Securing IaC

The Attack and Defense Way

Nitin Sharma, Security Engineer

Salesforce, India

Synopsis

$ whoami

$ Introduction to IaC

$ Cloud Security Scenario: 2020-2021

$ IaC and DevSecOps

$ Different IaC Methods (Orchestration vs Configuration Management)

$ IaC with Terraform and Security Best Practices

$ Terraform Security - The Attack & Defense (checkov & tfsec with TerraGoat)

$ whoami

- Security Engineer since 2016
- Started career doing security automation for cloud and on-prem
- Inevitable love for Linux and Python
- AWS Community Builder (cohort Nov. 2021)
- Published Author - “Securing Docker: The Attack and Defense Way”
- GIAC GCSA, CCSKv4, Terraform Associate, AWS 4x-Certified, CySA+, Pentest+, etc.
- Interested in Cloud and Container Security Research, currently exploring Blockchain

$ IaC - Infrastructure As Code

What is it and why do I care ?

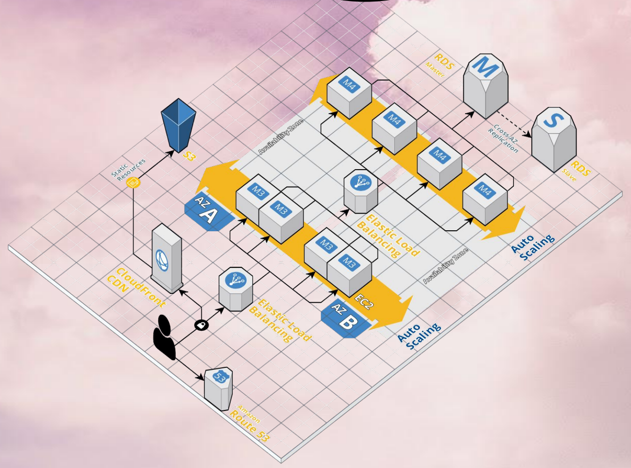
Infrastructure as Code is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.

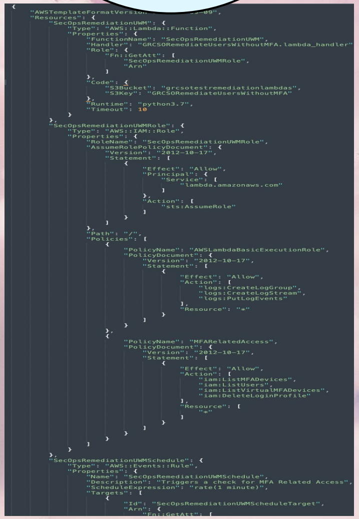
Source: Wikipedia

$ IaC - Infrastructure As Code

Current IaC Practices…

Design → Code → Deploy





```
{
    "AWSTemplateFormatVersion          -09",
    "Resources": {
        "SecOpsRemediationUWM": {
            "Type": "AWS::Lambda::Function",
            "Properties": {
                "FunctionName": "SecOpsRemediationUWM",
                "Handler": "GRCSORemediateUsersWithoutMFA.lambda_handler"
                "Role": {
                    "Fn::GetAtt": [
                        "SecOpsRemediationUWMRole",
                        "Arn"
                    ]
                },
                "Code": {
                    "S3Bucket": "grcsotestremediationlambdas",
                    "S3Key": "GRCSORemediateUsersWithoutMFA"
                },
                "Runtime": "python3.7",
                "Timeout": 10
            }
        },
        "SecOpsRemediationUWMRole": {
            "Type": "AWS::IAM::Role",
            "Properties": {
                "RoleName": "SecOpsRemediationUWMRole",
                "AssumeRolePolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": [
                                    "lambda.amazonaws.com"
                                ]
                            },
                            "Action": [
                                "sts:AssumeRole"
                            ]
                        }
                    ]
                },
                "Path": "/",
                "Policies": [
                    {
                        "PolicyName": "AWSLambdaBasicExecutionRole",
                        "PolicyDocument": {
                            "Version": "2012-10-17",
                            "Statement": [
                                {
                                    "Effect": "Allow",
                                    "Action": [
                                        "logs:CreateLogGroup",
                                        "logs:CreateLogStream",
                                        "logs:PutLogEvents"
                                    ],
                                    "Resource": "*"
                                }
                            ]
                        }
                    },
                    {
                        "PolicyName": "MFARelatedAccess",
                        "PolicyDocument": {
                            "Version": "2012-10-17",
                            "Statement": [
                                {
                                    "Effect": "Allow",
                                    "Action": [
                                        "iam:ListMFADevices",
                                        "iam:ListUsers",
                                        "iam:ListVirtualMFADevices",
                                        "iam:DeleteLoginProfile"
                                    ],
                                    "Resource": [
                                        "*"
                                    ]
                                }
                            ]
                        }
                    }
                ]
            }
        },
        "SecOpsRemediationUWMSchedule": {
            "Type": "AWS::Events::Rule",
            "Properties": {
                "Name": "SecOpsRemediationUWMSchedule",
                "Description": "Triggers a check for MFA Related Access",
                "ScheduleExpression": "rate(1 minute)",
                "Targets": [
                    {
                        "Id": "SecOpsRemediationUWMScheduleTarget",
                        "Arn": {
                            "Fn::GetAtt": [
```

$ Benefits of IaC

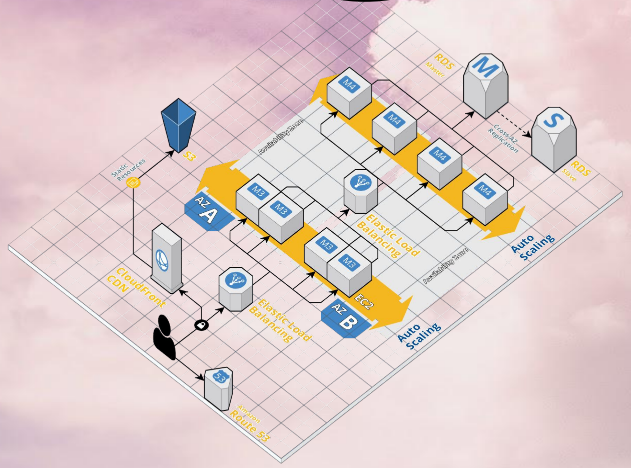
Less work, more manageable…

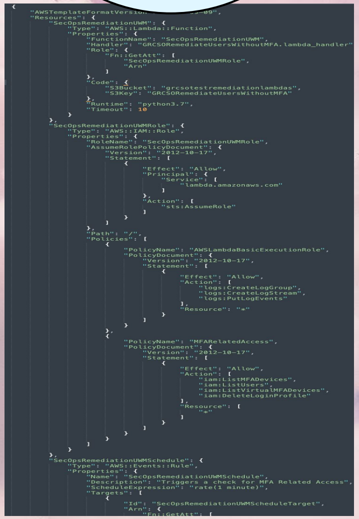
- Introduces the familiar Git workflow to managing infrastructure.
- Infrastructure can be deployed in a repeatable fashion.
- Increases deployment and build pipeline efficiency.
- Codifies runbooks eventually minimizing the human interference.
- Helps in infrastructure audits and troubleshooting incidents.
- Hassle free cloud inventory management.

$ IaC - Infrastructure As Code

How it is prevailing in the IT industry ?

Design → Code ✗ → Deploy





```
{
    "AWSTemplateFormatVersion [illegible]-09",
    "Resources": {
        "SecOpsRemediationUWM": {
            "Type": "AWS::Lambda::Function",
            "Properties": {
                "FunctionName": "SecOpsRemediationUWM",
                "Handler": "GRCSORemediateUsersWithoutMFA.lambda_handler"
                "Role": {
                    "Fn::GetAtt": [
                        "SecOpsRemediationUWMRole",
                        "Arn"
                    ]
                },
                "Code": {
                    "S3Bucket": "grcsotestremediationlambdas",
                    "S3Key": "GRCSORemediateUsersWithoutMFA"
                },
                "Runtime": "python3.7",
                "Timeout": 10
            }
        },
        "SecOpsRemediationUWMRole": {
            "Type": "AWS::IAM::Role",
            "Properties": {
                "RoleName": "SecOpsRemediationUWMRole",
                "AssumeRolePolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": [
                                    "lambda.amazonaws.com"
                                ]
                            },
                            "Action": [
                                "sts:AssumeRole"
                            ]
                        }
                    ]
                },
                "Path": "/",
                "Policies": [
                    {
                        "PolicyName": "AWSLambdaBasicExecutionRole",
                        "PolicyDocument": {
                            "Version": "2012-10-17",
                            "Statement": [
                                {
                                    "Effect": "Allow",
                                    "Action": [
                                        "logs:CreateLogGroup",
                                        "logs:CreateLogStream",
                                        "logs:PutLogEvents"
                                    ],
                                    "Resource": "*"
                                }
                            ]
                        }
                    },
                    {
                        "PolicyName": "MFARelatedAccess",
                        "PolicyDocument": {
                            "Version": "2012-10-17",
                            "Statement": [
                                {
                                    "Effect": "Allow",
                                    "Action": [
                                        "iam:ListMFADevices",
                                        "iam:ListUsers",
                                        "iam:ListVirtualMFADevices",
                                        "iam:DeleteLoginProfile"
                                    ],
                                    "Resource": [
                                        "*"
                                    ]
                                }
                            ]
                        }
                    }
                ]
            }
        },
        "SecOpsRemediationUWMSchedule": {
            "Type": "AWS::Events::Rule",
            "Properties": {
                "Name": "SecOpsRemediationUWMSchedule",
                "Description": "Triggers a check for MFA Related Access",
                "ScheduleExpression": "rate(1 minute)",
                "Targets": [
                    {
                        "Id": "SecOpsRemediationUWMScheduleTarget",
                        "Arn": {
                            "Fn::GetAtt": [
```

$ Cloud Security Scenario (2020)

Cloud Security and IaC - What's the connection ?

Nearly 200K Insecure Templates in Use

43% of Cloud Databases are NOT encrypted

60% of Cloud Storage services have logging disabled

76% of organisations expose SSH (port 22)

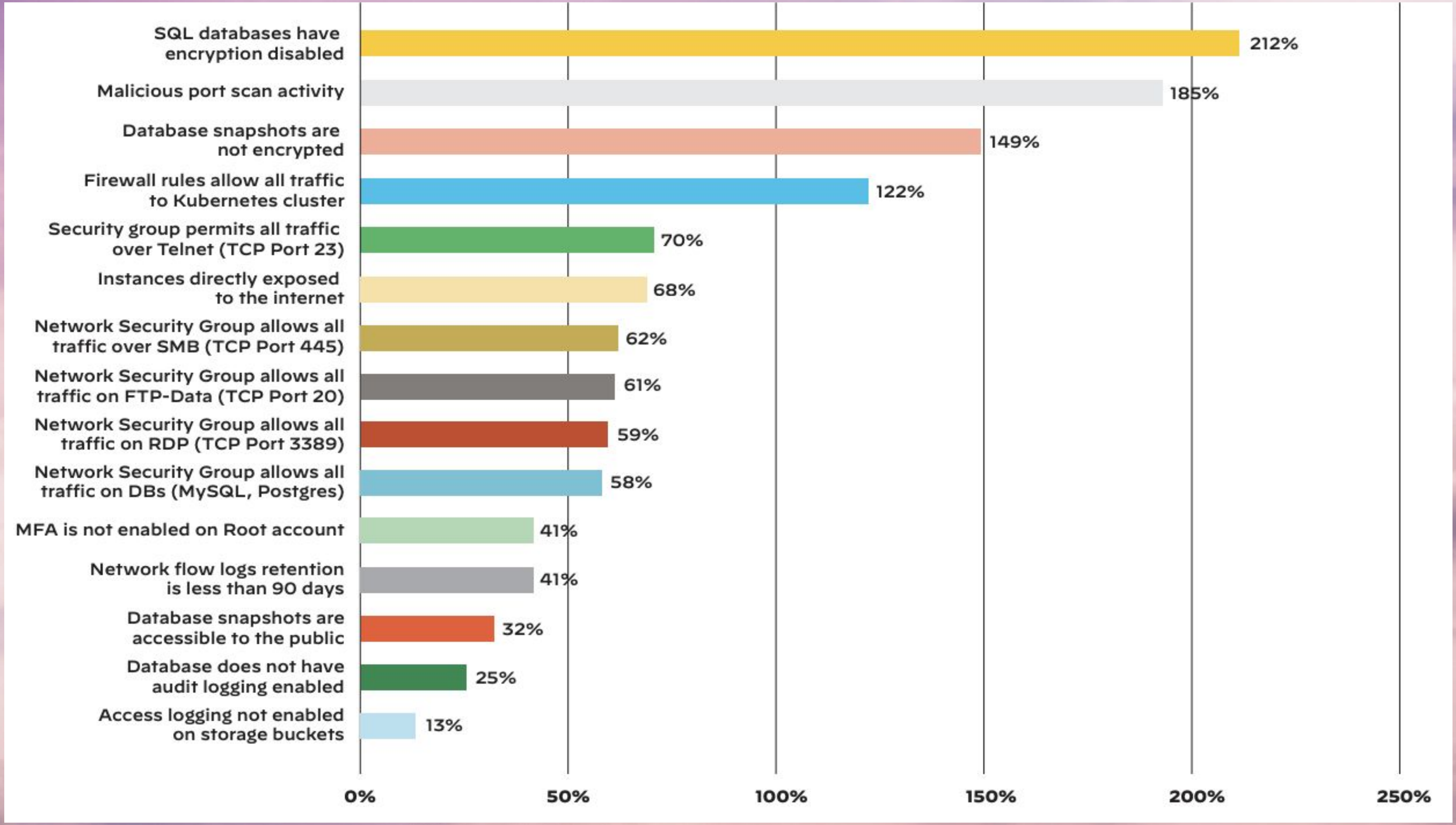
69% of organisations expose RDP (port 3389)

27% of organisations use outdated versions of TLS

Source: Unit42 Cloud Threat Report Spring 2020

$ Increase in Security Incidents (2021)

Cloud Security and IaC - What’s the connection ?



Source: Unit42 Cloud Threat Report 1H 2021

$ Increase in Security Incidents (2021)

Cloud Security and IaC - The connection is Shift Left Philosophy…

- By design, IaC doesn’t present itself as an immediate risk or attack surface. But because IaC is governed by engineering and DevOps, security teams may often overlook it, instead focusing on monitoring cloud resources already in production (where risk is more immediately addressable).

- For example, a publicly exposed S3 bucket might seem like more of an immediate risk than code that could result in exposed buckets if the right protective layers aren’t in place.

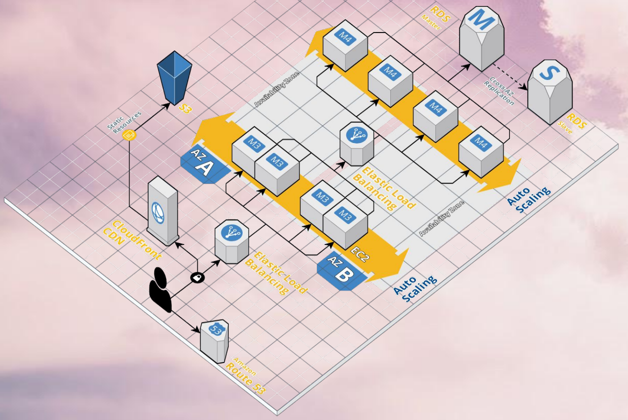
- While there is some truth to that, you may be overlooking some benefits and risks of ignoring IaC’s security implications.

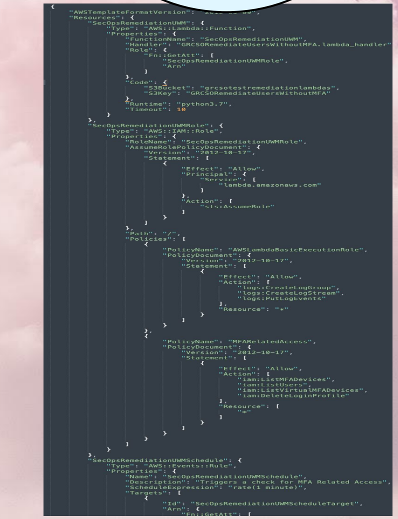
- IaC Security comes to rescue us under the hood as a part of Shift Left Philosophy.

$ IaC and DevSecOps

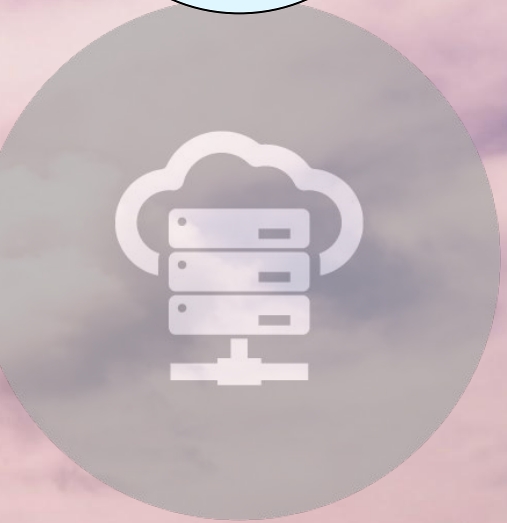
One more stage of Testing…

Design → Code → Security Testing → Deploy





```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "SecOpsRemediationUWM": {
            "Type": "AWS::Lambda::Function",
            "Properties": {
                "FunctionName": "SecOpsRemediationUWM",
                "Handler": "GRCSORemediateUsersWithoutMFA.lambda_handler"
                "Role": {
                    "Fn::GetAtt": [
                        "SecOpsRemediationUWMRole",
                        "Arn"
                    ]
                },
                "Code": {
                    "S3Bucket": "grcsotestremediationlambdas",
                    "S3Key": "GRCSORemediateUsersWithoutMFA"
                },
                "Runtime": "python3.7",
                "Timeout": 10
            }
        },
        "SecOpsRemediationUWMRole": {
            "Type": "AWS::IAM::Role",
            "Properties": {
                "RoleName": "SecOpsRemediationUWMRole",
                "AssumeRolePolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": [
                                    "lambda.amazonaws.com"
                                ]
                            },
                            "Action": [
                                "sts:AssumeRole"
                            ]
                        }
                    ]
                },
                "Path": "/",
                "Policies": [
                    {
                        "PolicyName": "AWSLambdaBasicExecutionRole",
                        "PolicyDocument": {
                            "Version": "2012-10-17",
                            "Statement": [
                                {
                                    "Effect": "Allow",
                                    "Action": [
                                        "logs:CreateLogGroup",
                                        "logs:CreateLogStream",
                                        "logs:PutLogEvents"
                                    ],
                                    "Resource": "*"
                                }
                            ]
                        }
                    },
                    {
                        "PolicyName": "MFARelatedAccess",
                        "PolicyDocument": {
                            "Version": "2012-10-17",
                            "Statement": [
                                {
                                    "Effect": "Allow",
                                    "Action": [
                                        "iam:ListMFADevices",
                                        "iam:ListUsers",
                                        "iam:ListVirtualMFADevices",
                                        "iam:DeleteLoginProfile"
                                    ],
                                    "Resource": [
                                        "*"
                                    ]
                                }
                            ]
                        }
                    }
                ]
            }
        },
        "SecOpsRemediationUWMSchedule": {
            "Type": "AWS::Events::Rule",
            "Properties": {
                "Name": "SecOpsRemediationUWMSchedule",
                "Description": "Triggers a check for MFA Related Access",
                "ScheduleExpression": "rate(1 minute)",
                "Targets": [
                    {
                        "Id": "SecOpsRemediationUWMScheduleTarget",
                        "Arn": {
                            "Fn::GetAtt": [
```



$ Different IaC Methods

Everything is automation but…

Orchestration

Arranging or co-ordinating multiple systems. Orchestration is how one can automate a process or workflow that involves many steps across multiple disparate systems.

Configuration Management

A collection of activities focused on establishing and maintaining the integrity of products and systems, through control of the process for initializing, changing, and monitoring the configuration of those products and systems throughout the Systems Development LifeCycle.

$ Different IaC Methods

Tools Tools Everywhere…

Orchestration

- Docker
- Kubernetes
- Terraform
- Cloudformation

Configuration Management

- Chef
- Puppet
- Ansible
- Saltstack

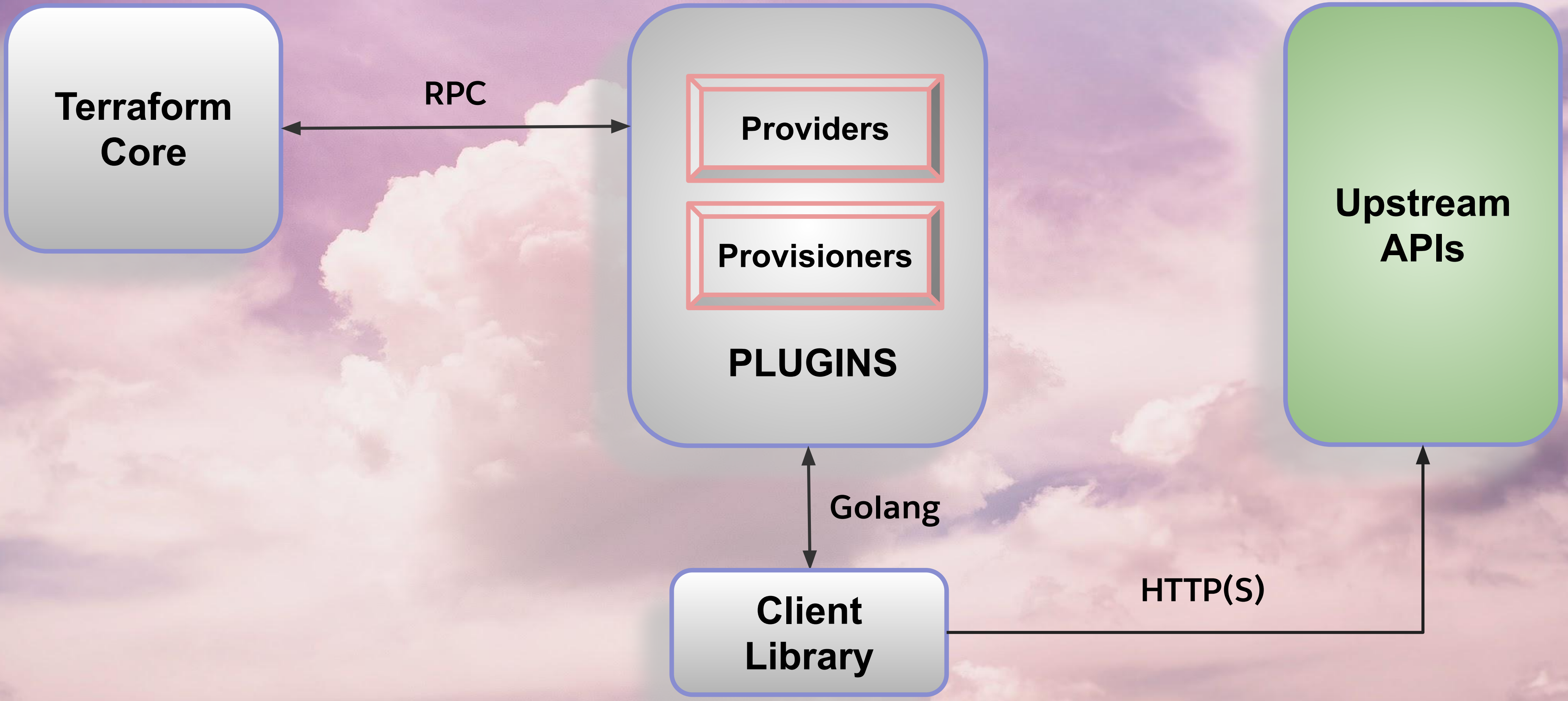
$ IaC with Terraform - Introduction

A brief outline…

- A provisioning declarative open-source tool
- Based on HCL - Hashicorp Configuration Language - Written in Golang
- Simple, Modular, Composable; Immutable
- Applies Graph Theory to IaC Paradigm
- Composes multiple tiers (SaaS/PaaS/IaaS)
- Plugin Based Architecture Model (executable binaries in Go communicate via RPC)

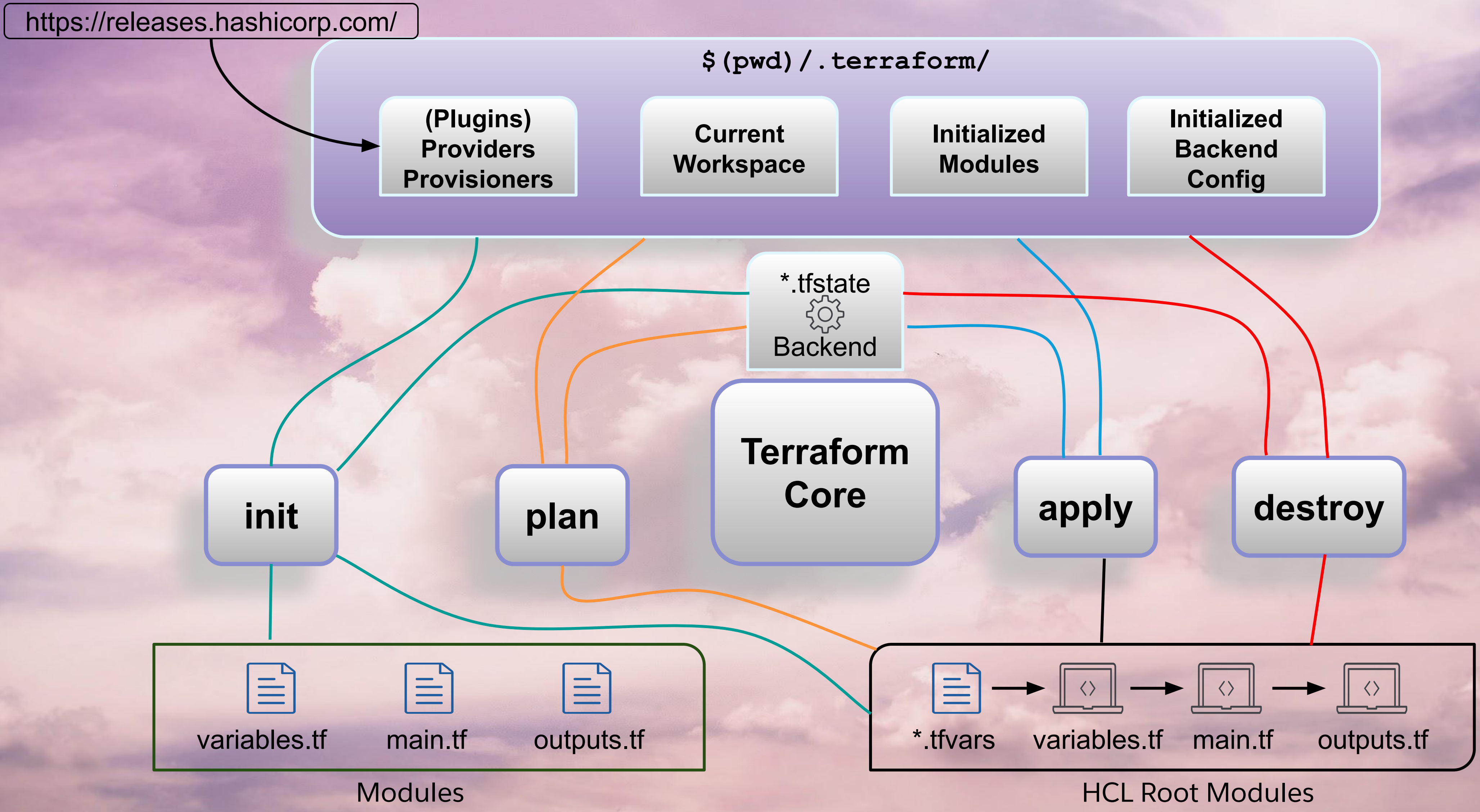
$ IaC with Terraform - How it works ?

High Level Overview…



$ IaC with Terraform - How it works ?

Little In-depth…



$ IaC with Terraform - Benefits

What does Terraform even got you ?

- Dependency graph calculations for infrastructure resources
- Lifecycle management of cloud resources (**CRUD**)
- Cross Cloud Resource Management under one umbrella
- Declarative Language for defining cloud resource topology
  [A Centralised PUSH Approach]
- In majority of cases, Terraform creates all resources as immutable where it doesn't waste time fiddling with deployed resources.
  [Speed vs. Experimentation Advantages]

$ IaC - Security Best Practices

Isn't it a lot of security ?

Operational Security

Pertaining to how the engineering and security are collaborating in the IaC space with adequate process oriented security.

Architectural Security

Reflects how well the security is prioritised for an environment to be provisioned based on the provider security best practices, basically the Shift Left Security Testing.

IaC Tool Security

Ensure the operating IaC Pipelines are following the best practices to avoid any security concerns.

$ IaC - Operational Security (Terraform)

Must but overlooked more often…

- Deployment pipelines must provide appropriate access controls (e.g. Google or AWS IAM Policies) on who is allowed to run a Terraform 'apply' in a given environment with a given set of input vars. Hardening of server is must.

- Compartmentalise related resources into separate state files.

- Infra should be broken into smallest possible units of management, with this reflected in the layout of the manifests in the project.

- Per environment things such as vars, workspaces, etc. should be clearly namespaced so that changes are not applied to wrong environment.

$ IaC - Architectural Security (Terraform)

DevOps won't care if Infra is functional…

- Based on the provider being utilised, these vary greatly and have significant role in security of the infrastructure being provisioned.

- Terraform plans with state files provide us with this great ability - "Prevention is better than cure"

- Cloud misconfigurations can be detected and eliminated. e.g. Leaky S3 Buckets.

- There are different set of tools available as open source to test and identify the Terraform plans.

$ IaC - Terraform Security

Are you aware of this ?

- Do NOT commit '.tfstate' file to the codebase repos. [Huge Risk]
- Configure S3/Postgres backend for '.tfstate' with restricted access, encryption at rest and versioning.
- DO NOT expose any SECRETS in an unencrypted way to end users either in output vars or in plan/apply output. Use SENSITIVE parameter for creds.
- Set TF_LOG and TF_LOG_PATH for troubleshooting.
- Say NO to hard coded values, double references, etc.
- To prevent mutating commands, state locking can be utilised.

$ Terraform Security - checkov, tfsec and TerraGoat

Time for some hands-on…

- *checkov* is an open-source static analysis and policy-as-code engine for Terraform, CloudFormation, Kubernetes, Azure Resource Manager and Serverless Framework. [Scans both Code and Plan]

- *tfsec* is an open-source static analysis security scanner for Terraform. It is designed to run locally and in the CI pipelines with developer-friendly output and fully documented checks. [Scans only Code and not Plan]

- *TerraGoat* is a vulnerable-by-design open-source Terraform project designed to give DevOps engineers a place to learn how to identifying misconfigured IaC modules and test them without polluting our own professional AWS accounts.

$ Terraform Security - checkov, tfsec and TerraGoat

Time for some hands-on…

Demo Time

$ References

- Infrastructure as Code - Wikipedia [https://en.wikipedia.org/wiki/Infrastructure_as_code]
- Unit42 Cloud Threat Report - Spring 2020
- Unit42 Cloud Threat Report - 1H 2021
- Terraform Documentation [https://www.terraform.io/docs]
- tfsec - Github [https://aquasecurity.github.io/tfsec/v0.63.1/]
- checkov - Github [https://github.com/bridgecrewio/checkov]
- TerraGoat - Github [https://github.com/bridgecrewio/terragoat/]

Thank You...