Red Hat Developer

developers.redhat.com   redhat-developer   @rhdevelopers

Cheat sheet

OpenShift disconnected installation

Red Hat OpenShift is a comprehensive Kubernetes platform that simplifies the deployment, management, and scaling of applications, offering support for both containers and virtual machines within a single environment.

In Internet-connected environments, OpenShift installation is straightforward with installer-provisioned infrastructure, a guided, automated method ideal for cloud providers or on-premises setups. However, in disconnected environments, especially those with stringent security requirements, installations become more complex, requiring mirroring of all necessary content locally and tricks to simulate an internet connection for OpenShift's functionality. This cheat sheet walks you through the process.

Note: These steps described in this resource are based on Red Hat OpenShift 4.15.

Step 1: Download & configure software on the connected bastion host

1. Validate enough space exists

Ensure that enough space is available to download the software; at least 75 GB of space is required. You can check this by running the following command:

```
$ df -h
```

2. Download OpenShift software

```
$ wget
https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/stable-4.15/
openshift-client-linux.tar.gz
```

```
$ wget
https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/stable-4.15/
oc-mirror.tar.gz
$ wget
https://mirror.openshift.com/pub/openshift-v4/x86_64/clients/ocp/stable-4.15/
openshift-install-linux.tar.gz
$ wget https://developers.redhat.com/content-gateway/file/pub/openshift-v4/
clients/mirror-registry/1.3.10/mirror-registry.tar.gz
```

3. Configure fapolicyd

`fapolicyd` (File Access Policy Daemon) is a security tool designed to control file access and execution on Linux systems. It is primarily used to enforce policies regarding which files can be executed or accessed, thereby enhancing the security posture of the system. By managing a list of allowed and denied actions, `fapolicyd` helps prevent the execution of unauthorized or potentially harmful software.

With a STIG (Security Technical Implementation Guide)-compliant environment, `fapolicy` automatically blocks any binary that is not an RPM. Therefore, we will need to add it to the `fapolicy` trust:

```
#We need to disable FApolicyD for the install
$ systemctl stop fapolicyd.service
# We will need to add our tools to the fapolicyd trust
$ sudo fapolicyd-cli --file add /usr/local/bin/oc-mirror
$ sudo fapolicyd-cli --update
#We need to start FApolicyD for the install
$ systemctl start fapolicyd.service; systemctl status fapolicyd.service
```

4. Acquire pull secrets

You will need your pull secret to authenticate with the Red Hat servers. In the same folder, issue the following commands:

```
$ mkdir ~/.docker
$ echo '<YOUR_RED_HAT_PULL_SECRET>' > ~/.docker/config.json
```

5. Configure oc-mirror and download Kubernetes Operators

Red Hat uses a tool called `oc-mirror` to help package up the necessary OpenShift components. `oc-mirror` requires a config file called `imageset.yaml`. This config file tells the tool exactly what components to pull down.

You can use the following imageset config file to pull the containers and operators that we recommend for most deployments. If you need additional operators that are not included in this file, we will explain how to accomplish this in the following section.

```
#This will be the imageconfig.yaml. Cut and paste into your command line.

$ cat >> imageset.yaml<< EOF
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
archiveSize: 4
mirror:
  platform:
```

```
 architectures:
  - amd64
 channels:
 - name: stable-4.15
  type: ocp
  minVersion: 4.15.12
 graph: true
operators:
- catalog: registry.redhat.io/redhat/redhat-operator-index:v4.15
 packages:
  - name: kubevirt-hyperconverged
   channels:
    - name: stable
  - name: kubernetes-nmstate-operator
   channels:
    - name: stable
  - name: cincinnati-operator
   channels:
    - name: v1
  - name: cluster-logging
   channels:
    - name: stable-5.8
  - name: compliance-operator
   channels:
    - name: stable
  - name: web-terminal
   channels:
    - name: fast
  - name: file-integrity-operator
   channels:
    - name: stable
  - name: lvms-operator
   channels:
    - name: stable-4.15
  - name: odf-operator
   channels:
    - name: stable-4.15
  - name: odf-csi-addons-operator
```

```
        channels:
          - name: stable-4.15
      - name: ocs-operator
        channels:
          - name: stable-4.15
      - name: mcg-operator
        channels:
          - name: stable-4.15
      - name: local-storage-operator
        channels:
          - name: stable
      - name: mtv-operator
        channels:
          - name: release-v2.6
      - name: nfd
        channels:
          - name: stable
  additionalImages:
  - name: registry.redhat.io/rhel9/rhel-guest-image:latest
  - name: registry.redhat.io/rhel8/rhel-guest-image:latest
  - name: registry.redhat.io/ubi8/ubi:latest
  - name: registry.redhat.io/ubi9/ubi:latest
  - name: registry.redhat.io/rhel8/support-tools
  - name: registry.redhat.io/rhel9/support-tools
  - name: registry.redhat.io/openshift4/ose-must-gather:latest
  - name: registry.redhat.io/odf4/odf-must-gather-rhel9:v4.15
  - name: registry.redhat.io/container-native-virtualization/cnv-must-gather-
rhel9:v4.15.1
EOF
```

The preceding command downloads some of the more common operators. If you would like to pull down additional ones, you can do so by running the following command:

```
$ for i in $(oc-mirror list operators --catalogs --version=4.15 | grep
registry); do $(oc-mirror list operators --catalog=$i --version=4.15 > $(echo
$i | cut -b 27- | rev | cut -b 7- | rev).txt); done
```

This will pull all operator names for OpenShift 4.15 and save them as a text file in your local directory. Update your `imageconfig.yaml` with the specific operators you want.

6. Run the oc-mirror command

On the connected bastion, run the `oc-mirror` command and pull down all images that are needed to stand up OpenShift and Day 2 operations. This process might take a while. You also need to ensure that `umask` is set to `0022` on STIG'd Red Hat Enterprise Linux (RHEL) servers to allow the `oc-mirror` process.

```
$ umask 0022
$ oc-mirror --config <imageset.yaml> file://.
```

Step 2: Transfer software to disconnected network

From here, you need to transfer the files over to the disconnected side however you see fit—`rsync`, `scp`, `sneakernet`, burn to a DVD, host it in an Apache server, etc.

You will need to move over the entire directory to include the binaries: `oc`, `oc-mirror`, `mini-quay`, etc. Use the following `scp` command to transfer files over from the connected STIGed bastion to the disconnected STIG bastion:

```
# scp -r * <username>@<disconnected_bastion_host>:.
$ scp -r * allen@airgap-stig:.
```

Step 3: Configure disconnected bastion host

On the disconnected side, let's unzip our ingredients and check to make sure everything is good to go.

You will also need to ensure that you update `fapolicy` to ensure that you do not run into issues.

```
#Untar our ingredients
$ tar xvf mirror-registry.tar.gz
$ tar xvf oc-mirror.tar.gz
$ tar xvf openshift-client-linux.tar.gz
$ tar xvf openshift-install-linux.tar.gz

#Move the tools and change ownership make executable
$ sudo cp {mirror-registry,oc,oc-mirror,openshift-install} /usr/local/bin/
$ sudo chown -R $USER /usr/local/bin/{mirror-registry,oc,oc-mirror,openshift-
install}
$ sudo chmod +x /usr/local/bin/{mirror-registry,oc,oc-mirror,openshift-
install}
$ sudo restorecon -v /usr/local/bin/{mirror-registry,oc,oc-mirror,openshift-
install}


#We need to disable FApolicyD for the install
```

```
$ systemctl stop fapolicyd.service

#We will need to add our tools to the fapolicyd trust
$ sudo fapolicyd-cli --file add /usr/local/bin/mirror-registry
$ sudo fapolicyd-cli --file add /usr/local/bin/oc
$ sudo fapolicyd-cli --file add /usr/local/bin/oc-mirror
$ sudo fapolicyd-cli --file add /usr/local/bin/openshift-install
$ sudo fapolicyd-cli --update

#We need to start FApolicyD for the install
$ systemctl start fapolicyd.service; systemctl status fapolicyd.service

#The STIG auto applies max user namespaces to zero, therefore rootless podman
will error out
#Query the current value
$ sysctl -a | grep max_user_namespaces
user.max_user_namespaces = 62372

#Query / change the stored value
$ grep namespaces /etc/sysctl.d/99-sysctl.conf

#Apply the new value
$ sysctl -p /etc/sysctl.d/99-sysctl.conf
```

Step 4: Configure mirror registry on bastion host

Because OpenShift is deployed as a set of containers, a registry is necessary to operate properly. In connected environments, OpenShift would use Red Hat's container registry for the initial installation, but in disconnected environments this will not be available. Therefore, you need to stand one up to serve this purpose.

The STIG modifies the user `bashrc` and profile to default to `0077`. During the mirroring process to your local registry, we also build your default catalog source. During that process, we need to ensure that the `umask` is set to `0022` so that OpenShift can read those files within the built container. This is necessary because by default, OpenShift cannot run containers as root for security reasons.

1. Install Red Hat Quay

First, install Red Hat Quay as follows:

```
#We had to change the umask from 0077 to 0022 for the quay install, due to the
fact that it always ssh'es into localhost and it get the umask from the
default /etc/bashrc and /etc/profile
$ sed :%s/077/022/gc /etc/bashrc
$ sed :%s/077/022/gc /etc/profile

#Install the mirror registry (Quay)
#We will need to open up port 8443 for Quay
$ firewall-cmd --add-port 8443/tcp --permanent
$ firewall-cmd --reload

#If you have a cert and key that you would like to use instead of having Quay
create a self-signed one please add the following
---
      --sslCert string          The path to the SSL certificate Quay should
use
      --sslCheckSkip            Whether or not to check the certificate
hostname against the SERVER_HOSTNAME in config.yaml.
      --sslKey string           The path to the SSL key Quay should use
  -H, --targetHostname string   The hostname of the target you wish to install
Quay to. This defaults to $HOST (default "rhel8-mirror-stig.airgap.dota-
lab.iad.redhat.com")
  -u, --targetUsername string   The user on the target host which will be used
for SSH. This defaults to $USER (default "allen")
----

$ mkdir -p /data/mirror-registry
$ ./mirror-registry install --quayHostname <Full FQDN> --quayRoot
/data/mirror-registry --quayStorage /data/mirror-registry --targetUsername
<user_name> -k ~/.ssh/my_ssh_key --initUser admin --initPassword redhat123
```

2. Test if Quay is up and functioning

Next, verify the installation:

```
#Copy certificates that created - Only do this if Quay created the certs for
you. Quay will create self-signed certs during the install process.

$ sudo cp -v /data/mirror-registry/quay-rootCA/rootCA.pem
/etc/pki/ca-trust/source/anchors/
$ sudo update-ca-trust
$ podman login <Full FQDN>:8443
```

> **Note:** if you are running into issues installing Quay, try using an alternative registry, as outlined in Appendix A.

3. Upload the bundled images into the registry

Once you've confirmed Quay is running, use the following commands to transfer the bundled images into the mirror registry:

```
#Starting the initial image transfer into local Quay - this may take a while
$ oc-mirror --max-per-registry 1 --from mirror_seq1_000000.tar docker://<Full
FQDN>:8443

#You should notice that your local Quay would start showing images under
https://<Full FQDN>:8443/repository

#You know if have completed correctly when the directory
oc-mirror-workspace/results-*/imageContentSourcePolicy.yaml
```

Step 5: Create install-config

Let's create the `install-config`; this is the file that tells CoreOS how to configure itself upon installation. You can find a sample `install-config.yaml` in the OpenShift documentation.

1. Grab pull secret

First, you need to grab the pull secret that will be copied into the config:

```
#We will need to get our auth from our local quay registry
$ podman login --authfile config.json <Full FQDN>:8443
$ jq -c . config.json ###This is your $<PULL_SECRET> that you will need to
copy
```

2. Cut and paste YAML

Next, cut and paste the following code into the command prompt (be sure to update it with the pull secret and environment-specific details):

```
$ vi install-config.yaml
---
apiVersion: v1
additionalTrustBundlePolicy: Always
baseDomain: dota-lab.iad.redhat.com <---  #Your domain name
metadata:
 name: stig-sno   <---  #Your domain name will get appended to this
compute:
- hyperthreading: Enabled
 name: worker
 replicas: 0
controlPlane:
 hyperthreading: Enabled
 name: master
 replicas: 3
networking:
 clusterNetwork:
 - cidr: 10.128.0.0/14   <---  #IP space for the local network made for within
OCP
   hostPrefix: 23
 machineNetwork:
 - cidr: 172.31.255.0/16 <---  #IP space of your physical box
 networkType: OVNKubernetes
 serviceNetwork:
 - 172.30.0.0/16         <---  #IP space for the local network made for within
OCP
platform:
 baremetal: <--- For a SNO installs only, replace baremetal with 'none:{}'
   apiVIPs:
   - 172.31.255.200 <---  #IP reserved for apiVIP, not needed for SNO
   ingressVIPs:
   - 172.31.255.201 <---  #IP reserved for ingressVIP, not needed for SNO
fips: true
pullSecret: '$PULLSECRET' <--- #Run this command $jq -c . config.json
```

```
sshKey: '$PUBLIC_RSA_KEY' <--- #Pull your pub key $cat ~/.ssh/id_rsa.pub

#You will need to copy the CA that Quay created from
#/data/mirror-registry/quay-rootCA/rootCA.pem
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  -----END CERTIFICATE-----
The output will look like the following:

##At the end of the oc-mirror command you will receive this bit of info
imageContentSources:
- mirrors:
 - <Full FDQN>:8443/openshift-release-dev/ocp-release
 source: quay.io/openshift-release-dev/ocp-release
- mirrors:
 - <Full FDQN>:8443/openshift-release-dev/ocp-v4.0-art-dev
 source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

Step 6: Create agent-config

Let's also create an `agent-config.yaml` file, which tells OpenShift how the network is configured:

```
$ vi agent-config.yaml
---
#Here is a sample agent-config.yaml for a static IP

apiVersion: v1beta1
kind: AgentConfig
metadata:
#same name as the cluster in install-config.yaml
  name: stig-sno
rendezvousIP: {{ static_ip_of_box }}
# All fields below are optional
additionalNTPSources: {{ ntp_ip }}
hosts:
# If a host is listed, then at least one interface
# needs to be specified.
- hostname: stig-sno-master-0
  role: master
  rootDeviceHints:
    deviceName: {{ /dev/sda }}
  # MAC Addresses of the NIC
  # interfaces are used to identify the host to apply this configuration to
  interfaces:
    - macAddress: {{ mac_address_1 }}
      name: eth0
    - macAddress: {{ mac_address_2 }}
      name: eth1
    - macAddress: {{ mac_address_3 }}
      name: eth2
    - macAddress: {{ mac_address_4 }}
      name: eth3
  # networkConfig contains the network configuration for the host in NMState
format.
  networkConfig:
    interfaces:
      - name: eth0
        type: ethernet
        state: up
        mac-address: {{ mac_address_1 }}
        ipv4:
          enabled: true
          address:
          #static IP of the box
            - ip: {{ static_ip_of_box }}
              prefix-length: 24
          dhcp: false
      - name: eth1
        type: ethernet
        state: down
        mac-address: {{ mac_address_2 }}
        ipv4:
          enable: false
      - name: eth2
        type: ethernet
        state: down
        mac-address: {{ mac_address_3 }}
    ipv4:
     enable: false
   - name: eth3
    type: ethernet
    state: down
    mac-address: {{ mac_address_4 }}
    ipv4:
     enable: false
  dns-resolver:
   config:
    server:
      - {{ router_ip }}
  routes:
   config:
    - destination: 0.0.0.0/0
     next-hop-address:{{ gateway_ip }}
     next-hop-interface: eth0
     table-id: 254
```

Step 7: Create OpenShift boot ISO

Agent-based installation is a subcommand of the OpenShift installer. It generates a bootable ISO image containing all of the assets required to deploy an OpenShift cluster, with an available OpenShift release image.

1. Install nmstatectl for networking support

Before you generate the image, you need to install the `nmstatectl` binary so that the agent config can do its networking for the CoreOS nodes:

```
$ sudo dnf install /usr/bin/nmstatectl -y
```

2. Create boot ISO Image

Then, create the custom ISO image so you can boot it:

```
$ umask 0022; openshift-install agent create image
```

Step 8: Load the ISO

Attach media and fire up those server(s). Make note of the URL, username, and password to access the GUI upon completion:

```
#Go to your IPMI and boot the ISO
#Attach the media and start the server! - check for UEFI!

#Back on your disconnected bastion complete the install
$ openshift-install agent wait-for bootstrap-complete
$ openshift-install agent wait-for install-complete

time="2024-05-28T13:44:54-04:00" level=info msg="Install complete!"
time="2024-05-28T13:44:54-04:00" level=info msg="To access the cluster as the
system:admin user when using 'oc', run\n    export
KUBECONFIG=/home/allen/stig/auth/kubeconfig"
time="2024-05-28T13:44:54-04:00" level=info msg="Access the OpenShift web-
console here: https://console-openshift-console.apps.stig-sno.dota-
lab.iad.redhat.com"
time="2024-05-28T13:44:54-04:00" level=info msg="Login to the console with
user: \"kubeadmin\", and password: \"abc-7v9AL-Bd7fI-xyz\""
```

Step 9: Post-installation tasks

1. Disable the default OperatorHub catalog sources

Operator catalogs that source content provided by Red Hat and community projects are configured for OperatorHub by default during an OpenShift Container Platform installation. In a restricted network environment, you must disable the default catalogs as a cluster administrator. You can then configure OperatorHub to use local catalog sources.

```
#Disable the sources for the default catalogs by adding
disableAllDefaultSources: true to the OperatorHub object:
$ export KUBECONFIG=auth/kubeconfig
$ oc patch OperatorHub cluster --type json \
    -p '[{"op": "add", "path": "/spec/disableAllDefaultSources", "value":
true}]'
```

2. Updating the catalog source

We need to tell OpenShift that we have a local mirror of all of the operators. Within your bastion host, you will see that `oc-mirror` created a directory called `oc-mirror-workspace`. Within that you will see a `results-xyz` directory. You will need to apply that directory as follows:

```
$ oc apply -f oc-mirror-workspace/results-xyz/catalogSource-redhat-operator-index.yaml
$ oc apply -f oc-mirror-workspace/results-xyz/imageContentSourcePolicy.yaml
$ oc apply -f oc-mirror-workspace/results-xyz/release-signatures/
```

Troubleshooting

The following list offers tips for troubleshooting common problems when performing bare metal OpenShift Container P installs:

- DNS issues—make sure your bastion host uses the same DNS as the servers.
- Certs do not link up with NTP, cert is ahead of NTP.
    - If needed, add NTP IP to `agent-config.yaml`.
- If the registry is not up, check the following:
    - Certs
    - File system issues with STIG
    - Ownership of files
- The `pull-secret` is not updated with the correct authentication JSON.
- `agent-config.yaml` with interfaces not called out in both “the upper section and lower section.”
- MAC addresses on `agent-config.yaml` are correct.

Tracked issues

Refer to the following list for current issues with the installer in a STIG environment.

`oc-mirror`:

- [OCPBUGS-26078] oc-mirror creates index image incorrectly with non-default umask
- https://issues.redhat.com/browse/OCPBUGS-31536

Docs:

- https://issues.redhat.com/browse/OCPBUGS-32231
- https://issues.redhat.com/browse/OCPBUGS-26078
- https://issues.redhat.com/browse/OCPBUGS-23386

- https://issues.redhat.com/browse/OCPBUGS-23549
- https://issues.redhat.com/browse/OCPBUGS-30607
- https://issues.redhat.com/browse/OCPBUGS-37931

References & documentation

- Red Hat Disconnected documentation (OpenShift Container Platform 4.15)
- Better securing the future: Navigating Red Hat OpenShift disconnected installations with the agent-based installer – Red Hat Blog
- OpenShift Data Foundation 4.15 - Disconnected prep
- Installing OpenShift Virtualization 4.15

Appendix A: Host the local mirror in a Docker registry

If you are facing issues with Quay, you can use the a lightweight Docker registry to host the local mirror as follows.

1. First, open up port 5000 to get through the firewall:

```
$ firewall-cmd --add-port 5000/tcp --permanent
$ firewall-cmd --reload
```

2. Next, create some directories for the container registry. Here, we are setting up and changing the owner to our current user:

```
$ mkdir -p /opt/registry/{auth,certs,data}
$ sudo chown -R $USER /opt/registry
```

3. Afterwards, create a certificate for your registry to use. If you have your own certificate, you can skip the OpenSSL step.

```
$ cd /opt/registry/certs
$ openssl req -newkey rsa:4096 -nodes -sha256 -keyout domain.key -x509 -days 
365 -addext "subjectAltName = DNS:<quay.disco.iad.redhat.com>" -out domain.crt
#Country Name (2 letter code) [XX]:<US>
#State or Province Name (full name) []: <Washington>
#Locality Name (eg, city) [Default City]: <DC>
#Organization Name (eg, company) [Default Company Ltd]: <Red Hat>
#Organizational Unit Name (eg, section) []: <Disco>
#Common Name (eg, your name or your server's hostname) []:<registry.dota-
lab.iad.redhat.com>
#Email Address []:<your-email-address>test@redhat.com
```

The common name is the one that matters; the rest of these can be pretty much any value, but the common name must be the correct name for your machine in order for the certificate to properly resolve.

4. Next, add simple password authentication to your registry. We will use the username `openshift` and the password `redhat` for demonstration purposes. If you do not have `htpasswd`, you can always install it with a `yum install httpd-tools` command.

```
$ htpasswd -bBc /opt/registry/auth/htpasswd <username> <password>
#For example:
#$ htpasswd -bBc /opt/registry/auth/htpasswd openshift redhat
```

5. Now you can set up your registry to run. Use the password and certificate we created and automatically start in case the virtual machine ever restarts:

```
$ podman run -d --name mirror-registry \
-p 5000:5000 --restart=always \
-v /opt/registry/data:/var/lib/registry:z \
-v /opt/registry/auth:/auth:z \
-e "REGISTRY_AUTH=htpasswd" \
-e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
-e "REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd" \
-v /opt/registry/certs:/certs:z \
-e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt \
-e REGISTRY_HTTP_TLS_KEY=/certs/domain.key \
docker.io/library/registry:2
```

6. Validate the connection to the registry server:

```
$ curl -u openshift:redhat https://<Full FDQN>:5000/v2/_catalog
```

7. You see the certificate is not trusted, but we can temporarily ensure it works by ignoring the certificate verification with a `-k`.

```
$ curl -u <username>:<password> -k https://<Full FDQN>:5000/v2/_catalog
#For example
$ curl -u openshift:redhat -k https://registry.rhel8-mirror-stig.dota-
lab.iad.redhat.com:5000/v2/_catalog
```

8. It works! Now we need to fix the trust issues by copying the certificate to the systems trust store and updating the CA trust.

```
$ sudo cp /opt/registry/certs/domain.crt /etc/pki/ca-trust/source/anchors/
$ sudo update-ca-trust
```

9. Now, test it again to ensure everything works:

```
$ curl -u <username>:<password>  https://<Full FDQN>:5000/v2/_catalog
```

Appendix B: The Compliance Operator

All versions of OpenShift entitle the user to a Kubernetes Operator called the Compliance Operator. You can use this operator to automate the application of STIG controls on the system. For a helpful explanation of this process, read the blog post Accelerate STIG compliance with Red Hat OpenShift's built-in security features: From 40 CAT I items to 7.

The Defense Information Systems Agency (DISA) published the Red Hat OpenShift Container Platform 4.12 Security Technical Implementation Guide (OpenShift STIG).

You can learn more about the compliance operator by referring to the official Red Hat OpenShift documentation.