FortiSIEM - Disaster Recovery Procedures - Elasticsearch

Version 5.2.6

08/15/2020
FortiSIEM 5.2.6 Disaster Recovery Procedures - Elasticsearch

TABLE OF CONTENTS

Change Log

Date	Change Description
04/25/2018	Initial version of FortiSIEM - Disaster Recovery Procedures
08/19/2019	Revision 1: Updated the location of the image download site.
11/25/2019	Revision 2: Updated the recovery procedures.
03/30/2020	Release of Disaster Recovery Procedures for 5.3.0.
08/15/2020	Revision 3: All new content for Disaster Recovery.

Disaster Recovery

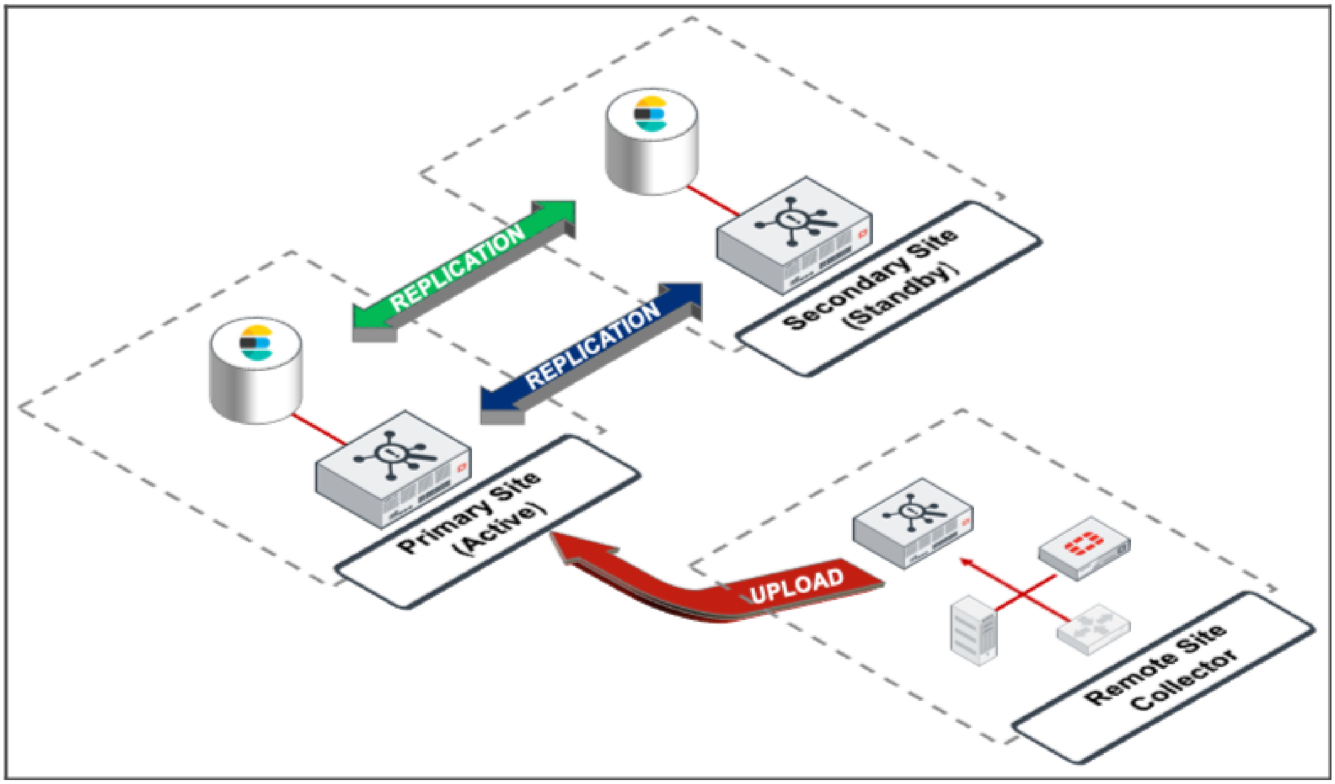
The following sections describe how to enable and work with the FortiSIEM Disaster Recovery (DR) feature.

Introduction

Understanding the FortiSIEM DR Feature

FortiSIEM has a replication feature, designed for those customers who require full disaster recovery capabilities, where one site is designated to be the Primary (active) and the other the Secondary (standby) site. The two systems replicate the Primary sites databases and data.

This requires a second fully licensed FortiSIEM system, where the Primary and Secondary Sites are identically setup in terms of Supervisor, Workers, and event storage in this case Elasticsearch.

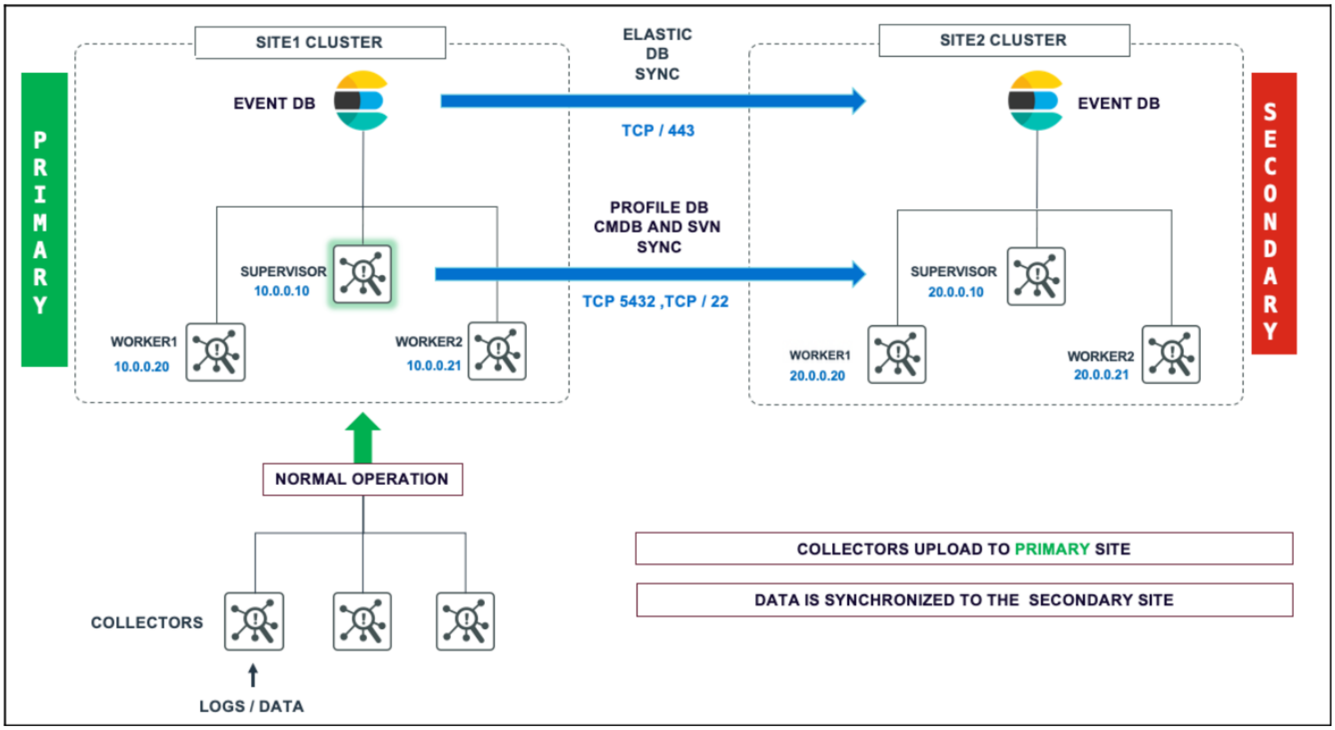


Under normal operations, if collectors are being used, these upload to the Primary site and will buffer by design when this site is not available. If DR is used, and a disaster occurs, then these same collectors will revert to uploading to the Secondary site which will now be designated as the Primary/Active site.

FortiSIEM runs as a cluster (or single node for a SMB) with Super, Worker, Report Server, and Collectors nodes.

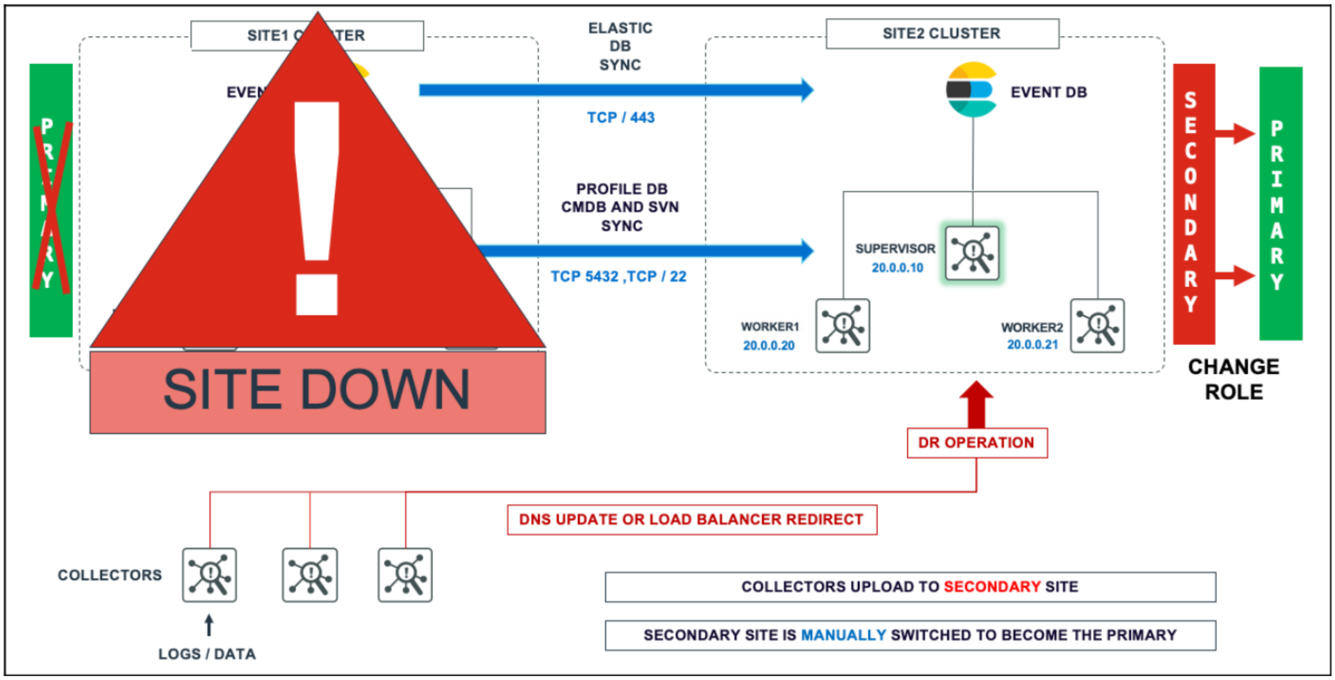
To provide DR features, FortiSIEM must have a Secondary system ready on standby to take over operations, with the following databases replicated from the Primary site:

- The CMDB residing in a PostGreSQL database.
- Device configurations residing in SVN on the Supervisor node.
- Profile data residing on SQLite databases on the Supervisor node.
- Event DB - Elasticsearch database.



When disaster strikes:

1. The Secondary must become the Primary FortiSIEM.
2. DNS Changes must be made so that users will logon to Secondary Supervisor, and that Collectors will send events to Secondary Workers.



When the Old Primary is recovered and powered up, it will sync missing data with the Secondary site (the Active Primary FortiSIEM).

When the user decides to return to the pre-disaster setup, the user can switch the roles of Primary and Secondary.

Prerequisites for a Successful DR Implementation

- Two separate FortiSIEM licenses - one for each site.
- The installation at both sites must be identical - workers, storage type, that is: Elasticsearch DB setup (Coordinator, Master, and Data Nodes), archive setup, report server setup, hardware resources (CPU, Memory, Disk) of the FortiSIEM nodes.
- DNS Names are used for the Supervisor nodes at the two sites. Make sure that users, collectors, and agents can access both supervisor nodes by their DNS names.
- DNS Names are used for the Worker upload addresses.
- TCP Ports for HTTPS (TCP/443), SSH (TCP/22) and PostGreSQL (TCP/5432) are open between both sites.
- It is recommended to use DNS Names for the Elasticsearch Coordinator nodes.

Understanding the Requirements for DNS Names

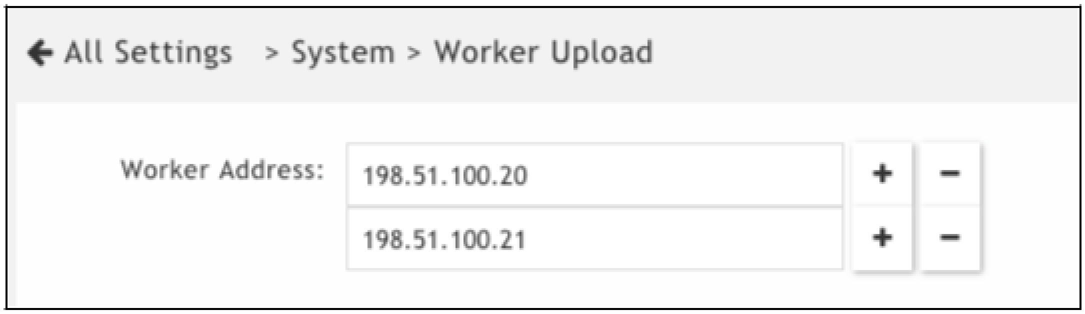
It is important to understand your FortiSIEM environment and plan ahead in terms of communications from users, agents and collectors.

Worker Upload

- Performing Collector Registration
- Agent Communications

Each entry in the Worker Upload address list is given to Collectors at registration (and periodically in communication to the Supervisor) to instruct where to upload customer event data.

An example is shown below, where the customer has *not* followed best practice advice and used IP Addresses and not FQDNs.



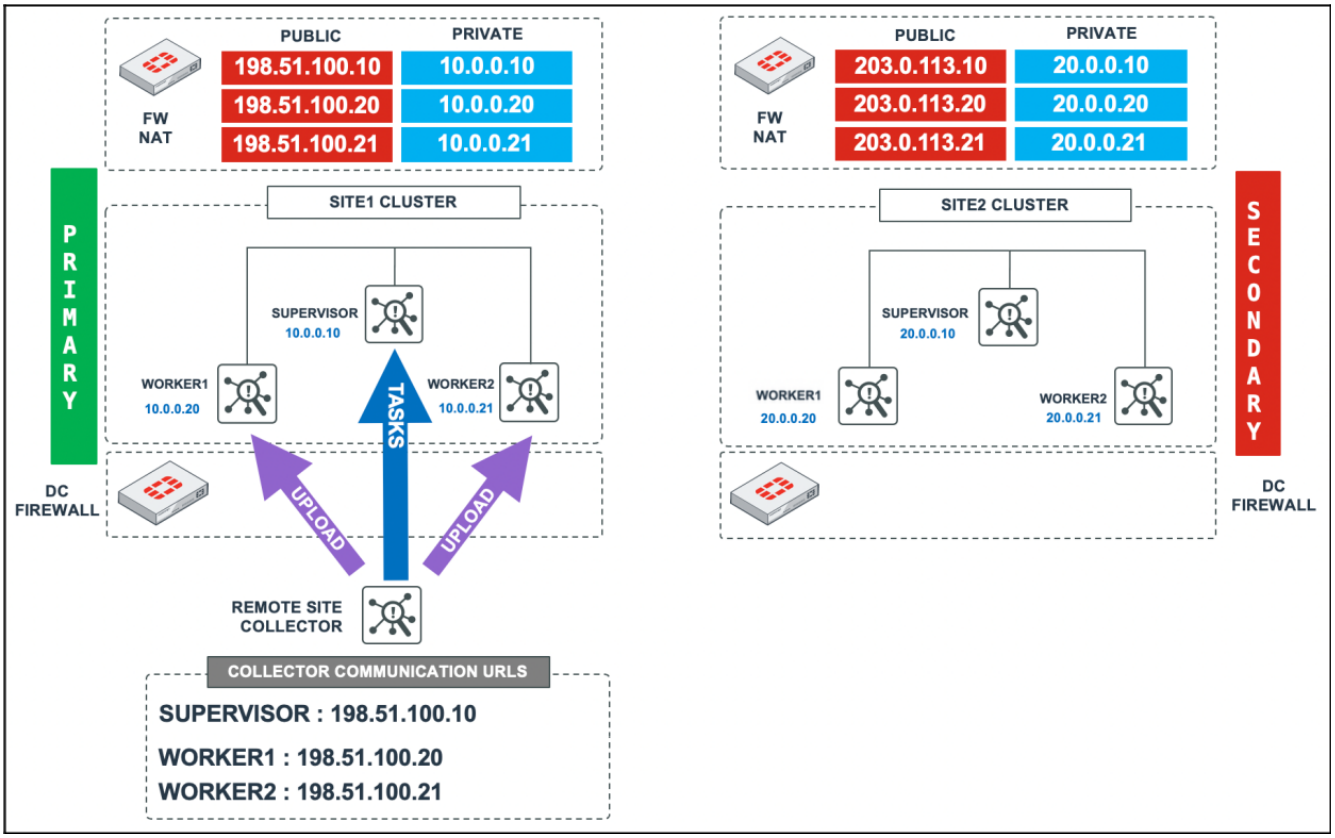
Worker Address: 198.51.100.20

198.51.100.21

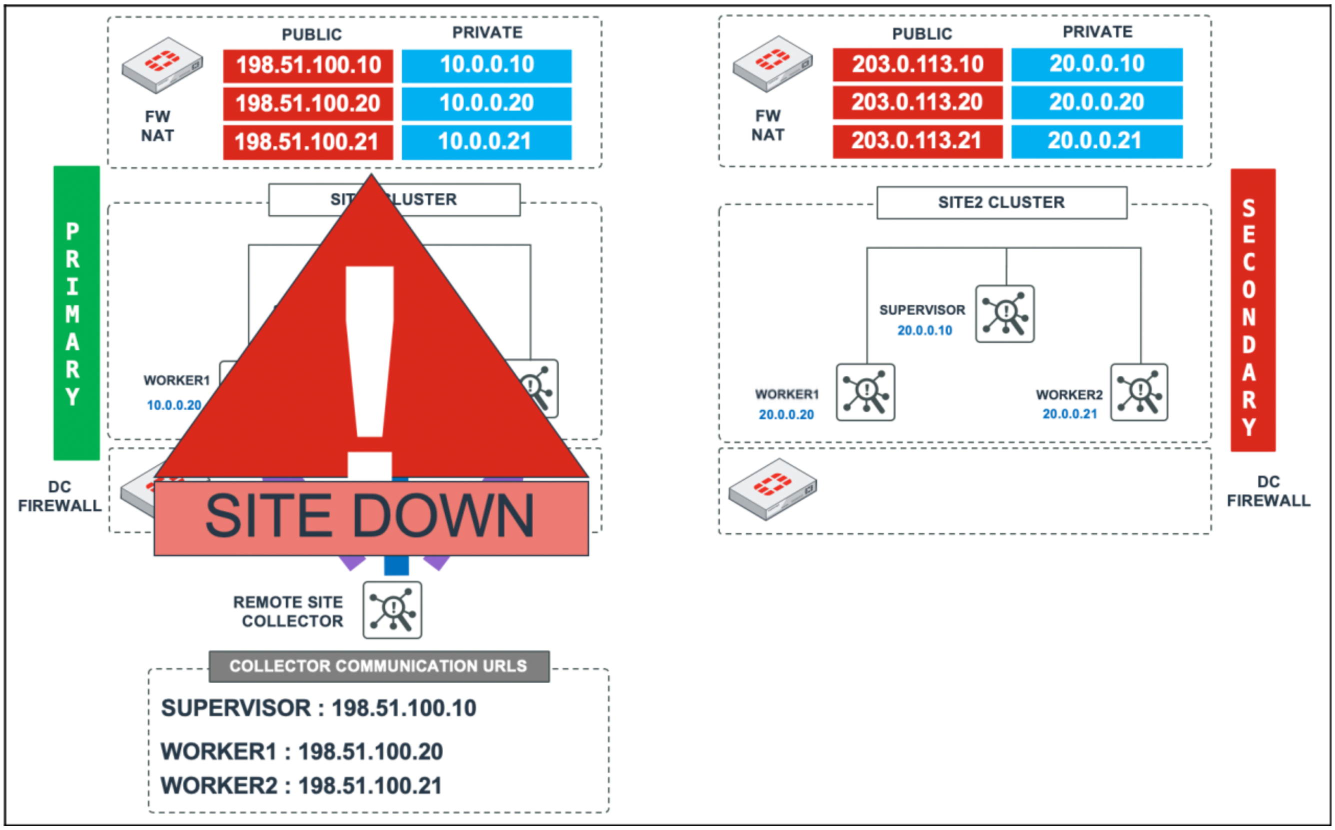
In addition to the Worker Upload entries, Collectors also maintain communication with the Supervisor node, to receive jobs/tasks and report Collector health data. When Collectors register for the first time with the Supervisor node, this communication address is stored for this purpose.

Why is using IP addresses for Collector registration and Worker Upload settings bad when it comes to DR planning?

Consider the environment below where only IP addresses have been used. During normal operations Collector traffic flows to the Workers at the Primary site and the Collector maintains communications with the Supervisor. This all works fine until the Primary site has a disaster.



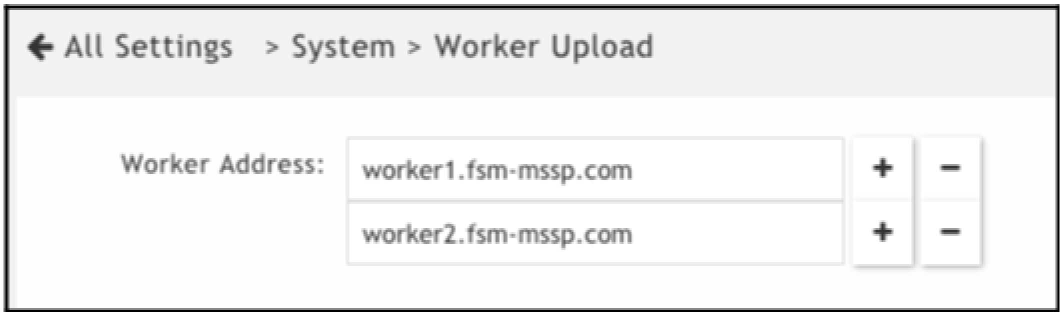
At this point, when the Primary node is unavailable the remote Collector nodes are essentially hard-coded (by IP) to talk to the Primary site only. Even if the Secondary node is up and operational and promoted to be the Primary node, Collectors are unable to upload logs or get any tasks from the Supervisor node due to the old Primary sites IPs being used.



A much better approach is to utilize DNS.

This allows name resolution to control which Supervisor, Primary or Secondary is currently active and which worker addresses to attempt to upload customer data to. DNS “A” records are created for the Supervisor nodes at both sites, and a “CNAME” is used to determine which is active, which has a small time to live (TTL) value.

The Worker Upload settings reference DNS addresses:



Worker Address: worker1.fsm-mssp.com

worker2.fsm-mssp.com

External DNS Example

Node	DNS Record Type	Name	IP/Alias
Supervisor (Primary)	A	site1.fsm-mssp.com	198.51.100.10
Supervisor (Secondary)	A	site2.fsm-mssp.com	203.0.113.10
Active Supervisor	CNAME	site.fsm-mssp.com	site1.fsm-mssp.com

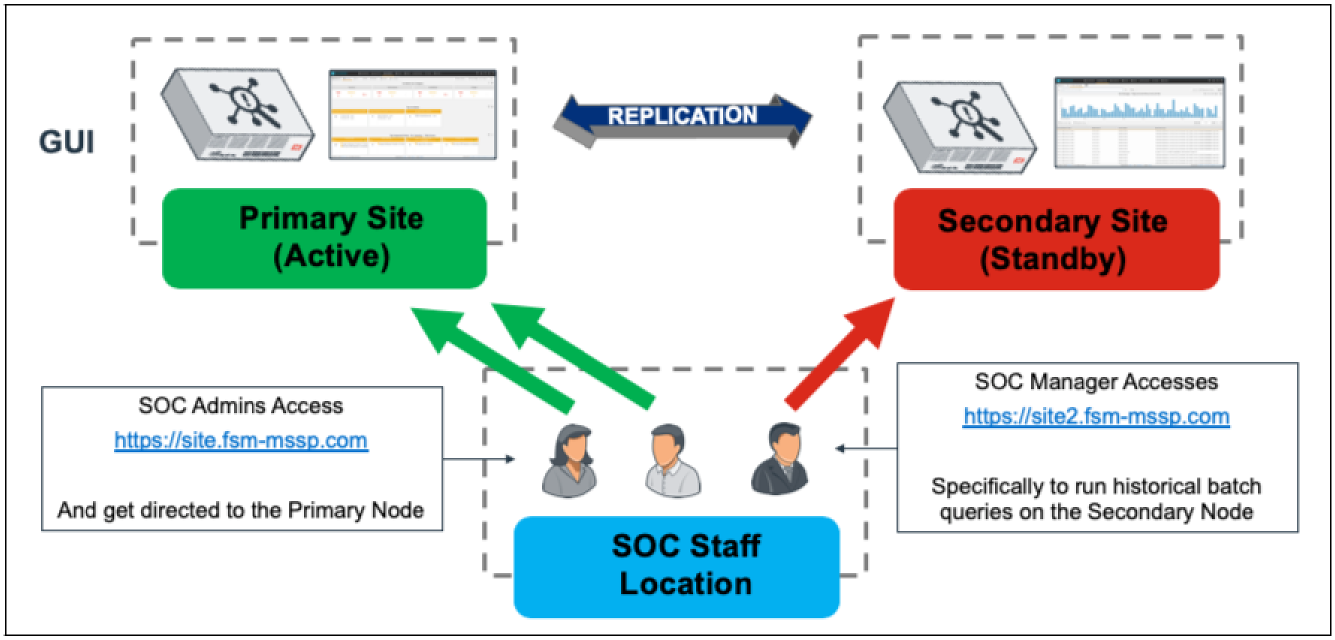
Node	DNS Record Type	Name	IP/Alias
Worker1 (Primary)	A	worker1.fsm-mssp.com	198.51.100.20
Worker2 (Primary)	A	worker2.fsm-mssp.com	198.51.100.21

For the internal DNS records, again both internal Supervisor addresses are listed with a CNAME to determine the current Primary GUI to logon to for SOC operators. (If public certificates are being used, then a Wildcard cert should be used to achieve this).

Internal DNS Example

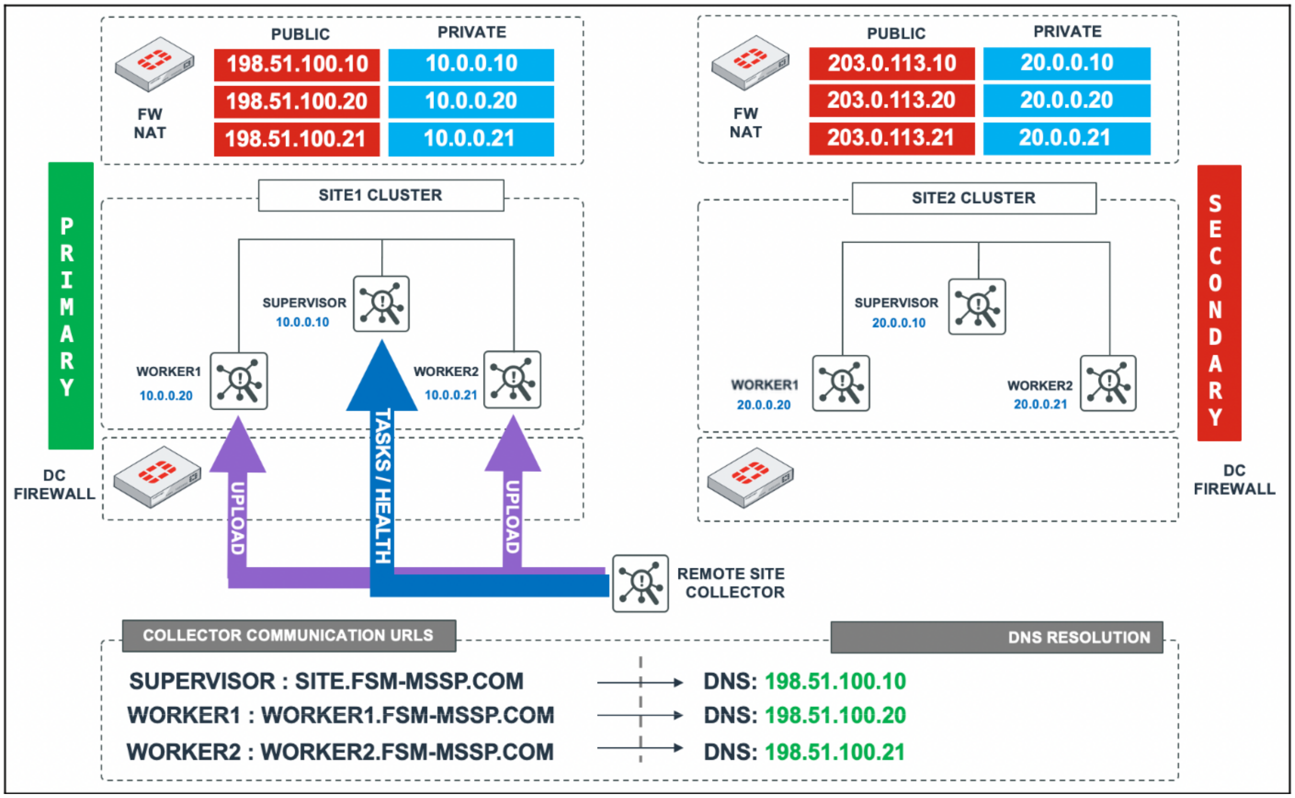
Node	DNS Record Type	Name	IP/Alias
Supervisor (Primary)	A	site1.fsm-mssp.com	10.0.0.10
Supervisor (Secondary)	A	site2.fsm-mssp.com	20.0.0.10
Active Supervisor	CNAME	site.fsm-mssp.com	site1.fsm-mssp.com

By utilizing internal DNS, then SOC operators can always access the active Supervisor GUI via site.fsm-mssp.com, but as will be discussed later, the Secondary Standby Supervisor can always be accessed if required.

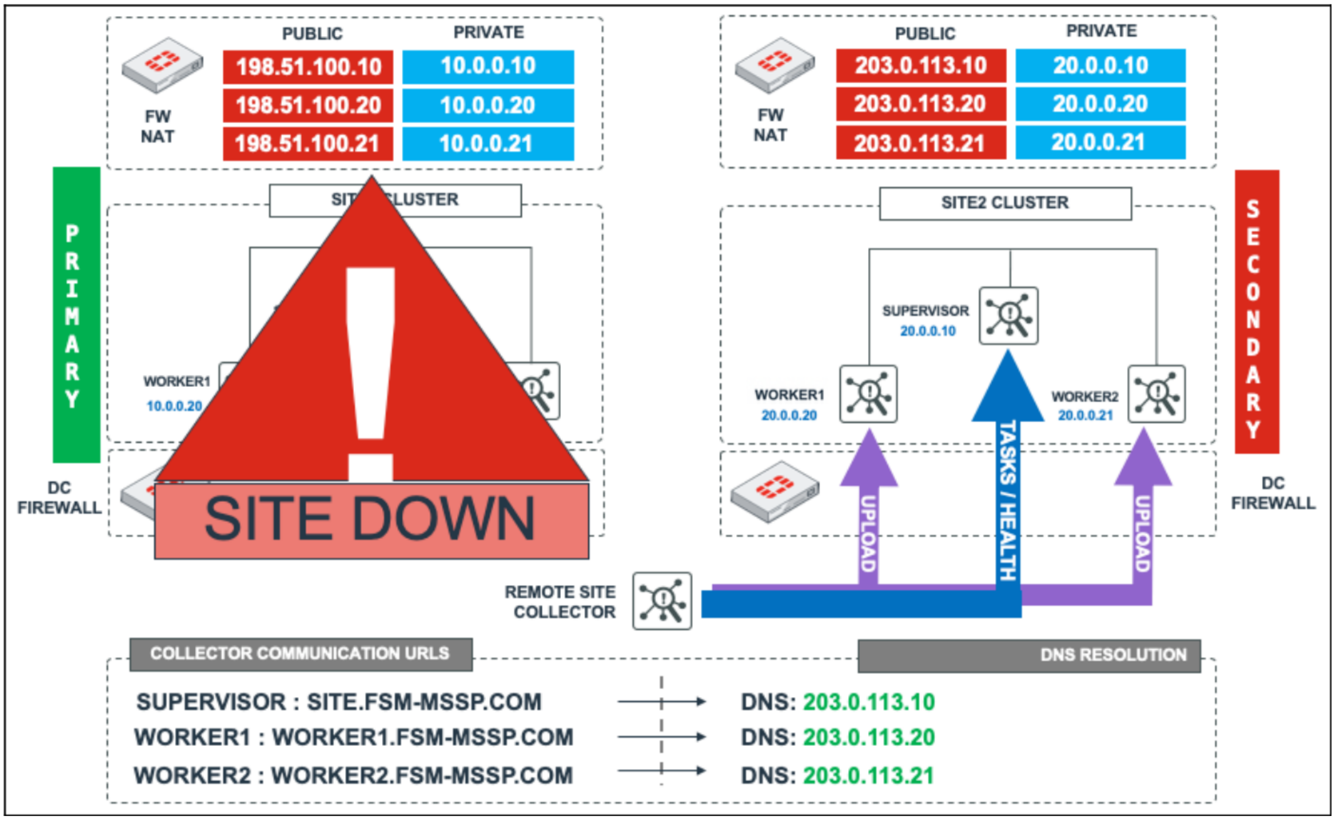

GUI
REPLICATION
Primary Site (Active)
Secondary Site (Standby)
SOC Admins Access
https://site.fsm-mssp.com
And get directed to the Primary Node
SOC Manager Accesses
https://site2.fsm-mssp.com
Specifically to run historical batch queries on the Secondary Node
SOC Staff Location

Note: Any DNS changes, are made MANUALLY in the event of a failover.

As can be seen below, using DNS the Collectors are instructed to talk to the Active site.



And in the event of a failure at the Primary Site, they can be easily instructed to communicate with the Supervisor and Workers at the Secondary site which will be manually switched to be the Primary Role site.



Note : In addition to DNS changes being made manually, the process for promoting the Secondary Supervisor to be the Primary Role Supervisor node is also made manually in the FortiSIEM GUI.

Performing Collector Registration

When registering Collectors, you should ignore the Supervisor-IP requirement, and instead use the CNAME for the Active Supervisor node.

```
[root@collector ~]# phProvisionCollector

Usage: phProvisionCollector --add <Organization-user-name> <Organization-user-password>
      <Supervisor-IP> <Organization-name> <Collector-name>
```

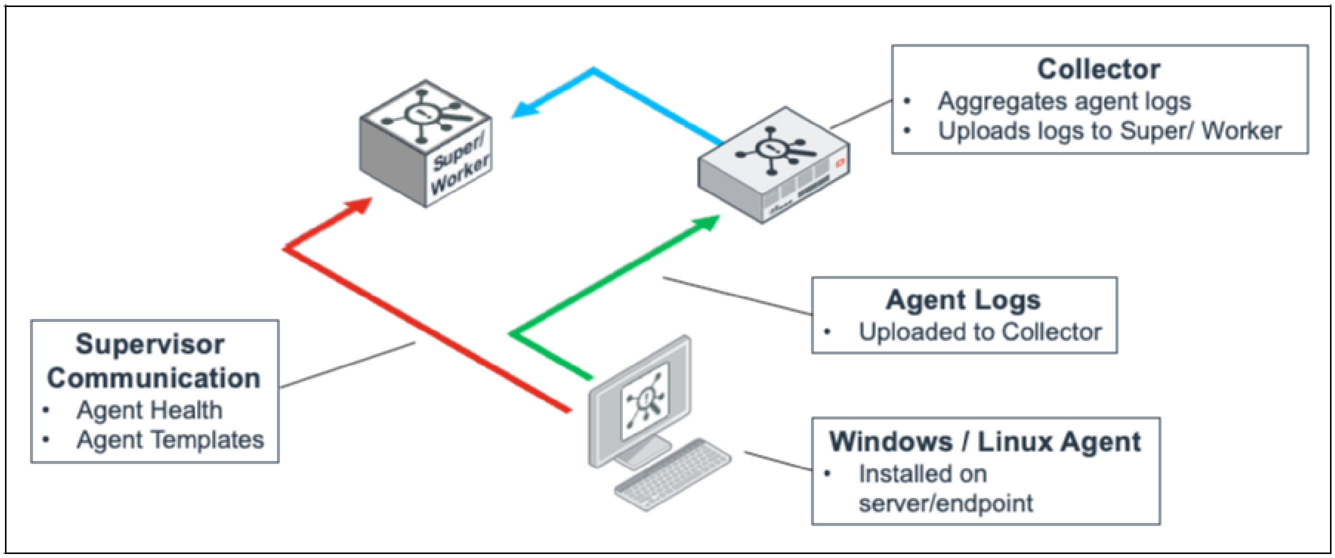
An example using `site.fsm-mssp.com` is shown below. Since Collectors always communicate with the Supervisor node, communications can be easily restored to the Primary via a simple DNS change.

```
[root@collector ~]# phProvisionCollector --add admin admin*1 site.fsm-mssp.com super
      collector.fsm-mssp.com

Continuing to provision the Collector
Adding Collector (collector.fsm-mssp.com) to Super (site.fsm-mssp.com) with Organization
      (super)
This collector is registered successfully, and will be rebooted soon.
```

Agent Communications

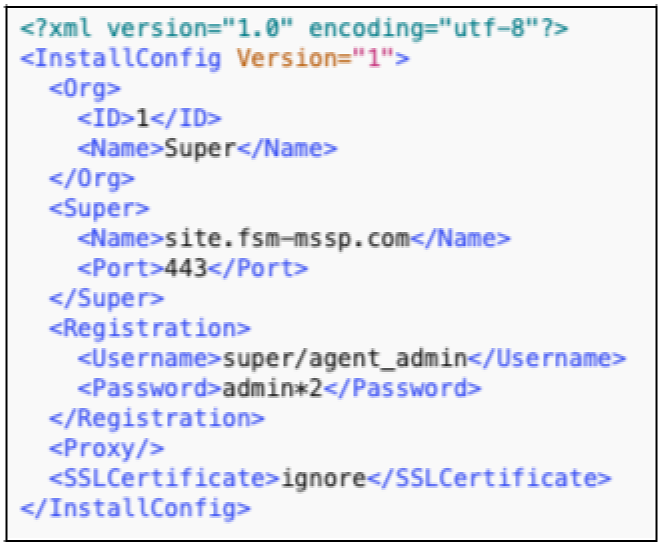
The communications for FortiSIEM Windows and Linux agents follow a similar path to the above. Agents register with the Supervisor node, and maintain this communication to receive updated templates and report health. One or more

Collectors are assigned to each agent as the node or nodes to deliver event data.



For best practice, agent registration should use the Supervisor CNAME. This way, if the Primary Site is a totally destroyed, you can still easily ensure agent communication to the DR site Supervisor via a simple DNS change and still make template changes etc.

The Windows installation file `installSettings.xml` is shown:



```
<?xml version="1.0" encoding="utf-8"?>
<InstallConfig Version="1">
  <Org>
    <ID>1</ID>
    <Name>Super</Name>
  </Org>
  <Super>
    <Name>site.fsm-mssp.com</Name>
    <Port>443</Port>
  </Super>
  <Registration>
    <Username>super/agent_admin</Username>
    <Password>admin*2</Password>
  </Registration>
  <Proxy/>
  <SSLCertificate>ignore</SSLCertificate>
</InstallConfig>
```

The same concept also applies to deploying Linux agents.

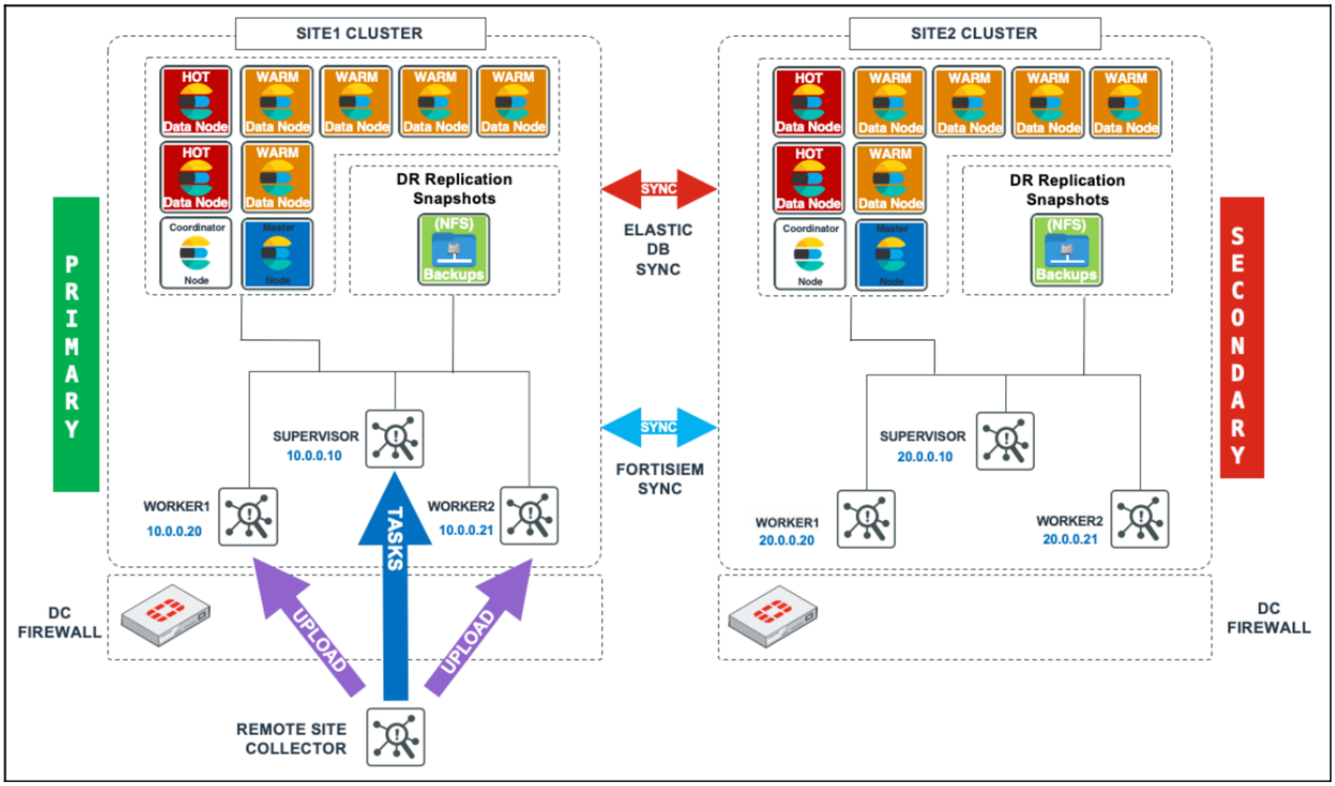
Configuring Elasticsearch for Replication

- Understanding Elasticsearch Replication
- Configuring Elasticsearch for Snapshots
- Defining Elasticsearch in FortiSIEM

Deployments using Elasticsearch are a little more complex than the traditional NoSQL event database. Elasticsearch has different node types in Coordinator, Master, and Data nodes (FortiSIEM supports Hot and Warm) which needs to be an identical setup at both sites.

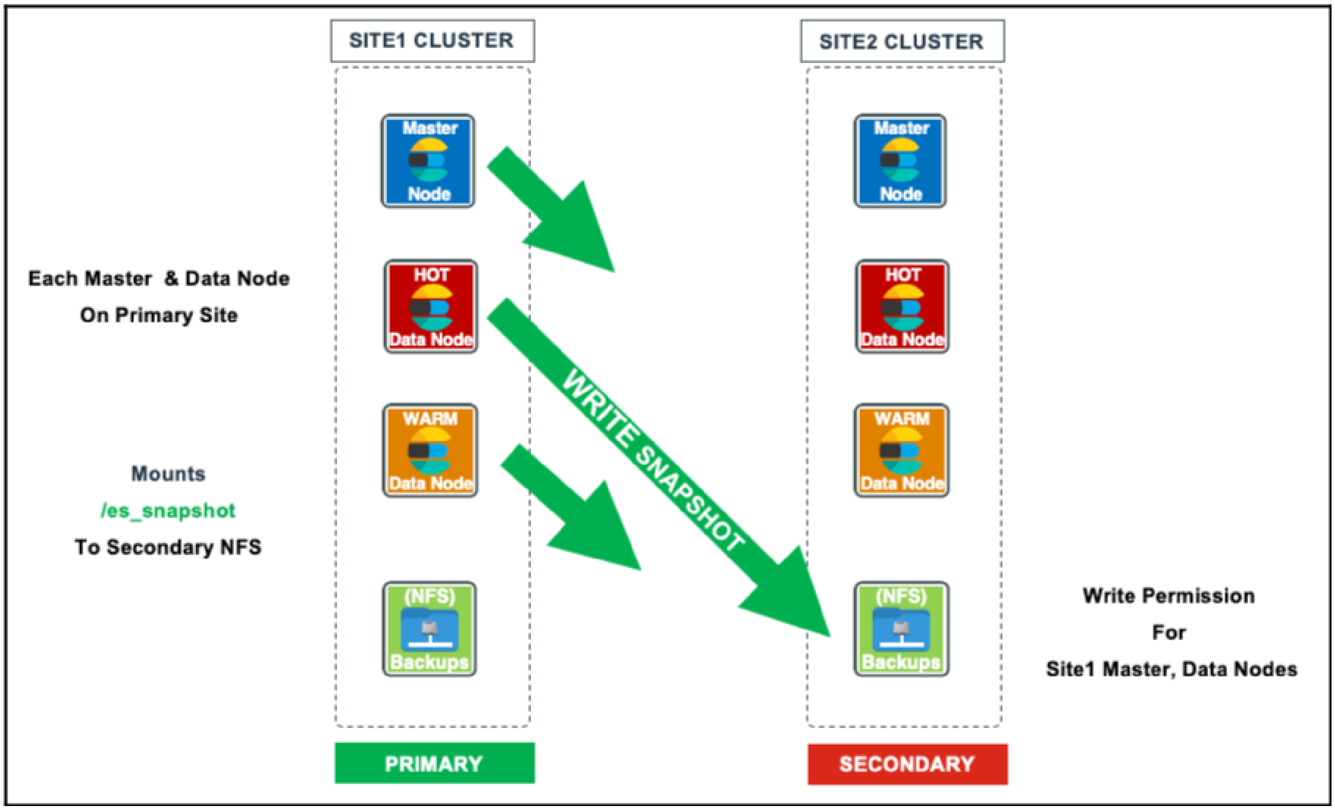
Understanding Elasticsearch Replication

Elasticsearch has its own replication features, known as **snapshots**, built into the database. FortiSIEM takes advantage of these features.

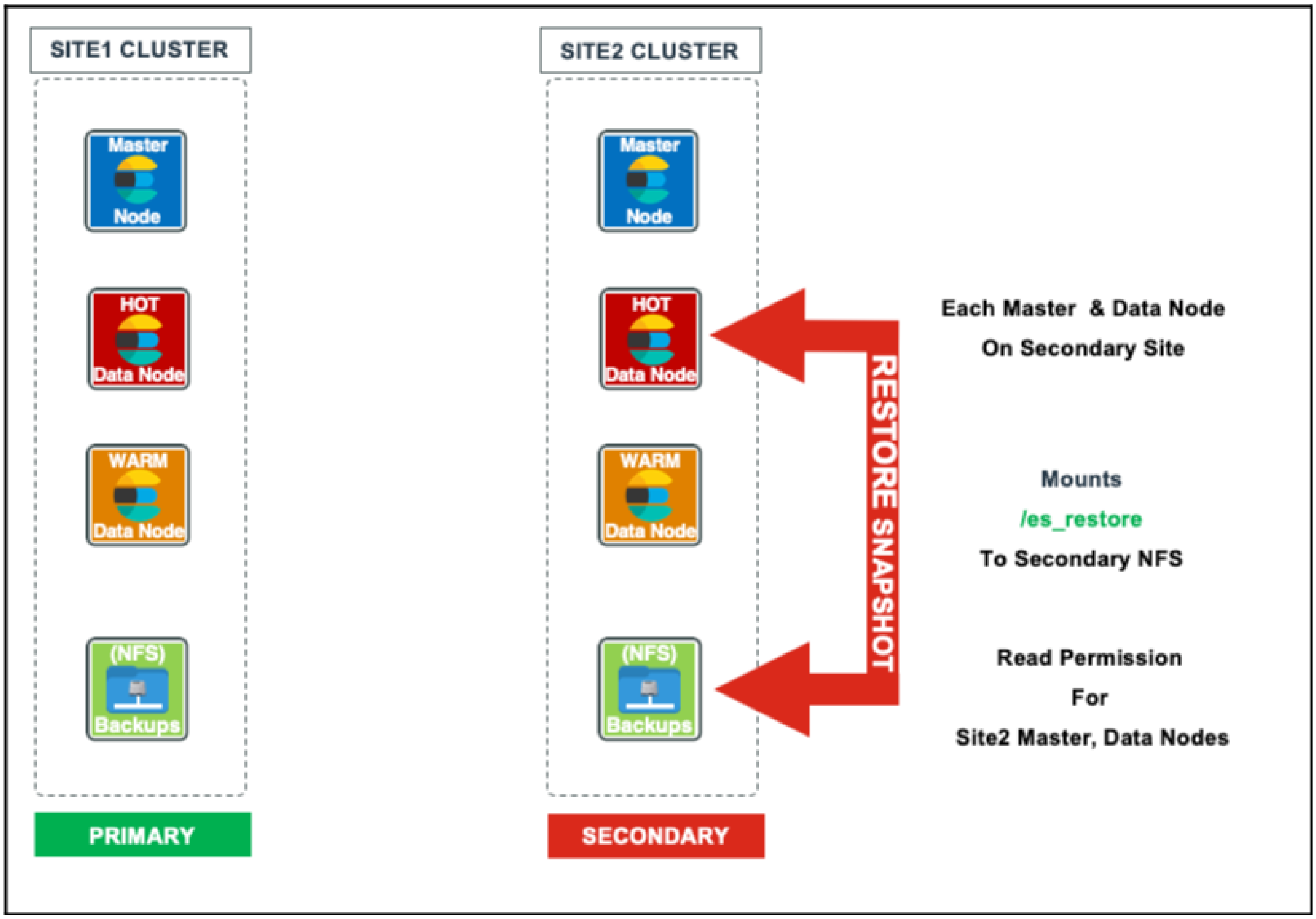


This requires an appropriately sized NFS resource at each site to store the snapshots produced by Elasticsearch.

Each Master and Data node (both Hot and Warm) require two NFS mount points, one to the opposite sites NFS to write the snapshots, and one on the local site to read and restore the snapshot files. FortiSIEM has a snapshot replication timer which instructs Elasticsearch to produce a snapshot on a set internal. Under normal operation the snapshot of the data is taken and written to the Secondary sites NFS share.



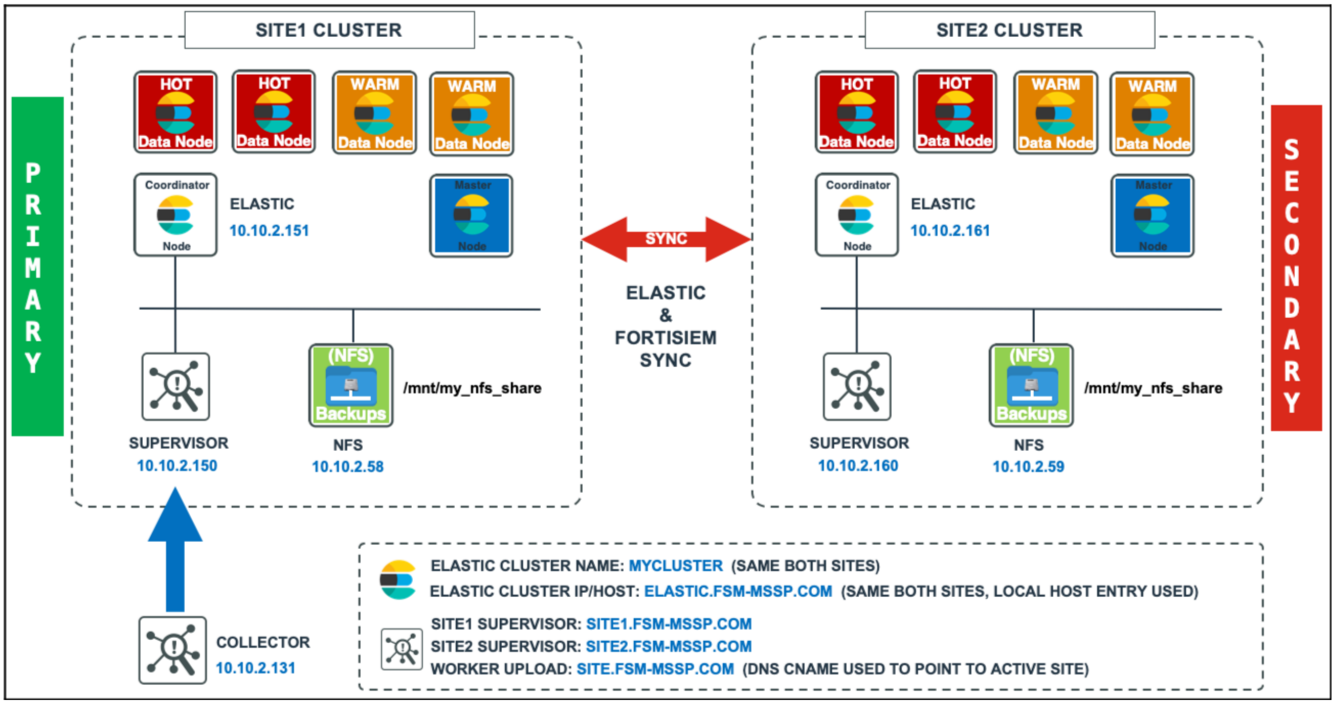
The Secondary site will then read and restore the snapshot into its Elasticsearch database. The snapshot files will be deleted once this is completed.



In the event of a DR failover, this process is reversed, that is, the Primary node (promoted on the Secondary site) will snapshot its data to the NFS on the old Primary site (when the elastic database is operational), and the Secondary nodes will restore.

LAB Used for Testing

The following details for demonstrating Elasticsearch DR used the sample lab and IP addresses:



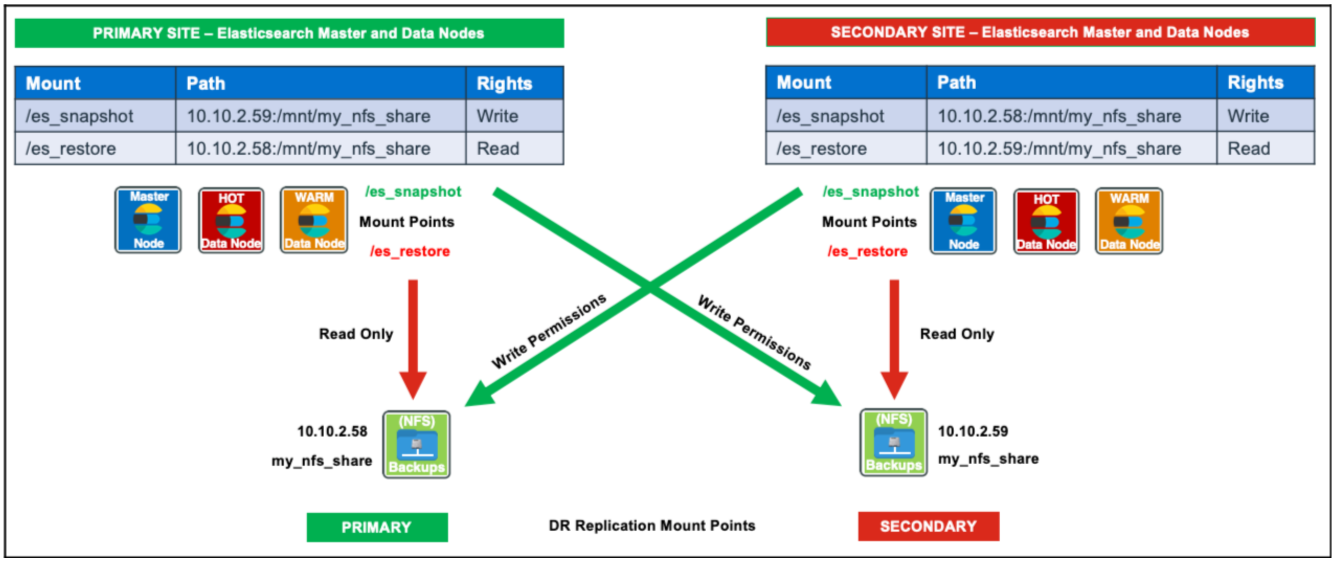
ELASTIC CLUSTER NAME: MYCLUSTER (SAME BOTH SITES)
ELASTIC CLUSTER IP/HOST: ELASTIC.FSM-MSSP.COM (SAME BOTH SITES, LOCAL HOST ENTRY USED)

SITE1 SUPERVISOR: SITE1.FSM-MSSP.COM
SITE2 SUPERVISOR: SITE2.FSM-MSSP.COM
WORKER UPLOAD: SITE.FSM-MSSP.COM (DNS CNAME USED TO POINT TO ACTIVE SITE)

Configuring Elasticsearch for Snapshots

- Primary Elasticsearch Cluster – Master and Data Nodes
- Secondary Elasticsearch Cluster - Master and Data Nodes

On each Master and every Data node two mount points are required, as pictured below. These can be named anything you like, but for the purposes of the diagram below the mounts `es_snapshot` and `es_restore` are used.



PRIMARY SITE – Elasticsearch Master and Data Nodes

Mount	Path	Rights
/es_snapshot	10.10.2.59:/mnt/my_nfs_share	Write
/es_restore	10.10.2.58:/mnt/my_nfs_share	Read

SECONDARY SITE – Elasticsearch Master and Data Nodes

Mount	Path	Rights
/es_snapshot	10.10.2.58:/mnt/my_nfs_share	Write
/es_restore	10.10.2.59:/mnt/my_nfs_share	Read

On each Master and Data nodes use the following commands to mount to the NFS shares. (**Note:** Ubuntu linux was used for Elasticsearch in the examples, you may or not need to use `sudo` to add the commands in your environment) :

Primary Elasticsearch Cluster – Master and Data Nodes

The following is sample code to make a directory for the remote snapshot, mount the local directory, make a directory for the local restore, and mount the local directory:

```
#Make a directory for the remote snapshot
sudo mkdir /es_snapshot

#Mount the local directory es_snapshot to the remote site NFS share
sudo mount -t nfs -o nfsvers=3 10.10.2.59:/mnt/my_nfs_share /es_snapshot

#Make a directory for the local restore
sudo mkdir /es_restore

#Mount the local directory es_restore to the local site NFS share
sudo mount -t nfs -o nfsvers=3 10.10.2.58:/mnt/my_nfs_share /es_restore
```

Also add these entries to the `fstab` file:

```
#Edit the fstab file via
Sudo vi /etc/fstab

#Add the additional entries
10.10.2.59:/mnt/my_nfs_share /es_snapshot nfs defaults 0 0
10.10.2.58:/mnt/my_nfs_share /es_restore nfs defaults 0 0

#Exit and save the file (:wq!)
```

Use the following commands on every Master and Data node to edit the `elasticsearch.yml` file. Add the `repo` path (**Note:** the `rpm` distribution of Elasticsearch was used, your `elasticsearch.yml` file may be in a different location).

```
#Edit the elasticsearch.yml file
sudo vi /etc/elasticsearch/elasticsearch.yml

#Add the entry
path.repo: ["/es_snapshot","/es_restore"]

#Exit and save the file (:wq!)
```

Restart and check the service status of Elasticsearch using the following commands:

```
sudo systemctl restart elasticsearch.service
sudo systemctl status elasticsearch.service
```

Finally, you must tell the Primary site Elasticsearch cluster what the snapshot and restore repo names and storage types are. Use the following CURL command to the Coordinator node:

Snapshot Repo

```
curl -X PUT "10.10.2.151:9200/_snapshot/dr_snapshot_repo" -H 'Content-Type: application/json'
     -d'
{
  "type": "fs",
  "settings": {
     "location": "/es_snapshot",
     "compress": true
  }
}
'
```

Restore Repo

```
curl -X PUT "10.10.2.151:9200/_snapshot/dr_restore_repo" -H 'Content-Type: application/json' -
     d'
{
   "type": "fs",
   "settings": {
      "location": "/es_restore",
      "compress": true
   }
}
'
```

For both the Snapshot and Restore `repo curl` requests, Elasticsearch should respond with the following message:

```
{"acknowledged":true}
```

Secondary Elasticsearch Cluster – Master and Data Nodes

The following is sample code to make a directory for the remote snapshot, mount the local directory, make a directory for the local restore, and mount the local directory:

```
#Make a directory for the remote snapshot
sudo mkdir /es_snapshot

#Mount the local directory es_snapshot to the remote site NFS share
sudo mount -t nfs -o nfsvers=3 10.10.2.58:/mnt/my_nfs_share /es_snapshot

#Make a directory for the local restore
sudo mkdir /es_restore

#Mount the local directory es_restore to the local site NFS share
sudo mount -t nfs -o nfsvers=3 10.10.2.59:/mnt/my_nfs_share /es_restore
```

Also add these entries to the `fstab` file:

```
#Edit the fstab file via
sudo vi /etc/fstab

#Add the additional entries
10.10.2.58:/mnt/my_nfs_share /es_snapshot nfs defaults 0 0
10.10.2.59:/mnt/my_nfs_share /es_restore nfs defaults 0 0

#Exit and save the file (:wq!)
```

Use the following commands on every Master and Data nodes, to edit the `elasticsearch.yml` file and add the `repo` path (**Note:** the `rpm` distribution of Elasticsearch was used, your `elasticsearch.yml` file may be in a different location).

```
#Edit the elasticsearch.yml file
sudo vi /etc/elasticsearch/elasticsearch.yml

#Add the entry
path.repo: ["/es_snapshot","/es_restore"]

#Exit and save the file (:wq!)
```

Use the following commands to restart and check the service status of Elasticsearch:

```
sudo systemctl restart elasticsearch.service
sudo systemctl status elasticsearch.service
```

Finally, you must tell the Secondary site Elasticsearch cluster what the snapshot and restore repo names and storage types are. Use the following CURL command to the Coordinator node:

Snapshot Repo

```
curl -X PUT "10.10.2.161:9200/_snapshot/dr_snapshot_repo" -H 'Content-Type: application/json'
      -d'
{
   "type": "fs",
   "settings": {
      "location": "/es_snapshot",
      "compress": true
   }
}
'
```

Restore Repo

```
curl -X PUT "10.10.2.161:9200/_snapshot/dr_restore_repo" -H 'Content-Type: application/json' -
      d'
{
   "type": "fs",
   "settings": {
      "location": "/es_restore",
      "compress": true
   }
}
'
```

For both the Snapshot and Restore repo curl requests, Elasticsearch should respond with the following message:

```
{"acknowledged":true}
```

Verify the `repo` configuration by issuing the following CURL request to the Primary site Coordinator node and the Secondary site

Coordinator node :

```
#Example shown to Primary site Coordinator curl -XGET "http://10.10.2.151:9200/_snapshot/_
      all?pretty"
{
   "dr_snapshot_repo" : {
      "type" : "fs",
      "settings" : {
         "compress" : "true",
         "location" : "/es_snapshot"
      }
   },
   "dr_restore_repo" : {
      "type" : "fs",
      "settings" : {
         "compress" : "true",
         "location" : "/es_restore"
      }
   }
}
```

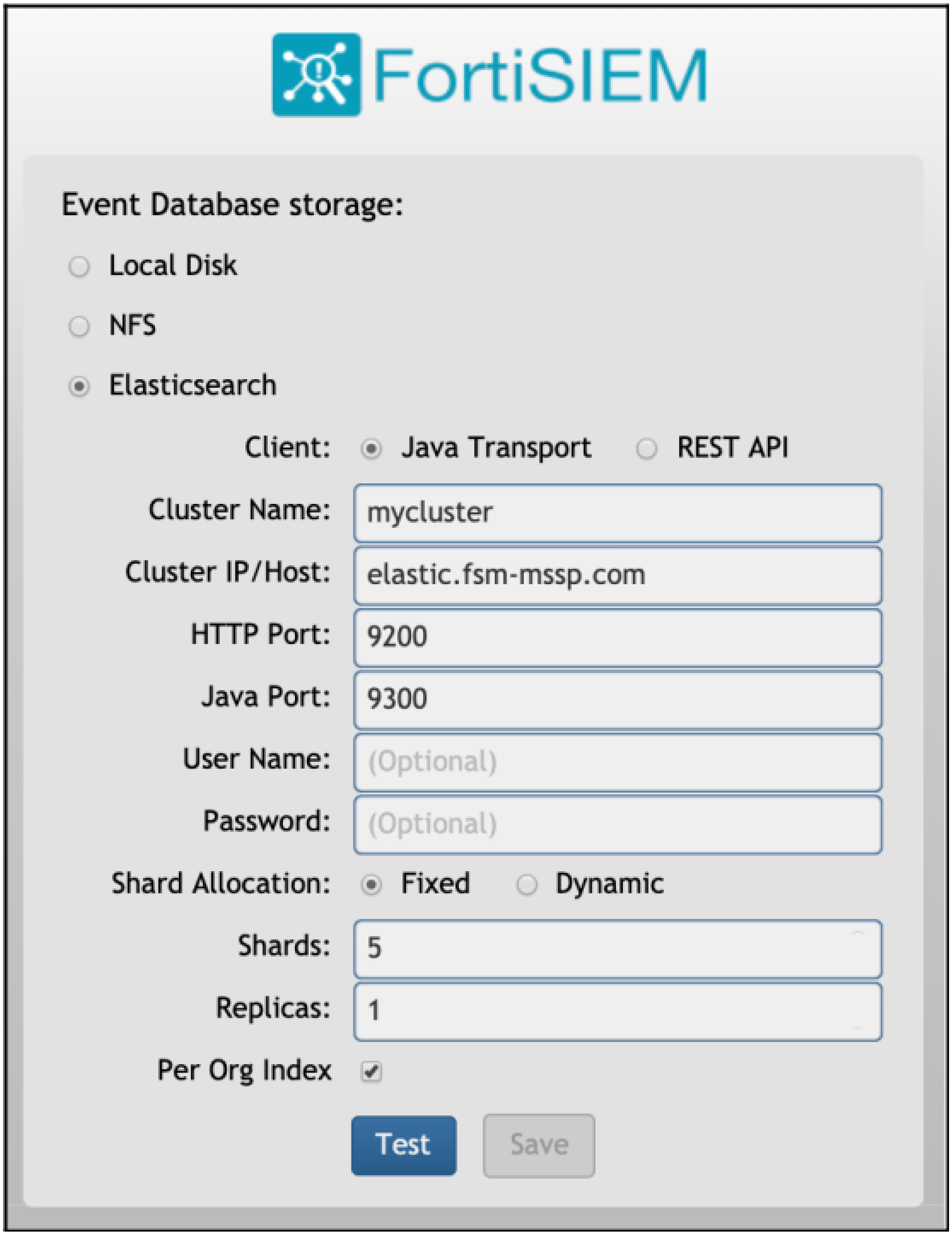
The snapshot configuration in Elasticsearch is now complete.

Defining Elasticsearch in FortiSIEM

Not technically required, but found to be a best practice, add a local host entry for the local site Elasticsearch Coordinator node on the local site Supervisor:

```
#Host Entry for Primary site Coordinator on the Primary Supervisor
10.10.2.151 elastic.fsm-mssp.com

#Host Entry for Secondary site Coordinator on the Secondary Supervisor
10.10.2.161 elastic.fsm-mssp.com
```

The Elasticsearch setup in FortiSIEM, was defined as the following for both Primary site and Secondary site Supervisor nodes:



Event Database storage:

○ Local Disk

○ NFS

● Elasticsearch

Client: ● Java Transport ○ REST API

Cluster Name: mycluster

Cluster IP/Host: elastic.fsm-mssp.com

HTTP Port: 9200

Java Port: 9300

User Name: (Optional)

Password: (Optional)

Shard Allocation: ● Fixed ○ Dynamic

Shards: 5

Replicas: 1

Per Org Index ☑

Configuring Disaster Recovery

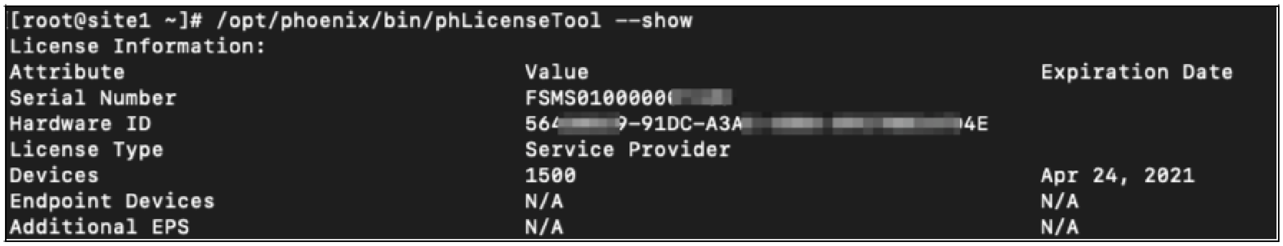
The following sections describe how to configure FortiSIEM primary and secondary nodes for disaster recovery.

- FortiSIEM Primary Node
- FortiSIEM Secondary Node

FortiSIEM Primary Node

On the Primary FortiSIEM node in the GUI:

1. Navigate to **Admin > Settings > Database > Replicate** (or **Replication** in 5.3+).
2. Select **Enable Replication**.
3. For the **Primary**, enter the **Host** and **IP** information.
4. For the **UUID**, obtain the **Hardware ID** value through an **SSH session** on the Primary by entering the following command:
   ```
   /opt/phoenix/bin/phLicenseTool --show
   ```
   For example:



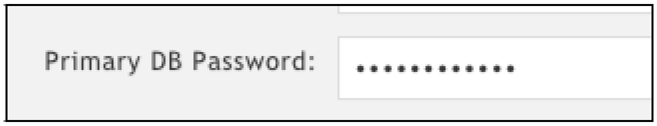
```
[root@site1 ~]# /opt/phoenix/bin/phLicenseTool --show
License Information:
Attribute                      Value                          Expiration Date
Serial Number                  FSMS0100000[redacted]
Hardware ID                    564[redacted]9-91DC-A3A[redacted]4E
License Type                   Service Provider
Devices                        1500                           Apr 24, 2021
Endpoint Devices               N/A                            N/A
Additional EPS                 N/A                            N/A
```

5. For the **CMDB Replication** mount point enter `/something` (this can be any fake mount point). (**Note:** this value is not actually used today).
6. Under **Configuration and Profile Replication**, generate the **SSH Public Key** and **SSH Private Key Path** by entering the following in your SSH session:
   ```
   su - admin
   ssh-keygen -t rsa -b 4096

   #Leave the file location as default, and press enter at the passphrase prompt.
   ```
   The output will appear similar to the following:
   ```
   Generating public/private rsa key pair.
   Enter file in which to save the key (/opt/phoenix/bin/.ssh/id_rsa):
   Created directory '/opt/phoenix/bin/.ssh'.
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /opt/phoenix/bin/.ssh/id_rsa.
   Your public key has been saved in /opt/phoenix/bin/.ssh/id_rsa.pub.
   The key fingerprint is:
   e5:90:b1:3f:a4:f6:b7:7d:f6:8d:e7:0b:0e:97:cf:8b admin@site1.fsm-mssp.com
   The key's randomart image is:
   +--[ RSA 4096]----+
   | ....|
   | . . E. o|
   ```
7. For the SSH Public Key enter the following command, and copy ALL the output into the field:
   ```
   cat /opt/phoenix/bin/.ssh/id_rsa.pub
   ```
8. For the **SSH Private Key Path**, enter the following into the field: `/opt/phoenix/bin/.ssh/id_rsa`.
9. Exit the `admin` user in the SSH session by entering the following command:
   ```
   exit
   ```

10. Select a **Replication Frequency**, with a minimum of 10 minutes.
    Note: This value is used for SVN and ProfileDB synchronization as well as how often Elasticsearch snapshots are taken.
11. Ensure Elasticsearch Snapshot is ticked, to have data replication between Elasticsearch instances, and then complete the following information:

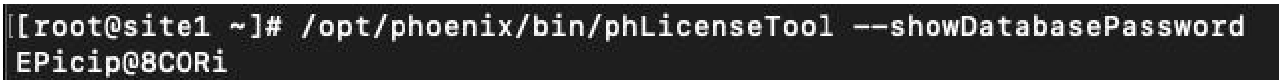
Write Mount Point	/es_snapshot
Read Mount Point	/es_restore
Write Repository	/dr_snapshot_repo
Read Repository	dr_restore_repo

12. Finally, run the following command in the primary SSH session and enter the output under the Role: Secondary, **Primary DB Password** field.
    Note: The **Primary DB Password** field initially looks like it has a populated value. This is **false**, and the following step must be completed.



Primary DB Password: •••••••••••••

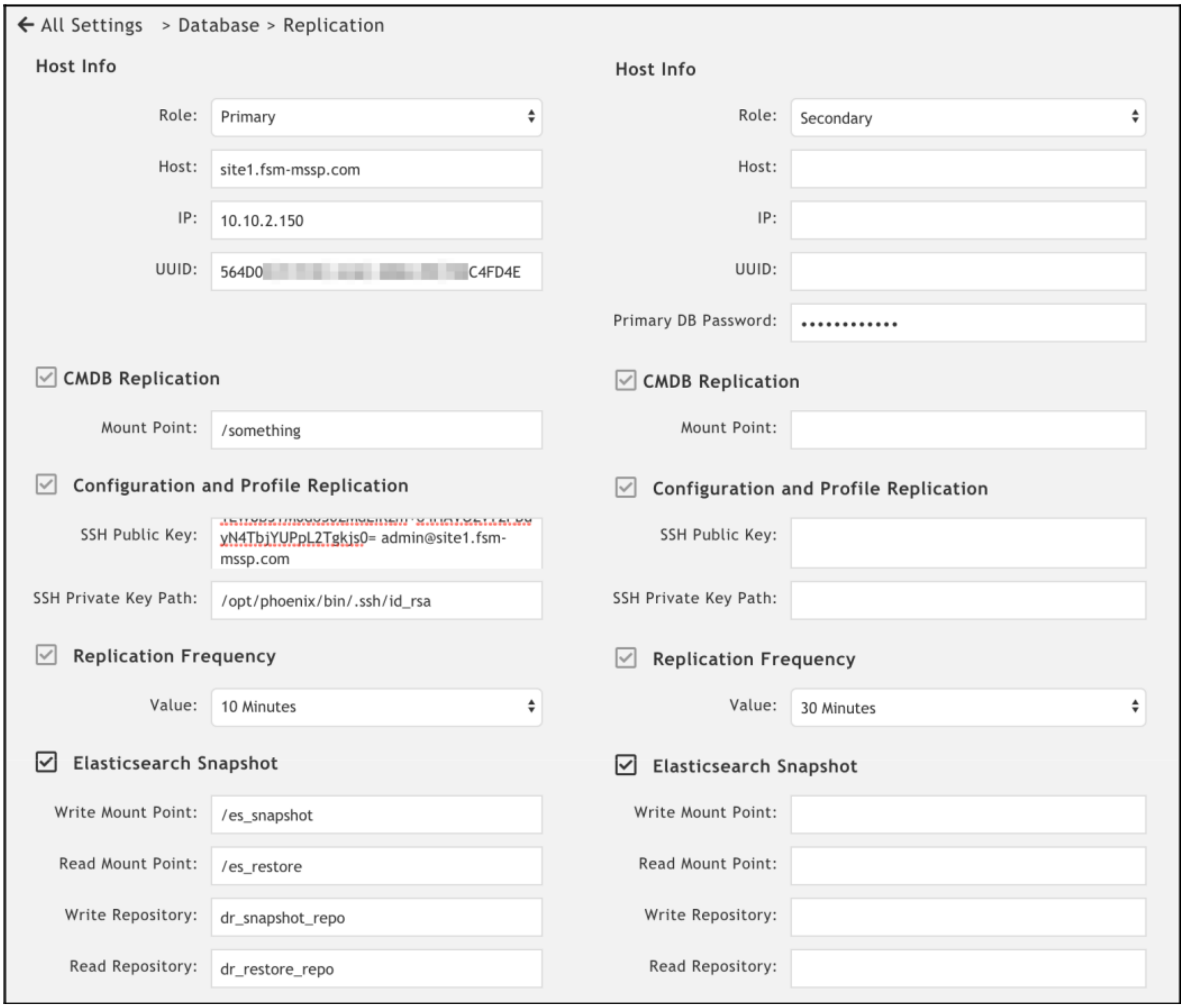
```
/opt/phoenix/bin/phLicenseTool -showDatabasePassword
```



```
[[root@site1 ~]# /opt/phoenix/bin/phLicenseTool --showDatabasePassword
EPicip@8CORi
```

Keep a copy of this password for Step 4 under FortiSIEM Secondary Node.

The completed Primary role details will appear similar to the following:



← All Settings > Database > Replication

Host Info

Role: Primary

Host: site1.fsm-mssp.com

IP: 10.10.2.150

UUID: 564D0[redacted]C4FD4E

☑ CMDB Replication

Mount Point: /something

☑ Configuration and Profile Replication

SSH Public Key: yN4TbjYUPpL2Tgkjs0= admin@site1.fsm-mssp.com

SSH Private Key Path: /opt/phoenix/bin/.ssh/id_rsa

☑ Replication Frequency

Value: 10 Minutes

☑ Elasticsearch Snapshot

Write Mount Point: /es_snapshot

Read Mount Point: /es_restore

Write Repository: dr_snapshot_repo

Read Repository: dr_restore_repo

Host Info

Role: Secondary

Host:

IP:

UUID:

Primary DB Password: ••••••••••••

☑ CMDB Replication

Mount Point:

☑ Configuration and Profile Replication

SSH Public Key:

SSH Private Key Path:

☑ Replication Frequency

Value: 30 Minutes

☑ Elasticsearch Snapshot

Write Mount Point:

Read Mount Point:

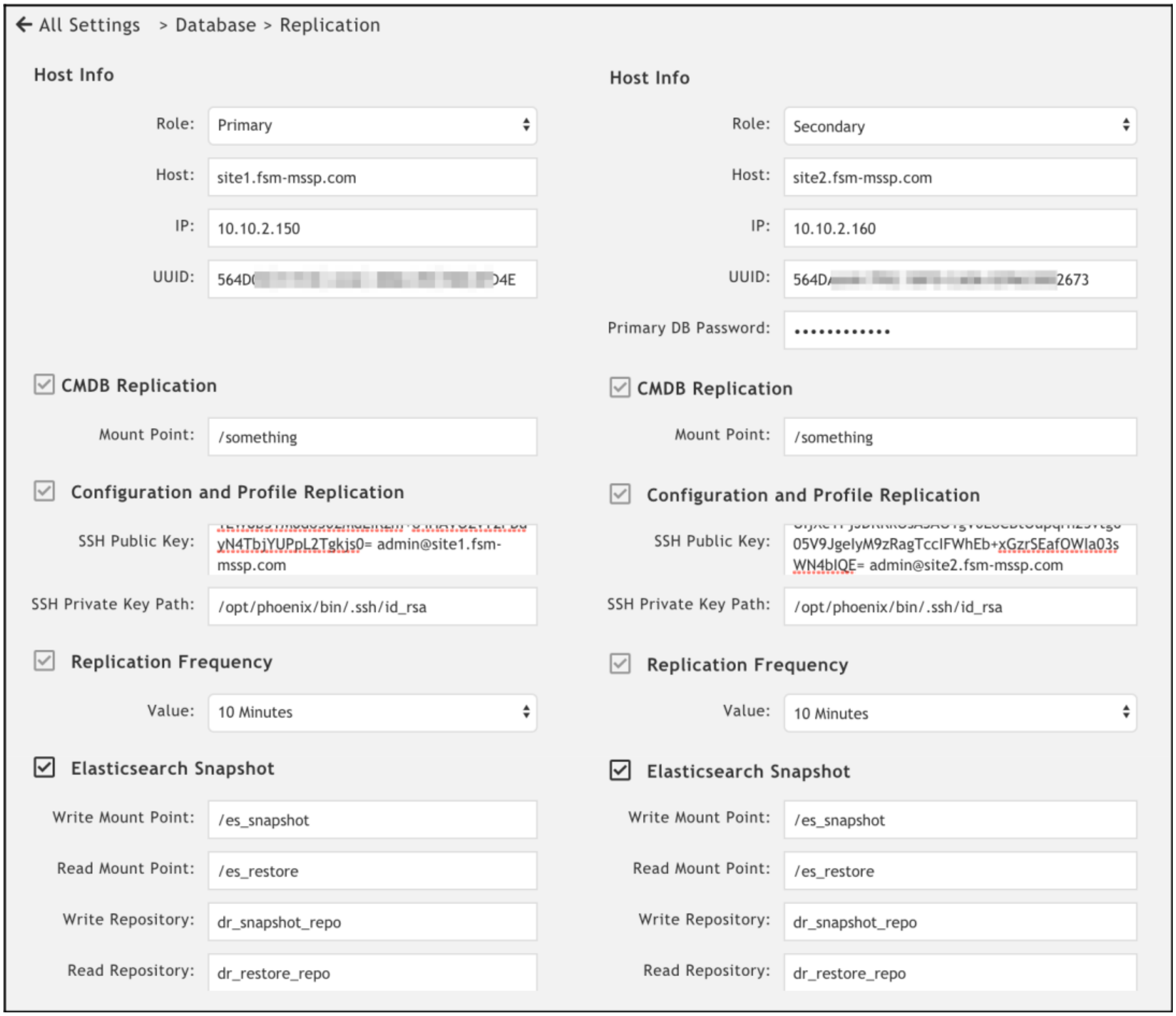
Write Repository:

Read Repository:

Now move on to configuring the Secondary nodes details.

13. For the **Secondary**, enter the **Host** and **IP** information.
14. For the **UUID**, obtain the **Hardware ID** value through an SSH session on the secondary node by entering the following command:
    ```
    /opt/phoenix/bin/phLicenseTool --show
    ```
15. For the **CMDB Replication** mount point enter `/something` (this can be any fake mount point). **Note:** this value is not actually used today.
16. Under **Configuration and Profile Replication**, generate the **SSH Public Key** and **SSH Private Key Path** by entering the following in your SSH session on your secondary node:
    ```
    su - admin
    ssh-keygen -t rsa -b 4096

    #Leave the file location as default, and press enter at the passphrase prompt.
    ```
17. For the **SSH Public Key** enter the following command, and copy **all** of the output into the field:
    ```
    cat /opt/phoenix/bin/.ssh/id_rsa.pub
    ```
18. For the **SSH Private Key Path**, enter the following into the field: `/opt/phoenix/bin/.ssh/id_rsa`.

19. Exit the admin user in the SSH session by entering the following command:
    `exit`
20. Select the same **Replication Frequency** and **Elasticsearch Snapshot** details as were set on the Primary node.



← All Settings > Database > Replication

Host Info

Role: Primary
Host: site1.fsm-mssp.com
IP: 10.10.2.150
UUID: 564D[redacted]4E

☑ CMDB Replication
Mount Point: /something

☑ Configuration and Profile Replication
SSH Public Key: yN4TbjYUPpL2Tgkjs0= admin@site1.fsm-mssp.com
SSH Private Key Path: /opt/phoenix/bin/.ssh/id_rsa

☑ Replication Frequency
Value: 10 Minutes

☑ Elasticsearch Snapshot
Write Mount Point: /es_snapshot
Read Mount Point: /es_restore
Write Repository: dr_snapshot_repo
Read Repository: dr_restore_repo

Host Info

Role: Secondary
Host: site2.fsm-mssp.com
IP: 10.10.2.160
UUID: 564D[redacted]2673
Primary DB Password: ••••••••••••

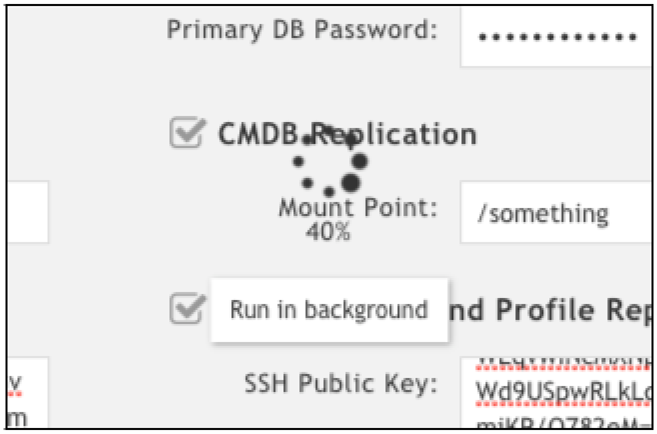
☑ CMDB Replication
Mount Point: /something

☑ Configuration and Profile Replication
SSH Public Key: 05V9JgelyM9zRagTccIFWhEb+xGzrSEafOWla03sWN4bIQE= admin@site2.fsm-mssp.com
SSH Private Key Path: /opt/phoenix/bin/.ssh/id_rsa

☑ Replication Frequency
Value: 10 Minutes

☑ Elasticsearch Snapshot
Write Mount Point: /es_snapshot
Read Mount Point: /es_restore
Write Repository: dr_snapshot_repo
Read Repository: dr_restore_repo

21. Click **Export** and download a file named `replicate.json`. **Note:** This file contains all of the DR settings, except the Primary DB Password.
22. Click **Apply**.
    Note: This should result in the following message in the GUI, where it will stick at 40% until the Secondary node configuration is completed.



Primary DB Password: ············

☑ CMDB Replication

Mount Point: /something

40%

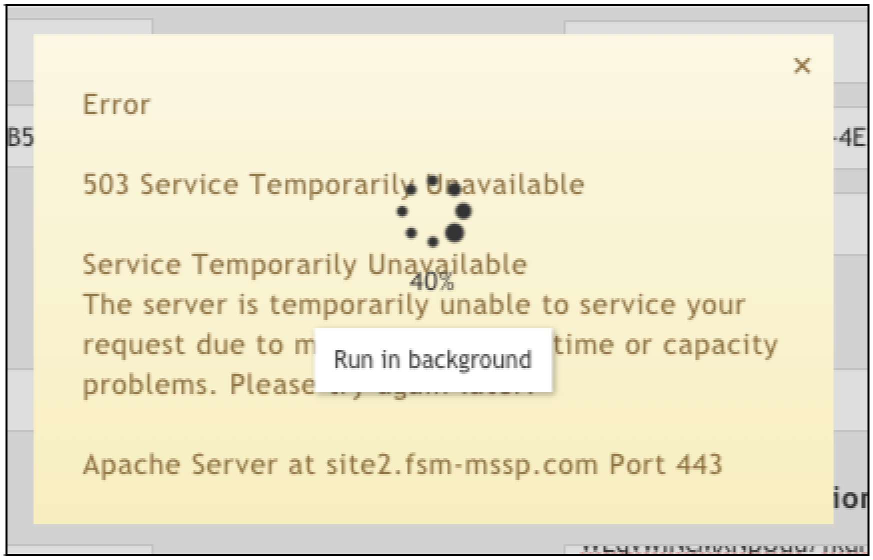
☑ Run in background nd Profile Rep

SSH Public Key: Wd9USpwRLkLc

FortiSIEM Secondary Node

On the Secondary FortiSIEM node, log into the FortiSIEM GUI:

1. Navigate to **Admin > Settings > Database > Replicate** (or **Replication** in 5.3+).
2. Select **Enable Replication**.
3. Click **Import**, and select the `replicate.json` file downloaded from the Primary node.
4. Copy the **Primary DB Password**, from Step 12 in FortiSIEM Primary Node.
   If you do not have the password handy, run the following command on the Primary nodes SSH session and enter the output under the **Primary DB Password** field.
   ```
   #On the PRIMARY node
   /opt/phoenix/bin/phLicenseTool -showDatabasePassword
   ```
5. Click **Apply**.
   At this point, the Secondary node will display the following while the backend scripts are disabling services, etc.

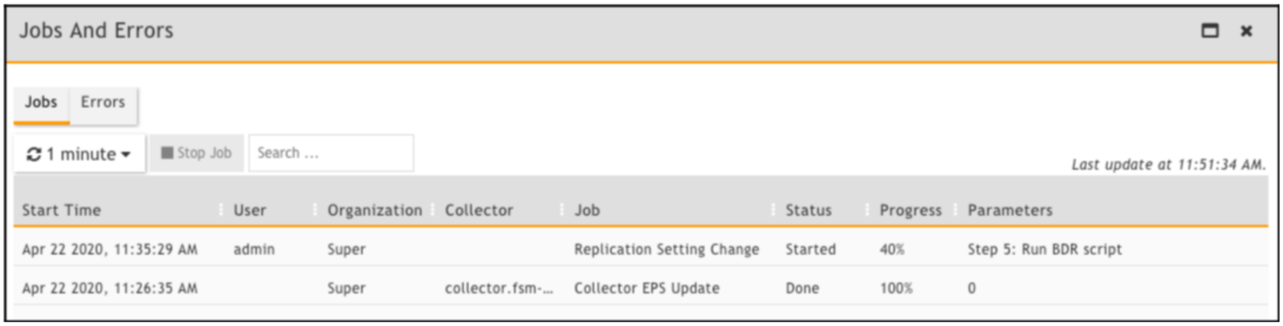


```
Error

503 Service Temporarily Unavailable

Service Temporarily Unavailable
The server is temporarily unable to service your
request due to m                  time or capacity
problems. Please

Apache Server at site2.fsm-mssp.com Port 443
```

40%

Run in background

Note: There will be disruption of services on both nodes, while the setup is taking place behind the scenes. While initial replication is taking place, you can view the status on the Primary node, Jobs, and Errors (Red Alert Symbol, top right of GUI) on what Step (out of 10) the process is currently at.



Last update at 11:51:34 AM.

Start Time	User	Organization	Collector	Job	Status	Progress	Parameters
Apr 22 2020, 11:35:29 AM	admin	Super		Replication Setting Change	Started	40%	Step 5: Run BDR script
Apr 22 2020, 11:26:35 AM		Super	collector.fsm-...	Collector EPS Update	Done	100%	0

Backend logs will better display the current status of the replication and DR scripts being run.

Troubleshooting Disaster Recovery Setup

- Backend Logs
- Alternative Logs
- FortiSIEM Services Status on Primary and Secondary Node
- Understanding FortiSIEM Operations in DR Mode
- Verify Elasticsearch Snapshots for Data Replication

Backend Logs

On both the Primary and Secondary nodes, use the `cat` or `tail -f` command to view the backend logs:

`/opt/phoenix/config/pgMasterRep/bdrlog`

Note: This process can take a while. The output below was a new installation with minimal test data and it took around 5 minutes to complete, for a live system it will take a lot longer. (It is recommended to `tail -f` the log).

Successful Enablement of Disaster Recovery on the Primary node

```
[root@site1 ~]# cat /opt/phoenix/config/pgMasterRep/bdrlog
bdr_connection_count for 10.10.2.150 is
back up pg_hba.conf and postgresql.conf
setting bdr configuration ...
inserting pg_hba records ...
finished setting bdr configuration
restart postgresql9.4
ext_btree_gist_count is 0
ext_bdr_count is 0
bdr_node1_count is 0
please wait the bdr building ...
no primary file exist, add primary file
Waiting for Secondary 10.10.2.160 to finish up synch Primary CMDB
Waiting for Secondary 10.10.2.160 to finish up synch Primary CMDB
Waiting for Secondary 10.10.2.160 to finish up synch Primary CMDB
Waiting for Secondary 10.10.2.160 to finish up synch Primary CMDB
Waiting for Secondary 10.10.2.160 to finish up synch Primary CMDB
Waiting for Secondary 10.10.2.160 to finish up synch Primary CMDB
Waiting for Secondary 10.10.2.160 to finish up synch Primary CMDB
Waiting for Secondary 10.10.2.160 to finish up synch Primary CMDB
Secondary 10.10.2.160 finished synch Primary CMDB
```

Successful Enablement of Disaster Recovery on the Secondary node

```
[root@site2 ~]# cat /opt/phoenix/config/pgMasterRep/bdrlog
slave - bdr_connection_count for 10.10.2.150 is
Backup unsynchable system properties from ph_sys_conf before replicating CMDB from Primary
     CMDB
dump file ph_sys_server.sql and ph_sys_conf.sql ...
Shutdown App Server to preparing synch CMDB from primary
Stopping crond: [ OK ]
Stopping postgresql-9.4 service: [ OK ]
wait port 5432 to stop...
port 5432 stopped
join connection according cmdb buffer ... master ip = 10.10.2.150, slave ip = 10.10.2.160
bdr_init_copy: starting ...

Getting remote server identification ...
Detected 1 BDR database(s) on remote server
Updating BDR configuration on the remote node:
  phoenixdb: creating replication slot ...
  phoenixdb: creating node entry for local node ...
Creating base backup of the remote node...
149666/149666 kB (100%), 1/1 tablespace
Creating restore point on remote node ...
Bringing local node to the restore point ...
Transaction log reset
Initializing BDR on the local node:
  phoenixdb: adding the database to BDR cluster ...
All done
please wait the connection building ...
synching CMDB from Primary, status= c
Done synching CMDB from Primary
DELETE 1
DELETE 8
DELETE 0
DELETE 55
DELETE 2
ERROR: relation "ph_sys_collector_trail" does not exist
LINE 1: delete from ph_sys_collector_trail

import sql ph_sys_server.sql ...
COPY 1
COPY 1
Restoring non-replicable system properties
COPY 3
Stop running all quartz jobs on secondary
restart App Server ...
Starting crond: [ OK ]
ALTER ROLE
Done replication CMDB
```

Alternative Logs

It is also possible to track the DR scripts by examining the `phoenix.log` file. Use the grep command on both Primary and Secondary nodes to track progress.

```
grep "521-ReplicationRoleChange" /opt/phoenix/log/phoenix.log
2020-05-03T22:34:44.273567+02:00 site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=6903,[phLogDetail]=521-
      ReplicationRoleChange, Step 1.1: check command type 2020-05-03T22:34:44.273830+02:00
      site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=6912,[phLogDetail]=521-
      ReplicationRoleChange, Step 1.2: check command data 2020-05-03T22:34:44.274070+02:00
      site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=6919,[phLogDetail]=521-
      ReplicationRoleChange, Step 2: load replication setting 2020-05-03T22:34:44.328209+02:00
      site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=6934,[phLogDetail]=521-
      ReplicationRoleChange, Step 3: handle replication role change 2020-05-
      03T22:34:44.329293+02:00 site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=6953,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.1: handle replication role change on super
2020-05-03T22:34:44.329407+02:00 site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=6956,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.2: prepare role info 2020-05-03T22:34:44.370127+02:00 site1
      phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=6979,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.3: update SSH keys 2020-05-03T22:34:44.453706+02:00 site1
      phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=6992,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.4: update SSH configurations
2020-05-03T22:34:44.513790+02:00 site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=7007,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.5: run database replication script 2020-05-
      03T22:41:30.363477+02:00 site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=7029,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.6: wait appsvr back 2020-05-03T22:41:30.455080+02:00 site1
      phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=7038,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.7: update service and SVN password for the first time 2020-
      05-03T22:41:31.002774+02:00 site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=7235,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.7.1: get sevice user 2020-05-03T22:41:31.002985+02:00 site1
      phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=7243,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.7.2: get secondary host 2020-05-03T22:41:31.012831+02:00
      site1 phMonitorSupervisor[14092]:
```

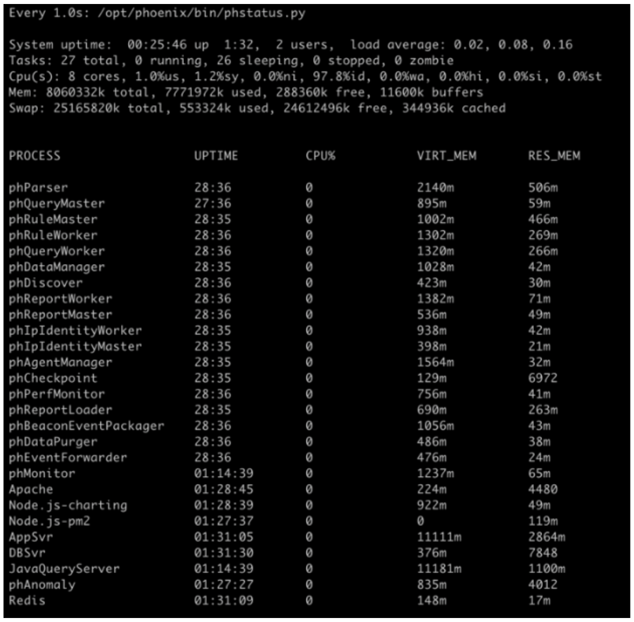
```
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=7262,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.7.3: update secondary 2020-05-03T22:41:31.177562+02:00
      site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=7050,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.8: restart processes on super
2020-05-03T22:41:31.236232+02:00 site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=7058,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.9: notify processes on super 2020-05-
      03T22:41:31.326463+02:00 site1 phMonitorSupervisor[14092]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phMonitorSupervisor,
      [fileName]=phMonitorProcess.cpp,[lineNumbe r]=7068,[phLogDetail]=521-
      ReplicationRoleChange, Step 3.10: finish role change on super
```

FortiSIEM Services Status on Primary and Secondary Node

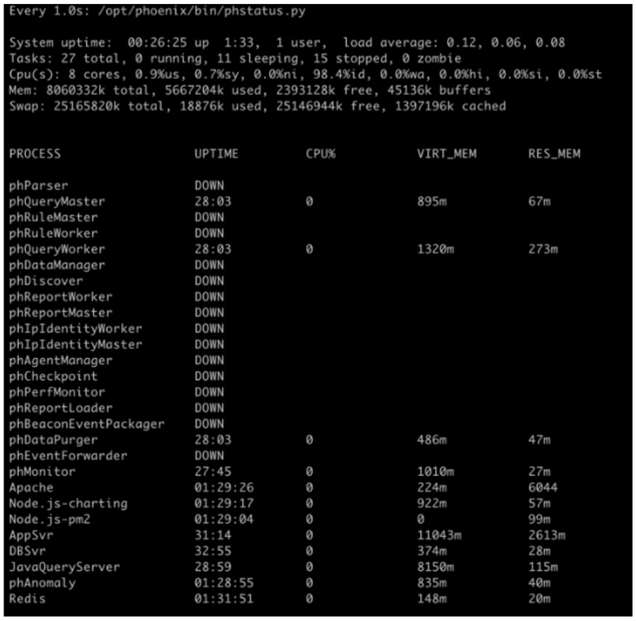
On the Primary node, all FortiSIEM `ph*` services will be in an "up" state. (They will all restart, but it may take up to 3 to 5 minutes to restart.)

On the Secondary node, most `ph*` services will be "down" except for `phQueryMaster`, `phQueryWorker`, `phDataPurger`, and `phMonitor`.

This can be seen in the following images. They illustrate the Primary Node and Secondary Node after a full CMDB sync:



```
Every 1.0s: /opt/phoenix/bin/phstatus.py

System uptime:  00:25:46 up  1:32,  2 users,  load average: 0.02, 0.08, 0.16
Tasks: 27 total, 0 running, 26 sleeping, 0 stopped, 0 zombie
Cpu(s): 8 cores, 1.0%us, 1.2%sy, 0.0%ni, 97.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 8060332k total, 7771972k used, 288360k free, 11600k buffers
Swap: 25165820k total, 553324k used, 24612496k free, 344936k cached


PROCESS                  UPTIME         CPU%           VIRT_MEM       RES_MEM

phParser                 28:36          0              2140m          506m
phQueryMaster            27:36          0              895m           59m
phRuleMaster             28:35          0              1002m          466m
phRuleWorker             28:36          0              1302m          269m
phQueryWorker            28:36          0              1320m          266m
phDataManager            28:35          0              1028m          42m
phDiscover               28:36          0              423m           30m
phReportWorker           28:36          0              1382m          71m
phReportMaster           28:36          0              536m           49m
phIpIdentityWorker       28:35          0              938m           42m
phIpIdentityMaster       28:35          0              398m           21m
phAgentManager           28:35          0              1564m          32m
phCheckpoint             28:35          0              129m           6972
phPerfMonitor            28:36          0              756m           41m
phReportLoader           28:35          0              690m           263m
phBeaconEventPackager    28:36          0              1056m          43m
phDataPurger             28:36          0              486m           38m
phEventForwarder         28:36          0              476m           24m
phMonitor                01:14:39       0              1237m          65m
Apache                   01:28:45       0              224m           4480
Node.js-charting         01:28:39       0              922m           49m
Node.js-pm2              01:27:37       0              0              119m
AppSvr                   01:31:05       0              11111m         2864m
DBSvr                    01:31:30       0              376m           7848
JavaQueryServer          01:14:39       0              11181m         1100m
phAnomaly                01:27:27       0              835m           4012
Redis                    01:31:09       0              148m           17m
```



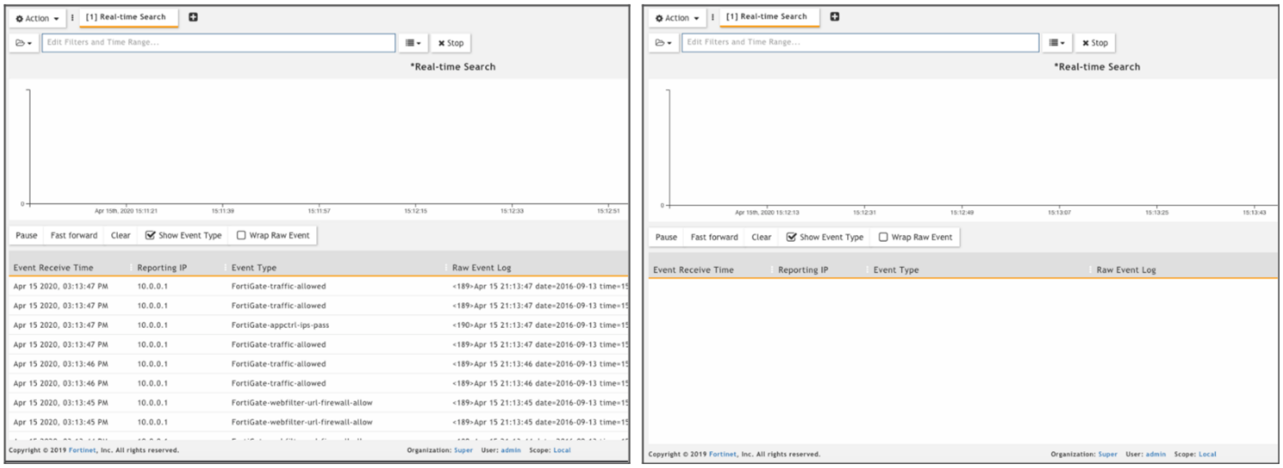
```
Every 1.0s: /opt/phoenix/bin/phstatus.py

System uptime:  00:26:25 up  1:33,  1 user,  load average: 0.12, 0.06, 0.08
Tasks: 27 total, 0 running, 11 sleeping, 15 stopped, 0 zombie
Cpu(s): 8 cores, 0.9%us, 0.7%sy, 0.0%ni, 98.4%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 8060332k total, 5667204k used, 2393128k free, 45136k buffers
Swap: 25165820k total, 18876k used, 25146944k free, 1397196k cached


PROCESS                  UPTIME         CPU%           VIRT_MEM       RES_MEM

phParser                 DOWN
phQueryMaster            28:03          0              895m           67m
phRuleMaster             DOWN
phRuleWorker             DOWN
phQueryWorker            28:03          0              1320m          273m
phDataManager            DOWN
phDiscover               DOWN
phReportWorker           DOWN
phReportMaster           DOWN
phIpIdentityWorker       DOWN
phIpIdentityMaster       DOWN
phAgentManager           DOWN
phCheckpoint             DOWN
phPerfMonitor            DOWN
phReportLoader           DOWN
phBeaconEventPackager    DOWN
phDataPurger             28:03          0              486m           47m
phEventForwarder         DOWN
phMonitor                27:45          0              1010m          27m
Apache                   01:29:26       0              224m           6044
Node.js-charting         01:29:17       0              922m           57m
Node.js-pm2              01:29:04       0              0              99m
AppSvr                   31:14          0              11043m         2613m
DBSvr                    32:55          0              374m           28m
JavaQueryServer          28:59          0              8150m          115m
phAnomaly                01:28:55       0              835m           40m
Redis                    01:31:51       0              148m           20m
```

Understanding FortiSIEM Operations in DR Mode

When operating in DR Replication mode, there are a few things to bear in mind:

- Both the Primary and Secondary nodes GUI are available for login.
- The CMDB is set in a multi-master mode, so any changes on the Secondary are replicated over to the Primary.
- Although the CMDB can be edited from either site, it is recommended to do all edits on the Primary site.
- Analytical queries and reports can be run from either node.
- Performing Real-Time queries: You will see results only on the Primary node, as this is done in memory before storage.

  Primary vs Secondary – Real-Time Search

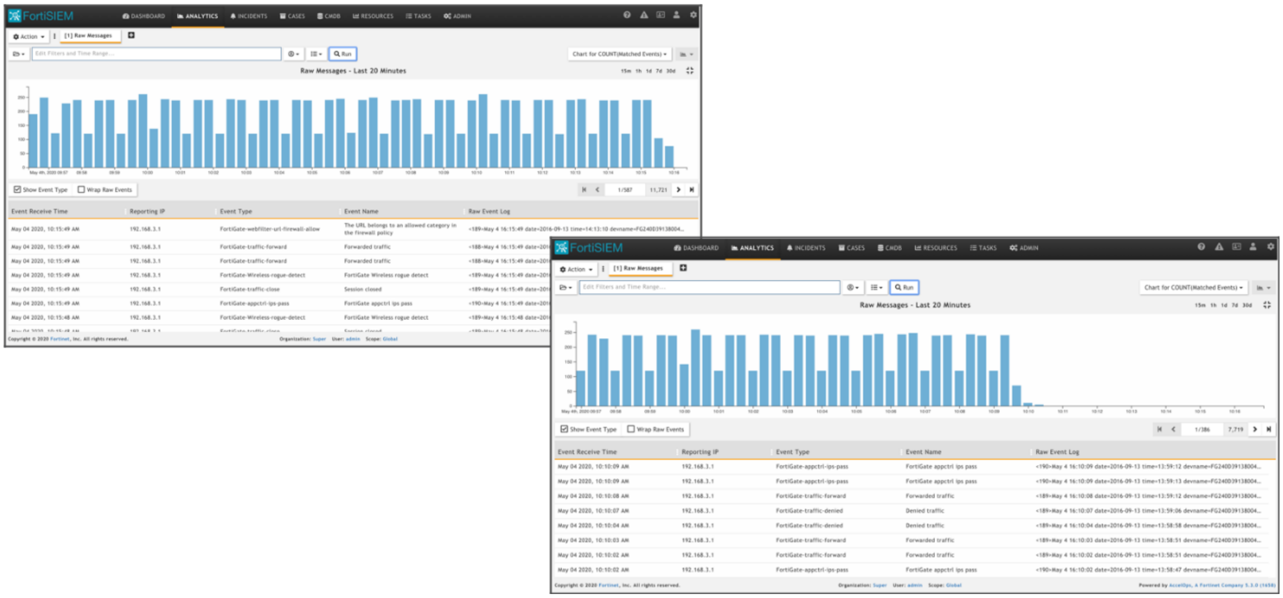


Event Receive Time	Reporting IP	Event Type	Raw Event Log
Apr 15 2020, 03:13:47 PM	10.0.0.1	FortiGate-traffic-allowed	<189>Apr 15 21:13:47 date=2016-09-13 time=15
Apr 15 2020, 03:13:47 PM	10.0.0.1	FortiGate-traffic-allowed	<189>Apr 15 21:13:47 date=2016-09-13 time=15
Apr 15 2020, 03:13:47 PM	10.0.0.1	FortiGate-appctrl-ips-pass	<190>Apr 15 21:13:47 date=2016-09-13 time=15
Apr 15 2020, 03:13:47 PM	10.0.0.1	FortiGate-traffic-allowed	<189>Apr 15 21:13:47 date=2016-09-13 time=15
Apr 15 2020, 03:13:46 PM	10.0.0.1	FortiGate-traffic-allowed	<189>Apr 15 21:13:46 date=2016-09-13 time=15
Apr 15 2020, 03:13:46 PM	10.0.0.1	FortiGate-traffic-allowed	<189>Apr 15 21:13:46 date=2016-09-13 time=15
Apr 15 2020, 03:13:45 PM	10.0.0.1	FortiGate-webfilter-url-firewall-allow	<189>Apr 15 21:13:45 date=2016-09-13 time=15
Apr 15 2020, 03:13:45 PM	10.0.0.1	FortiGate-webfilter-url-firewall-allow	<189>Apr 15 21:13:45 date=2016-09-13 time=15

Event Receive Time	Reporting IP	Event Type	Raw Event Log

- Performing Historical Queries: Bear in mind the data on the Secondary node will be slightly out of date, dependent upon how much data is being replicated **and what the replication frequency is**, but this is ideal for running large complex queries on the Secondary without impacting the Primary's performance. Bear in mind this will also be the same for Triggering Events on the Incident Tab.

  Primary vs Secondary – Historical Search (Last 20 Minutes)



Event Receive Time	Reporting IP	Event Type	Event Name	Raw Event Log
May 04 2020, 10:15:49 AM	192.168.3.1	FortiGate-webfilter-url-firewall-allow	The URL belongs to an allowed category in the firewall policy	<189>May 4 16:15:49 date=2016-09-13 time=14:13:10 devname=FG240D39138004...
May 04 2020, 10:15:49 AM	192.168.3.1	FortiGate-traffic-forward	Forwarded traffic	<188>May 4 16:15:49 date=201
May 04 2020, 10:15:49 AM	192.168.3.1	FortiGate-traffic-forward	Forwarded traffic	<188>May 4 16:15:49 date=201
May 04 2020, 10:15:49 AM	192.168.3.1	FortiGate-Wireless-rogue-detect	FortiGate Wireless rogue detect	<189>May 4 16:15:49 date=201
May 04 2020, 10:15:49 AM	192.168.3.1	FortiGate-traffic-close	Session closed	<189>May 4 16:15:49 date=201
May 04 2020, 10:15:49 AM	192.168.3.1	FortiGate-appctrl-ips-pass	FortiGate appctrl ips pass	<190>May 4 16:15:49 date=201
May 04 2020, 10:15:48 AM	192.168.3.1	FortiGate-Wireless-rogue-detect	FortiGate Wireless rogue detect	<189>May 4 16:15:48 date=201

Event Receive Time	Reporting IP	Event Type	Event Name	Raw Event Log
May 04 2020, 10:10:09 AM	192.168.3.1	FortiGate-appctrl-ips-pass	FortiGate appctrl ips pass	<190>May 4 16:10:09 date=2016-09-13 time=13:59:12 devname=FG240D39138004...
May 04 2020, 10:10:09 AM	192.168.3.1	FortiGate-appctrl-ips-pass	FortiGate appctrl ips pass	<190>May 4 16:10:09 date=2016-09-13 time=13:59:13 devname=FG240D39138004...
May 04 2020, 10:10:08 AM	192.168.3.1	FortiGate-traffic-forward	Forwarded traffic	<189>May 4 16:10:08 date=2016-09-13 time=13:59:12 devname=FG240D39138004...
May 04 2020, 10:10:07 AM	192.168.3.1	FortiGate-traffic-denied	Denied traffic	<189>May 4 16:10:07 date=2016-09-13 time=13:59:06 devname=FG240D39138004...
May 04 2020, 10:10:04 AM	192.168.3.1	FortiGate-traffic-denied	Denied traffic	<189>May 4 16:10:04 date=2016-09-13 time=13:58:58 devname=FG240D39138004...
May 04 2020, 10:10:03 AM	192.168.3.1	FortiGate-traffic-forward	Forwarded traffic	<189>May 4 16:10:03 date=2016-09-13 time=13:58:51 devname=FG240D39138004...
May 04 2020, 10:10:02 AM	192.168.3.1	FortiGate-traffic-forward	Forwarded traffic	<189>May 4 16:10:02 date=2016-09-13 time=13:58:51 devname=FG240D39138004...
May 04 2020, 10:10:02 AM	192.168.3.1	FortiGate-appctrl-ips-pass	FortiGate appctrl ips pass	<190>May 4 16:10:02 date=2016-09-13 time=13:58:47 devname=FG240D39138004...

- Any notifications or scheduled report deliveries are performed on the Primary node only. (Since most of the required `ph*` processes are down on the Secondary).

Verify Elasticsearch Snapshots for Data Replication

Since the Secondary node GUI can be logged into, an Historical search will verify any event data being replicated. You can also observe this behavior from the Elasticsearch nodes.

- Snapshots Written to NFS Mounts
- Snapshot Tracking in Elasticsearch
- FortiSIEM Supervisor Node Tracking
- Snapshot and Restore Errors on FortiSIEM Backend

Snapshots Written to NFS Mounts

A simple directory listing of the NFS mounts will confirm data being written by Elasticsearch.

Primary Elasticsearch Nodes

Here is the mount for `/es_snapshot` under normal operation, on one of the Primary site Elasticsearch nodes. The response confirms that data is written to the Secondary site:

```
# Snapshot files written to Secondary site NFS Mount /es_snapshot
elk@pri-co-master:~$ ls -la /es_snapshot/
total 44
drwxrwsrwx+ 3 96 96 4096 May 3 18:48 .
drwxr-xr-x 24 root root 4096 May 1 21:35 ..
-rw-rw-rw-+ 1 elasticsearch 96 839 May 3 18:43 index-551
-rw-rw-rw-+ 1 elasticsearch 96 622 May 3 18:48 index-552
-rw-rw-rw-+ 1 elasticsearch 96 8 May 3 18:48 index.latest
drwxrwxrwx+ 7 elasticsearch 96 4096 May 3 18:28 indices
-rw-rw-rw-+ 1 elasticsearch 96 103 May 3 18:43 meta-ZWL6HHFvQL6MO6zcvvBkTg.dat
-rw-rw-rw-+ 1 elasticsearch 96 298 May 3 18:43 snap-ZWL6HHFvQL6MO6zcvvBkTg.dat
```

An inspection of the `/es_restore` directory shows that it is empty, as expected. The directory would show data only in the event of the Primary FortiSIEM being changed to Secondary.

```
# Snapshot files written to Primary site NFS Mount /es_restore
elk@pri-co-master:~$ ls -la /es_restore/
total 8
drwxrwsrwx+ 2 96 96 6 May 1 17:47 .
drwxr-xr-x 24 root root 4096 May 1 21:35 ..
```

Secondary Elasticsearch Nodes

Here is the mount for `/es_snapshot` under normal operation, on one of the Secondary site Elasticsearch nodes. The mount shows that it is empty as expected:

```
# Snapshot files written to Primary site NFS Mount /es_snapshot
elk@sec-co-master:~$ ls -la /es_snapshot
total 8
drwxrwsrwx+ 2 96 96 6 May 1 17:47 .
drwxr-xr-x 24 root root 4096 May 1 21:50 ..
```

An inspection of the Secondary Elasticsearch nodes reveal that they have data ready for restoration.

```
# Snapshot files written to Secondary site NFS Mount /es_restore
elk@sec-co-master:~$ ls -la /es_restore
total 44
```

```
drwxrwsrwx+ 3 96 96 4096 May 3 18:48 .
drwxr-xr-x 24 root root 4096 May 1 21:50 ..
-rw-rw-rw-+ 1 elasticsearch 96 839 May 3 18:43 index-551
-rw-rw-rw-+ 1 elasticsearch 96 622 May 3 18:48 index-552
-rw-rw-rw-+ 1 elasticsearch 96 8 May 3 18:48 index.latest
drwxrwxrwx+ 7 elasticsearch 96 4096 May 3 18:28 indices
-rw-rw-rw-+ 1 elasticsearch 96 103 May 3 18:43 meta-ZWL6HHFvQL6MO6zcvvBkTg.dat
-rw-rw-rw-+ 1 elasticsearch 96 298 May 3 18:43 snap-ZWL6HHFvQL6MO6zcvvBkTg.dat
```

Snapshot Tracking in Elasticsearch

Elasticsearch provides Snapshot and Restore APIs that FortiSIEM takes advantage of. These APIs can be viewed via the following CURL commands:

Viewing the Snapshots on Either Node (Primary ES Cluster Shown)

```
curl -XGET "http://10.10.2.151:9200/_snapshot/dr_snapshot_repo/_all?pretty"
{
   "snapshots" : [
      {
         "snapshot" : "fortisiem-snapshot-2020.05.03-21:20:23",
         "uuid" : "jLmCNrrVRNKlVPEBwDJtGA",
         "version_id" : 6080899,
         "version" : "6.8.8",
         "indices" : [
            "fortisiem-event-2020.05.03-2000",
            "fortisiem-summary-2020.05.03",
            "fortisiem-lookups-v2",
            "fortisiem-incident-2020.05",
            "fortisiem-event-2020.05.03-1"
         ],
         "include_global_state" : false,
         "state" : "SUCCESS",
         "start_time" : "2020-05-03T21:20:23.916Z",
         "start_time_in_millis" : 1588540823916,
         "end_time" : "2020-05-03T21:20:24.769Z",
         "end_time_in_millis" : 1588540824769,
         "duration_in_millis" : 853,
         "failures" : [ ],
         "shards" : {
            "total" : 25,
            "failed" : 0,
            "successful" : 25
         }
      }
   ]
}
```

A shorter version of the CURL output can be viewed by the following command.

```
curl -XGET "http://10.10.2.151:9200/_cat/snapshots/dr_snapshot_repo?v&s=id"
id      status start_epoch start_time end_epoch
end_time duration indices successful_shards failed_shards total_shards
fortisiem-snapshot-2020.05.03-21:20:23 SUCCESS 1588540823 21:20:23 1588540824
21:20:24    853ms     5     25     0     25
```

Viewing the Restores on Either Node (Secondary ES Cluster Shown)

```
curl -XGET "http://10.10.2.161:9200/_snapshot/dr_restore_repo/_all?pretty"
{
  "snapshots" : [
    {
      "snapshot" : "fortisiem-snapshot-2020.05.03-21:20:23",
      "uuid" : "jLmCNrrVRNKlVPEBwDJtGA",
      "version_id" : 6080899,
      "version" : "6.8.8",
      "indices" : [
        "fortisiem-event-2020.05.03-2000",
        "fortisiem-summary-2020.05.03",
        "fortisiem-lookups-v2",
        "fortisiem-incident-2020.05",
        "fortisiem-event-2020.05.03-1"
      ],
      "include_global_state" : false,
      "state" : "SUCCESS",
      "start_time" : "2020-05-03T21:20:23.916Z",
      "start_time_in_millis" : 1588540823916,
      "end_time" : "2020-05-03T21:20:24.769Z",
      "end_time_in_millis" : 1588540824769,
      "duration_in_millis" : 853,
      "failures" : [ ],
      "shards" : {
        "total" : 25,
        "failed" : 0,
        "successful" : 25
      }
    }
  ]
}
```

A shorter version of the CURL output can be viewed by the following command.

```
curl -XGET "http://10.10.2.161:9200/_cat/snapshots/dr_restore_repo?v&s=id"
id status      start_epoch start_time end_epoch
end_time duration indices successful_shards failed_shards total_shards fortisiem-snapshot-
     2020.05.03-21:20:23 SUCCESS 1588540823 21:20:23 1588540824
21:20:24 853ms      5      25      0      25
```

Note: Elasticsearch seems to use GMT timestamps for its snapshots.

FortiSIEM Supervisor Node Tracking

The snapshot, restore, and snapshot deletion can be tracked on the FortiSIEM Supervisor Primary and Secondary nodes, via the following command:

```
grep "snapshot" /opt/phoenix/log/phoenix.log
```

Primary FortiSIEM Supervisor

The `phDataPurger` process will log when a snapshot is taken and to what repository in Elasticsearch.

```
2020-05-03T23:20:24.793695+02:00 site1 phDataPurger[12216]:
```

```
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phDataPurger,[fileName]=ESDisa
     sterRecoveryManager.cpp,[lineNumber]=78,[phLogDetail]=Elasticsearch Snapshot succeeded.
     Repository: dr_snapshot_repo, Snapshot: fortisiem-snapshot-2020.05.03-21:20:23
```

Secondary FortiSIEM Supervisor

The Secondary FortiSIEM will show the restore reported by Elasticsearch with the same command as above.

```
2020-05-03T23:25:21.124793+02:00 site2 phDataPurger[19047]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phDataPurger,[fileName]=ESDisa
     sterRecoveryManager.cpp,[lineNumber]=102,[phLogDetail]=Elasticsearch Restore succeeded.
     Repository: dr_restore_repo, Snapshot: fortisiem-snapshot-2020.05.03-21:20:23
```

The Secondary will also report the deletion of the Snapshot file.

```
2020-05-03T23:35:25.824556+02:00 site2 phDataPurger[19047]:
[PH_GENERIC_INFO]:[eventSeverity]=PHL_INFO,[procName]=phDataPurger,[fileName]=Elasti
     csearchCluster.cpp,[lineNumber]=1131,[phLogDetail]=Elasticsearch snapshot deletion
     succeeded. Snapshot fortisiem-snapshot-2020.05.03-21:20:23
```

Snapshot and Restore Errors on FortiSIEM Backend

If things do not go as planned and you find snapshots are failing, use the following command to look for the following error messages:

```
grep "PH_DATAPURGER_DR" /opt/phoenix/log/phoenix.log
```

Here the `repo` was not mounted properly:

```
<11>Apr 29 17:50:04 site1 phDataPurger[17363]:
[PH_DATAPURGER_DR_ES_SNAPSHOT_FAILED]:[eventSeverity]=PHL_ERROR,[procName]=phDataPur ger,
     [fileName]=ESDisasterRecoveryManager.cpp,[lineNumber]=86,[errReason]=Elasticsear ch
     Snapshot failed. Reason: ,[phLogDetail]=Elasticsearch Snapshot failed. Repository: dr_
     snapshot_repo, Snapshot: fortisiem-snapshot-2020.04.29-15:50:04
```

This error was reported on the Secondary FortiSIEM node:

```
2020-04-29T17:50:26.060137+02:00 site2 phDataPurger[31599]:
[PH_DATAPURGER_DR_ES_SNAPSHOTS_GET_FAILED]:[eventSeverity]=PHL_ERROR,[procName]=phDa taPurger,
     [fileName]=ESDisasterRecoveryManager.cpp,[lineNumber]=154,[errReason]=No good snapshots
     in repository: dr_restore_repo,[phLogDetail]=Elasticsearch snapshots getting failed.
     Repository: dr_restore_repo
```

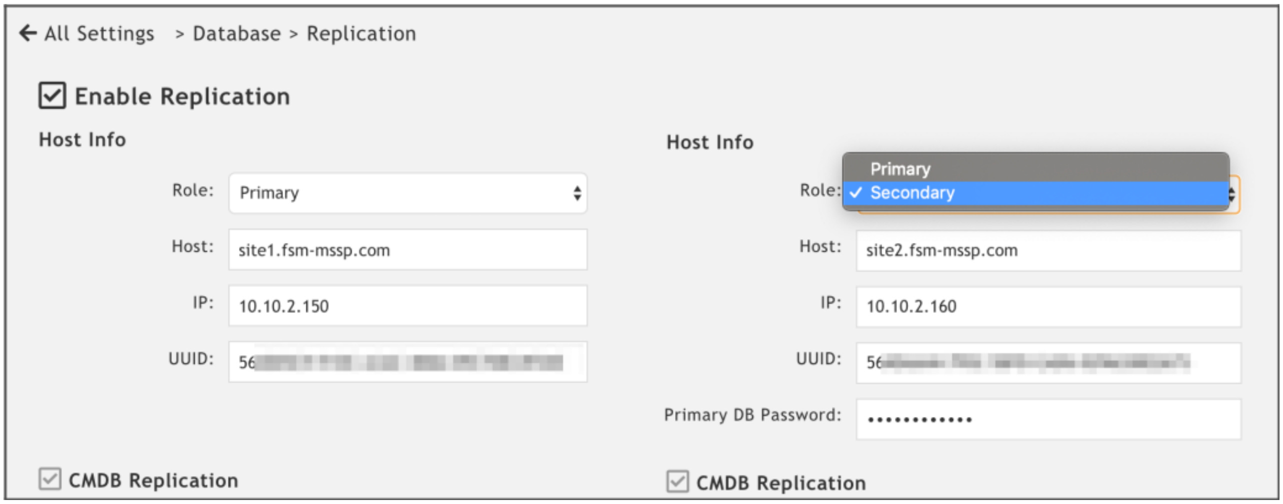
DR Change When the Primary site is Unavailable

It is important to note that it is a manual process to promote the Secondary node to be the Primary.

As soon as the Primary node is unavailable (that is, down/unavailable), any collector nodes will start to buffer their uploads, as the Worker Upload addresses they deliver to will be unavailable.

On the Secondary FortiSIEM node, log into the GUI:

1. Navigate to **Admin > Settings > Database > Replicate** (or **Replication** in 5.3+).
2. Change the **Role selector** for the Secondary node to be **Primary**.



← All Settings > Database > Replication

☑ Enable Replication

Host Info

Role: Primary

Host: site1.fsm-mssp.com

IP: 10.10.2.150

UUID: 56[redacted]

☑ CMDB Replication

Host Info

Role: Primary / ✓ Secondary

Host: site2.fsm-mssp.com

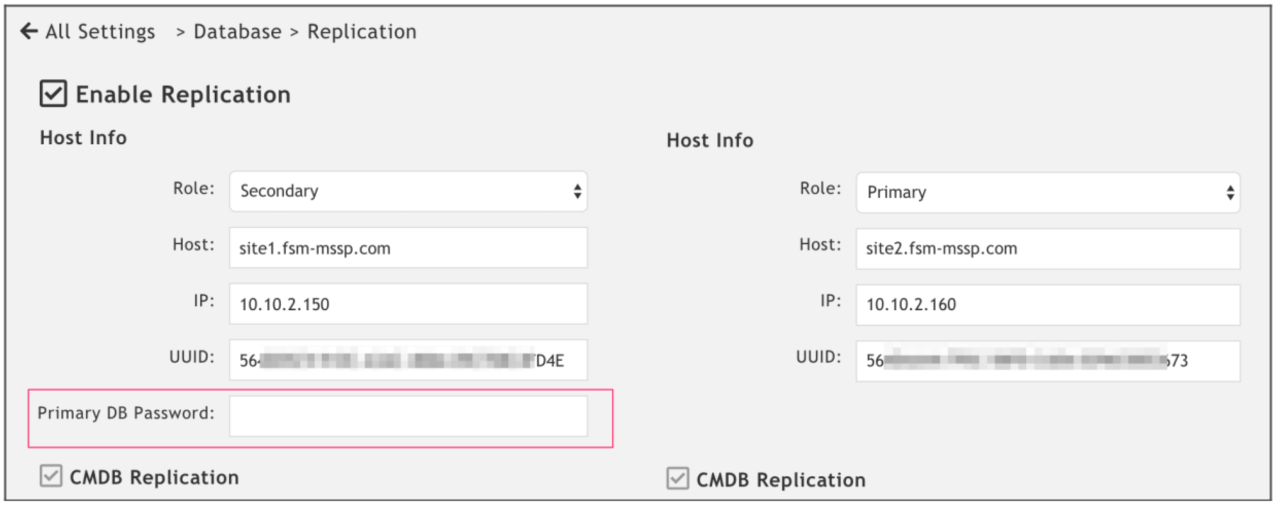
IP: 10.10.2.160

UUID: 56[redacted]

Primary DB Password: ••••••••••••

☑ CMDB Replication

3. Notice how the original Primary Role has now switched to Secondary, and the **PrimaryDB** Password field moves across to the left.



← All Settings > Database > Replication

☑ Enable Replication

Host Info

Role: Secondary

Host: site1.fsm-mssp.com

IP: 10.10.2.150

UUID: 56[redacted]D4E

Primary DB Password:

☑ CMDB Replication

Host Info

Role: Primary

Host: site2.fsm-mssp.com

IP: 10.10.2.160

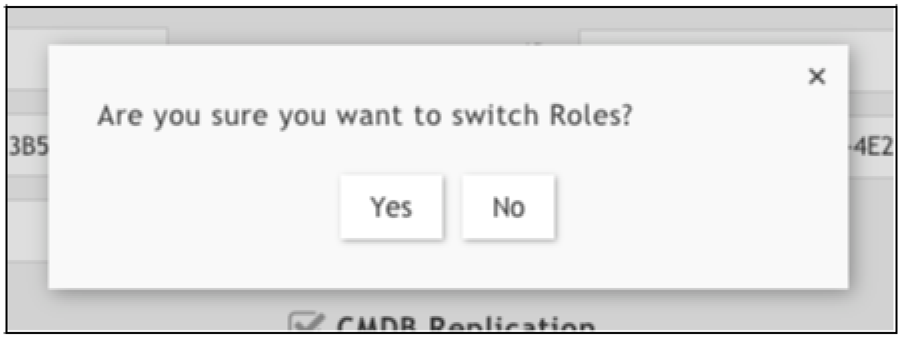
UUID: 56[redacted]73

☑ CMDB Replication

This field needs to be input again, but it can be obtained from an SSH session to the Secondary now, as it now has the same database as the Primary. Run the following command and paste the output into the **Primary DB Password** field.

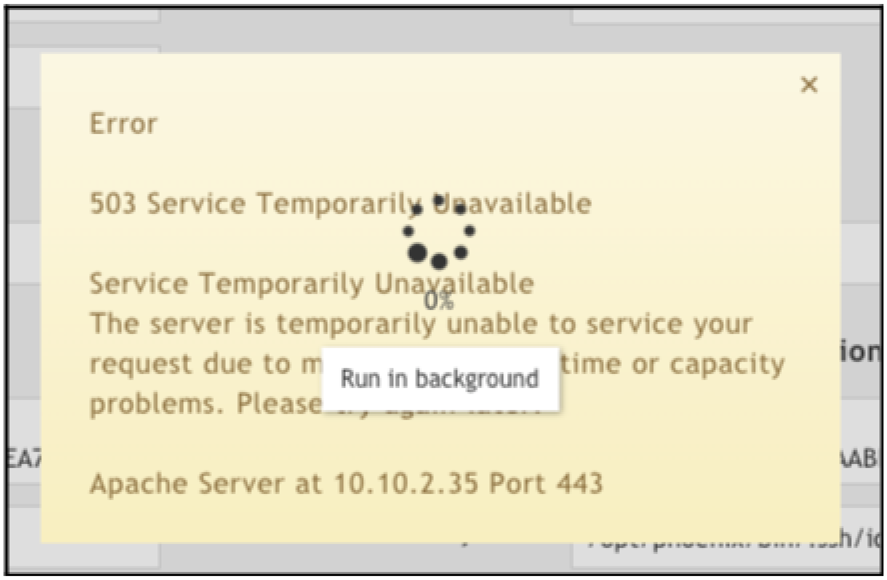
```
#On the SECONDARY node
/opt/phoenix/bin/phLicenseTool -showDatabasePassword
```

4. Click **Apply**.
5. Click **Yes** to the warning, `Are you sure you want to switch Roles?`.



Are you sure you want to switch Roles?

At this time, the following will appear in the GUI and it will seem to disconnect and the DR scripts will be run in the background.



Error

503 Service Temporarily Unavailable

Service Temporarily Unavailable
The server is temporarily unable to service your request due to m time or capacity problems. Please

Apache Server at 10.10.2.35 Port 443

After a short period of time, all the backend processes will start and the GUI will return to the login page.

If you run a Real-Time search you will probably find no data is still being received. This is because a DNS change is now required for the shared DNS addresses for the Supervisor node and the Worker upload settings, in this example case:

DNS Address	Old Value	New Value
site.fsm-mssp.com	CNAME -> site1.fsm-mssp.com	CNAME -> site2.fsm-mssp.com
worker1.fsm-mssp.com	198.51.100.20	203.0.113.20
Worker2.fsm-mssp.com	198.51.100.21	203.0.113.21

Change the DNS addresses and data will start to flow in normally.

Note: When the original Primary is recovered and powered back on, it will detect this and take on the Secondary role automatically, although it will take minutes, not seconds.

Change-Over Where Both Systems are Operational

Operationally, there may be a need to perform a DR change over while both nodes are actually up and running.

Again, to note, this is a manual process of promoting the Secondary node to be the Primary.

On the Primary FortiSIEM node, log into the GUI:

1. Navigate to **Admin -> Settings -> Database -> Replicate** (or **Replication** in 5.3+).
2. Change the **Role selector** for the Primary node to be **Secondary**.
3. Populate the **Primary DB Password** field.
   Run the following command on either the Primary or Secondary node via SSH:
   ```
   #On the PRIMARY or SECONDARY node
   /opt/phoenix/bin/phLicenseTool --showDatabasePassword
   ```
4. Click **Apply**, and respond **Yes** to the warning, “`Are you sure you want to switch Roles?`”.
   Note: The extra steps below are very important. You will have a cluster which thinks it has two Primary nodes if you do not follow the two steps below.
5. Switch to the Secondary node GUI, and navigate to **Admin > Settings > Database > Replicate** (or **Replication** in 5.3+).
6. Change the Roles (unless the CMDB sync has already updated).
7. Click **Apply**.

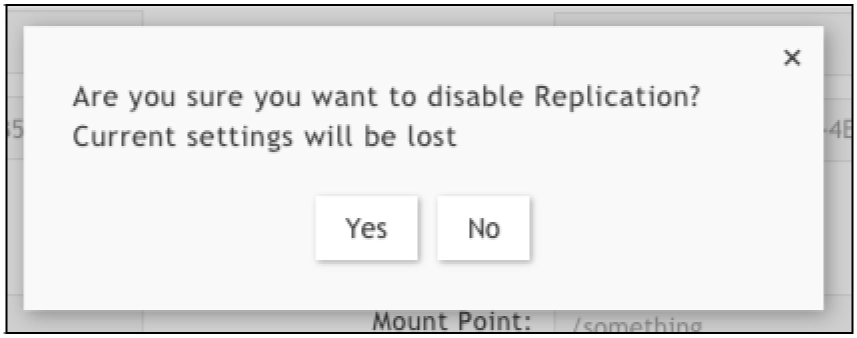
Remember to change the DNS addresses after the migration.

Turning Off the Disaster Recovery Feature

There are cases where the DR Replication feature needs to be disabled, such as performing upgrades.

On the Primary FortiSIEM node, log into the GUI:

1. Navigate to **Admin > Settings > Database > Replicate** (or **Replication** in 5.3+).
2. Deselect the **Enable Replication** check box.
3. Respond **Yes** to the warning regarding disabling the Replication.



Are you sure you want to disable Replication?
Current settings will be lost

4. Click **Apply**.
5. Wait for the response `Replicate settings applied`.

Since the database is shared, this only needs to be performed on one node.

But, due to a bug in 5.2.8, it can only be re-enabled from the opposite node, Secondary in this case.