Модуль С

Автоматизация развертывания приложений, по методологии DevOps

Используемые технологии:

* Ansible, Ansible-galaxy;
* Docker, Docker-compose;
* Web-server Nginx;
* Базовая инициализация ОС;
* Балансировка нагрузки;
* Работа с основными компонентами ОС Debian на базе Linux;
* SQL, MySQL
* Python

В данном сценарии кандидаты получают доступ к базовой инфраструктуре, а также описание состояния системы, которого необходимо достичь. На основе данной информации кандидатам необходимо решить задачу автоматизации

развертывания приложения и всех его основных компонентов. Для тестирования своего решения кандидатам предоставляется доступ к изолированной группе ресурсов, которая содержит базовые экземпляры платформ для размещения приложения.

ПРОВЕРКА ТЕСТОВЫХ ЗАДАНИЙ

Проверка заданий будет происходить следующим образом. Всем участникам необходимо скопировать, написанные ими:

1. Плейбуки Ansible

2. Конфигурационный файл Nginx

В директорию /root/check. Проверка конфигурационных файлов будет выполняться в ручном режиме комиссией, проверяющей задания. В случае если какую-либо задачу не получилось описать в конфигурационном файле, то за её ручное выполнение участник ¼ балла, чем за полностью автоматизированную задачу. Стоимость заданий приведена в конце данного файла.

Часть 1

Подготовка рабочего окружения

**Вы как DevOps инженер работаете на компьютере c названием – DevOps.**

**Компьютер Windows User НЕ УЧАВСТВУЕТ в настройке, он используется для проверки корректности выполнения задания участником.**

**Компьютер Router(NAT) не используется для настройки, доступ к нему для участника запрещен!!!**

**Логин/пароль на все компьютеры root/toor**

**ЧАСТЬ 1 – Подготовка**

1. Настройте IP адреса в соответствии с топологией сети.

2. Настройте ssh для подключения root пользователя на 22 порт

3. Установить на компьютер DevOps пакет Ansible.

4. Сделайте инвентарный файл со следующими группами:

* Все сервера
* Сервера Приложений
* Сервер базы данных

5. Используя Ansible и коллекцию galaxy произведите следующие настройки:

5.1 Установите на все компьютеры следующее программное обеспечение:

* Curl
* Vim
* Procs
* Wget
* git

6. Создайте пользователей на всех серверах:

* Administrator / P@ssw0rd
* Moderator / didSomeoneSayModerator?!

6.1 На сервере Database создайте пользователя dbAdmin / ilovedb

7. Отключите SElinux на всех серверах

**ЧАСТЬ 2 – Развертывание приложения**

**Репозиторий с приложением и инструкцией располагается по следующему адресу:** [**https://github.com/Hramoff/prof.DevOps.git**](https://github.com/Hramoff/prof.DevOps.git)

1. На сервере “WEB” установите Nginx и настройте конфигурационный файл:

1.1Файл с конфигом должен располагаться в директории sites-available;

1.2 Настройте балансировку нагрузки:

* Используйте Метод наименьшего числа соединений
* На сервер 10.10.10.12 установите веса со значением 5;
* На сервер 10.10.10.10 установите веса со значением 5;
* На сервер 10.10.10.11 установите веса со значением 3;

1.3 Установите на сервер 10.10.10.12, 10.10.10.11 количество неудачных попыток, равных 3;

1.4 Установите максимально число активных соединений на сервер 10.10.10.11 в количестве 100;

1.5 Логи с ошибками должны сохраняться в директорию /root/log/ с названием nginx\_error\_log;

1.6 Логи с подключениями должны сохраняться в директорию /root/log/ с названием nginx\_access\_log;

1.7 Сайт должен обслуживать доменное имя finitastory.com;

1.8 Сайт должен быть доступен по доменному имени;

1.9Сервер прослушивает 80 порт;

1.10Все сервера в балансировке прослушивают порт 8000;

2. Подготовьте Dockerfile:

2.1Используйте образ python3.9;

2.2 Используйте рабочую директорию /FS;

2.3 Скопируйте директорию с проектом в директорию /FS;

2.4 Установите зависимости;

2.5 Сделайте проброс портов – порт приложения на порт 8000;

2.6 Укажите инструкцию для запуска приложения;

3. Используя Ansible скопируйте данный Dockerfile на каждый Application server:

3.1 Используя коллекцию galaxy установите дополнение для Docker;

3.2 Произведите сборку Dockerfile в образ приложения, с названием – FinitaStory;

4. Подготовьте Docker-compose;

4.1 Используйте любую подходящую версию;

4.2 Имя сервиса – FinitaStory;

4.3 Пробросьте 8000:8000 порт;

4.4 Скопируйте docker-compose на каждый Application Server и запустите

5. Настройка базы данных:

* Используйте docker-compose:

5.1 Любая актуальная и рабочая версия;

5.2 Образ MySQL;

5.3 Имя сервиса - FINITA\_BD;

5.4 Порт – стандартный порт MySQL;

5.5 Используйте environment для настройки авторизации в БД. Логин “root”, пароль “P@ssw0rd” ;

5.6 Используйте Volume для проброса папки с Базой Данных;

5.7 Для управления базой данных установите adminer;

5.8 Каждый контейнер должен перезагружаться сам, после выхода его из строя;

5.9 Скопируйте с помощью Ansible docker-compose на сервер с базой данных;

5.10 Скопируйте бекап базы данных на сервер с базой данных;

5.11 Используя Ansible запустите контейнер с базой данных;

6. Проверка работоспособности приложения:

6.1 Произведите регистрацию нового пользователя;

6.2 Произведите вход нового пользователя:

Схема сети представлена в соответствии с рисунком 1.

