

Using ServiceAccounts

Relevant Documentation

- [Configure Service Accounts for Pods](#)
- [Managing Service Accounts](#)
- [Using RBAC Authorization](#)

Exam Tips

- ServiceAccounts allow processes within containers to authenticate with the Kubernetes API server.
- You can set the Pod's ServiceAccount with `serviceAccountName` in the Pod spec.
- The Pod's ServiceAccount token is automatically mounted to the Pod's containers.

Lesson Reference

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

Create a ServiceAccount.

```
vi my-sa.yml
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: my-sa
automountServiceAccountToken: true
```

```
kubectl apply -f my-sa.yml
```

Create a Pod that uses the new ServiceAccount.

```
vi sa-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: sa-pod
spec:
  serviceAccountName: my-sa
  containers:
  - name: busybox
    image: radial/busyboxplus:curl
    command: ['sh', '-c', 'while true; do curl -s --header "Authorization: Bearer $(cat /var/run/secrets/kubernetes.io/serviceaccount/token)" --cacert /var/run/secrets/kubernetes.io/serviceaccount/ca.crt https://kubernetes/api/v1/namespaces/default/pods; sleep 5; done']
```

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

Check the Pod logs.

```
kubectl logs sa-pod
```

At this point, the Pod log should display an error message indicating that the ServiceAccount does not have appropriate permissions to perform the requested API call.