Практическое задание №10  
Часть 1. DevOps.

Разработать веб-приложение и настроить автоматизированный конвейер для его деплоя. Приложение будет развернуто на виртуальной машине с использованием Docker, Ansible и GitHub Actions.

1. Создание веб-приложения. Разработайте простое веб-приложение на языке по вашему выбору (например, Python/Flask, Node.js, Ruby on Rails, Golang). Приложение может быть одним из тех, что писали ранее, может быть совершенно новым.

2. Создание Docker-контейнера. Напишите Dockerfile для вашего веб-приложения и убедитесь, что Dockerfile правильно копирует исходный код, устанавливает зависимости и запускает приложение. Важно: Dockerfile должен быть оптимизирован для минимального размера образа.

3. Настройка Ansible. Создайте Ansible playbook для развертывания Docker-контейнера с вашим приложением на удаленном сервере. Убедитесь, что playbook устанавливает Docker на сервере, если он еще не установлен, и запускает контейнер с вашим приложением.

4. Настройка GitHub Actions. Создайте репозиторий на GitHub и загрузите туда ваше приложение, Dockerfile и Ansible playbook. Настройте GitHub Actions workflow, который будет автоматически запускаться при каждом пуше в репозиторий. Workflow должен включать сборку Docker-образа, загрузку образа в Docker Hub (или другой регистр контейнеров), выполнение Ansible playbook для развертывания нового образа на удаленном сервере.

1. mkdir webapp # Создание каталога для веб-приложения

cd webapp # Переход в созданную папку

nano app.py # Создание главного Python-файла с Flask-приложением

nano index.html #Создание html файла

python3 -m venv venv # Создание виртуального окружения

source venv/bin/activate # Активация виртуального окружения

pip install flask # Установка Flask внутри виртуального окружения

python3 app.py # Запуск веб-приложения

2.sudo apt install docker.io -y # Установка Docker

nano Dockerfile # Создание Dockerfile  
docker login # Вход в Docker Hub

docker login -u norjonoy # Вход с указанием пользователя  
docker build -t norjonoy/webapp . # Сборка Docker-образа

docker images # Просмотр списка доступных образов  
sudo docker push norjonoy/webapp # Загрузка контейнера в Docker Hub

docker push norjonoy/webapp # Повторная отправка образа  
ansible-playbook -i inventory deploy.yml # Развертывание через Ansible   
  
3. sudo apt install ansible -y # Установка Ansible  
nano deploy.yml # Создание Ansible Playbook  
nano inventory #Cоздание inventory file  
ansible-playbook -i inventory deploy.yml # Запуск финального развертывания  
  
4.  
git remote add origin <https://github.com/tayirova/final_assignment.git>  
git branch -m main # Создание главной ветки main 203 git add . # Добавление всех файлов в индекс  
git push -u origin main # Финальная отправка изменений в репозиторий

Часть 2. СУБД Oracle.

Надо будет создать БД, создать в ней пользователей с различными привилегиями, создать таблицы с данными, также создать скрипты для бэкапа используя как Data Pump, так и Rman.

Предполагается, что у вас осталась установленная ранее СУБД Oracle. Если нет - установите СУБД (версия 19c) на виртуальную машину любым удобным для вас способом.

1. Создайте новую БД с помощью DBCA. Постарайтесь задокументировать этот процесс и описать параметры, которые встречаются в ходе установки.

2. Создайте пользователей с различными уровнями привилегий (администратор базы данных, пользователь, ограниченный пользователь (только чтение данных)). Каждому пользователю назначьте отдельное табличное пространство.

3. Для каждого пользователя создайте таблицы и наполните их данными.   
4. Создайте скрипт для для экспорта одной любой схемы с помощью инструмента data pump. Опишите также процесс импорта этой схемы.  
5. Попробуйте создать rman скрипт для бэкапа всей базы данных. Осуществите запуск rman-

скрипта из cron-а. Опишите скрипт, процесс запуска, покажите логи выполнения скрипта.

1. dbcaA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**2.Создание пользователей и таблиц**

1. Создание администратора базы данных (admin\_user):

CREATE USER admin\_user IDENTIFIED BY 123123;

GRANT DBA TO admin\_user;

ALTER USER admin\_user DEFAULT TABLESPACE users QUOTA UNLIMITED ON users;

2. Создание обычного пользователя (regular\_user):

CREATE USER regular\_user IDENTIFIED BY UserPass123;

GRANT CONNECT, RESOURCE TO regular\_user;

ALTER USER regular\_user DEFAULT TABLESPACE users QUOTA 50M ON users;

3. Создание пользователя с доступом только для чтения (readonly\_user):

CREATE USER readonly\_user IDENTIFIED BY 123123;

GRANT CONNECT TO readonly\_user;

GRANT SELECT ANY TABLE TO readonly\_user;

ALTER USER readonly\_user DEFAULT TABLESPACE users QUOTA 10M ON users;

**3. Создание таблиц и добавление данных**

Для regular\_user:

ALTER SESSION SET CURRENT\_SCHEMA = regular\_user;

CREATE TABLE table\_for\_user (

id NUMBER PRIMARY KEY,

name VARCHAR2(2));

INSERT INTO table\_for\_user VALUES (1, 'a1');

INSERT INTO table\_for\_user VALUES (2, 'a2');

INSERT INTO table\_for\_user VALUES (3, 'a3');

INSERT INTO table\_for\_user VALUES (4, 'a4');

Для admin\_user:

ALTER SESSION SET CURRENT\_SCHEMA = admin\_user;

CREATE TABLE table\_of\_admin (

id NUMBER PRIMARY KEY,

name VARCHAR2(20)

);

INSERT INTO table\_of\_admin VALUES (1, 'admin1');

Доступ readonly\_user к таблицам:

GRANT SELECT ON regular\_user.table\_for\_user TO readonly\_user;

GRANT SELECT ON admin\_user.table\_of\_admin TO readonly\_user;

Экспорт схемы regular\_user:

**1. Создание скрипта экспорта:**

expdp system/AdminPass123 schemas=regular\_user \

directory=DATA\_PUMP\_DIR \

dumpfile=regular\_user\_export.dmp \

logfile=regular\_user\_export.log

**2. Импорт схемы:**

impdp admin\_user/password schemas=regular\_user \

directory=DATA\_PUMP\_DIR \

dumpfile=regular\_user\_export.dmp \

logfile=regular\_user\_import.log \

table\_exists\_action=replace

**5. Бэкап базы данных с помощью RMAN**

1.Создание скрипта RMAN:

RUN {

ALLOCATE CHANNEL ch1 DEVICE TYPE DISK;

BACKUP DATABASE FORMAT '/u01/app/oracle/backup/ORCL19C\_%U.bak';

RELEASE CHANNEL ch1;}

2.Запуск скрипта:

rman target / @/home/oracle/full\_backup.rman

Настройка cron для автоматизации:

1.

crontab -e

0 2 \* \* \* /home/oracle/full\_backup.sh

Скрипт full\_backup.sh:

#!/bin/bash

rman target / @/home/oracle/full\_backup.rman

chmod +x /home/oracle/full\_backup.sh