Runbook for Terraform Modules

Google Cloud Platform

Date: 10/7/2021

Author: Veera Vedantham

Prepared for: Google Cloud Platform

Document type: Run Book

Contents

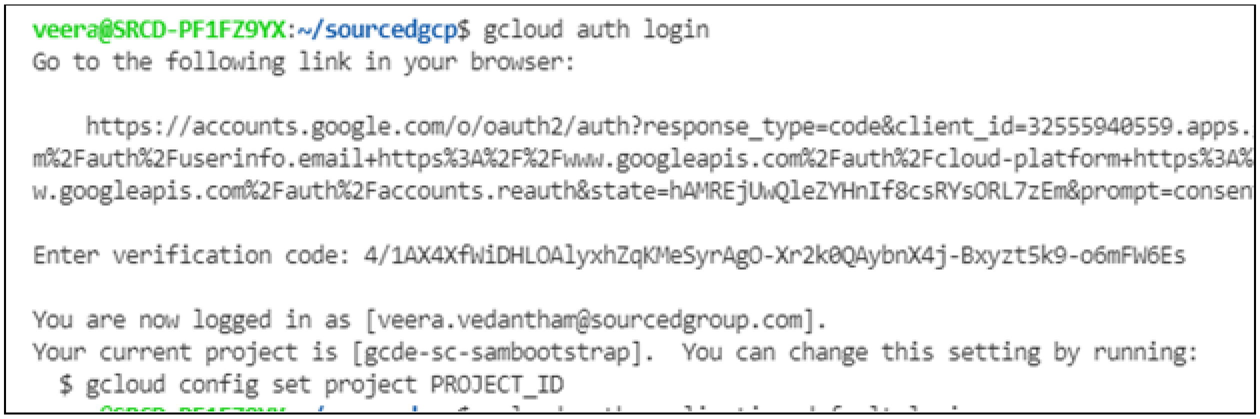
Prerequisites

- Linux Machine or Windows Subsystem for Linux (WSL)
- Terraform 0.14.10
- Terragrunt v0.31.0
- Google Cloud SDK 349.0.0
- Code editor e.g., VS Code

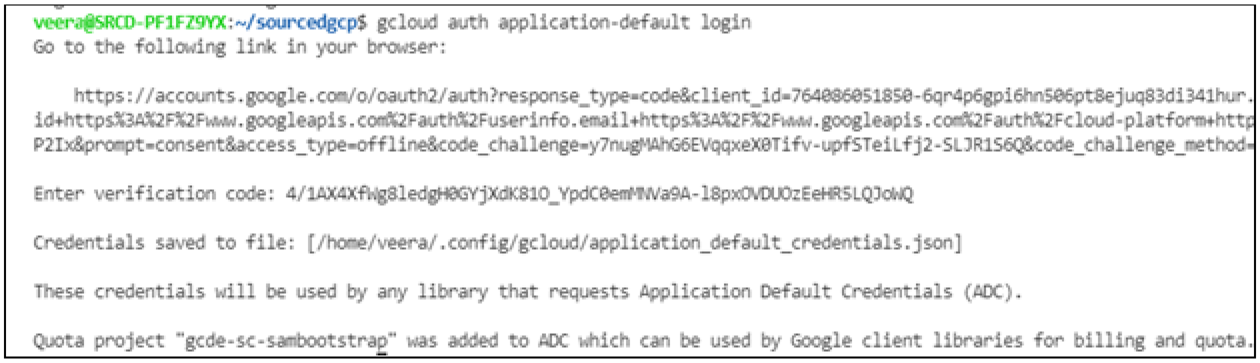
Steps to make changes to terraform modules

1. Login to Google cloud using both the below commands

   # gcloud auth login - Login to the cloud SDK through user account
   # gcloud auth application-default login - Application Default Credentials (ADC) set for Google API calls



```
veera@SRCD-PF1FZ9YX:~/sourcedgcp$ gcloud auth login
Go to the following link in your browser:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.
m%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%
w.googleapis.com%2Fauth%2Faccounts.reauth&state=hAMREjUwQleZYHnIf8csRYsORL7zEm&prompt=consen

Enter verification code: 4/1AX4XfWiDHLOAlyxhZqKMeSyrAgO-Xr2k0QAybnX4j-Bxyzt5k9-o6mFW6Es

You are now logged in as [veera.vedantham@sourcedgroup.com].
Your current project is [gcde-sc-sambootstrap].  You can change this setting by running:
  $ gcloud config set project PROJECT_ID
```



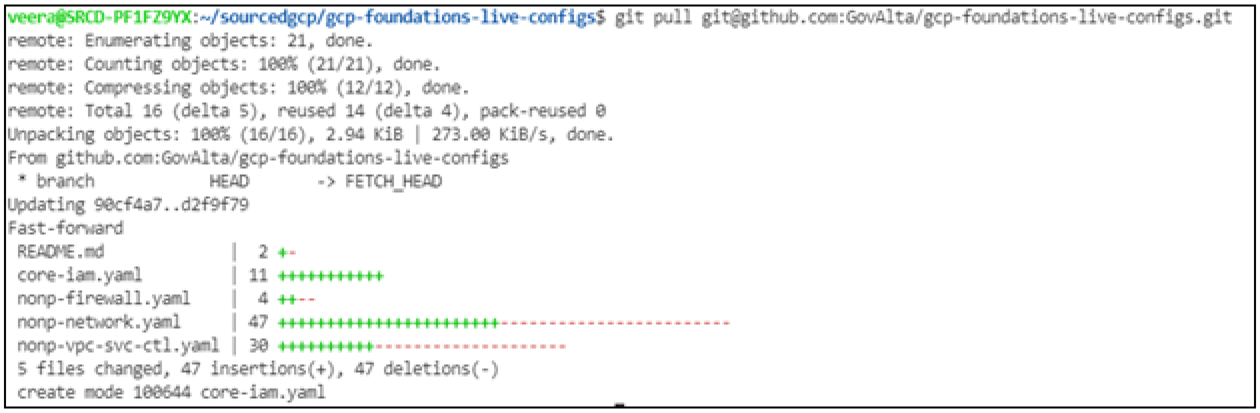
```
veera@SRCD-PF1FZ9YX:~/sourcedgcp$ gcloud auth application-default login
Go to the following link in your browser:

    https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=764086051850-6qr4p6gpi6hn506pt8ejuq83di341hur.
id+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+http
P2Ix&prompt=consent&access_type=offline&code_challenge=y7nugMAhG6EVqqxeX0Tifv-upfSTeiLfj2-SLJR1S6Q&code_challenge_method=

Enter verification code: 4/1AX4XfWg8ledgH0GYjXdK810_YpdC0emMNVa9A-l8pxOVDUOzEeHR5LQJoWQ

Credentials saved to file: [/home/veera/.config/gcloud/application_default_credentials.json]

These credentials will be used by any library that requests Application Default Credentials (ADC).

Quota project "gcde-sc-sambootstrap" was added to ADC which can be used by Google client libraries for billing and quota.
```

2. Clone all the repositories from GitHub to the local machine using the *git clone* command.

   # git clone git@github.com:GovAlta/terraform-gcp-guardrails.git

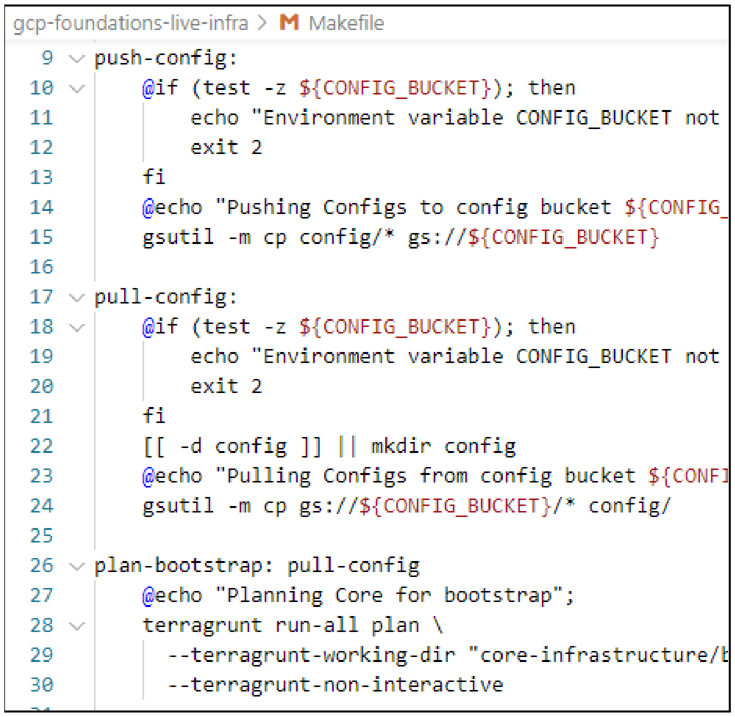
3. If making changes to an existing repository, pull the latest code from GitHub to the corresponding local directory

   # git pull git@github.com:GovAlta/gcp-foundations-live-configs.git



```
veera@SRCD-PF1FZ9YX:~/sourcedgcp/gcp-foundations-live-configs$ git pull git@github.com:GovAlta/gcp-foundations-live-configs.git
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 16 (delta 5), reused 14 (delta 4), pack-reused 0
Unpacking objects: 100% (16/16), 2.94 KiB | 273.00 KiB/s, done.
From github.com:GovAlta/gcp-foundations-live-configs
 * branch            HEAD       -> FETCH_HEAD
Updating 90cf4a7..d2f9f79
Fast-forward
 README.md            |  2 +-
 core-iam.yaml        | 11 +++++++++++
 nonp-firewall.yaml   |  4 ++--
 nonp-network.yaml    | 47 +++++++++++++++++++++++------------------------
 nonp-vpc-svc-ctl.yaml | 30 ++++++++++--------------------
 5 files changed, 47 insertions(+), 47 deletions(-)
 create mode 100644 core-iam.yaml
```

4. Switch to *gcp-foundations-live-infra* repository to run all *make <target>* commands. The list of all make <target> commands to be run are configured in *MakeFile* of this repository. For example, *make push-config* pushes new config file changes into cloud storage buckets.

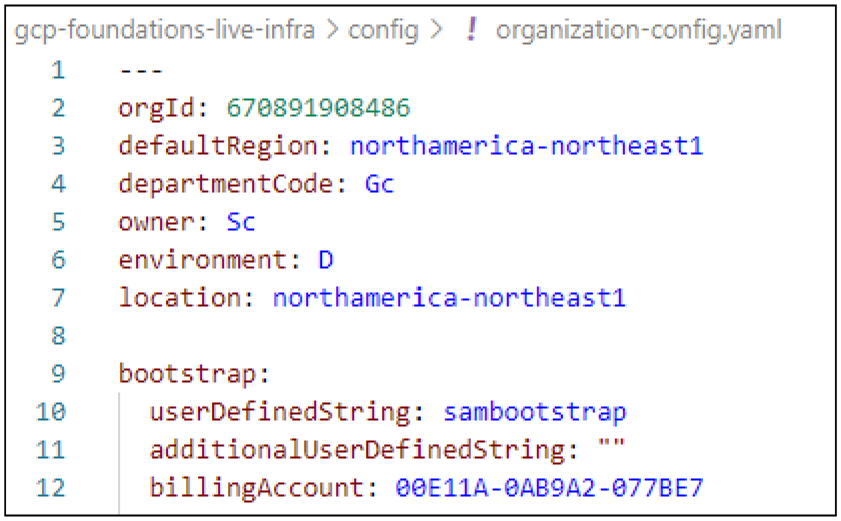


gcp-foundations-live-infra > Makefile

```
push-config:
    @if (test -z ${CONFIG_BUCKET}); then
        echo "Environment variable CONFIG_BUCKET not
        exit 2
    fi
    @echo "Pushing Configs to config bucket ${CONFIG_
    gsutil -m cp config/* gs://${CONFIG_BUCKET}

pull-config:
    @if (test -z ${CONFIG_BUCKET}); then
        echo "Environment variable CONFIG_BUCKET not
        exit 2
    fi
    [[ -d config ]] || mkdir config
    @echo "Pulling Configs from config bucket ${CONFI
    gsutil -m cp gs://${CONFIG_BUCKET}/* config/

plan-bootstrap: pull-config
    @echo "Planning Core for bootstrap";
    terragrunt run-all plan \
      --terragrunt-working-dir "core-infrastructure/b
      --terragrunt-non-interactive
```

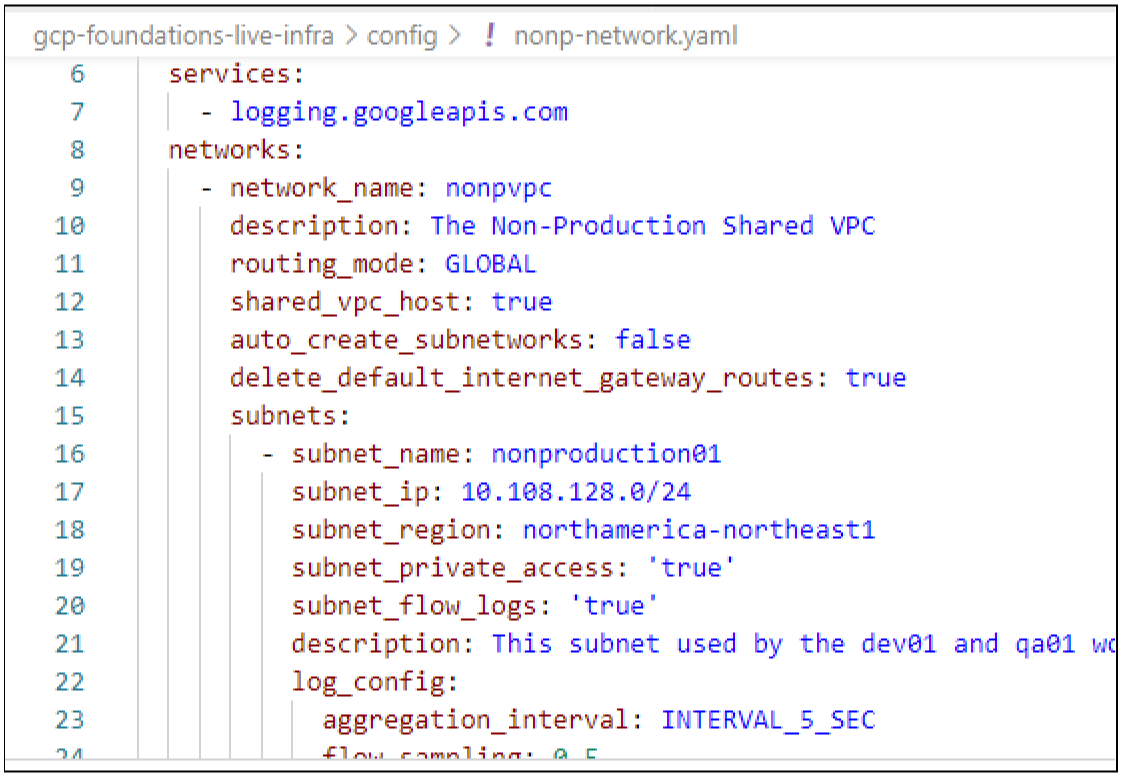
5. Create a **config** directory in *gcp-foundations-live-infra* repository and copy sample config files from *gcp-foundations-live-configs* repository to this directory.

6. Make sure to pass inputs to GCP from terraform modules through config files viz. organization-config.yaml, nonp-network.yaml, etc.



gcp-foundations-live-infra > config > ! organization-config.yaml

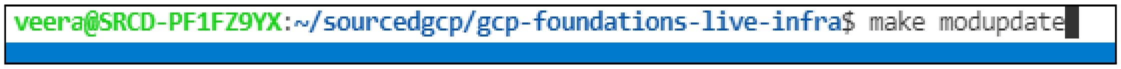
```yaml
---
orgId: 670891908486
defaultRegion: northamerica-northeast1
departmentCode: Gc
owner: Sc
environment: D
location: northamerica-northeast1

bootstrap:
  userDefinedString: sambootstrap
  additionalUserDefinedString: ""
  billingAccount: 00E11A-0AB9A2-077BE7
```



gcp-foundations-live-infra > config > ! nonp-network.yaml

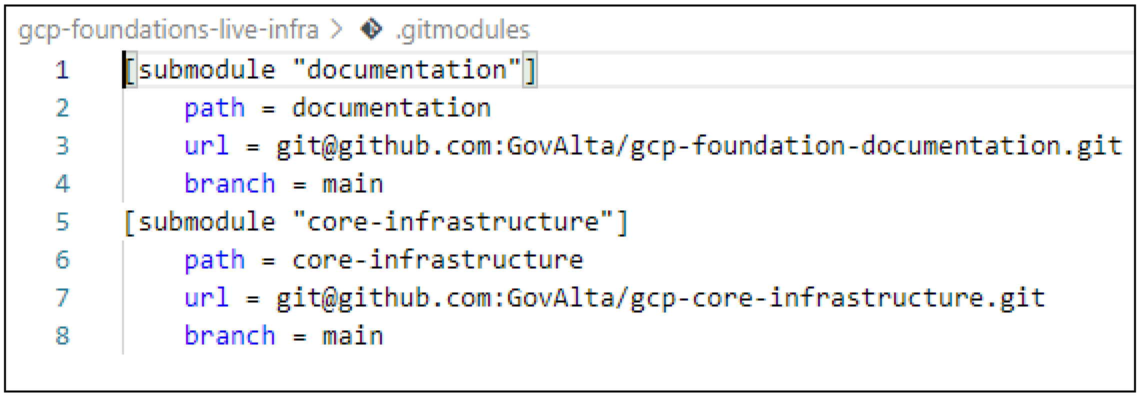
```yaml
services:
  - logging.googleapis.com
networks:
  - network_name: nonpvpc
    description: The Non-Production Shared VPC
    routing_mode: GLOBAL
    shared_vpc_host: true
    auto_create_subnetworks: false
    delete_default_internet_gateway_routes: true
    subnets:
      - subnet_name: nonproduction01
        subnet_ip: 10.108.128.0/24
        subnet_region: northamerica-northeast1
        subnet_private_access: 'true'
        subnet_flow_logs: 'true'
        description: This subnet used by the dev01 and qa01 wo
        log_config:
          aggregation_interval: INTERVAL_5_SEC
          flow_sampling: 0.5
```

7. When there are changes in *gcp-core-infrastructure* and *gcp-foundation-documentation* modules, make sure to pull all those changes into *gcp-foundations-live-infra* repository submodules using the *make modupdate* command.



```
veera@SRCD-PF1FZ9YX:~/sourcedgcp/gcp-foundations-live-infra$ make modupdate
```

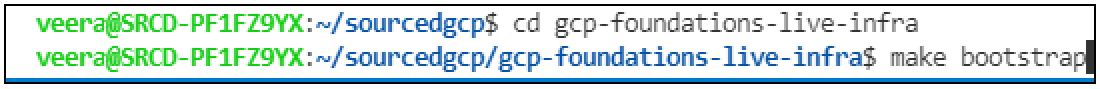
The *make modupdate* command will pull the latest changes into *gcp-foundations-live-infra* repository using configurations in *. gitmodules* file in this repository.



gcp-foundations-live-infra > .gitmodules

```
[submodule "documentation"]
    path = documentation
    url = git@github.com:GovAlta/gcp-foundation-documentation.git
    branch = main
[submodule "core-infrastructure"]
    path = core-infrastructure
    url = git@github.com:GovAlta/gcp-core-infrastructure.git
    branch = main
```

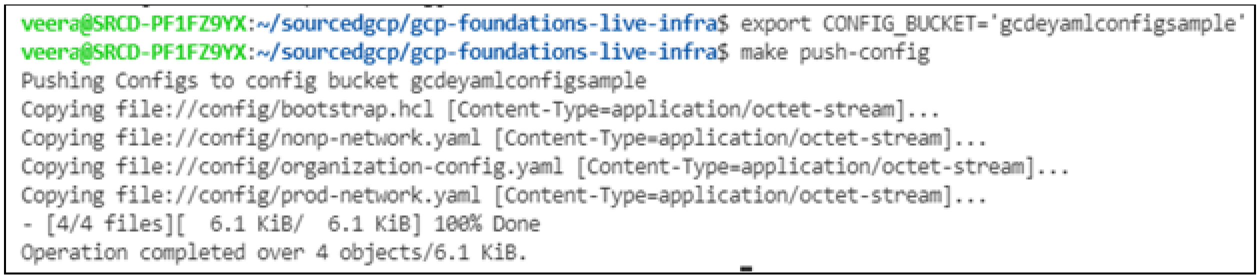
8. Run *make bootstrap* command from *gcp-foundations-live-infra* repository to set up a bootstrap project in GCP.



```
veera@SRCD-PF1FZ9YX:~/sourcedgcp$ cd gcp-foundations-live-infra
veera@SRCD-PF1FZ9YX:~/sourcedgcp/gcp-foundations-live-infra$ make bootstrap
```

9. Set the CONFIG_BUCKET for your project using *export CONFIG_BUCKET=""* command. The bucket name is derived from the output of the previous *make bootstrap* command.

   Use the *make push-config* command to push the latest config files into the yaml-config cloud storage buckets so that the latest input values are picked up by subsequent runs of bootstrap.



```
veera@SRCD-PF1FZ9YX:~/sourcedgcp/gcp-foundations-live-infra$ export CONFIG_BUCKET='gcdeyamlconfigsample'
veera@SRCD-PF1FZ9YX:~/sourcedgcp/gcp-foundations-live-infra$ make push-config
Pushing Configs to config bucket gcdeyamlconfigsample
Copying file://config/bootstrap.hcl [Content-Type=application/octet-stream]...
Copying file://config/nonp-network.yaml [Content-Type=application/octet-stream]...
Copying file://config/organization-config.yaml [Content-Type=application/octet-stream]...
Copying file://config/prod-network.yaml [Content-Type=application/octet-stream]...
- [4/4 files][  6.1 KiB/  6.1 KiB] 100% Done
Operation completed over 4 objects/6.1 KiB.
```

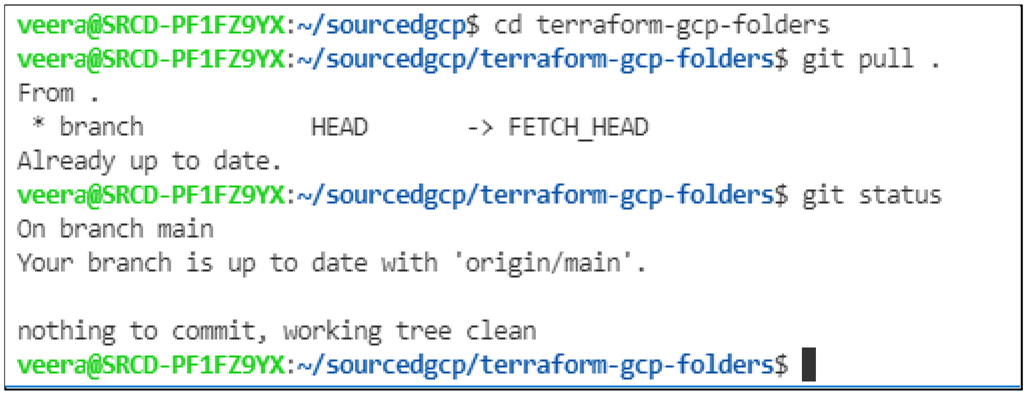
10. Any code changes to git repositories should be made in the “feature/xxx” branch, committed, pushed in and Pull Request must be created to be approved and merged to the “main” branch after peer verification.

    As a best practice, direct code changes in the “main” branch must be avoided.

Steps to make code changes in “feature” branch

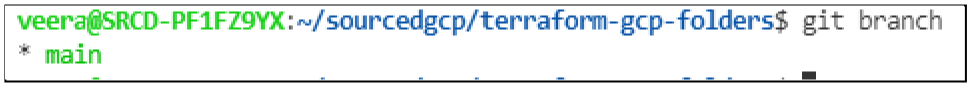
As an example, if code changes are required in the *terraform-gcp-folders* repository, below steps must be followed.

1. Switch to *terraform-gcp-folders* directory.
2. Make sure to pull the latest code using *git pull .* command.
3. Check the status of the repository using *git status*. It should have the latest code and should NOT have any staged files in the local area which are not committed.



```
veera@SRCD-PF1FZ9YX:~/sourcedgcp$ cd terraform-gcp-folders
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$ git pull .
From .
 * branch            HEAD       -> FETCH_HEAD
Already up to date.
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$
```

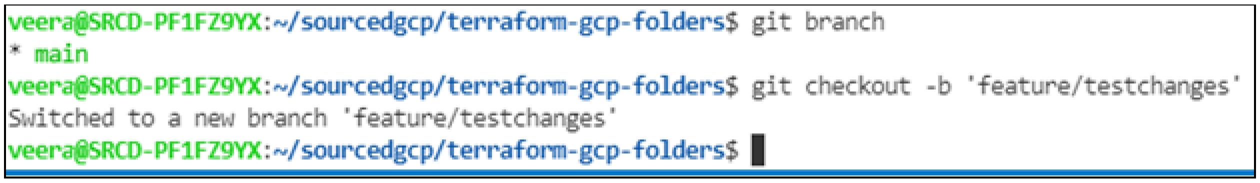
4. Run the *git branch* command to check which branch your local repository is pointing to currently.



```
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$ git branch
* main
```

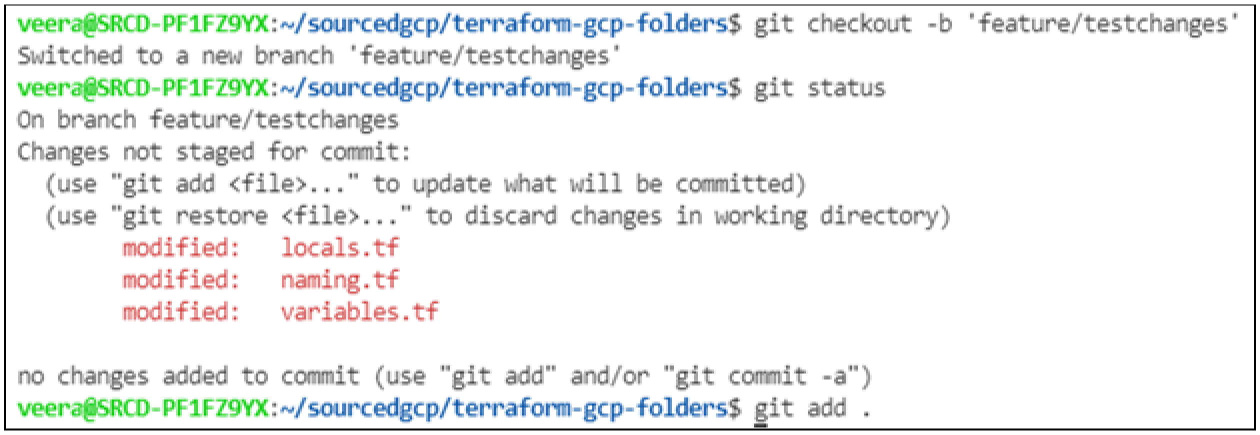
5. Create a new feature branch for the repository using the *git checkout* command.

   # git checkout -b ‘feature/testchanges’



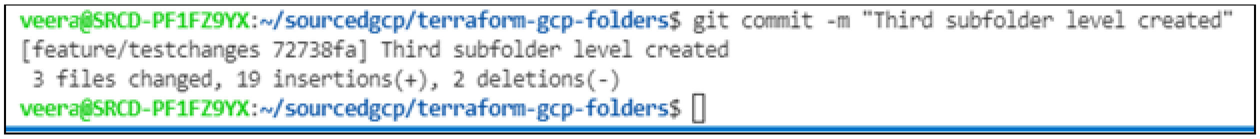
```
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$ git branch
* main
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$ git checkout -b 'feature/testchanges'
Switched to a new branch 'feature/testchanges'
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$
```

6. The local repository now points to the feature branch as can be seen in the above screen.
7. Make the required code changes in main.tf, variables.tf, locals.tf, naming.tf, outputs.tf and Makefile as per the use case.
8. After the code changes are made, please check the *git status*. The changes to be committed are visible.



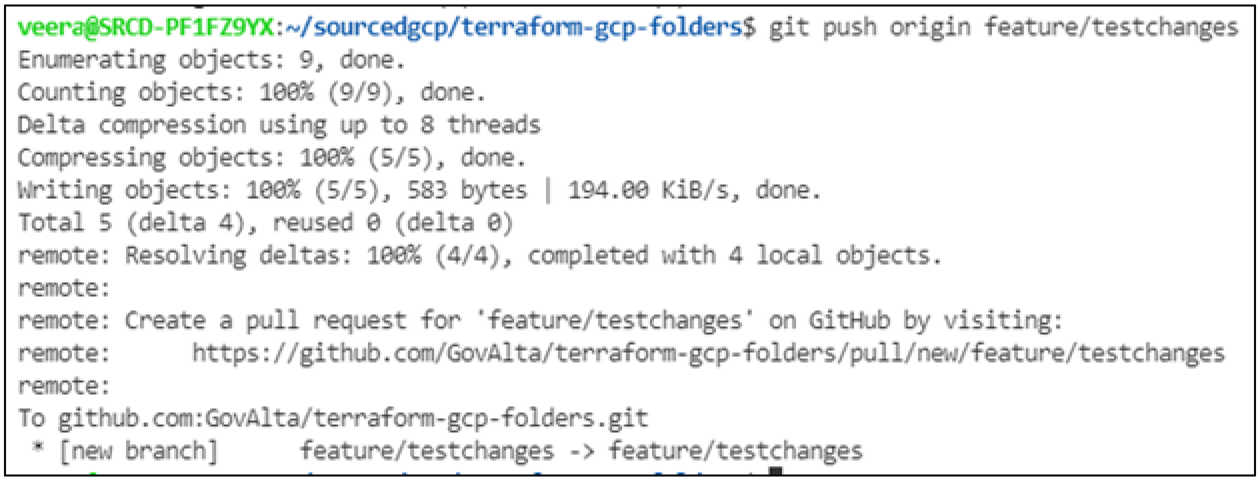
```
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$ git checkout -b 'feature/testchanges'
Switched to a new branch 'feature/testchanges'
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$ git status
On branch feature/testchanges
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   locals.tf
        modified:   naming.tf
        modified:   variables.tf

no changes added to commit (use "git add" and/or "git commit -a")
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$ git add .
```

9. Add the changes using "*git add .*" command.
   Commit the changes using the command *git commit -m "<your commit msg>"*



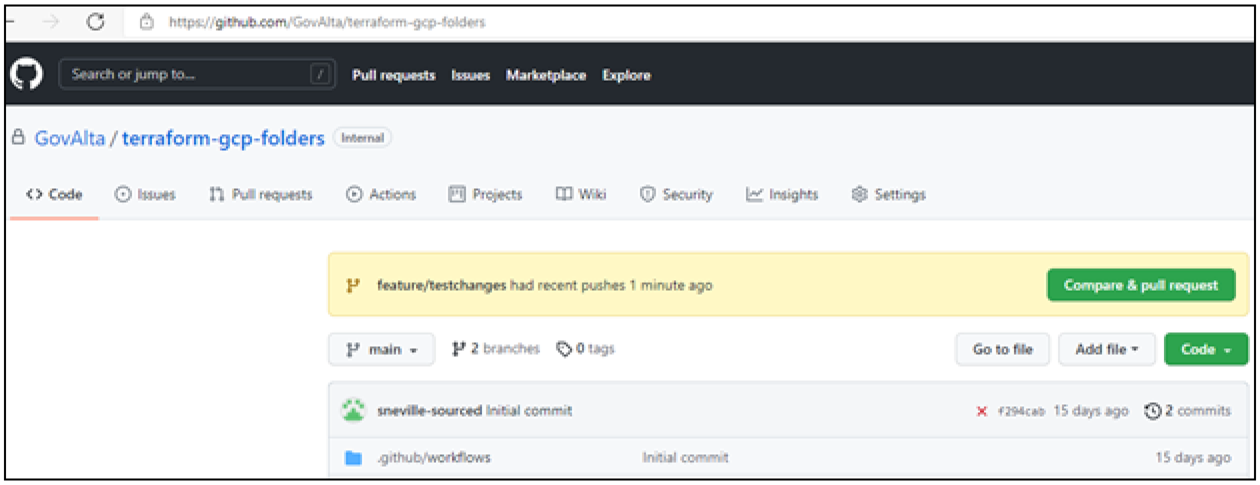
```
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$ git commit -m "Third subfolder level created"
[feature/testchanges 72738fa] Third subfolder level created
 3 files changed, 19 insertions(+), 2 deletions(-)
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$
```

10. Push the changes to the feature branch – *git push origin feature/testchanges*

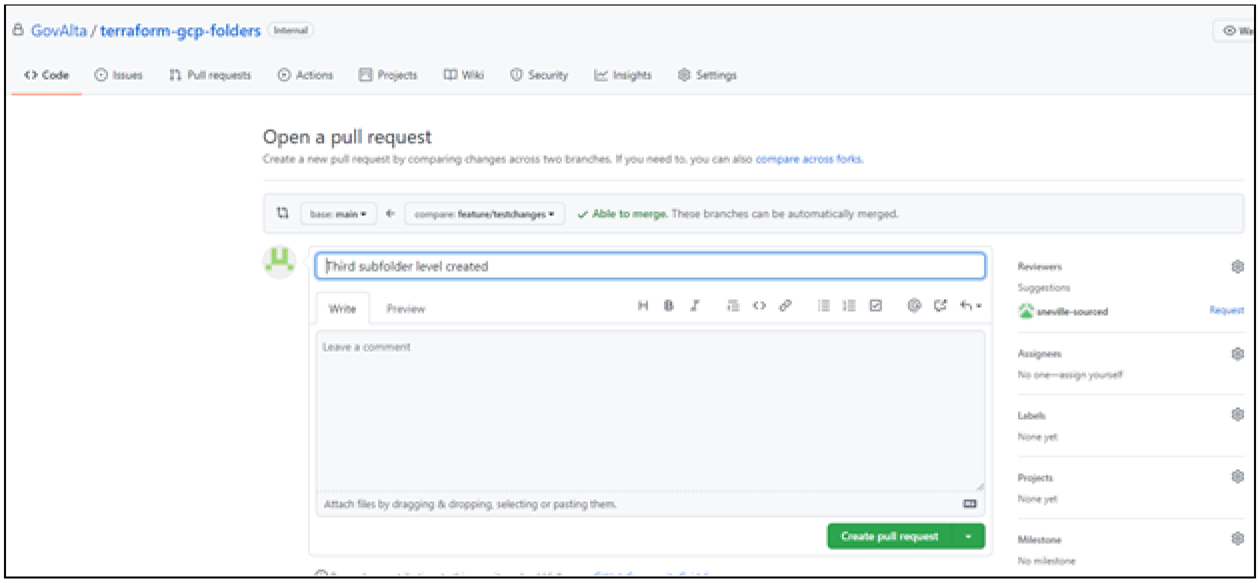


```
veera@SRCD-PF1FZ9YX:~/sourcedgcp/terraform-gcp-folders$ git push origin feature/testchanges
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 583 bytes | 194.00 KiB/s, done.
Total 5 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
remote:
remote: Create a pull request for 'feature/testchanges' on GitHub by visiting:
remote:      https://github.com/GovAlta/terraform-gcp-folders/pull/new/feature/testchanges
remote:
To github.com:GovAlta/terraform-gcp-folders.git
 * [new branch]      feature/testchanges -> feature/testchanges
```

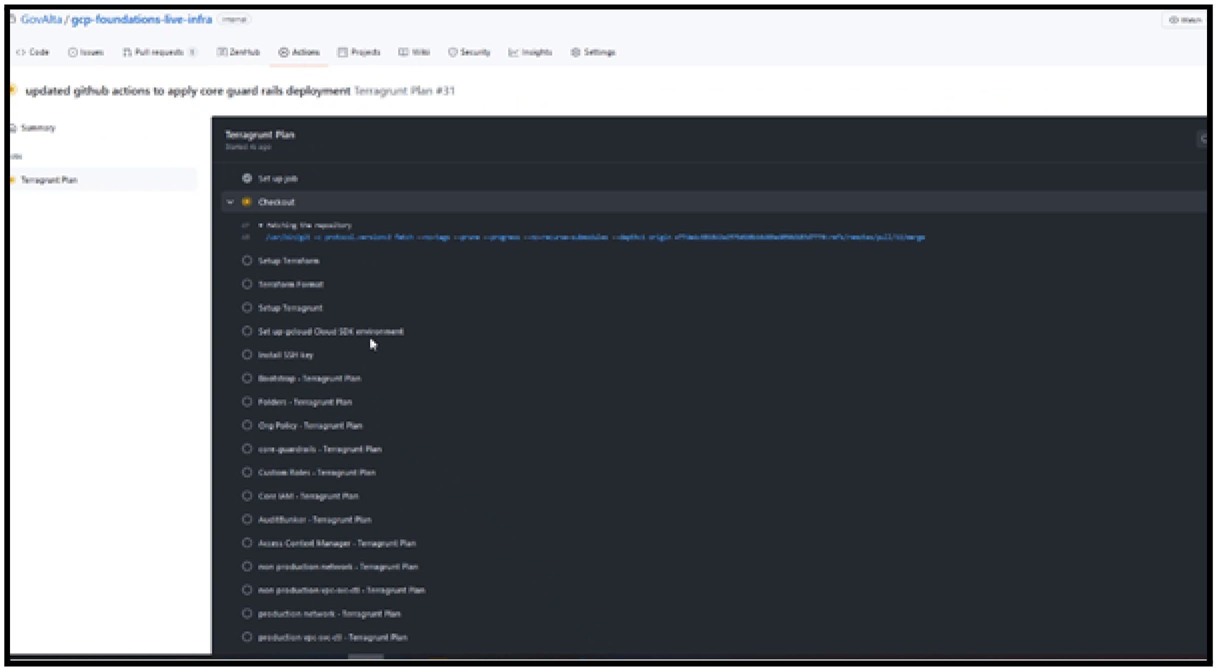
11. Once the code changes are complete, switch to the Github repository to create the Pull Request and view the Terragrunt plan being generated.



12. Click on "Request" under the "Reviewers" section and then click on "Create pull request".


GovAlta / terraform-gcp-folders
Internal
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Open a pull request
Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.
base: main
compare: feature/testchanges
Able to merge. These branches can be automatically merged.
Third subfolder level created
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Create pull request
Reviewers
Suggestions
sneville-sourced
Request
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone

13. The automated pipeline is triggered and the terragrunt plan is generated which can be checked and approved in the Github Pull Requests tab by the reviewer.

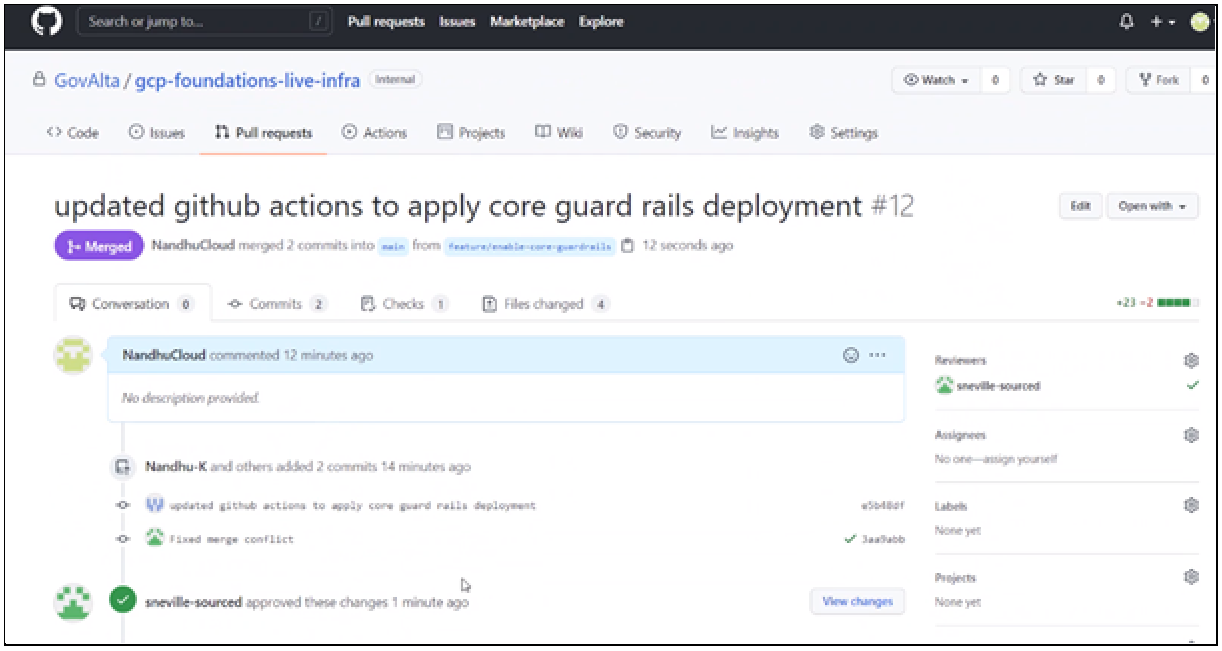


GovAlta / gcp-foundations-live-infra

updated github actions to apply core guard rails deployment Terragrunt Plan #31

Terragrunt Plan

- Set up job
- Checkout
- Setup Terraform
- Terraform Format
- Setup Terragrunt
- Set up gcloud Cloud SDK environment
- Install SSH key
- Bootstrap - Terragrunt Plan
- Folders - Terragrunt Plan
- Org Policy - Terragrunt Plan
- core-guardrails - Terragrunt Plan
- Custom Roles - Terragrunt Plan
- Core IAM - Terragrunt Plan
- AuditBunker - Terragrunt Plan
- Access Context Manager - Terragrunt Plan
- non production-network - Terragrunt Plan
- non production-vpc-sc-dl - Terragrunt Plan
- production-network - Terragrunt Plan
- production-vpc-sc-dl - Terragrunt Plan



GovAlta / gcp-foundations-live-infra Internal

updated github actions to apply core guard rails deployment #12

Merged NandhuCloud merged 2 commits into `main` from `feature/enable-core-guardrails` 12 seconds ago

Conversation 0 · Commits 2 · Checks 1 · Files changed 4 · +23 −2

NandhuCloud commented 12 minutes ago

No description provided.

Nandhu-K and others added 2 commits 14 minutes ago

```
updated github actions to apply core guard rails deployment    e5b48df
Fixed merge conflict                                          ✓ 3aa9abb
```

sneville-sourced approved these changes 1 minute ago

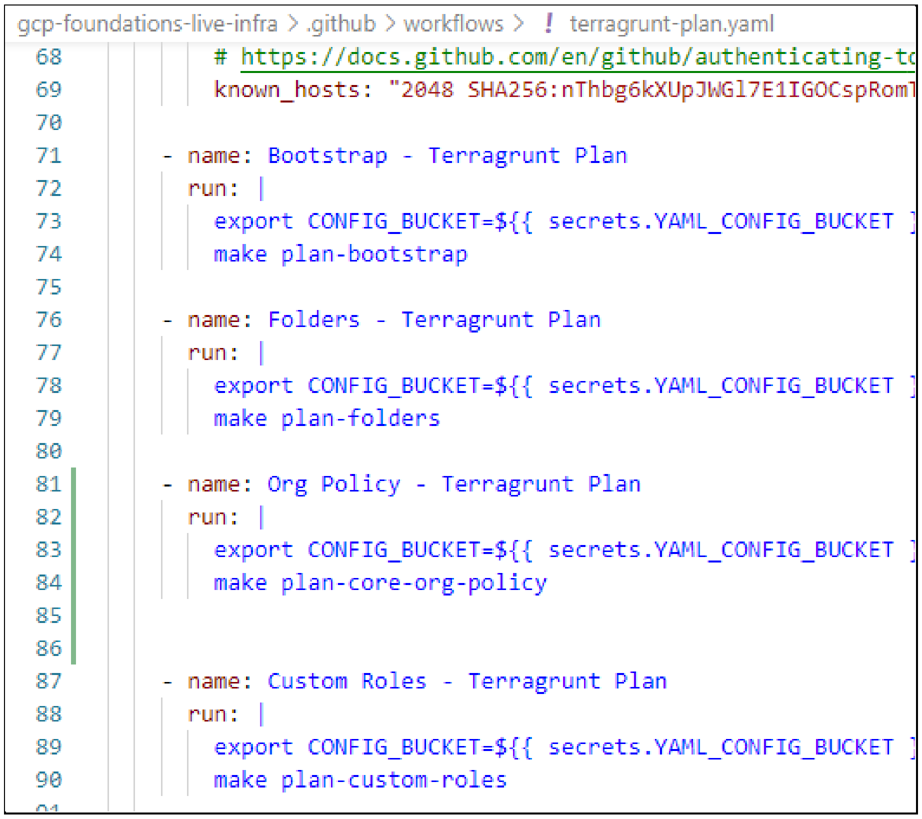
Reviewers: sneville-sourced ✓
Assignees: No one—assign yourself
Labels: None yet
Projects: None yet

14. Once the Pull Request is approved and merged into the main branch in Github, the plan created in the previous step is applied and the actual GCP resources are provisioned, and this can be monitored like the Terragrunt plan above.

15. The automated Github actions workflow is triggered by the *.github/workflows* configurations under *gcp-foundations-live-infra* repository.

    All the Terragrunt plan and apply commands are configured in the corresponding YAML files in this repository.

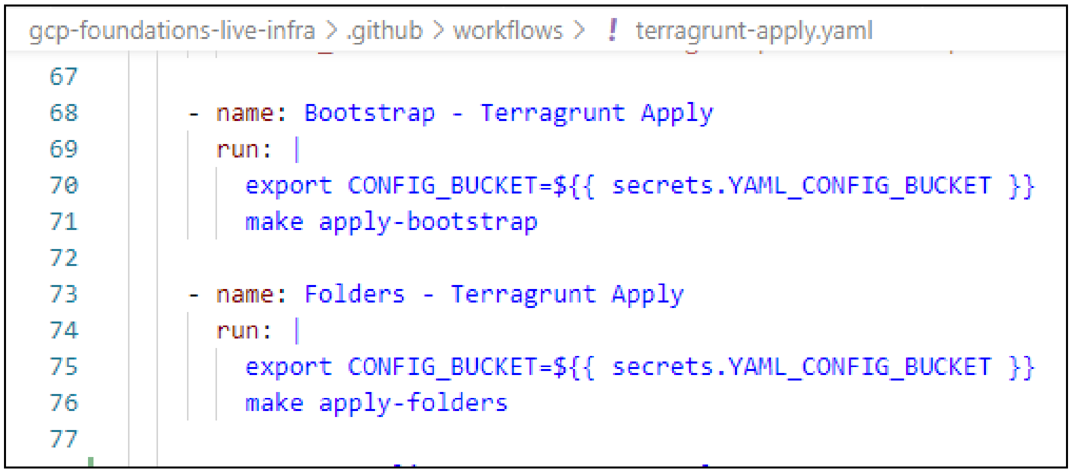
Terragrunt-Plan



gcp-foundations-live-infra > .github > workflows > ! terragrunt-plan.yaml

```
        # https://docs.github.com/en/github/authenticating-to
        known_hosts: "2048 SHA256:nThbg6kXUpJWGl7E1IGOCspRoml

    - name: Bootstrap - Terragrunt Plan
      run: |
        export CONFIG_BUCKET=${{ secrets.YAML_CONFIG_BUCKET }
        make plan-bootstrap

    - name: Folders - Terragrunt Plan
      run: |
        export CONFIG_BUCKET=${{ secrets.YAML_CONFIG_BUCKET }
        make plan-folders

    - name: Org Policy - Terragrunt Plan
      run: |
        export CONFIG_BUCKET=${{ secrets.YAML_CONFIG_BUCKET }
        make plan-core-org-policy


    - name: Custom Roles - Terragrunt Plan
      run: |
        export CONFIG_BUCKET=${{ secrets.YAML_CONFIG_BUCKET }
        make plan-custom-roles
```

Terragrunt-Apply



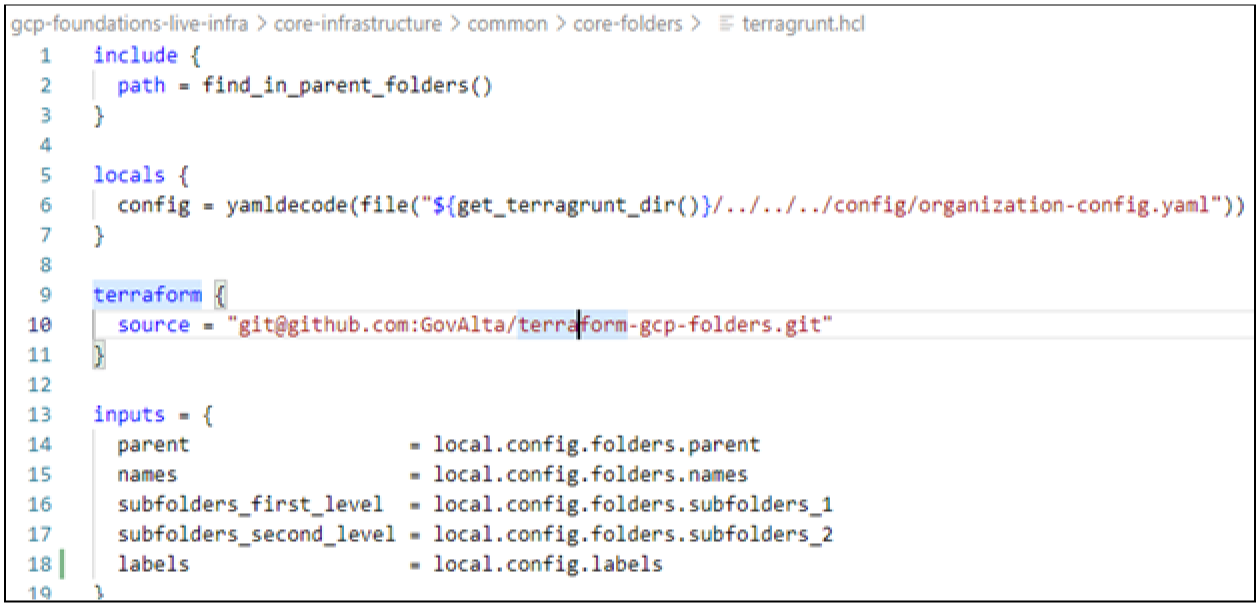
gcp-foundations-live-infra > .github > workflows > ! terragrunt-apply.yaml

```
      - name: Bootstrap - Terragrunt Apply
        run: |
          export CONFIG_BUCKET=${{ secrets.YAML_CONFIG_BUCKET }}
          make apply-bootstrap

      - name: Folders - Terragrunt Apply
        run: |
          export CONFIG_BUCKET=${{ secrets.YAML_CONFIG_BUCKET }}
          make apply-folders

```

Terragrunt.hcl

This file is the starting point for the Terragrunt workflow and includes the path to parent Terragrunt.hcl files, path to config files to be loaded by Terragrunt, terraform source for this Terragrunt and any inputs to be passed to the terraform source module from Terragrunt.

Sample File



gcp-foundations-live-infra > core-infrastructure > common > core-folders > terragrunt.hcl

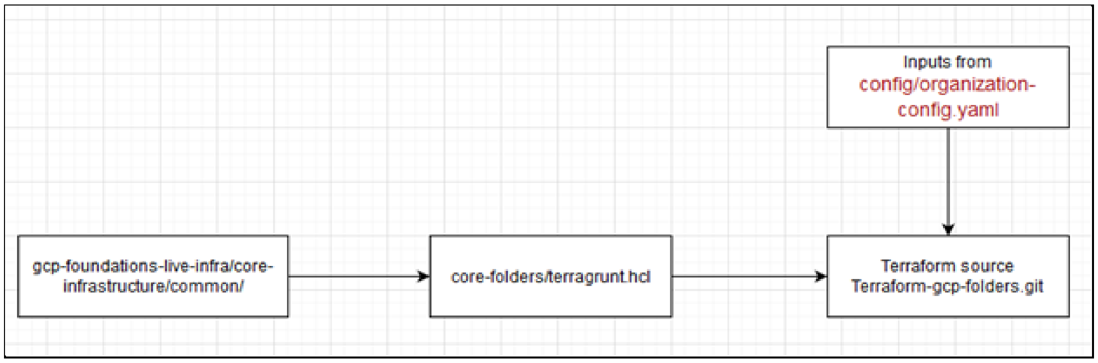
```
include {
  path = find_in_parent_folders()
}

locals {
  config = yamldecode(file("${get_terragrunt_dir()}/../../../config/organization-config.yaml"))
}

terraform {
  source = "git@github.com:GovAlta/terraform-gcp-folders.git"
}

inputs = {
  parent                 = local.config.folders.parent
  names                  = local.config.folders.names
  subfolders_first_level  = local.config.folders.subfolders_1
  subfolders_second_level = local.config.folders.subfolders_2
  labels                 = local.config.labels
}
```

Terragrunt Blocks

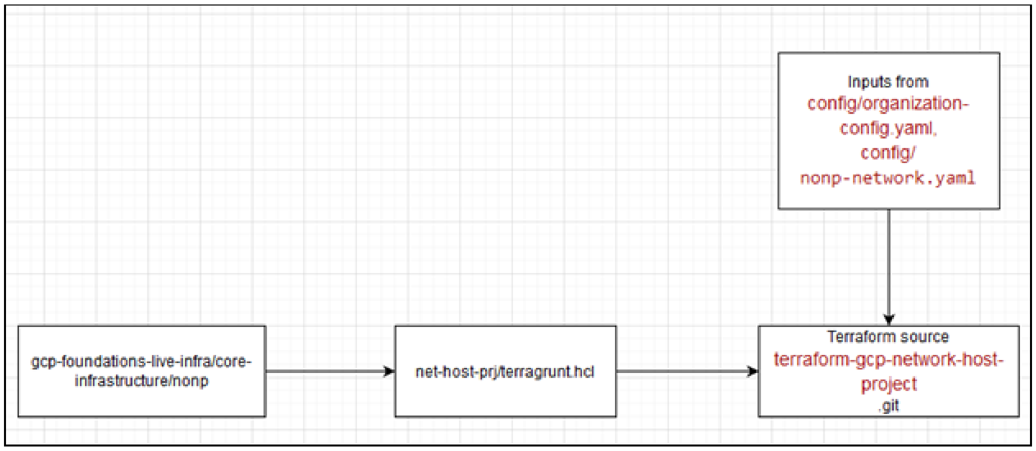
- **Locals** – loads the config file needed for this module
- **Terraform** – loads the terraform source for this Terragrunt.
- **Inputs** – passes the input values to terraform source based on values configured in the config file.
- **Include** – includes the Terragrunt.hcl source files from the parent directories.

Terragrunt Code Flow

Sample 1:


Inputs from config/organization-config.yaml
gcp-foundations-live-infra/core-infrastructure/common/
core-folders/terragrunt.hcl
Terraform source Terraform-gcp-folders.git

Sample 2:


Inputs from config/organization-config.yaml, config/ nonp-network.yaml
gcp-foundations-live-infra/core-infrastructure/nonp
net-host-prj/terragrunt.hcl
Terraform source terraform-gcp-network-host-project .git