Creating Kubernetes Pods

⌛ 00:53:25 Exit Lab ✓ Complete Lab

⏱ 1 hour duration Practitioner Rate this lab

VIDEOS GUIDE

Creating Kubernetes Pods

In this lab, we're going to create a simple Kubernetes pod. There's already a Kubernetes cluster set up in the lab environment that we can log in to and interact with. What we're going to do is create an Nginx web server as a pod running in that cluster.

Our Assignment

There are some specifications we need to be mindful of though:

- Use the `nginx` container image.
- The container needs to have a containerPort of `80` .
- Override the default `command` parameter, and set it to `nginx` .
    - This will require adding some of the default arguments back in: Pass in the `-g daemon off; -q` args to run Nginx in quiet mode.
- Create the pod in a namespace called `web` .

Once we've created the pod, we'll check its status with a couple of `kubectl` commands.

Get into the Server

First off, we need to log in. We'll use the credentials shown in the lab to log in to the public IP we can see there:

```
[user@$host ~]$ ssh cloud_user@<SERVER IP>
```

Create the yaml File

Once we're logged in, we can go ahead and start setting up our pod. First, we'll have to create a yaml file. Use whatever editor you like. The lab video shows vi.

```
[cloud_user@host ~]$ vi nginx.yaml
```

The yaml File Explained

This is what we will need to put into the file:

- `apiVersion` : We'll set it to `v1` .
- `kind` : This is the type of object we want to create. In our case, it will be `Pod` .
- `metadata` : This is where we'll give the pod a name (`nginx`) and say which namespace name we're going to use (`web`).
- `spec` : Here, we'll specify a container name and image (`nginx` for each).
    - `command` : We're using a custom command (`nginx`), which will override the container's default settings.
    - `args` : This will put some of those defaults we need (`-g` , `daemon off`) back in, as well as a custom argument (`-q`) to specify we want to run in *quiet* mode.
    - `ports` : We'll specify here what port we want for a container port. It's `80` in this case.

The yaml File Contents

The whole `nginx.yaml` file should look like this when we're done:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  namespace: web
spec:
  containers:
  - name: nginx
    image: nginx
    command: ["nginx"]
    args: ["-g", "daemon off;", "-q"]
    ports:
    - containerPort: 80
```

Creating the Pod Itself

Making the yaml file concludes the first leg of our journey. Any configurations we need are defined and ready to go. Now, we've got to actually create the pod using the definitions we specified in that file.

Create the pod with this:

```
[cloud_user@host ~]$ kubectl create -f ~/nginx.yaml
```

We'll get a `pod/nginx created` message. But let's make sure the pod was created. We should see the pod listed with this command:

```
[cloud_user@host ~]$ kubectl get pods -n web
```

Make sure `-n web` is in that command, so that we're looking in the `web` namespace instead of the default one. In the command's output, we should see a `STATUS` column, and it should say `Running`.

Just for good measure, let's describe the pod and look at some of the details. Like we did with the `get pods` command, don't forget the `-n web`.

```
[cloud_user@host ~]$ kubectl describe pod nginx -n web
```

In Conclusion

All of the information we see here is showing us we now have a healthy pod up and running in our Kubernetes cluster. Congratulations! We made it!

Tools

Lab Diagram   Instant Terminal

Credentials

How do I connect?

Cloud Server Kube Master

Username

cloud_user

Password

|BE!J4Fo

Kube Master Public IP

54.226.1.180

Launch Instant Terminal

How do I connect?

Additional Resources

Your company is getting ready to launch a new website, and they need you to set up an nginx web server in their Kubernetes cluster. The nginx server will need to be accessible via network in the future, so you will need to expose port 80 as a containerPort for the nginx container. Your team has also asked you to ensure that nginx runs in quiet mode for the time being to cut down on unnecessary log output. You can do this by setting the command to `nginx` and passing the following arg to the container: `-g daemon off; -q` . As this nginx server belongs to the Web team, you will need to create it in the team's `web` namespace.

To summarize:

- Use the `nginx` container image.
- The container needs a containerPort of `80` .
- Set the command to `nginx`
- Pass in the `-g daemon off; -q` args to run nginx in quiet mode.
- Create the pod in the `web` namespace.

Once the pod is created, you should be able to find it with `kubectl get pods -n web` . Once the pod is created, you can get more information about its current status with `kubectl describe pod nginx -n web` .

Learning Objectives

0 of 2 completed

□ **Create a yaml file containing the pod spec for the nginx pod.**

□ **Create the pod.**