

# Introduction to Cloud Computing – Exercise 9

**Scope:** Helm charts

Admission:

The goal of the lab is to get acquainted with the Helm package manager for Kubernetes

## 1. Getting to know Helm

Find information about Helm. Describe in the report in a few words what it is for, and what are its benefits.

## 2. Installation of Helm

Use the instructions on the <https://helm.sh/docs/intro/install/> page to install the Helm add-on. Use the package manager installation (From Apt (Debian/Ubuntu section)).

To make sure that the installation was successful, perform

```
helm -h
```

## 3. Repositories

Once we have Helm installed, we can start looking for ready-made charts (understood as installation "maps"). To see available repositories, call

```
helm repo list
```

As Helm does not have a greyhound repository configured, an error was returned. To add a new repository, call

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

At this point, you can call the command again

```
Helmet repo list
```

You should see the added repository. Now that we have a repository configured, we can start searching them to find ready-made greyhounds.

Let's look for ready-made greyhounds associated with *apache*

```
Helm search repo apache
```

In the repository there is a very large variety of ready-made solutions. You may notice that in the list we have the name, the version of the chart, the version of the application and the description. Try searching the repository for other keywords. In the inventory, you receive only the latest versions of the software. To search for other versions, add `--version` to the keyword.

If you want to delete a repository, you can use *the helm remove NAZWA\_REPOZYTORIUM* command, but let's not do it now.

## 4. MySql installation using Helm chart

Before we start a new installation, clean up Kubernetes. Open three command line windows. In the first, call

```
Kubect1 get pods
```

You should get information that there are no pods – if you still have any bottoms running – remove them.

In the second window, call

```
Minikube ssh
```

This command will allow you to go inside the minikube. Then call the command

```
Docker images
```

For a clean minikube installation, you should not see a MySql image here (if there is, this is due to previous tasks performed in the lab).

Return to the first terminal window. Create a new Helm installation now, where when giving the name (*mydb*) we must remember that it is unique in a given namespace.

```
Helm install mydb bitnami/mysql
```

As a result, we should see the information after the installation. Make a copy of them to a text editor.

Then call `kubectl get pods` to see if a new mysql instance appears.

Now go back to the text editor and follow the command from the *Services* department to check if our database is available. If you received a return message, it means that the server is running.

Find the command in the copied content to find out what the password is for the database user. Save them in a text editor. Then execute We can see that the password is stored in secret (... `get secret...` ) means that it is generated when the pod is created. Helm chart, which we launched to install MySQL did a number of actions for us - > created a pod, service, secret and many other things.

Execute the commands in the *Try connect to your database* section in the third terminal window. To connect, you will need the password you received earlier. To make sure that we have successfully connected to the database, call

```
Show databases
```

If you have correctly seen the available databases, everything is configured correctly. At this point, you can return to the second pane and call again

```
Docker images
```

We should be able to see the new Mysql images.

## 5. Uninstalling the package/deployment

View all installations using

```
Helm list
```

We should see the currently installed mysql server. This is a list of all packages in the base namespace

To remove the package use the `uninstall` command. Call

```
Helm uninstall mydb
```

After the server package is properly uninstalled, list the active pods to see that there is no more mysql server.

## 6. Variable management

Restart the minikube. Call

```
Minikube delete
```

And then

```
Minikube start
```

Open a new terminal window. We will now maintain this helmet installation, and we will pass our variables inside the installation. To specify a variable with a user password for my sql, we can call the command (**but do not call**)

```
Helm install mydb bitnami/mysql -set auth.rootPassword = test1234
```

In this case, when we want to specify one / two variables, we can use them by command. The right approach is to use yaml files. In this case, we create a new yaml file, let's call it *values.yaml* save it in a convenient place. Put in file:

```
auth:
  rootPassword: "test1234"
```

Now we can call the command

```
Helm install mydb bitnami/mysql --values PATH_TO_FILE
```

Once everything is successful, you can use the command to display the password "echo Password..." See if she returned a good password.

To make sure everything is configured correctly, run the client (described above) and see if the client worked.