### **How to Create local private registry server for OpenShift 4.3 deployment**

### **Introduction:**

### In order to deploy OpenShift container platform in restricted environment (also termed as disconnected and Air-Gap cluster), a private

### registry server is needed. This registry server runs as container and serves as content provider for OCP 4.3.x. For it to serve

### contents to deployment, all the required contents (images) must be first mirrored to this registry server.

### The private registry can be created on NPS server as well as on any non-NPS node. The node can be a VM or a physical server. User will require to fill all details in input\_vars.yaml and then will have to trigger playbook. The flag **“registry\_on\_nps:”,** decides if registry script should be executed on NPS node or non-NPS. Value **“false”** means registry will be configured on server other than NPS node. Post successful completion the automation creates binary, named **'openshift-install'** and a text file, named **'mirrorimage.txt'** in '/root' directory on same server on which this script is executed. These files will have to be transferred to NPS node in directory "/var/nps/ISO/".

### The detailed procedure is described below.

### **Pre-Requisites:**

### Download **openshift client (oc)** binary and **pull secret** from RedHat OpenShift portal. Please refer to **“Download RHOCP and HPE Nimble CSI driver artifacts”** of run-book for more information on downloading artefacts.

### Internet connectivity must be available to the node from where this script is executed.

### A valid RedHat subscription credentials with OpenShift entitlements is needed.

### **Hardware requirements:**

* RHEL 7.x server, this can be a Virtual Machine as well as physical server.

Though there are no minimum resource requirements has been mentioned by RedHat, but as per our validation, we recommend following:

* + RAM: 4096GB
  + vCPU: 2
  + Disk: 300GB+ (in /root partition)

**Note:-** Though around 125GB (in /root partition ) is sufficient to store **deployment images**, **OpenShift-Samples operator**

**images** and **red hat operators images** but due to the fact that this is disconnected environment and hence may need to mirror

images from other operator hubs if needed and **application images** too need to be first mirrored on to this registry

server, enough storage should be present on the node from where this script is executed.

### **Software requirements:**

* Ansible version > 2.8 is required

### **Usage:**

### Download the toolkit software and navigate to registry-utils/offline-registry. User inputs are collected in file 'input\_vars.yaml'.

### **1. Fill input\_vars.yaml**

### cluster\_info:

### proxy:

### proxy\_port: 'proxy port number' *===> Organizational proxy port number*

### proxy\_user: "proxy username" *===> Organizational username*

### proxy\_password: "proxy password" *===> Organizational proxy password*

### proxy\_ip: < proxy ip used for connecting to internet > *===> Organizational proxy IPv4 address*

### registry\_info:

### username: <username> *===> username for private registry server*

### password: <password> *===> password for private registry server*

### registry\_host\_fqdn: <fqdn of registry host> *===> FQDN of server on which registry container is executed.*

### port: *===> Port number for registry server*

### certs\_info:

### C: "" *===> country\_name*

### ST: "" *===> state\_or\_province\_name*

### L: "" *===> locality\_name*

### O: "" *===> organization\_name*

### OU: "" *===> organizational\_unit\_name*

### duration: "<number of days>" *===> Number of days till this certificate is valid.*

### email\_address: registry@example.com *===> emailAddress field of the certificate signing request subject*

### redhat\_subscription:

### username: <username> *===> RH customer portal username*

### password: <password> *===> RH customer portal password*

### poolid: <pool id> *===> pool id with openshift entitlements*

### artefacts:

### openshift\_client\_path: "<path to oc binary .gz>" *===> Absolute path of openshift client binary downloaded from.*

### *RedHat portal*

### pull\_secret\_path: "<path to pull-secret.txt>" *===> Absolute path of pull-secret.txt downloaded from RedHat*

### *portal*

### track\_folder: /var/run/rhocp/

### openshift\_version: <openshift image tag version> *===> Obtained it from below link:*

### *URL:*[*https://quay.io/repository/openshift-release-*](https://quay.io/repository/openshift-release-)*dev/ocp-release?tab=tags*

### ***example: 4.3.10-x86\_64***

### registry\_on\_nps: false *===> keep it false if private registry is to be created on Non-NPS*

### *server.*

### **2. Execution**

### The playbook, create\_registry\_server.yaml located in “nps-rhocp/registry-utils/offline-registry directory”, configures private registry in following 2 phases:

### It registers the server on which this script is being executed to RedHat CDN

### Configures the private registry and mirrors contents from quay.io

The private registry creation can be run in following modes, with root permission.:

a) To configure the registry run:

# ansible-playbook create\_registry\_server.yaml

Supported Tags:

=============

a) When only subscription is to be executed:

# ansible-playbook create\_registry\_server.yaml --tags "subscription"

b) When subscription is done and if re-subscription is not needed, only registry creation play can be executed:

**# ansible-playbook create\_registry\_server.yaml --tags "configure"**

c) When registry is configured and only validation check is to be performed:

**# ansible-playbook create\_registry\_server.yaml --tags "validate"**

### **Verification:**

Follow following procedure to validate if registry is installed successfully:

From the registry server execute following (as root user):

**Automation**:

1. # ansible-playbook create\_registry\_server.yaml --tags "validate"

**Manual**:

1. curl -u <registry-username>:<registry-password> -k https://<registry-fqdn>:<user provided port>/v2/\_catalog

Desired output on console:

{"repositories":["ocp4/openshift4"]}

1. docker login -u <registry-username> -p <registry-password> <registry-fqdn>:<user provided port>

Desired output on console:

Login Succeeded

### Troubleshooting:

In case of any failure you can just re-run the playbook.