Importing Pre-Existing Infrastructure

With the latest version of Terraform (v1.5.0) it is possible to generate terraform from existing infrastructure in some cases.

Here I give an example of importing Okta groups but this is available in a lot of other providers like AWS and GCP.

IMPORT BLOCK

The two pieces of information required are the id of the resource being created ie a GCP Bucket or VM and the name you wish to call your resource in terraform.
The type of ID that is required can be located in the import section of the Terraform documentation.
Terraform Registry
It may not always be obvious and may need some investigation. In the case below, the API shows the ID yet the GUI does not.

With this available you can create a file with the import block
I created a file named import.tf

```
import {
   to = resource.name_you_want
   id = "the id of resource"
  }
```

Here is an example of mine for creating code for a rule from okta

```
import {
   to = okta_group_rule.commercial_growth_to_confluence_commercial
   id = "0prbov8r7r3ezIBpS697"
  }
```

GENERATE TERRAFORM

Now we wish to generate the terraform code

```
terraform plan -generate-config-out=<outputfilename.tf>
```

example

```
terraform plan -generate-config-out=okta_rules.tf
```

This will create a file `okta_rules_import.tf`. This is done by terraform accessing your infrastructure and copying this to a terraform file.

As the code will inform, this code needs checking and may create information not need or you were unaware of.
It will also need adapting to ensure those pieces that need to reply on another ie a user and role(s), will need adapting to point in the correct place.

This will allow you to create the infrastructure that is of known working manner from your infrastructure and is a great

help.

Once all the infrastructure is created, you should be able to commit your code for review BUT at this point the terraform will appear to need to create all of this infrastructure and so you will need to import the infrastructure to state so that is is being controlled by terraform.

This should be the last thing you do and make others aware as they may find an anomaly if they are creating plans at the same time as their code will not yet have the terraform yet the state believes that it should and as such will believe that the terraform has been removed and highlight it is to be destroyed.

```
terraform import resource.chosenname "ID"
```

example

```
terraform import  okta_group_rule.commercial_growth_to_confluence_commercial "0prbov8r7r3ezIBpS697"
```

You should now have code that creates `0` new resources (there maybe subtle changes ie labels).
Now you can create a merge request.

Quick Guide

```
#####################################
####    Create an import Block    ####
####         In a tf file         ####
#####################################
I found the resource id with
curl -X GET https://raftai.okta.com/api/v1/groups/rules -H 'Authorization: SSWS <API KEY> ' | jq -r '.
[]|.name,.id'

import {
   to = okta_group_rule.commercial_growth_to_confluence_commercial
   id = "0prbov8r7r3ezIByS697"
  }

########################################
####    Import the Resource Block    ####
########################################
Run on CLI
terraform plan -generate-config-out=okta_rules_import.tf

########################################
####          Commit Code            ####
########################################

Agree import

############################################
####    Import The Resource into State    ####
############################################
terraform import  okta_group_rule.commercial_growth_to_confluence_commercial "0prbov8r7r3ezIBpS697"

```

```
Check no extras being created and merge request
Merge

Bingo
```