

ПРОДВИНУТЫЙ ИНТЕНСИВ

# Terraform МЕГА

СЛЕДЯЩИЙ + Yandex Cloud

## Внутреннее устройство

Павел Селиванов



# План

## State

- Зачем нужен state
- Структура
- Хранение
- Секретные данные

## Выстраивание зависимостей

## Провайдеры

# Terraform state

Файл в формате JSON, в котором  
Terraform хранит связь между  
**описанными** и реально **созданными**  
объектами



# Terraform State

```
"mode": "managed",  
  "type": "yandex_vpc_network",  
  "name": "this",  
  "instances": [  
    {  
      "id": "enpkjvob52vucbju3hb1",  
      .....  
    }  
  ]  
}
```



# Terraform State

В стейте хранятся зависимости между объектами

```
{
  "type": "yandex_vpc_subnet",
  "name": "this",
  "instances": [
    {
      "dependencies": [
        "yandex_vpc_network.this"
      ]
    }
  ]
}
```



# Terraform state

При выполнении операций Terraform  
выполняет обновление стейта из  
реальной инфраструктуры

# Terraform state

При выполнении операций Terraform  
выполняет обновление стейта из  
реальной инфраструктуры

Это может быть медленно  
на больших инфраструктурах,  
поэтому есть флаг *-refresh=false*

# Remote State

- Хранить локально – плохая идея!
- Для удаленного хранения есть разные бэкенды
  - Обычно это S3, но нужен механизм локов
- Хорошая альтернатива – Kubernetes backend

```
terraform {  
  backend "kubernetes" {  
    secret_suffix = "state"  
    config_path   = "~/.kube/config"  
  }  
}
```



# Шеф, все пропало!

```
$ terraform import \  
    <resource type>.<resource name> \  
    <resource id from cloud>
```



# Важно!

Terraform подразумевает  
маппинг объектов в стейте и  
конфигурации 1-to-1!



## Sensitive data

```
variable "db_password" {  
  type      = string  
  sensitive = true  
}
```

----- скрывать вывод переменной

```
output "db_password" {  
  value = var.db_password  
}
```

**Error:** Output refers to sensitive values

# Sensitive data

Передавать секретные данные нужно  
через переменные окружения

```
$ TF_VAR_<variable name>=<value>
```

# Sensitive data

State все равно хранит в  
открытом виде, поэтому  
**state = sensitive data**



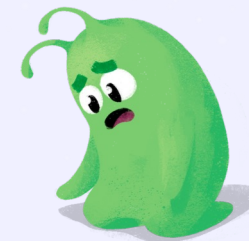
# Зависимости

Terraform самостоятельно определяет зависимости между объектами, ориентируясь на ссылки между объектами

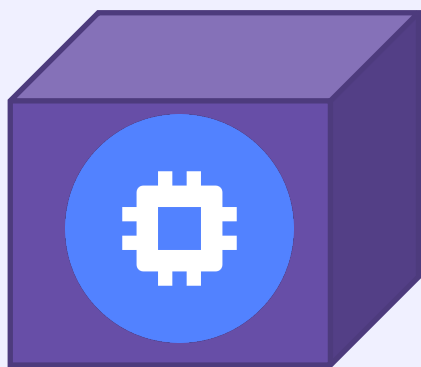
# Зависимости

Terraform самостоятельно определяет зависимости между объектами, ориентируясь на ссылки между объектами

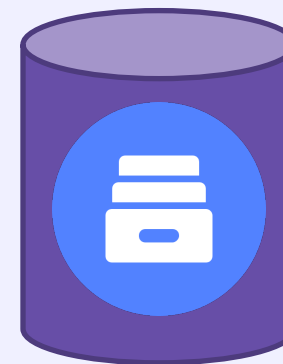
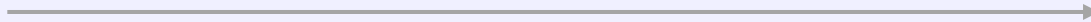
Но это может работать не всегда!



# Depends\_on



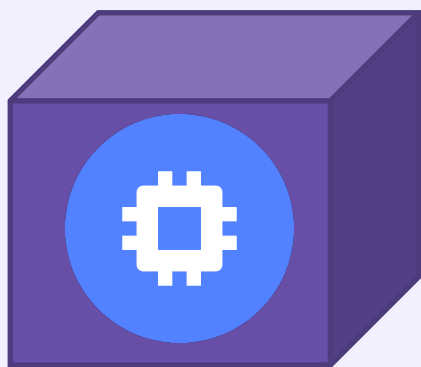
**Application instance**



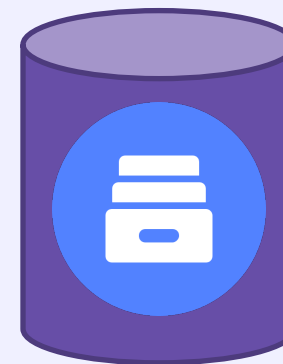
**S3 bucket**



# Depends\_on



Application instance



S3 bucket

```
depends_on =  
[yandex_storage_bucket.this] }
```

# Graph

Позволяет сгенерировать визуальный граф объектов

```
$ terraform graph | dot -Tpng > png.png
```



<http://www.graphviz.org/>



# Провайдеры

- Соединительный мост между Terraform и API «провайдера»
- Не занимаются магией, просто вызывают методы API
- Бинарники, написанные на Go с использованием стандартных библиотек Hashicorp
- <http://learn.hashicorp.com/collections/terraform/providers>



ПРОДВИНУТЫЙ ИНТЕНСИВ

# Terraform МЕГА

СЛЕДРМ + Yandex Cloud

■ Спасибо!

