

VOTIRO

Secure File Gateway - Implementation

Troubleshooting Playbook

Votiro Support
November 2020

Contents

- 1 Introduction 3**
- 2 Verify Fallback State 4**
- 3 Connectivity Checks and Disk Speed 5**
 - 3.1 Perform Connectivity Check between Nodes 5
 - 3.2 Perform Internet Connectivity Check 5
 - 3.3 Verify Network Configuration and Unique Hostname for each Node 6
 - 3.4 Check Disk Speed 7
- 4 Installation Checks 8**
 - 4.1 Install the Virtual Appliance 8
 - 4.2 Check Cluster Health 8
 - 4.3 Check Redis Cluster Health 9
 - 4.4 Check Cluster Health - Pods are not in Running State after the Installation \ Upgrade 9
 - 4.5 Check Cluster Health - Pods in imagepullback \ ImageErr10
- 5 Post Implementation Checks11**
 - 5.1 License Creation - Problem Exporting License 11
 - 5.2 Test a File - File is blocked due to an Error in Positive Selection 11

1 Introduction

The purpose of this Troubleshooting Playbook is to provide technical support for your implementation of Votiro's Secure File Gateway.

The playbook is split into sections to help you identify the most likely stage the problem occurred, determine the source of the problem, and resolve it. The four sections relating to the stages of the implementation are:

- [Verify Fallback State](#) – this stage verifies the cluster could revert quickly in case of any issues with the deployment.
- [Connectivity Checks and Disk Speed](#) – this stage checks the connectivity between the Nodes, and that the disk speed is sufficient for the cluster operation.
- [Installation Checks](#) – this stage checks that the cluster was deployed successfully and that all cluster components are running as expected.
- [Post Implementation Checks](#) – this stage checks the license creation and file sanitization, ensuring the **Test File** feature is functioning correctly.

Should you require additional assistance troubleshooting your implementation, contact Votiro's Support team by email at support@votiro.com.

2 Verify Fallback State

Action:

Verify a VM snapshot exists, if on a vmware environment.

Diagnosis:

- No snapshot was taken.

Resolution:

Ensure a snapshot of each node is taken, then use the following command:

```
initcluster.
```

3 Connectivity Checks and Disk Speed

3.1 Perform Connectivity Check between Nodes

Action:

Check connectivity from the first node to each of the other nodes, using the *Ping* command .

Check connectivity from node 0 to each of the other nodes, using the *SSH* command.

Diagnosis:

- *Ping* is unreachable.
- Unable to SSH from Node 0 to the other nodes.

Resolution:

Perform these additional checks:

- Segmentation between the 3 nodes.
- FW or any other networking component between nodes.
- Other SSH traffic is allowed to pass in the segment.

3.2 Perform Internet Connectivity Check

Action:

Perform an internet connectivity check, using the following command:

```
#curl https://votiroregistry.azurecr.io/v2/ -v
```

Diagnosis:

- cURL command returns a timeout response.

Resolution:

If there is a proxy, configure it and check connectivity, using the following command:

```
#curl -x 'http://USER:PASSWORD@PROXY' -v  
https://votiroregistry.azurecr.io/v2/
```

Note

USER:PASS is for a proxy that requires authentication.

After PROXY you should add port: PROXY_IP:PROXY_PORT

If there is no proxy and no internet connectivity, disable **Safebrowsing** and use the *initcluster* command in **Offline** mode.

3.3 Verify Network Configuration and Unique Hostname for each Node

Action:

To verify your network configuration, use the *nmtui* tool.

Check the hostname, using the following command:

```
#hostname.
```

Check the IP address using the following command:

```
#ip addr | grep ens160.
```

Diagnosis:

- Hostnames configured with uppercase or underscore.
- Hostnames are not configured – all hostnames are the same.
- IP addresses were not configured.

Resolution:

If the hostnames or IPs are not unique, configure using the command `#nmtui`

1. Edit a connection.
2. Set up IPv4 Configuration to be **Manual**, click **Show**.
3. Add Address, Gateway, DNS servers.
4. Click **OK**, then click **Back**.
5. Set the system hostname (you may use the FQDN) (all lowercase, numbers and “-“):
6. Quit the *nmtui*.
7. Restart the machine, using the following command:

```
#reboot.
```

Once the machines are back up, check the *hostname* and *IP* again to confirm the change.

3.4 Check Disk Speed

Action:

To check the disk speed, use the *healthcheck* script. The disk speed check is embedded towards the end of the output on your screen.

Connect to each node, then use the following command:

```
#hdparm -Tt /dev/sda
```

The output you should receive is as follows:

```
/dev/sda:
Timing cached reads:   11150 MB in  1.99 seconds = 5593.75 MB/sec
Timing buffered disk reads: 718 MB in  3.00 seconds = 238.96 MB/sec
```

Diagnosis:

- The *hdparm* command is not found.
- OVF for Version 9.3 or earlier, does not have the utility built-in.

Resolution:

If there is internet connectivity, manually install *hdparm*, using the following command:

```
#sudo yum install hdparm
```

During the install you will be prompted to proceed with the install, enter **Y**.

The expected output for *hdparm* is at least the values in the screenshot in the **Action** section above. In case you get lower results, check the type of disks being used in the environment, and what competing activity there is on the host.

4 Installation Checks

4.1 Install the Virtual Appliance

Action:

To check the status of your installation, use the *initcluster* script.

Diagnosis:

- The *initcluster* execution failed.

Resolution:

Check the *Votiro-setup.log* file , using the following command:

```
#cat votiro-setup.log
```

4.2 Check Cluster Health

Action:

To verify all components are running correctly, use this short sanity-check on your system. The *healthcheck* script can be found in the OVF.

To execute the *healthcheck* script, navigate to the */root* folder, then use the following command:

```
#./healthcheck
```

The expected results are:

- No pods with issues.
- No errors in Redis pods.
- All elastic pods are running.

You may proceed to open the UI when the following conditions are met:

- All pods are running.
- The Redis cluster is in its desired state.
- Disk speeds are at the required rates.

Login to the Management Dashboard and upload a file using the **Test File** feature on the **Policies** page.

Diagnosis:

- One of the nodes' state is **Not Ready**.

Resolution:

Check for errors, using the following command:

```
#kubectl describe node NodeName -n votiro
```

4.3 Check Redis Cluster Health

Action:

Check the Redis cluster, using the *healthcheck* script.

Diagnosis:

- Redis cluster status: fail.
- Cluster known nodes: less than 6.
- Cluster size: less than 3.

Nodes details:

- Master-fail.
- Slave-fail.
- Redis nodes are in *crashloopback* state.

Resolution:

Reset redis, using the following command:

```
#!/root/reser-redis.sh
```

4.4 Check Cluster Health - Pods are not in Running State after the Installation \ Upgrade

Action:

Check the status of pods, using the following command:

```
#kubectl get pods -n votiro
```

Pod status: imagepullbackoff error \ crashloopbackoff

Diagnosis:

- To further understand what happened on the affected pod, use the following command:

```
#kubectl describe pod <pod-name> -n Votiro | grep -A20 Events
```

Scroll down to see the Events.

Resolution:

If the error is clear, act accordingly to resolve the issue.

If not address support with the describe output.

4.5 Check Cluster Health - Pods in imagepullback \ ImageErr

Action:

To further understand the issue with the affected pod, use the following command:

```
#kubectl describe pod <pod-name> -n Votiro | grep -A20 Events
```

Diagnosis:

- The image attempting to be loaded does not exist, using the following command:

```
#Docker images | grep "<docker_image>"
```

Resolution:

Load the images manually.

Navigate to the **Upgrade** folder, and use the following command:

```
docker load -i images.tar
```

5 Post Implementation Checks

5.1 License Creation - Problem Exporting License

Action:

Open the Management Dashboard and navigate to **System Setup > License**. Click **Generate**.

A LicensePackage.zip file is generated.

Diagnosis:

- License is not exported.
- Problem in the UI in the license exporting phase.

Resolution:

Obtain the *systemId* manually:

1. Login to Node1.
2. Use command:

```
#kubectl get cm system-config -n votiro -o yaml | grep systemId:
```
3. Request a license by sending the *systemId*, customer details and the license expiration date, to Votiro support by email at support@votiro.com. Alternatively, generate a license using the Votiro license server.

5.2 Test a File - File is blocked due to an Error in Positive Selection

Action:

Files with URLs may be blocked if the Virtual Appliance cannot reach the internet.

Diagnosis:

- Proxy is blocking access.
- Geo location block in FW.
- Any other networking issues.

Resolution:

Disable **Safebrowsing** manually, using these steps:

1. Login to Node 1.
2. Use the following command:

```
#kubectl edit cm system-config -n votiro -o yaml
```

3. Set the value of parameter **SafebrowsingIsEnabled:** to **False**.

4. Restart the relevant pods, using the following command:

```
- #kubectl get pods -n votiro | awk '/officex|office-  
|pdf|eml/{print $1}' | xargs kubectl delete -n votiro pod
```