

ONAP offline deployment

21.6.2018

- Many of the corporate deployments are w/o public internet connectivity
- To have full control over deployed ONAP
(avoid build/runtime dependencies which are pulled from internet)
=> *“might lead to unpredicted behavior”*
- To introduce custom ONAP solutions, not upstreamed (having common repository for those in local nexus)

“The main principle in place is that all

docker images

git repos

pip packages

deb/rpm packages

cloudify blueprints

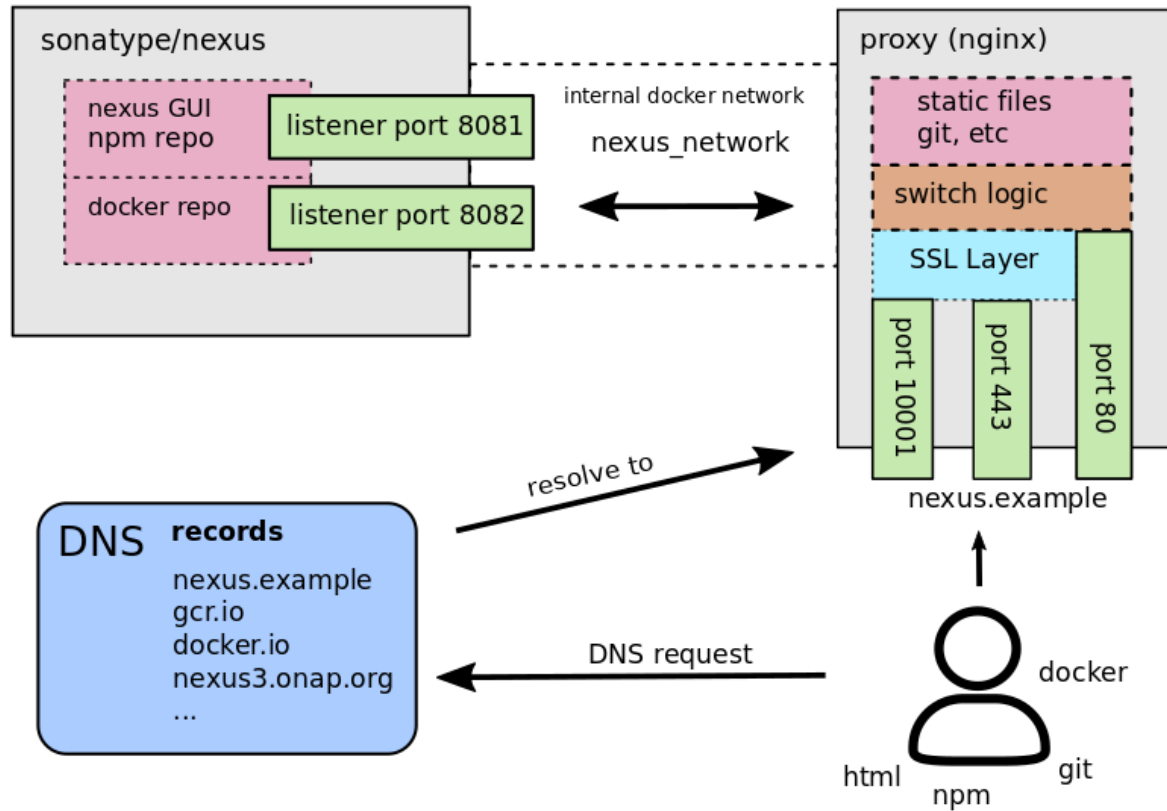
files

...

which are needed during ONAP OOM installation are downloaded during preparation/packaging part and pulled into local nexus accessible via nginx proxy.

Original DNS Domains remains unchanged in original deployment scripts and are simulated by nginx.”

Basic Concept



Challenge: *“How to collect all online dependencies, store it in the right way and use them later for offline deployment ?”*

Solution: *introducing own local nexus*

Current approach: *due to many dependencies in runtime, the only reasonable way for now is to perform “online” ONAP deployment and collect all artifacts for offline deployment*

Target approach: *runtime dependencies are considered as bugs or bad practice, we need to continuously challenge them and remove if possible (e.g. by adding them into build time), when runtime downloaded artifacts are gone we can continuously improve tooling for collecting artifacts like pre-pull image collecting script*

Preparation of local nexus:

Step 1) get standard sonatype/nexus3 image

Step 2) configure:

> docker (hosted) repo

> npm (hosted) repo

> Security > Realms (Add Docker Bearer / Npm Bearer Token Realm)

> Security > Users (Add docker user)

Step 3) Collect all required images and push them into this “local nexus”

E.g. # docker push \$image_name:\$tag

Step 4) Collect all required npm packages and publish them into “local nexus”

E.g. # npm publish \$i --access public

Apart of local nexus stuff and as a part of “offline deployment tooling” following artifacts are also collected & used for deployment:

- a) git-repos (so, demo.git, testsuite, ...)*
- b) rpm/deb packages (docker, make, ansible pkgs, dnsmasq, nfs common, ...)*
- c) nginx/nexus images*
- d) helm / kubectl / rancher binaries*
- e) Few special files under following dirs (make them accessible via nginx)*
 - github.com/*
 - nexus.onap.org/*
 - www.getcloudify.org/*
 - www.springframework.org/*

OOM related tweaks: "There are couple of places in OOM project we need to patch to get ONAP working for offline deployments, mainly it's about setting up repository to our local nexus or pushing root certificate into pods necessary for https traffic towards our simulated dns domains.

E.g.
./oom/kubernetes/dcaegen2/charts/dcae-cloudify-manager/templates/deployment.yaml

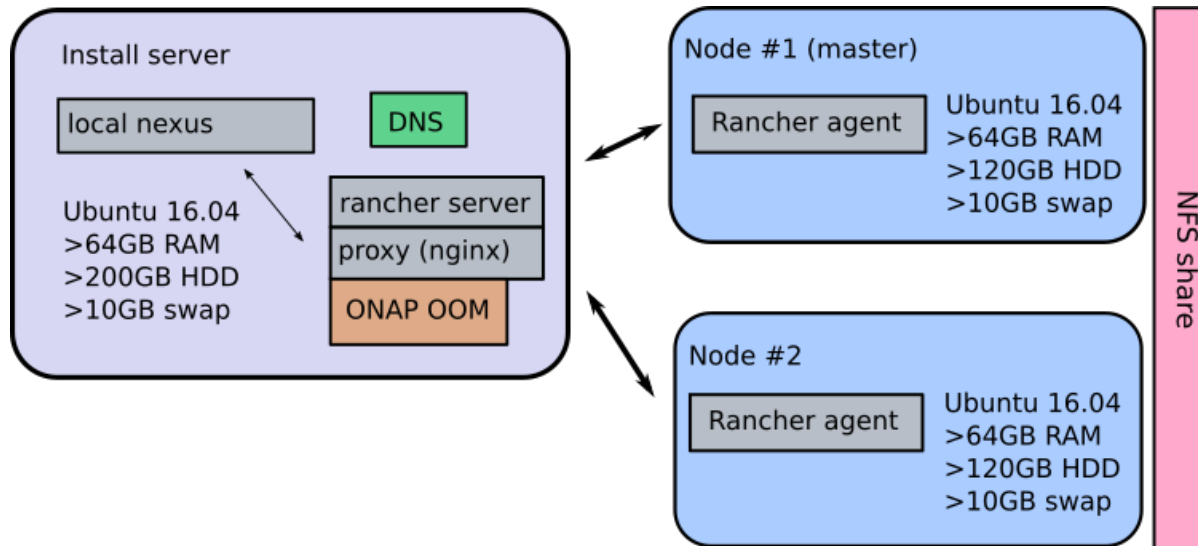
```
<     - mountPath: /etc/pki/ca-trust/source/anchors
<     name: root-ca
85,86d82
<         echo -e '\nREQUESTS_CA_BUNDLE="/etc/ssl/certs/ca-bundle.crt"' >>
/etc/sysconfig/cloudify-restservice
<         update-ca-trust extract
100,102d95
<     - name: root-ca
<     hostPath:
<     path: /usr/local/share/ca-certificates/extra
```


ONAP offline deployment

Installation procedure consists of 2 steps:

Step 1) Deploy infrastructure from repository

- On Install server – setup nexus, nginx, rancher server and DNS (simulating used dns domains), docker, modified OOM repo
- On k8s nodes – deploy docker, rancher agent, setup NFS share across nodes



After Step 1 k8s cluster is configured

```
root@oom-beijing-xenial-offline-master:~/install# kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
oom-beijing-xenial-offline-compute1 Ready    <none>   32m    v1.8.10-rancher1
oom-beijing-xenial-offline-compute2 Ready    <none>   27m    v1.8.10-rancher1
```

DNS Domains are simulated

```
root@oom-beijing-prerc2-master:~# cat /etc/hosts
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

10.2.2.7 oom-beijing-preRC2-master
10.2.2.5 oom-beijing-preRC2-compute1
10.2.2.3 oom-beijing-preRC2-compute2
10.2.2.7 gcr.io git.rancher.io gerrit.onap.org registry-1.docker.io docker.io
10.2.2.7 registry.npmjs.org nexus3.onap.org nexus.onap.org docker.elastic.co
10.2.2.7 www.getcloudify.org www.springframework.org registry.hub.docker.com
10.2.2.7 nexus.oom-beijing-preRC2-master
root@oom-beijing-prerc2-master:~#
```

Step 2) Deploy ONAP (OOM way) on top of it