

k8s - Task #1

Goals

- Install kubectl in Cloud9
- Get basic cluster information

- Similar to Docker - Task #1-4, but with k8s
- Run bare Ubuntu 20.04 to install python and necessary modules for simple numpy computation
- Use a private repository to run an interactive JARVIS image
- Copy files back and forth between pods and Cloud9
- For [kubectl vs Docker](#) references

Estimated Time: 45-60 minutes

Prerequisites

All steps below should be performed in your personal Cloud9 environment.

MAKE SURE to disable temporary credential management under Cloud9 Settings -> AWS Settings.

Steps

Environment Setup

1. Your Cloud9 environment includes a folder named "*init_scripts*"
Type the following commands to install necessary CLI tools and apply configuration for k8s:

```
# source ~/environment/init_scripts/cli_tools.sh  
# source ~/.bash_profile
```

Now your Cloud9 environment should be ready for k8s interaction.

Cluster Details

1. Type the following command to find out which nodes are available in the cluster:

```
# kubectl get nodes
```

And the following to get additional details:

```
# kubectl get nodes -o wide
```

And for even further details:

```
# kubectl describe nodes
```

In many ways, the above commands can be applied to any k8s object (pods, jobs, deployments etc.).

Working in a Cloud Container

1. Start a new pod/container based on bare Ubuntu 20.04:

```
# kubectl run -it test-ubuntu --restart=Never --image=ubuntu:20.04 /bin/bash
```

Note the arguments are similar to docker but slightly different.

It is necessary to supply a pod name (*test-ubuntu* here).

For kubectl *run* options check [here](#).

The restart policy is set so that k8s will not start the pod again after you exit.

2. Instead of the Dockerfile used in “Dockers - Task 1”, perform the same commands as in each **RUN** directive to install python3 and numpy.
3. Open a python shell (python3 / ipython3)
4. Write a tiny program that computes 20 random numbers
You may use: [numpy.random.random — NumPy v1.16 Manual](#)
5. Save those numbers to a file located in the home directory of the container
6. Exit the python shell and verify the file contents are OK
7. **REMEMBER** the full file path
8. Exit the container

Copying Files

1. In the Cloud9 terminal get again the list of pods running:

```
# kubectl get pods
```

Note the ID given to the pod.

2. Use kubectl [cp](#) subcommand to copy the resulting file (with random numbers) from the pod into the Cloud9 environment.
It is very similar to Linux scp usage.
3. Ensure file contents are what you expect.

JARVIS Service

1. Like when starting a bare ubuntu pod above
2. Use a similar command to start a new interactive pod running our jarvis-service from your private repository (need to specify the full URL of the image)
3. Create a text file in your Cloud9 environment that contains line messages to JARVIS
4. Use kubectl to copy the file from Cloud9 to the pod
5. Process the text with JARVIS into an output file inside the pod
6. Copy the resulting file from the pod to your Cloud9 environment and examine its contents