1.5.2. Структура модулей конкурсного задания

Модуль Б. (*Настройка технических и программных средств информационно-коммуникационных систем*) (инвариант)

Время на выполнение модуля 10 часов

Задания:

Часть 1 (1 день - 5 часов)

1) Базовая настройка

a) Настройте имена устройств согласно топологии

1. Используйте полное доменное имя

b) Сконфигурируйте адреса устройств на свое усмотрение.

1. Для офиса HQ выделена сеть 192.168.11.0/28
2. Для офиса DT выделена сеть 192.168.33.0/28
3. Для туннелей между офисами выделена сеть 10.10.10.0/28

i. Туннель должен вмещать минимально возможное количество адресов

4. Данные сети необходимо разделить на подсети для каждого vlan.

i. VLAN111 должна вмещать не более 32 адресов

ii. VLAN222 должна вмещать не более 12 адресов

iii. VLAN333 должна вмещать не более 8 адресов

c) На всех устройства (кроме FW-DT) создайте пользователя userssh с паролем Pa$$w0rd

1. Пользователь userssh должен иметь возможность запуска утилиты sudo без дополнительной аутентификации.
2. На маршрутизаторах пользователь userssh должен обладать максимальными привилегиями.

2) Настройка коммутации

a) Настройте коммутаторы SW1-HQ, SW2-HQ, SW3-HQ.

1. Используйте Open vSwitch
2. Имя коммутатора должно совпадать с коротким именем устройства
    i. Используйте заглавные буквы
3. Передайте все физические порты коммутатору.
4. Обеспечьте включение портов, если это необходимо
5. Создайте на коммутаторах интерфейсы управления и именем MGMT.
    i. Для интерфейсов управления используйте vlan330.
6. Настройте протокол основного дерева
    i. Корнем дерева должен выступать SW1-HQ.

b) Настройте коммутатор SW-DT

1. В качестве коммутатора используйте соответствующий виртуальный коммутатор.

c) Для каждого офиса устройства должны находиться в соответствующих VLAN

1. Клиенты – vlan111,
2. Сервера – в vlan222,
3. Администраторы – в vlan333.

3) Настройте подключения маршрутизаторов к провайдеру

a) Для подключения R-DT к провайдеру необходимо использовать последний адрес из сети 172.16.4.0/28.

b) Для подключения R-HQ к провайдеру необходимо должен использовать последний адрес из сети 172.16.5.0/28.

c) Провайдер использует первый адрес из каждой сети

4) Настройка динамической трансляции адресов

a) Настройте на маршрутизаторах динамическую трансляцию адресов.

b) Все устройства во всех офисах должны иметь доступ к сети Интернет

5) Настройка протокола динамической конфигурации хостов

а) На R-HQ и R-DT настройте протокол динамической конфигурации хостов для клиентов (CLI)

1. Адрес сети – согласно топологии

i. Исключите адрес шлюза по умолчанию из диапазона выдаваемых адресов

2. Адрес шлюза по умолчанию – в соответствии с топологией

i. Шлюзом для сети HQ является маршрутизатор R-HQ

ii. Шлюзом для сети DT является межсетевой экран FW-DT

3. DNS-суффикс – au-team.irpo

4. Настройте клиентов на получение динамических адресов.

6) Между офисами DT и HQ необходимо сконфигурировать ip туннель

а) Используйте GRE

7) Настройте динамическую маршрутизацию OSPF

а) Между офисами DT и HQ

1. Маршрутизаторы должны быть защищены от вброса маршрутов с любых интерфейсов, кроме тех, на которых обмен маршрутами явно требуется.

2. Обеспечьте защиту протокола маршрутизации посредством парольной защиты

i. Используйте пароль Pa$$w0rd

b) Между R-DT и FW-DT

1. R-DT должен узнавать о сетях, подключенных к FW-DT по OSPF.

2. FW-DT должен получать маршрут по умолчанию и другие необходимые маршруты от R-DT через OSPF.

3. R-DT должен быть защищен от вброса маршрутов с любых интерфейсов, кроме тех, на которых обмен маршрутами явно требуется.

8) Настройка DNS для SRV1-HQ и SRV1-DT

а) Реализуйте основной DNS сервер компании на SRV1-HQ

1. Для всех устройств обоих офисов необходимо создать записи A и PTR.

2. Для всех сервисов предприятия необходимо создать записи CNAME.

3. Загрузка записей с SRV1-HQ должна быть разрешена только для SRV1-DT

b) Сконфигурируйте SRV1-DT, как резервный DNS сервер.

c) Все устройства должны быть настроены на использование обоих внутренних DNS серверов.

1. Для офиса HQ основным DNS сервером является SRV1-HQ

2. Для офиса DT основным DNS сервером является SRV1-DT

d) В качестве DNS сервера пересылки используйте любой общедоступный DNS сервер

9) Настройте синхронизацию времени между сетевыми устройствами по протоколу NTP.

a) В качестве сервера должен выступать SRV1-HQ

1. Используйте стратум 4

2. Используйте ntp2.vniiftri.ru в качестве внешнего сервера синхронизации времени

b) Все устройства должны синхронизировать своё время с SRV1-HQ.

1. Используйте chrony, где это возможно

c) Используйте на всех устройствах московский часовой пояс.

10) Реализация доменной инфраструктуры SAMBA AD

a) Сконфигурируйте основной доменный контроллер на SRV1-HQ

1. Используйте модуль BIND9_DLZ

2. Создайте 15 пользователей user1-user30 с паролем P@ssw0rd.

3. Пользователи user1-user5 должны входить в состав группы group1.

4. Пользователи user6-user10- должны входить в состав группы group2.

5. Пользователи user11-user15 должны входить в состав группы group3.

6. Создайте подразделения CLI и ADMIN

i. Поместите клиентов в подразделения в зависимости от их роли.

7. Клиентами домена являются ADMIN-DT, CLI-DT, ADMIN-HQ, CLI-HQ.

f) В качестве резервного контроллера домена используйте SRV1-DT.

1. Используйте модуль BIND9_DLZ

h) Реализуйте общую папку на SRV1-HQ

1. Используйте название SAMBA

2. Используйте расположение /opt/data

Часть 2 (2 день - 5 часов)

11) Управление доменом с помощью ADMC

a) Управление доменом с помощью ADMC осуществляться с ADMIN-HQ

b) Для подразделения CLI настройте политику изменения рабочего стола на картинку компании, а также запретите использование пользователям изменение сетевых настроек и изменение графических параметров рабочего стола.

c) Для подразделения ADMIN реализуйте подключение общей папки SAMBA с использованием доменных политик.

12. Настройка межсетевого экрана

1. Сервера и Администраторы офиса DT должны иметь доступ ко всем устройствам

2. Клиенты офиса DT должны иметь доступ только к серверам

3. Разрешите ICMP-запросы администраторами офиса DT на внутренние интерфейсы межсетевого экрана

13) Реализация бекапа общей папки на сервере SRV1-HQ с использованием systemctl

a) Бекап должен архивировать все данные в формат tar.gz и хранить в директории /var/bac/.

1. Архивация должна производиться благодаря юниту типа service с названием backup.

2. Сервис должен включатся автоматический при загрузке.

b) Время выполнение бекапа каждый день в 8 часов вечера.

1. Используйте юнит типа timer для выполнения.

2. Если устройство будет выключено, то архивация производится сразу после запуска.

14) Развертывание приложений в Docker на SRV2-DT

a) Создайте локальный Docker Registry.

b) Напишите Dockerfile для приложения web.

1. В качестве базового образа используйте nginx:alpine

2. Содержание index.html

```
<html>
  <body>
    <center><h1><b>WEB</b></h1></center>
  </body>
</html>
```

3. Соберите образ приложения web и загрузите его в ваш Registry.

i. Используйте номер версии 1.0 для вашего приложения

ii. Образ должен быть доступен для скачивания и дальнейшего запуска на локальной машине

c) Разверните Docker контейнер используя образ из локального Registry.

1. Имя контейнера web
2. Контейнер должно работать на порту 80

3. Обеспечьте запуск контейнера после перезагрузки компьютера

15) Настройка системы централизованного мониторинга

a) В качестве сервера системы централизованного мониторинга используйте SRV3-DT

b) В качестве системы централизованного мониторинга используйте Zabbix

1. В качестве сервера баз данных используйте PostgreSQL

i. Имя базы данных: zabbix

ii. Пользователь базы данных: zabbix

iii. Пароль пользователя базы данных: zabbixpwd

2. В качестве веб-сервера используйте Apache

c) Система централизованного мониторинга должна быть доступна для внутренних пользователей по адресу http://<IP адрес SRV3-DT>/zabbix

1. Администратором системы мониторинга должен быть пользователь Admin с паролем P@ssw0rd

2. Часовой пояс по умолчанию должен быть Europe/Moscow

d) Настройте узел системы централизованного мониторинга

1. В качестве узлов сети используйте устройства SRV1-DT, SRV2-DT, SRV3-DT, SRV-HQ.

2. Имя узла сети должно соответствовать полному имени устройства

16) Настройте веб-сервер nginx как обратный прокси-сервер на SRV1-DT

a) При обращении по доменному имени www.au.team, клиента должно перенаправлять на SRV2-DT на контейнер web

b) При обращении по доменному имени zabbix.au.team клиента должно перенаправлять на SRV3-DT на сервис Zabbix

c) Если необходимо, настройте сетевое оборудование для обеспечения работы требуемых сервисов.

18) Настройка узла управления Ansible

a) Настройте узел управления на базе ADMIN-DT

1. Используйте стандартную пакетную версию ansible.

b) Сконфигурируйте инвентарь

1. Инвентарь должен располагаться по пути /etc/ansible/inventory.

i. Настройте запуск данного инвентаря по умолчанию

2. Инвентарь должен содержать три группы устройств:

i. Networking (R-DT, R-HQ)

ii. Servers (SRV1-HQ, SRV1-DT, SRV2-DT, SRV3-DT)

iii. Clients (ADMIN-HQ, ADMIN-DT, CLI-HQ, CLI-DT)

3. Реализуйте доступ ко всем устройствам с учетом настроек SSH

i. Подключение осуществляется по пользователю sshuser

ii. Используйте корректный интерпретатор Python

iii. Отключите проверку SSH–ключа на хосте

c) Выполните тестовую команду “ping” средствами ansible

1. Убедитесь, что все устройства отвечают “pong” без предупреждающих сообщений

2. Убедитесь, что команды ansible выполняются от пользователя user без использования sudo

18) Настройка резервного копирования

a) На ADMIN-HQ развернуть Кибер Бекап 17 версии
b) Настроить организацию wsr
c) Настроить пользователя с правами администратора на сервере Кибер Бекап wsradmin с паролем Pa$$w0rd
d) Установить на CLI-HQ агент Кибер Бекап с функциями узла хранилища и подключить его при помощи токена
e) Подключить в качестве устройства хранения блочное устройство sdb в формате xfs (устройство должно быть примонтировано в папку /backups)
f) Создать план полного резервного копирования для сервера ADMIN-HQ
g) Выполнить полное резервное копирование ADMIN-HQ на узел хранения

Модуль Г. (*Обеспечение отказоустойчивости*) (вариант)

Время на выполнение модуля 5 часов

Задания:

1) Подготовка машины ControlVM

a) Общие указания:

1. Вся проверка выполнения задания будет осуществляться с машины ControlVM.

2. НЕ удаляйте инстанс ControlVM после завершения задания.

b) Создание и настройка инстанса ControlVM:

1. Создайте виртуальный инстанс с именем **ControlVM** и подключите его к сети интернет.

2. Установите следующие параметры для виртуальной машины:

i. **Тип виртуальной машины:** 2 vCPU, 4 ГБ RAM.

ii. **Размер диска:** 30 ГБ.

iii. **Тип диска:** SSD.

3. Отключите функции мониторинга и резервного копирования для данного инстанса.

4. В качестве операционной системы выберите **Альт Сервер 10**.

5. Настройте инстанс для разрешения внешних подключений по протоколу SSH.

6. Сохраните ключевую пару для доступа на вашем локальном ПК на рабочем столе с расширением **.pem**.

c) Настройка внешнего подключения к ControlVM:

1. Установите на локальный ПК клиент SSH **PuTTY**.

2. Создайте в PuTTY профиль с именем **cloud**.

3. Убедитесь в возможности установления соединения с инстансом ControlVM с локального ПК через PuTTY, без необходимости ввода дополнительных параметров.

4. Для подключения используйте имя пользователя **altlinux** и ранее сохранённую ключевую пару.

2) Подготовка облачной инфраструктуры:

a) Требования к виртуальным машинам:

1. Основные характеристики:

i. Операционная система: Альт p10 StarterKit/Альт Сервер p10-cloud

ii. Количество vCPU: 1.

iii. Объём оперативной памяти: 1024 МБ.

iv. Объём диска: 10 ГБ./30 ГБ

v. Тип диска: HDD.

b) Подготовьте сценарий автоматизации развёртывания облачной инфраструктуры:

1. Создание виртуальных машин и сетей:

i. Виртуальные машины и сети должны быть созданы строго в соответствии с предложенной топологией (см. Топология ниже).

ii. Имена виртуальных машин, сетей, подсетей и маршрутизаторов должны соответствовать именованиям, указанным в Топологии.

iii. Обеспечьте правильное подключение виртуальных машин к соответствующим сетям в рамках заданной топологии.

2. Безопасность и доступ:

i. Разрешите трафик по протоколу ICMP для всех виртуальных машин для диагностики сетевых подключений.

ii. Назначьте IP-адреса всем машинам. Сохраните внешние IP-адреса всех машин в файле /home/altlinux/white.ip на машине ControlVM.

iii. Настройте аутентификацию на основе открытых ключей для SSH.

iv. В случае предоставления внешнего доступа к виртуальным машинам, разрешите его только по протоколу SSH (публичный ключ, пароль отключён) и только с соответствующих IP-адресов.

3. Балансировка нагрузки:

i. Создайте балансировщик нагрузки и распределите трафик между серверами Web1 и Web2 (см. Топология).

ii. Ограничьте внешний доступ к балансировщику только протоколами HTTP и HTTPS. Все остальные порты должны быть закрыты.

iii. Балансировка нагрузки должна использовать алгоритм round robin.

iv. При обращении на внешний адрес балансировщика нагрузки должен выводиться ответ от приложения, работающего на внутренних серверах Web1 и Web2.

4. Настройка подключения:

i. Настройте машину WebAdm так, чтобы она могла подключаться по SSH с использованием пользователя altlinux и пароля «Pa$$w0rd» к серверам Web1 и Web2 с помощью VPN туннеля.

ii. Убедитесь, что машина ControlVM может подключаться к машине WebAdm используя ключевую пару пользователя altlinux по SSH через её глобальный IP-адрес.

3) Создание и настройка скрипта на машине ControlVM:

a) Создание скрипта автоматизации:

1. На машине ControlVM создайте скрипт cloudinit.sh.

2. В качестве рабочей директории используйте путь /home/altlinux/bin.

3. Скрипт должен использовать файл конфигурации /home/altlinux/bin/cloud.conf для настройки подключения к облачному провайдеру.

4. При проверке задания, эксперты могут изменить настройки только в файле cloud.conf. Другие файлы редактироваться не будут.

В файле cloud.conf допускается оставление комментариев, поясняющих назначение параметров.

b) Требования к скрипту:

1. Скрипт должен быть разработан таким образом, чтобы его можно было выполнять из любой директории без необходимости указания полного пути к исполняемому файлу.

2. Для выполнения задания используйте инструменты для автоматизации развёртывания инфраструктуры.

3. Скрипт должен включать механизмы проверки доступности созданных ресурсов и их правильного функционирования, включая доступность Web-серверов через балансировщик нагрузки.

4) Развертывание приложений в Docker

a) Общие требования:

Все действия выполняются на машине ControlVM. Выполнить развертывание Python-скрипта в Docker, настроить WordPress с использованием Docker Compose и развернуть базовый стек ELK для сбора и отображения логов.

b) Развертывание Python-скрипта в Docker

1. Напишите Python-скрипт в домашней директории пользователя py.py, который выполняет следующие задачи:

i. Проверяет наличие файла input.txt в рабочей директории root.

ii. Выводит сообщение с содержимым.

iii. Если файла input.txt нет, выводит сообщение об ошибке.

2. Создайте Dockerfile для Python-скрипта file-copy-python:

i. Используйте базовый образ python:3.8-alpine.

ii. Python-скрипт py.py должен выполняться внутри контейнера.

iii. Реализуйте копирование файла input.txt в контейнер (этот файл может содержать произвольный текст).

iv. Контейнер при запуске должен выводит содержимое файла input.txt, после чего завершать свою работу.

3. Сборка и запуск контейнера:

i. Соберите Docker-образ с именем file-copy-python.yml.

ii. Запустите контейнер и убедитесь, что содержимое файла выводится файл input.txt.

c) Развертывание WordPress с использованием Docker Compose

3.1. Создание файла wordpress.yml:

i. В домашней директории пользователя создайте файл wordpress.yml, описывающий стек контейнеров для WordPress и MySQL.

3.2. Конфигурация стека Docker Compose:

i. Определите два сервиса:

1) **wordpress:**
   - Используйте образ **wordpress:latest**.
   - Свяжите с сетью wordpress-network.
   - Прокиньте порт 80 для доступа к WordPress извне.
   - Настройте необходимые переменные окружения (WORDPRESS_DB_HOST, WORDPRESS_DB_USER, WORDPRESS_DB_PASSWORD, WORDPRESS_DB_NAME и тд.).

2) **mysql:**
   - Используйте образ **mysql:5.7**.
   - Свяжите с сетью wordpress-network.
   - Создайте volume для хранения данных базы данных.
   - Настройте необходимые переменные окружения (MYSQL_DATABASE,MYSQL_USER, MYSQL_PASSWORD, MYSQL_ROOT_PASSWORD и тд.).

3.3. Запуск стека:

i. Запустите Docker Compose с файлом wordpress.yml.

ii. Убедитесь, что WordPress доступен по указанному порту и готов к настройке.

4) Развертывание базового стека ELK

a) Создание файла elk.yml:

1. В домашней директории пользователя создайте файл elk.yml, описывающий стек контейнеров для Elasticsearch, Logstash и Kibana.

b) Конфигурация стека Docker Compose:

1. Определите три сервиса:

1) elasticsearch:

- Используйте образ **elasticsearch:7.10.1**.
- Прокиньте порт 9200 для доступа к Elasticsearch API.

2) logstash:

- Используйте образ **logstash:7.10.1**.
- Настройте Logstash для получения данных и отправки их в Elasticsearch.

3) kibana:

- Используйте образ **kibana:7.10.1**.
- Прокиньте порт 5601 для доступа к веб-интерфейсу Kibana.

c) Запуск стека:

1. Запустите Docker Compose с файлом elk.yml.
2. Убедитесь, что все сервисы работают и Kibana доступна по порту 5601.

5) Развёртывания облачных сервисов

a) На машине ControlVM создайте скрипт /home/altlinux/bin/DeployApp.sh.

1. Скрипт должен выполняться из любой директории без явного указания пути к исполняемому файлу.

b) Подготовьте web-приложение App1

1. Скачайте файлы app1.py и Dockerfile по адресу:

https://github.com/auteam-usr/moscow39

2. Соберите образ приложения и загрузите его в локальный репозиторий Docker на ваше усмотрение.

c) Команда DeployApp.sh должна запускать средства автоматизации для настройки операционных систем.

1. Разверните web-приложение App1 из репозитория Docker на виртуальных машинах Web1 и Web2.

2. Обеспечьте балансировку нагрузки между Web1 и Web2.

3. Обеспечьте внешний доступ к web-приложению по протоколу https.

4. При обращении по протоколу http должно выполняться автоматическое перенаправления на протокол https.

5. Обеспечивать доверие сертификату не требуется.

6) Завершение работы

a) По окончании рабочего времени освободите ресурсы облачного провайдера, использованные для автоматически созданных объектов.

b) Удалите все автоматически созданные виртуальные машины, сети, объекты и другие ресурсы.

c) Внимание: НЕ удаляйте **ControlVM** и ресурсы, необходимые для её функционирования.

d) Важно: если в облачной инфраструктуре останутся объекты, кроме тех, которые необходимы для работы **ControlVM** или создаются по умолчанию, проверка выполнения задания не будет проводиться.