

Deploy Connected K8s Clusters

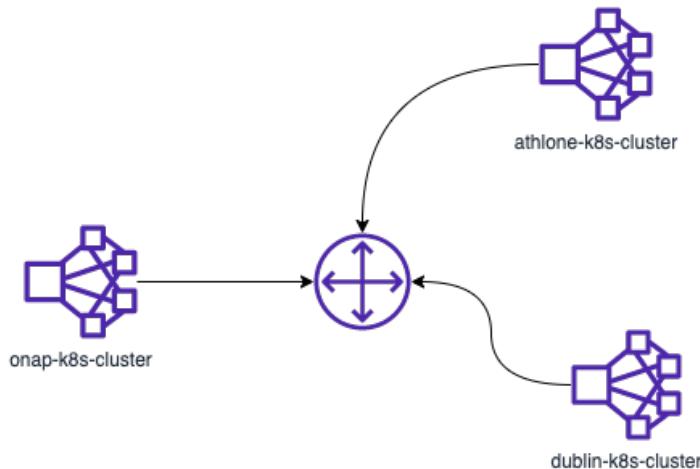
For deploying multiple connected clusters we are using [infra/engine](#) project.

```
git clone "ssh://nauman@gerrit.nordix.org:29418/infra/engine"
```

Deployment

We are going to deploy the following k8s clusters and then connect them together afterward.

- onap-k8s-cluster
- dublin-k8s-cluster
- athlone-k8s-cluster



For deploying onap-k8s-cluster we are using 10.1.0.0/24 as cluster network CIDR.

```
CIDR_BLOCK="10.1.0.0/24"
HEAT_STACK_NAME="onap-k8s-stack"
bash -x ./deploy.sh
-c k8-multus-plugins
-s kubernetes
-b master
-d ubuntu1804
-p file://${HOME}/nordix/infra/provisioner/heat/playbooks/roles/create-stack/files/heat-environment-xerces-
ubuntu1804.yaml
-i file://${HOME}/nordix/infra/provisioner/heat/playbooks/roles/create-stack/files/heat-idf.yaml
-o /home/ubuntu/.openstack/est_onap.sh
```

For deployment of dublin and athlone clusters, we need to use different CIDR block and stack name env vars.

```
CIDR_BLOCK="10.2.0.0/24"
HEAT_STACK_NAME="dublin-k8s-stack"
bash -x ./deploy.sh
-c k8-multus-plugins
-s kubernetes
-b master
-d ubuntu1804
-p file://${HOME}/nordix/infra/provisioner/heat/playbooks/roles/create-stack/files/heat-environment-xerces-
ubuntu1804.yaml
-i file://${HOME}/nordix/infra/provisioner/heat/playbooks/roles/create-stack/files/heat-idf.yaml
-o /home/ubuntu/.openstack/est_onap.sh
```

```

CIDR_BLOCK="10.3.0.0/24"
HEAT_STACK_NAME="athlone-k8s-stack"
bash -x ./deploy.sh
-c k8-multus-plugins
-s kubernetes
-b master
-d ubuntu1804
-p file:///${HOME}/nordix/infra/provisioner/heat/playbooks/roles/create-stack/files/heat-environment-xerces-
ubuntu1804.yaml
-i file:///${HOME}/nordix/infra/provisioner/heat/playbooks/roles/create-stack/files/heat-idf.yaml
-o /home/ubuntu/.openstack/est_onap.sh

```

Networking

After successful execution of above commands first we need to release floating IPs used by `athlone-k8s-stack` and `dublin-k8s-stack` mapped to their jumphosts.

Now we have to remove gateway interfaces from `router.dublin-k8s-stack` having address `10.2.0.1` and `router.athlone-k8s-stack` having address `10.3.0.1` using openstack web interface network/routing section

And add them to `router.onap-k8s-stack` by selecting the following subnets

`10.2.0.0/24` (`subnet.dublin-k8s-stack`)

`10.3.0.0/24` (`subnet.athlone-k8s-stack`)

with

`10.2.0.1` and `10.3.0.1` as respective gateway ip addresses.

Security Groups

For allowing traffic between different network subnets we have to create new rules in following security groups.

internal-sg.onap-k8s-stack

Add two new rules that allows all tcp traffic on all ports from following security groups

- `internal-sg.dublin-k8s-stack`
- `internal-sg.athlone-k8s-stack`

internal-sg.dublin-k8s-stack

Similarly add two new rules that allows all tcp traffic on all ports from following security groups

- `internal-sg.onap-k8s-stack`
- `internal-sg.athlone-k8s-stack`

internal-sg.athlone-k8s-stack

Also add two new rules that allows all tcp traffic on all ports from following security groups

- `internal-sg.onap-k8s-stack`
- `internal-sg.dublin-k8s-stack`

Kube Config

Post deployment we have kubeconfig files in following jumphosts

- `jumphost.onap-k8s-stack` (`ssh root@$jumphost.onap-k8s-stack-ip`)
- `jumphost.dublin-k8s-stack` (`ssh root@$jumphost.dublin-k8s-stack-ip`)
- `jumphost.athlone-k8s-stack` (`ssh root@$jumphost.athlone-k8s-stack-ip`)

After ssh into all jump-host copy contents of `~/.kube/config` and merge and place them in `jumphost.onap-k8s-stack`.

 We have removed the floating ip addresses of dublin & athlone jumphost instances but we can ssh into them using their private ip address through onap jumphost

The merged kubeconfig will look similar to

```

apiVersion: v1
kind: Config
preferences: {}
current-context: onap
clusters:
- cluster:
    certificate-authority-data: ...
    server: https://10.1.0.185:6443
    name: cluster.onap
- cluster:
    certificate-authority-data: ...
    server: https://10.3.0.11:6443
    name: cluster.athlone
- cluster:
    certificate-authority-data: ...
    server: https://10.2.0.206:6443
    name: cluster.dublin
contexts:
- context:
    cluster: cluster.onap
    user: kubernetes-admin-cluster.onap
    name: onap
- context:
    cluster: cluster.athlone
    user: kubernetes-admin-cluster.athlone
    name: athlone
- context:
    cluster: cluster.dublin
    user: kubernetes-admin-cluster.dublin
    name: dublin
users:
- name: kubernetes-admin-cluster.onap
  user:
    client-certificate-data: ...
    client-key-data: ...
- name: kubernetes-admin-cluster.athlone
  user:
    client-certificate-data: ...
    client-key-data: ...
- name: kubernetes-admin-cluster.dublin
  user:
    client-certificate-data: ...
    client-key-data: ...

```

After placing the merged kubeconfig in onap jumphost the following command should output cluster info

```

kubectl cluster-info --context onap
kubectl cluster-info --context dublin
kubectl cluster-info --context athlone
# Kubernetes master is running at https://10.1.0.185:6443

```