* Для того что бы исключить проблем с выгрузкой в GitHub конфиденциальных данных, необходимо в корне репозитория создать файл .gitignore, в нем указываются файлы которые мы хотим игнорировать для выгрузки в репозиторий, это обычно те файлы в которых хранятся конфиденциальные данные.

В нашем случае при работе с terraform, мы можем создать такой .gitignore:

- исключаем локальные файлы .terraform

- исключаем файлы tfstate

- исключаем файлы журнала сбоев crash.log

- исключаем все файлы .tfvars, которые могут содержать конфиденциальные данные, такие как пароль, закрытые ключи и другие секреты. Они не должны быть частью версии контролировать, поскольку они являются точками данных, которые потенциально конфиденциальны и подвержены изменяться в зависимости от окружения. \*.tfvars \*.tfvars.json

- игнорировать файлы переопределения, так как они обычно используются для локального переопределения ресурсов override.tf

- исключаем файлы tfplan которые в себе имеют вывод плана инфраструктуры

- исключить файлы конфигурации CLI terraform.rc

- исключить файл key.json который в себе имеет данные для сервисного аккаунта, закрытый и публичный ключ, и так же в каком облаке работаем.

- исключаем файл providers.tf который содержит данные для подключения к облаку, личный токен, зону, folder\_id, cloud\_id

- так же исключим cloud-init.yaml в котором мы передаем данные сервисного аккаунта для ВМ, и публичный ключ.

В нашем случае мы получаем такой .gitignore:

\*\*/.terraform/\*

\*.tfstate

\*.tfstate.\*

crash.log

crash.\*.log

\*.tfvars

\*.tfvars.json

override.tf

override.tf.json

\*\_override.tf

\*\_override.tf.json

.terraformrc

terraform.rc

key.json

providers.tf

\*.lock.tf

\*.lock.hcl

cloud-init.yaml

* Так же можем использовать iam token, для подключения сервисного аккаунта, iam token это, уникальная последовательность символов, которая выдается пользователю после прохождения аутентификации. С помощью этого токена пользователь авторизуется в api yandex cloud и выполняет операции с ресурсами.

## Получить IAM-токен можем с помощью CLI

yc iam key create --service-account-name my-robot --output key.json

создаем профиль который будет использоваться для выполнения операций от имени сервисного аккаунта

yc config profile create my-robot-profile

укажем в конфигурации профиля авторизованный ключ сервисного аккаунта

yc config set service-account-key key.json

Теперь можем получить IAM-токен для сервисного аккаунта:

yc iam create-token

После получения токена, мы его заносим в providers.tf token

* В дальнейшем что бы, не было утечки данных, можем использовать ПО checkov, snyk.