Configuring SecurityContext for Containers

Relevant Documentation

- Configure a Security Context for a Pod or Container

Exam Tips

- A container's SecurityContext allows you to control advanced security-related settings for the container.
- Set the container's user ID (UID) and group ID (GID) with `securitContext.runAsUser` and `securityContext.runAsGroup`.
- Enable or disable privilege escalation with `securityContext.allowPrivilegeEscalation`.
- Make the container root filesystem read-only with `securityContext.readOnlyRootFilesystem`.

Lesson Reference

Log in to the **control plane node**.

Create a Pod that uses some custom securityContext settings.

These settings will:

- Run the container as user ID 3000.
- Run the container with group ID 4000.
- Disable privilege escalation mode on the container process.

```
vi securitycontext-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: securitycontext-pod
spec:
  containers:
  - name: busybox
    image: busybox:stable
    command: ['sh', '-c', 'while true; do echo Running...; sleep 5; done']
    securityContext:
      runAsUser: 3000
      runAsGroup: 4000
      allowPrivilegeEscalation: false
      readOnlyRootFilesystem: true
```

```
kubectl apply -f securitycontext-pod.yml
```

Check the user and group ID used by the container.

```
kubectl exec securitycontext-pod -- id
```

Try to write to a file inside the container.

```
kubectl exec securitycontext-pod -- sh -c "echo test > test.txt"
```

This will fail, since user ID 3000 has not been given permissions to write to the file.