Terraform

Infrastructure as Code

AWS Cloud

REQUIREMENTS:

- 3 ENVIRONMENTS – DEV, TST, PRD
  - DEV
    - 1 SERVER = ~~T2.SMALL~~ → T2.MEDIUM
  - TST
    - 1 SERVER = ~~T2.SMALL~~ → T2.MEDIUM
  - PRD
    - 2 SERVERS = ~~T2.MEDIUM~~ → T2.LARGE

NON-PRD ACCOUNT

DEV

DEV-INSTANCE-1

~~T2.SMALL~~

T2.MEDIUM

TST

TST-INSTANCE-1

~~T2.SMALL~~

T2.MEDIUM

PRD ACCOUNT

PRD

PRD-INSTANCE-1

~~T2.MEDIUM~~

T2.LARGE

PRD-INSTANCE-2

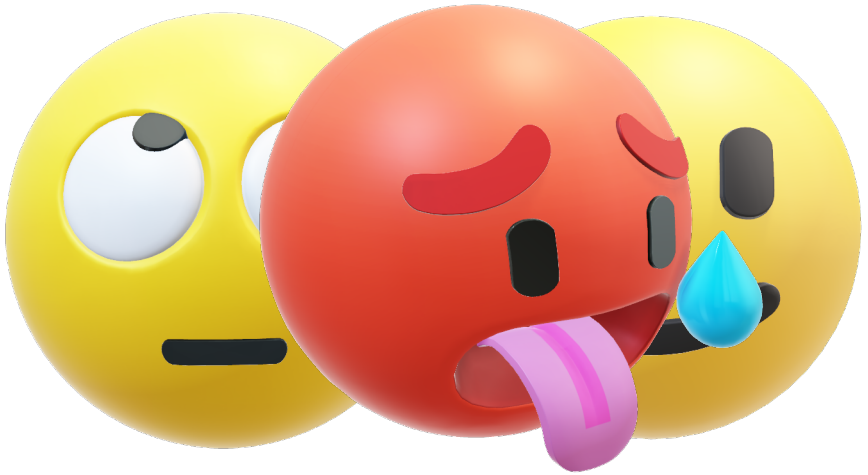
~~T2.MEDIUM~~

T2.LARGE

AWS Cloud

REQUIREMENTS:

- 3 ENVIRONMENTS – DEV, TST, PRD
  - DEV
    - ~~1~~ SERVER → 2 SERVERS
      - ~~T2.MEDIUM~~ → T2.LARGE
  - TST
    - ~~1~~ SERVER → 2 SERVERS
      - ~~T2.MEDIUM~~ → T3.LARGE
  - PRD
    - ~~2~~ SERVERS → 4 SERVERS
      - ~~T2.LARGE~~ → M5.XLARGE



NON-PRD ACCOUNT

DEV

DEV-INSTANCE-1

~~T2.MEDIUM~~

T2.LARGE

DEV-INSTANCE-2

T2.LARGE

TST

TST-INSTANCE-1

~~T2.MEDIUM~~

T3.LARGE

TST-INSTANCE-2

T3.LARGE

PRD ACCOUNT

PRD

PRD-INSTANCE-1

~~T2.LARGE~~

M5.XLARGE

PRD-INSTANCE-3

M5.XLARGE

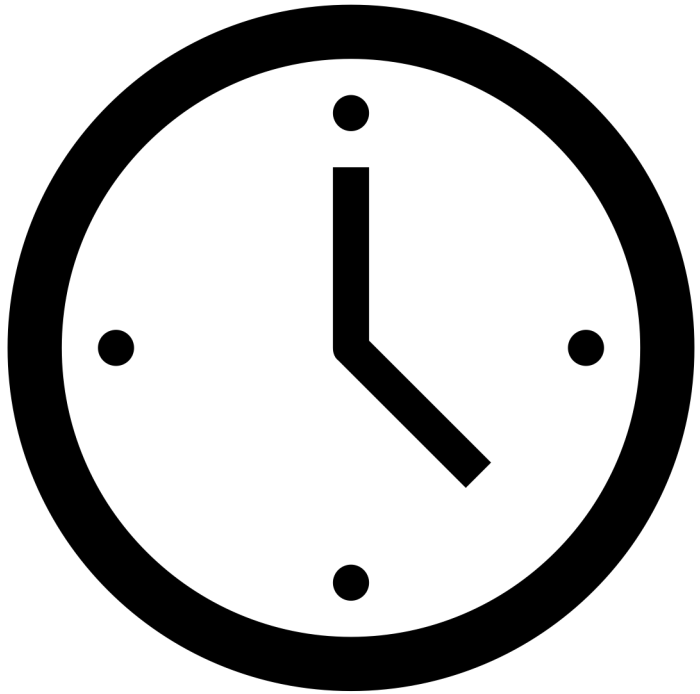
PRD-INSTANCE-2

~~T2.LARGE~~

M5.XLARGE

PRD-INSTANCE-4

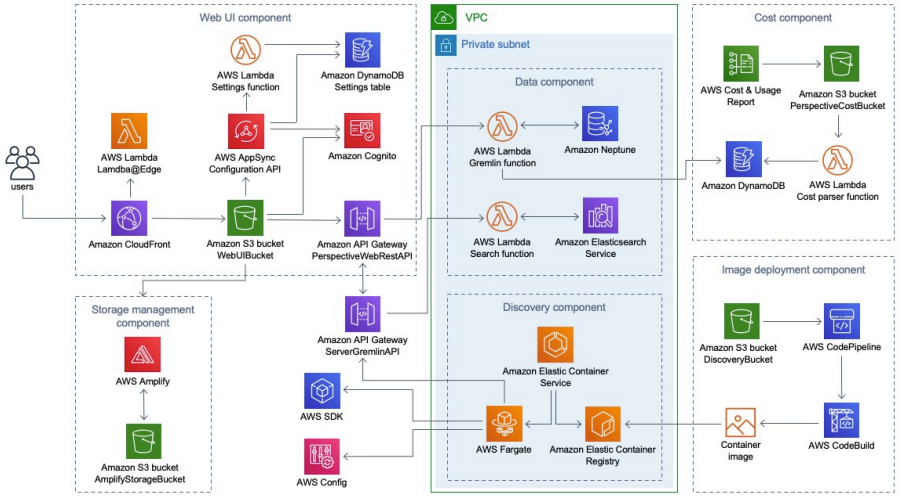
M5.XLARGE



FAST FORWARD 6 MONTHS…

INFRASTRUCTURE NOW HAS ADDITIONAL RESOURCES E.G. DATABASE, ALB, AUTOSCALING, ETC – ALL MANUALLY CREATED ACROSS MULTIPLE ENVIRONMENTS.

NOBODY REMEMBERS *EXACTLY* WHAT THEY DID IN EACH ENVIRONMENT



AND NOW…

NEW EMPLOYEE TO TRAIN

TERRAFORM

- TERRAFORM IS A TOOL FOR BUILDING, CHANGING, AND VERSIONING INFRASTRUCTURE
  - **WRITE -** CONFIGURATION FILES DESCRIBES COMPONENTS NEEDED TO RUN A SINGLE APPLICATION
  - **PLAN -** DESCRIBES WHAT IT WILL DO TO REACH THE DESIRED STATE
  - **APPLY -** EXECUTES IT TO BUILD THE DESCRIBED INFRASTRUCTURE.

Dev Configuration

- Environment = Dev
- Instance Count = 1
- Instance Size = t2.small

Tst Configuration

- Environment = tst
- Instance Count = 1
- Instance Size = t2.small

Prd Configuration

- Environment = prd
- Instance Count = 2
- Instance Size = t2.medium

Let terraform do it for you

DEV

Dev-instance-1

t2.small

TST

Tst-instance-1

t2.small

PRD

PRD-instance-1

t2.medium

PRD-instance-2

t2.medium

Terraform Folder Structure
What does it do?

- Infrastructure
  - Environments
    - Dev
      - Database
        - variables.tfvars
        - terragrunt.hcl
      - EC2-Instance
        - terragrunt.hcl
        - variables.tfvars
    - Tst
      - Database
        - Variables.tfvars
        - Terragrunt.hcl
      - EC2-Instance
        - Variables.tfvars
        - Terragrunt.hcl
    - Prd
      - Database
        - Variables.tfvars
        - Terragrunt.hcl
      - EC2-Instance
        - Variables.tfvars
        - Terragrunt.hcl
  - General Resources
    - Non-prd
      - Account.tfvars
      - General.tfvars
      - Global.tfvars
    - Prd
      - Account.tfvars
      - General.tfvars
      - Global.tfvars

Create an ec2 instance

Here's what I want

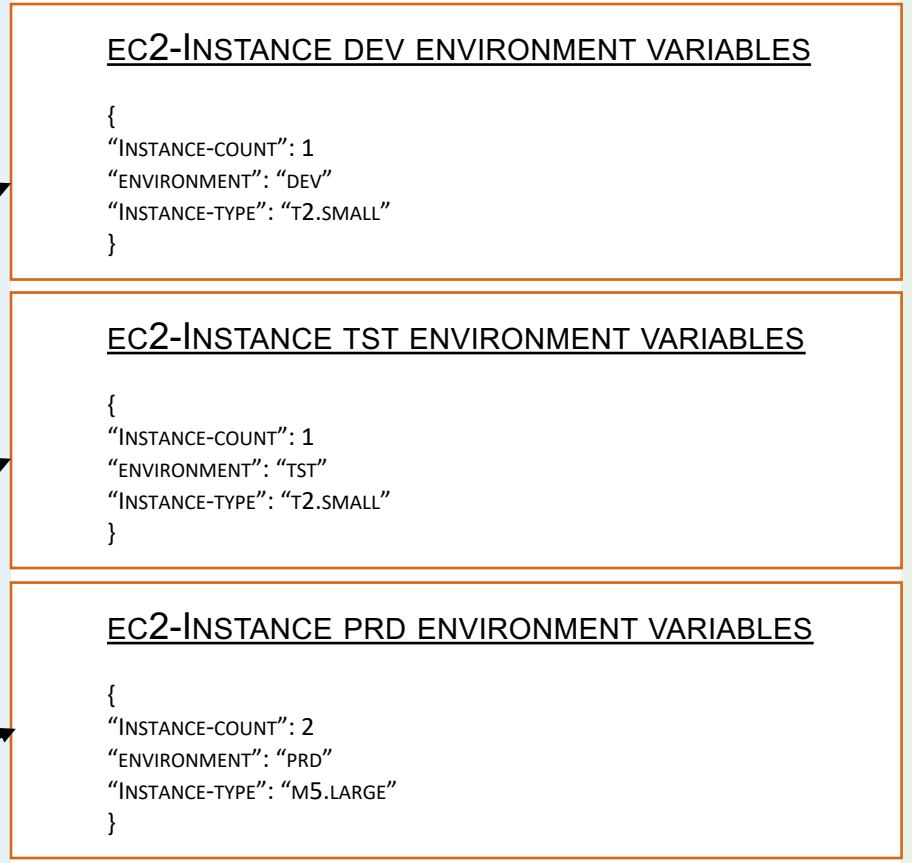
ec2-Instance dev environment variables

{
"Instance-count": 1
"environment": "dev"
"Instance-type": "t2.small"
}

- Modules
  - Database
    - Main.tf
    - Outputs.tf
    - Vars.tf
  - EC2-Instance
    - Main.tf
    - Outputs.tf
    - Vars.tf

How environment variables are defined

- Infrastructure
  - Environments
    - Dev
      - Database
        - variables.tfvars
        - terragrunt.hcl
      - EC2-Instance
        - variables.tfvars
        - terragrunt.hcl
    - Tst
      - Database
        - variables.tfvars
        - terragrunt.hcl
      - EC2-Instance
        - variables.tfvars
        - terragrunt.hcl
    - Prd
      - Database
        - variables.tfvars
        - terragrunt.hcl
      - EC2-Instance
        - variables.tfvars
        - terragrunt.hcl
  - General Resources
    - Non-prd
      - account.tfvars
      - general.tfvars
      - global.tfvars
    - Prd
      - account.tfvars
      - general.tfvars
      - global.tfvars



- Modules
  - Database
    - main.tf
    - outputs.tf
    - vars.tf
  - EC2-Instance
    - main.tf
    - outputs.tf
    - vars.tf

What's in the modules?

- Infrastructure
  - Environments
    - Dev
      - Database
        - variables.tfvars
        - terragrunt.hcl
      - EC2-Instance
        - variables.tfvars
        - terragrunt.hcl
    - Tst
      - Database
        - Variables.tfvars
        - Terragrunt.hcl
      - EC2-Instance
        - Variables.tfvars
        - Terragrunt.hcl
    - Prd
      - Database
        - Variables.tfvars
        - Terragrunt.hcl
      - EC2-Instance
        - Variables.tfvars
        - Terragrunt.hcl
  - General Resources
    - Non-prd
      - Account.tfvars
      - General.tfvars
      - Global.tfvars
    - Prd
      - Account.tfvars
      - General.tfvars
      - Global.tfvars

ec2-instance dev environment variables

```
{
"Instance-count": 1
"environment": "dev"
"Instance-type": "t2.small"
}
```

ec2-instance main.tf

```
resource "aws_instance" "ec2" {
        name = "${var.environment}-instance-${count.index}"
        count = var.instance-count
        instance_type = var.instance
}
```

ec2-instance vars.tf

```
Variable "Instance-count"{}
Variable "environment" {}
Variable "Instance-type"{
        default = "t2.small"
}
```

- Modules
  - Database
    - Main.tf
    - Outputs.tf
    - Vars.tf
  - EC2-Instance
    - Main.tf
    - Outputs.tf
    - Vars.tf

INFRASTRUCTURE

DEV CONFIGURATION

- ENVIRONMENT = DEV
- INSTANCE COUNT = 1
- INSTANCE SIZE = T2.SMALL

PLAN

NO CHANGES

TST CONFIGURATION

- ENVIRONMENT = TST
- INSTANCE COUNT = 1
- INSTANCE SIZE = T2.SMALL

PLAN

NO CHANGES

PRD CONFIGURATION

- ENVIRONMENT = PRD
- INSTANCE COUNT = 2
- INSTANCE SIZE = M5.LARGE

PLAN APPLY

MODIFY INSTANCE SIZE
T2.MEDIUM → M5.LARGE

MODULES

AWS INFRASTRUCTURE

DEV

DEV-INSTANCE-1

T2.SMALL

TST

TST-INSTANCE-1

T2.SMALL

PRD

PRD-INSTANCE-1

T2.MEDIUM

PRD-INSTANCE-2

T2.MEDIUM

Modules

Dev Configuration

- Version = 1.0
- Environment = Dev
- Instance Count = 1
- Instance Size = t2.small

PLAN

No Changes

Dev

- Version 1.0

Dev-instance-1

t2.small

Tst Configuration

- Version = 1.0
- Environment = tst
- Instance Count = 1
- Instance Size = t2.small

PLAN

No Changes

Tst

- Version 1.0

Tst-instance-1

t2.small

PRD Configuration

- version = 1.0
- Environment = prd
- Instance Count = 2
- Instance Size = t2.large

Prd Configuration

- version = 1.1
- Environment = prd
- Instance Count = 2
- Instance Size = M5.large

PLAN

No Changes

Prd

- Version 1.1
- Version 1.0

prd-instance-1

m5.large

prd-instance-2

m5.large

Why use terraform

- Automation
  - deployment process can be automated
  - Speed - computer can carry out the deployment steps far faster than a person
  - Safety - automated process will be more consistent, more repeatable, and not prone to manual error.
- Documentation
  - represent the state of your infrastructure in source files that anyone can read
- Version Control
  - entire history of your infrastructure captured in the commit log
- Validation
  - code review of all changes (plan & apply)
- Reuse
  - package your infrastructure into reusable modules, so that instead of doing every deployment for every product in every environment from scratch, you can build on top of existing modules.