**POD**

In Virtualization, basic scheduling unit is Virtual Machine(VM)

In Containerization, basic scheduling unit is Containerization

In Kubernetes, what is basic scheduling unit?

**PODS**

What are pods on kubernetes?

What is difference between container and pods?

How to create PODS in kubernetes?

**Objective:**

**----------**

1. What is pod?

2. POD deployment (how pods are deployment in kubernetes)

3. Multi-container (how to use multiple container in single pod)

4. POD networking (How does pod networking works)

5. Inter & Intra pod communication (what is Inter-pod networking && what is Intra-pod networking)

6. POD Lifecycle (stages of pod lifecycle)

7. POD Manifest file (how to write pod manifest file)

# What is Pod?

-- Pod is known as Atomic unit of scheduling

{

In virtualization Environment, virtual machine is the Atomic unit of scheduling

In Containerization Environment, Dockeris the Atomic unit of scheduling

In kubernetes Environment, Pod is the Atomic unit of scheduling

}

Docker

Virtualization

VM

Container

POD

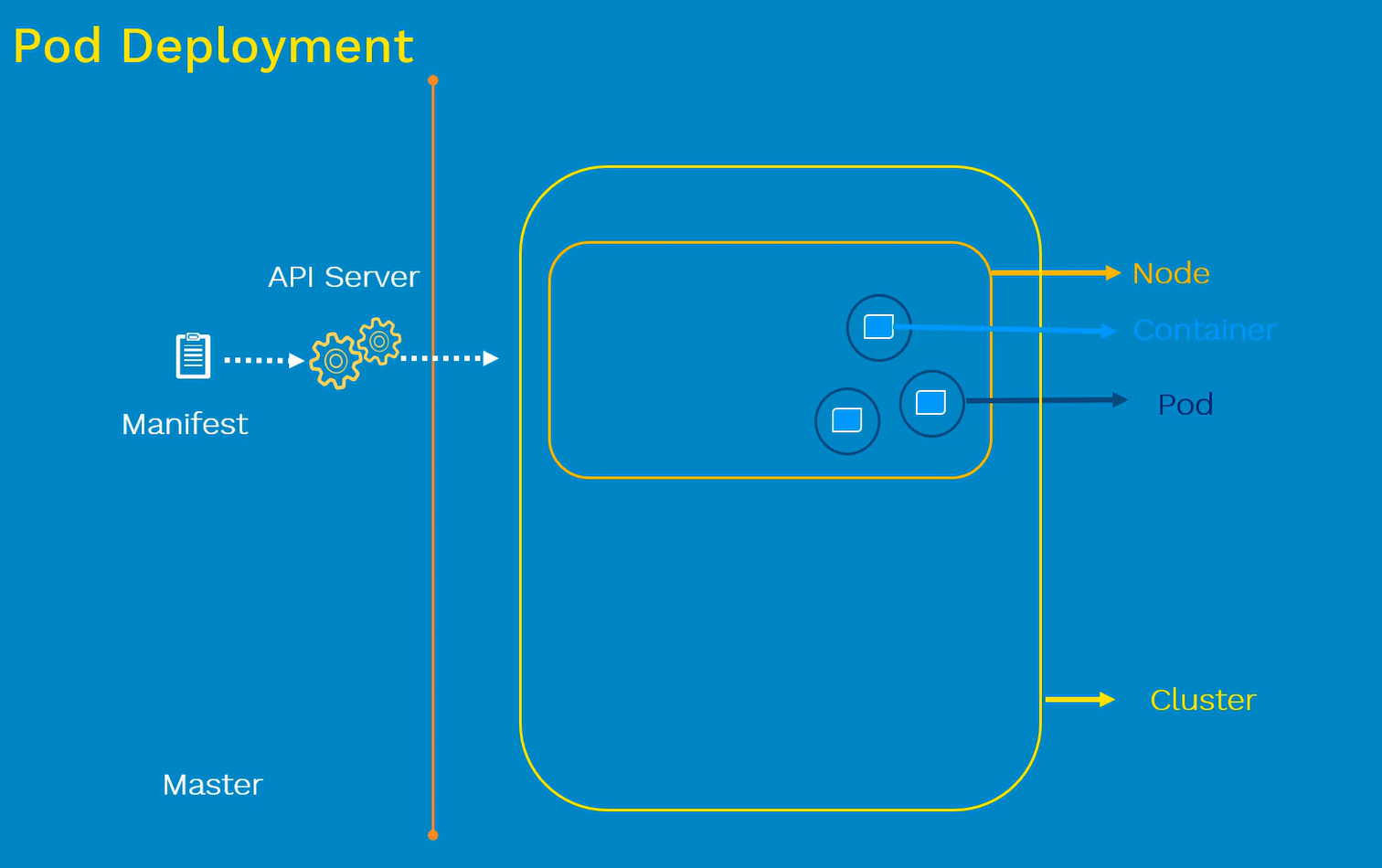
Kubernetes

**POD Deployment:**

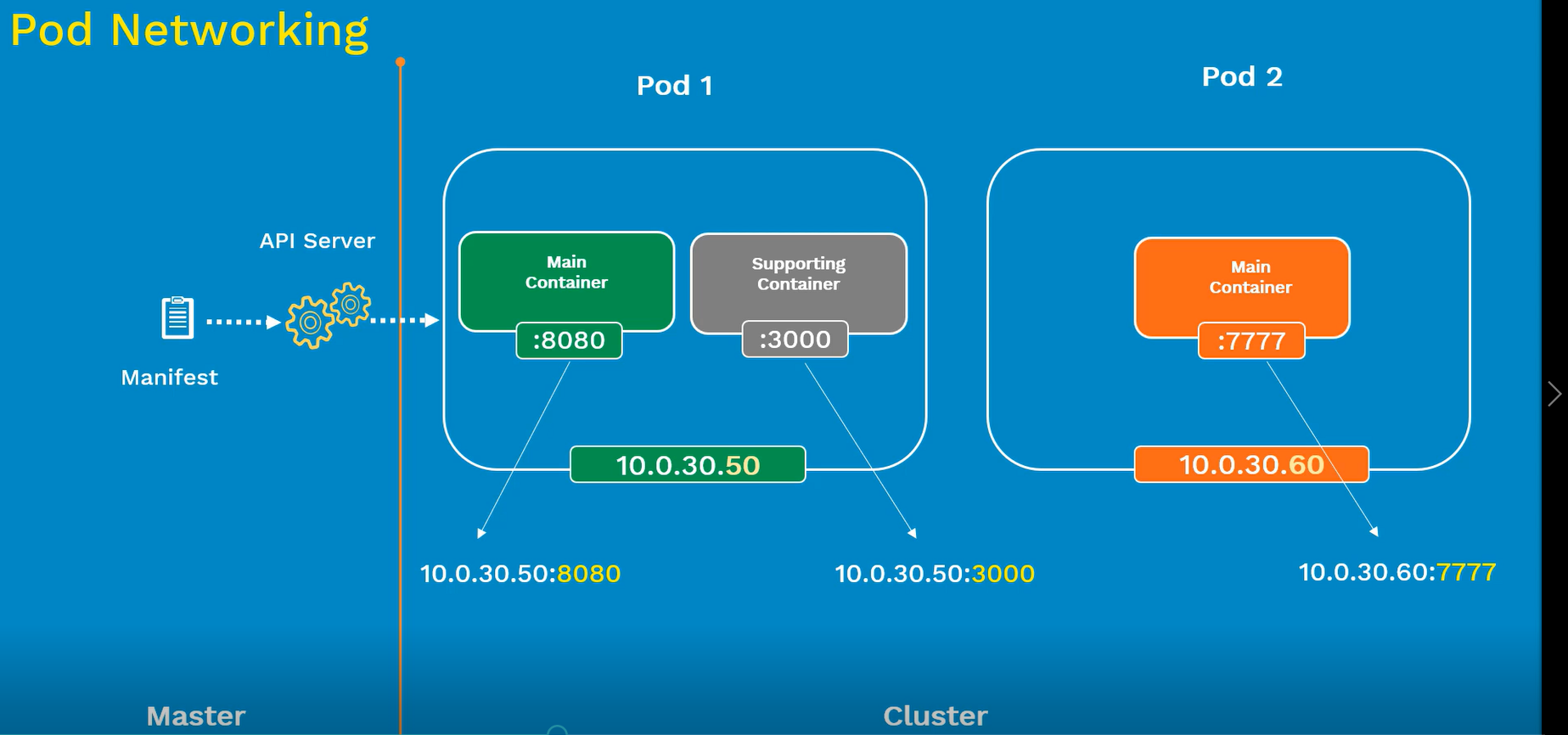
How the pod is deployed and scaled inside kubernetes?

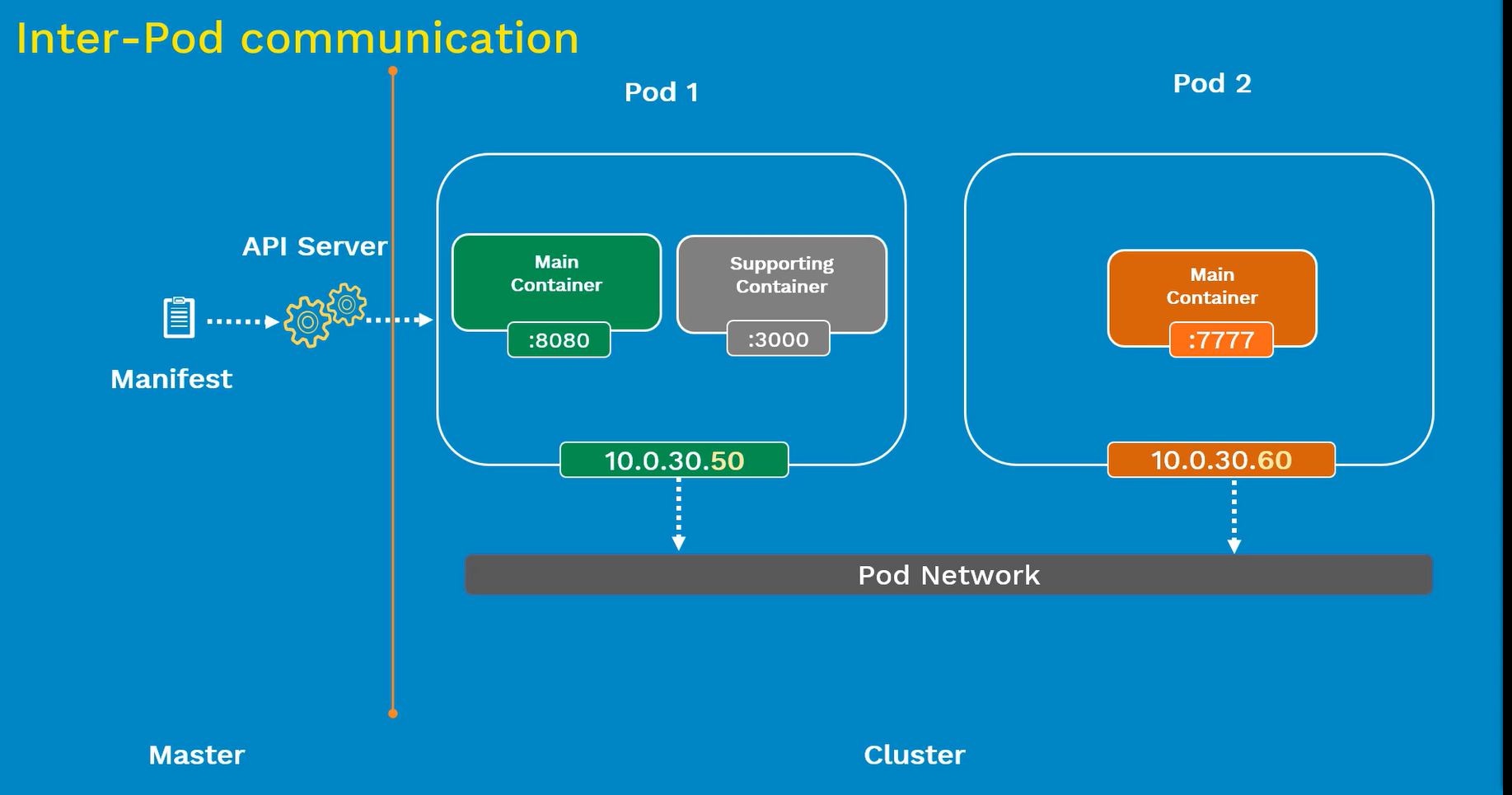
TO deploy the pod, 🡪 first we have to write pod manifest file that contain container images that we have to write and deploy to the API server on the **master node**

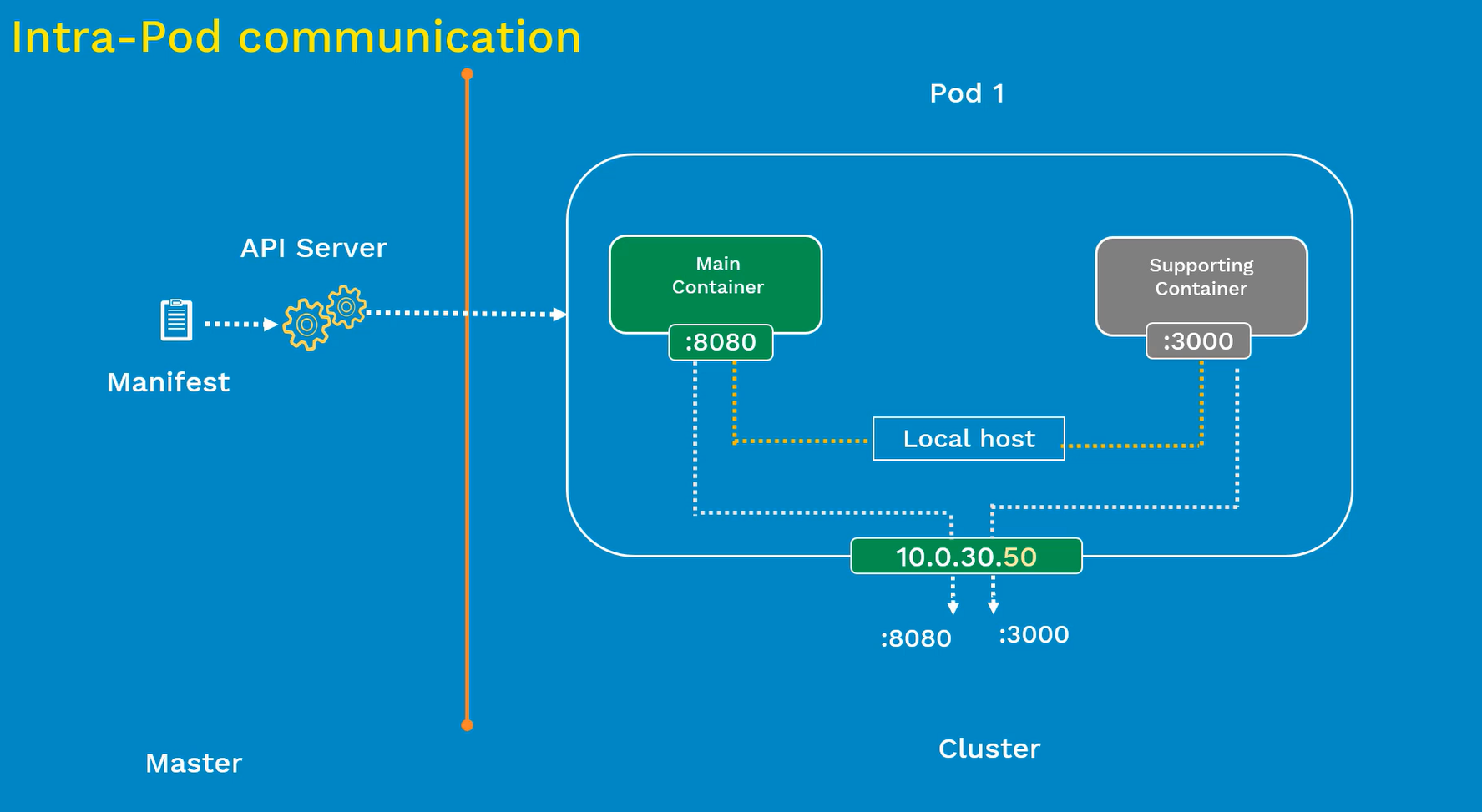
**API server and scheduler components** on the master node 🡪 decides and deploy the PODS on the **worker nodes** respectively

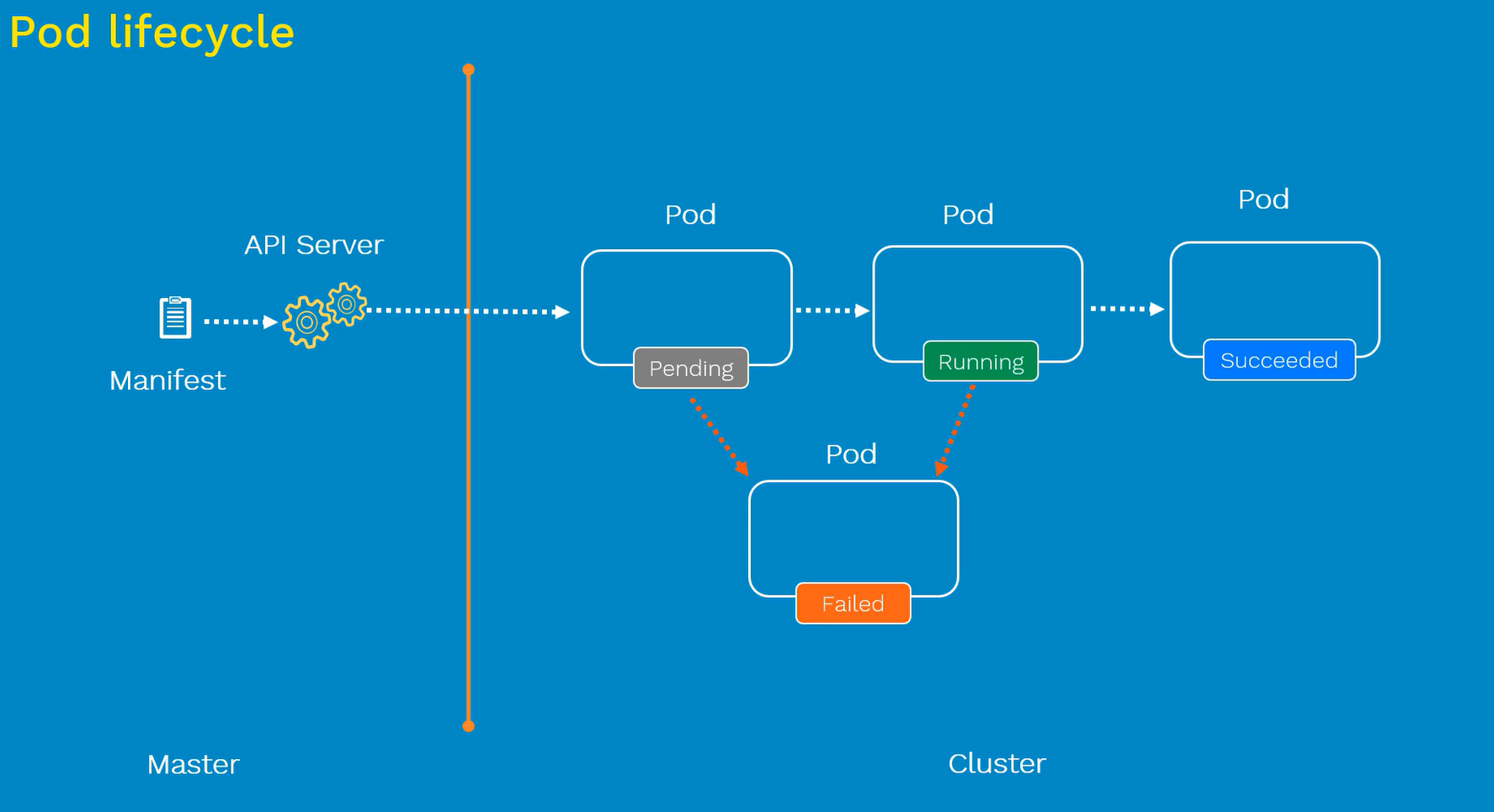


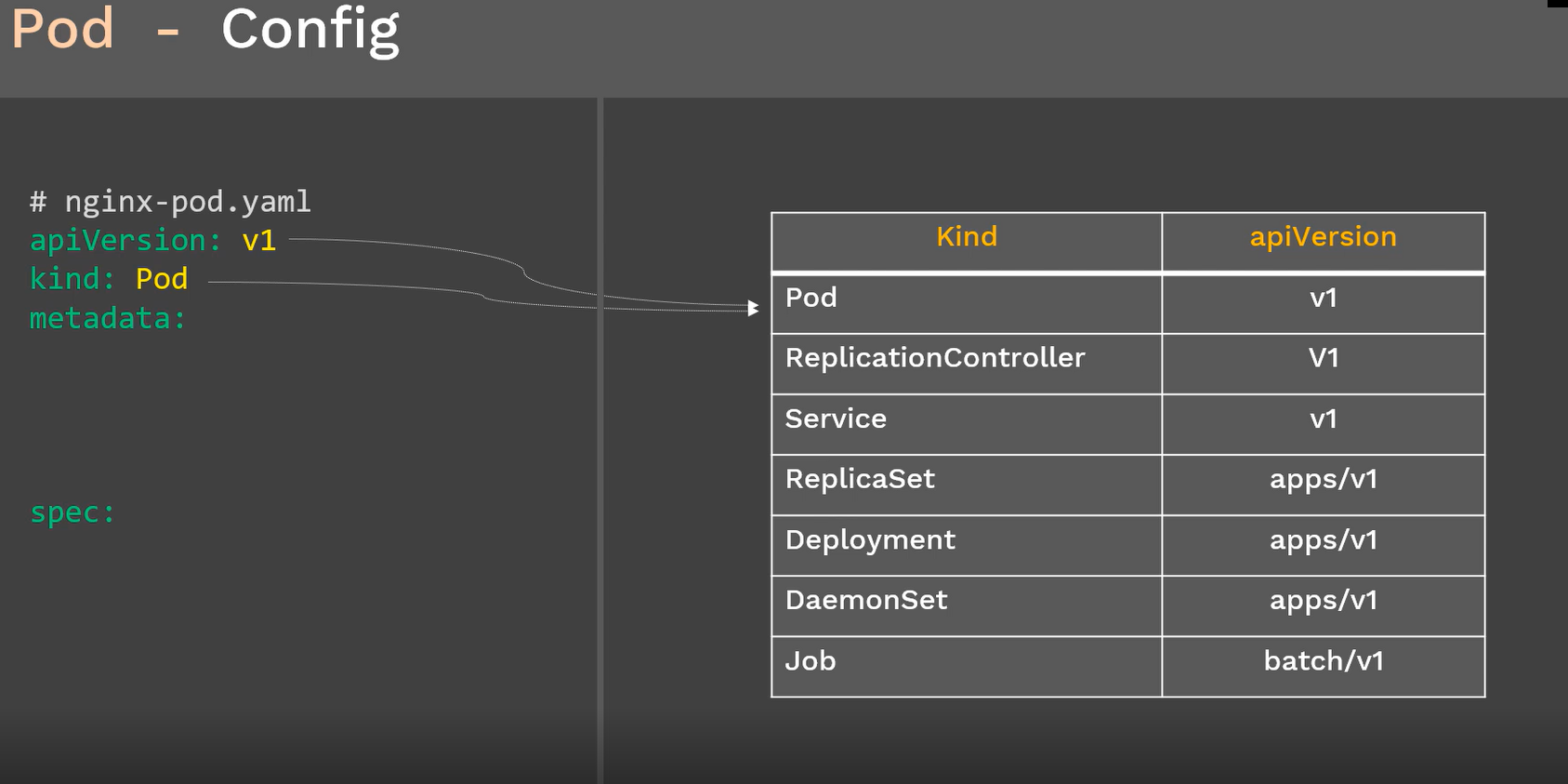
Kubernetes doesn’t deploy the containers directly on the worker nodes -🡪 containers are encapsulated inside the pods



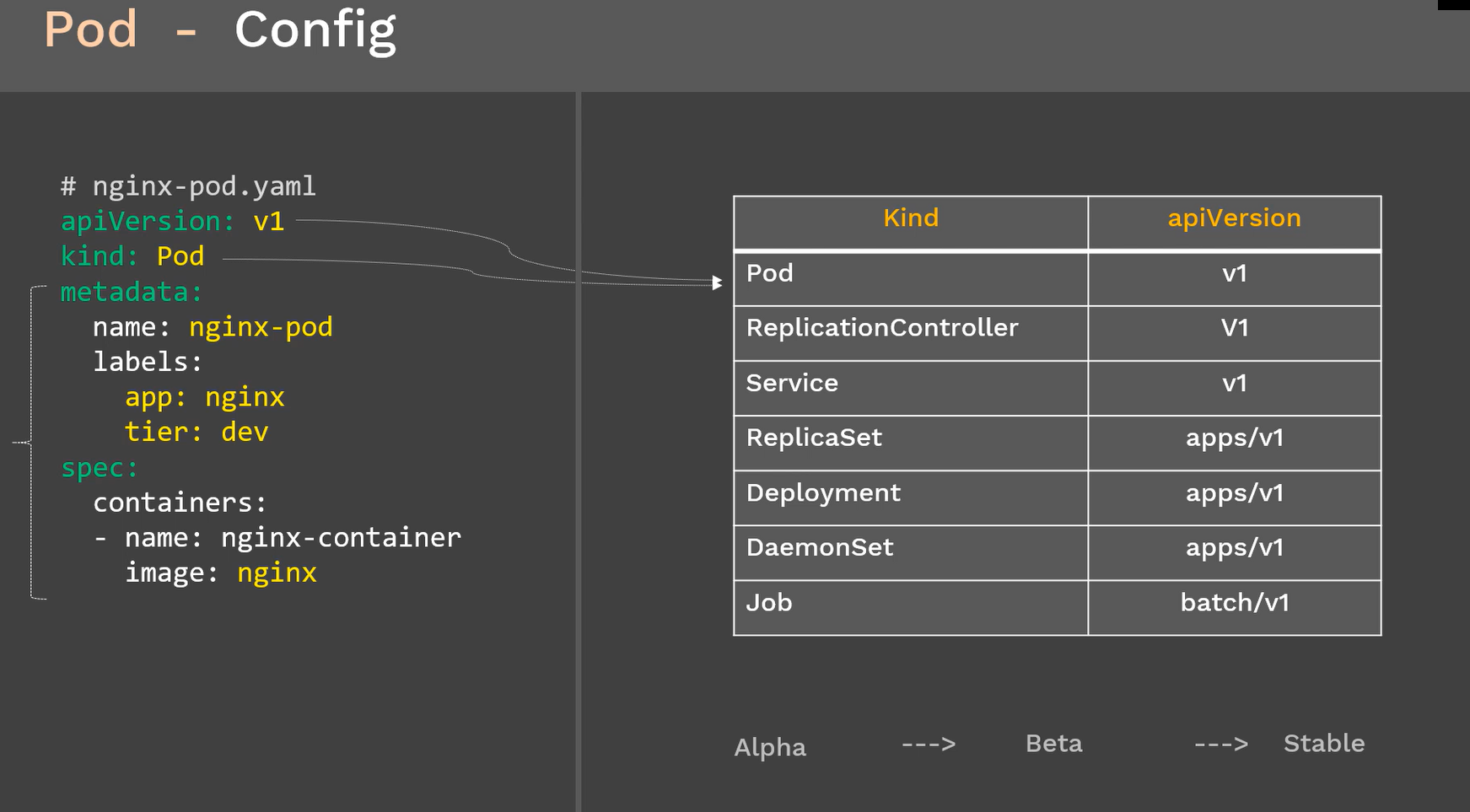


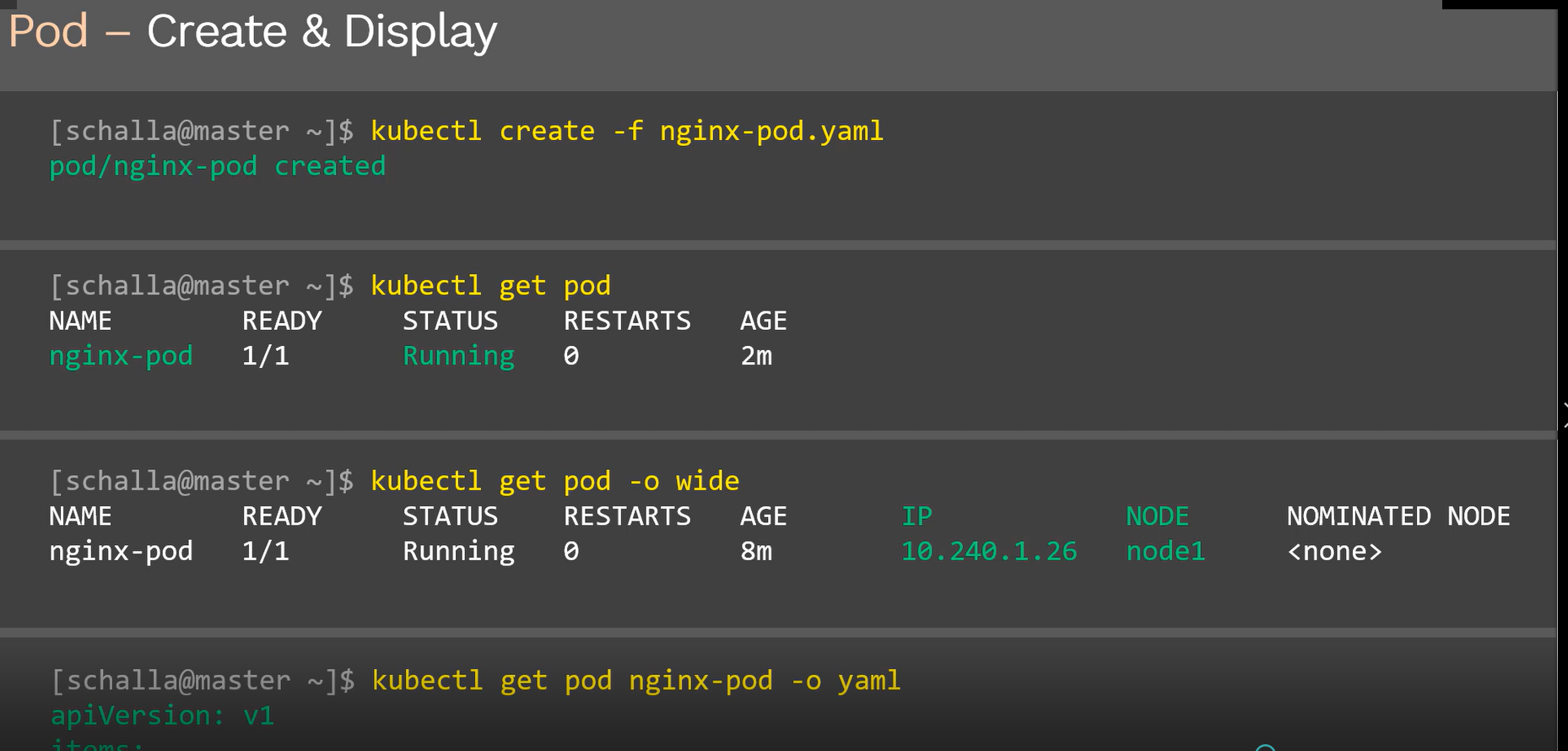


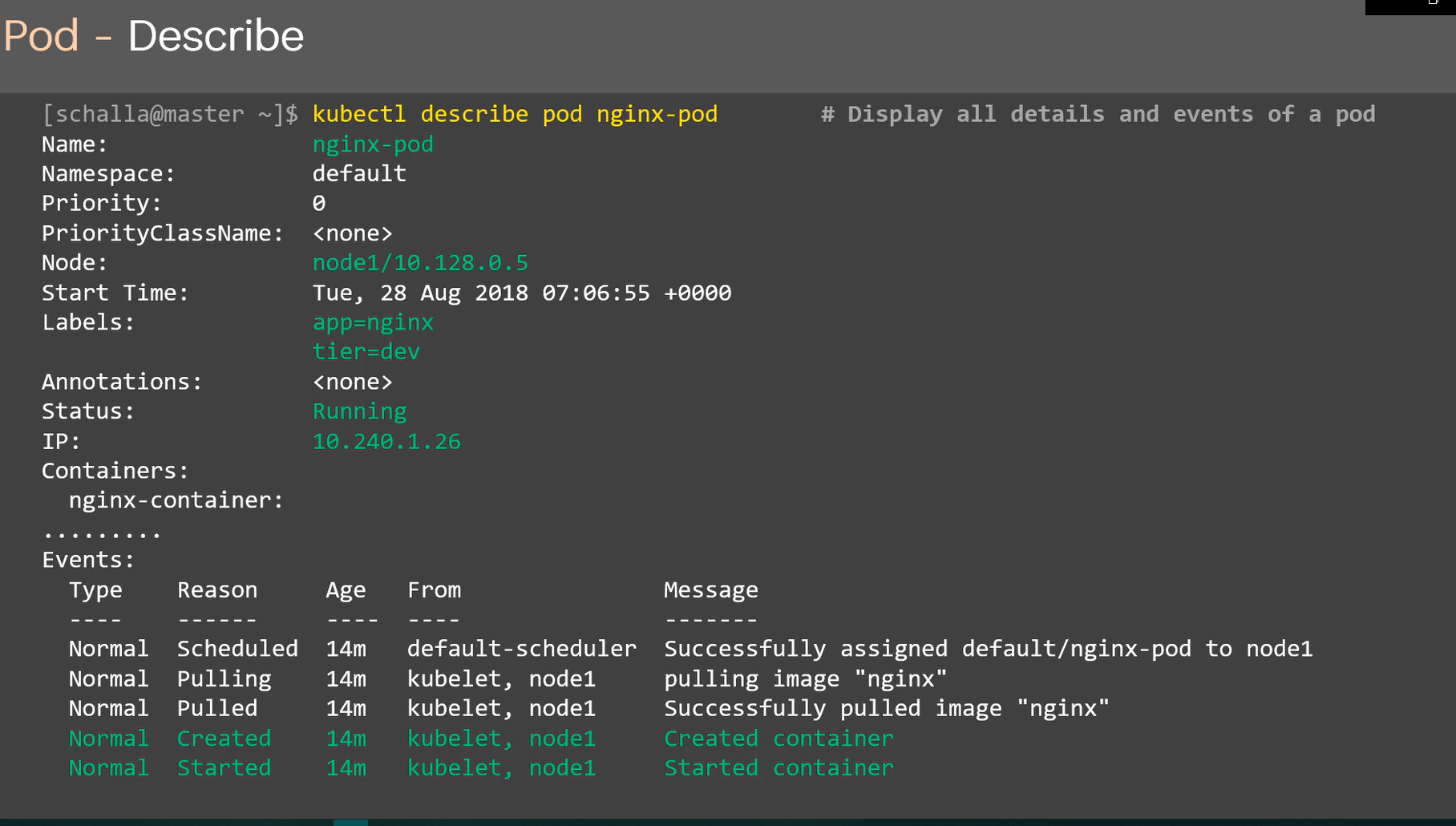


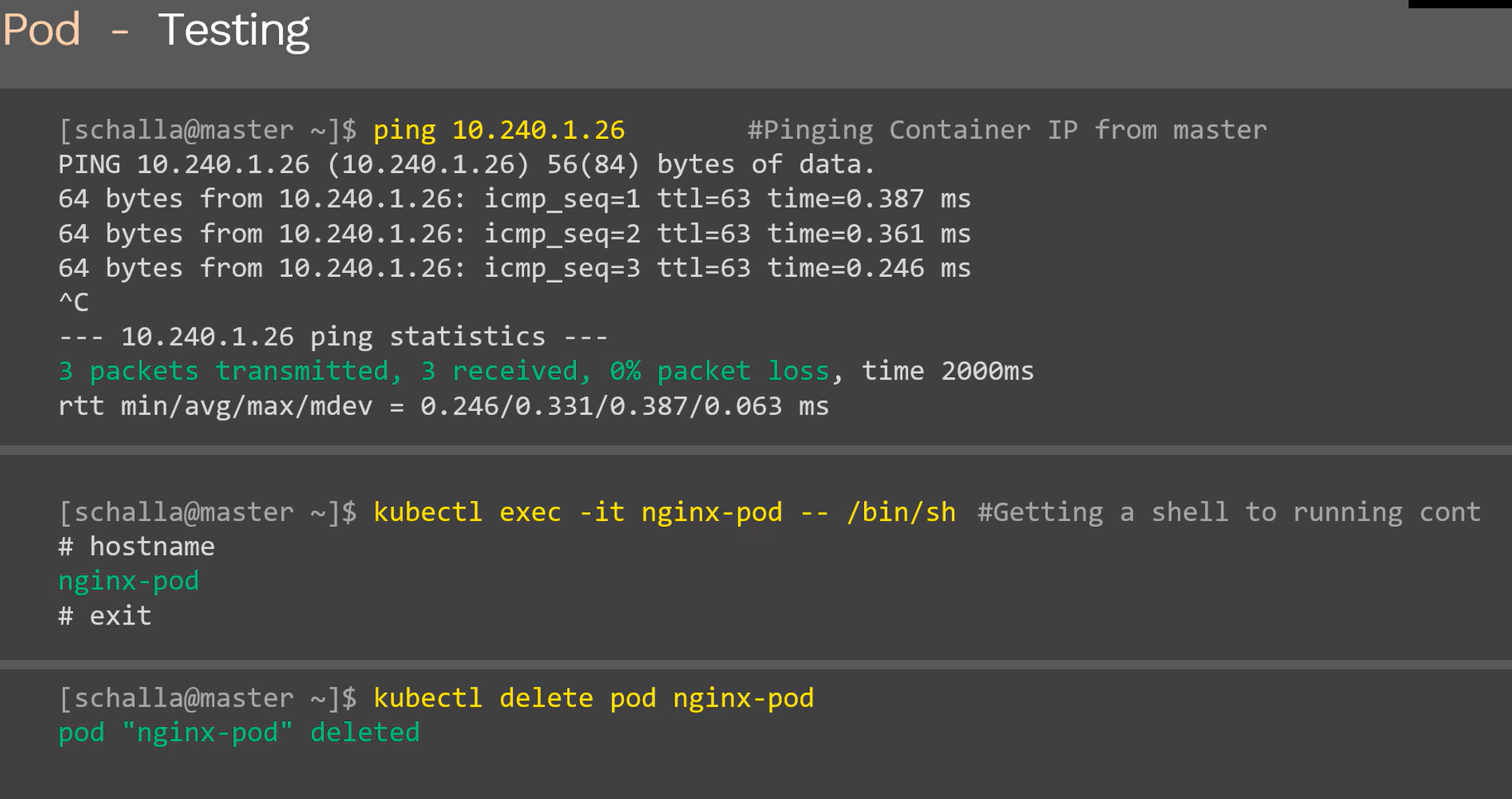


**Pod Manifest file:**









**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Demo:**

# 1. <https://labs.play-with-k8s.com/>

# nginx-pod.yaml

---

apiVersion: v1

kind: Pod

metadata:

labels:

app: nginx

tier: dev

name: nginx-Pod

spec:

containers:

-

image: nginx

name: nginx-container

**apiVersion** – It defines the version number which kubernetes object belongs to

**kind:** what is this kind of object we are creating. Eg: pod.. etc

**metadata:** It consist of two fields. 1. Name 2. Lables

* **Name**: name of the pod
* **Labels**: These are used to logically group all related pods for displaying and managing. It helps in filtering and organizing
* It is an optional field.
* My recommendation is to define labels always

**Spec: Pod is like a wrapper around the container, pod can have multiple containers. We can give environment variables, port numbers, name, image.**

**Eg:**  In the above example, we are downloading the image on the docker hub and provide the name of the container

# 2. **Create and Display Pod**

**kubectl create if nginx-pod.yaml –** create pod by using this command

**kubectl get pod -**  list the pods

**kubectl get pod –o wide -**  list the pods using –o option

**kubectl get pod nginx-pod –o yaml -**  it displays the pod configuration in yaml format

**kubectl describe pod nginx-pod -**  provide the complete details of the pod

# 3. Test & Delete

# To get inside the pod

**kubectl exec –it <pod-name> -- /bin/sh**

# create Test HTML page on the pod

Cat <<EOF > /usr/share/nginx/html/test.html

<!DOCTYPE html>

<html>

<head>

<title>Testing....</title>

</head>

<body>

<h1 style="color:(90,70,250);">Hello, Kubernetes....!</h1>

<h2>Congratulations, you passed :-) </h2>

</body>

</html>

EOF

exit 🡪 exit from pod

# Expose the pod by using nodeport

**kubectl expose pod nginx-pod --type=NodePort --port=80**

# Command to display the expose details

**kubectl describe svc ngnix-pod**

**Test: http://<node-IP>:<NodePort>/test.html**

# Delete Pod & svc

**kubectl delete svc <service-name> (ngnix-pod)**

**kubectl delete pod <pod-name> (nginx-pod)**