

Helms Chart Repository

- Helm Charts Repository, on page 1

Helm Charts Repository

Helm is a tool that you can use to manage packages in the Kubernetes environment. You can use Helm to modify the current deployment. The helm repository is a repository in which you can find the sample value files for XRd and review all available settings that can be configured during the deployment of XRd on K8s.

To deploy XRd on your Amazon Elastic Container Service for Kubernetes (EKS) cluster, use the appropriate XRd Helm Chart. The below-mentioned application charts are available in the XRd Helm Repository:

- xrd-control-plane, for running XRd Control Plane containers.
- xrd-vrouter, for running XRd vRouter containers.

Follow the instructions provided in the Helm documentation to install and setup Helm.

The following commands serve as options for interacting with XRd's Helm repository:

- To add the Helm repository, enter the `helm repo add xrd https://ios-xr.github.io/xrd-helm` command.
- To retrieve the latest versions of packages from a previously added Helm repository, enter the `helm repo update xrd` command.
- To view the charts available in a specific Helm repository, enter the `helm search repo xrd` command.
- To install the chart, enter the `helm install my-<chart-name> xrd/<chart-name> command.`
- To uninstall the chart, enter the `helm delete my-<chart-name> xrd/<chart-name> command.`
- To download and view files for a published chart without installing them, enter the `helm pull chartrepo/chartname` command.

The process of creating a Kubernetes StatefulSet object through XRd charts follows the generation of a singular replica.

Sample Helm Values for XRd vRouter

The following are sample Helm values files for cloud router deployment on m5[n].12xlarge, m5[n].24xlarge, or m6in.16xlarge instances. These examples uses default resource requirements from the data sheet.

Perform the following updates before use:

- Set `image.repository` to a container repository URL containing XRd vRouter images.
- Update `image.tag` depending on the tagging scheme for your container repository.
- Update `config.username` and `config.password` with a username and password of your choice to use for the XR root user.
- Update `interfaces` and `mgmtInterfaces` depending on your network topology.

Example: m5[n].12xlarge or m5[n].24xlarge XRd Workload Values File

```
# Sample values file for running an XRd workload on an m5[n].12xlarge
# or m5[n].24xlarge EC2 instance.
#
# This values file is for the xrd-vrouter chart at version 1.0.0.

# Image configuration
image:
  # Repository for the container image (required).
  repository: "my-repo/xrd-vrouter"
  # Image tag (required).
  tag: "25.2.1"

# Pod resource configuration.
resources:
  limits:
    # All this memory must be available on NUMA node 0 (i.e. CPU socket 0).
    memory: 16Gi
    hugepages-1Gi: 6Gi

# Security context for the XRd container.
# Privileged mode is currently required for XRd in K8s due to device
# access requirements.
securityContext:
  privileged: true

# Persistent storage controls.
persistence:
  enabled: true
  size: "6Gi"
  accessModes:
  - ReadWriteOnce
  storageClass: "gp2"

# XR configuration and boots scripts.
config:
  # XR root username and password.
  username: "myusername"
  password: "mypassword"

# XRd line interfaces.
# N.B. This requires there to be 15 interfaces attached to the worker node -
# one for the K8s cluster networking and 14 for XRd.
interfaces:
- type: pci
  config:
    last: 14

# XRd management interfaces.
mgmtInterfaces: []
```

```
# CPU settings.
cpu:
  # The cpuset to use for XRd vRouter.
  cpuset: "7-23""

# PCI interface driver.
pciDriver: "igb_uio"
```

Example: m6in.16xlarge XRd Workload Values File

```
# Sample values file for running an XRd workload on an m6in.16xlarge EC2
# instance.
#
# This values file is for the xrd-vrouter chart at version 1.0.0.

# Image configuration
image:
  # Repository for the container image (required).
  repository: "my-repo/xrd-vrouter"
  # Image tag (required).
  tag: "25.2.1"

# Pod resource configuration.
resources:
  limits:
    memory: 16Gi
    hugepages-1Gi: 6Gi

# Security context for the XRd container.
# Privileged mode is currently required for XRd in K8s due to device
# access requirements.
securityContext:
  privileged: true

# Persistent storage controls.
persistence:
  enabled: true
  size: "6Gi"
  accessModes:
  - ReadWriteOnce
  storageClass: "gp2"

# XR configuration and boots scripts.
config:
  # XR root username and password.
  username: "myusername"
  password: "mypassword"

# XRd line interfaces.
# N.B. This requires there to be 15 interfaces attached to the worker node -
# one for the K8s cluster networking and 14 for XRd.
interfaces:
- type: pci
  config:
    last: 14

# CPU settings.
cpu:
  # The cpuset to use for XRd vRouter.
  cpuset: "15-31"

# PCI interface driver.
pciDriver: "igb_uio"
```

Sample Helm Values for XRd Control Plane

The following is a sample Helm values file for an XRd Control Plane deployment. This example uses default resource requirements from the data sheet.

Perform the following updates before use:

- Set `image.repository` to a container repository URL containing XRd Control Plane images.
- Update `image.tag` depending on the tagging scheme for your container repository.
- Update `config.username` and `config.password` with a username and password of your choice to use for the XR root user.
- Update `interfaces` and `mgmtInterfaces` depending on your network topology.

Example: XRd SR-PCE Values File on AWS EKS

```
#Sample values file for running an XRd SR-PCE workload on AWS EKS.
#
# This values file is for the xrd-control-plane chart at version 1.0.0.

# Image configuration
image:
  # Repository for the container image (required).
    repository: "my-repo/xrd-control-plane"
  # Image tag (required).
    tag: "25.2.1"
# Pod resource configuration.
# See https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
  resources:
  limits:
  memory: 16Gi

# Security context for the XRd container.
# Privileged mode is currently required for XRd in K8s due to device
# access requirements.
securityContext:
  privileged: true

# Persistent storage controls.
persistence:
  enabled: true
  size: "6Gi"
  accessModes:
  - ReadWriteOnce
  storageClass: "gp2"

# XR configuration and boots scripts.
config:
  # XR root username and password.
  username: "myusername"
  password: "mypassword"

# XRd line interfaces.

interfaces:
  - type: ens6
  config:
  type: host-d

# XRd management interfaces.
```

```
mgmtInterfaces:
- type: defaultCni
chksum: true
```