Infrastructure as Code

- Terraform
  - Terraform Top Commands
  - Terraform HandBook
  - Import JSON Code from API+jq
- Vagrant
  - Vagrant HandBook

Terraform

Terraform

Terraform Top Commands

- **Terraform CLI Guide**

Инициализация рабочего каталога

```
terraform init
```

Валидация конфигурационных файлов

```
terraform validate
```

Просмотр плана выполнения

```
terraform plan
```

Применение изменений (авто подтверждение)

```
terraform apply -auto-approve
```

Уничтожение всей управляемой инфраструктуры (авто подтверждение)

```
terraform destroy -auto-approve
```

Поиск ssh-ключа по NAME

```
curl -X GET \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <do_key>' \
  'https://api.digitalocean.com/v2/account/keys' \
  -G \
  --data-urlencode 'page=1' \
  --data-urlencode 'per_page=999' \
  | jq '.ssh_keys[] | select(.name=="xeonmp22_key1") | .id'
```

Удаление ключа оп ID

```
curl -X DELETE \
  -H 'Authorization: Bearer <do_key>' \
  "https://api.digitalocean.com/v2/account/keys/<key_id>"
```

Создание/Удаление заданного ресурса

```
terraform apply -target=digitalocean_droplet.web2 -auto-approve
terraform apply -target=aws_route53_record.www2 -auto-approve
terraform destroy -target=digitalocean_droplet.web2 -auto-approve
```

Terraform HandBook

- **Установка Terraform**

- **Terraform Guides: Official +Yandex.Cloud**

> **Terraform Providers: Official +Github +Yandex.Cloud**

PROVIDER	API
GitLab Repo	GitLab API
Yandex Cloud	Yandex API
Digital Ocean_droplet	Digital Ocean_API
Vscale_compute instance **+Official**	Vscale_API
Hetzner Cloud server	Hetzner Cloud_API
AWS_instance AWS_Route53_record AWS_Route53_zone	HTTP API + API Gatewa

> **Источник developer.hashicorp.com**

- Environment variables
- Variables
- External provider
- Functions
- Count
- Output Values

- Interpolation Syntax
- Provisioner remote-exec
- Provisioner local-exec

Другие источники

- Google Load Balancers
- Import JSON Code from API+jq
- Утилита curl
- Утилита jq

Terraform

Import JSON Code from API+jq

??????? ?????? ??????? ?????

> В обратных кавычках Curl запрос с отфильтрованным fingerprint-ом ключа

```
sudo tee ~/key_convert << 'EOF'
#!/bin/bash
echo "{
\"fingerprint\": `curl -X GET \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <key>' \
  'https://api.digitalocean.com/v2/account/keys?page=1&per_page=999' \
  | jq '.ssh_keys[] | select(.name=="REBRAIN.SSH.PUB.KEY") | .fingerprint'`
}"
EOF
```

?????????? ???? ? ???????????? terraform

> Функция **lookup** извлекает значение указанного ключа из набора

```
data "external" "key_conv" {
  program = ["bash", "${path.root}/key_convert"]
}

resource "digitalocean_droplet" "ubuntu01" {
  ***
  ssh_keys = [
    "${lookup(data.external.key_conv.result, "fingerprint", "what?")}"
  ]
}
```

?????? jq-???????? ?????? ??????

Выгрузка значения fingerprint конкретного ключа по его name:

```
curl -X GET \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <key>' \
  'https://api.digitalocean.com/v2/account/keys?page=1&per_page=999' \
  | jq '.ssh_keys[] | select(.name=="REBRAIN.SSH.PUB.KEY") | .fingerprint'
```

Выгрузка значений name всех ключей:

```
curl -X GET \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <key>' \
  'https://api.digitalocean.com/v2/account/keys?page=1&per_page=999' \
  | jq '.ssh_keys[] | .name'
```

Vagrant

Vagrant HandBook

????????? Vagrnat

Digital Ocean Droplet Example

Предварительная настройка Plugins/BOX

```
vagrant plugin install vagrant-host-shell
vagrant plugin install vagrant-digitalocean
vagrant box add digital_ocean https://github.com/devopsgroup-io/vagrant-
digitalocean/raw/master/box/digital_ocean.box
```

Vagrantfile

```
Vagrant.configure('2') do |config|
  config.vm.define "droplet1" do |config|
      config.vm.provider :digital_ocean do |provider, override|
        override.ssh.private_key_path = '~/.ssh/id_rsa'
        override.vm.box = 'digital_ocean'
        override.vm.box_url = "https://github.com/devopsgroup-io/vagrant-
digitalocean/raw/master/box/digital_ocean.box"
        override.nfs.functional = false
        override.vm.allowed_synced_folder_types = :rsync
        provider.token = 'YOUR TOKEN'
        provider.image = 'ubuntu-18-04-x64'
        provider.region = 'nyc1'
        provider.size = 's-1vcpu-1gb'
```

```
        provider.backups_enabled = false
        provider.private_networking = false
        provider.ipv6 = false
        provider.monitoring = false
      end
  end
```

Vagrant Top Commands

Режимы работы Vagrant:

```
vagrant up
vagrant destroy
vagrant ssh
vagrant status
```

Удаление кэша старых машин:

```
vagrant global-status --prune
```

Запуска/rebuild машин с LOGированием:

```
VAGRANT_LOG=info vagrant up
VAGRANT_LOG=info vagrant rebuild
```