FORTRA

VCM
2.25

Implementation and Installation Guide
May 2026

202605181119

Table of Contents

Before You Begin

Read and understand this section before you install VCM.

You must install VCM (on one system or on multiple systems) and a collector. The following are prerequisites and requirements for installing and deploying VCM in your environment:

Installation Requirements

Single System Deployment

Deployment configurations vary based on the following requirements for fault tolerance and configuration factors that affect the performance of the application:

- The number of users using the application concurrently
- The number of collected systems
- The collection rates for each metric group
- The number of instances collected for multiple instance metric groups
- The number of processes collected by system
- The number of disks and file systems collected by system

When there are no requirements for fault tolerance and the default collection settings are used, VCM can be installed on a single system. The following configuration supports up to 50 collectors and is intended as a software trial:

- 4 CPU cores
- 16 GB memory
- 125 GB of available disk space

> **NOTE:**
> Contact Fortra if your deployment includes more than 50 collectors.

The following are required for installing VCM on a single system:

- A dedicated system where you plan on installing VCM
- Administrator access to the system where you plan on installing VCM

To install and run VCM, your configuration must meet the following requirements:

Platform	Operating System Level
Red Hat Enterprise Linux on x64 systems	8 or 9

Multiple System Deployment

The deployment recommendations in this document are based on test deployments used to validate support for the collection of 5000 systems with default collection settings. It is assumed that each collected system averages less than 5 collected resources per metric group (CPUs, disks, file systems, and network interfaces). Data volumes can vary greatly when using non-default collection settings. Consult with Fortra when considering deploying non-default collection settings at scale.

The three system deployment described here provides support for the collection of 5000 systems with default collection settings. This deployment breakdown also forms the basis for larger deployments as your collection needs grow. For example, additional streaming and database systems might be required to support additional data collection or non-default data collection settings.

The following are required for installing VCM on multiple systems:

- All installation systems are running Red Hat Enterprise Linux (RHEL) 8 or 9.
- Root access to the installation systems

> **NOTE:**
> The installation steps in the subsequent subsections use the example system names system1, system2, and system3 to reference the systems used for the installation. Actual system DNS hostnames or IP addresses should be used for your deployment.

For more information on deploying VCM on multiple systems, see ***Deploying VCM on Multiple Systems* on page 15**.

Three System Deployment Example

The following is an example of a three system deployment. Fortra recommends deploying VCM in the following order:

System 3: Database and Data Management Components

- Installed components:
  - Apache Cassandra
  - vityl-cassandra
  - vityl-data-management
  - vityl-spark-controller
  - vityl-spark-worker
  - vityl-summarization
- 8 CPU cores
- 32 GB memory minimum (64 GB is recommended)
- Inbound Ports:
  - 4702
  - 4703
  - 9042
- Outbound Ports (select one of the following):
  - Allow all outgoing ports
  - Allow the following ports:
    - 2181
    - 3703
    - 9092
- Filesystem information:

Location	Size	Owner	Permissions	Description Description
/opt/vityl	8 GB	root	Owner Read/Write/Execute, Group Read/Execute, Other Read/Execute (755)	VCM application files
/var/log/cassandra	1 GB	cassandra	Owner Read/Write/Execute, Group Read/Execute, Other Read/Execute (755)	Apache Cassandra logging

Location	Size	Owner	Permissions	Description Description
/var/opt/vityl/spark	2 GB	vityl-cm	Owner Read/Write/Execute, Group Read/Execute, Other Read/Execute (755)	Local storage for Apache Spark
/var/lib/cassandra/data	750 GB	cassandra	Owner Read/Write/Execute, Group Read/Execute, Other Read/Execute (755)	Apache Cassandra data. This disk should support a minimum of 5000 IOPS and 15 MB/s write throughput. This directory must be empty for Apache Cassandra to function correctly.

Location	Size	Owner	Permissions	Description Description
/var/lib/cassandra/commitlog	15 GB	cassandra	Owner Read/Write/Execute, Group Read/Execute, Other Read/Execute (755)	Apache Cassandra commit log. This file system should be on a separate disk from the Apache Cassandra data for best performance. This directory must be empty for Apache Cassandra to function correctly.

System 2: Data Streaming Components

- Installed components:
  - vityl-kafka
  - vityl-storm
  - vityl-storm-ui
  - vityl-storm-logviewer
  - vityl-storm-nimbus
  - vityl-storm-supervisor
  - vityl-storm-topologies
- 8 CPU cores
- 32 GB memory

- Inbound Ports:
  - 2181
  - 4700
  - 4701
  - 9092
- Outbound Ports (select one of the following):
  - Allow all outgoing ports
  - Allow the following ports:
    - 443
    - 2181
    - 3703
    - 9042
    - 9092
- Filesystem Information

Location	Size	Owner	Permissions	Description
/opt/vityl	8 GB	root	Owner Read/Write/Execute, Group Read/Execute, Other Read/Execute (755)	VCM application files
/var/lib/kafka	200 GB	vityl-cm	Owner Read/Write/Execute, Group Read/Execute (750)	Apache Kafka data. This disk should support a minimum of 5000 IOPS and 15 MB/s write throughput. This directory must be empty for Apache Kafka to function correctly.
/var/lib/zookeeper	2 GB	vityl-cm	Owner Read/Write/Execute, Group Read/Execute (750)	Apache Zookeeper data

Location	Size	Owner	Permissions	Description
/var/log/kafka	1 GB	vityl-cm	Owner Read/Write/Execute, Group Read/Execute, Other Read/Execute (755)	Apache Zookeeper and Apache Kafka logging
/var/opt/vityl/storm-local	3 GB	vityl-cm	Owner Read/Write/Execute, Group Read/Execute, Other Read/Execute (755)	Local storage for Apache Storm

System 1: Front-End Components and Capacity Plans

- Installed components:
  - PostgreSQL
  - vityl-console
  - vityl-console-ui
  - vityl-collector-api
  - vityl-metrics-api
  - vityl-capacity-plans
  - vityl-data-analytics
- 8 CPU cores
- 32 GB memory
- Inbound Ports:
  - 443
  - 3703
  - 3708
  - 3709
- Outbound Ports (select one of the following):
  - Allow all outgoing ports
  - Allow the following ports:
    - 2181
    - 9042
    - 9092
- Filesystem Information

Location	Size	Owner	Permissions	Description
/opt/vityl	8 GB	root	Owner Read/Write/Execute, Group Read/Execute, Other Read/Execute (755)	VCM application files
/var/lib/pgsql	20 GB	postgres	Owner Read/Write/Execute (700)	PostgreSQL data. This disk should support a minimum of 5000 IOPS and 15 MB/s write throughput.

Collector

The following are required for installing the VCM Collector component:

- Root access to the system where you plan on installing the Collector component.
- External internet access on the system where you plan on installing the Collector component or a local repository to hold the software packages (Red Hat Enterprise Linux and CentOS systems only)

To install and run the Collector component, your configuration must meet the following requirements:

Platform	Operating System Level
Red Hat Enterprise Linux on x64 systems	7, 8, or 9
Red Hat Enterprise Linux on IBM POWER systems	8
SuSE Linux Enterprise Server on IBM POWER systems	15
Ubuntu Server LTS Linux on x64 systems	20.04, 22.04, 24.04
Microsoft Windows Server on x64 systems	2012 R2, 2016, 2019, 2022, or 2025

Container and Kubernetes Pod Collection Requirements

When properly configured, the collector can collect performance data for containers and Kubernetes pods in the following environments:

- Stand alone and clustered Docker hosts
- Kubernetes hosts using the Docker container runtime
- Kubernetes hosts using the CRI-O container runtime

The collector can be installed as a service or as a sidecar container on systems running containers or Kubernetes pods. For information on installing collectors on Linux systems, see ***Installing Collectors on Linux Systems* on page 22**. For information on running the collectors on Linux systems as containers, see ***Running Collectors on Linux Systems as a Container* on page 25**.

Metadata for Kubernetes pods is collected using the kubectl command line interface (CLI). Metadata for containers can be collected from a variety of command line interfaces. Metadata collection for containers is attempted in the following order. Values from the first successful collection method are used.

1. kubectl
2. docker
3. podman
4. crictl

Configure permission for the vityl-cm user to call the appropriate CLI for your environment. If necessary for your environment, you can also configure the collector to run as the root user to enable this collection (see ***Running Collectors on Linux Systems as the Root User* on page 23**).

Installing VCM on a Single System

VCM can be installed on a single system.

To install VCM on a single system:

1. Download the **Vityl Bootstrap** package from the **Fortra Support Portal** at https://support.fortra.com.
2. Move the **vityl-bootstrap** RPM package to the target system.
3. Open a terminal window on the target system.
4. Install the **vityl-bootstrap** package using the following command:

   ```
   sudo yum install ./vityl-bootstrap-x.x.x-1.el8.noarch.rpm
   ```

   This places all VCM packages in the /opt/vityl/boot directory and registers it as a yum repository for use in the local system.
5. Install the rest of the **VCM** packages using the following command:

   ```
   sudo yum install vityl-performance-monitor
   ```
6. Run the following command:

   ```
   /opt/vityl/util/stormHostnameChange.sh <system DNS name or IP address>
   ```

   where system DNS name or IP address is of the system where vityl-storm* components are installed.
7. Run the following command:

   ```
   /opt/vityl/util/uiListenerChange.sh <system DNS name or IP address>
   ```

   where system DNS name or IP address is of the system where nginx is installed.

Deploying VCM on Multiple Systems

Installation should be performed in the following order:

1. *Database and Data Management Components*
2. *Data Streaming Components*
3. *Front-End Components and Capacity Plans*

NOTE:
Required file systems are described for each VCM system. If additional filesystems are provisioned to meet these requirements, Fortra recommends mounting these filesystems at the root of the filesystem (for example, /data/cassandra/data, /data/cassandra/commitlog, /data/kafka/data, and so on). After installing VCM, move files from the default locations to the newly provisioned filesystems and create symbolic links between the default locations and the newly provisioned locations. Installing the software before making filesystem changes ensures that directory ownership and permissions are configured correctly.

Additional Installation Ports

Collector System

> **NOTE:** The Collector system only runs the Collector component.

- Inbound Ports:
  - None
- Outbound Ports (select one of the following):
  - Allow all outgoing ports
  - Allow the following ports:
    - 2181
    - 9092

End User System

- Inbound Ports:
  - None
- Outbound Ports (select one of the following):
  - Allow all outgoing ports
  - Allow the following ports:
    - 443
    - 3709
    - 4700
    - 4701
    - 4702
    - 4703

Deploying Database and Data Management Components (system3)

To deploy the database and data management components:

1. Download the **Vityl Bootstrap** package from the **Fortra Support Portal** at https://support.fortra.com/.
2. Move the **vityl-bootstrap** RPM package to the target system.
3. Open a terminal window on the target system.
4. Install the **vityl-bootstrap** package using the following command:

   ```
   sudo yum install ./vityl-bootstrap-2.X.0-1.el8.noarch.rpm
   ```

   This places all VCM packages in the /opt/vityl/boot directory and registers it as a yum repository for use in the local system.
5. Install the **vityl-utils** package using the following command:

   ```
   sudo yum install vityl-utils
   ```

   This places the deployment scripts in the /opt/vityl/util directory.
6. Change to the **/opt/vityl/util** directory.
7. Edit the **vityl-systems.env.sh** file and specify the DNS name or IP address for SYSTEM1, SYSTEM2, and SYSTEM3.
8. Run the **deploySystem3.sh** deployment script.

   Database and data management components are installed and automatically configured to communicate with SYSTEM1 and SYSTEM2. If firewalld is installed, required external ports are opened as part of the public zone.

Deploying Data Streaming Components (system2)

To deploy the data streaming components:

1. Download the **Vityl Bootstrap** package from the **Fortra Support Portal** at https://support.fortra.com/.
2. Move the **vityl-bootstrap** RPM package to the target system.
3. Open a terminal window on the target system.
4. Install the **vityl-bootstrap** package using the following command:

   ```
   sudo yum install ./vityl-bootstrap-2.X.0-1.el8.noarch.rpm
   ```

   This places all VCM packages in the /opt/vityl/boot directory and registers it as a yum repository for use in the local system.
5. Install the **vityl-utils** package using the following command:

   ```
   sudo yum install vityl-utils
   ```

   This places Vityl deployment scripts in the /opt/vityl/util directory.
6. Change to the **/opt/vityl/util** directory.
7. Edit the **vityl-systems.env.sh** file and specify the DNS name or IP address for SYSTEM1, SYSTEM2, and SYSTEM3.
8. Run the **deploySystem2.sh** deployment script.

   Database and data management components are installed and automatically configured to communicate with SYSTEM1 and SYSTEM3. If firewalld is installed, required external ports are opened as part of the public zone.

Deploying Front-End Components and Capacity Plans (system1)

To deploy the front-end components and Capacity Plans:

1. Download the **Vityl Bootstrap** package from the **Fortra Support Portal** at https://support.fortra.com/.

   Move the **vityl-bootstrap** RPM package to the target system.
2. Open a terminal window on the target system.
3. Install the **vityl-bootstrap** package using the following command:

   ```
   sudo yum install ./vityl-bootstrap-2.X.0-1.el8.noarch.rpm
   ```

   This places all VCM packages in the /opt/vityl/boot directory and registers it as a yum repository for use in the local system.
4. Install the **vityl-utils** package using the following command:

   ```
   sudo yum install vityl-utils
   ```

   This places the deployment scripts in the /opt/vityl/util directory.
5. Change to the **/opt/vityl/util** directory.
6. Edit the file vityl-systems.env.sh and set the values for SYSTEM1, SYSTEM2, and SYSTEM3 in this file. The DNS name or IP address of each system should be used for these values.
7. Run the **deploySystem1.sh** deployment script.

   Database and data management components are installed and automatically configured to communicate with SYSTEM2 and SYSTEM3. If firewalld is installed, required external ports are opened as part of the public zone.

Installing Collectors

You can use collectors to collect, stream, and manage performance metrics and heartbeat data in real-time intervals. Raw data from servers is collected, streamed, and converted into a format that is stored in a database and analyzed using Performance Monitor.

You must install a collector on each system you want to monitor, which can include the system where VCM is installed.

Before you install a collector, you must do the following:

- Update the Kafka Connection URI to point to where Apache Kafka is installed and running.

  The Kafka Connection URI can be a list of Kafka brokers. For example, kafkaBroker1:9092,kafkaBroker2:9092. This might be required if you have collectors that can only connect to a specific network address. Collectors might be in error if a Kafka broker is unreachable. As long as the collector can connect to one Kafka broker, the data can be streamed.

  To update the Kafka Connection URI, click the Configuration menu item from the Collectors user interface and click the Kafka URI button. Type the new URI and click Update.

  After a collector is initiated, data collection cannot start until you assign a configuration to the collector.
- Review and grant user privileges as needed.

  Some metric collection requires elevated user privileges. By default, the collector is installed to run with lower privileges.

  You can adjust the user privileges when and where appropriate (see ***Running Collectors on Linux Systems as the Root User* on page 23**) to fit your data collection needs.

  The following Disk IO metrics in the Process data group require root privileges on Linux systems:
  - Data
  - Reads
  - Writes

The following Process data metrics require local administrative privileges on Microsoft Windows systems:

- Full Command
- Login

> **NOTE:**
> Workload generation that depends on these fields cannot function correctly when these fields are not collected.

Collector Requirements

The following are required for installing the Collector component of VCM:

- Root access to the system where you plan on installing the Collector component.
- External internet access on the system where you plan on installing the Collector component or a local repository to hold the software packages (Red Hat Enterprise Linux systems only)

To install and run the Collector component, your configuration must meet the following requirements:

Platform	Operating System Level
Red Hat Enterprise Linux on x64 systems	7, 8, or 9
Red Hat Enterprise Linux on IBM POWER systems	8
SuSE Linux Enterprise Server on IBM POWER systems	15
Ubuntu Server LTS Linux on x64 systems	20.04, 22.04, or 24.04
Microsoft Windows Server on x64 systems	2012 R2, 2016, 2019, 2022, or 2025

Data Collection

The data collection process can be described in the following tiers:

- Data collection
- Data streaming
- Data processing and storage

VCM data collectors can be deployed to Linux and Microsoft Windows systems. These collectors read configuration data from Apache ZooKeeper, perform the requested data collection, and stream collected data to Apache Kafka.

Apache Kafka retains up to 12 hours of streamed data by default, making this data available to multiple consumers. You can run multiple Apache Kafka nodes to support large message volumes and to provide fault tolerance.

Apache Storm processes consume collected data in real time and process this data for storage to Apache Cassandra. You can use multiple Apache Storm and Apache Cassandra nodes to support large message volumes and to provide fault tolerant storage.

Installing Collectors on Linux Systems

> **NOTE:** Administrator access is required to install a collector.

To install a Collector on Linux systems:

1. Open a terminal window on the system where you want to install the collector.
2. Copy the **Collector** package from the system where you installed Vityl Bootstrap. There are several ways to accomplish this, including but not limited to ftp, scp, curl, rpm, and wget.

   For example, for scp, you can use the following:

   ```
   scp SERVERNAME:/opt/vityl/boot/vityl-collector\* /tmp/
   ```
3. Install the collector using one of the following commands, depending on your system:

   For Ubuntu systems:

   ```
   sudo dpkg --install /tmp/vityl-collector_x.x.x-x-ubuntuxx.xx_
   amd64.deb
   ```

   For Red Hat Enterprise Linux and SuSe Enterprise Server systems:

   ```
   sudo rpm --install /tmp/vityl-collector-x.x.x-x.elx.x.rpm
   ```

4. From **/etc/vityl/collector**, edit the **zookeeper.properties** file.

   The zookeeper.properties file is the properties file used to connect the collector to Apache ZooKeeper.

   By default, the zookeeper.properties file contains the following information:

   ```
   ZooKeeperConnectionString=127.0.0.1:2181
   ```

   where:

   ZooKeeperConnectionString is the network connection and port used to connect to Apache ZooKeeper. By default, 127.0.0.1:2181. Apache ZooKeeper is used to configure the metrics and heartbeat data for each collector and to provide connection information to the Apache Kafka stream.
5. Enable and start the collector using the following commands:

   ```
   sudo systemctl enable vityl-collector
   sudo systemctl start vityl-collector
   ```

   > **NOTE:**
   > To see the system after you install and start a new collector, refresh your datasources in VCM or wait until the top of the hour.

6. Verify the collector is running using the following command:

   ```
   sudo systemctl status vityl-collector
   ```
7. Assign a configuration to the collector by doing the following:
   a. From the **Collectors** user interface, click the **Collectors** menu item.
   b. Locate the collector and assign a configuration.

Running Collectors on Linux Systems as the Root User

> **NOTE:**
> Administrator access is required to change collector service settings.

To run a collector as the root user on systemd based initialization systems:

1. Edit the collector service definition using the following command:

   ```
   sudo systemctl edit vityl-collector
   ```
2. Add the following text to this service override file:

   ```
   [Service]
   User=root
   Group=root
   ```
3. Save your changes and exit.

4. Restart the collector using the following command:

   ```
   sudo systemctl restart vityl-collector
   ```

Running Collectors on Linux Systems as a Container

You can run the collector as a container on Linux x64 Docker or Kubernetes hosts. Monitoring the host system requires the host's root filesystem to be bind-mounted in /host on the container. The container also needs to use the host's network stack and PID namespace to collect all network and process data. The VCM_ZOOKEEPER_CONNECT environment variable must be supplied to the container to provide the connection string for Apache ZooKeeper.

To run the collector on Linux systems as a container:

1. Open a terminal window on the system where you want to install the collector.
2. Copy the **vityl-collector-2.*x*.*x*-image.tar.gz** file to a Docker host.
3. Load the image using the following command:

   `docker load < vityl-collector-2.x.x-image.tar.gz`
4. For Kubernetes deployment (and optionally for Docker deployment), re-tag and push the image to your registry using the following commands:

   `docker image tag vityl-collector:2.x.x registry.example.com/vityl-collector:2.x.x`

   `docker push registry.example.com/vityl-collector:2.x.x`
5. Start the container.

Example Docker Command

```
docker container run -d --tty --restart=unless-stopped --
network="host"
--pid="host" --env "VCM_ZOOKEEPER_CONNECT=your-vcm-zookeeper:2181"
--env "VCM_GUID_TYPE=host"
--volume /:/host:ro --volume
/var/run/docker.sock:/host/var/run/docker.sock/:ro
--hostname $HOSTNAME
--name vityl-collector vityl-collector:2.x.x
```

Example YAML Definition for Kubernetes Deployments

```
apiVersion: v1
kind: Pod
metadata:
  name: vityl-collector
spec:
  hostPID: true
  hostNetwork: true
  containers:
  - name: vityl-collector
```

```
    image: registry.example.com/vityl-collector:2.x.x
    env:
    - name: VCM_ZOOKEEPER_CONNECT
      value: "your-vcm-zookeeper:2181"
    - name: VCM_GUID_TYPE
      value: "host"
    volumeMounts:
    - mountPath: /host
      name: host-root
      readOnly: true
    command: ["/usr/bin/vityl-collector"]
    args: ["-e", "-c", "/etc/vityl/collector"]
  volumes:
  - name: host-root
    hostPath:
      path: /
      type: Directory
```

Installing Collectors on Microsoft Windows Systems

> **NOTE:** Administrator access is required to install a collector.

After you install a Collector on Microsoft Windows systems, you can change the collector configuration and reset the recovery options on failure.

To install a collector on a Microsoft Windows system:

1. Copy the **Collector** package from the system where you installed Vityl Bootstrap. The installer is located in the following location:

   `/opt/vityl/boot/vityl-collector-x.x.x.exe`
2. Run the installer.
3. Edit the **zookeeper.properties** file located in the **C:\ProgramData\Vityl\Collector** directory and set the **ZooKeeperConnectionString** according to your deployment.

   The zookeeper.properties file is used to connect the collector to Apache ZooKeeper.

   By default, the zookeeper.properties file contains the following information:

   ```
   ZooKeeperConnectionString=127.0.0.1:2181
   ```

   where:

   ZooKeeperConnectionString is the network connection and port used to connect to Apache ZooKeeper. Apache ZooKeeper is used to configure the metrics and heartbeat data for each collector and to provide connection information to the Apache Kafka stream.
4. From **Control Panel** > **Services**, start the **Collector** service.

5. From **Control Panel** > **Services**, set the Collector service startup type to **Automatic**.

   NOTE:
   To see the system after you install and start a new collector, refresh your datasources in VCM or wait until the top of the hour.

6. Assign a configuration to the collector by doing the following:
   a. From the **Collectors** user interface, click the **Collectors** menu item.
   b. Locate the collector and assign a configuration.

To install a collector on a Microsoft Windows system using a command prompt:

NOTE:
You must have Administrator privileges to install a collector from a command prompt.

1. Copy the **Collector** package from the system where you installed **Vityl Bootstrap**. The installer is located in the following location:

   `/opt/vityl/boot/vityl-collector-x.x.x.exe`

2. Run the following command:

   `vityl-collector-x.x.x.exe /exenoui /qn /l*v install.log`

3. Edit the **zookeeper.properties** file located in the **C:\ProgramData\Vityl\Collector** directory and set the **ZooKeeperConnectionString** according to your deployment.

   The zookeeper.properties file is used to connect the collector to Apache ZooKeeper.

   By default, the zookeeper.properties file contains the following information:

   ```
   ZooKeeperConnectionString=127.0.0.1:2181
   ```

   where:

   ZooKeeperConnectionString is the network connection and port used to connect to Apache ZooKeeper. Apache ZooKeeper is used to configure the metrics and heartbeat data for each collector and to provide connection information to the Apache Kafka stream.

4. Start the service using the following command:

   `net start vitylcollector`

5. Change the vitylcollector service configuration from a manual to an automatic startup using the following command:

   `sc config vitylcollector start= auto`

   NOTE:
   Run the sc command even if the service is started and operating correctly.

6. Assign a configuration to the collector by doing the following:
    a. From the **Collectors** user interface, click the **Collectors** menu item.
    b. Locate the collector and assign a configuration.

Configuring Collectors When Deploying With Automation

The zookeeper.properties file (located in the /etc/vityl/collector/ directory for Linux systems and in the C:\ProgramData\vityl\collector\ directory for Microsoft Windows systems) stores connection information for Apache ZooKeeper. When started, a collector reads this file and uses this connection information to connect to Apache ZooKeeper to retrieve collector configuration information. The collector configuration defines what metrics the collector collects, which collection intervals to use, the location of Apache Kafka for streaming the data, and so on.

If a file named zookeeper.properties exists in the /etc/vityl/collector directory for Linux systems or in the C:\ProgramData\vityl\collector\ directory for Microsoft Windows systems, the collector uses this file and does not overwrite it during installation. When using automated deployment tools to install and configure vityl-collector, the zookeeper.properties file can be modified to include the Apache ZooKeeper connection information for your deployment. After installing your first collector, you can use the associated zookeeper.properties file as a template for creating the file to use in your automated collector deployments.

Docker Collection

Collectors can be configured to collect Docker container metrics when running on a Docker host. When installed, the Collector component runs under the vityl-cm system user account. This account needs read access to the Docker domain socket (which is created by default in the /var/run/docker.sock file). This is commonly accomplished by adding the user to the docker group.

For example,

```
usermod -aG docker vityl-cm
```

> **NOTE:**
> If the collector is running when the vityl-cm user is added to the Docker group, running the above command does not affect the running collector process. Restart the collector to make sure that Docker data collection can be performed.

Kubernetes Collection

Collectors can be configured to collect Kubernetes pod metrics when running on a Kubernetes node. When installed, the Collector component runs under the vityl-cm system user account. This account can read pod performance metrics from the cgroup file system but cannot interact with Kubernetes services to collect pod related meta data (such as pod names, pod namespaces, and the pod-to-containers relationship). When pod meta data is not available, pod performance metrics are reported by pod identifier only. To collect the additional meta data, Fortra recommends configuring the collector to run as the root user (see ***Running Collectors on Linux Systems as the Root User* on page 23**).

On Kubernetes clusters, collectors can be installed using a DaemonSet to make sure that each node in the cluster runs a copy of the collector. Additionally, you can configure your Kubernetes cluster to allow the collector to run the kubectl command line interface (CLI) to collect metadata on Kubernetes pods and containers. An example yaml definition for this configuration is provided in the /opt/vityl/util/examples/kubernetes/vityl-collector.yaml file, which is installed as part of the vityl-utils package. The vityl-utils package is installed as part of your VCM deployment.

Using Environment Variables to Alter Collector Behavior

You can alter the collector runtime behavior using environment variables to ensure that your deployment needs are met. Environment variable values take precedence when the values conflict with information defined by collector configuration files.

Environment Variable Name	Behavior
VCM_ ZOOKEEPER_ CONNECT	This variable overrides the zookeeper connection string. It can be used to override the value provided by the zookeeper.properties file.
VCM_GUID_ TYPE	This variable overrides the collector unique identifier behavior. Valid values are host, file, and preset. When this variable is not specified, it is set to file by default. The file setting causes the Collector to use the guid.properties file on the collected system filesystem. This file is generated automatically by the collector when it is not already present. The preset value allows you to provide your own GUID using the VCM_ GUID environment variable. The host value causes the collector to build a GUID from the system hostname. This ensures that a given hostname always generates the same GUID value. This is useful when deploying the collector in an environment with ephermal storage, such as inside a container on a Docker or Kubernetes host.

Environment Variable Name	Behavior
VCM_GUID	This variable defines a user provided GUID for use with the VCM_GUID_TYPE preset value. It must conform to GUID standards: it should follow the 8-4-4-4-12 hex character pattern (hyphens are optional) and include valid version and variant values.
VCM_HOST_GUID_NAMESPACE	This variable defines the default namespace used when the host is specified with the VCM_GUID_TYPE environment variable. By default, when this is not specified, the GUID Verson 5 standard defined DNS namespace is used when generating the host GUID. When this value is specified, it is used as the GUID namespace. This value can be used to ensure that system hostname collisions do not generate identical GUID values in your environment. For example, in your environment, system names use a DNS short name. You are deploying the collector to multiple Docker or Kubernetes clusters and you expect DNS short names on these systems to be duplicated across the clusters. This value can be used to specify a namespace for each cluster to ensure that GUID collisions do not occur with your collected data.
VCM_HOSTNAME	This variable overrides the hostname otherwise provided by the host operating system. When using this value, make sure that hostnames are not duplicated with different collector deployments.
VCM_FS_PREFIX	This variable overrides the default filesystem prefix used when collecting data from host files. This value is used to allow the collector to collect data from the host filesystem when running inside a container. The collector container image provided by Fortra uses the default location /host. This environment variable allows you change this location, which can be useful if you are building your own container for the collector. The collector expects a read-only mount of the root host filesystem at this location.
VCM_FS_PREFIX	This variable overrides the default filesystem prefix used when collecting data from host files. This value is used to allow the collector to collect data from the host filesystem when running inside a container. The collector container image provided by Fortra uses the default location /host. This environment variable allows you change this location, which can be useful if you are building your own container for the collector. The collector expects a read-only mount of the root host filesystem at this location.

Environment Variable Name	Behavior
VCM_FS_ PREFIX	This variable overrides the default filesystem prefix used when collecting data from host files. This value is used to allow the collector to collect data from the host filesystem when running inside a container. The collector container image provided by Fortra uses the default location /host. This environment variable allows you change this location, which can be useful if you are building your own container for the collector. The collector expects a read-only mount of the root host filesystem at this location.
VCM_POD_ PROPERTIES_ COLLECT	This variable overrides the default Kubernetes pod properties collection behavior. Valid values are host, all, and none. When a value is not specified, this variable is set to host by default. By default, pod properties are collected on every host in the cluster running the collector. Each node sends property data for the pods running on the node. Collecting pod properties on every node can create undesirable overhead in large environments. To avoid this overhead, you can use the all value to cause the collector to collect and send pod properties for all pods in the cluster from a single node. The none value disables pod properties collection.
VCM_POD_ PROPERTIES_ INTERVAL	This variable overrides the default Kubernetes pod properties collection interval. Valid values are: 1m, 5m, 15m, and 1h. These values correspond to the allowed metric collection intervals of 1 minute, 5 minutes, 15 minutes, and 1 hour. When unspecified, the default 1 minute collection interval is used.
VCM_ CONTAINER_ PROPERTIES_ INTERVAL	This variable overrides the default container properties collection interval. Valid values are: 1m, 5m, 15m, and 1h. These values correspond to the allowed metric collection intervals of 1 minute, 5 minutes, 15 minutes, and 1 hour. When unspecified, the default 1 minute collection interval is used.

After You Are Done

Read the following for additional information and your next steps.

Configuring VCM for Your Environment

Use the following steps to configure VCM for your environment:

1. Access VCM using the default credentials.

   For more information, see ***Accessing VCM* on page 35**.
2. Apply license key and configure collection settings.

   For more information, see ***Applying the License Key and Configuring Collection Settings* on page 35**.
3. Apply configurations to collectors.

   For more information, see ***Applying Configurations to Collectors* on page 36**.
4. Activate VCM and VCM configuration datasources.

   In Automated Analytics, activate the datasources to show collector data in Performance Monitor. For more information, see the *Automated Analytics Administrator Guide*.
5. Validate your deployment by adding collectors to the backend system(s) and using them to verify you have data available in Performance Monitor.

   For more information, see the *VCM User Guide*.
6. Add datasources in Automated Analytics.

   Add any additional datasources that are important to you (VMware vCenters, cloud datasources, and so on). For more information, see the *VCM User Guide*.
7. Data Reporting and Analysis

   Add groups for Key Performance Indicator analysis (Rightsizing, Health, Risk, and Chargeback). For more information, see the *VCM User Guide*.

Activating VCM and VCM Configuration Datasources

1. From the **Edit Datasources** dialog, click the **VCM** datasource.

   The VCM datasource is a default datasource that is created in an inactive state. You must connect to this datasource.

> **NOTE:**
> The VCM datasource is created in an inactive state by default. You need to connect this datasource to retrieve metric data stored by installed collectors.

The properties of the datasource are displayed.

2. Click **Connect**.
3. Click the **All Datasources** link to close the datasource properties.
4. From the **Edit Datasources** dialog, click the **VCM Configuration** datasource.

> **NOTE:**
> The VCM Configuration datasource is created in an inactive state by default. You need to connect this datasource to retrieve status information from installed collectors.

The properties of the datasource are displayed.

5. Click **Connect**.
6. Click the **All Datasources** link to close the datasource properties.

Adding Datasources in Automated Analytics

Datasources are used to retrieve data for creating capacity plans and calculating health. A datasource is created automatically during installation to retrieve data stored by VCM collectors. If you have data from other monitoring solutions, you can create additional datasources. For information on creating datasources in Automated Analytics, click Help from the Automated Analytics interface and locate "Managing Datasources" in the *Automated Analytics Administrator Guide*.

Service Names and Ports

The VCM services use the following ports:

Service Name	Port
vityl-cassandra	7000 7001 9042 9160
nginx	80 443
postgresql-17	5432

Service Name	Port
vityl-collector	N/A
vityl-collector-api	3703
vityl-console	3702
vityl-console-ui	3705
vityl-data-analytics	3708 3709 3710
vityl-data-management	N/A
vityl-kafka	9092
vityl-message-broker	8162 3707
vityl-metrics-api	3701
vityl-model-solver	N/A
vityl-plans-api	3706
vityl-spark-controller	6066 7077 4702
vityl-spark-worker	4703
vityl-storm-logviewer	4701
vityl-storm-nimbus	6627
vityl-storm-supervisor	6700 6701 6702 6703 6704 6705 6706 6707 6708 6709
vityl-storm-ui	4700
vityl-zookeeper	2181

Validating the Deployment

Fortra recommends that you perform an end-to-end validation of your deployment before you deploy any collector.

To validate your deployment:

1. Deploy your collectors to each installation system. For more information on collector deployment, see ***Installing Collectors* on page 20**.
2. Use the **Collector** component to validate that each collector is running.
3. Use the **Performance Monitor** component to validate that data for each system is available.

   > **NOTE:**
   > If the system is not available in the Performance Monitor component, refresh your datasources by clicking Refresh Datasources in the Datasources component.

Accessing VCM

To access VCM:

1. Open a browser window and enter the following URL:

   `https://`*`system`*

   where *system* is where VCM is installed.
2. Enter your username and password.

   By default, admin and admin.

Applying the License Key and Configuring Collection Settings

To apply your license key and configure collection settings:

1. Navigate to **https://*system1*** and log in with the default credentials (the username is admin and the password is admin).
2. Insert your license key when prompted.
3. Navigate to the **Collectors** component and change the **Kafka URI** field to the DNS hostname of system2. Specify **9092** as the port number. This configures the location of Apache Kafka.

Applying Configurations to Collectors

You must apply a configuration to a collector before it can collect data. This is done in the Collectors user interface of VCM. For information on applying a configuration to a collector, see the *VCM User Guide*.

Changing Collector Configuration After Installing or Upgrading Collectors

After installing or upgrading a Collector on Microsoft Windows systems, the startup type and the account that the Collector runs under are reset to the default settings. The default startup type is Manual and the default account is LocalService. You can change these settings using the service control tool (sc.exe).

> **NOTE:**
> The below commands must be run from a command prompt using the Administrator account.

To change the Collector configuration:

1. Change the startup type to automatic using the following command:

   ```
   sc config vitylcollector start= auto
   ```

2. Change the account the Vityl Collector service runs under to the LocalSystem account using the following command:

   ```
   sc config vitylcollector obj= ".\LocalSystem" password= ""
   ```

   > **NOTE:**
   > You can run the above commands at the same time using the following command in their place:
   > `sc config vitylcollector obj= ".\LocalSystem" password= "" start= auto`

3. Restart the Vityl Collector service using the following commands:

   ```
   sc stop vitylcollector
   sc start vitylcollector
   ```

Configuring Strong Passwords

VCM passwords can be configured in /etc/vityl/console/appsettings.json to enforce stronger password use.

By default the settings are:

"UserAccount": {

"Password": {

"RequireDigit": false,

"RequireLowerCase": false,

"RequireNonAlphanumeric": false,

"RequireUppercase": false,

"RequiredUniqueChars": 0,

"RequiredLength": 0

}

},

Modify the settings and restart vityl-console (systemctl restart vityl-console) for stronger password enforcement. Enforcement is done on new user creation or password change. Existing passwords will continue to work.

Encryption Keys

When VCM is installed, encryption keys will get automatically generated (unless they existed previously).

To generate new keys:

Enter the command:

```
opt/vityl/pass/generate-key.sh -t <type>
```

where <type> is one of the following:

• 'password' - Key used to encrypt passwords in configuration files

• 'token' - Key used by the OAuth Proxy component of the VCM token handler in NGINX

If a key file already exists for the type of key being generated, the file will not be overwritten by default. To create a new key, either rename the existing file (if you wish to save the existing key), or invoke generate-key.sh with the '-f' option to force it overwrite the existing file. For example, `generate-key.sh -t pass -f`

Enabling Password Encryption in ActiveMQ

1. Log in as root user to the system where ActiveMQ is running.
2. Edit the **/etc/vityl/message-broker/security.xml** file.
   a. From the configurationEncryptor bean, find the line that specifies the encryption key (<property name="password" value="admin"/>). The default encryption key is "admin".
   b. Update the default encryption key to a new one. This should only consist of alphanumeric characters.
   c. Save the file.
3. Run **/opt/vityl/message-broker/bin/encrypt-passwords.sh**.

   This encrypts any unencrypted passwords in credentials-enc.properties and jetty-realm.properties.
4. Restart the message broker service.
5. Verify the passwords in credentials-enc.properties are encrytped and enclosed by ENC().

   EXAMPLE: activemq.password=ENC(L/qwBI5CJjRttcd+BrKa1Zv7xz/deDYa).
6. Verify the passwords in jetty-realm.properties are encrypted and enclosed by ENC():

   EXAMPLE: admin: ENC(8eUBE+35B1HnKzq2ZeIg3Mrrq2PjTDZQcIn4pK3imN0=), admin.
7. Verify that VCM is functioning and you can log into the ActiveMQ UI.
8. If everything is functioning as expected, go to **/etc/vityl/message-broke**r and remove any .bak files containing the unencrypted passwords.

   NOTE: If you have previously upgraded VCM, there may be .rpmsave files that contain unencrypted passwords. You can remove these files.

Configuring Audit Logging and Message Formatting

You can configure audit logging and message formatting to ensure proper logging of user activities for compliance and troubleshooting. Configuration for User Access Logging is stored in `/etc/vityl/console/appsettings.json`. Audit logs are written to `/var/log/vityl/console/audit-YYYYMMDD.log`.

Default Configuration

```
"Audit": {
 "Enabled": true,
 "MinimumLevel": "Information",
 "RetentionDays": 90,
 "MaxFileSizeMB": 3
}
```

where:

- "Enabled": true - When this is set to true, audit events are logged to files. When this is set to false, no audit logging occurs.
- "MinimumLevel": "Information" - Sets the minimum log level for audit events.
  - Options: "Debug", "Information", "Warning", "Error", and "Fatal"
- "RetentionDays": 90 - Automatic cleanup - how many days of audit logs to keep.
  - Keeps: audit-20251125.log (today) back to audit-20250826.log (90 days ago).
  - Deletes: Files older than 90 days are automatically removed.
- "MaxFileSizeMB": 3 - Prevents individual log files from becoming too large.
  - When a daily log file reaches 3 MB, Serilog creates additional files with sequence numbers.
  - For example,

    ```
    audit-20251125.log (up to 3 MB)
    audit-20251125_001.log (if the first file exceeded 3 MB)
    audit-20251125_002.log (if needed for high-volume days)
    ```

Log File Format

```
{
 "timeStamp":"2025-12-17T18:56:11.9761846Z",
 "source":"10.60.154.131",
 "userId":"admin",
```

```
 "eventType":"login",
 "componentName":"Authentication",
 "status":"success"
}
```

where

- **timeStamp** is the date and time that the event occurred.
- **source** is the source IP address of the request.
- **userId** is the authenticated user performing the action.
- **eventType** is the specific audit event type.
- **componentName** is the system component handling the event.
- **status** is the outcome of the action.

Data Reporting and Analysis

VCM runs two automated analytic processes that provide health and risk key performance indicators (KPIs) for collected systems. By default, risk KPIs run daily while health KPIs run hourly.

When deploying in production, Fortra recommends that you configure KPI runs using several system groups. Configure groups of homogenous systems (grouped by system type, application, service, data center, boundaries, and so on) to perform health and risk KPI analysis. Configuring multiple sets of smaller groups can improve KPI analysis performance (versus analyzing a few, very large groups).

To configure KPI runs:

1. Configure and run **KPI analytics**.

   For information on creating a group for these systems, see the *Key Performance Indicators* in the *VCM User Guide*.
2. Validate that health and risk values are available in the Key Performance Indicators component.

Data Summarization and Retention

VCM runs a daily process to manage stored data. This process provides data retention and summarization. The retention process is based on day boundaries, allowing you to configure how many days each supported time interval is retained. To more efficiently support long-term data storage, each time interval is summarized to the next larger time interval.

This process is performed with Apache Spark. Apache Spark provides a framework for running tasks in parallel across multiple worker systems, providing the ability to scale retention and summarization processing horizontally as data volumes increase.

Configuring Data Summarization

Metric data can be collected at intervals as low as 1 second. Storing data for long periods of time for historical reporting or for capacity planning purposes using short collection intervals can require a large amount of disk space. To provide higher granularity metric data for longer term storage, metric data is summarized once a day. All collected metric data is stored with Coordinated Universal Time (UTC) timestamps. By default, this process runs one hour after midnight (UTC time).

This process summarizes all data that is stored in the 3rd day prior to the current day. For example, if it is the 4th day of the month, the process summarizes data that is stored in the database for the 1st day of the month. On the 5th day, the 2nd day is processed, and so on.

Depending on your local time zone, you might need to adjust the summarization run time of to occur when real-time monitoring usage is unexpected.

For example, if you are deploying VCM in Tokyo, the default summarization offset time results in summarization taking place at 9:00 a.m. in Tokyo time. However, you can change the "dailyExecutionOffsetInMinutes" value to 780 minutes (13 hours) to change your summarization time to 10:00 p.m. local time (1:00 p.m. UTC time).

To adjust the summarization execution time:

1. Open the **/etc/vityl/data-management/application.conf.json** file.
2. Change the **dailyExecutionOffsetInMinutes** value as needed for your deployment.

   Under the summarization section, the dailyExecutionOffsetInMinutes key value defines the number of minutes after UTC midnight that should elapse before the summarization process begins. The default value for dailyExecutionOffsetInMinutes is 60.
3. Save your changes.
4. Restart the **Vityl Data Management** service to apply your changes using the following command:

   ```
   sudo systemctl restart vityl-data-management
   ```

SELinux and Firewall Settings

During installation, the SELinux and firewalld settings are updated to allow VCM to run.

If you enable SELinux or firewalld after installation, update the settings using the following command:

```
/opt/vityl/console/security-helper.sh
```

For firewalld, the helper script enables a minimal set of firewalld services within the default zone, which is typically the public zone.

> **NOTE:**
> If a 502 Bad Gateway error occurs when trying to connect to the 443 SSL port, rerun the security-helper.sh script.

Configuring Additional Automated Analytics Instances

Automated Analytics performs data retrieval and analysis functions for the VCM application. In some deployment scenarios, multiple instances of Automated Analytics can be deployed to provide additional capacity for these functions.

When multiple instances are deployed, a single instance is considered the primary instance, which provides the user interface for Automated Analytics to the end user. Additional instances are considered secondary instances, which provide additional capacity for data retrieval and analysis functions and do not provide a user interface.

Consult your Fortra representative when deploying VCM to determine if additional instances are required for your deployment.

Installing Secondary Automated Analytics Instances

Primary Automated Analytics Instances

A primary Automated Analytics instance delivers the user interface to the end user to configure itself and all secondary Automated Analytics instances. Existing Automated Analytics instances are considered primary instances.

Before you can add a secondary Automated Analytics instance, reconfigure the postgresql-17 server to allow remote connections. You also might need to make firewall adjustments to allow remote connections to the ports for vityl-message-broker (port 3707) and postgresql-17 (port 5432).

POSTGRESQL-17

Run the following script to configure Postgresql to allow remote connections:

```
/opt/vityl/postgres/allowRemoteConnections.sh
```

Secondary Automated Analytics Instances

To add a secondary Automated Analytics instance, install vityl-data-analytics on an additional Linux virtual machine. After it is installed, do the following to configure the secondary instances:

USER.PROPERTIES

In the /opt/vityl/data-analytics/webapps/surveyor/WEB-INF/user.properties file, change the hostname in the messaging.broker.url property as follows:

```
messaging.broker.url=failover://
(ssl://localhost:3707)?randomize=false
```

Replace localhost with the hostname or the IP address of the primary Automated Analytics instance to connect to the broker running on the primary instance. Add the messaging.secondary property to the other messaging.broker properties as follows:

```
messaging.secondary=true
```

Broker Truststore

After the primary Automated Analytics instance is up and running and port 3707 is accessible, run the following command with the hostname or IP address in place of primary:

```
sudo /opt/vityl/data-analytics/bin/add-broker-cert.sh primary:3707
```

The script creates a truststore to allow this host to validate the SSL certificate from the message broker and restarts the vityl-data-analytics service, which activates your configuration changes.

Vityl Plans API Settings

To see all datasource and module information and prevent unexpected behavior of the status pages, add the following setting to the AnalyticService section of the appsettings.json file located in the /etc/vityl/plans-api/ directory:

```
"AnalyticsServiceHosts" : x
```

where *x* is the current number of Automated Analytics instances.

Upgrading Secondary Automated Analytics Instances

Regardless of if you are upgrading all components or just the Automated Analytics instance, you should start the upgrade process with the primary Automated Analytics instance. The primary instance could be on the same server as the other Automated Analytics components. To locate the primary instance, look at the user.properties file. Primary instances do not include the messaging.secondary=true property.

To upgrade secondary Automated Analytics instances:

1. Stop all **Automated Analytics** instances.
2. Upgrade the **primary Automated Analytics** instance.

   If you are updating all VCM components and the primary Automated Analytics instance is on the same server as the other VCM components being upgraded, the update vityl-* includes the primary Automated Analytics update.
3. Upgrade your **VCM components** as needed.
4. Upgrade all **secondary Automated Analytics** instances.
5. Stop and restart each instance.

   The order in which you stop and restart instances is not important. You can perform multiple instance upgrades consecutively on different servers.
6. For each instance, verify that the instance is connected to the broker by doing the following:
   a. Go to the **primary Automated Analytics** instance.
   b. Click **Tools** and select **Data Service**.

      The upgraded Automated Analytics instance appears in the Servers list.
   c. Click **Enable Data Synchronization**.

Configuring Apache Kafka Data Retention

Collected metric and heartbeat data is sent to Apache Kafka. Apache Kafka retains this data for a period of time, allowing a window of time to process the data before it is removed. By default, Apache Kafka retains data for 168 hours for all topics. You can change the retention for each topic to meet your application resiliency and disk space usage requirements.

To configure the retention for a topic:

Run the following command:

```
kafka-configs --bootstrap-server hostname:port --alter --entity-type
topics
--entity-name topic name --add-config 'retention.ms=retention time'
```

where:

hostname:port	is the name of the host or the IP address of the system running Apache Kafka, plus the port (9092 by default). On a single system, you can use localhost or 127.0.0.1.
topic name	is the name of the Apache Kafka topic to alter (Heartbeat or Metric)
retention time	is the data retention for the topic in milliseconds

Example

The following example shows altering the Apache Kafka retention to 12 hours for Heartbeat and Metric data:

```
kafka-configs --bootstrap-server localhost:9092 --alter --entity-
type topics
--entity-name Metric --add-config 'retention.ms=43200000'

kafka-configs --bootstrap-server localhost:9092 --alter --entity-
type topics
--entity-name Heartbeat --add-config 'retention.ms=43200000'
```

To validate Kafka topics with data retention:

If the Metric or Heartbeat topic already exists, you can validate the retention setting for the topic using the following command:

```
kafka-configs --describe --bootstrap-server localhost:9092
--entity-name=topicName --entity-type=topics
```

Deployment Monitoring

After you install and deploy VCM, you can monitor your deployment using Apache Storm, Apache Spark, or VCM.

Monitoring the Deployment With Apache Storm

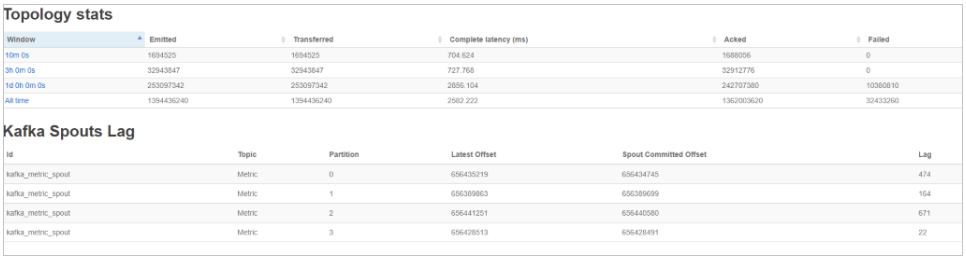
When processing large volumes of data, data can be received by Apache Kafka faster than it can be processed by Apache Storm. The Apache Storm framework provides a Lag measurement that allows you to monitor the current position of the Apache Storm processing versus the lead position in Apache Kafka.

To view the Apache Storm lag value:

1. Access **Apache Storm** at **http://system:4700**.
2. Locate the **Topology Summary** section.
3. Click the **VitylMetricTopology** link.
4. Locate the **Kafka Spouts Lag** section. On the far right, a Lag value for each Apache Kafka Metric topic partition is displayed.

A single Lag value is not meaningful. Monitor the change in this value over time based on the shortest collection interval. For example, the shortest collection interval is 1 minute. Monitoring the Lag value every minute and seeing an increasing value over time indicates that more data is arriving than is being processed each minute and changes are needed to support the current message volume.

The following is an example of Kafka Lag information from the Apache Storm user interface:



Topology stats

Window	Emitted	Transferred	Complete latency (ms)	Acked	Failed
10m 0s	1694525	1694525	704.624	1688056	0
3h 0m 0s	32943847	32943847	727.768	32912776	0
1d 0h 0m 0s	253097342	253097342	2856.104	242707380	10388610
All time	1394436240	1394436240	2582.222	1362003620	32433260

Kafka Spouts Lag

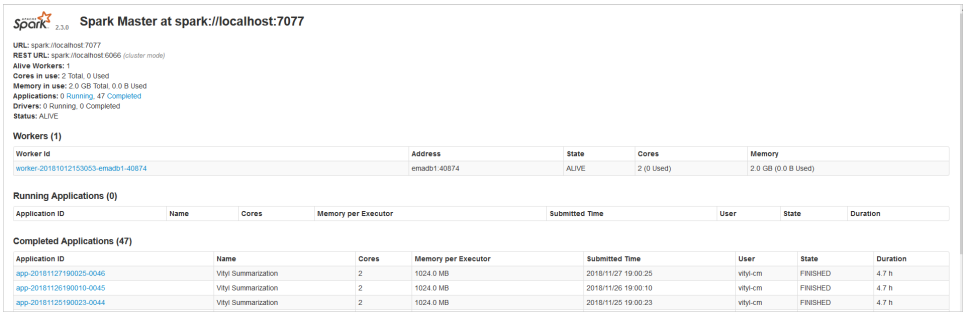
Id	Topic	Partition	Latest Offset	Spout Committed Offset	Lag
kafka_metric_spout	Metric	0	656435219	656434745	474
kafka_metric_spout	Metric	1	656389863	656389699	164
kafka_metric_spout	Metric	2	656441251	656440580	671
kafka_metric_spout	Metric	3	656428513	656428491	22

Monitoring the Deployment With Apache Spark

The Apache Spark summarization and retention process runs daily. By default, this process starts 1 hour after midnight (UTC time). This ensures that all of yesterday's data is available for summarization. You can change this time offset as needed.

Apache Spark allows you to monitor your summarization processes at http://system:4702. From this user interface, you can monitor the run duration for each daily job. If this process is taking too long, consult with Fortra for scaling assistance.

The following example shows a sample Apache Spark status display:



Monitoring the Deployment With VCM

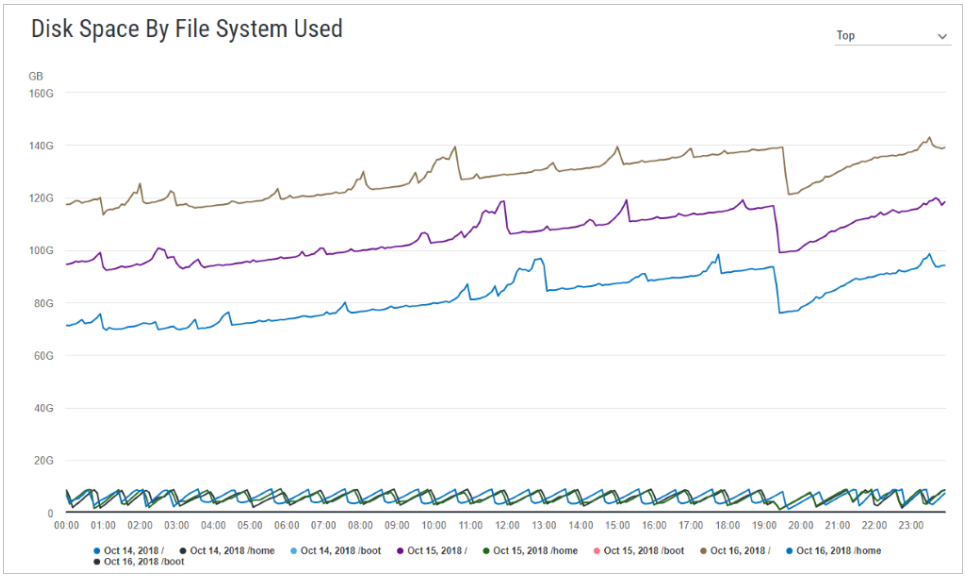
As a performance monitoring tool, VCM can report on itself, helping you understand how your system resources are used and when to make scaling changes.

The Performance Monitor Overview charts provide a good starting point. Monitor CPU, memory, disk IO, and disk space usage on all VCM back-end systems as you would any other application in your enterprise. If performance problems are encountered, Fortra can help you determine the source of the issue and recommend deployment changes.

Cassandra Disk Space Usage

The following example shows the growth of disk space usage on an Apache Cassandra system. This example shows 3 days of data (overlaid) where the 1-minute retention has not been reached. After the 1-day retention is reached, the gap between days is decreased but continues to grow until the 5-minute, 15-minute, and 1-hour retentions are reached. At that time, you can expect disk space usage to level off.

In this example, the top 3 lines are the data files while the bottom 3 lines are the commit logs. You can see the commit logs stay within a range. You also see the occasional small and larger spikes in the used space for the data files. These spikes occur when Apache Cassandra runs a compaction on a table, which means you need to allow for some overhead in disk space to handle these jobs.



Contacting Fortra

Please contact Fortra for questions or to receive information about VCM. You can contact us to receive technical bulletins, updates, program fixes, and other information via electronic mail, Internet, or fax.

Fortra Portal

For additional resources, or to contact Technical Support, visit the Fortra Support Portal.

For support issues, please provide the following:

- Check this guide's table of contents and index for information that addresses your concern.
- Gather and organize as much information as possible about the problem including job/error logs, screen shots or anything else to document the issue.