AWS IaC

Good & Bad Practices

Michał Mikołajczyk
AWS Meetup Silesia 6/2024

Cześć!

Michał Mikołajczyk
AWS Solutions Architect / Cloud Engineer

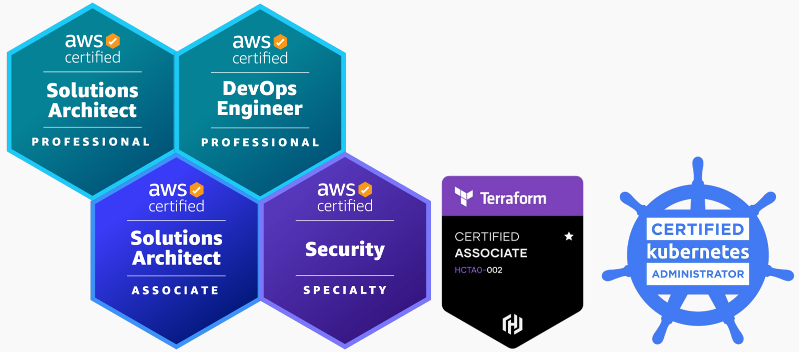
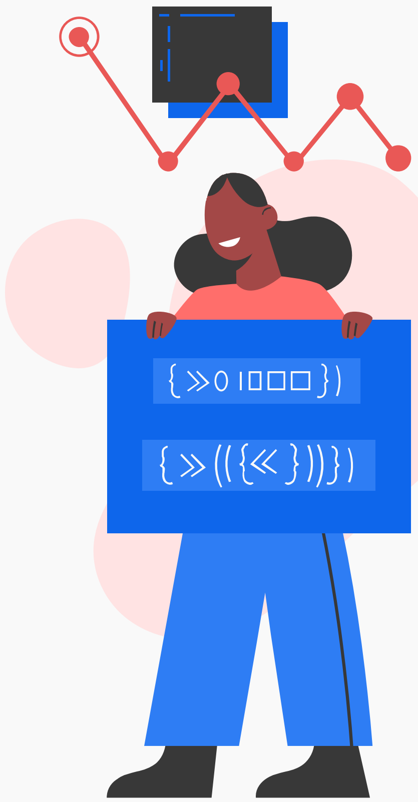


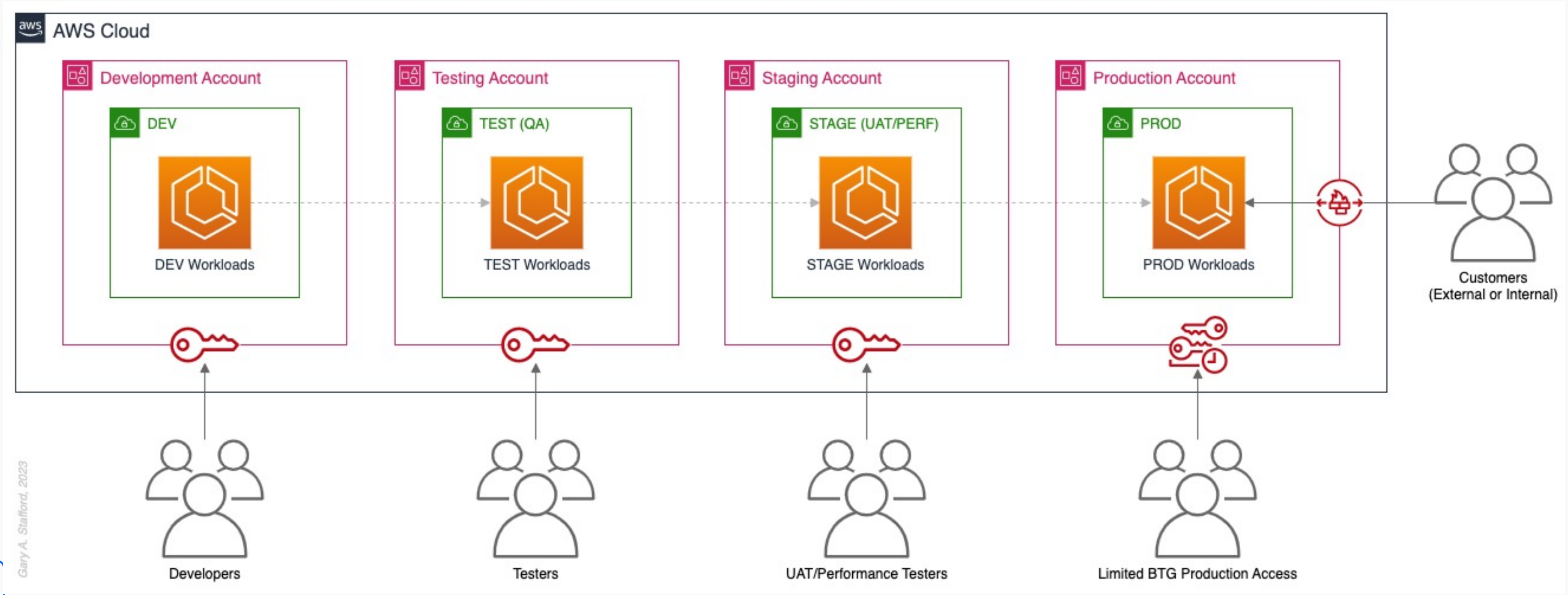
Table of contents

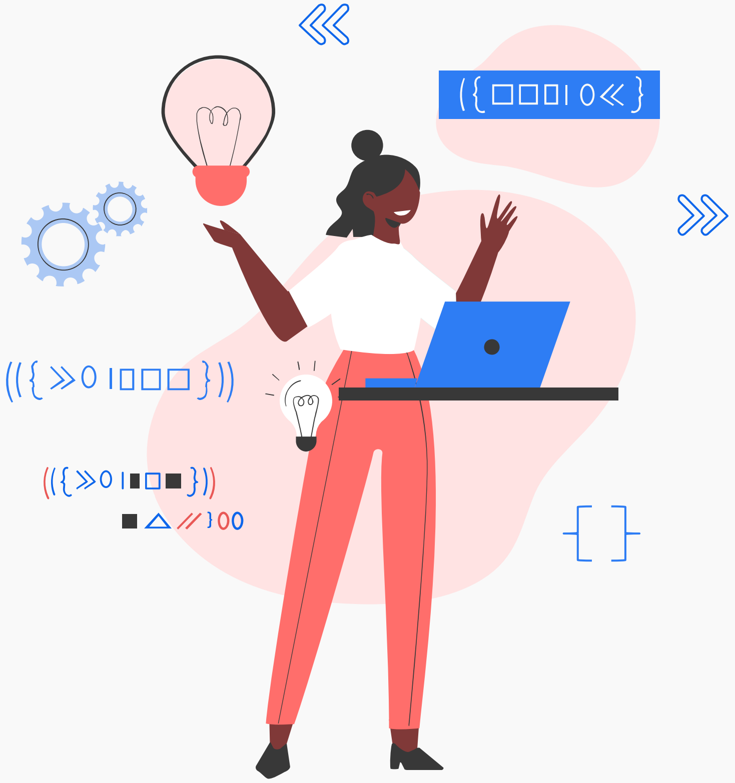


01

Introduction

AWS Environemnts





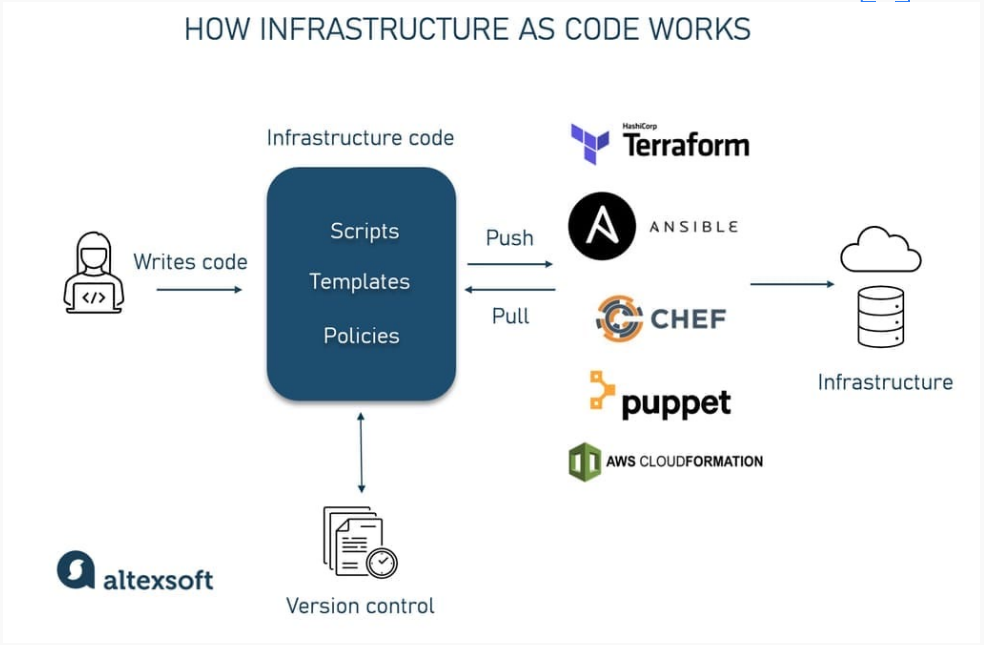
02

Theory

AWS IaC Recap

Infrastructure as Code

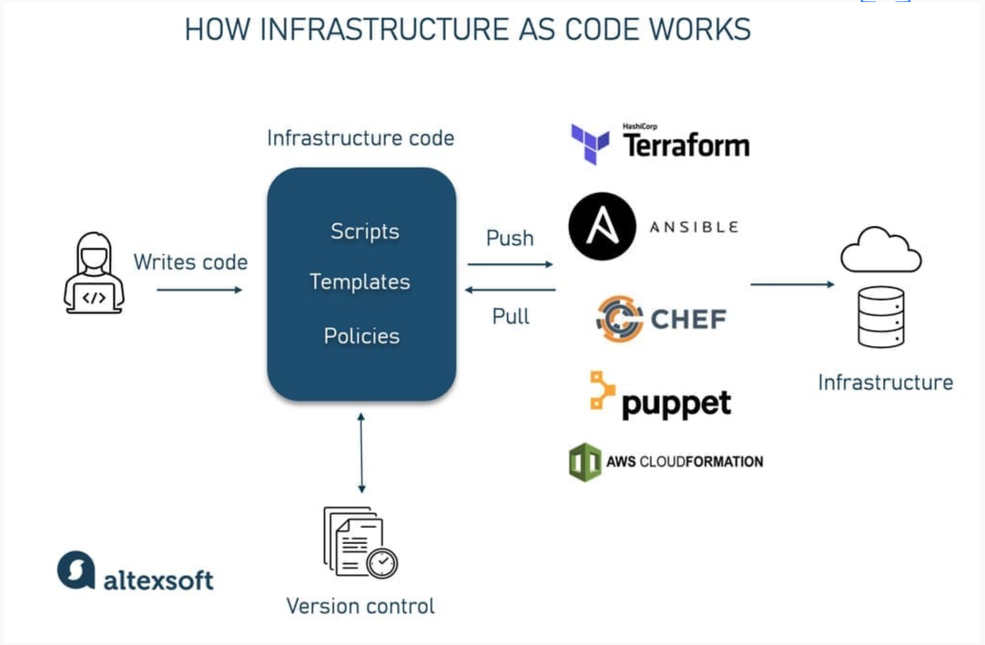
is the ability to provision and support your computing infrastructure using code instead of manual processes and settings.



source: https://www.altexsoft.com/blog/infrastructure-as-code/

Declarative Resource Management

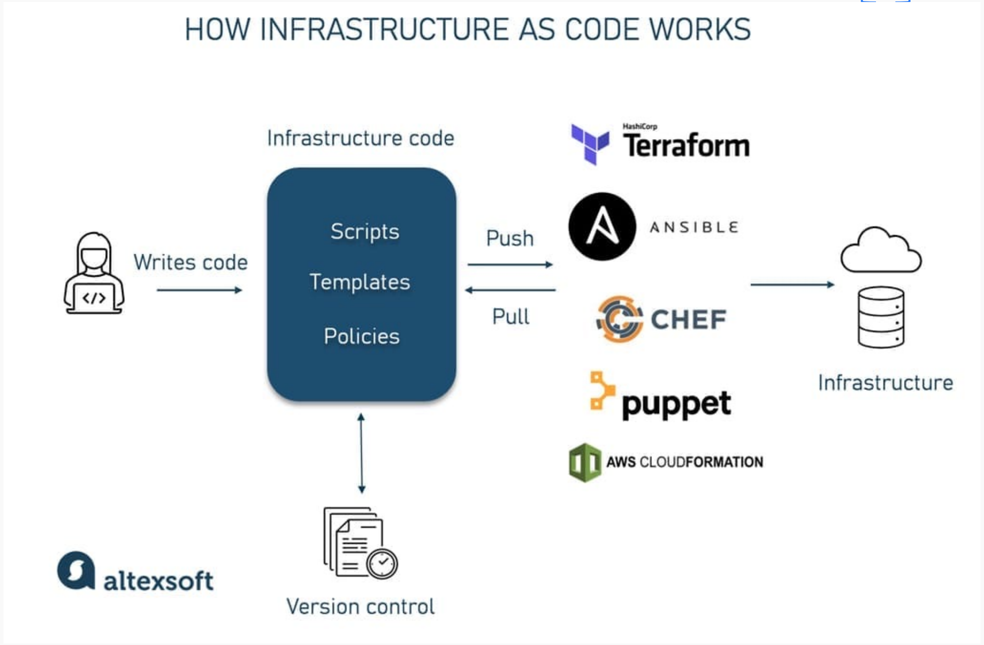
Declarative templates enable consistent resource configuration and simplify replication across environments.



source: https://www.altexsoft.com/blog/infrastructure-as-code/

Idempotent Deployments

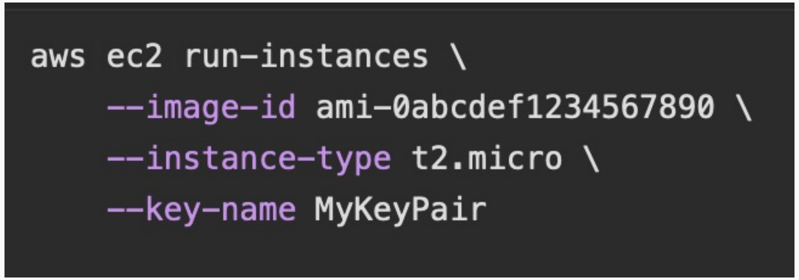
No matter how many times you run your IaC and, what your starting state is, you will end up with the same end state



source: https://www.altexsoft.com/blog/infrastructure-as-code/

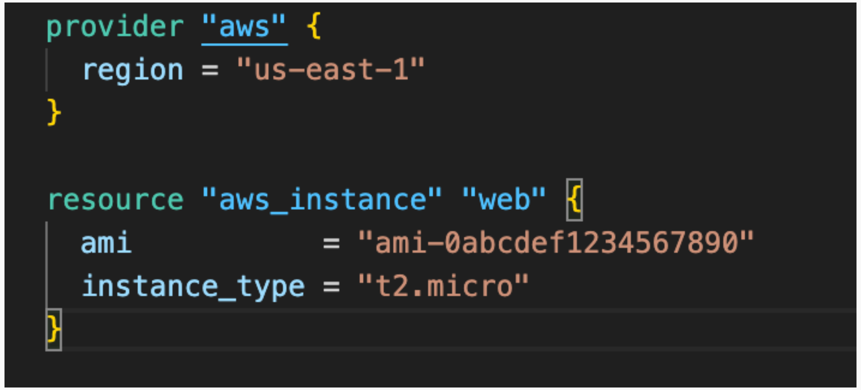
Scripts vs IaC

AWS CLI



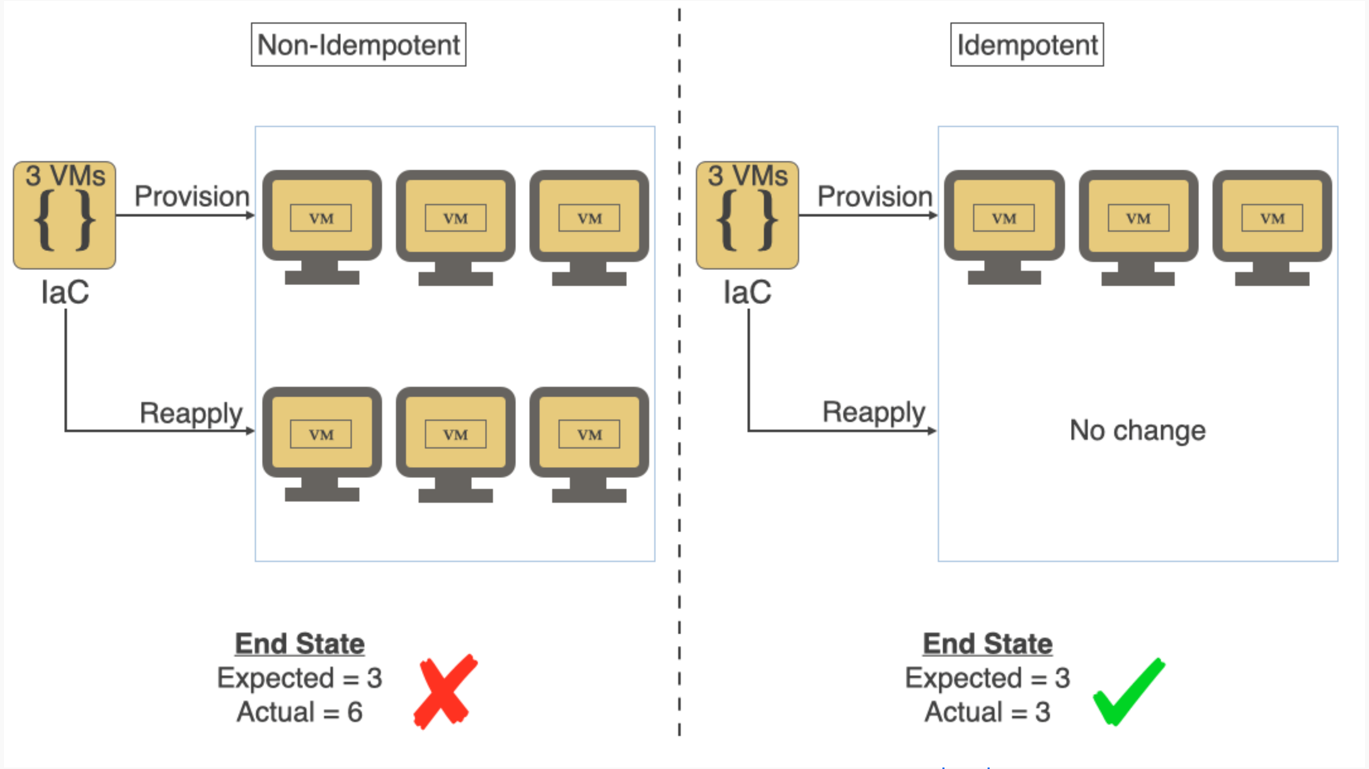
```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --key-name MyKeyPair
```

Terraform



```
provider "aws" {
  region = "us-east-1"
}

resource "aws_instance" "web" {
  ami           = "ami-0abcdef1234567890"
  instance_type = "t2.micro"
}
```

Idempotency

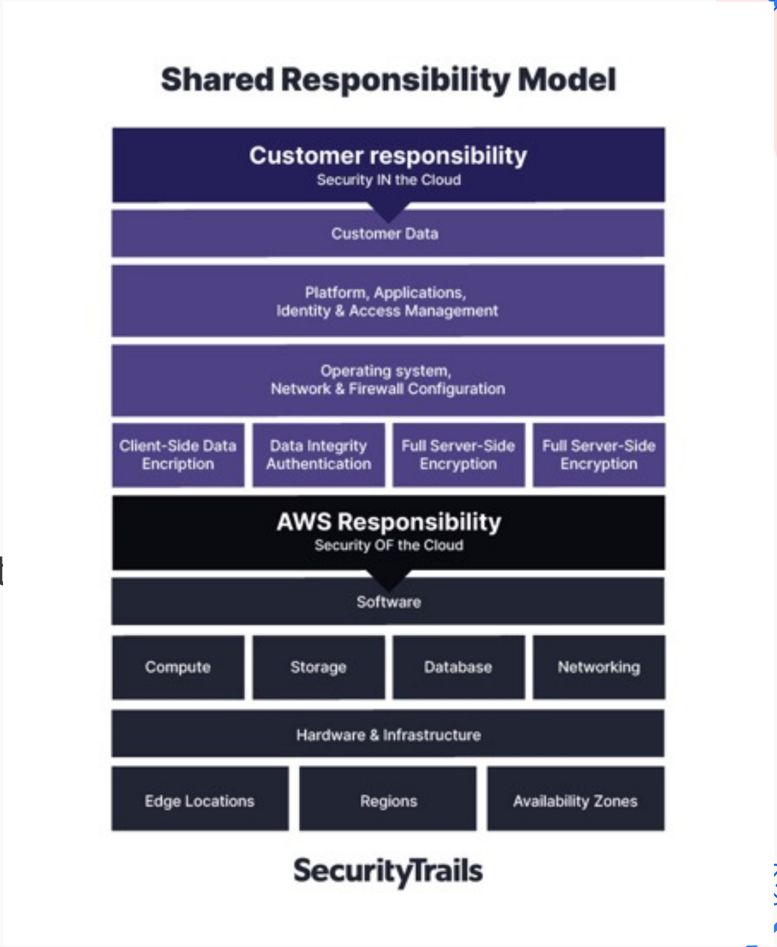


source: https://www.cloudknit.io/blog/principles-patterns-and-practices-for-effective-infrastructure-as-code

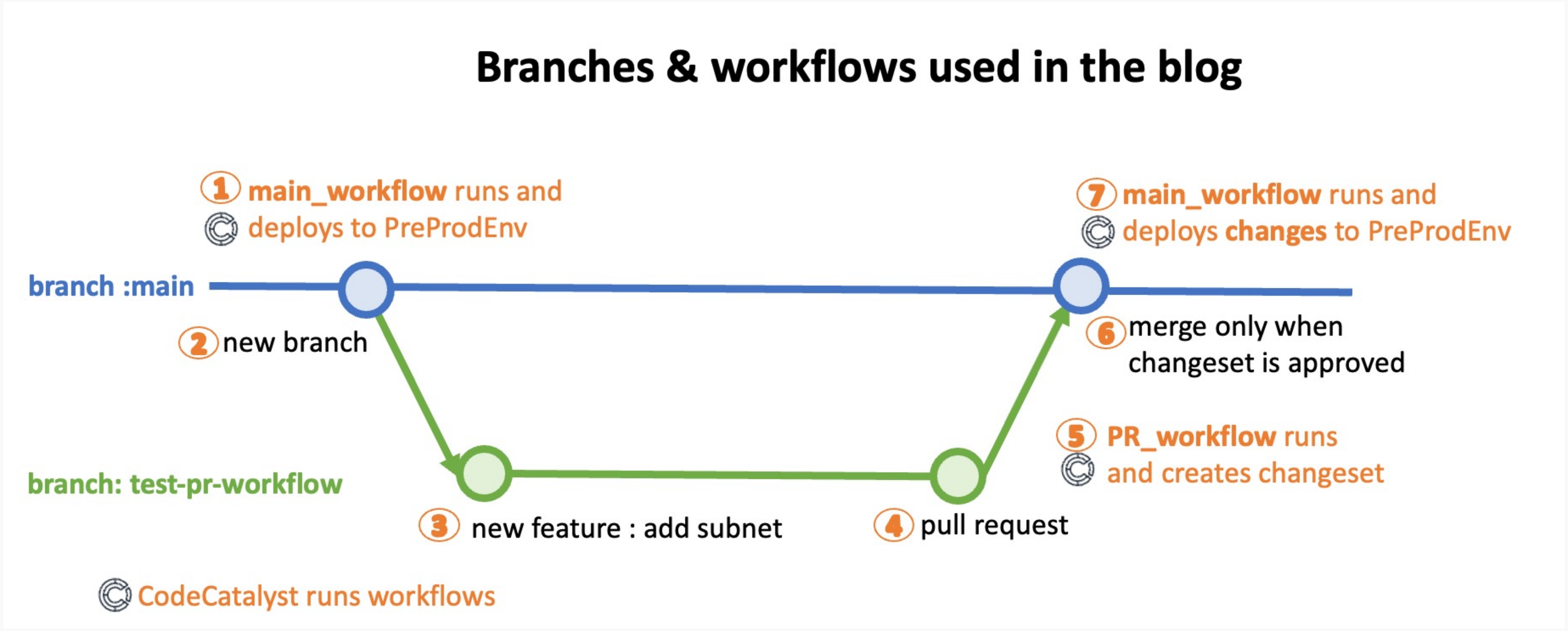
Good & Bad Practices

Why We Need Best Practices

- Misconfigured IAM Roles
- S3 Buckets with Public Access
- Unplanned Costs due to EC2 and S3 Misconfigurations
- Downtime Due to Misconfigured Autoscaling
- Resource Deletion from Incorrect Version Cont
- Network Exposure through Incorrect Security Group Rules
- Downtime Due to Misconfigured Networking Configuration


Shared Responsibility Model
Customer responsibility
Security IN the Cloud
Customer Data
Platform, Applications, Identity & Access Management
Operating system, Network & Firewall Configuration
Client-Side Data Encription
Data Integrity Authentication
Full Server-Side Encryption
Full Server-Side Encryption
AWS Responsibility
Security OF the Cloud
Software
Compute
Storage
Database
Networking
Hardware & Infrastructure
Edge Locations
Regions
Availability Zones
SecurityTrails

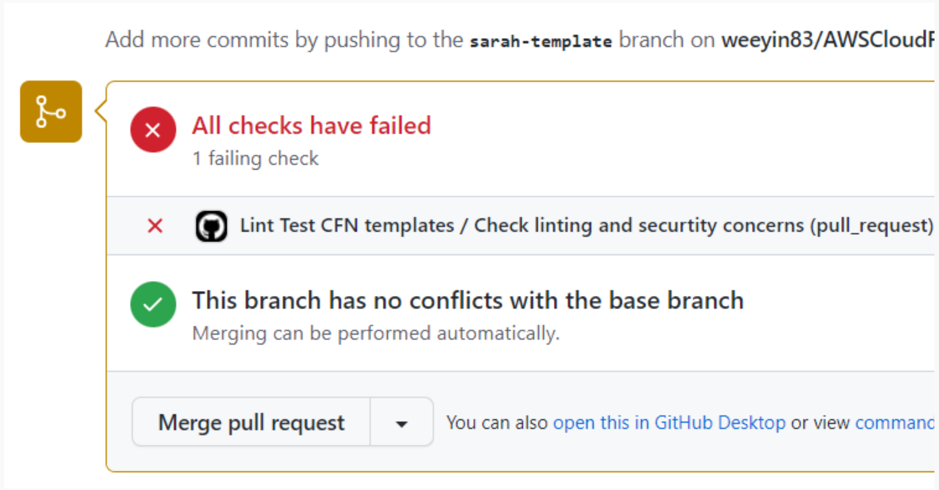
Controlled Deployment to Environments



https://community.aws/content/2dukZbCqB0wWgvV6VIioWVNJ959/build-ci-cd-pipeline-iac-cloudformation

Controlled Deployment to Environments

Do:
- Use Pull Requests
- Use PR checks
- Use Manual Triggers


Add more commits by pushing to the sarah-template branch on weeyin83/AWSCloudF

All checks have failed
1 failing check

Lint Test CFN templates / Check linting and securtity concerns (pull_request)

This branch has no conflicts with the base branch
Merging can be performed automatically.

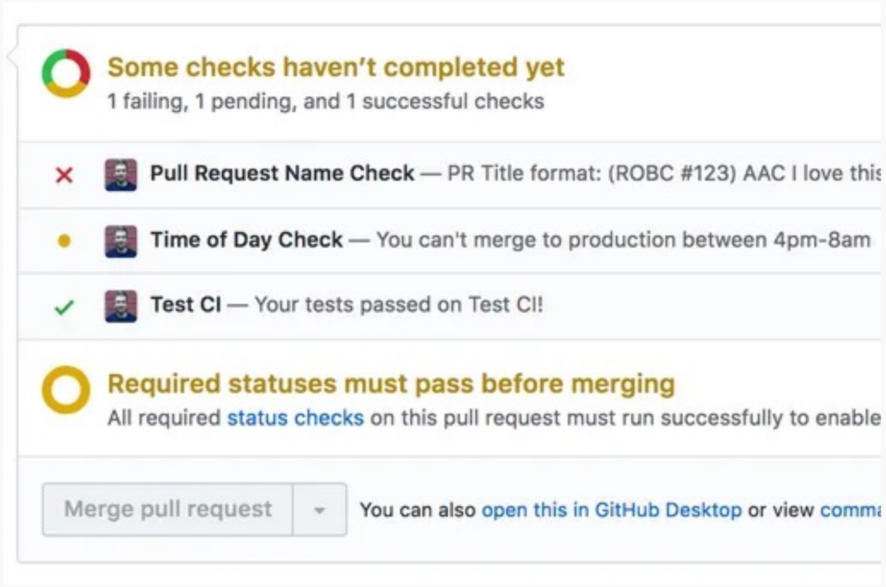
You can also open this in GitHub Desktop or view commanc

https://www.techielass.com/source-control-your-aws-cloudformation-templates-with-github/

Controlled Deployment to Environments

Do:
- Use Pull Requests
- Use PR checks
- Use Manual Triggers

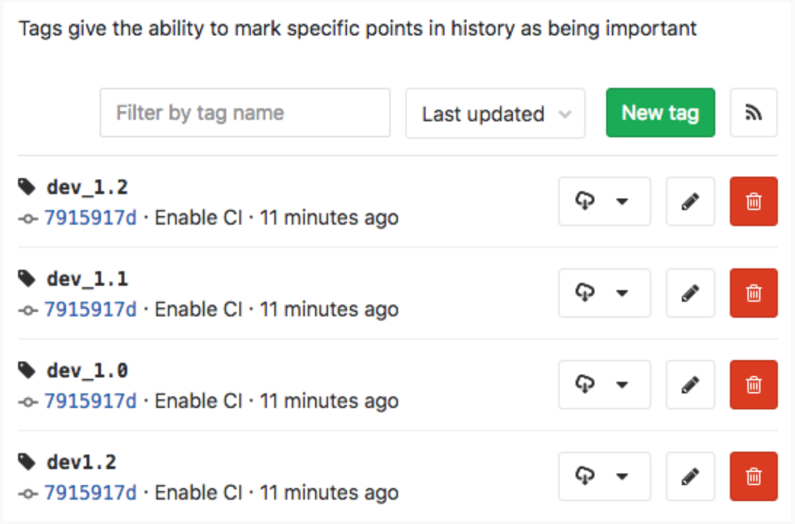
Don't:
- Deploy to DEV on every push to a feature branch
- Bypass existing rules


Some checks haven't completed yet
1 failing, 1 pending, and 1 successful checks
✗ Pull Request Name Check — PR Title format: (ROBC #123) AAC I love this
● Time of Day Check — You can't merge to production between 4pm-8am
✓ Test CI — Your tests passed on Test CI!
Required statuses must pass before merging
All required status checks on this pull request must run successfully to enable

You can also open this in GitHub Desktop or view comma

https://blog.theodo.com/2018/04/prevent-lazy-github-practices-in-10-minutes-using-aws-lambdas/

Tag based deployments


Tags give the ability to mark specific points in history as being important

Filter by tag name

Last updated

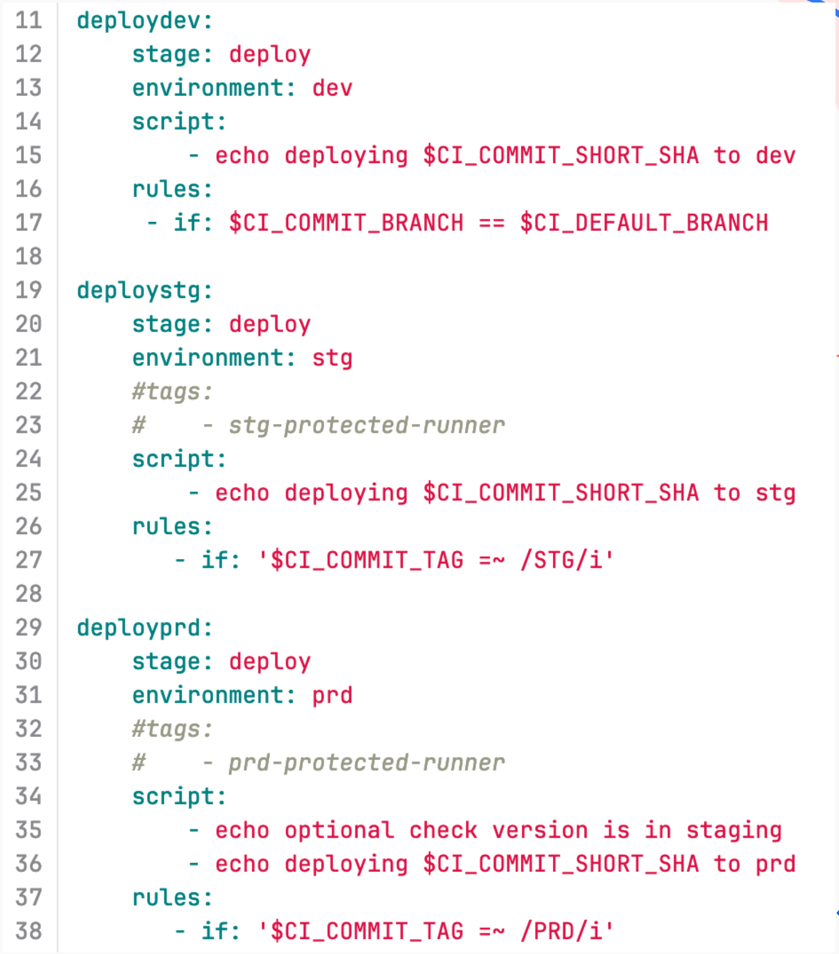
New tag

dev_1.2
7915917d · Enable CI · 11 minutes ago

dev_1.1
7915917d · Enable CI · 11 minutes ago

dev_1.0
7915917d · Enable CI · 11 minutes ago

dev1.2
7915917d · Enable CI · 11 minutes ago

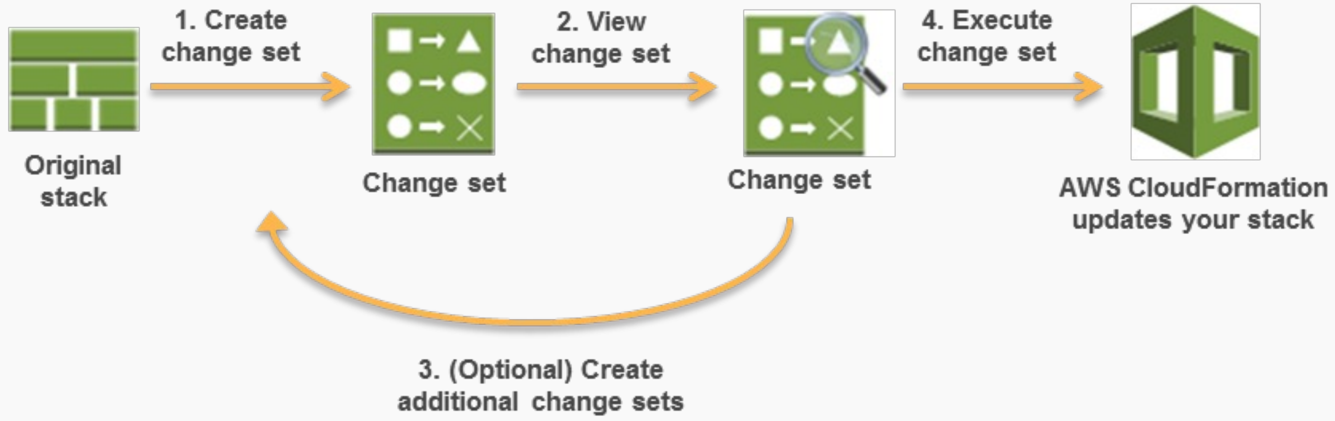

```
deploydev:
    stage: deploy
    environment: dev
    script:
        - echo deploying $CI_COMMIT_SHORT_SHA to dev
    rules:
     - if: $CI_COMMIT_BRANCH == $CI_DEFAULT_BRANCH

deploystg:
    stage: deploy
    environment: stg
    #tags:
    #     - stg-protected-runner
    script:
        - echo deploying $CI_COMMIT_SHORT_SHA to stg
    rules:
       - if: '$CI_COMMIT_TAG =~ /STG/i'

deployprd:
    stage: deploy
    environment: prd
    #tags:
    #     - prd-protected-runner
    script:
        - echo optional check version is in staging
        - echo deploying $CI_COMMIT_SHORT_SHA to prd
    rules:
       - if: '$CI_COMMIT_TAG =~ /PRD/i'
```

https://gitlab.com/kaihendry/p2p-demo/-/blob/main/.gitlab-ci.yml

Pre-Deployment Change Analysis

Do:

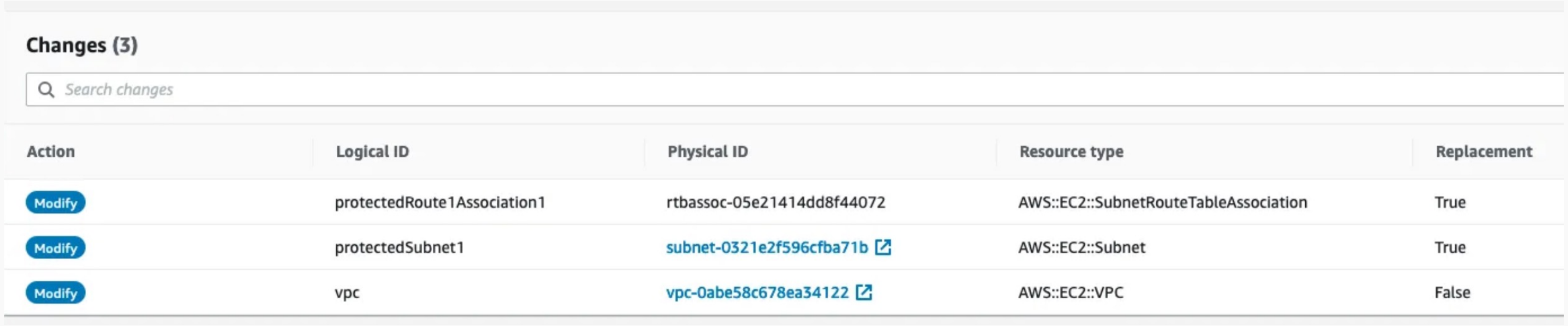
- Review CloudFormation ChangeSets & Terraform Plan output



Pre-Deployment Change Analysis

Do:

- Review CloudFormation ChangeSets & Terraform Plan output



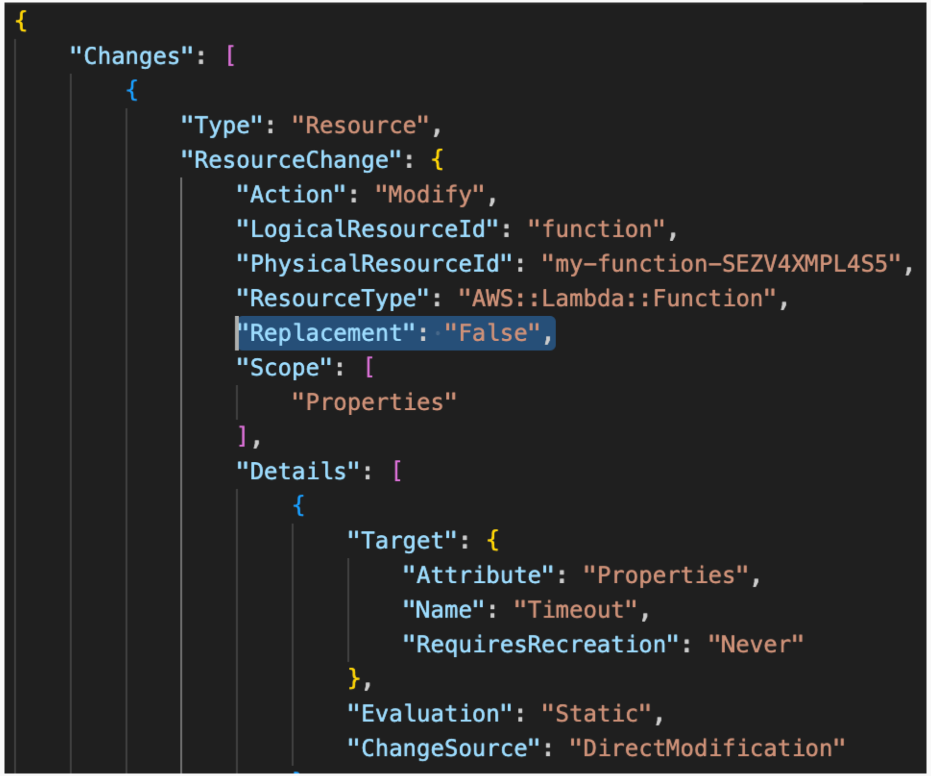
Changes (3)

Search changes

Action	Logical ID	Physical ID	Resource type	Replacement
Modify	protectedRoute1Association1	rtbassoc-05e21414dd8f44072	AWS::EC2::SubnetRouteTableAssociation	True
Modify	protectedSubnet1	subnet-0321e2f596cfba71b	AWS::EC2::Subnet	True
Modify	vpc	vpc-0abe58c678ea34122	AWS::EC2::VPC	False

Pre-Deployment Change Analysis

Do:
- Review CloudFormation ChangeSets



```
{
    "Changes": [
        {
            "Type": "Resource",
            "ResourceChange": {
                "Action": "Modify",
                "LogicalResourceId": "function",
                "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
                "ResourceType": "AWS::Lambda::Function",
                "Replacement": "False",
                "Scope": [
                    "Properties"
                ],
                "Details": [
                    {
                        "Target": {
                            "Attribute": "Properties",
                            "Name": "Timeout",
                            "RequiresRecreation": "Never"
                        },
                        "Evaluation": "Static",
                        "ChangeSource": "DirectModification"
```

Pre-Deployment Change Analysis

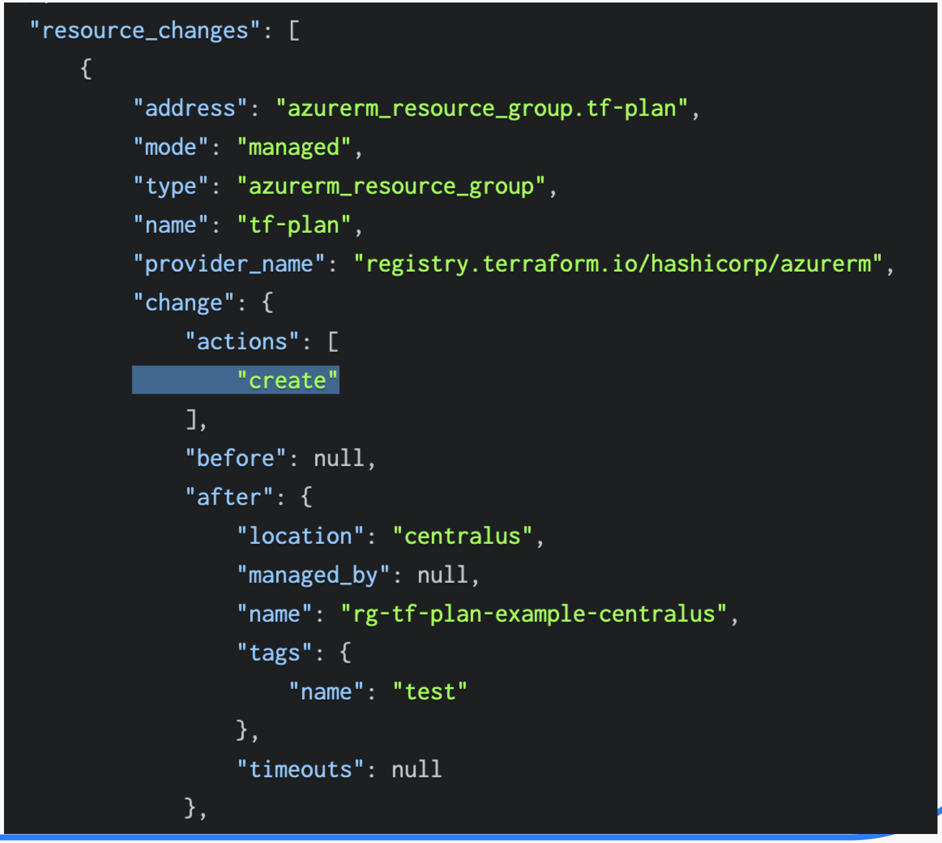
Do:

- Review CloudFormation ChangeSets & Terraform Plan output

Terraform plan example

terraform plan -out=myplan.tfplan

terraform show -json myplan.tfplan > myplan.json



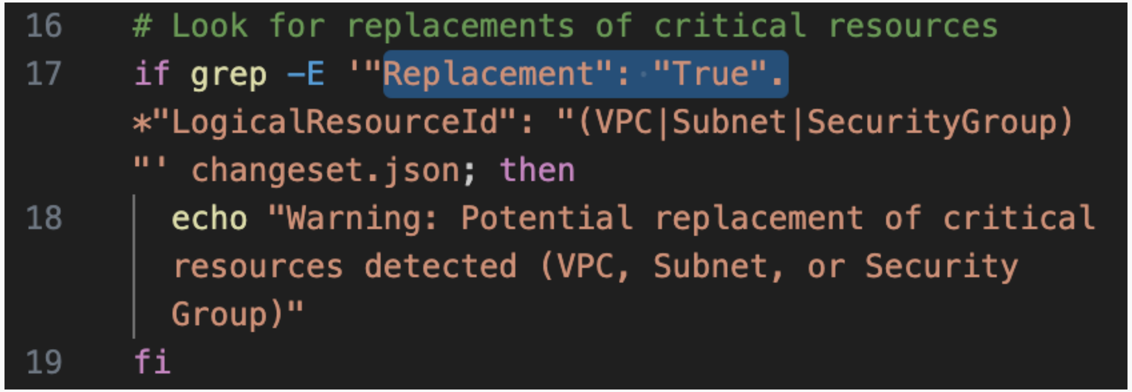
```
"resource_changes": [
    {
        "address": "azurerm_resource_group.tf-plan",
        "mode": "managed",
        "type": "azurerm_resource_group",
        "name": "tf-plan",
        "provider_name": "registry.terraform.io/hashicorp/azurerm",
        "change": {
            "actions": [
                "create"
            ],
            "before": null,
            "after": {
                "location": "centralus",
                "managed_by": null,
                "name": "rg-tf-plan-example-centralus",
                "tags": {
                    "name": "test"
                },
                "timeouts": null
            },
```

source: https://spacelift.io/blog/terraform-plan

Pre-Deployment Change Analysis

Do:

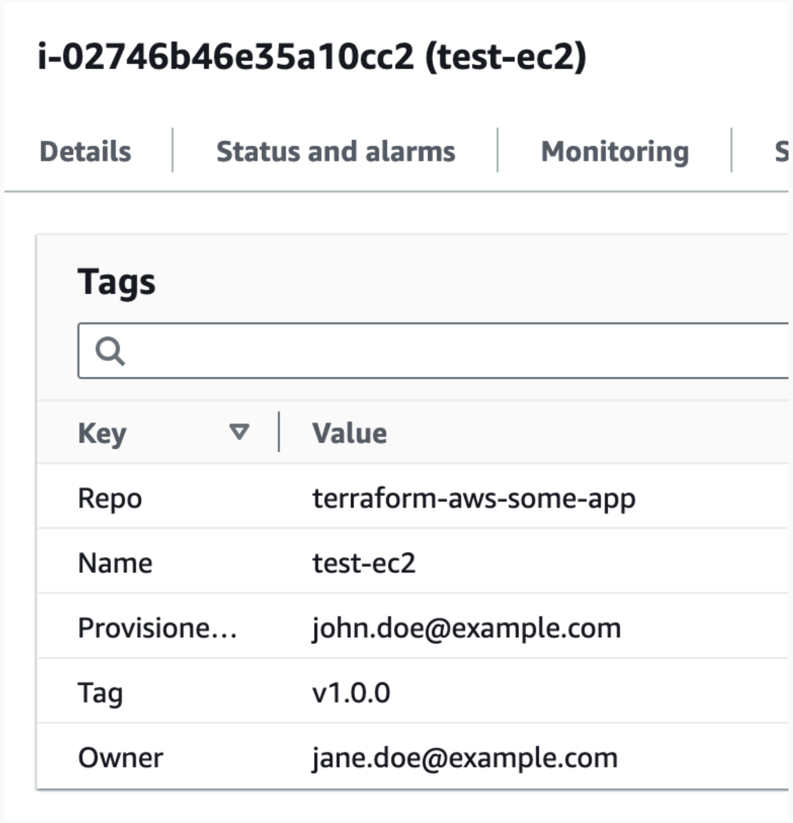
- Review CloudFormation ChangeSets & Terraform Plan output
- Automate the process by using scripts



```
# Look for replacements of critical resources
if grep -E '"Replacement": "True".*"LogicalResourceId": "(VPC|Subnet|SecurityGroup)"' changeset.json; then
  echo "Warning: Potential replacement of critical resources detected (VPC, Subnet, or Security Group)"
fi
```

Detailed Tagging

- **GITHUB_REPOSITORY**: The owner and repository name. For example, octocat/Hello-World.
- **GITHUB_SHA**: The commit SHA that triggered the workflow.
- **GITHUB_REF**: The branch or tag ref that triggered the workflow.
- **GITHUB_WORKFLOW**: The name of the workflow.
- **GITHUB_ACTOR**: The name of the person or app that initiated the workflow.



Tags

Key	Value
Repo	terraform-aws-some-app
Name	test-ec2
Provisione…	john.doe@example.com
Tag	v1.0.0
Owner	jane.doe@example.com

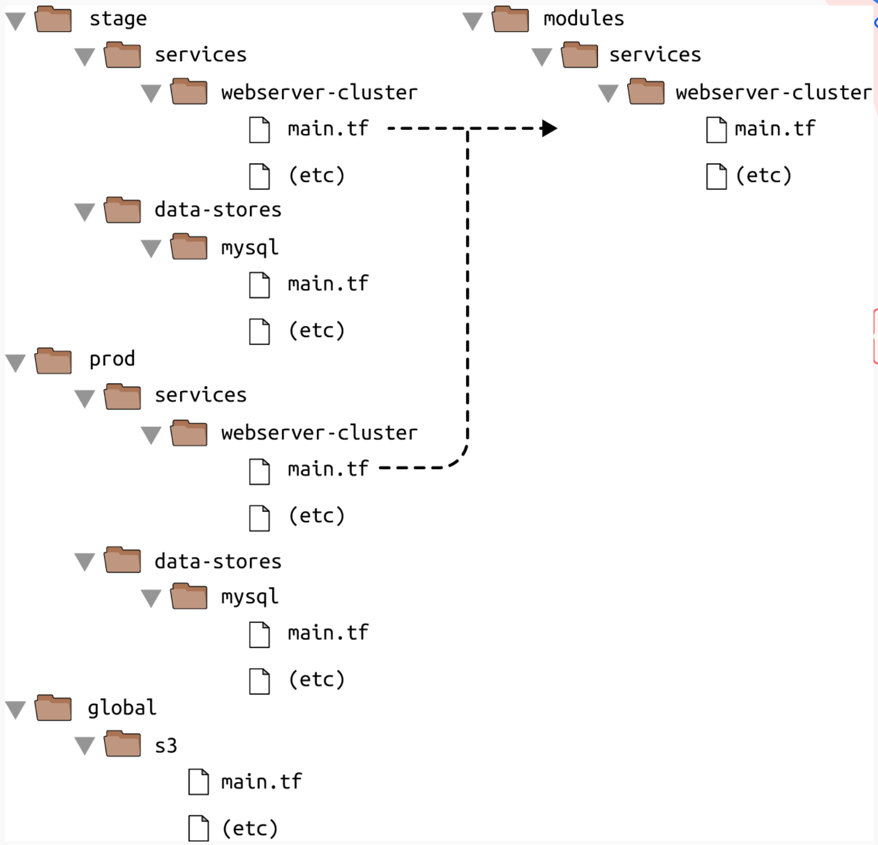
Extract reusable code (Modules)

Do:

- Extract Code Internally
- Extract Code to outside Repo
- Use Versioning

Don't:

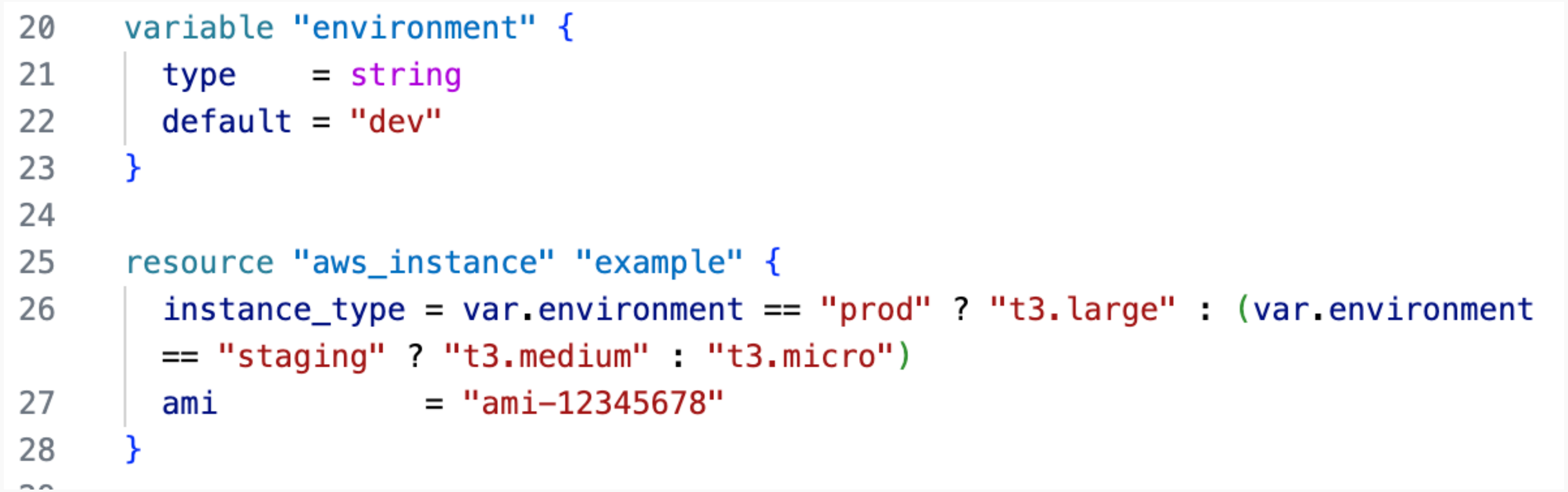
- Hardcode values inside modules
- Over-Engineer modules
- Over-Nest modules



```
▼ stage                                      ▼ modules
    ▼ services                                   ▼ services
        ▼ webserver-cluster                          ▼ webserver-cluster
              main.tf  ----------------------->           main.tf
              (etc)                                       (etc)
    ▼ data-stores
        ▼ mysql
              main.tf
              (etc)
▼ prod
    ▼ services
        ▼ webserver-cluster
              main.tf
              (etc)
    ▼ data-stores
        ▼ mysql
              main.tf
              (etc)
▼ global
    ▼ s3
          main.tf
          (etc)
```

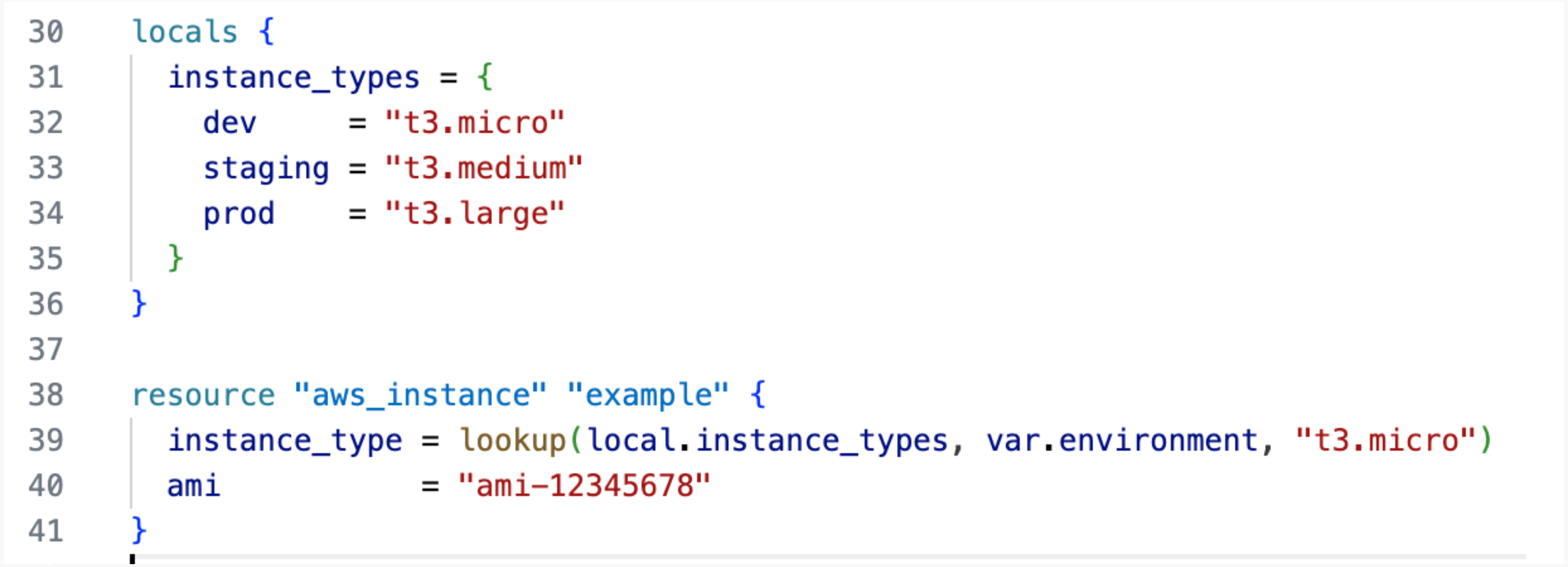
https://blog.gruntwork.io/how-to-create-reusable-infrastructure-with-terraform-modules-25526d65f73d

Write easy to read code (Terraform)



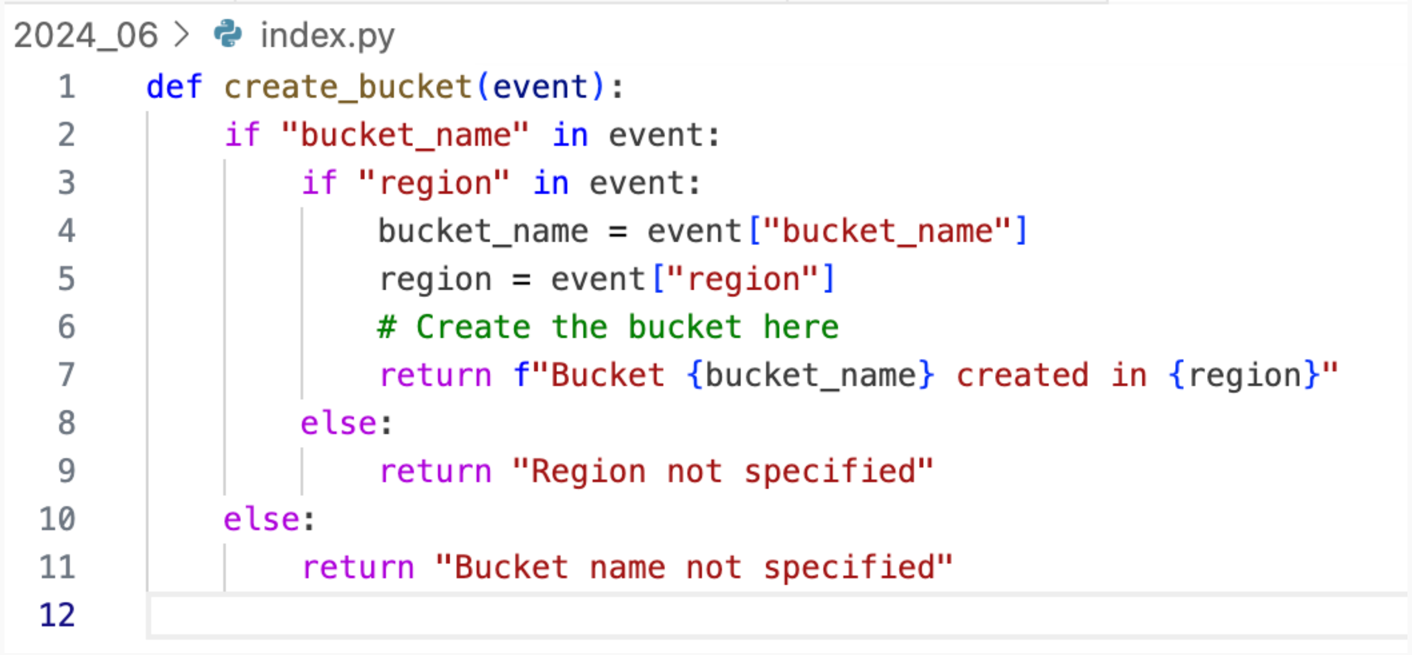
```
variable "environment" {
  type    = string
  default = "dev"
}

resource "aws_instance" "example" {
  instance_type = var.environment == "prod" ? "t3.large" : (var.environment
  == "staging" ? "t3.medium" : "t3.micro")
  ami           = "ami-12345678"
}
```

Write easy to read code (Terraform)



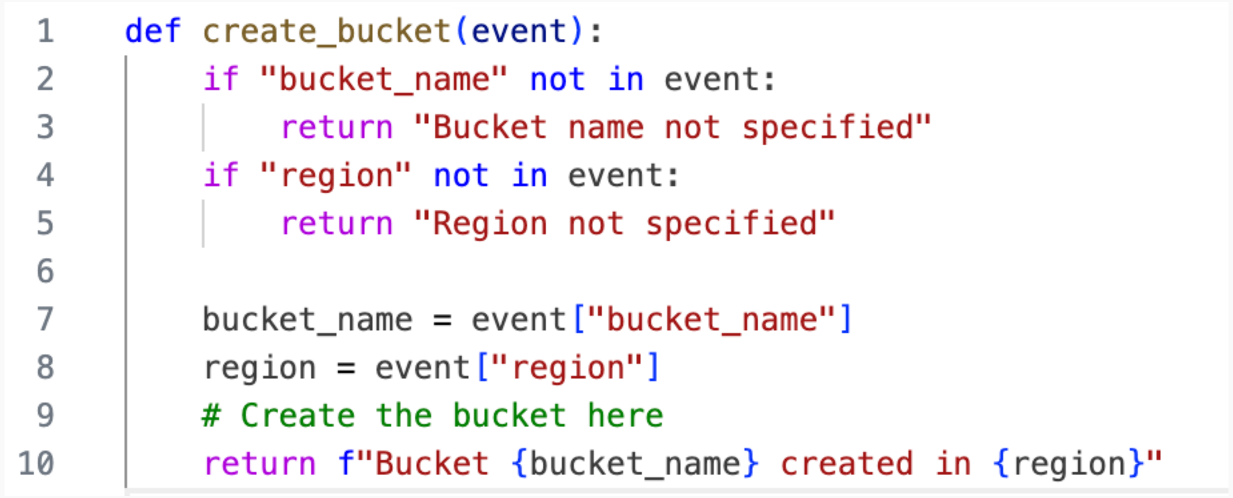
```
locals {
  instance_types = {
    dev     = "t3.micro"
    staging = "t3.medium"
    prod    = "t3.large"
  }
}

resource "aws_instance" "example" {
  instance_type = lookup(local.instance_types, var.environment, "t3.micro")
  ami           = "ami-12345678"
}
```

Write easy to read code (Python)


2024_06 > index.py

```python
def create_bucket(event):
    if "bucket_name" in event:
        if "region" in event:
            bucket_name = event["bucket_name"]
            region = event["region"]
            # Create the bucket here
            return f"Bucket {bucket_name} created in {region}"
        else:
            return "Region not specified"
    else:
        return "Bucket name not specified"

```

Write easy to read code (Python)



```python
def create_bucket(event):
    if "bucket_name" not in event:
        return "Bucket name not specified"
    if "region" not in event:
        return "Region not specified"

    bucket_name = event["bucket_name"]
    region = event["region"]
    # Create the bucket here
    return f"Bucket {bucket_name} created in {region}"
```

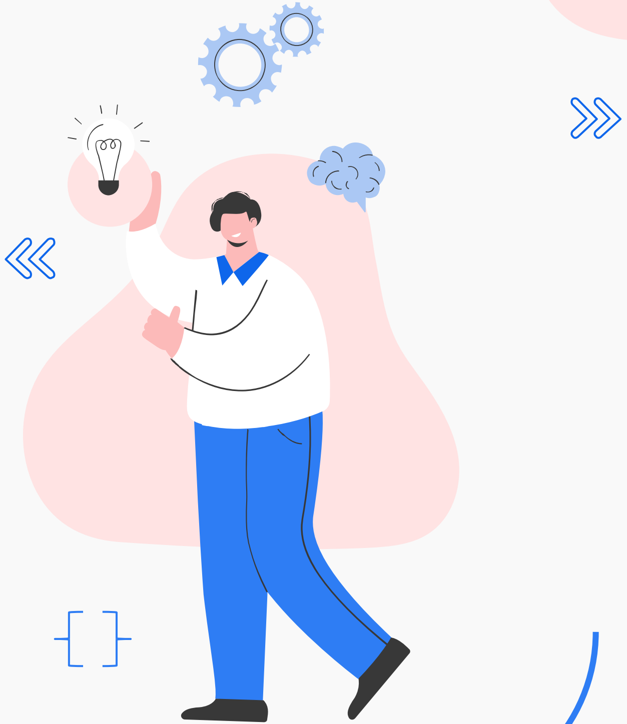
Security Considerations for IaC

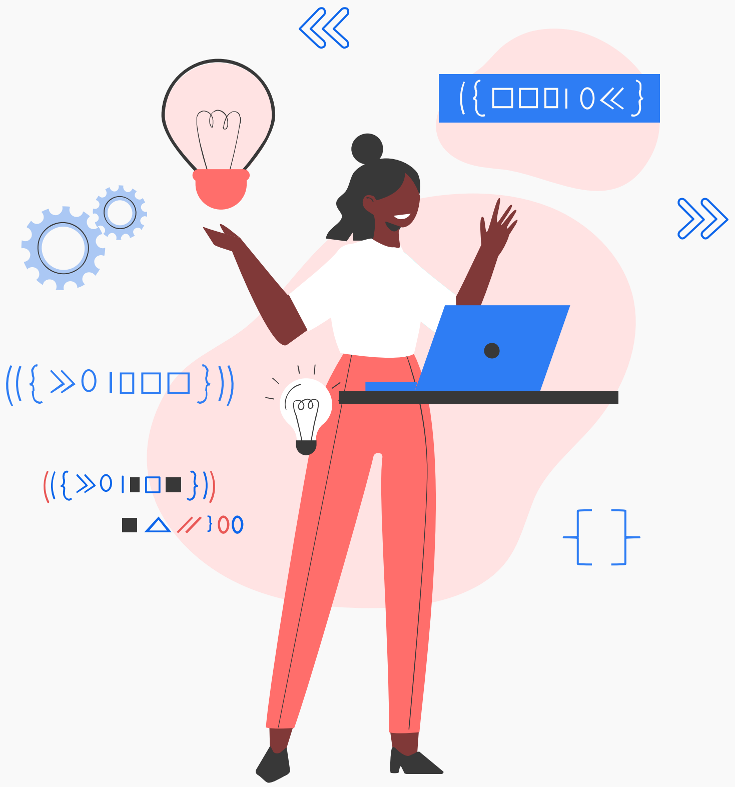
- Use Version Control with Access Restriction
- Avoid Hardcoding Sensitive Data- Use AWS Secrets Manager or AWS SSM Parameter Store
- Apply IAM Permission Boundaries and least privilege for deployment roles.
- Enable Automated Security Scanning
- Do not log sensitive data in pipelines
- Review and Use IaC Change Previews

Helpful Tools

AWS IaC helpers

- Checkov, Tfsec
- Snyk
- AWS CloudFormation Guard
- Trust, but Verify

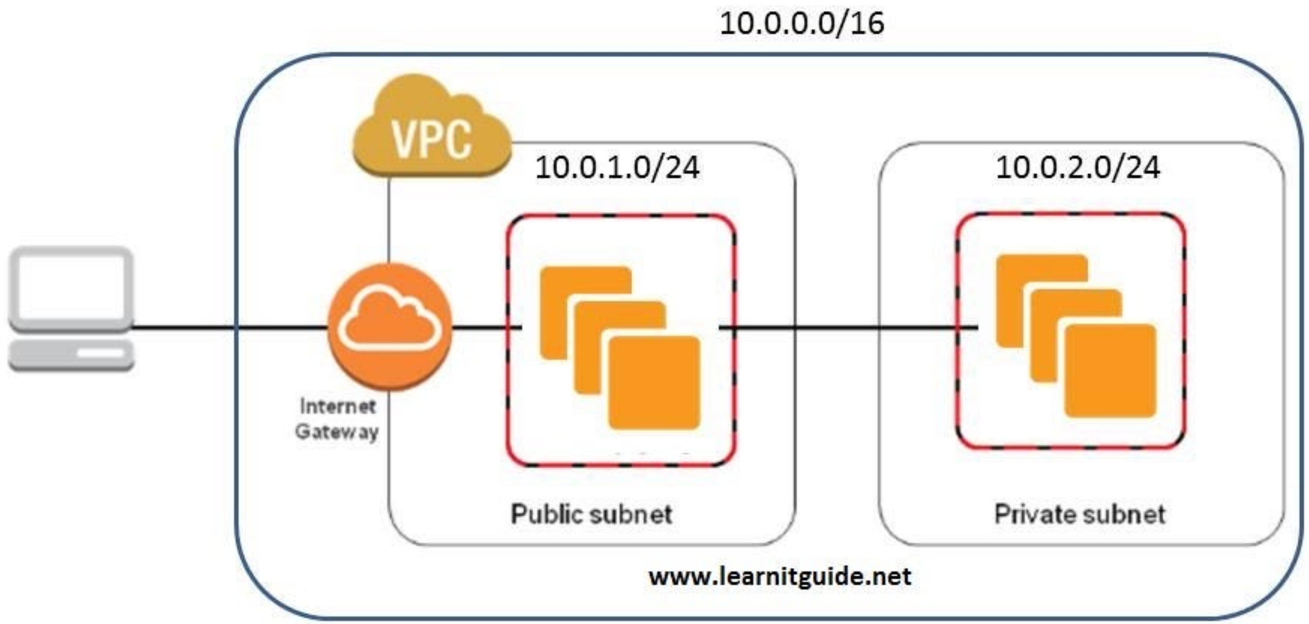


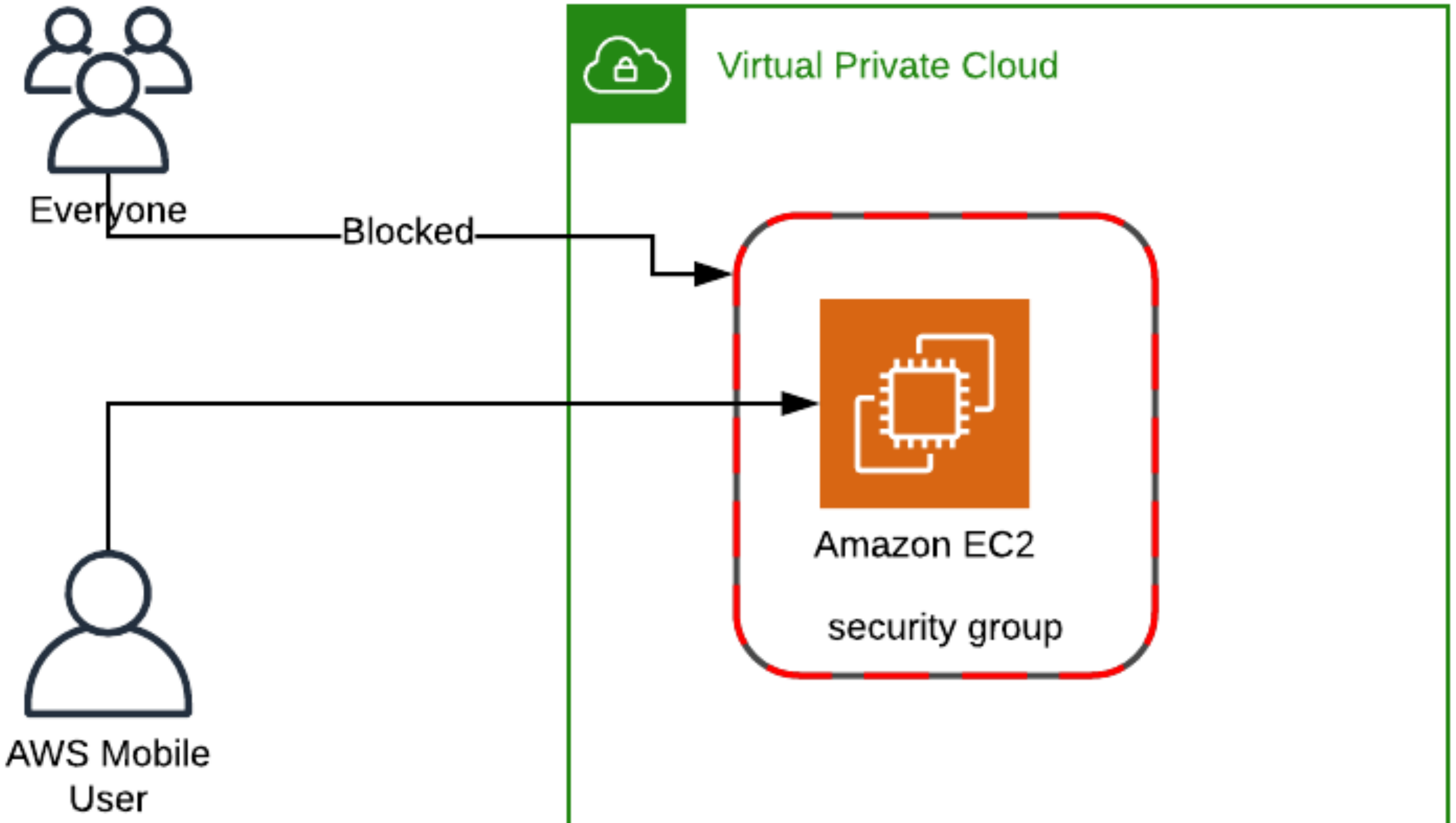


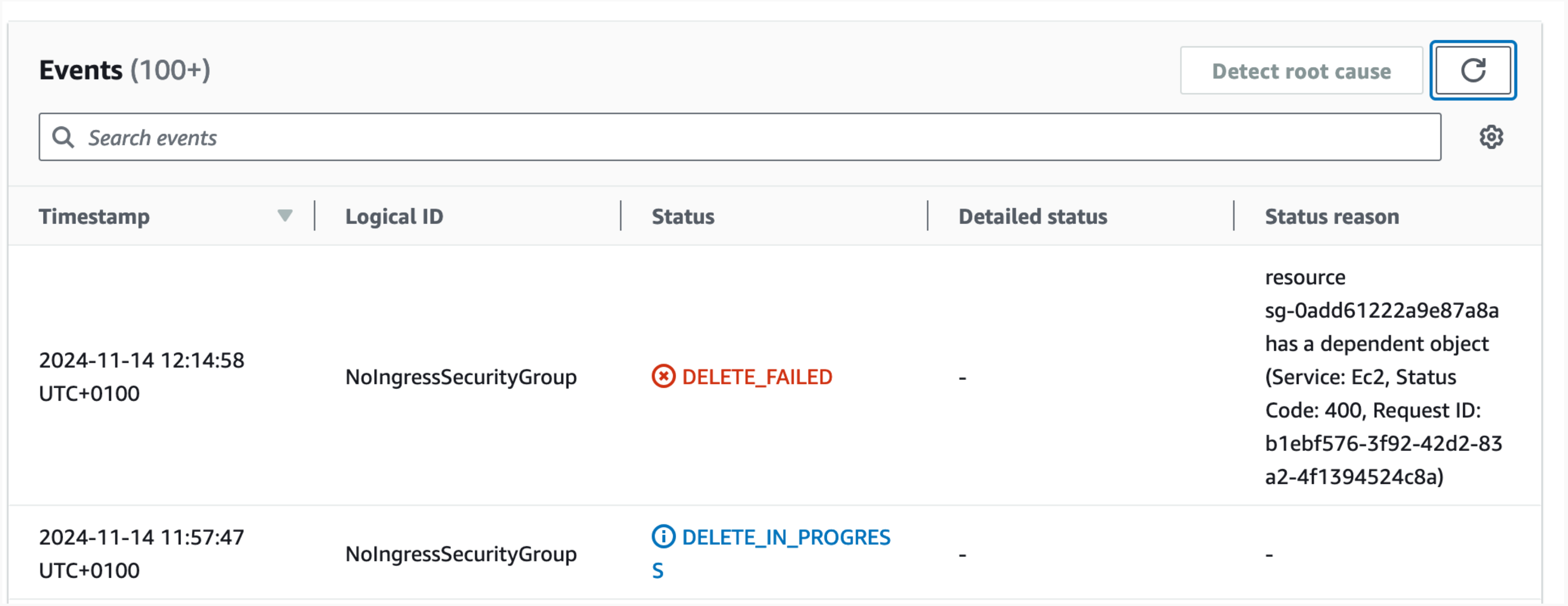
03

Demo Time

Let the Demo Gods be with Us


10.0.0.0/16
VPC
10.0.1.0/24
10.0.2.0/24
Internet Gateway
Public subnet
Private subnet
www.learnitguide.net


Everyone
Blocked
Virtual Private Cloud
Amazon EC2
security group
AWS Mobile User

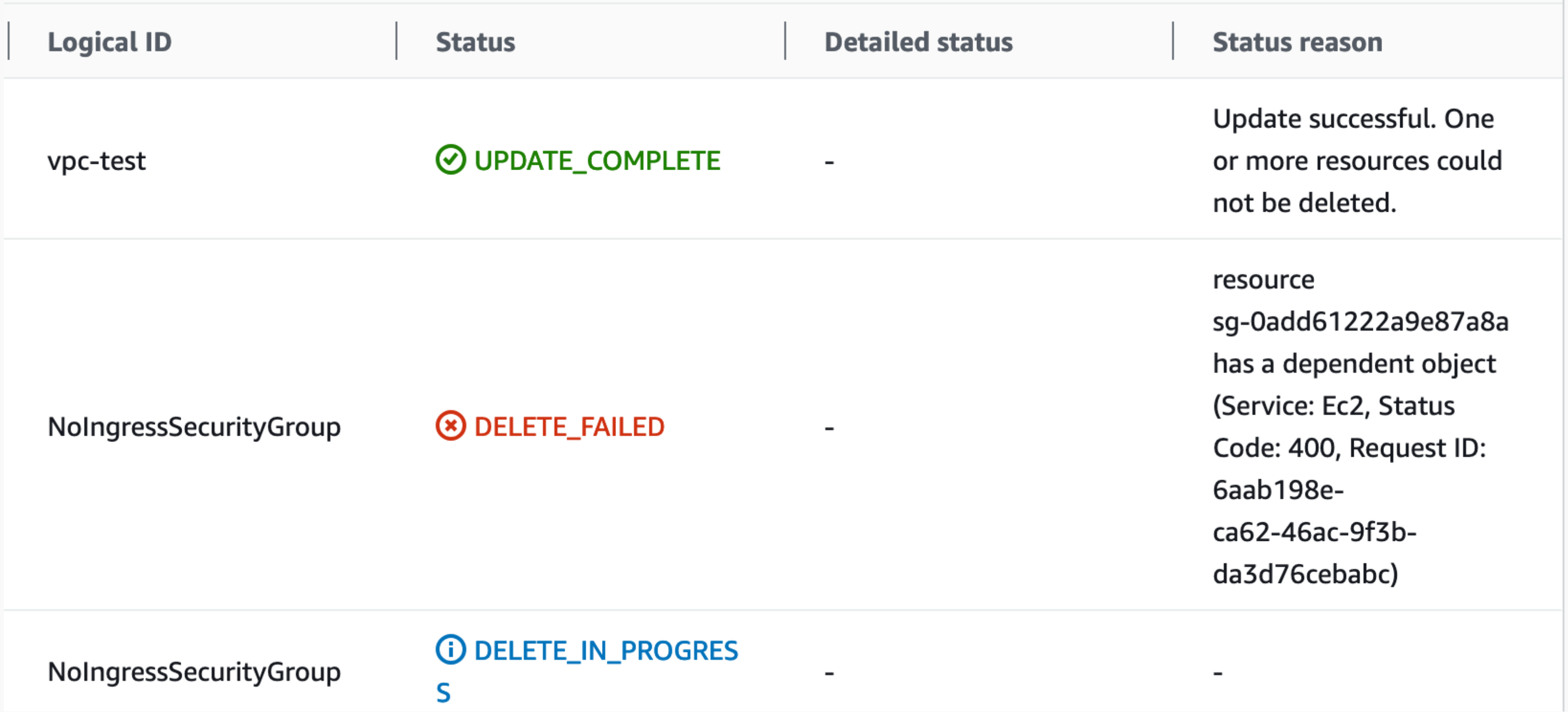


Events (100+)

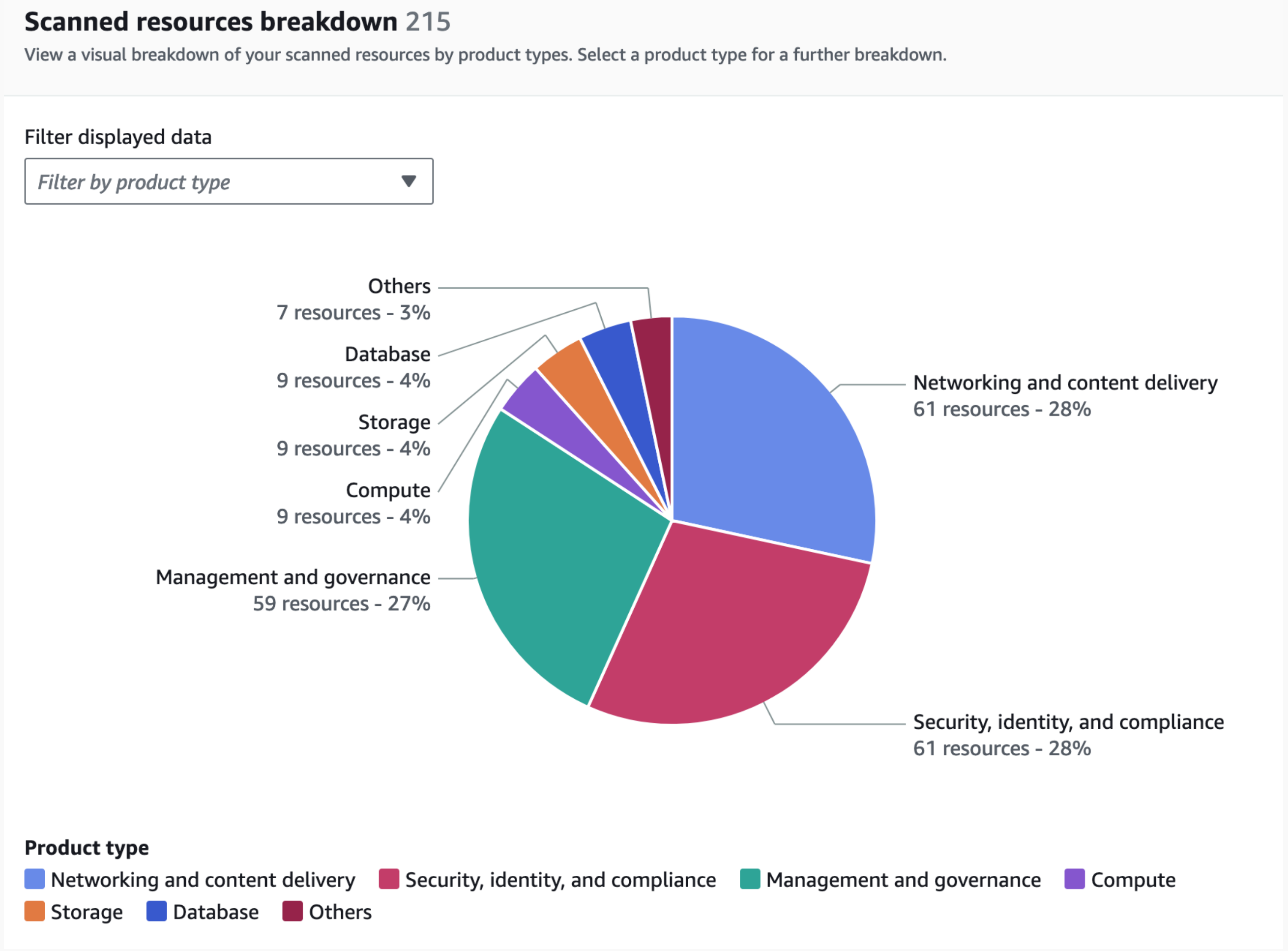
Detect root cause

Search events

Timestamp	Logical ID	Status	Detailed status	Status reason
2024-11-14 12:14:58 UTC+0100	NoIngressSecurityGroup	DELETE_FAILED	-	resource sg-0add61222a9e87a8a has a dependent object (Service: Ec2, Status Code: 400, Request ID: b1ebf576-3f92-42d2-83a2-4f1394524c8a)
2024-11-14 11:57:47 UTC+0100	NoIngressSecurityGroup	DELETE_IN_PROGRESS	-	-



Logical ID	Status	Detailed status	Status reason
vpc-test	✓ UPDATE_COMPLETE	-	Update successful. One or more resources could not be deleted.
NoIngressSecurityGroup	⊗ DELETE_FAILED	-	resource sg-0add61222a9e87a8a has a dependent object (Service: Ec2, Status Code: 400, Request ID: 6aab198e-ca62-46ac-9f3b-da3d76cebabc)
NoIngressSecurityGroup	ⓘ DELETE_IN_PROGRESS	-	-



Scanned resources breakdown 215

View a visual breakdown of your scanned resources by product types. Select a product type for a further breakdown.

Filter displayed data

Filter by product type

Others
7 resources - 3%

Database
9 resources - 4%

Storage
9 resources - 4%

Compute
9 resources - 4%

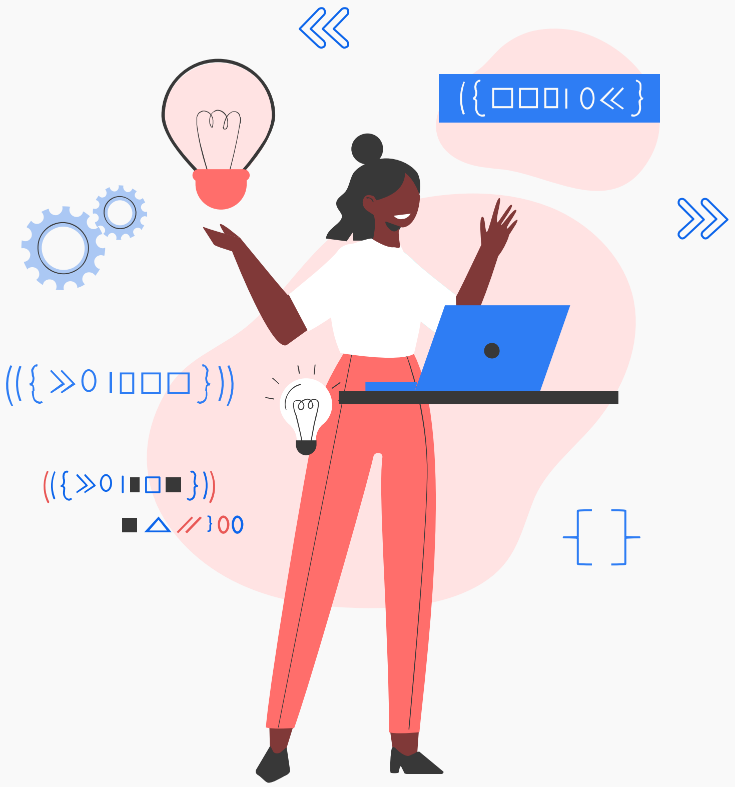
Management and governance
59 resources - 27%

Networking and content delivery
61 resources - 28%

Security, identity, and compliance
61 resources - 28%

Product type

🟦 Networking and content delivery 🟥 Security, identity, and compliance 🟩 Management and governance 🟪 Compute
🟧 Storage 🟦 Database 🟥 Others



04

Summary

AWS Best & Worst Practices

“Your friends and family understand what you do”

—Positive Affirmations for SREs

https://www.youtube.com/watch?v=ia8Q51ouA_s

Thanks!

Reach out:

MichalMikolajczyk.pl@gmail.com

Resources:

CloudNinja.pl

github.com/MichalMiko/aws_meetup

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**