

Introduction to Cloud Computing – Exercise 5

Scope: Labels, Health, Application upgrades

Introduction:

In this laboratory we will get acquainted with labels and supervision

1. Labels

Download related files from lab 5. See `helloworld-pod-with-labels.yml`. See that a labels section has been added.

To create a new deployment, run the command

```
Kubectl create -f helloworld-pod-with-labels.yml
```

If everything was successful, you should be able to see the new *one created*.

```
Kubectl get pods
```

To see what *labels* are assigned to pods, call:

```
Kubectl get pods --show-labels
```

Do labels coincide with the yml file?

Labels can also be changed during operation. To change the label, call

```
Kubectl label pod/helloworld app=helloworldapp --overwrite
```

See if the labels have changed. **Change the label of the running application to your own, take a screenshot.**

To remove the label, we can do it using the command

```
Kubectl label pod/helloworld app-
```

You can see that the label has been removed.

2. Search by labels

In this section you will learn how you can search for the information you need using labels. Initialize the new structure using `sample-infrastructure-with-labels.yml`

```
Kubectl create -f sample-infrastructure-with-labels.yml
```

You should be able to see that many pods have been created.

To see them all, call

```
Kubectl get pods
```

In this example, the generated pods are supposed to simulate an online store, where we have different departments that different teams are working on. They also contain different versions of the application (dev, production, ...).

Wait until all pods are in the Running state.

Then, for example, to search for all containing the variable `env=production`, call

```
Kubectl get pods --selector env=production
```

To make sure that the correct pods are returned, call

```
Kubectl get pods --selector env=production --show-labels
```

Add your own label `student=your_name` to several pods, then search for them, take a screenshot, and show all and take a screenshot.

3. Advanced search by labels

In the case of labels, we can use advanced search. We would like to search for pods that have a `release-version` label from 1.0 to 2.0.

```
Kubectl get pods --selector 'release-version in (1.0,2.0)'
```

To verify the correct search, call the previous command with the parameter `- show labels`.

4. Delete by label

Just as `kubectl` allows advanced search by labels, it also allows you to remove individual pods that match the search criteria. For this purpose, we can use the command `kubectl delete pods --selector our_label`.

Using the given command, remove the pods with the label you added. Take a dump before, the typed command, and the dump after the deletion.

Importantly, just as labels were used for pods, they can also be used for other Kubernetes objects (service, deployment, ...)

5. Health checks

See `helloworld-with-probes.yml`

```
Nano helloworld-with-probes.yml
```

You may notice that in the spec section, we have two additional sections `readinessProbe` and `livenessProbe`. Call

```
Kubectl create -f helloworld-with-probes.yml
```

See deployments

```
Kubectl get deployment
```

Then call the command to make sure that our deployment is in the state we want (all states agree)

```
Kubectl get replicaset
```

In this case, all of them go as it should. Now consider the case when not everything goes according to plan.

For this purpose, we have prepared a second file. See *the file `helloworld-with-bad-readiness-probe.yml`* The change that occurred is a different Prop for `http` port. This will result in a container that has its `containerPort` set to port 80 and a `probe` port to port 90 to unresponsive.

Launch the app

```
Kubectl create -f helloworld-with-bad-readiness-probe.yml
```

After starting, wait for a while. See deployments and pods. You will notice that the new deployment is not ready.

Despite everything, the pod is able to run. To view more information, follow the

```
Kubectl describe under NAZWA_PODA
```

The section that interests us is *Events*. **Do you find any bug there?**

For *readinessProbe*, see the file `helloworld-with-bad-liveness-probe.yml` and the `livenessProbe` settings.

Deploy by using this file. Check available pods with the command

```
Kubectl get pods
```

And deployments with the command

```
Kubectl get deployments
```

What happened after a while?

To see more precisely, execute the *describe* command on the given pod.

What is the difference and what `readinessProbe` and `livenessProbe` inform us about

6. Upgrade/Rollback

Open *helloworld-black.yml* file is very similar to the earlier file `helloworld` but choose here a different image – `helloworld:black`

Create a new deployment

```
Kubectl create -f helloworld-black.yml --record
```

We have added *the* `--record` parameter, this parameter will allow us to keep the history of undoing changes.

Once properly launched, it should open our service

```
Minikube service navbar-service
```

If the page does not open, try to give our service `nodePort` (as in the previous laboratory).

As you will see on the site, the navbar at the top of the page is black. We will try to change its color to another. We have already prepared a second image with a blue navigation bar. To replace the image, for a running service use the command

```
Kubect1 set image deployment/navbar-deployment helloworld=karthequian/helloworld:blue
```

See available deployments. And then refresh your page, you should see a changed bar at the top of the page. [Take a screenshot](#)

Call

```
Kubect1 get rs
```

You can see that we have two replications with different guides at the end, and one has 3 pods and the other zero.

See available pods and deployments. What happened was that by uploading a new image, a new replication was created and therefore new pods.

Let's see the story

```
Kubect1 rollout history ployment/navbar-deployment
```

You may see two revisions. This is made possible by adding the *-record* argument

To perform a rollback to a specific version, do:

```
Kubect1 rollout undo deployment/navbar-deployment -to-revision=NUMER_REWIZJI
```

[Take a screenshot after restoring the previous version of our site.](#)