RBAC - Administrator Howto

Role-based access control adds the ability to perform authentication and authorization of activities performed against a Kubernetes cluster. Authentication is concerned with the "who" and authorization is concerned with the "what".

User information is stored in OpenLDAP running in a container on your CaaSP Admin Node. You use standard LDAP administration tools for managing these users.

By default, when you create the first user in Velum during bootstrap of your cluster, that user is granted "Cluster Administrator" rights within Kubernetes. You can add additional users with these rights by adding new entries into the LDAP directory.

Before performing any administrative functions on the OpenLDAP instance, you will need to retrieve your OpenLDAP admin account password. To do this, SSH to your admin node and run the following command:

```
docker exec -it $(docker ps | grep openldap | awk '{print $1}') cat /var/lib/misc/infra-s
```

Adding new users

To add a new user, create a LDIF file like this:

```
dn: uid=<userid>,ou=People,dc=infra,dc=caasp,dc=local
objectClass: person
objectClass: inetOrgPerson
objectClass: top

uid: <userid>
userPassword: <password hash>
givenname: <first name>
cn: <full name>
sn: <surname>
mail: <email address>
```

To generate the password hash, install the `openldap2` package:

```
sudo zypper install openldap2
```

Then generate the password hash:

```
/usr/sbin/slappasswd
```

Be sure to replace all the parameters in the template above, indicated as `<....>`. The `mail` attribute value is used as the login to Velum and Kubernetes.

Populate your OpenLDAP server with this LDIF file:

```
ldapadd -H ldap://<ADMIN NODE IP>:389 -ZZ -D cn=admin,dc=infra,dc=caasp,dc=local -w <LDAP
```

Adding users to the Administrators group

To add this new user to the existing Administrators group, create a new LDIF file like this:

```
dn: cn=Administrators,ou=Groups,dc=infra,dc=caasp,dc=local
changetype: modify
add: uniqueMember
uniqueMember: uid=<userid>,ou=People,dc=infra,dc=caasp,dc=local
```

Be sure to replace all the parameters in the template above, indicated as `<....>`.

Populate your OpenLDAP server with this LDIF file:

```
ldapmodify -H ldap://<ADMIN NODE IP>:389 -ZZ -D cn=admin,dc=infra,dc=caasp,dc=local -w <L
```

Adding a new group

Scenario: You have users that you want to grant access to manage a single namespace in Kubernetes.

LDAP

To do this, first create your users as mentioned in **Adding new users** above, then create a new group:

```
dn: cn=<group name>,ou=Groups,dc=infra,dc=caasp,dc=local
objectclass: top

objectclass: groupOfUniqueNames
cn: <group name>
uniqueMember: uid=<member1>,ou=People,dc=infra,dc=caasp,dc=local
uniqueMember: uid=<member2>,ou=People,dc=infra,dc=caasp,dc=local
uniqueMember: uid=<member3>,ou=People,dc=infra,dc=caasp,dc=local
uniqueMember: uid=<member4>,ou=People,dc=infra,dc=caasp,dc=local
uniqueMember: uid=<member5>,ou=People,dc=infra,dc=caasp,dc=local
```

Repeat the `uniqueMember` attribute for every member of this group.

Populate your OpenLDAP server with this LDIF file:

```
ldapadd -H ldap://<ADMIN NODE IP>:389 -ZZ -D cn=admin,dc=infra,dc=caasp,dc=local -w <LDAP
```

Kubernetes

You must create a role binding to allow this new LDAP group access in Kubernetes.

Create a Kubernetes deployment descriptor like this:

```
---
# Define the Role's permissions in Kubernetes
kind: Role
apiVersion: rbac.authorization.k8s.io/v1beta1
```

```
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: <group name>
  namespace: <applicable namespace>
rules:
- apiGroups: [""]
  resources: [""]
  resourceNames: [""]
  verbs: [""]
---
# Map a LDAP group to this Kubernetes role
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: <group name>
  namespace: <applicable namespace>
subjects:
- kind: Group
  name: <name of LDAP group>
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: <group name>
  namespace: <applicable namespace>
  apiGroup: rbac.authorization.k8s.io
```

Add this role and binding to Kubernetes:

```
kubectl apply -f <DEPLOYMENT DESCRIPTOR FILE>
```

See https://kubernetes.io/docs/admin/authorization/rbac/for more details on authorization in Kubernetes.