EPAM University Programs

DevOps external course

Module 2 Virtualization and Cloud Basic

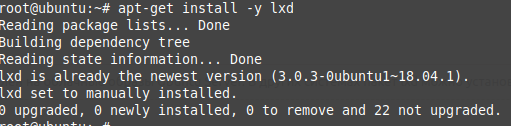
TASK 2.4

Работа с lxc в Ubuntu

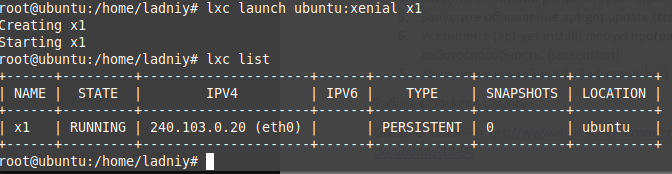
Documentation - <https://help.ubuntu.com/lts/serverguide/lxd.html>

<https://linuxcontainers.org/lxd/getting-started-cli/>

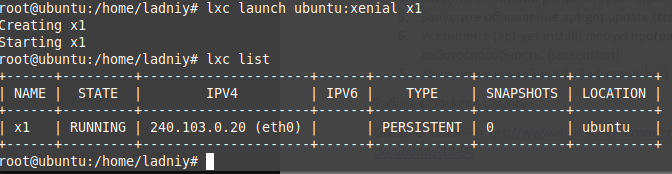
1. Установить lxc (screenshot)



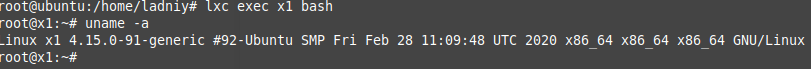
1. Запустить lxc launch для любой из версий Убунту (screenshot)



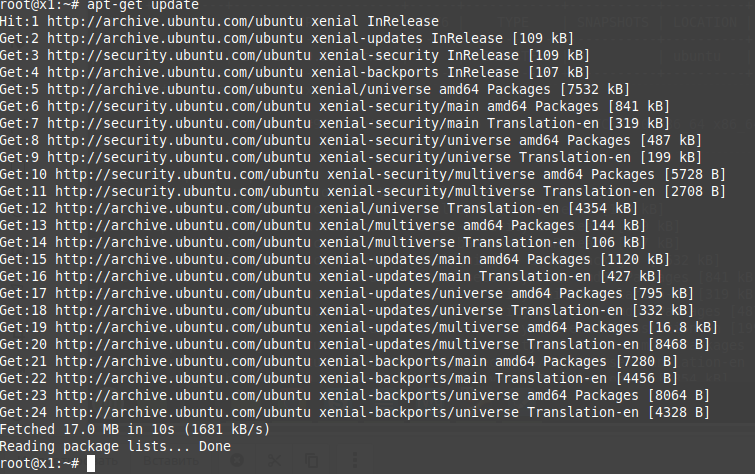
1. По окончании загрузки убедиться, что машина стартовала lxc list (screenshot)



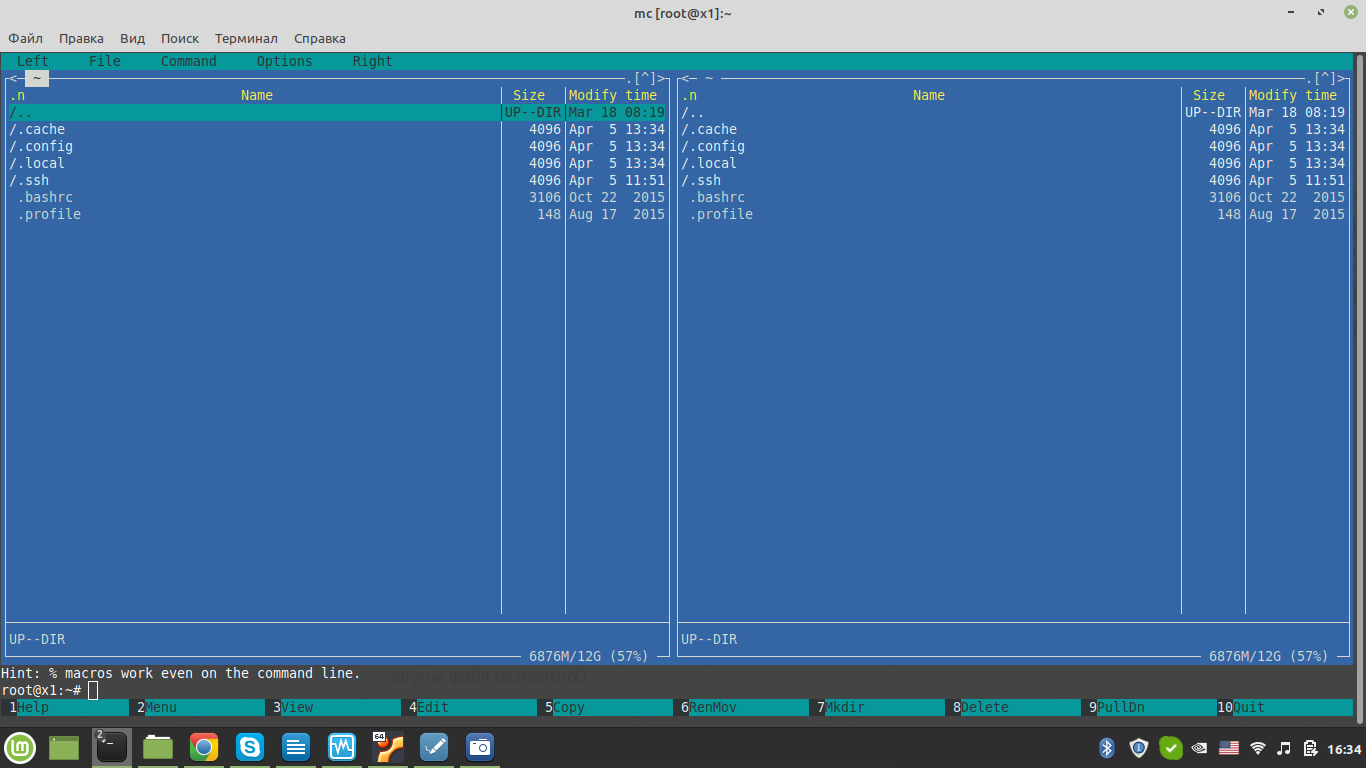
1. Зайдите в контейнер с командной строкой bash /bin/bash (screenshot)



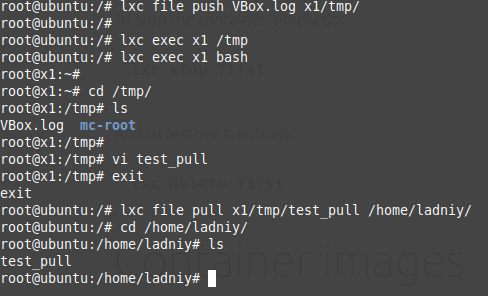
1. Запустите обновление apt-get update (screenshot)



1. Установите (apt-get install) любую программу в контейнер. Например mc. Проверьте работоспособность. (screenshot)



1. Загрузите в контейнер файл (screenshot) и скачайте с контейнера другой файл (screenshot).

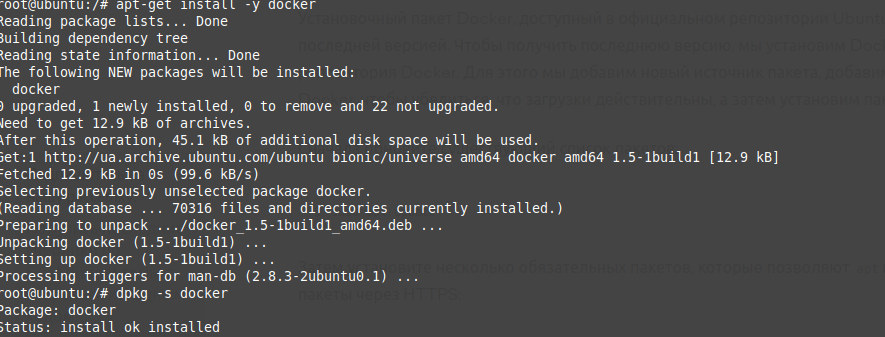


Работа с Docker в Ubuntu

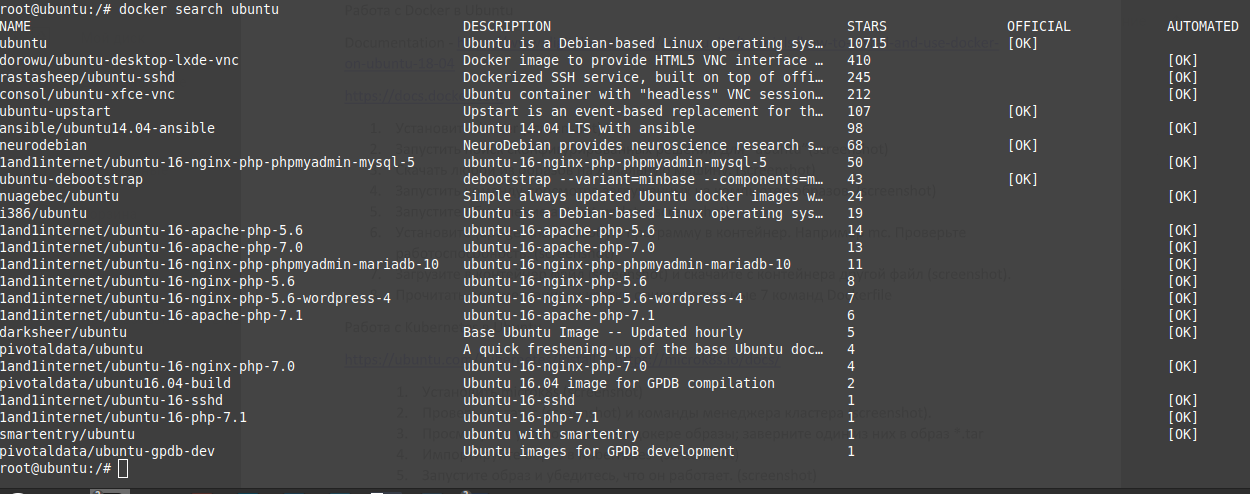
Documentation - <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04>

[https://docs.docker.com](https://docs.docker.com/)

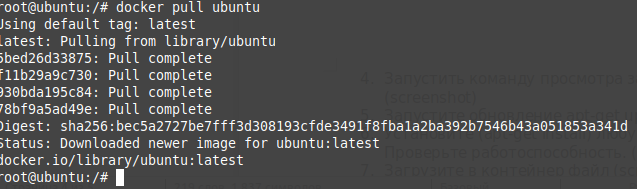
1. Установить docker (screenshot)



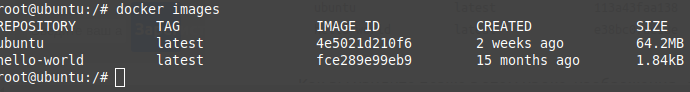
1. Запустить поиск сконфигурированных решений для “ubuntu”(screenshot)



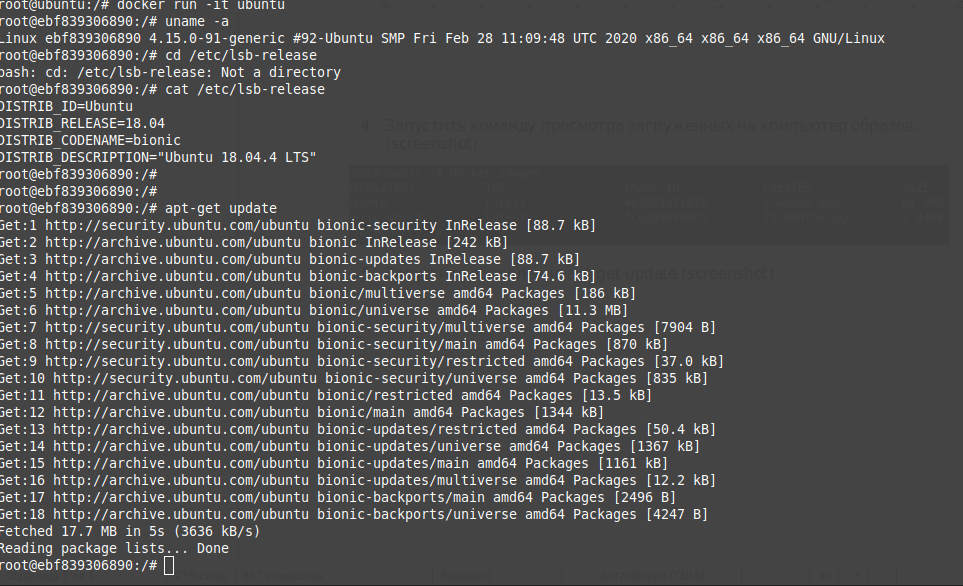
1. Скачать любой из образов на локальную машину. (screenshot)



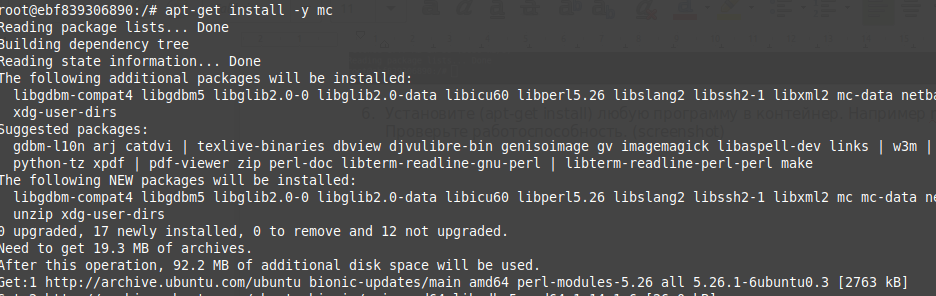
1. Запустить команду просмотра загруженных на компьютер образов. (screenshot)

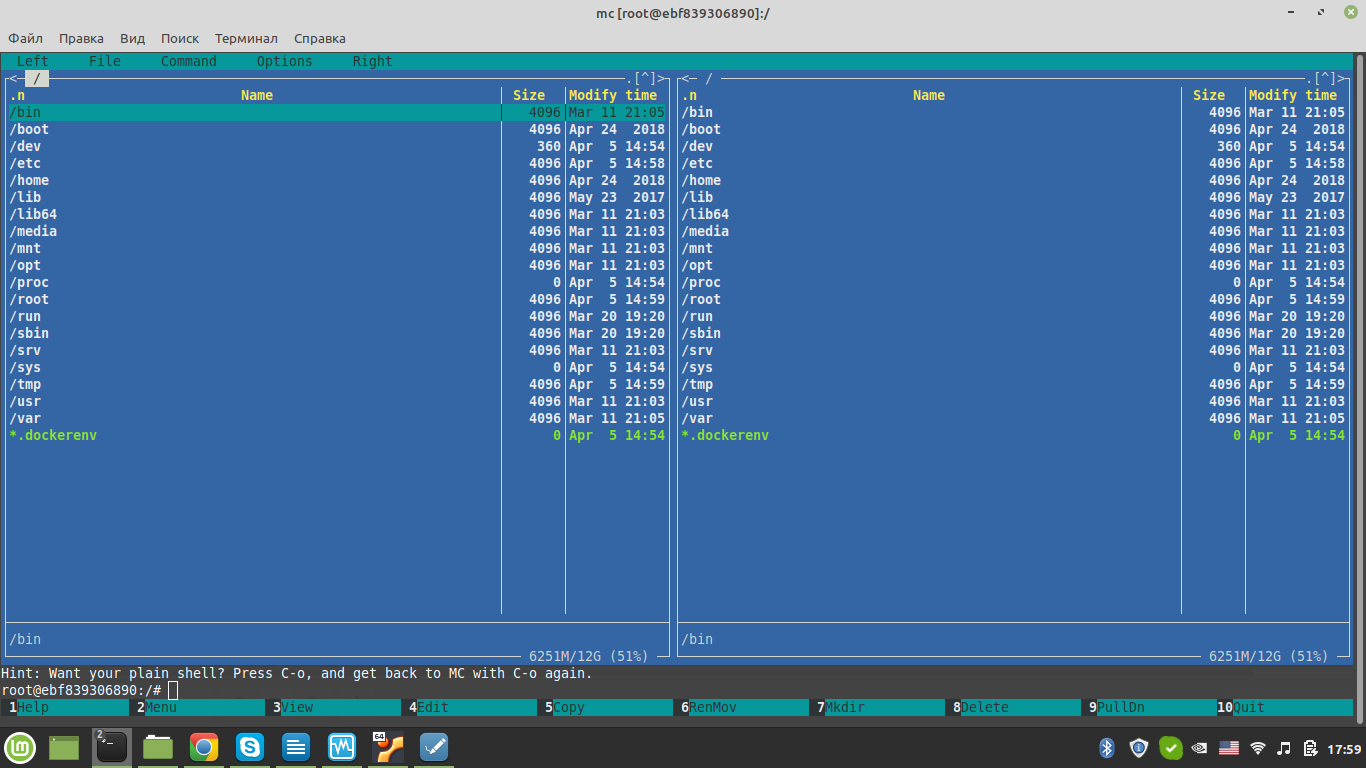


1. Запустите обновление apt-get update (screenshot)

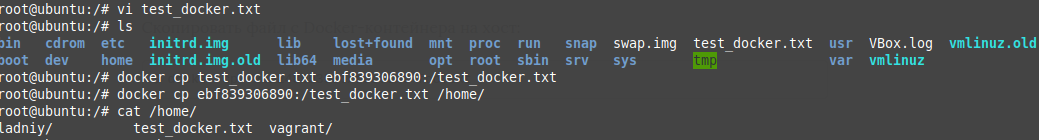


1. Установите (apt-get install) любую программу в контейнер. Например mc. Проверьте работоспособность. (screenshot)





1. Загрузите в контейнер файл (screenshot) и скачайте с контейнера другой файл (screenshot).



1. Прочитать документацию и кратко описать основные 7 команд Dockerfile

Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build users can create an automated build that executes several command-line instructions in succession.

**FROM** The FROM keyword tells Docker to use a base image that matches the provided name and tag when building the image. The basic image, in addition, is also called the parent image.

**LABEL** The LABEL statement (label) allows you to add metadata to the image. In the case of the file under consideration now, it includes the contact information of the creator of the image. Declaring labels does not slow down the image assembly process or increase its size. They only contain useful information about the Docker image, so it is recommended that they be included in the file.

**ENV** The ENV instruction allows you to set constant environment variables that will be available in the container during its execution. In the previous example, after creating the container, you can use the ADMIN variable.

**RUN** The RUN instruction allows you to create a layer during image build. After its execution, a new layer is added to the image, its state is fixed. The RUN instruction is often used to install additional packages in images. In the previous example, the RUN apk update && apk upgrade statement tells Docker that the system needs to update packages from the base image. Following these two commands is the && apk add bash command, indicating that bash needs to be installed in the image.

**COPY** The COPY instruction is presented in our file like this: COPY. ./app. She tells Docker that she needs to take files and folders from the local context of the assembly and add them to the current working directory of the image. If the destination directory does not exist, this instruction will create it.

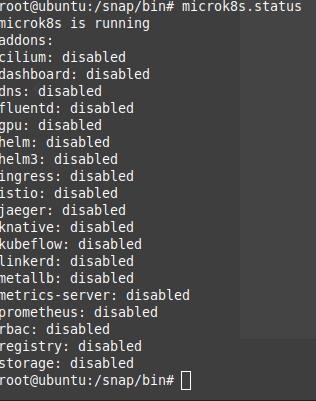
**ADD** The ADD instruction allows you to solve the same problems as COPY, but a couple more use cases are associated with it. So, using this instruction, you can add files downloaded from remote sources to the container, as well as unpack local .tar files.

**CMD** The CMD instruction provides Docker with a command to execute when the container starts. The results of this command are not added to the image during its assembly. In our example, this command launches the my\_script.py script at runtime.

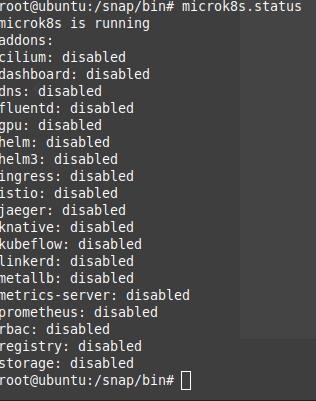
Работа с Kubernetes в Ubuntu

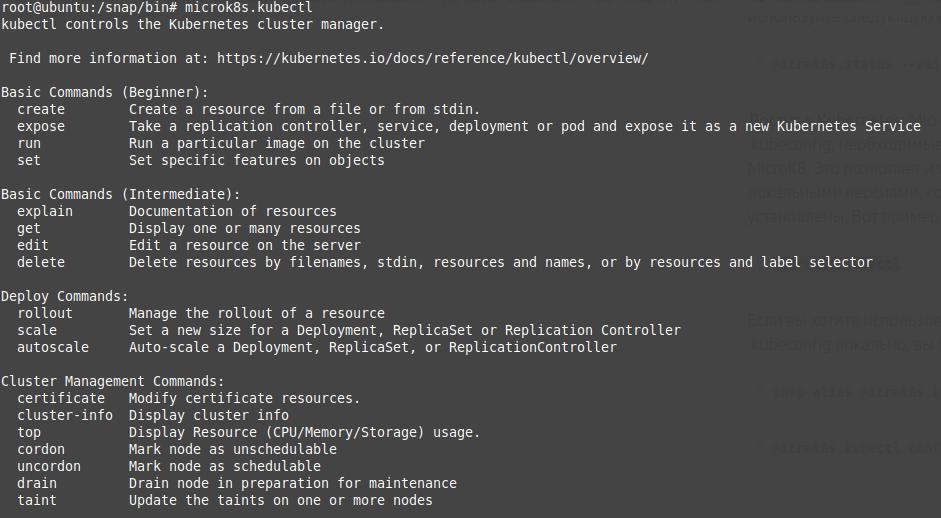
<https://ubuntu.com/kubernetes/install> ; <https://microk8s.io/docs/>

1. Установить microk8s (screenshot)

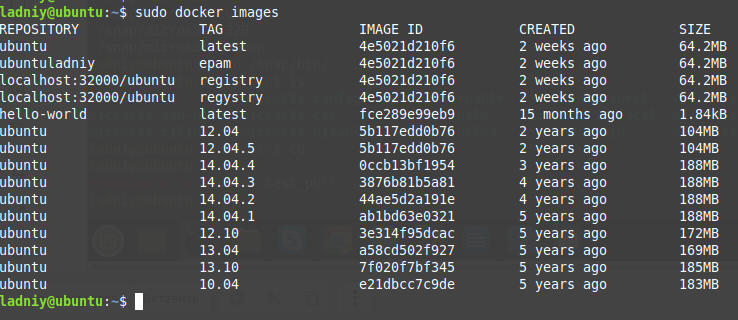


1. Проверьте статус (screenshot) и команды менеджера кластера (screenshot).





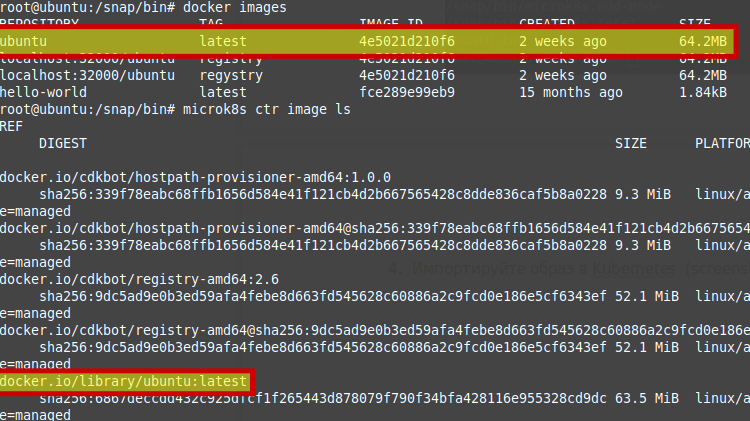
1. Просмотрите установленные в докере образы; заверните один из них в образ \*.tar





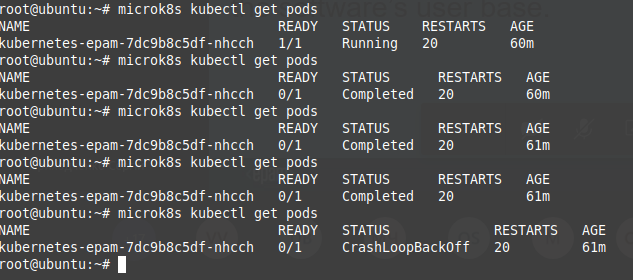


1. Импортируйте образ в Kubernetes (screenshot)



1. Запустите образ и убедитесь, что он работает. (screenshot)





образ стартует, рассыпается и заново стартует. Если можно я сдам ДЗ в таком виде чтобы успеть по срокам и завтра еще раз с нуля всё пересоздам. Спасибо.