Конкурсное задание включает в себя следующие разделы:

1. Форма участия в конкурсе: 2

2. Общее время на выполнение задания: 3

3. Задание для конкурса 3

4. Модули задания и необходимое время 3

5. Критерии оценки. 10

6. Приложения к заданию. 11

1. ФОРМА УЧАСТИЯ В КОНКУРСЕ: ИНДИВИДУАЛЬНЫЙ КОНКУРС.

2. ОБЩЕЕ ВРЕМЯ НА ВЫПОЛНЕНИЕ ЗАДАНИЯ: 15Ч.

3. ЗАДАНИЕ ДЛЯ КОНКУРСА

Конкурсное задание состоит из трех независимых модулей, которые участники выполняют последовательно в конкурсные дни C1, C2 и С3. Каждый модуль оценивается в день его проведения с использованием автоматизированных средств проверки. Инфраструктурные детали реализации конкурсного задания являются секретными до момента начала выполнения задания.

4. МОДУЛИ ЗАДАНИЯ И НЕОБХОДИМОЕ ВРЕМЯ

Таблица 1.

|  |  |  |
| --- | --- | --- |
| Наименование модуля | Соревновательный день (С1, С2, С3) | Время на задание |
| Модуль А: «Пуско-наладка информационной инфраструктуры» | С1 | 5 |
| Модуль B: «Аудит состояния инфраструктуры и устранение неисправностей» | С2 | 5 |
| Модуль С: «Автоматизация развертывания приложения» | С3 | 5 |

# Модуль C: Автоматизация развертывания приложения.

В рамках задания по автоматизации развёртывания приложений от Вас требуется создание сценария настройки инфраструктуры с использованием программного обеспечения Ansible. Конечной целью является функционирование обозначенных сервисов при развёртывании вашего сценария на эталонном стенде.

**Вводная информация:**

Компания «SOVNET.SU» является лидером на рынке разработки прикладных программных интерфейсов для анализа данных автоматизированных систем управления технологическими процессами.

В 2022 году, с целью снижения капитальных затрат на содержание собственной информационной инфраструктуры, компания реализовывает пилотный проект по переходу на модель «инфраструктура как сервис» (IaaS) проводит отбор кандидатов для дальнейшей работы над данным проектом.

Для выполнения данного задания Вам потребуется аккаунт GitHub.

Исходные файлы размещены в Git-репозитории:

**https://github.com/andgabs/skill\_hmao**

В вашем распоряжении тестовый стенд состоящий из 5 виртуальных машин под управлением:

Debian: PJD-01, PJD-02 и DEP.

CentOS: PJC-01, PJD-03

Топология прилагается. (Приложение С Рисунок 1)

Первоначальный доступ к виртуальным машинам осуществляется по протоколу SSH через узел DEP, пользователь root, пароль toor.

Сетевые параметры были сконфигурированы предварительно Инженером нашей компании. Каждая машина имеет доступ в Интернет. Вы можете использовать текущую инфраструктуру для тестирования своего проекта.

Обратите внимание, что после того, как группа проверки склонирует ваш репозиторий на проверочный стенд, будет производиться только запуск **playbook.yaml** файла. Продумайте то, что потребуется для его выполнения.

После выполнения задания, вам требуется собрать все необходимые файлы, которые необходимы для выполнения задания и отправить playbook.yaml с зависимостями на свой GitHub аккаунт.

Создайте отдельный репозиторий с именем, hmao-$lastname (в качестве $lastname, используйте свою фамилию), тип репозитория - Public.

**Автоматизация базовой конфигурации виртуальных машиню**

Контролирующим хостом является DEP. На виртуальной машине предустановлены пакеты ansible, sshpass, git. Все узлы доступны по именам.

Вам необходимо реализовать Ansible playbook для автоматизации базовой конфигурации виртуальных машин. Ansible playbook должен выполнять следующие задачи конфигурации:

* Выполнить клонирование Git-репозитория в корневую директорию узла DEP.
* Обновить списки пакетов, выполнить установку lynx на DEP.
* Обновить списки пакетов, выполнить установку curl (на все кроме DEP).
* Активировать межсетевой экран Firewalld (при необходимости установить). Разрешить порты TCP 80,8080,1022.
* Изменить конфигурационный файл SSH:
  + Порт подключения должен быть 1022;
  + Разрешить авторизацию по публичному ключу.
* Создать пользователей для доступа к виртуальным машинам. Для каждого пользователя необходимо разместить публичный ключ.

*Исходя из соображений безопасности, файл со списком пользователей, паролей и публичными ключами расположенный в директории Users/privvars.yml Git-репозитория, зашифрован при помощи Ansible Vault. Используйте временный пароль gK2VEOxxEK9n для дешифрования файла.*

* Выполнить установку пакета docker.
* Выполнить установку пакета docker-compose.
* Создайте директорию /skillcloud-nginx, скопируйте файлы необходимые для сборки образов, а также index.html и nginx.conf файлы.
* Перезапустить SSH и Firewalld.
* Выполнить перезагрузку виртуальных машин.

*Развертывание отказоустойчивого веб-сервера на базе Nginx:*

* Выполните сборку образов на виртуальных машинах, используя dockerfile в директории skillcloud-nginx.
  + Задайте имена виртуальных машин в файле nginx.conf.
  + Задайте теги при сборке образов, site:site и balance:balance.
* Запустить docker-compose.yml файл для сборки контейнеров (при необходимости предусмотрите внесение изменений).
* Учтите, что в случае отказа одной из виртуальных машины, сайт должен быть доступен.

*Убедитесь, что балансировка работает корректно, при помощи остановки docker-compose на одной, двух или трех виртуальных машинах. В качестве инструмента проверки используйте curl.*

5. КРИТЕРИИ ОЦЕНКИ.

Таблица 2.

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Баллы | | |
| Судейские аспекты | Объективная оценка | Общая оценка |
| Пуско-наладка информационной инфраструктуры | 0 | 20 | 20 |
| Аудит состояния инфраструктуры и устранение неисправностей | 0 | 15 | 15 |
| Автоматизация развертывания приложения | 0 | 15 | 15 |
| Итого |  | 50 | 50 |

6. ПРИЛОЖЕНИЯ К ЗАДАНИЮ.

Приложение C

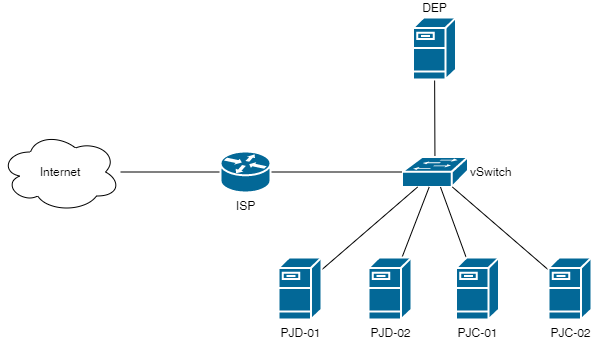


Рисунок 1 – Топология сети