[MS-CPREST-Diff]:

Control Plane REST API

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
10/16/2019	1.0	New	Released new document.
12/18/2019	2.0	Major	Significantly changed the technical content.
3/5/2020	3.0	Major	Significantly changed the technical content.
6/22/2020	4.0	Major	Significantly changed the technical content.
8/4/2020	5.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Control Plane REST API protocol specifies an HTTP-based web service API that deploys data services and applications into a managed cluster environment, and then communicates with its management service APIs to manage high-value data stored in relational databases that have been integrated with high-volume data resources within a dedicated cluster.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

Apache Hadoop: An open-source framework that provides distributed processing of large data sets across clusters of computers that use different programming paradigms and software libraries.

Apache Knox: A gateway system that provides secure access to data and processing resources in an Apache Hadoop cluster.

Apache Spark: A parallel processing framework that supports in-memory processing to boost the performance of big-data analytic applications.

Apache ZooKeeper: A service that is used to maintain synchronization in highly available systems.

application: A participant that is responsible for beginning, propagating, and completing an atomic transaction. An application communicates with a transaction manager in order to begin and complete transactions. An application communicates with a transaction manager in order to marshal transactions to and from other applications. An application also communicates in application-specific ways with a resource manager in order to submit requests for work on resources.

Basic: An authentication access type supported by HTTP as defined by [RFC2617].

Bearer: A type of token that provides an authentication access type supported by HTTP as defined by [RFC6750].

big data cluster: A grouping of high-value relational data with high-volume big data that provides the computational power of a cluster to increase scalability and performance of applications.

cluster: A group of computers that are able to dynamically assign resource tasks among nodes in a group.

container: A unit of software that isolates and packs an application and its dependencies into a single, portable unit.

control plane: A logical plane that provides management and security for a Kubernetes cluster. It contains the controller, management proxy, and other services that are used to monitor and maintain the cluster.

control plane service: The service that is deployed and hosted in the same Kubernetes namespace in which the user wants to build out a big data cluster. The service provides the core functionality for deploying and managing all interactions within a Kubernetes cluster.

controller: A replica set that is deployed in a big data cluster to manage the functions for deploying and managing all interactions within the control plane service.

create retrieve update delete (CRUD): The four basic functions of persistent storage. The "C" stands for create, the "R" for retrieve, the "U" for update, and the "D" for delete. CRUD is used to denote these conceptual actions and does not imply the associated meaning in a particular technology area (such as in databases, file systems, and so on) unless that associated meaning is explicitly stated.

distinguished name (DN): In the Active Directory directory service, the unique identifier of an object in Active Directory, as described in [MS-ADTS] and [RFC2251].

Docker: An open-source project for automating the deployment of applications as portable, self-sufficient containers that can run on the cloud or on-premises.

domain controller (DC): A server that controls all access in a security domain.

Domain Name System (DNS): A hierarchical, distributed database that contains mappings of domain names to various types of data, such as IP addresses. DNS enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

Hadoop Distributed File System (HDFS): A core component of Apache Hadoop, consisting of a distributed storage and file system that allows files of various formats to be stored across numerous machines or nodes.

JavaScript Object Notation (JSON): A text-based, data interchange format that is used to transmit structured data, typically in Asynchronous JavaScript + XML (AJAX) web applications, as described in [RFC7159]. The JSON format is based on the structure of ECMAScript (Jscript, JavaScript) objects.

JSON Web Token (JWT): A type of token that includes a set of claims encoded as a JSON object. For more information, see [RFC7519].

Kubernetes: An open-source container orchestrator that can scale container deployments according to need. Containers are the basic organizational units from which applications on Kubernetes run.

Kubernetes cluster: A set of computers in which each computer is called a node. A designated master node controls the cluster, and the remaining nodes in the cluster are the worker nodes. A Kubernetes cluster can contain a mixture of physical-machine and virtual-machine nodes.

Kubernetes namespace: Namespaces represent subdivisions within a cluster. A cluster can have multiple namespaces that act as their own independent virtual clusters.

management proxy: A pod that is deployed in the control plane to provide users with the ability to interact with deployed applications to manage the big data cluster.

master instance: A server instance that is running in a big data cluster. The master instance provides various kinds of functionality in the cluster, such as for connectivity, scale-out query management, and metadata and user databases.

NameNode: A central service in HDFS that manages the file system metadata and where clients request to perform operations on files stored in the file system.

node: A single physical or virtual computer that is configured as a member of a cluster. The node has the necessary software installed and configured to run containerized applications.

persistent volume: A volume that can be mounted to Kubernetes to provide continuous and unrelenting storage to a cluster.

pod: A unit of deployment in a Kubernetes cluster that consists of a logical group of one or more containers and their associated resources. A pod is deployed as a functional unit in and represents a process that is running on a Kubernetes cluster.

pool: A logical grouping of pods that serve a similar function in a big data cluster deployment.

replica set: A group of pods that mirror each other in order to maintain a stable set of data that runs at any given time across one or more nodes.

storage class: A definition that specifies how storage volumes that are used for persistent storage are to be configured.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

universally unique identifier (UUID): A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and RPC objects. UUIDs are highly likely to be unique. UUIDs are also known as globally unique identifiers (GUIDs) and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the UUID.

YAML Ain't Markup Language (YAML): A Unicode-based data serialization language that is designed around the common native data types of agile programming languages. YAML v1.2 is a superset of JSON.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[ApacheHadoop] Apache Software Foundation, "Apache Hadoop", https://hadoop.apache.org/

[ApacheKnox] Apache Software Foundation, "Apache Knox", https://knox.apache.org/

[ApacheSpark] Apache Software Foundation, "Apache Spark", https://spark.apache.org/

[ApacheZooKeeper] Apache Software Foundation, "Welcome to Apache ZooKeeper", https://zookeeper.apache.org/

[JSON-Schema] Internet Engineering Task Force (IETF), "JSON Schema and Hyper-Schema", http://json-schema.org/

[Kubernetes] The Kubernetes Authors, "Kubernetes Documentation", https://kubernetes.io/docs/home/

[MS-TDS] Microsoft Corporation, "Tabular Data Stream Protocol".

[REST] Fielding, R., "Architectural Styles and the Design of Network-based Software Architectures", 2000, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, http://www.rfc-editor.org/rfc/rfc3986.txt

[RFC4120] Neuman, C., Yu, T., Hartman, S., and Raeburn, K., "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005, https://www.rfc-editor.org/rfc/rfc4120.txt

[RFC4559] Jaganathan, K., Zhu, L., and Brezak, J., "SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows", RFC 4559, June 2006, http://www.rfc-editor.org/rfc/rfc4559.txt

[RFC7230] Fielding, R., and Reschke, J., Eds., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014, http://www.rfc-editor.org/rfc/rfc7230.txt

[RFC7231] Fielding, R., and Reschke, J., Eds., "Hypertext Transfer Protocol -- HTTP/1.1: Semantics and Content", RFC7231, June 2014, http://www.rfc-editor.org/rfc/rfc7231.txt

[RFC7519] Internet Engineering Task Force, "JSON Web Token (JWT)", http://www.rfc-editor.org/rfc/rfc7519.txt

[RFC791] Postel, J., Ed., "Internet Protocol: DARPA Internet Program Protocol Specification", RFC 791, September 1981, http://www.rfc-editor.org/rfc/rfc791.txt

[RFC793] Postel, J., Ed., "Transmission Control Protocol: DARPA Internet Program Protocol Specification", RFC 793, September 1981, http://www.rfc-editor.org/rfc/rfc793.txt

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 8259, December 2017, https://www.rfc-editor.org/rfc/rfc8259.txt

[Swagger2.0] SmartBear Software, "What Is Swagger?", OpenAPI Specification (fka Swagger), version 2.0, https://swagger.io/docs/specification/2-0/what-is-swagger/

[YAML1.2] Ben-Kiki, O., Evans, C., and dot NET, I., "YAML Ain't Markup Language (YAML) Version 1.2", 3rd edition, October 2009, https://yaml.org/spec/1.2/spec.html

1.2.2 Informative References

[JSON-RegEx] Droettboom, M., "Understanding JSON Schema: Regular Expressions", https://json-schema.org/understanding-json-schema/reference/regular_expressions.html

[MSDOCS-ConfigBDC] Microsoft Corporation, "Configure Apache Spark and Apache Hadoop in Big Data Clusters", https://docs.microsoft.com/en-us/sql/big-data-cluster/configure-spark-hdfs?view=sql-server-ver15

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.txt

1.3 (Updated Section) Overview

The Control Plane REST API protocol specifies a protocol to communicate with the control plane. The control plane acts as an abstraction layer in which users can create and manage big data clusters inside a Kubernetes namespace [Kubernetes] without communicating directly with the Kubernetes cluster or the services and tools deployed in it. It provides convenient APIs to allow the user to manage the lifecycle of resources deployed in the cluster.

All client and server communications are formatted in JavaScript Object Notation (JSON), as specified in [RFC8259].

The protocol uses RESTful web service APIs that allow users to do the following:

- Create a Kubernetes cluster in which to manage, manipulate, and monitor a big data cluster.
- Manage the lifecycle of a big data cluster, including authentication and security.
- Manage the lifecycle of machine learning applications and other resources that are deployed in the cluster.
- Manage the lifecycle of Hadoop Distributed File System (HDFS) mounts mounted remotely.
- Use monitoring tools deployed in the Kubernetes cluster to observe or report the status of the big data cluster.

The control plane consists of a controller replica set, a management proxy, and various pods that provide log and metrics collection for pods in the cluster.

This protocol defines the deployment of a ~~Big Data Cluster~~big data cluster with the most basic topography and the default configurations for the resources in the initial cluster manifest (see section 3.1.5.1.1). Depending on the configuration sent to the Control Plane REST API, the Apache Spark [ApacheSpark] and Apache Hadoop [ApacheHadoop] resources in the cluster can be customized by including additional settings in the initial cluster manifest. All configurations are based on the configuration settings defined in the documentation for each component. Supported and unsupported configurations are listed in [MSDOCS-ConfigBDC].

This protocol can be authenticated by using either Basic authentication or token authentication. Additionally, if the Control Plane is deployed with Active Directory configured, Active Directory can be used to retrieve a JSON Web Token (JWT) token which can then be used to authenticate the Control Plane REST APIs.

All requests are initiated by the client, and the server responds in JSON format, as illustrated in the following diagram.

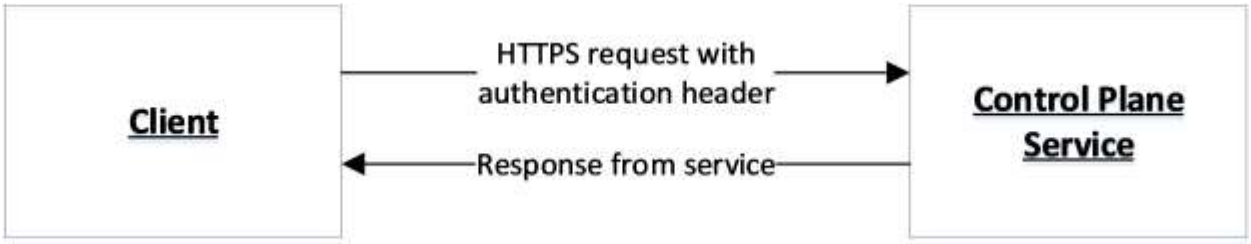

Figure 1: Communication flow

1.4 Relationship to Other Protocols

The Control Plane REST API protocol transmits messages by using HTTPS [RFC7230] [RFC2818] over TCP [RFC793].

The following diagram shows the protocol layering.

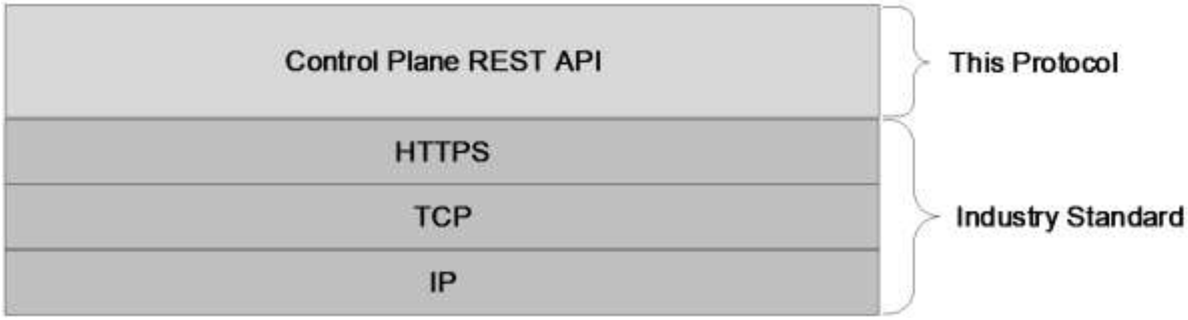

Figure 2: Protocol layering

1.5 Prerequisites/Preconditions

A controller and controller database has to be deployed in the Kubernetes cluster before the Control Plane REST API can be used. The controller is deployed by using Kubernetes APIs [Kubernetes].

1.6 Applicability Statement

This protocol supports exchanging messages between a client and the control plane service.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The Control Plane REST API protocol consists of a set of RESTful [REST] web services APIs, and client messages MUST use HTTPS over TCP/IP, as specified in [RFC793] [RFC7230] [RFC7231].

The management service is granted permission by the cluster administrator to manage all resources within the cluster, including but not limited to authentication. Implementers can configure their servers to use standard authentication, such as HTTP Basic and token authentication.

This protocol does not require any specific HTTP ports, character sets, or transfer encoding.

2.2 Common Data Types

2.2.1 Namespaces

None.

2.2.2 HTTP Methods

This protocol uses HTTP methods GET, POST, PATCH, and DELETE.

2.2.3 (Updated Section) HTTP Headers

This protocol defines the following common HTTP headers in addition to the existing set of standard HTTP headers.

HTTP headers	Description
X-RequestID	An optional universal unique identifier (UUID) that can be included to help map a request through the control plane service.

2.2.3.1 X-RequestID

A request to the control plane service can include an **X-RequestID** header that is included in all subsequent calls within the control plane service. This header can help with following a request through the control plane service logs.

2.2.4 URI Parameters

Every resource that supports CRUD operations uses common JSON elements [JSON-Schema] in any request or response.

This protocol defines the following common URI parameters [RFC3986].

URI parameters	Description
clusterIp	The IP address of a connectable node in the cluster.
controllerPort	A port that is defined by the user during control plane creation and exposed on the cluster for the controller.

URI parameters	Description
bdcName	The name of the big data cluster that is being manipulated.

2.2.4.1 clusterIp

The *clusterIp* parameter contains the IP address of a node in the cluster that is accessible to the user. This is often the same address that tools, such as the kubectl tool that manages the Kubernetes cluster, use to connect to the cluster.

2.2.4.2 controllerPort

The *controllerPort* parameter is defined in the controller. The value of this parameter is specified before controller deployment.

2.2.4.3 bdcName

The *bdcName* parameter provides the name of the deployed big data cluster. The *bdcName* parameter matches the Kubernetes cluster into which the big data cluster is deployed.

2.2.5 (Updated Section) JSON Elements

Every resource type that supports this protocol supports CRUD operations and uses common JSON elements [JSON-Schema] in any request or response.

The element names that are defined, implicitly or explicitly, in the schemas that are defined in this section can be regular expressions [JSON-RegEx]. For example, "([A-Za-z_.-])+" is a regular expression that matches any string that uses any combination of alphabetical characters, "_"s, "."s, and ~~"."~~"-"s. When used to define an array, the element name "*" is a placeholder for the individual elements within the array that do not have names.

The following table summarizes the set of common resource types that are consumed or produced by this protocol.

Structure	Section	Description
Big Data Cluster	2.2.5.1	Defines the big data cluster that is deployed.
Storage	2.2.5.2	Defines the persistent storage that is deployed.
App	2.2.5.3	Defines the application scripts that are deployed.
Token	2.2.5.4	Defines the response if token type authentication is used.
Home	2.2.5.5	Defines the responses that are used to determine whether the control plane service is available.

2.2.5.1 (Updated Section) Big Data Cluster

The **Big Data Cluster** resource type defines the big data cluster resources that are created and managed by the protocol that is defined by this specification inside a Kubernetes namespace [Kubernetes].

This protocol defines the following set of **Big Data Cluster** JSON schemas.

Schema	Section	Description
Big Data Cluster Spec Schema	2.2.5.1.1	Defines the big data cluster that is deployed.
Big Data Cluster Error Response Schema	2.2.5.1.2	Describes the current errors.
Big Data Cluster Information Schema	2.2.5.1.3	Provides information about the big data cluster.
Big Data Cluster Status Schema	2.2.5.1.4	Retrieves the status of the big data cluster.
Big Data Cluster Service Status Schema	2.2.5.1.5	Retrieves the status of a service.
Big Data Cluster Service Resource Status Schema	2.2.5.1.6	Retrieves the requested status of a resource in a service.
Big Data Cluster Upgrade Schema	2.2.5.1.7	Upgrades the images in the big data cluster and control plane.
Big Data Cluster Endpoints List Schema	2.2.5.1.8	Retrieves ~~an~~a list of endpoints that are exposed for a component.
Big Data Cluster Endpoint Schema	2.2.5.1.9	Exposes an endpoint outside the big data cluster.

2.2.5.1.1 Big Data Cluster Spec Schema

The **Big Data Cluster Spec** schema defines the big data cluster JSON object that is deployed inside a Kubernetes namespace [Kubernetes].

The JSON of this schema is as follows.

```
"$schema": "http://json-schema.org/draft-07/schema#",
"$id": "http://example.com/root.json",
"type": "object",
"definitions": {
  "comment": "section 2.2.5.1.1.1"
},
"required": [
  "apiVersion",
  "metadata",
  "spec"
],
"properties": {
  "apiVersion": {
    "comment": "section 2.2.5.1.1.2"
  },
  "metadata": {
    "$ref": "#/definitions/metadata"
  },
  "spec": {
    "comment": "section 2.2.5.1.1.3"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
definitions	object	Section 2.2.5.1.1.1
apiVersion	object	Section 2.2.5.1.1.2
metadata	object	Section 2.2.5.1.1.1.9
spec	object	Section 2.2.5.1.1.3

2.2.5.1.1.1 definitions

The **definitions** element specifies the resources that are deployed in the big data cluster.

The JSON of this element is as follows.

```
"definitions": {
  "storage": {
    "comment": "section 2.2.5.1.1.1.1"
  },
  "storageInfo": {
    "comment": "section 2.2.5.1.1.1.2"
  },
  "docker": {
    "comment": "section 2.2.5.1.1.1.3"
  },
  "security": {
    "comment": "section 2.2.5.1.1.1.4"
  },
  "spark": {
    "comment": "section 2.2.5.1.1.1.5"
  },
  "hdfs": {
    "comment": "section 2.2.5.1.1.1.6"
  },
  "sql": {
    "comment": "section 2.2.5.1.1.1.7"
  },
  "gateway": {
    "comment": "section 2.2.5.1.1.1.8"
  },
  "metadata": {
    "comment": "section 2.2.5.1.1.1.9"
  },
  "replicas": {
    "type": "integer"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
storage	object	Section 2.2.5.1.1.1.1
storageInfo	object	Section 2.2.5.1.1.1.2
docker	object	Section 2.2.5.1.1.1.3
security	object	Section 2.2.5.1.1.1.4

Element name	Type	Description
spark	object	Section 2.2.5.1.1.1.5
hdfs	object	Section 2.2.5.1.1.1.6
sql	object	Section 2.2.5.1.1.1.7
gateway	object	Section 2.2.5.1.1.1.8
metadata	object	Section 2.2.5.1.1.1.9
replicas	integer	Specifies the number of pods that are deployed in the big data cluster.

2.2.5.1.1.1.1 storage

The **storage** element specifies the structured data that define the Kubernetes persistent storage that is used in the cluster.

The JSON of this element is as follows.

```
"storage": {
  "required": [
    "logs",
    "data"
  ],
  "properties": {
    "data": {
      "$ref": "#/definitions/storageInfo"
    },
    "logs": {
      "$ref": "#/definitions/storageInfo"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
logs	object	Section 2.2.5.1.1.1.2
data	object	Section 2.2.5.1.1.1.2

2.2.5.1.1.1.2 storageInfo

The **storageInfo** element specifies the structured data that are used to create persistent volumes in Kubernetes [Kubernetes].

The JSON of this element is as follows.

```
"storageInfo": {
  "required": [
    "className",
    "accessMode",
    "size"
  ],
  "properties": {
    "className": {
      "type": "string"
    },
    "accessMode": {
      "comment": "section 2.2.5.1.1.1.2.1"
    },
    "size": {
      "type": "string",
      "example": "10Gi"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
className	string	Specifies the name of the Kubernetes storage class that is used to create persistent volumes.
accessMode	object	Section 2.2.5.1.1.1.2.1
size	string	Specifies the size of a persistent volume.

2.2.5.1.1.1.2.1 (Updated Section) accessMode

The **accessMode** element specifies the way in which a user or group of users can interact with data in the persistent volumes.

The JSON of this element is as follows.

```
"accessMode": {
  "enum": [
    "ReadWriteOnce",
    "ReadOnlyMany",
    "ReadWriteMany"
  ]
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
ReadWriteOnce	string	Specifies that read-write access is allowed ~~only one time~~to a single node in the cluster.
ReadOnlyMany	string	Specifies that read-only access is allowed to multiple nodes in the cluster.
ReadWriteMany	string	Specifies that read-write access is allowed to multiple nodes in the cluster.

2.2.5.1.1.1.3 docker

The **docker** element specifies the structured data that define where to find the Docker image.

```
"docker": {
  "required": [
    "registry",
    "repository",
    "imageTag",
    "imagePullPolicy"
  ],
  "properties": {
    "registry": {
      "type": "string",
      "example": "repo.contoso.com"
    },
    "repository": {
      "type": "string"
    },
    "imageTag": {
      "type": "string",
      "example": "latest"
    },
    "imagePullPolicy": {
      "comment": "section 2.2.5.1.1.1.3.1"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
registry	string	Specifies the registry in which the Docker image is located.
repository	string	Specifies the repository in which the Docker image is located.
imageTag	string	Specifies the full image name of the Docker image to be pulled.
imagePullPolicy	object	Section 2.2.5.1.1.1.3.1

2.2.5.1.1.1.3.1 imagePullPolicy

The **imagePullPolicy** element specifies the policy that pulls the Docker image into the cluster.

The JSON of this element is as follows.

```
"imagePullPolicy": {
  "enum": [
    "Always",
    "IfNotPresent"
  ]
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
Always	string	Specifies that the Docker image is always pulled.
IfNotPresent	string	Specifies that the Docker image pull is skipped if it already exists locally on the node.

2.2.5.1.1.1.4 (Updated Section) security

The **security** element specifies the collection of data that define the security settings for the cluster.

The JSON of this element is as follows.

```
"security": {
  "type": "object",
  "required": [
    "activeDirectory",
    "allowRunAsRoot",
    ~~“allowNodeMetrics”,~~"allowNodeMetricsCollection",
    ~~“allowPodMetrics”~~"allowPodMetricsCollection",
    "automountServiceAccountToken"
  ],
  "properties": {
    "activeDirectory": {
      "comment": "section 2.2.5.1.1.1.4.1"
    },
    "allowRunAsRoot": {
      "$id": "#/properties/allowRunAsRoot",
      "type": "boolean"
    },
    "allowDumps": {
      "$id": "#/properties/allowDumps",
      "type": "boolean"
    },
    "~~allowNodeMetrics~~allowNodeMetricsCollection": {
      "$id": "#/properties/~~allowNodeMetrics~~allowNodeMetricsCollection",
      "type": "boolean"
    },
    "~~allowPodMetrics~~allowPodMetricsCollection": {
      "$id": "#/properties/~~allowPodMetrics~~allowPodMetricsCollection",
      "type": "boolean"
    },
    "automountServiceAccountToken": {
      "$id": "#/properties/automountServiceAccountToken",
      "type": "boolean"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
activeDirectory	object	Section 2.2.5.1.1.1.4.1
allowRunAsRoot	boolean	SHOULD<1> specify a Boolean that indicates that the containers in the cluster are run as privileged users.
allowDumps	boolean	SHOULD<2> specify a Boolean that indicates that the containers in the big data cluster can

Element name	Type	Description
		produce dump files that can be used for debugging.
~~allowNodeMetrics~~allowNodeMetricsCollection	boolean	SHOULD<3> specify a Boolean that indicates that metrics for the nodes in big data cluster can be collected.
~~allowPodMetrics~~allowPodMetricsCollection	boolean	SHOULD<4> specify a Boolean that indicates that metrics for the pods running in the big data cluster can be collected.
automountServiceAccountToken	boolean	SHOULD<5> specify a Boolean that indicates that the default Kubernetes [Kubernetes] service account is mounted on each pod in the big data cluster.

2.2.5.1.1.1.4.1 (Updated Section) activeDirectory

The **activeDirectory** element SHOULD<6> specify the Active Directory settings for the cluster.

The JSON of this element is as follows.

```
"activeDirectory": {
  "$id": "#/properties/activeDirectory",
  "type": "object",
  "required": [
    "useInternalDomain",
    "useSubdomain",
    "ouDistinguishedName",
    "dnsIpAddresses",
    "domainControllerFullyQualifiedDns",
    "domainDnsName",
    "clusterAdmins",
    "clusterUsers"
  ],
  "properties": {
    "useInternalDomain": {
      "$id": "#/properties/activeDirectory/properties/useInternalDomain",
      "type": "boolean"
    },
    "useSubdomain": {
      "$id": "#/properties/activeDirectory/properties/useSubdomain",
      "type": "boolean"
    },
    "ouDistinguishedName": {
      "$id": "#/properties/activeDirectory/properties/ouDistinguishedName",
      "type": "string"
    },
    "dnsIpAddresses": {
      "comment": "section 2.2.5.1.1.1.4.1.1"
    },
    "domainControllerFullyQualifiedDns": {
      "comment": "section 2.2.5.1.1.1.4.1.2"
    },
    "subdomain": {
      "$id": "#/properties/activeDirectory/properties/subdomain",
      "type": "string"
    },
    "accountPrefix": {
      "$id": "#/properties/activeDirectory/properties/accountPrefix",
      "type": "string"
    },
    "realm": {
      "$id": "#/properties/activeDirectory/properties/realm",
      "type": "string"
    },
    "domainDnsName": {
      "$id": "#/properties/activeDirectory/properties/domainDnsName",
      "type": "string"
    },
    "clusterAdmins": {
      "comment": "section 2.2.5.1.1.1.4.1.3"
    },
    "clusterUsers": {
      "comment": "section 2.2.5.1.1.1.4.1.4"
    },
    "appOwners": {
      "comment": "section 2.2.5.1.1.1.4.1.5"
    },
    "appUsers": {
      "comment": "section 2.2.5.1.1.1.4.1.6"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
useInternalDomain	boolean	SHOULD<7> specify a Boolean that indicates whether to use a domain that is hosted within the cluster. This element ~~should be~~is set to "false" for production deployments.
useSubdomain	boolean	SHOULD<8> specify a Boolean that indicates whether a Domain Name System (DNS) subdomain is used for the big data cluster.
ouDistinguishedName	string	SHOULD<9> specify the distinguished name (DN) of an organizational unit to which all Active Directory accounts created by the cluster deployment are added. For example, if the domain is named ~~“~~"contoso.local~~”,~~", the value of this field is ~~“~~"OU=BDC,DC=contoso,DC=local~~”.~~".
dnsIpAddresses	array	Section 2.2.5.1.1.1.4.1.1
domainControllerFullyQualifiedDns	array	Section 2.2.5.1.1.1.4.1.2
subdomain	string	SHOULD<10> specify a unique DNS subdomain to use for the cluster.
accountPrefix	string	SHOULD<11> specify a unique prefix to use for all Active Directory accounts that the big data cluster generates.
realm	string	SHOULD<12> specify the scope in which the Active Directory server has the authority to authenticate activity.
domainDnsName	string	SHOULD<13> specify the name of the Active Directory domain.

Element name	Type	Description
clusterAdmins	array	Section 2.2.5.1.1.1.4.1.3
clusterUsers	array	Section 2.2.5.1.1.1.4.1.4
appOwners	array	Section 2.2.5.1.1.1.4.1.5
appUsers	array	Section 2.2.5.1.1.1.4.1.6

2.2.5.1.1.1.4.1.1 dnsIpAddresses

The **dnsIpAddresses** element SHOULD<14> specify a list of Domain Name System (DNS) IP addresses of domain controllers (DCs).

The element name "*" is a placeholder for the individual elements within the array that do not have names.

The JSON of this element is as follows.

```
"dnsIpAddresses": {
  "$id": "#/properties/activeDirectory/properties/dnsIpAddresses",
  "type": "array",
  "items": {
    "$id": "#/properties/activeDirectory/properties/dnsIpAddresses/items",
    "type": "string"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
*	string	Specifies a valid IPv4 IP address that conforms to [RFC791].

2.2.5.1.1.1.4.1.2 domainControllerFullyQualifiedDns

The **domainControllerFullyQualifiedDns** element SHOULD<15> specify a list of fully qualified domain names of DCs.

The element name "*" is a placeholder for the individual elements within the array that do not have names.

The JSON of this element is as follows.

```
      "domainControllerFullyQualifiedDns": {
        "$id": "#/properties/activeDirectory/properties/domainControllerFullyQualifiedDns",
        "type": "array",
        "items": {
          "$id":
    "#/properties/activeDirectory/properties/domainControllerFullyQualifiedDns/items",
          "type": "string"
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
*	string	Specifies an element that has a name that is defined by a regular expression that matches any string.

2.2.5.1.1.1.4.1.3 clusterAdmins

The **clusterAdmins** element SHOULD<16> specify an array of the names of the Active Directory group that is assigned administrator permissions. This element can contain only one Active Directory group name.

The element name "*" is a placeholder for the individual elements within the array that do not have names.

The JSON of this element is as follows.

```
"clusterAdmins": {
  "$id": "#/properties/activeDirectory/properties/clusterAdmins",
  "type": "array",
  "items": {
    "$id": "#/properties/activeDirectory/properties/clusterAdmins/items",
    "type": "string"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
*	string	Specifies an element that has a name that is defined by a regular expression that matches any string. For further details, see [RFC4120].

2.2.5.1.1.1.4.1.4 clusterUsers

The **clusterUsers** element SHOULD<17> specify an array of the names of the Active Directory groups that have regular permissions in the cluster.

The element name "*" is a placeholder for the individual elements within the array that do not have names.

The JSON of this element is as follows.

```
"clusterUsers": {
  "$id": "#/properties/activeDirectory/properties/clusterUsers",
  "type": "array",
  "items": {
    "$id": "#/properties/activeDirectory/properties/clusterUsers/items",
    "type": "string"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
*	string	Specifies an element that has a name that is defined by a regular expression that matches any string. For further details, see [RFC4120].

2.2.5.1.1.1.4.1.5 appOwners

The **appOwners** element SHOULD<18> specify an array of the names of the Active Directory groups that have permissions to create, delete, and run applications in the cluster.

The element name "*" is a placeholder for the individual elements within the array that do not have names.

The JSON of this element is as follows.

```
"appOwners": {
  "$id": "#/properties/activeDirectory/properties/appOwners",
  "type": "array",
  "items": {
    "$id": "#/properties/activeDirectory/properties/appOwners/items",
    "type": "string"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
*	string	Specifies an element that has a name that is defined by a regular expression that matches any string. For further details, see [RFC4120].

2.2.5.1.1.1.4.1.6 appUsers

The **appUsers** element SHOULD<19> specify an array of the names of the Active Directory groups that have permissions to run applications in the cluster.

The element name "*" is a placeholder for the individual elements within the array that do not have names.

The JSON of this element is as follows.

```
"appUsers": {
  "$id": "#/properties/activeDirectory/properties/appUsers",
  "type": "array",
  "items": {
    "$id": "#/properties/activeDirectory/properties/appUsers/items",
    "type": "string"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
*	string	Specifies an element that has a name that is defined by a regular expression that matches any string. For further details, see [RFC4120].

2.2.5.1.1.1.5 (Updated Section) spark

The **spark** element specifies the structured data that define the settings for Apache Spark [ApacheSpark] in the cluster.

The JSON of this element is as follows.

```
"spark": {
  "patternProperties": {
    ~~"patternProperties": {~~
~~      "([~~"^([A-Za-z~~_.-])+":~~_.-])+$": {
      "type": "string"
    }
  ~~}~~},
~~    }~~
  "additionalProperties": false
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
~~([~~^([A-Za-z~~_.-])+~~_.-])+$	string	Specifies the structured data that define the settings for Apache Spark in the cluster.

2.2.5.1.1.1.6 (Updated Section) hdfs

The **hdfs** element specifies the structured data that define the settings for the HDFS [ApacheHadoop] resource in the cluster.

The JSON of this element is as follows.

```
"hdfs": {
  "patternProperties": {
    ~~"patternProperties": {~~
~~      "([~~"^([A-Za-z~~_.-])+":~~_.-])+$": {
      "type": "string"
    }
  ~~}~~},
~~    }~~
  "additionalProperties": false
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
~~([~~^([A-Za-z~~_.-])+~~_.-])+$	string	Specifies the structured data that define the settings for the **HDFS** resource in the cluster. For

Element name	Type	Description
		further information about the application of settings for the **hdfs** element, see [MSDOCS-ConfigBDC].

2.2.5.1.1.1.7 (Updated Section) sql

The **sql** element specifies the structured data that define the settings for the **SQL Server** resource in the cluster.

The **sql** element MUST use the **hadr.enabled** instance of the element to enable high availability for the **master** SQL Server instances.

The JSON of this element is as follows.

```
  "sql": {
    "~~properties~~patternProperties": {
      ~~"([~~"^([A-Za-z~~_.-])+":~~_.-])+$": {
        "type": "string"
      }
    ~~}~~},
    "additionalProperties": false
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
~~([~~^([A-Za-z~~_.-])+~~_.-])+$	string	Specifies the structured data that define the settings for the **SQL Server** resource in the cluster.

2.2.5.1.1.1.8 (Updated Section) gateway

The **gateway** element specifies the structured data that define the settings for the **gateway** resource in the cluster.

The JSON of this element is as follows.

```
  "gateway": {
    "patternProperties": {
      ~~"patternProperties": {~~
~~        "([~~"^([A-Za-z~~_.-])+":~~_.-])+$": {
        "type": "string"
      }
    ~~  }~~},
~~    }~~
    "additionalProperties": false
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
~~([~~^([A-Za-z~~_.-])+~~_.-])+$	string	Specifies the structured data that define the settings for the **gateway** resource in the cluster.

2.2.5.1.1.1.9 (Updated Section) metadata

The **metadata** element specifies the structured data that provide information about a specified JSON object in the cluster.

The JSON of this element is as follows.

```
"metadata": {
  "required": [
    "kind",
    "name"
  ],
  "properties": {
    ~~"([A-Za-z_.-])+":~~"kind": {
      "type": "string"
    ~~}~~},
  ~~}~~  "name": {
      "type": "string"
  },
  "additionalProperties": false
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
kind	string	Specifies the type of the JSON object.
name	string	Specifies the name of the JSON object.
~~([A-Za-z_.-])+~~	~~string~~	~~Specifies the structured data that provide information about a specified JSON object in the cluster.~~

2.2.5.1.1.2 apiVersion

The **apiVersion** element specifies the Kubernetes [Kubernetes] API version that is used in the cluster.

The JSON of this element is as follows.

```
"apiVersion": {
  "$id": "#/properties/apiVersion",
  "const": "v1"
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
v1	string	Specifies that the value of the API version MUST be "v1".

2.2.5.1.1.3 spec

The **spec** element specifies the structured data that define the **resources** and **services** that are deployed in the big data cluster.

The JSON of this element is as follows.

```
"spec": {
  "$id": "#/properties/spec",
  "type": "object",
  "required": [
    "resources",
    "services"
  ],
  "properties": {
    "security": {
      "$ref": "#/definitions/security"
    },
    "resources": {
      "comment": "section 2.2.5.1.1.3.1"
    },
    "services": {
      "comment": "section 2.2.5.1.1.3.2"
    },
    "docker": {
      "$ref": "#/definitions/docker"
    },
    "storage": {
      "$ref": "#/definitions/storage"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
resources	object	Section 2.2.5.1.1.3.1
services	object	Section 2.2.5.1.1.3.2
security	object	Section 2.2.5.1.1.1.4
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1

2.2.5.1.1.3.1 resources

The **resources** element of the **spec** element specifies the structured data for the resources that are deployed in the big data cluster.

The JSON of this element is as follows.

```
      "resources": {
        "$id": "#/properties/spec/properties/resources",
        "type": "object",
        "required": [
          "sparkhead",
          "storage",
          "nmnode",
          "master",
          "compute",
          "appproxy",
          "zookeeper",
          "gateway",
          "data"
        ],
        "properties": {
          "sparkhead": {
            "comment": "section 2.2.5.1.1.3.1.1"
          },
          "storage": {
            "comment": "section 2.2.5.1.1.3.1.2"
          },
          "nmnode": {
            "comment": "section 2.2.5.1.1.3.1.3"
          },
          "master": {
            "comment": "section 2.2.5.1.1.3.1.4"
          },
          "compute": {
            "comment": "section 2.2.5.1.1.3.1.5"
          },
          "appproxy": {
            "comment": "section 2.2.5.1.1.3.1.6"
          },
          "zookeeper": {
            "comment": "section 2.2.5.1.1.3.1.7"
          },
          "gateway": {
            "comment": "section 2.2.5.1.1.3.1.8"
          },
          "spark": {
            "comment": "section 2.2.5.1.1.3.1.9"
          },
          "data": {
            "comment": "section 2.2.5.1.1.3.1.10"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
sparkhead	object	Section 2.2.5.1.1.3.1.1
storage	object	Section 2.2.5.1.1.3.1.2
nmnode	object	Section 2.2.5.1.1.3.1.3
master	object	Section 2.2.5.1.1.3.1.4
compute	object	Section 2.2.5.1.1.3.1.5
appproxy	object	Section 2.2.5.1.1.3.1.6
zookeeper	object	Section 2.2.5.1.1.3.1.7

Element name	Type	Description
gateway	object	Section 2.2.5.1.1.3.1.8
spark	object	Section 2.2.5.1.1.3.1.9
data	object	Section 2.2.5.1.1.3.1.10

2.2.5.1.1.3.1.1 sparkhead

The **sparkhead** resource provides all the management services components that maintain the Apache Spark instances [ApacheSpark] in the big data cluster.

The JSON of this resource is as follows.

```
      "sparkhead": {
        "$id": "#/properties/spec/properties/resources/properties/sparkhead",
        "type": "object",
        "required": [
          "spec",
        ],
        "properties": {
          "clusterName": {
            "$id":
    "#/properties/spec/properties/resources/properties/sparkhead/properties/clusterName",
            "type": "string"
          },
          "security": {
            "$ref": "#/definitions/security"
          },
          "spec": {
            "comment": "section 2.2.5.1.1.3.1.1.1"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
clusterName	string	Specifies the name of the big data cluster into which **sparkhead** resources are deployed.
security	object	Section 2.2.5.1.1.1.4
spec	object	Section 2.2.5.1.1.3.1.1.1

2.2.5.1.1.3.1.1.1 spec

The **spec** element of the **sparkhead** resource specifies the structured data that define the **sparkhead** resources that are deployed in the cluster.

The JSON of this element is as follows.

```
      "spec": {
        "$id": "#/properties/spec/properties/resources/properties/sparkhead/properties/spec",
        "type": "object",
        "required": [
          "replicas",
        ],
        "properties": {
          "replicas": {
            "$id":
    "#/properties/spec/properties/resources/properties/sparkhead/properties/spec/properties/repli
    cas",
            "type": "integer"
          },
          "docker": {
            "$ref": "#/definitions/docker"
          },
          "storage": {
            "$ref": "#/definitions/storage"
          },
          "settings": {
            "comment": "section 2.2.5.1.1.3.1.1.2"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
replicas	integer	Specifies the number of pods that are deployed for the **sparkhead** resource.
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1
settings	object	Section 2.2.5.1.1.3.1.1.2

2.2.5.1.1.3.1.1.2 spec.settings

The **spec.settings** element of the **sparkhead** resource specifies the settings of a **sparkhead** resource.

The JSON of this element is as follows.

```
    "settings": {
      "$id":
"#/properties/spec/properties/resources/properties/sparkhead/properties/spec/properties/setti
ngs",
      "type": "object",
      "required": [
        "spark",
      ],
      "properties": {
        "spark": {
          "$ref": "#/definitions/spark"
        }
      }
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
spark	object	Section 2.2.5.1.1.1.5

2.2.5.1.1.3.1.2 storage

The **storage** resource provides of a group of disks that are aggregated and managed as a single unit and used to ingest and store data from HDFS [ApacheHadoop].

If multiple storage pools are deployed, a "-#" suffix is appended to the resource name to denote ordinality, for example, "storage-0". This suffix is a positive integer that can range from 0 to n-1, where n is the number of storage pools that are deployed.

The JSON of this resource is as follows.

```
      "storage": {
        "$id": "#/properties/spec/properties/resources/properties/storage",
        "type": "object",
        "required": [
          "metadata",
          "spec"
        ],
        "properties": {
          "clusterName": {
            "$id": 
    "#/properties/spec/properties/resources/properties/storage/properties/clusterName",
            "type": "string"
          },
          "metadata": {
            "$ref": "#/definitions/metadata"
          },
          "security": {
            "$ref": "#/definitions/security"
          },
          "spec": {
            "comment": "section 2.2.5.1.1.3.1.2.1"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
metadata	object	Section 2.2.5.1.1.1.9
spec	object	Section 2.2.5.1.1.3.1.2.1
clusterName	string	Specifies the name of the big data cluster in which the **storage** resources are deployed.
security	object	Section 2.2.5.1.1.1.4

2.2.5.1.1.3.1.2.1 (Updated Section) spec

The **spec** element of the **storage** resource specifies the structured data that define the **storage** resources deployed in the big data cluster.

The JSON of this element is as follows.

```
      "spec": {
        "$id": "#/properties/spec/properties/resources/properties/storage/properties/spec",
        "type": "object",
        "required": [
          "type",
          "replicas",
          "settings"
        ],
        "properties": {
          "type": {
            "$id":
    "#/properties/spec/properties/resources/properties/storage/properties/spec/properties/type",
            "type": "~~integer~~string"
          },
          "replicas": {
            "$id":
    "#/properties/spec/properties/resources/properties/storage/properties/spec/properties/replica
    s",
            "type": "integer"
          },
          "docker": {
            "$ref": "#/definitions/docker"
          },
          "storage": {
            "$ref": "#/definitions/storage"
          },
          "settings": {
            "comment": "section 2.2.5.1.1.3.1.2.2"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
type	~~integer~~string	Specifies the value that is used to define the type of the resource. The value of the **storage** resource MUST be ~~the following:~~ ~~4 – storage pool~~"Storage".
replicas	integer	Specifies the number of pods that are deployed for use by the **storage** resource.
settings	object	Section 2.2.5.1.1.3.1.2.2
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1

2.2.5.1.1.3.1.2.2 spec.settings

The **spec.settings** element of the **storage** resource specifies the settings of a **storage** resource.

The JSON of this element is as follows.

```
      "settings": {
```

```
    "$id":
"#/properties/spec/properties/resources/properties/storage/properties/spec/properties/setting
s",
    "type": "object",
    "required": [
      "spark",
      "sql",
      "hdfs"
    ],
    "properties": {
      "spark": {
        "$ref": "#/definitions/spark"
      },
      "sql": {
        "$ref": "#/definitions/sql"
      },
      "hdfs": {
        "$ref": "#/definitions/hdfs"
      }
    }
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
spark	object	Section 2.2.5.1.1.1.5
sql	object	Section 2.2.5.1.1.1.7
hdfs	object	Section 2.2.5.1.1.1.6

2.2.5.1.1.3.1.3 nmnode

The **nmnode** resource provides the NameNode central service in HDFS [ApacheHadoop].

If multiple NameNode pools are deployed, a "-#" suffix is appended to the resource name to denote ordinality, for example, "nmnode-0". This suffix is a positive integer that can range from 0 to n-1, where n is the number of NameNodes that are deployed.

The JSON of this resource is as follows.

```
  "nmnode": {
    "$id": "#/properties/spec/properties/resources/properties/nmnode",
    "type": "object",
    "required": [
      "spec",
    ],
    "properties": {
      "clusterName": {
        "$id":
"#/properties/spec/properties/resources/properties/nmnode/properties/clusterName",
        "type": "string"
      },
      "security": {
        "$ref": "#/definitions/security"
      },
      "spec": {
        "comment": "section 2.2.5.1.1.3.1.3.1"
      }
    }
```

```
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
clusterName	string	Specifies the name of the big data cluster into which **nmnode** resources are deployed.
security	object	Section 2.2.5.1.1.1.4
spec	object	Section 2.2.5.1.1.3.1.3.1

2.2.5.1.1.3.1.3.1 spec

The **spec** element of the **nmnode** resource specifies the structured data that define the **nmnode** resources that are deployed in the cluster.

The JSON of this element is as follows.

```
      "spec": {
        "$id": "#/properties/spec/properties/resources/properties/nmnode/properties/spec",
        "type": "object",
        "required": [
          "replicas",
        ],
        "properties": {
          "replicas": {
            "$id": 
    "#/properties/spec/properties/resources/properties/nmnode/properties/spec/properties/replicas
    ",
            "type": "integer"
          },
          "docker": {
            "$ref": "#/definitions/docker"
          },
          "storage": {
            "$ref": "#/definitions/storage"
          },
          "settings": {
            "comment": "section 2.2.5.1.1.3.1.3.2"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
replicas	integer	Specifies the number of pods that are deployed for use by the **nmnode** resource.
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1
settings	object	Section 2.2.5.1.1.3.1.3.2

2.2.5.1.1.3.1.3.2 spec.settings

The **spec.settings** element of the **nmnode** resource specifies the settings of a **nmnode** resource.

The JSON of this element is as follows.

```
  "settings": {
    "$id":
"#/properties/spec/properties/resources/properties/nmnode/properties/spec/properties/settings
",
    "type": "object",
    "required": [
      "hdfs",
    ],
    "properties": {
      "hdfs": {
        "$ref": "#/definitions/hdfs"
      }
    }
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
hdfs	object	Section 2.2.5.1.1.1.6

2.2.5.1.1.3.1.4 master

The **master** resource provides various kinds of functionality in the big data cluster, such as for connectivity, scale-out query management, metadata and user databases, and machine learning services.

The JSON of this resource is as follows.

```
  "master": {
    "$id": "#/properties/spec/properties/resources/properties/master",
    "type": "object",
    "required": [
      "metadata",
      "spec"
    ],
    "properties": {
      "clusterName": {
        "$id":
"#/properties/spec/properties/resources/properties/master/properties/clusterName",
        "type": "string"
      },
      "metadata": {
        "$ref": "#/definitions/metadata"
      },
      "security": {
        "$ref": "#/definitions/security"
      },
      "spec": {
        "comment": "section 2.2.5.1.1.3.1.4.1"
      }
    }
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
metadata	object	Section 2.2.5.1.1.1.9
spec	object	Section 2.2.5.1.1.3.1.4.1
clusterName	string	Specifies the name of the big data cluster into which the **master** resource is deployed.
security	object	Section 2.2.5.1.1.1.4

2.2.5.1.1.3.1.4.1 (Updated Section) spec

The **spec** element of the **master** resource specifies the structured data that define the **master** resource that is deployed in the big data cluster.

The JSON of this element is as follows.

```
  "spec": {
    "$id": "#/properties/spec/properties/resources/properties/master/properties/spec",
    "type": "object",
    "required": [
      "type",
      "replicas",
      "endpoints"
    ],
    "properties": {
      "type": {
        "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/type",
        "type": "~~integer~~string"
      },
      "replicas": {
        "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/replicas
",
        "type": "integer"
      },
      "docker": {
        "$ref": "#/definitions/docker"
      },
      "storage": {
        "$ref": "#/definitions/storage"
      },
      "endpoints": {
        "comment": "section 2.2.5.1.1.3.1.4.2"
      },
      "settings": {
        "comment": "section 2.2.5.1.1.3.1.4.4"
      }
    }
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
type	~~integer~~string	Specifies the value that is used to define the resource type. The value of the **master** resource

Element name	Type	Description
		MUST be ~~the following:~~ ~~1 – master instance –~~ "Master".
replicas	integer	Specifies the number of pods that are deployed for use by the **master** resource.
endpoints	array	Section 2.2.5.1.1.3.1.4.2
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1
settings	object	Section 2.2.5.1.1.3.1.4.4

2.2.5.1.1.3.1.4.2 (Updated Section) spec.endpoints

The **spec.endpoints** element of the **master** resource defines externally available endpoints that can be used to access a resource.

The JSON of this element is as follows.

```
  "endpoints": {
    "$id": 
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s",
    "type": "array",
    "items": {
      "$id": 
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items",
      "type": "object",
      "required": [
        "name",
        "serviceType",
        "port"
      ],
      "properties": {
        "name": {
          "$id": 
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/name",
          "type": "string"
        },
        "serviceType": {
          "comment": "section 2.2.5.1.1.3.1.4.3"
        },
        "port": {
          "$id": 
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/port",
          "type": "integer",
          "examples": [
            31433,
          ]
        },
        "dnsName": {
          "$id": 
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/~~port~~dnsName",
          "type": "string"
        },
        "dynamicDnsUpdate": {
          "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/~~port~~dynamicDnsUpdate",
          "type": "boolean"
        }
      }
    }
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
name	string	Specifies the name of the endpoint.
serviceType	object	Section 2.2.5.1.1.3.1.4.3
port	integer	Specifies the port on which to access the endpoint.
dnsName	string	Specifies the DNS name of the endpoint.
dynamicDnsUpdate	boolean	Specifies a Boolean that indicates whether the DNS entry for the endpoint is to be automatically updated by the control plane.

2.2.5.1.1.3.1.4.3 (Updated Section) spec.endpoints.serviceType

The **spec.endpoints.serviceType** element of the **master** resource defines ~~the~~what type of Kubernetes ~~Service type~~ [Kubernetes] ~~that~~Service exposes the endpoint.

The JSON of this element is as follows.

```
  "serviceType": {
    "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/serviceType",
    "enum": [
      "NodePort",
      "LoadBalancer"
    ]
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
NodePort	string	Specifies that the endpoint is exposed by a Kubernetes **NodePort** ServiceType ~~[Kubernetes]~~.
LoadBalancer	string	Specifies that the endpoint is exposed by a Kubernetes **LoadBalancer** ServiceType ~~[Kubernetes]~~.

2.2.5.1.1.3.1.4.4 spec.settings

The **spec.settings** element of the **master** resource specifies the settings of the **master** resource.

The JSON of this element is as follows.

```
      "settings": {
        "$id": 
    "#/properties/spec/properties/resources/properties/master/properties/spec/properties/settings
    ",
        "type": "object",
        "required": [
          "sql",
        ],
        "properties": {
          "sql": {
            "$ref": "#/definitions/sql"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
sql	object	Section 2.2.5.1.1.1.7

2.2.5.1.1.3.1.5 compute

The **compute** resource specifies of a group of one or more pods that provides scale-out computational resources for the big data cluster.

If multiple compute pools are deployed, a "-#" suffix is appended to the resource name to denote ordinality, for example, "compute-0". This suffix is a positive integer that can range from 0 to n-1, where n is the number of compute pools that are deployed.

The JSON of this resource is as follows.

```
      "compute": {
        "$id": "#/properties/spec/properties/resources/properties/compute",
        "type": "object",
        "required": [
          "metadata",
          "spec"
        ],
        "properties": {
          "clusterName": {
            "$id": 
    "#/properties/spec/properties/resources/properties/compute/properties/clusterName",
            "type": "string"
          },
          "metadata": {
            "$ref": "#/definitions/metadata"
          },
          "security": {
            "$ref": "#/definitions/security"
          },
          "spec": {
            "comment": "section 2.2.5.1.1.3.1.5.1"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
metadata	object	Section 2.2.5.1.1.1.9
spec	object	Section 2.2.5.1.1.3.1.5.1
clusterName	string	Specifies the name of the big data cluster into which **compute** resources are deployed.
security	object	Section 2.2.5.1.1.1.4

2.2.5.1.1.3.1.5.1 (Updated Section) spec

The **spec~~.settings~~** element of the **compute** resource specifies the structured data that define a **compute** resource.

The JSON of this element is as follows.

```
  "spec": {
    "$id": "#/properties/spec/properties/resources/properties/compute/properties/spec",
    "type": "object",
    "required": [
      "type",
      "replicas"
    ],
    "properties": {
      "type": {
        "$id": "#/properties/spec/properties/resources/properties/compute
/properties/spec/properties/type",
        "type": "~~integer~~string"
      },
      "replicas": {
        "$id": "#/properties/spec/properties/resources/properties/compute
/properties/spec/properties/replicas",
        "type": "integer"
      },
      "docker": {
        "$ref": "#/definitions/docker"
      },
      "storage": {
        "$ref": "#/definitions/storage"
      },
      "settings": {
        "comment": "section 2.2.5.1.1.3.1.5.2"
      }
    }
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
type	~~integer~~string	Specifies the value that is used to define the resource type. The value of the **compute** resource MUST be ~~the following:~~ ~~2 - compute pool -~~ "Compute".
replicas	integer	Specifies the number of pods that are deployed

Element name	Type	Description
		for use by the **compute** resource.
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1
settings	object	Section 2.2.5.1.1.3.1.5.2

2.2.5.1.1.3.1.5.2 spec.settings

The **spec.settings** element of the **compute** resource specifies the settings of a **compute** resource.

The JSON of this element is as follows.

```
    "settings": {
      "$id": 
  "#/properties/spec/properties/resources/properties/compute/properties/spec/properties/setting
  s",
      "type": "object",
      "required": [
        "sql",
      ],
      "properties": {
        "sql": {
          "$ref": "#/definitions/sql"
        }
      }
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
sql	object	Section 2.2.5.1.1.1.7

2.2.5.1.1.3.1.6 appproxy

The **appproxy** resource specifies a pod that is deployed in the control plane and provides users with the ability to interact with the applications that are deployed in the big data cluster.

The JSON of this resource is as follows.

```
    "appproxy": {
      "$id": "#/properties/spec/properties/resources/properties/appproxy",
      "type": "object",
      "required": [
        "spec",
      ],
      "properties": {
        "clusterName": {
          "$id": 
  "#/properties/spec/properties/resources/properties/appproxy/properties/clusterName",
          "type": "string"
        },
        "security": {
          "$ref": "#/definitions/security"
        },
        "spec": {
          "comment": "section 2.2.5.1.1.3.1.6.1"
        }
      }
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
clusterName	string	Specifies the name of the big data cluster into which **appproxy** resources are deployed.
security	object	Section 2.2.5.1.1.1.4
spec	object	Section 2.2.5.1.1.3.1.6.1

2.2.5.1.1.3.1.6.1 spec

The **spec** element of the **appproxy** resource specifies the structured data that define the **appproxy** resources that are deployed in the big data cluster.

The JSON of this element is as follows.

```
      "spec": {
        "$id": "#/properties/spec/properties/resources/properties/appproxy/properties/spec",
        "type": "object",
        "required": [
          "replicas",
          "endpoints"
        ],
        "properties": {
          "replicas": {
            "$id": 
    "#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/replic
    as",
            "type": "integer"
          },
          "docker": {
            "$ref": "#/definitions/docker"
          },
          "storage": {
            "$ref": "#/definitions/storage"
          },
          "endpoints": {
            "comment": "section 2.2.5.1.1.3.1.6.2"
          },
          "settings": {
            "$id": 
    "#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/settin
    gs",
            "type": "object"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
replicas	integer	Specifies the number of pods that are deployed for use by the **appproxy** resource.
endpoints	array	Section 2.2.5.1.1.3.1.6.2
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1
settings	object	Reserved for future use. This element MUST be ignored.

2.2.5.1.1.3.1.6.2 (Updated Section) spec.endpoints

The **spec.endpoints** element for the **appproxy** resource defines the endpoints that can be used to interact with the **appproxy**.

The JSON of this element is as follows.

```
      "endpoints": {
        "$id":
    "#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/endpoi
    nts",
        "type": "array",
        "items": {
          "$id":
    "#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/endpoi
    nts/items",
          "type": "object",
          "required": [
            "name",
            "serviceType",
            "port"
          ],
          "properties": {
            "name": {
              "comment": "section 2.2.5.1.1.3.1.6.3"
            },
            "serviceType": {
              "comment": "section 2.2.5.1.1.3.1.6.4"
            },
            "port": {
              "$id":
    "#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/endpoi
    nts/items/properties/port",
              "type": "integer",
              "examples": [
                30778,
              ]
            },
            "dnsName": {
              "$id":
    "#/properties/spec/properties/resources/properties/~~master~~appproxy/properties/spec/properties/
    endpoints/items/properties/~~port~~dnsName",
              "type": "string"
            },
            "dynamicDnsUpdate": {
              "$id":
    "#/properties/spec/properties/resources/properties/~~master~~appproxy/properties/spec/properties/
    endpoints/items/properties/~~port~~dynamicDnsUpdate",
              "type": "boolean"
            }
            }
        }
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
name	object	Section 2.2.5.1.1.3.1.6.3
serviceType	object	Section 2.2.5.1.1.3.1.6.4
port	integer	Specifies the port on which the service is exposed.
dnsName	string	Specifies the DNS name registered for the endpoint that is registered to the DC for a deployment with Active Directory enabled.
dynamicDnsUpdate	boolean	Specifies a Boolean that indicates whether to register automatically the DNS service in the Active Directory DNS records.

2.2.5.1.1.3.1.6.3 spec.endpoints.name

The **spec.endpoints.name** element for the **appproxy** resource defines the name of the endpoint.

The JSON of this element is as follows.

```
    "name": {
      "$id": 
  "#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/endpoi
  nts/items/properties/name",
      "const": "AppServiceProxy"
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
AppServiceProxy	string	Specifies the name of the endpoint for the **appproxy** resource.

2.2.5.1.1.3.1.6.4 (Updated Section) spec.endpoints.serviceType

The **spec.endpoints.serviceType** element for the **appproxy** resource defines what type of Kubernetes ~~service~~[Kubernetes] Service exposes the endpoint.

The JSON of this element is as follows.

```
    "serviceType": {
      "$id": 
  "#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/endpoi
  nts/items/properties/serviceType",
      "enum": [
        "NodePort",
        "LoadBalancer"
      ]
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
NodePort	string	Specifies that the endpoint is exposed by a Kubernetes **NodePort** ServiceType ~~[Kubernetes].~~.
LoadBalancer	string	Specifies that the endpoint is exposed by a Kubernetes **LoadBalancer** ServiceType.

2.2.5.1.1.3.1.7 zookeeper

The **zookeeper** resource contains instances of Apache ZooKeeper [ApacheZooKeeper] that are used to provide a service that maintains synchronization in highly available systems in the big data cluster.

The JSON of this element is as follows.

```
    "zookeeper": {
      "$id": "#/properties/spec/properties/resources/properties/zookeeper",
      "type": "object",
      "required": [
        "spec",
      ],
      "properties": {
        "clusterName": {
          "$id":
  "#/properties/spec/properties/resources/properties/zookeeper/properties/clusterName",
          "type": "string"
        },
        "security": {
          "$ref": "#/definitions/security"
        },
        "spec": {
          "comment": "section 2.2.5.1.1.3.1.7.1"
        }
      }
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
clusterName	string	Specifies the name of the big data cluster into which **zookeeper** resources are deployed.
security	object	Section 2.2.5.1.1.1.4
spec	object	Section 2.2.5.1.1.3.1.7.1

2.2.5.1.1.3.1.7.1 spec

The **spec** element of the **zookeeper** resource specifies the structured data that define the **zookeeper** resources that are deployed in the big data cluster.

The JSON of this element is as follows.

```
      "spec": {
        "$id": "#/properties/spec/properties/resources/properties/zookeeper/properties/spec",
        "type": "object",
        "required": [
          "replicas",
        ],
        "properties": {
          "replicas": {
            "$id":
    "#/properties/spec/properties/resources/properties/zookeeper/properties/spec/properties/repli
    cas",
            "type": "integer"
          },
          "docker": {
            "$ref": "#/definitions/docker"
          },
          "storage": {
            "$ref": "#/definitions/storage"
          },
          "settings": {
            "comment": "section 2.2.5.1.1.3.1.7.2"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
replicas	integer	Specifies the number of pods that are deployed for use by the **zookeeper** resource.
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1
settings	object	Section 2.2.5.1.1.3.1.7.2

2.2.5.1.1.3.1.7.2 spec.settings

The **spec.settings** element of the **zookeeper** resource specifies the settings of a **zookeeper** resource.

The JSON of this element is as follows.

```
      "settings": {
        "$id":
    "#/properties/spec/properties/resources/properties/zookeeper/properties/spec/properties/setti
    ngs",
        "type": "object",
        "required": [
          "hdfs",
        ],
        "properties": {
          "hdfs": {
```

```
          "$id":
    "#/properties/spec/properties/resources/properties/zookeeper/properties/spec/properties/setti
    ngs/properties/hdfs",
          "type": "object"
        }
      }
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
hdfs	object	Section 2.2.5.1.1.1.6

2.2.5.1.1.3.1.8 gateway

The **gateway** resource uses the Apache Knox [ApacheKnox] gateway system to provide a secure endpoint by which to connect to Hadoop [ApacheHadoop].

The JSON of this resource is as follows.

```
      "gateway": {
        "$id": "#/properties/spec/properties/resources/properties/gateway",
        "type": "object",
        "required": [
          "spec",
        ],
        "properties": {
          "clusterName": {
            "$id":
    "#/properties/spec/properties/resources/properties/gateway/properties/clusterName",
            "type": "string"
          },
          "security": {
            "$ref": "#/definitions/security"
          },
          "spec": {
            "comment": "section 2.2.5.1.1.3.1.8.1"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
clusterName	string	Specifies the name of the big data cluster into which **gateway** resources are deployed.
security	object	Section 2.2.5.1.1.1.4
spec	object	Section 2.2.5.1.1.3.1.8.1

2.2.5.1.1.3.1.8.1 spec

The **spec** element of the **gateway** resource specifies the structured data that define the **gateway** resources that are deployed in the big data cluster.

The JSON of this element is as follows.

```
      "spec": {
        "$id": "#/properties/spec/properties/resources/properties/gateway/properties/spec",
        "type": "object",
        "required": [
          "replicas",
          "endpoints"
        ],
        "properties": {
          "replicas": {
            "$id": 
    "#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/replica
    s",
            "type": "integer"
          },
          "docker": {
            "$ref": "#/definitions/docker"
          },
          "storage": {
            "$ref": "#/definitions/storage"
          },
          "endpoints": {
            "comment": "section 2.2.5.1.1.3.1.8.2"
          },
          "settings": {
            "$id": 
    "#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/setting
    s",
            "type": "object"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
replicas	integer	Specifies the number of pods to deploy for use by the **gateway** resource.
endpoints	array	Section 2.2.5.1.1.3.1.8.2
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1
settings	object	Reserved for future use. This element MUST be ignored.

2.2.5.1.1.3.1.8.2 (Updated Section) spec.endpoints

The **spec.endpoints** element for the **gateway** resource defines the endpoints that can be used to interact with the **gateway** resource.

The JSON of this element is as follows.

```
      "endpoints": {
        "$id": 
    "#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoin
    ts",
        "type": "array",
        "items": {
          "$id":
    "#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoin
    ts/items",
          "type": "object",
          "required": [
            "name",
            "serviceType",
            "port"
          ],
          "properties": {
            "name": {
              "comment": "section 2.2.5.1.1.3.1.8.3"
            },
            "serviceType": {
              "comment": "section 2.2.5.1.1.3.1.8.4"
            },
            "port": {
              "$id":
    "#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoin
    ts/items/properties/port",
              "type": "integer"
            },
            "dnsName": {
              "$id":
    "#/properties/spec/properties/resources/properties/~~master~~gateway/properties/spec/properties/e
    ndpoints/items/properties/~~port~~dnsName",
              "type": "string"
            },
            "dynamicDnsUpdate": {
              "$id":
    "#/properties/spec/properties/resources/properties/~~master~~gateway/properties/spec/properties/e
    ndpoints/items/properties/~~port~~dynamicDnsUpdate",
              "type": "boolean"
            }
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
name	object	Section 2.2.5.1.1.3.1.8.3
serviceType	object	Section 2.2.5.1.1.3.1.8.4
port	integer	Specifies the port on which the service is exposed.
dnsName	string	Specifies the DNS name registered for the endpoint that is registered to the DC for a deployment with Active Directory enabled.
dynamicDnsUpdate	boolean	Specifies a Boolean that indicates whether to register automatically the DNS service in the Active Directory DNS records.

2.2.5.1.1.3.1.8.3 spec.endpoints.name

The **spec.endpoints.name** element of the **gateway** resource specifies the name of the endpoint.

The JSON of this element is as follows.

```
      "name": {
        "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoin
ts/items/properties/name",
        "const": "Knox"
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
Knox	string	Specifies the name of the Apache Knox [ApacheKnox] endpoint for the **gateway** resource.

2.2.5.1.1.3.1.8.4 (Updated Section) spec.endpoints.serviceType

The **spec.endpoints.serviceType** element of the **gateway** resource ~~specifies the~~defines what type of Kubernetes [Kubernetes] Service~~ type that~~ exposes the endpoint.

The JSON of this element is as follows.

```
      "serviceType": {
        "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoin
ts/items/properties/serviceType",
        "enum": [
          "NodePort",
          "LoadBalancer"
        ]
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
NodePort	string	Specifies that the endpoint is exposed by a Kubernetes **NodePort** ServiceType ~~[Kubernetes].~~.
LoadBalancer	string	Specifies that the endpoint is exposed by a Kubernetes **LoadBalancer** ServiceType.

2.2.5.1.1.3.1.9 spark

The **spark** resource specifies the settings structured data that define the spark service settings.

The JSON schema for this resource is as follows.

```
        "spark": {
          "$id": "#/properties/spec/properties/resources/properties/spark",
          "type": "object",
          "required": [
            "metadata",
            "spec"
          ],
          "properties": {
            "clusterName": {
              "$id":
    "#/properties/spec/properties/resources/properties/spark/properties/clusterName",
              "type": "string"
            },
            "security": {
              "$ref": "#/definitions/security"
            },
            "metadata": {
              "$ref": "#/definitions/metadata"
            },
            "spec": {
              "comment": "section 2.2.5.1.1.3.1.9.1"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
clusterName	string	Specifies the name of the big data cluster into which **spark** resources are deployed.
security	object	Section 2.2.5.1.1.1.4
metadata	object	Section 2.2.5.1.1.1.9
spec	object	Section 2.2.5.1.1.3.1.9.1

2.2.5.1.1.3.1.9.1 (Updated Section) spec

The **spec** element of the **spark** resource specifies the structured data that define the **spark** resources that are deployed in the big data cluster.

The JSON schema for this resource is as follows.

```
      "spec": {
        "$id": "#/properties/spec/properties/resources/properties/spark/properties/spec",
        "type": "object",
        "required": [
          "type",
          "replicas"
        ],
        "properties": {
          "type": {
            "$id":
    "#/properties/spec/properties/resources/properties/spark/properties/spec/properties/type",
            "type": "~~integer~~string"
          },
          "replicas": {
            "$id": "#/properties/spec/properties/resources/properties/spark/properties
    /spec/properties/replicas",
            "type": "integer"
          },
          "docker": {
            "$ref": "#/definitions/docker"
          },
          "storage": {
            "$ref": "#/definitions/storage"
          },
          "settings": {
            "comment": "section 2.2.5.1.1.3.1.9.2"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
type	~~integer~~string	Specifies the value that is used to define the resource type. The value of the **spark** resource MUST be ~~the following:~~ ~~6 – spark pool~~"Spark".
replicas	integer	Specifies the number of pods that are deployed for use by the **spark** resource.
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1
settings	object	Section 2.2.5.1.1.3.1.9.2

2.2.5.1.1.3.1.9.2 spec.settings

The **spec.settings** element of the **spark** resource specifies the settings of a **spark** resource.

The JSON of this element is as follows.

```
  "settings": {
    "$id": "#/properties/spec/properties/resources/properties/spark/properties/spec
/properties/settings",
    "type": "object",
    "required": [
      "spark",
      "hdfs"
    ],
    "properties": {
      "spark": {
        "$ref": "#/definitions/spark"
      },
      "hdfs": {
        "$ref": "#/definitions/hdfs"
      }
    }
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
spark	object	Section 2.2.5.1.1.1.5
hdfs	object	Section 2.2.5.1.1.1.6

2.2.5.1.1.3.1.10 data

The **data** resource specifies a group of pods that provide persistent storage in the big data cluster.

If multiple data pools are deployed, a "-#" suffix is appended to the resource name to denote ordinality, for example, "data-0". This suffix is a positive integer that can range from 0 to n-1, where n is the number of data pools that are deployed.

The JSON schema for this resource is as follows.

```
      "data": {
        "$id": "#/properties/spec/properties/resources/properties/data",
        "type": "object",
        "required": [
          "metadata",
          "spec"
        ],
        "properties": {
          "clusterName": {
            "$id": 
    "#/properties/spec/properties/resources/properties/data/properties/clusterName",
            "type": "string"
          },
          "security": {
            "$ref": "#/definitions/security"
          },
          "metadata": {
            "$ref": "#/definitions/metadata"
          },
          "spec": {
            "comment": "section 2.2.5.1.1.3.1.10.1"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
metadata	object	Section 2.2.5.1.1.1.9
spec	object	Section 2.2.5.1.1.3.1.10.1
clusterName	string	Specifies the name of the big data cluster into which **data** resources are deployed.
security	object	Section 2.2.5.1.1.1.4

2.2.5.1.1.3.1.10.1 (Updated Section) spec

The **spec** element of the **data** resource specifies the structured data that define the **data** resources that are deployed in the big data cluster.

The JSON of this element is as follows.

```
      "spec": {
        "$id": "#/properties/spec/properties/resources/properties/data/properties/spec",
        "type": "object",
        "required": [
          "type",
          "replicas"
        ],
        "properties": {
          "type": {
            "$id":
    "#/properties/spec/properties/resources/properties/data/properties/spec/properties/type",
            "type": "~~integer~~string"
          },
          "replicas": {
            "$id":
    "#/properties/spec/properties/resources/properties/data/properties/spec/properties/replicas",
            "type": "integer"
          },
          "docker": {
            "$ref": "#/definitions/docker"
          },
          "storage": {
            "$ref": "#/definitions/storage"
          },
          "settings": {
            "comment": "section 2.2.5.1.1.3.1.10.2"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
type	~~integer~~string	Specifies the value that is used to define the resource type. The value of the **data** resource MUST be ~~the following:~~ ~~3 – data pool~~"Data".
replicas	integer	Specifies the number of pods that are deployed for use by the **data** resource.
docker	object	Section 2.2.5.1.1.1.3
storage	object	Section 2.2.5.1.1.1.1
settings	object	Section 2.2.5.1.1.3.1.10.2

2.2.5.1.1.3.1.10.2 spec.settings

The **spec.settings** element of the **data** resource specifies the settings of a **data** resource.

The JSON of this element is as follows.

```
  "settings": {
    "$id":
"#/properties/spec/properties/resources/properties/data/properties/spec/properties/settings",
    "type": "object",
    "required": [
      "sql",
    ],
    "properties": {
      "sql": {
        "$ref": "#/definitions/sql"
      }
```

```
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
sql	object	Section 2.2.5.1.1.1.7

2.2.5.1.1.3.2 services

The **services** element of the **spec** element consists of the services that are deployed to distribute and manage the resources in the big data cluster.

The JSON of this element is as follows.

```
      "services": {
        "$id": "#/properties/spec/properties/services",
        "type": "object",
        "required": [
          "sql",
          "hdfs",
          "spark"
        ],
        "properties": {
          "sql": {
            "comment": "section 2.2.5.1.1.3.2.1"
          },
          "hdfs": {
            "comment": "section 2.2.5.1.1.3.2.2"
          },
          "spark": {
            "comment": "section 2.2.5.1.1.3.2.3"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
sql	object	Section 2.2.5.1.1.3.2.1
hdfs	object	Section 2.2.5.1.1.3.2.2
spark	object	Section 2.2.5.1.1.3.2.3

2.2.5.1.1.3.2.1 sql

The **sql** service distributes and manages the SQL Server service in the big data cluster.

The JSON of this service is as follows.

```
      "sql": {
        "$id": "#/properties/spec/properties/services/properties/sql",
        "type": "object",
        "required": [
          "resources",
        ],
        "properties": {
          "resources": {
            "comment": "section 2.2.5.1.1.3.2.1.1"
          },
          "settings": {
            "$ref": "#/definitions/sql"
          }
        }
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
resources	array	Section 2.2.5.1.1.3.2.1.1
settings	object	Section 2.2.5.1.1.1.7

2.2.5.1.1.3.2.1.1 resources

The **resources** element of the **sql** service specifies an array of resources that use the SQL Server service in the big data cluster.

The JSON of this element is as follows.

```
  "resources": {
    "$id": "#/properties/spec/properties/services/properties/sql/properties/resources",
    "type": "array",
    "items": [
      {
        "const": "master"
      },
      {
        "const": "compute"
      },
      {
        "const": "data"
      },
      {
        "const": "storage"
      }
    ]
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
master	string	Specifies the **master** resource.
compute	string	Specifies the **compute** resource.
data	string	Specifies the **data** resource.
storage	string	Specifies the **storage** resource.

2.2.5.1.1.3.2.2 hdfs

The **hdfs** service distributes and manages the HDFS [ApacheHadoop] storage and file system in the big data cluster.

The JSON of this service is as follows.

```
"hdfs": {
  "$id": "#/properties/spec/properties/services/properties/hdfs",
  "type": "object",
  "required": [
    "resources",
  ],
  "properties": {
    "resources": {
      "comment": "section 2.2.5.1.1.3.2.2.1"
    },
    "settings": {
      "$ref": "#/definitions/hdfs"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
resources	array	Section 2.2.5.1.1.3.2.2.1
settings	object	Section 2.2.5.1.1.1.6

2.2.5.1.1.3.2.2.1 resources

The **resources** element of an **hdfs** service specifies the array of resources that are deployed to an HDFS service in the big data cluster.

The JSON of this element is as follows.

```
"resources": {
  "$id": "#/properties/spec/properties/services/properties/hdfs/properties/resources",
  "type": "array",
  "items": [
    {
      "const": "nmnode"
    },
    {
      "const": "zookeeper"
    },
    {
      "const": "storage"
    },
    {
      "const": "sparkhead"
    },
    {
      "const": "spark"
    }
  ]
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
nmnode	string	Specifies the **nmnode** resource.
zookeeper	string	Specifies the **zookeeper** resource.
storage	string	Specifies the **storage** resource.
sparkhead	string	Specifies the **sparkhead** resource.
spark	string	Specifies the **spark** resource.

2.2.5.1.1.3.2.3 spark

The **spark** service provides support for in-memory processing to boost the performance of the big-data analytic applications in the big data cluster.

The JSON of this service is as follows.

```
"spark": {
  "$id": "#/properties/spec/properties/services/properties/spark",
  "type": "object",
  "required": [
    "resources",
    "settings"
  ],
  "properties": {
    "resources": {
      "comment": "section 2.2.5.1.1.3.2.3.1"
    },
    "settings": {
      "$ref": "#/definitions/spark"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
resources	array	Section 2.2.5.1.1.3.2.3.1
settings	object	Section 2.2.5.1.1.1.5

2.2.5.1.1.3.2.3.1 resources

The **resources** element of a **spark** service specifies an array of resources that define the resources that are deployed in the **spark** services.

The JSON of this element is as follows.

```
"resources": {
  "$id": "#/properties/spec/properties/services/properties/spark/properties/resources",
  "type": "array",
  "items": [
    {
      "const": "sparkhead"
    },
    {
      "const": "storage"
    },
    {
      "const": "spark"
    }
  ]
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
sparkhead	string	Specifies the **sparkhead** resource.
storage	string	Specifies the **storage** resource.
spark	string	Specifies the **spark** resource.

2.2.5.1.2 Big Data Cluster Error Response Schema

The **Big Data Cluster Error Response** schema specifies the response body that describes the current errors in the big data cluster.

The JSON of this schema is as follows.

```
"$schema": "http://json-schema.org/draft-07/schema#",
"type": "object",
"title": "The Root Schema",
"required": [
  "code",
  "reason",
  "data"
],
"properties": {
  "code": {
    "$id": "#/properties/code",
    "type": "integer",
    "title": "The Code Schema",
    "default": 0,
    "examples": [
      500,
    ]
  },
  "reason": {
    "$id": "#/properties/reason",
    "type": "string",
    "default": "",
    "examples": [
      "An unexpected exception occurred.",
    ]
  },
  "data": {
    "$id": "#/properties/data",
    "type": "string",
    "default": "",
    "examples": [
      "Null reference exception",
    ]
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
code	integer	Specifies the HTTP status code that results from the operation.
reason	string	Specifies the description of the error that occurred.
data	string	Specifies the data on which the error occurred.

2.2.5.1.3 (Updated Section) Big Data Cluster Information Schema

The **Big Data Cluster Information** schema provides the specified information about the big data cluster.

The JSON of this schema is as follows.

```
"$schema": "http://json-schema.org/draft-07/schema#",
"type": "object",
"required": [
  "~~code~~state",
  ~~"state",~~
  "spec"
],
"properties": {
  ~~"code": {~~
    ~~"$id": "#/properties/code",~~
    ~~"type": "integer"~~
  ~~},~~
  "state": {
    "$id": "#/properties/state",
    "type": "string",
    "title": "The State Schema"
  },
  "spec": {
    "$id": "#/properties/spec",
    "type": "string"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
~~code~~	~~integer~~	~~Specifies the HTTP status code that results from the operation.~~
state	string	Specifies the state of the big data cluster.
spec	string	Specifies a JSON string that represents the JSON model. The value of the **spec** element is escaped by the server before it is sent to the client. The **spec** value is then unescaped by the client to create a valid JSON document.

2.2.5.1.4 Big Data Cluster Status Schema

The **Big Data Cluster Status** schema retrieves the specified status information about the big data cluster.

The JSON of this schema is as follows.

```
"$schema": "http://json-schema.org/draft-07/schema#",
"type": "object",
"required": [
  "bdcName",
  "state",
  "healthStatus",
  "details",
  "services"
],
"properties": {
  "bdcName": {
    "type": "string"
  },
  "state": {
    "type": "string"
  },
  "healthStatus": {
    "type": "string"
  },
  "details": {
    "type": "string"
  },
  "services": {
    "comment": "section 2.2.5.1.4.1"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
bdcName	string	Specifies the name of the big data cluster for which status is requested.
state	string	Specifies the state of the big data cluster deployment.
healthStatus	string	Specifies the health status of the big data cluster.
details	string	Specifies the additional information that is requested about the status of the big data cluster.
services	array	Section 2.2.5.1.4.1

2.2.5.1.4.1 services

The **services** element provides a list of the names of the services about which status information is requested.

The JSON of this schema is as follows.

```
"services": {
  "type": "array",
  "title": "The Services Schema",
  "items": {
    "type": "object",
    "title": "The Items Schema",
    "required": [
      "serviceName",
      "state",
      "healthStatus",
      "details",
      "resources"
    ],
    "properties": {
      "serviceName": {
        "type": "string"
      },
      "state": {
        "type": "string"
      },
      "healthStatus": {
        "type": "string"
      },
      "details": {
        "type": "string"
      },
      "resources": {
        "comment": "section 2.2.5.1.4.1.1"
      }
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
serviceName	string	Specifies the name of the service to be queried.
state	string	Specifies the state of the specified deployed service.
healthStatus	string	Specifies the health status of the specified deployed service.
details	string	Specifies the additional information that is requested about the deployed service.
resources	array	Section 2.2.5.1.4.1.1

2.2.5.1.4.1.1 resources

The **resources** element specifies an array of resources that are used in a service about which status information is requested.

The JSON of this element is as follows.

```
"resources": {
  "type": "array",
  "title": "The Resources Schema",
  "items": {
    "type": "object",
    "title": "The Items Schema",
    "required": [
      "resourceName",
      "state",
      "healthStatus",
      "details",
      "instances"
    ],
    "properties": {
      "resourceName": {
        "type": "string"
      },
      "state": {
        "type": "string"
      },
      "healthStatus": {
        "type": "string"
      },
      "details": {
        "type": "string"
      },
      "instances": {
        "comment": "section 2.2.5.1.4.1.1.1"
      }
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
resourceName	string	Specifies the name of the deployed resource to be queried.
state	string	Specifies the state of the deployed resource.
healthStatus	string	Specifies the health status of the deployed resource.
details	string	Specifies the additional information that is requested about the deployed resource.
instances	array	Section 2.2.5.1.4.1.1.1

2.2.5.1.4.1.1.1 instances

The **instances** element contains an array that includes information about all the instances of a specific resource in the cluster.

The JSON of this element is as follows.

```
    "instances": {
      "type": "array",
      "title": "The Instances Schema",
      "items": {
        "type": "object",
        "title": "The Items Schema",
        "required": [
          "instanceName",
          "state",
          "healthStatus",
          "details",
          "dashboards"
        ],
        "properties": {
          "instanceName": {
            "type": "string"
          },
          "state": {
            "type": "string"
          },
          "healthStatus": {
            "type": "string"
          },
          "details": {
            "type": "string"
          },
          "dashboards": {
            "comment": "section 2.2.5.1.4.1.1.1.1"
          }
        }
      }
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
instanceName	string	Specifies the name of the instance to be queried.
state	string	Specifies the state of the instance.
healthStatus	string	Specifies the health status of the instance.
details	string	Specifies the additional information that is requested about the instance.
dashboards	object	Section 2.2.5.1.4.1.1.1.1

2.2.5.1.4.1.1.1.1 dashboards

The **dashboards** element defines a series of URLs that can be used to retrieve logs and metrics for a specific resource instance.

The JSON of this element is as follows.

```
"dashboards": {
  "type": "object",
  "title": "The Dashboards Schema",
  "required": [
    "nodeMetricsUrl",
    "sqlMetricsUrl",
    "logsUrl"
  ],
  "properties": {
    "nodeMetricsUrl": {
      "type": "string",
      "examples": [
        "https://0.0.0.0:30777/api/v1/bdc/instances/master/status/nodemetrics/ui",
      ]
    },
    "sqlMetricsUrl": {
      "type": "string",
      "examples": [
        "https://0.0.0.0:30777/api/v1/bdc/instances/master/status/sqlmetrics/ui",
      ]
    },
    "logsUrl": {
      "type": "string",
      "examples": [
        "https://0.0.0.0:30777/api/v1/bdc/instances/master/status/logs/ui",
      ]
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
nodeMetricsUrl	string	Specifies a URL that can be used to view metrics about the nodes in the cluster on which the service is running.
sqlMetricsUrl	string	Specifies a URL that can be used to view metrics about the SQL Server metrics for the resource instance.
logsUrl	string	Specifies a URL that can be used to view the logs for the resource instance.

2.2.5.1.5 Big Data Cluster Service Status Schema

The **Big Data Cluster Service Status** schema retrieves the requested status information about the services in the big data cluster.

The JSON of this schema is as follows.

```
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/root.json",
  "type": "object",
  "required": [
    "serviceName",
    "state",
    "healthStatus",
    "details",
    "resources"
  ],
  "properties": {
    "serviceName": {
      "type": "string"
    },
    "state": {
      "type": "string"
    },
    "healthStatus": {
      "type": "string"
    },
    "details": {
      "type": "string"
    },
    "resources": {
      "comment": "section 2.2.5.1.5.1"
    }
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
serviceName	string	Specifies the name of the deployed service to be queried.
state	string	Specifies the state of the deployed service.
healthStatus	string	Specifies the health status of the deployed service.
details	string	Specifies the additional information requested about the deployed service.
resources	array	Section 2.2.5.1.5.1

2.2.5.1.5.1 resources

The **resources** element specifies the array of resources that are deployed in the big data cluster.

The JSON of this element is as follows.

```
"resources": {
  "$id": "#/properties/resources",
  "type": "array",
  "title": "The Resources Schema",
  "items": {
    "$id": "#/properties/resources/items",
    "type": "object",
    "title": "The Items Schema",
    "required": [
      "resourceName",
      "state",
      "healthStatus",
      "details",
      "instances"
    ],
    "properties": {
      "resourceName": {
        "type": "string"
      },
      "state": {
        "type": "string"
      },
      "healthStatus": {
        "type": "string"
      },
      "details": {
        "type": "string"
      },
      "instances": {
        "comment": "section 2.2.5.1.5.1.1"
      }
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
resourceName	string	Specifies the name of the deployed resource to be queried.
state	string	Specifies the state of the deployed resource.
healthStatus	string	Specifies the health status of the deployed resource.
details	string	Specifies the additional information that is requested about the deployed resource.
instances	array	Section 2.2.5.1.5.1.1

2.2.5.1.5.1.1 instances

The **instances** element contains an array that includes information about all the instances of a specific resource in the cluster.

The JSON of this element is as follows.

```
"instances": {
  "type": "array",
  "title": "The Instances Schema",
  "items": {
    "type": "object",
    "title": "The Items Schema",
    "required": [
      "instanceName",
      "state",
      "healthStatus",
      "details",
      "dashboards"
    ],
    "properties": {
      "instanceName": {
        "type": "string"
      },
      "state": {
        "type": "string"
      },
      "healthStatus": {
        "type": "string"
      },
      "details": {
        "type": "string"
      },
      "dashboards": {
        "comment": "section 2.2.5.1.5.1.1.1"
      }
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
instanceName	string	Specifies the name of the instance to be queried.

Element name	Type	Description
state	string	Specifies the state of the instance.
healthStatus	string	Specifies the health status of the instance.
details	string	Specifies the additional information that is requested about the instance.
dashboards	object	Section 2.2.5.1.5.1.1.1

2.2.5.1.5.1.1.1 dashboards

The **dashboards** element defines a series of URLs that can be used to retrieve logs and metrics for a specific resource instance.

The JSON of this element is as follows.

```
"dashboards": {
  "type": "object",
  "title": "The Dashboards Schema",
  "required": [
    "nodeMetricsUrl",
    "sqlMetricsUrl",
    "logsUrl"
  ],
  "properties": {
    "nodeMetricsUrl": {
      "type": "string"
    },
    "sqlMetricsUrl": {
      "type": "string"
    },
    "logsUrl": {
      "type": "string"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
nodeMetricsUrl	string	Specifies a URL that can be used to view metrics about the nodes in the cluster on which the service is running.
sqlMetricsUrl	string	Specifies a URL that can be used to view metrics about the SQL Server metrics for the resource instance.
logsUrl	string	Specifies a URL that can be used to view the logs for the resource instance.

2.2.5.1.6 Big Data Cluster Service Resource Status Schema

The **Big Data Cluster Service Resource Status** schema retrieves the requested status information about a resource that is deployed in a service in the big data cluster.

The JSON of this schema is as follows.

```
"$schema": "http://json-schema.org/draft-07/schema#",
"$id": "http://example.com/root.json",
"type": "object",
"title": "The Root Schema",
"required": [
  "resourceName",
  "state",
  "healthStatus",
  "details",
  "instances"
],
"properties": {
  "resourceName": {
    "$id": "#/properties/resourceName",
    "type": "string"
  },
  "state": {
    "$id": "#/properties/state",
    "type": "string"
  },
  "healthStatus": {
    "$id": "#/properties/healthStatus",
    "type": "string"
  },
  "details": {
    "$id": "#/properties/details",
    "type": "string"
  },
  "instances": {
    "comment": "section 2.2.5.1.6.1"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
resourceName	string	Specifies the name of the deployed resource to be queried.
state	string	Specifies the state of the deployed resource.
healthStatus	string	Specifies the health status of the deployed resource.
details	string	Specifies the additional information that is requested about the deployed resource.
instances	array	Section 2.2.5.1.6.1

2.2.5.1.6.1 instances

The **instances** element contains an array that contains information about all the instances of a specific resource in the cluster.

The JSON of this element is as follows.

```
"instances": {
  "$id": "#/properties/instances",
  "type": "array",
  "items": {
    "type": "object",
    "title": "The Items Schema",
    "required": [
      "instanceName",
      "state",
      "healthStatus",
      "details",
      "dashboards"
    ],
    "properties": {
      "instanceName": {
        "type": "string"
      },
      "state": {
        "type": "string"
      },
      "healthStatus": {
        "type": "string"
      },
      "details": {
        "type": "string"
      },
      "dashboards": {
        "comment": "section 2.2.5.1.6.1.1"
      }
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
instanceName	string	Specifies the name of the instance to be queried.
state	string	Specifies the state of the instance.
healthStatus	string	Specifies the health status of the instance.
details	string	Specifies the additional information that is requested about the instance.
dashboards	object	Section 2.2.5.1.6.1.1

2.2.5.1.6.1.1 dashboards

The **dashboards** element defines a series of URLs that can be used to retrieve logs and metrics for a specific resource instance.

The JSON of this element is as follows.

```
"dashboards": {
  "type": "object",
  "title": "The Dashboards Schema",
  "required": [
    "nodeMetricsUrl",
    "sqlMetricsUrl",
    "logsUrl"
  ],
  "properties": {
    "nodeMetricsUrl": {
      "type": "string"
    },
    "sqlMetricsUrl": {
      "type": "string"
    },
    "logsUrl": {
      "type": "string"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
nodeMetricsUrl	string	Specifies a URL that can be used to view metrics about the nodes in the cluster on which the service is running.
sqlMetricsUrl	string	Specifies a URL that can be used to view metrics about the SQL Server metrics for the resource instance.
logsUrl	string	Specifies a URL that can be used to view the logs for the resource instance.

2.2.5.1.7 Big Data Cluster Upgrade Schema

The **Big Data Cluster Upgrade** schema is used to upgrade the big data cluster.

The JSON of this schema is as follows.

```
"$schema": "http://json-schema.org/draft-07/schema",
"type": "object",
"required": [
    "targetVersion",
    "targetRepository"
],
"properties": {
    "targetVersion": {
        "$id": "#/properties/targetVersion",
        "type": "string",
    },
    "targetRepository": {
        "$id": "#/properties/targetRepository",
        "type": "string",
    }
 }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
targetVersion	string	Specifies the Docker image tag that all containers in the cluster are to be updated to.
targetRepository	string	Specifies the Docker repository from which to

Element name	Type	Description
		retrieve the Docker image. This element is used when the desired repository differs from the repository that is currently being used by the big data cluster.

2.2.5.1.8 Big Data Cluster Endpoints List Schema

The **Big Data Cluster Endpoints List** schema is used to retrieve a listing of information about the endpoints that are exposed for a component.

The JSON of this schema is as follows.

```
"$schema": "http://json-schema.org/draft-07/schema#",
"$id": "http://example.com/root.json",
"type": "array",
"title": "The Root Schema",
"items": {
  "$id": "#/items",
  "type": "object",
  "required": [
    "name",
    "description",
    "endpoint",
    "protocol"
  ],
  "properties": {
    "name": {
      "$id": "#/items/properties/name",
      "type": "string",
      "title": "The Name Schema"
    },
    "description": {
      "$id": "#/items/properties/description",
      "type": "string"
    },
    "endpoint": {
      "$id": "#/items/properties/endpoint",
      "type": "string"
    },
    "protocol": {
      "comment": "section 2.2.5.1.8.1"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
name	string	Specifies the name of the list of endpoints that are exposed for a component.
description	string	Specifies what the endpoint is used to access.
endpoint	string	Specifies the endpoint that is exposed outside of the cluster.
protocol	object	Section 2.2.5.1.8.1

2.2.5.1.8.1 protocol

The **protocol** element specifies the network protocol that is used to access the endpoint.

The JSON of this schema is as follows.

```
"protocol": {
  "enum": [
    "https",
    "tds"
  ]
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
https	string	Specifies that the endpoint is accessed through the HTTPS protocol [RFC7230] [RFC7231].
tds	string	Specifies that the endpoint is accessed through the Tabular Data Stream (TDS) protocol [MS-TDS].

2.2.5.1.9 Big Data Cluster Endpoint Schema

The **Big Data Cluster Endpoint** schema is used to expose an endpoint outside the cluster.

The JSON of this schema is as follows.

```
"$schema": "http://json-schema.org/draft-07/schema#",
"$id": "http://example.com/root.json",
"type": "object",
"required": [
  "name",
  "description",
  "endpoint",
  "protocol"
],
"properties": {
  "name": {
    "$id": "#/properties/name",
    "type": "string",
    "title": "The Name Schema"
  },
  "description": {
    "$id": "#/properties/description",
    "type": "string"
  },
  "endpoint": {
    "$id": "#/properties/endpoint",
    "type": "string"
  },
  "protocol": {
    "comment": "section 2.2.5.1.9.1"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
name	string	Specifies the name of the endpoint that is exposed for a component.
description	string	Specifies what the endpoint is used to access.
endpoint	string	Specifies the endpoint that is exposed outside of the cluster.
protocol	object	Section 2.2.5.1.9.1

2.2.5.1.9.1 protocol

The **protocol** element specifies the network protocol that is used to access the endpoint.

The JSON of this schema is as follows.

```
"protocol": {
  "enum": [
    "https",
    "tds"
  ]
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
https	string	Specifies that the endpoint is accessed through the https protocol [RFC7230] [RFC7231].
tds	string	Specifies that the endpoint is accessed through the TDS protocol [MS-TDS].

2.2.5.2 Storage

The **Storage** resource type defines the persistent storage that is used in the big data cluster.

This protocol defines the following set of common **Storage** JSON schemas.

Schema	Section	Description
Storage Response Schema	2.2.5.2.1	Specifies the response body that describes the state of a mounted remote HDFS storage volume.

2.2.5.2.1 Storage Response Schema

The **Storage Response** schema specifies the response body that describes the state of a remote HDFS storage volume that is mounted in the big data cluster.

The following is the schema in JSON format.

```
        "$schema": "http://json-schema.org/draft-07/schema#",
        "type": "object",
        "title": "Storage Response Schema",
        "required": [
          "mount",
          "remote",
          "state",
          "error"
        ],
        "properties": {
          "mount": {
            "$id": "#/properties/mount",
            "type": "string"
          },
          "remote": {
            "$id": "#/properties/remote",
            "type": "string"
          },
          "state": {
            "comment": "section 2.2.5.2.1.1"
          },
          "error": {
            "$id": "#/properties/error",
            "type": "string"
          }
        }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
mount	string	Specifies the path of the HDFS mount.
remote	string	Specifies the HDFS mount point to attach the mount to.
state	object	Section 2.2.5.2.1.1
error	string	Specifies that the mount is unhealthy. This field is populated only if the mount is unhealthy.

2.2.5.2.1.1 state

The **state** element defines the valid state values of the HDFS mount deployment.

The JSON schema for this element is as follows.

```
      "state": {
        "$id": "#/properties/state",
        "enum": [
          "Initial",
          "Creating",
          "WaitingForCreate",
          "Updating",
          "WaitingForUpdate",
          "Ready",
          "Deleting",
          "WaitingForDelete",
          "Deleted",
          "Error"
        ]
      }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
Initial	string	Indicates that the request to create the HDFS mount was received.
Creating	string	Indicates that HDFS mount creation is being initiated.
WaitingForCreate	string	Indicates that HDFS mount creation is in progress.
Updating	string	Indicates that the HDFS mount has received an update request.
WaitingForUpdate	string	Indicates that the HDFS mount is in the process of updating.
Ready	string	Indicates that the HDFS mount is ready.
Deleting	string	Indicates that the HDFS mount is initiating deletion.
WaitingForDelete	string	Indicates that the HDFS mount is in the process of deletion.
Deleted	string	Indicates that the HDFS mount has been deleted.
Error	string	Indicates that the HDFS mount is in an error state.

2.2.5.3 App

The **App** resource type defines the application scripts that are deployed in the big data cluster.

This protocol defines the following set of common **App** JSON schemas.

Schema	Section	Description
App Description Schema	2.2.5.3.1	Defines an overview of an application.
App Run Result	2.2.5.3.2	Describes the results of running an application.

2.2.5.3.1 (Updated Section) App Description Schema

The **App Description** schema defines an overview of an application that is deployed in the big data cluster.

The following is the schema in JSON format.

```
        "$schema": "http://json-schema.org/draft-07/schema#",
        "type": "~~array~~object",
        "title": "App Result Schema",
        "definitions": {
          "comment": "section 2.2.5.3.1.1"
        },
        "items": {
          "$id": "#/items",
          "type": "object",
          "required": [
            "name",
            "internal_name",
            "version",
            "input_param_defs",
            "output_param_defs",
            "state",
            "links"
          ],
          "properties": {
            "name": {
              "$id": "#/items/properties/name",
              "type": "string"
            },
            "internal_name": {
              "$id": "#/items/properties/internal_name",
              "type": "string"
            },
            "version": {
              "$id": "#/items/properties/version",
              "type": "string"
            },
            "input_param_defs": {
              "comment": "section 2.2.5.3.1.2"
            },
            "output_param_defs": {
              "comment": "section 2.2.5.3.1.3"
            },
            "state": {
              "comment": "section 2.2.5.3.1.4"
            },
            "links": {
              "comment": "section 2.2.5.3.1.5"
            }
          }
        }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
definitions	object	Section 2.2.5.3.1.1
name	string	Specifies the name of the application that is deployed.
internal_name	string	Specifies the name of the application that is used internally within the cluster.
version	string	Specifies the version of the application that is deployed.
input_param_defs	array	Section 2.2.5.3.1.2
output_param_defs	array	Section 2.2.5.3.1.3
state	object	Section 2.2.5.3.1.4
links	object	Section 2.2.5.3.1.5

2.2.5.3.1.1 definitions

The **definitions** element specifies schema elements that can be referenced by other elements within the schema.

The JSON schema for this element is as follows.

```
"definitions": {
  "link": {
    "comment": "section 2.2.5.3.1.1.1"
  },
  "parameter": {
    "comment": "section 2.2.5.3.1.1.2"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
link	object	Section 2.2.5.3.1.1.1
parameter	object	Section 2.2.5.3.1.1.2

2.2.5.3.1.1.1 link

The **link** element defines a URL that can be used to interact with the deployed application.

The JSON schema for this element is as follows.

```
"link": {
  "type": "object",
  "properties": {
    "*": {
      "type": "string"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
*	string	Specifies a URL that matches the regular expression and can be used to interact with the deployed application. For further details, see [RFC3986].

2.2.5.3.1.1.2 parameter

The **parameter** element is a data structure that represents an application parameter.

The JSON schema for this element is as follows.

```
"parameter": {
  "required": [
    "name",
    "type"
  ],
  "properties": {
    "name": {
      "type": "string"
    },
    "type": {
      "comment": "section 2.2.5.3.1.1.2.1"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
name	string	Specifies the name of the parameter being defined.
type	object	Section 2.2.5.3.1.1.2.1

2.2.5.3.1.1.2.1 (Updated Section) type

The **type** element defines the valid data types of application parameters.

The JSON schema for this element is as follows.

```
"type": {
  "enum": [
    "str",
    "int",
    "dataframe",
    "data.frame",
    "float",
    "matrix",
    "vector",
    "bool"
  ]
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
str	string	Specifies that the parameter is a string.
int	string	Specifies that the parameter is an integer.
dataframe	string	Specifies that the parameter is a Data Frame.
data.frame	string	~~Specifies that~~Synonym for the ~~parameter is a Data Frame~~**dataframe** element.
float	string	Specifies that the parameter is a float.
matrix	string	Specifies that the parameter is a matrix.

Element name	Type	Description
vector	string	Specifies that the parameter is a vector.
bool	string	Specifies that the parameter is a Boolean.

2.2.5.3.1.2 input_param_defs

The **input_param_defs** element is an array of parameter objects that represent the inputs that can be passed to the application.

```
"input_param_defs": {
  "$id": "#/items/properties/input_param_defs",
  "type": "array",
  "description": "Array of input parameters for the deployed app",
  "items": {
    "$ref": "#/definitions/parameter"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
parameter	object	Section 2.2.5.3.1.1.2

2.2.5.3.1.3 output_param_defs

The **output_param_defs** element is an array of parameter objects that represent the outputs of the application.

The JSON schema for this element is as follows.

```
"output_param_defs": {
  "$id": "#/items/properties/output_param_defs",
  "type": "array",
  "items": {
    "$ref": "#/definitions/parameter"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
parameter	object	Section 2.2.5.3.1.1.2

2.2.5.3.1.4 state

The **state** element defines the valid states of the application's deployment.

The JSON schema for this element is as follows.

```
"state": {
  "$id": "#/items/properties/state",
  "enum": [
    "Initial",
    "Creating",
    "WaitingForCreate",
    "Updating",
    "WaitingForUpdate",
    "Ready",
    "Deleting",
    "WaitingForDelete",
    "Deleted",
    "Error"
  ]
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
Initial	string	Indicates that the application is about to be created.
Creating	string	Indicates that the application has begun creation.
WaitingForCreate	string	Indicates that the application is in the process of creation.
Updating	string	Indicates that the application is about to be updated.
WaitingForUpdate	string	Indicates that the application is in the process of being updated.
Ready	string	Indicates that the application is in the Ready state.
Deleting	string	Indicates that the application is preparing to be deleted.
WaitingForDelete	string	Indicates that the application is in the process of deletion.
Deleted	string	Indicates that the application has been deleted.
Error	string	Indicates that an error has occurred.

2.2.5.3.1.5 links

The **links** element is a data structure that represents a URL that can be used to access the deployed application.

The JSON schema for this element is as follows.

```
"links": {
  "$id": "#/properties/links",
  "type": "object",
  "required": [
    "app",
    "swagger"
  ],
  "properties": {
    "app": {
      "$id": "#/properties/links/properties/app",
      "type": "string"
    },
    "swagger": {
      "$id": "#/properties/links/properties/swagger",
      "type": "string"
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
app	string	Specifies an endpoint by which to access the deployed application.
swagger	string	Specifies an endpoint to a Swagger editor [Swagger2.0] that can be used directly to send requests to the deployed application.

2.2.5.3.2 App Run Result Schema

The **App Run Result** schema describes the results of running an application.

The following is the schema in JSON format.

```
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "required": [
    "success",
    "errorMessage",
    "outputFiles",
    "consoleOutput",
    "changedFiles"
  ],
  "properties": {
    "success": {
      "$id": "#/properties/success",
      "type": "boolean"
    },
    "errorMessage": {
      "$id": "#/properties/errorMessage",
      "type": "string"
    },
    "outputParameters": {
      "comment": "section 2.2.5.3.2.1"
    },
    "outputFiles": {
      "$id": "#/properties/outputFiles",
      "type": "object"
    },
    "consoleOutput": {
      "$id": "#/properties/consoleOutput",
      "type": "string"
    },
    "changedFiles": {
      "$id": "#/properties/changedFiles",
      "type": "array"
    }
    }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
success	boolean	Specifies a Boolean that indicates whether the application method succeeded.
errorMessage	string	Specifies a string that describes the reason the application method failed.
outputParameters	object	Section 2.2.5.3.2.1
outputFiles	object	Specifies an array of file names that resulted from the application operation.
consoleOutput	string	Specifies the text output that resulted from the application method.
changedFiles	array	Specifies an array of file names that were modified as a result of the application operation.

2.2.5.3.2.1 outputParameters

The **outputParameters** element contains a list of output parameters that resulted from the application method. A result is optional and can be of any type, based on the output_param_defs element as specified in section 2.2.5.3.1.3.

The JSON schema for this element is as follows.

```
"outputParameters": {
  "$id": "#/properties/outputParameters",
  "type": "object",
  "properties": {
    "result": {
      "$id": "#/properties/outputParameters/properties/result",
      "type": "string,
        "enum": [
          "str",
          "int",
          "dataframe",
          "data.frame",
          "float",
          "matrix",
          "vector",
          "bool"
      ]
    }
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
result	integer	Specifies the value that is returned by application execution.

Element name	Type	Description
type	string	Section 2.2.5.3.1.1.2.1

2.2.5.4 Token

The **Token** resource type defines the authentication that is used in the big data cluster.

This protocol defines the following set of common **Token** JSON schemas.

Schema	Section	Description
Token Response Schema	2.2.5.4.1	Defines the response body that is returned when a user requests an authentication token.

2.2.5.4.1 Token Response Schema

The **Token Response** schema defines the response body that is returned when a user requests an authentication token.

The following is the schema in JSON format.

```
"$schema": "http://json-schema.org/draft-07/schema#",
"type": "object",
"required": [
  "token_type",
  "access_token",
  "expires_in",
  "expires_on",
  "token_id"
],
"properties": {
  "token_type": {
    "$id": "#/properties/token_type",
    "type": "string"
  },
  "access_token": {
    "$id": "#/properties/access_token",
    "type": "string"
  },
  "expires_in": {
    "$id": "#/properties/expires_in",
    "type": "integer"
  },
  "expires_on": {
    "$id": "#/properties/expires_on",
    "type": "integer"
  },
  "token_id": {
    "$id": "#/properties/token_id",
    "type": "string"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
token_type	string	Specifies a returned token type that is Bearer.
access_token	string	Specifies the JWT token that is generated for the request.
expires_in	integer	Specifies the number of seconds for which the token is valid after being issued.
expires_on	integer	Specifies the date on which the token expires, which is based on the number of seconds since the Unix Epoch.
token_id	string	Specifies a unique ID that was generated for the token request.

2.2.5.5 Home

The **Home** resource type defines the response that is used to determine whether the control plane service is listening for requests.

This protocol defines the following set of common **Home** JSON schemas.

Schema	Section	Description
Ping Response Schema	2.2.5.5.1	Specifies the availability of the control plane.
Info Response Schema	2.2.5.5.2	Specifies a JSON model that contains metadata about the control plane.

2.2.5.5.1 Ping Response Schema

The **Ping Response** schema specifies the availability information that is used to determine whether the Control Plane REST API is responsive.

The following is the schema in JSON format.

```
"$schema": "http://json-schema.org/draft-07/schema#",
"$id": "http://example.com/root.json",
"type": "object",
"title": "The Root Schema",
"required": [
  "code",
  "message"
],
"properties": {
  "code": {
    "comment": "section 2.2.5.5.1.1"
  },
  "message": {
    "comment": "section 2.2.5.5.1.2"
  }
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
code	object	Section 2.2.5.5.1.1
message	object	Section 2.2.5.5.1.2

2.2.5.5.1.1 code

The **code** element specifies the HTTP Response Status code that is returned in response to the ping made to the controller.

The JSON schema for this element is as follows.

```
"code": {
  "$id": "#/properties/code",
  "const": 200
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
200	integer	Indicates that the control plane is responsive.

2.2.5.5.1.2 message

The **message** element contains a message that describes the status of the controller.

The JSON schema for this element is as follows.

```
"message": {
  "$id": "#/properties/message",
  "const": "Controller is available."
}
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
Controller is available.	string	Indicates that the control plane has been verified as being available.

2.2.5.5.2 Info Response Schema

The **Info Response** schema specifies a JSON model that contains metadata about the deployed control plane.

The following is the schema in JSON format.

```
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/root.json",
  "type": "object",
  "title": "The Root Schema",
  "required": [
    "version",
    "buildTimestamp"
  ],
  "properties": {
    "version": {
      "$id": "#/properties/version",
      "type": "string"
    },
    "buildTimestamp": {
      "$id": "#/properties/buildTimestamp",
      "type": "string"
    }
  }
```

This protocol defines the following common JSON schema elements.

Element name	Type	Description
version	string	Specifies the version of the controller that was pinged.
buildTimestamp	string	Specifies the build and time information for the message that was returned in response to the ping that was made to the controller.

3 Protocol Details

3.1 Common Details

If an HTTP operation is unsuccessful, the server MUST return the error as JSON content in the response. The format of the JSON response is provided in the Response Body sections of the methods that can performed during HTTP operations.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

For a client to use this protocol, the client MUST have a healthy control plane service that is running in a Kubernetes cluster.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 (Updated Section) Message Processing Events and Sequencing Rules

The methods of the APIs that are defined in this section can use resources that are defined according to the resource types in section 2.2.5.

The following APIs are invoked and managed by using the control plane service.

API	Section	Description
Big Data Cluster	3.1.5.1	The APIs that are used to perform create retrieve update delete (CRUD) operations against the big data cluster resource that is deployed in the Kubernetes cluster.
Control	3.1.5.2	The APIs that are used to perform ~~CRUD~~management operations against the control plane.
Storage	3.1.5.3	The APIs that are used to perform CRUD operations on external mounts that are mounted in the HDFS instance of the big data cluster.
App Deploy	3.1.5.4	The APIs that are used to perform CRUD operations against a standalone application that is deployed in a pod in the cluster.
Token	3.1.5.5	The APIs that are used to create a token that can be included as a header in an application call in the cluster.
Home Page	3.1.5.6	The APIs that are used to monitor whether the control plane service is listening for requests.

The URL of the message that invokes the resource is formed by concatenating the following components:

- The absolute URI to the control plane service.
- A string that represents the endpoint to be accessed.
- The remainder of the desired HTTP URL as described in the following sections.

Requests require a Basic authentication header or a JWT authentication token [RFC7519] (see section 3.1.5.5) to be attached to the request. However, if the control plane is set up by using Active Directory, an exception for this is the Token API, as described in section 3.1.5.5.1, and which requires either a Basic authentication header or a negotiation header [RFC4559].

For example, to retrieve the state of a currently deployed cluster that is named "test", the following request is sent by using Basic authentication.

```
        curl -k -u admin:<adminPassword> --header "X-RequestID: 72b674f3-9288-42c6-a47b-
             948011f15010" https://<clusterIp>:<controllerPort>/api/v1/bdc/status
```

adminPassword: The administrator password for the cluster that was defined during control plane service setup.

k: The parameter that is required because the cluster uses self-signed certificates. For more information, see section 5.1.

header: The parameter that adds the **X-RequestID** header to the request.

The following request, for example, is sent by using a negotiation header.

```
    curl -k -X POST https://control.bdc.local:30080/api/v1/token -H "Content-Length: 0"
             --negotiate
```

negotiate: The control plane authenticates the request by using negotiation. An empty username and password are sent in the request.

3.1.5.1 (Updated Section) Big Data Cluster

A **Big Data Cluster** API represents a big data cluster resource that is deployed in a Kubernetes cluster in a Kubernetes namespace of the same name.

The methods of this API are invoked by using the following URI.

```
    https://<clusterIp>:<controllerPort>/api/v1/bdc
```

The following methods can be performed during HTTP operations on this resource.

Method	Section	Description
Create Big Data Cluster	3.1.5.1.1	Creates a **big data cluster** resource.
Delete Big Data Cluster	3.1.5.1.2	Deletes a ~~Big Data Cluster~~big data cluster resource.
Get Big Data Cluster Logs	3.1.5.1.3	Retrieves logs from a ~~Big Data Cluster~~big data cluster resource.
Get Big Data Cluster Status	3.1.5.1.4	Retrieves the status of a ~~Big Data Cluster~~big data cluster resource.

Method	Section	Description
Get Big Data Cluster Information	3.1.5.1.5	Retrieves the status and configuration of a ~~Big Data Cluster~~big data cluster resource.
Get Service Status	3.1.5.1.6	Retrieves the statuses of all resources in a service in a ~~Big Data Cluster~~big data cluster resource.
Get Service Resource Status	3.1.5.1.7	Retrieves the status of a resource in a service in a ~~Big Data Cluster~~big data cluster resource.
Redirect to Metrics Link	3.1.5.1.8	Redirects the client to a metrics dashboard.
Upgrade Big Data Cluster	3.1.5.1.9	Updates the Docker images that are deployed in a ~~Big Data Cluster~~big data cluster resource.
Get All Big Data Cluster Endpoints	3.1.5.1.10	Retrieves a list of all endpoints exposed by a ~~BDC~~big data cluster resource.
Get ~~BDC~~Big Data Cluster Endpoint	3.1.5.1.11	Retrieves the endpoint information for a specific endpoint in the ~~BDC~~big data cluster resource.

3.1.5.1.1 Create Big Data Cluster

The **Create Big Data Cluster** method creates a big data cluster in the Kubernetes cluster.

This method is invoked by sending a POST operation to the following URI:

```
https://<clusterIp>:<controllerPort>/api/v1/bdc
```

The HTTP response message for the **Create Big Data Cluster** method can include the following status codes.

HTTP status code	Description
200	The cluster specification was accepted, and creation of the big data cluster has been initiated.
400	The control plane service failed to parse the cluster specification.
400	A cluster with the provided name already exists.
500	An unexpected error occurred while parsing the cluster specification.
500	An internal error occurred while initiating the create event for the cluster.
500	The operation failed to store the list of the data pool nodes in metadata storage.
500	The operation failed to store the list of the storage pool nodes in metadata storage.

3.1.5.1.1.1 Request Body

The request body is a JSON object that is defined by the **Big Data Cluster Spec** schema, as specified in section 2.2.5.1.1. For an example of this request body, see section 4.2.

3.1.5.1.1.2 Response Body

If the request is successful, no response body is returned.

If the request fails, the response body is a JSON object that is defined by the **Big Data Cluster Error Response** schema, as specified in section 2.2.5.1.2. For an example of this response body, see section 4.4.

3.1.5.1.1.3 (Updated Section) Processing Details

This method creates a new ~~Big Data Cluster~~big data cluster resource.

3.1.5.1.2 (Updated Section) Delete Big Data Cluster

The **Delete Big Data Cluster** method deletes the ~~Big Data Cluster~~big data cluster resource that is deployed in the cluster.

It is invoked by sending a DELETE operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/bdc
```

The HTTP response message for the **Delete Big Data Cluster** method can include the following status codes.

HTTP status code	Description
200	Deletion of the ~~Big Data Cluster~~big data cluster resource was initiated.
500	Deletion of the ~~Big Data Cluster~~big data cluster resource failed due to an internal error.

3.1.5.1.2.1 Request Body

The request body is empty. There are no parameters.

3.1.5.1.2.2 Response Body

The response body is empty.

3.1.5.1.2.3 (Updated Section) Processing Details

This method deletes a ~~Big Data Cluster~~big data cluster resource.

3.1.5.1.3 (Updated Section) Get Big Data Cluster Logs

The **Get Big Data Cluster Logs** method retrieves the logs from the ~~Big Data Cluster~~big data cluster resource.

This method is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/bdc/log?offset=<offsetNumber>
```

offset: A parameter that allows a partial log to be returned. If the value of **offset** is 0, the whole log is returned. If the value of **offset** is nonzero, the log that is returned starts at the byte located at the **offset** value.

The HTTP response message for the **Get Big Data Cluster Logs** method can include the following status code.

HTTP status code	Description
200	The logs are successfully returned.

3.1.5.1.3.1 Request Body

The request body is empty.

3.1.5.1.3.2 Response Body

The response body contains the contents of the log file. The log starts with the **offset** value and continues to the end of the log.

3.1.5.1.3.3 Processing Details

The client is responsible for tracking the offset into the file when a partial log is retrieved. To do so, the client adds the previous **offset** value to the length of the log returned. This value represents the new **offset** value.

3.1.5.1.4 (Updated Section) Get Big Data Cluster Status

The **Get Big Data Cluster Status** method retrieves the status of all resources in a ~~Big Data Cluster~~big data cluster resource.

This method is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/bdc/status?all=[<true/false>]
```

all: If the query parameter is set to "all", additional information is provided about all instances that exist for each resource in all the services.

The HTTP response message for the **Get Big Data Cluster Status** method can include the following status codes.

HTTP status code	Description
200	The status of the ~~Big Data Cluster~~big data cluster resource was returned successfully.
404	No ~~Big Data Cluster~~big data cluster resource is currently deployed.
500	The operation failed to retrieve the status of the ~~Big Data Cluster~~big data cluster resource.

3.1.5.1.4.1 Request Body

The request body is empty.

3.1.5.1.4.2 Response Body

The response body is a JSON object that is defined by the **Big Data Cluster Status** schema, as specified in section 2.2.5.1.4. For examples of the response body, see sections 4.3 and 4.5.

3.1.5.1.4.3 Processing Details

None.

3.1.5.1.5 (Updated Section) Get Big Data Cluster Information

Th **Get Big Data Cluster Information** method retrieves the status and configuration of the ~~Big Data Cluster~~big data cluster resource.

This method is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/bdc
```

The HTTP response message for the **Get Big Data Cluster Information** method can include the following status codes.

HTTP status code	Description
200	**Big** ~~Data Cluster~~data cluster resource information was returned successfully.
404	No ~~Big Data Cluster~~big data cluster resource is currently deployed.
500	The operation failed to retrieve the information for the currently deployed ~~Big Data Cluster~~big data cluster resource.

3.1.5.1.5.1 Request Body

The request body is empty.

3.1.5.1.5.2 Response Body

The response body is a JSON object that is defined by the **Big Data Cluster Information** schema, as specified in section 2.2.5.1.3. For an example of the response body, see section 4.6.

3.1.5.1.5.3 Processing Details

None.

3.1.5.1.6 (Updated Section) Get Service Status

The **Get Service Status** method retrieves the statuses of all services in a specified service in the ~~Big Data Cluster~~big data cluster resource.

It is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/bdc/services/<serviceName>/status?all=[<true/fals
e>]
```

serviceName: The name of the service for which to retrieve the status. The value can be one of the following:

- **SQL**: The status of SQL nodes in the cluster.
- **HDFS**: The status of all HDFS nodes in the cluster.
- **Spark**: The status of all Spark nodes in the cluster.
- **Control**: The status of all components in the control plane.

all: If the query parameter is set to "all", additional information is provided about all instances that exist for each resource in the specified service.

The HTTP response message for the **Get Service Status** method can include the following status codes.

HTTP status code	Description
200	**Service** status was returned successfully.
404	The **service** that is specified by **serviceName** does not exist.
500	An unexpected exception occurred.

3.1.5.1.6.1 Request Body

The request body is empty.

3.1.5.1.6.2 Response Body

The response body is a JSON object that is defined by the **Big Data Cluster Service Status** schema, as specified in section 2.2.5.1.5. For an example of the response body, see section 4.7.

3.1.5.1.6.3 Processing Details

None.

3.1.5.1.7 (Updated Section) Get Service Resource Status

The **Get Service Resource Status** method retrieves the status of a resource within a specified service in the ~~Big Data Cluster~~big data cluster resource.

It is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/bdc/services/<serviceName>/resource/<resourceName
>/status?all=[<true/false>]
```

serviceName: The name of the service for which to retrieve the status. The value can be one of the following:

- **SQL**: The status of SQL nodes in the cluster.
- **HDFS**: The status of all HDFS nodes in the cluster.
- **Spark**: The status of all Spark nodes in the cluster.

- **Control**: The status of all components in the control plane.

resourceName: The name of the resource for which to retrieve the status.

all: If the query parameter is set to "all", additional information is provided about all instances that exist for each resource in the specified service.

The HTTP response message for the **Get Service Resource Status** method can include the following status codes.

HTTP status code	Description
200	Service resource status was returned successfully.
404	The service that is specified by **serviceName** or resource that is specified by **resourceName** does not exist.
500	An unexpected exception occurred.

3.1.5.1.7.1 Request Body

The request body is empty.

3.1.5.1.7.2 Response Body

The response body is a JSON object that is defined by the **Big Data Cluster Service Resource Status** schema, as specified in section 2.2.5.1.6. For an example of the response body, see section 4.8.

3.1.5.1.7.3 Processing Details

None.

3.1.5.1.8 Redirect to Metrics Link

The **Redirect to Metrics Link** method redirects the client to a URL that displays metrics for components in the big data cluster.

It is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/bdc/instances/<instanceName>/status/<linkType>/ui
```

instanceName: The name of the instance for which to retrieve the URI.

linkType: The type of link to retrieve. The value can be one of the following:

- **SqlMetrics**: Metrics for any SQL instances that are running in the requested instance.
- **NodeMetrics**: Metrics for the node that contains the pod on which the instance is running.
- **Logs**: A link to a dashboard that contains the logs from the requested instance.

The HTTP response message for the **Redirect to Metrics Link** method can include the following status codes.

HTTP status code	Description
302	The redirect was successful.
404	The resource that is specified in the request does not exist.
500	The server is unable to redirect the client.

3.1.5.1.8.1 Request Body

The request body is empty.

3.1.5.1.8.2 Response Body

The response body is empty.

3.1.5.1.8.3 Processing Details

None.

3.1.5.1.9 (Updated Section) Upgrade Big Data Cluster

The **Upgrade Big Data Cluster** method updates the Docker images that are deployed in the ~~Big Data Cluster~~big data cluster resource.

It is invoked by sending a PATCH operation to the following URI.

```
    https://<clusterIp>:<controllerPort>/api/v1/bdc
```

The HTTP response message for the **Upgrade Big Data Cluster** method can include the following status codes.

HTTP status code	Description
200	A Docker images upgrade was initiated.
400	The upgrade request is invalid.
500	An unexpected error occurred while processing the upgrade.

3.1.5.1.9.1 Request Body

The response body is a JSON object that is defined by the **Big Data Cluster Upgrade** schema, as specified in section 2.2.5.1.7. For an example of the request body, see section 4.9.

3.1.5.1.9.2 Response Body

If the request is successful, no response body is returned.

If the request fails, the response body is a JSON object that is defined by the **Big Data Cluster Error Response** schema, as specified in section 2.2.5.1.2. For an example of the response body, see section 4.4.

3.1.5.1.9.3 Processing Details

This method updates the Docker images that are deployed in the big data cluster.

3.1.5.1.10 (Updated Section) Get All Big Data Cluster Endpoints

The **Get All Big Data Cluster Endpoints** method retrieves a list of all endpoints exposed by a ~~Big Data Cluster~~big data cluster resource.

It is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/bdc/endpoints
```

The HTTP response message for the **Get All Big Data Cluster Endpoints** method can include the following status codes.

HTTP status code	Description
200	The big data cluster endpoints were successfully returned.

3.1.5.1.10.1 Request Body

The request body is empty.

3.1.5.1.10.2 Response Body

The response body is a JSON object that is defined by the **Big Data Cluster Endpoints List** schema, as specified in section 2.2.5.1.8. For an example of the response body, see section 4.10.

3.1.5.1.10.3 Processing Details

None.

3.1.5.1.11 (Updated Section) Get Big Data Cluster Endpoint

The **Get Big Data Cluster Endpoint** method retrieves the endpoint information for a specific endpoint in the ~~Big Data Cluster~~big data cluster resource.

It is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/bdc/endpoints/<endpointName>
```

endpointName: The name of the endpoint for which to retrieve information. This value can be one of the following:

- **gateway**: Gateway to access HDFS [ApacheHadoop] files and Apache Spark [ApacheSpark].
- **spark-history**: Portal for managing and monitoring Apache Spark jobs.
- **yarn-ui**: Portal for accessing Apache Spark monitoring and diagnostics.
- **app-proxy**: Proxy for running commands against applications deployed in the cluster.
- **mgmtproxy**: Proxy for accessing services which monitor the health of the cluster.

- **logsui**: Dashboard for searching through cluster logs.
- **metricsui**: Dashboard for searching through cluster metrics.
- **controller**: Endpoint for accessing the controller.
- **sql-server-master**: SQL Server master instance front end.
- **webhdfs**: HDFS file system proxy.
- **livy**: Proxy for running Apache Spark statements, jobs, and applications.

The HTTP response message for the **Get Big Data Cluster Endpoint** method can include the following status codes.

HTTP status code	Description
200	The ~~Big Data Cluster~~big data cluster endpoint was successfully returned.
404	The ~~Big Data Cluster~~big data cluster endpoint was not found.

3.1.5.1.11.1 Request Body

The request body is empty.

3.1.5.1.11.2 Response Body

The response body is a JSON object that is defined by the **Big Data Cluster Endpoint** schema, as specified in section 2.2.5.1.9. For an example of the response body, see section 4.11.

3.1.5.1.11.3 Processing Details

None.

3.1.5.2 Control

The **Control** API describes the state of the control plane.

The methods of this API are invoked by using the following URI.

```
    https://<clusterIp>:<controllerPort>/api/v1/control
```

The following methods can be performed by using HTTP operations on this resource.

Method	Section	Description
Get Control Status	3.1.5.2.1	Retrieve the status of the control plane.
Upgrade Control	3.1.5.2.2	Upgrade the control plane.
Redirect to Metrics Link	3.1.5.2.3	Redirects the client to a URI that displays metrics for a resource in the control plane.
Get Control Resource Status	3.1.5.2.4	Retrieves the status of a resource in the control plane.

3.1.5.2.1 Get Control Status

The **Get Control Status** method is used to retrieve the statuses of all components in the control plane.

This method is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/control?all=<true>
```

For an example of the curl request, see section 4.1.

The HTTP response message for the **Get Control Status** method can include the following status codes.

HTTP status code	Description
200	The control plane statuses were returned successfully.
500	An unexpected error occurred.

3.1.5.2.1.1 Request Body

The request body is empty.

3.1.5.2.1.2 Response Body

The response body is a JSON object that is defined by the **Big Data Cluster Service Status** schema, as specified in section 2.2.5.1.5. For an example of the response body, see section 4.7.

3.1.5.2.1.3 Processing Details

None.

3.1.5.2.2 Upgrade Control

The **Upgrade Control** method is used to update images currently deployed in the control plane.

This method is invoked by sending a PATCH operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/control
```

The HTTP response message for the **Upgrade Control** method can include the following status codes.

HTTP status code	Description
200	The control plane was upgraded successfully.
500	An unexpected error occurred while upgrading the control plane.

3.1.5.2.2.1 Request Body

The response body is a JSON object that is defined by the **Big Data Cluster Upgrade** schema, as specified in section 2.2.5.1.7. For an example of the request body, see section 4.9.

3.1.5.2.2.2 Response Body

If the request is successful, no response body is returned.

If the request fails, the response body is a JSON object that is defined by the **Big Data Cluster Error Response** schema, as specified in section 2.2.5.1.2. For an example of the response body, see section 4.4.

3.1.5.2.2.3 Processing Details

This method is used to update the Docker images that are deployed in the control plane.

3.1.5.2.3 Redirect to Metrics Link

The **Redirect to Metrics Link** method redirects the client to a URL that displays metrics for components in a cluster.

It is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/control/instances/<instanceName>/status/<linkType
>/ui
```

instanceName: The name of the pod for which to retrieve the URI.

linkType: The type of link to retrieve. The value can be one of the following:

- **SqlMetrics**: Metrics for an SQL instances that are running in the requested instance.
- **NodeMetrics**: Metrics for the node that contains the pod on which the instance is running.
- **Logs**: A link to a dashboard that contains the logs from the requested instance.

The HTTP response message for the **Redirect to Metrics Link** method can include the following status codes.

HTTP status code	Description
302	The redirect was successful.
400	The resource that is specified in the request does not exist.
500	The server is unable to redirect the client.

3.1.5.2.3.1 Request Body

The request body is empty.

3.1.5.2.3.2 Response Body

The response body is empty.

3.1.5.2.3.3 Processing Details

None.

3.1.5.2.4 Get Control Resource Status

The **Get Control Resource Status** method retrieves the status of a resource in the control plane.

It is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/control/resources/<resourceName>/status?all=[<tru
e/false>]
```

resourceName: The name of the resource for which to retrieve the status.

all: If the query parameter is set to "all", additional information is provided about all instances that exist for each resource in the specified service.

The HTTP response message for the **Get Control Resource Status** method can include the following status codes.

HTTP status code	Description
200	The resource status was returned successfully.
404	The resource that is specified by **resourceName** does not exist.
500	An unexpected exception occurred.

3.1.5.2.4.1 Request Body

The request body is empty.

3.1.5.2.4.2 Response Body

The response body is a JSON object that is defined by the **Big Data Cluster Service Resource Status** schema, as specified in section 2.2.5.1.6. For an example of the response body, section 4.8.

3.1.5.2.4.3 Processing Details

None.

3.1.5.3 (Updated Section) Storage

The **Storage** API specifies a remote file system that is mounted to a path in the cluster’s local HDFS.

The methods of this API are invoked by using the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/storage/mounts
```

The following methods can be performed by using HTTP operations on this resource.

Method	Section	Description
Get Mount Status	3.1.5.3.1	Retrieve the status of a specified mount in the cluster.

Method	Section	Description
Get All Mount Statuses	3.1.5.3.2	Retrieve the status of all mounts in the cluster.
Create Mount	3.1.5.3.3	Create a mount.
Delete Mount	3.1.5.3.4	Delete a mount.
Refresh ~~mount~~Mount	3.1.5.3.5	Refresh a mount.

3.1.5.3.1 Get Mount Status

The **Get Mount Status** method is used to retrieve the status of one or more HDFS mounts in the cluster.

This method is invoked by sending a GET operation to the following URI.

```
    https://<clusterIp>:<controllerPort>/api/v1/storage/mounts/<mountPath>
```

mountPath: The directory of the mount.

The HTTP response message for the **Get Mount Status** method can include the following status codes.

HTTP status code	Description
200	The mount status was returned successfully.
404	The mount that is specified by **mountPath** does not exist.

3.1.5.3.1.1 Request Body

The request body is empty.

3.1.5.3.1.2 Response Body

The response body is a JSON object that is defined by the **Storage Response** schema, as specified in section 2.2.5.2.1. For an example of the response body, see section 4.12.

3.1.5.3.1.3 Processing Details

This method is used to retrieve the status of one or more HDFS **mounts** in the cluster.

3.1.5.3.2 Get All Mount Statuses

The **Get All Mount Statuses** method is used to retrieve the statuses of all HDFS **mounts** in the cluster.

This method is invoked by sending a GET operation to the following URI.

```
    https://<clusterIp>:<controllerPort>/api/v1/storage/mounts/
```

The HTTP response message for the **Get All Mount Statuses** method can include the following status code.

HTTP status code	Description
200	All mount statuses were returned successfully.

3.1.5.3.2.1 Request Body

The request body is empty.

3.1.5.3.2.2 Response Body

The response body is a JSON object that is defined by the **Storage Response** schema, as specified in section 2.2.5.2.1. For an example of the response body, see section 4.12.

3.1.5.3.2.3 Processing Details

This method is used to retrieve the status of all HDFS **mounts** in the cluster.

3.1.5.3.3 Create Mount

The **Create Mount** method creates an HDFS mount within the cluster.

This method is invoked by sending a POST operation to the following URI.

```
    https://<clusterIp>:<controllerPort>/api/v1/storage/mounts?remote=<remote>&mount=<mount>
```

remote: The URI of the store to mount.

mount: The local HDFS path for the mount point.

The HTTP response message for the **Create Mount** method can include the following status codes.

HTTP status code	Description
202	Mount creation was successfully initiated.
400	The specified mount already exists.
500	An internal error occurred while initiating the create event for the specified mount.
500	An unexpected error occurred while processing the mount credentials.

3.1.5.3.3.1 Request Body

The request body is a request in JSON format in which each element corresponds to an authentication element that is needed to access the remote file system. The authentication elements that are required vary from provider to provider.

3.1.5.3.3.2 Response Body

The response body is empty.

3.1.5.3.3.3 Processing Details

The client can use the GET operation to monitor the creation of the mount.

3.1.5.3.4 Delete Mount

The **Delete Mount** method deletes a mounted HDFS mount.

This method is invoked by sending a DELETE operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/storage/mounts?mount=<mount>
```

mount: The mount point to delete.

The HTTP response message for the **Delete Mount** method can include the following status codes.

HTTP status code	Description
202	The delete request was accepted.
400	The delete request is invalid.
404	The specified mount does not exist.
500	The method failed to delete the specified mount.

3.1.5.3.4.1 Request Body

The request body is empty.

3.1.5.3.4.2 Response Body

If the request is successful, no response body is returned.

If the request fails, the response body is a JSON object that is defined by the **Big Data Cluster Error Response** schema, as specified in section 2.2.5.1.2. For an example of the response body, see section 4.4.

3.1.5.3.4.3 Processing Details

The client can use the **Get Mount Status** method to monitor the deletion of the mount.

3.1.5.3.5 Refresh Mount

The **Refresh Mount** method refreshes a currently mounted mount to update the files and permissions that are stored in HDFS.

It is invoked by sending a POST operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/storage/mounts/refresh?mount=<mount>
```

mount: The mount to refresh.

The HTTP response message for the **Refresh Mount** method can include the following status codes.

HTTP status code	Description
202	The refresh request was accepted.
400	The refresh request is invalid.
404	The specified mount does not exist.
500	The method failed to refresh specified mount.

3.1.5.3.5.1 Request Body

The request body is empty.

3.1.5.3.5.2 Response Body

If the request is successful, no response body is returned.

If the request fails, the response body is as a JSON object that is defined by the **Big Data Cluster Error Response** schema, as specified in section 2.2.5.1.2. For an example of a response body, see section 4.4.

3.1.5.3.5.3 Processing Details

None.

3.1.5.4 (Updated Section) App Deploy

The **App Deploy** ~~resource~~API specifies an R or Python script that can be deployed or is deployed in the **cluster**.

The methods of this API are invoked by using the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/app
```

The following methods can be performed by using HTTP operations on this resource.

Method	Section	Description
Get App	3.1.5.4.1	Retrieve the status of the application.
Get App Versions	3.1.5.4.2	Retrieve the status of all deployed applications.
Get All Apps	3.1.5.4.3	Retrieve the status of one or more deployed applications.
Create App	3.1.5.4.4	Create an application.
Update App	3.1.5.4.5	Update a deployed application.
Delete App	3.1.5.4.6	Delete a deployed application.
Run App	3.1.5.4.7	Send inputs to a deployed application.
Get App Swagger Document	3.1.5.4.8	Retrieve a Swagger document that describes the application that is deployed.

3.1.5.4.1 Get App

The **Get App** method returns a description of a deployed application with the specified name and version.

This method is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api//v1/app/name/<version>
```

name: The name of the deployed application.

version: The version of the deployed application.

The HTTP response message for the **Get App** method can include the following status codes.

HTTP status code	Description
200	The description of the deployed application was successfully returned.
404	The application cannot be found.

3.1.5.4.1.1 Request Body

The request body is empty.

3.1.5.4.1.2 Response Body

The response body is a JSON object that is defined by the **App Description** schema, as specified in section 2.2.5.3.1. For an example of a response body, see section 4.13.

3.1.5.4.1.3 Processing Details

This method returns a list of statuses for all applications of the specified name.

3.1.5.4.2 Get App Versions

The **Get App Versions** method returns a list of all versions of the named applications that are deployed in the cluster.

This method is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/app/<name>
```

name: The name of a deployed application.

The HTTP response message for the **Get App Versions** method can include the following status codes.

HTTP status code	Description
200	A list of all versions of the deployed application was successfully returned.

HTTP status code	Description
404	The application cannot be found.

3.1.5.4.2.1 Request Body

The request body is empty.

3.1.5.4.2.2 Response Body

The response body is an array of **app** descriptions in a JSON object that is defined by the **App Description** schema, as specified in section 2.2.5.3.1. For an example of a response body, see section 4.13.

3.1.5.4.2.3 Processing Details

This method returns the status of all versions of a specific **app**.

3.1.5.4.3 Get All Apps

The **Get All Apps** method is used to retrieve a list of the descriptions of all applications that are deployed in the cluster.

This method is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/app
```

The HTTP response message for the **Get All Apps** method can include the following status code.

HTTP status code	Description
200	The statuses of all the applications were retrieved successfully.

3.1.5.4.3.1 Request Body

The request body is empty.

3.1.5.4.3.2 Response Body

The response body is an array of **app** descriptions in a JSON object that is defined by the **App Description** schema, as specified in section 2.2.5.3.1. For an example of a response body, see section 4.13.

3.1.5.4.3.3 Processing Details

This method returns the description of all applications that are deployed in the cluster.

3.1.5.4.4 Create App

The **Create App** method is used to create an application in the cluster.

This method is invoked by sending a POST operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/app
```

The HTTP response message for the **Create App** method can include the following status codes.

HTTP status code	Description
201	The application was created successfully, and its status is available by using the **Location** header link.
400	The request is invalid.
409	An application with the specified version already exists.

3.1.5.4.4.1 (Updated Section) Request Body

The request body is a ZIP file that is created for use in deploying an application in the big data cluster. The ZIP file needs to contain a specification file that specifies the runtime for the application and the big data cluster-supported application that is to be deployed. For an example of a request body, see section 4.14.

The specification file, that MUST use the filename "spec.yaml", is written in YAML Ain't Markup Language (YAML) [YAML1.2] and contains the following elements.

Element	Description
name	The name of the runtime application.
version	The version of the application.
runtime	The language the application uses. The valid options are the following: ▪ R – Specifies to use the R runtime. ▪ Python – Specifies to use the Python runtime. ▪ SSIS – Specifies to use the SQL Server Integration Services (SSIS) runtime. ▪ MLEAP – Specifies to use the MLEAP runtime.
src	The full path to the location of the application.
entrypoint	The function that is called upon execution of the **Create App** method.
replicas	The number of replicas that are needed.
poolsize	The pool size that the application needs to scale.
inputs	The input parameters for the application that are defined by a mapping between parameter name and parameter type.
outputs	The output parameter types that the application returns.

3.1.5.4.4.2 Response Body

The response body is empty.

3.1.5.4.4.3 Processing Details

This method is used to create an **app** resource in the cluster.

3.1.5.4.5 Update App

The **Update App** method is used to update an application that is deployed in the cluster.

The **Update App** method is invoked by sending a PATCH operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/app
```

The HTTP response message for the **Update App** method can include the following status codes.

HTTP status code	Description
201	The application was updated. The update status is available by using a GET operation.
400	The request is invalid.
404	The specified application cannot be found.

3.1.5.4.5.1 Request Body

The request body is a ZIP file that is created for use in updating an application in the big data cluster. For details about the creation of this ZIP file, see section 3.1.5.4.4.1. For an example of a request body, see section 4.14.

3.1.5.4.5.2 Response Body

The response body is empty.

3.1.5.4.5.3 Processing Details

This method is used to update an already deployed application.

3.1.5.4.6 Delete App

The **Delete App** method is used to delete an application that is deployed in the cluster.

This method can be invoked by sending a DELETE operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/app/<name>/<version>
```

name: The name of the deployed application.

version: The version of the deployed application.

The HTTP response message for **Delete App** method can include the following status codes.

HTTP status code	Description
202	The request was accepted, and the application will be deleted.
404	The specified application cannot be found.

3.1.5.4.6.1 Request Body

The request body is empty.

3.1.5.4.6.2 Response Body

The response body is empty.

3.1.5.4.6.3 Processing Details

This method is used to delete an **app** resource in the cluster.

3.1.5.4.7 Run App

The **Run App** method is used to send a request to an application that is deployed in the cluster.

This method can be invoked by sending a POST operation to the following URI.

```
https://<clusterIp>:<appProxyPort>/api/v1/app/<name>/<version>/run
```

appProxyPort: The port that is defined by the user during control plane creation and exposed on the cluster for the **appproxy** resource.

name: The name of the deployed application.

version: The version of the deployed application.

The HTTP response message for **Run App** method can include the following status codes.

HTTP status code	Description
202	The request is accepted, and the application will be run with the passed-in parameters.
404	The specified application cannot be found.

3.1.5.4.7.1 (Updated Section) Request Header

The request MUST use Bearer authentication. This is done by including an **Authorization** HTTP header that contains a Bearer token. The header ~~should look~~looks like the following.

```
'Authorization: Bearer <token>'
```

token: The token string that is returned when a token is retrieved. For further details, see section 3.1.5.5.1.

3.1.5.4.7.2 Request Body

The request body is a JSON object that is defined by the **App Run Result** schema, as specified in section 2.2.5.3.2.

The elements in this JSON object match the names and types that are defined for the **input_params_defs** element in section 2.2.5.3.1.2.

For an example of a request body, see section 4.15.

3.1.5.4.7.3 Response Body

The response body is a JSON object that is defined by the **App Run Result** schema, as specified in section 2.2.5.3.2. For an example of a response body, see section 4.16.

3.1.5.4.7.4 Processing Details

This method is used to delete an **app** resource in the cluster.

3.1.5.4.8 Get App Swagger Document

The **Get App Swagger Document** method is used to retrieve a Swagger [Swagger2.0] document that can be passed into a Swagger editor to describe the application that is deployed in the cluster.

This method can be invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/app/<name>/<version>/swagger.json
```

name: The name of the deployed application.

version: The version of the deployed application.

The HTTP response message for the **Get App Swagger Document** method can include the following status codes.

HTTP status code	Description
202	The request is accepted, and the application's Swagger document is returned in the response.
404	The specified application cannot be found.

3.1.5.4.8.1 Request Body

The request body is empty.

3.1.5.4.8.2 Response Body

The response body is a JSON file that conforms to the Swagger 2.0 specification [Swagger2.0].

3.1.5.4.8.3 Processing Details

This method retrieves a Swagger document that can be used to describe a deployed **app** resource in the cluster.

3.1.5.5 Token

The **Token** API is a JWT [RFC7519] token that can be used as a form of authentication to use an application.

The methods of this API are invoked by using the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/token
```

The following methods can be performed by using HTTP operations on this resource.

Method	Section	Description
Create Token	3.1.5.5.1	Create and retrieve a token.

3.1.5.5.1 Create Token

The **Create Token** method is used to create a JWT Bearer token.

This method can be invoked by sending a POST operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/token
```

In addition to a Basic authentication header, this method can be accessed by using a negotiation [RFC4559] header.

The HTTP response message for the **Create Token** method can include the following status codes.

HTTP Status code	Description
200	The requested token was created.
400	The request is invalid.

3.1.5.5.1.1 Request Body

The request body is empty.

3.1.5.5.1.2 Response Body

The response is a JSON object that is defined by the **Token Response** schema, as specified in section 2.2.5.4.1. For an example of a response body, see section 4.17.

3.1.5.5.1.3 Processing Details

This method is used to create a JWT Bearer token.

3.1.5.6 Home Page

The **Home Page** API is used to determine whether the control plane service is listening for requests.

The methods of this API are invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/
```

The following methods can be performed by using HTTP operations on this resource.

Method	Section	Description
Get Home Page	3.1.5.6.1	Retrieve the controller home page.
Ping Controller	3.1.5.6.2	Determine whether the controller is responsive.
Info	3.1.5.6.3	Retrieve information about the cluster.

3.1.5.6.1 Get Home Page

The **Get Home Page** method is used to retrieve the home page of the controller. This API can be used to check whether the control plane service is running.

This method is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/
```

The HTTP response message for the **Get Home Page** method can include the following status code.

HTTP status code	Description
200	The home page was returned successfully.

3.1.5.6.1.1 Request Body

The request body is empty.

3.1.5.6.1.2 Response Body

The response body is empty.

3.1.5.6.1.3 Processing Details

None.

3.1.5.6.2 Ping Controller

The **Ping Controller** method is used to determine whether the control plane REST API is responsive.

This method is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/
```

The HTTP response message for the **Ping Controller** method can include the following status code.

HTTP status code	Description
200	The control plane is responsive.

3.1.5.6.2.1 Request Body

The request body is empty.

3.1.5.6.2.2 Response Body

The response body is a JSON object that is defined by the **Ping Response** schema, as specified in section 2.2.5.5.1. For an example of a response body, see section 4.18.

3.1.5.6.2.3 Processing Details

None.

3.1.5.6.3 Info

The **Info** method is used to retrieve information about the currently deployed cluster.

This method is invoked by sending a GET operation to the following URI.

```
https://<clusterIp>:<controllerPort>/api/v1/info
```

The HTTP response message for the **Info** method can include the following status code.

HTTP status code	Description
200	The Info page was returned successfully.

3.1.5.6.3.1 Request Body

The request body is empty.

3.1.5.6.3.2 Response Body

The response body is a JSON object that is defined by the **Info Response** schema, as specified in section 2.2.5.5.2. For an example of a response body, see section 4.19.

3.1.5.6.3.3 Processing Details

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Cluster Admin Details

The client role of this protocol is simply a pass-through and requires no additional timers or other state. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

4 Protocol Examples

4.1 (Updated Section) Check the Control Plane Status

The client sees whether the control plane is ready to accept creation of a big data cluster by sending the following request. If the control plane is ready, the GET operation ~~should return~~returns a ~~200~~message with a status of 200. The following is an example of a request that is sent by the **Get Control Status** method, as defined in section 3.1.5.2.1.

Request:

```
curl -k --request GET -u admin:***** https://localhost:30080/api/v1/~~ping~~control?all=<true>
```

4.2 (Updated Section) Create a Big Data Cluster

If the GET operation returns a message with a 200 status, the client can proceed to create a big data cluster by sending a request that uses the following sample configuration for a cluster named "mssql-cluster". The following is an example of a request body that is sent by the **Create Big Data Cluster** method, as defined in section 3.1.5.1.1.

Request:

```
curl -k -- request PATCH -u admin:***** https://localhost:30080/api/v1/bdc
```

Body:

```
{
  "apiVersion": "v1",
  "metadata": {
    "kind": "Cluster",
    "name": "mssql-cluster"
  },
  "spec": {
    "resources": {
      "nmnode-0": {
        "spec": {
          "replicas": 1
        }
      },
      "sparkhead": {
        "spec": {
          "replicas": 1
        }
      },
      "zookeeper": {
        "spec": {
          "replicas": 0
        }
      },
      "gateway": {
        "spec": {
          "replicas": 1,
          "endpoints": [
            {
              "name": "Knox",
              "dnsName": "",
              "serviceType": "NodePort",
              "port": 30443
            }
          ]
```

```
            }
          },
          "appproxy": {
            "spec": {
              "replicas": 1,
              "endpoints": [
                {
                  "name": "AppServiceProxy",
                  "dnsName": "",
                  "serviceType": "NodePort",
                  "port": 30778
                }
              ]
            }
          },
          "master": {
            "metadata": {
              "kind": "Pool",
              "name": "default"
            },
            "spec": {
              "type": "Master",
              "replicas": 3,
              "endpoints": [
                {
                  "name": "Master",
                  "dnsName": "",
                  "serviceType": "NodePort",
                  "port": 31433
                },
                {
                  "name": "MasterSecondary",
                  "dnsName": "",
                  "serviceType": "NodePort",
                  "port": 31436
                }
              ],
              "settings": {
                "sql": {
                  "hadr.enabled": "true"
                }
              }
            }
          },
          "compute-0": {
            "metadata": {
              "kind": "Pool",
              "name": "default"
            },
            "spec": {
              "type": "Compute",
              "replicas": 1
            }
          },
          "data-0": {
            "metadata": {
              "kind": "Pool",
              "name": "default"
            },
            "spec": {
              "type": "Data",
              "replicas": 2
            }
          },
          "storage-0": {
            "metadata": {
              "kind": "Pool",
              "name": "default"
            },
            "spec": {
```

```
          "type": "Storage",
          "replicas": 2,
          "settings": {
            "spark": {
              "includeSpark": "true"
            }
          }
        }
      }
    },
    "services": {
      "sql": {
        "resources": [
          "master",
          "compute-0",
          "data-0",
          "storage-0"
        ]
      },
      "hdfs": {
        "resources": [
          "nmnode-0",
          "zookeeper",
          "storage-0",
          "sparkhead"
        ],
        "settings": {
        }
      },
      "spark": {
        "resources": [
          "sparkhead",
          "storage-0"
        ],
        "settings": {
       }
      }
    }
  }
}
```

4.3 (Updated Section) Check on the Progress of the Big Data Cluster Deployment

The user can check the status of the creation of the big data cluster by sending the following request. After the status response is returned as ~~“~~"ready~~”,~~", the client can begin to use the big data cluster. The following is an example of a response body that is returned for the **Get Big Data Cluster Status** method, as defined in section 3.1.5.1.4.

Request:

```
curl -k -- request GET -u admin:***** https://localhost:30080/api/v1/bdc/status
```

Response:

```
{
  "bdcName": "mssql-cluster",
  "state": "ready",
  "healthStatus": "healthy",
  "details": null,
  "services": [
    {
      "serviceName": "sql",
      "state": "ready",
      "healthStatus": "healthy",
      "details": null,
      "resources": [
        {
          "resourceName": "master",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet master is healthy",
          "instances": null
        },
        {
          "resourceName": "compute-0",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet compute-0 is healthy",
          "instances": null
        },
        {
          "resourceName": "data-0",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet data-0 is healthy",
          "instances": null
        },
        {
          "resourceName": "storage-0",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet storage-0 is healthy",
          "instances": null
        }
      ]
    },
    {
      "serviceName": "hdfs",
      "state": "ready",
      "healthStatus": "healthy",
      "details": null,
      "resources": [
        {
          "resourceName": "nmnode-0",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet nmnode-0 is healthy",
          "instances": null
        },
        {
          "resourceName": "zookeeper",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet zookeeper is healthy",
          "instances": null
        },
        {
          "resourceName": "storage-0",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet storage-0 is healthy",
          "instances": null
        },
        {
          "resourceName": "sparkhead",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet sparkhead is healthy",
          "instances": null
        }
      ]
    },
    {
      "serviceName": "spark",
      "state": "ready",
      "healthStatus": "healthy",
      "details": null,
      "resources": [
        {
          "resourceName": "sparkhead",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet sparkhead is healthy",
          "instances": null
        },
        {
          "resourceName": "storage-0",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet storage-0 is healthy",
          "instances": null
        }
      ]
    },
    {
      "serviceName": "control",
      "state": "ready",
      "healthStatus": "healthy",
      "details": null,
      "resources": [
        {
          "resourceName": "controldb",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet controldb is healthy",
          "instances": null
        },
        {
          "resourceName": "control",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "ReplicaSet control is healthy",
          "instances": null
        },
        {
          "resourceName": "metricsdc",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "DaemonSet metricsdc is healthy",
          "instances": null
        },
        {
          "resourceName": "metricsui",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "ReplicaSet metricsui is healthy",
          "instances": null
        },
        {
          "resourceName": "metricsdb",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet metricsdb is healthy",
          "instances": null
        },
        {
          "resourceName": "logsui",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "ReplicaSet logsui is healthy",
          "instances": null
        },
        {
          "resourceName": "logsdb",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet logsdb is healthy",
          "instances": null
        },
        {
          "resourceName": "mgmtproxy",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "ReplicaSet mgmtproxy is healthy",
          "instances": null
        }
      ]
    },
    {
      "serviceName": "gateway",
      "state": "ready",
      "healthStatus": "healthy",
      "details": null,
      "resources": [
        {
          "resourceName": "gateway",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "StatefulSet gateway is healthy",
          "instances": null
        }
      ]
    },
    {
      "serviceName": "app",
      "state": "ready",
      "healthStatus": "healthy",
      "details": null,
      "resources": [
        {
          "resourceName": "appproxy",
          "state": "ready",
          "healthStatus": "healthy",
          "details": "ReplicaSet appproxy is healthy",
          "instances": null
        }
      ]
    }
  ]
}
```

4.4 Receive an Error in the Big Data Cluster Deployment

The following is an example of a response body that is returned for a failed request that uses the **Create Big Data Cluster** method, as defined in section 3.1.5.1.1.

```
{
  "code": 400,
  "reason": "INVALID JSON",
  "data": "The control plane service failed to parse the cluster specification."
}
```

4.5 Check on the Status of the Big Data Cluster

The following is an example of a response body that is returned for the **Get Big Data Cluster Status** schema, as defined in section 3.1.5.1.4.

```
{
    "bdcName": "bdc",
    "state": "ready",
    "healthStatus": "healthy",
    "details": null,
    "services": [
        {
            "serviceName": "sql",
            "state": "ready",
            "healthStatus": "healthy",
            "details": null,
            "resources": [
                {
                    "resourceName": "master",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet master is healthy",
                    "instances": null
                },
                {
                    "resourceName": "compute-0",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet compute-0 is healthy",
                    "instances": null
                },
                {
                    "resourceName": "data-0",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet data-0 is healthy",
                    "instances": null
                },
                {
                    "resourceName": "storage-0",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet storage-0 is healthy",
                    "instances": null
                }
            ]
        },
        {
            "serviceName": "hdfs",
            "state": "ready",
            "healthStatus": "healthy",
            "details": null,
            "resources": [
                {
                    "resourceName": "nmnode-0",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet nmnode-0 is healthy",
                    "instances": null
                },
                {
                    "resourceName": "zookeeper",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet zookeeper is healthy",
                    "instances": null
                },
                {
                    "resourceName": "storage-0",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet storage-0 is healthy",
                    "instances": null
                }
            ]
        },
        {
            "serviceName": "spark",
            "state": "ready",
            "healthStatus": "healthy",
            "details": null,
            "resources": [
                {
                    "resourceName": "sparkhead",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet sparkhead is healthy",
                    "instances": null
                },
                {
                    "resourceName": "storage-0",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet storage-0 is healthy",
                    "instances": null
                }
            ]
        },
        {
            "serviceName": "control",
            "state": "ready",
            "healthStatus": "healthy",
            "details": null,
            "resources": [
                {
                    "resourceName": "controldb",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": null,
                    "instances": null
                },
                {
                    "resourceName": "control",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": null,
                    "instances": null
                },
                {
                    "resourceName": "metricsdc",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "DaemonSet metricsdc is healthy",
                    "instances": null
                },
                {
                    "resourceName": "metricsui",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "ReplicaSet metricsui is healthy",
                    "instances": null
                },
                {
                    "resourceName": "metricsdb",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet metricsdb is healthy",
                    "instances": null
                },
                {
                    "resourceName": "logsui",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "ReplicaSet logsui is healthy",
                    "instances": null
                },
                {
                    "resourceName": "logsdb",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet logsdb is healthy",
                    "instances": null
                },
                {
                    "resourceName": "mgmtproxy",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "ReplicaSet mgmtproxy is healthy",
                    "instances": null
                }
            ]
        },
        {
            "serviceName": "gateway",
            "state": "ready",
            "healthStatus": "healthy",
            "details": null,
            "resources": [
                {
                    "resourceName": "gateway",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "StatefulSet gateway is healthy",
                    "instances": null
                }
            ]
        },
        {
            "serviceName": "app",
            "state": "ready",
            "healthStatus": "healthy",
            "details": null,
            "resources": [
                {
                    "resourceName": "appproxy",
                    "state": "ready",
                    "healthStatus": "healthy",
                    "details": "ReplicaSet appproxy is healthy",
                    "instances": null
                }
            ]
        }
    ]
}
```

4.6 (Updated Section) Receive the State and Configuration of the Big Data Cluster

The following is an example of a response body that is returned for the **Get Big Data Cluster Information** method, as defined in section 3.1.5.1.5.

```
   "state": "Ready",
   "spec": {
      "apiVersion": "v1",
      "metadata": {
         "kind": "BigDataCluster",
         "name": "mssql-cluster"
      },
      "spec": {
         "resources": {
            "gateway": {
               "clusterName": "mssql-cluster",
               "spec": {
                  "replicas": 1,
                  "docker": {
                     "registry": "mcr.contoso.com",
                     "repository": "mssql/bdc",
                     "imageTag": "latest",
                     "imagePullPolicy": "Always"
                  },
                  "storage": {
                     "data": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "15Gi"
                     },
                     "logs": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "10Gi"
                     }
                  },
                  "endpoints": [
                     {
                        "name": "Knox",
                        "serviceType": "NodePort",
                        "port": 30443,
                        "dynamicDnsUpdate": true
                     }
                  ],
                  "settings": {
                     "gateway": {
                        "gateway-site.gateway.httpclient.socketTimeout": "90s",
                        "gateway-site.sun.security.krb5.debug": "true"
                     }
                  }
               }
            },
            "appproxy": {
               "clusterName": "mssql-cluster",
               "spec": {
                  "replicas": 1,
                  "docker": {
                     "registry": "mcr.contoso.com",
                     "repository": "mssql/bdc",
                     "imageTag": "latest",
                     "imagePullPolicy": "Always"
                  },
                  "storage": {
                     "data": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "15Gi"
                     },
                     "logs": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "10Gi"
                     }
                  },
                  "endpoints": [
                     {
                        "name": "AppServiceProxy",
                        "serviceType": "NodePort",
                        "port": 30778,
                        "dynamicDnsUpdate": true
                     }
                  ],
                  "settings": {}
               }
            },
            "storage-0": {
               "clusterName": "mssql-cluster",
               "metadata": {
                  "kind": "Pool",
                  "name": "default"
               },
               "spec": {
                  "type": ~~4,~~"Storage",
                  "replicas": 2,
                  "docker": {
                     "registry": "mcr.contoso.com",
                     "repository": "mssql/bdc",
                     "imageTag": "latest",
                     "imagePullPolicy": "Always"
                  },
                  "storage": {
                     "data": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "15Gi"
                     },
                     "logs": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "10Gi"
                     }
                  },
                  "settings": {
                     "spark": {
                        "includeSpark": "true",
                        "yarn-site.yarn.nodemanager.resource.memory-mb": "18432",
                        "yarn-site.yarn.scheduler.maximum-allocation-vcores": "6",
                        "yarn-site.yarn.scheduler.maximum-allocation-mb": "18432",
                        "spark-defaults-conf.spark.executor.instances": "3",
                        "yarn-site.yarn.nodemanager.resource.cpu-vcores": "6",
                        "spark-defaults-conf.spark.executor.cores": "1",
                        "spark-defaults-conf.spark.driver.memory": "2g",
                        "spark-defaults-conf.spark.driver.cores": "1",
                        "yarn-site.yarn.scheduler.capacity.maximum-am-resource-percent":
"0.3",
                        "spark-defaults-conf.spark.executor.memory": "1536m",
                        "capacity-scheduler.yarn.scheduler.capacity.maximum-applications":
"10000",
                        "capacity-scheduler.yarn.scheduler.capacity.resource-calculator":
"org.apache.hadoop.yarn.util.resource.DominantResourceCalculator",
                        "capacity-scheduler.yarn.scheduler.capacity.root.queues": "default",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.capacity":
"100",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.user-limit-
factor": "1",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.maximum-
capacity": "100",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.state":
"RUNNING",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.maximum-
application-lifetime": "-1",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.default-
application-lifetime": "-1",
                        "capacity-scheduler.yarn.scheduler.capacity.node-locality-delay":
"40",
                        "capacity-scheduler.yarn.scheduler.capacity.rack-locality-additional-
delay": "-1",
                        "hadoop-env.HADOOP_HEAPSIZE_MAX": "2048",
                        "yarn-env.YARN_RESOURCEMANAGER_HEAPSIZE": "2048",
                        "yarn-env.YARN_NODEMANAGER_HEAPSIZE": "2048",
                        "mapred-env.HADOOP_JOB_HISTORYSERVER_HEAPSIZE": "2048",
                        "hive-env.HADOOP_HEAPSIZE": "2048",
                        "livy-conf.livy.server.session.timeout-check": "true",
                        "livy-conf.livy.server.session.timeout-check.skip-busy": "true",
                        "livy-conf.livy.server.session.timeout": "2h",
                        "livy-conf.livy.server.yarn.poll-interval": "500ms",
                        "livy-env.LIVY_SERVER_JAVA_OPTS": "-Xmx2g",
                        "spark-defaults-conf.spark.r.backendConnectionTimeout": "86400",
                        "spark-history-server-conf.spark.history.fs.cleaner.maxAge": "7d",
                        "spark-history-server-conf.spark.history.fs.cleaner.interval": "12h",
                        "spark-env.SPARK_DAEMON_MEMORY": "2g",
                        "yarn-site.yarn.log-aggregation.retain-seconds": "604800",
                        "yarn-site.yarn.nodemanager.log-aggregation.compression-type": "gz",
                        "yarn-site.yarn.nodemanager.log-aggregation.roll-monitoring-interval-
seconds": "3600",
                        "yarn-site.yarn.scheduler.minimum-allocation-mb": "512",
                        "yarn-site.yarn.scheduler.minimum-allocation-vcores": "1",
                        "yarn-site.yarn.nm.liveness-monitor.expiry-interval-ms": "180000"
                     },
                     "sql": {},
                     "hdfs": {
                        "hdfs-site.dfs.replication": "2",
                        "hdfs-site.dfs.ls.limit": "500",
                        "hdfs-env.HDFS_NAMENODE_OPTS": "-Dhadoop.security.logger=INFO,RFAS -
Xmx2g",
                        "hdfs-env.HDFS_DATANODE_OPTS": "-Dhadoop.security.logger=ERROR,RFAS -
Xmx2g",
                        "hdfs-env.HDFS_AUDIT_LOGGER": "INFO,RFAAUDIT",
                        "core-
site.hadoop.security.group.mapping.ldap.search.group.hierarchy.levels": "10",
                        "core-site.fs.permissions.umask-mode": "077",
                        "core-site.hadoop.security.kms.client.failover.max.retries": "20",
                        "kms-site.hadoop.security.kms.encrypted.key.cache.size": "500",
                        "zoo-cfg.tickTime": "2000",
                        "zoo-cfg.initLimit": "10",
                        "zoo-cfg.syncLimit": "5",
                        "zoo-cfg.maxClientCnxns": "60",
                        "zoo-cfg.minSessionTimeout": "4000",
                        "zoo-cfg.maxSessionTimeout": "40000",
                        "zoo-cfg.autopurge.snapRetainCount": "3",
                        "zoo-cfg.autopurge.purgeInterval": "0",
                        "zookeeper-java-env.JVMFLAGS": "-Xmx1G -Xms1G",
                        "zookeeper-log4j-properties.zookeeper.console.threshold": "INFO"
                     }
                  }
               }
            },
            "sparkhead": {
               "clusterName": "mssql-cluster",
               "spec": {
                  "replicas": 2,
                  "docker": {
                     "registry": "mcr.contoso.com",
                     "repository": "mssql/bdc",
                     "imageTag": "latest",
                     "imagePullPolicy": "Always"
                  },
                  "storage": {
                     "data": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "15Gi"
                     },
                     "logs": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "10Gi"
                     }
                  },
                  "settings": {
                     "hdfs": {
                        "hdfs-site.dfs.replication": "2",
                        "hdfs-site.dfs.ls.limit": "500",
                        "hdfs-env.HDFS_NAMENODE_OPTS": "-Dhadoop.security.logger=INFO,RFAS -
Xmx2g",
                        "hdfs-env.HDFS_DATANODE_OPTS": "-Dhadoop.security.logger=ERROR,RFAS -
Xmx2g",
                        "hdfs-env.HDFS_AUDIT_LOGGER": "INFO,RFAAUDIT",
                        "core-
site.hadoop.security.group.mapping.ldap.search.group.hierarchy.levels": "10",
                        "core-site.fs.permissions.umask-mode": "077",
                        "core-site.hadoop.security.kms.client.failover.max.retries": "20",
                        "kms-site.hadoop.security.kms.encrypted.key.cache.size": "500",
                        "zoo-cfg.tickTime": "2000",
                        "zoo-cfg.initLimit": "10",
                        "zoo-cfg.syncLimit": "5",
                        "zoo-cfg.maxClientCnxns": "60",
                        "zoo-cfg.minSessionTimeout": "4000",
                        "zoo-cfg.maxSessionTimeout": "40000",
                        "zoo-cfg.autopurge.snapRetainCount": "3",
                        "zoo-cfg.autopurge.purgeInterval": "0",
                        "zookeeper-java-env.JVMFLAGS": "-Xmx1G -Xms1G",
                        "zookeeper-log4j-properties.zookeeper.console.threshold": "INFO"
                     },
                     "spark": {
                        "yarn-site.yarn.nodemanager.resource.memory-mb": "18432",
                        "yarn-site.yarn.scheduler.maximum-allocation-vcores": "6",
                        "yarn-site.yarn.scheduler.maximum-allocation-mb": "18432",
                        "spark-defaults-conf.spark.executor.instances": "3",
                        "yarn-site.yarn.nodemanager.resource.cpu-vcores": "6",
                        "spark-defaults-conf.spark.executor.cores": "1",
                        "spark-defaults-conf.spark.driver.memory": "2g",
                        "spark-defaults-conf.spark.driver.cores": "1",
                        "yarn-site.yarn.scheduler.capacity.maximum-am-resource-percent":
"0.3",
                        "spark-defaults-conf.spark.executor.memory": "1536m",
                        "capacity-scheduler.yarn.scheduler.capacity.maximum-applications":
"10000",
                        "capacity-scheduler.yarn.scheduler.capacity.resource-calculator":
"org.apache.hadoop.yarn.util.resource.DominantResourceCalculator",
                        "capacity-scheduler.yarn.scheduler.capacity.root.queues": "default",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.capacity":
"100",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.user-limit-
factor": "1",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.maximum-
capacity": "100",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.state":
"RUNNING",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.maximum-
application-lifetime": "-1",
                        "capacity-scheduler.yarn.scheduler.capacity.root.default.default-
application-lifetime": "-1",
                        "capacity-scheduler.yarn.scheduler.capacity.node-locality-delay":
"40",
                        "capacity-scheduler.yarn.scheduler.capacity.rack-locality-additional-
delay": "-1",
                        "hadoop-env.HADOOP_HEAPSIZE_MAX": "2048",
                        "yarn-env.YARN_RESOURCEMANAGER_HEAPSIZE": "2048",
                        "yarn-env.YARN_NODEMANAGER_HEAPSIZE": "2048",
                        "mapred-env.HADOOP_JOB_HISTORYSERVER_HEAPSIZE": "2048",
                        "hive-env.HADOOP_HEAPSIZE": "2048",
                        "livy-conf.livy.server.session.timeout-check": "true",
                        "livy-conf.livy.server.session.timeout-check.skip-busy": "true",
                        "livy-conf.livy.server.session.timeout": "2h",
                        "livy-conf.livy.server.yarn.poll-interval": "500ms",
                        "livy-env.LIVY_SERVER_JAVA_OPTS": "-Xmx2g",
                        "spark-defaults-conf.spark.r.backendConnectionTimeout": "86400",
                        "spark-history-server-conf.spark.history.fs.cleaner.maxAge": "7d",
                        "spark-history-server-conf.spark.history.fs.cleaner.interval": "12h",
                        "spark-env.SPARK_DAEMON_MEMORY": "2g",
                        "yarn-site.yarn.log-aggregation.retain-seconds": "604800",
                        "yarn-site.yarn.nodemanager.log-aggregation.compression-type": "gz",
                        "yarn-site.yarn.nodemanager.log-aggregation.roll-monitoring-interval-
seconds": "3600",
                        "yarn-site.yarn.scheduler.minimum-allocation-mb": "512",
                        "yarn-site.yarn.scheduler.minimum-allocation-vcores": "1",
                        "yarn-site.yarn.nm.liveness-monitor.expiry-interval-ms": "180000"
                     }
                  }
               }
            },
            "data-0": {
               "clusterName": "mssql-cluster",
               "metadata": {
                  "kind": "Pool",
                  "name": "default"
               },
               "spec": {
                  "type": ~~3,~~"Data",
                  "replicas": 2,
                  "docker": {
                     "registry": "mcr.contoso.com",
                     "repository": "mssql/bdc",
                     "imageTag": "latest",
                     "imagePullPolicy": "Always"
                  },
                  "storage": {
                     "data": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "15Gi"
                     },
                     "logs": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "10Gi"
                     }
                  },
                  "settings": {
                     "sql": {}
                  }
               }
            },
            "compute-0": {
               "clusterName": "mssql-cluster",
               "metadata": {
                  "kind": "Pool",
                  "name": "default"
               },
               "spec": {
                  "type": ~~2,~~"Compute",
                  "replicas": 1,
                  "docker": {
                     "registry": "mcr.contoso.com",
                     "repository": "mssql/bdc",
                     "imageTag": "latest",
                     "imagePullPolicy": "Always"
                  },
                  "storage": {
                     "data": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "15Gi"
                     },
                     "logs": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "10Gi"
                     }
                  },
                  "settings": {
                     "sql": {}
                  }
               }
```

```
      },
      "master": {
         "clusterName": "mssql-cluster",
         "metadata": {
            "kind": "Pool",
            "name": "default"
         },
         "spec": {
            "type": ~~1,~~"Master",
            "replicas": 1,
            "docker": {
               "registry": "mcr.contoso.com",
               "repository": "mssql/bdc",
               "imageTag": "latest",
               "imagePullPolicy": "Always"
            },
            "storage": {
               "data": {
                  "className": "local-storage",
                  "accessMode": "ReadWriteOnce",
                  "size": "15Gi"
               },
               "logs": {
                  "className": "local-storage",
                  "accessMode": "ReadWriteOnce",
                  "size": "10Gi"
               }
            },
            "endpoints": [
               {
                  "name": "Master",
                  "serviceType": "NodePort",
                  "port": 31433,
                  "dynamicDnsUpdate": true
               }
            ],
            "settings": {
               "sql": {
                  "hadr.enabled": "false"
               }
            }
         }
      },
      "nmnode-0": {
         "clusterName": "mssql-cluster",
         "spec": {
            "replicas": 2,
            "docker": {
               "registry": "mcr.contoso.com",
               "repository": "mssql/bdc",
               "imageTag": "latest",
               "imagePullPolicy": "Always"
            },
            "storage": {
               "data": {
                  "className": "local-storage",
                  "accessMode": "ReadWriteOnce",
                  "size": "15Gi"
               },
               "logs": {
                  "className": "local-storage",
                  "accessMode": "ReadWriteOnce",
                  "size": "10Gi"
               }
            },
            "settings": {
               "hdfs": {
                  "hdfs-site.dfs.replication": "2",
                  "hdfs-site.dfs.ls.limit": "500",
                        "hdfs-env.HDFS_NAMENODE_OPTS": "-Dhadoop.security.logger=INFO,RFAS -
Xmx2g",
                        "hdfs-env.HDFS_DATANODE_OPTS": "-Dhadoop.security.logger=ERROR,RFAS -
Xmx2g",
                        "hdfs-env.HDFS_AUDIT_LOGGER": "INFO,RFAAUDIT",
                        "core-
site.hadoop.security.group.mapping.ldap.search.group.hierarchy.levels": "10",
                        "core-site.fs.permissions.umask-mode": "077",
                        "core-site.hadoop.security.kms.client.failover.max.retries": "20",
                        "kms-site.hadoop.security.kms.encrypted.key.cache.size": "500",
                        "zoo-cfg.tickTime": "2000",
                        "zoo-cfg.initLimit": "10",
                        "zoo-cfg.syncLimit": "5",
                        "zoo-cfg.maxClientCnxns": "60",
                        "zoo-cfg.minSessionTimeout": "4000",
                        "zoo-cfg.maxSessionTimeout": "40000",
                        "zoo-cfg.autopurge.snapRetainCount": "3",
                        "zoo-cfg.autopurge.purgeInterval": "0",
                        "zookeeper-java-env.JVMFLAGS": "-Xmx1G -Xms1G",
                        "zookeeper-log4j-properties.zookeeper.console.threshold": "INFO"
                     }
                  }
               }
            },
            "zookeeper": {
               "clusterName": "mssql-cluster",
               "spec": {
                  "replicas": 3,
                  "docker": {
                     "registry": "mcr.contoso.com",
                     "repository": "mssql/bdc",
                     "imageTag": "latest",
                     "imagePullPolicy": "Always"
                  },
                  "storage": {
                     "data": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "15Gi"
                     },
                     "logs": {
                        "className": "local-storage",
                        "accessMode": "ReadWriteOnce",
                        "size": "10Gi"
                     }
                  },
                  "settings": {
                     "hdfs": {
                        "hdfs-site.dfs.replication": "2",
                        "hdfs-site.dfs.ls.limit": "500",
                        "hdfs-env.HDFS_NAMENODE_OPTS": "-Dhadoop.security.logger=INFO,RFAS -
Xmx2g",
                        "hdfs-env.HDFS_DATANODE_OPTS": "-Dhadoop.security.logger=ERROR,RFAS -
Xmx2g",
                        "hdfs-env.HDFS_AUDIT_LOGGER": "INFO,RFAAUDIT",
                        "core-
site.hadoop.security.group.mapping.ldap.search.group.hierarchy.levels": "10",
                        "core-site.fs.permissions.umask-mode": "077",
                        "core-site.hadoop.security.kms.client.failover.max.retries": "20",
                        "kms-site.hadoop.security.kms.encrypted.key.cache.size": "500",
                        "zoo-cfg.tickTime": "2000",
                        "zoo-cfg.initLimit": "10",
                        "zoo-cfg.syncLimit": "5",
                        "zoo-cfg.maxClientCnxns": "60",
                        "zoo-cfg.minSessionTimeout": "4000",
                        "zoo-cfg.maxSessionTimeout": "40000",
                        "zoo-cfg.autopurge.snapRetainCount": "3",
                        "zoo-cfg.autopurge.purgeInterval": "0",
                        "zookeeper-java-env.JVMFLAGS": "-Xmx1G -Xms1G",
                        "zookeeper-log4j-properties.zookeeper.console.threshold": "INFO"
                     }
                  }
               }
            }
         },
         "services": {
            "sql": {
               "resources": [
                  "master",
                  "compute-0",
                  "data-0",
                  "storage-0"
               ],
               "settings": {}
            },
            "hdfs": {
               "resources": [
                  "nmnode-0",
                  "zookeeper",
                  "storage-0",
                  "sparkhead"
               ],
               "settings": {}
            },
            "spark": {
               "resources": [
                  "sparkhead",
                  "storage-0"
               ],
               "settings": {
                  "yarn-site.yarn.nodemanager.resource.memory-mb": "18432",
                  "yarn-site.yarn.scheduler.maximum-allocation-vcores": "6",
                  "yarn-site.yarn.scheduler.maximum-allocation-mb": "18432",
                  "spark-defaults-conf.spark.executor.instances": "3",
                  "yarn-site.yarn.nodemanager.resource.cpu-vcores": "6",
                  "spark-defaults-conf.spark.executor.cores": "1",
                  "spark-defaults-conf.spark.driver.memory": "2g",
                  "spark-defaults-conf.spark.driver.cores": "1",
                  "yarn-site.yarn.scheduler.capacity.maximum-am-resource-percent": "0.3",
                  "spark-defaults-conf.spark.executor.memory": "1536m"
               }
            }
         },
         "docker": {
            "registry": "mcr.contoso.com",
            "repository": "mssql/bdc",
            "imageTag": "latest",
            "imagePullPolicy": "Always"
         },
         "storage": {
            "data": {
               "className": "local-storage",
               "accessMode": "ReadWriteOnce",
               "size": "15Gi"
            },
            "logs": {
               "className": "local-storage",
               "accessMode": "ReadWriteOnce",
               "size": "10Gi"
            }
         }
      }
   }
}
```

4.7 Receive Service Status Information

The following is an example of a response body that is sent by the **Get Service Status** method, as defined in section 3.1.5.1.6.

```
{
  "serviceName": "sql",
  "state": "ready",
  "healthStatus": "healthy",
  "details": null,
  "resources": [
    {
      "resourceName": "master",
      "state": "ready",
      "healthStatus": "healthy",
      "details": "StatefulSet master is healthy",
      "instances": [
        {
          "instanceName": "master-0",
          "state": "running",
          "healthStatus": "healthy",
          "details": "Pod master-0 is healthy",
          "dashboards": {
            "nodeMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/master-
0/status/nodemetrics/ui",
            "sqlMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/master-
0/status/sqlmetrics/ui",
            "logsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/master-0/status/logs/ui"
          }
        }
      ]
    },
    {
      "resourceName": "compute-0",
      "state": "ready",
      "healthStatus": "healthy",
      "details": "StatefulSet compute-0 is healthy",
      "instances": [
        {
          "instanceName": "compute-0-0",
          "state": "running",
          "healthStatus": "healthy",
          "details": "Pod compute-0-0 is healthy",
          "dashboards": {
            "nodeMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/compute-0-
0/status/nodemetrics/ui",
            "sqlMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/compute-0-
0/status/sqlmetrics/ui",
            "logsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/compute-0-
0/status/logs/ui"
          }
        }
      ]
    },
    {
      "resourceName": "data-0",
      "state": "ready",
      "healthStatus": "healthy",
      "details": "StatefulSet data-0 is healthy",
      "instances": [
        {
          "instanceName": "data-0-0",
          "state": "running",
          "healthStatus": "healthy",
          "details": "Pod data-0-0 is healthy",
          "dashboards": {
            "nodeMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/data-0-
0/status/nodemetrics/ui",
            "sqlMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/data-0-
0/status/sqlmetrics/ui",
            "logsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/data-0-0/status/logs/ui"
          }
        },
        {
          "instanceName": "data-0-1",
          "state": "running",
          "healthStatus": "healthy",
          "details": "Pod data-0-1 is healthy",
          "dashboards": {
            "nodeMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/data-0-
1/status/nodemetrics/ui",
            "sqlMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/data-0-
1/status/sqlmetrics/ui",
            "logsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/data-0-1/status/logs/ui"
          }
        }
      ]
    },
    {
      "resourceName": "storage-0",
      "state": "ready",
      "healthStatus": "healthy",
      "details": "StatefulSet storage-0 is healthy",
      "instances": [
        {
          "instanceName": "storage-0-0",
          "state": "running",
          "healthStatus": "healthy",
          "details": "Pod storage-0-0 is healthy",
          "dashboards": {
            "nodeMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/storage-0-
0/status/nodemetrics/ui",
            "sqlMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/storage-0-
0/status/sqlmetrics/ui",
            "logsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/storage-0-
0/status/logs/ui"
          }
        },
        {
          "instanceName": "storage-0-1",
          "state": "running",
          "healthStatus": "healthy",
          "details": "Pod storage-0-1 is healthy",
          "dashboards": {
            "nodeMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/storage-0-
1/status/nodemetrics/ui",
            "sqlMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/storage-0-
1/status/sqlmetrics/ui",
            "logsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/storage-0-
1/status/logs/ui"
          }
        }
      ]
    }
  ]
}
```

4.8 Receive Service Resource Status Information

The following is an example of a response body that is sent by the **Get Service Resource Status** method, as defined in section 3.1.5.1.7.

```
{
  "resourceName": "master",
  "state": "ready",
  "healthStatus": "healthy",
  "details": "StatefulSet master is healthy",
  "instances": [
    {
      "instanceName": "master-0",
      "state": "running",
      "healthStatus": "healthy",
      "details": "Pod master-0 is healthy",
      "dashboards": {
        "nodeMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/master-
0/status/nodemetrics/ui",
        "sqlMetricsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/master-
0/status/sqlmetrics/ui",
        "logsUrl": "https://0.0.0.0:30777/api/v1/bdc/instances/master-0/status/logs/ui"
      }
    }
  ]
}
```

4.9 Upgrade the Big Data Cluster

The following is an example of a request body that is sent by the **Upgrade Big Data Cluster** method, as defined in section 3.1.5.1.9.

```
{
    "targetVersion": "latest",
    "targetRepository": "foo/bar/baz"
}
```

4.10 Receive a List of Information About All the Exposed Big Data Cluster Endpoints

The following is an example of a response body that is sent by the **Get All Big Data Cluster Endpoints** method, as defined in section 3.1.5.1.10.

```
[
   {
      "name":"gateway",
      "description":"Gateway to access HDFS files, Spark",
      "endpoint":"https://10.91.138.80:30443",
      "protocol":"https"
   },
   {
      "name":"spark-history",
      "description":"Spark Jobs Management and Monitoring Dashboard",
      "endpoint":"https://10.91.138.80:30443/gateway/default/sparkhistory",
      "protocol":"https"
   },
   {
      "name":"yarn-ui",
      "description":"Spark Diagnostics and Monitoring Dashboard",
      "endpoint":"https://10.91.138.80:30443/gateway/default/yarn",
      "protocol":"https"
   },
   {
      "name":"app-proxy",
      "description":"Application Proxy",
      "endpoint":"https://10.91.138.80:30778",
      "protocol":"https"
   },
   {
      "name":"mgmtproxy",
      "description":"Management Proxy",
      "endpoint":"https://10.91.138.80:30777",
      "protocol":"https"
   },
   {
      "name":"logsui",
      "description":"Log Search Dashboard",
      "endpoint":"https://10.91.138.80:30777/kibana",
      "protocol":"https"
   },
   {
      "name":"metricsui",
      "description":"Metrics Dashboard",
      "endpoint":"https://10.91.138.80:30777/grafana",
      "protocol":"https"
   },
   {
      "name":"controller",
      "description":"Cluster Management Service",
      "endpoint":"https://10.91.138.80:30080",
      "protocol":"https"
   },
   {
      "name":"sql-server-master",
      "description":"SQL Server Master Instance Front-End",
      "endpoint":"10.91.138.80,31433",
      "protocol":"tds"
   },
   {
      "name":"webhdfs",
      "description":"HDFS File System Proxy",
      "endpoint":"https://10.91.138.80:30443/gateway/default/webhdfs/v1",
      "protocol":"https"
   },
   {
      "name":"livy",
      "description":"Proxy for running Spark statements, jobs, applications",
      "endpoint":"https://10.91.138.80:30443/gateway/default/livy/v1",
      "protocol":"https"
   }
]
```

4.11 Expose a Big Data Cluster Endpoint

The following is an example of a response body that is sent by the **Get Big Data Cluster Endpoint** method, as defined in section 3.1.5.1.11.

```
    {
        "name":"gateway",
        "description":"Gateway to access HDFS files, Spark",
        "endpoint":"https://10.91.138.80:30443",
        "protocol":"https"
     }
```

4.12 Check Mount Status

The following is an example of a response body that is sent by the **Get Mount Status** method, as defined in section 3.1.5.3.1.

```
    {
        "mount": "/mnt/test",
        "remote": "abfs://foo.bar",
        "state": "Ready",
        "error": ""
    }
```

4.13 Receive an Application Specification

The following is an example of a response body that is sent by the **Get App** method, as defined in section 3.1.5.4.1.

```
{
  "name": "hello-py",
  "internal_name": "app1",
  "version": "v1",
  "input_param_defs": [
    {
      "name": "msg",
      "type": "str"
    },
    {
      "name": "foo",
      "type": "int"
    }
  ],
  "output_param_defs": [
    {
      "name": "out",
      "type": "str"
    }
  ],
  "state": "Ready",
  "links": {
    "app": "https://10.127.22.96:30777/api/app/hello-py/v1",
    "swagger": "https://10.127.22.96:30777/api/app/hello-py/v1/swagger.json"
  }
}
```

4.14 Deploy an Application

The following is an example of a request body that is sent by the **Create App** method, as defined in section 3.1.5.4.4, or the **Update App** method, as defined in section 3.1.5.4.5. To deploy an application that uses the Python runtime, the following files are bundled into a ZIP file.

```
#add.py
def add(x, y):
  result = x+y
  return result
result=add(x,y)


#spec.yaml
name: add-app
version: v1
runtime: Python
src: ./add.py
entrypoint: add
replicas: 1
poolsize: 1
inputs:
   x: int
   y: int
output:
   result: int
```

4.15 Run an Application

The following is an example of a request body that is sent by the **Run App** method, as defined in section 3.1.5.4.7.

```
{
  "x":5,
  "y": 37
}
```

4.16 Check on the Results of a Request to Run an Application

The following is an example of a response body that is sent by the **Run App** method, as defined in section 3.1.5.4.7.

```
{
  "success": true,
  "errorMessage": "",
  "outputParameters": {
    "result": 42
  },
    "outputFiles": {},
    "consoleOutput": "",
    "changedFiles": []
 }
```

4.17 Receive an Authentication Token

The following is an example of a response body that is sent by the **Create Token** method, as defined in section 3.1.5.5.1.

```
{
    "token_type": "Bearer",
    "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlIjpbImFwcCIsImNvbnRyb2xsZXIiLCJtZXRhZGF0YSJdLCJ
uYmYiOjE1NTQ5MTM0MjIsImV4cCI6MTU1NDk0OTQyMSwiaWF0IjoxNTU0OTEzNDIyLCJpc3MiOiJtc3NxbC5taWNyb3Nv
ZnQuY29tIiwiYXVkIjoibXNzcWwubWljcm9zb2Z0LmNvbSJ9.qKTG4PsGxDDFbjnZnE__3NWxEqCS9X9kc9B9IpR_UTY"
,
    "expires_in": 36000,
    "expires_on": 1554949422,
    "token_id":
"YsaMFgi1Re72fyfd7dZz6twfgjCy7jb49h1IVKkHMZt0QpqO7noNte6Veu0x8h3PD7msPDiR9z9drWyJvZQ6MPWD0wNz
mRrvCQ+v7dNQV8+9e9N4gZ7iE5vDP6z9hBgrggh8w4FeVSwCYYZiOG67OTzF2cnCfhQ8Gs+AjJWso3ga5lHqIKv34JNgO
ONp5Vpbu5iHGffZepgZ4jaIDIVd3ByogHtq+/c5pjdwLwoxH47Xuik0wNLLwiqktAWOv1cxDXOivkaGbJ6FDtJR4tPuNg
RLjNuz9iAZ16osNDyJ7oKyecnt4Tbt+XerwlyYYrjDWcW92qtpHX+kWnDrnmRn1g=="
}
```

4.18 Receive a Response Returned from a Ping on the Controller

The following is an example of a response body that is sent by the **Ping Controller** method, as defined in section 3.1.5.6.2.

```
{
    "code": 200,
    "message": "Controller is available."
}
```

4.19 Receive Cluster Information

The following is an example of a response body that is sent by the **Info** method, as defined in section 3.1.5.6.3.

```
{
   "version":"1.0",
   "buildTimestamp":"Thu Aug 01 03:32:28 GMT 2019"
}
```

5 Security

5.1 Security Considerations for Implementers

Unless specified otherwise, all authentication is done by way of Basic authentication.

The Control Plane Rest API protocol uses self-signed certificates. A user of this protocol needs to skip certificate verification when sending HTTP operations.

5.2 Index of Security Parameters

None.

6 Appendix A: Full JSON Schema

For ease of implementation, the following sections provide the full JSON schemas for this protocol.

Schema name	Section
Big Data Cluster	6.1
Storage	6.2
App	6.3
Token	6.4
Home	6.5

6.1 Big Data Cluster

The Big Data Cluster schemas are the following.

Schema name	Section
Big Data Cluster Spec Schema	6.1.1
Big Data Cluster Error Response Schema	6.1.2
Big Data Cluster Information Schema	6.1.3
Big Data Cluster Status Schema	6.1.4
Big Data Cluster Service Status Schema	6.1.5
Big Data Cluster Service Resource Status Schema	6.1.6
Big Data Cluster Upgrade Schema	6.1.7
Big Data Cluster Endpoints List Schema	6.1.8
Big Data Cluster Endpoint Schema	6.1.9

6.1.1 (Updated Section) Big Data Cluster Spec Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/root.json",
  "type": "object",
  "definitions": {
    "storage": {
      "required": [
        "logs",
        "data"
      ],
      "properties": {
        "data": {
          "$ref": "#/definitions/storageInfo"
        },
        "logs": {
          "$ref": "#/definitions/storageInfo"
        }
      }
    },
    "storageInfo": {
      "required": [
        "className",
        "accessMode",
        "size"
      ],
      "properties": {
        "className": {
          "type": "string"
        },
        "accessMode": {
          "enum": [
            "ReadWriteOnce",
            "ReadOnlyMany",
            "ReadWriteMany"
          ]
        },
        "size": {
          "type": "string",
          "example": "10Gi"
        }
      }
    },
    "docker": {
      "required": [
        "registry",
        "repository",
        "imageTag",
        "imagePullPolicy"
      ],
      "properties": {
        "registry": {
          "type": "string",
          "example": "repo.contoso.com"
        },
        "repository": {
          "type": "string"
        },
        "imageTag": {
          "type": "string",
          "example": "latest"
        },
        "imagePullPolicy": {
          "enum": [
            "Always",
            "IfNotPresent"
          ]
        }
      }
    },
    "security": {
      "type": "object",
      "required": [
        "activeDirectory",
        "allowRunAsRoot",
        "~~allowNodeMetrics~~allowNodeMetricsCollection",
        "~~allowPodMetrics~~allowPodMetricsCollection",
        "automountServiceAccountToken"
      ],
      "properties": {
        "activeDirectory": {
          "$id": "#/properties/activeDirectory",
          "type": "object",
          "required": [
            "useInternalDomain",
            "useSubdomain",
            "ouDistinguishedName",
            "dnsIpAddresses",
            "domainControllerFullyQualifiedDns",
            "domainDnsName",
            "clusterAdmins",
            "clusterUsers"
          ],
          "properties": {
            "useInternalDomain": {
              "$id": "#/properties/activeDirectory/properties/useInternalDomain",
              "type": "boolean"
            },
            "useSubdomain": {
              "$id": "#/properties/activeDirectory/properties/useSubdomain",
              "type": "boolean"
            },
            "ouDistinguishedName": {
              "$id": "#/properties/activeDirectory/properties/ouDistinguishedName",
              "type": "string"
            },
            "dnsIpAddresses": {
              "$id": "#/properties/activeDirectory/properties/dnsIpAddresses",
              "type": "array",
              "items": {
                "$id": "#/properties/activeDirectory/properties/dnsIpAddresses/items",
                "type": "string"
              }
            },
            "domainControllerFullyQualifiedDns": {
              "$id":
"#/properties/activeDirectory/properties/domainControllerFullyQualifiedDns",
              "type": "array",
              "items": {
                "$id":
"#/properties/activeDirectory/properties/domainControllerFullyQualifiedDns/items",
                "type": "string"
              }
            },
            "subdomain": {
              "$id": "#/properties/activeDirectory/properties/subdomain ",
              "type": "string"
            },
            "accountPrefix": {
              "$id": "#/properties/activeDirectory/properties/accountPrefix ",
              "type": "string"
            },
            "realm": {
              "$id": "#/properties/activeDirectory/properties/realm",
              "type": "string"
            },
            "domainDnsName": {
              "$id": "#/properties/activeDirectory/properties/domainDnsName",
              "type": "string"
            },
            "clusterAdmins": {
              "$id": "#/properties/activeDirectory/properties/clusterAdmins",
              "type": "array",
              "items": {
                "$id": "#/properties/activeDirectory/properties/clusterAdmins/items",
                "type": "string"
              }
            },
            "clusterUsers": {
              "$id": "#/properties/activeDirectory/properties/clusterUsers",
              "type": "array",
              "items": {
                "$id": "#/properties/activeDirectory/properties/clusterUsers/items",
                "type": "string"
              }
            },
            "appOwners": {
              "$id": "#/properties/activeDirectory/properties/appOwners",
              "type": "array",
              "items": {
                "$id": "#/properties/activeDirectory/properties/appOwners/items",
                "type": "string"
              }
            },
            "appUsers": {
              "$id": "#/properties/activeDirectory/properties/appUsers",
              "type": "array",
              "items": {
                "$id": "#/properties/activeDirectory/properties/appUsers/items",
                "type": "string"
              }
            }
          }
        },
        "allowRunAsRoot": {
          "$id": "#/properties/allowRunAsRoot",
          "type": "boolean"
        },
        "allowDumps": {
          "$id": "#/properties/allowDumps",
          "type": "boolean"
        },Collection
        "~~allowNodeMetrics~~allowNodeMetricsCollection": {
          "$id": "#/properties/~~allowNodeMetrics~~allowNodeMetricsCollection",
          "type": "boolean"
        },
        "~~allowPodMetrics~~allowPodMetricsCollection": {
          "$id": "#/properties/~~allowPodMetrics~~allowPodMetricsCollection",
          "type": "boolean"
        },
        "automountServiceAccountToken": {
          "$id": #/properties/automountServiceAccountToken",
          "type": "boolean"
        }
      }
    },
    "spark": {
      "patternProperties": {
        ~~"patternProperties": {~~
          ~~"([~~"^([A-Za-z~~_.-])+":~~_.-])+$": {
          "type": "string"
        }
      ~~  }~~},
      ~~}~~"additionalProperties": false
    },
    "hdfs": {
      "patternProperties": {
        ~~"patternProperties": {~~
          ~~"([~~"^([A-Za-z~~_.-])+":~~_.-])+$": {
          "type": "string"
        }
      ~~  }~~},
      ~~}~~"additionalProperties": false
    },
    "sql": {
      "patternProperties": {
        ~~"patternProperties": {~~
          ~~"([~~"^([A-Za-z~~_,-])+":~~_,-])+$": {
         "type": "string"
        }
      ~~  }~~},
      ~~}~~"additionalProperties": false
    },
    "gateway": {
      "patternProperties": {
        ~~"patternProperties": {~~
```

```
~~          "([~~"^([A-Za-z~~_.-])+":~~_.-])+$": {
          "type": "string"
        }
      ~~  }~~},
      ~~}~~"additionalProperties": false
    },
    "metadata": {
      "required": [
        "kind",
        "name"
      ],
      "properties": {
        ~~"([A-Za-z_.-])+":~~"kind": {
          "type": "string"
        ~~}~~},
      ~~}~~  "name": {
          "type": "string"
        },
      "additionalProperties": false
    },
    "replicas": {
      "type": "integer"
    }
  },
  "required": [
    "apiVersion",
    "metadata",
    "spec"
  ],
  "properties": {
    "apiVersion": {
      "$id": "#/properties/apiVersion",
      "const": "v1"
    },
    "metadata": {
      "$ref": "#/definitions/metadata"
    },
    "spec": {
      "$id": "#/properties/spec",
      "type": "object",
      "required": [
        "resources",
        "services"
      ],
      "properties": {
        "security": {
          "$ref": "#/definitions/security"
        },
        "resources": {
          "$id": "#/properties/spec/properties/resources",
          "type": "object",
          "required": [
            "sparkhead",
            "storage",
            "nmnode",
            "master",
            "compute",
            "appproxy",
            "zookeeper",
            "gateway",
            "data"
          ],
          "properties": {
            "sparkhead": {
              "$id": "#/properties/spec/properties/resources/properties/sparkhead",
              "type": "object",
              "required": [
                "spec"
              ],
              "properties": {
                "clusterName": {
                  "$id":
"#/properties/spec/properties/resources/properties/sparkhead/properties/clusterName",
                  "type": "string"
                },
                "security": {
                  "$ref": "#/definitions/security"
                },
                "spec": {
                  "$id":
"#/properties/spec/properties/resources/properties/sparkhead/properties/spec",
                  "type": "object",
                  "required": [
                    "replicas"
                  ],
                  "properties": {
                    "replicas": {
                      "$id":
"#/properties/spec/properties/resources/properties/sparkhead/properties/spec/properties/repli
cas",
                      "type": "integer"
                    },
                    "docker": {
                      "$ref": "#/definitions/docker"
                    },
                    "storage": {
                      "$ref": "#/definitions/storage"
                    },
                    "settings": {
                      "$id":
"#/properties/spec/properties/resources/properties/sparkhead/properties/spec/properties/setti
ngs",
                      "type": "object",
                      "required": [
                        "spark"
                      ],
                      "properties": {
                        "spark": {
                          "$ref": "#/definitions/spark"
                        }
                      }
                    }
                  }
                }
              }
            },
            "storage": {
              "$id": "#/properties/spec/properties/resources/properties/storage",
              "type": "object",
              "required": [
                "metadata",
                "spec"
              ],
              "properties": {
                "clusterName": {
                  "$id":
"#/properties/spec/properties/resources/properties/storage/properties/clusterName",
                  "type": "string"
                },
                "metadata": {
                  "$ref": "#/definitions/metadata"
                },
                "security": {
                  "$ref": "#/definitions/security"
                },
                "spec": {
                  "$id":
"#/properties/spec/properties/resources/properties/storage/properties/spec",
                  "type": "object",
                  "required": [
                    "type",
                    "replicas",
                    "settings"
                  ],
                  "properties": {
                    "type": {
                      "$id":
"#/properties/spec/properties/resources/properties/storage/properties/spec/properties/type",
                      "type": "~~integer~~string"
                    },
                    "replicas": {
                      "$id":
"#/properties/spec/properties/resources/properties/storage/properties/spec/properties/replica
s",
                      "type": "integer"
                    },
                    "docker": {
                      "$ref": "#/definitions/docker"
                    },
                    "storage": {
                      "$ref": "#/definitions/storage"
                    },
                    "settings": {
                      "$id":
"#/properties/spec/properties/resources/properties/storage/properties/spec/properties/setting
s",
                      "type": "object",
                      "required": [
                        "spark",
                        "sql",
                        "hdfs"
                      ],
                      "properties": {
                        "spark": {
                          "$ref": "#/definitions/spark"
                        },
                        "sql": {
                          "$ref": "#/definitions/sql"
                        },
                        "hdfs": {
                          "$ref": "#/definitions/hdfs"
                        }
                      }
                    }
                  }
                }
              }
            },
            "nmnode": {
              "$id": "#/properties/spec/properties/resources/properties/nmnode",
              "type": "object",
              "required": [
                "spec"
              ],
              "properties": {
                "clusterName": {
                  "$id":
"#/properties/spec/properties/resources/properties/nmnode/properties/clusterName",
                  "type": "string"
                },
                "security": {
                  "$ref": "#/definitions/security"
                },
                "spec": {
                  "$id":
"#/properties/spec/properties/resources/properties/nmnode/properties/spec",
                  "type": "object",
                  "required": [
                    "replicas"
                  ],
                  "properties": {
                    "replicas": {
                      "$id":
"#/properties/spec/properties/resources/properties/nmnode/properties/spec/properties/replicas
",
                      "type": "integer"
                    },
                    "docker": {
                      "$ref": "#/definitions/docker"
                    },
                    "storage": {
                      "$ref": "#/definitions/storage"
                    },
                    "settings": {
                      "$id":
"#/properties/spec/properties/resources/properties/nmnode/properties/spec/properties/settings
",
                      "type": "object",
                      "required": [
                        "hdfs"
                      ],
                      "properties": {
                        "hdfs": {
                          "$ref": "#/definitions/hdfs"
                        }
                      }
                    }
                  }
                }
              }
            },
            "master": {
              "$id": "#/properties/spec/properties/resources/properties/master",
              "type": "object",
              "required": [
                "metadata",
                "spec"
              ],
              "properties": {
                "clusterName": {
                  "$id":
"#/properties/spec/properties/resources/properties/master/properties/clusterName",
                  "type": "string"
                },
                "metadata": {
                  "$ref": "#/definitions/metadata"
                },
                "security": {
                  "$ref": "#/definitions/security"
                },
                "spec": {
                  "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec",
                  "type": "object",
                  "required": [
                    "type",
                    "replicas",
                    "endpoints"
                  ],
                  "properties": {
                    "type": {
                      "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/type",
                      "type": "~~integer~~string"
                    },
                    "replicas": {
                      "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/replicas
",
                      "type": "integer"
                    },
                    "docker": {
                      "$ref": "#/definitions/docker"
                    },
                    "storage": {
                      "$ref": "#/definitions/storage"
                    },
                    "endpoints": {
                      "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s",
                      "type": "array",
                      "items": {
                        "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items",
                        "type": "object",
                        "required": [
                          "name",
                          "serviceType",
                          "port"
                        ],
                        "properties": {
                          "name": {
                            "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/name",
                            "type": "string"
                          },
                          "serviceType": {
                            "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/serviceType",
                            "enum": [
                              "NodePort",
                              "LoadBalancer"
                            ]
                          },
                          "port": {
                            "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/port",
                            "type": "integer",
                            "examples": [
                              31433
                            ]
                          },
                          "dnsName": {
                            "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/~~port~~dnsName",
                            "type": "string"
                          },
                          "dynamicDnsUpdate": {
                            "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/~~port~~dynamicDnsUpdate",
                            "type": "boolean"
                          }
                        }
                      }
                    },
                    "settings": {
                      "$id":
"#/properties/spec/properties/resources/properties/master/properties/spec/properties/settings
",
                      "type": "object",
                      "required": [
                        "sql"
                      ],
                      "properties": {
                        "sql": {
                          "$ref": "#/definitions/sql"
                        }
                      }
                    }
                  }
                }
              }
            },
            "compute": {
              "$id": "#/properties/spec/properties/resources/properties/compute",
              "type": "object",
              "required": [
                "metadata",
                "spec"
              ],
              "properties": {
                "clusterName": {
                  "$id":
"#/properties/spec/properties/resources/properties/compute/properties/clusterName",
                  "type": "string"
                },
                "metadata": {
                  "$ref": "#/definitions/metadata"
                },
                "security": {
                  "$ref": "#/definitions/security"
                },
                "spec": {
                  "$id":
"#/properties/spec/properties/resources/properties/compute/properties/spec",
                  "type": "object",
                  "required": [
                    "type",
                    "replicas"
                  ],
                  "properties": {
                    "type": {
                      "$id":
"#/properties/spec/properties/resources/properties/compute/properties/spec/properties/type",
                      "type": "~~integer~~string"
                    },
                    "replicas": {
                      "$id":
"#/properties/spec/properties/resources/properties/compute/properties/spec/properties/replica
s",
                      "type": "integer"
                    },
                    "docker": {
                      "$ref": "#/definitions/docker"
                    },
                    "storage": {
                      "$ref": "#/definitions/storage"
                    },
                    "settings": {
                      "$id":
"#/properties/spec/properties/resources/properties/compute/properties/spec/properties/setting
s",
                      "type": "object",
                      "required": [
                        "sql"
                      ],
                      "properties": {
                        "sql": {
                          "$ref": "#/definitions/sql"
                        }
                      }
                    }
                  }
                }
              }
            },
            "appproxy": {
              "$id": "#/properties/spec/properties/resources/properties/appproxy",
              "type": "object",
              "required": [
                "spec"
              ],
              "properties": {
                "clusterName": {
                  "$id": 
"#/properties/spec/properties/resources/properties/appproxy/properties/clusterName",
                  "type": "string"
                },
                "security": {
                  "$ref": "#/definitions/security"
                },
                "spec": {
                  "$id": 
"#/properties/spec/properties/resources/properties/appproxy/properties/spec",
                  "type": "object",
                  "required": [
                    "replicas",
                    "endpoints"
                  ],
                  "properties": {
                    "replicas": {
                      "$id": 
"#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/replic
as",
                      "type": "integer"
                    },
                    "docker": {
                      "$ref": "#/definitions/docker"
                    },
                    "storage": {
                      "$ref": "#/definitions/storage"
                    },
                    "endpoints": {
                      "$id": 
"#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/endpoi
nts",
                      "type": "array",
                      "items": {
                        "$id": 
"#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/endpoi
nts/items",
                        "type": "object",
                        "required": [
                          "name",
                          "serviceType",
                          "port"
                        ],
                        "properties": {
                          "name": {
                            "$id": 
"#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/endpoi
nts/items/properties/name",
                            "const": "AppServiceProxy"
                          },
                          "serviceType": {
                            "$id": 
"#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/endpoi
nts/items/properties/serviceType",
                            "enum": [
                              "NodePort",
                              "LoadBalancer"
                            ]
                          },
                          "port": {
                            "$id":
"#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/endpoints/items/properties/port",
                            "type": "integer",
                            "examples": [
                              30778
                            ]
                          },
                          "dnsName": {
                            "$id":
"#/properties/spec/properties/resources/properties/~~master~~appproxy/properties/spec/properties/endpoints/items/properties/~~port~~dnsName",
                            "type": "string"
                          },
                          "dynamicDnsUpdate": {
                            "$id":
"#/properties/spec/properties/resources/properties/~~master~~appproxy/properties/spec/properties/endpoints/items/properties/~~port~~dynamicDnsUpdate",
                            "type": "boolean"
                          }
                        }
                      }
                    },
                    "settings": {
                      "$id":
"#/properties/spec/properties/resources/properties/appproxy/properties/spec/properties/settings",
                      "type": "object"
                    }
                  }
                }
              }
            },
            "zookeeper": {
              "$id": "#/properties/spec/properties/resources/properties/zookeeper",
              "type": "object",
              "required": [
                "spec"
              ],
              "properties": {
                "clusterName": {
                  "$id":
"#/properties/spec/properties/resources/properties/zookeeper/properties/clusterName",
                  "type": "string"
                },
                "security": {
                  "$ref": "#/definitions/security"
                },
                "spec": {
                  "$id":
"#/properties/spec/properties/resources/properties/zookeeper/properties/spec",
                  "type": "object",
                  "required": [
                    "replicas"
                  ],
                  "properties": {
                    "replicas": {
                      "$id":
"#/properties/spec/properties/resources/properties/zookeeper/properties/spec/properties/replicas",
                      "type": "integer"
                    },
                    "docker": {
                      "$ref": "#/definitions/docker"
                    },
                    "storage": {
                      "$ref": "#/definitions/storage"
                    },
                    "settings": {
                      "$id":
"#/properties/spec/properties/resources/properties/zookeeper/properties/spec/properties/setti
ngs",
                      "type": "object",
                      "required": [
                        "hdfs"
                      ],
                      "properties": {
                        "hdfs": {
                          "$id":
"#/properties/spec/properties/resources/properties/zookeeper/properties/spec/properties/setti
ngs/properties/hdfs",
                          "type": "object"
                        }
                      }
                    }
                  }
                }
              }
            },
            "gateway": {
              "$id": "#/properties/spec/properties/resources/properties/gateway",
              "type": "object",
              "required": [
                "spec"
              ],
              "properties": {
                "clusterName": {
                  "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/clusterName",
                  "type": "string"
                },
                "security": {
                  "$ref": "#/definitions/security"
                },
                "spec": {
                  "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec",
                  "type": "object",
                  "required": [
                    "replicas",
                    "endpoints"
                  ],
                  "properties": {
                    "replicas": {
                      "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/replica
s",
                      "type": "integer"
                    },
                    "docker": {
                      "$ref": "#/definitions/docker"
                    },
                    "storage": {
                      "$ref": "#/definitions/storage"
                    },
                    "endpoints": {
                      "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoin
ts",
                      "type": "array",
                      "items": {
                        "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoin
ts/items",
                        "type": "object",
                        "required": [
                          "name",
                          "serviceType",
                          "port"
                        ],
                        "properties": {
                          "name": {
                            "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoin
ts/items/properties/name",
                            "const": "Knox"
                          },
                          "serviceType": {
                            "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoin
ts/items/properties/serviceType",
                            "enum": [
                              "NodePort",
                              "LoadBalancer"
                            ]
                          },
                          "port": {
                            "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoin
ts/items/properties/port",
                            "type": "integer"
                          },
                          "dnsName": {
                            "$id":
"#/properties/spec/properties/resources/properties/~~master~~gateway/properties/spec/properties/e
ndpoints/items/properties/~~port~~dnsName",
                            "type": "string"
                          },
                          "dynamicDnsUpdate": {
                            "$id":
"#/properties/spec/properties/resources/properties/~~master~~gateway/properties/spec/properties/e
ndpoints/items/properties/~~port~~dynamicDnsUpdate",
                            "type": "boolean"
                          }
                        }
                      }
                    },
                    "settings": {
                      "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/setting
s",
                      "type": "object"
                    }
                  }
                }
              }
            },
            "spark": {
              "$id": "#/properties/spec/properties/resources/properties/spark",
              "type": "object",
              "required": [
                "metadata",
                "spec"
              ],
              "properties": {
                "clusterName": {
                  "$id":
"#/properties/spec/properties/resources/properties/spark/properties/clusterName",
                  "type": "string"
                },
                "security": {
                  "$ref": "#/definitions/security"
                },
                "metadata": {
                  "$ref": "#/definitions/metadata"
                },
                "spec": {
                  "$id":
"#/properties/spec/properties/resources/properties/spark/properties/spec",
                  "type": "object",
                  "required": [
                    "type",
                    "replicas"
                  ],
                  "properties": {
                    "type": {
                      "$id":
"#/properties/spec/properties/resources/properties/spark/properties/spec/properties/type",
                      "type": "~~integer~~string"
                    },
                    "replicas": {
                      "$id":
"#/properties/spec/properties/resources/properties/spark/properties/spec/properties/replicas"
,
                      "type": "integer"
                    },
                    "docker": {
                      "$ref": "#/definitions/docker"
                    },
                    "storage": {
                      "$ref": "#/definitions/storage"
                    },
                    "settings": {
                      "$id":
"#/properties/spec/properties/resources/properties/spark/properties/spec/properties/settings"
,
                      "type": "object",
                      "required": [
                        "spark",
                        "hdfs"
                      ],
                      "properties": {
                        "spark": {
                          "$ref": "#/definitions/spark"
                        },
                        "hdfs": {
                          "$ref": "#/definitions/hdfs"
                        }
                      }
                    }
                  }
                }
              }
            },
            "data": {
              "$id": "#/properties/spec/properties/resources/properties/data",
              "type": "object",
              "required": [
                "metadata",
                "spec"
              ],
              "properties": {
                "clusterName": {
                  "$id":
"#/properties/spec/properties/resources/properties/data/properties/clusterName",
                  "type": "string"
                },
                "security": {
                  "$ref": "#/definitions/security"
                },
                "metadata": {
                  "$ref": "#/definitions/metadata"
                },
                "spec": {
                  "$id":
"#/properties/spec/properties/resources/properties/data/properties/spec",
                  "type": "object",
                  "required": [
                    "type",
                    "replicas"
                  ],
                  "properties": {
                    "type": {
                      "$id":
"#/properties/spec/properties/resources/properties/data/properties/spec/properties/type",
                      "type": "~~integer~~string"
                    },
                    "replicas": {
                      "$id":
"#/properties/spec/properties/resources/properties/data/properties/spec/properties/replicas",
                      "type": "integer"
                    },
                    "docker": {
                      "$ref": "#/definitions/docker"
                    },
                    "storage": {
                      "$ref": "#/definitions/storage"
                    },
                    "settings": {
                      "$id":
"#/properties/spec/properties/resources/properties/data/properties/spec/properties/settings",
                      "type": "object",
                      "required": [
                        "sql"
                      ],
                      "properties": {
                        "sql": {
                          "$ref": "#/definitions/sql"
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        },
        "services": {
          "$id": "#/properties/spec/properties/services",
          "type": "object",
          "required": [
            "sql",
            "hdfs",
            "spark"
          ],
          "properties": {
            "sql": {
              "$id": "#/properties/spec/properties/services/properties/sql",
              "type": "object",
              "required": [
                "resources"
              ],
              "properties": {
                "resources": {
                  "$id":
"#/properties/spec/properties/services/properties/sql/properties/resources",
                  "type": "array",
                  "items": [
                    {
                      "const": "master"
                    },
                    {
                      "const": "compute"
                    },
                    {
                      "const": "data"
                    },
                    {
                      "const": "storage"
                    }
                  ]
                },
                "settings": {
                  "$ref": "#/definitions/sql"
                }
              }
            },
            "hdfs": {
              "$id": "#/properties/spec/properties/services/properties/hdfs",
              "type": "object",
              "required": [
                "resources"
              ],
              "properties": {
                "resources": {
                  "$id": 
"#/properties/spec/properties/services/properties/hdfs/properties/resources",
                  "type": "array",
                  "items": [
                    {
                      "const": "nmnode"
                    },
                    {
                      "const": "zookeeper"
                    },
                    {
                      "const": "storage"
                    },
                    {
                      "const": "sparkhead"
                    },
                    {
                      "const": "spark"
                    }
                  ]
                },
                "settings": {
                  "$ref": "#/definitions/hdfs"
                }
              }
            },
            "spark": {
              "$id": "#/properties/spec/properties/services/properties/spark",
              "type": "object",
              "required": [
                "resources",
                "settings"
              ],
              "properties": {
                "resources": {
                  "$id": 
"#/properties/spec/properties/services/properties/spark/properties/resources",
                  "type": "array",
                  "items": [
                    {
                      "const": "sparkhead"
                    },
                    {
                      "const": "storage"
                    },
                    {
                      "const": "spark"
                    }
                  ]
                },
                "settings": {
                  "$ref": "#/definitions/spark"
                }
              }
              }
            }
          },
          "docker": {
            "$ref": "#/definitions/docker"
          },
          "storage": {
            "$ref": "#/definitions/storage"
          }
        }
      }
    }
  }
```

6.1.2 Big Data Cluster Error Response Schema

```
  {
    "$schema": "http://json-schema.org/draft-07/schema#",
    "type": "object",
    "title": "The Root Schema",
    "required": [
      "code",
      "reason",
      "data"
    ],
    "properties": {
      "code": {
        "$id": "#/properties/code",
        "type": "integer",
        "title": "The Code Schema",
        "default": 0,
        "examples": [
          500
        ]
      },
      "reason": {
        "$id": "#/properties/reason",
        "type": "string",
        "default": "",
        "examples": [
          "An unexpected exception occurred."
        ]
      },
      "data": {
        "$id": "#/properties/data",
        "type": "string",
        "default": "",
        "examples": [
          "Null reference exception"
        ]
      }
    }
  }
```

6.1.3 (Updated Section) Big Data Cluster Information Schema

```
  {
    "$schema": "http://json-schema.org/draft-07/schema#",
    "type": "object",
    "required": [
~~      "code",~~
      "state",
      "spec"
    ],
    "properties": {
~~      "code": {~~
        ~~      "$id": "#/properties/code",~~
        ~~      "type": "integer"~~
        ~~    },~~
      "state": {
        "$id": "#/properties/state",
        "type": "string",
        "title": "The State Schema"
      },
      "spec": {
        "$id": "#/properties/spec",
        "type": "string"
      }
    }
  }
```

6.1.4 Big Data Cluster Status Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "required": [
    "bdcName",
    "state",
    "healthStatus",
    "details",
    "services"
  ],
  "properties": {
    "bdcName": {
      "type": "string"
    },
    "state": {
      "type": "string"
    },
    "healthStatus": {
      "type": "string"
    },
    "details": {
      "type": "string"
    },
    "services": {
      "type": "array",
      "title": "The Services Schema",
      "items": {
        "type": "object",
        "title": "The Items Schema",
        "required": [
          "serviceName",
          "state",
          "healthStatus",
          "details",
          "resources"
        ],
        "properties": {
          "serviceName": {
            "type": "string"
          },
          "state": {
            "type": "string"
          },
          "healthStatus": {
            "type": "string"
          },
          "details": {
            "type": "string"
          },
          "resources": {
            "type": "array",
                "title": "The Resources Schema",
                "items": {
                  "type": "object",
                  "title": "The Items Schema",
                  "required": [
                    "resourceName",
                    "state",
                    "healthStatus",
                    "details",
                    "instances"
                  ],
                  "properties": {
                    "resourceName": {
                      "type": "string"
                    },
                    "state": {
                      "type": "string"
                    },
                    "healthStatus": {
                      "type": "string"
                    },
                    "details": {
                      "type": "string"
                    },
                    "instances": {
                      "type": "array",
                      "title": "The Instances Schema",
                      "items": {
                        "type": "object",
                        "title": "The Items Schema",
                        "required": [
                          "instanceName",
                          "state",
                          "healthStatus",
                          "details",
                          "dashboards"
                        ],
                        "properties": {
                          "instanceName": {
                            "type": "string"
                          },
                          "state": {
                            "type": "string"
                          },
                          "healthStatus": {
                            "type": "string"
                          },
                          "details": {
                            "type": "string"
                          },
                          "dashboards": {
                            "type": "object",
                            "title": "The Dashboards Schema",
                            "required": [
                              "nodeMetricsUrl",
                              "sqlMetricsUrl",
                              "logsUrl"
                            ],
                            "properties": {
                              "nodeMetricsUrl": {
                                "type": "string",
                                "examples": [

    "https://0.0.0.0:30777/api/v1/bdc/instances/master/status/nodemetrics/ui"
                                ]
                              },
                              "sqlMetricsUrl": {
                                "type": "string",
                                "examples": [
    "https://0.0.0.0:30777/api/v1/bdc/instances/master/status/sqlmetrics/ui"
                                ]
                              },
                              "logsUrl": {
                                "type": "string",
                                "examples": [

    "https://0.0.0.0:30777/api/v1/bdc/instances/master/status/logs/ui"
                                ]
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
```

6.1.5 Big Data Cluster Service Status Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/root.json",
  "type": "object",
  "required": [
    "serviceName",
    "state",
    "healthStatus",
    "details",
    "resources"
  ],
  "properties": {
    "serviceName": {
      "type": "string"
    },
    "state": {
      "type": "string"
    },
    "healthStatus": {
      "type": "string"
    },
    "details": {
      "type": "string"
    },
    "resources": {
      "$id": "#/properties/resources",
      "type": "array",
      "title": "The Resources Schema",
      "items": {
        "$id": "#/properties/resources/items",
        "type": "object",
        "title": "The Items Schema",
        "required": [
          "resourceName",
          "state",
          "healthStatus",
          "details",
          "instances"
        ],
        "properties": {
          "resourceName": {
            "type": "string"
          },
          "state": {
            "type": "string"
          },
          "healthStatus": {
            "type": "string"
          },
          "details": {
            "type": "string"
          },
          "instances": {
            "type": "array",
            "title": "The Instances Schema",
            "items": {
              "type": "object",
              "title": "The Items Schema",
              "required": [
                "instanceName",
                "state",
                "healthStatus",
                "details",
                "dashboards"
              ],
              "properties": {
                "instanceName": {
                  "type": "string"
                },
                "state": {
                  "type": "string"
                },
                "healthStatus": {
                  "type": "string"
                },
                "details": {
                  "type": "string"
                },
                "dashboards": {
                  "type": "object",
                  "title": "The Dashboards Schema",
                  "required": [
                    "nodeMetricsUrl",
                    "sqlMetricsUrl",
                    "logsUrl"
                  ],
                  "properties": {
                    "nodeMetricsUrl": {
                      "type": "string"
                    },
                    "sqlMetricsUrl": {
                      "type": "string"
                    },
                    "logsUrl": {
                      "type": "string"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

6.1.6 Big Data Cluster Service Resource Status Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/root.json",
  "type": "object",
  "title": "The Root Schema",
  "required": [
    "resourceName",
    "state",
    "healthStatus",
    "details",
    "instances"
  ],
  "properties": {
    "resourceName": {
      "$id": "#/properties/resourceName",
      "type": "string"
    },
    "state": {
      "$id": "#/properties/state",
      "type": "string"
    },
    "healthStatus": {
      "$id": "#/properties/healthStatus",
      "type": "string"
    },
    "details": {
      "$id": "#/properties/details",
      "type": "string"
    },
    "instances": {
      "$id": "#/properties/instances",
      "type": "array",
      "items": {
        "type": "object",
        "title": "The Items Schema",
        "required": [
          "instanceName",
          "state",
          "healthStatus",
          "details",
          "dashboards"
        ],
        "properties": {
          "instanceName": {
            "type": "string"
          },
          "state": {
            "type": "string"
          },
          "healthStatus": {
            "type": "string"
          },
          "details": {
            "type": "string"
          },
          "dashboards": {
            "type": "object",
            "title": "The Dashboards Schema",
            "required": [
              "nodeMetricsUrl",
              "sqlMetricsUrl",
              "logsUrl"
            ],
            "properties": {
              "nodeMetricsUrl": {
                "type": "string"
              },
              "sqlMetricsUrl": {
                "type": "string"
              },
              "logsUrl": {
                "type": "string"
              }
            }
          }
        }
      }
    }
  }
}
```

6.1.7 Big Data Cluster Upgrade Schema

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "type": "object",
    "required": [
        "targetVersion",
        "targetRepository"
    ],
    "properties": {
        "targetVersion": {
            "$id": "#/properties/targetVersion",
            "type": "string",
        },
        "targetRepository": {
            "$id": "#/properties/targetRepository",
            "type": "string",
        }
    }
}
```

6.1.8 Big Data Cluster Endpoints List Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/root.json",
  "type": "array",
  "title": "The Root Schema",
  "items": {
    "$id": "#/items",
    "type": "object",
    "required": [
      "name",
      "description",
      "endpoint",
      "protocol"
    ],
    "properties": {
      "name": {
        "$id": "#/items/properties/name",
        "type": "string",
        "title": "The Name Schema"
      },
      "description": {
        "$id": "#/items/properties/description",
        "type": "string"
      },
      "endpoint": {
        "$id": "#/items/properties/endpoint",
        "type": "string"
      },
      "protocol": {
        "enum": [
          "https",
          "tds"
        ]
      }
    }
  }
}
```

6.1.9 Big Data Cluster Endpoint Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/root.json",
  "type": "object",
  "required": [
    "name",
    "description",
    "endpoint",
    "protocol"
  ],
  "properties": {
    "name": {
      "$id": "#/properties/name",
      "type": "string",
      "title": "The Name Schema"
    },
    "description": {
      "$id": "#/properties/description",
      "type": "string"
    },
    "endpoint": {
      "$id": "#/properties/endpoint",
      "type": "string"
    },
    "protocol": {
      "enum": [
        "https",
        "tds"
      ]
    }
  }
}
```

6.2 Storage

The Storage schemas are the following.

Schema name	Section
Storage Response Schema	6.2.1

6.2.1 Storage Response Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "title": "Storage Response Schema",
  "required": [
    "mount",
    "remote",
    "state",
    "error"
  ],
  "properties": {
    "mount": {
      "$id": "#/properties/mount",
      "type": "string"
    },
    "remote": {
      "$id": "#/properties/remote",
      "type": "string"
    },
    "state": {
      "$id": "#/properties/state",
      "enum": [
        "Initial",
        "Creating",
        "WaitingForCreate",
        "Updating",
        "WaitingForUpdate",
        "Ready",
        "Deleting",
        "WaitingForDelete",
        "Deleted",
        "Error"
      ]
    },
    "error": {
      "$id": "#/properties/error",
      "type": "string"
    }
  }
}
```

6.3 App

The App schemas are the following.

Schema name	Section
App Description Schema	6.3.1
App Run Result Schema	6.3.2

6.3.1 (Updated Section) App Description Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "~~array~~object",
  "title": "App Result Schema",
  "definitions": {
    "link": {
      "type": "object",
      "properties": {
        "*": {
          "type": "string"
        }
      }
    },
    "parameter": {
      "required": [
        "name",
        "type"
      ],
      "properties": {
        "name": {
          "type": "string"
        },
        "type": {
          "enum": [
            "str",
            "int",
            "dataframe",
            "data.frame",
            "float",
            "matrix",
            "vector",
            "bool"
          ]
        }
      }
    }
  },
  "items": {
    "$id": "#/items",
    "type": "object",
    "required": [
      "name",
      "internal_name",
      "version",
      "input_param_defs",
      "output_param_defs",
      "state",
      "links"
    ],
    "properties": {
      "name": {
        "$id": "#/items/properties/name",
        "type": "string"
      },
      "internal_name": {
        "$id": "#/items/properties/internal_name",
        "type": "string"
      },
      "version": {
        "$id": "#/items/properties/version",
        "type": "string"
      },
      "input_param_defs": {
        "$id": "#/items/properties/input_param_defs",
        "type": "array",
        "description": "Array of input parameters for the deployed app",
        "items": {
          "$ref": "#/definitions/parameter"
        }
      },
      "output_param_defs": {
        "$id": "#/items/properties/output_param_defs",
        "type": "array",
        "items": {
          "$ref": "#/definitions/parameter"
        }
      },
      "state": {
        "$id": "#/items/properties/state",
        "enum": [
          "Initial",
          "Creating",
          "WaitingForCreate",
          "Updating",
          "WaitingForUpdate",
          "Ready",
          "Deleting",
          "WaitingForDelete",
          "Deleted",
          "Error"
        ]
      },
      "links": {
        "$id": "#/properties/links",
        "type": "object",
        "required": [
          "app",
          "swagger"
        ],
        "properties": {
          "app": {
            "$id": "#/properties/links/properties/app",
            "type": "string"
          },
          "swagger": {
            "$id": "#/properties/links/properties/swagger",
            "type": "string"
          }
        }
      }
    }
  }
}
```

6.3.2 App Run Result Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "required": [
    "success",
    "errorMessage",
    "outputFiles",
    "consoleOutput",
    "changedFiles"
  ],
  "properties": {
    "success": {
      "$id": "#/properties/success",
      "type": "boolean"
    },
    "errorMessage": {
      "$id": "#/properties/errorMessage",
      "type": "string"
    },
    "outputParameters": {
      "$id": "#/properties/outputParameters",
      "type": "object",
      "properties": {
        "result": {
          "$id": "#/properties/outputParameters/properties/result",
          "enum": [
            "str",
            "int",
            "dataframe",
            "data.frame",
            "float",
            "matrix",
            "vector",
            "bool"
          ]
        }
      }
    },
    "outputFiles": {
      "$id": "#/properties/outputFiles",
      "type": "object"
    },
    "consoleOutput": {
      "$id": "#/properties/consoleOutput",
      "type": "string"
    },
    "changedFiles": {
      "$id": "#/properties/changedFiles",
      "type": "array"
    }
  }
}
```

6.4 Token

Schema name	Section
Token Response Schema	6.4.1

6.4.1 Token Response Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "required": [
    "token_type",
    "access_token",
    "expires_in",
    "expires_on",
    "token_id"
  ],
  "properties": {
    "token_type": {
      "$id": "#/properties/token_type",
      "type": "string"
    },
    "access_token": {
      "$id": "#/properties/access_token",
      "type": "string"
    },
    "expires_in": {
      "$id": "#/properties/expires_in",
      "type": "integer"
    },
    "expires_on": {
      "$id": "#/properties/expires_on",
      "type": "integer"
    },
    "token_id": {
      "$id": "#/properties/token_id",
      "type": "string"
    }
  }
}
```

6.5 Home

The Home schemas are the following.

Schema name	Section
Ping Response Schema	6.5.1
Info Response Schema	6.5.2

6.5.1 Ping Response Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/root.json",
  "type": "object",
  "title": "The Root Schema",
  "required": [
    "code",
    "message"
  ],
  "properties": {
    "code": {
      "$id": "#/properties/code",
      "const": 200
    },
    "message": {
      "$id": "#/properties/message",
      "const": "Controller is available."
    }
  }
}
```

6.5.2 Info Response Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://example.com/root.json",
  "type": "object",
  "title": "The Root Schema",
  "required": [
    "version",
    "buildTimestamp"
  ],
  "properties": {
    "version": {
      "$id": "#/properties/version",
      "type": "string"
    },
    "buildTimestamp": {
      "$id": "#/properties/buildTimestamp",
      "type": "string"
    }
  }
}
```

7 (Updated Section) Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft SQL Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 2.2.5.1.1.1.4: The **allowRunAsRoot** element is not supported by SQL Server earlier than SQL Server 2019 Cumulative Update 5 (CU5). In SQL Server 2019 Cumulative Update 2 (CU2), SQL Server 2019 Cumulative Update 3 (CU3), and SQL Server 2019 Cumulative Update 4 (CU4), the equivalent element **privileged** was used. SQL Server 2019 and SQL Server 2019 Cumulative Update 1 (CU1) do not support this functionality.

<2> Section 2.2.5.1.1.1.4: The **allowDumps** element is not supported by SQL Server earlier than SQL Server 2019 CU5.

<3> Section 2.2.5.1.1.1.4: The **~~allowNodeMetrics~~allowNodeMetricsCollection** element is not supported by SQL Server earlier than SQL Server 2019 CU5.

<4> Section 2.2.5.1.1.1.4: The **~~allowPodMetrics~~allowPodMetricsCollection** element is not supported by SQL Server earlier than SQL Server 2019 CU5.

~~<5~~<5> Section 2.2.5.1.1.1.4: The **automountServiceAccountToken** element is not supported by SQL Server earlier than SQL Server 2019 Cumulative Update 6 (CU6).

<6> Section 2.2.5.1.1.1.4.1: The **security.activeDirectory** element is not supported by SQL Server earlier than SQL Server 2019 CU2.

<7> Section 2.2.5.1.1.1.4.1: The **security.activeDirectory.useInternalDomain** element is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 ~~Cumulative Update 1 (~~CU1~~),~~, the equivalent element **security.useInternalDomain** was used.

<8> Section 2.2.5.1.1.1.4.1: The **useSubdomain** element is not supported by SQL Server earlier than SQL Server 2019 CU5.

<9> Section 2.2.5.1.1.1.4.1: The **security.activeDirectory.ouDistinguishedName** element is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent element **security.ouDistinguishedName** was used.

<10> Section 2.2.5.1.1.1.4.1~~<9> Section 2.2.5.1.1.1.4.1:~~: The **subdomain** element is not supported by SQL Server earlier than SQL Server 2019 CU5.

<11> Section 2.2.5.1.1.1.4.1: The **accountPrefix** element is not supported by SQL Server earlier than SQL Server 2019 CU5.

<12> Section 2.2.5.1.1.1.4.1: The **security.activeDirectory.realm** element is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent element **security.realm** was used.

<13> Section 2.2.5.1.1.1.4.1: The **security.activeDirectory.domainDnsName** element is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent element **security.domainDnsName** was used.

<14> Section 2.2.5.1.1.1.4.1.1: The **security.activeDirectory.dnsIpAddresses** element is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent element **security.dnsIpAddresses** was used.

<15> Section 2.2.5.1.1.1.4.1.2: The **security.activeDirectory.domainControllerFullyQualifiedDns** element is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent element **security.domainControllerFullyQualifiedDns** was used.

<16> Section 2.2.5.1.1.1.4.1.3: The **security.activeDirectory.clusterAdmins** element is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent element **security.clusterAdmins** was used.

<17> Section 2.2.5.1.1.1.4.1.4: The **security.activeDirectory.clusterUsers** element is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent element **security.clusterUsers** was used.

<18> Section 2.2.5.1.1.1.4.1.5: The **security.activeDirectory.appOwner** element is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent element **security.appOwner** was used.

<19> Section 2.2.5.1.1.1.4.1.6: The **security.activeDirectory.appUsers** element is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent element **security.appUsers** was used.

8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
2.2.5.1.1.1.2.1 accessMode	Clarified definition of 'ReadWriteOnce'.	Major
2.2.5.1.1.1.4 security	Updated schema to add details about 'automountServiceAccountToken' element, to revise element names to 'allowNodeMetricsCollection' and 'allowPodMetricsCollection", and to clarify 'allowRunAsRoot' product behavior note.	Major
2.2.5.1.1.1.5 spark	Clarified 'patternProperties' definition.	Major
2.2.5.1.1.1.6 hdfs	Clarified 'patternProperties' definition.	Major
2.2.5.1.1.1.7 sql	Clarified 'patternProperties' definition.	Major
2.2.5.1.1.1.8 gateway	Clarified 'patternProperties' definition.	Major
2.2.5.1.1.1.9 metadata	Revised schema to remove 'patternProperties' and add definitions for 'kind' and 'name' in schema.	Major
2.2.5.1.1.3.1.4.1 spec	Changed definition of property 'type' element from 'integer' to 'string' and clarified that its value is 'Master'.	Major
2.2.5.1.3 Big Data Cluster Information Schema	Removed 'code' from schema.	Major
2.2.5.3.1 App Description Schema	Replaced 'array' with 'object' for type of schema.	Major
6.1.1 Big Data Cluster Spec Schema	Updated schema to add details about new 'automountServiceAccountToken' element, to revise element names to 'allowNodeMetricsCollection' and 'allowPodMetricsCollection", and to make numerous other revisions to the schema.	Major
6.1.3 Big Data Cluster Information Schema	Revised schema.	Major

Section	Description	Revision class
6.3.1 App Description Schema	Revised schema.	Major

9 Index