

[MS-CPREST-Diff]:

Control Plane REST API

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

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. |

Table of Contents

| | | |
|-------------|--|-----------|
| 1 | Introduction | 7 |
| 1.1 | (Updated Section) Glossary | 7 |
| 1.2 | References | 9 |
| 1.2.1 | Normative References | 9 |
| 1.2.2 | (Updated Section) Informative References | 10 |
| 1.3 | (Updated Section) Overview | 10 |
| 1.4 | Relationship to Other Protocols | 11 |
| 1.5 | (Updated Section) Prerequisites/Preconditions | 12 |
| 1.6 | Applicability Statement | 12 |
| 1.7 | Versioning and Capability Negotiation | 12 |
| 1.8 | Vendor-Extensible Fields | 12 |
| 1.9 | Standards Assignments | 12 |
| 2 | Messages | 13 |
| 2.1 | Transport | 13 |
| 2.2 | Common Data Types | 13 |
| 2.2.1 | Namespaces | 13 |
| 2.2.2 | HTTP Methods | 13 |
| 2.2.3 | HTTP Headers | 13 |
| 2.2.3.1 | X-RequestID | 13 |
| 2.2.4 | URI Parameters | 13 |
| 2.2.4.1 | clusterIp | 14 |
| 2.2.4.2 | controllerPort | 14 |
| 2.2.4.3 | bdcName | 14 |
| 2.2.5 | (Updated Section) JSON Elements | 14 |
| 3 | Protocol Details | 19 |
| 3.1 | Common Details | 19 |
| 3.1.1 | Abstract Data Model | 19 |
| 3.1.2 | Timers | 19 |
| 3.1.3 | Initialization | 19 |
| 3.1.4 | Higher-Layer Triggered Events | 19 |
| 3.1.5 | Message Processing Events and Sequencing Rules | 19 |
| 3.1.5.1 | (Updated Section) Big Data Cluster | 20 |
| 3.1.5.1.1 | Create BDC | 26 |
| 3.1.5.1.1.1 | (Updated Section) Request Body | 27 |
| 3.1.5.1.1.2 | Response Body | 29 |
| 3.1.5.1.1.3 | Processing Details | 30 |
| 3.1.5.1.2 | Delete BDC | 30 |
| 3.1.5.1.2.1 | Request Body | 30 |
| 3.1.5.1.2.2 | Response Body | 30 |
| 3.1.5.1.2.3 | Processing Details | 30 |
| 3.1.5.1.3 | Get BDC Logs | 30 |
| 3.1.5.1.3.1 | Request Body | 31 |
| 3.1.5.1.3.2 | Response Body | 31 |
| 3.1.5.1.3.3 | Processing Details | 31 |
| 3.1.5.1.4 | Get BDC Status | 31 |
| 3.1.5.1.4.1 | Request Body | 31 |
| 3.1.5.1.4.2 | Response Body | 31 |
| 3.1.5.1.4.3 | Processing Details | 34 |
| 3.1.5.1.5 | Get BDC Information | 34 |
| 3.1.5.1.5.1 | Request Body | 35 |
| 3.1.5.1.5.2 | (Updated Section) Response Body | 35 |
| 3.1.5.1.5.3 | Processing Details | 44 |
| 3.1.5.1.6 | Get Service Status | 44 |

| | | |
|--------------|-----------------------------------|----|
| 3.1.5.1.6.1 | Request Body | 45 |
| 3.1.5.1.6.2 | Response Body | 45 |
| 3.1.5.1.6.3 | Processing Details | 47 |
| 3.1.5.1.7 | Get Service Resource Status | 47 |
| 3.1.5.1.7.1 | Request Body | 47 |
| 3.1.5.1.7.2 | Response Body | 48 |
| 3.1.5.1.7.3 | Processing Details | 48 |
| 3.1.5.1.8 | Redirect to Metrics Link | 48 |
| 3.1.5.1.8.1 | Request Body | 49 |
| 3.1.5.1.8.2 | Response Body | 49 |
| 3.1.5.1.8.3 | Processing Details | 49 |
| 3.1.5.1.9 | Upgrade BDC | 49 |
| 3.1.5.1.9.1 | Request Body | 49 |
| 3.1.5.1.9.2 | Response Body | 50 |
| 3.1.5.1.9.3 | Processing Details | 50 |
| 3.1.5.1.10 | Get All BDC Endpoints | 50 |
| 3.1.5.1.10.1 | Request Body | 50 |
| 3.1.5.1.10.2 | Response Body | 50 |
| 3.1.5.1.10.3 | Processing Details | 51 |
| 3.1.5.1.11 | Get BDC Endpoint | 51 |
| 3.1.5.1.11.1 | Request Body | 52 |
| 3.1.5.1.11.2 | Response Body | 52 |
| 3.1.5.1.11.3 | Processing Details | 52 |
| 3.1.5.2 | Control | 52 |
| 3.1.5.2.1 | Get Control Status | 53 |
| 3.1.5.2.1.1 | Request Body | 53 |
| 3.1.5.2.1.2 | Response Body | 53 |
| 3.1.5.2.1.3 | Processing Details | 53 |
| 3.1.5.2.2 | Upgrade Control | 53 |
| 3.1.5.2.2.1 | Request Body | 54 |
| 3.1.5.2.2.2 | Response Body | 54 |
| 3.1.5.2.2.3 | Processing Details | 54 |
| 3.1.5.2.3 | Redirect to Metrics Link | 54 |
| 3.1.5.2.3.1 | Request Body | 55 |
| 3.1.5.2.3.2 | Response Body | 55 |
| 3.1.5.2.3.3 | Processing Details | 55 |
| 3.1.5.2.4 | Get Control Resource Status | 55 |
| 3.1.5.2.4.1 | Request Body | 56 |
| 3.1.5.2.4.2 | Response Body | 56 |
| 3.1.5.2.4.3 | Processing Details | 56 |
| 3.1.5.3 | Storage | 56 |
| 3.1.5.3.1 | Get Mount Status | 56 |
| 3.1.5.3.1.1 | Request Body | 57 |
| 3.1.5.3.1.2 | Response Body | 57 |
| 3.1.5.3.1.3 | Processing Details | 57 |
| 3.1.5.3.2 | Get All Mount Statuses | 57 |
| 3.1.5.3.2.1 | Request Body | 57 |
| 3.1.5.3.2.2 | Response Body | 57 |
| 3.1.5.3.2.3 | Processing Details | 58 |
| 3.1.5.3.3 | Create Mount | 58 |
| 3.1.5.3.3.1 | Request Body | 58 |
| 3.1.5.3.3.2 | Response Body | 58 |
| 3.1.5.3.3.3 | Processing Details | 58 |
| 3.1.5.3.4 | Delete Mount | 58 |
| 3.1.5.3.4.1 | Request Body | 59 |
| 3.1.5.3.4.2 | Response Body | 59 |
| 3.1.5.3.4.3 | Processing Details | 59 |
| 3.1.5.3.5 | Refresh Mount | 59 |

| | | |
|-------------|------------------------------------|----|
| 3.1.5.3.5.1 | Request Body | 59 |
| 3.1.5.3.5.2 | Response Body | 60 |
| 3.1.5.3.5.3 | Processing Details | 60 |
| 3.1.5.4 | (Updated Section) App Deploy | 60 |
| 3.1.5.4.1 | Get App | 61 |
| 3.1.5.4.1.1 | Request Body | 62 |
| 3.1.5.4.1.2 | Response Body | 62 |
| 3.1.5.4.1.3 | Processing Details | 62 |
| 3.1.5.4.2 | Get App Versions | 63 |
| 3.1.5.4.2.1 | Request Body | 63 |
| 3.1.5.4.2.2 | Response Body | 63 |
| 3.1.5.4.2.3 | Processing Details | 63 |
| 3.1.5.4.3 | Get All Apps | 63 |
| 3.1.5.4.3.1 | Request Body | 63 |
| 3.1.5.4.3.2 | Response Body | 64 |
| 3.1.5.4.3.3 | Processing Details | 64 |
| 3.1.5.4.4 | Create App | 64 |
| 3.1.5.4.4.1 | Request Body | 64 |
| 3.1.5.4.4.2 | Response Body | 64 |
| 3.1.5.4.4.3 | Processing Details | 64 |
| 3.1.5.4.5 | Update App | 64 |
| 3.1.5.4.5.1 | Request Body | 65 |
| 3.1.5.4.5.2 | Response Body | 65 |
| 3.1.5.4.5.3 | Processing Details | 65 |
| 3.1.5.4.6 | Delete App | 65 |
| 3.1.5.4.6.1 | Request Body | 65 |
| 3.1.5.4.6.2 | Response Body | 65 |
| 3.1.5.4.6.3 | Processing Details | 65 |
| 3.1.5.4.7 | Run App | 66 |
| 3.1.5.4.7.1 | Request Header | 66 |
| 3.1.5.4.7.2 | Request Body | 66 |
| 3.1.5.4.7.3 | Response Body | 66 |
| 3.1.5.4.7.4 | Processing Details | 67 |
| 3.1.5.4.8 | Get App Swagger Document | 67 |
| 3.1.5.4.8.1 | Request Body | 67 |
| 3.1.5.4.8.2 | Response Body | 67 |
| 3.1.5.4.8.3 | Processing Details | 67 |
| 3.1.5.5 | Token | 67 |
| 3.1.5.5.1 | Create Token | 68 |
| 3.1.5.5.1.1 | Request Body | 68 |
| 3.1.5.5.1.2 | Response Body | 68 |
| 3.1.5.5.1.3 | Processing Details | 69 |
| 3.1.5.6 | Home Page | 69 |
| 3.1.5.6.1 | Get Home Page | 69 |
| 3.1.5.6.1.1 | Request Body | 70 |
| 3.1.5.6.1.2 | Response Body | 70 |
| 3.1.5.6.1.3 | Processing Details | 70 |
| 3.1.5.6.2 | Ping Controller | 70 |
| 3.1.5.6.2.1 | Request Body | 70 |
| 3.1.5.6.2.2 | Response Body | 70 |
| 3.1.5.6.2.3 | Processing Details | 70 |
| 3.1.5.6.3 | Info | 70 |
| 3.1.5.6.3.1 | Request Body | 71 |
| 3.1.5.6.3.2 | Response Body | 71 |
| 3.1.5.6.3.3 | Processing Details | 71 |
| 3.1.6 | Timer Events | 71 |
| 3.1.7 | Other Local Events | 71 |
| 3.2 | Cluster Admin Details | 71 |

| | | |
|----------|---|------------|
| 4 | Protocol Examples | 72 |
| 4.1 | Request to Check Control Plane Status | 72 |
| 4.2 | Request to Create Big Data Cluster | 72 |
| 4.3 | Check on Big Data Cluster Deployment Progress | 74 |
| 5 | Security | 78 |
| 5.1 | Security Considerations for Implementers | 78 |
| 5.2 | Index of Security Parameters | 78 |
| 6 | Appendix A: Full JSON Schema | 79 |
| 6.1 | Big Data Cluster | 79 |
| 6.1.1 | (Updated Section) Big Data Cluster Spec Schema | 79 |
| 6.1.2 | Big Data Cluster Error Response Schema | 96 |
| 6.1.3 | Big Data Cluster Information Schema | 96 |
| 6.1.4 | Big Data Cluster Status Schema | 97 |
| 6.1.5 | Big Data Cluster Service Status Schema | 99 |
| 6.1.6 | Big Data Cluster Service Resource Status Schema | 101 |
| 6.1.7 | Big Data Cluster Endpoints List Schema | 102 |
| 6.1.8 | Big Data Cluster Endpoint Schema | 102 |
| 6.2 | Storage | 103 |
| 6.2.1 | Storage Response Schema | 103 |
| 6.3 | App | 104 |
| 6.3.1 | App Description Schema | 104 |
| 6.3.2 | App Run Result Schema | 105 |
| 6.4 | Token | 106 |
| 6.4.1 | Token Response Schema | 106 |
| 6.5 | Home | 107 |
| 6.5.1 | Ping Response Schema | 107 |
| 6.5.2 | Info Response Schema | 107 |
| 7 | (Updated Section) Appendix B: Product Behavior | 108 |
| 8 | Change Tracking | 110 |
| 9 | Index | 111 |

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 (Updated Section) 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 YARN:** A resource manager and job scheduler that is used by Apache Hadoop.~~

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

app proxy: A pod that is deployed in the control plane and provides users with the ability to interact with the applications deployed in the big data cluster. **Also called application proxy.**

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: 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.

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.

~~**Spark driver:** A process that maintains the context for the Apache Spark application and schedules work to the Spark executors in the cluster.~~

~~**Spark executor:** A worker node process that runs the individual tasks in an Apache Spark application.~~

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", January 2013, <http://json-schema.org/>

[Kubernetes] The Kubernetes Authors, "Kubernetes Documentation", version 1.14, <https://kubernetes.io/docs/home/>

[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>

[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>

[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 (Updated Section) Informative References

[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.

~~Depending on the configuration sent to the Control Plane REST API, the user can customize the topography of the cluster.~~

~~This protocol defines the deployment of a 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 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.

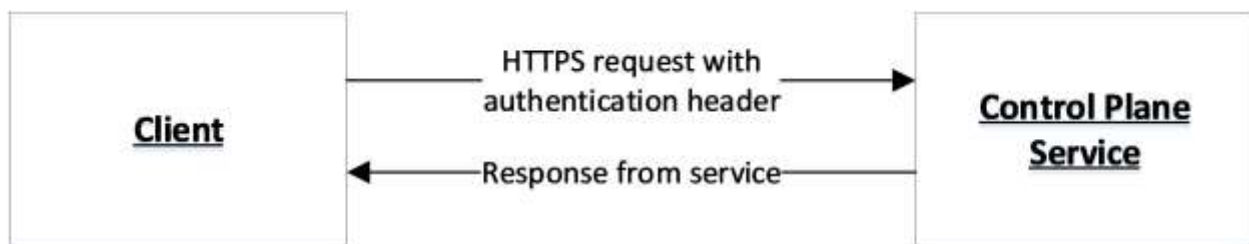


Figure 1: Communication flow

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.

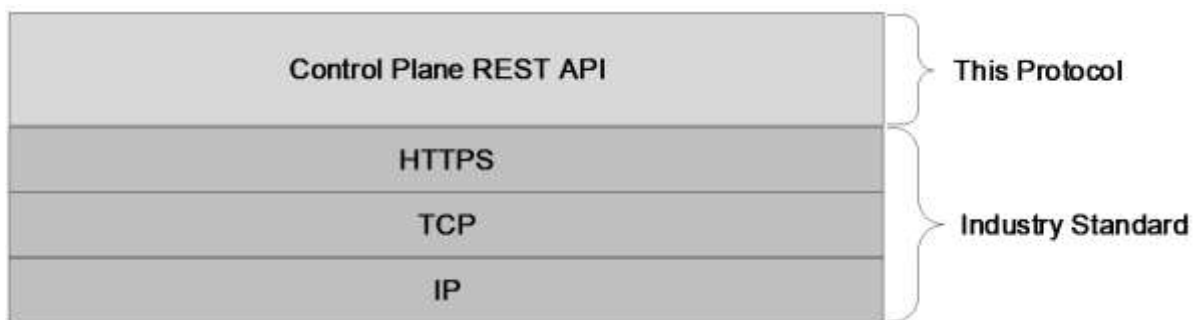


Figure 2: Protocol layering

1.5 (Updated Section) 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 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 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 properties [JSON-Schema] in any request or response.

The following table summarizes a set of common URI parameters [RFC3986] that are defined in this protocol.

| 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 |

| URI parameters | Description |
|----------------|---|
| | the cluster for the controller. |
| 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 to be deployed.

2.2.5 (Updated Section) JSON Elements

Data structures that are defined in this section flow through this protocol in JSON format and are defined in JSON schema [JSON-Schema].

This protocol defines the following common JSON schema properties. All properties are required.

| Property | Description |
|-----------------|---|
| metadata | Structured data that provides information about the JSON object. |
| metadata.kind | Structured data that describes the type of object that is to be created. |
| metadata.name | Structured data that provides the name of the component that is to be created. |
| docker | Structured data that defines where to find the docker image. |
| docker.registry | Specifies the registry where a docker image is located. |
| type | Enumeration that is used to define the type of a resource. The possible values are mapped as follows: 0 – other: Any big data cluster resource that is not defined by another type in this |

| Property | Description |
|------------------------|---|
| | <p>enumeration.</p> <p>1 – master instance: A big data cluster resource that manages connectivity and provides an entry point to make scale-out queries and machine learning services in the cluster.</p> <p>2 – compute pool: A big data cluster resource that consists of a group of one or more pods that provides scale-out computational resources for the cluster.</p> <p>3 – data pool: A big data cluster resource that consists of a group of pods that provides persistent storage for the cluster.</p> <p>4 – storage pool: A big data cluster resource that consists of a group of disks that is aggregated and managed as a single unit and used to ingest and store data from HDFS.</p> <p>5 – sql pool: A big data cluster resource that consists of multiple master instances. If a resource with this type is included, a resource with a type value set to 1 MUST not be present.</p> <p>6 – spark pool: A resource that consists of components that are related to Apache Spark [ApacheSpark].</p> |
| docker.repository | Specifies the repository where a docker image is located. |
| docker.imageTag | Specifies the image tag for the docker image to be pulled. |
| docker.imagePullPolicy | Specifies the image pull policy for the docker image. |
| storage | Structured data that defines persistent storage to be used in the cluster. |
| storage.className | Specifies the name of the Kubernetes [Kubernetes] storage class that is used to create persistent volumes. |
| storage.accessMode | Specifies the access mode for Kubernetes persistent volumes. |
| storage.size | Specifies the size of the persistent volume. |

| Property | Description |
|--|---|
| endpoints | An array of endpoints that is exposed for a component. |
| endpoint | An endpoint that is exposed outside of the cluster. |
| endpoint.name | Specifies the name of the endpoint that is exposed outside of the cluster. |
| endpoint.serviceType | Specifies the Kubernetes service type that exposes the endpoint port. |
| endpoint.port | Specifies the port on which the service is exposed. |
| <code>endpoint.dnsName</code> | Specifies the Domain Name System (DNS) name that is registered for the endpoint, that is registered to the domain controller (DC) for a deployment with Active Directory enabled. |
| <code>dynamicDnsUpdate</code> | Specifies a Boolean to indicate whether to register automatically the DNS service in the Active Directory DNS records. |
| replicas | Specifies the number of Kubernetes pods to deploy for a component. |
| <code>hadoop.sql</code> | Configuration The collection of data that specifies the settings for the Apache Hadoop [ApacheHadoop] that is running SQL Server in the this cluster. |
| <code>hadoop.yarn.sql.hadr.enabled</code> | Specifies the structured data that describes the configuration for Apache YARN [ApacheHadoop] that is used by Hadoop. Specifies whether SQL Server high availability is enabled. |
| <code>hadoop.yarn.nodeManager.security</code> | Specifies the structured The collection of data that describes specifies the security settings for the node manager cluster. |
| <code>hadoop.yarn.nodeManager.memory.security.activeDirectory</code> | SHOULD <1>Specifies in MB the maximum amount of memory that is available to the node manager for the YARN that is used by Hadoop. specify the Active Directory settings for the cluster. |

| Property | Description |
|---|--|
| <code>hadoop.yarn.nodeManager.vcores.security.activeDirectory.useInternalDomain</code> | SHOULD<2> Specifies the number of virtual cores to allocate to the node manager for the YARN that is used by Hadoop. Specify whether to use a domain that is hosted within the cluster. This property should be set to "false" for production deployments. |
| <code>hadoop.yarn.schedulerMax.security.activeDirectory.ouDistinguishedName</code> | SHOULD<3> Specifies the structured data that describes the settings for the YARN scheduler that is used by Hadoop. Specify the distinguished name (DN) of an organizational unit to which all Active Directory accounts created by the cluster deployment are added. If the domain is named "contoso.local", the value for this field would be "OU=BDC,DC=contoso,DC=local". |
| <code>hadoop.yarn.schedulerMax.memory.security.activeDirectory.activeDnsIpAddresses</code> | SHOULD<4> Specifies the maximum number of MBs of memory that is available to the YARN scheduler. Specify a list of IP addresses of DCs. |
| <code>hadoop.yarn.capacityScheduler.security.activeDirectory.domainControllerFullyQualifiedDns</code> | Specifies the structured data that describes the settings for the YARN scheduler's capacity in Hadoop. SHOULD<5> Specify a list of fully qualified domain names of DCs. |
| <code>hadoop.yarn.capacityScheduler.maxAmPercent.security.activeDirectory.realm</code> | Specifies the maximum percentage of resources that are available to be used by the YARN Scheduler in Hadoop. SHOULD<6> Specify the Active Directory realm. |
| <code>spark.security.activeDirectory.domainDnsName</code> | The collection of data that describes the settings for Apache Spark [ApacheSpark] in the cluster. SHOULD<7> Specify the name of the Active Directory domain. |
| <code>spark.driverMemory.security.activeDirectory.clusterAdmins</code> | Specifies the amount of memory that is available to the Spark driver. SHOULD<8> Specify an Active Directory group which will get administrator permissions. This parameter takes only one AD group. |
| <code>spark.driverCores.security.activeDirectory.clusterUsers</code> | Specifies the number of cores that are available to the Spark driver. SHOULD<9> Specify a |

| Property | Description |
|---|---|
| | list of Active Directory groups that have regular permissions in the cluster. |
| spark.executorInstancessecurity.activeDirectory.appOwners | Describes the number of executors that are available for use by the Spark driver.SHOULD<10> specify a list of Active Directory groups that have permissions to create, delete, and run applications in the cluster. |
| spark.executorMemorysecurity.activeDirectory.appUsers | Specifies the maximum number of bytes that are available to each Spark executor.SHOULD<11> specify a list of Active Directory groups that have permissions to run any applications in the cluster. |
| spark.executorCoressecurity.privileged | Specifies the maximum number of cores that are available to each Spark executor.SHOULD<12> specify whether the containers in the cluster are run as privileged users. |

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 be performed during HTTP operations.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation can maintain to participate in this protocol. The organization is provided to help explain how this protocol works. This document does not require that implementations of the Control Plane REST API protocol adhere to this model, provided the external behavior of the implementation is consistent with that specified in this document.

The following resources are managed by using this protocol:

- Big Data Cluster (section 3.1.5.1)
- Control (section 3.1.5.2)
- Storage (section 3.1.5.3)
- App Deploy (section 3.1.5.4)
- Token (section 3.1.5.5)
- Home Page (section 3.1.5.6)

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 Message Processing Events and Sequencing Rules

The following resources are created and managed by using the control plane service.

| Resource | Section | Description |
|------------------|---------|---|
| Big Data Cluster | 3.1.5.1 | The big data cluster that is deployed in the Kubernetes cluster. |
| Control | 3.1.5.2 | The API that describes the state of the control plane. |
| Storage | 3.1.5.3 | An external mount that is mounted in the HDFS instance of the big data cluster. |

| Resource | Section | Description |
|------------|---------|--|
| App Deploy | 3.1.5.4 | A standalone Python or R script that is deployed in a pod in the cluster. |
| Token | 3.1.5.5 | 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 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 (BDC)** resource represents a big data cluster that is deployed in a Kubernetes cluster in a Kubernetes namespace of the same name.

This resource is 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 BDC | 3.1.5.1.1 | Creates a big data cluster resource. |
| Delete BDC | 3.1.5.1.2 | Deletes a BDC resource. |
| Get BDC Logs | 3.1.5.1.3 | Retrieves logs from a BDC resource. |
| Get BDC Status | 3.1.5.1.4 | Retrieves the status of a BDC resource. |
| Get BDC Information | 3.1.5.1.5 | Retrieves the status and configuration of a BDC resource. |
| Get Service Status | 3.1.5.1.6 | Retrieves the statuses of all resources in a service in a BDC resource. |
| Get Service Resource Status | 3.1.5.1.7 | Retrieves the status of a resource in a service in a BDC resource. |
| Redirect to Metrics Link | 3.1.5.1.8 | Redirects the client to a metrics dashboard. |
| Upgrade BDC | 3.1.5.1.9 | Updates the docker images that are deployed in a BDC resource. |
| Get All BDC Endpoints | 3.1.5.1.10 | Retrieves a list of all endpoints exposed by a BDC resource. |
| Get BDC Endpoint | 3.1.5.1.11 | Retrieves the endpoint information for a specific endpoint in the BDC resource. |

The following properties are valid. All properties are required as specified in the table.

| Property | Required | Description |
|--|----------|--|
| apiVersion | Yes | Kubernetes [Kubernetes] API version that is being used in the big data cluster. The value of this property MUST be "v1". |
| metadata | Yes | See definition of metadata in section 2.2.5. |
| spec | Yes | Structured data that define what to deploy in the big data cluster. |
| spec.docker | | See definition of docker in section 2.2.5. |
| spec.storage | | See definition of storage in section 2.2.5. |
| spec.hadoop | Yes | Structured data that define Apache Hadoop [ApacheHadoop] settings. See section 2.2.5. |
| spec.resources.clusterName | | Specifies the name of the big data cluster into which the resources are being deployed. |
| spec.resources.sparkhead | Yes | Structured data that define the sparkhead resource, which contains all the management services for maintaining Apache Spark instances [ApacheSpark]. |
| spec.resources.sparkhead.spec.replicas | Yes | Specifies the number of replicas to deploy for the sparkhead resource. |
| spec.resources.sparkhead.spec.docker | | See definition of docker in section 2.2.5. |

| Property | Required | Description |
|--|----------|--|
| spec.resources.sparkhead.spec.storage | | See definition of storage in section 2.2.5. |
| spec.resources.sparkhead.spec.settings | | Specifies the structured data that define settings for the sparkhead resource. |
| spec.resources.sparkhead.spec.settings.spark | | See definition of spark in section 2.2.5. Specifies the Spark settings for the sparkhead resource. |
| spec.resources.sparkhead.hadoop.spec.settings.hdfs | | See definition of hadoop in section 2.2.5. Specifies HDFS settings for the sparkhead resource. |
| spec.resources.sparkhead.security | | Specifies the security settings for the sparkhead resource. |
| spec.resources.storage | Yes | Structured data that define the settings for the storage resource. 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. |
| spec.resources.storage.clusterName | | Specifies the name of the big data cluster into which the storage resource is being deployed. |
| spec.resources.storage.metadata | Yes | Specifies the metadata of the big data cluster into which the storage resource is being deployed. |
| spec.resources.storage.spec.type | Yes | Specifies the type of pool that is already deployed in the storage resource. The value of this property MUST be 4, as defined for the type property in section 2.2.5. |
| spec.resources.storage.spec.replicas | Yes | Specifies the number of pods to deploy for the storage resource. |
| spec.resources.storage.spec.docker | | See definition of docker in section 2.2.5. |
| spec.resources.storage.spec.settings | | Specifies the settings for the storage resource. |
| spec.resources.storage.spec.settings.spark | | See definition of spark in section 2.2.5. |
| spec.resources.storage.spec.settings.sql | | Specifies the SQL settings for the storage resource. |
| spec.resources.storage.spec.settings.hdfs | | Specifies the HDFS settings for the storage resource. |
| spec.resources.storage.hadoopsecurity | | See definition of Hadoop in section 2.2.5. Specifies the security settings for the storage resource. |
| spec.resources.master | Yes | Structured data that define settings for master SQL instance. |
| spec.resources.master.clusterName | | Specifies the name of the big data cluster into which the SQL master resource is being |

| Property | Required | Description |
|--|----------|---|
| | | deployed. |
| spec.resources.master.metadata | Yes | See definition of metadata in section 2.2.5. |
| spec.resources.master.spec.type | Yes | Specifies the type of pool to deploy. The value of this property MUST be 1, as defined for the type property in section 2.2.5. |
| spec.resources.master.spec.replicas | Yes | Specifies the number of pods to deploy for the SQL master resource. |
| spec.resources.master.spec.docker | | See definition of docker in section 2.2.5. |
| spec.resources.master.spec.dnsName | | Specifies the DNS name that is registered for the master instance that is registered to the domain controller (DC) for a deployment with Active Directory enabled. |
| spec.resources.master.spec.endpoints | Yes | See definition of endpoints in section 2.2.5. |
| spec.resources.master.spec.settings.sql | | Specifies the SQL settings for the SQL master resource. |
| spec.resources.master.spec.settings.sql.hadr.enabled | | Specifies the setting to enable high availability for the master SQL instances. |
| spec.resources.master. hadoop security | | See definition of hadoop in section 2.2.5. Specifies the security settings for the SQL master resource. |
| spec.resources.compute | Yes | Structured data that defines the settings for compute resource. 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. |
| spec.resources.compute.clusterName | | Specifies the name of the big data cluster into which the resource is being deployed. |
| spec.resources.compute.metadata | Yes | See definition of metadata in section 2.2.5. |
| spec.resources.compute.spec.type | Yes | Specifies the type of pool to deploy. The value of this property MUST be 3, as defined for the type property in section 2.2.5. |
| spec.resources.compute.spec.replicas | Yes | Specifies the number of pods to deploy for the compute resource. |
| spec.resources.compute.spec.docker | | See definition of docker in section 2.2.5. |
| spec.resources.compute.spec.settings | | Specifies the settings for the compute resource. |
| spec.resources.compute.spec.settings.sql | | Specifies the SQL settings for the compute resource. |
| spec.resources.compute.security | | Specifies the security settings for the compute resource. |
| spec.resources.data | Yes | Structured data that defines the settings for |

| Property | Required | Description |
|--|----------|---|
| | | data pool resource. 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. |
| spec.resources.data.clusterName | | Specifies the name of the big data cluster into which the resource is being deployed. |
| spec.resources.data.metadata | Yes | See definition of metadata in section 2.2.5. |
| spec.resources.data.spec.type | Yes | Specifies the type of pool to deploy. The value of this property MUST be 3, as defined for the type property in section 2.2.5. |
| spec.resources.data.spec.replicas | Yes | Specifies the number of pods to deploy for the data resource. |
| spec.resources.data.spec.docker | | See definition of docker in section 2.2.5. |
| spec.resources.data.spec.settings | | Specifies the settings for the data resource. |
| spec.resources.data.spec.settings.sql | | Specifies the SQL settings for the data resource. |
| spec.resources.data.hadoopsecurity | | See definition of hadoop in section 2.2.5. Specifies the security settings for the data resource. |
| spec.resources.nmnode | Yes | Structured data that define settings for the NameNode resource. 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. |
| spec.resources.nmnode.clusterName | | Specifies the name of the big data cluster into which the resource is being deployed. |
| spec.resources.nmnode.metadata | Yes | See definition of metadata in section 2.2.5. |
| spec.resources.nmnode.spec.replicas | Yes | Specifies the number of pods to deploy for the NameNode resource. |
| spec.resources.nmnode.spec.docker | | See definition of docker in section 2.2.5. |
| spec.resources.nmnode.spec.settings | | Specifies the settings for the NameNode resource. |
| spec.resources.nmnode.spec.settings.hdfs | | Specifies the HDFS settings for the NameNode resource. |
| spec.resources.nmnode.hadoopspec.storage | | See definition of hadoop in section 2.2.5. Specifies the storage settings for the NameNode resource. |
| spec.resources.nmnode.security | | Specifies the security settings for the NameNode resource. |

| Property | Required | Description |
|---|----------|--|
| spec.resources.appproxy | Yes | Structured data that define the settings for the application proxy (app proxy) resource. |
| spec.resources.appproxy.clusterName | | Specifies the name of the big data cluster into which the resource is being deployed. |
| spec.resources.appproxy.metadata | Yes | See definition of metadata in section 2.2.5. |
| spec.resources.appproxy.spec.replicas | Yes | Specifies the number of pods to deploy for the appapplication proxy resource. |
| spec.resources.appproxy.spec.docker | | See definition of docker in section 2.2.5. |
| spec.resources.appproxy.spec.settings | | Specifies the settings for the appapplication proxy resource. |
| spec.resources.appproxy.spec.endpoints | Yes | See definition of endpoints in section 2.2.5. |
| spec.resources.appproxy. hadoopsecurity | | See definition of hadoop in section 2.2.5. Specifies the security settings for the application proxy resource. |
| spec.resources.zookeeper | Yes | Specifies the structured data that defines the settings for the zookeeper resource, which contains instances of Apache ZooKeeper [ApacheZooKeeper] that are used to provide synchronization in Hadoop. |
| spec.resources.zookeeper.clusterName | | Specifies the name of the big data cluster into which the resource is being deployed. |
| spec.resources.zookeeper.metadata | Yes | See definition of metadata in section 2.2.5. |
| spec.resources.zookeeper.spec.replicas | Yes | Specifies the number of pods to deploy for the zookeeperZooKeeper resource. |
| spec.resources.zookeeper.spec.docker | | See definition of docker in section 2.2.5. |
| spec.resources.zookeeper.spec.settings | | Specifies the settings for the zookeeperZooKeeper resource. |
| spec.resources.zookeeper.spec. settings .hdfs | | Specifies the HDFS configuration settings for the zookeeperZooKeeper resource. |
| spec.resources.zookeeper. hadoopsecurity | | See definition of hadoop in section 2.2.5. Specifies the security settings for the ZooKeeper resource. |
| spec.resources.gateway | Yes | Structured data that defines the settings for the gateway resource. The gateway resource contains Apache Knox [ApacheKnox] and provides a secure endpoint to connect to Hadoop. |
| spec.resources.gateway.clusterName | | Specifies the name of the big data cluster into which the resource is being deployed. |
| spec.resources.gateway.metadata | Yes | See definition of metadata in section 2.2.5. |
| spec.resources.gateway.spec.replicas | Yes | Specifies the number of pods to deploy for the gateway resource. |
| spec.resources.gateway.spec.docker | | See definition of docker in section 2.2.5. |

| Property | Required | Description |
|---|----------|--|
| spec.resources.gateway.spec.settings | | Specifies the settings for the gateway resource. |
| spec.resources.gateway.spec.endpoints | Yes | See definition of endpoints in section 2.2.5. |
| spec.resources.gateway.spec.dnsNamesecurity | | Specifies the Domain Name System (DNS) name that is registered security settings for the gateway resource that is registered to the domain controller (DC) for a deployment with Active Directory enabled. |
| spec.resources.gateway.hadoop | | See definition of hadoop in section 2.2.5. |
| spec.services | Yes | Structured data that define the service settings and the resources in which the service is present. |
| spec.services.sql | Yes | Specifies the structured data that define the SQL Server service settings. |
| spec.services.sql.resources | Yes | Specifies an array of resources that use the SQL Server service. |
| spec.services.sql.settings | | Specifies the configuration settings for the SQL Server service. |
| spec.services.hdfs | Yes | Specifies the structured data that define the HDFS service settings. |
| spec.services.hdfs.resources | Yes | Specifies an array of resources that use the HDFS service. |
| spec.services.hdfs.settings | | Specifies the configuration settings for the HDFS service. |
| spec.services.spark | Yes | Specifies the structured data that define the Spark service settings. See section 2.2.5. |
| spec.services.spark.resources | Yes | Specifies an array of resources that define which resources use the Spark service. |
| spec.services.spark.settings | Yes | Specifies the configuration settings for the Spark service. See section 2.2.5. |

3.1.5.1.1 Create BDC

The **Create BDC** 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 response message for the **Create BDC** method can result in the following status codes.

| HTTP status code | Description |
|------------------|--|
| 200 | The cluster specification was accepted, and creation of the big data cluster has |

| HTTP status code | Description |
|------------------|---|
| | 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. |

The state of the BDC deployment is retrieved by using the **Get BDC Status** method as specified in section 3.1.5.1.4.

3.1.5.1.1.1 (Updated Section) Request Body

The request body is a JSON object in the format that is shown in the following example.

```
{
  "apiVersion": "v1",
  "metadata": {
    "kind": "BigDataCluster",
    "name": "mssql-cluster"
  },
  "spec": {
    "hadoop": {
      "yarn": {
        "nodeManager": {
          "memory": 18432,
          "vcores": 6
        },
        "schedulerMax": {
          "memory": 18432,
          "vcores": 6
        },
        "capacityScheduler": {
          "maxAmPercent": 0.3
        }
      }
    },
    "resources": {
      "nmnode-0": {
        "spec": {
          "replicas": 2
        }
      },
      "sparkhead": {
        "spec": {
          "replicas": 2
        }
      },
      "zookeeper": {
        "spec": {
          "replicas": 3
        }
      }
    },
    "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": "false>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": 23,
        "settings": {
          "spark": {
            "includeSpark": "true"
          }
        }
      }
    }
  },
  "services": {
    "sql": {
      "resources": [
        "master",
        "compute-0",
        "data-0",
        "storage-0"
      ]
    },
    "hdfs": {
      "resources": [
        "nmnode-0",
        "zookeeper",
        "storage-0",
        "sparkhead"
      ],
      "settings": {
        "hdfs-site.dfs.replication": "3"
      }
    },
    "spark": {
      "resources": [
        "sparkhead",
        "storage-0"
      ],
      "settings": {
        "DriverMemory": "spark-defaults-conf.spark.driver.memory": "2g",
        "DriverCores": "spark-defaults-conf.spark.driver.cores": "1",
        "ExecutorInstances": "spark-defaults-conf.spark.executor.instances": "3",
        "ExecutorMemory": "spark-defaults-conf.spark.executor.memory": "1536m",
        "ExecutorCores": "spark-defaults-conf.spark.executor.cores": "1",
        "yarn-site.yarn.nodemanager.resource.memory-mb": "18432",
        "yarn-site.yarn.nodemanager.resource.cpu-vcores": "6",
        "yarn-site.yarn.scheduler.maximum-allocation-mb": "18432",
        "yarn-site.yarn.scheduler.maximum-allocation-vcores": "6",
        "yarn-site.yarn.scheduler.capacity.maximum-am-resource-percent": "0.3"
      }
    }
  }
}

```

The JSON schema for the **Create BDC** method is presented in section 6.1.1.

3.1.5.1.1.2 Response Body

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

If the request fails, a JSON object of the format that is shown in the following example is returned.

```
{
  "code": 500,
  "reason": "An unexpected exception occurred.",
  "data": "Null reference exception"
}
```

The JSON schema for the response body is presented in section 6.1.2.

3.1.5.1.1.3 Processing Details

This method creates a new **cluster** resource.

3.1.5.1.2 Delete BDC

The **Delete BDC** method deletes the BDC 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 response message for the **Delete BDC** method can result in the following status codes.

| HTTP status code | Description |
|------------------|---|
| 200 | BDC deletion was initiated. |
| 500 | BDC deletion 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 Processing Details

This method deletes a **BDC** resource.

3.1.5.1.3 Get BDC Logs

The **Get BDC Logs** method retrieves the logs from the **BDC** 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 non-zero, the log that is returned starts at the byte located at the offset value.

The response message for the **Get BDC Logs** method can result in 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 is 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 to the length of the log returned. This value represents the new offset value.

3.1.5.1.4 Get BDC Status

The **Get BDC Status** method retrieves the status of all resources in a **BDC** 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 response message for the **Get BDC Status** method can result in the following status codes.

| HTTP status code | Description |
|------------------|---|
| 200 | The state of the BDC resource was returned successfully. |
| 404 | No BDC resource is currently deployed. |
| 500 | The operation failed to retrieve the status of the BDC 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 in the format that is shown in the following example.

```
{
  "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": "approxy",
            "state": "ready",
            "healthStatus": "healthy",
            "details": "ReplicaSet approxy is healthy",
            "instances": null
        }
    ]
}
]
}
}

```

The JSON schema for this response is presented in section 6.1.4.

3.1.5.1.4.3 Processing Details

None.

3.1.5.1.5 Get BDC Information

The **Get BDC Information** method retrieves the status and configuration of the **BDC** resource.

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

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

The response message for the **Get BDC Information** method can result in the following status codes.

| HTTP status code | Description |
|------------------|--|
| 200 | BDC resource information was returned successfully. |
| 404 | No BDC resource currently deployed. |
| 500 | Failed to retrieve the information for the currently deployed BDC resource. |

3.1.5.1.5.1 Request Body

The request body is empty.

3.1.5.1.5.2 (Updated Section) Response Body

The response body is a JSON object that includes the following properties.

| Property | Description |
|----------|---|
| code | The HTTP status code that results from the operation. |
| state | The state of the BDC resource (see section 3.1.5.1.4). |
| spec | A JSON string that represents the JSON model as presented in section 6.1.1. |

The response body is a JSON object in the format that is shown in the following example. The value of the **spec** property is escaped by the server before it is sent to the client. The **spec** property value is then unescaped by the client to create a valid JSON document.

```
{
  "code": 200,
  "state": "Ready",
  "spec": {
    "apiVersion": "v1",
    "kind": "BigDataCluster",
    "name": "test",
    "spec": "mssql-cluster"
  }
}
```

```
"accessMode": "ReadWriteOnce",
"size": "10Gi"},
},
"endpoints": {
  "name": "App": "Knox",
  "serviceType": "NodePort",
  "port": 30778,
  "dynamicDnsUpdate": true
},
"settings": {
  "hadoop": {
    "yarn": {
      "nodeManager": {
        "memory": 18432,
        "vcores": 6,
        "schedulerMax": {
          "memory": 18432,
          "vcores": 6,
          "capacityScheduler": {
            "maxAmPercent": 0.3
          }
        }
      },
      "compute": {
        "gateway": {
          "gateway-site.gateway.httpClient.socketTimeout": "90s",
          "gateway-site.sun.security.krb5.debug": "true"
        }
      }
    }
  },
  "appproxy": {
    "clusterName": "test",
    "metadata": {
      "kind": "Pool",
      "name": "default"
    },
    "spec": {
      "type": 2,
      "replicas": 1,
      "docker": {
        "registry": "repo.corp",
        "repository": "mssql/microsoft.com",
        "imageTag": "rel",
        "imagePullPolicy": "IfNotPresent"
      },
      "storage": {
        "data": {
          "className": "local-storage",
          "accessMode": "ReadWriteOnce",
          "size": "15Gi",
          "logs": {
            "className": "local-storage",
            "accessMode": "ReadWriteOnce",
            "size": "10Gi",
            "settings": {
              "sql": {}
            }
          }
        }
      }
    }
  },
  "hadoop": {
    "yarn": {
      "nodeManager": {
        "memory": 18432,
        "vcores": 6,
        "schedulerMax": {
          "memory": 18432,
          "vcores": 6,
          "capacityScheduler": {
            "maxAmPercent": 0.3
          }
        }
      },
      "compute": {
        "gateway": {
          "clusterName": "test",
          "spec": {
            "replicas": 1,
            "docker": {
              "registry": "repo.corp.microsoft.com",
              "repository": "mssql-private-preview",
              "imageTag": "rel",
              "imagePullPolicy": "IfNotPresent"
            },
            "storage": {
              "data": {
                "className": "local-storage",
                "accessMode": "ReadWriteOnce",
                "size": "15Gi",
                "logs": {
                  "className": "local-storage",
                  "accessMode": "ReadWriteOnce",
                  "size": "10Gi"
                }
              }
            }
          }
        }
      }
    }
  }
}
```

```

"endpoints": [
  {
    "name": "AppServiceProxy",
    "serviceType": "NodePort",
    "port": 30443,
    "settings": {},
    "dynamicDnsUpdate": true
  },
  {
    "settings": {}
  },
  {
    "storage-0": {
      "clusterName": "mssql-cluster",
      "metadata": {
        "kind": "Pool",
        "name": "default"
      },
      "spec": {
        "type": 4,
        "replicas": 2,
        "docker": {
          "registry": "mcr.microsoft.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",
            "capacityScheduler": {
              "maxAmPercent": "0.3"
            },
            "nmnode-0": {
              "clusterName": "test",
              "spec": {
                "replicas": 1,
                "docker": {
                  "registry": "repo.corp.microsoft.com",
                  "repository": "mssql-private-preview",
                  "imageTag": "rel",
                  "imagePullPolicy": "IfNotPresent"
                },
                "storage": {
                  "data": {
                    "className": "local-storage",
                    "accessMode": "ReadWriteOnce",
                    "size": "15Gi"
                  },
                  "logs": {
                    "className": "local-storage",
                    "accessMode": "ReadWriteOnce",
                    "size": "10Gi"
                  }
                },
                "settings": {
                  "hdfs": {},
                  "hadoop": {
                    "yarn": {
                      "nodeManager": {
                        "memory": "18432",
                        "vcores": "6"
                      },
                      "capacityScheduler": {
                        "maxAmPercent": "0.3"
                      },
                      "sparkhead": {
                        "clusterName": "test",
                        "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,RFAUDIT",
"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.microsoft.com",

```

```

"repository": "mssql-private/bdc",
"preview": "imageTag": "latest",
"imagePullPolicy": "IfNotPresent", "Always"
},
"storage": {
  "data": {
    "className": "local-storage",
    "accessMode": "ReadWriteOnce",
    "size": "15Gi",
  },
  "logs": {
    "className": "local-storage",
    "accessMode": "ReadWriteOnce",
    "size": "10Gi",
    "settings": {
      "spark": {
        "ExecutorMemory": "1536m",
        "ExecutorInstances": "3",
        "ExecutorCores": "1",
        "DriverCores": "1",
        "DriverMemory": "2g",
        "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,RFAUDIT",
      "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": {
        "nodeManager": {
          "resource.memory": "18432",
          "vcores": "6",
          "schedulerMax": {
            "memory": "18432",
            "vcores": "6",
            "capacityScheduler": {
              "maxAmPercent": "0.3"
            }
          },
          "zookeeper": {
            "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",

```



```
"compute-0": {
  "clusterName": "test", "kind": "Pool", "name": "default",
  "spec": {
    "type": 3, "replicas": 2,
    "docker": {
      "registry": "mcr.microsoft.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": {}
      }
    },
    "yarn": {
      "nodeManager": {
        "memory": 18432, "vcores": 6, "schedulerMax": {
          "memory": 18432, "vcores": 6, "capacityScheduler": {
            "maxAmPercent": 0.3
          }
        }, "master": {
          "clusterName": "mssql-cluster",
          "metadata": {
            "kind": "Pool", "name": "default"
          },
          "spec": {
            "type": 1, "replicas": 1,
            "docker": {
              "registry": "mcr.microsoft.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": {
            "hdr.enabled": "false"
        },
        "hadoop": {
            "yarn": {
                "nodeManager": {
                    "memory": "18432",
                    "vcores": "6",
                    "schedulerMax": {
                        "memory": "18432",
                        "vcores": "6",
                        "capacityScheduler": {
                            "maxAmPercent": "0.3"
                        }
                    }
                },
                "spark": {
                    "resources": {
                        "sparkhead": {
                            "storage-0": {
                                "settings": {
                                    "ExecutorMemory": "1536m",
                                    "ExecutorInstances": "3",
                                    "ExecutorCores": "1",
                                    "DriverCores": "1",
                                    "DriverMemory": "2g"
                                },
                                "sql": {
                                    "resources": {
                                        "master": {
                                            "compute-0": {
                                                "data-0": {
                                                    "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,RFAUDIT",
                                        "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": {
                                            "storage-java-env.JVMFLAGS": "-Xmx1G -Xms1G",
                                            "zookeeper-log4j-properties.zookeeper.console.threshold": "INFO"
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        },
        "nmnode-0": {
            "clusterName": "mssql-cluster",
            "spec": {
                "replicas": 2,
                "docker": {
                    "registry": "mcr.microsoft.com",
                    "repository": "mssql/bdc",
                    "imageTag": "latest",
                    "imagePullPolicy": "Always"
                },
                "storage-0": {
                    "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,RFAUDIT",
                        "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": {
                            "storage-java-env.JVMFLAGS": "-Xmx1G -Xms1G",
                            "zookeeper-log4j-properties.zookeeper.console.threshold": "INFO"
                        }
                    }
                }
            }
        },
        "zookeeper": {
            "clusterName": "mssql-cluster",
            "spec": {
                "replicas": 3,

```



```

    },
    "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.microsoft.com",
    "repository": "mssql-private-bdc",
    "imageTag": "rc1",
    "imagePullPolicy": "IfNotPresent"
  },
  "storage": {
    "data": {
      "className": "local-storage",
      "accessMode": "ReadWriteOnce",
      "size": "15Gi"
    },
    "logs": {
      "className": "local-storage",
      "accessMode": "ReadWriteOnce",
      "size": "10Gi"
    }
  }
}

```

The JSON schema for this response is presented in section 6.1.3.

3.1.5.1.5.3 Processing Details

None.

3.1.5.1.6 Get Service Status

The **Get Service Status** method retrieves the statuses of all services in a specified service in the **BDC** resource.

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

```
https://<clusterIp>:<controllerPort>/api/v1/bdc/services/<serviceName>/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.

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 response message for the **Get Service Status** method can result in 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 in the format that is shown in the following example.

```
{
  "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"
    }
  ]
}

```

The full JSON schema for this response is presented in section 6.1.5.

3.1.5.1.6.3 Processing Details

None.

3.1.5.1.7 Get Service Resource Status

The **Get Service Resource Status** method retrieves the status of a resource within a specified service in the **BDC**.

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 response message for the **Get Service Resource Status** method can result in 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 of the format that is shown in the following example.

```
{
  "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"
      }
    }
  ]
}
```

A full JSON schema is presented in section 6.1.6.

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 **BDC**.

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 response message for the **Redirect to Metrics Link** method can result in the following status codes.

| HTTP status code | Description |
|------------------|------------------------------|
| 302 | The redirect was successful. |

| HTTP status code | Description |
|------------------|---|
| 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 Upgrade BDC

The **Upgrade BDC** method updates the docker images that are deployed in the **BDC** resource.

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

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

The response message for the **Upgrade BDC** method can result in the following status codes.

| HTTP status code | Description |
|------------------|--|
| 200 | BDC upgrade was initiated. |
| 400 | The request is invalid. |
| 500 | An unexpected error occurred while processing the upgrade. |

3.1.5.1.9.1 Request Body

The request body is a JSON object that includes the following properties.

| Property | Description |
|------------------|---|
| targetVersion | The docker image tag that is used to update all containers in the cluster. |
| targetRepository | The docker repository from which to retrieve the docker images. This parameter is used when the desired repository differs from the repository that is currently being used by the big data cluster. |

The request body is a JSON object in the format that is shown in the following example.

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

}

3.1.5.1.9.2 Response Body

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

If the request fails, a JSON object as described in section 6.1.2 is returned.

3.1.5.1.9.3 Processing Details

This method upgrades the **BDC** resource.

3.1.5.1.10 Get All BDC Endpoints

The **Get All BDC Endpoints** method retrieves a list of all endpoints exposed by a BDC resource.

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

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

The response message for the **Get All BDC Endpoints** method can result in the following status codes.

| HTTP status code | Description |
|------------------|---|
| 200 | The BDC 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 of the format that is shown in the following example.

```
[
  {
    "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"
  }
]

```

A full JSON schema for this response is presented in section 6.1.7.

3.1.5.1.10.3 Processing Details

None.

3.1.5.1.11 Get BDC Endpoint

The **Get BDC Endpoint** method retrieves the endpoint information for a specific endpoint in the BDC 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 files and Spark.
- **spark-history**: Portal for managing and monitoring Apache Spark [ApacheSpark] jobs.
- **yarn-ui**: Portal for accessing Apache Spark monitoring and diagnostics.
- **app-proxy**: Proxy for running commands against applications deployed in the BDC.
- **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 response message for the **Get BDC Endpoint** method can result in the following status codes.

| HTTP status code | Description |
|------------------|---|
| 200 | The BDC endpoint was successfully returned. |
| 404 | The BDC 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 of the format that is shown in the following example.

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

A full JSON schema for this response is presented in section 6.1.8.

3.1.5.1.11.3 Processing Details

None.

3.1.5.2 Control

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

This resource is 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. |

The following property is valid.

| Property | Description |
|---------------|--|
| targetVersion | The docker image tag that is used to update all containers 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>`

The response message for the **Get Control Status** method can result in 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 in the same format as described in section 3.1.5.1.6.2.

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 response message for the **Upgrade Control** method can result in 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 request body is a JSON object that includes the following properties.

| Property | Description |
|------------------|---|
| targetVersion | The docker image tag that is used to update all containers in the control plane. |
| targetRepository | The docker repository from which to retrieve the docker images. This parameter is used when the desired repository differs from the repository that is currently being used by the big data cluster. |

The request body is a JSON object in the format that is shown in the following example.

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

3.1.5.2.2.2 Response Body

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

If the request fails, a JSON object as described in section 6.1.2 is returned.

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 response message for the **Redirect to Metrics Link** method can result in 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=[<true/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 response message for the **Get Control Resource Status** method can result in 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 in the same format as described in section 3.1.5.1.7.2.

3.1.5.2.4.3 Processing Details

None.

3.1.5.3 Storage

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

This resource is 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. |
| 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 | 3.1.5.3.5 | Refresh a mount. |

The following properties are valid.

| Property name | Description |
|---------------|---|
| mount | The path of the HDFS mount. |
| remote | The HDFS mount point to attach the mount to. |
| state | The status of the HDFS mount deployment. |
| error | The mount is unhealthy. This field is populated only if the mount is unhealthy. |

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 response message for the **Get Mount Status** method can result in 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 of the format that is shown in the following example.

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

The full JSON schema for the response is presented in section 6.2.1.

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 response message for the **Get All Mount Statuses** method can result in 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 contains an array of JSON objects in the format that is described in section 3.1.5.3.1.2.

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 response message for the **Create Mount** method can result in 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 property corresponds to an authentication property that is needed to access the remote file system. The authentication properties 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 response message for the **Delete Mount** method can result in 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, there is no response body.

For an unsuccessful request, the response body contains a JSON object of the type **Cluster Error Response** as described in section 6.1.2.

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 response message for the **Refresh Mount** method can result in 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

On an unsuccessful request, the response body contains a JSON object of the type **Response**.

3.1.5.3.5.3 Processing Details

None.

3.1.5.4 (Updated Section) App Deploy

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

This resource is 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. |

The following properties are valid.

| Property Name | Description |
|---------------|--|
| name | Name of the application that is being deployed. |
| internal_name | Name for the application that is used internally within the cluster. |
| state | State of the application's deployment. Valid values are the following: <ul style="list-style-type: none">InitialCreatingUpdatingWaitingForUpdateReadyDeleting |

| Property Name | Description |
|-------------------|---|
| | <ul style="list-style-type: none"> WaitingForDelete Deleted Error |
| version | Version of the app being deployed. |
| input_param_defs | Array of parameters that represent the inputs that can be passed to the application. |
| parameter | Structured data representing an application parameter. A parameter consists of a name and a type . |
| parameter.name | Name of the parameter. |
| parameter.type | Type of parameter. Valid values are the following: <ul style="list-style-type: none"> str int dataframe data.frame float Matrix vector bool |
| output_param_defs | Array of parameters that represent the outputs of the application. |
| links | Array of links . |
| link | Structured data that represents a URL that can be used to access the deployed application. |
| link.app | An endpoint to access to deployed application. |
| link.swagger | An endpoint to a Swagger editor [Swagger2.0]. The editor can be used to directly send requests to the deployed application. |
| success | Describes whether an application method succeeded. |
| errorMessage | Describes the reason an application method failed. |
| outputParameters | List of output parameters that resulted from the method. See output_param_defs . |
| outputFiles | Array of file names that resulted from the application operation. |
| consoleOutput | Describes the text output that resulted from the application method. |
| changedFiles | Array of file names that were modified from the application operation. |

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 response message for the **Get App** method can result in 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 formatted as shown in the following example.

```
{
  "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"
  }
}
```

The JSON schema for the response is presented in section 6.3.1

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 deployed **app** resource.

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

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

name: The name of the deployed application.

The response message for the **Get App Versions** method can result in the following status codes.

| HTTP status code | Description |
|------------------|---|
| 200 | A list of all versions of the deployed application was successfully returned. |
| 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 contains an array of **app** descriptions in a JSON object in the format that is described in section 3.1.5.4.1.1.

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 deployed in the cluster.

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

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

The response message for the **Get All Apps** method can result in 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 contains an array of descriptions for all applications that are deployed in the cluster in a JSON object in the format that is described in section 3.1.5.4.1.2.

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 **app** resource in the cluster.

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

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

The response message for the **Create App** method can result in 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 Request Body

The request body contains a ZIP file that has been stored for later access by the cluster. The ZIP file contains a specification with the filename "spec.yaml" that is written in YAML [YAML1.2] as well as the Python script, R script, SQL Server Integration Services (SSIS) application, or MLEAP model, a format for serializing machine learning pipelines, that is to be deployed.

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 a deployed **app** resource.

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

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

The response message for the **Update App** method can result in 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 contains a ZIP file that has been stored for future access by the cluster. The ZIP file contains a specification with the filename "spec.yaml" that is written in YAML [YAML1.2] as well as the updated Python script, R script, SQL Server Integration Services (SSIS) application, or MLEAP model, a format for serializing machine learning pipelines, that is to be deployed.

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 **app** resource 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 response message for **Delete App** method can result in 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 a deployed **app** resource.

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 app proxy.

name: The name of the deployed application.

version: The version of the deployed application.

The response message for **Run App** method can result in 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 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 like the following.

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

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

3.1.5.4.7.2 Request Body

The request body contains a JSON object in the format that is shown in the following example.

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

The properties in this JSON object match the names and types that are described in **appModel.input_params_defs** in section 3.1.5.3.

3.1.5.4.7.3 Response Body

The response body is a JSON object in the format that is shown in the following example.

```
{
  "success": true,
}
```

```

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

```

The full schema definition is presented in section 6.3.2.

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.

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 response message for the **Get App Swagger Document** method can result in 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** resource is a JWT [RFC7519] token that can be used as a form of authentication to use an application.

It can be 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. |

The following properties are valid. All properties are required.

| Property | Description |
|--------------|---|
| token_type | The token type that is returned MUST be Bearer. |
| access_token | The JWT token that is generated for the request. |
| expires_in | The number of seconds for which the token is valid after being issued. |
| expires_on | The date on which the token expires. The date is based on the number of seconds since the Unix Epoch. |
| token_id | Unique ID that was generated for the token request. |

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 response message for the **Create Token** method can result in 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 in the format that is shown in the following example.

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 response message for the **Ping Controller** method can result in 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 is a JSON object in the format that is shown in the following example.

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

The full schema definition is presented in section 6.5.1.

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 response message for the **Info** method can result in 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 is a JSON object in the format that is shown in the following example.

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

The full schema definition is presented in section 6.5.2.

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

In this example, the client deploys a big data cluster to the server.

4.1 Request to Check 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 a 200 status.

Request:

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

4.2 Request to Create Big Data Cluster

If the GET operation returns a 200 status, the client can proceed to create a big data cluster by sending the following request that uses the following sample configuration for a cluster named "mssql-cluster".

Request:

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

Body:

```
{
  "apiVersion": "v1",
  "metadata": {
    "kind": "BigDataCluster",
    "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 Check on Big Data Cluster Deployment Progress

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.

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": "approxy",
            "state": "ready",
            "healthStatus": "healthy",
            "details": "ReplicaSet approxy is healthy",
            "instances": null
        }
    ]
}
]
}

```

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 |
|-------------|---------|
| BDC | 6.1 |
| Storage | 6.2 |
| App | 6.3 |
| Token | 6.4 |
| Home | 6.5 |

6.1 Big Data Cluster

6.1.1 (Updated Section) Big Data Cluster Spec Schema

```
{
  "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",

```



```

    },
    "hadoop": {
      "required": [
        "yarnPrivileged"
      ],
      "properties": {
        "yarnActiveDirectory": {
          "$ref": "#/definitions/yarn"
        }
      }
    },
    "spark": {
      "id": "#/properties:/activeDirectory",
      "driverMemory": {
        "type": "object",
        "required": [
          "useInternalDomain",
          "ouDistinguishedName",
          "dnsIpAddresses",
          "domainControllerFullyQualifiedDns",
          "domainDnsName",
          "clusterAdmins",
          "clusterUsers"
        ],
        "properties": {
          "useInternalDomain": {
            "$id": "#/properties/activeDirectory/properties/useInternalDomain",
            "type": "boolean"
          },
          "ouDistinguishedName": {
            "$id": "#/properties/activeDirectory/properties/ouDistinguishedName",
            "type": "string",
            "example": "2g"
          },
          "dnsIpAddresses": {
            "driverCores": {
              "$id": "#/properties/activeDirectory/properties/dnsIpAddresses",
              "type": "integer"array",
              "items": {
                "executorInstances": {
                  "type": "integer"
                },
                "executorMemory": {
                  "$id": "#/properties/activeDirectory/properties/dnsIpAddresses/items",
                  "type": "string",
                  "example": "1536m"
                },
                "executorCores": {
                  "domainControllerFullyQualifiedDns": {
                    "$id":
                    "#/properties/activeDirectory/properties/domainControllerFullyQualifiedDns",
                    "type": "integer"array",
                    "items": {
                      "$id":
                      "#/properties/activeDirectory/properties/domainControllerFullyQualifiedDns/items",
                      "type": "string"
                    }
                  },
                  "realm": {
                    "$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"
    }
  }
},
"privileged": {
  "$id": "#/properties/privileged",
  "type": "boolean"
}
},
"spark": {
  "patternProperties": {
    "patternProperties": {
      "*": {
        "type": "string"
      }
    }
  }
},
"hdfs": {
  "patternProperties": {
    "patternProperties": {
      "hdfs-site.dfs.replication": {
        "type": "string"
      }
    }
  }
},
"sql": {
  "properties": {
    "*": {
      "type": "string"
    }
  }
},
"gateway": {
  "patternProperties": {
    "patternProperties": {
      "*": {
        "type": "string"
      }
    }
  }
}
},
"metadata": {
  "required": [

```

```

        "kind",
        "name"
    ],
    "properties": {
        "kind": {
            "type": "string"
        },
        "name": {
            "type": "string"
        }
    },
    "replicas": {
        "type": "integer"
    }
},
"$schema": "http://json-schema.org/draft-07/schema#",
"$id": "http://example.com/root.json",
"type": "object",
"required": [
    "apiVersion",
    "metadata",
    "spec"
],
"properties": {
    "apiVersion": {
        "$id": "#/properties/apiVersion",
        "const": "v1"
    },
    "metadata": {
        "$ref": "#/definitions/metadata"
    },
    "spec": {
        "$id": "#/properties/spec",
        "type": "object",
        "required": [
            "hadoop",
            "resources",
            "services"
        ],
        "properties": {
            "hadoopsecurity": {
                "$ref": "#/definitions/hadoopsecurity"
            },
            "resources": {
                "$id": "#/properties/spec/properties/resources",
                "type": "object",
                "required": [
                    "sparkhead",
                    "storage-0",
                    "nmnode-0",
                    "master",
                    "compute-0",
                    "approxxy",
                    "zookeeper",
                    "gateway",
                    "data-0"
                ],
                "properties": {
                    "sparkhead": {
                        "$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": {
          "$id":
"/properties/spec/properties/resources/properties/sparkhead/properties/spec/properties/repl
icas",
          "type": "integer"
        },
        "docker": {
          "$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": {
              "$ref": "#/definitions/spark"
            }
          }
        }
      }
    }
  },
  "hadoop": {
    "$ref": "#/definitions/hadoop"
  }
},
"storage-0": {
  "$id": "#/properties/spec/properties/resources/properties/storage-0",
  "type": "object",
  "required": [
    "metadata",
    "spec"
  ],
  "properties": {
    "clusterName": {
      "$id": "#/properties/spec/properties/resources/properties/storage-
0/properties/clusterName",
      "type": "string"
    },
    "metadata": {
      "$ref": "#/definitions/metadata"
    },
    "security": {
      "$ref": "#/definitions/security"
    }
  },
  "spec": {
    "$id": "#/properties/spec/properties/resources/properties/storage-
0/properties/spec",
    "type": "object",
    "required": [
      "type",

```

```

        "replicas",
        "settings"
    ],
    "properties": {
        "type": {
            "$id": "#/properties/spec/properties/resources/properties/storage-
0/properties/spec/properties/type",
            "type": "integer"
        },
        "replicas": {
            "$id": "#/properties/spec/properties/resources/properties/storage-
0/properties/spec/properties/replicas",
            "type": "integer"
        },
        "docker": {
            "$ref": "#/definitions/docker"
        },
        "storage": {
            "$ref": "#/definitions/storage"
        },
        "settings": {
            "$id": "#/properties/spec/properties/resources/properties/storage-
0/properties/spec/properties/settings",
            "type": "object",
            "required": [
                "spark",
                "sql",
                "hdfs"
            ],
            "properties": {
                "spark": {
                    "$ref": "#/definitions/spark"
                },
                "sql": {
                    "$id": "#/properties/spec/properties/resources/properties/storage-
0/properties/spec/properties/settings/properties/sql",
                    "type": "object",
                    "required": [
                        "hdfs"
                    ],
                    "properties": {
                        "hdfs": {
                            "$id": "#/properties/spec/properties/resources/properties/storage-
0/properties/spec/properties/settings/properties/hdfs",
                            "type": "object",
                            "required": [
                                "hadoop"
                            ],
                            "properties": {
                                "hadoop": {
                                    "$ref": "#/definitions/hadoophdfs"
                                }
                            }
                        }
                    }
                }
            }
        },
        "hadoop": {
            "$ref": "#/definitions/hadoophdfs"
        }
    }
},
"nmnode-0": {
    "$id": "#/properties/spec/properties/resources/properties/nmnode-0",
    "type": "object",
    "required": [
        "spec"
    ],
    "properties": {
        "clusterName": {
            "$id": "#/properties/spec/properties/resources/properties/nmnode-
0/properties/clusterName",
            "type": "string"
        },
        "security": {
            "$ref": "#/definitions/security"
        }
    }
}

```

```

    },
    "spec": {
      "$id": "#/properties/spec/properties/resources/properties/nmnode-
0/properties/spec",
      "type": "object",
      "required": [
        "replicas"
      ],
      "properties": {
        "replicas": {
          "$id": "#/properties/spec/properties/resources/properties/nmnode-
0/properties/spec/properties/replicas",
          "type": "integer"
        },
        "docker": {
          "$ref": "#/definitions/docker"
        },
        "storage": {
          "$ref": "#/definitions/storage"
        },
        "settings": {
          "$id": "#/properties/spec/properties/resources/properties/nmnode-
0/properties/spec/properties/settings",
          "type": "object",
          "required": [
            "hdfs"
          ],
          "properties": {
            "hdfs": {
              "$id": "#/properties/spec/properties/resources/properties/nmnode-
0/properties/spec/properties/settings/properties/hdfs",
              "type": "object"
            }
          }
        },
        "hadoop": {
          "$ref": "#/definitions/hadoophdfs"
        }
      }
    }
  },
  "master": {
    "$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": {
            "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/type",
            "type": "integer"
        },
        "replicas": {
            "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/replicas",
            "type": "integer"
        },
        "docker": {
            "$ref": "#/definitions/docker"
        },
        "storage": {
            "$ref": "#/definitions/storage"
        },
        "endpoints": {
            "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoints",
            "type": "array",
            "items": {
                "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoints/items",
                "type": "object",
                "required": [
                    "name",
                    "serviceType",
                    "port"
                ],
                "properties": {
                    "name": {
                        "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoints/items/properties/name",
                        "const": "Master",
                        "type": "string"
                    },
                    "serviceType": {
                        "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoints/items/properties/serviceType",
                        "enum": [
                            "NodePort",
                            "LoadBalancer"
                        ]
                    },
                    "port": {
                        "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoints/items/properties/port",
                        "type": "integer",
                        "examples": [
                            31433
                        ]
                    }
                }
            }
        },
        "dynamicDnsUpdate": {
            "dnsName": {
                "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoints/items/properties/port",
                "type": "string"
            }
        }
    }
}

```

```

        "$id":
        "#/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoint
s/items/properties/port",
        "type": "boolean"
    }
    },
    "settings": {
        "$id":
        "#/properties/spec/properties/resources/properties/master/properties/spec/properties/settings
",
        "type": "object",
        "required": [
            "sql"
        ],
        "properties": {
            "sql": {
                "$id":
                "#/properties/spec/properties/resources/properties/master/properties/spec/properties/settings
/properties/sql",
                "type": "object",
                "required": [
                    "hadr.enabled"
                ],
                "properties": {
                    "hadr.enabled": {
                        "$id":
                        "#/properties/spec/properties/resources/properties/master/properties/spec/properties/settings
/properties/sql/properties/hadr.enabled",
                        "enum": [
                            "false",
                            "true"
                        ]
                    }
                }
            }
        }
    },
    "hadoop": {
        "$ref": "#/definitions/hadoopsql"
    }
}
},
"compute-0": {
    "$id": "#/properties/spec/properties/resources/properties/compute-0",
    "type": "object",
    "required": [
        "metadata",
        "spec"
    ],
    "properties": {
        "clusterName": {
            "$id": "#/properties/spec/properties/resources/properties/compute-
0/properties/clusterName",
            "type": "string"
        },
        "metadata": {
            "$ref": "#/definitions/metadata"
        },
        "security": {
            "$ref": "#/definitions/security"
        }
    },
    "spec": {

```



```

    "$id": "#/properties/spec/properties/resources/properties/compute-
0/properties/spec",
    "type": "object",
    "required": [
      "type",
      "replicas"
    ],
    "properties": {
      "type": {
        "$id": "#/properties/spec/properties/resources/properties/compute-
0/properties/spec/properties/type",
        "type": "integer"
      },
      "replicas": {
        "$id": "#/properties/spec/properties/resources/properties/compute-
0/properties/spec/properties/replicas",
        "type": "integer"
      },
      "docker": {
        "$ref": "#/definitions/docker"
      },
      "storage": {
        "$ref": "#/definitions/storage"
      },
      "settings": {
        "$id": "#/properties/spec/properties/resources/properties/compute-
0/properties/spec/properties/settings",
        "type": "object",
        "required": [
          "sql"
        ],
        "properties": {
          "sql": {
            "$id": "#/properties/spec/properties/resources/properties/compute-
0/properties/spec/properties/settings/properties/sql",
            "type": "object"
          }
        }
      },
      "hadoop": {
        "$ref": "#/definitions/hadoopsql"
      }
    }
  },
  "approxxy": {
    "$id": "#/properties/spec/properties/resources/properties/approxxy",
    "type": "object",
    "required": [
      "spec"
    ],
    "properties": {
      "clusterName": {
        "$id":
        "#/properties/spec/properties/resources/properties/approxxy/properties/clusterName",
        "type": "string"
      },
      "security": {
        "$ref": "#/definitions/security"
      },
      "spec": {
        "$id":
        "#/properties/spec/properties/resources/properties/approxxy/properties/spec",
        "type": "object",
        "required": [

```

```

        "replicas",
        "endpoints"
    ],
    "properties": {
        "replicas": {
            "$id":
"/properties/spec/properties/resources/properties/approxy/properties/spec/properties/replicas",
            "type": "integer"
        },
        "docker": {
            "$ref": "#/definitions/docker"
        },
        "storage": {
            "$ref": "#/definitions/storage"
        },
        "endpoints": {
            "$id":
"/properties/spec/properties/resources/properties/approxy/properties/spec/properties/endpoints",
            "type": "array",
            "items": {
                "$id":
"/properties/spec/properties/resources/properties/approxy/properties/spec/properties/endpoints/items",
                "type": "object",
                "required": [
                    "name",
                    "serviceType",
                    "port"
                ],
                "properties": {
                    "name": {
                        "$id":
"/properties/spec/properties/resources/properties/approxy/properties/spec/properties/endpoints/items/properties/name",
                        "const": "AppServiceProxy"
                    },
                    "serviceType": {
                        "$id":
"/properties/spec/properties/resources/properties/approxy/properties/spec/properties/endpoints/items/properties/serviceType",
                        "enum": [
                            "NodePort",
                            "LoadBalancer"
                        ]
                    },
                    "port": {
                        "$id":
"/properties/spec/properties/resources/properties/approxy/properties/spec/properties/endpoints/items/properties/port",
                        "type": "integer",
                        "examples": [
                            30778
                        ]
                    }
                }
            }
        },
        "dnsName": {
            "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoints/items/properties/port",
            "type": "string"
        },
        "dynamicDnsUpdate": {
            "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoints/items/properties/port",
            "type": "boolean"
        }
    }
}

```

```

    },
    "settings": {
      "$id":
"/properties/spec/properties/resources/properties/approxy/properties/spec/properties/settin
gs",
      "type": "object"
    }
  },
  "hadoop": {
    "$ref": "#/definitions/hadoop"
  }
},
"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": {
          "$id":
"/properties/spec/properties/resources/properties/zookeeper/properties/spec/properties/repli
cas",
          "type": "integer"
        },
        "docker": {
          "$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": {
              "$id":
"/properties/spec/properties/resources/properties/zookeeper/properties/spec/properties/setti
ngs/properties/hdfs",
              "type": "object"
            }
          }
        }
      }
    }
  }
},
  "hadoop": {
    "$ref": "#/definitions/hadoop"
  }
}

```

```

    }
  },
  "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": {
            "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/replicas",
            "type": "integer"
          },
          "docker": {
            "$ref": "#/definitions/docker"
          },
          "storage": {
            "$ref": "#/definitions/storage"
          },
          "endpoints": {
            "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoints",
            "type": "array",
            "items": {
              "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoints/items",
              "type": "object",
              "required": [
                "name",
                "serviceType",
                "port"
              ],
              "properties": {
                "name": {
                  "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoints/items/properties/name",
                  "const": "Knox"
                },
                "serviceType": {
                  "$id":
"#/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoints/items/properties/serviceType",
                  "enum": [
                    "NodePort",
                    "LoadBalancer"
                  ]
                }
              }
            }
          }
        }
      }
    }
  }
}

```

```

        "port": {
            "$id":
"/properties/spec/properties/resources/properties/gateway/properties/spec/properties/endpoints/items/properties/port",
            "type": "integer"
        },
        "dnsName": {
            "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoints/items/properties/port",
            "type": "string"
        },
        "dynamicDnsUpdate": {
            "$id":
"/properties/spec/properties/resources/properties/master/properties/spec/properties/endpoints/items/properties/port",
            "type": "boolean"
        }
    }
},
"settings": {
    "$id":
"/properties/spec/properties/resources/properties/gateway/properties/spec/properties/settings",
    "type": "object"
}
},
"hadoop": {
    "$ref": "#/definitions/hadoop"
}
},
"data-0": {
    "$id": "#/properties/spec/properties/resources/properties/data-0",
    "type": "object",
    "required": [
        "metadata",
        "spec"
    ],
    "properties": {
        "clusterName": {
            "$id": "#/properties/spec/properties/resources/properties/data-0/properties/clusterName",
            "type": "string"
        },
        "security": {
            "$ref": "#/definitions/security"
        },
        "metadata": {
            "$ref": "#/definitions/metadata"
        },
        "spec": {
            "$id": "#/properties/spec/properties/resources/properties/data-0/properties/spec",
            "type": "object",
            "required": [
                "type",
                "replicas"
            ],
            "properties": {
                "type": {
                    "$id": "#/properties/spec/properties/resources/properties/data-0/properties/spec/properties/type",
                    "type": "integer"
                },
                "replicas": {
                    "$id": "#/properties/spec/properties/resources/properties/data-0/properties/spec/properties/replicas",

```

```
        "type": "integer"
      },
      "docker": {
        "$ref": "#/definitions/docker"
      },
      "storage": {
        "$ref": "#/definitions/storage"
      },
      "settings": {
        "$id": "#/properties/spec/properties/resources/properties/data-
0/properties/spec/properties/settings",
        "type": "object",
        "required": [
          "sql"
        ],
        "properties": {
          "sql": {
            "$id": "#/properties/spec/properties/resources/properties/data-
0/properties/spec/properties/settings/properties/sql",
            "type": "object"
          }
        }
      },
      "hadoop": {
        "$ref": "#/definitions/hadoopsql"
      }
    }
  },
  "services": {
    "$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": "mastercompute-0"
              },
              {
                "const": "compute-data-0"
              },
              {
                "const": "data-0"
              }
            ]
          }
        }
      }
    }
  }
}
```

```

        "const": "storage-0"
    }
  ],
  "settings": {
    "$id": "#/properties/spec/properties/services/propertiesref":
    "#/definitions/sql/properties/settings",
    "type": "object"
  }
},
"hdfs": {
  "$id": "#/properties/spec/properties/services/properties/hdfs",
  "type": "object",
  "required": [
    "resources"
  ],
  "properties": {
    "resources": {
      "$id":
      "#/properties/spec/properties/services/properties/hdfs/properties/resources",
      "type": "array",
      "items": [
        {
          "const": "nmnode-0"
        },
        {
          "const": "zookeeper"
        },
        {
          "const": "storage-0"
        },
        {
          "const": "sparkhead"
        }
      ]
    },
    "settings": {
      "$id": "#/properties/spec/properties/services/propertiesref":
      "#/definitions/hdfs/properties/settings",
      "type": "object"
    }
  },
  "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-0"
          }
        ]
      },
      "settings": {
        "$ref": "#/definitions/spark"
      }
    }
  }
}

```



```

    "$id": "#/properties/code",
    "type": "integer"
  },
  "state": {
    "$id": "#/properties/state",
    "type": "string",
    "title": "The State Schema"
  },
  "spec": {
    "$id": "#/properties/spec",
    "type": "string"
  }
}
}
}

```

6.1.4 Big Data Cluster Status Schema

```

{
  "definitions": {},
  "$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-
0/status/nodemetrics/ui"
                ],
              },
              "sqlMetricsUrl": {
                "type": "string",
                "examples": [

```


6.1.6 Big Data Cluster Service Resource Status Schema

```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$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",
              }
            }
          }
        }
      }
    }
  }
}
```



```

    "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

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

6.3.1 App Description Schema

```
{  
  "definitions": {  
    "link": {  
      "type": "object",  
      "properties": {  
        "^.*$": {  
          "type": "string"  
        }  
      }  
    },  
    "parameter": {  
      "required": [  
        "name",  
        "type"  
      ],  
      "properties": {  
        "name": {  
          "type": "string"  
        },  
        "type": {  
          "enum": [  
            "str",  
            "int",  
            "dataframe",  
            "data.frame",  
            "float",  
            "matrix",  
            "vector",  
            "bool"  
          ]  
        }  
      }  
    }  
  }  
},  
"$schema": "http://json-schema.org/draft-07/schema#",  
"type": "array",  
"title": "App Result Schema",  
"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": {
```



```

],
"properties": {
  "success": {
    "$id": "#/properties/success",
    "type": "boolean",
  },
  "errorMessage": {
    "$id": "#/properties/errorMessage",
    "type": "string",
  },
  "outputParameters": {
    "$id": "#/properties/outputParameters",
    "type": "object",
    "required": [
      "result"
    ],
    "properties": {
      "result": {
        "$id": "#/properties/outputParameters/properties/result",
        "type": "integer"
      }
    }
  },
  "outputFiles": {
    "$id": "#/properties/outputFiles",
    "type": "object",
  },
  "consoleOutput": {
    "$id": "#/properties/consoleOutput",
    "type": "string",
  },
  "changedFiles": {
    "$id": "#/properties/changedFiles",
    "type": "array",
  }
}
}
}

```

6.4 Token

6.4.1 Token Response Schema

```

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

6.5.1 Ping Response Schema

```

{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$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

```

{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$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: The **security.activeDirectory** property is not supported by SQL Server earlier than SQL Server 2019 Cumulative Update 2 (CU2).

<2> Section 2.2.5: The **security.activeDirectory.useInternalDomain** property 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 property **security.useInternalDomain** was used.

<3> Section 2.2.5: The **security.activeDirectory.ouDistinguishedName** property is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent property **security.ouDistinguishedName** was used.

<4> Section 2.2.5: The **security.activeDirectory.activeDnsIpAddresses** property is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent property **security.activeDnsIpAddresses** was used.

<5> Section 2.2.5: The **security.activeDirectory.domainControllerFullyQualifiedDns** property is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent property **security.domainControllerFullyQualifiedDns** was used.

<6> Section 2.2.5: The **security.activeDirectory.realm** property is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent property **security.realm** was used.

<7> Section 2.2.5: The **security.activeDirectory.domainDnsName** property is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent property **security.domainDnsName** was used.

<8> Section 2.2.5: The **security.activeDirectory.clusterAdmins** property is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent property **security.clusterAdmins** was used.

<9> Section 2.2.5: The **security.activeDirectory.clusterUsers** property is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent property **security.clusterUsers** was used.

<10> Section 2.2.5: The **security.activeDirectory.appOwner** property is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent property **security.appOwner** was used.

<11> Section 2.2.5: The **security.activeDirectory.appUsers** property is not supported by SQL Server earlier than SQL Server 2019 CU2. In SQL Server 2019 and SQL Server 2019 CU1, the equivalent property **security.appUsers** was used.

<12> Section 2.2.5: The **security.privileged** property is not supported by SQL Server earlier than SQL Server 2019 CU2.

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 |
|------------------------------------|---|----------------|
| 1.2.2 Informative References | Added [MSDOCS-ConfigBDC] reference. | Minor |
| 1.3 Overview | Added clarification about configuring the Spark and Hadoop components. | Major |
| 1.5 Prerequisites/Preconditions | Added [Kubernetes] reference. | Major |
| 2.2.5 JSON Elements | Added for the security settings to be used when a cluster is deployed with Active Directory. | Major |
| 2.2.5 JSON Elements | Added descriptions of endpoint,dnsName, dynamicDnsUpdate, sql, sql.hadr.enabled, multiple security, multiple security.activeDirectory, and security.privileged properties, and removed descriptions of multiple hadoop and multiple spark properties. | Major |
| 3.1.5.1 Big Data Cluster | Added descriptions of multiple hdfs, multiple security, multiple dnsName properties, and removed descriptions of multiple hadoop and multiple dnsName properties. | Major |
| 3.1.5.1.1.1 Request Body | Replaced request body example for SQL Server 2019 with request body example for SQL Server 2019 CU3. | Minor |
| 3.1.5.1.5.2 Response Body | Replaced response body example for SQL Server 2019 with response body example for SQL Server 2019 CU3. | Minor |
| 6.1.1 Big Data Cluster Spec Schema | Replaced Big Data Cluster Spec Schema for SQL Server 2019 with Big Data Cluster Spec Schema for SQL Server 2019 CU3. | Minor |

9 Index

A

Applicability 12

C

Capability negotiation 12

Change tracking 110

Common

- Abstract data model 19

- Higher-layer triggered events 19

- Initialization 19

- Message processing events and sequencing rules 19

- Other local events 71

- Timer events 71

- Timers 19

E

Elements 14

Examples

- Check on Big Data Cluster Deployment Progress example 74

- Request to Check Control Plane Status example 72

- Request to Create Big Data Cluster example 72

F

Fields - vendor-extensible 12

Full JSON schema 79

G

Glossary 7

H

Headers

- X-RequestID 13

HTTP headers 13

HTTP methods 13

I

Implementer - security considerations 78

Index of security parameters 78

Informative references 10

Introduction 7

J

JSON schema 79

M

Messages

- transport 13

N

Namespaces 13

Normative references 9

O

Overview (synopsis) 10

P

Parameters - security index 78

Preconditions 12

Prerequisites 12

Product behavior 108

Protocol Details

Cluster Admin 71

Common 19

Protocol examples

Check on Big Data Cluster Deployment Progress 74

Request to Check Control Plane Status 72

Request to Create Big Data Cluster 72

R

References

informative 10

normative 9

Relationship to other protocols 11

S

Security

implementer considerations 78

parameter index 78

Standards assignments 12

T

Tracking changes 110

Transport 13

elements 14

HTTP headers 13

HTTP methods 13

namespaces 13

V

Vendor-extensible fields 12

Versioning 12

X

X-RequestID 13