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

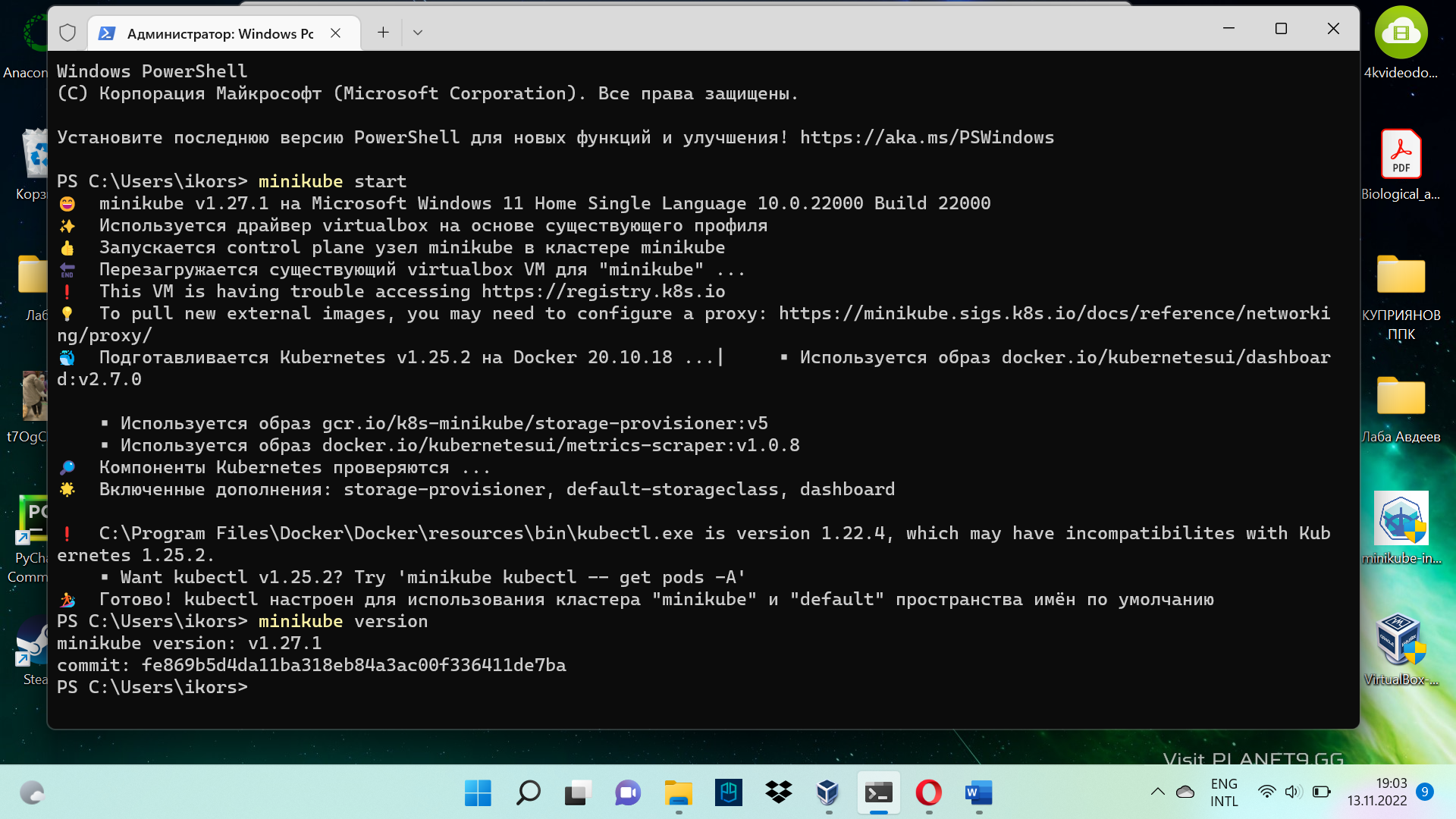
[Задание 1. Установка helm. 4](#_Toc119255037)

[Задание 2. Установка JupyterHub. 6](#_Toc119255038)

[Заключение 12](#_Toc119255046)

# **Задание 1. Установка helm.**

Запустим консоль(Администратор)



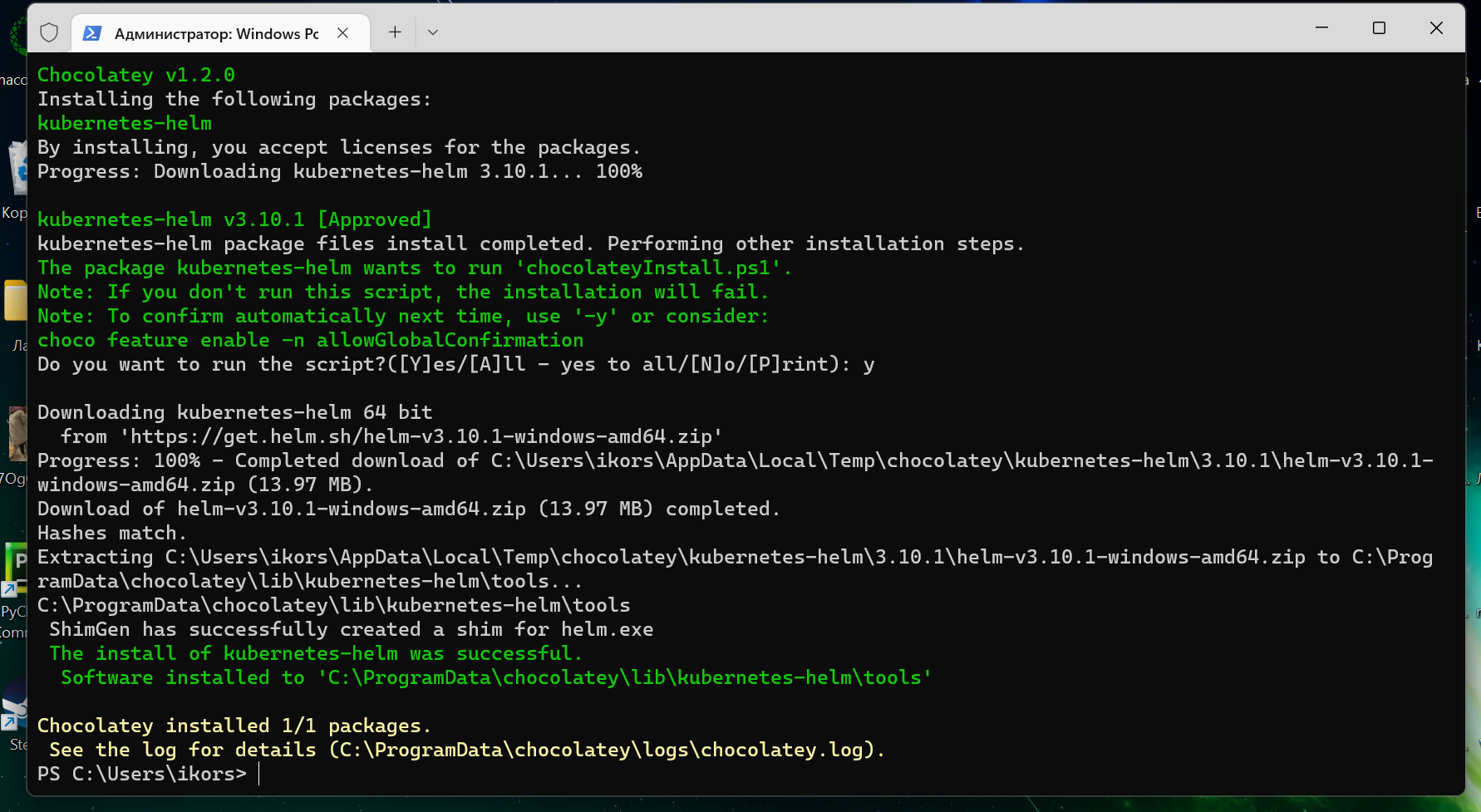
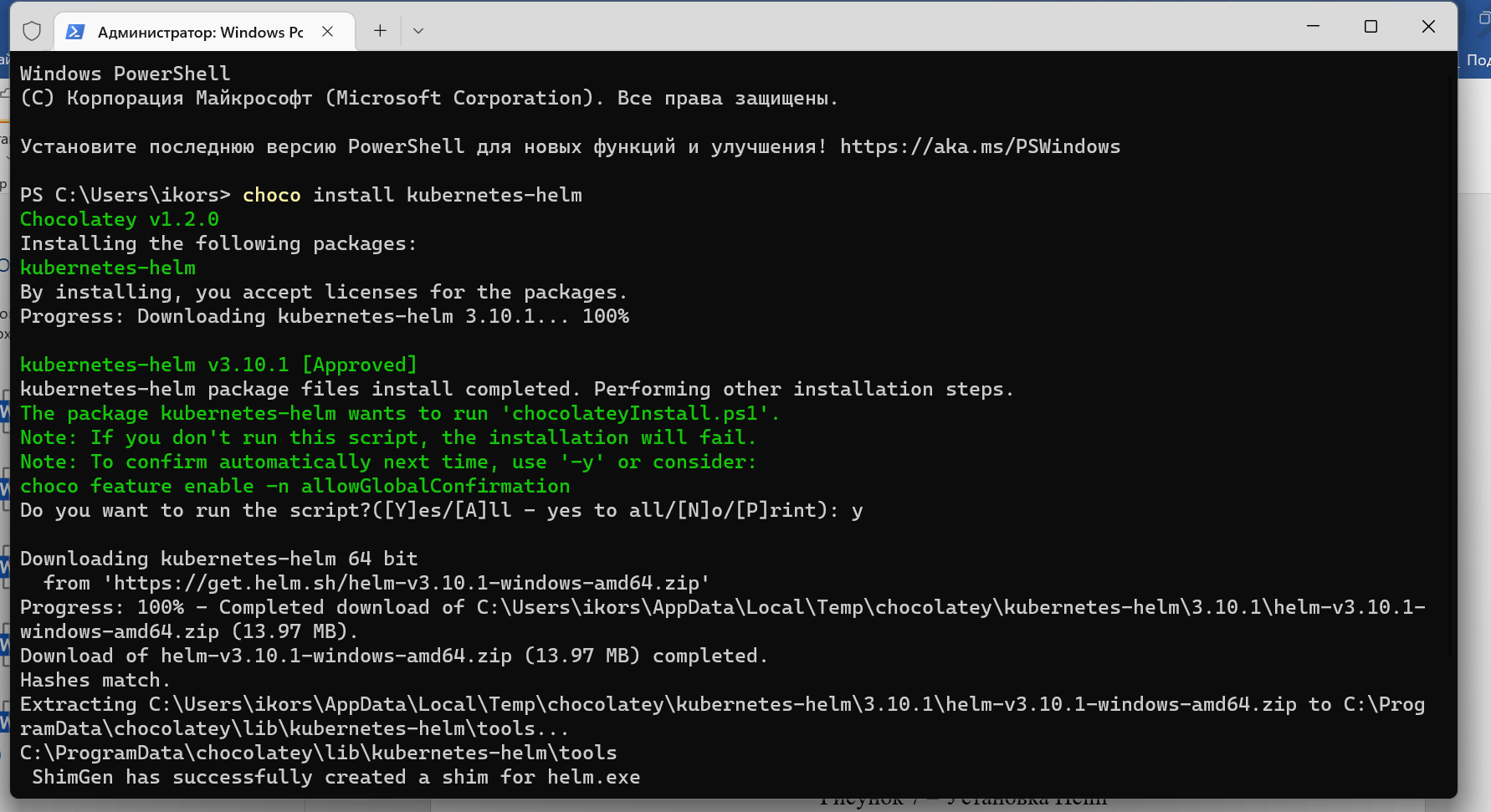
Зачем устанавливать helm?

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/](https://helm.sh/Его) указана команда для установки Helm :

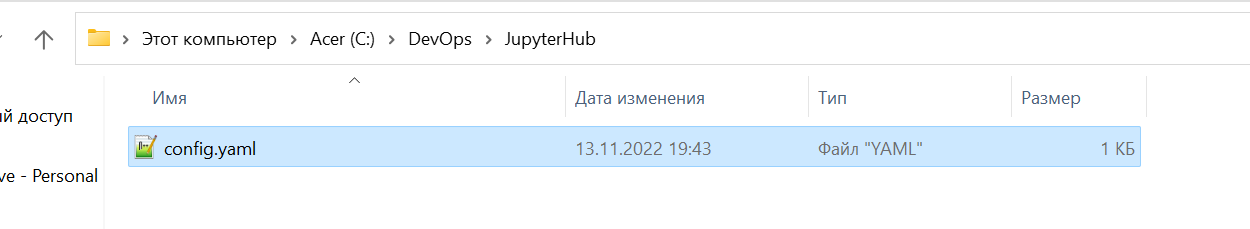
> choco install kubernetes-helm



Теперь можно поставить Jupyterhub.

# **Задание 2. Установка JupyterHub.**

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



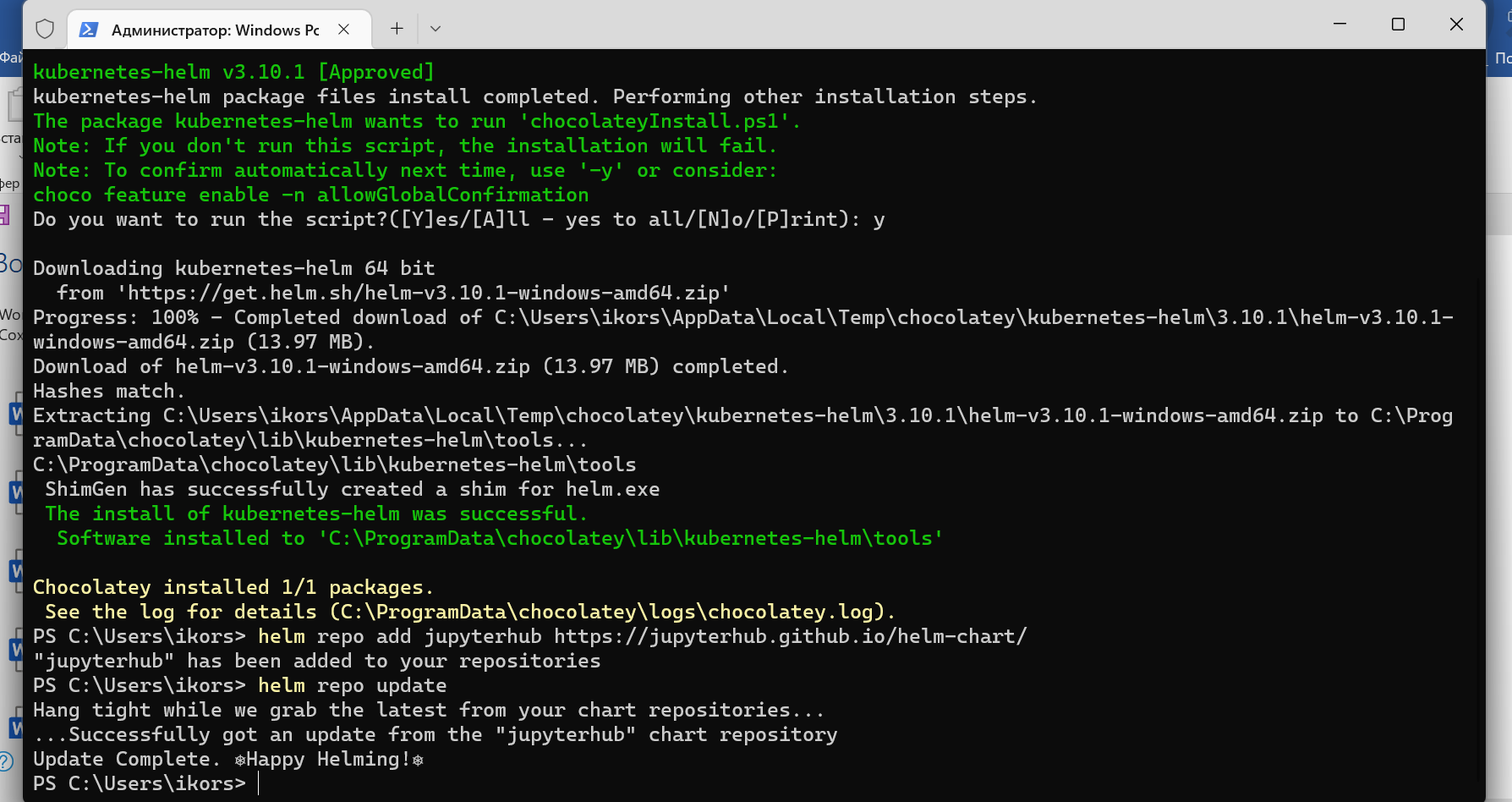


С помощью уже установленного менеджера пакетов, надо добавить к себе репозиторий JupyterHub.

Для этого выполним следующие команды:

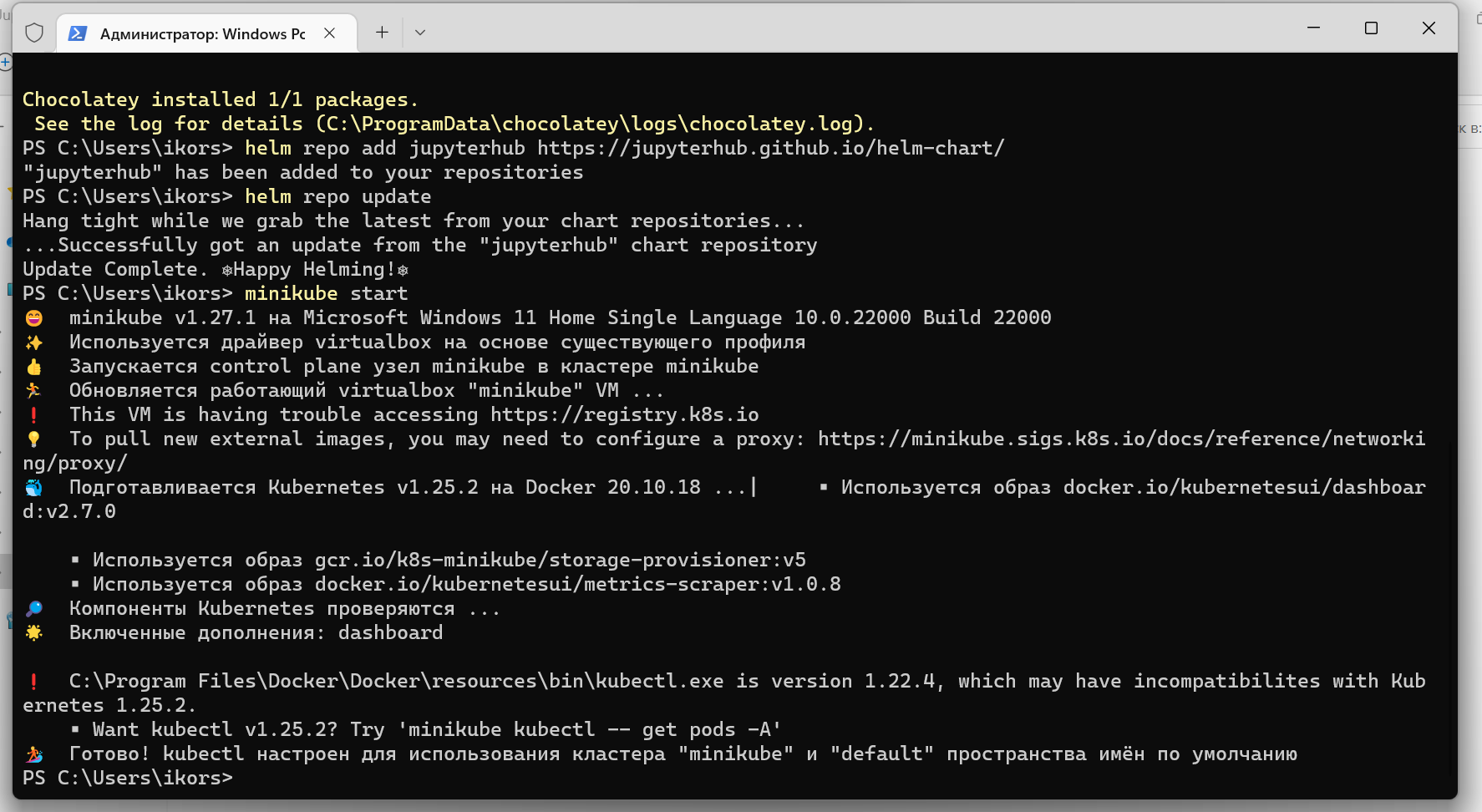
> helm repo add jupyterhub https://jupyterhub.github.io/helm-chart/

> helm repo update

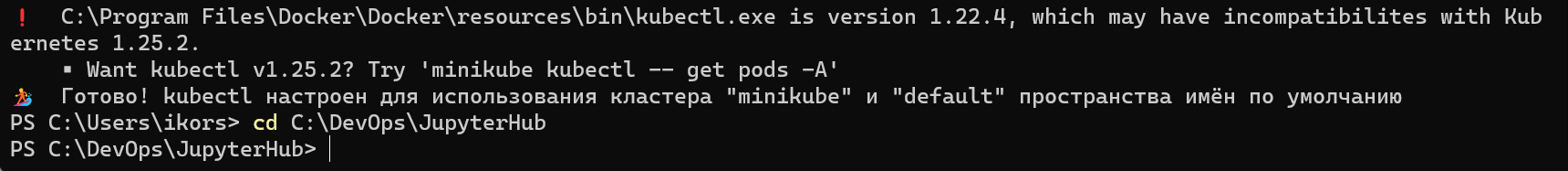


Пора запустить локальный кластер Kubernetes. Для этого воспользуемся командой:

> minikube start



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

 Произведём его настройку при помощи следующей команды:

helm install <helm-release-name> jupyterhub/jupyterhub \

--namespace <k8s-namespace> \

--create-namespace \

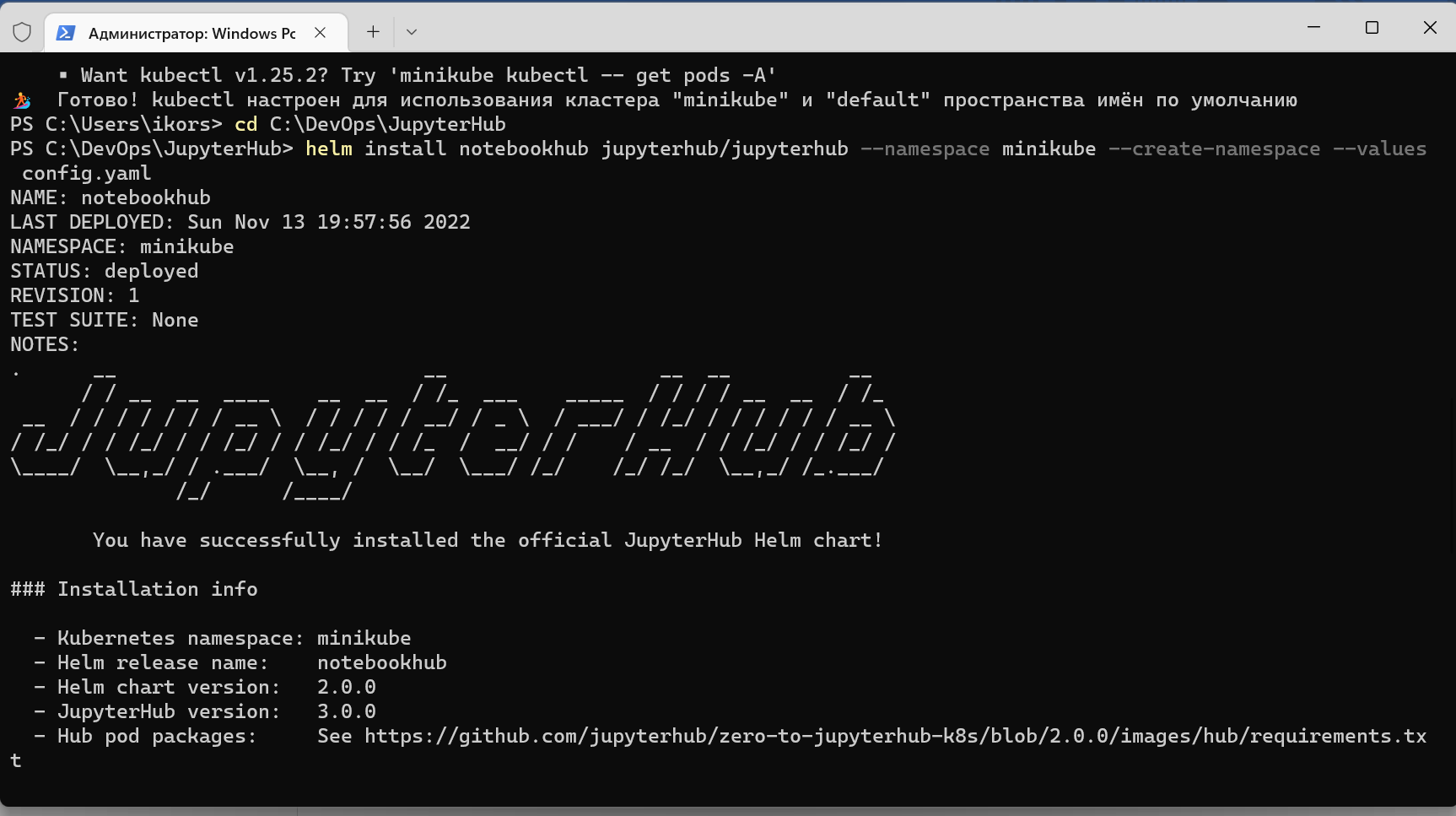
--values config.yaml

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.

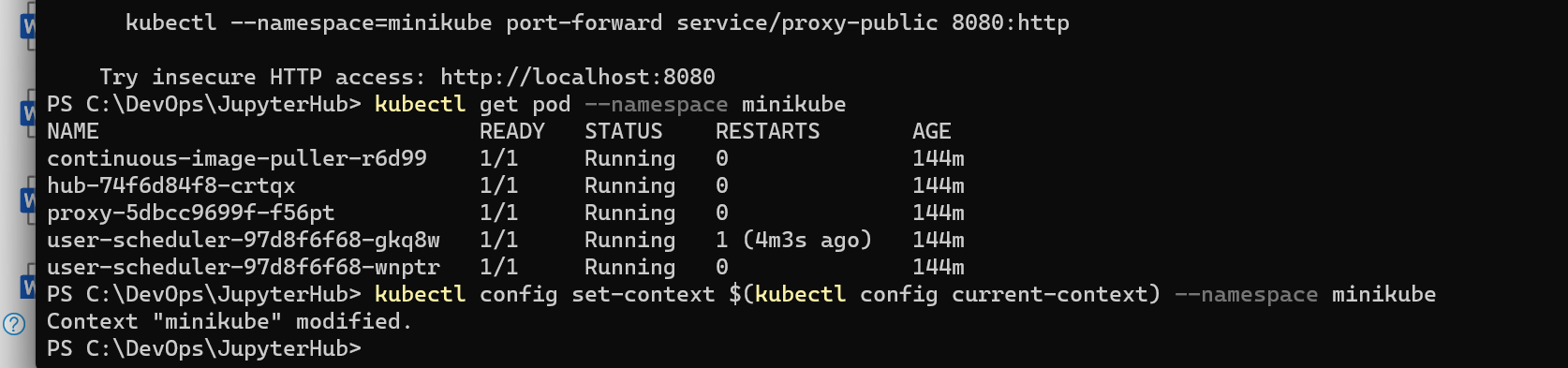
В моем случае эта команда выглядит следующим образом:

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



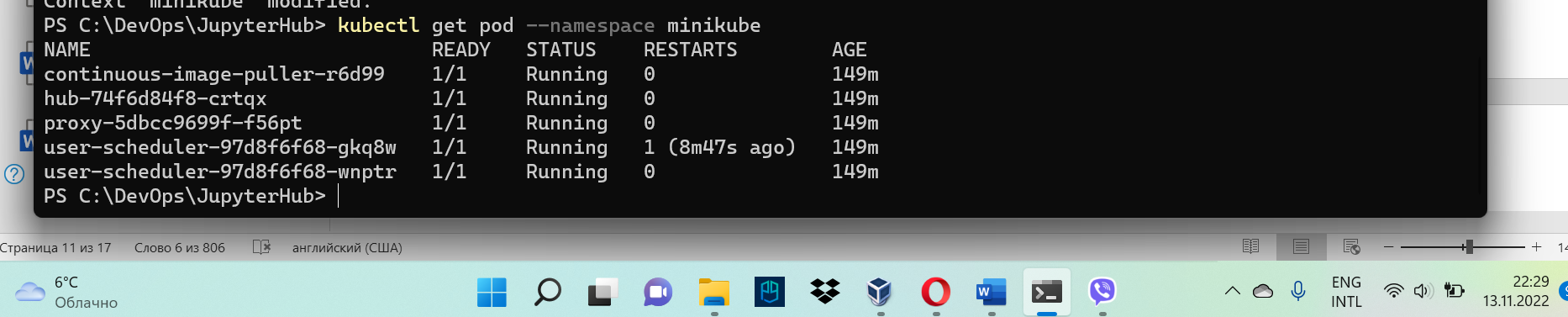
Теперь необходимо провести установку по умолчанию для флага пространства имён, для этого вернёмся на предыдущий терминал

> kubectl get pod --namespace minikube

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

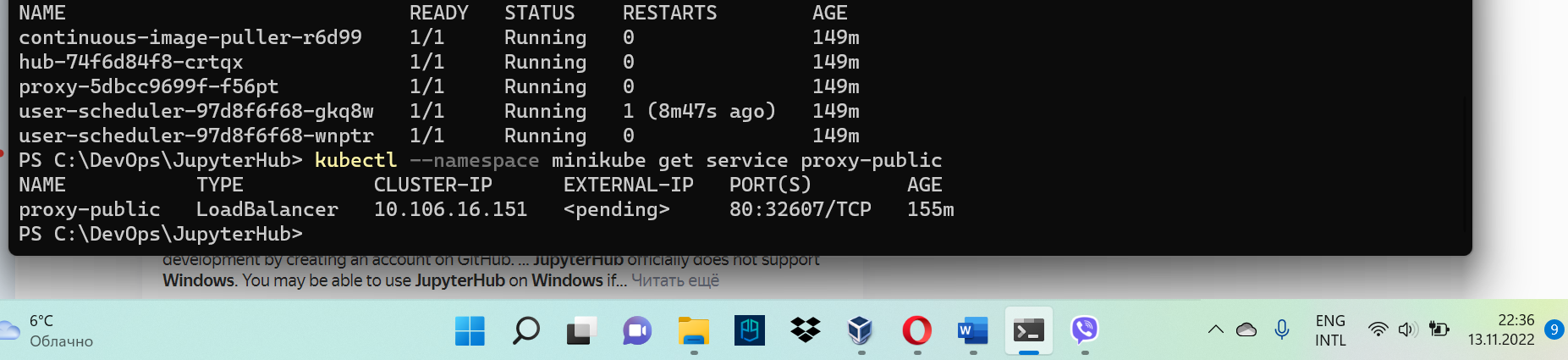
Теперь посмотрим, запущены ли proxy и hub.

> kubectl get pod --namespace minikube



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

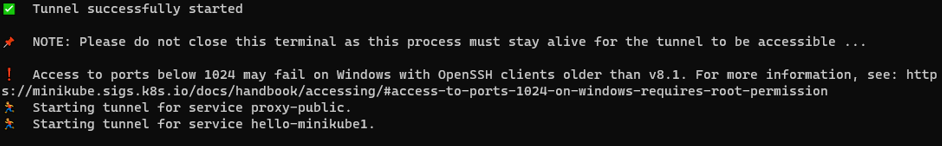
> kubectl --namespace minikube get service proxy-public



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



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

Развертывание Kubernetes:

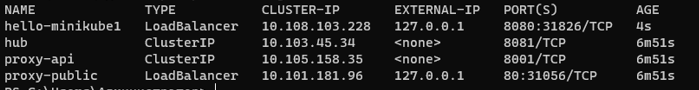
> kubectl create deployment hello-minikube1 --image=k8s.gcr.io/echoserver:1.4

Создаём сервис Kubernetes с типом LoadBalancer:

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

Проверяем внешний IP:

>kubectl get svc

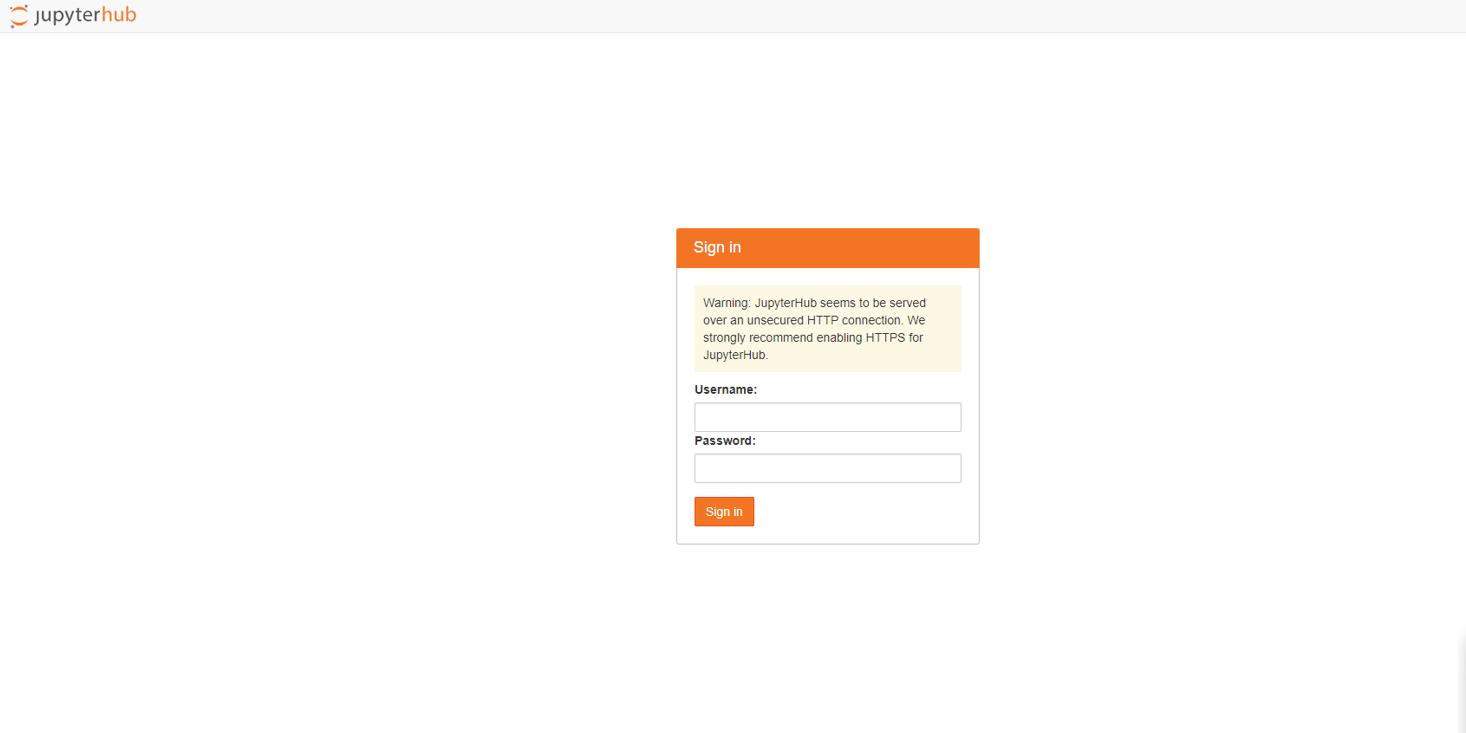


Получилось!!!

Теперь в графе EXTERNAL-IP появился адресс:

<http://127.0.0.1>

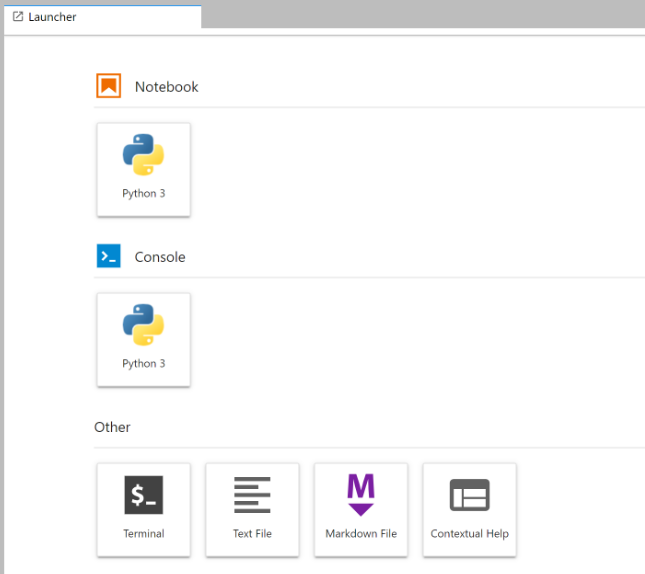
Теперь мы можем перейти по ссылке в браузер и окажемся в окне авторизации JupyterHub.



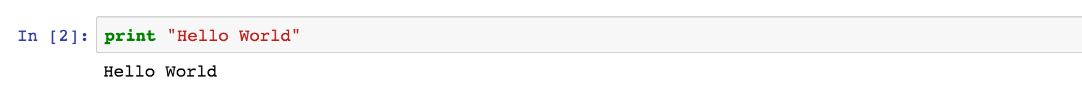
Придумываем логин и пароль, я взял комбинацию логин: vlad, пароль: korsh.

Теперь, когда мы смогли авторизироваться, 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»



# **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.