Задание B5.2.8

Мы с вами рассмотрели установку ***Terraform****-*клиента, как объявлять и использовать переменные, установку провайдера, а также узнали, как найти нужный провайдер.

В этом задании вам нужно написать пример конфигурации, которая бы могла взаимодействовать с ***Yandex* *Cloud***. Необязательно создавать настоящую конфигурацию на стороне ***Yandex****-*провайдера.

Для реального взаимодействия с ***Yandex****-*провайдером необходимо предварительно создать аккаунт в ***Yandex* *Cloud***, создать ***service* *account*** и ***service\_account\_key\_file*** — это ключ, по которому вы будете проходить аутентификацию на стороне провайдера.

**Задание**:

* 1. Установить ***Terraform*** на локальный компьютер.
  2. Установить ***Terraform***-провайдер для работы с ***Yandex* *Cloud.***
  3. Указать фиксированную версию ***Terraform***-провайдера ***Yandex****-*провайдера.

Задание B5.3.7

В этом юните мы почерпнули множество новой информации, пора начать использовать эти знания. Давайте поработаем с реальной инфраструктурой и перенесем наши наработки к реальному провайдеру.

Продолжаем работу с заданием B5.2.8, где мы уже установили ***Terraform****-*клиент и описали работу с ***Yandex****-*провайдером.

Для выполнения заданий вам необходима подготовленная учетная запись в ***Yandex* *Cloud*** с возможностью создавать новые ресурсы.

В качестве задания нам нужно развернуть с помощью ***Terraform***новую виртуальную машину (инстанс в ***Yandex* *Cloud***), выдать внешний ***IP*** для доступа к содержимому виртуалки, чтобы зайти на веб-страницу на этой машине.

В качестве образа виртуальной машины будем использовать публичный образ ***LEMP***.

[***LEMP***](https://cloud.yandex.ru/docs/solutions/web/lamp-lemp)— это популярные наборы компонентов для развертывания веб-приложений и динамических сайтов. Этот образ в базовой конфигурации имеет страницу приветствия, которую мы и хотим увидеть в после применения нашего тестового задания.

В предыдущем юните мы рассмотрели, как работать с ресурсами, таким же способом мы создаем инстанс. В этом же юните мы узнали, как работать с источниками данных и как получать данные из уже подготовленных ресурсов.

После выполнения задания выполните удаление ресурсов, чтобы избежать дальнейших списаний.

**Важно**: не храните пароли и другие приватные данные в *Git*-репозитории.

**Задание**:

* 1. Получите *OAuth*-токен для доступа к [***Yandex Сloud***](https://cloud.yandex.com/en-ru/docs/iam/concepts/authorization/oauth-token)*.*
  2. Найдите ***cloud*\_*id*** и ***folder*\_*id*** в панели управления ***Yandex Сloud****.*
  3. Создайте инстанс (виртуальная машина в облаке) со следующими характеристиками:
     + ***CPU 2***;
     + ***RAM 2 GB****;*
     + образ — ***LEMP****;*
     + регион подсети — ***ru*-*central*1-*a****;*
     + остальные характеристики можете установить по своему усмотрению.
  4. Образ в инстанс нужно предоставить из связанного источника данных data yandex\_compute\_image.
  5. Для работы с инстансом вы должны ему выдать внешний ***IP***, для этого вам нужно создать ресурсы ниже и передать их значения так же, как и образ:
     + ***VPC* *network****,*
     + ***VPC* *subnet****.*
  6. Проверим, что все успешно поднялось, для этого:
     + зайдем по *SSH* на только что созданный инстанс;
     + откроем в браузере ***IP*** только что созданного инстанса, там должна быть страница приветствия ***nginx*** с текстом ***«Welcome to nginx!»***.

Задание B5.4.4

Продолжаем задание B5.3.7, в котором у нас уже существует реальная инфраструктура в ***Yandex Cloud***.

В этом юните мы увидели, как разместить файл состояния в удаленном хранилище на примере ***s3 bucket****.* Точно так же можно разместить ***Terraform* *state*** в любом другом разрешенном в документации хранилище.

В нашем задании нужно разместить файл состояния в *Yandex Cloud*-хранилище.

Чтобы разместить файл состояния в *Yandex Cloud*, предлагается использовать ***s3***-провайдер. Официальная документация к описанию этого провайдера прикреплена ниже.

Для начала вам нужно будет создать сам ***bucket*** (хранилище), его можно создать как через *web-*интерфейс ***Yandex Cloud***, так и с помощью ***Terraform*.** Мы будем создавать через ***Terraform***, но положить эти инструкции необходимо в отдельную директорию.

После этого вы сможете добавить инструкции в ***Terraform*** для хранения в ***Yandex Cloud***-хранилище.

**Задание**:

Необходимо поместить файл состояния в хранилище ***Yandex Cloud***, для этого:

* + 1. Создаём хранилище в отдельной директории:
       - Для создания хранилища вам необходимо создать [***static access key***](https://cloud.yandex.com/docs/iam/concepts/authorization/access-key) (***Access* *and* *Secret***).
       - При создании хранилища вам нужно указать ***access*\_*key*** и ***secret*\_*key***. Конечно как связанный ресурс ***Terraform***. Имя хранилища должно быть уникальное на весь ***Yandex Cloud***.   
         Для этого нам нужно:  
         - cоздать ***service*-*account****;*- предоставить права на запись для этого ***service*-*account***;- создать ключи доступа для этого ***service*-*account****;*- чтобы переиспользовать ***access*\_*key*** и ***secret*\_*key*** в нашем проекте, добавляем вывод этих значений в ***output*.**
    2. Добавляем в настройки задания B5.3.6 информацию о том, что файл состояния теперь должен хранится в удаленном хранилище:
       - Прописываем в настройках имя созданного хранилища.
       - Добавляем недавно созданные ключи ***access*\_*key*** и ***secret*\_*key*** в настройки доступа к хранилищу.

Если у вас возникли сложности с созданием ***service*-*account*** и ключей, то код решения можно взять [здесь](https://gist.github.com/kksudo/4f282188a686da9e95fcc3fe156d0367) и добавить туда создание вашего хранилища.

Задание B5.5.3

Продолжаем задание B5.4.4. В нем мы подняли хранилище, и попросили ***Terraform*** хранить состояние в удаленном хранилище.

В этом юните мы узнали о продвинутой концепции использования ***Terraform***, а именно, об использовании модулей.

Давайте применим эти знания на практике. Для этого мы создадим ещё один инстанс, а общие части вынесем в модуль, который сами создадим. Таким образом, в основной ***Terraform***-инструкции инстансы у нас будут создаваться через модуль, который мы напишем сами.

Обратите внимание на ошибки, которые могут возникнуть в процессе вынесения кода в модуль:

* + ***cidr* *VPC* *subnet*** не должен дублироваться;
  + ***cidr* *VPC* *subnet* *name*** также не должен повторяться, если используется одна сеть ***VPC***.

**Задание:**

* 1. Добавить еще один инстанс в ***Yandex* *Cloud*** с теми же характеристиками, за исключением нескольких моментов:
     + - образ нужно использовать ***LAMP***;
       - регион подсети указать ***ru-central1-b***.
  2. Вынести повторяющийся код для создания инстансов в модуль. Инстансы должны находиться в одной ***VPC****-*сети, но в разных подсетях и разных зонах.
  3. Переписать инфраструктурный код, чтобы для создания инстансов использовался созданный нами модуль.
  4. Модуль разместить в локальной директории рядом с основным проектом.

Задание B5.6.7

Продолжаем задание B5.5.3.

Теперь нам нужно применить лучшие практики для работы с ***Terraform***, если мы не сделали этого ранее.  
В качестве зависимостей к утилите ***pre*-*commit*-*Terraform*** вам потребуется установить:

* + [***Terraform-docs***](https://github.com/SkillfactoryCoding/DEVOPS-Infrastructure-terraform-docs)  
    Для того чтобы появилась документация, вам нужно создать файл и добавить туда это содержимое:

<!-- BEGINNING OF PRE-COMMIT-TERRAFORM DOCS HOOK -->

<!-- END OF PRE-COMMIT-TERRAFORM DOCS HOOK -->

* + [***tflint***](https://github.com/SkillfactoryCoding/DEVOPS-Infrastructure-tflint)  
    [Правила](https://github.com/SkillfactoryCoding/DEVOPS-Infrastructure-tflint/tree/master/docs/rules), которые можно включить дополнительно в проверке. Выглядеть подобный конфиг может таким образом:
  + repos:
  + — repo: git://github.com/antonbabenko/pre-commit-Terraform
  + rev: v1.50.0
  + hooks:
  + — id: Terraform\_tflint
  + args:
  + — '--args=--only=Terraform\_deprecated\_interpolation'
  + — '--args=--only=Terraform\_deprecated\_index'
  + [***pre-commit***](https://pre-commit.com/#install)

**Задание:**

* + 1. Добавьте версионирование используемых провайдеров/других ресурсов, если не сделали этого ранее.
    2. Настройте на локальном компьютере [*pre-commit-Terraform*](https://github.com/SkillfactoryCoding/DEVOPS-infrastructure-pre-commit-terraform) вместе с [*Tflint*](https://github.com/SkillfactoryCoding/DEVOPS-Infrastructure-tflint)*.*В процессе настройки обратите внимание на нюансы:
       - * Ваш проект должен быть в гит-репозитории.
         * Должен быть создан конфиг для *pre*-*commit*-*Terraform.*В конфиге должны присутствовать такие проверки:

           - terraform\_fmt  
           - terraform\_validate  
           - terraform\_docs   
           - terraform\_tflint

* + 1. Проверьте проект с помощью установленного линтера. Можно запустить ***pre-commit-Terraform*** таким образом: pre-commit run -a.

В результате выполнения этого задания вы получите больше опыта взаимодействия с ***Terraform***-линтерами, важно попробовать «на вкус» работу линтеров, чтобы это взаимодействие стало более прозрачным.

Практикуйтесь и изучайте дополнительные материалы, приведенные в юните и за его пределами. :)

Продолжаем наше задание B5.6.7.

В этом задании мы сделаем наши инстансы более гибкими за счет масштабирования, для этого мы должны использовать ***Load* *Balancer***. Подготовим ***Load* *Balancer*** и необходимые связанные ресурсы для работы с существующими инстансами.

В конечном итоге у нас должна быть возможность горизонтально масштабировать инстансы, находящиеся за ***Load* *Balancer***.

[**Сетевые балансировщики (*Load Balancer*)**](https://cloud.yandex.ru/docs/network-load-balancer/concepts/) равномерно распределяют нагрузку по облачным ресурсам и отслеживают их состояние. Это позволяет повысить доступность и отказоустойчивость ваших приложений и облачной сетевой инфраструктуры.

При первом запросе будем получать ответ от первого инстанса, при одном из последующих мы должны получить ответ от другого инстанса. В одном случае мы увидим страницу приветствия ***nginx***(нужно вставить в браузер ***IP*** от созданного инстанса, который мы получили в ***output*** нашей ***Terraform***-конфигурации), в другом — ***Apache***.

**Задание**:

* 1. Добавить в проект ***Network* *Load* *Balancer*** таким образом, чтобы за ним были созданные нами инстансы.
  2. Трафик должен делиться поровну между инстансами.
  3. Доступ к инстансам должен быть возможен как через ***IP* *Load* *Balancer***, так и через их публичные ***IP****.*