Подготовка облачной инфраструктуры

Задание:

- а) Требования к виртуальным машинам:
 - Основные характеристики:
 - i. Операционная система: Альт p10 StarterKit/Альт Сервер p10-cloud
 - іі. Количество vCPU: 1.
 - ііі. 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:
 - і. Виртуальные машины и сети должны быть созданы строго в соответствии с предложенной топологией (см. Топология ниже).
 - іі. Имена виртуальных машин, сетей, подсетей и маршрутизаторов должны соответствовать именованиям, указанным в Топологии.
 - ііі. Обеспечьте правильное подключение виртуальных машин к соответствующим сетям в рамках
 - заданной топологии.
- 2. Безопасность и доступ:
 - о i. Разрешите трафик по протоколу ICMP для всех виртуальных машин для диагностики сетевых подключений.
 - іі. Назначьте IP-адреса всем машинам. Сохраните внешние IP-адреса всех машин в файле
 - /home/altlinux/white.ip на машине ControlVM.
 - ііі. 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:
 - о і. Создайте балансировщик нагрузки и распределите трафик между серверами Web1 и Web2 (см. Топология).
 - іі. Ограничьте внешний доступ к балансировщику только протоколами НТТР и НТТРЅ. 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:
 - і. Настройте машину WebAdm так, чтобы она могла подключаться по SSH с использованием пользователя altlinux и пароля «P@ssw0rd» к серверам Web1 и Web2 с помощью VPN туннеля.
 - іі. Убедитесь, что машина ControlVM может подключаться к машине WebAdm используя ключевую пару пользователя altlinux по SSH через её глобальный IP-адрес.

Вариант реализации: ControlVM

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

 Создаём файл network.tf и описываем последовательно сетевую часть для развёртывания данной инфраструктуры с поэтапным запуском и наблюдением созданных ресурсов:

vim network.tf

- Помещаем следующее содержимое:
 - Создадим виртуальную сеть с именем INTERNET в соответствие с топологией;
 - В созданной виртуальной сети создадим одноимённую подсеть, т.к. все сетевые параметры задаются на подсеть в рамках сети (с произвольными параметрами);
 - В данном случае подсеть INTERNET имеет IP-адрес 192.168.200.0/24, DHCP-пул адресов для раздачи из данной подсети с 192.168.200.100 по 192.168.200.200, в качестве шлюза по умолчанию в данной подсети будет использоваться IP-адрес 192.168.200.1, в качестве DNS IP-адрес 77.88.8.8;
 - Также данную подсеть INTERNET необходимо добавить в ранее созданный виртуальный маршрутизатор с именем cloud (чтобы не создавать новый), который необходим для работы ControlVM и не должен быть удалён по завершению работы, для того чтобы для всех подключённых инстансов к данной подсети был доступ в сеть Интернет, т.к. на данном виртуальном маршрутизаторе ранее был выставлен чек-бокс для SNAT, и именно через него ControlVM и все остальные будут иметь доступ в сеть Интернет;



• Выполняем проверку синтаксиса и структуры файлов конфигурации Terraform

terraform validate

Результат:

[altlinux@controlvm bin]\$ terraform validate Success! The configuration is valid. [altlinux@controlvm bin]\$ ■

• Проверяем конфигурацию перед развёртыванием, смотрим план развёртывания ресурсов:

terraform plan

• Результат:

 По плану будет создано (добавлено) 3 ресурса: Сеть INTERNET, Подсеть INTERNET и добавление в виртуальный маршрутизатор cloud подсети INTERNET:

[altinum@controlym bin]\$ terraform plan
data.openstack_networking_router_vz.router: weading data.openstack_networking_router_v2.router: Read complete after 2s [id=ba5d5a05-064a-4556-bab4-020266cdba07]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols: + create
Terraform will perform the following actions:
<pre>w openside/pictual kig_pictual kig_ inclusion kig_ inclusion resource "openside/pictual kig_pictual kig_ inclusion kig_ inclusion + admin_state_up = true + admin_state_up = true + admin_state_up + admin_state_up + availability_zone_hints = (known after apply) + availability_zone_hints = (known after apply) + availability_zone_hints = (known after apply) + atd = (known after apply) + atd = (known after apply) + atd = (known after apply) + name = "INTERNET" + part_security_enabled = (known after apply) + region = (known after apply) + shared = (known after apply) + transt_id = (known after apply) + transt_id</pre>
+ segments (known after apply) }
<pre># openstack_networking_router_interface_v2.router_interface will be created resource "openstack_networking_router_interface_v2" "router_interface" { + force_destroy = false id = (known after apply) + port_id = (known after apply) + router_id = (known after apply) + router_id = (known after apply) + submet_id = (known after apply) }</pre>
<pre># openstack.networking_subnet_v2.subnet will be created * resource "openstack_networking_subnet_v2" "subnet" { + all_tags = (known after apply) + cidr = "192.168.200.0/24" * dns_nameservers = [* "77.88.8.8",</pre>
] enable_dhcp = true gateway_ip = "192.168.200.1" + id = (known after apply) + ipversion = 4 + ipve_address_mode = (known after apply) + ipve_ra_mode = (known after apply) + name = "INTERNET" + network_id = (known after apply) + no_gateway = false
<pre>- service_types = (known after apply) + tenant_id = (known after apply) - allocation_pool {</pre>
• Запускаем развёртывание данных ресурсов:
terraform apply
 Подтверждаем развёртывание введя уеs:
Plan: 3 to add, 0 to change, 0 to destroy.
Do you want to perform these actions? Terraform will perform the actions described above. Only 'yes' will be accepted to approve.
Enter a value: yes
• Результат:
Plan: 3 to add, 0 to change, 0 to destroy. Do you want to perform these actions? Terraform will perform the actions described above. Only 'yes' will be accepted to approve. Enter a value: yes
<pre>openstack_networking_network_v2.network: Creating openstack_networking_network_v2.network: Creating complete after 7s [id=59fc148b-b768-472e-8830-a534c7c0264e] openstack_networking_subnet_v2.subnet: Creating openstack_networking_router_interface_v2.router_interface: Creating openstack_networking_router_interface_v2.router_interface: Creating openstack_networking_router_interface_v2.router_interface: Still creating flos elapsed] openstack_networking_router_interface_v2.router_interface: Creating complete after 13s [id=6576962b-6c57-46e1-b843-e40ad371d71a] Apply_complete! Resources: 3 added, 0 changed, 0 destroyed. o[alLinnadecontrol/wm bin]s</pre>
 Проверяем наличие созданных ресурсов средствами openstack-cli: Сеть с подсетью:

	Name	Subnets			
59fc148b-b768-472e-8830-a534c7c0264e 88d8e8c2-3b17-425e-b9d8-f7f7b3b360aa c87b7204-0dd5-4679-a97a-8b96b34bb2d6	INTERNET public cloud	1d7e8d18 41a04224 2e112e29	-b0ec-497d-9a9c-b726064efc50 -4412-4617-9785-58bb9e3afd9a, ffa6a7f8- -7c2c-413d-b964-145bb0338305	3fc5-4d50-afda-0ff1759	
[altlinux@controlvm bin]\$ openstacki	nsecure subr	net list	——		
ID.	Name		Network	Subnet	
1d7e8d18-b0ec-497d-9a9c-b726064efc50	INTERNET		- 59fc148b-b768-472e-8830-a534c7c0264e	192.168.200.0/24	
2e112e29-7c2c-413d-b964-145bb0338305 41a04224-4412-4617-9785-58bb9e3afd9a	fip-service-subnet		c87b7204-0dd5-4679-a97a-8b96b34bb2d6 88d8e8c2-3b17-425e-b9d8-f7f7b3b360aa 88d8e8c2-3b17-425e-b9d8-f7f7b3b360aa	192.168.100.0/24 192.168.15.0/24 169.254.32.0/23	

Проверяем наличие созданных ресурсов средствами веб-интерфейса:
 Ость с подсетью:

K	иберпроте	кт				Projecti	~ 4 @
16	Вычисления 🔳	Сети			INTERNET		×
	🕐 Вергуальные машены	(±) Панся	٩		й уданить		
	Offpular	Mun +	Управление (Р-адресами	Twn +			
	A Cent	A doud	Bunosena	Виртуальная	Конфигурация сети		
	() VPN		Выхочени	Виртуальные	Pale	INTERNET	1
	Маршрупизаторы	A public	boweee.	<i>Warrecian</i>	Den	Виртуальная	
	Оповающие IP-адопса				Идентификатор сели	59fc148b-b768-472e-8830-a534c7c0264e	9
	 Дотни безопасности Воликовровцики напрупи Воликовровцики напрупи Stiti-клини 				Падсети 🕳		-
					Версан Р подсети	1Pol	
					COR	192.168.200.0/24	
					illence	192.168.200.7	
					Скравр ОНСР	Включено	
					Hynu IP superon	192.168.200.100 - 192.168.200.200	
					Ceperpa: DNS	77.88.8.8	

• Подсеть добавленная в существующий маршрутизатор (для доступа в сеть Интернет из данной подсети):

KI	ИБЕРПРОТЕ	кт	
∿•	Вычисления 💽	Маршрутизаторы	
0	😵 Виртуальные машины	Поиск	
	🖨 Тома	Имя †	Статус 🧅
	🖧 Сети		🥝 Запущена
	Маршрутизаторы		
	Плавающие IP-адреса		
	Руппы безопасности Балансиоовщики нагрузки		
	SSH-ключи		

КИБЕРПРОТЕКТ

14-	Вычисления 🔃	Вычы	Вычисления) Маршрутизаторы) Маршрутизатор							
	😚 Виртуальные машины	ИНТЕ	РФЕЙСЫ СТАТИЧЕСКИЕ МАРШРУТЫ							
	 Образи Нома Сети 		tot Q							
			IP-адрес +	Статус 🕴	Twn	Cen.				
	U VPN		192.158.15.124	🗿 Запушена	Вжешный шлоз	public				
	ដ Мәршрулизаторы		100 100 100 1							
	 Плавающие Р-адреса Пруппы безопасности 		192.188.100.1	🥥 запущена	внутренний интерфенс	cloud				
			192.168.200.1	🧔 Запушена	Внутренний интерфейс	NTERNET				
	🔒 Балансировщики нагрузки									
	COLUMN HIM IN									

- В файл network.tf добавляем следующий код:
 - Создаём порты в сети INTERNET в одноимённой подсети для каждого инстанса и балансировщика нагрузки
 - Обращаясь уже к ранее созданным ресурсам, а именно к идентификаторам id ресурсов с именемами network и subnet

```
# Создадим порт для инстанса WebADM
resource "openstack_networking_port_v2" "port_webadm" {
                 = "webadm"
  name
   network_id = openstack_networking_network_v2.network.id
  fixed_ip {
    subnet_id = openstack_networking_subnet_v2.subnet.id
     ip_address = "192.168.200.20"
  }
}
# Создадим порт для инстанса WEB1
resource "openstack_networking_port_v2" "port_web1" {
  name
                 = "web1"
   network_id = openstack_networking_network_v2.network.id
  fixed ip {
     subnet_id = openstack_networking_subnet_v2.subnet.id
      ip_address = "192.168.200.21"
  }
}
# Создадим порт для инстанса WEB2
resource "openstack_networking_port_v2" "port_web2" {
   name
                  = "web2"
  network_id = openstack_networking_network_v2.network.id
  fixed ip {
     subnet_id = openstack_networking_subnet_v2.subnet.id
      ip_address = "192.168.200.22"
  }
}
# Создадим порт для балансировщика нагрузки Load Balancer
resource "openstack_networking_port_v2" "port_loadbalancer" {
  name
                  = "Load Balancer"
  network_id = openstack_networking_network_v2.network.id
   fixed ip {
     subnet_id = openstack_networking_subnet_v2.subnet.id
     ip_address = "192.168.200.23"
  }
}
 • Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов:
terraform apply
        • Подтверждаем развёртывание введя уез:
  Plan: 4 to add, 0 to change, 0 to destroy.
 Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
    Enter a value: ves

    Результат:

   Enter a value: yes
openstack_networking_port_v2.port_webadm: Creating...
openstack_networking_port_v2.port_web2: Creating...
openstack_networking_port_v2.port_web1: Creating...
openstack_networking_port_v2.port_web1: Creating...
openstack_networking_port_v2.port_web2: Creation complete after 8s [id=ca24637d-8927-49cd-a63b-1c294a0d2f7d]
openstack_networking_port_v2.port_webadm: Creation complete after 8s [id=s24637d-8927-49cd-a63b-1c294a0d2f7d]
openstack_networking_port_v2.port_webadm: Creation complete after 8s [id=s24624637d-8927-49cd-a63b-1c294a0d2f7d]
openstack_networking_port_v2.port_webadm: Creation complete after 8s [id=s23762a-a45c-d478-4926-b143-56c57313a4d]
openstack_networking_port_v2.port_web1: Creation complete after 8s [id=s237648d2-7d9d-44c5-aa77-9152aaa21ed4]
Apply completel Resources: 4 added, 0 changed, 0 destroyed.
[altlinux@controlvm bin]$
  • Проверяем наличие созданных ресурсов средствами openstack-cli:
        • в веб-интерфейсе нет возможности посмотреть созданные порты;
```

100	Nane	WAC Address	Fixed IP Addresses	Statu
2071 e132.cc/22.4a/b-2712.4a/64/25.c4b/a77 300 1202.cc/22.4a/b-2712.4a/64/25.c4b/a77 300 1202.ch/2518-4713.64/b47-48/25/64/4C/c 301 1202.ch/2518-4713-64/b47-48/25-46/c2 301 1202.ch/2518-451-452-46-20-24/51-46/25 550 1267-54/64-1-543-452-64/25-64/517464 18865428-c578-451-b403-466973776743 19276451-427-464-1-5431-562-6517313404 503 74802-7404-452-64315-14420 503 74802-7404-452-64315-14420 503 74802-7404-452-64315-14420 503 74802-7404-452-64315-14420 503 74802-7404-452-64315-14420 512 7451-523-7442-6431-12234400417100 778 1822-725-734-745-653-6431-1449 512 7451-523-744-5431-1449 512 7451-543-5431-1449 512 7451-543-5431-1449 512 7451-543-5431-1449 512 7451-543-5431-1449 512 7451-543-543-545-543-54-543-544-5431-1449 512 7451-543-543-545-543-544-5431-1449 512 7451-543-543-543-545-543-544-544	webath, web1 Load Balancer web2	fa: 16: 3e: 51:84: c3 fa: 16: 3e: 80:4e: 23 fa: 16: 3e: 80:4e: 23 fa: 16: 3e: 69:75: 2c fa: 16: 3e: 69:75: 2c fa: 16: 3e: 61: c3; dc fa: 16: 3e: 61: c3; dc fa: 16: 3e: 101:e4; c9 fa: 16: 3e: 101:e4; c9 fa: 16: 3e: 07: 2a; b5 fa: 16: 3e: c7: 2a; b5 fa: 16: 3e: c27: 2a; b5 fa: 16: 3e: c27: 2a; b5 fa: 16: 3e: c27: 5a; fa: 16: 5a; fa: 16: 5a; fa:	Ip_address='192_168'100.126', subnet_it='2e112a29-7c2c-413d-b544-145b033800' ip_address='192_168'100.12', subnet_it='2e112a29-7c2c-413d-b544-145b033800' ip_address='192_168'20', subnet_it='16768d13 bbc-4704', subnet_it='17668d13 bbc-4704', subnet	ACTEV ACTEV ACTEV ACTEV ACTEV ACTEV ACTEV DOWN DOWN DOWN ACTEV

- В файл network.tf добавляем следующий код:
 - Создаём плавающие IP-адреса ("публичные") для каждого инстанса и балансировщика нагрузки;
 - Плавающие IP-адреса должны браться из "Публичной" сети указав её имя, в данном случае public;



- Имя "публичной" сети можно посмотреть через openstack-cli:
 - Та сеть, которую ранее не создавали в ручную для работы ControlVM (cloud) и средствами Terraform в соответствие с топологией (INTERNET);

ID	Name	Subnets
59fc148b-b768-472e-8830-a534c7c0264e	INTERNET	1d7e8d18-b0ec-497d-9a9c-b726064efc50
88d8e8c2-3b17-425e-b9d8-f7f7b3b360aa	public	41a04224-4412-4617-9785-58bb9e3afd9a, ffa6a7f8-3fc5-4d50-afda-0ff17591463d
c87b7204-0dd5-4679-a97a-8b96b34bb2d6	cloud	2e112e29-72c-413d-9a64-145bb0338305

Имя "публичной" сети можно посмотреть через веб-интерфейс:

KUBEPHPOTEKT

иртуальные машины	🛫 ФИЛЕ	-		
	Trees on the	поиск	Q	
ома		Имя †	Управление IP-адресами	Τν
ети		🔏 cloud	Включено	Bi
PN		NTERNET	Включено	B
Ларшрутизаторы		🐥 public	Включено	Φι
ілавающие іР-адреса руппы безопасности алансировщики нагрузки				
lл P a	аршрутизаторы авающие IP-адреса уппы безопасности лансировщики нагрузки Н-ключи	аршрутизаторы авающие IP-адреса уппы безопасности лансировщики нагрузки Н-ключи	аршрутизаторы Сробности авающие IP-адреса уппы безопасности лансировщики нагрузки	аршрутизаторы Включено вавающие IP-адреса улпы безопасности лансировщики нагрузки

Enter a value: yes
<pre>openstack_networking_floatingip_v2.floatingip_webadm: Creating openstack_networking_floatingip_v2.floatingip_web1: Creating openstack_networking_floatingip_v2.floatingip_web1: Creating openstack_networking_floatingip_v2.floatingip_web2: Still creating openstack_networking_floatingip_v2.floatingip_web2: Still creating openstack_networking_floatingip_v2.floatingip_web2: Still creating openstack_networking_floatingip_v2.floatingip_web1: Still creating openstack_networking_floatingip_v2.floatingip_web1: Still creating openstack_networking_floatingip_v2.floatingip_web1: Still creating [10s elapsed] openstack_networking_floatingip_v2.floatingip_web1: Still creating [10s elapsed] openstack_networking_floatingip_v2.floatingip_web1: Still creation [10s elapsed] openstack_networking_floatingip_v2.floatingip_web1: Creation complete after 11s [id=13336582-a17a-4de5-8e33-a0151662cb77] openstack_networking_floatingip_v2.floatingip_web1: Creation complete after 11s [id=750a2a96-6d20-4f15-8lac-fcb4ddb85192] openstack_networking_floatingip_v2.floatingip_web8: Creation complete after 11s [id=750a2a96-6d20-4f15-8lac-fcb4ddb85192] openstack_networking_floatingip_v2.floatingip_loadbalancer: Creation complete after 11s [id=750a2a96-6d20-4f15-8lac-fcb4ddb85192] openstack_networking_floatingip_v2.floatingip_loadbalancer: Creation complete after 11s [id=750a2a96-6d20-4f15-8lac-fcb4db8566027]</pre>
Apply complete! Resources: 4 added. 0 changed. 0 destroyed. [altlinux@controlvm bin]%

• Проверяем наличие созданных ресурсов средствами openstack-cli:

Плавающие IP-адреса (4 шт.):

stliv.edcontrolivi bir]s openstackInsecure floating ip list									
ID	Floating IP Address	Fixed IP Address	Port	Floating Network	Project				
13396582-a17a-4de5-8e33-a0151662cb/7 750a2a66-6405-4415-81ac-fc644d865192 92cb452c-a435-47e3-ad1e-c00fd29a73e cc559743-dd65-4cba-ed65-cd9b46a66027 dc648177-ac4c-4407-8391-15130960f207	192.168.15.161 192.168.15.99 192.168.15.97 192.168.15.53 192.168.15.176	None None 192,158,100,128 None None	None None 33967560-533e-4535-X089-882a25c36ecc None None	88d9e8c2-3b17-425e-b9d8-f7f7b3b360aa 88d9e8c2-3b17-425e-b9d8-f7f7b3b360aa 88d9e8c2-3b17-425e-b9d8-f7f7b3b360aa 88d9e8c2-3b17-425e-b9d8-f7f7b3b360aa 88d9e8c2-3b17-425e-b9d8-f7f7b3b360aa	40eb49:7b35546948e7e1a1a39941814 40eb49:7b35546948e7e1a1a39941814 40eb49:7b35546948e7e1a1a39941814 40eb49:7b35546948e7e1a1a39941814 40eb49:7b35546948e7e1a1a39941814				
Taltlinu@controlyn bin15									

Проверяем наличие созданных ресурсов средствами веб-интерфейса:
 Плавающие IP-адреса (4 шт.):

КИБЕРПРОТЕКТ

1∕⊷	Вычисления 🖪	Пла	ваюц	цие IP-адреса			
Ø	Виртуальные машины	Пон	ск	٩			
	О Тома		IP-a/	ipec 🔋	Статус	Сеть	Назначен
	🙏 Сети		®	192.168.15.161	🚫 Неактивен	public	-
	U VPN			192.168.15.99	🚫 Неактивен	public	-
	🗱 Маршрутизаторы		®	192.168.15.97	🙆 Запущена	public	ContraIVM
	➤		®	192.168.15.53	🚫 Неактивен	public	-
	 Группы безоласности Балансировщики нагрузки 		®	192.168.15.176	🚫 Неактивен	public	-
	SSH-ключи						

- В файл network.tf добавляем следующий код:
 - Создавая ассоциацию ранее созданного плавающего IP-адреса с ещё ранее созданным портов;
 Для каждого инстанса и балансировщика нагрузки;

<pre># Создадим для WebADM accoциацию плавающего IP и порт (публичного и приватного IP адресов) resource "openstack_networking_floatingip_associate_v2" "association_webadm" { port_id = openstack_networking_port_v2.port_webadm.id floating_ip = openstack_networking_floatingip_v2.floatingip_webadm.address }</pre>
<pre># Создадим для WEB1 ассоциацию плавающего IP и порт (публичного и приватного IP адресов) resource "openstack_networking_floatingip_associate_v2" "association_web1" { port_id = openstack_networking_port_v2.port_web1.id floating_ip = openstack_networking_floatingip_v2.floatingip_web1.address }</pre>
<pre># Создадим для WEB2 ассоциацию плавающего IP и порт (публичного и приватного IP адресов) resource "openstack_networking_floatingip_associate_v2" "association_web2" { port_id = openstack_networking_port_v2.port_web2.id floating_ip = openstack_networking_floatingip_v2.floatingip_web2.address }</pre>
<pre># Создадим для Load Balancer ассоциацию плавающего IP и порт (публичного и приватного IP адресов) resource "openstack_networking_floatingip_associate_v2" "association_loadbalancer" { port_id = openstack_networking_port_v2.port_loadbalancer.id floating_ip = openstack_networking_floatingip_v2.floatingip_loadbalancer.address }</pre>

iter a value: yes	
<pre>stack_networking_floatingip_associate_v2.association_loadbalancer: Creating stack_networking_floatingip_associate_v2.association_webadm: Creating stack_networking_floatingip_associate_v2.association_web2: Creation complete after 6s [id=750.a296-6420-4ff7-881a-cfcb4ddb8192] stack_networking_floatingip_associate_v2.association_webdm: Creation complete after 6s [id=750.a296-6420-4ff7-881a-cfcb4ddb8192] stack_networking_floatingip_associate_v2.association_loadbalancer: Creation complete after 6s [id=c559f43-dd55-4cba-aed6-cdab66a6677-64476-8891-64674787-881-64867-8476-8891-6486747867-8476-8891-6486747867-8476-8891-8816-86664877-8476-8891-8866-84768891-8866-8476-8891-886-8476-8891-886-8476-891-886-8476-891-886-8476-891-886-8476-891-886-8476-891-886-8476-891-886-8476-891-886-8476-891-886-8476-886-8476-8891-886-8476-886-8476-8861-886-8476-886-8476-8861-891-886-8476-886-8476-8861-886-8476-886-8476-8861-886-8476-886-8476-8861-886-8476-886-8476-886-8476-886-8476-8861-866-8476-886-8476-886-8476-886-8476-886-8476-886-8476-866-8476-866-8476-866-8476-866-8476-866-8476-866-847686-8476-866-847686-8476-866-8476-866-8476-866-8476-866-8476-866-8476-866</pre>	027]
y complete! Resources: 4 added, 0 changed, 0 destroyed.	

Проверяем в веб-интерфейсе статус плавающих IP-адресов сменился с Неактивен на Запущена:
 Также появились IP-адреса BM (но сами BM (инстансы) ещё не созданы) из подсети INTERNET (192.168.200.0/24

созданной ранее);

ĸ	ИБЕРПРОТЕ	кт					Project3
14-	Вычисления 🖽	Пла	авающие IP-адреса				
Θ	🕲 Виртуальные мадины	Dur	n q				
	Образы		Potper a	Статус	Cette	Hadmanetw	iP-adpec BM
	A Cene		192,168.15.161	 Запущена 	public	-	192,168.200.22
	() VPN		192,168,15.99	🧔 Запущена	public	-	192.168.200.20
	11 Маршруплаторы		192,168.15.97	🔿 Запущена	public	ControlyM	192.168.190.128
-	🔸 🎯 Плаваюцьк IP адреса		(P) 192.168.15.53	🥥 Запущена	public		192.168.200.25
	 бруппы безогаснаста баланстровщами напрузка 		() ISE 168.15.176	🖨 Запущена	public	-	192.368.200.21
	D SSH-ktronen						

 Создаём файл security.tf и описываем последовательно часть касающуюся сетевой безопасности (с точки зрения требования задания) для развёртывания данной инфраструктуры с поэтапным запуском и наблюдением созданных ресурсов:

vim network.tf

• Помещаем следующее содержимое:

Создаём две группы безопасности для будущего контроля ICMP и SSH



Результат:

Enter a value: yes openstack_networking_secgroup_v2.secgroup_2: Creating... openstack_networking_secgroup_v2.secgroup_1: Creating... openstack_networking_secgroup_v2.secgroup_2: Creation complete after 1s [id=d4594ebe-bcae-4396-a991-c9cd6c2b16a6] openstack_networking_secgroup_v2.secgroup_1: Creation complete after 1s [id=3b22146c-c84e-4b50-9b2b-e232a770a66b] Apply complete! Resources: 2 added, 0 changed, 0 destroyed. [allingsecontrolverbin]s

- Проверяем наличие созданных ресурсов средствами openstack-cli:
 - Группы безапасности:

ID	Name	Description	Project	Tags
3b22146c-c84e-4b50-9b2b-e232a770a66b ca0f6f34-12a8-4c36-a466-4b30a6ea807d d4594ebe-bcae-4396-a991-c9cd6c2b16a6	t ICMP default SSH	- I IOMP Default security group SSH	* 4deb49c7b35546948e7e1a1a39941814 4deb49c7b35546948e7e1a1a39941814 4deb49c7b35546948e7e1a1a39941814 4deb49c7b35546948e7e1a1a39941814	

- Проверяем наличие созданных ресурсов средствами веб-интерфейса:
 - Группы безапасности:

KUBEPHPOTEKT

🖊 Вычисления 🔃	Гру	ппы безопасности		
🚱 🛛 Виртуальные машины	Пон	× Q		
 Образы Пома 		Vines 🕇	ID	Описание
👗 Сети		default.	ca0f6f34-12a8	Default security group
O VPN		CMP ICMP	3b22146c-c84	ICMP
🗱 Маршрутизаторы		€ SSH	d4594ebe-bca	SSH
Плавающие IP-адреса				
🗕 🖕 🔠 Группы безопасности				
🔥 Балансировщики нагружи				

В файл security.tf добавляем следующий код:

- Создавая правило, в ранее созданную группу безопасности обращаясь по идентификатору id созданного ресурса secgroup_1 для фильтрации по протоколу icmp с любых IP-адресов для входящего трафика;
- Создавая правило, в ранее созданную группу безопасности обращаясь по идентификатору id созданного ресурса secgroup_2 для фильтрации по протоколу tcp на порт 22 (SSH) с любых IP-адресов для входящего трафика;
- Ограничивая тем самом доступ к будущим инстансам по ICMP и SSH из внешних сетей;

# Добавляем правило	о в группу безопасности для ICMP
resource "openstacl	<pre><_networking_secgroup_rule_v2" "secgroup_rule_icmp" {</pre>
direction	= "ingress"
ethertype	= "IPv4"
protocol	= "icmp"
<pre>remote_ip_prefix</pre>	= "0.0.0.0/0"
<pre>security_group_id</pre>	<pre>d = openstack_networking_secgroup_v2.secgroup_1.id</pre>
}	
# Добавляем правило	о в группу безопасности для SSH
resource "openstack	<_networking_secgroup_rule_v2" "secgroup_rule_ssh" {
direction	= "ingress"
ethertype	= "IPv4"
protocol	= "tcp"
port_range_min	= 22
port_range_max	= 22
<pre>remote_ip_prefix</pre>	= "0.0.0/0"
<pre>security_group_id</pre>	<pre>d = openstack_networking_secgroup_v2.secgroup_2.id</pre>
}	

Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply:
 Результат:



nstack_networking_secgroup_rule_v2.secgroup_rule_ssh: Creating... nstack_networking_secgroup_rule_v2.secgroup_rule_icmp: Creating... nstack_networking_secgroup_rule_v2.secgroup_rule_icmp: Creation complete after 1s [id=ad88027a-77d7-4a24-90dc-03b86b081360] nstack_networking_secgroup_rule_v2.secgroup_rule_issh: Creation complete after 1s [id=ad88027a-63fe-41d2-961e-3406e0a4aa21] oly complete! Resources: 2 added, 0 changed, 0 destroyed.

- Проверяем наличие созданных ресурсов средствами openstack-cli:
 - Правила в группах безопасности:

[eltline.control= hin]t operatocki	isecure securi	ty group rul	le list 🔶	at and an and				
		[Ethertype	IP Range	Port Range		Remote Security Group	Remote Address Group	
) None	IP/6				None	None	
70 2173a6dc-c1ad-44c9-834a-349767c844bc	Noré				1 Ingress	ca016134-12a8-4c36-a466-4b30a6ea807d	None	ca016134-12a8-4c36-a466-4b3(a6ea80
70 34da24db-167f-4fb6-8630-41553eDe35f0	None) egrass		Norse	01594ebe-bcae-4396-a991-c0cd6c2b16
a6 31913dde-fe56-416c-babt-cedcba26595c	None	1 IPv6			ingress	ca0f6f34-12a8-4c36-a466-4b30a6ea807d	None	ca0f6f34-12a8-4c36-a456-4b30a6ea80
7d 4ab8a53a-63fe-41d2-961e-3406e0a4aa21					ingress	None	None	64594ebe-bcae-4396-a991-c9cd6c2b16
a6 4b32153c-9b33-455e-8741-136466d71434	None	IPv4	1.0.0.0.0/0		egress		None	3b22146c-c84e-4b50-9b2b-e232a770a6
6b 8448a4o4 467b 473d ac7f 934cb6cb8593	None	IPy6			ingress		None	ca016134-12x8-4c36-a466-4b30a6ex80
7d 9421176F+3853+49dd+8e54+951194e21a6F	Nore	TPr4			ingress	Nore	None:	ca0f6f34-12a8-4c36-a466-4b30a6ea80
7d a21b9d83-88d5-4cba-ae73-cad40be28d65	None	1.1996			egress	None	Note:	d4594ebe-bcae-4396-a991-c9cd6c2616
a5 a7217f79-4a8c+4069-hac0-bh7306c#1c9f	Nore	I IPv6			egress.	Norse		3b22146c-c84e-4b50-9b2b-e232a770a5
6b ad88027a-77d7-4a24-90dc+03b85b081360	1 taip	1944	1.0.0.0.0/0		ingress	1 Nooe	None	3b22146c+c84e+4b50+9b2b+e232a779a6
6b f19ad581-757f-4a3a-abff-8962b1b9cbd5 7d	Note	[IPyd	1.0.0.0.0/0		egress	None	None	ca0f6f34-12a8-4c34-a466-4b30a6ea80
•								

• Проверяем наличие созданных ресурсов средствами веб-интерфейса: • Правила в группах безопасности:



KUBEPHPOTEKT

44	Вычисления 🔝	Группы безопасности			SSH				>
0		Toro: Q			× stramorta - 1	В Удалить			
	🖨 Tova	Vivin 1 default	10 ca0%5f34-1228	Drucawie Default security group	Dyami	u	Capiloras	Hamperson DM	
	Q VPN	Contraction (Contraction)	35221466-c84	KMP	Для входящего	трафика 🔫	-	 Добланть 	
	 Маршруткаторе Плавикоция IP-адреса 	🛱 25H 🔶	d4294abe-box	35H	Пратокол Ф	Диапазон портов	Источных Ф		
	→@ Групц-Безоласности				SSH	22	0.0.0.0/0	-	8
	B 224 Kitelian				Для исходнция	о трафика		🕀 Добашить	3
					Пратокол Ф	Диятикон партов	Hasherson O		
					/kedo#	1-65535	0.0.0.00		ŧ.
					Definit	1-65535	::Mi		胞

- В файл security.tf добавляем следующий код:
 - Поскольку в дальнейшем по заданию на инстансах WEB1 и WEB2 должно быть развёрнуто веб-приложение, а значит они должны принимать входящий трафик на порты 80/tcp и 443/tcp (HTTP и HTTPS)
 - Поскольку в дальнейшем по заданию необходимо реализовать туннельное соединение между инстансами WebADM и WEB1, WEB2, а значит WebADM должен принимать входящий трафик на порт реализуемого VPN-соединения, например WireGuard (51820/udp);
 - Таким образом необходимо создать ещё 1 или 2 группы безопасности с соответствующими правилами;
 - В данном примере создаются две группы безопасности с именами WEB и VPN и три правила: два для группы WEB и одно для группы VPN;

```
# Создаём группу безопасность для WEB-трафика (HTTP/HTTPS)
resource "openstack_networking_secgroup_v2" "secgroup_3" {
 name = "WEB"
  description = "WEB for HTTP/HTTPS"
}
# Добавляем первое правило в группу безопасности для WEB (HTTP)
resource "openstack_networking_secgroup_rule_v2" "secgroup_rule_http" {
                 = "ingress"
 direction
                   = "IPv4"
  ethertype
                   = "tcp"
 protocol
 port_range_min = 80
port_range_max = 80
  remote_ip_prefix = "0.0.0.0/0"
  security_group_id = openstack_networking_secgroup_v2.secgroup_3.id
}
# Добавляем второе правило в группу безопасности для WEB (HTTPS)
resource "openstack_networking_secgroup_rule_v2" "secgroup_rule_https" {
               = "ingress"
  direction
                   = "IPv4"
  ethertype
                   = "tcp'
 protocol
 port_range_min = 443
port_range_max = 443
  remote_ip_prefix = "0.0.0.0/0"
 security_group_id = openstack_networking_secgroup_v2.secgroup_3.id
}
# Создаём группу безопасность для VPN (WireGuard)
resource "openstack_networking_secgroup_v2" "secgroup_4" {
  name = "VPN"
  description = "VPN (Wireguard)"
}
# Добавляем правило в группу безопасности для VPN (WireGuard)
resource "openstack_networking_secgroup_rule_v2" "secgroup_rule_vpn" {
               = "ingress"
= "IPv4"
 direction
 ethertype
                   = "udp"
 protocol
 port_range_min = 51820
port_range_max = 51820
 remote_ip_prefix = "0.0.0.0/0"
 security_group_id = openstack_networking_secgroup_v2.secgroup_4.id
}
 • Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply:
```

```
Enter a value: yes

openstack_networking_secgroup_v2.secgroup_4: Creating...

openstack_networking_secgroup_v2.secgroup_1: Creating ...

openstack_networking_secgroup_v2.secgroup_1: Creating complete after 1s [id=e31e4cda-5c26-4148-b683-dfdd64a6d35b]

openstack_networking_secgroup_rule_v2.secgroup_rule_http: Creating...

openstack_networking_secgroup_rule_v2.secgroup_rule_http: Creating...

openstack_networking_secgroup_rule_v2.secgroup_rule_http: Creating...

openstack_networking_secgroup_rule_v2.secgroup_rule_http: Creating...

openstack_networking_secgroup_rule_v2.secgroup_rule_http: Creating...

openstack_networking_secgroup_rule_v2.secgroup_rule_vpn: Creating...

openstack_networking_secgroup_rule_v2.secgroup_rule_vpn: Creating complete after 0s [id=88d9fc5e-44d9-4c71-8838-8cf705639a47]

openstack_networking_secgroup_rule_v2.secgroup_rule_vpn: Creation complete after 0s [id=7ee89ddd=8cd1-4ee5-b61e-03c5641f2caa]

openstack_networking_secgroup_rule_v2.secgroup_rule_tptps: Creation complete after 0s [id=10e2aa67-3b0b-4b57-a8be-d259d1784424]

Apply_complete1 Resources: 5 added, 0 changed, 0 destroyed.

[allingecontrolvem_bin]; ]
```

Проверяем наличие созданных ресурсов средствами openstack-cli:
 Правила в группах безопасности:

Результат:

10	IP Protocol	Ethertype	IP Range	Port Range	Direction	Remote Security Group	
а М	l	4	¥	A	A	4	
0986faa0-3719-497a-aa05-68e422ae9b64	None	IPv6			egress	None	
074	E aver	1.75.4		L MARKANY	L reference and	1 10000	
10e28867+3000+4057-880e+025901784424 35h i	tcp	1 1 994	1 0.0.0.0/0	443:443	ingress) None	
2173a60c-c1ad-44c9-834a-349f67c844bc	None	IPv4	0.0.0.0/0		ingress	ca0f6f34-12a8-4c36-a466-4b30a6ea807	1
07d							
201a8a46-16c3-4a5d-a39c-662d73c31896	None	I IPv6	1 ::/0		egress	None	
34da24db-167f-4fb6-8630-41553e0e35f0	None	I IPv4	1.0.0.0.0/0		l egress	1 None	
6a6							
3f9f3dde-fe56-4f6c-babf-cedcba2b595c	None	IPv6	::/0		ingress	ca0f6f34-12a8-4c36-a466-4b30a6ea807	1
U/G 1_4ab8a53a_63fe_41d2_961e_3406e0a4aa21	ten	1 TPod	1 0 0 0 0/0	1 22-22	Ingress	1 None	
646	1.3582	1		1	1.4966633	1 notes	
4b32153c-9b33-455e-8741+136466d71434	None	IPv4	0.0.0.0/0		egress	None	
660	I MARKA	I TRUE	1.0.0.0/0		Ladrand	1 Money	
e3d	1 None	TLA4	1 0.0.0.070		egress	1 none	
66796b6f-4828-4cf6-8ad9-15b4c055cad6	None	IPv4	0.0.0.0/0		egress	None	
356			ha a anan				
//ee8900d-8cd1+4ee5-b61c-03c5641f2caa a3d	l udp	LPV4	0.0.0.0/0	51820:51820	ingress	None	
1 8448a4e4-467b-473d-ae7f-934cb0cb8593	None	IPv6	1 :::/0		ingress	None	
07d							
88d9fc5e-44d9-4c71-8838-8cf705639a47	tcp	IPv4	0.0.0.0/0	80:80	ingress	None	
530 0421176f-3853-49dd-8e54-951194e21a6f	None	I-TPud	1.0.0.0.0/0		1 ingross	1 None	
07d					1 angless		
a21b9d83-88d5-4cba-ae73-cad40be28d66	None	IPv6			egress	None	
6a6	(Marine)	L TRUE			1 202229	1 Month	
66b	l none	TLAG			egress	1 none	
ad88027a-77d7-4a24-90dc-03b86b081360	icmp	IPv4	0.0.0.0/0		ingress	None	
66b		Contractor -					
0094ab46-810C-4576-8901-64a72C05a35a	None	EPvb	1 3 3 5 0		egress	None	
1 f19ad581-757f-4a3a-abff-8962b1b9cbd5	None	IPv4	1 0.0.0.0/0		egress	None	
07d							

• Проверяем наличие созданных ресурсов средствами веб-интерфейса: • Правила в группах безопасности:



KUBEPHPOTEKT

16	Вычисления ⊡	Группы безопасности			VPN				1
۲	🔁 Bayryarassan takuttete	Tiwos Q			P Hasemira B	Уданить			
	 Обрани Тома 	itun	0	Описание	Правил	r Cor	аста	Rassovennue IM	
	J. Cens () VPN	Contention of the second secon	3022146c-c84	Default security group				E talana	
	11 Маршрунскатори @ Паражошче IP адреса	<u>бі</u> 55н	d4594ebe-bca+.	55H	Протакал ()	диапазон портов	Источник Ф	(c) phoneses	8
Ļ	🕳 🏦 Труппы безспасности	wes	estate-sc2.	Web (Wreguard)	LOP	51820	00000		ß
	SSH kind und seine subjetent				для исходящего	трофина		🕘 Добавить	1
					Протокол @	Диальное портов	Hanavener O		
					ðistoil.	1 - 65535	00000		ß
					- distoil -	1-65530	0.49.0		ŧ.

• В файл network.tf добавляем следующий код:

- Назначая ранее созданные группы безопасности на ранее созданные порты для каждого инстанса, обращаясь по идентификаторам id созданных ранее ресурсов;
- На порт для инстанса WebADM назначаем группы безопасности ICMP, SSH (явное требование задания) и VPN, т.к. в дальнейшем данный инстанс будет выступать в роле VPN-сервера и должен принимать соединения на порт указанный в правиле для данной группы безопасности;
- На порт для инстансов WEB1 и WEB2 назначаем группы безопасности ICMP, SSH (явное требование задания) и WEB, т.к. в дальнейшем на данных инстансах будет развёрнуто веб-приложение и должны приниматься соединения на порты указанные в правилах для данной группы безопасности;

```
# Назначим группы (ICMP, SSH, VPN) безопасности на порт для инстанса WebADM
resource "openstack_networking_port_secgroup_associate_v2" "security_group_associate_webadm" {
   port id
                     = openstack_networking_port_v2.port_webadm.id
    enforce
                      = true
    security_group_ids = [
     openstack_networking_secgroup_v2.secgroup_1.id,
     openstack_networking_secgroup_v2.secgroup_2.id,
     openstack_networking_secgroup_v2.secgroup_4.id
    1
}
# Назначим группы (ICMP, SSH, WEB) безопасности на порт для инстанса WEB1
resource "openstack_networking_port_secgroup_associate_v2" "security_group_associate_web1" {
               = openstack_networking_port_v2.port_web1.id
   port_id
    enforce
                      = true
    security_group_ids = [
     openstack_networking_secgroup_v2.secgroup_1.id,
     openstack_networking_secgroup_v2.secgroup_2.id,
     openstack_networking_secgroup_v2.secgroup_3.id
   ]
}
# Назначим группы (ICMP, SSH, WEB) безопасности на порт для инстанса WEB2
resource "openstack_networking_port_secgroup_associate_v2" "security_group_associate_web2" {
    port_id
                    = openstack_networking_port_v2.port_web2.id
    enforce
                      = true
    security_group_ids = [
     openstack_networking_secgroup_v2.secgroup_1.id,
     openstack_networking_secgroup_v2.secgroup_2.id,
      openstack_networking_secgroup_v2.secgroup_3.id
    ]
}
```

Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply:
 Результат:



 Создаём файл instance.tf и описываем последовательно часть для развёртывания необходимых инстансов (виртуальных машин) с поэтапным запуском и наблюдением созданных ресурсов:

```
vim instance.tf
```

• Помещаем следующее содержимое:

- Т.к. все ресурсы и сети должны быть созданы в соответствие с топологией по условиям задания, а на топологии ControlVM также должна быть подключена к сети INTERNET;
- То добавляем к инстансу ControlVM ещё один сетевой интерфейс из подсети INTERNET и назначаем
- фиксированный ІР-адрес;
- Необходимо указать идентификатор id виртуальной машины ControlVM

Подключаем в ControlVM интерфейс из подсети "INTERNET" в соответствие с топологией
resource openstack_compute_internace_attach_vz controlvm {
instance_id = "96306ddf-a488-4e35-8a8f-25696821efe5"
<pre>network_id = openstack_networking_network_v2.network.id</pre>
fixed_ip = "192.168.200.10"
<pre>depends_on = [openstack_networking_subnet_v2.subnet]</pre>
}

• instance_id (идентификатор) можно получить средствами openstack-cli:

ID	Name	Status	Networks	Image	Flavor
+ 96306ddf-a488-4e35-8a8f-25696821efe5	ControlVM	ACTIVE	<pre>cloud=192.168.100.128, 192.168.15.9</pre>	7 N/A (booted from volume)	medium

• instance_id (идентификатор) можно получить средствами веб-интерфейса:



Enter a value: yes
openstack_compute_interface_attach_v2.controlwn: Creating openstack_compute_interface_attach_v2.controlwn: Still creating [10s elapsed] openstack_compute_interface_attach_v2.controlwn: Creation complete after 17s [id=96306ddf-p488-4e35-8a8f-2569682tefe5/bea9153c-e7cd-41f7-8592-35047e452ed3
Apply complete! Resources: 1 added. 0 changed, 0 destroyed. [altlinux@controlym bin]%
 Проверяем наличие второго сетевого интерфейса из подсети INTERNET средствами openstack-cli:

[altlinux@controlvm_bin]\$_openstack addresses	insecure server show ControlWV grep "address"
[altlinux@controlvm bin]\$	

• Проверяем наличие второго сетевого интерфейса из подсети INTERNET средствами веб-интерфейса:

K	берпроте	кт			Project1 ~	0 0
16	Вычисления 🔝	Виртуальные машины	~	ControlVM		×
0	 Вергуальные машины Обраще 	C C C C C C C C C C C C C C C C C C C	Cranye i	Decourse O Repearpyonts	© выеличнить + Алтаратное переза	qrysea
	A Tona	🖗 Carloshti 🔶 ———————————————————————————————————	0 затуация	Orising	Монитории	
	 ум. Марирутанцара Марирутанцара Тарлак болгански Алерска Тарлак болгански Алерски А болгански Алерски Д болгански Алерски Д болгански Алерски Д болгански Алерски 			Casolicmaa Inus Odges Presso expression (1) Texes Texe 00 Texe 00 Texes Tegestembores (1) a 0397 Aartseutorescours um pauge 085 Cerressione sinnergoljoit (1) deatt Metages (c) Metry but (1) accession	ControlMal an envire p10 schooladty, 64.spmm2 Minansamens seconpublic sciencepic bost- Tasa 12.01 r.63, strinub Sarrysst-ward medium (2 LIP, 4.746 COS) Circusa-boso Bacteriane Pageler 151.10.1053/L.73946 Macanashina 1	radel / /
				INTERNET MAC appent to 16 Jan male 37, Coson-sal 1	Р адрес: 192 тійі. 200 18. Группи безелеріністи: 1	2

• В файл instance.tf добавляем следующий код:

- Описывая конфигурацию инстансов WebADM, WEB1 и WEB2;
- См. ниже информацию о переменных var.* и файле cloud-init.yml;

```
# Создаём инстанс с именем "WebADM"
resource "openstack_compute_instance_v2" "webadm" {
 source openstack_compute_instante_v2 v
name = "WebADM"
flavor_id = var.flavor_id
key_pair = var.key_pair
user_data = file("cloud-init.yml")
 block_device {
                              = var.image_id
    uuid
    uuid = var.ima;
source_type = "image"
volume_size = 10
boot index = 0
   boot_index = 0
destination_type = "volume"
   delete_on_termination = true
  }
  network {
   port = openstack_networking_port_v2.port_webadm.id
  }
}
# Создаём инстанс с именами "WEB1"
resource "openstack_compute_instance_v2" "web1" {
 rame = "WEB1"
flavor_id = var.flavor_id
key_pair = var.key_pair
user_data = file("cloud-init.yml")
  block_device {
   uuid = var.image_id
source_type = "image"
volume_size = 10
   boot_index = 0
destination_type = "volume"
   delete_on_termination = true
  }
  network {
   port = openstack_networking_port_v2.port_web1.id
  }
}
# Создаём инстанс с именами "WEB2"
resource "openstack_compute_instance_v2" "web2" {
              = "WEB2"
= var.flavor_id
= var.key_pair
= file("cloud-init.yml")
  name
  flavor_id
  key_pair
  user_data
  block_device {
  source_type
volume_size
boot_index
                               = var.image_id
   uuid
                              = "image"
                               = 10
   boot_index = 0
destination_type = "volume"
   delete_on_termination = true
  }
  network {
   port = openstack_networking_port_v2.port_web2.id
  }
}
```

• В файл variables.tf добавляем следующий код:

- Определяя значение переменных указывая идентификаторы id для соответствующих (существующих) ресурсов;
- См. ниже где брать идентификаторы;
- Учитывая основные требования задания для создаваемых инстансов:

. Основные характеристики:

- i. Операционная система: Альт p10 StarterKit/Альт Сервер p10-cloud
- іі. Количество vCPU: 1.
- iii. Объём оперативной памяти: 1024 MБ.
- iv. Объём диска: 10 ГБ/30 ГБ

```
# ID для образа "alt-p10-cloud-x86_64.qcow2" (Starterkit)
variable "image_id" {
   type = string
    default = "92e78753-4f88-40eb-a10a-5a7fb9bfc106"
}
# ID для шаблона 1 vCPU 1 RAM (openstack --insecure flavor list)
variable "flavor_id" {
   type = string
    default = "1f64883c-1cdd-45e4-ac8a-82ab57a12fdf"
}
# Имя ssh-ключа
variable "key_pair" {
    type = string
    default = "cloud"
}
```

• Для переменной image_id значение можно получить: Средствами openstack-cli:

ID	Name	Status
+	+	+
92e78753-4f88-40eb-a10a-5a7fb9bfc106	alt-p10-cloud-x86_64.qcow2	active
23a88949-aaab-4a97-93fb-1b24f5fd1cb7	alt-server-p10-cloud-x86_64.gcow2	active
5499d36e-3eb6-43c0-b1e6-a1d228ce5c86	cirros	active

Средствами веб-интерфейса:

K	иберпроте	кт				tyajacet 🗢
4-	Вычисления 🔃	Образы			alt-p10-cloud-x86_64.qc	ow2
Ø	😨 Виртуальные машяны	🔹 Toece 🔍			🔒 Создать том 👍 Загрузить с	браз
1	O Ofpano	Hus +	Crarye i	Twen		
1	A Cetta	📇 alt-p10-doud-x86,54.gcmv2 +	- O Artunen	Шаблан	Сведения	
	() VPN	ah-server-p10-cloud x86_64.gcow2	о Актирен	Шаблон	Charlyc	👌 Актявен
	12 Маршрутисаторы	Cirros	Актирен	Шаблон	Passep	385 Me6
	 Пливающие IP-адреся Пруппы безопасности 				Идентификатор образа	92c78753-4/88-40cb-a10a-5a7fb9cfc106
	🔥 Балансировцики нагруски			· · · · ·	Свойства	
	SSH execution				Hue	alt-p10-cloud-x86_64.qcow2
					Twn OC	Generic Linux
					Минимальный размер тома	3746

- Для переменной flavor_id значение можно получить:
 - Средствами openstack-cli:

		1				
ID	Name	RAM	Disk	Ephemeral	VCPUS	IS Public
***************************************	+		+			
100	tiny	512	0	0		True
101	small	2048	0			True
102	medium	4096	0	0	2	True
103	large	8192	i 0 i	0	4	True
104	xlarge	16384	0	0	8	True
1f64883c-1cdd-45e4-ac8a-82ab57a12fdf	minimal	1024	0	0		True

- Средствами веб-интерфейса нет возможности;
- Для переменной key_pair значение можно получить: Средствами openstack-cli:

[altlinux	<pre>@controlvm bin]\$ openstackinsecure keypair lis</pre>	st		
+		-+-		-+
Name	Fingerprint		Туре	
+		-+-		
cloud	a8:54:b3:44:c4:a7:2a:dc:56:ef:ff:cb:3b:ff:89:ab		ssh	
+				
Lal + Linus	decontrolum biold			

Средствами веб-интерфейса:

КИБЕРПРОТЕКТ

∿∘	Вычисления ⊡	SSH-ключи
0	 Виртуальные машины Образы Тома Сеги VPN Маршрутизаторы Плавающие IP-адреса Группы безопасности Балансировщики нагрузки SSH-ключи 	Поиск Q

Средствами Terraform передаём именно тот ключ, на основе которого осуществляется доступ с рабочего места до ControlVM

 Чтобы при необходимости можно было подключиться по SSH к каждому инстансу по "публичному" (Плавающему) IP адресу, что также является требованиями задания;

```
    Создаём файл cloud-init.yml:
```

vim cloud-init.yml

- Помещаем следующее содержимое:
 - Задавая пароль по требованиям задания P@ssw0rd для каждого создаваемого инстанса;
 - Также передаём значение публичной части SSH-ключа (генерация ssh-ключей см. ниже), чтобы в дальнейшем был доступ с ControlVM до каждого инстанса по SSH;
 - И именно по ключам, т.к. по условиям задания подключение по SSH на основе открытых ключей;
 - Доступ по SSH с ControlVM до каждого инстанса необходим для дальнейшей работы Ansible;



w key's randomart inage is:				
++{RSA>2348]>+>++				
0 8 = +1				
n 0 + 1				
405				
00.00				
,E O, B U, [
+ =+++0.				
tere [SPA250] const				
<pre>iltllnnwlcontrolwn/bin]1 cat =/.ssh/idire</pre>	la , pub			
It-rsa AAAAB3NzaCiyc2EAAAADADABABAADACDFz1	MisEG9ToRndaRRT DXsDndDgAE7r FPNsP1Dr9tmo4QsvmMdF3LR	gOTR5p53cP1zUgxfstxKCfvW1AfWCE31JW6kaV8	BBENN FOG jCBenNe93cIK741Fk3cZnrHHUJ5vMd	sD=hsn206V#OwScUtipi9B?Ag0F0EsL
TwiEeejJCOerQULASQJM785NTcRN523v5HhJRyW	4xrfBhP70e2j1Hwy+jWCVTpO/W714Q5EtcVegSZup20oJ6K/	%GpgjLoP4eU1r8EVIWLG25AJLI#1FL5hTsOwa74	20N+TWFSC4G6PFXtMLdZ3ebPJ2n1c#pydrw50Ee	NTy4vPA2bWED altlinux@controlvn

• Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply: • Результат:

Plan: 3 to add, 0 to change, 0 to destroy.
Do you want to perform these actions? Terraform will perform the actions described above. Only 'yes' will be accepted to approve.
Enter a value: yes
<pre>openstack_compute_instance_v2.web1: Creating openstack_compute_instance_v2.web2: Creating openstack_compute_instance_v2.web2: Creating openstack_compute_instance_v2.web1: Still creating [10s elapsed] openstack_compute_instance_v2.web2: Still creating [10s elapsed] openstack_compute_instance_v2.web2: Still creating [20s elapsed] openstack_compute_instance_v2.web2: Creation complete after 25s [id=88398bf3-2c63-48c9-817b-5ded169725c7] openstack_compute_instance_v2.web2: Creation complete after 25s [id=86398bf3-2c63-48c9-817b-5ded169725c7] openstack_compute_instance_v2.web2: Creation complete after 25s [id=86398bf3-2c63-48c9-817b-5ded169725c7] openstack_compute_instance_v2.web3dm: Creation complete after 25s [id=86398bf3-2c63-48c9-817b-5ded169725c7]</pre>
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

• Проверяем наличие созданных инстансов средствами openstack-cli:

	Nane	Status	Networks	Image	Flavor
74e2bb78-e034-463c-a2ad-ca0d843d47d3 88398bf3-2c63-48c9-817b-5ded160725c7 c4ce7221-58a9-4804-a190-89f438526921 96306ddf-a488-4e35-8a8t-25696821efe5	WEB1 WEB2 WebADM ControlVM	ACTIVE ACTIVE ACTIVE ACTIVE	INTENNET-192.168.15.176, 192.168.200.21 INTENNET-192.168.15.161, 192.168.200.22 INTENNET-192.168.15.99, 192.168.200.20 INTENNET-192.168.200.10; claud=192.168.100.128, 192.168.15.97	N/A (booted from volume) N/A (booted from volume) N/A (booted from volume) N/A (booted from volume) N/A (booted from volume)	minimal minimal minimal medium

• Проверяем наличие созданных инстансов средствами веб-интерфейса:

K		кт								Project.1 w
4.	Вычисления 🔳	Виртуа/	ьные машины							
0	😨 Виртуальные мацанны	# dwnerp	Toiro	q					(+ Cou	дать виртуальную
1)	O Ofpane	1	er †		Статус	іР-адрес	eUN +	037 4	Хранилище Б	Тона
	& Ceta	0 8	ControlVM		О Запущена	192.168.100.128, 192.16.,	2	4 Fat5	30 Fv6	1
	(I) VPN	0 0	WEB!		😋 Запущена	192.168.200.21	4	1 FeB	10 rv6	
	Марарутизаторы	0 6	WEB2		🗿 Запущана	192.168.208.22	18	106	10 Net	1
	 Плаванодне Р-адреса Сруппы безопасности 	0 6	WARADIA		о запущена	192.168.209.20	3	t futfi	10 Folk	1

• Также проверяем что:

• Каждый инстанс подключён к сети INTERNET с неоходимыми группами безопасности:





• Каждый Плавающий IP-адрес назначем на инстансы (за исключением 1-го для балансировки нагрузки):

КИБЕРПРОТЕКТ

Вычисления 🔳	Плавающие IP-адреса				
🗇 Виртуальные машины					
Образы	с Сеть	Назначен			
🖧 Сети	inyujena public	WEB2			
O VPN	нущена public	WebADM			
# Маршрутизаторы	пущена public	ControlVM			
Плавающие IP-адреса	inyujeva public	-			
 Пруппы безопасности Балансировщики нагрузки 	пущена public	WEB1			
Жаршрутизаторы Плавающие IP-здреса Труппы безопасности Балансировщики нагрузки Ø SSH-килена	пущена риblic пущена риblic пущена риblic				

 Создаём файл loadbalancer.tf и описываем последовательно часть для развёртывания балансировщика нагрузки с поэтапным запуском и наблюдением созданных ресурсов:

vim loadbalancer.tf

- Помещаем следующее содержимое:
 - Создадим балансировщик нагрузки, который будет подключён в соответствие с топологией к подсети INTERNET;

openstack 1b loadbalancer v2.loadbalancer:	Creating	
openstack Ib loadbalancer v2.loadbalancer:	Still creating	[10s elapsed]
openstack lb loadbalancer v2.loadbalancer:	Still creating	[20s elapsed]
openstack 1b loadbalancer v2.loadbalancer:	Still creating	[30s elapsed]
openstack lb loadbalancer v2.loadbalancer:	Still creating	[40s elapsed]
openstack lb loadbalancer v2.loadbalancer:	Still creating	[50s elapsed]
openstack lb loadbalancer v2.loadbalancer:	Still creating	[1mOs elapsed]
openstack lb loadbalancer v2.loadbalancer:	Still creating	[1m10s elapsed]
openstack Ib loadbalancer v2.loadbalancer:	Still creating	[1m20s elapsed]
openstack 1b loadbalancer v2.loadbalancer:	Still creating	[1m30s elapsed]
openstack lb loadbalancer v2.loadbalancer:	Still creating	[1m40s elapsed]
openstack lb loadbalancer v2.loadbalancer:	Still creating	[1m50s elapsed]
openstack_lb_loadbalancer_v2.loadbalancer:	Creation complete	after 1m51s [id=7c63d7bb-d2d8-4062-9fdd-92b91cc3b149]
Apply complete! Resources: 1 added, 0 chang	ged, 0 destroyed.	
faltlinuv@controlvm.htnl5		

• Проверяем наличие созданного балансировщика нагрузки средствами openstack-cli:

	nane	project_id	vip_address	provisioning_status	operating_status	provider
7c63d7bb-d2d8-4062-9fdd-92b91cc3b149	Load Balancer	4deb49c7b35546948e7c1a1a39941814	192.168.200.143	ACTIVE	ONLINE	anphora

• Проверяем наличие созданного балансировщика нагрузки средствами веб-интерфейса:

KI	ИБЕРПРОТЕ	кт			
≁	Вычисления 💽	Бал	ансировщики нагруз	ки	
0	 Виртуальные машины Образы Тома Сети VPN Маршругизаторы Плавающие IP-адреса Группы безопасности 		Имя † Cond Balancer	Cranyc 🖕	IP-адрес 🖕 192.168.200.143
	 Балансировщики нагрузки 55Н-ключи 				

- В файл loadbalancer.tf добавляем следующий код:
- Определяя правила в ранее созданном балансировщике нагрузки для HTTP и HTTPS трафика;

# Создаём правило	о в балансировщике нагрузки для HTTP					
resource "opensta	ack_lb_listener_v2" "listener_http" {					
name	= "HTTP"					
protocol	= "TCP"					
protocol_port	= "80"					
loadbalancer_id	d = openstack_lb_loadbalancer_v2.loadbalancer.id					
}						
# Создаём правило	о в балансировщике нагрузки для HTTPS					
<pre>resource "openstack_lb_listener_v2" "listener_https" {</pre>						
name	= "HTTPS"					
protocol	= "TCP"					
protocol_port	= "443"					
loadbalancer_id	d = openstack_lb_loadbalancer_v2.loadbalancer.id					
}						

Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply:
 Результат:



Создаём целевую группу для HTTP с указанием алгоритма балансировки "ROUND ROBIN" resource "openstack_lb_pool_v2" "pool_http" { name = "HTTP" protocol = "HTTP"
Ib_method = "ROUND_ROBIN"
listener_id = openstack_lb_listener_v2.listener_http.id
}
Создаём целевую группу для HTTPS с указанием алгоритма балансировки "ROUND ROBIN"
resource "openstack lb pool v2" "pool https" {
name = "HTTPS"
protocol = "HTTPS"
lb_method = "ROUND_ROBIN"
listener_id = openstack_lb_listener_v2.listener_https.id
}

Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply:
 Результат:

Enter a value	et yes
openstack_lb_pd openstack_lb_pd openstack_lb_pd openstack_lb_pd	col_v2.pcol_https: Creating col_v2.pcol_http: Creating col_v2.pcol_https: Creation complete after 8s [id=d77602e2-34a9-4422-86dc-a795652eb21b] col_v2.pcol_https: Creation complete after 8s [id=924b756e-d0a6-4ea5-9f5e-730241dbd0ce]
Apply complete	I Resources: 2 added, 0 changed, 0 destroyed.

Проверяем созданные целевые группы средствами openstack-cli:

	name	project_id	provisioning_status	protocol	lb_algorithm	admin_state_up
d77602e2-34a9-4422-86dc-a795652eb21b	HTTPS	4deb49c7b35546948e7e1a1a39941814	ACTIVE	HTTPS	ROUND_ROBIN	True
924b756e-d0a6-4ea5-9f5e-730241dbd0ce	HTTP	4deb49c7b35546948e7e1a1a39941814	ACTIVE		ROUND_ROBIN	True

• Проверяем созданные целевые группы средствами веб-интерфейса:

KUEEPHPOTEKT

Вычисления 💽	Бал	ансировщики нагрузки 🜔 Load Balancer 🗲 🗕		
🔀 Виртуальные машины	Nev	ex Q		
Образы А Тома		Пул балансировки	Статус	Состояние участник
🚑 Сети		💑 ТСР на порте 80 -> НТТР на порте	🔉 Запущен	-
0 VPN		A TCP на порте 443 → HTTPS на порте	О Запущен	2
🚼 Маршрутизаторы				
Плавающие IP-адреса				
👌 Группы безопасности				
🔥 Балансировщики нагрузки				
SSH-ICHIO444				

• В файл loadbalancer.tf добавляем следующий код:

• Добавляя инстансы WEB1 и WEB2 в ранее созданные целевые группы для HTTP и HTTPS;

```
# Добавляем инстанс WEB1 в целевую группу HTTP
resource "openstack_lb_member_v2" "member_web1_http" {
               = "WEB1"
 name
 subnet id
               = openstack_networking_subnet_v2.subnet.id
 pool_id = openstack_lb_pool_v2.pool_http.id
address = "192.168.200.21"
 protocol_port = "80"
}
# Добавляем инстанс WEB2 в целевую группу HTTP
resource "openstack_lb_member_v2" "member_web2_http" {
              = "WEB2"
 name
 subnet_id = openstack_networking_subnet_v2.subnet.id
 pool_id = openstack_lb_pool_v2.pool_http.id
address = "192.168.200.22"
 protocol_port = "80"
}
# Добавляем инстанс WEB1 в целевую группу HTTPS
resource "openstack_lb_member_v2" "member_web1_https" {
               = "WEB1"
 name
 subnet id = openstack networking subnet v2.subnet.id
 pool_id = openstack_lb_pool_v2.pool_https.id
address = "192.168.200.21"
 protocol_port = "80"
}
# Добавляем инстанс WEB2 в целевую группу HTTPS
resource "openstack_lb_member_v2" "member_web2_https" {
               = "WEB2"
 name
 subnet id
               = openstack_networking_subnet_v2.subnet.id
 pool_id = openstack_lb_pool_v2.pool_https.id
address = "192.168.200.22"
  protocol_port = "80"
}
```

Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply:
 Результат:

```
Enter a value: yes

openstack_lb_member_v2.member_web1_http: Creating...

openstack_lb_member_v2.member_web2_https: Creating...

openstack_lb_member_v2.member_web2_http: Creating...

openstack_lb_member_v2.member_web1_http: Still creating... [10s elapsed]

openstack_lb_member_v2.member_web1_http: Still creating... [10s elapsed]

openstack_lb_member_v2.member_web2_http: Still creating... [10s elapsed]

openstack_lb_member_v2.member_web1_http: Creation complete after 18s [id=3af7c156-4ab3-499f-8439-5ade4c52a664]

openstack_lb_member_v2.member_web1_http: Still creating... [20s elapsed]

openstack_lb_member_v2.member_web2_http: Still creating... [20s elapsed]

openstack_lb_member_v2.member_web2_http: Creation complete after 12s [id=3ad740cd-8ce4-4222-ab32-fc27b8fd86dc]

openstack_lb_member_v2.member_web1_http: Creation complete after 24s [id=12e61e9f-2925-4841-b98c-930a6c628457]

Apply complete1 Resources: 4 added, 0 changed, 0 destroyed.

o [altlinux@controlvm bin]$
```

[•] Проверяем наличие инстансов в целевых группах средствами openstack-cli:

[altlinux@controlym bin]% openstackin	isecure	Loadbalancer member list HTTP 🛶					8
id	nane	project_id	provisioning_status	address	protocol_port	operating_status	weight
3ed740cd-8ce4-4222-ab32-fc27b8fd86dc 12e61e9f-2925-4841-b98c-930a6c628457	WEB2 WEB1	4deb49c7b35546948e7e1a1a39941814 4deb49c7b35546948e7e1a1a39941814	ACTIVE ACTIVE	192.168.200.22 192.168.200.21	80 80	NO_MONITOR	1
<pre>[altlinux@controlym bin]% openstack</pre>	isecure	loadbalancer nember list HTTPS 🥌					
10	nane	project_id	provisioning status	address	protocol_port	operating_status	weight
161a06b6-0dd5-4b99-978d-ffd75935e7b2 3af7c156-4ab3-499f-8439-5ade4c52a664	VEB1 VEB2	4deb49c7b35546948e7e1a1a39941814 4deb49c7b35546948e7e1a1a39941814	ACTIVE ACTIVE	192.168.200.21 192.168.200.22	80 80	NO_MONITOR	

• Проверяем наличие инстансов в целевых группах средствами веб-интерфейса:





• В файл loadbalancer.tf добавляем следующий код:

• Добавляя проверку доступности инстансов в каждой целевой группе;

# Создзём пл	орерку лоступности инстансов в нелевой группе HTTD
# создаем пр	sobepty decivine in whitehold b denebow i pynne in ir
resource of	enstack_to_monitor_vz monitor_nttp {
name	= "monitor HIIP"
pool_id	<pre>= openstack_lb_pool_v2.pool_http.id</pre>
type	= "PING"
delay	= "10"
timeout	= "4"
max_retrie	es = "5"
}	
# Создаём пр	юверку доступности инстансов в целевой группе HTTPS
resource "op	<pre>venstack_lb_monitor_v2" "monitor_https" {</pre>
name	= "monitor HTTPS"
pool_id	<pre>= openstack_lb_pool_v2.pool_https.id</pre>
type	= "PING"
delay	= "10"
timeout	= "4"
max retrie	es = "5"
}	

Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply:
 Результат:

Enter a value: yes
openstack_lb_monitor_v2.monitor_http: Creating openstack_lb_monitor_v2.monitor_https: Creating
openstack_lb_monitor_v2.monitor_https: Creation complete after 7s [id=0919f2d1-4a3b-4c01-92a6-280be85111ac] openstack_lb_monitor_v2.monitor_http: Creation complete after 7s [id=b0f72673-2d1a-4f48-8299-cc2913090583]
Apply complete! Resources: 2 added, 0 changed, 0 destroyed. [altlinux@controlvm bin]\$ [altlinux@controlvm bin]\$
 Проверяем доступность мониторинга инстансов в целевых группах средствами openstack-cli:

[altlinux@controlum bin]\$ openstackin	nsecure	loadbalancer nember list HTTP 🔫					
id	name	project_id	provisioning_status	address	protocol_port	operating_status	weight
3ed740cd-8ce4-4222-ab32-fc27b8fd86dc 12e61e9f-2925-4841-b98c-930a6c628457	WEB2 WEB1	4deb49c7b35546948e7e1a1a39941814 4deb49c7b35546948e7e1a1a39941814	ACTIVE ACTIVE	192.168.200.22 192.168.200.21	80 80	ONL THE ONL THE	1
[altlinum@controlwn_bin]\$ openstacki	secure	loadbalancer member list HTTPS 📹		******			*******
id	name	project_id	provisioning_status	address	protocol_port	operating_status	weight
161a06b6+0dd5+4b99+978d+FFd75935e7b2 3af7c156+4ab3+499F-8439-5ade4c52a664	WEB1 WEB2	4deb49c7b35546948e7e1a1a39941814 4deb49c7b35546948e7e1a1a39941814	ACTIVE ACTIVE	192.168.200.21 192.168.200.22	80 80	ONLINE ONLINE	
Faltlinexecontroles bints	******						

• Проверяем доступность мониторинга инстансов в целевых группах средствами веб-интерфейса:

Балансировщики нагрузки

*	линьтр Понтя	٩		~	+ 60	эздать балансировщих нагру	узки
	Visas 1	Статус 🐇	IP-адрес 👍	Плавающой IP-ад_ 🕴	Состояние участников	Всего участников 🔒	0
	A Load Balancer	🗿 Запущен	192.168.200.143	-	-	-4	
					о Исправен 4		

Создаём файл output.tf:

vim output.tf

- Помещаем следующее содержимое:
 - Для печати на экран "внешний (публичные)" плавающие IP-адреса всех инстансов;
 - Понадобится в дальнейшем для сохранения вывода в файл по требованиям задания;



Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply:
 Результат:

Enter a value: yes	
Apply complete! Resources: 0 added, 0 changed, 0 destro	yed.
Outputs:	
LoadBalancer = "192.168.15.53" WEB1 = "192.168.15.176" WEB2 = "192.168.15.161" WebADM = "192.168.15.99" [alt]nux@controlwn.bin]\$	

- На текущий момент вся инфраструктура развёрнута в соответствие с требованиями задания, за исключением настройки VPN-туннеля и SSH по паролю в рамках туннельного соединения
- Реализуем недостающий функционал средствами связки Terraform и Ansible
- Создаём файл templates.tf:

vim templates.tf

- Помещаем следующее содержимое:
 - Поскольку нельзя предугадать какие IP-адреса будут выданы инстансам в роли "внешних" (Плавающих)
 - реализуем возможность генерации автоматического инвентаря для Ansible средствами Terraform; Т.к. в конечном варианте ожидается запуск единого скрипта который развёрнёт инфраструктуру а затем её
 - настроит;

```
data "template_file" "inventory" {
    template = file("/home/altlinux/bin/_templates/inventory.tpl")
    vars = {
       user = "altlinux"
       webadm = join("", [openstack_compute_instance_v2.webadm.name, " ansible_host=",
openstack_networking_floatingip_v2.floatingip_webadm.address])
       web1 = join("", [openstack_compute_instance_v2.web1.name, " ansible_host=",
openstack_networking_floatingip_v2.floatingip_web1.address])
       web2 = join("", [openstack_compute_instance_v2.web2.name, " ansible_host=",
openstack_networking_floatingip_v2.floatingip_web2.address])
   }
}
resource "local_file" "save_inventory" {
  content = data.template_file.inventory.rendered
   filename = "/home/altlinux/bin/ansible/inventory"
}
```

• Создаём директорию "_templates" где будет хранить шаблоны:

mkdir _templates

• Создаём сам файл шаблона для автоматической генерации инвентаря для Ansible:

vim _templates/inventory.tpl

• Помещаем в него следующее содержимое:

 В результате Terraform должен будет генерировать инвентарь для Ansible по пути /home/altlinux/bin/ansible/inventory описанный в ini-формате;

```
${webadm}
${webal}
${web2}
[all:vars]
ansible_ssh_user = ${user}
ansible_ssh_extra_args = '-o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no'
ansible_python_interpreter = /usr/bin/python3
```

• Создаём директорию для Ansible:

mkdir ansible

Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply:
 о Результат:

Enter a value: yes
<pre>local_file.save_inventory: Creating local_file.save_inventory: Creation complete after 0s [id=1446cc2e1188e76e3f28823e7247b944c88e63fc]</pre>
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
Outputs:
LoadBalancer = "192.168.15.53" WEB1 = "192.168.15.176" WEB2 = "192.168.15.161" WEb20M = "192.168.15.99" [altlinux@controlvm bin]\$

• Проверяем наличие файла /home/altlinux/bin/ansible/inventory и его содержимое:



• Устанавливаем ansible:

sudo apt-get update && sudo apt-get install -y ansible

• Проверяем корректность сгенерированного инвентарного файла:

ansible -i ansible/inventory -m ping all

• Результат:

Ansible успешно может подключиться с ControlVM до всех инстансов для дальнейшей их настройки;



- В файл templates.tf добавляем следующий код:
 - Добавляя возможность Terraform автоматически генерировать конфигурационные файлы для WireGuard;
 - Поскольку нет информации о "внешнем" (Плавающим) IP-адресе который будет назначем WebADM при новом развёртывании проекта;
 - Он необходим для конфигурационного файла WireGuard для параметра Endpoint;

```
data "template_file" "web1_wg0" {
   template = file("/home/altlinux/bin/_templates/web1_wg0.tpl")
    vars = {
     floatingip_webadm = join("",[openstack_networking_floatingip_v2.floatingip_webadm.address])
   }
}
resource "local_file" "save_web1_wg0" {
  content = data.template_file.web1_wg0.rendered
  filename = "/home/altlinux/bin/ansible/wireguard/web1_wg0.conf"
}
data "template_file" "web2_wg0" {
   template = file("/home/altlinux/bin/_templates/web2_wg0.tpl")
   vars = {
     floatingip_webadm = join("",[openstack_networking_floatingip_v2.floatingip_webadm.address])
   }
}
resource "local file" "save web2 wg0" {
  content = data.template_file.web2_wg0.rendered
  filename = "/home/altlinux/bin/ansible/wireguard/web2_wg0.conf"
}
```

• Создаём сам файл шаблона конфигурационного файла туннельного соединения для инстанса WEB1:

vim _templates/web1_wg0.tpl

• Помещаем в него следующее содержимое:

```
[Interface]
Address = 10.20.30.2/29
PrivateKey = +0oP8y4QuTZL7P5Esr7YV7p/GqleqSTE5Mf6R1E6y38=
# WebAdm
[Peer]
PublicKey = Pn1bzrpZoBob/VeAvWDQfazwrZh18WarKSKzid2K1wc=
AllowedIPs = 10.20.30.0/29
PersistentKeepalive = 10
Endpoint = ${floatingip_webadm}:51820
```

• Создаём сам файл шаблона конфигурационного файла туннельного соединения для инстанса WEB2:

vim _templates/web2_wg0.tpl

• Помещаем в него следующее содержимое:

```
[Interface]
Address = 10.20.30.3/29
PrivateKey = UEN3Lbpm+NyXsw+XMEZPKHAaMocFRP0KfDZvIiu+BUk=
# WebAdm
```

[Peer]
PublicKey = Pn1bzrpZoBob/VeAvWDQfazwrZh18WarKSKzid2K1wc=
AllowedIPs = 10.20.30.0/29
PersistentKeepalive = 10
Endpoint = \${floatingip_webadm}:51820

• Создаём директорию для куда Terraform сохранит сгенерированные конфигурационные файлы:

mkdir ansible/wireguard

Проверяем конфигурацию и план terraform validate и terraform plan и запускаем развёртывание ресурсов terraform apply:
 Результат:

```
Enter a value: yes

local_file.save_web1_wg0: Creating...

local_file.save_web1_wg0: Creation complete after 0s [id=e48082e72e6a45f68c784df5a548a741909d4fd1]

local_file.save_web2_wg0: Creation complete after 0s [id=aae9a96c0222f4de114f5dc2b787ed93711b2127]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

LoadBalancer = "192.168.15.53"

WEB1 = "192.168.15.161"

WebADM = "192.168.15.161"

WebADM = "192.168.15.99"

[altlinux@control.wm bin]$
```

Проверяем наличие конфигурационных файлов для туннельных соединений и их содержимое:
 Должен корректно подставляться параметр Endpoint:



• Создаём конфигурационный файл туннельного соединения для инстанса WebADM:

vim ansible/wireguard/webadm_wg0.conf

- Помещаем в него следующее содержимое:
 - Т.к. с точки зрения WireGuard инстанс WebADM будет выступать сервером, то данный файл может быть статическим и не требуем какой-либо динамически получаемой информации;

```
[Interface]
Address = 10.20.30.1/29
ListenPort = 51820
PrivateKey = wHpaZtEfAYGxVXhwzUa80S1EbBXqrjDW8E1Z73pzTUI=
# WEB1
[Peer]
PublicKey = dZuyzPZVpJ4gyvMpJRC0I3PBGEIYe1x1MZuYyYHjfT0=
AllowedIPs = 10.20.30.2/32
# WEB2
[Peer]
PublicKey = ZSTT8CxQsxJ1KU9b05PoSG6iLFBb8hQtIc4Kyox4AnU=
AllowedIPs = 10.20.30.3/32
```

• Создаём playbook который будет реализовывать настройку VPN-соединения средствами WireGuard между WebADM и WEB1, WEB2:

vim ansible/wireguard_playbook.yml

• Помещаем в него следующее содержимое:

--- name: Install packages hosts: all

become: true

tasks:

 name: Install Wireguard community.general.apt_rpm: name:

 wireguard-tools
 wireguard-tools-wg-quick

- state: latest update_cache: true
- name: Settings WebAdm Wireguard server hosts: WebADM become: true

tasks:

 name: Create directory '/etc/wireguard' ansible.builtin.file: path: /etc/wireguard state: directory mode: '0755'

- name: Copy file 'wg0.conf' ansible.builtin.copy: src: wireguard/webadm_wg0.conf dest: /etc/wireguard/wg0.conf
- name: Started and enabled wg0 ansible.builtin.systemd: name: wg-quick@wg0 state: started enabled: true
- name: Settings WEB1 Wireguard client hosts: WEB1 become: true

tasks:

- name: Create directory '/etc/wireguard' ansible.builtin.file: path: /etc/wireguard state: directory mode: '0755'
- name: Copy file 'wg0.conf' ansible.builtin.copy: src: wireguard/web1_wg0.conf dest: /etc/wireguard/wg0.conf
- name: Started and enabled wg0 ansible.builtin.systemd: name: wg-quick@wg0 state: started enabled: true
- name: Settings WEB2 Wireguard client hosts: WEB2 become: true

tasks:

- name: Create directory '/etc/wireguard' ansible.builtin.file: path: /etc/wireguard state: directory mode: '0755'
- name: Copy file 'wg0.conf' ansible.builtin.copy: src: wireguard/web2_wg0.conf dest: /etc/wireguard/wg0.conf
- name: Started and enabled wg0 ansible.builtin.systemd: name: wg-quick@wg0 state: started enabled: true

• Устанавливаем коллекцию необходимую для работы модуля community.general.apt_rpm:

ansible-galaxy collection install community.general



• Запускаем playbook-сценарий для настройки VPN-туннеля:

ansible-playbook -i ansible/inventory ansible/wireguard_playbook.yml

• Результат:

[altlinux@controlvm bin]	1 ansible-p	laybook -i an	sible/inventory a	nsihle/wireg	uard_playbook	.yml	
PLAY [Install packages]	*******	********	************	*******		*******	********
TASK [Gathering Facts] * bk: [WebADM] ok: [WEB2] ok: [WEB1]	*******	******	******	********	*****	******	*******
TASK [Install Wireguard] changed: [WebADM] changed: [WEB2] changed: [WEB1]	*********		*************	*********		***********	*********
PLAY [Settings WebAdm Wi	reguard ser	ver] *******	***************	*********	***********	**********	*******
TASK [Gathering Facts] * ok: [WebADM]	*********	*********	*****	*********	*********	*********	*********
TASK [Create directory ' ok: [WebADM]	/etc/wiregu	ard'] ******		**********		*********	**********
TASK [Copy file 'wg0.cor changed: [WebADM]	f.] *******	********	********	*******	***********	********	********
TASK [Started and enable changed: [WebADM]	d wg0] ****	*******	*****	******	********	*****	*******
PLAY [Settings WEB1 Wire	guard clien	t] *********	******	******	******	******	********
TASK [Gathering Facts] * ok: [WEB1]	*******	*******	*****	******	******	*******	******
TASK [Create directory ' ok: [WEB1]	/etc/wiregu	ard'] *******	*****	******	*******	**********	*******
TASK [Copy file 'wg0.cor changed: [WEB1]	t.J ******	********	*****	*********	******	******	*********
TASK [Started and enable changed: [WEB1]	d wg0] ****	**********	**************	-****		********	
PLAY [Settings WEB2 Wire	guard clien	t] **********	******	*******	**********	*******	**********
TASK [Gathering Facts] * ok: [WEB2]	*******	*********	*************	*****	**********	**********	********
TASK [Create directory ' ok: [WEB2]	/etc/wiregu	ard'] ******		*******	******	******	******
TASK [Copy file 'wg0.cor changed: [WEB2]	1.] *******	*******	*****	*****	******	******	******
TASK [Started and enable changed: [WEB2]	d wg0] ****	********	******	*******	*********	*******	*********
PLAY RECAP							
NEB1 WEB2	: ok=5 ok=5	changed=3 changed=3 changed=2	unreachable=0 unreachable=0 unreachable=0	failed=0 failed=0 failed=0	skipped=0 skipped=0 skipped=0	rescued=0 rescued=0	ignored=0 ignored=0
		changeoes.	mis cachante-0	Halleu-V	skuppen-0	rescueu-v	rguot eu=0

ltlinux@controlvm bin]\$

• Проверяем наличие VPN-соединения:

• Подключаемся по SSH с ControlVM на WebADM по Плавающему IP-адресу;

• Смотрим туннельные соединения;

• Проверяем связность;



- Создаём playbook который будет реализовывать:
 - WebADM так, чтобы она могла подключаться по SSH с использованием пользователя altlinux и пароля «P@ssw0rd» к инстансам WEB1 и WEB2 с помощью VPN туннеля

vim ansible/ssh_playbook.yml

• Помещаем в него следующее содержимое:

```
- - -
- hosts: WEB1 WEB2
 become: true
 tasks:
   - name: Setting 'sshd_config' file
     ansible.builtin.lineinfile:
      line: "{{ item }}"
       path: /etc/openssh/sshd_config
       state: present
     with_items:
        - "PasswordAuthentication no"
        - "Match address 10.20.30.0/29"
        . .
              PasswordAuthentication yes"
   - name: Restarted sshd
      ansible.builtin.systemd:
       name: sshd
       state: restarted
 • Запускаем playbook-сценарий для настройки SSH по VPN-туннелю:
```

ansible-playbook -i ansible/inventory ansible/ssh_playbook.yml

• Результат:

[altlinux@controlvm_bin]\$ ansible-p	laybook -i an	sible/inventory a	insible/ssh_p	laybook.yml		
PLAY [WEB1 WEB2] ************************	*********	******	******	*******	******	*******
TASK [Gathering Facts] ************ ok: [WEB2] ok: [WEB1]	********	******	****	***********	*********	*******
<pre>TASK [Setting 'sshd_config' file] * ok: [WEB1] => (item=PasswordAuthent ok: [WEB2] => (item=PasswordAuthent changed: [WEB2] => (item=Match addr changed: [WEB1] => (item=Match addr changed: [WEB1] => (item= Passwo changed: [WEB2] => (item= Passwo</pre>	ication no) Ication no) ess 10.20.30. ess 10.20.30. rdAuthenticat rdAuthenticat	0/29) 0/29) ion yes) ion yes)	****	****	*****	*****
TASK [Restarted sshd] ************ changed: [WEB1] changed: [WEB2]	*******	*****	****	*********	******	*****
PLAY RECAP ***********************	*********	************	******	********		*********
WEB1 : 0k=3	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
WEB2 : ok=3	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
[altlinux@controlvm bin]\$						

- Проверяем:
 - Подключение с WebADM по туннельным IP-адресам на основе пароля:



• Также проверяем что подключение по SSH из внешних сете (не по туннелю) доступно только по ключу:



• Таким образом, на текущий момент получаем следующую структуру файлов и каталогов в рамках каталога /home/altlinux/bin на ControlVM:



Последнее изменение: среда, 15 января 2025, 16:18

Обратная связь

Подпишитесь

Вы используете гостевой доступ (Вход) Сводка хранения данных

Тема оформления сайта разработана conecti.me