White Paper

# Automation for Big Data Infrastructure and Applications

## BlueData EPIC™ Software Platform

This technical white paper describes how the BlueData EPIC platform automates the set up, configuration, deployment, and management of Big Data infrastructure and applications.

AUTHORED BY ANTHONY HERNANDEZ -- (415)786-2081 -- anthony94122@outlook.com

# Table of Contents

AUTHORED BY ANTHONY HERNANDEZ -- (415)786-2081 -- anthony94122@outlook.com

# 1. Introduction

The **BlueData Elastic Private Instant Clusters** software platform (referred to as **"EPIC"** throughout this white paper) enables enterprises to implement Big-Data-as-a-Service (BDaaS) running on their own datacenter infrastructure, in the public cloud, or in a hybrid architecture.

Automation is a key component of any "as-a-service" solution. BDaaS automation can greatly reduce the cost and effort required to install and maintain Big Data clusters. BlueData EPIC supports automation at all levels of the Big Data infrastructure stack, including:

- Big Data platform infrastructure
- Big Data application environment
- Big Data application-specific infrastructure
- Pipeline infrastructure for multiple Big Data applications

Automation is a weighty decision, especially for a complex application that can impact the entire enterprise. The speed with which the organization accepts the utility and correctness of the automated process limits how quickly automation can be introduced to an enterprise. BlueData EPIC lets you control the level of automation used to manage the Big Data applications in your specific environment.

Users of the BlueData EPIC software platform can start by manually controlling their Big Data virtual clusters. BlueData EPIC stands up Linux nodes (using managed Docker containers in a secure network environment) that are exact replicas of their existing physical hosts. Users can then install and configure Big Data software packages using the same familiar commands. Users who are more comfortable with automation can allow BlueData EPIC to both create the Linux environments and then install and integrate the Big Data software packages into existing enterprise services such as Active Directory, Kerberos, and Single Sign-On. Users who are at ease with automation can choose to let BlueData EPIC configure their Big Data clusters completely. Bringing up a fully-configured Big Data cluster that is ready for use by a data scientist is as simple as a few mouse clicks in the EPIC interface.

BlueData understands that the application execution environment inside a modern enterprise is not a "one size fits all" proposition. Two enterprises may run the exact same Big Data application; however, how the application is configured and run within each enterprise can vary greatly. This white paper describes the power and flexibility of the BlueData EPIC platform for automating the set up, configuration, deployment, and management of your Big Data applications in a containerized BDaaS environment.

# 2. The Power of Automation

One of the classic definitions of *automation* is “a mechanical device, operated electronically, that functions automatically, without continuous input from an operator.” BlueData modifies this definition slightly within the BDaaS context: We say that automation means replacing the manual execution of tedious, repetitious, and difficult deployment and configuration tasks with intelligent software. The process of installing, configuring, and maintaining Big Data clusters is full of such manual tasks, and is therefore ripe for automation.

## Time to Value

TechTarget defines *time to value* as the period of time between a request for a specific value and the initial delivery of that value[1]. In this context, the value is a desirable quantifiable (tangible) or abstract (intangible) business goal.

With BDaaS, the request is for an operational Big Data environment. Request delivery is complete when a data scientist can use the Big Data environment to access the necessary data using familiar tools in a secure and reliable manner. Delivering such a Big Data environment involves many steps. Manual steps or unexpected stoppages in the creation process delay the delivery of the Big Data environment and thus the time to value. BlueData EPIC minimizes (or altogether eliminates) these delays.

Figure 1 illustrates a typical Big Data deployment with delays required for manual steps, versus the rapid time to value for a BlueData EPIC deployment with automation.

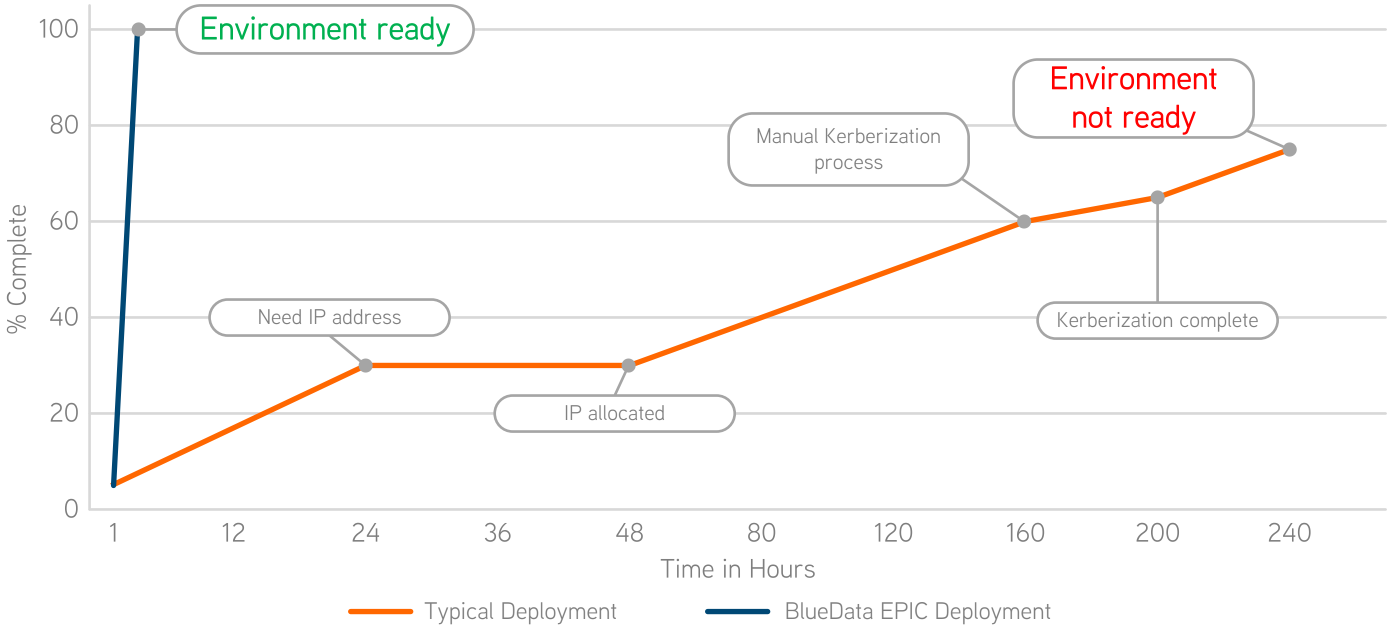


*Figure 1: Time to value in a typical Big Data deployment compared to BlueData EPIC*

## Manual versus Automation

It is tempting to perform complex and infrequent tasks manually, instead of taking the time to build an automated process. The rationale behind this temptation is that the infrequency of the task does not justify investing in automation; however, this rationale does not apply to Big Data deployments.

The use of Big Data applications is rapidly accelerating within most enterprises, and once-rare tasks associated with Big Data application infrastructure are becoming commonplace. It is therefore better to invest in automation up front, thereby minimizing the future need for manual intervention throughout the life cycle of the Big Data application infrastructure.

[1] *Source: http://whatis.techtarget.com/definition/time-to-value-TtV*

AUTHORED BY ANTHONY HERNANDEZ -- (415)786-2081 -- anthony94122@outlook.com

# 3. Types of Big Data Infrastructure Automation

Infrastructure automation takes many forms and is implemented at many levels in the software stack. This section describes how the BlueData EPIC software platform addresses the various layers of infrastructure automation for Big Data applications.

## Big Data Platform Infrastructure

Within the context of *Big Data platform infrastructure*, automation refers to the automated provisioning and management of the required infrastructure resources, regardless of the specific Big Data application that will be using those resources.

BlueData EPIC treats common platform infrastructure resources (CPU, RAM, and storage) as a bundle called a *flavor*. Within BlueData EPIC, a virtual node is a Docker container that is created when creating a Big Data cluster. Each virtual node is automatically deployed with a node flavor (i.e. a specific allocation of CPU, RAM, and storage) for that container.

Several aspects of the application-independent Big Data platform infrastructure extend beyond the basic resources defined by a node flavor, including:

- **Container placement:** Placing containers for Big Data applications is frequently more involved than simply selecting on-premises datacenter infrastructure versus a public cloud service (e.g. AWS), because the cluster must be placed on a host that meets all application-specific requirements. For example, TensorFlow requires GPU resources, and this application must therefore be placed on a host that includes GPU hardware. Additional factors come into play, including (but not limited to) CPU and RAM requirements, the tenant the container belongs to, and enterprise-specific placement rules.
- **Hybrid resource management:** BlueData EPIC supports managing Big Data clusters on either a public cloud service and/or in on-premises enterprise datacenters. A single BlueData EPIC deployment that uses both public cloud and on-premises resources is referred to as a *hybrid* environment. BlueData EPIC can automatically and transparently deploy a Big Data cluster in either environment, according to user-determined placement criteria.

## Big Data Application Environment

The term *Big Data application environment infrastructure* refers to those services, processes, and procedures that all applications running within an enterprise data center environment must conform to. BlueData EPIC supports automatic integration of Big Data clusters with these standard enterprise processes, such as:

- **Docker registry:** BlueData EPIC can be integrated with an enterprise's secure Docker repository, which allows storing the Docker application images used by BlueData EPIC in a common location where they can be scanned and audited according to enterprise policies.
- **Remote storage system access:** Data security and the systems used to secure data are crucial components of Big Data analytics. The innovative DataTap technology provided by BlueData EPIC surfaces existing remote data storage devices to a Big Data cluster using the secure and well-understood Hadoop Distributed File System (HDFS) protocol.
- **Integration with enterprise services:** In this context, *enterprise services* refers to those systems that all applications running within a given enterprise must use in order to provide secure, seamless user access to corporate computing resources. These systems typically involve some form of user authorization and authentication. BlueData EPIC automatically configures Big Data clusters to integrate with these services (e.g. AD/LDAP, Kerberos, SSO), greatly reducing the time and effort required to install a Big Data cluster while ensuring that all systems are properly secured.
    - **Environment AD/LDAP Configuration:** Most enterprises employ either Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) to authorize users and manage authentication. BlueData EPIC management services can be configured to use an existing AD/LDAP service at installation time. Further, Big Data clusters managed by EPIC can be configured to use AD/LDAP to grant specific users the right to use specific Big Data clusters. This gives the enterprise a single, familiar "source of truth" for managing user access to Big Data compute resources and data.

- **Environment Kerberos Configuration:** Kerberos is "an authentication protocol for trusted hosts on untrusted networks" that controls user access to Big Data compute resources. Only users with the correct Kerberos credentials can submit Big Data jobs to Hadoop clusters protected by Kerberos. BlueData EPIC automatically configures Big Data clusters built from common Hadoop distributions (e.g. Cloudera, Hortonworks) to integrate with Kerberos.
- **Single Sign-On (SSO):** Single Sign-On (SSO) allows a user to authenticate themselves to an enterprise's security system once. Thereafter, they may access multiple computing applications without having to reenter their credentials. SSO is implemented by integrating with an enterprise-wide Identity Provider (IdP).
- **Data access Kerberos configuration:** Beyond protecting access to Big Data compute resources, Kerberos also implements user access control to data residing in a HDFS data lake. The BlueData EPIC DataTap functionality supports access to Kerberos-protected HDFS file systems using either proxy or passthrough authentication for maximum flexibility. Proxy configuration allows a single Kerberos principal to be used, and permits non-kerberized compute clusters to access data on a Kerberos-protected HDFS. Passthrough configuration supports the full HDFS Kerberos security architecture.
- **Key Management Service (KMS):** Most HDFS data lakes use Transparent Data Encryption (TDE) to protect data both at rest and transit. Key Management Service (KMS) software manages the encryption/decryption keys used by TDE systems. BlueData EPIC can configure a Big Data cluster to use a KMS service when accessing data via DataTap.

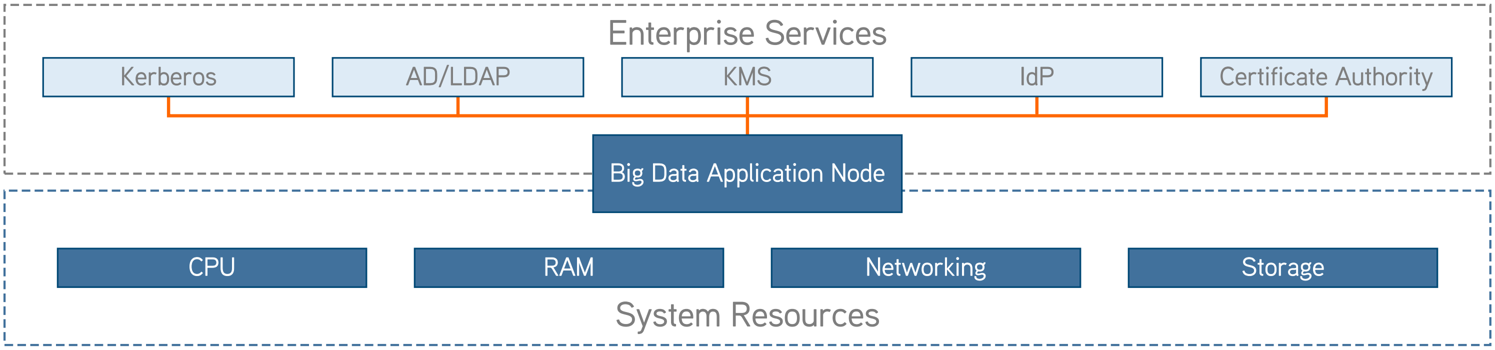


*Figure 2: Big Data node infrastructure and enterprise services.*

## Big Data Application-Specific Infrastructure

The term *application-specific infrastructure* refers to the Big Data frameworks (e.g. Hadoop, Spark, Kafka, Cassandra) that underly a specific Big Data application. With BlueData EPIC, the automated configuration of application-specific infrastructure begins after placing a Big Data application node (i.e. a Docker container) on a host and integrating it with enterprise services. This includes:

- **Big Data application configuration for container resources:** Each Big Data node deploys with a given flavor that defines the amount of CPU, memory, and storage capacity for that container. Different nodes in the same Big Data cluster can be assigned different node flavors based on their *node role* (see below).
    - **CPU:** Certain application configuration parameters must be set to be compatible with the CPU resources available to the node. For example, a node running a Hadoop YARN NodeManager service must have the parameter `{ctp0pqfgocpcigt0tguqwteg0erw/xeqtgu` set to less than or equal to the number of CPU cores allocated to that node. BlueData EPIC automatically configures CPU-dependent application-specific parameters based on the CPU resources defined in the flavor associated with each node.
    - **Memory:** Certain application configuration parameters must also be set to be compatible with the amount of RAM available to the node. For example, a node running a Hadoop YARN NodeManager service must have the parameter `{ctp0pqfgocpcigt0tguqwteg0ogoqt{/od` set to less than or equal to the MB of RAM allocated to that node. BlueData EPIC automatically configures the RAM-dependent application-specific parameters based on the memory resources defined in the flavor associated with each node.
    - **Storage:** Some Big Data applications require more, or less, local persistent storage. BlueData EPIC allows you to constrain the minimum amount of disk storage a node

running a specific Big Data application can use. This controls the amount of "spill space" or temporary storage available to the services running within a given node. If the Big Data application being configured includes HDFS services, then the persistent storage capacity assigned to the container defines the capacity for HDFS within the cluster.

- **Enable and configure per-node role services:** Different sets of Big Data services can be enabled/disabled per node role. BlueData EPIC assigns a node role to each node in a Big Data cluster. This flexibility provides the ability to customize which specific service(s) run on each node in a cluster.
- **Surface access to services and interfaces:** Each service in a Big Data cluster is accessed via an IP address and port pair. BlueData EPIC allows the enterprise to control which Big Data services are visible to the user and how they can be accessed.
- **Secure Sockets Layer:** Secure Socket Layer (SSL) is a security technology that establishes an encrypted link between a web server and a browser using trusted (either signed of self-signed) certificates. BlueData EPIC can automatically configure SSL connections to Big Data services accessed externally by data scientists (e.g. Spark), as well as between internal cluster services (e.g. Hadoop ResourceMananger and NodeManager).

Application-specific configuration begins once infrastructure configuration has been completed for the Big Data application. This includes:

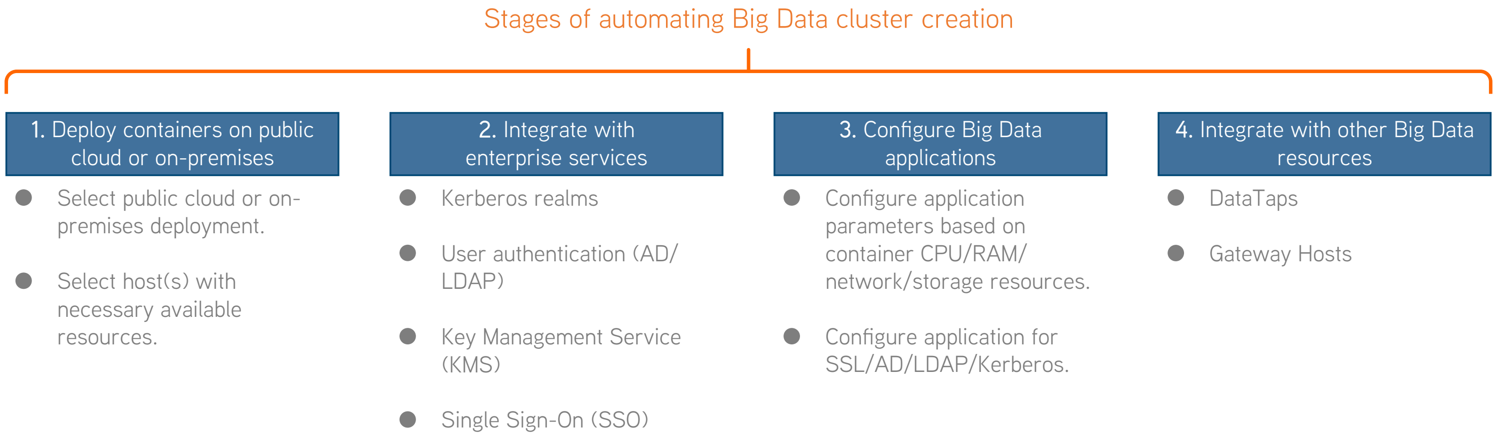
- **Application Kerberos configuration:** Integrating with an enterprise Kerberos system is mostly application-independent; however, some specific Big Data applications require special configuration. For example, each of the main Hadoop services can be configured to use a different Kerberos principal and be issued a secure keytab file that verifies principal authenticity. These Kerberos components enhance cluster security by preventing rogue software from executing "man-in-the-middle" attacks. BlueData EPIC supports automatic or manual principal and keytab creation.
- **Application AD/LDAP configuration:** BlueData EPIC can automate an application-specific LDAP integration or customization for an individual user or group of users.
- **Inserting manual steps into the cluster management process:** BlueData EPIC supports enterprises at any stage in their Big Data journey. Some Big Data cluster creation/management steps may require special integration with existing enterprise processes. BlueData EPIC supports running Big Data clusters in Isolated Mode that leaves the final configuration of a partially-configured cluster to an administrator, as described in "BlueData EPIC Tools for Automation" on page 7.

## Pipeline Infrastructure for Multiple Big Data Applications

The term *pipeline infrastructure for multiple Big Data applications* refers to any software or network configuration needed to allow operating a series of co-operating or interconnected applications in a Big Data pipeline, such as a Kafka service ingesting data into a Hadoop cluster. BlueData EPIC automates this deployment and management, as follows:

- **Integration with existing Big Data clusters and services:** Big Data clusters created on the BlueData EPIC platform can be configured to access data or services from clusters running on bare-metal, such as when a BlueData EPIC DataTap provides access to an existing HDFS data lake.
- **Addition of Gateway nodes:** Nodes attached to a Big Data Cluster can run "gateway" services that allow administrators to limit the container(s) that end users (such as data scientists) can access to run Big Data applications. Gateway nodes are separate and distinct from the nodes that run the Big Data application services.
- **Analytical clusters:** There are a number of business intelligence, ETL (Extract, Transform, and Load), and data science applications that run in special clusters that we call *analytical clusters*. These work with other Big Data clusters (typically Hadoop and Spark) to process large data volumes. BlueData EPIC supports analytical clusters in the following ways:
    - BlueData EPIC can launch a standalone analytical cluster from an EPIC application image built solely for that analytical application. These clusters can connect to one or more Hadoop/Spark clusters running in the same EPIC tenant.
    - BlueData EPIC can launch an integrated analytical cluster from an EPIC application image that contains the configuration information for both a Hadoop/Spark cluster and the analytical application cluster. These clusters need not connect to a different Hadoop/Spark cluster, because the Hadoop/Spark applications reside in the same cluster as the analytical application.
    - An analytical cluster can run outside the context of any BlueData EPIC environment and still access Hadoop/Spark clusters running within an EPIC environment.

Figure 3 summarizes the types and stages of automation at different levels in the Big Data cluster and application infrastructure stack.



Stages of automating Big Data cluster creation

1. Deploy containers on public cloud or on-premises

- Select public cloud or on-premises deployment.
- Select host(s) with necessary available resources.

2. Integrate with enterprise services

- Kerberos realms
- User authentication (AD/ LDAP)
- Key Management Service (KMS)
- Single Sign-On (SSO)

3. Configure Big Data applications

- Configure application parameters based on container CPU/RAM/ network/storage resources.
- Configure application for SSL/AD/LDAP/Kerberos.

4. Integrate with other Big Data resources

- DataTaps
- Gateway Hosts

*Figure 3: Stages of automating the Big Data cluster creation process*

# 4. BlueData EPIC Tools for Automation

BlueData EPIC includes a suite of tools that simplify adding the automation processes described in "Types of Big Data Infrastructure Automation" on page 3, including:

- **Docker files:** Docker files are the first component of the BlueData EPIC automation. They are a mechanism for defining which software bundles are required for the correct operation of a Big Data environment. A complete, well-defined Docker file makes it unnecessary to install any additional software into a cluster to make it operational.
- **ActionScripts:** ActionScripts are shell scripts that can run on one or more node(s) in a Big Data Cluster. They allow administrators to quickly run operations that need to be executed on multiple nodes in a cluster, such as installing software packages or updating configuration files. ActionScripts can be executed as "bootstrap" scripts at cluster creation time, or any time after a cluster has been created.

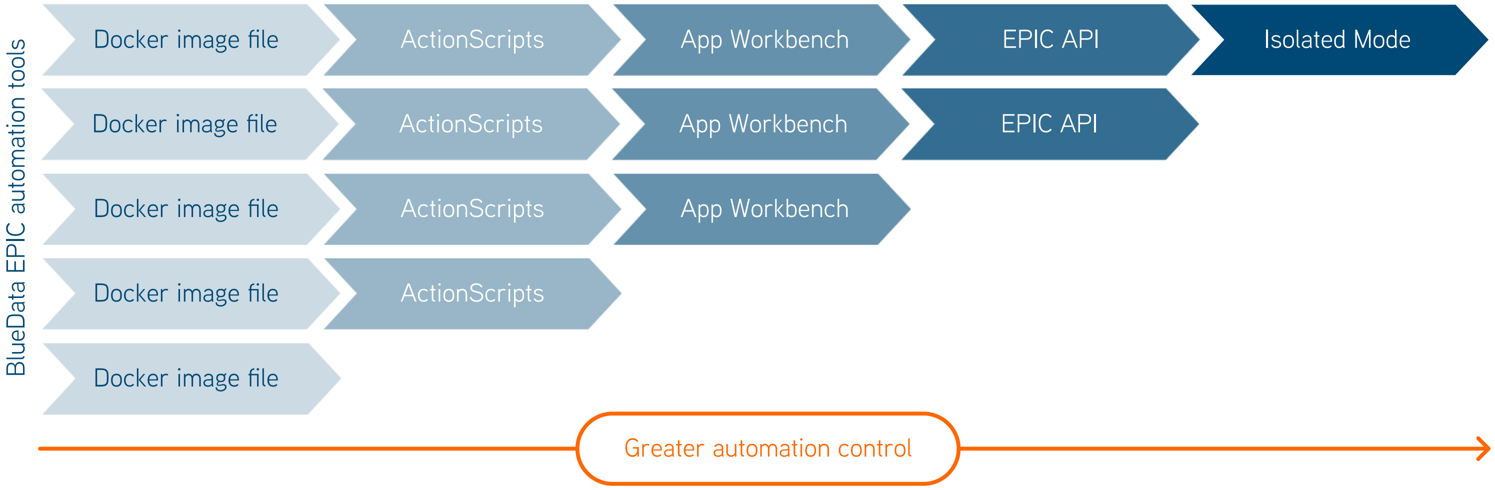


*Figure 4: BlueData EPIC tools for various levels of automation*

- **App Workbench:** The BlueData App Workbench is a set of tools that allow users to design, implement, and deploy customized Big Data applications and deployment processes to fit their businesses needs in a BlueData EPIC environment. App Workbench helps application authors insert configuration and automation scripts into the cluster creation process.
- **RESTful API for DevOps and automation:** The BlueData EPIC API consists of commands for creating, managing, and destroying Big Data clusters. Authors can use the EPIC API to integrate cluster management into various enterprise workflow management systems, such as ServiceNow. App Workbench authors can also use this API to add calls to their automation scripts that query various EPIC configuration parameters. The data returned by EPIC can be used to fine-tune application configuration and execution. This API framework consists of three primary components:
  - **BlueData agent:** EPIC installs this agent in all Big Data nodes in clusters instantiated in the BlueData EPIC environment. The agent communicates with the EPIC management services and controls cluster operation. Agent configuration occurs on cluster startup and is completely transparent to both the application and related configuration scripts.
  - **Python API:** The DFaXNKD API is a Python library that surfaces access to the BlueData EPIC API functionality in a common shell script language interface.
  - **Application configuration bundle:** This set of configuration files and scripts written by a BlueData EPIC application image developer is used in conjunction with the App Workbench to author an image for the BlueData EPIC App Store. Data scientists and data analysis teams can easily spin up multi-node clusters once these images have been added to the App Store and then installed in the EPIC platform.

- **Isolated Mode:** Isolated Mode is an operational mode that limits cluster access. Running a cluster in Isolated Mode allows authorized users to access the cluster for maintenance or updates using either automated scripts or manual commands, while preventing other users from accessing the services running in that cluster.

  Clusters can be placed into Isolated Mode at any time. For example, a new cluster can be created in Isolated Mode. This allows application-specific configuration (e.g. Kerberos protection) before the cluster becomes available for tenant-wide access. This scenario avoids any security issues that may arise from running an unprotected cluster. You can also configure a bootstrap ActionScript to run automatically as soon as a new cluster enters Isolated Mode.

# 5. BlueData EPIC Automation Scenarios

Every organization is at a different stage of their Big Data journey, and is thus ready for differing levels of automation. The scenarios outlined in this section illustrate how to use the BlueData EPIC tools described in this white paper to enable the automation of Big Data infrastructure and applications, starting with the simplest scenario and building to more advanced automation schemes.

## Crawl

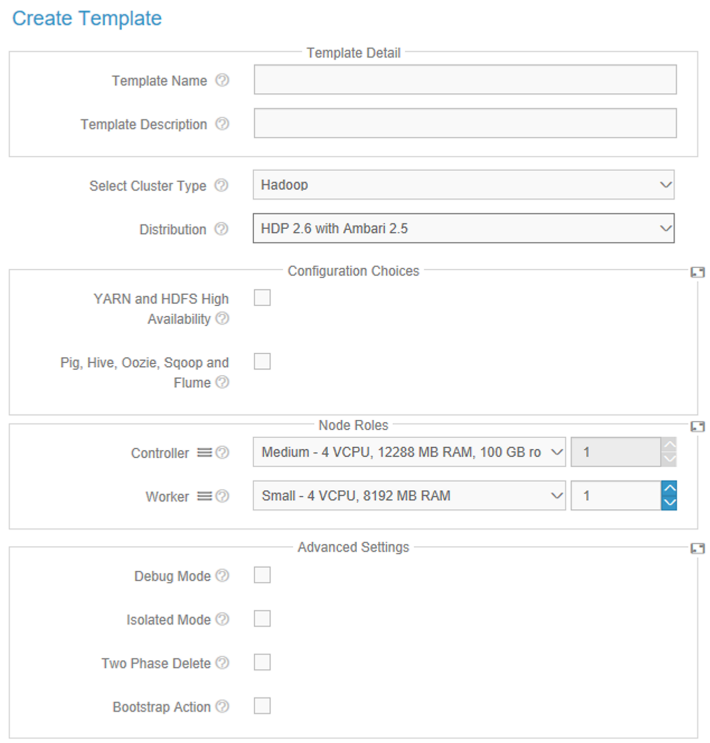
The BlueData EPIC App Store is a collection of pre-configured Docker application images for common Big Data application images. The App Store provides sample Docker images and starting configurations for particular versions of popular distributions, frameworks, and applications (e.g. Cloudera CDH 5.12.1, Hortonworks HDP 2.6, Spark 2.1.1).

Each of these images uses the tools described in this white paper to automatically bring up and configure Big Data clusters.

The simplest automation scenario within BlueData EPIC entails running the Big Data application images that are included with BlueData EPIC and accessible via the App Store, with no configuration or other modifications required. You simply employ the cluster "template" functionality within BlueData EPIC to specify:

- CPU, memory, and storage allocations for each node role in a cluster.
- Number of nodes of each role type in the cluster.
- Location of the data storage systems available to the cluster.
- Whether the cluster should be configured in HA (high availability) mode.
- Specific Big Data service(s) to enable.

You can save specific cluster configurations as templates that users can invoke to create new clusters via a single mouse click, as shown in Figure 5 (right).



Create Template

Template Detail

Template Name
Template Description

Select Cluster Type: Hadoop
Distribution: HDP 2.6 with Ambari 2.5

Configuration Choices

YARN and HDFS High Availability □
Pig, Hive, Oozie, Sqoop and Flume □

Node Roles

Controller: Medium - 4 VCPU, 12288 MB RAM, 100 GB ro — 1
Worker: Small - 4 VCPU, 8192 MB RAM — 1

Advanced Settings

Debug Mode □
Isolated Mode □
Two Phase Delete □
Bootstrap Action □

*Figure 5: Create Template screen in BlueData EPIC*

## Walk

The next level of automation starts by modifying an App Store application image. This allows you to add to and update your App Store with the latest versions of those Big Data applications, or make other modifications (e.g. adding a security patch to the base image).

You can use the App Workbench to "unpack" these application image bundles. Once unpacked, the image developer can view the existing scripts and customization processes. They can update or customize the scripts and then generate a new application image reflecting these changes. The image will appear as a new tile in the BlueData EPIC App Store.

Figures 6 through 8 on the following page illustrate how you can:

1. Start with an existing application image in the App Store, such as Cloudera CDH 5.10.1.
2. Unpack the contents of the Docker file.
3. Modify the application image, such as for a newer version of CDH.

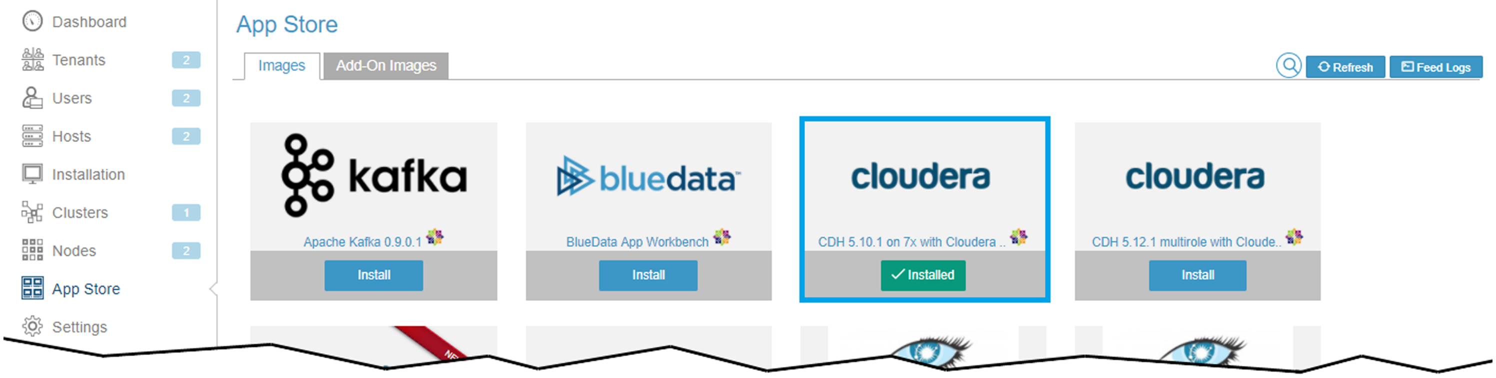
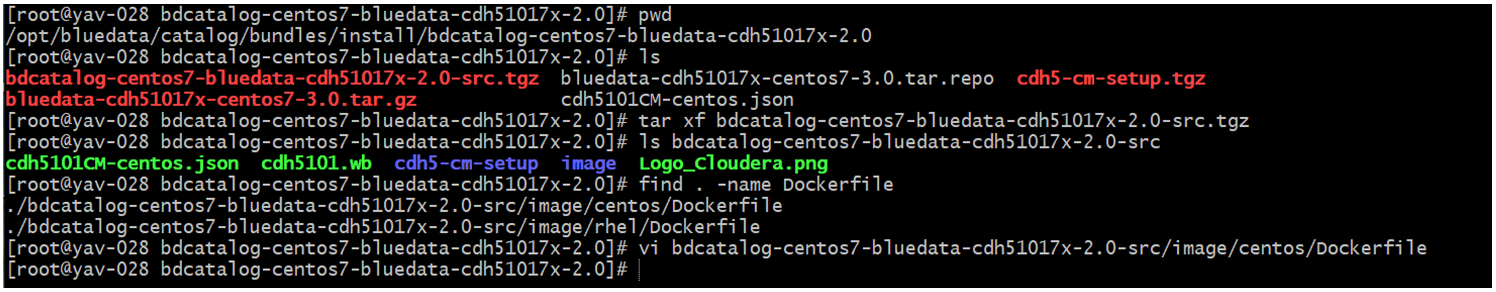
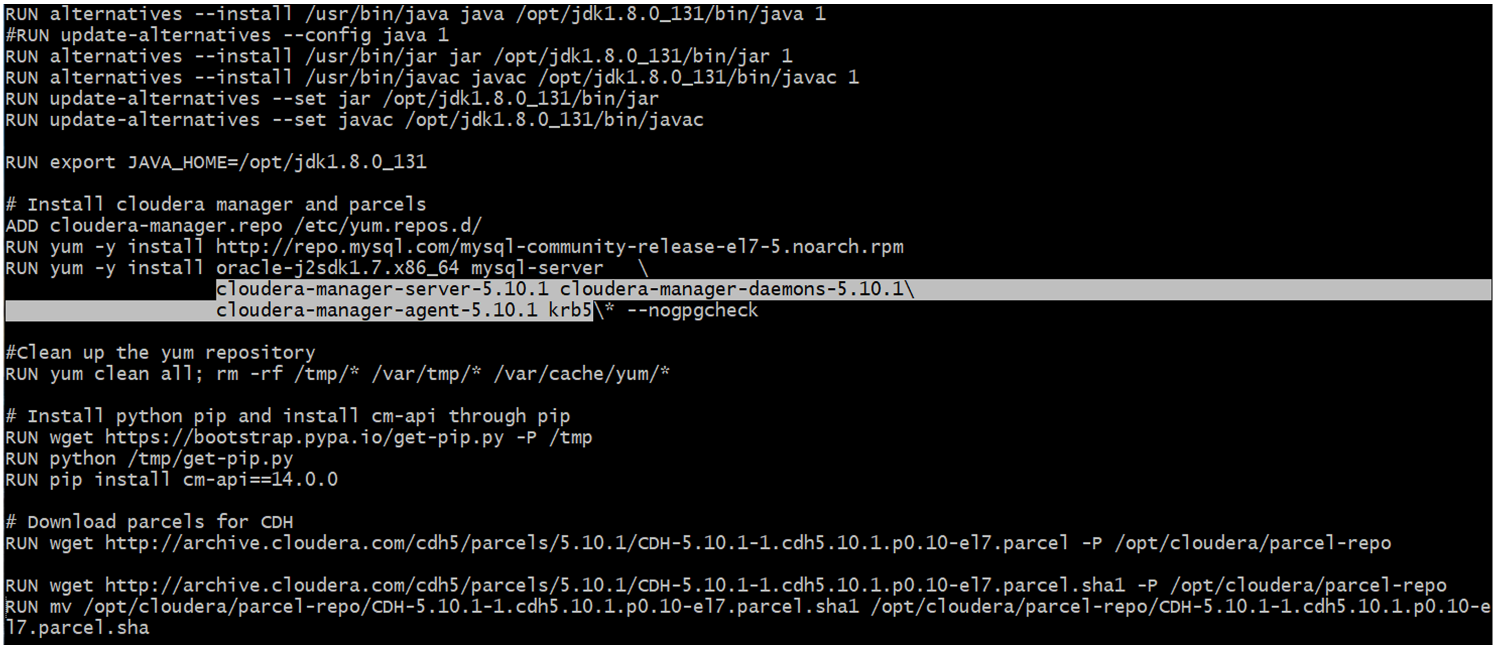


*Figure 6: CDH 5.10.1 application image in the BlueData EPIC App Store*



```
[root@yav-028 bdcatalog-centos7-bluedata-cdh51017x-2.0]# pwd
/opt/bluedata/catalog/bundles/install/bdcatalog-centos7-bluedata-cdh51017x-2.0
[root@yav-028 bdcatalog-centos7-bluedata-cdh51017x-2.0]# ls
bdcatalog-centos7-bluedata-cdh51017x-2.0-src.tgz  bluedata-cdh51017x-centos7-3.0.tar.repo  cdh5-cm-setup.tgz
bluedata-cdh51017x-centos7-3.0.tar.gz             cdh5101CM-centos.json
[root@yav-028 bdcatalog-centos7-bluedata-cdh51017x-2.0]# tar xf bdcatalog-centos7-bluedata-cdh51017x-2.0-src.tgz
[root@yav-028 bdcatalog-centos7-bluedata-cdh51017x-2.0]# ls bdcatalog-centos7-bluedata-cdh51017x-2.0-src
cdh5101CM-centos.json  cdh5101.wb  cdh5-cm-setup  image  Logo_Cloudera.png
[root@yav-028 bdcatalog-centos7-bluedata-cdh51017x-2.0]# find . -name Dockerfile
./bdcatalog-centos7-bluedata-cdh51017x-2.0-src/image/centos/Dockerfile
./bdcatalog-centos7-bluedata-cdh51017x-2.0-src/image/rhel/Dockerfile
[root@yav-028 bdcatalog-centos7-bluedata-cdh51017x-2.0]# vi bdcatalog-centos7-bluedata-cdh51017x-2.0-src/image/centos/Dockerfile
[root@yav-028 bdcatalog-centos7-bluedata-cdh51017x-2.0]#
```

*Figure 7: Unpacked contents of the CDH 5.10.1 application image*



```
RUN alternatives --install /usr/bin/java java /opt/jdk1.8.0_131/bin/java 1
#RUN update-alternatives --config java 1
RUN alternatives --install /usr/bin/jar jar /opt/jdk1.8.0_131/bin/jar 1
RUN alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_131/bin/javac 1
RUN update-alternatives --set jar /opt/jdk1.8.0_131/bin/jar
RUN update-alternatives --set javac /opt/jdk1.8.0_131/bin/javac

RUN export JAVA_HOME=/opt/jdk1.8.0_131

# Install cloudera manager and parcels
ADD cloudera-manager.repo /etc/yum.repos.d/
RUN yum -y install http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
RUN yum -y install oracle-j2sdk1.7.x86_64 mysql-server    \
                   cloudera-manager-server-5.10.1 cloudera-manager-daemons-5.10.1\
                   cloudera-manager-agent-5.10.1 krb5\* --nogpgcheck

#Clean up the yum repository
RUN yum clean all; rm -rf /tmp/* /var/tmp/* /var/cache/yum/*

# Install python pip and install cm-api through pip
RUN wget https://bootstrap.pypa.io/get-pip.py -P /tmp
RUN python /tmp/get-pip.py
RUN pip install cm-api==14.0.0

# Download parcels for CDH
RUN wget http://archive.cloudera.com/cdh5/parcels/5.10.1/CDH-5.10.1-1.cdh5.10.1.p0.10-el7.parcel -P /opt/cloudera/parcel-repo

RUN wget http://archive.cloudera.com/cdh5/parcels/5.10.1/CDH-5.10.1-1.cdh5.10.1.p0.10-el7.parcel.sha1 -P /opt/cloudera/parcel-repo
RUN mv /opt/cloudera/parcel-repo/CDH-5.10.1-1.cdh5.10.1.p0.10-el7.parcel.sha1 /opt/cloudera/parcel-repo/CDH-5.10.1-1.cdh5.10.1.p0.10-e
l7.parcel.sha
```

*Figure 8: Example of modifying a Docker file to install different versions of the Cloudera-manager-agent code*

## Run

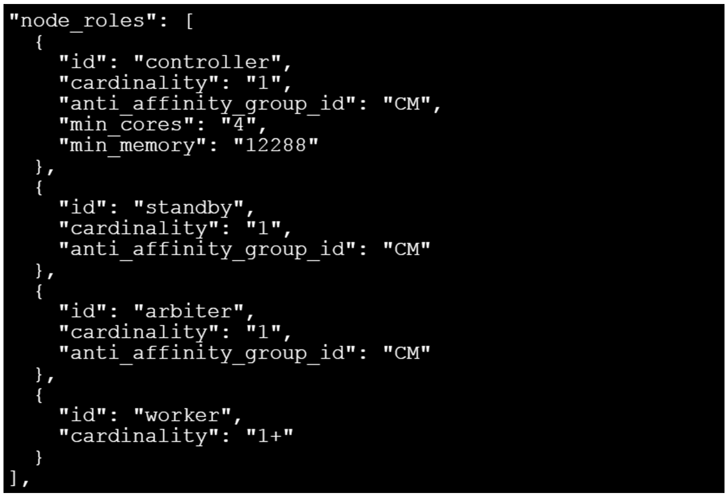
A more advanced scenario involves using BlueData EPIC to execute entirely new Big Data application images. BlueData EPIC runs Big Data applications without requiring any code changes whatsoever. Using the App Workbench and following the examples provided in the included App Store application images allows you to bundle any Big Data framework or application into a BlueData EPIC application image that includes both cluster configuration and automation instructions.

This is a common scenario as the Big Data journey evolves within an enterprise. Every organization probably has its own preferred Big Data applications, frameworks, and data science tools. Individual data science teams within a single organization may also have their own preferred toolsets. For example, one team may want to use Spark with RStudio, while another team may prefer Spark with a Jupyter notebook.

BlueData EPIC delivers unmatched flexibility to meet the needs of these different teams by enabling a "bring your own app" model of creating your own Docker applications images for your preferred Big Data applications and data processing frameworks. For example:

- Analysts with a Hadoop cluster dedicated for ETL may want an application like Talend pre-wired for immediate use as an Edge node.
- Data science teams may want to add machine learning applications like H2O or TensorFlow for their testing, development, prototyping, and experimentation.

Figure 9 illustrates how BlueData EPIC supports this scenario with multiple, flexible node roles. The metadata JSON file in each installed Big Data application can define the node roles that will be used by that application. For example, a Spark application image may include Jupyter, RStudio, Zeppelin, and/or Edge node roles in addition to the standard Controller and Worker roles.



```
"node_roles": [
  {
    "id": "controller",
    "cardinality": "1",
    "anti_affinity_group_id": "CM",
    "min_cores": "4",
    "min_memory": "12288"
  },
  {
    "id": "standby",
    "cardinality": "1",
    "anti_affinity_group_id": "CM"
  },
  {
    "id": "arbiter",
    "cardinality": "1",
    "anti_affinity_group_id": "CM"
  },
  {
    "id": "worker",
    "cardinality": "1+"
  }
],
```

*Figure 9: JSON code snippet displaying the simple syntax for declaring multiple roles of nodes in a Big Data cluster (below)*