

Terraform

- [Terraform Top Commands](#)
- [Terraform HandBook](#)
- [Import JSON Code from API+jq](#)

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 по ID

```
curl -X DELETE \  
  -H 'Authorization: Bearer <do_key>' \  
  <id>
```

```
"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	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](#)

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'
```