Table of Contents

Network Policy Lab

In this lab, you explore the use of network policy to protect a simple application, and then you give a separate project access to the database pod.

Goals

- Enable `ovs-networkpolicy` plug-in
- Set up project request template with network policy
- Allow access between related projects

---

1. Enable Network Policy Plug-in

The `networkpolicy` SDN plug-in is not enabled by default in OpenShift. You must reconfigure all of the masters and nodes and restart the appropriate processes in order to use it.

1. Confirm which SDN plug-in is running right now:

```
oc get clusternetwork
```

Sample Output

```
NAME      CLUSTER NETWORKS   SERVICE NETWORK   PLUGIN NAME
default   10.1.0.0/16:9      172.30.0.0/16     redhat/openshift-ovs-subnet
```

2. Run the following Ansible commands to change the plug-in and restart the services:

```
# reconfigure masters and restart
ansible masters -m shell -a "sed -i -e 's/openshift-ovs-subnet/openshift-
ovs-networkpolicy/g' /etc/origin/master/master-config.yaml"
ansible masters -m shell -a "/usr/local/bin/master-restart api"
ansible masters -m shell -a "/usr/local/bin/master-restart controllers"

# be patient - look at the logs and make sure it comes up.

# reconfigure all node_group config maps
oc get cm -n openshift-node -o yaml | sed -e 's/ovs-subnet/ovs-
networkpolicy/' | oc apply -f -

# restart the nodes pods
ansible nodes -m shell -a "systemctl restart atomic-openshift-node"

# be patient - look at the logs and make sure it comes up.

# restart the opensvswitch pods
oc delete pods -n openshift-sdn --all

oc get pods -n openshift-sdn -o wide -w
```

3. Verify the cluster network plug-in again:

```
oc get clusternetwork
```

Sample Output

```
NAME      CLUSTER NETWORKS   SERVICE NETWORK   PLUGIN NAME
default   10.1.0.0/16:9      172.30.0.0/16     redhat/openshift-ovs-
networkpolicy
```

2. Set Up Project Request Template with Network Policy

Network policies are scoped to namespaces. There are no default policies in OpenShift. In this section, you create some policies so that new projects are protected. You do this by creating, editing, and enabling a default project template.

1. Label the default namespace `name=default`:

```
oc label namespace default name=default
```

2. Create a bootstrap project request template file:

```
oc adm create-bootstrap-project-template -o yaml > template.yaml
```

3. Append a network policy to the template file that does the following:
   - Allows all pods within the project to intercommunicate
   - Allows access from pods in the default namespace to all pods in the project

```
objects:
- apiVersion: networking.k8s.io/v1
  kind: NetworkPolicy
  metadata:
    name: allow-same-and-default-namespace
  spec:
    ingress:
    - from:
      - podSelector: {}
    - from:
      - namespaceSelector:
          matchLabels:
            name: default
```

4. Create the template in the default namespace:

```
oc create -f template.yaml -n default
```

5. Edit all of the `master-config.yaml` files on all of the masters to point the `projectRequestTemplate` parameter to the new `default/project-request` template, and then restart the appropriate control-plane pods to pick up the changes:

```
ansible masters -m lineinfile -a 'path=/etc/origin/master/master-
config.yaml regexp="projectRequestTemplate" line="
projectRequestTemplate: \"default/project-request\""'

ansible masters -m shell -a '/usr/local/bin/master-restart api;
/usr/local/bin/master-restart controllers'
```

6. Validate the default project request template:

   a. First, examine an existing namespace with no network policies:

   ```
   oc get netpol -n default
   ```

   b. Create a new namespace and examine the network policies:

   ```
   oc new-project test-netpol
   oc get netpol -n test-netpol
   oc get netpol -n test-netpol -o yaml
   ```

3. Allow Access Between Related Projects

In this section, you create two projects. Due to the default policies you created above, the projects are isolated from each other—pods in one project cannot access pods in another. You create a network policy to allow access from the application in one project to the database in another project.

3.1. Set Up Projects and Get Credentials

1. Create a **`collective`** project with a Perl/Dancer database application that has persistent storage:

   ```
   oc new-project collective
   oc new-app dancer-mysql-persistent
   ```

2. Create a **`workers`** project in which a CakePHP application named "contributor" contributes data to the **`commons`** database in the **`collective`** namespace:

   ```
   oc new-project workers
   oc new-app cakephp-mysql-persistent
   ```

3. Get the database credentials from the **`collective`** project's secrets for the contributor to use:

```
# oc get secret dancer-mysql-persistent -o yaml -n collective | awk
'/database/ { print $2 }'

# or

oc get secret dancer-mysql-persistent -o yaml -n collective
```

Sample Output

```
apiVersion: v1
data:
  database-password: YkRyeUgwSjU=
  database-user: dXNlckpZTw==
  keybase:
dHF1NDd1cjdhdml5MDVmcTE4cmlrOHJoamN4c2dkZGxpY3h1ZWNxY3h0dDd2ZDhvcmd2Zm5rZX
g1djdrdmQ3dnc0b2MzcmZ5bHZyM3c4NHEyNmpmb2N4OHQxcW04N3VxMDEybnZrbzB4NGRmdGVj
bWt5Yjh2cGpxN2lmZDhpYQ==
kind: Secret
metadata:
  annotations:
    openshift.io/generated-by: OpenShiftNewApp
  creationTimestamp: 2018-11-29T19:16:57Z
  labels:
    app: dancer-mysql-persistent
    template: dancer-mysql-persistent
  name: dancer-mysql-persistent
  namespace: collective
  resourceVersion: "156028"
  selfLink: /api/v1/namespaces/collective/secrets/dancer-mysql-persistent
  uid: 572e7549-f40b-11e8-96f7-025a09226ffc
type: Opaque
```

4. Decode the **`database-password`** and **`database-user`** values to get their real values and make note of them:

```
# oc get secret dancer-mysql-persistent -n collective -o yaml | awk
'/database/ { print $2 }' | xargs -I {} sh -c "echo {} | base64 -d; echo "

# or

echo YkRyeUgwSjU= | base64 -d
echo dXNlckpZTw== | base64 -d
```

5. Examine the service layer of the **collective** project:

```
oc get svc -n collective
```

Sample Output

```
NAME                       TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
database                   ClusterIP   172.30.73.81     <none>
5432/TCP   20h
rails-pgsql-persistent   ClusterIP   172.30.163.207   <none>
8080/TCP   20h
```

6. Validate the hostname of the database service in the collective namespace:

```
# host database.collective.svc
```

Sample Output

```
database.collective.svc.cluster.local has address 172.30.73.81
```

3.2. Create Network Policy to Connect Across Projects

In this section, you connect from the **workers** database pod's **mysql** client to the **collective** database using the **mysql** client.

1. Use **rsh** to access the **workers** database pod:

```
oc project workers
oc get pods
oc rsh mysql-<database pod>
```

2. Use the **`database-password`** and **`database-user`** values you noted earlier to create a table with some data for the contributors to add to:

```
mysql -u <database-user> -p"<database-password>" -h
database.collective.svc -D sampledb
```

   - Expect this command to hang.
3. Press **Ctrl+C** to abort the command.
4. Press **Ctrl+D** to exit the pod.
5. Label the **`workers`** namespace so it can be referenced in a network policy:

```
oc label namespace workers name=workers
```

6. Create the network policy to allow access to the **`collective`** project's database labeled **`name=database`** on port **`3306`** from the **`workers`** project labeled **`name=workers`**.

```
cat << EOF |
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-workers-project-to-database
spec:
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          name: workers
    ports:
    - port: 3306
      protocol: TCP
  podSelector:
    matchLabels:
      name: database
EOF
oc create -n collective -f -
```

When determining ingress, you can only indicate entire namespaces. You cannot use **`podSelector`** to select pods in other namespaces.

7. Try connecting to the MySQL database from the **`workers`** project again:

```
oc rsh mysql-<database pod>

mysql -u <database-user> -p"<database-password>" -h
database.collective.svc -D sampledb
```

Sample Output

```
mysql: [Warning] Using a password on the command line interface can be
insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4138
Server version: 5.7.21 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql>
```

- Expect to have connectivity between the databases.

4. Clean Up Environment

1. Delete the two projects you created:

```
oc delete project collective
oc delete project workers
```

Build Version: 2.12 : Last updated 2019-03-27 09:54:17 EDT