

HashiCorp

Terraform

Начинайте использовать, если еще не.

План вебинара

Какие задачи решает Terraform

Как Terraform это делает

Terraform demo

Где и как учить Terraform дальше

Между блоками будут короткие QA сессии



DevOps coach



Сергей Василенко
[LinkedIn](#) | [Technical Blog](#)



Platform Software Engineer

Инфраструктура



На проекте (обычно) больше, чем одно окружение: dev, staging, production...

Инфраструктура

 На проекте (обычно) больше, чем одно окружение: dev, staging, production...

 Время жизни у разных окружений – разное

 Разные окружения отличаются по конфигурации ресурсов (CPU, RAM, Storage)

Инфраструктура

🕒 На проекте (обычно) больше, чем одно окружение: dev, staging, production...

💰 Время жизни у разных окружений – разное

💰 Разные окружения отличаются по конфигурации ресурсов (CPU, RAM, Storage)

😞 Разные окружения не должны отличаться по функциональным элементам

😞 Количество функциональных элементов большое, ими нужно управлять

Инфраструктура

Вспомните:

- Сколько ресурсов вам пришлось настроить для запуска приложения?
- Сколько времени вы потратили на это?
- Было ли ощущение “мы что-то забыли” в процессе?

А вы могли бы без ошибок повторить эту настройку?

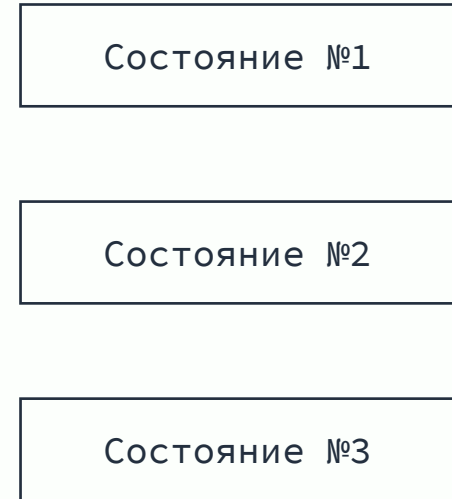
А если я попрошу вас это сделать несколько раз?

Инфраструктура — модели управления

Mutable



Immutable



Инфраструктура **как Код** (IaC)

Императивный стиль

Описание **действий**, которые приводят к конечному результату:

- Действие "А"
- Действие "Б"
- Действие "В"
- ...
- ... Шаг за шагом

Декларативный стиль

Описание конечного **результата**:

- Параметр "А" = значение X
- Параметр "Б" = значение Y
- Параметр "В" = значение Z
- ...
- ... Желаемое состояние конфигурации

Инфраструктура **как Код** (IaC)

Императивная модель

```
$ aws ec2 run-instances \  
--image-id "ami-03d315ad33b9d49c4" \  
--count 1 --instance-type t3.nano
```

```
$ aws ec2 run-instances \  
--image-id "ami-03d315ad33b9d49c4" \  
--count 2 --instance-type t3.nano
```

Декларативная модель

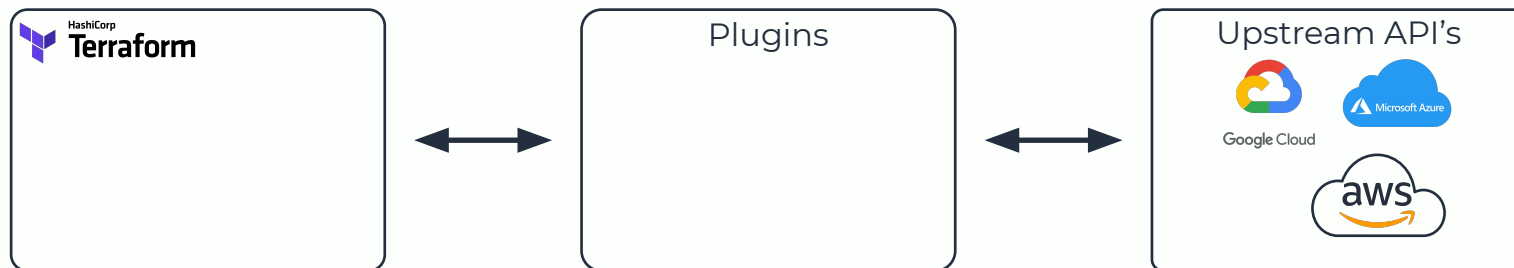
```
resource "aws_instance" "example" {  
    count          = 1  
    ami            = "ami-03d315ad33b9d49c4"  
    instance_type = "t3.nano"  
}
```

```
resource "aws_instance" "example" {  
    count          = 2  
    ami            = "ami-03d315ad33b9d49c4"  
    instance_type = "t3.nano"  
}
```

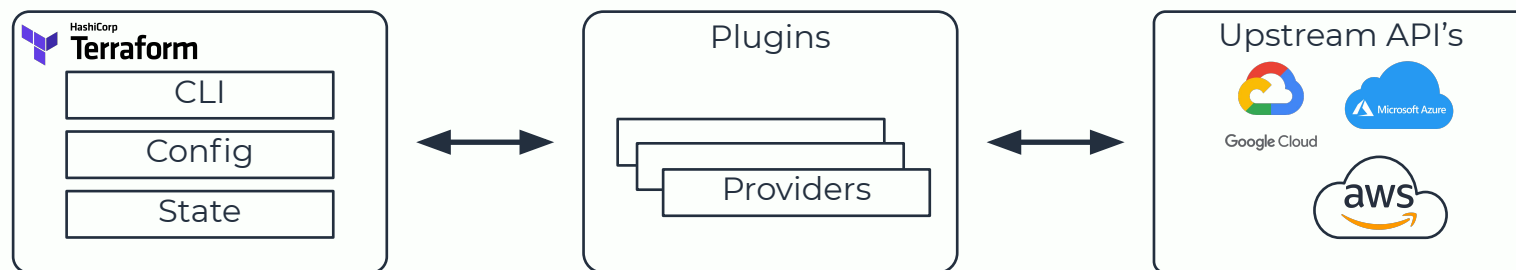
Короткая Q/A сессия



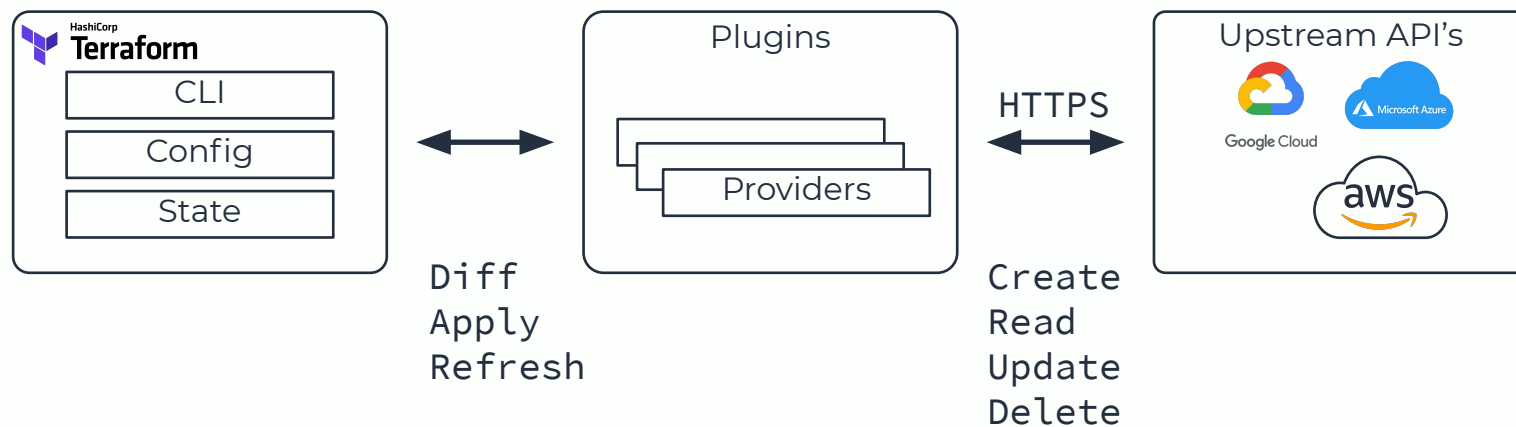
Terraform: как устроен и как работает



Terraform: как устроен и как работает



Terraform: как устроен и как работает



Короткая Q/A сессия



Terraform module

Terraform module — набор конфигурационных файлов (*.tf) в директории

Конфигурационный файл — набор блоков и аргументов; HCL синтаксис

Terraform module

Terraform module — набор конфигурационных файлов (*.tf) в директории

Конфигурационный файл — набор блоков и аргументов; HCL синтаксис

Пример:

```
resource "aws_vpc" "main" {  
  cidr_block = var.base_cidr_block  
}
```

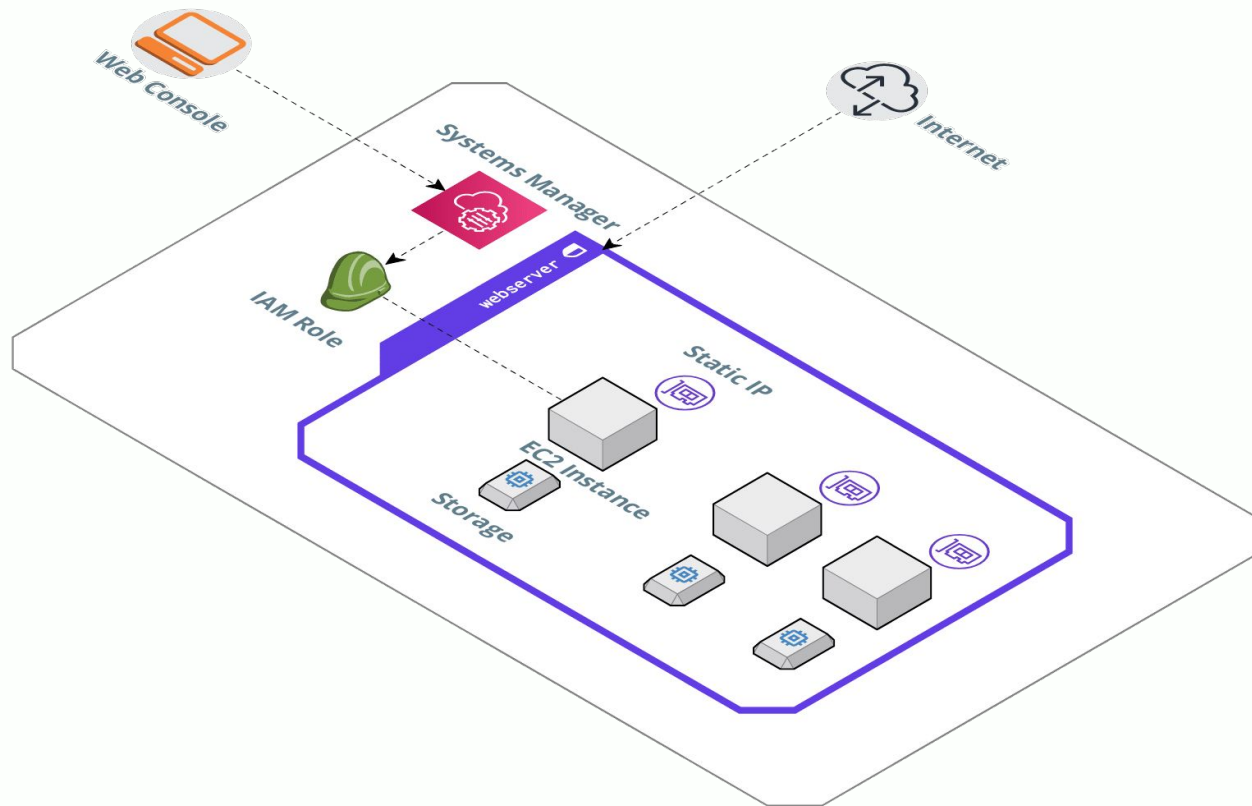
```
<BLOCK TYPE> "<BLOCK LABEL>" "<BLOCK LABEL>" {  
  # Block body  
  <IDENTIFIER> = <EXPRESSION> # Argument  
}
```

Имя аргумента

Значение аргумента





} Блок

Инфраструктура для Demo



Screen sharing Demo

Почему стоит использовать Terraform

-  Скорость и (условная) простота эксплуатации
-  Стандартизация конфигурации окружений
-  Минимизация рисков — человеческих ошибок
-  Интеграция с инструментами развертывания и прочей автоматизацией

Где и как учиться дальше

Курсы

Портал HashiCorp Learn:

- [Write Terraform Configuration](#)
- [Reuse Configuration with Modules](#)
- [Provision Infrastructure Deployed with Terraform](#)
- [Manage Terraform State](#)

- [Study Guide - Terraform Associate Certification](#)

Вдохновляющие видео на канале [HashiCorp YouTube](#)

Практика, практика, практика!

Q/A сессия

