Micro Focus Security
ArcSight Database

Software Version: 9.2.1-16

Deployment Guide

Document Release Date: August, 2020

Software Release Date: August, 2020

Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14
1QN UK

https://www.microfocus.com

Copyright Notice

Confidential computer software. Valid license from Micro Focus required for possession, use or copying. The information contained herein is subject to change without notice.

The only warranties for Micro Focus products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein.

No portion of this product's documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's internal use, without the express written permission of Micro Focus.

Notwithstanding anything to the contrary in your license agreement for Micro Focus ArcSight software, you may reverse engineer and modify certain open source components of the software in accordance with the license terms for those particular components. See below for the applicable terms.

U.S. Governmental Rights. For purposes of your license to Micro Focus ArcSight software, “ commercial computer software” is defined at FAR 2.101. If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation and other technical data subject to the terms of the Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulation (“ FAR”) and its successors. If acquired by or on behalf of any agency within the Department of Defense (“ DOD”), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of the Agreement as specified in 48 C.F.R. 227.7202-3 of the DOD FAR Supplement (“ DFARS”) and its successors. This U.S. Government Rights Section 18.11 is in lieu of, and supersedes, any other FAR, DFARS, or other clause or provision that addresses government rights in computer software or technical data.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft

Corporation. UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated

- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document,

go to: ArcSight Product Documentation on the Micro Focus Security Community

Support

Contact Information

Phone	A list of phone numbers is available on the Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs

Contents

Installing the Database

This section provides information about configuring the Database server and installing the database.

> **Note:** Before you install the database, make sure to estimate the storage needed for the incoming EPS (event per second) and event size, and also to evaluate the retention policy accordingly.

Configuring the Database Server

The server configuration is based on an HPE ProLiant DL380 Gen9 server with 48 cores and 128 GB memory.

> **Note:** The configuration settings for the server described in this section assume the hardware is HPE ProLiant DL380 Gen9 server with 48 cores and 128 GB memory. If you're not using this type of hardware, adjusting the configuration settings may result in better performance.

To avoid performance issues with large workloads, the Database server should be a dedicated server.

> **Note:** Database data should be backed-up routinely. For more information, please see "Backing Up and Restoring the Database" on page 36.

To configure the Database server:

1. Provision the server running on any of the following operating systems with at least 2 GB of swap space:

   - CentOS - Versions 6.x or 7.x
   - RHEL - Versions 6.x or 7.x

   > **Note:** In case pre-check on swap space fails after provisioned 2 GB on swap, provision swap with 2.2 GB should solve the problem.

2. Add the following parameters to `/etc/sysctl.conf`. You must reboot the server for the changes to take effect.

Parameter	Description
`net.core.somaxconn = 1024`	Increases the number of incoming connections

`net.core.wmem_max = 16777216`	Sets the send socket buffer maximum size in bytes
`net.core.rmem_max = 16777216`	Sets the receive socket buffer maximum size in bytes
`net.core.wmem_default = 262144`	Sets the receive socket buffer default size in bytes
`net.core.rmem_default = 262144`	Controls the default size of receive buffers used by sockets

`net.core.netdev_max_backlog = 100000`	Increase the length of the network interface input queue
`net.ipv4.tcp_mem = 16777216 16777216 16777216`	
`net.ipv4.tcp_wmem = 8192 262144 8388608`	
`net.ipv4.tcp_rmem = 8192 262144 8388608`	
`net.ipv4.udp_mem = 16777216 16777216 16777216`	
`net.ipv4.udp_rmem_min = 16384`	
`net.ipv4.udp_wmem_min = 16384`	
`vm.swappiness = 1`	Defines the amount and frequency at which the kernel copies RAM contents to a swap space For more information, see Check for Swappiness.

3. Add the following parameters to `/etc/rc.local`. You must reboot the server for the changes to take effect.

> **Note:** The following commands assume that sdb is the data drive (i.e. /opt), and sda is the operating system/catalog drive.

Parameter	Description
`echo deadline > /sys/block/sdb/queue/scheduler`	Resolve FAIL (S0150)
`/sbin/blockdev --setra 4096 /dev/sdb`	Resolve FAIL (S0020) Vertica resides on `/dev/sdb`
`echo always > /sys/kernel/mm/transparent_hugepage/enabled`	
`cpupower frequency-set --governor performance`	Resolve WARN (S0140/S0141) (**CentOS only**)

4. To increase the process limit, add the following to `/etc/security/limits.d/20-nproc.con`:

```
* soft nproc 10240
* hard nproc 10240
* soft nofile 65536
* hard nofile 65536
* soft core unlimited
* hard core unlimited
```

5. In `/etc/default/grub`, append line `GRUB_CMDLINE_LINUX` with `intel_idle.max_cstate=0 processor.max_cstate=1`. For example:

   ```
   GRUB_CMDLINE_LINUX="vconsole.keymap=us crashkernel=auto
   vconsole.font=latarcyrheb-sun16 rhgb quiet intel_idle.max_cstate=0
   processor.max_cstate=1"
   ```

   ```
   grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

6. Use `iptables` to disable the firewall **WARN (N0010):**

   ```
   iptables -F
   iptables -t nat -F
   iptables -t mangle -F
   iptables -X
   systemctl mask firewalld
   systemctl disable firewalld
   systemctl stop firewalld
   ```

   For more information, see Firewall Considerations.

Port Availability

Database requires several ports to be open on the local network. It is not recommended to place a firewall between nodes (all nodes should be behind a firewall), but if you must use a firewall between nodes, ensure the following ports are available:

Port	Protocol	Service	Note
22	TCP	sshd	Required by Administration Tools and the Management Console Cluster Installation wizard.
5433	TCP	Database	Database client (vsql, ODBC, JDBC, etc) port.
5434	TCP	Database	Intra- and inter-cluster communication.
5433	UDP	Database	Database spread monitoring.
5438	TCP	Database Management Console	Used as Management Console-to-node and node-to-node (agent) communication port. This port replaced 5444 in the Single node installation.
5444	TCP	Database Management Console	MC-to-node and node-to-node (agent) communications port. See Changing MC or Agent Ports.
5450	TCP	Database Management Console	Port used to connect to MC from a web browser and allows communication from nodes to the MC application/web server.
4803	TCP	Spread	Client connections.

Port	Protocol	Service	Note
4803	UDP	Spread	Daemon to daemon connections.
4804	UDP	Spread	Daemon to daemon connections.
6543	UDP	Spread	Monitor to daemon connection.

7. Set SELinux to permissive mode:

   ```
   In /etc/selinux/config
   SELINUX=permissive
   ```

   For more information, see SELinux Configuration.

8. Configure the BIOS for maximum performance:

   System Configuration > **BIOS/Platform Configuration (RBSU)** > **Power Management** > **HPE Power Profile** > **Maximum Performance**

9. Reboot the system, and then use the `ulimit -a` command to verify that the limits were increased.

Enabling Password-less Communication

This section describes how to configure password-less communication from the node1 server to all of the node servers in the cluster.

Note: You must repeat the authentication process for all nodes in the cluster.

To configure password-less communication:

1 On the node1 server, run the `ssh-keygen` command:

```
ssh-keygen -q -t rsa
```

2 Copy the key from node1 to all of the nodes, including node1, using the node IP address:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@11.111.111.111
```

The system displays the key fingerprint and requests to authenticate with the node server.

3 Enter the required credentials for the node.

The operation is successful when the system displays the following message:

```
Number of key(s) added: 1
```

4 To verify successful key installation, run the following command from node1 to the target node to verify that node1 can successfully log in:

```
ssh root@11.111.111.111
```

Determine FIPS Configuration

Follow the steps in "Setting FIPS on Database Server " on page 34to enable or disable FIPS.

To Install Database

After you configured the Database server and enabled password-less SSH access, install the database.

1. On the Database cluster node1 server, create a folder forthe Database database installer script:

   ```
   mkdir $db_install-DIR
   ```

   Note: `$db_install-DIR` must not be under `/root`. Also, ensure that the folder name is not arcsight-database, else the installation will fail.

2. From the Download Installation Packages section, copy the database bits, `db-installer_ 3.2.0-4.tar.gz, to $db_install-DIR`

3. Extract the `.tar` file:

   ```
   cd $db_install-DIR
   tar xvfz db_installer_3.2.0-4.tar.gz
   ```

4. Edit the `config/db_user.properties` file. The `hosts` property isrequired.

Property	Description
`hosts`	A comma separated list of the Database database servers in IPv4 format (for example, 1.1.1.1, 1.1.1.2, 1.1.1.3). If it is necessary to construct the cluster, avoid using local loopback (localhost, 127.0.0.1, etc.).
`db_retention_day`	Used for the data retention policy.

5. Install Database:

   ```
   ./db_installer install
   ```

   When prompted, create the database administrator user, app admin user, and the search user. Database now supports multiple users:

   - **Database administrator:** Credentials required to access the database host to perform database related operations, i.e. setup, configuration, and debugging.
   - **App admin user:** A regular user with granted permissions (db, schema, resource pool). Credentials required when configuring Database from the CDF Management Portal for Interset.
   - **Search user:** A user designated for search operations. Credentials required when configuring Database from the CDF Management Portal. This is not applicable for Interset.
   - **Ingest user:** Should not be used or changed, this user is internally used for Database-scheduler,

i.e. ingestion.

For a list of options that you can specify when installing Database, see ./db_installer Options.

6. Database cluster status should be monitored constantly, for more information, please see "Monitoring the Database " on page 50

   - **Database nodes status:** Ensures all nodes are up
   - **Database nodes storage status:** Ensures storage is sufficient

Complete Database Setup

Follow the steps below to complete the Database Setup.

1. Login to the database node1 as root:

```
cd $db_install-DIR
```

2. Create the schema:

```
./db_installer create-schema
```

3. In order to create the Kafka scheduler, run the below commands:

   - If SSL is disabled:

   ```
   ./sched_ssl_setup --disable-ssl
   ```

   - If SSLis enabled, see "Database SSLRoot Certificate Support" on the next page.

4. Create the Kafka scheduler:

```
./kafka_scheduler create <Transformation_Hub_Node_1_IP>:9092
```

Note: Scheduler will obtain the Transformation Hub node information from kafka broker.

For a list of options that you can specify when installing the scheduler, see Kafka Scheduler Options.

5. Check the Database status:

```
./db_installer status
```

6. Check the scheduler status, event-copy progress, and messages:

```
./kafka_scheduler status
./kafka_scheduler events
./kafka_scheduler messages
```

./db_installer Options

To specify an option, type `./db_installer <Option_Name>`.

Option Name	Description
install	Installs the database
uninstall	Uninstalls the database and deletes data and users
create-schema	Creates the database schema for Interset
delete-schema	Deletes the Interset database schema
start-db	Starts the database with the `dba_password` specified in `db_credentials.properties`
stop-db	Stops the database
status	Prints the database cluster status

Kafka Scheduler Options

To specify an option, type `./kafka_scheduler <Option_Name>`.

Option Name	Description
update	Updates the scheduler
start	Starts the scheduler and begins copying data from all registered Kafka brokers
stop	Stops the scheduler and ends copying data from all registered Kafka brokers
delete	Deletes all registered Kafka instances from the scheduler
status	Prints the following information and log status for a running or stopped scheduler: - Current Kafka cluster assigned to the scheduler - Name and database host where the active scheduler is running - Name, database host, and process ID of every running scheduler (active or backup)
events	Prints event copy progress for the scheduler
messages	Prints scheduler messages

Configuring the Database with SSL

Certificate Creation:

Create a self-signed CA:

```
openssl req -newkey rsa:4096 -sha256 -keyform PEM -keyout ca.key -x509 \

-days 3650 -outform PEM -out ca.crt \

-subj "/C=US/ST=California/L=Santa Clara/O=Micro Focus/OU=Arcsight/\

CN=RootCA/emailAddress=admin@microfocus.com"   -nodes
```

Generate the Certificate for Vertica

1. Create the server key:

```
openssl genrsa -out vertica.key 4096 -nodes -sha256
Generating RSA private key, 4096 bit long modulus

..................................................................................................................++

.......................................................................

..............................................++  e is 65537

(0x10001)
```

2. Create Server certificate signing request:

```
openssl req -new -key vertica.key -out vertica.csr \
-subj "/C=US/ST=California/L=Santa Clara/O=Micro Focus/OU=Arcsight/\
CN=Vertica/emailAddress=admin@microfocus.com" -nodes -sha256
```

3. Sign the Certificate Signing Request with self-signed CA:

```
openssl x509 -req -in vertica.csr -CA ca.crt -CAkey ca.key \
-CAcreateserial -extensions server -days 3650 -outform PEM -sha256 \
-out vertica.crt
Signature ok

subject=/C=US/ST=California/L=Santa Clara/O=Micro
Focus/OU=Arcsight/CN=FQDN/emailAddress=admin@microfocus.com

Getting CA Private Key
```

Create the Vertica Scheduler Client Certificate

1. Create the certificate key for the Vertica scheduler:

```
openssl genrsa -out scheduler.key 4096
Generating RSA private key, 4096 bit long modulus

.........................++

.........................++

e is 65537 (0x10001)
```

2. Create the Vertica scheduler client certificate signing request:

```
openssl req -new -key scheduler.key -out scheduler.csr \
-subj "/C=US/ST=California/L=Santa Clara/O=Micro Focus/OU=Arcsight/\
CN=Scheduler/emailAddress=admin@microfocus.com" -nodes -sha256
```

3. Sign the certificate signing request:

```
openssl x509 -req -in scheduler.csr -CA ca.crt -CAkey ca.key \

-CAcreateserial -extensions client -days 3650 -outform PEM -sha256 \

-out scheduler.crt
```

Signature ok

subject=/C=US/ST=California/L=Santa Clara/O=Micro Focus/OU=Arcsight/CN=scheduler/emailAddress=admin@arcsight.com

Getting CA Private Key

Change the key files permissions

Run the following command:

```
chmod 600 ca.key vertica.key scheduler.key
```

Installing Self-Signed CA during the Transformation Hub Installation

1. Install the Transformation Hub. For more information see the Transformation Hub Deployment guide available from the Micro Focus Community.
2. Access the CDF UI
3. After infrastructure services have been deployed, copy the generated ca.crt and ca.key to the Transformation Hub server /tmp directory and Install the self-signed CA

   ```
   /opt/arcsight/kubernetes/scripts/cdf-updateRE.sh write \
   --re-key=/tmp/ca.key --re-crt=/tmp/ca.crt
   ```

   Dry run to check the certificate/key files.

   Success! Enabled the pki secrets engine at: RE_dryrun/

   Success! Data written to: RE_dryrun/config/ca

   Success! Disabled the secrets engine (if it existed) at:

   RE_dryrun/ Dry run succeeded.

   Submitting the certificate/key files to platform. CA for external communication

   will be replaced. Success! Disabled the secrets engine (if it existed) at: RE/

   Success! Enabled the pki secrets

   engine at: RE/ Success! Data

   written to: RE/config/ca Success!

   Data written to: RE/roles/coretech

   Success! Data written to:

   RE/config/urls

   Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl apply

secret/nginx-default-secret

configured configmap/public-

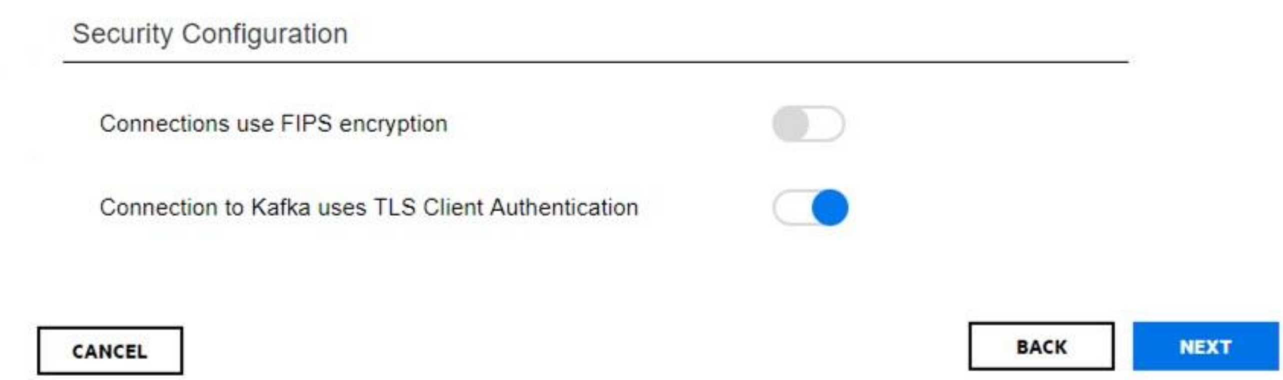
ca-certificates patched

configmap/public-ca-

certificates patched

4. Proceed with the Transformation Hub installation and into the configuration page

> **Note:** TLS Client Authentication and FIPS need to be enabled at this time. Client Authentication and FIPS cannot be enabled or disabled in the Transformation Hub **Reconfigure** page.



Security Configuration

Connections use FIPS encryption ○

Connection to Kafka uses TLS Client Authentication ●

Enabling Database SSL

1. Copy the following files to the database server /tmp directory:
   - vertica.crt
   - vertica.key
   - schedule.crt
   - schedule.key
   - ca.crt
2. Change the certificate key file ownership:

```
chown <dbadmin user> vertica.key scheduler.key
```

3. Enable the database server SSL

```
./db_ssl_setup --enable-ssl --vertica-cert-path /tmp/vertica.crt \
--vertica-key-path /tmp/vertica.key --client-ca-path /tmp/ca.crt
```

Verification:

4. Log in to database server as dbadmin user

```
mkdir ~/.vsql

cp /tmp/scheduler.crt ~/.vsql/client.crt

cp /tmp/scheduler.key ~/.vsql/client.key

cp /tmp/ca.crt ~/.vsql/root.crt

chmod 600 ~/.vsql/client.key
```

5. Log in to database cluster node1 as root user:

```
rm -rf /tmp/vertica.crt /tmp/vertica.key /tmp/issue_ca.crt /tmp/ca.crt
```

6. Check the database connection:

```
vsql -m require
```

Password:

Expected result:

```
 SSL connection (cipher: DHE-RSA-AES256-GCM-SHA384, bits: 256, protocol: 
TLSv1.2)
```

Run the following command:

```
dbadmin=> select user,authentication_method, ssl_state from sessions where
session_id  = current_session();
```

Expected result:

```
current_user | authentication_method | ssl_state
--------------+-----------------------+-----------

dbadmin | Password | Mutual

(1 row)
```

Enabling SSL in Scheduler

To enable SSL in scheduler, run the following command:

```
./sched_ssl_setup --enable-ssl --sched-cert-path /tmp/scheduler.crt \
--sched-key-path /tmp/scheduler.key --vertica-ca-path /tmp/ca.crt \
--kafka-ca-path /tmp/ca.crt
```

Creating Scheduler with SSL Enabled

To create Scheduler with SSL enabled, run the following command:

```
$db-install-DIR/kafka_scheduler create <WorkerNode1>:9093
```

Enabling SSL with the Database

1. Browse to https://<virtual-server-FQDN>:5443, if it is a multiple master, or https://<master- FQDN>:5443, if it is a single master.
2. Click **DEPLOYMENT > Deployments**.
3. Click the **...** icon under **REFRESH** and select **Reconfigure.** A new tab will be opened.
4. Select **FUSION**, and scroll down to **Database Configuration**.
5. Under **Database Configuration**, enable **Use SSL for Database Connections**
6. Copy the Database ca certificate into the **Database Certificate(s)** field, make sure not to include any blank spaces or missing line breaks to prevent a handshake authentication failure.
7. Click **SAVE**. This will restart the search engine pod for the SSL changes to take effect

Enabling SSL between Interset and the Database

This section provides information for enabling SSL between Interset and the database so that Interset can communicate with the database securely.

Creating Root Certificate

On a server:

Create a new ca key and cert

1. Create parameters for ca key

   ```
   mkdir /root/ca

   cd /root/ca

   mkdir certs crl newcerts private

   chmod 700 private

   touch index.txt

   echo 1000 > serial
   ```

2. Create the `/root/ca/openssl.cnf` file

   `vi /root/ca/openssl.cnf` and add the following example contents:

   ```
   # OpenSSL root CA configuration file.

   # Copy to `/root/ca/openssl.cnf`.

   [ ca ]

   default_ca = CA_default

   [ CA_default ]

   # Directory and file

   locations. dir    = /root/ca

   certs             = $dir/certs

   crl_dir           = $dir/crl

   new_certs_dir     = $dir/newcerts
   ```

```
database          =

$dir/index.txt

serial            = $dir/serial
RANDFILE          = $dir/private/.rand

# The root key and root certificate.

private_key       = $dir/private/ca.key

certificate       = $dir/certs/ca.crt

# For certificate revocation lists.

crlnumber         = $dir/crlnumber

crl               = $dir/crl/ca.crl.pem

crl_extensions    = crl_ext

default_crl_days  = 30

# SHA-1 is deprecated, so use SHA-2 instead.

default_md        = sha256

name_opt          = ca_default

cert_opt          = ca_default

default_days      = 375

preserve          = no

policy            = policy_strict

[ policy_strict ]

# The root CA should only sign intermediate certificates that match.

# See the POLICY FORMAT section of `man ca`.

countryName             = match

stateOrProvinceName     = match

organizationName        = match

organizationalUnitName  = optional

commonName              = supplied

emailAddress            = optional

[ policy_loose ]

# Allow the intermediate CA to sign a more diverse range of certificates.

# See the POLICY FORMAT section of the `ca` man page.
```

```
countryName             = optional

stateOrProvinceName   =   optional

localityName            = optional

organizationName    =     optional

organizationalUnitName  = optional

commonName              = supplied

emailAddress            = optional

[ req ]

# Options for the `req` tool (`man req`).

default_bits        = 2048

distinguished_name  = req_distinguished_name

string_mask         = utf8only

# SHA-1 is deprecated, so use SHA-2 instead.

default_md          = sha256

# Extension to add when the -x509 option is used.

x509_extensions     = v3_ca

[ req_distinguished_name ]

countryName                     = US

stateOrProvinceName             = California

localityName                    = Sunnyvale

0.organizationName              = EntCorp

organizationalUnitName          = Arcsight

commonName                      =  Common Name

emailAddress                    =  Email Address

# Optionally, specify some defaults.

countryName_default             = GB

stateOrProvinceName_default     = England

localityName_default            =
0.organizationName_default      = abcd

organizationalUnitName_default  =

emailAddress_default            =

[ v3_ca ]

# Extensions for a typical CA (`man x509v3_config`).

subjectKeyIdentifier = hash

authorityKeyIdentifier = keyid:always,issuer

basicConstraints = critical, CA:true

keyUsage = critical, digitalSignature, cRLSign, keyCertSign

[ v3_intermediate_ca ]

# Extensions for a typical intermediate CA (`man x509v3_config`).

subjectKeyIdentifier = hash

authorityKeyIdentifier = keyid:always,issuer

basicConstraints = critical, CA:true, pathlen:0

keyUsage = critical, digitalSignature, cRLSign, keyCertSign

[ usr_cert ]

# Extensions for client certificates (`man x509v3_config`).

basicConstraints = CA:FALSE

nsCertType = client, email

nsComment = "OpenSSL Generated Client Certificate"

subjectKeyIdentifier = hash

authorityKeyIdentifier = keyid,issuer

keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment

extendedKeyUsage = clientAuth, emailProtection

[ server_cert ]

# Extensions for server certificates (`man x509v3_config`).

basicConstraints = CA:FALSE

nsCertType = server
```

```
nsComment = "OpenSSL Generated Server Certificate"

subjectKeyIdentifier = hash

authorityKeyIdentifier = keyid,issuer:always

keyUsage = critical, digitalSignature, keyEncipherment

extendedKeyUsage = serverAuth

[ crl_ext ]

# Extension for CRLs (`man x509v3_config`).

authorityKeyIdentifier=keyid:always

[ ocsp ]

# Extension for OCSP signing certificates (`man ocsp`).

basicConstraints = CA:FALSE

subjectKeyIdentifier = hash

authorityKeyIdentifier = keyid,issuer

keyUsage = critical, digitalSignature

extendedKeyUsage = critical, OCSPSigning
```

3. Generate the new ca root key

```
cd /root/ca

openssl genrsa -out private/ca.key 4096

chmod 400 private/ca.key
```

4. Create the new ca cert

```
openssl req -config openssl.cnf \

-key private/ca.key \

-new -x509 -days 365 -sha256 -extensions v3_ca \

-out certs/ca.crt

...

If you enter '.', the field will be left blank.

-----

US [GB]:US
```

```
California [England]:California

Sunnyvale []:Sunnyvale

EntCorp [abcd]:

Arcsight []:Arcsight

Common Name []:root ca

Email Address []:admin@abcd.com
```

5. Verify the root ca

```
chmod 444 certs/ca.crt

openssl x509 -noout -text -in certs/ca.crt
```

Creating an Intermediate Certificate

1. Create parameters for intermediate key

```
mkdir /root/ca/intermediate/

cd /root/ca/intermediate

mkdir certs crl csr newcerts private

chmod 700 private

touch index.txt

echo 1000 > serial

echo 1000 > /root/ca/intermediate/crlnumber
```

   a. Create the /root/ca/intermediate/openssl.cnf file

```
vi /root/ca/intermediate/openssl.cnf  and add the following contents -
make sure the dir is unique for each intermediate cert created:

[ ca ]

default_ca = CA_default

[ CA_default ]

# Directory and file locations.

dir               = /root/ca/intermediate

certs             = $dir/certs

crl_dir           = $dir/crl
```

```
new_certs_dir     = $dir/newcerts

database          = $dir/index.txt

serial            = $dir/serial

RANDFILE          = $dir/private/.rand

# The root key and root certificate.

private_key       = $dir/private/intermediate.key

certificate       = $dir/certs/intermediate.crt

# For certificate revocation lists.

crlnumber         = $dir/crlnumber

crl               = $dir/crl/intermediate.crl.pem

crl_extensions    = crl_ext

default_crl_days  = 30

# SHA-1 is deprecated, so use SHA-2 instead.

default_md        = sha256

name_opt          = ca_default

cert_opt          = ca_default

default_days      = 375

preserve          = no

policy            = policy_loose

[ policy_strict ]

# The root CA should only sign intermediate certificates that match.

# See the POLICY FORMAT section of `man ca`.

countryName             = match

stateOrProvinceName     = match

organizationName        = match

organizationalUnitName  = optional

commonName              = supplied

emailAddress            = optional
```

```
[ policy_loose ]
# Allow the intermediate CA to sign a more diverse range of
certificates.
# See the POLICY FORMAT section of the `ca` man page.
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName    =     optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
[ req ]
# Options for the `req` tool (`man req`).
default_bits        = 2048
distinguished_name  = req_distinguished_name
string_mask         = utf8only
# SHA-1 is deprecated, so use SHA-2 instead.
default_md          = sha256
# Extension to add when the -x509 option is used.
x509_extensions     = v3_ca
[ req_distinguished_name ]
# See <https://en.wikipedia.org/wiki/Certificate_signing_request>.
countryName                     = Country Name (2 letter code)
stateOrProvinceName             = State or Province Name
localityName                    = Locality Name
0.organizationName              = Organization Name
organizationalUnitName          = Organizational Unit Name
commonName                      = Common Name
```

```
emailAddress                    = Email Address
# Optionally, specify some defaults.
countryName_default             = GB
stateOrProvinceName_default     = England
localityName_default            =
0.organizationName_default      = abcd
organizationalUnitName_default  =
emailAddress_default            =
[ v3_ca ]
# Extensions for a typical CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
[ v3_intermediate_ca ]
# Extensions for a typical intermediate CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true, pathlen:0
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
[ usr_cert ]
# Extensions for client certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = client, email
nsComment = "OpenSSL Generated Client Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment
```

```
extendedKeyUsage = clientAuth, emailProtection

[ server_cert ]

# Extensions for server certificates (`man x509v3_config`).

basicConstraints = CA:FALSE

nsCertType = server

nsComment = "OpenSSL Generated Server Certificate"

subjectKeyIdentifier = hash

authorityKeyIdentifier = keyid,issuer:always

keyUsage = critical, digitalSignature, keyEncipherment

extendedKeyUsage = serverAuth

[ crl_ext ]

# Extension for CRLs (`man x509v3_config`).

authorityKeyIdentifier=keyid:always

[ ocsp ]

# Extension for OCSP signing certificates (`man ocsp`).

basicConstraints = CA:FALSE

subjectKeyIdentifier = hash

authorityKeyIdentifier = keyid,issuer

keyUsage = critical, digitalSignature

extendedKeyUsage = critical, OCSPSigning
```

b. Generate the new Intermediate ca key

```
cd /root/ca

openssl genrsa -out intermediate/private/intermediate.key 4096
```

c. Create the intermediate ca certificate signing request (csr)

```
chmod 400 intermediate/private/intermediate.key

openssl req -config intermediate/openssl.cnf -new -sha256 \

-key intermediate/private/intermediate.key \

-out intermediate/csr/intermediate.csr.pem
```

```
...
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:US
State or Province Name [England]:California
Locality Name []:Sunnyvale
Organization Name [abcd]:
Organizational Unit Name []:Arcsight
Common Name []:intermediate ca
Email Address []:admin@abcd.com
```

d. Create the new Intermediate ca cert

```
cd /root/ca
openssl ca -config openssl.cnf -extensions v3_intermediate_ca \
-days 3650 -notext -md sha256 \
-in intermediate/csr/intermediate.csr.pem \
-out intermediate/certs/intermediate.crt
Sign the certificate? [y/n]: y
1 out of 1 certificate requests certified, commit? [y/n]y
chmod 444 intermediate/certs/intermediate.crt
```

e. Verify the Intermediate ca

```
openssl x509 -noout -text \
-in intermediate/certs/intermediate.crt
```

f. Verify the Intermediate cert against the root ca

```
openssl verify -CAfile certs/ca.crt \
intermediate/certs/intermediate.crt
# intermediate.crt: OK
```

Creating CA chains Certificate

```
cd /root/ca

cat certs/ca.crt intermediate/certs/intermediate.crt > chain.crt
```

Creating Database server Certificate

a. Create database key

```
openssl genrsa -out vertica.key 4096
```

b. Create database server certificate signing request

```
openssl req -new -key vertica.key -out vertica.csr  -subj
"/C=US/ST=California/L=Santa Clara/O=Micro
Focus/OU=Arcsight/CN=Vertica/emailAddress=admin@abcd.com" -nodes -sha256
```

c. Sign the certificate signing request

```
openssl x509 -req -in vertica.csr -CA
intermediate/certs/intermediate.crt -CAkey
intermediate/private/intermediate.key -CAcreateserial -extensions server
-days 3650 -outform PEM -out vertica.crt
```

d. Verify the scheduler client certificate

```
openssl verify -CAfile chain.crt vertica.crt

vertica.crt: OK
```

Creating scheduler client Certificate

1. Create client key

```
openssl genrsa -out scheduler.key 4096
```

2. Create client certificate signing request

```
openssl req -new -key scheduler.key -out scheduler.csr  -subj
"/C=US/ST=California/L=Santa Clara/O=Micro
Focus/OU=Arcsight/CN=Scheduler/emailAddress=admin@abcd.com" -nodes -sha256
```

3. Sign the certificate signing request

```
openssl x509 -req -in scheduler.csr -CA
intermediate/certs/intermediate.crt -CAkey
```

```
intermediate/private/intermediate.key -CAcreateserial -extensions client -
days 3650 -outform PEM -out scheduler.crt
```

4. Verify the scheduler client certificate

```
openssl verify -CAfile chain.crt scheduler.crt

scheduler.crt: OK
```

Installing self-signed CA during the TH installation

1. Install Transformation Hub

```
cdf-2020.02/install  --k8s-home /opt/arcsight/kubernetes -u admin ......
```

2. Access the CDF UI

```
https://n15-214-128-h125.arcsight.com:3000
```

After infrastructure services are deployed, wait for the **Preparation Complete** page to be displayed.

3. Installing intermediate certificate and key

```
scp previously generated intermediate.key, intermediate.crt, and ca.crt to
Master node1's /opt/cert.

On Master node1

mkdir /opt/cert

scp previously generated intermediate.key, intermediate.crt, and ca.crt
to /opt/cert

cd /opt/cert

/opt/arcsight/kubernetes/scripts/cdf-updateRE.sh write --re-
key=/opt/cert/intermediate.key --re-crt=/opt/cert/intermediate.crt  --re-
ca=/opt/cert//ca.crt

...

Dry run to check the certificate/key files.

Success! Enabled the pki secrets engine at: RE_dryrun/

Success! Data written to: RE_dryrun/config/ca

Success! Disabled the secrets engine (if it existed) at: RE_dryrun/

Dry run succeeded.
```

```
Submitting the certificate/key files to platform. CA for external
communication will be replaced.

Success! Disabled the secrets engine (if it existed) at: RE/

Success! Enabled the pki secrets engine at: RE/

Success! Data written to: RE/config/ca

Success! Data written to: RE/roles/coretech

Success! Data written to: RE/config/urls

Warning: kubectl apply should be used on resource created by either
kubectl create --save-config or kubectl apply

secret/nginx-default-secret configured

configmap/public-ca-certificates patched

configmap/public-ca-certificates patched
```

4. Continue with the installation.
5. Under **Transformation Hub > Security Configuration page**, turn ON **Connection to kafka uses TLS Client Authentication.**

> **Note:** TLS Client Authentication and FIPS need to enabled at this time if the system is planning to use TSL client authentication and FIPS. Client Authentication in post-deployment can't be changed after this point.

6. Continue with the Transformation Hub/Interset suite deployment.
7. Enable SSL on Database cluster

```
On Vertica server node1,

mkdir /opt/cert

scp created chain.crt scheduler.crt scheduler.key vertica.crt vertica.key
intermediate.key to /opt/cert

chown -R $dbadmin:$dbadmin /opt/cert
```

8. Enable database server SSL

```
cd to $db_install-DIR

./db_ssl_setup --enable-ssl --vertica-cert-path /opt/cert/vertica.crt --
vertica-key-path /opt/cert/vertica.key --client-ca-path
/opt/cert/chain.crt

...
```

```
2020-07-15 14:27:11,422 DEBUG Installing Certs/Keys for SSL: Return code:
0, Out put: Parameters set successfully

2020-07-15 14:27:11,451 DEBUG Enabling EnableSSL flag: Return code: 0,
Output: A LTER DATABASE

WARNING 4324: Parameter EnableSSL will not take effect until database
restart

2020-07-15 14:27:11,451 INFO ENABLED SSL/TLS MODE FOR VERTICA

...

Starting Vertica on all nodes. Please wait, databases with a large catalog
may take a while to initialize.

...

Database investigate: Startup Succeeded. All Nodes are UP
```

9. Verify database SSL
   a. Login to dabase node1 server as $dbadmin

```
mkdir ~/.vsql
cp /opt/cert/scheduler.crt ~/.vsql/client.crt
cp /opt/cert/scheduler.key ~/.vsql/client.key
cp /opt/cert/chain.crt ~/.vsql/root.crt
chmod 400 ~/.vsql/client.key
```

   b. Check the database

connection `vsql -m`

```
require Password:
Expected result:
SSL connection (cipher: DHE-RSA-AES256-GCM-SHA384, bits: 256, protocol:
TLSv1.2)
dbadmin=>select user,authentication_method, ssl_state from sessions
where session_id = current_session();
Expected result:
current_user | authentication_method | ssl_state
--------------+-----------------------+-----------
dbadmin | Password | Mutual (1 row)
```

Configure SSL Connection for Database from Management Portal

1. Browse t o the management portal at https://<virtual_FQDN>:5443, or at https://<master_node1_ FQDN>:5443.
2. Click **DEPLOYMENT**, and select **Deployments.**
3. Click the **Three Dots** (Browse) on the far right and choose **Reconfigure**. A new screen will be opened in a separate tab.

4. Go to **Fusion > Database Configuration >**
   a. Turn ON **Use SSL for Database Connections**
   b. Copy `/opt/cert/chain.crt` to the Database Certificate(s) field

Enabling SSL in Scheduler

```
cd $db_install-DIR

./sched_ssl_setup --enable-ssl --sched-cert-path /opt/cert/scheduler.crt --
sched-key-path /opt/cert/scheduler.key --vertica-ca-path /opt/cert/chain.crt
--vertica-ca-key /opt/cert/intermediate.key --kafka-ca-path
/opt/cert/chain.crt

....

Entry for alias vertica_caroot successfully imported.

Import command completed: 1 entries successfully imported, 0 entries failed
or cancelled

...

2020-07-15 14:44:08,566 INFO Key pair imported successfully into
/opt/installer/wrk/ks.pkcs12

2020-07-15 14:44:10,040 DEBUG Import Key Pair: Return code: 0, Output:
Importing keystore /opt/installer/wrk/ks.pkcs12 to
/opt/installer/wrk/scheduler.keystore.bcfks...

Entry for alias scheduler_key successfully imported.

Import command completed: 1 entries successfully imported, 0 entries failed
or cancelled

2020-07-15 14:44:10,040 INFO Key pair imported successfully into
/opt/installer/wrk/scheduler.keystore.bcfks

2020-07-15 14:44:10,041 INFO Created file /opt/installer/wrk/vkconfig.cnf
successfully

...
```

Creating Scheduler with SSL Enabled

```
./kafka_scheduler create <TH_WorkerNode1>:9093
```

Setting FIPS on Database Server

In order to enable FIPS mode in Interset we have to set the OS in FIPS mode.

To enable FIPS in the OS

1. Run the below commands:

```
yum install dracut-fips
yum install dracut-fips-aesni
rpm -q prelink && sed -i '/^PRELINKING/s,yes,no,' /etc/sysconfig/prelink
```

Ignore the error if prelink was not installed.

```
mv -v /boot/initramfs-$(uname -r).img{,.bak}
dracut
grubby --update-kernel=$(grubby --default-kernel) --args=fips=1
uuid=$(findmnt -no uuid /boot)
[[ -n $uuid ]] && grubby --update-kernel=$(grubby --default-kernel) \
--args=boot=UUID=${uuid}
reboot
```

b. To verify if FIPS has been enabled, run the following command:

```
sysctl crypto.fips_enabled
```

Expected Result: `crypto.fips_enabled =  1`

To disable FIPS

1. Run the below

commands: `yum remove`

```
dracut-fips dracut --
force
grubby --update-kernel=$(grubby --default-kernel) --remove-args=fips=1
reboot
```

2. To verify if FIPS has been disabled, run the following command:

```
sysctl crypto.fips_enabled
```

Expected Result: crypto.fips_enabled = 0

Managing the Database

This section provides information about managing the database.

Backing Up and Restoring the Database

You should back up and restore the database before you upgrade it or before you add or remove a database node.

Consider the following when backing up and restoring the database:

- The backup process can consume additional storage. The amount of space that the backup consumes depends on the size of your catalog and any objects that you drop during the backup. The backup process releases this storage after the backup is complete.
- You can only restore backups to the same version of database. For example, you cannot back up Database 9.1.0 and restore it to Database 9.2.1.
- Ingesting events into the database during backup might exclude the most recently ingested events from the backup. To ensure that all events are backed up, stop ingestion before you start the backup.
- For optimal network performance, each database node should have its own backup host.
- Use one directory on each database node to store successive backups.
- You can save backups to the local folder on the database node, if there is enough space available, or to a remote server.
- You can perform backups on ext3, ext4, NFS and XFS file systems.

Preparing the Backup Host

Micro Focus recommends that each backup host have space for at least twice the database node footprint size. Consider your long-term backup storage needs.

If you are using a single backup location, you can use the following database operation to estimate the required storage space for the database cluster:

```
dbadmin=> select sum(used_bytes) as total_used_bytes from v_monitor.storage_
containers;

total_used_bytes

------------------

 5717700329

(1 row)
```

If you are using multiple backup locations, one per node, use the following database operation to estimate the required storage space:

```
dbadmin=> select node_name, sum(used_bytes) as total_used_bytes from v_
monitor.storage_containers group by node_name;

 node_name | total_used_bytes

------------------------+---------------------

 v_investigate_node0002 | 1906279083

 v_investigate_node0003 | 1905384292

 v_investigate_node0001 | 1906036954

(3 rows)
```

Remote backup hosts must have SSH access.

The database administrator must have password-less SSH access from database node1 to the backup hosts, as well as from the restored database node1.

To set up password-less SSH:

1. Log in to the backup server.
2. Create user `$dbadmin`.

   `$dbadmin` is the administrator for the database cluster.
3. Ensure that `$dbadmin` has write permission to the dedicated directory where you will store the backup.
4. Log in to database node1 as `root`.
5. Change to the database administrator:

```
# su -l $dbadmin
```

6. Setup password-less SSH for all backup servers:

```
# ssh-copy-id -i ~/.ssh/id_rsa.pub $dbadmin@$back_up_server_ip
```

Preparing Backup Configuration File

Database includes sample configuration files that you can copy, edit, and deploy for your various *vbr* tasks. Database automatically installs these files at `/opt/vertica/share/vbr/example_configs.`

For more information, please see: Sample VBR .ini Files.

The default number of restore points (`restorePointLimit`) is 52, assuming a weekly backup for one year. Using multiple restore points gives you the option to recover from one of several backups. For

example, if you specify 3, you have 1 current backup and 3 backup

archives. We use `backup_restore_full_external.ini` as an

example.

```
# su - $dbadmin

# cp /opt/vertica/share/vbr/example_configs/backup_restore_full_external.ini
db_backup.ini

# vi db_backup.ini
```

Note: You must save a copy of `db_backup.ini` for future tasks.

Note: The following is an example for reference only .`v_investigate_node000*` is hard coded.

`dbName = investigate` is hard coded.

```
# cat db_backup.ini

; This sample vbr configuration file shows full or object backup and restore
to a separate remote backup-host for each respective database host.

; Section headings are enclosed by square brackets.

; Comments have leading semicolons (;) or pound signs (#).

; An equal sign separates options and values.

; Specify arguments marked '!!Mandatory!!' explicitly.

; All commented parameters are set to their default value.

; ------------------------------------------- ;

;;; BASIC PARAMETERS ;;;

; ------------------------------------------- ;
```

```
[Mapping]

; !!Mandatory!! This section defines what host and directory will store the
backup for each node.

; node_name = backup_host:backup_dir

; In this "parallel backup" configuration, each node backs up to a distinct
external host.

; To backup all database nodes to a single external host, use that single
hostname/IP address in each entry below.

v_investigate_node0001 = 192.168.1.1:/opt/dbadmin/backups

v_investigate_node0002 = 192.168.1.2:/opt/dbadmin/backups

v_investigate_node0003 = 192.168.1.3:/opt/dbadmin/backups

[Misc]

; !!Recommended!! Snapshot name. Object and full backups should always have
different snapshot names.

; Backups with the same snapshotName form a time sequence limited by
restorePointLimit.

; SnapshotName is used for naming archives in the backup directory, and for
monitoring and troubleshooting.

; Valid characters: a-z A-Z 0-9 - _

snapshotName = Vertica_backup_09_09_2019

[Database]

; !!Recommended!! If you have more than one database defined on this Vertica
cluster, use this parameter to specify which database to backup/restore.

dbName = investigate

; If this parameter is True, vbr prompts the user for the database password
every time.

; If False, specify the location of password config file in 'passwordFile'
parameter in [Misc] section.

dbPromptForPassword = True

; ------------------------------------------- ;

;;; ADVANCED PARAMETERS ;;;

; ------------------------------------------- ;

[Misc]

; The temp directory location on all database hosts.

; The directory must be readable and writeable by the dbadmin, and must
implement POSIX style fcntl lockf locking.
```

```
tempDir = /tmp
; How many times to retry operations if some error occurs.
retryCount = 2
; Specifies the number of seconds to wait between backup retry attempts, if a
failure occurs.
retryDelay = 1
; Specifies the number of historical backups to retain in addition to the
most recent backup.
; 1 current + n historical backups
restorePointLimit = 52
; Full path to the password configuration file
; Store this file in directory readable only by the dbadmin
; (no default)
; passwordFile = /path/to/vbr/pw.txt
; When enabled, Vertica confirms that the specified backup locations contain
; sufficient free space and inodes to allow a successful backup. If a backup
; location has insufficient resources, Vertica displays an error message
explaining the shortage and
; cancels the backup. If Vertica cannot determine the amount of available
space
; or number of inodes in the backupDir, it displays a warning and continues
; with the backup.
enableFreeSpaceCheck = True
; When performing a backup, replication, or copycluster, specifies the
maximum
; acceptable difference, in seconds, between the current epoch and the backup
epoch.
; If the time between the current epoch and the backup epoch exceeds the
value
; specified in this parameter, Vertica displays an error message.
SnapshotEpochLagFailureThreshold = 3600
[Transmission]
; Specifies the default port number for the rsync protocol.
port_rsync = 50000
```

```
; Total bandwidth limit for all backup connections in KBPS, 0 for unlimited. Vertica distributes

; this bandwidth evenly among the number of connections set in concurrency_ backup.

total_bwlimit_backup = 0

; The maximum number of backup TCP rsync connection threads per node.

; Optimum settings depend on your particular environment.

; For best performance, experiment with values between 2 and 16.

concurrency_backup = 2

; The total bandwidth limit for all restore connections in KBPS, 0 for unlimited

total_bwlimit_restore = 0

; The maximum number of restore TCP rsync connection threads per node.

; Optimum settings depend on your particular environment.

; For best performance, experiment with values between 2 and 16.

concurrency_restore = 2

[Database]

; Vertica user name for vbr to connect to the database.

; This setting is rarely needed since dbUser is normally identical to the database administrator

dbUser = $dbadmin
```

Backing Up the Database

The `$dbadmin` user must perform the backup from the database node1 of the cluster.

> **Note:** vbr Command Reference.

To back up the database:

1. Stop Kafka scheduler

   Login to database node1 as `root`

   ```
   # cd $db_install-DIR
   # ./kafka_scheduler stop
   ```
2. Initialize backup location

```
# su - $dbadmin
# vbr -t init --config-file db_backup.ini
Initializing backup locations.

Backup locations initialized.
```

3. Back up data:

```
# vbr -t backup -c db_backup.ini
Enter vertica password:
Starting backup of database investigate.
Participating nodes: v_investigate_node0001,v_investigate_node0002,v_
investigate_node0003.
Snapshotting database.
Snapshot complete.
Approximate bytes to copy: 270383427 of 270383427 total.
[================================================]   100%
Copying backup metadata.
Finalizing backup.
Backup complete!
```

4. Verify that the backup files were written to the backup locations:

```
# ssh 192.168.1.1 ls /opt/dbadmin/backups
backup_manifest
Objects
Snapshots
# ssh 192.168.1.2 ls /opt/dbadmin/backups
backup_manifest
Objects
Snapshots
# ssh 192.168.1.3 ls /opt/dbadmin/backups
backup_manifest
Objects
Snapshots
```

Backing Up Database Incrementally

Incremental backups use the same setup as a full backup and only back up what changed from the previous full backup. When you perform a full backup using the same configuration file, subsequent backups are incremental. When you start an incremental backup, the `vbr` tool displays a backup size that is a portion of the total backup size. This portion represents the delta changes that will be backed up during the incremental backup.

Run the following command to perform an incremental backup:

```
# vbr --task backup --config-file db_backup.ini
```

Verifying the Integrity of the Backup

Use the `full-check` option to verify the integrity of the database backup. The option reports the following:

- Incomplete restore points
- Damaged restore points
- Missing backup files
- Unreferenced files

To verify the backup integrity, run the following command:

```
# vbr --task full-check --config-file db_backup.ini

Enter vertica password:

Checking backup consistency.

List all snapshots in backup location:

Snapshot name and restore point: Vertica_backup_09_09_2019_20190909_010826,
nodes:['v_investigate_node0001', 'v_investigate_node0002', 'v_investigate_
node0003'].

Regenerating backup manifest for location rsync://
[192.168.1.1]:50000/opt/dbadmin/backups

Regenerating backup manifest for location rsync://
[192.168.1.2]:50000/opt/dbadmin/backups

Regenerating backup manifest for location rsync://
[192.168.1.3]:50000/opt/dbadmin/backups

Snapshots that have missing objects(hint: use 'vbr --task remove' to delete
these snapshots):

Backup locations have 0 unreferenced objects

Backup locations have 0 missing objects
Backup consistency check complete.
```

Managing Backups

This section describes how to view and delete backups. To view available backups, run the following command:

```
# vbr --task listbackup --config-file db_backup.ini

Enter vertica password:

backup backup_type epoch objects include_patterns exclude_patterns nodes
(hosts) version file_system_type

Vertica_backup_09_09_2019_20190909_010826 full 6058

 v_investigate_node0001(192.168.10.11), v_investigate_node0002
(192.168.10.12), v_investigate_node0003(192.168.10.13) v9.2.1-6 [Linux]

The backup name includes the backup time-stamp.
```

Backup times-tamp can be found by using listbackup option, i.e. `20190909_010826` from `Vertica_ backup_09_09_2019_20190909_010826.`

To delete a backup, run the following command:

```
# vbr --task remove --config-file db_backup.ini --archive 20190909_010826

Enter vertica password:

Removing restore points: 20190909_010826

Remove complete!
```

Preparing to Restore Database Data

Before you restore database data, ensure that your environment meets the following requirements:

- You can only restore backups to the same version of database from which you made the backup. For example, you cannot backup Database 9.1.0 and restore it to Database 9.2.1.
- You can restore backup to the original cluster where the backup was generated. However, all data ingested to the database after backup will be lost. If backup is restored to a new cluster, you must restore to a cluster that is identical to the cluster from which you made the backup (same or larger disk size). Ensure that the cluster meets the following requirements:
  - The target database is created and empty.
  - The target database name matches the backup database name.
  - The target database is stopped.
  - All database nodes in the target cluster are running.

- All database node names in the target cluster match the names from the backup.

Restoring the Database

The `$dbadmin`user must restore from the database node1 of the cluster.

To set up password-less SSH:

1. Log in to the target database node1 as root.

2. Change to the database administrator:

```
# su -l $dbadmin
```

3. Setup password-less SSH for all backup servers:

```
# ssh-copy-id -i ~/.ssh/id_rsa.pub $dbadmin@$back_up_server_ip
```

To restore the database:

1. Build a target database cluster that is identical to the original cluster.
2. Log in to the target database node1 and stop the database:

   ```
   # cd $db_install-DIR
   # ./db_installer stop-db
   ```

3. Become the `$dbadmin` user:

   ```
   # su -l $dbadmin
   ```

4. Copy `db_backup.ini` to `/home/$dbadmin`.
5. Restore the backup data:

   ```
   # vbr --task restore --config-file db_backup.ini
   ```

   The output should be similar to the following:

   ```
   Enter vertica password:
   Starting full restore of database Investigate.
   Participating nodes: v_investigate_node0001, v_investigate_node0002, v_
   investigate_node0003.
   Restoring from restore point: investigate_backup_20190909_010826
   Determining what data to restore from backup.
   [================================================]   100%
   Approximate bytes to copy: 270383427 of 270383427 total.
   Syncing data from backup to cluster nodes.
   [================================================]   100%
   Restoring catalog.
   Restore complete!
   ```

6. Start the database:

```
# exit
# ./db_installer start-db
```

The output should be similar to the following:

```
Starting nodes:
v_investigate_node0001 (127.0.0.1)
Starting Vertica on all nodes. Please wait, databases with a large catalog
may take a while to initialize.
Node Status: v_investigate_node0001: (DOWN)
Node Status: v_investigate_node0001: (DOWN)
Node Status: v_investigate_node0001: (DOWN)
Node Status: v_investigate_node0001: (DOWN)
Node Status: v_investigate_node0001: (UP)
Database Investigate started successfully
```

7. Start the Kafka scheduler:

```
# ./kafka_scheduler start
```

Configuring the Watchdog and Event Retention Time Policy on the Database

About Watchdog

A watchdog process automatically runs once a day to monitor cluster status and storage utilization. When watchdog detects a cluster node is in DOWN state, it will try to restart the node.

When storage utilization reaches the defined threshold (default is 95%), watchdog will start to purge data until utilization is under threshold.

To modify the default threshold:

1. Login to database cluster node1 as root
2. Change the database installer directory:

```
cd $db_install-DIR
```

3. Change the storage threshold value:

```
vi db.properties
```

```
STORAGE_THRESHHOLD= <new value>
```

For better disk management you can also put in place a data retention policy alongside watchdog.

Data Retention Policy

The retention period can range from 1 to 366 days. The data retention policy is based on calendar days. Calendar day is based on event's Normalized Event Time (NET).

Time-based data retention is disabled by default. When you enable it, the default retention period is 90 days, but that can be modified at any time. If you run the data retention script on 6/30/2019 and the

`db_retention_days property` is set to 90, then data older than 04/01/2019 will be deleted. You can purge data in real time or by using a scheduled cron job. Confirmation is needed when retention

period is set to less than 30 days.

> **Note:** Database data needs to be backed-up routinely. The backup policy is defined by the user. Always evaluate (-e option) retention policy before purging data.

To enable data retention:

1. Login to database cluster node1 as root
2. Change the database installer directory:

   ```
   cd $db_install-DIR
   ```
3. Check the cluster nodes disk usage

   ```
   ./db_installer status
   ```

   Check the `disk_space_free_percent` field to determine the retention day
4. Ensure your database is backed up.

   For more information, see "Backing Up the Database" on page 41.
5. Enable data retention policy:

   ```
   cd $db_install-DIR/config
   vi db_user.properties
   ```
   Uncomment `#db_retention_days=90`

6. Verify the number of days of data in the database:

```
cd $db_install-DIR/scripts
./retention_policy_util.sh -t
```

The result should be similar to the following:

```
----------------------------------------------------------------------
Investigate has 100 day(s) with time-range: [2017-10-26 - 2018-02-06].
----------------------------------------------------------------------
```

> **Note:** There are more than 100 calendar days between 2017-10-26 and 2018-02-06. The results above show that there are only 100 event days, meaning that 100 days have incoming events. Certain calendar days did not have incoming events.

7. To change the default retention period, enter the following command:

```
./retention_policy_util.sh -u <Number_of_Days>
```

To enable automatic purging based on event retention time period:

1. To create the purge process, enter the following command:

```
./retention_policy_util.sh -s
```

> **Note:** A cron job is scheduled to purge data daily.

2. To verify the created cron job, enter the following command:

```
./retention_policy_util.sh -l
```

Expected results:

```
-------------------------------------------------------------------------------------------

Current retention value is set to: 90 day(s)
-------------------------------------------------------------------------------------------

Current cronjob is running:
(59 23 * * * /opt/installer/scripts/retention_policy_util.sh -p &>>
/opt/installer/vertica-installer.log)
--------------------------------------------------------------------------------------------
```

3. To preview the purge results, enter the following command:

```
./retention_policy_util.sh -e
```

The results should be similar to the following:

```
***********************************************************************
No data will be purged. This is only evaluation for your retention policy
************************************************************************
```

```
Will purge time range : [ 2017-10-26 - 2017-10-31 ].
Will purge day 1, (2017-10-26)
Will purge day 2, (2017-10-27)
Will purge day 3, (2017-10-28)
Will purge day 4, (2017-10-29)
Will purge day 5, (2017-10-31)
***** done *****
```

4. To purge data in real time, enter the following command:

   ```
   ./retention_policy_util.sh -p
   ```
5. To disable the purge cron job, enter the following command:

   ```
   ./retention_policy_util.sh -d
   ```
6. To verify the disabled cron job, enter the following command:

   ```
   ./retention_policy_util.sh -l
   ```
   Expected results:

   ```
   ---------------------------------------------------------------------------------------
   Current retention value is set to: 90 day(s)
   ---------------------------------------------------------------------------------------
   ```

Monitoring the Database

You can monitor the Database by using a watchdog or commands.

Using Watchdog

Database includes a watchdog, which monitors the database nodes, to automatically purge data when the disk usage exceeds storage threshold and to automatically restart the node when the database node goes down.

Using Commands

You can monitor the status of the data by using the following commands:

```
$db_install-DIR/db_installer status

$db_install-DIR/kafka_scheduler status
```

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Deployment Guide (Database 9.2.1-16)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to arcsight_doc@microfocus.com.

We appreciate your feedback!