A CLOUD GURU

Working with kubectl

Will Boyd
DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Working with kubectl

What Is kubectl?

`kubectl get`

`kubectl describe`

`kubectl create`

`kubectl apply`

`kubectl delete`

`kubectl exec`

Hands-On Demonstration

Will Boyd

DEVOPS TRAINING ARCHITECT

A CLOUD GURU

What Is kubectl?

kubectl is a command line tool that allows you to interact with Kubernetes. kubectl uses the Kubernetes API to communicate with the cluster and carry out your commands.

“You can use kubectl to deploy applications, inspect and manage cluster resources, and view logs.”

A CLOUD GURU

kubectl get

Use `kubectl get` to list objects in the Kubernetes cluster.

- `-o` — Set output format.
- `--sort-by` — Sort output using a JSONPath expression.
- `--selector` — Filter results by label.

```
$ kubectl get <object type> <object name> -o
<output> --sort-by <JSONPath> --selector
<selector>
```

A CLOUD GURU

kubectl describe

You can get detailed information about Kubernetes objects using `kubectl describe`.

```
$ kubectl describe <object type> <object name>
```

A CLOUD GURU

kubectl create

Use `kubectl create` to create objects.

Supply a YAML file with `-f` to create an object from a YAML descriptor stored in the file.

If you attempt to create an object that already exists, an error will occur.

```
$ kubectl create -f <file name>
```

A CLOUD GURU

kubectl apply

`kubectl apply` is similar to `kubectl create`. However, if you use `kubectl apply` on an object that already exists, it will modify the existing object, if possible.

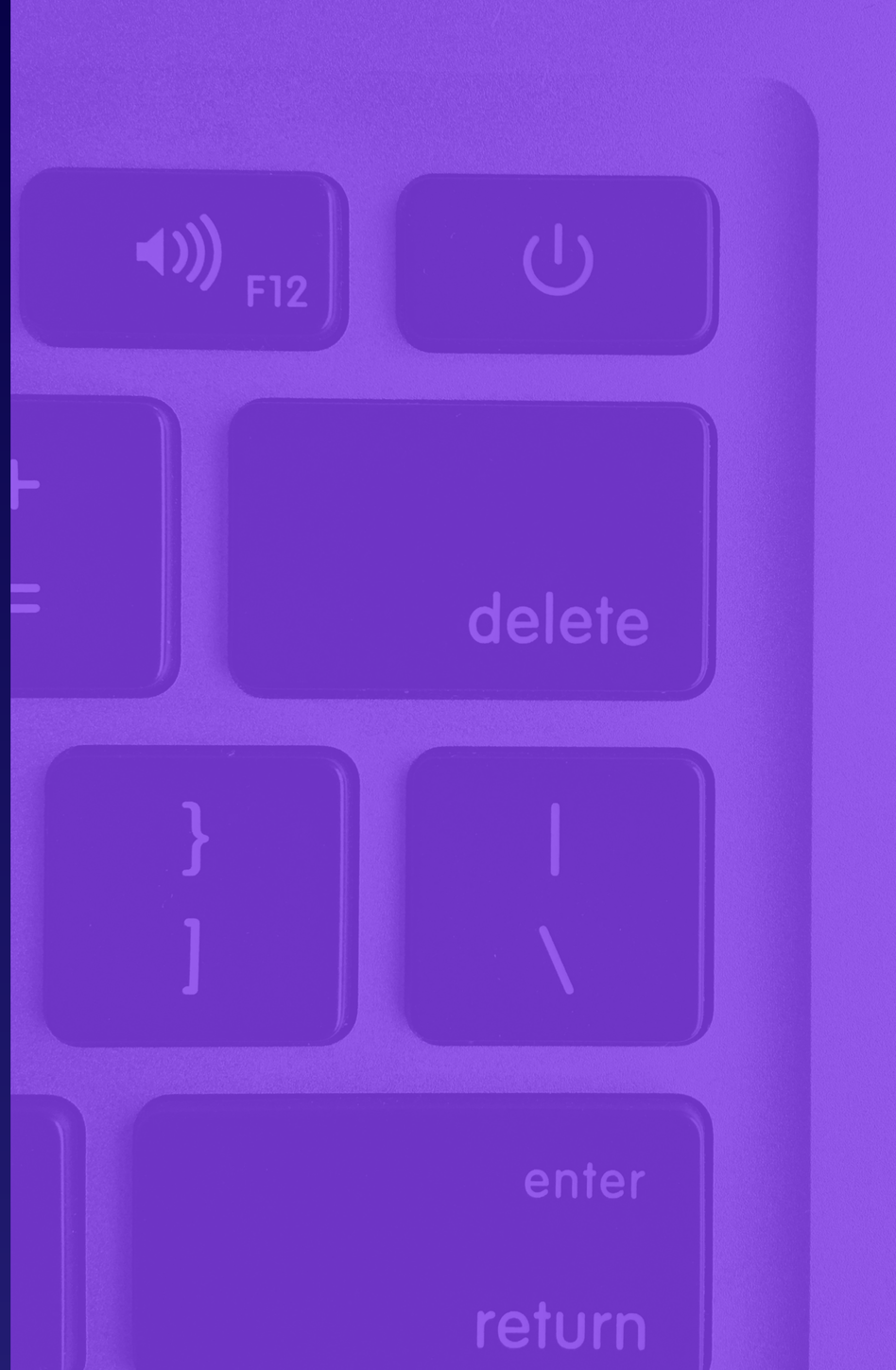
```
$ kubectl apply -f <file name>
```

A CLOUD GURU

kubectl delete

Use `kubectl delete` to delete objects from the cluster.

```
$ kubectl delete <object type> <object name>
```



kubectl exec

`kubectl exec` can be used to run commands inside containers. Keep in mind that, in order for a command to succeed, the necessary software must exist within the container to run it.

For pods with multiple containers, specify the container name with `-c`.

```
$ kubectl exec <pod name> -c <container name> --
<command>
```

A CLOUD GURU

Hands-On Demonstration

LESSON BREAKDOWN

Working with kubectl

What Is kubectl?

`kubectl get`

`kubectl describe`

`kubectl create`

`kubectl apply`

`kubectl delete`

`kubectl exec`

Hands-On Demonstration

Will Boyd

DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

kubectl Tips

Will Boyd
DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

kubectl Tips

Imperative Commands

Quick Sample YAML

Export YAML from an Object

Use the Docs

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

Imperative Commands

A CLOUD GURU

Declarative

- Define objects using data structures such as YAML or JSON.

VS

Imperative

- Define objects using kubectl commands and flags. Some people find imperative commands faster. Experiment and see what works for you!

```
$ kubectl create deployment my-deployment --
image=nginx
```

A CLOUD GURU

Quick Sample YAML

Use the **`--dry-run`** flag to run an imperative command without creating an object. Combine it with **`-o yaml`** to quickly obtain a sample YAML file you can manipulate!

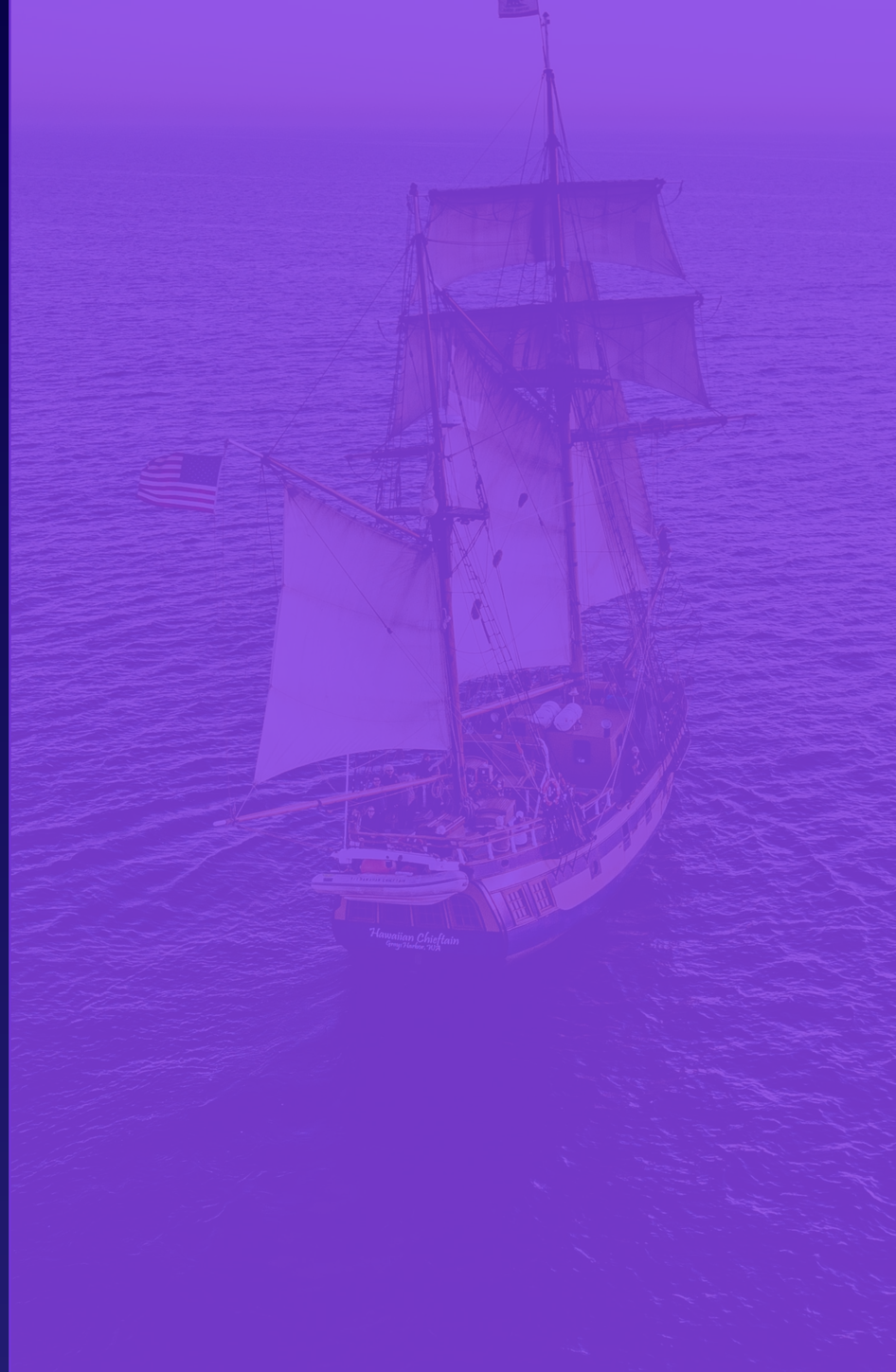
```
$ kubectl create deployment my-deployment
--image=nginx --dry-run -o yaml
```

A CLOUD GURU

Record a Command

Use the **`--record`** flag to record the command that was used to make a change.

```
$ kubectl scale deployment my-deployment
replicas=5 --record
```



Use the Docs

You can often find YAML examples in the Kubernetes documentation. You are allowed to use this documentation during the exam. Feel free to copy and paste example YAML and/or commands from the docs!

A CLOUD GURU

Hands-On Demonstration

LESSON BREAKDOWN

kubectl Tips

Imperative Commands

Quick Sample YAML

Export YAML from an Object

Use the Docs

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Managing K8s Role-Based Access Control (RBAC)

Will Boyd
DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Managing K8s Role-Based Access Control (RBAC)

RBAC in K8s

RBAC Objects

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

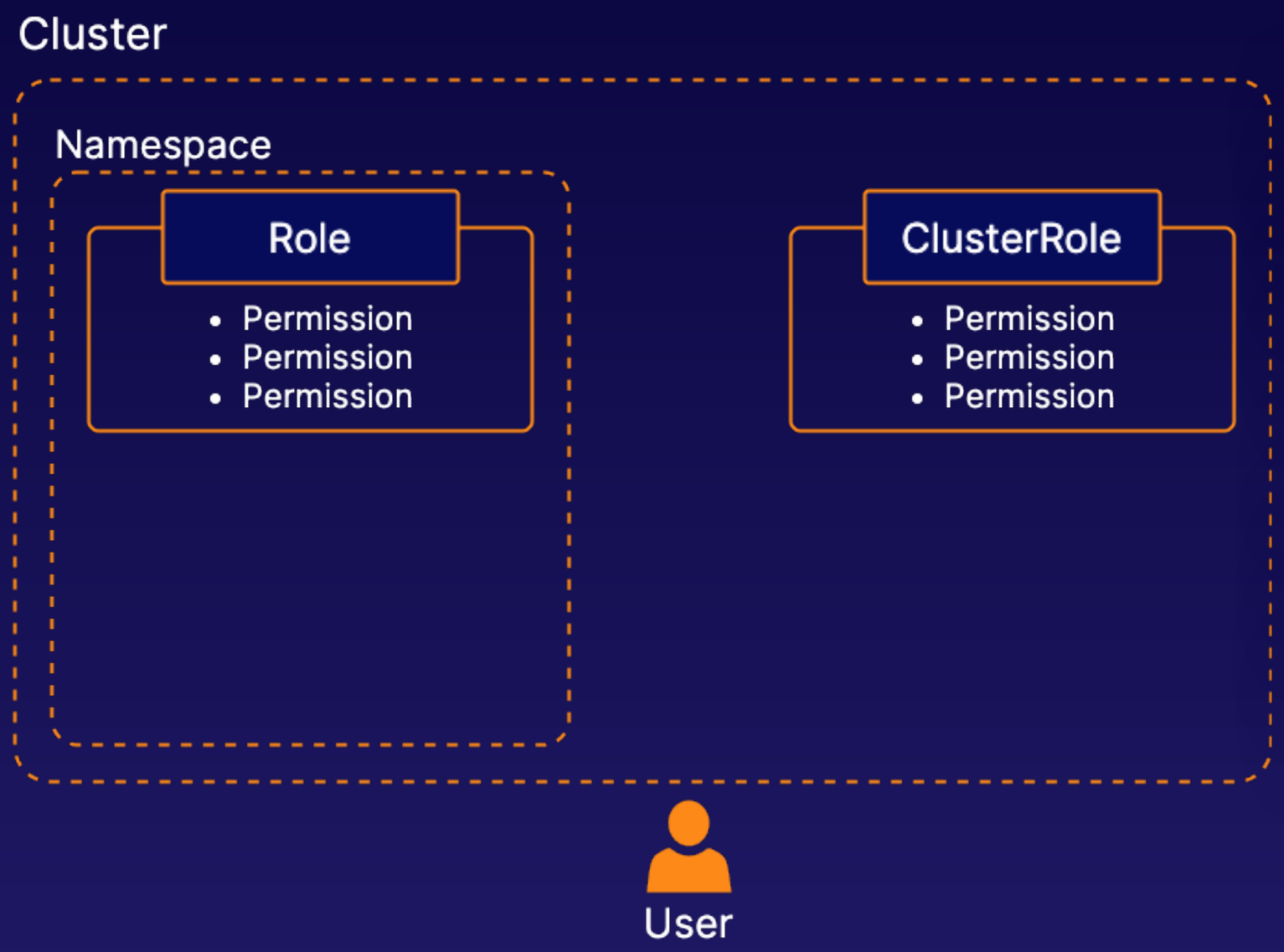
RBAC in K8s

Role-based access control (RBAC) in K8s allows you to control what users are allowed to do and access within your cluster.

For example, you can use RBAC to allow developers to read metadata and logs from Kubernetes pods but not make changes to them.

RBAC Objects

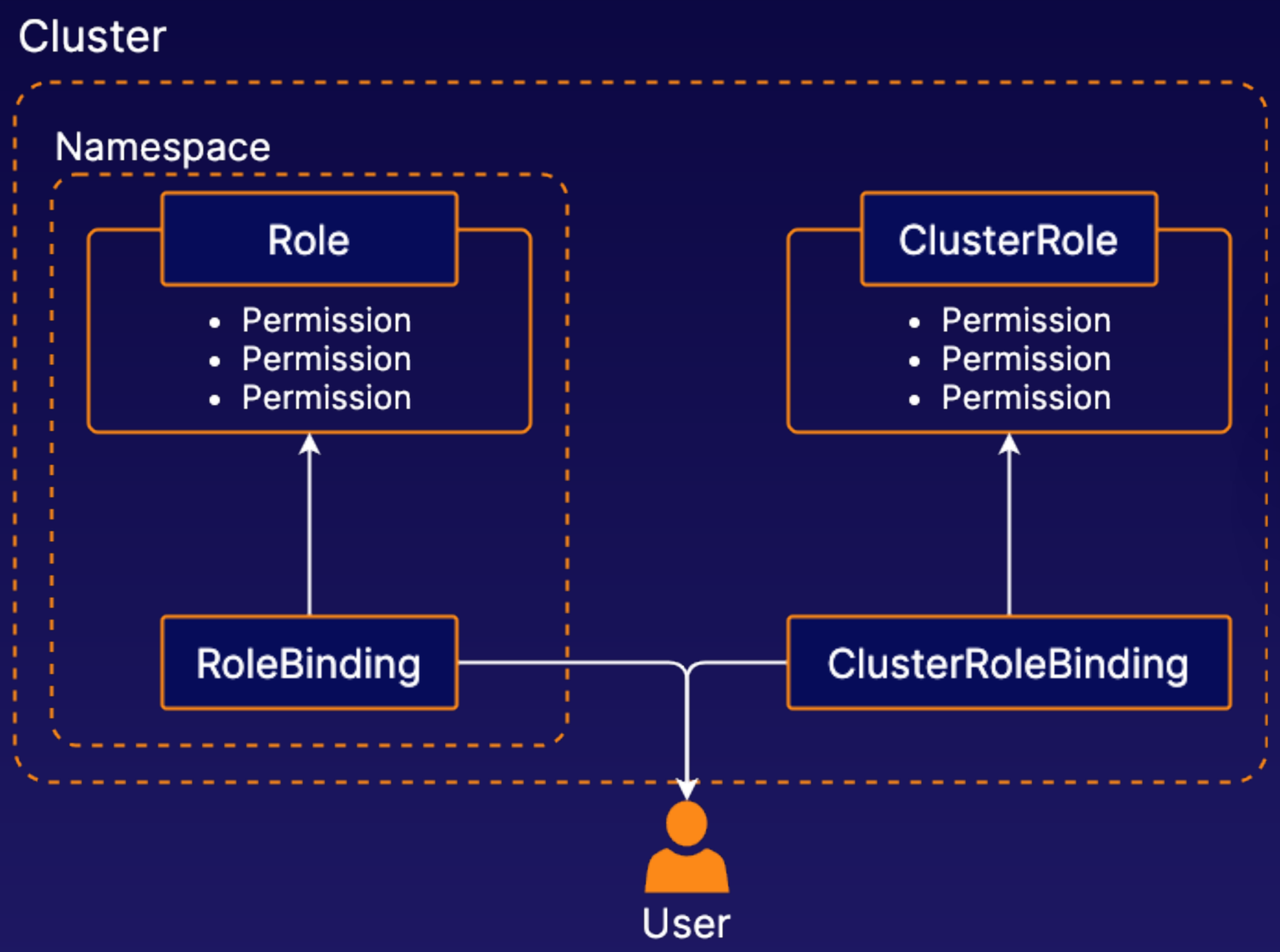
RBAC Objects



Roles and **ClusterRoles** are Kubernetes objects that define a set of permissions. These permissions determine what users can do in the cluster.

A Role defines permissions within a particular namespace, and a ClusterRole defines cluster-wide permissions not specific to a single namespace.

RBAC Objects



RoleBinding and **ClusterRoleBinding** are objects that connect Roles and ClusterRoles to users.

A CLOUD GURU

Hands-On Demonstration

LESSON BREAKDOWN

Managing K8s Role-Based Access Control (RBAC)

RBAC in K8s

RBAC Objects

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Creating Service Accounts

Will Boyd
DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Creating Service Accounts

What Is a Service Account?

Creating Service Accounts

Binding Roles to Service Accounts

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

What Is a Service Account?

In K8s, a **service account** is an account used by container processes within Pods to authenticate with the K8s API.

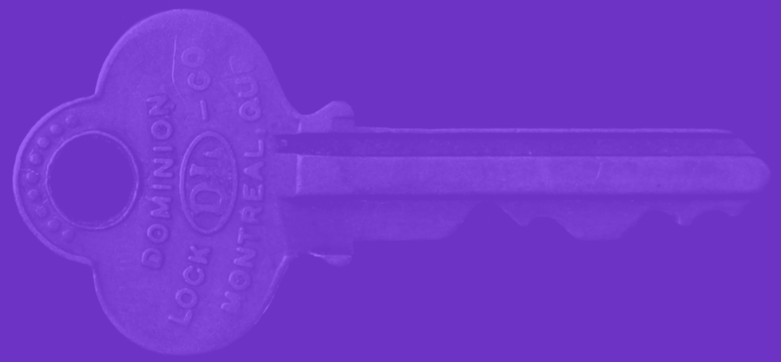
If your Pods need to communicate with the K8s API, you can use service accounts to control their access.

A CLOUD GURU

Creating Service Accounts

A service account object can be created with some YAML just like any other K8s object.

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: my-serviceaccount
```



Binding Roles to Service Accounts

You can manage access control for service accounts, just like any other user, using **RBAC** objects.

Bind service accounts with ClusterRoles or ClusterRoleBindings to provide access to K8s API functionality.

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: sa-pod-reader
subjects:
- kind: ServiceAccount
  name: my-serviceaccount
  namespace: default
roleRef:
  kind: Role
  name: pod-reader
  apiGroup: rbac.authorization.k8s.io
```

A CLOUD GURU

Hands-On Demonstration

LESSON BREAKDOWN

Creating Service Accounts

What Is a Service Account?

Creating Service Accounts

Binding Roles to Service Accounts

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

A CLOUD GURU

Inspecting Pod Resource Usage

Will Boyd
DEVOPS TRAINING ARCHITECT

LESSON BREAKDOWN

Inspecting Pod Resource Usage

Kubernetes Metrics Server

`kubectl top`

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU

Kubernetes Metrics Server

In order to view metrics about the resources pods and containers are using, we need an add-on to collect and provide that data. One such add-on is **Kubernetes Metrics Server**.

A CLOUD GURU

kubectl top

With `kubectl top`, you can view data about resource usage in your pods and nodes. `kubectl top` also supports flags like `--sort-by` and `--selector`.

```
$ kubectl top pod --sort-by <JSONPATH> --
selector <selector>
```

A CLOUD GURU

Hands-On Demonstration

LESSON BREAKDOWN

Inspecting Pod Resource Usage

Kubernetes Metrics Server

`kubectl top`

Hands-On Demonstration

Will Boyd
DEVOPS TRAINING ARCHITECT

A CLOUD GURU