

BOOSTING TERRAGRUNT PERFORMANCE IN ATLANTIS WITH run-all AND PROVIDER CACHING

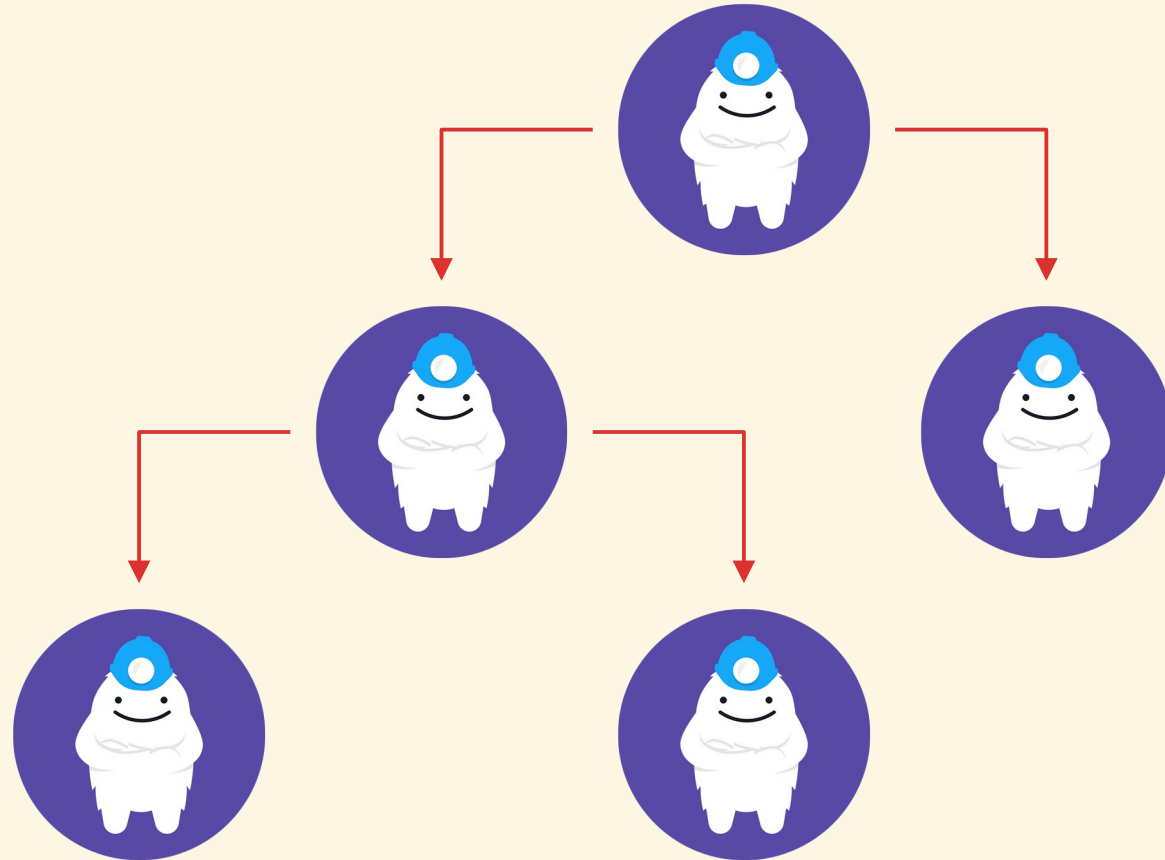
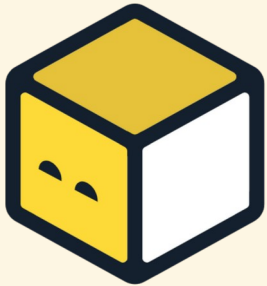
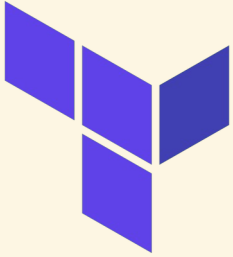
a practical configuration example

Who is this guy?

- SRE at RiksTV, a Norwegian TV channels distributor
- Automation and Infrastructure-as-code junkie
- Previously:
 - Worked for Telenor, Opera Software, ++
 - CFEngine power user, CFEngine Champion 2012
 - speaker at CfgMgmtCamp, FOSDEM, Italian DevOps Meeting
- Amateur runner (5k, 10k, a couple half marathons)
- See LinkedIn for more, I am not here to speak about myself!



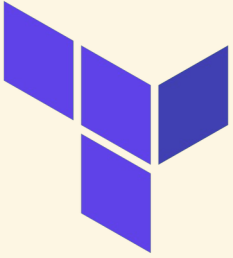
Prerequisites



Atlantis?



Our environment



What is Atlantis?

Atlantis

Q Search

CtrlK

HomeGuideDocsContributingBlog

Atlantis

Terraform Pull Request Automation

Running Terraform Workflows with Ease

Get Started

What is Atlantis?

Join us on Slack





Fewer Mistakes

Catch errors in Terraform plan output before applying changes. Ensure changes are applied before merging.



Empower Developers

Developers can safely submit Terraform pull requests without credentials. Require approvals for applies.



Instant Audit Logs

Detailed logs for infrastructure changes, approvals, and user actions. Configure approvals for production changes.



Proven at Scale

Used by top companies to manage over 600 repos with 300 developers. In production since 2017.



Self-Hosted

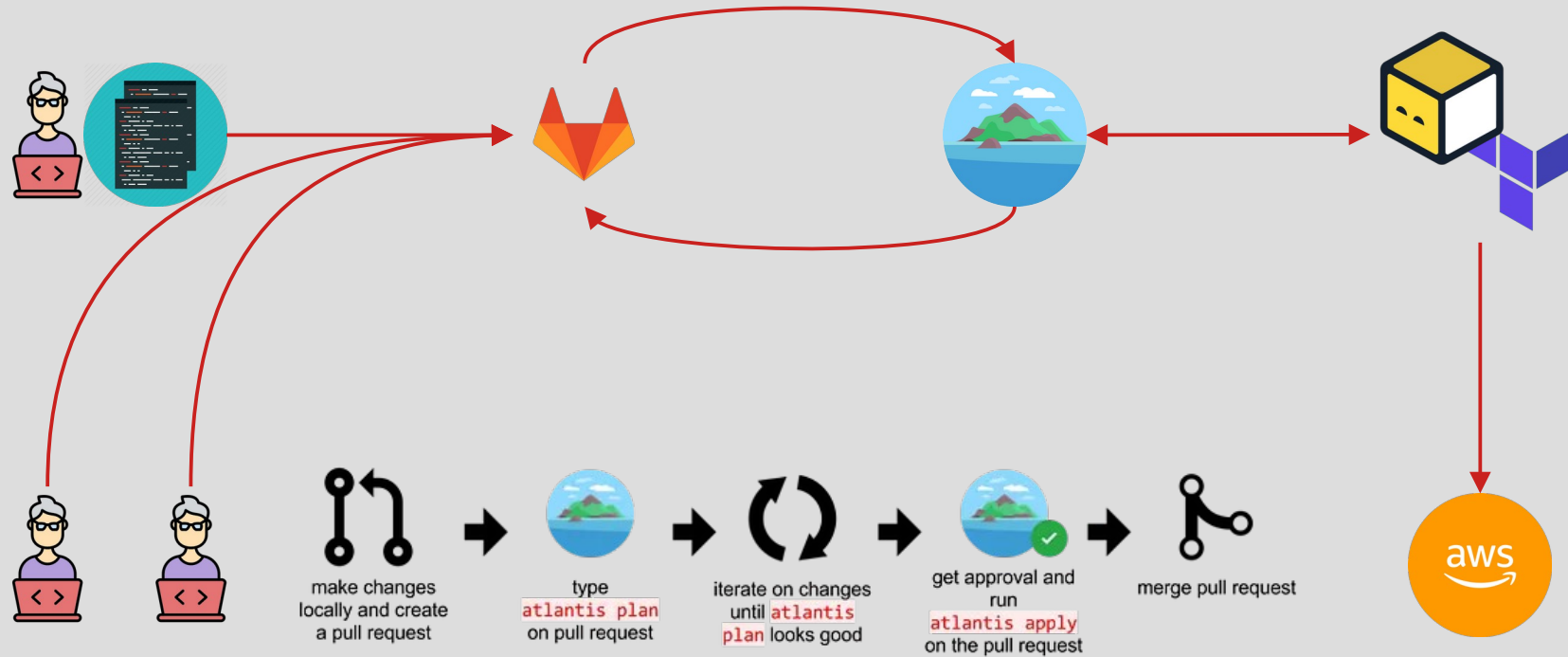
Your credentials remain secure. Deployable on VMs, Kubernetes, Fargate, etc. Supports GitHub, GitLab, Bitbucket, Azure DevOps.



Open Source

Atlantis is an open source project with strong community support, powered by volunteer contributions.

Working with Atlantis



Better code reviews

rikstv / sre / RikstV.Terraform.Infra / Merge requests / !575

Merged Remove noncurrent objects in Nexus S3 bucket

Overview 11 Commits 3 Pipelines 4 Changes 1

Terraform will perform the following actions:

```
# module.s3bucket.aws_s3_bucket_lifecycle_configuration
~ resource "aws_s3_bucket_lifecycle_configuration" "lifecycle_configuration" {
  id = "lifecycle_configuration"
  # (3 unchanged attributes hidden)

+ rule {
+   id      = "RemoveNonCurrentVersions"
+   status = "Enabled"

+   filter {
+   }

+   noncurrent_version_expiration {
+     noncurrent_days = 1
+   }
+ }

+ rule {
+   id      = "AbortIncompleteMultipartUpload"
+   status = "Enabled"

+   filter {
+   }
+ }

# (1 unchanged block hidden)
}

# module.s3bucket.aws_s3_bucket_versioning
~ resource "aws_s3_bucket_versioning" "versioning" {
  id = "versioning"
  # (2 unchanged attributes hidden)
```

Add a to-do item

Assignee

Marco Marongiu

Edit

Reviewer

All required approvals given [Assign](#)

Arman Nourifar

Edit

Labels

None

Edit

Milestone

None

Edit

Time tracking

No estimate or time spent

+ -

3 Participants

Atlantis workflows

Atlantis

Guide

Test Drive

Testing locally

Docs

Installing Atlantis

Configuring Atlantis

Overview

Server Configuration

Server Side Repo Config

Pre Workflow Hooks

Post Workflow Hooks

Conftest Policy Checking

Custom Workflows

Repo and Project Permissions

Repo Level atlantis.yaml

Upgrading atlantis.yaml

Command Requirements

Checkout Strategy

Terraform Versions

Terraform Cloud

Sending Notifications via Webhooks

Q Search

Ctrl K

Home

Guide

Docs

Contributing

Blog

Custom Workflows

Custom workflows can be defined to override the default commands that Atlantis runs.

Usage

Custom workflows can be specified in the Server-Side Repo Config or in the Repo-Level `atlantis.yaml` files.

Notes:

- If you want to allow repos to select their own workflows, they must have the `allowed_overrides: [workflow]` setting. See [server-side repo config use cases](#) for more details.
- If in addition you also want to allow repos to define their own workflows, they must have the `allow_custom_workflows: true` setting. See [server-side repo config use cases](#) for more details.

Use Cases

.tfvars files

Given the structure:

On this page

Usage

Use Cases

.tfvars files

Adding extra arguments ...

Custom init/plan/apply ...

CDKTF

Terragrunt

Running custom comma...

Custom Backend Config

Add directory and repo ...

Reference

Workflow

Stage

Step

Atlantis workflows

Atlantis

Guide

Test Drive

Testing locally

Docs

Installing Atlantis

Configuring Atlantis

Overview

Server Configuration

Server Side Repo Config

Pre Workflow Hooks

Post Workflow Hooks

ConfTest Policy Checking

Custom Workflows

Repo and Project Permissions

Repo Level atlantis.yaml

Upgrading atlantis.yaml

Command Requirements

Checkout Strategy

Terraform Versions

Terraform Cloud

Sending Notifications via Webhooks

Q Search

CtrlK

Home

Guide

Docs

Contributing

Blog

Terragrunt

Atlantis supports running custom commands in place of the default Atlantis commands. We can use this functionality to enable [Terragrunt](#).

You can either use your repo's `atlantis.yaml` file or the Atlantis server's `repos.yaml` file.

Given a directory structure:

```
graph TD
    . --> live
    live --> prod
    prod --> terragrunt_hcl1[terragrunt.hcl]
    live --> staging
    staging --> terragrunt_hcl2[terragrunt.hcl]
```

If using the server `repos.yaml` file, you would use the following config:

```
# repos.yaml
# Specify TERRAGRUNT_TFPATH environment variable to accommodate setting --default-backend-config
# Generate json plan via terragrunt for policy checks
repos:
- id: "/"
  workflow: terragrunt
workflows:
  terragrunt:
    plan:
      steps:
      - env:
        name: TERRAGRUNT_TFPATH
```

On this page

Usage

Use Cases

.tfvars files

Adding extra arguments ...

Custom init/plan/apply ...

CDKTF

Terragrunt

Running custom comma...

Custom Backend Config

Add directory and repo ...

Reference

Workflow

Stage

Step

atlantis.yaml (snippets)

```
- autoplan:
  enabled: true
  when_modified:
    - '*.hcl'
    - '*.tf*'
    - '**/*.hcl'
    - '**/*.tf*'
    - ../../terragrunt.hcl
    - ../stacks/iam/*.tf*
    - ../stacks/network/*.tf*
    - ../stacks/prefixlists/*.tf*
    - ../stacks/securitygroups/*.tf*
dir: accounts/rikstv
name: accounts_rikstv
workspace: accounts_rikstv
```

```
- autoplan:
  enabled: true
  when_modified:
    - '*.hcl'
    - '*.tf*'
    - '**/*.hcl'
    - '**/*.tf*'
    - ../../../../terragrunt.hcl
dir: apps/bi/foobar
name: apps_bi_foobar
workspace: apps_bi_foobar
```

terragrunt-atlantis-config

Files

master

Go to file

> .github

> cmd

> test_examples

.gitignore

.goxc.json

CONTRIBUTORS.svg

Dockerfile

LICENSE

Makefile

README.md

go.mod

go.sum

main.go

terragrunt-atlantis-config / README.md

Preview Code Blame 234 lines (171 loc) · 17.5 KB

Raw Copy Download Edit Menu



Terragrunt Atlantis Config

Generate Atlantis Config for Terragrunt projects.

What is this?

[Atlantis](#) is an awesome tool for Terraform pull request automation. Each repo can have a YAML config file that defines Terraform module dependencies, so that PRs that affect dependent modules will automatically generate `terraform plan`s for those modules.

[Terragrunt](#) is a Terraform wrapper, which has the concept of dependencies built into its configuration.

This tool creates Atlantis YAML configurations for Terragrunt projects by:

- Finding all `terragrunt.hcl` in a repo
- Evaluating their `dependency`, `terraform`, `locals`, and other source blocks to find their dependencies
- Creating a Directed Acyclic Graph of all dependencies
- Constructing and logging YAML in Atlantis' config spec that reflects the graph

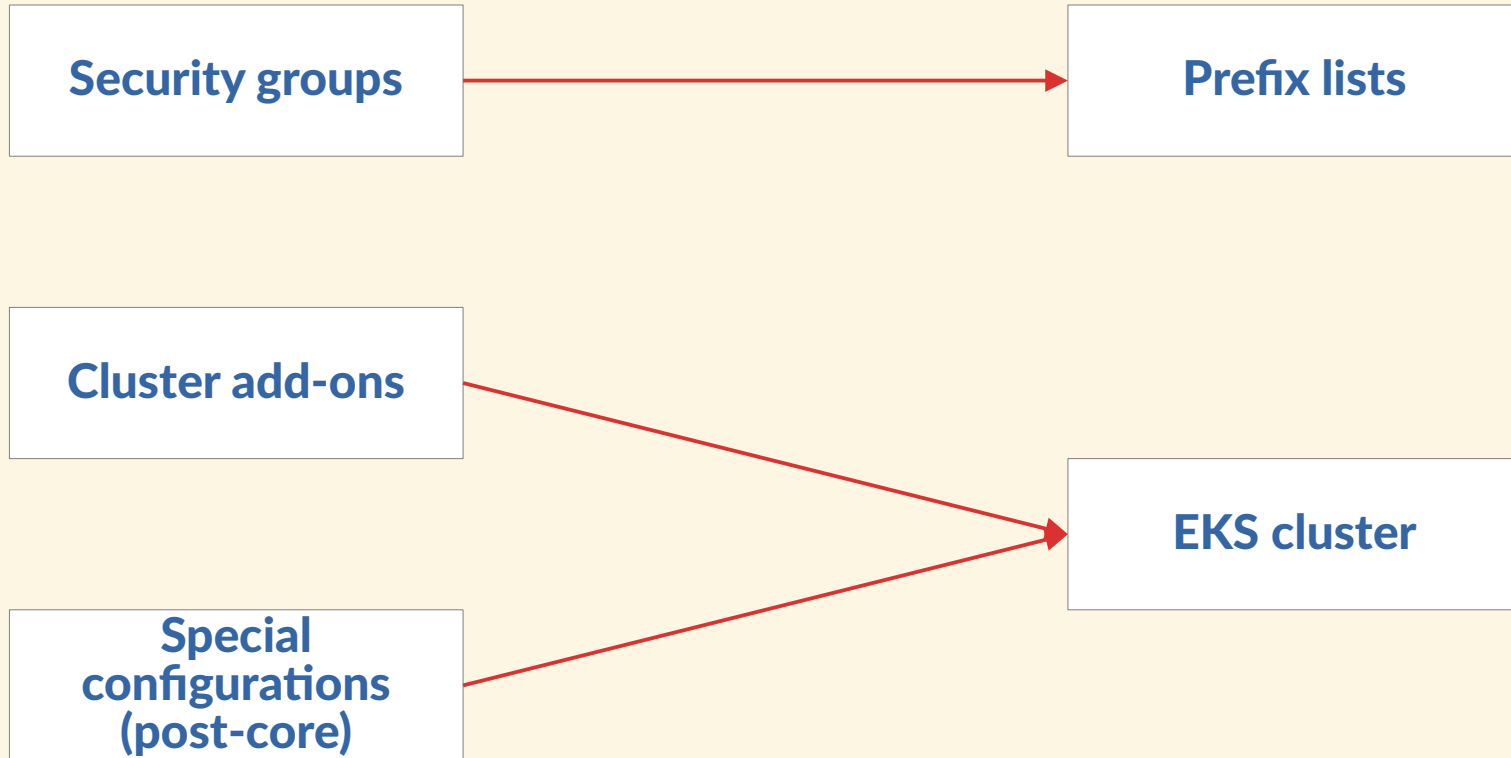
This is especially useful for organizations that use monorepos for their Terragrunt config (as we do at Transcend), and have thousands of lines of config.

Integrate into your Atlantis Server

The recommended way to use this tool is to install it onto your Atlantis server, and then use a [Pre-Workflow hook](#) to run it after every clone. This way, Atlantis can automatically determine what modules should be planned/applied for any change to your repository.

To get started, add a `pre_workflow_hooks` field to your `repos` section of your [server-side repo config](#):

Stack dependencies



This is slow...

- parallelising doesn't help much
- provider caching not concurrency-safe

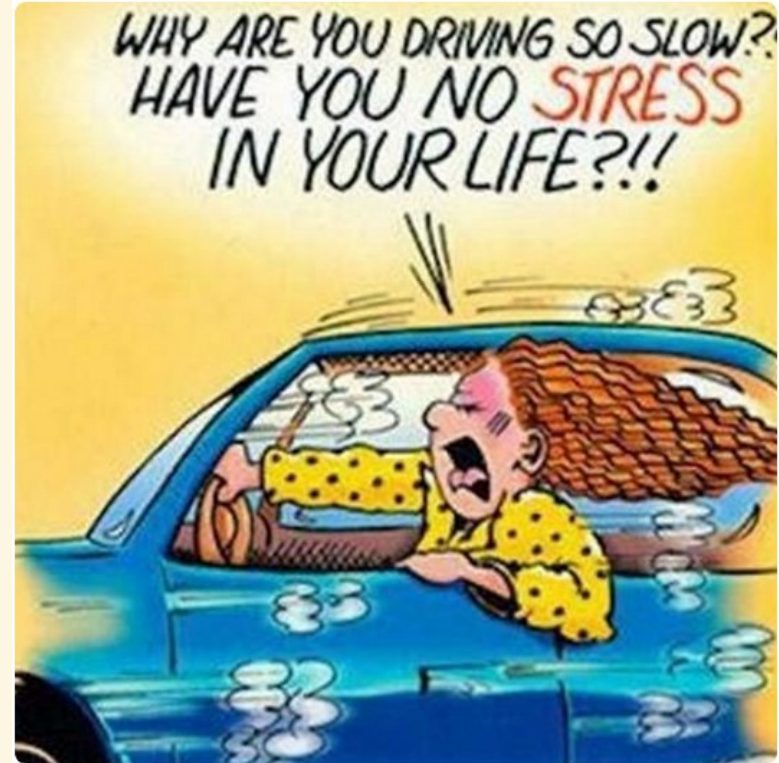


Image from <https://www.pinterest.com/pin/7881368073450681/>

Possible solutions

a proxy cache on premise?

just live with that and be happy?

...or something in between?

run-all + terragrunt provider caching

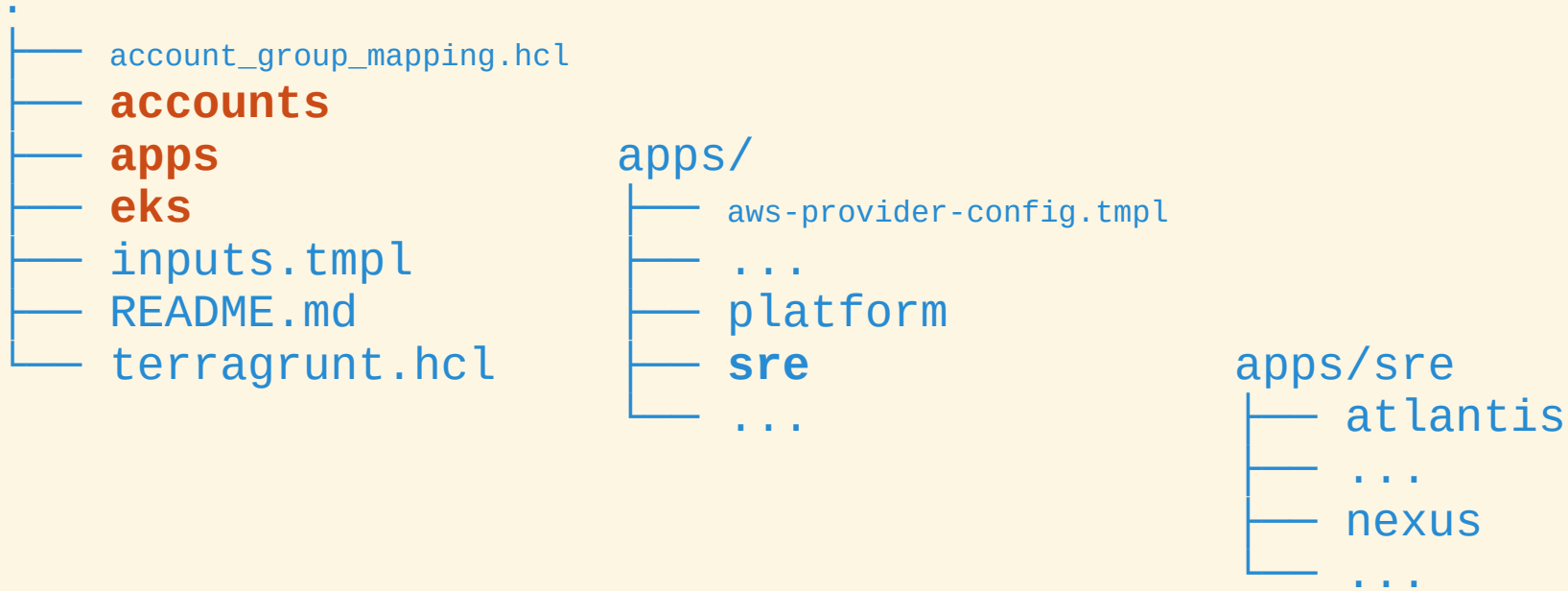
Terragrunt provider caching

EXPERIMENTAL FEATURE!

```
extra_arguments "terraform_terragrunt_caching" {  
  commands = ["init", "plan", "apply", "show", "import", "providers"]  
  env_vars = {  
    TERRAGRUNT_PROVIDER_CACHE      = 1  
    TERRAGRUNT_PROVIDER_CACHE_DIR = local.plugin_cache_dir  
    TF_PLUGIN_CACHE_DIR            = local.plugin_cache_dir  
  }  
}
```

Building a terragrunt run-all workflow

Repo structure and stacks



Structure of a stack

```
apps/sre/nexus/  
├── context.hcl  
├── dev  
│   └── terragrunt.hcl  
├── prod  
│   └── terragrunt.hcl  
├── README.md  
└── _stack  
    ├── additional_providers.tf  
    ├── db.tf  
    ├── ec2.tf  
    ├── main.tf  
    ├── s3.tf  
    └── variables.tf
```

- **context.hcl**: metadata
- **environments** (dev, prod...) **with terragrunt.hcl**
- **_stack**: terraform code for the resources of the stack

The simplest terragrunt.hcl

```
include "root" {  
  path = find_in_parent_folders()  
}  
  
terraform {  
  source = "../_stack"  
}
```

Episode 1: Nice try

rikstv / sre / RikstV.DockerImages.Atlantis / Merge requests / !35

Merged Implement run-all feature/runall into main

Overview 0 Commits 8 Pipelines 1 Changes 4

Add a to-do item

atlantis-repos-config.yaml

+3 -3 Viewed

```
27 27      - run:
30 30          # Allow for targetted plans/applies as not supported for Terraform wrappers by default
31 -      command: terragrunt plan -input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /g' | tr -d '\\') -no-color -out $PLANFILE
    31 +      command: terragrunt run-all plan --terragrunt-working-dir .. -input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /g' |
      tr -d '\\') -no-color -out $PLANFILE
32 32      output: hide
33 33      - run: |
34 34          terragrunt show $PLANFILE
@@ -38,7 +38,7 @@ workflows:
38 38          # Reduce Terraform suggestion output
39 39          name: TF_IN_AUTOMATION
40 40          value: "true"
41 -      - run: terragrunt apply -input=false $PLANFILE
    41 +      - run: terragrunt run-all apply --terragrunt-working-dir .. -input=false $PLANFILE
42 42      - env:
43 43          name: TERRAGRUNT_NON_INTERACTIVE
44 44          value: "true"
@@ -48,7 +48,7 @@ workflows:
48 48          name: TF_VAR_author
49 49          command: 'git show -s --format="%ae" $HEAD_COMMIT'
50 50          # Allow for imports as not supported for Terraform wrappers by default
51 -      - run: terragrunt import -input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /' | tr -d '\\')
    51 +      - run: terragrunt run-all import --terragrunt-working-dir .. -input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /' | tr -d
      '\\')
```

Episode 2: The project

rikstv / sre / RikstV.DockerImages.Atlantis / Merge requests / !35

Merged Implement run-all feature/runall into main

Overview 0 Commits 8 Pipelines 1 Changes 4

Add a to-do item

atlantis-repos-config.yaml

+2 -2 Viewed

	↑	@@ -4,7 +4,7 @@ repos:
4	4	apply_requirements: [mergeable, approved]
5	5	pre_workflow_hooks:
6	6	# Run https://github.com/transcend-io/terragrunt-atlantis-config but with some extra filtering using yq
7	-	- run: terragrunt-atlantis-config-with-yq-hook generate --output atlantis.yaml --autoplan --create-project-name --create-workspace --parallel="\$TERRAGRUNT_ATLANTIS_CONFIG_PARALLEL" --ignore_projects "atlantis"
7	+	- run: terragrunt-atlantis-config-with-yq-hook generate --project-hcl-files context.hcl --output atlantis.yaml --autoplan --create-project-name --create-workspace --parallel="\$TERRAGRUNT_ATLANTIS_CONFIG_PARALLEL" --ignore_projects "atlantis"
8	8	
9	9	# This should be identical to the real one (rikstv.terraform.infra), but is used for testing atlantis
10	10	# The webhook in the 'atlantistesting' repo is not configured to trigger atlantis in prod
	↕	@@ -12,7 +12,7 @@ repos:
12	12	workflow: terragrunt
13	13	apply_requirements: [mergeable, approved]
14	14	pre_workflow_hooks:
15	-	- run: terragrunt-atlantis-config-with-yq-hook generate --output atlantis.yaml --autoplan --create-project-name --create-workspace --parallel="\$TERRAGRUNT_ATLANTIS_CONFIG_PARALLEL" --ignore_projects "atlantis"
15	+	- run: terragrunt-atlantis-config-with-yq-hook generate --project-hcl-files context.hcl --output atlantis.yaml --autoplan --create-project-name --create-workspace --parallel="\$TERRAGRUNT_ATLANTIS_CONFIG_PARALLEL" --ignore_projects "atlantis"
16	16	# Workflow adapted from:
17	17	# https://www.runatlantis.io/docs/custom-workflows.html#terragrunt
18	18	workflows:
	↓	

Episode 2: The project (cont.)

rikstv / sre / rikstV.DockerImages.Atlantis / Merge requests / !35

Merged Implement run-all feature/runall into main

Overview 0 Commits 8 Pipelines 1 **Changes 4**

Add a to-do item

atlantis-repos-config.yaml

+3 -3 Viewed

```
31 -       command: terragrunt run-all plan --terragrunt-working-dir .. -input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /g' |
    tr -d '\\') -no-color -out $PLANFILE
31 +       command: terragrunt run-all plan -input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /g' | tr -d '\\') -no-color -out
    $PLANFILE
32 32         output: hide
33 33     - run: |
34 34         terragrunt show $PLANFILE
↕ @@ -38,7 +38,7 @@ workflows:
38 38         # Reduce Terraform suggestion output
39 39         name: TF_IN_AUTOMATION
40 40         value: "true"
41 -     - run: terragrunt run-all apply --terragrunt-working-dir .. -input=false $PLANFILE
41 +     - run: terragrunt run-all apply -input=false $PLANFILE
42 42     - env:
43 43         name: TERRAGRUNT_NON_INTERACTIVE
44 44         value: "true"
↕ @@ -48,7 +48,7 @@ workflows:
48 48         name: TF_VAR_author
49 49         command: 'git show -s --format="%ae" $HEAD_COMMIT'
50 50         # Allow for imports as not supported for Terraform wrappers by default
51 -     - run: terragrunt run-all import --terragrunt-working-dir .. -input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /' | tr -d
    '\\')
51 +     - run: terragrunt run-all import -input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /' | tr -d '\\')
52 52     state: rm;
```

Episode 3: All in all

rikstv / sre / RikstV.DockerImages.Atlantis / Merge requests / !35

Merged Implement run-all feature/runall into main

Overview 0 Commits 8 Pipelines 1 **Changes 4**

Add a to-do item



Use `run-all` across the board

Marco Marongiu authored 3 months ago



6046dafe



atlantis-repos-config.yaml

+2 -2 Viewed



@@ -31,7 +31,7 @@ workflows:

31	31	command: terragrunt run-all plan -input=false \$(printf '%s' \$COMMENT_ARGS sed 's/,/ /g' tr -d '\\')
		\$PLANFILE
32	32	output: hide
33	33	- run:
34	-	terragrunt show \$PLANFILE
	34	+ terragrunt run-all show \$PLANFILE

35	35	apply:
36	36	steps:
37	37	- env:



@@ -51,4 +51,4 @@ workflows:

51	51	- run: terragrunt run-all import -input=false \$(printf '%s' \$COMMENT_ARGS sed 's/,/ /' tr -d '\\')
52	52	state_rm:
53	53	steps:
54	-	- run: terragrunt state rm \$(printf '%s' \$COMMENT_ARGS sed 's/,/ /' tr -d '\\')
	54	+ - run: terragrunt run-all state rm \$(printf '%s' \$COMMENT_ARGS sed 's/,/ /' tr -d '\\')

Episode 3: All in all (cont.)

```
| Error: Failed to load plugin schemas
```

```
| Error while loading schemas for plugin components: 2 problems:
```

```
| - Failed to obtain provider schema: Could not load the schema for provider  
| registry.terraform.io/hashicorp/helm: failed to instantiate provider  
| "registry.terraform.io/hashicorp/helm" to obtain schema: unavailable  
| provider "registry.terraform.io/hashicorp/helm".  
| - Failed to obtain provider schema: Could not load the schema for provider  
| registry.terraform.io/magodo/restful: failed to instantiate provider  
| "registry.terraform.io/magodo/restful" to obtain schema: unavailable  
| provider "registry.terraform.io/magodo/restful"..
```

Episode 4: Grand finale

rikstv / sre / RikstV.DockerImages.Atlantis / Merge requests / !35

Merged Implement run-all feature/runall into main

Overview 0 Commits 8 Pipelines 1 Changes 4

Add a to-do item

atlantis-repos-config.yaml

+3 -3 Viewed

```
@@ -28,17 +28,17 @@ workflows:
  28 28         value: "true"
  29 29     - run:
  30 30         # Allow for targetted plans/applies as not supported for Terraform wrappers by default
  31 -         command: terragrunt run-all plan -input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /g' | tr -d '\\') -no-color -out
           $PLANFILE
  31 +         command: terragrunt run-all plan -input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /g' | tr -d '\\') -no-color -out
           atlantis.tfplan
  32 32         output: hide
  33 33     - run: |
  34 -         terragrunt run-all show $PLANFILE
  34 +         terragrunt run-all show atlantis.tfplan
  35 35     apply:
  36 36     steps:
  37 37     - env:
  38 38         # Reduce Terraform suggestion output
  39 39         name: TF_IN_AUTOMATION
  40 40         value: "true"
  41 -         - run: terragrunt run-all apply -input=false $PLANFILE
  41 +         - run: terragrunt run-all apply -input=false atlantis.tfplan
  42 42     - env:
  43 43         name: TERRAGRUNT_NON_INTERACTIVE
  44 44         value: "true"
```

A peek in the Atlantis container

- `.../__selftest__/nonprod/.terragrunt-cache/GE.../0u.../_stack/atlantis.tfplan`
- `.../__selftest__/uat/.terragrunt-cache/yy.../0u.../_stack/atlantis.tfplan`
- `.../__selftest__/prod/.terragrunt-cache/0e.../0u.../_stack/atlantis.tfplan`

Episode 5: The final touch

rikstv / sre / RikstV.DockerImages.Atlantis / Merge requests / !40

Merged Improve workflow, upgrade Atlantis feature/debug-atlantis-con... into main

Overview 0 Commits 14 Pipelines 14 Changes 4

Add a to-do item

atlantis-repos-config.yaml

+53 -49 Viewed

```
29         - run:
30             # Allow for targetted plans/applies as
             not supported for Terraform wrappers by default
```

```
31 -         command: terragrunt run-all plan -input=false
           $(printf '%s' $COMMENT_ARGS | sed 's/,/ /g' | tr -d '\n' -n
           color -out atlantis.tfplan
```

```
32         output: hide
```

```
33         - run: |
```

```
34 -         terragrunt run-all show atlantis.tfplan
```

```
35         apply:
```

```
36         steps:
```

```
37         - env:
```

```
@@ -41,7 +45,7 @@ workflows:
```

```
41         - env:
```

```
42             name: TERRAGRUNT_NON_INTERACTIVE
```

```
43             value: "true"
```

```
44 -         - run: terragrunt run-all apply -input=false
           atlantis.tfplan
```

```
45         import:
```

```
46         steps:
```

```
47         - env:
```

```
33         - run:
34             # Allow for targetted plans/applies as
             not supported for Terraform wrappers by default
```

```
35 +         command: terragrunt run-all plan -
           input=false $(printf '%s' $COMMENT_ARGS | sed 's/,/ /g' | tr -d
           '\n') -no-color -out ../../../../atlantis.tfplan
```

```
36         output: hide
```

```
37         - run: |
```

```
38 +         terragrunt run-all show ../../../../
           atlantis.tfplan
```

```
39         apply:
```

```
40         steps:
```

```
41         - env:
```

```
@@ -41,7 +45,7 @@ workflows:
```

```
45         - env:
```

```
46             name: TERRAGRUNT_NON_INTERACTIVE
```

```
47             value: "true"
```

```
48 +         - run: terragrunt run-all apply -input=false
           ../../../../atlantis.tfplan
```

```
49         import:
```

```
50         steps:
```

```
51         - env:
```

terragrunt-out-dir

Ran Apply for 6 projects:

```
{
  "level": "warn",
  "ts": "2025-01-19T16:47:27.438Z",
  "caller": "events/apply_command_runner.go:223",
  "msg": "unable to update commit status: POST
1 https://mygitserver.example.com/api/v4/projects/rikstv/sre/rikstv.terraform.infra.a
▶ tlantistesting/statuses/682ac035b55d8193a729b02edef6f8e71c8944ab: 400 {message:
2 Cannot transition status via :run from :running (Reason(s): Status cannot
  transition via \"run\")}",
  "json": {
    "repo": "rikstv/sre/rikstv.terraform.infra.atlantistesting",
    "pull": "15"
  },
  "stacktrace": "github.com/runatlantis/atlan...
4 (*ApplyCommandRunner).updateCommitStatus\n\tgithub.com/runatlantis/atlan...
  server/events/apply_command_runner.go:223\n\tgithub.com/runatlantis/atlan...
  events.(*ApplyCommandRunner).Run\n\tgithub.com/runatlantis/atlan...
5 apply_command_runner.go:181\n\tgithub.com/runatlantis/atlan...
  (*DefaultCommandRunner).RunCommentCommand\n\tgithub.com/runatlantis/atlan...
6 server/events/command_runner.go:383"
}
```

Apply Summary

6 projects, 6 successful, 0 failed, 0 errored

✔ atlantis/pre_workflow_hook: Pre workflow ...

The recipe, summarised

- Enable provider caching
- Start from the standard terragrunt workflow
- Check which parts of the code you can consider stacks, and mark them clearly for Atlantis in some way...
- ...or, if you add terragrunt-atlantis-config, make it recognise stacks correctly (we used the pre-existing context.hcl, in your case it may be different)
- Replace all terragrunt commands with terragrunt run-all
- Replace \$PLANFILE with a relative path (must use .tfplan as the extension, land outside the terragrunt cache, and never clash with other plans)

Look out!

- Atlantis is active, not yet mature: has bugs, slow releases
- with Gitlab, use at least 0.31
- `atlantis apply` not working properly
- may break augmented terraform command-line options

Questions?



THANK YOU FOR ATTENDING!

**This presentation will soon be available on
syslog.me for download.**

References and attributions

- Atlantis' terragrunt custom workflow:
<https://www.runatlantis.io/docs/custom-workflows.html#terragrunt>
- terragrunt-atlantis-config:
<https://github.com/transcend-io/terragrunt-atlantis-config>
- Atlantis on Fargate terraform module
<https://registry.terraform.io/modules/terraform-aws-modules/atlantis/aws/latest>
- <https://github.com/runatlantis/atlantis/issues/3280>