**Deploying Your First Application in Kubernetes - Study Notes (Day 33)**

**Course: Complete DevOps Course**  
**Instructor: Abhishek**

**1. Introduction**

**Prerequisites**

✅ Watch **Day 30-32** (Kubernetes vs. Docker, Architecture, Installation)  
✅ Understand:

* Kubernetes **cluster architecture**
* Why Kubernetes > Docker (Auto-healing, Scaling, Enterprise support)

**2. Key Concepts**

**1. Pods: The Smallest Unit in Kubernetes**

* **Definition:** A **Pod** is a wrapper for **one or more containers** with shared resources (networking, storage).
* **Why Pods?**
  + Standardizes container deployment via **YAML files** (vs. Docker CLI commands).
  + Enables **multi-container patterns** (e.g., sidecar, init containers).

**Pod vs. Docker Container**

| **Feature** | **Docker Container** | **Kubernetes Pod** |
| --- | --- | --- |
| **Deployment** | docker run -d -p 80:80 nginx | Defined in pod.yaml (declarative) |
| **Networking** | Isolated | Shared IP for containers in the same Pod |
| **Use Case** | Single-container apps | Multi-container apps (e.g., logging sidecars) |

**2. kubectl: Kubernetes CLI**

* **Purpose:** Interact with Kubernetes clusters (like docker CLI for Docker).
* **Common Commands:**
* kubectl get pods # List Pods
* kubectl describe pod nginx # Debug Pod
* kubectl logs nginx # View Pod logs
* kubectl delete pod nginx # Delete Pod

**3. Hands-On: Deploying a Pod**

**Step 1: Install Tools**

1. **Install kubectl:**
2. curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
3. chmod +x kubectl && sudo mv kubectl /usr/local/bin/
4. **Install Minikube (Local Cluster):**
5. curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
6. sudo install minikube-linux-amd64 /usr/local/bin/minikube

**Step 2: Start Minikube Cluster**

minikube start --driver=docker # Use Docker as VM driver

kubectl get nodes # Verify cluster

**Step 3: Create a Pod**

1. **pod.yaml:**
2. apiVersion: v1
3. kind: Pod
4. metadata:
5. name: nginx
6. spec:
7. containers:
8. - name: nginx
9. image: nginx:1.14.2
10. ports:
11. - containerPort: 80
12. **Deploy:**
13. kubectl apply -f pod.yaml
14. kubectl get pods -o wide # Check Pod IP

**Step 4: Access the Pod**

minikube ssh # SSH into cluster

curl http://<POD\_IP> # Access Nginx

exit

**Step 5: Debugging**

kubectl describe pod nginx # Inspect Pod details

kubectl logs nginx # View application logs

**4. Next Steps: Deployments**

* **Problem:** Pods lack **auto-healing/scaling**.
* **Solution:** Use **Deployments** (wrapper over Pods).
* apiVersion: apps/v1
* kind: Deployment
* metadata:
* name: nginx-deployment
* spec:
* replicas: 3
* template:
* spec:
* containers:
* - name: nginx
* image: nginx:1.14.2

**5. Cheat Sheet**

| **Task** | **Command** |
| --- | --- |
| List Pods | kubectl get pods |
| Delete Pod | kubectl delete pod <name> |
| Debug Pod | kubectl describe pod <name> |
| View Logs | kubectl logs <pod-name> |

📌 **Pro Tip:** Bookmark the [Kubectl Cheat Sheet](https://kubernetes.io/docs/reference/kubectl/cheatsheet/).

**6. Summary**

* **Pods** = Kubernetes’ smallest deployable unit (1+ containers).
* **kubectl** = CLI to manage Kubernetes (like docker for Docker).
* **Minikube** = Local Kubernetes cluster for practice.
* **Next:** Deployments for **auto-scaling/healing**.

🚀 **Assignment:**

1. Deploy a Pod with a custom app (e.g., Python/Go).
2. Debug it using kubectl describe/logs.

📢 **Feedback?** Comment below! 👍 **Like & Share** if this helped!

**End of Notes** 🎉

# \*\*Deploying Your First Application in Kubernetes - Study Notes (Day 33)\*\*

\*\*Course: Complete DevOps Course\*\*

\*\*Instructor: Abhishek\*\*

---

## \*\*1. Introduction\*\*

### \*\*Prerequisites\*\*

✅ Watch \*\*Day 30-32\*\* (Kubernetes vs. Docker, Architecture, Installation)

✅ Understand:

- Kubernetes \*\*cluster architecture\*\*

- Why Kubernetes > Docker (Auto-healing, Scaling, Enterprise support)

---

## \*\*2. Key Concepts\*\*

### \*\*1. Pods: The Smallest Unit in Kubernetes\*\*

- \*\*Definition:\*\* A \*\*Pod\*\* is a wrapper for \*\*one or more containers\*\* with shared resources (networking, storage).

- \*\*Why Pods?\*\*

- Standardizes container deployment via \*\*YAML files\*\* (vs. Docker CLI commands).

- Enables \*\*multi-container patterns\*\* (e.g., sidecar, init containers).

#### \*\*Pod vs. Docker Container\*\*

| \*\*Feature\*\* | \*\*Docker Container\*\* | \*\*Kubernetes Pod\*\* |

|-------------------|------------------------------------|----------------------------------------|

| \*\*Deployment\*\* | `docker run -d -p 80:80 nginx` | Defined in `pod.yaml` (declarative) |

| \*\*Networking\*\* | Isolated | Shared IP for containers in the same Pod |

| \*\*Use Case\*\* | Single-container apps | Multi-container apps (e.g., logging sidecars) |

### \*\*2. `kubectl`: Kubernetes CLI\*\*

- \*\*Purpose:\*\* Interact with Kubernetes clusters (like `docker` CLI for Docker).

- \*\*Common Commands:\*\*

```sh

kubectl get pods # List Pods

kubectl describe pod nginx # Debug Pod

kubectl logs nginx # View Pod logs

kubectl delete pod nginx # Delete Pod

```

---

## \*\*3. Hands-On: Deploying a Pod\*\*

### \*\*Step 1: Install Tools\*\*

1. \*\*Install `kubectl`:\*\*

```sh

curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

chmod +x kubectl && sudo mv kubectl /usr/local/bin/

```

2. \*\*Install Minikube (Local Cluster):\*\*

```sh

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube

```

### \*\*Step 2: Start Minikube Cluster\*\*

```sh

minikube start --driver=docker # Use Docker as VM driver

kubectl get nodes # Verify cluster

```

### \*\*Step 3: Create a Pod\*\*

1. \*\*`pod.yaml`:\*\*

```yaml

apiVersion: v1

kind: Pod

metadata:

name: nginx

spec:

containers:

- name: nginx

image: nginx:1.14.2

ports:

- containerPort: 80

```

2. \*\*Deploy:\*\*

```sh

kubectl apply -f pod.yaml

kubectl get pods -o wide # Check Pod IP

```

### \*\*Step 4: Access the Pod\*\*

```sh

minikube ssh # SSH into cluster

curl http://<POD\_IP> # Access Nginx

exit

```

### \*\*Step 5: Debugging\*\*

```sh

kubectl describe pod nginx # Inspect Pod details

kubectl logs nginx # View application logs

```

---

## \*\*4. Next Steps: Deployments\*\*

- \*\*Problem:\*\* Pods lack \*\*auto-healing/scaling\*\*.

- \*\*Solution:\*\* Use \*\*Deployments\*\* (wrapper over Pods).

```yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

replicas: 3

template:

spec:

containers:

- name: nginx

image: nginx:1.14.2

```

---

## \*\*5. Cheat Sheet\*\*

| \*\*Task\*\* | \*\*Command\*\* |

|------------------------|--------------------------------------|

| List Pods | `kubectl get pods` |

| Delete Pod | `kubectl delete pod <name>` |

| Debug Pod | `kubectl describe pod <name>` |

| View Logs | `kubectl logs <pod-name>` |

📌 \*\*Pro Tip:\*\* Bookmark the [Kubectl Cheat Sheet](https://kubernetes.io/docs/reference/kubectl/cheatsheet/).

---

## \*\*6. Summary\*\*

- \*\*Pods\*\* = Kubernetes’ smallest deployable unit (1+ containers).

- \*\*`kubectl`\*\* = CLI to manage Kubernetes (like `docker` for Docker).

- \*\*Minikube\*\* = Local Kubernetes cluster for practice.

- \*\*Next:\*\* Deployments for \*\*auto-scaling/healing\*\*.

🚀 \*\*Assignment:\*\*

1. Deploy a Pod with a custom app (e.g., Python/Go).

2. Debug it using `kubectl describe/logs`.

📢 \*\*Feedback?\*\* Comment below! 👍 \*\*Like & Share\*\* if this helped!

---

\*\*End of Notes\*\* 🎉