Notes

NIST Special Publication 1500-8

THIS IS A DRAFT

NIST Big Data Interoperability Framework: Volume 8, Reference Architecture Interface

NIST Big Data Public Working Group
Reference Architecture Subgroup

Version 0.1
October 19, 2016

http://dx.doi.org/10.6028/NIST.SP.1500-8

National Institute of Standards and Technology
U.S. Department of Commerce

NIST Special Publication 1500-6
Information Technology Laboratory

NIST Big Data Interoperability Framework: Volume 8, Reference Architecture Interface

Version 0.1

NIST Big Data Public Working Group (NBD-PWG)
Reference Architecture Subgroup
National Institute of Standards and Technology
Gaithersburg, MD 20899

http://dx.doi.org/10.6028/NIST.SP.1500-8

October 2016

U. S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Under Secretary of Commerce for Standards and Technology and Director

National Institute of Standards and Technology (NIST) Special Publication 1500-8
20 pages (October 19, 2016)

NIST Special Publication series 1500 is intended to capture external perspectives related to NIST standards, measurement, and testing-related efforts. These external perspectives can come from industry, academia, government, and others. These reports are intended to document external perspectives and do not represent official NIST positions.

Comments on this publication may be submitted to Wo Chang

National Institute of Standards and Technology
Attn: Wo Chang, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8900) Gaithersburg, MD 20899-8930
Email: SP1500comments@nist.gov

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at NIST promotes the U.S. economy and public welfare by providing technical leadership for the Nations measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology (IT). ITLs responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. This document reports on ITLs research, guidance, and outreach efforts in IT and its collaborative activities with industry, government, and academic organizations.

Abstract

Big Data is a term used to describe the large amount of data in the networked, digitized, sensor-laden, information-driven world. While opportunities exist with Big Data, the data can overwhelm traditional technical approaches, and the growth of data is outpacing scientific and technological advances in data analytics. To advance progress in Big Data, the NIST Big Data Public Working Group (NBD-PWG) is working to develop consensus on important fundamental concepts related to Big Data. The results are reported in the NIST Big Data Interoperability Framework series of volumes. This volume, Volume 6, summarizes the work performed by the NBD-PWG to characterize Big Data from an architecture perspective, presents the NIST Big Data Reference Architecture (NBDRA) conceptual model, and discusses the components and fabrics of the NBDRA.

Keywords

Application Provider; Big Data; Big Data characteristics; Data Consumer; Data Provider; Framework Provider; Management Fabric; reference architecture; Security and Privacy Fabric; System Orchestrator; use cases.

Acknowledgements

This document reflects the contributions and discussions by the membership of the NBD-PWG, co- chaired by Wo Chang of the NIST ITL, Robert Marcus of ET-Strategies, and Chaitanya Baru, University of California San Diego Supercomputer Center.

The document contains input from members of the NBD-PWG: . NIST SP1500-8, Version 1 has been collaboratively authored by the NBD-PWG. As of the date of publication, there are over . NIST acknowledges the specific contributions to this volume by the following NBD-PWG members: .

The editors for this document were , and Wo Chang.

Add the members of the Working group

Add number of contributors

Add specific members

Add the editors

1 Executive Summary (Draft 0.0001)

This document, NIST Big Data Interoperability Framework: Volume 8, Reference Architecture Interfaces, was prepared by the NIST Big Data Public Working Group (NBD-PWG) Reference Architecture Subgroup to establish the operational interfaces for management interactions and dataflow with needed resources between Reference Architecture components which are defined in the Volume 6 Reference Architecture.

These interfaces, referred to as the NIST Big Data Reference Architecture Interfaces (NBDRAI), was crafted by addressing the model (structure and representation) and protocol (execution mechanism) to the control of the Big Data application lifecycle under the direction of the System Orchestrator to other NBDRA components. Specific interfaces and the interactions between NBDRA components shall be described and defined.

The NIST Big Data Interoperability Framework consists of seven volumes, each of which addresses a specific key topic, resulting from the work of the NBD-PWG. The seven volumes are:

- Volume 1, Definitions
- Volume 2, Taxonomies
- Volume 3, Use Cases and General Requirements
- Volume 4, Security and Privacy
- Volume 5, Architectures White Paper Survey
- Volume 6, Reference Architecture
- Volume 7, Standards Roadmap
- Volume 8, Interfaces

The NIST Big Data Interoperability Framework will be released in three versions, which correspond to the three development stages of the NBD-PWG work. The three stages aim to achieve the following with respect to the NIST Big Data Reference Architecture (NBDRA).

Stage 1: Identify the high-level Big Data reference architecture key components, which are technology–, infrastructure–, and vendor–agnostic.

Stage 2: Define general interfaces between the NBDRA components.

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces.

Potential areas of future work for the Subgroup during stage 2 are highlighted in Section 1.5 of this volume. The current effort documented in this volume reflects concepts developed within the rapidly evolving field of Big Data.

2 Introduction

2.1 Background

There is broad agreement among commercial, academic, and government leaders about the remarkable potential of Big Data to spark innovation, fuel commerce, and drive progress. Big Data is the common term used to describe the deluge of data in todays networked, digitized, sensor-laden, and information- driven world. The availability of vast data resources carries the potential to answer questions previously out of reach, including the following:

- How can a potential pandemic reliably be detected early enough to intervene?
- Can new materials with advanced properties be predicted before these materials have ever been synthesized?
- How can the current advantage of the attacker over the defender in guarding against cyber- security threats be reversed?

There is also broad agreement on the ability of Big Data to overwhelm traditional approaches. The growth rates for data volumes, speeds, and complexity are outpacing scientific and technological advances in data analytics, management, transport, and data user spheres. Despite widespread agreement on the inherent opportunities and current limitations of Big Data, a lack of consensus on some important fundamental questions continues to confuse potential users and stymie progress. These questions include the following:

- What attributes define Big Data solutions?
- How is Big Data different from traditional data environments and related applications?
- What are the essential characteristics of Big Data environments?
- How do these environments integrate with currently deployed architectures?
- What are the central scientific, technological, and standardization challenges that need to be addressed to accelerate the deployment of robust Big Data solutions?

Within this context, on March 29, 2012, the White House announced the Big Data Research and Development Initiative. The initiatives goals include helping to accelerate the pace of discovery in science and engineering, strengthening national security, and transforming teaching and learning by improving the ability to extract knowledge and insights from large and complex collections of digital data.

Six federal departments and their agencies announced more than $200 million in commitments spread across more than 80 projects, which aim to significantly improve the tools and techniques needed to access, organize, and draw conclusions from huge volumes of digital data. The initiative also challenged industry, research universities, and nonprofits to join with the federal government to make the most of the opportunities created by Big Data.

Motivated by the White House initiative and public suggestions, the National Institute of Standards and Technology (NIST) has accepted the challenge to stimulate collaboration among industry professionals to further the secure and effective adoption of Big Data. As one result of NISTs Cloud and Big Data Forum held on January 1517, 2013, there was strong encouragement for NIST to create a public working group for the development of a Big Data Interoperability Framework. Forum participants noted that this roadmap should define and prioritize Big Data requirements, including interoperability, portability, reusability, extensibility, data usage, analytics, and technology infrastructure. In doing so, the roadmap would accelerate the adoption of the most secure and effective Big Data techniques and technology. On June 19, 2013, the NIST Big Data Public Working Group (NBD-PWG) was launched with extensive participation by industry, academia, and government from across the nation. The scope of the NBD-PWG involves forming a community of interests from all sectorsincluding industry, academia, and governmentwith the goal of developing consensus on definitions, taxonomies, secure reference architectures, security and privacy, andfrom thesea standards roadmap. Such a consensus would create a vendor-neutral, technology- and infrastructure-independent framework that would enable Big Data stakeholders to identify and use the best analytics tools for their processing and visualization requirements on the most suitable computing platform and cluster, while also allowing value-added from Big Data service providers.

The NIST Big Data Interoperability Framework consists of seven volumes, each of which addresses a specific key topic, resulting from the work of the NBD-PWG. The seven volumes are:

- Volume 1, Definitions
- Volume 2, Taxonomies
- Volume 3, Use Cases and General Requirements
- Volume 4, Security and Privacy
- Volume 5, Architectures White Paper Survey
- Volume 6, Reference Architecture
- Volume 7, Standards Roadmap
- Volume 8, Interfaces

The NIST Big Data Interoperability Framework will be released in three versions, which correspond to the three stages of the NBD-PWG work. The three stages aim to achieve the following with respect to the NIST Big Data Reference Architecture (NBDRA.)

Stage 1: Identify the high-level Big Data reference architecture key components, which are technology-, infrastructure-, and vendor-agnostic;

Stage 2: Define general interfaces between the NBDRA components; and

Stage 3: Validate the NBDRA by building Big Data general applications through the general interfaces.

Potential areas of future work for the Subgroup during stage 2 are highlighted in Section 1.5 of this volume. The current effort documented in this volume reflects concepts developed within the rapidly evolving field of Big Data.

2.2 Scope and Objectives of the Reference Architectures

SUBGROUP Reference architectures provide an authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions. Reference architectures generally serve as a foundation for solution architectures and may also be used for comparison and alignment of instantiations of architectures and solutions. The goal of the NBD-PWG Reference Architecture Subgroup is to develop an open reference architecture for Big Data that achieves the following objectives:

- Provides a common language for the various stakeholders;
- Encourages adherence to common standards, specifications, and patterns;
- Provides consistent methods for implementation of technology to solve similar problem sets;
- Illustrates and improves understanding of the various Big Data components, processes, and systems, in the context of a vendor- and technology-agnostic Big Data conceptual model;
- Provides a technical reference for U.S. government departments, agencies, and other consumers to understand, discuss, categorize, and compare Big Data solutions; and
- Facilitates analysis of candidate standards for interoperability, portability, reusability, and extendibility.

The NBDRA is a high-level conceptual model crafted to serve as a tool to facilitate open discussion of the requirements, design structures, and operations inherent in Big Data. The NBDRA is intended to facilitate the understanding of the operational intricacies in Big Data. It does not represent the system architecture of a specific Big Data system, but rather is a tool for describing, discussing, and developing system- specific architectures using a common framework of reference. The model is not tied to any specific vendor products, services, or reference implementation, nor does it define prescriptive solutions that inhibit innovation. The NBDRA does not address the following:

- Detailed specifications for any organizations operational systems;
- Detailed specifications of information exchanges or services; and
- Recommendations or standards for integration of infrastructure products.

2.3 Report Production

A wide spectrum of Big Data architectures have been explored and developed as part of various industry, academic, and government initiatives. The development of the NBDRA and material contained in this volume involved the following steps:

1. Announce that the NBD-PWG Reference Architecture Subgroup is open to the public to attract and solicit a wide array of subject matter experts and stakeholders in government, industry, and academia;
2. Gather publicly available Big Data architectures and materials representing various stakeholders, different data types, and diverse use cases;

3. Examine and analyze the Big Data material to better understand existing concepts, usage, goals, objectives, characteristics, and key elements of Big Data, and then document the findings using NISTs Big Data taxonomies model (presented in NIST Big Data Interoperability Framework: Volume 2, Taxonomies); and
4. Develop a technology-independent, open reference architecture based on the analysis of Big Data material and inputs received from other NBD-PWG subgroups.

2.4 Report Structure

The organization of this document roughly corresponds to the process used by the NBD-PWG to develop the NBDRA. Following the introductory material presented in Section 1, the remainder of this document is organized as follows:

- Section 2 contains high-level, system requirements in support of Big Data relevant to the design of the NBDRA and discusses the development of these requirements.
- Section 3 presents the generic, technology-independent NBDRA conceptual model.
- Section 4 discusses the five main functional components of the NBDRA.
- Section 5 describes the system and life cycle management considerations related to the NBDRA management fabric.
- Section 6 briefly introduces security and privacy topics related to the security and privacy fabric of the NBDRA.
- Appendix A summarizes deployment considerations.
- Appendix B lists the terms and definitions in this document.
- Appendix C provides examples of Big Data logical data architecture options.
- Appendix D defines the acronyms used in this document.
- Appendix E lists general resources that provide additional information on topics covered in this document and specific references in this document.

2.5 Future Work on this Volume (Next Step is Validation)

This document (Version 1) presents the overall NBDRA components and fabrics with high-level description and functionalities. Version 2 activities will focus on the definition of general interfaces between the NBDRA components by performing the following:

- Select use cases from the 62 (51 general and 11 security and privacy) submitted use cases or other, to be identified, meaningful use cases;
- Work with domain experts to identify workflow and interactions among the NBDRA components and fabrics;
- Explore and model these interactions within a small-scale, manageable, and well-defined confined environment; and

- Aggregate the common data workflow and interactions between NBDRA components and fabrics and package them into general interfaces. Version 3 activities will focus on validation of the NBDRA through the use of the defined NBDRA general interfaces to build general Big Data applications. The validation strategy will include the following:
- Implement the same set of use cases used in Version 2 by using the defined general interfaces;
- Identify and implement a few new use cases outside the Version 2 scenarios; and
- Enhance general NBDRA interfaces through lessons learned from the implementations in Version 3 activities.

The general interfaces developed during Version 2 activities will offer a starting point for further refinement by any interested parties and is not intended to be a definitive solution to address all implementation needs.

3 High-Level NBDRA Interface Requirements

The Volume 6 Reference Architecture document provides a list of comprehensive high-level reference architecture requirements. To enable interoperability between the NBDRA components, a list of well- defined NBDRA interface is needed.

3.1 System Orchestration Requirements

3.2 Application Providers Requirements

3.3 Data Providers Requirements

3.4 Framework Providers Requirements

3.5 Data Consumers Requirements

3.6 Security and Privacy Fabric Requirements

3.7 System Management Requirements

System Orchestration Requirements

Application Providers Requirements

Data Providers Requirements

Framework Providers Requirements

Data Consumers Requirements

Security and Privacy Fabric Requirements

System Management Requirements

4 NBDRA Interface Approach

NBDRA Interface Approach

- Orchestrate via Application Provider to other RA components o System Orchestrator (Data Scientists) uses the BD Application Provider as the command center to orchestrate dataflow from Data Provider, carryout the BD application lifecycle with the help of the BD Framework Provider, and enable Data Consumer to consume Big Data processing results
- Agnostic can plug-in any specific technologies, analytics, IaaSs to support Big Data applications at any environments with and without many CPUs/Cores/GPUs/other accelerators: o Laptop/desktop o Server o Data centers o clouds
- Customizable parameters in two-levels o High-level: with defaults parameters o Low-level: call out specific parameters and environmental needs
- Packaging analytics algorithms/tools (as payload) via standard interface (as transport) to achieve interoperability across application domains with the goals for analytics to be o Re-usable available analytics packages for adoption o Deployable customizable analytics tools for deployment o Operational adjustable

NBDRA Components To develop the use cases, publically available information was collected for various Big Data architectures in nine broad areas, or application domains. Participants in the NBD-PWG Use Case and Requirements Subgroup and other interested parties provided the use case details via a template, which helped to standardize the responses and facilitate subsequent analysis and comparison of the use cases.

Figure 2: NIST Big Data Reference Architecture (NBDRA)

5 NBDRA Functional Interfaces

As outlined in Section 3, the five main functional components of the NBDRA represent the different technical roles within a Big Data system. The functional components are listed below and discussed in subsequent sections.

System Orchestrator: Defines and integrates the required data application activities into an operational vertical system;

Big Data Application Provider: Executes a data life cycle to meet security and privacy requirements as well as System Orchestrator-defined requirements;

Data Provider: Introduces new data or information feeds into the Big Data system;

Big Data Framework Provider: Establishes a computing framework in which to execute certain transformation applications while protecting the privacy and integrity of data; and

Data Consumer: Includes end users or other systems that use the results of the Big Data Application Provider.

5.1 System Orchestrator to BD Application Provider INTERFACE

The System Orchestrator role includes defining and integrating the required data application activitieS Into an operational vertical system. Typically, the System Orchestrator involves a collection of more specific roles, performed by one or more actors, which manage and orchestrate the operation of the Big Data system. These actors may be human components, software components, or some combination of the two. The function of the System Orchestrator is to configure and manage the other components of the Big Data architecture to implement one or more workloads that the architecture is designed to execute. The workloads managed by the System Orchestrator may be assigning/provisioning framework components to individual physical or virtual nodes at the lower level, or providing a graphical user interface that supports the specification of workflows linking together multiple applications and components at the higher level. The System Orchestrator may also, through the Management Fabric, monitor the workloads and system to confirm that specific quality of service requirements are met for each workload, and may actually elastically assign and provision additional physical or virtual resources to meet workload requirements resulting from changes/surges in the data or number of users/transactions.

5.2 BD Application Provider Interface

The Big Data Application Provider role executes a specific set of operations along the data life cycle to meet the requirements established by the System Orchestrator, as well as meeting security and privacy requirements. The Big Data Application Provider is the architecture component that encapsulates the business logic and functionality to be executed by the architecture. The Big Data Application Provider activities include the following:

- Collection
- Preparation
- Analytics
- Visualization
- Access

5.2.1 Collection

In general, the collection activity of the Big Data Application Provider handles the interface with the Data Provider. This may be a general service, such as a file server or web server configured by the System Orchestrator to accept or perform specific collections of data, or it may be an application-specific service designed to pull data or receive pushes of data from the Data Provider. Since this activity is receiving data at a minimum, it must store/buffer the received data until it is persisted through the Big Data Framework Provider. This persistence need not be to physical media but may simply be to an in-memory queue or other service provided by the processing frameworks of the Big Data Framework Provider. The collection activity is likely where the extraction portion of the Extract, Transform, Load (ETL)/Extract, Load, Transform (ELT) cycle is performed. At the initial collection

stage, sets of data (e.g., data records) of similar structure are collected (and combined), resulting in uniform security, policy, and other considerations. Initial metadata is created (e.g., subjects with keys are identified) to facilitate subsequent aggregation or look-up methods.

5.2.2 Preparation

The preparation activity is where the transformation portion of the ETL/ELT cycle is likely performed, although analytics activity will also likely perform advanced parts of the transformation. Tasks performed by this activity could include data validation (e.g., checksums/hashes, format checks), cleansing (e.g., eliminating bad records/fields), outlier removal, standardization, reformatting, or encapsulating. This activity is also where source data will frequently be persisted to archive storage in the Big Data Framework Provider and provenance data will be verified or attached/associated. Verification or attachment may include optimization of data through manipulations (e.g., deduplication) and indexing to optimize the analytics process. This activity may also aggregate data from different Data Providers, leveraging metadata keys to create an expanded and enhanced data set.

5.2.3 Analytics

The analytics activity of the Big Data Application Provider includes the encoding of the low-level business logic of the Big Data system (with higher-level business process logic being encoded by the System Orchestrator). The activity implements the techniques to extract knowledge from the data based on the requirements of the vertical application. The requirements specify the data processing algorithms for processing the data to produce new insights that will address the technical goal. The analytics activity will leverage the processing frameworks to implement the associated logic. This typically involves the activity providing software that implements the analytic logic to the batch and/or streaming elements of the processing framework for execution. The messaging/communication framework of the Big Data Framework Provider may be used to pass data or control functions to the application logic running in the processing frameworks. The analytic logic may be broken up into multiple modules to be executed by the processing frameworks which communicate, through the messaging/communication framework, with each other and other functions instantiated by the Big Data Application Provider.

5.2.4 Visualization

The visualization activity of the Big Data Application Provider prepares elements of the processed data and the output of the analytic activity for presentation to the Data Consumer. The objective of this activity is to format and present data in such a way as to optimally communicate meaning and knowledge. The visualization preparation may involve producing a text-based report or rendering the analytic results as some form of graphic. The resulting output may be a static visualization and may simply be stored through the Big Data Framework Provider for later access. However, the visualization activity frequently interacts with the access activity, the analytics activity, and the Big Data Framework Provider (processing and platform) to provide interactive visualization of the data to the Data Consumer based on parameters provided to the access activity by the Data Consumer. The visualization activity may be completely application-implemented,

leverage one or more application libraries, or may use specialized visualization processing frameworks within the Big Data Framework Provider.

5.2.5 Access

The access activity within the Big Data Application Provider is focused on the communication/interaction with the Data Consumer. Similar to the collection activity, the access activity may be a generic service such as a web server or application server that is configured by the System Orchestrator to handle specific requests from the Data Consumer. This activity would interface with the visualization and analytic activities to respond to requests from the Data Consumer (who may be a person) and uses the processing and platform frameworks to retrieve data to respond to Data Consumer requests. In addition, the access activity confirms that descriptive and administrative metadata and metadata schemes are captured and maintained for access by the Data Consumer and as data is transferred to the Data Consumer. The interface with the Data Consumer may be synchronous or asynchronous in nature and may use a pull or push paradigm for data transfer.

5.3 BD Application Provider to Data Provider Interface

The Data Provider role introduces new data or information feeds into the Big Data system for discovery, access, and transformation by the Big Data system. New data feeds are distinct from the data already in use by the system and residing in the various system repositories. Similar technologies can be used to access both new data feeds and existing data. The Data Provider actors can be anything from a sensor, to a human inputting data manually, to another Big Data system.

5.4 BD Application Provider to Framework Provider Interface

The Big Data Framework Provider typically consists of one or more hierarchically organized instances of the components in the NBDRA IT value chain (Figure 2). There is no requirement that all instances at a given level in the hierarchy be of the same technology. In fact, most Big Data implementations are hybrids that combine multiple technology approaches in order to provide flexibility or meet the complete range of requirements, which are driven from the Big Data Application Provider.

5.4.1 Infrastructure Frameworks

This Big Data Framework Provider element provides all of the resources necessary to host/run the activities of the other components of the Big Data system. Typically, these resources consist of some combination of physical resources, which may host/support similar virtual resources. These resources are generally classified as follows:

Virtual Cluster


```
{
    "profile":{
    "firstname": "Gregor",
    "Lastname": "von Laszewski",
    "e−mail": "laszewski@gmail.com",
    "username": "gregor"
    },
    "cluster": {
    "name": "myCLuster",
    "label": "c0",
```

```
    "provider": ["openstack",
                 "aws",
                 "azure",
                 "eucalyptus"],
    "endpoint": {
        "url": "https",
        "passwd": "secret"
    }
    },
    "computer": {
    "name": "myComputer",
    "label": "server-001",
    "ip": "127.0.0.1",
    "memoryGB": 16
    }
}
```

5.4.2 Data Platform Frameworks

Data Platform Frameworks provide for the logical data organization and distribution combined with the associated access application programming interfaces (APIs) or methods. The frameworks may also

5.4.3 Processing Frameworks

The processing frameworks for Big Data provide the necessary infrastructure software to support implementation of applications that can deal with the volume, velocity, variety, and variability of data. Processing frameworks define how the computation and processing of the data is organized. Big Data applications rely on various platforms and technologies to meet the challenges of scalable data analytics and operation.

5.4.4 Messaging/Communications Frameworks

Messaging and communications frameworks have their roots in the High Performance Computing (HPC) environments long popular in the scientific and research communities. Messaging/Communications Frameworks were developed to provide APIs for the reliable queuing, transmission, and receipt of data

5.4.5 Resource Management Framework

As Big Data systems have evolved and become more complex, and as businesses work to leverage limited computation and storage resources to address a broader range of applications and business challenges, the requirement to effectively manage those resources has grown significantly. While tools for resource management and elastic computing have expanded and matured in response to the needs of cloud providers and virtualization technologies, Big Data introduces unique requirements for these tools. However, Big Data frameworks tend to fall more into a distributed computing paradigm, which presents additional challenges.

5.5 BD Application Provider to Data Consumer Interface

Similar to the Data Provider, the role of Data Consumer within the NBDRA can be an actual end user or another system. In many ways, this role is the mirror image of the Data Provider, with the entire Big Data framework appearing like a Data Provider to the Data Consumer. The activities associated with the Data Consumer role include the following:

- Search and Retrieve

- Download
- Analyze Locally
- Reporting
- Visualization
- Data to Use for Their Own Processes

6 NBDRA Security and Privacy Fabric

INTERFACE

Security and privacy

6.1 Security and Privacy

The characteristics of Big Data pose system management challenges on traditional management Big Data Life cycle Management

7 NBDRA Management Fabric Interfaces

Management Fabric

7.1 System Management

The characteristics of Big Data pose system management challenges on traditional management Big Data Life cycle Management

8 Conclusion

Conclusion

A Acronyms

get acronyms from nist

ACID Atomicity, Consistency, Isolation, Durability

API application programming interface

ASCII American Standard Code for Information Interchange

BASE Basically Available, Soft state, Eventual consistency

NIST National Institute of Standards

NBD-PWG NIST Big Data Public Working Group

ITL Information Technology Laboratory

NBDRA NIST Big Data Reference Architecture

NBDRAI NIST Big Data Reference Architecture Interface

DevOps a clipped compound of "software DEVelopment" and "information technology OPerationS"

IaaS Infrastructure as a Service

SaaS Software as a Service

OS Operating System

WWW World Wide Web

HTTP HyperText Transfer Protocol

HTTPS HTTP Secure

REST REpresentational State Transfer

B Resources and References

cleanup section refs and integrate with bibtex

B.1 General Resources

The following resources provide additional information related to Big Data architecture.

Big Data Public Working Group, NIST Big Data Program, National Institute for Standards and Technology, June 26, 2013, `http://bigdatawg.nist.gov`.

Doug Laney, 3D Data Management: Controlling Data Volume, Velocity, and Variety, Gartner, February 6, 2001, [1].

integrate section refs

B.2 Document References

Contributors are members of the NIST Big Data Public Working Group who dedicated great effort to prepare and substantial time on a regular basis to research and development in support of this document. Many of the architecture use cases were originally collected by the NBD-PWG Use Case and Requirements Subgroup and can be accessed at http://bigdatawg.nist.gov/usecases.php.

The White House Office of Science and Technology Policy, Big Data is a Big Deal, OSTP Blog, accessed February 21, 2014, `http://www.whitehouse.gov/blog/2012/03/29/big-data-big-deal`.

Office of the Assistant Secretary of Defense, Reference Architecture Description, U.S. Department of Defense, June 2010, `http://dodcio.defense.gov/Portals/0/Documents/DIEA/Ref_Archi_Description_Final_v1_18Jun10.pdf`.

References

[1] D. Laney, "3D Data Management: Controlling Data Volume, Velocity, and Variety," Gartner, Tech. Rep., 2001. [Online]. Available: http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Data-Volume-Velocity-and-Variety.pdf

[2] R. Fielding, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content," Internet Requests for Comments, RFC Editor, RFC 7231, 2014. [Online]. Available: https://tools.ietf.org/html/rfc7231

A Responses

Each call to the API should return the appropriate response, as described in RFC7231[2].

```
200: Ok
201: Created
204: No content
400: Bad Request
401: Unauthorized
404: Not found
405: Method not allowed
406: Not acceptable
409: Conflict
500: Internal server error
501: Not implemented
503: Service unavailable
507: Insufficient storage
```

B Virtual Cluster

Example

```
{
    "profile":{
    "firstname": "Gregor",
    "Lastname": "von Laszewski",
    "e-mail": "laszewski@gmail.com",
    "username": "gregor"
    },
    "cluster": {
    "name": "myCLuster",
    "label": "c0",
    "provider": ["openstack",
                "aws",
                "azure",
                "eucalyptus"],
    "endpoint": {
        "url": "https",
        "passwd": "secret"
    }
    },
    "computer": {
    "name": "myComputer",
    "label": "server-001",
    "ip": "127.0.0.1",
    "memoryGB": 16
    }
}
```

Type

```
profile = {
    'schema': {
        'firstname': {
            'type': 'string'
        },
        'Lastname': {
            'type': 'string'
        },
        'e—mail': {
            'type': 'string'
        },
        'username': {
            'type': 'string'
        }
    }
}

cluster = {
    'schema': {
        'name': {
            'type': 'string'
        },
        'label': {
            'type': 'string'
        },
        'provider': {
            'type': 'list',
            'schema': {
                'type': 'string'
            }
        },
        'endpoint': {
            'type': 'dict',
            'schema': {
                'url': {
                    'type': 'string'
                },
                'passwd': {
                    'type': 'string'
                }
            }
        }
    }
}

computer = {
    'schema': {
        'name': {
            'type': 'string'
        },
        'label': {
            'type': 'string'
        },
        'ip': {
            'type': 'string'
        },
        'memoryGB': {
            'type': 'integer'
        }
    }
}
```

C Layers

Example

```
{
    "requires": {
        "java": {"version": "1.8", "implementation": "OpenJDK",
                 "supervisord": {},
                 "zookeeper" : {}
                }
    },

    "deployers": {
        "ansible": "git://github.com/cloudmesh_roles/hadoop"
    },

    "parameters": {
        "use_hdfs": true,
        "use_yarn": false,
```

```
        "num_journalnodes": 1,
        "num_namenodes": 1,
        "num_datanodes": 1,
        "num_resourcemanagers": 1,
        "num_historyservers": 1
    }
}
```

Type

```
requires = {
    'schema': {
        'java': {
            'type': 'dict',
            'schema': {
                'version': {
                    'type': 'string'
                },
                'implementation': {
                    'type': 'string'
                },
                'supervisord': {
                    'type': 'dict',
                    'schema': {}
                },
                'zookeeper': {
                    'type': 'dict',
                    'schema': {}
                }
            }
        }
    }
}

deployers = {
    'schema': {
        'ansible': {
            'type': 'string'
        }
    }
}

parameters = {
    'schema': {
        'use_hdfs': {
            'type': 'boolean'
        },
        'use_yarn': {
            'type': 'boolean'
        },
        'num_journalnodes': {
            'type': 'integer'
        },
        'num_namenodes': {
            'type': 'integer'
        },
        'num_datanodes': {
            'type': 'integer'
        },
        'num_resourcemanagers': {
            'type': 'integer'
        },
        'num_historyservers': {
            'type': 'integer'
        }
    }
}
```

D Deployment

Example

```
{
    "cluster": {"name": "myCluster"},
    "provider": {"name": "aws"},
    "stack": { "layers": ["hadoop", "spark", "postgresql"]}
}
```

Type

```
cluster = {
    'schema': {
        'name': {
            'type': 'string'
        }
    }
}

provider = {
    'schema': {
        'name': {
            'type': 'string'
        }
    }
}

stack = {
    'schema': {
        'layers': {
            'type': 'list',
            'schema': {
                'type': 'string'
            }
        }
    }
}
```

E API Endpoints

E.1 Virtual Cluster

```
/cluster:
  description: This endpoint provides methods for operating on the full set of Clusters.
  get:
    description: List available cluster
    responses: [200, 404]
    result: list of Cluster objects
  post:
    description: Create and launch cluster
    responses: [201, 406]
    result: 'Location' header stores the link to resource
  delete:
    description: Delete all clusters
    responses: [200, 400]
    result: list of Cluster objects that were deleted

/cluster/{id}:
  description: This endpoint provides methods for manipulating a single Cluster.
  get:
    description: View the cluster
    responses: [200, 404]
    result: a Cluster object
  delete:
    description: Destroy and delete the cluster
    responses: [200, 404]
    result: the Cluster object that as deleted
  patch:
```

```
    description: Update components of the cluster. Useful for properties
    responses: [203, 406]
    result: none
  put:
    description: Replace the definition of a cluster, by tearing down and relaunching
    responses: [204, 404]
    result: none

/cluster/{id}/inventory:
  description: |
    This endpoing allows inventory files to be retreived for the
    cluster.  The inventory files contains the IP address and log in
    information for the cluster and may be used by configuration
    managers such as Ansible to manage content on the cluster.
  get:
    description: Get the Ansible inventory for the cluster
    responses: [200, 404]
    result: string, the inventory
```

E.2 Layers

```
/stack:
  get: List available stacks
  post: Create a new composition
  delete: Delete all stack compositions

/stack/{id}:
  get: Retrieve a composition
  put: Replace a composition
  patch: Modify a composition. Primarily intended to add/remove/modify layers
  delete: Delete a composition
```

E.3 Deployments

```
/deploy:
  get: List current stack deployments
  post: Create a new deployments
  delete: Delete all deployments

/deploy/{id}:
  get: Retrieve a deployment
  put: Replace a deployment
  delete: Delete a deployment

/deploy/{id}/status:
  get: Retrive the status of a deployment
```