ITALY
OpenInfra Days

Scaling Terraform From Startup to Enterprise

Paolo Tonin - Flowing

FLOWING® ideato

Paolo Tonin

Cloud & DevOps

paolo.tonin@flowing.it

Scaling Terraform

- **Intro**
- What is Terraform
- 4 Stages of Adoption
- Stage Next

Scaling Terraform

- ~~**Intro**~~
- **What is Terraform**
- 4 Stages of Adoption
- Stage Next

FLOWING® ideato

HashiCorp

Terraform

Write, Plan, and Create Infrastructure as Code

From Infrastructure Services to multiple Cloud

Not IaaS services

Cloudflare

UltraDNS

Fastly

BitBucket

Datadog

(and more…)

Virtualization and IaaS

OpenStack

OpenNebula

VMware vCloud

Softlayer

Cloudstack

(and more…)

Cloud provider

Amazon

Google Cloud

Microsoft Azure

DigitalOcean

Alibaba Cloud

(and more…)

Pro

- Multiple Provider
- Low learning curve
- Declarative HCL language, not procedural code
- Dry runs
- Infrastructure versioning
- Terraform Registry

Cons

- No rolling upgrades*
- HCL Limitations (logic, loops)*
- Not “Drift detection”*

Scaling Terraform

- ~~**Intro**~~
- ~~**What is Terraform**~~
- **4 Stages of Adoption**
  - Stage 1 Manual
  - Stage 2 Semi Automated
  - Stage 3
  - Stage 4
- Stage Next

FLOWING® ideato

4 Stage of Adoption

FLOWING® ideato

Stage 1

Scaling Terraform - Stage 1 Manual

- Web Consoles / Manual CLI
- Single environment(s)
- Mutable Infrastructure (Pets server)

Scaling Terraform - Stage 1 Manual

- Web Consoles / Manual CLI
- Single environment(s)
- Mutable Infrastructure (Pets server)
- *“Infrastructure as Code”* 🤔

Scaling Terraform - Stage 1 Manual

- Web Consoles / Manual CLI
- Single environment(s)
- Mutable Infrastructure (Pets server)
- ~~*“Infrastructure as Code”*~~ ops.txt

- ....

Scaling Terraform - Stage 1 Manual

Technical	Operational
Reproducibility	**Auditing**
Change Management	**Consistency**
Architecture	**Knowledge Sharing**

Scaling Terraform - Stage 1 Manual

- ~~**Intro**~~
- ~~**What is Terraform**~~
- **4 Stages of Adoption**
  - ~~Stage 1 Manual~~
  - Stage 2 Semi automated
  - Stage 3
  - Stage 4
- Stage Next

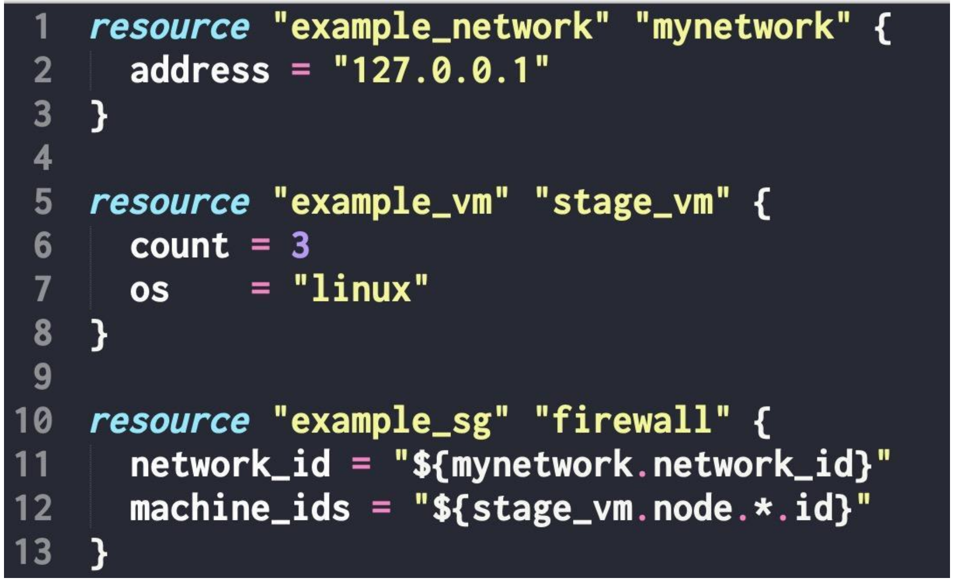
FLOWING® ideato

Stage 2 - Semi automated

Scaling Terraform - Stage 2 Semi automated

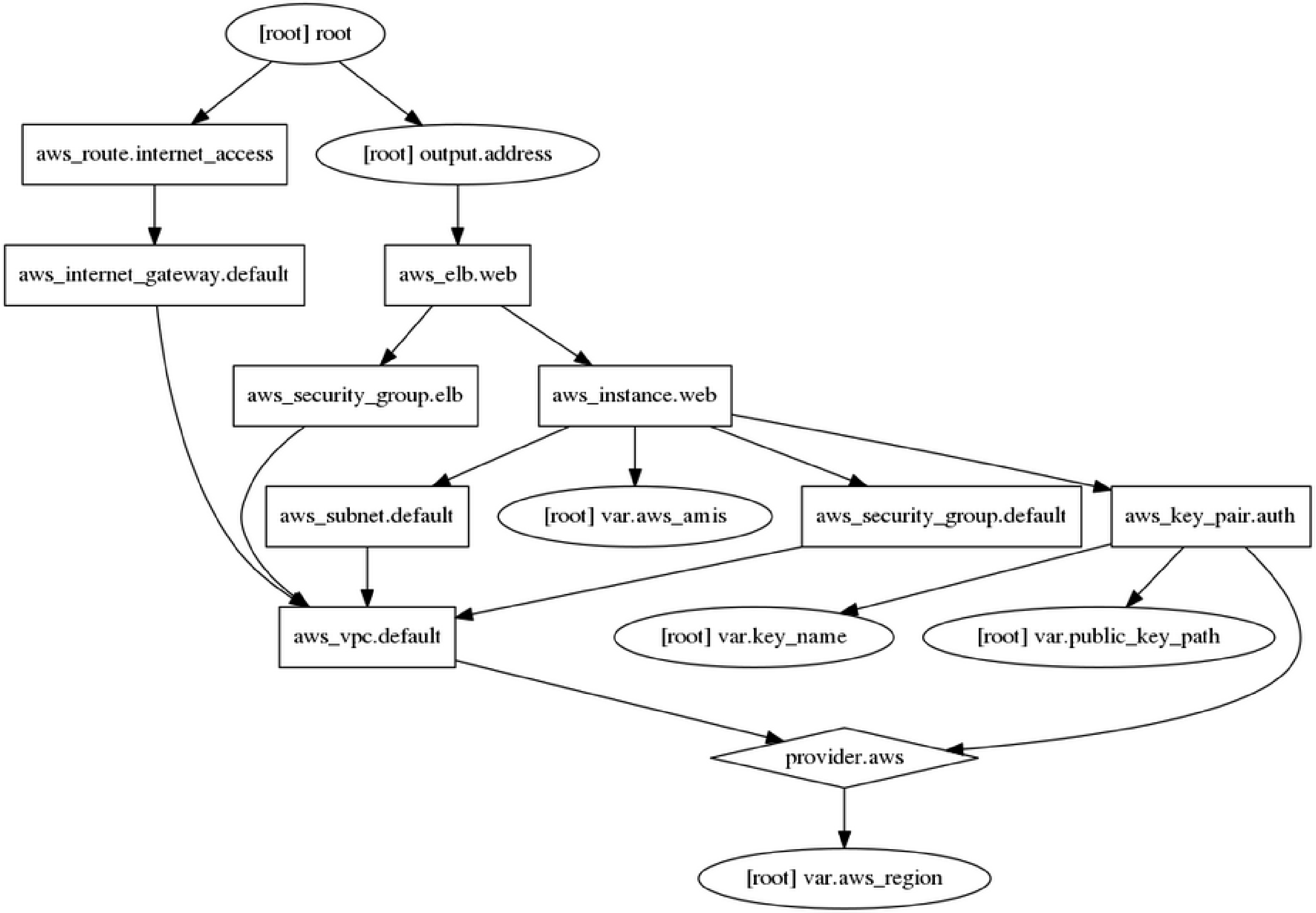
- Adopting Infrastructure As Code (also Dockerfiles)
- Machine images (also playbooks and others / Packer)
- Web Consoles
- **Introduce Terraform!**
  - **Configuration!**
  - **Automation!**
  - **Iteration!**

Modeling Infrastructure with Configuration



```
resource "example_network" "mynetwork" {
  address = "127.0.0.1"
}

resource "example_vm" "stage_vm" {
  count = 3
  os    = "linux"
}

resource "example_sg" "firewall" {
  network_id = "${mynetwork.network_id}"
  machine_ids = "${stage_vm.node.*.id}"
}
```

- **Plan, Apply, Iterate**



FLOWING® ideato

Scaling Terraform - Stage 2 Semi automated

Technical	Operational
Reproducibility	Auditing
Change Management	Consistency
Architecture	Knowledge Sharing

FLOWING® ideato

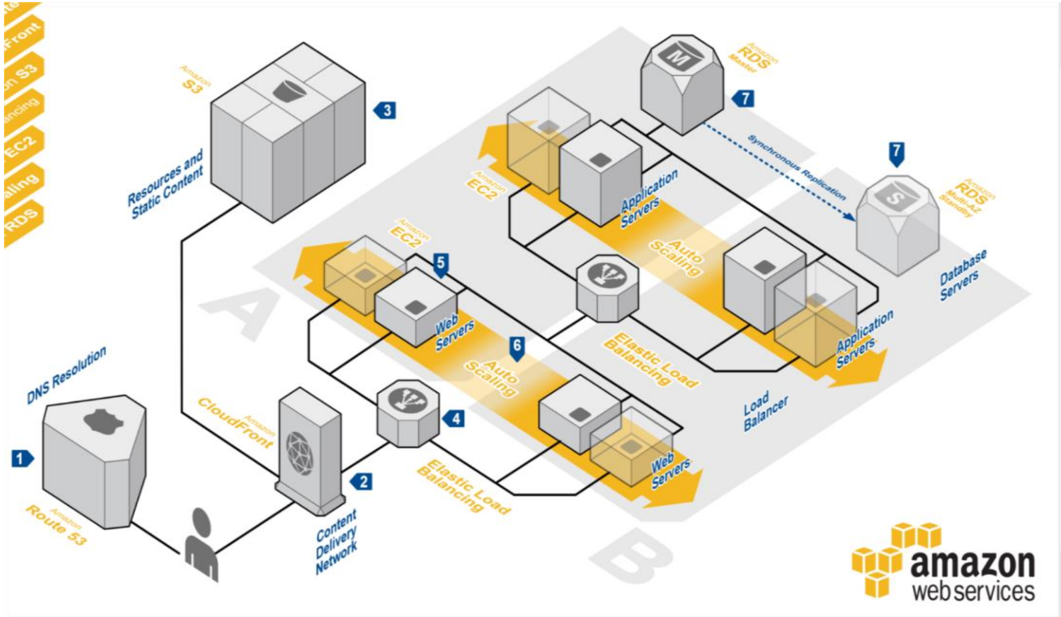
Scaling Terraform - Stage 2 Semi automated

Technical	Operational
Reproducibility	Auditing
Change Management	Consistency
Architecture	Knowledge Sharing
State management	Operations

Let’s “import” our infrastructure

FLOWING®
ideato

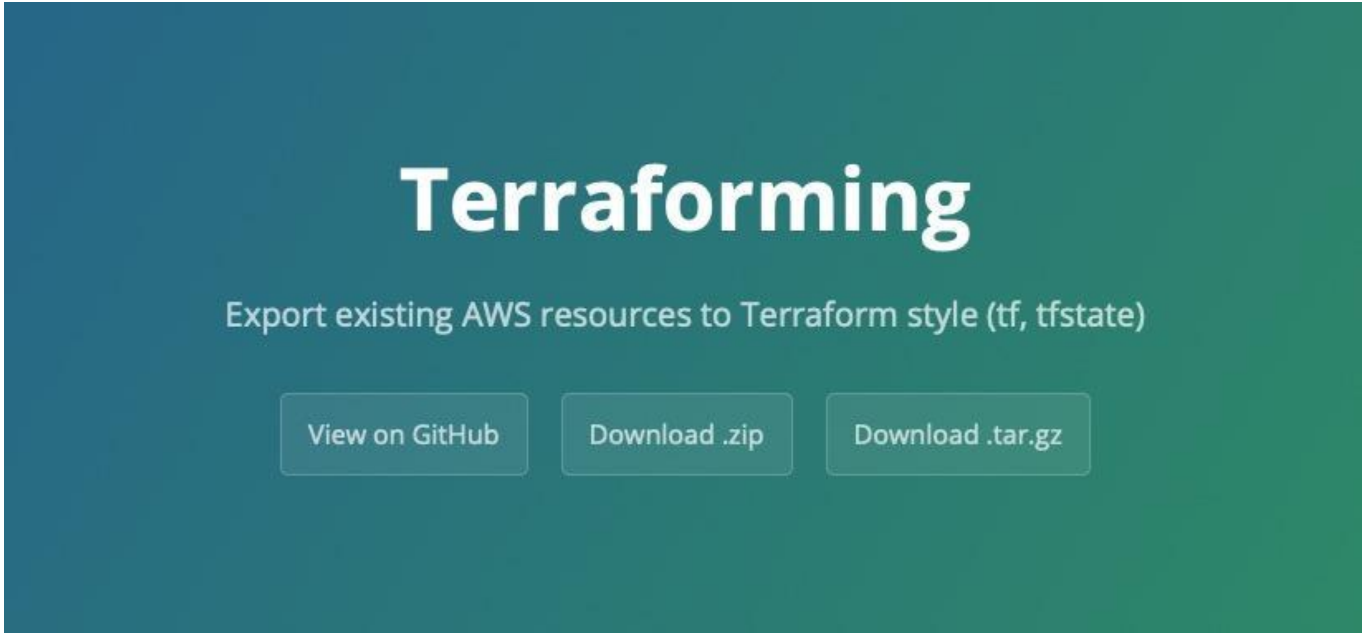
Scaling Terraform - Stage 2 Semi automated



Scaling Terraform - Stage 2 Semi automated

```
$ cat ec2_instance.tf

resource "aws_instance" "prod_ec2" {

  # ...instance configuration...

}

$ terraform import aws_instance.prod_ec2 i-abcd1234
```

Scaling Terraform - Stage 2 Semi automated



Terraforming

Export existing AWS resources to Terraform style (tf, tfstate)

Scaling Terraform

- ~~**Intro**~~
- ~~**What is Terraform**~~
- **4 Stages of Adoption**
  - ~~Stage 1 Manual~~
  - ~~Stage 2 Semi automated~~
  - Stage 3
  - Stage 4
- Stage Next

FLOWING® ideato

Stage 3

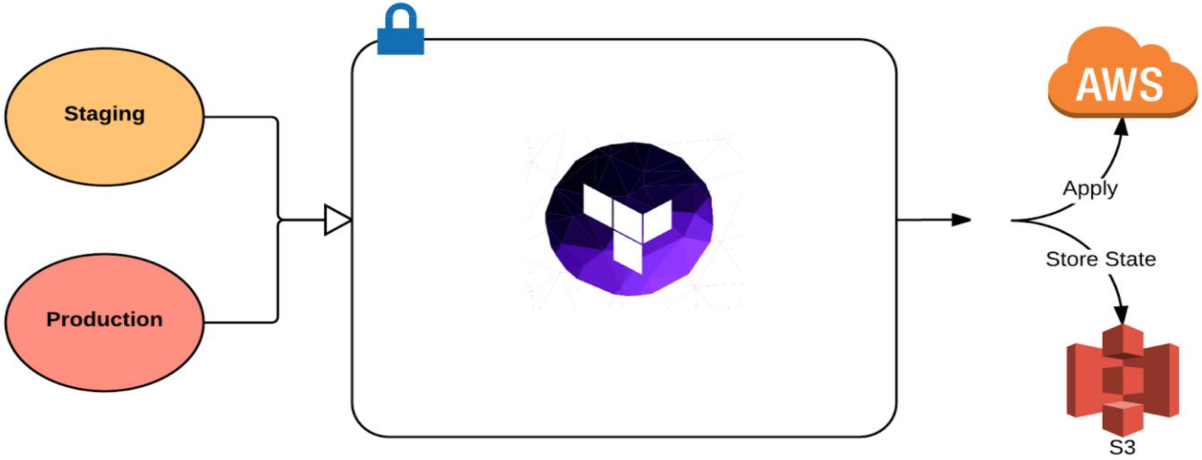
- Workspaces
- Packaged Components
- Modules
- Inputs and Outputs
- Config Mgmt Hooks

Scaling Terraform - Stage 3

- Organizational adoption
- Multiple Environments
- Collaboration
- Version Control

Scaling Terraform - Stage 3

- Organizational adoption
- Multiple Environments (Workspaces)



Scaling Terraform - Stage 3

- Organizational adoption
- Multiple Environments (Workspaces)

```
$ terraform workspace new staging

You're now on a new, empty workspace. Workspaces isolate
their state, so if you run "terraform plan" Terraform
will not see any existing state for this configuration.
```

Scaling Terraform - Stage 3

- Organizational adoption
- Multiple Environments (Workspaces) - separate state files

```
resource "aws_instance" "example" {

  count = "${terraform.workspace == "prod" ? 5 : 1}"

  # … other params

}
```

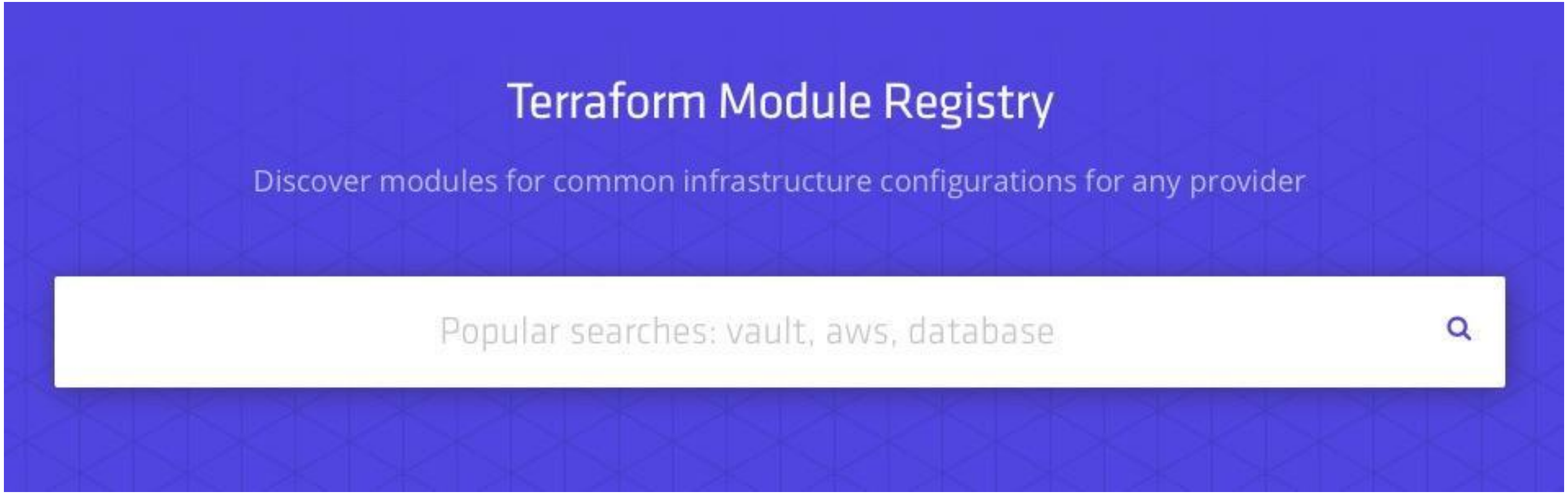
Scaling Terraform - Stage 3

- Organizational adoption
- Multiple Environments (Workspaces) - separate state files

```
resource "aws_vpc" "mycompany_vpc" {

  name = "VPC ${terraform.workspace == "default" ?
“Prod” : “Staging”}"

  # … other params

}
```

- ~~Workspaces~~
- Packaged Components

Scaling Terraform - Stage 3



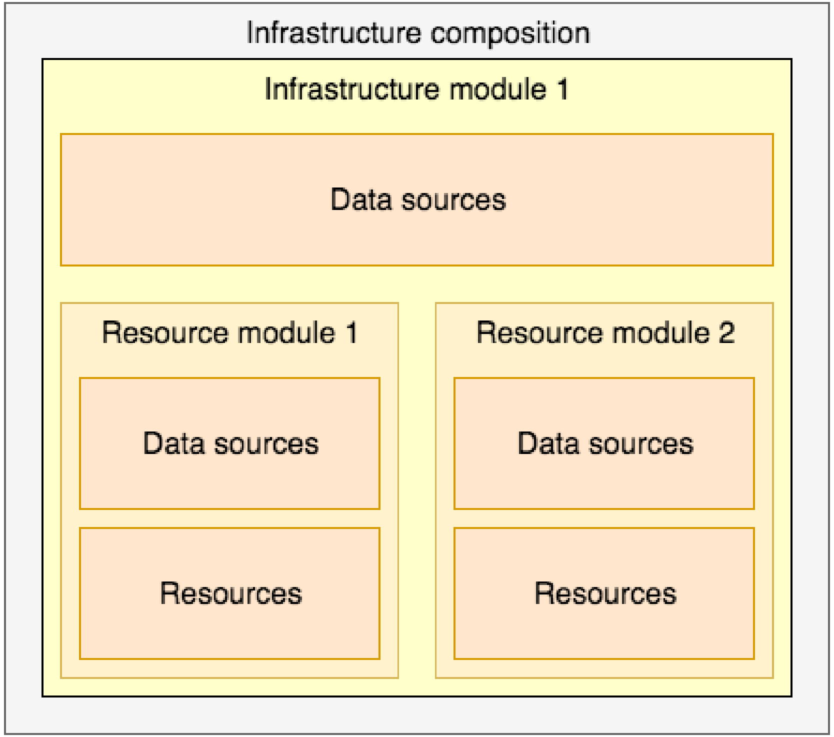
Terraform Module Registry

Discover modules for common infrastructure configurations for any provider

Popular searches: vault, aws, database

- ~~Workspaces~~
- ~~Packaged Components~~
- Modules

Scaling Terraform - Stage 3



Infrastructure composition

Infrastructure module 1

Data sources

Resource module 1

Data sources

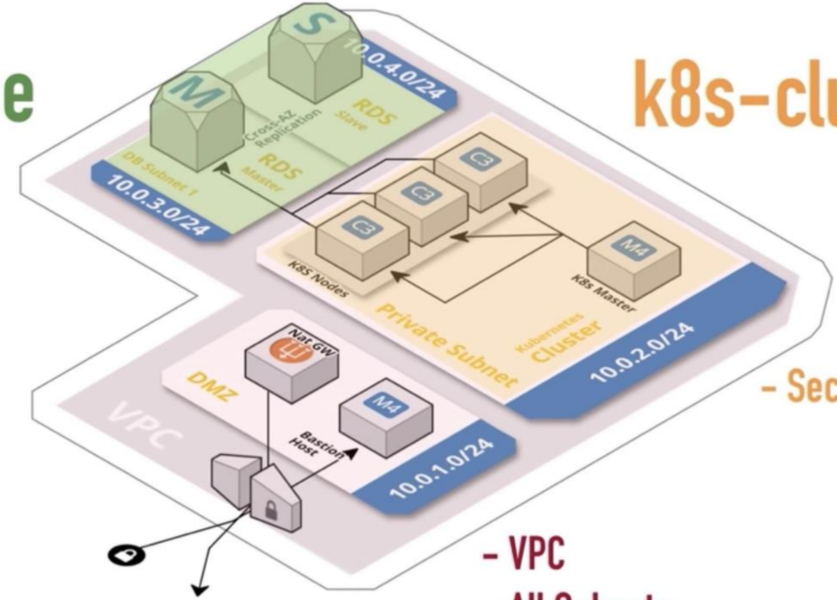
Resources

Resource module 2

Data sources

Resources

Scaling Terraform - Stage 3



database

- Amazon RDS
- DB Subnet Group

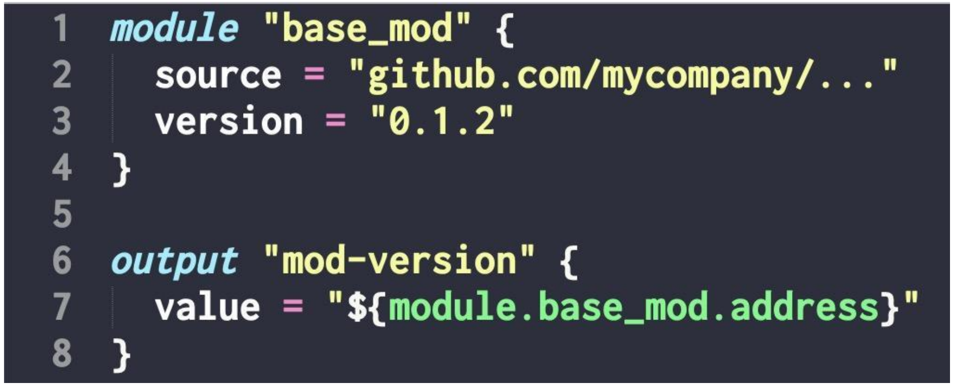
k8s-cluster

- Instances
- Security Groups

core

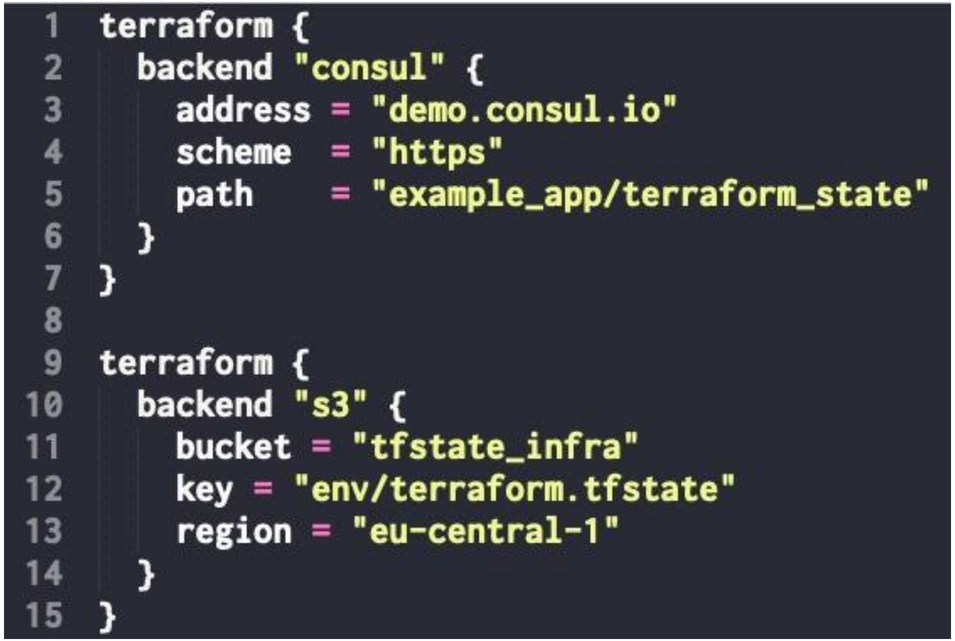
- VPC
- All Subnets
- Core Routing & Gateways
- Bastion Host (OpenVPN server)

```
$ tree base-mod/

├── README.md

├── main.tf

└── variables.tf

└── outputs.tf
```



```
module "base_mod" {
  source = "github.com/mycompany/..."
  version = "0.1.2"
}

output "mod-version" {
  value = "${module.base_mod.address}"
}
```

- Version control (PR)
- Remote state



```
terraform {
  backend "consul" {
    address = "demo.consul.io"
    scheme  = "https"
    path    = "example_app/terraform_state"
  }
}

terraform {
  backend "s3" {
    bucket = "tfstate_infra"
    key = "env/terraform.tfstate"
    region = "eu-central-1"
  }
}
```

Scaling Terraform - Stage 3

Technical	Operational
Reproducibility	Auditing
Change Management	Consistency
Architecture	Knowledge Sharing
State management	Operations

Scaling Terraform - Stage 3

Technical	Operational
Reproducibility	Auditing
Change Management	Consistency
Architecture	Knowledge Sharing
State management	Operations

Scaling Terraform - Stage 3

- ~~**Intro**~~
- ~~**What is Terraform**~~
- **4 Stages of Adoption**
  - ~~Stage 1 Manual~~
  - ~~Stage 2 Semi automated~~
  - ~~Stage 3~~
  - Stage 4
- Stage Next

FLOWING® ideato

FLOWING® ideato

Stage 4

- VCS Integration
- Team permissions
- “Run Terraform for me”

Developer

“I need to create new VMs for my new env; but i don’t know yet how many”

FLOWING®
ideato

Operations
“Ok, it’s not a problem!”

FLOWING®
ideato

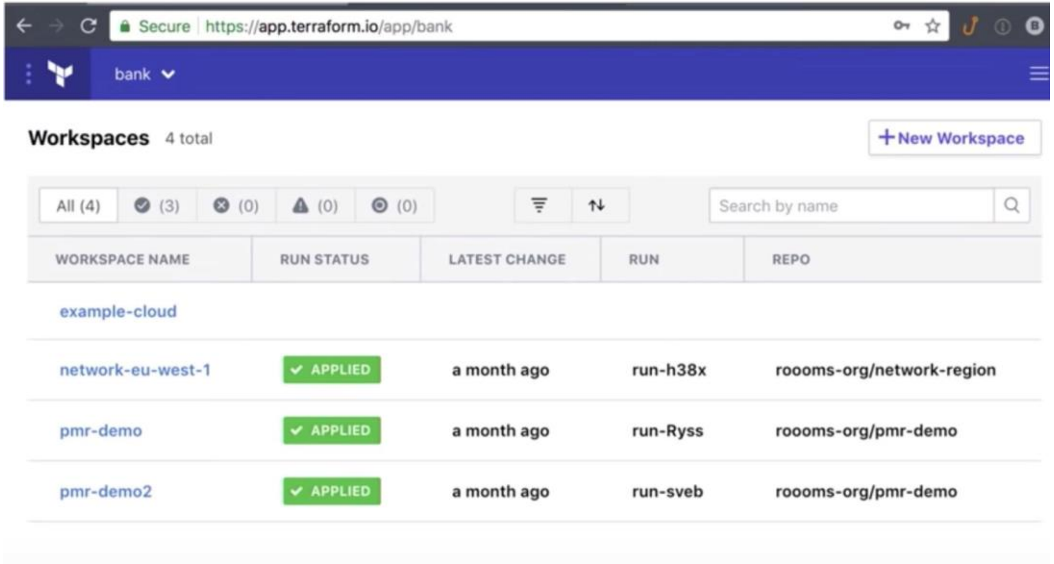
Scaling Terraform - Stage 4

Operations
Create new Reservation (Scheduled Reserved or Convertible)
Create new IAM and assign permission for this RI
Restrict IAM Policy for using RI and EC2 family

Scaling Terraform - Stage 4

Developers
Configure AWS CLI with IAM user
Create Terraform module for EC2 resources

Terraform Enterprise

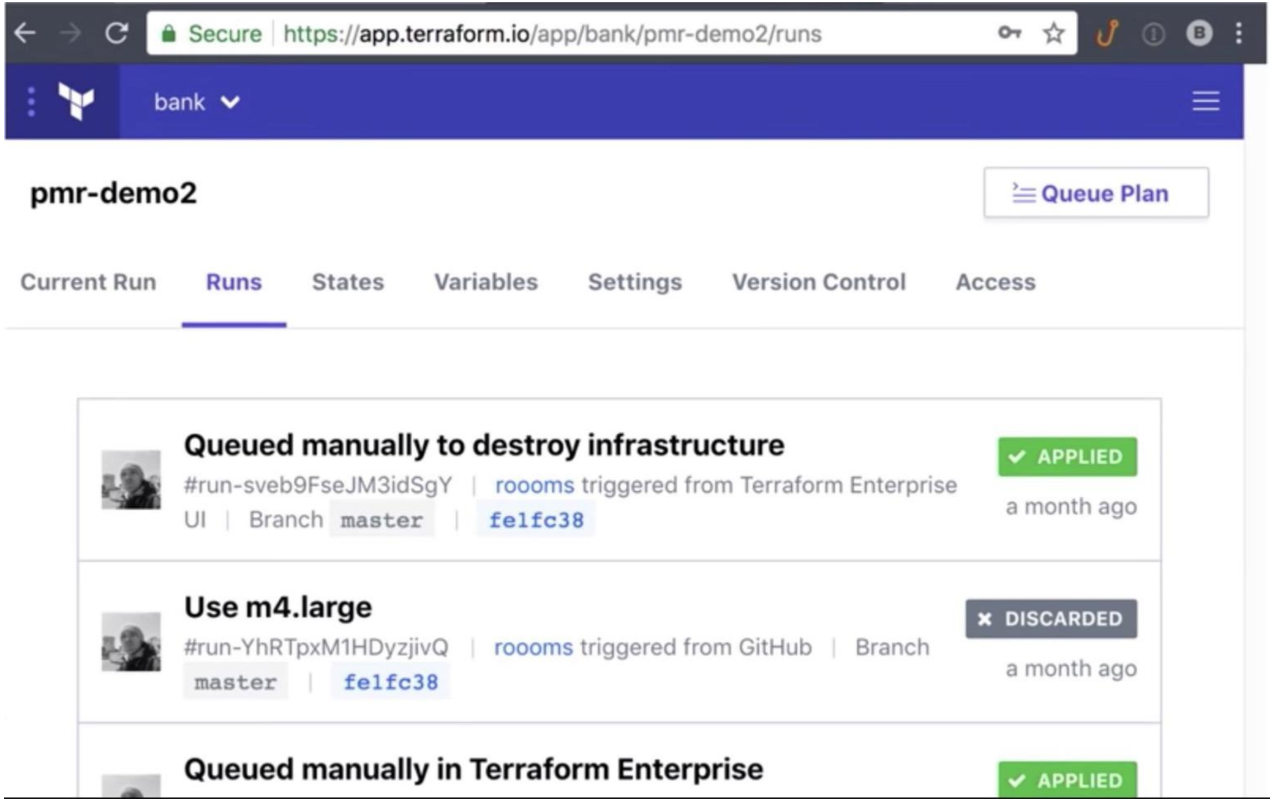


Workspaces 4 total

All (4) | ✓ (3) | ☒ (0) | ▲ (0) | ● (0)

WORKSPACE NAME	RUN STATUS	LATEST CHANGE	RUN	REPO
example-cloud				
network-eu-west-1	✓ APPLIED	a month ago	run-h38x	roooms-org/network-region
pmr-demo	✓ APPLIED	a month ago	run-Ryss	roooms-org/pmr-demo
pmr-demo2	✓ APPLIED	a month ago	run-sveb	roooms-org/pmr-demo

Terraform Enterprise



pmr-demo2

Queued manually to destroy infrastructure
#run-sveb9FseJM3idSgY | rooooms triggered from Terraform Enterprise UI | Branch `master` | `fe1fc38`
✓ APPLIED
a month ago

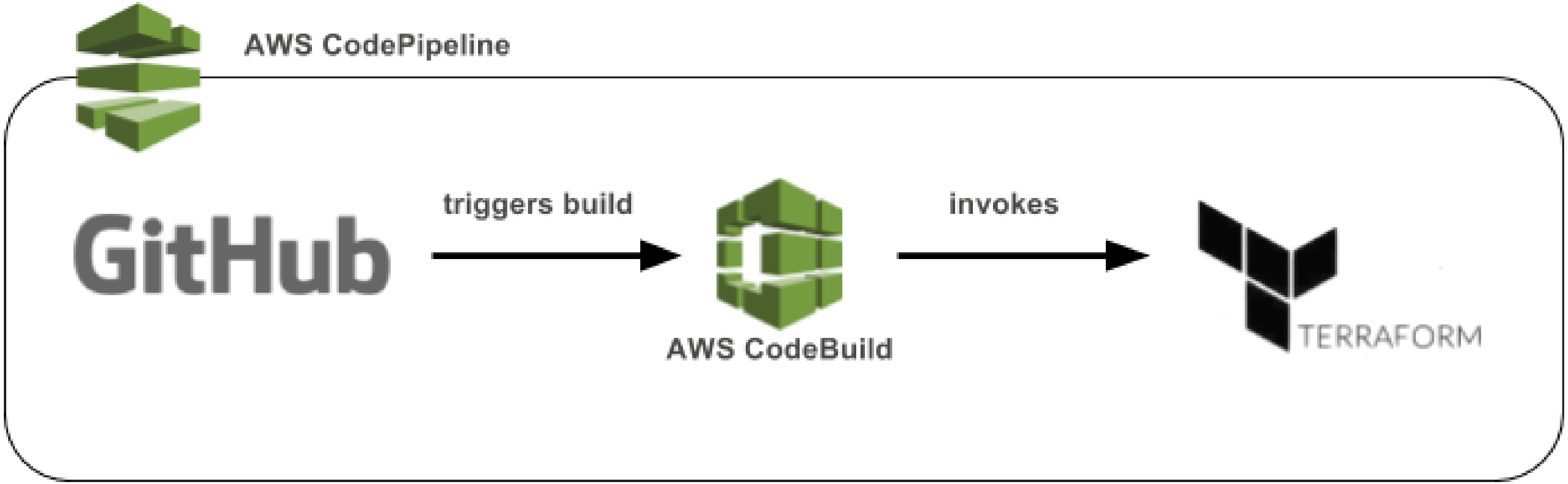
Use m4.large
#run-YhRTpxM1HDyzjivQ | rooooms triggered from GitHub | Branch `master` | `fe1fc38`
✖ DISCARDED
a month ago

Queued manually in Terraform Enterprise
✓ APPLIED

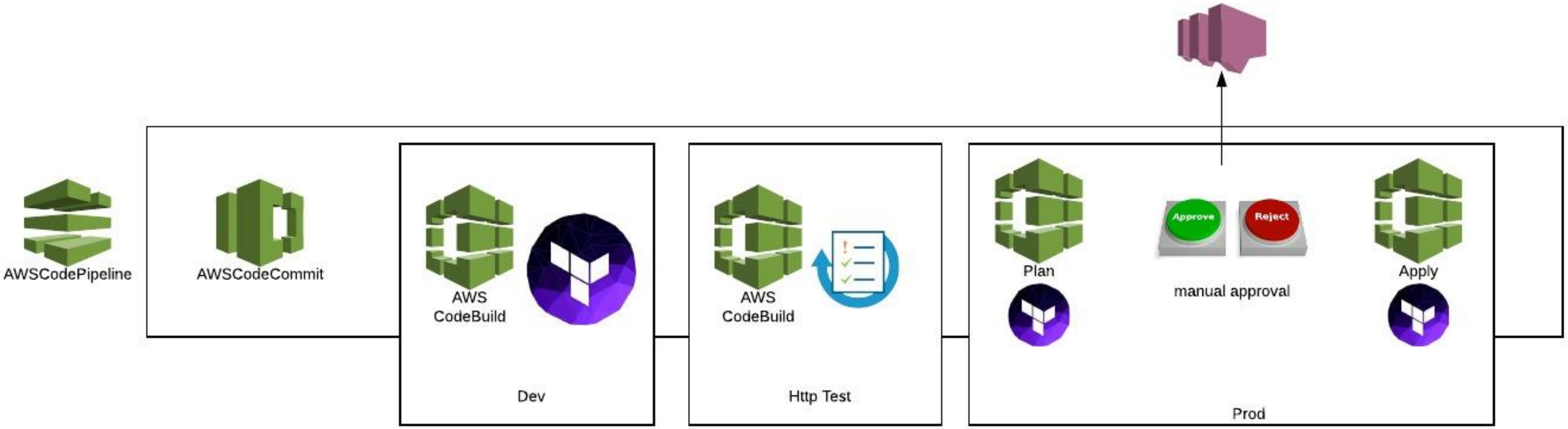
“Run Terraform for me”

FLOWING®
ideato

Scaling Terraform - Stage 4



Scaling Terraform - Stage 4



Grazie!

FLOWING® ideato

Paolo Tonin

Cloud & DevOps

paolo.tonin@flowing.it