VSP Upgrade

For Containers Using Helm Charts

Virsec Security Platform 2.5

www.virsec.com
support@virsec.com

Contents

About this Document

This document provides information about VSP Upgrade in containers using Helm Charts.

1. LFR

1. Log in to the Artifactory site using Virsec-provided credentials from the local machine
2. Navigate to the directory **vsp > ReleaseNumber > Helm**
3. Right-click on the file `vsp-lfr-<version>.tgz`, listed on the page and download it to the local system
4. Install VSP LFR by executing the below steps:
   a. Log in to the Management Node
   b. Copy the downloaded file `vsp-lfr-<version>.tgz`
5. Execute the below command to display the configurable parameters:
   a. `helm inspect values ./vsp-lfr-<version>.tgz`
6. Execute the below commands to install LFR and display the LFR URL (using either Method 1 or 2)
   a. **Method 1: Using helm install command**
      i. **For On-Premise environments:**
         1. **Helm 3**

```
helm upgrade vsp-lfr ./vsp-lfr-<RELEASE_VERSION>.tgz --set
artifactory.username="<ARTIFACTORY_USERNAME>" --set
artifactory.password='<ARTIFACTORY_PASSWORD>' --namespace virsec
```

         2. **Helm 2**

```
helm upgrade --name vsp-lfr ./vsp-lfr-<RELEASE_VERSION>.tgz --set
artifactory.username="<ARTIFACTORY_USERNAME>" --set
artifactory.password='<ARTIFACTORY_PASSWORD>' --namespace virsec
```

      ii. **For AWS EKS/ GOOGLE GKE environments:**
         1. **Helm 3**

```
helm upgrade vsp-lfr ./vsp-lfr-<RELEASE_VERSION>.tgz --set
cloudProvider=eks  --set
artifactory.username="<ARTIFACTORY_USERNAME>" --set
artifactory.password='<ARTIFACTORY_PASSWORD>' --namespace virsec
```

         2. **Helm 2**

```
helm upgrade --name vsp-lfr ./vsp-lfr-<RELEASE_VERSION>.tgz --set
cloudProvider=eks --set
artifactory.username="<ARTIFACTORY_USERNAME>" --set
artifactory.password='<ARTIFACTORY_PASSWORD>' --namespace virsec
```

iii. **Verification:** Execute the below command to verify the upgrade:

```
helm status vsp-lfr
```

iv. Execute the below commands to retrieve LFR URL:

```
export SERVICE_IP=$(kubectl get svc --namespace virsec vsp-lfr --
template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{
end }}")
echo http://$SERVICE_IP:80
```

b. **Method 2: Using kubectl command**

i. **For On-Premise environments:**

1. **Helm 3**

```
helm template vsp-lfr ./vsp-lfr-<RELEASE_VERSION>.tgz --set
artifactory.username="<ARTIFACTORY_USERNAME>" --set
artifactory.password='<ARTIFACTORY_PASSWORD>' --namespace virsec
> vsp-lfr.yaml
```

2. **Helm 2**

```
helm template --name vsp-lfr ./vsp-lfr-<RELEASE_VERSION>.tgz --
set artifactory.username="<ARTIFACTORY_USERNAME>" --set
artifactory.password='<ARTIFACTORY_PASSWORD>' --namespace virsec
> vsp-lfr.yaml
```

ii. **For AWS EKS/ GOOGLE GKE environments:**

1. **Helm 3**

```
helm template vsp-lfr ./vsp-lfr-<RELEASE_VERSION>.tgz --set
cloudProvider=eks  --set
artifactory.username="<ARTIFACTORY_USERNAME>" --set
artifactory.password='<ARTIFACTORY_PASSWORD>' --namespace virsec
> vsp-lfr.yaml
```

2. **Helm 2**

```
helm template --name vsp-lfr ./vsp-lfr-<RELEASE_VERSION>.tgz --
set cloudProvider=eks --set
artifactory.username="<ARTIFACTORY_USERNAME>" --set
artifactory.password='<ARTIFACTORY_PASSWORD>' --namespace virsec
> vsp-lfr.yaml
```

iii. Execute the below command to deploy VSP LFR:

```
kubectl apply -f vsp-lfr.yaml
```

iv. Execute the below commands to retrieve LFR URL:

```
export SERVICE_IP=$(kubectl get svc --namespace virsec vsp-lfr --
template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{
end }}")
echo http://$SERVICE_IP:80
```

2. CMS

1. For upgrade from VSP CMS 2.4.x or previous versions, execute the commands: **Worker Node**
   a. `rm -rf /home/virsec/kafkavolume`
   b. `rm -rf /home/virsec/zookeepervolume`
2. For an **EKS environment**, execute the below commands to delete PV and PVC for `jreports-content` and `jreports-database`
   a. Execute the commands below for upgrade from all VSP versions **2.2.x and below:**
      i. **Delete PVC**

```
kubectl -n virsec delete pvc jreports-content-claim
kubectl -n virsec delete pvc jreports-database-claim
```

      ii. **Delete PV**

```
kubectl -n virsec delete pv jreports-content
kubectl -n virsec delete pv jreports-database
kubectl get  pv  | grep virsec/jreports-database-claim  | awk
'{print $1}' | xargs -I {} kubectl delete pv {}
kubectl get  pv  | grep virsec/jreports-content-claim  | awk '{print
$1}' | xargs -I {} kubectl delete pv {}
```

      iii. **Verification:** Ensure that the command below does not give any output list

```
kubectl get pv,pvc,storageclass --all-namespaces
```

   b. Execute the commands below for upgrade from **VSP 2.2.3 only:**
      i. **Delete PVC (SSL)**

```
kubectl delete pvc ssl-certs-content-claim -n virsec
```

      ii. **Delete PV (SSL)**

```
kubectl delete pv ssl-certs-content
kubectl get  pv  | grep virsec/ssl-certs-content-claim  | awk
'{print $1}' | xargs -I {} kubectl delete pv {}
```

3. Download the file `vsp_cleanup_cms.sh` from the Artifactory directory **vsp > ReleaseNumber > Helm**

a. To view the help menu, execute the below command:

`./vsp_cleanup_cms.sh -h`

```
ubuntu@ip-10-0-24-36:~/helm$ ./vsp_cleanup_cms.sh -h

./vsp_cleanup_cms.sh can be used for cleaning cms deployment and services while upgrading or
installaing new cleaned build through Helm Charts

usage: ./vsp_cleanup_cms.sh command [-h]
  -h print this usage
  -S cleaning all services; optional

Example usage:
    help:
        ./vsp_cleanup_cms.sh -h
    cleaning deployments only:
        ./vsp_cleanup_cms.sh
    cleaning deployments and services
        ./vsp_cleanup_cms.sh -S
```

b. Execute the same command using one of the below parameters to remove the previous version

   i. -S (Optional): For clean CMS setup. This deletes all the services of the previous setup (if any)

4. Log in to the Artifactory site using Virsec-provided credentials from the local machine
5. Navigate to the directory **vsp > ReleaseNumber > Helm**
6. Right-click on the file `cms-<version>.tgz`, listed on the page and download it to the local system
7. Upgrade VSP CMS by executing the below steps:
   a. Log in to the Management Node
   b. Copy the downloaded file `cms-<version>.tgz`
   c. Execute the below command to display the configurable parameters:

      `helm inspect values ./cms-<version>.tgz`
8. Create a custom value file to provide custom values for deployment as described below:
   a. `vi <CustomFileName>.yaml`

> NOTE:
> Only Multi-pod CMS deployment is supported

b. **Optional CMS Services Configuration:** To deploy optional CMS services, configure the parameters as described below:

i. Indicate `true` or `false` for all the optional services installation - Ticketing (Zendesk), Syslog, Splunk, Centralized logging, MSSP Portal, VSP APIs, Reporting

```
# deployment type and features options
multipods:
  enabled: true
  options:
    zendesk:
      enabled: true
    syslogService:
      enabled: true
    siemSplunkService:
      enabled: true
    centralizedloggingSystem:
      enabled: false
    mSSPPortal:
      enabled: true
    exposeVSPAPIs:
      enabled: true
    reporting:
      enabled: true
```

c. **Secure Kafka Options:** The options are available for Kafka are:

`0`: For Unsecure Kafka connection. By default, the value is set to 0 if not specified

`1`: For One-way SSL where the Client verifies the server

`2`: For Two-way SSL where both the Client and Server verify each other

i. **Sample Usage:** The example below depicts One-way SSL configuration:

```
helm install vsp-cms ./cms-<RELEASE_VERSION>.tgz --set
cloudProvider=eks --set kafka.secureKafkaMode="1" --namespace virsec
```

NOTE:
If the Probes are of version 2.4.x and below, ensure that only option 0 is used for Kafka. Do not use options 1 or 2. as they are not supported

d. **Method 1: Using helm upgrade command**

NOTE:
Provide the parameter "`-f <CustomFileName>.yaml`" in the below commands if selective optional CMS services need to be installed

i. **For On-Premise environments:**

1. **Helm 3**

```
helm upgrade vsp-cms ./cms-<RELEASE_VERSION>.tgz --set
upgrade=true --namespace virsec
```

2. **Helm 2**

```
helm upgrade --name vsp-cms ./cms-<RELEASE_VERSION>.tgz --set upgrade=true --namespace virsec
```

ii. **For AWS EKS/ GOOGLE GKE environments:**

1. **Helm 3**

```
helm upgrade vsp-cms  ./cms-<RELEASE_VERSION>.tgz --set upgrade=true --set cloudProvider=eks  --namespace virsec
```

2. **Helm 2**

```
helm upgrade --name vsp-cms   ./cms-<RELEASE_VERSION>.tgz --set upgrade=true --set cloudProvider=eks --namespace virsec
```

iii. **Verification:** Execute the below command to verify the upgrade:

1. `helm status vsp-cms`

iv. Execute the Provided commands to retrieve CMS URL:

```
$ export SERVICE_IP=$(kubectl get svc --namespace virsec vsp-cms --template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}")
$ echo https://$SERVICE_IP:443
```

e. **Method 2: Using kubectl command**

> NOTE:
> Provide the parameter "`-f <CustomFileName>.yaml`" in the below commands if selective optional CMS services need to be installed

i. **For On-Premise environments:**

1. **Helm 3**

```
helm template vsp-cms ./cms-<RELEASE_VERSION>.tgz  --set upgrade=true --namespace virsec > vsp-cms.yaml
```

2. **Helm 2**

```
helm template --name vsp-cms ./cms-<RELEASE_VERSION>.tgz --set upgrade=true --namespace virsec > vsp-cms.yaml
```

ii. **For AWS EKS/ GOOGLE GKE environments:**

1. **Helm 3**

```
helm template vsp-cms  ./cms-<RELEASE_VERSION>.tgz --set upgrade=true --set cloudProvider=eks  --namespace virsec > vsp-cms.yaml
```

2. **Helm 2**

```
helm template --name vsp-cms   ./cms-<RELEASE_VERSION>.tgz --set upgrade=true --set cloudProvider=eks --namespace virsec > vsp-cms.yaml
```

iii. Execute the below command to deploy VSP CMS:

```
kubectl apply -f vsp-cms.yaml
```

iv. Execute the below commands to retrieve CMS URL:

```
$ export SERVICE_IP=$(kubectl get svc --namespace virsec vsp-cms --template "{{ range (index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}")
$ echo https://$SERVICE_IP:443
```

NOTE:
If a proxy server is configured for internet access, ensure that the root certificate information is added to the property file, as described in the Deploy Custom SSL Certificates Section of the Maintenance document

-- END OF DOCUMENT --