Working with Your Kubernetes Cluster

Kien Bui
DevOps & Platform Engineer

Course Overview

Introduction

Exploring Kubernetes Architecture

Installing and Configuring Kubernetes

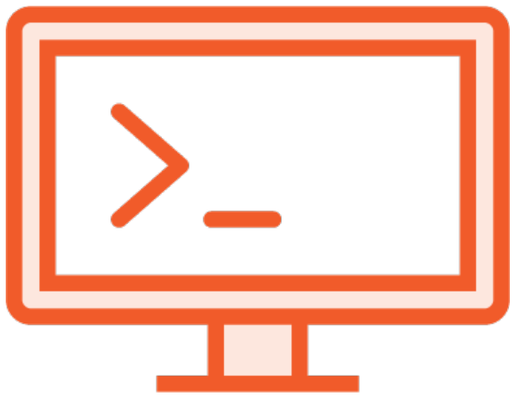
Working with Your Kubernetes Cluster

Overview

Using `kubectl` to Interact With Your Cluster

Application Deployments

Using kubectl

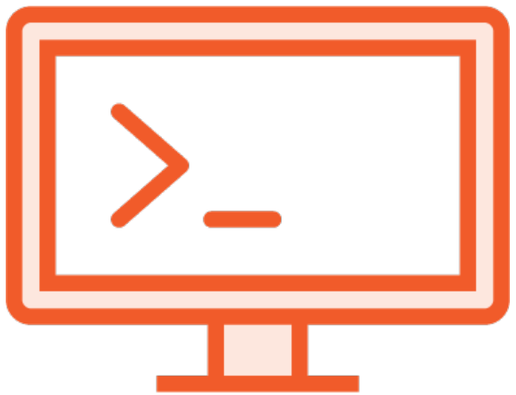


Primary CLI tool

Control your Kubernetes Cluster

- Operations - what you want to do
- Resources - what you want to do it to
- Output - if there’s output, its format

Operations - what do you want to do?



`apply/create` - create resource(s)

`run` - start a pod from an image

`explain` - documentation of resources

`delete` - delete resource(s)

`get` - list resources

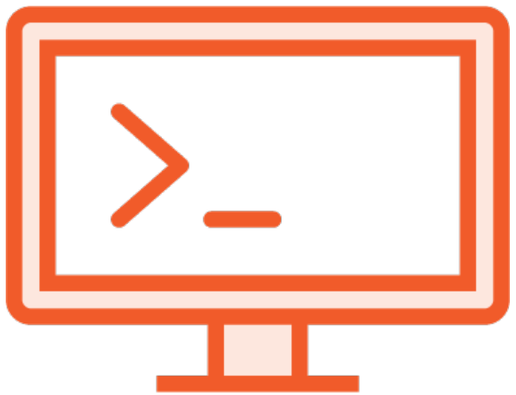
`describe` - detailed resource information

`exec` - execute a command on a container

`logs` - view logs on a container

https://kubernetes.io/docs/reference/kubectl/overview/#operations

Resources - what do you want to do it to?



`nodes (no)`

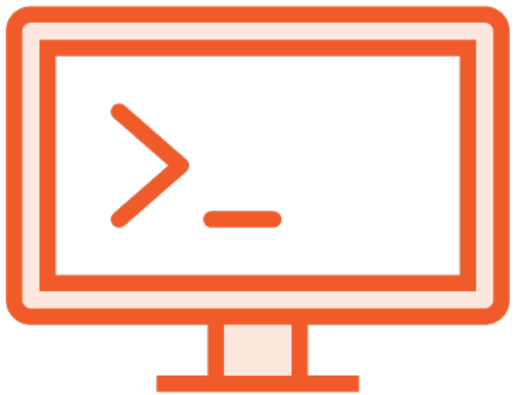
`pods (po)`

`services (svc)`

..and many more

https://kubernetes.io/docs/reference/kubectl/overview/#resource-types

Output



Specify `kubectl's` output format

`wide` - output additional info

`yaml` - YAML formatted API object

`json` - JSON formatted API object

`dry-run` - print an object without sending it to the API Server

https://kubernetes.io/docs/reference/kubectl/overview/#output-options

kubectl

kubectl	[command]	[type]	[name]	[flags]
kubectl	get	pods	pod1	--output=yaml
kubectl	create	deployment	nginx	--image=nginx

https://kubernetes.io/docs/reference/kubectl/kubectl/

https://kubernetes.io/docs/reference/kubectl/cheatsheet/

Demo

Using kubectl

- Nodes
- Pods
- API Resources
- Configure bash auto-completion

Application Deployment in Kubernetes

Imperative

```
kubectl create deployment nginx \
--image=nginx
```

```
kubectl run nginx --image=nginx
```

Declarative

Define our desired state in code

Manifest

YAML or JSON

```
kubectl apply -f deployment.yaml
```

Basic Manifest - Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello-world
 template:
    metadata:
     labels:
        app: hello-world
    spec:
      containers:
      - image: gcr.io/google-samples/hello-app:1.0
        name: hello-app
```

```
kubectl apply -f deployment.yaml
```

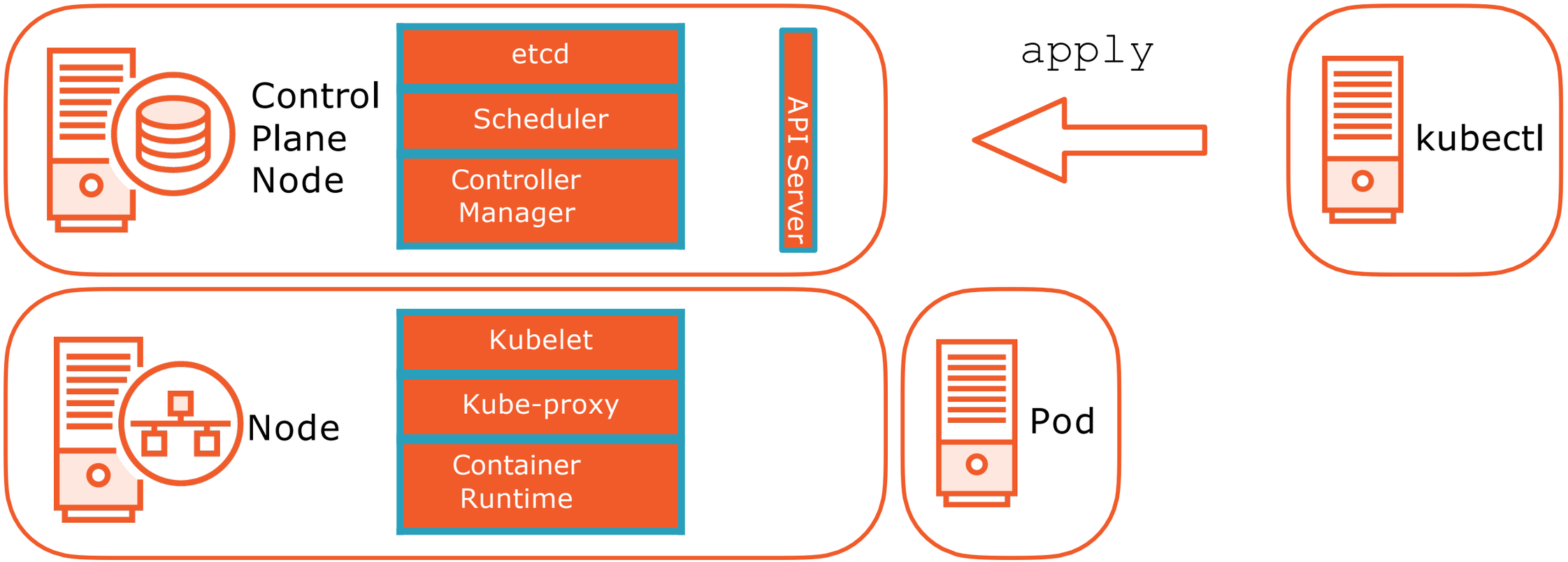
```
kubectl create deployment hello-world \
      --image=gcr.io/google-samples/hello-app:1.0 \
      --dry-run=client -o yaml > deployment.yaml


 kubectl apply -f deployment.yaml
```

Generating Manifests with dry-run

Deployments

Application Deployment Process



Demo

Imperatively and Declaratively

Deploying resources in your Cluster

- Deployments
- Pods
- Services

Making changes to existing resources

Summary

Using `kubectl` to Interact With Your Cluster

Application Deployments