МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

Институт информатики и кибернетики

Дисциплина «Информационная безопасность распределенных автоматизированных систем»

Отчет по лабораторной работе №3

«VM and RAID1»

Выполнил:

студент группы 6412-100503D

Шепляков Д.В.

Проверил:

Авдеев Е.В.

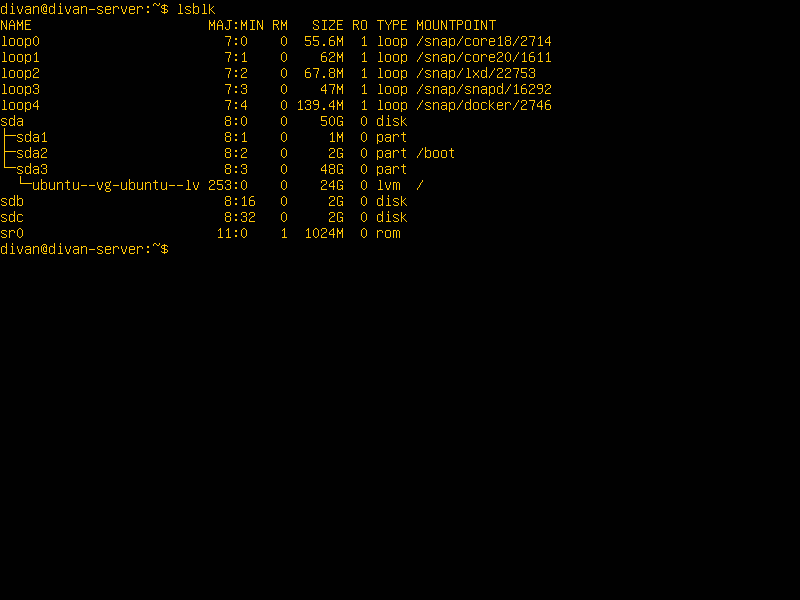
Самара 2023

**Ход работы**

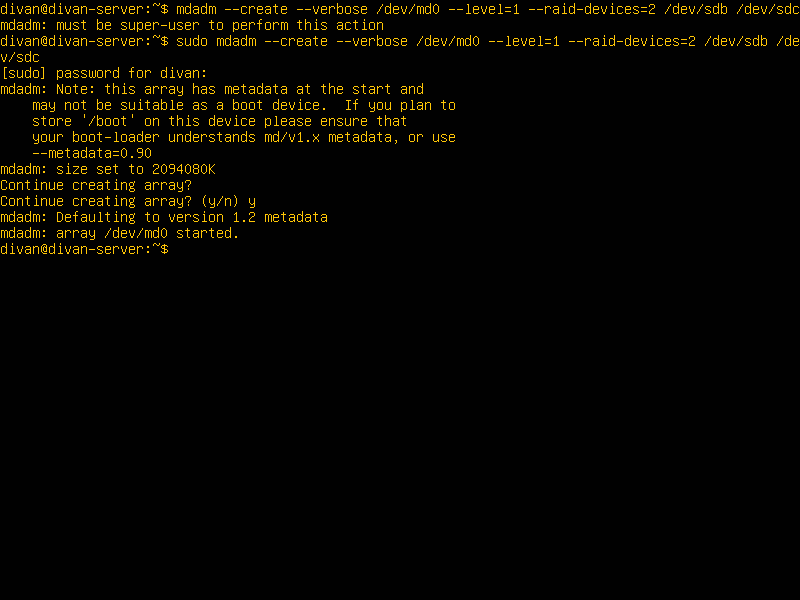
При создании новой виртуальной машины указываем 4 ядра и 4096 Мб оперативной памяти, а также жесткий диск размером 50 Гб.

RAID1

1. Смотрим идентификаторы двух нужных нам дисков, в данном случае были подключены два диска по 2 Гб.

Нужные диски - “sdb” и “sdc”.

2. Создаем массив, указывая название нового устройства и названия исходных дисков, а также уровень массива.



3. Создаем файловую систему на рейде.

sudo mkfs.ext4 -F /dev/md0

Создаем точку для монтирования новой файловой системы.

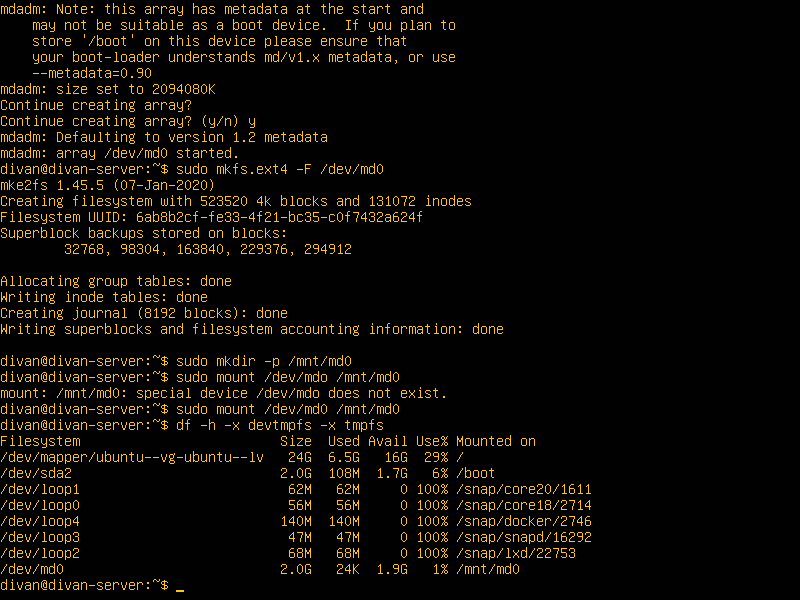
sudo mkdir -p /mnt/md0

Монтируем.

sudo mount /dev/md0 /mnt/md0

Проверяем, доступно ли новое файловое пространство.

df -h -x devtmpfs -x tmpfs



4. Чтобы при каждом запуске системы рейд собирался автоматически, нужно внести изменения в конфигурационный файл «/etc/mdadm/mdadm.conf». Затем, нужно обновить «initramfs» и добавить новую строку в файл «/etc/fstab», чтобы рейд автоматически монтировался при запуске системы.

sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf

sudo update-initramfs -u

echo '/dev/md0 /mnt/md0 ext4 defaults,nofail,discard 0 0' | sudo tee -a /etc/fstab



Minikube

1. При установке ОС выбираем предустановку Docker. Чтобы не использовать его от имени суперпользователя, исполняем следующие действия:

Создаем группу «docker».

sudo groupadd docker

Добавляем пользователя в эту группу.

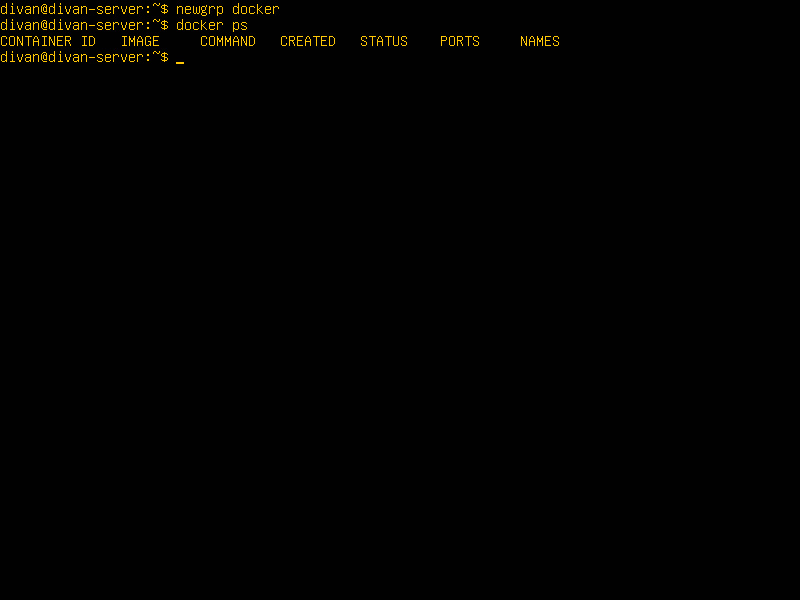
sudo usermod -aG docker $USER

Перезаходим в виртуальную машину и активируем изменения.

newgrp docker

Проверяем.

docker ps



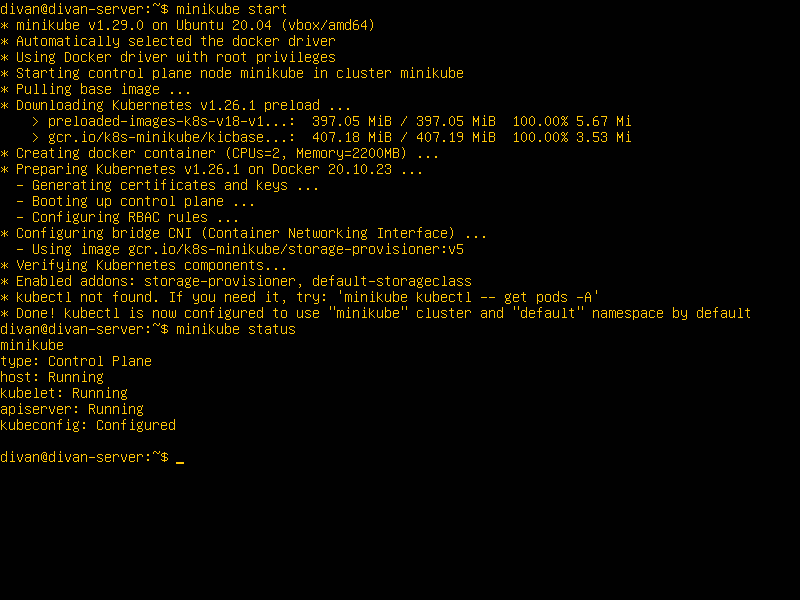
2. Устанавливаем Minikube.

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube



Запускаем. Minikube должен выбрать Docker автоматически.



Устанавливаем kubectl.

sudo apt-get update && sudo apt-get install -y apt-transport-https

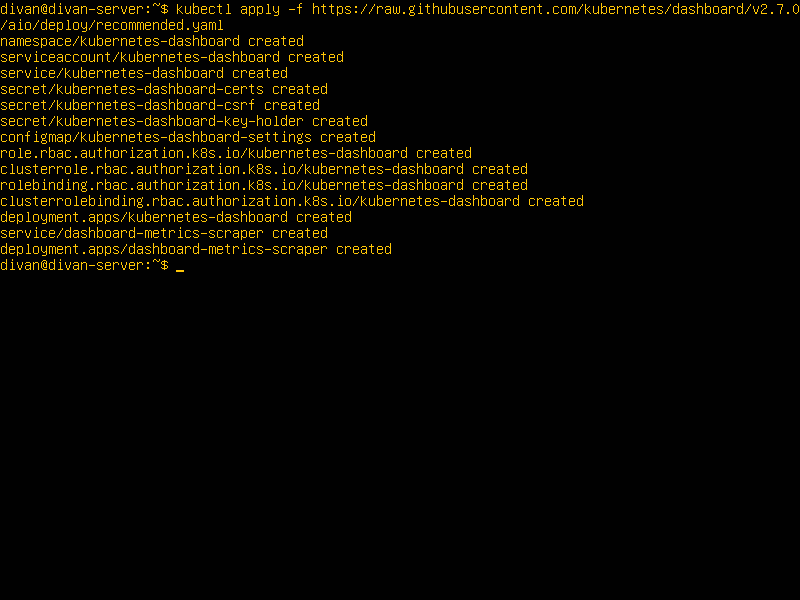
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update

sudo apt-get install -y kubectl

kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml



Создаем файл «sa-dash.yaml» со следующим содержимым,

apiVersion: v1

kind: ServiceAccount

metadata:

name: admin-user

namespace: kubernetes-dashboard

---

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: admin-user

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: cluster-admin

subjects:

- kind: ServiceAccount

name: admin-user

namespace: kubernetes-dashboard

и выполняем последнюю команду, но в качестве параметра передаем наш файл.

Создаем токен, который потребуется позже, дабы подключиться к серверу.

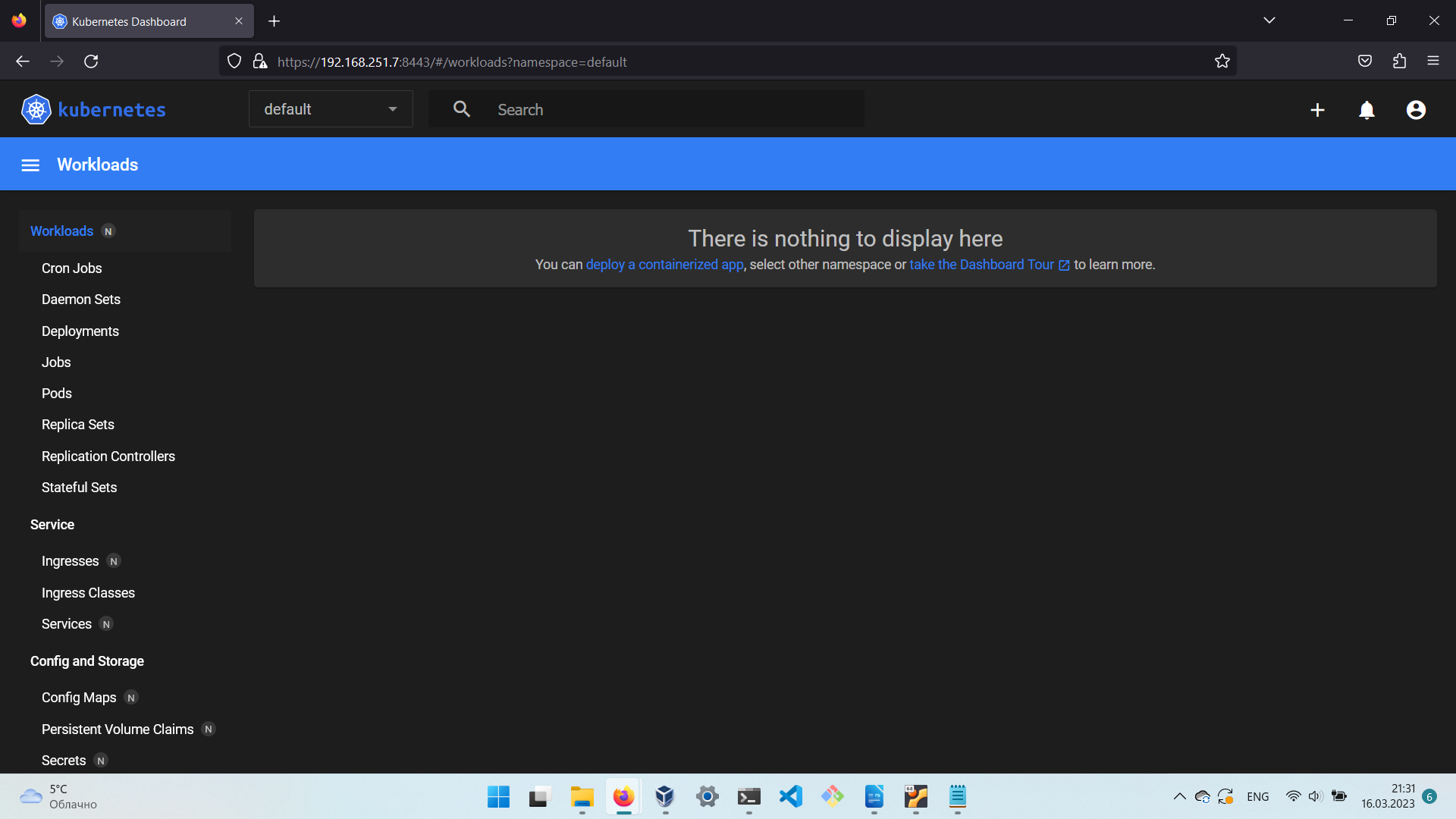
kubectl -n kubernetes-dashboard create token admin-user

Запускаем пару окон ssh, в которых прописываем следующие команды, запуская, таким образом сервер.

kubectl proxy --address 0.0.0.0 --accept-hosts '.\*'

kubectl port-forward -n kubernetes-dashboard service/kubernetes-dashboard 8443:443 --address 0.0.0.0

Теперь можно зайти в браузер и по адресу https://<ip виртуальной машины>:8443 ввести сгенерированный ранее токен, чтобы войти в систему.



Для примера, задеплоим приложение «hello-minikube».

