

Создание и настройка скрипта на машине ControlVM

Задача:

3) Создание и настройка скрипта на машине ControlVM:

- а) 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 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.
- б) 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.

Вариант реализации:

ControlVM

Все файлы создаются в контексте каталога `/home/altlinux/bin`, если не сказано иное

- Удаляем все ранее созданные ресурсы средствами **Terraform** для дальнейшего развёртывания средствами одного файла `cloudinit.sh`:

```
terraform destroy
```

- Подтверждаем удаление ресурсов:

```
Plan: 0 to add, 0 to change, 45 to destroy.

Changes to Outputs:
- LoadBalancer = "192.168.15.53" -> null
- WEB1         = "192.168.15.176" -> null
- WEB2         = "192.168.15.161" -> null
- WebADM       = "192.168.15.99"  -> null

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes
```

- Результат:
 - Должны удалиться все созданные ранее ресурсы средствами **Terraform** и остаться одна **ControlVM**, а также ресурсы необходимые для её работы;

```
Destroy complete! Resources: 45 destroyed.
[altlinux@controlvm bin]$
```

- Создаём файл `cloudinit.sh`:
 - В котором необходимо описать всю последовательность действий по запуску всех необходимых компонентов для развёртывания всех необходимой инфраструктуры (**Terraform**) с последующей её настройкой (**Ansible**);

```
cloudinit.sh
```

- Помещаем в него следующее содержимое:
 - Перечисляя ранее запускаемые в ручном режиме команды для проверки создаваемых ресурсов;
 - Добавляем проверки, т.к. по требованиям задания скрипт должен включать механизмы проверки доступности созданных ресурсов

```
#!/bin/bash

cd /home/altlinux/bin
source cloud.conf
terraform init
terraform apply -auto-approve
terraform output > /home/altlinux/white.ip
ansible-playbook -i ansible/inventory ansible/wireguard_playbook.yml
ansible-playbook -i ansible/inventory ansible/ssh_playbook.yml

echo "Проверяем доступность созданных инстансов, для каждого инстанса статус должен быть ACTIVE:"
echo ""
openstack --insecure server list

echo "Проверяем доступность созданного балансировщика нагрузки:"
echo ""
openstack --insecure loadbalancer list

echo "Проверяем доступность Web-серверов через балансировщик нагрузки:"
echo ""
openstack --insecure loadbalancer member list HTTP
openstack --insecure loadbalancer member list HTTPS
```

- Задаём права на исполнение для данного файла:

```
chmod +x cloudinit.sh
```

- Проверяем значение переменной окружения **PATH**:

```
echo $PATH
```

- Если есть путь **/home/altlinux/bin**, тогда данный скрипт **cloudinit.sh** будет выполняться из любой директории

```
altlinux@controlvm bin]$ echo $PATH
/home/altlinux/.vscodeium-server/bin/13ae69686c4390a9aee7b71b44337eb488319f26/bin/remote-cli:/home/altlinux/bin:/bin:/usr/bin:/usr/local/bin:/usr/games
altlinux@controlvm bin]$
```

- Выполняем запуск скрипта:

```
./cloudinit.sh
```

- Результат:
 - Скрипт обрабатывает без ошибок;
 - Средствами **Terraform** в рамках решения Кибер Инфраструктура развёртываются необходимые ресурсы, в соответствие с топологией и требованиями задания;
 - Информация о публичных IP-адреса сохраняется в файл в соответствие с требованиями задания;
 - Средствами **Ansible** происходит конфигурация развёрнутых ресурсов в соответствие с требованиями задания;
 - Выводится проверка созданных ресурсов;

```
Проверяем доступность созданных инстансов, для каждого инстанса статус должен быть ACTIVE:
+-----+-----+-----+-----+-----+-----+
| ID | Name | Status | Networks | Image | Flavor |
+-----+-----+-----+-----+-----+-----+
| 96077e0-1b85-4ef9-9d1d-59e89592ab8b | WEB2 | ACTIVE | INTERNET=192.168.15.251, 192.168.200.22 | N/A (booted from volume) | minimal |
| 19eff340-ab95-43c5-b970-719e274ea905 | WEB1 | ACTIVE | INTERNET=192.168.15.114, 192.168.200.21 | N/A (booted from volume) | minimal |
| 2cb02e71-a157-49f5-ad02-8e96844d3314 | WebADM | ACTIVE | INTERNET=192.168.15.194, 192.168.200.20 | N/A (booted from volume) | minimal |
| 96306ddf-a488-4e35-8a8f-25696821ef5e | ControlVM | ACTIVE | INTERNET=192.168.200.10; cloud=192.168.100.128, 192.168.15.97 | N/A (booted from volume) | medium |
+-----+-----+-----+-----+-----+-----+
Проверяем доступность созданного балансировщика нагрузки:
+-----+-----+-----+-----+-----+-----+
| id | name | project_id | vip_address | provisioning_status | operating_status | provider |
+-----+-----+-----+-----+-----+-----+
| 16c9aa2a-0d69-45af-887a-cd3bffa1068c3 | Load Balancer | 4deb49c7b35546948e7e1a1a39941814 | 192.168.200.200 | ACTIVE | ONLINE | amphora |
+-----+-----+-----+-----+-----+-----+
Проверяем доступность Web-серверов через балансировщик нагрузки:
+-----+-----+-----+-----+-----+-----+
| id | name | project_id | provisioning_status | address | protocol_port | operating_status | weight |
+-----+-----+-----+-----+-----+-----+
| 7c24fa6a-c8fa-4794-9c0e-9368b4323f89 | WEB2 | 4deb49c7b35546948e7e1a1a39941814 | ACTIVE | 192.168.200.22 | 80 | ONLINE | 1 |
| 32f779af-b39c-4be2-ba02-46e5a26ec79e | WEB1 | 4deb49c7b35546948e7e1a1a39941814 | ACTIVE | 192.168.200.21 | 80 | ONLINE | 1 |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
| id | name | project_id | provisioning_status | address | protocol_port | operating_status | weight |
+-----+-----+-----+-----+-----+-----+
| 0fdada4eb-9826-405b-89d7-c9d4723604be | WEB2 | 4deb49c7b35546948e7e1a1a39941814 | ACTIVE | 192.168.200.22 | 80 | ONLINE | 1 |
| 07daea96-5379-413e-b600-7d86bfd0dce0 | WEB1 | 4deb49c7b35546948e7e1a1a39941814 | ACTIVE | 192.168.200.21 | 80 | ONLINE | 1 |
+-----+-----+-----+-----+-----+-----+
altlinux@controlvm bin]$
```

КИБЕРПРОТЕКТ Project1

+

Настроить виджеты

ЦП

5

Используется: 5
Свободно: 0

ОЗУ

7 Гиб

Используется: 7 Гиб
Свободно: 0 байт

Плавающие IP-адреса

5

Используется: 5
Свободно: 0

Объём хранилища

60 Гиб

Использование: 60 Гиб

Статус VM

Всего VM: 4

Работает

4

Выполняется

0

Остановлен

0

Ошибки: 0

Список VM с наибольшим потреблением ресурсов

ИЦП	ОЗУ	Хранилище
ControlVM		0,86
WEB1		0,26
WEB2		0,25
WebADM		0,25

Показать все

[Обратная связь](#)

[Подпишитесь](#)

[Вы используете гостевой доступ \(Вход\)](#)

[Сводка хранения данных](#)

Тема оформления сайта разработана

[conecti.me](#)