

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»
(Самарский университет)

Институт информатики, математики и электроники
Факультет информатики
Кафедра суперкомпьютеров и общей информатики

Отчет по лабораторной работе №2

Дисциплина: «Методология разработки DevOps»

Тема: «**Helm. JupyterHub**»

Выполнил: Коршиков В.И.

Группа: 6233-010402D

Самара 2022

ЗАДАНИЕ

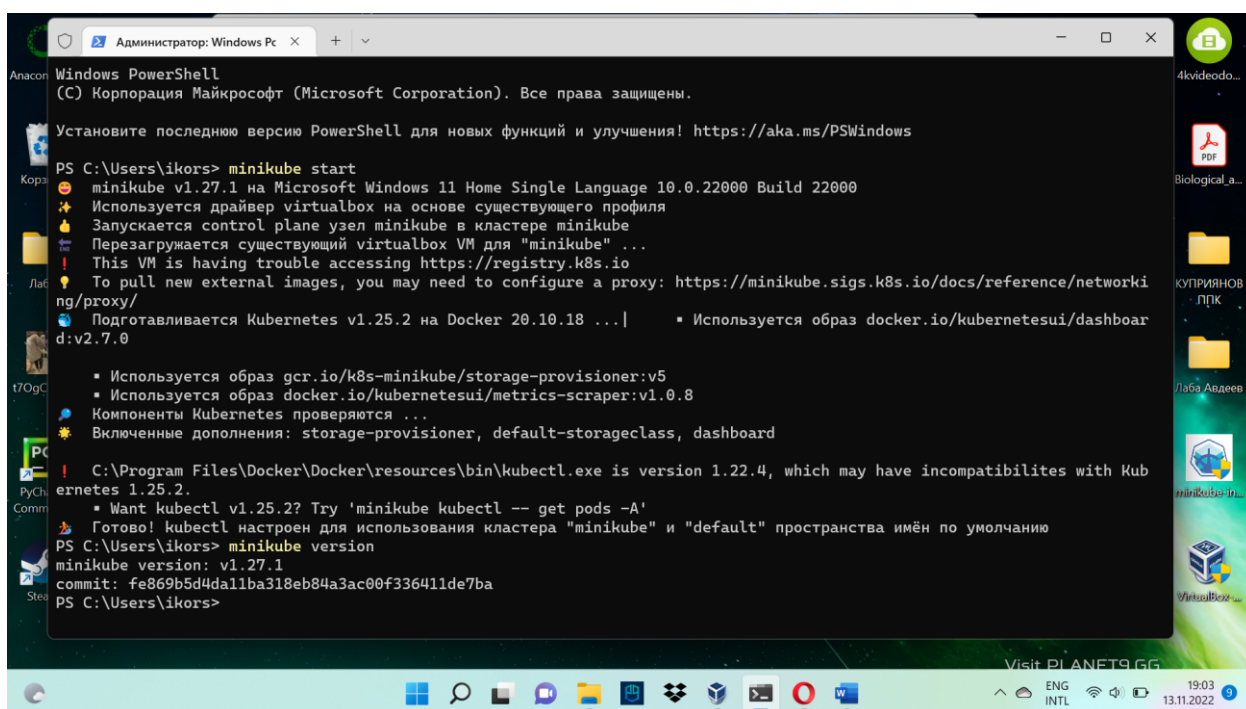
1. Install Helm, Jupyterhub according with <https://github.com/jupyterhub/zero-to-jupyterhub-k8s> <https://z2jh.jupyter.org/en/stable/>
2. Try to run test-notebooks as in following video [11:18] [Kube 99] Zero to Jupyterhub in Kubernetes | Getting Started Guide <https://youtu.be/Da1qn7-RHvY?t=676>
3. Make report with screens of:
4. JupiterHub notebook `print("Hello world")` or some simple command for your choice executed.

СОДЕРЖАНИЕ

| | |
|--------------------------------------|----|
| Задание 1. Установка helm. | 4 |
| Задание 2. Установка JupyterHub..... | 6 |
| Заключение | 12 |

Задание 1. Установка helm.

Запустим консоль(Администратор)



Зачем устанавливать helm?

Helm - это менеджер пакетов. С его помощью разработчики могут "легко управлять и развертывать приложения на кластере Kubernetes". У меня ОС Windows 11. Проще всего ставить его через консоль, но для этого надо поставить менеджер пакетов Chocolatey. Это можно сделать не выходя из терминала с помощью команды.

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object  
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

На сайте <https://helm.sh/> указана команда для установки Helm :

```
> choco install kubernetes-helm
```

```
Администратор: Windows Pc
Windows PowerShell
(С) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

PS C:\Users\ikors> choco install kubernetes-helm
Chocolatey v1.2.0
Installing the following packages:
kubernetes-helm
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-helm 3.10.1... 100%

kubernetes-helm v3.10.1 [Approved]
kubernetes-helm package files install completed. Performing other installation steps.
The package kubernetes-helm wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): y

Downloading kubernetes-helm 64 bit
from 'https://get.helm.sh/helm-v3.10.1-windows-amd64.zip'
Progress: 100% - Completed download of C:\Users\ikors\AppData\Local\Temp\chocolatey\kubernetes-helm\3.10.1\helm-v3.10.1-
windows-amd64.zip (13.97 MB).
Download of helm-v3.10.1-windows-amd64.zip (13.97 MB) completed.
Hashes match.
Extracting C:\Users\ikors\AppData\Local\Temp\chocolatey\kubernetes-helm\3.10.1\helm-v3.10.1-windows-amd64.zip to C:\Prog
ramData\chocolatey\lib\kubernetes-helm\tools...
C:\ProgramData\chocolatey\lib\kubernetes-helm\tools
ShimGen has successfully created a shim for helm.exe
```

```
Администратор: Windows Pc
Chocolatey v1.2.0
Installing the following packages:
kubernetes-helm
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-helm 3.10.1... 100%

kubernetes-helm v3.10.1 [Approved]
kubernetes-helm package files install completed. Performing other installation steps.
The package kubernetes-helm wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): y

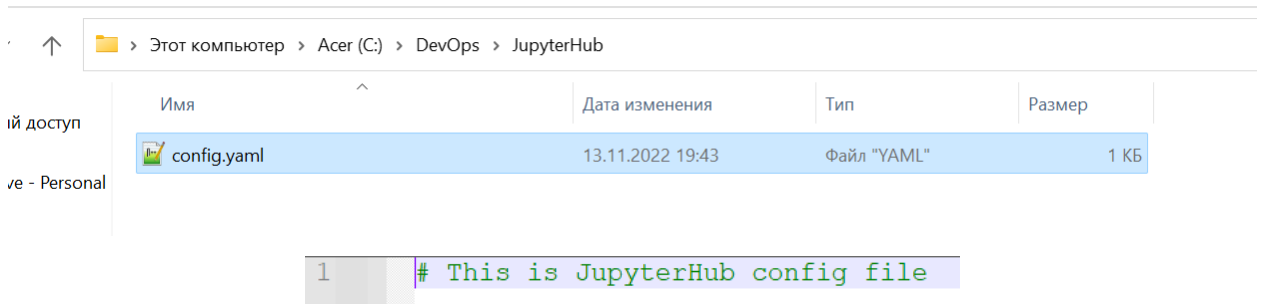
Downloading kubernetes-helm 64 bit
from 'https://get.helm.sh/helm-v3.10.1-windows-amd64.zip'
Progress: 100% - Completed download of C:\Users\ikors\AppData\Local\Temp\chocolatey\kubernetes-helm\3.10.1\helm-v3.10.1-
windows-amd64.zip (13.97 MB).
Download of helm-v3.10.1-windows-amd64.zip (13.97 MB) completed.
Hashes match.
Extracting C:\Users\ikors\AppData\Local\Temp\chocolatey\kubernetes-helm\3.10.1\helm-v3.10.1-windows-amd64.zip to C:\Prog
ramData\chocolatey\lib\kubernetes-helm\tools...
C:\ProgramData\chocolatey\lib\kubernetes-helm\tools
ShimGen has successfully created a shim for helm.exe
The install of kubernetes-helm was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\kubernetes-helm\tools'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Users\ikors>
```

Теперь можно поставить Jupyterhub.

Задание 2. Установка JupyterHub.

Перед установкой создадим конфигурационный файл *config.yaml*.
Запишем в него комментарий.

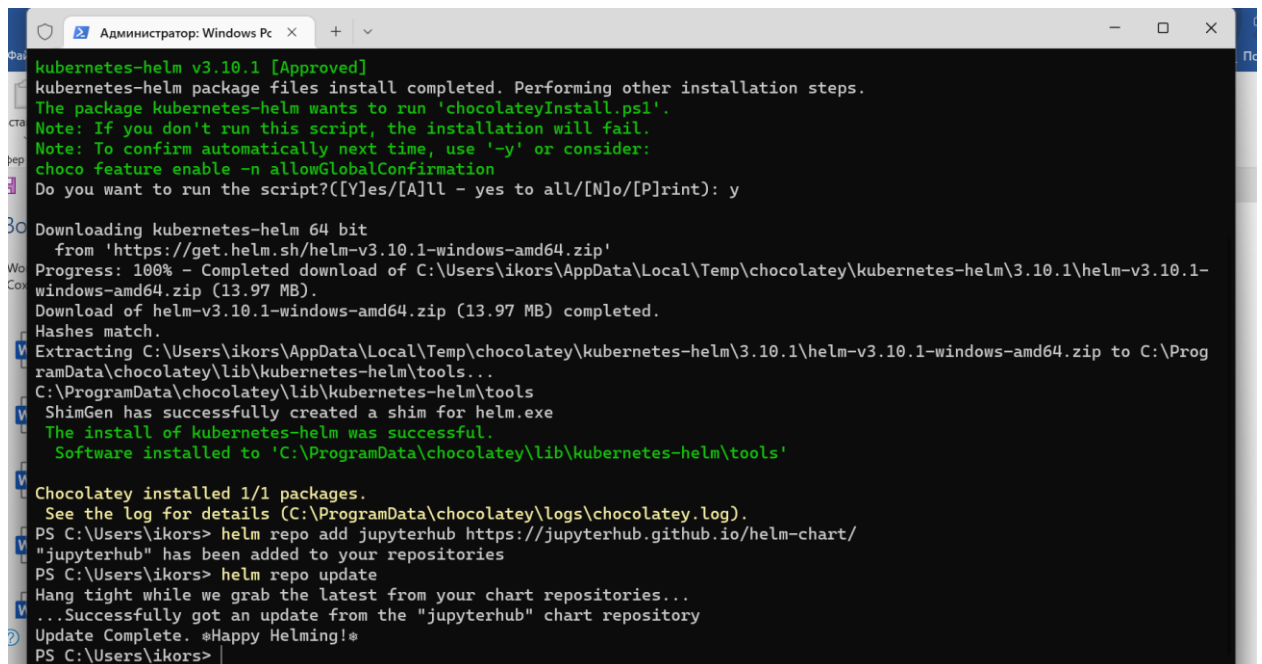


С помощью уже установленного менеджера пакетов, надо добавить к себе репозиторий JupyterHub.

Для этого выполним следующие команды:

```
> helm repo add jupyterhub https://jupyterhub.github.io/helm-chart/
```

```
> helm repo update
```



Пора запустить локальный кластер Kubernetes. Для этого воспользуемся
командой:

```
> minikube start
```

```
Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Users\ikors> helm repo add jupyterhub https://jupyterhub.github.io/helm-chart/
"jupyterhub" has been added to your repositories
PS C:\Users\ikors> helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "jupyterhub" chart repository
Update Complete. ✨Happy Helming!✨
PS C:\Users\ikors> minikube start
🐳 minikube v1.27.1 на Microsoft Windows 11 Home Single Language 10.0.22000 Build 22000
🔧 Используется драйвер virtualbox на основе существующего профиля
🔥 Запускается control plane узел minikube в кластере minikube
🔄 Обновляется работающий virtualbox "minikube" VM ...
❗ This VM is having trouble accessing https://registry.k8s.io
💡 To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
🌐 Подготавливается Kubernetes v1.25.2 на Docker 20.10.18 ... |   ▪ Используется образ docker.io/kubernetesui/dashboard:v2.7.0
▪ Используется образ gcr.io/k8s-minikube/storage-provisioner:v5
▪ Используется образ docker.io/kubernetesui/metrics-scraper:v1.0.8
🔍 Компоненты Kubernetes проверяются ...
🌟 Включенные дополнения: dashboard

❗ C:\Program Files\Docker\Docker\resources\bin\kubectl.exe is version 1.22.4, which may have incompatibilites with Kubernetes 1.25.2.
▪ Want kubectl v1.25.2? Try 'minikube kubectl -- get pods -A'
🔥 Готово! kubectl настроен для использования кластера "minikube" и "default" пространства имён по умолчанию
PS C:\Users\ikors>
```

Теперь пришла пора настроить конфигурационный файл. Монтируем в директорию, где он лежит:

```
❗ C:\Program Files\Docker\Docker\resources\bin\kubectl.exe is version 1.22.4, which may have incompatibilites with Kubernetes 1.25.2.
▪ Want kubectl v1.25.2? Try 'minikube kubectl -- get pods -A'
🔥 Готово! kubectl настроен для использования кластера "minikube" и "default" пространства имён по умолчанию
PS C:\Users\ikors> cd C:\DevOps\JupyterHub
PS C:\DevOps\JupyterHub>
```

Произведём его настройку при помощи следующей команды:

```
helm install <helm-release-name> jupyterhub/jupyterhub \
--namespace <k8s-namespace> \
--create-namespace \
--values config.yaml
```

где:

- *<helm-release-name>* относится к имени выпуска Helm, идентификатору, необходимому для различия установленных диаграммы.
- *<k8s-namespace>* относится к пространству имен Kubernetes, идентификатору, используемому для группировки ресурсов Kubernetes, в данном случае всех ресурсов Kubernetes, связанных с диаграммой JupyterHub.

```
> helm install notebookhub jupyterhub/jupyterhub --namespace minikube --create-namespace --values config.yaml
```

Теперь необходимо провести установку по умолчанию для флага пространства имён, для этого вернёмся на предыдущий терминал

```
> kubectl config set-context $(kubectl config current-context) --namespace minikube
```

```
> kubectl get pod --namespace minikube
```



```
Context minikube modified.
PS C:\DevOps\JupyterHub> kubectl get pod --namespace minikube
NAME                                READY   STATUS    RESTARTS   AGE
continuous-image-puller-r6d99      1/1     Running   0           149m
hub-74f6d84f8-crtqx                1/1     Running   0           149m
proxy-5dbcc9699f-f56pt             1/1     Running   0           149m
user-scheduler-97d8f6f68-gkq8w     1/1     Running   1 (8m47s ago) 149m
user-scheduler-97d8f6f68-wnpnr     1/1     Running   0           149m
PS C:\DevOps\JupyterHub>
```

Найдём IP-адрес, который можно использовать для доступа к JupyterHub. Выполняем команду и смотрим на EXTERNAL-IP proxy-public:

```
> kubectl --namespace minikube get service proxy-public
```

```
NAME                                READY   STATUS    RESTARTS   AGE
continuous-image-puller-r6d99      1/1     Running   0           149m
hub-74f6d84f8-crtqx                1/1     Running   0           149m
proxy-5dbcc9699f-f56pt             1/1     Running   0           149m
user-scheduler-97d8f6f68-gkq8w     1/1     Running   1 (8m47s ago) 149m
user-scheduler-97d8f6f68-wnpnr     1/1     Running   0           149m
PS C:\DevOps\JupyterHub> kubectl --namespace minikube get service proxy-public
NAME      TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
proxy-public LoadBalancer 10.106.16.151 <pending>     80:32607/TCP 155m
PS C:\DevOps\JupyterHub>
```

Вместо *EXTERNAL-IP* вместо IP-адреса указано *<pending>*, значит нам нужно провести присвоение IP-адреса. Для этого выполним инструкцию: <https://minikube.sigs.k8s.io/docs/handbook/accessing/>.

Первым делом необходимо запустить тоннель (аналог VPN) при помощи команды:

```
> minikube tunnel
```

```
✓ Tunnel successfully started

NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minikube.sigs.k8s.io/docs/handbook/accessing/#access-to-ports-1024-on-windows-requires-root-permission
Starting tunnel for service proxy-public.
Starting tunnel for service hello-minikube1.
```

После того, как запуск тоннеля выполнен, делаем следующие команды:

Развертывание Kubernetes:

```
> kubectl create deployment hello-minikube1 --image=k8s.gcr.io/echoserver:1.4
```

Создаём сервис Kubernetes с типом LoadBalancer:

```
> kubectl expose deployment hello-minikube1 --type=LoadBalancer --port=8080
```

Проверяем внешний IP:

```
> kubectl get svc
```

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE |
|-----------------|--------------|----------------|-------------|----------------|-------|
| hello-minikube1 | LoadBalancer | 10.108.103.228 | 127.0.0.1 | 8080:31826/TCP | 4s |
| hub | ClusterIP | 10.103.45.34 | <none> | 8081/TCP | 6m51s |
| proxy-api | ClusterIP | 10.105.158.35 | <none> | 8081/TCP | 6m51s |
| proxy-public | LoadBalancer | 10.101.181.96 | 127.0.0.1 | 80:31056/TCP | 6m51s |

Получилось!!!

Теперь в графе EXTERNAL-IP появился адрес:

<http://127.0.0.1>

Теперь мы можем перейти по ссылке в браузер и окажемся в окне авторизации JupyterHub.



Sign in

Warning: JupyterHub seems to be served over an unsecured HTTP connection. We strongly recommend enabling HTTPS for JupyterHub.

Username:

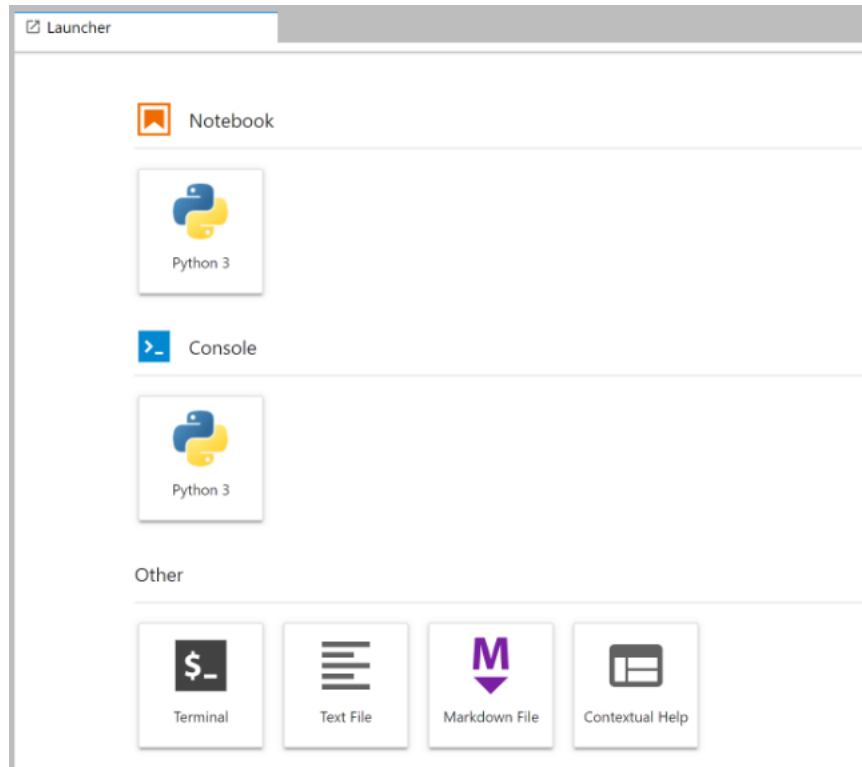
Password:

Sign in

Придумываем логин и пароль, я взял комбинацию логин: vlad, пароль: korsh.

Теперь, когда мы смогли авторизоваться, мы можем запускать ноутбуки.

После успешной авторизации, попадаем на главное окно, на котором нам предлагаются ноутбуки на выбор. Выбираем первый и продолжаем.



Теперь мы можем опробовать команду: «Hello World»

```
In [2]: print "Hello World"
```

```
Hello World
```

ЗАКЛЮЧЕНИЕ

В данной работе получены навыки работы с Chocolatey, helm, minikube, Kubernetes и JupyterHub. Произведено подключение к публичному прокси и написаны команды по выводу информации на notebook.