Vertica Reference Guide

Vertica is an analytics database platform that is specifically built for Big Data analytics. Vertica is built for organizations that generate large volumes of data frequently. Compared to traditional relational database systems (RDBMS), Vertica provides faster query performance and scalability while analyzing these large volumes of data. The most important features of Vertica include:

- Being a columnar database, it stores information in columns and only reads the columns required to answer a query.
- Highly scalable as Vertica runs on a cluster of commodity hardware. As and when the volume of data generated increases, you can add more nodes to the Vertica cluster to increase processing speeds.
- Compresses data size to a significant extent, which improves analytic performance while processing historical or trending data.
- Queries and loads data on a real-time basis up to 10X faster than traditional RDBMS.
- Efficiently handles multiple operations with multiple concurrent users.

Some other features include:

- Supports standard programming interfaces that include ODBC, JDBC, ADO.NET, and OLEDB.
- Uses database designer and administration that allows you to control Vertica with minimal administration effort.
- Analyzes semi-structured data along with structured data.

For more information on the Vertica database, see the documentation available at www.vertica.com.

The topics addressed in this guide are as follows:

ZENworks and Vertica

You can leverage the capabilities of Vertica when using the Dashboards feature in ZENworks. While all dashlets, with the exception of trending charts, will be displayed when connected to the existing RDBMS; to reduce any delays that you might face while querying large volumes of data in the dashlets, you can configure the Vertica database.

As Vertica stores and reads information from columns leading to faster processing speeds, the data in the dashlets will be processed and displayed much faster compared to an RDBMS database. Also, as Vertica has the ability to compress data size, it can store historical data, thereby enabling you to view historical trends related to patches and CVEs. For more information on the Dashboards feature in ZENworks, see the ZENworks 2020 Dashboard Reference guide.

Being an in-house Micro Focus product, the Vertica database is license free for up to 20 TB of data, beyond which you will have to purchase a license for Vertica. Vertica is packaged with your ZENworks 2020 virtual appliance build.

Points to Consider before moving to Vertica

This section is intended to help you make an informed decision about moving to the Vertica database. Some important aspects that you need to carefully consider before enabling Vertica in your zone are:

- ZENworks does not mandate the use of Vertica. You can consider configuring Vertica if:
  - You have more than 10 thousand devices in the zone and you use the Common Vulnerabilities and Exposure (CVE) feature. The CVE dashlets will load data at much faster speeds when Vertica is enabled in the zone.

NOTE: For zones with a lesser number of devices, you can continue using the existing RDBMS, as all features (except for trend charts) will continue to work seamlessly.

- You want to view the trending charts displayed within the Patch Tracker and CVE Tracker dashlets. Irrespective of the number of devices in the zone, trend data will be displayed only when Vertica is enabled.

- When you configure Vertica in the zone:
  - Vertica will only process data for the Dashboard feature.
  - ZENworks will process data from Vertica only for the Security dashlets. The bundle dashlets will read data from the existing RDBMS unless you manually configure these dashlets (Device Assignment Status dashlet and the Bundle Deployment Status dashlet) to obtain data from Vertica, which is documented in the Software Distribution Guide.

NOTE: The bundle dashlets will work seamlessly with the existing RDBMS, even when you have a sizable number of devices in the zone.

- ZENworks will continue to use the existing RDBMS as the primary database.
  - The RDBMS will continue to receive data from the ZENworks Primary Server. As an initial step, data will be migrated in bulk from the RDBMS to Vertica, after which the RDBMS will continuously sync data with Vertica at an interval of every 10 to 15 minutes, using the Apache Kafka and Apache ZooKeeper components.
  - The RDBMS will retain all data, except for the following which will be directly stored in Vertica:
    - CVE trend data and Patch trend data: If you remove Vertica from the zone, you will lose this data.
    - CVE Vulnerability Status data.
- ZENworks currently does not provide you with a control to disable Vertica automatically, if you no longer want to use it in your zone. You need to manually disable it in the zone, which is documented in Disabling Vertica and Reverting to RDBMS.

Prerequisites

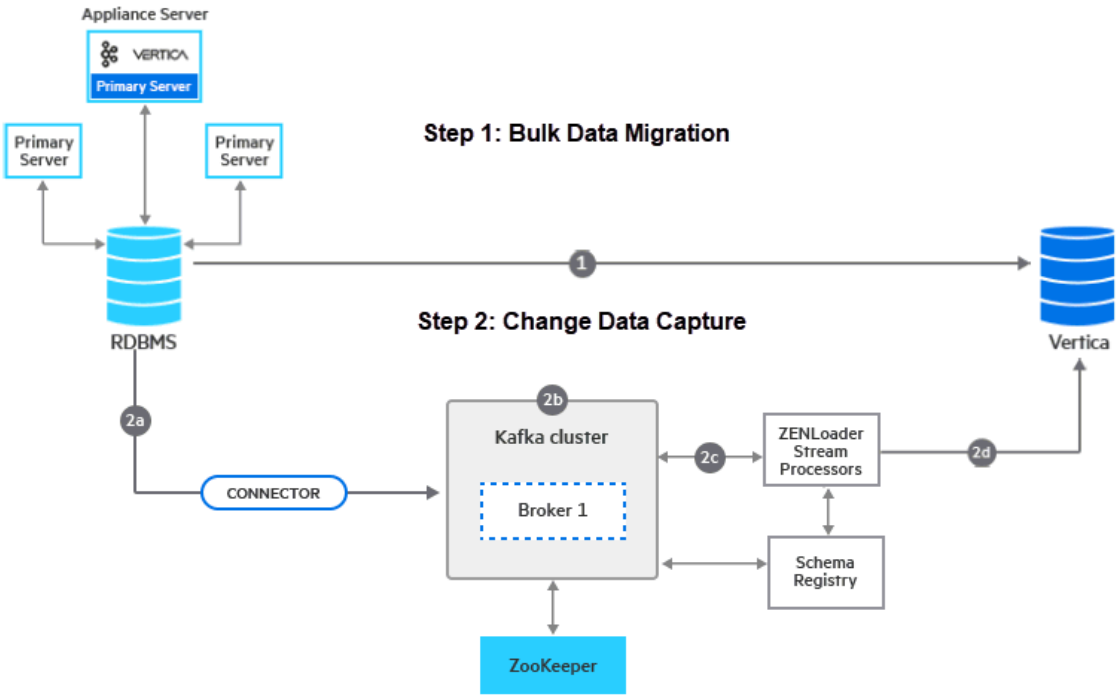
The prerequisites that need to be met before enabling and using Vertica are as follows:

- The system requirements as stated in the section System Requirements and Other Recommendations are met.
- To ensure appropriate memory is allocated for the Vertica and Kafka components and the ZENworks services on the server, run the **Calibrate Memory Configure action** on the server on which either Vertica and Kafka are installed. This action should be run after Vertica and Kafka are installed and before you proceed with bulk data migration. For more information, see Managing Vertica Memory Requirements.
- ZENworks 2020 virtual appliance build is deployed in the zone. Only Appliance servers can be a part of the Vertica and Kafka clusters.
- Primary Servers that have the Kafka, Vertica and Collection roles in them, require a two-way SSL certificate with client authentication enabled. If you are using an external CA certificate ensure that client authentication is enabled.
- The **Modify Zone Infrastructure** rights should be configured for your administrator. For more information on enabling these rights, see ZENworks Administrator Accounts and Rights Reference.

Workflow

Apache ZooKeeper that is packaged with the ZENworks 2020 build, is automatically enabled when the build is deployed on the first Primary Server. A brief summary of the steps to be performed to enable Vertica and to migrate your existing data to it, are as follows:

- **Enable Kafka in the zone:** Apache Kafka is already packaged with the ZENworks 2020 virtual appliance build and will initially be in a disabled state. You need to use the ZMAN utility to enable the Kafka cluster.
- **Enable Vertica in the zone:** Vertica is already packaged with the ZENworks 2020 virtual appliance build and will initially be in a disabled state. You need to use the ZMAN utility to configure Vertica in the zone. Ensure that you enable Kafka in the zone, before enabling Vertica.
- **Migrate data:** Migration of data from the existing RDBMS to Vertica includes two parts, which is represented in the following illustration:



Step 1 - Bulk Data Migration: As soon as Vertica is enabled in the zone, you need to perform bulk data migration from the existing RDBMS to Vertica by executing a Configure action. After migrating data in bulk, ZENworks will start processing queries from Vertica for all the dashlets in ZENworks. For more information, see Migrating Data to Vertica.

Step 2 - Change Data Capture: After the initial bulk migration process, as and when data is modified in the existing RDBMS, for these changes to reflect in Vertica, the RDBMS syncs with Vertica at an interval of every 15 minutes. Apache Kafka along with ZooKeeper is responsible for performing this sync action. For more information on this workflow, see Kafka Change Data Capture Workflow.

Viewing the Configuration Status in ZCC

To view the configuration status of the ZooKeeper, Kafka, Vertica clusters and the Bulk Data Migration process, you can navigate to the Getting Started page in ZCC. To navigate to this page, click **Configuration** > **Performance Upgrade**.

Each task on this page, includes an icon that indicates the following status:

- : the component is successfully configured.
- : the component is ready to be configured. For example, the Bulk Data Migration component will be ready to be configured only when Kafka and Vertica components are configured.
- : an error has occurred while configuring the component.

System Requirements and Other Recommendations

The minimum hardware requirements and other recommendations for the Vertica and Kafka servers (nodes) are:

- A single node cluster is recommended if you have less than 10000 devices in the zone.
- For a single node cluster, you can configure both Kafka and Vertica on a single node.
- If you have more than 10000 devices in your zone, then you require more than 1 Vertica node. As the minimum size of a cluster is 3 nodes, a minimum 3 node cluster is required for such zones.
- For multiple node cluster, it is recommended that Vertica, Kafka and the Collection role are configured on different servers. As Vertica and Kafka are Disk I/O (input/output) intensive, Kafka should be enabled on a server that does not have Disk I/O intensive roles configured on it. As the Collection role is also Disk I/O intensive, Kafka should not be configured on the same server on which the Collection role is enabled.
- The minimum hardware requirements for the Vertica and Kafka servers are available at the location https://www.novell.com/documentation/zenworks-2020/pdfdoc/vertica_system_requirements/vertica_system_requirements.pdf.The minimum hardware configuration was derived based on the number of devices in the zone and the number of administrators accessing the Dashboards simultaneously.
- Ensure that the hardware configuration for all the servers within the Vertica cluster are the same.
- Ensure that all the nodes in the cluster are in the subnet to enable internal cluster communication. For more information on the network configuration recommendations, see the Vertica documentation.

For more information on other recommendations, see the Vertica documentation.

ZooKeeper

ZooKeeper acts as a coordination service that provides flexible and robust synchronization within distributed systems. ZooKeeper forms an integral part of Kafka as it keeps track of the status of the Kafka cluster nodes and the Kafka topics and partitions. The leader election between the Kafka brokers is also performed by using Zookeeper when the leader node fails. For more information Kafka brokers and other Kafka terminologies, see Apache Kafka.

Apache ZooKeeper is already packaged with the ZENworks 2020 build. When the build is deployed on the first Primary Server, ZooKeeper is automatically enabled in the zone. However, you can expand the ZooKeeper cluster as per your scalability requirements and to provide fault tolerance.

For more information on adding additional servers to the ZooKeeper cluster, see Enabling ZooKeeper, Kafka, and Vertica on multiple servers.

For more information on monitoring the ZooKeeper cluster, see Monitoring ZooKeeper, Kafka and Vertica.

Apache Kafka

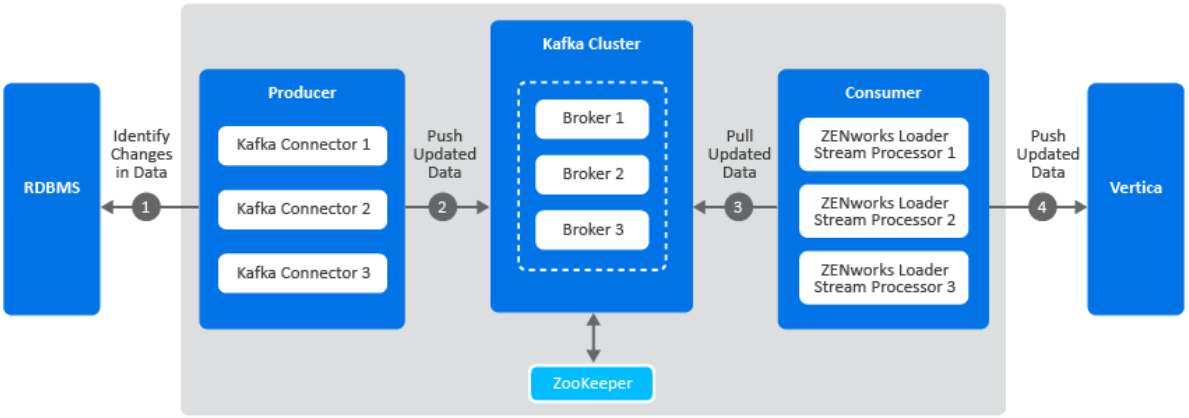
Apache Kafka is a distributed publish-subscribe messaging system that enables passing of messages from one system to another, while handling large volumes of data. Kafka can be enabled on a single Primary Server or can be run as a cluster on one or more servers that can span multiple data centers. Each server in the cluster is called a broker. Kafka is run as a cluster to ensure high availability of its services by replicating Kafka topics or messages to multiple Kafka brokers. Kafka requires ZooKeeper to co-ordinate between the servers within the Kafka cluster.

In the Vertica data migration workflow, Apache Kafka is required to sync data with Vertica, after the bulk data migration process. When Vertica is enabled, the ZENworks server will continue to send data to the existing RDBMS. After bulk data migration, if any data is added or modified in the RDBMS, Kafka syncs these changes with Vertica at an interval of every 10 to 15 minutes.

NOTE: To secure the Kafka cluster, Kafka allows clients to connect over SSL and uses the same Primary Server certificate on which it is installed. However, if the server certificate is issued by an external CA, then two-way SSL should be enabled for Kafka to work seamlessly.

Kafka Change Data Capture Workflow

The following diagram is a graphical representation of the Kafka workflow:



A description of each component in this architecture is as follows:

- **Kafka Cluster:** A group of servers or nodes that are connected to each other to achieve a common objective is a cluster. Each of the servers or nodes in the cluster, will have one instance of Kafka broker running.
- **Kafka Broker:** One or more servers that are added in a Kafka cluster are called brokers.
- **Apache ZooKeeper:** Kafka uses ZooKeeper to manage and co-ordinate between brokers. It notifies Kafka Producers and Consumers of any broker failures.

- **Kafka Producers:** The processes that publish messages to Kafka brokers. In this case, Kafka connectors are created as soon as Kafka is enabled. These connectors are created for each table in the RDBMS database and is responsible for identifying changes in these tables and publishing them to Kafka.
- **Kafka Consumers:** A service that reads data from Kafka brokers. In this case, the ZENworks loader stream processors subscribes to data from Kafka and passes this information to the Vertica database.

The Kafka workflow is as follows:

1. The Kafka connectors (Kafka producer) identifies changes in the respective RDBMS database tables.
2. These changes are published to a topic within a Kafka broker. Kafka maintains messages as categories and these categories are called topics. To modify the interval at which the Kafka connector identifies changes in the RDBMS tables and publishes this data to Kafka, you can update the connector-configs.xml file (available at `etc/opt/novell/zenworks`) and run the `zman srkccn` command to re-configure the connectors. For more information on this command, see Maintaining Kafka Connect.

   The topics are further broken down into partitions for speed and scalability. These partitions are replicated across multiple brokers, which is used for fault tolerance. The replicas are created based on the replication count specified while enabling the Kafka cluster. Each message within a partition is maintained in a sequential order, which is identified as an offset, also known as a position.
3. The ZENworks loader stream processors (consumers) subscribe to a topic, Kafka offers the current offset of the topic to the consumer and saves the offset in the ZooKeeper cluster.
4. The ZENworks loader stream processor receives the message and processes it to the Vertica database and the Kafka broker receives an acknowledgment of the processed message.

Kafka also uses a Schema Registry, which is a separate entity that producers and consumers talk to, for sending and retrieving schemas that describe the data models for the messages.

For more information on the Kafka Architecture, see the Confluent docs.

Enabling Kafka and Vertica on a single server

As stated earlier, ZooKeeper is already enabled on your first Primary Server. This section details the procedure to enable Kafka and Vertica on a single node (server). Both Kafka and Vertica can be enabled on the same server or two different servers.

The topics addressed in this section are:

- "Enabling Kafka" on page 7
- "Enabling Vertica" on page 9

Enabling Kafka

Apache Kafka is already packaged in your ZENworks 2020 Appliance build and will initially be in a disabled state. To enable Kafka, you need to execute ZMAN commands in the command line utility at both the cluster level and broker level. You need to first configure the Kafka cluster settings and subsequently add a broker to it.

Cluster Level:

- **server-role-kafka-configure-cluster (zman srkcc):** This command configures the Kafka cluster. While running this command, you need to specify the replication count. If the values for the remaining parameters are not specified, then the default values are considered. The parameters are:
  - **-c --replication count:** This ensures that messages remain available when a server in the cluster fails.Specify the number of copies to be maintained for each topic. For a single node cluster, this count can be set as 1.
  - **-l --logRetentionBytes:** Specify the maximum permissible size of the log, beyond which, the existing data is overwritten with the new data. By default the log size is unlimited.
  - **-t --zkSessionTimeout:** Specify the ZooKeeper session timeout (in milliseconds), which is the maximum time that the server waits to establish a connection to ZooKeeper. If the server fails to signal a heartbeat to ZooKeeper within this specified time period, then the server is considered to be dead. A heartbeat request helps identify if the server is still connected to the Kafka cluster. The default value is 30000 milliseconds.
  - **-r --retainDetectedLogsDuration:** Specify the maximum time to retain deleted logs. The default value is 86400000 milliseconds (1 day).
  - **-p --logCleanupPolicy:** Specify the default cleanup policy for segments that exceed the maximum permissible retention window. The possible values are Delete and Compact. The default value is Delete. The Delete policy will remove old segments when the retention time or size limit has reached. The Compact policy will enable log compaction on the topic, which ensures that Kafka will always retain at least the last known value for each message key within the log of data for a single topic partition.
  - **-s --schemaregistryport:** Specify the port on which the Schema Registry is running. The default value is 8081.
  - **-k --kafkaport:** Specify the port on which Kafka listens. The default value is 9093.
  - **-x --connectport:** Specify the port on which Kafka connect listens. The default value is 8083.

For example, `zman srkcc -c=1`

Broker Level

- **server-role-kafka-add-broker (zman srkab):** This command adds brokers or servers to the configured Kafka cluster. The parameters to be specified are: Specify the GUID, DNS or hostname of the server.
  - **--servers:** Specify the appliance server on which Kafka should be enabled.
  - **-i --ignorewarning message (optional):** As Kafka requires client authentication to be enabled in the certificate, if you are using an external CA certificate, ensure that client authentication is enabled and execute the command again with the option `i` set as true. Option `i` enables you to ignore the warning message and proceed with the execution of the command.

  For example: `zman srkab --servers=server1.microfocus.com`

  To add a Kafka broker, you need to ensure that the following prerequisites are met:
  - The Kafka cluster should already be configured.
  - The broker or server that is to be added to the cluster should be an Appliance server and should not already be added to the cluster.
  - The server should be accessible and should have a specific host name. If a server has multiple host names, then the command to add the Kafka broker might fail.

When this command is executed for a server, the Kafka, Kafka connect and Schema Registry services are enabled and started on the specified server.

For more information on debugging Kafka configuration issues, see Debugging Issues.

For more information on monitoring the Kafka cluster, see Monitoring ZooKeeper, Kafka and Vertica.

Enabling Vertica

NOTE: Before enabling Vertica, ensure that all the services within the server are up and running to avoid any port conflicts.

This section details the procedure to enable Vertica. The Vertica database is already packaged with the ZENworks 2020 virtual appliance build and will be in a disabled state. To configure Vertica in the zone, you need to execute the following command:

- **server-role-vertica-create-cluster (zman srvcc)** This command creates a Vertica cluster and adds a server based on the specified parameters. The parameters to be specified while executing this command are:
  - **--servers:** Specify the appliance server on which you want Vertica to be installed.
  - **K factor:** Specify the number of replicas of the data that should exist in the database. The K safety sets the fault tolerance for the Vertica cluster, where K can be set as 0 or 1. You need to specify a value based on the number of nodes in the cluster, which is measured as 2k+1. To enable Vertica on a single node cluster, the K factor can be specified as 0.

  For example: `zman srvcc --servers=server2.microfocus.com -k=0`

When this command is executed, Vertica is installed on the servers and the ZENworks database and its related schema are created.

IMPORTANT: After Vertica is enabled in the zone, ensure that you restart the ZENserver and ZENloader services on all the Primary Servers.

For more information on debugging Vertica configuration issues, see Debugging Issues.

For more information on removing a server, see Maintaining ZooKeeper, Kafka, and Vertica.

NOTE: After enabling Vertica, if you want to set a specific language option to be used for error messages and instructions, collating sequences, date formats, and so forth, then see the Vertica documentation. For more information on locales refer to the following links:

- https://www.vertica.com/docs/9.2.x/HTML/Content/Authoring/AdministratorsGuide/Diagnostics/Locales/AboutLocales.htm
- https://www.vertica.com/docs/9.2.x/HTML/Content/Authoring/AdministratorsGuide/Diagnostics/Locales/LocaleUTF8Support.htm
- https://www.vertica.com/docs/9.2.x/HTML/Content/Authoring/AdministratorsGuide/Diagnostics/Locales/SupportedLocales.htm

After changing the Language option, ensure that you are restart the Vertica database. For more information see, Stopping and Starting Vertica Services.

Managing Vertica Memory Requirements

As per the system requirements, the Appliance server on which Vertica is installed, should have minimum RAM size of 16 GB. Based on the RAM size, you need to allocate appropriate memory for Vertica, Kafka, the RDBMS (if embedded PostgreSQL is installed) and ZENworks services to function effectively on the Appliance server. ZENworks provides you with a Configure action that will automatically calibrate appropriate memory for these components.

This Configure action needs to be executed after installing Vertica and Kafka and before proceeding with bulk data migration. Subsequently, if you want to add more servers to the Vertica or the Kafka clusters, then you need to run this Configure action on each of the servers. The scenarios in which this action should be run are:

- Server on which Kafka and Vertica are installed
- Server on which only Vertica is installed
- Server on which embedded PostgreSQL and Vertica are installed
- Server on which embedded PostgreSQL and Kafka are installed
- Server on which only Kakfa is installed
- Server on which embedded PostgreSQL, Kafka and Vertica are installed

---

NOTE: The Configure action will fail to run, if the server does not meet the minimum memory requirement of 16 GB.

To run this Configure action:

1. In the command line utility of the Appliance server, execute the command: `novell-zenworks-configure -c CalibrateMemoryConfigureAction`
2. A message is displayed informing the user that the ZENworks services will automatically restart on the server after the Configure action is run. If you want to proceed with the memory calibration, then press enter to specify the default value as 1 or else enter the value 2.
3. Login as dbadmin and run the following commands to restart the Vertica database:
   a. `admintools -t stop_db -d zenworks -p <password> -F`
   b. `admintools -t start_db -d zenworks -p <password>`

      You can obtain the password by running the zman command `server-role-vertica-get-credentials (zman srvgc)`. You need to use the database superuser password to stop and start the Vertica database.

The newly configured values for each component are displayed. If you want to view the configured values at a later point in time, then run the following Configure action: `novell-zenworks-configure -c DisplayMemoryConfigureAction`.

Migrating Data to Vertica

NOTE: Before bulk data migration, ensure that you run a Configure action to allocate appropriate memory for ZENworks services and for the Vertica and Kafka components. For more information, see “Managing Vertica Memory Requirements” on page 10.

The data migration process to Vertica involves two steps:

- **Bulk Data Migration:** This process involves exporting of data in bulk to the configured tables in Vertica from the existing RDBMS. To initiate this process, from the command line utility of the Primary Server in which Vertica is installed, run the Configure action `novell-zenworks-configure -c VerticaDBMigrate -Doption=start`. If a Configure action that was executed previously had failed, then the start option will resume the operation from the point at which it had failed. To restart the migration process from the beginning, use the Force option. For example, `novell-zenworks-configure -c VerticaDBMigrate -Doption=force`. This option will remove existing data from the Vertica database.

  If the migration process fails, then check `novell-zenworks-configure.log` present at `/var/opt/novell/log/zenworks`. You can also view a detailed report at `/var/opt/novell/zenworks/vertica_migration_csv`.
- **Change Data Capture:** This event is automatically triggered after the bulk migration process. This process syncs modified or newly added data from the RDBMS to Vertica. The sync action is performed at an interval of every 10 to 15 minutes. For the syncing process, a queue action is initiated that creates Kafka connectors and topics for each table that is already migrated. The Kafka connectors identifies updated or modified tables in the RDBMS database and publishes this data to the appropriate topics. The ZENworks loader stream services (Kafka consumer) then subscribes to this data and sends it to the Vertica database.

  To monitor the status of the Data Sync process, navigate to the Diagnostics page in ZCC. For more information, see Monitoring ZooKeeper, Kafka and Vertica.

Enabling ZooKeeper, Kafka, and Vertica on multiple servers

This section details the procedure to move Vertica and its components from a single node cluster to a multiple node cluster. The prerequisites to be followed before adding servers to their individual clusters, are:

- Ensure that the hardware and software configuration of the servers that are to be added are the same as the configuration of the existing server within that cluster.
- Although there is no limit to the number of servers to be added in a cluster, ZENworks recommends a cluster size of 3 servers for ZooKeeper, Kafka and Vertica.

The topics addressed in this section are:

ZooKeeper

ZENworks recommends that you add odd number of servers to the cluster.

- **server_role_zookeeper_update_cluster (zman srzuc):** Execute this command to modify the cluster parameters. The parameters that can be specified using this command are:
  - **-l --leader-connect-port:** Define the leader port that the followers will use to connect to a leader in the cluster. The default port is 6790. However, you can specify an unused port between 6000 and 65535.
  - **-c --client-port:** Define the port on which ZooKeeper will listen in for incoming client connections. The default port is 6789. However, you can specify an unused port between 6000 and 65535.

- **-e --leader-elect-port:** : Define the election port that all servers in the cluster will use to elect the leader. The default port is 6791. However, you can specify an unused port between 6000 and 65535.
- **-t --tick-time:** Define the length of a single tick, which is a basic time unit in milliseconds, used by ZooKeeper to regulate heartbeats and timeouts. The default value is 2000.
- **-i --init-limit:** Define the amount of time, in ticks, to allow followers to connect to the leader. The default value is 10.
- **-s --sync-limit:** Define the amount of time, in ticks, to allow followers to sync with ZooKeeper. The default value is 5.
- **-x --max-connections:** Specify the maximum number of client connections that the ZooKeeper cluster can accept. The default value is 120.

For example: `zman srzuc -s=6`

You can now go ahead and add nodes (servers) to the ZooKeeper cluster:

- **server-role-zookeeper-add-server (zman srzas):** Execute this command to add nodes or servers to the cluster. The parameter to be specified is:
  - **--servers:** Specify a comma separated list of Primary Servers on which ZooKeeper should be enabled. You can specify the DNS, GUID or path of the server.

For example: `zman srzas --servers=server2.microfocus.com,server3.microfocus.com,server4.microfocus.com,server5.microfocus.com,`

Kafka

To add more servers in the Kafka cluster, you need to first add the servers to the cluster and then update the replication count to the number of servers added in the cluster.

- **server-role-kafka-add-broker (zman srkab):** This command adds brokers or servers to the configured Kafka cluster. The parameters to be specified are:
  - **--servers:** Specify a comma separated list of appliance servers on which Kafka should be enabled. You can specify the DNS, GUID or path of the servers.
  - **-i --ignorewarning message (optional):** As Kafka requires client authentication to be enabled in the certificate, if you are using an external CA certificate, ensure that client authentication is enabled and execute the command again with the option `i` set as true. Option `i` enables you to ignore the warning message and proceed with the execution of the command.

  For example: `zman srkab --servers=server1.microfocus.com,server2.microfocus.com`

  To add a Kafka broker, you need to ensure that the following prerequisites are met:

  - The server should be accessible and should have a specific host name.If a server has multiple host names, then the command to add the Kafka broker will fail.
  - When a broker is being added to the zone, tasks to remove or re-configure servers cannot be performed on other servers.

    When this command is executed for a server, the Kafka, Kafka connect and Schema Registry services are enabled and started on the specified server. The server is added to the cluster. When a broker is added in the cluster, ZENworks re-configures other existing brokers in the cluster. When re-configuration fails in one these brokers, you need to run the zman command (`zman srkrcb`) to re-configure the broker. For more information on this command, see Maintaining the Kafka Cluster.

- **server_role_kafka_update_cluster (zman srkuc):** Execute this command to modify the existing cluster parameters. The parameters are:
  - **-c --replication count:** This ensures that messages remain available when a server in the cluster fails.Specify the number of copies to be maintained for each topic. For a three node cluster, this count can be set as 3, which can indicates that a replica of the topic will be created on each server.
  - **-l --logRetentionBytes:** Specify the maximum permissible size of the log, beyond which, the existing data is overwritten with the new data. By default the log size is unlimited.
  - **-t --zkSessionTimeout:** Specify the ZooKeeper session timeout (in milliseconds), which is the maximum time that the server waits to establish a connection to ZooKeeper. If the server fails to signal a heartbeat to ZooKeeper within this specified time period, then the server is considered to be dead. A heartbeat request helps identify if the server is still connected to the Kafka cluster. The default value is 30000 milliseconds.
  - **-r --retainDetectedLogsDuration:** Specify the maximum time to retain deleted logs. The default value is 86400000 milliseconds (1 day).
  - **-p --logCleanupPolicy:** Specify the default cleanup policy for segments that exceed the maximum permissible retention window. The possible values are Delete and Compact. The default value is Delete. The Delete policy will remove old segments when the retention time or size limit has reached. The Compact policy will enable log compaction on the topic, which ensures that Kafka will always retain at least the last known value for each message key within the log of data for a single topic partition.
  - **-s --schemaregistryport:** Specify the port on which the schema registry is running. The default value is 8081.
  - **-k --kafkaport:** Specify the port on which Kafka listens. The default value is 9093.
  - **-x --connectport:** Specify the port on which Kafka connect listens. The default value is 8083.

    For example: `zman srkuc -c=3`

To remove a broker from the cluster or to re-configure a broker, see Maintaining the Kafka Cluster.

Vertica

To add more servers to the Vertica cluster, you need to add the servers and then update the K safety value of the Vertica cluster.

- **server-role-vertica-add-server (zman srvas):** This command adds servers to the existing cluster. Specify a comma separated list of servers by specifying the DNS, GUID or path of the server while executing this command.

  For example: `zman srvas --servers=server2.microfocus.com, server3.microfocus.com`

  IMPORTANT: After adding servers to the Vertica cluster, ensure that you restart the ZENserver and ZENloader services on all the Primary Servers.

- **server-role-vertica-update-cluster (zman srvuc):** This command lets you update the K-safety factor in the existing cluster. The K-safety value determines the number of replicas of data that should exist in a cluster. The K safety sets the fault tolerance for the Vertica cluster, where K can be set as 0 or 1. Specify a value

based on the number of nodes in the cluster, which is measured as 2k+1. For example: if the K factor is 1, then you need to have a minimum of 3 nodes. Vertica recommends a K-safety value of 1. For more information on K-safety, see Glossary.

For example: `zman srvuc -k=1`

Changing the ZooKeeper role from the first Primary Server to the second Primary Server

If the first Primary Server is up and running then you need to run the following zman commands:

1. **Add the second Primary Server to the ZooKeeper cluster:** Run the command `zman srzas --servers=<dns of 2nd primary>`.
2. **Remove the first Primary Server:** Run the command `zman srzrs --servers=<dns of 1st primary>`.

Connecting to Vertica

The dbadmin or the zenvertica user can connect to Vertica using the vsql tool. For more information on the vsql tool and to connect to the Vertica database, see the Vertica documentation.

To connect to Vertica, the user needs to specify the database credentials that can be obtained in either of the following ways:

- Execute the zman command `server-role-vertica-get-credentials (zman srvgc).`
- Obtain the credentials from the `zenworks-coreadmin` web service by referring to the following steps:
    1. On your browser, type the server address of the Primary Server that is part of the Vertica cluster, along with `zenworks-coreadmin`. For example: `https://10.0.0.0/zenworks-coreadmin.`
    2. Select **Test Service**.
    3. Select **Vertica Configuration** and click **Invoke**.

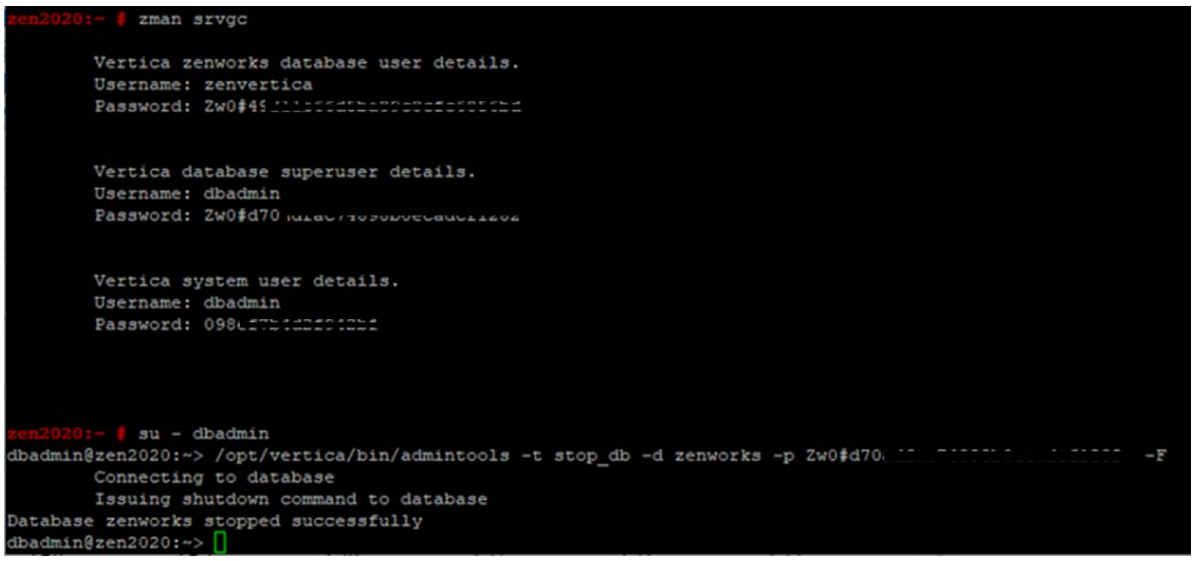
Stopping and Starting Vertica Services

To stop the Vertica service, you need to login as a dbadmin user and execute the following command:

```
$ /opt/vertica/bin/admintools -t stop_db -d db-name
```

```
[-p password] [-F]
```

Specify the dbadmin superuser password that can be obtained by executing the `zman server-role-vertica-get-credentials (zman srvgc)` command. Use the option -F (or --force) to override all user connections and force a shutdown.



To start Vertica services, execute the following command:

```
$ /opt/vertica/bin/admintools -t start_db -d <db-name> -p <password>
```

Maintaining ZooKeeper, Kafka, and Vertica

This section covers topics related to maintaining Vertica along with the Kafka and ZooKeeper clusters.

- "Maintaining the ZooKeeper Cluster" on page 15
- "Maintaining the Kafka Cluster" on page 16
- "Maintaining the Vertica Cluster" on page 17

NOTE: While executing the add or remove commands for a particular server, ensure that the server does not have multiple host names.

Maintaining the ZooKeeper Cluster

You can modify a ZooKeeper cluster and add or remove brokers from the existing cluster.

- **Updating the ZooKeeper Cluster:** Execute the command `server-role-zookeeper-update-cluster (zman srzuc)` to modify the cluster parameters. For more information on the parameters, see Enabling ZooKeeper, Kafka, and Vertica on multiple servers.
- **Adding Servers:** Execute the command `server-role-zookeeper-add-server (zman srzas)`. For more information, see Enabling ZooKeeper, Kafka, and Vertica on multiple servers.
- **Removing additional ZooKeeper servers:** Execute the command `server-role-zookeeper-remove-server (zman srzrs)`. This command removes any additional server from the existing cluster. Specify the DNS, GUID or path of the server while executing this command. At least one server in the zone must have a ZooKeeper role enabled on it. For example: `zman srvrs --servers=server1.microfocus.com`.
- **Viewing list of ZooKeeper servers:** Execute the command `server-role-zookeeper-list-cluster (zman srzlc)` to view the list of ZooKeeper servers in a cluster.

Maintaining the Kafka Cluster

You can modify a Kafka cluster by adding or removing brokers from the existing cluster. You can also modify the parameters of the Kafka cluster. The ZMAN commands that can be executed to manage the Kafka cluster are:

- **Removing brokers from the Kafka cluster:** Execute the command `server-role-kafka-remove-broker (zman srkrb)`. This command lets you remove Kafka brokers from an existing cluster. Specify the DNS, GUID or path of the server while executing this command.
- **Adding Kafka Brokers:** Execute the command `server-role-kafka-add-broker` (`zman srkab`). For more information, see Enabling ZooKeeper, Kafka, and Vertica on multiple servers.
- **Re-configuring Kafka broker:** Execute the command `server-role-reconfig-broker (zman srkrcb)`. This command re-configures the Kafka broker. Specify the DNS, GUID or path of the server while executing this command. When a broker is added to Kafka cluster, the remaining servers are re-configured by ZENworks. However, if re-configuration fails for a specific server, then run this command for that server. This command is also used for Disaster Recovery scenarios. For more information, see ZENworks Disaster Recovery Reference.
- **Updating Kafka cluster parameters:** Execute the command `server-role-kafka-update-cluster (zman srkuc)`. This command lets you modify the existing cluster parameters. For more information on each of the parameters that can be modified, see the parameters in the command `server-role-kafka-configure-cluster` in Kafka.
- **Viewing list of Kafka brokers:** Execute the command `server-role-kafka-list-cluster (zman srklc)` to view the list of Kafka brokers.

Maintaining Kafka Connect

When Kafka is enabled, Kafka connectors are created that is associated with each table in the RDBMS database. If any of these connectors are not running, then you might have to restart these connectors or re-configure the connectors.

To view the status of these connectors, navigate to the Diagnostic page in ZCC and view the status in the Data Sync Status section.

- **Restart Kafka Connectors:** Execute the command `server-role-kafka-restart-connectors (zman srkrcn)`. Specify a comma separated list of connectors. If any of the Kafka connectors as displayed in the Diagnostics page in ZCC, is not running, then you can try restarting the connectors by executing this command. For example: `zman srkrcn -c=zenconnectorzBundlefolderrights, zenconnectorzbundlerights`.
- **Reconfigure Kafka Connectors:** Execute the command `server-role-kafka-reconfigure-connectors (zman srkccn)` to reconfigure connectors based on the properties mentioned in the connector-configs.xml. This file is available at `etc/opt/novell/zenworks`. This.xml file lets you modify properties such as the interval at which the Kafka connector identifies changes in the RDBMS tables or the interval at which updated data is published to Kafka. Specify a comma separated list of connectors. For example: `zman srkrcn -c=zenconnectorzBundlefolderrights, zenconnectorzbundlerights`.
- **Re-create Kafka Connectors:** Execute the command `server-role-kafka-recreate-connectors (zman srkrcc)` to re-create connectors. Execute this command if you have migrated your primary database to another RDBMS database. For example: `zman srkrcc -f`.
- **Retrieve Kafka configuration details:** Execute the command `server-role-kafka-get-connector-config (zman srkgcc)`. For example: `zman srkgcc -c=zenconnectorzBundlefolderrights`.

- **Retrieve list of Kafka connectors:** Execute the command `zman server-role-kafka-list-connectors (zman srklcn)`.

Maintaining the Vertica Cluster

You can modify a Vertica cluster by adding or removing nodes from the existing cluster. You can also modify the K-safety factor value of an existing Vertica cluster. The ZMAN commands to be executed to modify the Vertica cluster are:

IMPORTANT: If you modify the Vertica cluster, ensure that you restart the ZENserver and ZENloader services on all the Primary Servers.

- **Adding Servers:** For more information, see Enabling ZooKeeper, Kafka, and Vertica on multiple servers.
- **Removing a server:** Execute the command `server-role-vertica-remove server (zman srvrs` to remove a server from the existing cluster. Specify the DNS, GUID or path of the server while executing this command. You can only remove one server at a time. Ensure that the K-safety factor remains effective after removing a server from the cluster. If you are decommissioning a server from the existing cluster, then before removing the server, it is recommended that you add another server to the cluster.
- **Updating the Vertica cluster:** Execute the command `server-role-vertica-update-cluster (zman srvuc)` to update the K-safety factor in the existing cluster.
- **Preparing the Vertica server for restore process:** Execute the command `server-role-vertica-prepare-server (zman srvps)` to prepare the Vertica server during the backup and restore process. For more information, see Replacing an Existing Primary Server with a New Primary Server.
- **Viewing list of Vertica servers:** Execute the command `server-role-vertica-list-cluster (zman srvlc)` to view the list of Vertica servers in a cluster.

Monitoring ZooKeeper, Kafka and Vertica

To monitor the overall health of the Vertica, ZooKeeper and Kafka clusters, you need to navigate to the Diagnostics page in ZCC. You can also view the status of the Data Sync process on this page. To navigate to the diagnostics page in ZCC, click **Diagnostics** in the left pane of ZCC.

For more information on debugging issues that are identified by the Diagnostics feature in ZCC, see Monitoring ZooKeeper, Kafka and Vertica.

NOTE: If client authentication is not enabled for external CA certificates, then the Data Sync Status and Kafka Cluster status will display correct values only if the Diagnostics page is accessed from a server in which the Kafka role is enabled.

The topics addressed in this section are:

Vertica Cluster

The statuses displayed are as follows:

 **Up**: indicates that all servers are up.

 **Down:** if at least one server is down in the zone, then based on the K factor the relevant status is displayed. For example, the status is displayed as down if at least one server is down and K factor is 0.

 **Warning:** if at least one server is down in the zone, then based on the K factor the relevant status is displayed. For example, the status is displayed as warning if at least one server is down and K factor is 1.

Zookeeper Cluster

The statuses displayed are as follows:

 **Up**: indicates that majority of servers in the cluster are up.

 **Down**: indicates that majority of servers in the cluster are down.

 **Warning:** if one of the servers is down.

Kafka Cluster

The statuses displayed are as follows:

 **Up**: indicates that all servers are up.

 **Down:** if at least one server is down in the zone, then based on the replication count the relevant status is displayed. For example, the status is displayed as down if at least one server is down in a three node cluster with replication count being 1.

 **Warning:** if at least one server is down in the zone, then based on the replication factor the relevant status is displayed. For example, the status is displayed as warning if at least one server is down in a three node cluster with replication factor being 2.

For the Kafka Cluster, the status of the Kafka Brokers, Kafka Connect and the Schema registry are also displayed.

Data Sync Status

The Data Sync Status panel enables you to identify whether the data sync process is running seamlessly or not.This panel displays the status of each Kafka connector that is responsible for syncing data from the RDBMS to Vertica. These connectors help in identifying the rows in the database tables that have been updated and publishes this data to Kafka. The ZENworks loader stream processor services then consumes this data from Kafka and sends it to Vertica.

Along with the connector name and its status, this panel displays the name of the ZENloader stream processors (consumer name) and its status and the names of the server on which the loader services are running. It displays the time at which the ZENloader stream processors last consumed data from Kafka and the time at which this data was last sent to Vertica. It also displays the number of pending records that are yet to be migrated from each Kafka connector to Vertica.

Debugging Issues

The topics addressed in this section are:

NOTE: If due to firewall restrictions, you are unable to access the Vertica, Kafka, or ZooKeeper services on another server, then to open the ports on that server, run the configure action `novell-zenworks-configure -c ClusterFirewallConfigureAction -Doperation=add/remove -Dservice=zookeeper/kafka/vertica`. For more information, see "Troubleshooting" on page 27

ZooKeeper

Configuration Issues

For any configuration related issue you can first refer to either one of the logs:

- `/var/opt/novell/log/zenworks/loader-messages.log`
- `/var/opt/novell/log/zenworks/zman.log`
- `%ZENWORKS_HOME%\logs\loader-messages.log`
- `%ZENWORKS_HOME%\logs\zman.log`

However, if information in these logs is insufficient, then it is recommended that you enable the ZooKeeper logs by uncommenting the lines below the comment `#zookeeper` in the `log4j.properties` file present in `/opt/novell/zenworks/share/tomcat` and then restarting the ZENworks server. You can then view the `zookeeper-messages.log` in the path `/var/opt/novell/log/zenworks/`.

General Debugging

If at any point in time, the ZooKeeper cluster is down, then check whether the port 6789 of the ZooKeeper server is not being used by any other component.

Kafka

Configuration Issues

For any configuration related issue you can refer to either one of the logs:

- `/var/opt/novell/log/zenworks/loader-messages.log`
- `/var/opt/novell/log/zenworks/zman.log`

General Debugging

If the status of **Kafka broker** in the Diagnostics page is **Not Running**, then check whether the port 9093 of the server is not being used by any other component. You can also check the Kafka logs within the server.logs file available at `/var/opt/novell/log/zenworks/kafka-logs`.

If the status of **Schema Registry** in the Diagnostics page is **Not Running,** then check schema-registry.log available at `/var/opt/novell/log/zenworks/kafka-logs`.

If the status of **Kafka Connect** in the Diagnostics page is **Not Running**, then check whether the port 8083 of the server is not being used by any other component. You can also check the kafka-connect.log available at `/var/opt/novell/log/zenworks/kafka-logs`.

To open the ports to enable communication with the Kafka service, you might have to run the configure action novell-zenworks-configure –c ClusterFirewallConfigureAction –Doperation=add/remove –Dservice=zookeeper/kafka/vertica

If due to some reason any of the Kafka connectors displayed in the **Data Sync** section of the **Diagnostics** page, have failed, then you need to reconfigure the Kafka connector and restart the Kafka-connect service. To reconfigure the connector run the following command:

```
zman server-role-kafka-reconfigure-connectors -c <name of the connector>
```

You can also retrieve the names of all the connectors in the command line utility by running the `zman server-role-kafka-list-connectors` command. After reconfiguring the Kafka connector, restart the Kafka-connect service by running the command `systemctl restart zenworks-connect.service`.

Also, if the RDBMS is down for more than an hour, the Kafka connectors will not be able to sync data between Vertica and the RDBMS. You need to restart the Kafka.connect service by running the command `systemctl restart zenworks-connect.service`.

Vertica

Configuration Issues

For failures related to Vertica configuration, you can check the following logs:

Installation Failure

Refer to the following logs:

- Check **vertica-install.log** within `/var/opt/novell/log/zenworks/`.
- Check **vertica-admin-tools.log** within `/var/opt/novell/log/zenworks`.
- Check **zman.log** within `/var/opt/novell/log/zenworks/`.

Vertica Management Console

As the Diagnostics page in ZCC provides only an overview of the status of the Vertica database, to get an in-depth view of the Vertica database operations, you can use the Vertica management console, which is a third-party tool. This tool can be installed and accessed from any server.

For more information on installing the Vertica management console, see the Vertica Documentation.

After logging in to the Management console, you need to select the **Import an existing database/cluster** option. Using this option, you can import multiple Vertica database servers. While importing the database, you need to provide the database credentials, which can be obtained in one of the following ways:

- Execute the zman command `server-role-vertica-get-credentials (zman srvgc).`
- Obtain the credentials from the `zenworks-coreadmin` web service by referring to the following steps:
  1. On your browser, type the server address of the Primary Server that is part of the Vertica cluster, along with `zenworks-coreadmin`. For example: `https://10.0.0.0/zenworks-coreadmin.`
  2. Select **Test Service**.
  3. Select **Vertica Configuration** and click **Invoke**.

For more information on using the Vertica management console, see the Vertica Documentation.

Backing up and Restoring the Vertica Database

As a part of maintenance activities, taking regular backups of the Vertica database is recommended. Vertica provides the vbr utility that enables you to perform a full backup and restore operation of the Vertica database.

This utility enables you to perform the following operations:

- Back up the database
- Back up specific objects like schemas or tables, in a database.
- Restore a database or individual objects from a backup.
- Copy a database to another cluster. For example, to promote a test cluster to production (for Enterprise mode only). For more information, see Vertica Documentation.
- Replicate individual objects such as schemas and tables to another cluster (for Enterprise mode only). For more information, see Vertica Documentation.
- List available backups.

The topics addressed in this section are:

- “Creating Configuration File for Vertica Full Local Backup” on page 22
- “Creating a Full Local Backup” on page 22
- “Restoring from a backup” on page 23

Creating Configuration File for Vertica Full Local Backup

To run the vbr utility, you need to specify a configuration file that contains information such as the backup location, name of the database and dbuser, the snapshot name, db administrator password file path, and retry count. Vertica provides a few sample configuration files available at `/opt/vertica/share/vbr/example_configs/backup_restore_full_local.ini` that can be used for various vbr tasks.

If the `dbPromptForPassword` parameter in the configuration file is true, then the vbr utility will prompt for a password every time you run the utility. However, to avoid user intervention while performing the backup operation, you can specify the location of the db administrator password config file in the `passwordFile` parameter. For example, `passwordFile=/path/to/vbr/pw.txt`. This file should be read protected by the db administrator user only. The format of the content in the pw.txt file should be:

[Passwords]

dbPassword=****

For more information on creating the configuration file, see the Vertica Documentation.

Creating a Full Local Backup

Prerequisites

- Ensure that you initialize the backup location by running the following command `vbr -t init -c full-backup.ini`.
- Your database should be up and running. All nodes need not be in a K-safe database. However, the nodes that are down are not backed up.
- All of the backup hosts should be up and running.
- Enough disk space should be available to store the backups.
- The user account of the user who starts the vbr utility should have write access to the target directories on the host backup location. This user can be dbadmin or another assigned role. However, you cannot run the vbr utility as root.
- Each backup has a unique file name. If you want to keep earlier backups, ensure that the `restorePointLimit` parameter is set to a number greater than 1 in the configuration file

Procedure

Run the vbr utility from a terminal. Use the database administrator account from an initiator node in the database cluster.

Run the following command and a backup will be created without any further intervention:

```
vbr -t backup -c full-backup.ini
```

Restoring from a backup

Prerequisites

Before performing a full restore from the backed up database, ensure that the following requirements are met:

- The database is down. You cannot perform the restore action when the database is running.
- All the backup hosts are available.
- The backup directory exists and contains the backups from which to restore the data.
- The cluster to which you are restoring the backup should have:
  - The same number of nodes as the one used to create the backup.
  - The same architecture (Enterprise or Eon mode) as the one used to create the backup.
  - Identical node names.

Procedure

To restore from the most recent backup, run the following command:

```
vbr -t restore -c full-backup.ini
```

To restore an archived backup, add the `--archive` parameter to the command line along with the directory name, which is the `date_timestamp` that identifies the archive to restore. For example, `vbr -t restore -c full-backup.ini --archive=20190712_085620`. This will restore from the archive created on 12th July 2019 and 8:56:20.

Disaster recovery for Vertica

ZENworks does not recommend maintaining a replica of the Vertica database to protect data in case of a failure in the zone. In the event of a failure of the Primary Server in which Vertica is enabled, you need to replace the device hosting the Primary Server with a new device that has the same identity, that is the same host name and IP address as that of the old device. For more information, see Replacing an Existing Primary Server with a New Primary Server in the ZENworks Disaster Recovery Reference.

After replacing the existing Primary Server with a new Primary Server, you need to re-execute the configure action to migrate data from the RDBMS to the Vertica database. For more information, see “Migrating Data to Vertica” on page 10. However, data like the patch CVE status data might not be available immediately unless you run patch scans for all the devices in the zone.

NOTE: In the event of a failure, any trending data that is created on Vertica cannot be recovered. However, Vertica provides a way to take a backup of specific tables in the database and enables you to import these onto another database. For more information, see Vertica Documentation. Therefore, as a precautionary measure, you can take periodic backup of these tables. After replacing the existing Primary Server with the new server, you can replicate and restore these tables from the backup location.

Shutting Down a Vertica Node

If you need to shut down a Vertica node for maintenance, then ensure that you refer to the following steps to ensure that there is no loss in data:

1. Login to the Vertica database using the dbadmin user and run the following commands:
   a. `select close_all_sessions();` For more information, see the Vertica Documentation.
   b. `select make_ahm_now();`
   c. `select do_tm_task('moveout');`
   d. `select do_tm_task('mergeout');`
   e. `select make_ahm_now();`
2. Stop the node using admintools command: `admintools -t stop_node -s` <HOSTS> where -s <HOSTS> is a comma separated list of hosts on which the Vertica process needs to be stopped.

Disabling Vertica and Reverting to RDBMS

ZENworks does not provide an option to automatically disable Vertica through ZENworks Control Center. If you want to disable Vertica and revert to the existing RDBMS, then refer to the following steps to manually disable the Vertica database. By following the procedure detailed in this section, Vertica will be disabled but will still be retained in the zone.

NOTE: When you revert to the RDBMS all trending data such as data related to CVE and Patch trends will be deleted.

---

1. Run the following query to set the value for the field `vertica.available` to false, in the `zOpaqueData` table in the database:

   ```
   update zopaquedata set data='false' where name = 'vertica.available'
   ```

   Subsequently, the Dashboard feature will start obtaining data from the RDBMS.
2. Stop and disable the following services:
   - **zenworks-connect.service:** `systemctl stop zenworks-connect.service` and `systemctl disable zenworks-connect.service`.
   - **zenworks-kafka.service:** `systemctl stop zenworks-kafka.service` and `systemctl disable zenworks-kafka.service`.
   - **zenworks-schema-registry.service:** `systemctl stop zenworks-schema-registry.service` and `systemctl disable zenworks-schema-registry.service`.
3. Execute the following commands:
   - `/opt/vertica/bin/admintools -t set_restart_policy -d zenworks -p 'never'`. Login as dbadmin to execute this command that sets the Restart policy of Vertica to never.
   - `/opt/vertica/bin/admintools -t stop_db -d zenworks -p <dbadmin_password> --force`. Execute this command to stop Vertica on the server. To obtain the `dbadmin` password, execute the zman command `server-role-vertica-get-credentials (zman srvgc)`.

4. Stop and disable Vertica services by executing the following commands:
   - `systemctl stop zenworks-vertica.service`
   - `systemctl disable zenworks-vertica.service`
5. Delete the file `/vastorage/etc/opt/novell/zenworks/loader/loader.cluster.id` and restart ZENloader services on the Vertica server.
6. Restart ZENloader and ZENserver services on all Primary Servers in the zone.

NOTE: Certain queue actions that run periodically will fail when Vertica is disabled. However, this will not have a functional impact on any of the ZENworks features.

If you want to enable Vertica again, then you need to execute the following steps:

Re-enabling Vertica

If after reverting to the RDBMS, you want to enable Vertica again, then execute the following steps:

1. Enable and start Vertica services by executing the following commands:
   - `systemctl enable zenworks-vertica.service`
   - `systemctl start zenworks-vertica.service`
2. Execute the following commands:
   - `/opt/vertica/bin/admintools -t set_restart_policy -d zenworks -p 'always'`. Login as dbadmin to execute this command that sets the Restart policy of Vertica to *always*.
   - `/opt/vertica/bin/admintools -t start_db -d zenworks -p <dbadmin_password>`. Execute this command to start Vertica on the server. To obtain the `dbadmin` password, execute the zman command `server-role-vertica-get-credentials (zman srvgc)`.
3. Enable and start the following services:
   - **zenworks-connect.service:** `systemctl enable zenworks-connect.service`

     and `systemctl start zenworks-connect.service`.
   - **zenworks-kafka.service:** `systemctl enable zenworks-kafka.service`

     and `systemctl start zenworks-kafka.service`.
   - **zenworks-schema-registry.service:** `systemctl enable zenworks-schema-registry.service`.

     and `systemctl start zenworks-schema-registry.service`.
4. Run the following query to set the value for the field `vertica.available` to true, in the `zOpaqueData` table in the database:

   ```
   update zopaquedata set data='true' where name = 'vertica.available'
   ```

   Subsequently, the Dashboard feature will start obtaining data from Vertica.
5. Create the file `/vastorage/etc/opt/novell/zenworks/loader/loader.cluster.id` on the Vertica server, with the content as `-hostname`, where hostname= hostname of the Vertica server.
6. Restart ZENloader and ZENserver services on all Primary Servers in the zone.

Disabling and Removing Vertica from the Zone

The previous section details the procedure to only disable Vertica in the zone. However, if you want to completely remove Vertica from the zone, then refer to the following section:

1. Run the zman command `server-role-vertica-get-credentials (zman srvgc)` on any server to obtain the Vertica database superuser password.
2. Using the obtained dbadmin password, run the following command to stop the Vertica database on one of the servers on which Vertica is installed:

   `/opt/vertica/bin/admintools -t stop_db -d zenworks -p <dbadmin_password> --force`.
3. Run the following drop database command on one of the servers on which Vertica is installed:

   `/opt/vertica/bin/admintools -t drop_db -d zenworks`.
4. Stop all the cluster nodes. To identify these nodes, run the following command on one of the servers on which Vertica is installed:

   `/opt/vertica/bin/admintools -t list_host`.

   If a node is listed, then run the following command to stop the node:

   `/opt/vertica/bin/admintools -t stop_node -s <comma separated list of hosts>`.
5. Disable the following services on the respective Vertica and Kafka servers.

   zenworks-connect.service: `systemctl stop zenworks-connect.service`

   and `systemctl disable zenworks-connect.service`.

   zenworks-kafka.service: `systemctl stop zenworks-kafka.service`

   and `systemctl disable zenworks-kafka.service`.

   zenworks-schema-registry.service: `systemctl stop zenworks-schema-registry.service`

   and `systemctl disable zenworks-schema-registry.service`.

   zenworks-vertica.service: `systemctl stop zenworks-vertica.service` and

   `systemctl disable zenworks-vertica.service`.
6. Delete the following datamodel files on all the servers.

   On a Linux server: Navigate to `/etc/opt/novell/zenworks/datamodel` and delete the files `zenvertica_dmaccounts.properties` and `zenvertica.xml`. Also, remove the entry `vertica=zenvertica` from the `dmmappings.properties` file.

   On a Windows server: Navigate to `C:\Program Files (x86)\Novell\ZENworks\conf\datamodel` and delete the files `zenvertica_dmaccounts.properties` and `zenvertica.xml`. Also, remove the entry `vertica=zenvertica` from the `dmmappings.properties` file.
7. Run the following query to remove the opaque entries related to Vertica from the database:

   `DELETE FROM ZOPAQUEDATA WHERE name LIKE '%vertica%'`
8. Run the following query to remove the opaque entries related to Kafka from the database:

   `DELETE FROM ZOPAQUEDATA WHERE name LIKE '%kafka%'`
9. Remove the following cluster ID file from all the Vertica nodes to stop the stream processors:

   `/etc/opt/novell/zenworks/loader/loader.cluster.id`.
10. Remove the Vertica role from the database:

```
DELETE FROM ZZENSERVERROLES roles WHERE ROLES = 'Vertica'
```

11 Remove the Kafka role from the database:

```
DELETE FROM ZZENSERVERROLES roles WHERE ROLES = 'Kafka'
```

12 Remove the Kafka connector information from the database:

```
DELETE FROM ZVCONSUMERMETRICS;
```

13 Restart the ZENserver and ZENloader services on all the servers.

Re-enabling Vertica

If after removing Vertica from the zone, you want to re-enable it, then refer to the following steps:

1. Procure the obfuscated license file from Micro Focus Customer Support and copy the file in the location `/etc/opt/novell/vertica`.
2. Enable Kafka. For more information, see Enabling Kafka.
3. Enable Vertica. For more information, see Enabling Vertica.
4. Execute the data migration configure action using the Force option `novell-zenworks-configure -c VerticaDBMigrate -Doption=force`.

Troubleshooting

This section provides solutions to problems that you might face with the Vertica database.

After partially disabling Vertica in the zone, upgrade to the next ZENworks release version fails

Explanation: Consider a scenario, where Vertica is configured on a Primary Server, and then needs to be disabled and removed from the zone. While disabling Vertica, a few errors occurred, due to which Vertica was only partially disabled. Subsequently, upgrade to the next ZENworks release version fails.

Action: Before proceeding with the Upgrade process, ensure that you have fully configured or fully disabled Vertica. While disabling Vertica in your zone, ensure that all the steps documented in are followed and no errors occur while executing the commands.

However, if you are still unable to disable Vertica completely from the zone, then run the following query:

```
delete from preglobalupgradelog where schematype='vertica'
```

A node in the Vertica cluster shuts down

Explanation: To provide fault tolerance, Vertica can be run as a cluster of nodes. However, at times the nodes in the cluster can abruptly shut down due to issues related to network connectivity and so on.

Action: Follow the steps documented in the Vertica documentation, to bring up the database when one of the nodes shuts down.

When nodes responsible for bringing a Vertica cluster down are restarted, then the remaining nodes are no longer a part of the cluster

Explanation: The K-factor specified while configuring Vertica, sets the fault tolerance for the Vertica cluster, where K can be set as 0 or 1. The Vertica cluster will be down, if more than 'K' number of nodes are down. For example, if the K factor is 1 in a 3 node cluster and 2 nodes are down, then the Vertica cluster will be down. When you restart services on these 2 nodes, the third node (that was up and running) will no longer be a part of the cluster.

Action: The command to restart the Vertica servers should be run for all the servers in the Vertica cluster. For more information on how to restart the servers in the Vertica cluster, see the Vertica documentation.

At least one Kafka connector has failed

Explanation: If for any reason any of the Kafka connectors have failed, which is displayed in the **Data Sync** panel of the Diagnostics page in ZCC, then the failed connectors will not be able to sync data from their associated RDBMS tables to Vertica.

Action: Run the command `zman server-role-kafka-reconfigure-connectors -c <name of the connector>` to reconfigure the failed Kafka connector.

You can retrieve the names of the all connectors within the command line utility by running the `zman server-role-kafka-list-connectors` command.

After reconfiguring the Kafka connectors, you need to restart the Kafka-connect service by running the command `systemctl restart zenworks-connect.service`.

Appendix 1: Mounting a New Hard Disk for Vertica Catalog

When one or more nodes are running on low disk space or have a failed disk, then beyond a certain threshold, Vertica starts rejecting transactions that update the catalog or data files. In Vertica, catalog is a set of files that contains information (metadata) about the objects in a database, such as the nodes, tables, constraints, and projections. The catalog files are replicated on all nodes in a cluster. To avoid low disk space or failed disk related issues, you can mount a new hard disk and store the catalog files on it.

Refer to the following steps to mount a new hard disk. You can ignore these steps if you have already mounted a disk.

1. Obtain the name of the new hard disk. You can use the command `fdisk -l`.
2. Format the new hard disk. For example, if the name of the new hard disk is `sdc`, then run the command `mkfs.ext4 /dev/sdc`.
3. Create a new directory and mount the hard disk to it. For example, if the new directory is `/vastorage/home/dbadmin/zenworks/NewDisk` then use the commands `mkdir -p /vastorage/home/dbadmin/zenworks/NewDisk` and `mount /dev/sdc /vastorage/home/dbadmin/zenworks/NewDisk`.

Refer to the following steps to move the existing data to the new directory location and make it the primary storage location for Vertica. The location `/vastorage/home/dbadmin/zenworks/NewDisk` has been used as an example in the following steps. Replace this location with the location where the new hard disk has been mounted:

1. Create a new directory within the mounted location and give permissions to it:

```
mkdir -p /vastorage/home/dbadmin/zenworks/NewDisk/v_zenworks_node0004_data/ &&
chown -R dbadmin:verticadba  /vastorage/home/dbadmin/zenworks/NewDisk/
v_zenworks_node0004_data/
```

NOTE: In the above command, `/vastorage/home/dbadmin/zenworks/NewDisk/` is the directory location where the hard disk is mounted and `v_zenworks_node0004_data` is the new directory.

---

2. Create a new db location by running the following SQL query in Vertica using any connection tool and specify the node to which the data should be moved.

   ```
   CREATE LOCATION '/vastorage/home/dbadmin/zenworks/NewDisk/
   v_zenworks_node0004_data/' NODE 'v_zenworks_node0001' USAGE 'DATA,TEMP' LABEL
   'T0_DATA_TEMP';
   ```

   NOTE: If you have more than one node, run this query for all the nodes. For example: `CREATE LOCATION '/vastorage/home/dbadmin/zenworks/NewDisk/v_zenworks_node0005_data/' NODE 'v_zenworks_node0002 USAGE 'DATA,TEMP' LABEL 'T0_DATA_TEMP';`, indicates that directory `/vastorage/home/dbadmin/zenworks/NewDisk/v_zenworks_node0005_data/'` is mounted in node 2.

   ---

3. Set object storage policy at db level by running the following query:

   ```
   select set_object_storage_policy('zenworks','T0_DATA_TEMP', true );
   ```

4. Apply the storage policy by running the following query:

   ```
   select enforce_object_storage_policy('zenworks');
   ```

5. Ensure that the new storage location is created successfully, by running the following queries:

   To check the storage location: `select * from storage_locations`

   To check the storage label: `select node_name,location_label from storage_containers where projection_id in (select projection_id from projections);`

   To check whether the new storage location is created or not (the older location will also be displayed till it is retired): `select is_retired,location_path from storage_locations;`

6. Retire the older location by running the following query:

   `SELECT RETIRE_LOCATION('/vastorage/home/dbadmin/zenworks/ v_zenworks_node0001_data/' , 'v_zenworks_node0001', true);` If you have more than one node, then repeat this for all the older locations of the nodes.

   To verify whether the location is retired, check if retire is marked True by running the following query:

   ```
   select is_retired,location_path from storage_locations;
   ```

   The retired location will be displayed as True till the location is deleted.

7. Delete the retired storage location by running the following query:

   `SELECT DROP_LOCATION('/vastorage/home/dbadmin/zenworks/ v_zenworks_node0001_data/' , 'v_zenworks_node0001');.`

   To check whether the location is deleted, run the following query:

   ```
   select is_retired,location_path from storage_locations;
   ```

After completing this procedure, it is recommended that you restart the database.

Glossary

This section provides a description of some of the terminologies that you will come across in this document.

- **Kafka Cluster:** A group of servers or nodes that are connected to each other to achieve a common objective is a cluster. Each of the servers or nodes in the cluster, will have one instance of Kafka broker running.
- **Kafka Broker:** One or more servers that are added in a Kafka cluster are called Brokers. These are typically message brokers that act as mediators between two systems, ensuring that messages are delivered to the correct systems.
- **Kafka Topics:** Kafka stores streams of records in categories called Topics. These topics are further broken down into partitions that can be stored across multiple servers and disks. Topics in Kafka are always multi-subscriber; that is, a topic can have zero, one, or many consumers that subscribe to the data written to it.
- **Kafka Partitions:** Kafka topics are further broken down into partitions that enables topics to be replicated across multiple brokers for fault tolerance.
- **Kafka Producers:** The processes that publish messages to one or more Kafka topics are called Producers. In this case, the producer is the RDBMS system.
- **Kafka Consumers:** The processes that consumes messages from the Kafka topics are called consumers. In this case, the consumer is Vertica.
- **Kafka Replicas:** A replica or a copy of a partition is essentially used to prevent loss of data.
- **Kafka Leader:** A node that handles all read and write requests for a partition is called a leader.
- **Kafka Follower:** A node that replicates the leader are called followers. If the leader fails, one of the followers will automatically become a leader.
- **Kafka Connect:** Kafka Connect is a tool included with Kafka that imports and exports data to Kafka. It is an extensible tool that runs connectors, which implement the custom logic for interacting with an external system. For example, a connector to an RDBMS system captures every change in data made in a table.
- **Kafka Schema Registry:** Kafka producers write data to Kafka topics and Kafka consumers read data from Kafka topics. Schema Registry helps ensure that producers write data with a schema that can be read by consumers, even as producers and consumers evolve their schemas.
- **Heartbeat Requests:** ZooKeeper sends heartbeat requests to determine whether the Kafka broker is alive.
- **Tick Time:** The unit of time used by ZooKeeper that is translated to milliseconds. All time-dependent operations for ZooKeeper are based on tick time.
- **Offset:** A Kafka topic receives messages across a set of partitions and each partition maintains these messages in a sequential order, which is identified as an offset, also known as a position.
- **K Factor:** Specify the number of replicas of the data that should exist in the database. The K safety sets the fault tolerance for the Vertica cluster, where K can be set as 0 or 1. Specify a value based on the number of nodes in the cluster, which is measured as 2k+1. For example: the K factor 0 is ideal for a single node cluster. If the K factor is 1, then you need to have a minimum of 3 nodes. Vertica recommends a K-safety value of 1.

  K-factor also determines the number of identical instance of the segmented projections to be created for fail safety. Vertica stores table data in projections that are a collection of table columns, which optimizes query execution. The projection data can be divided into multiple segments or can be maintained as a single unsegmented unit. To ensure high availability of data, unsegmented projections are replicated on all the nodes and a segmented projection should have K+1 identical instances (buddies). For example: if the K- factor is specified as 1, then 2 identical instances of the segment are maintained.

Legal Notice