4.5

4.5 Million

4.5 Million

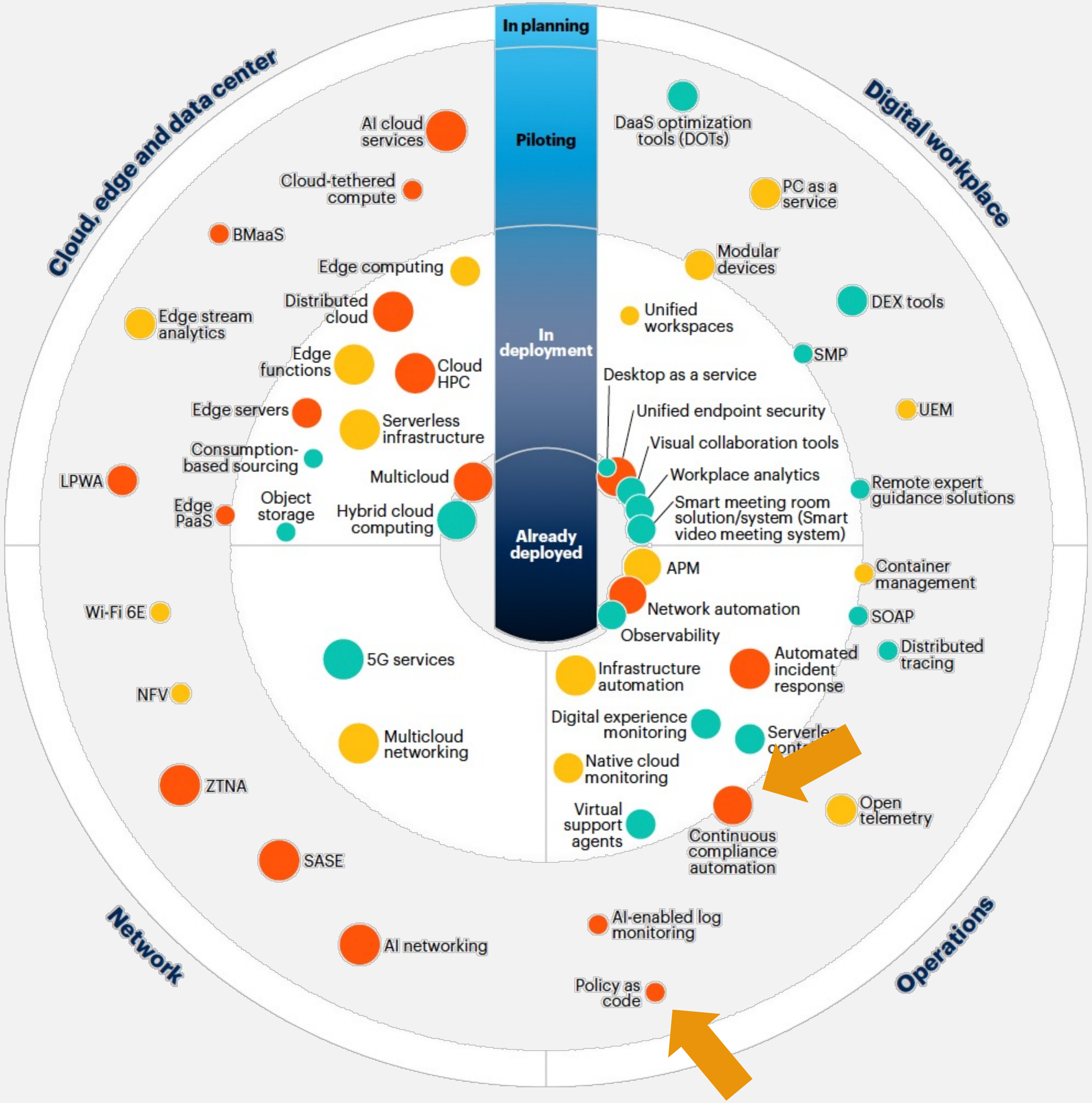
The majority of public cloud security incidents are due to misconfigured resources not from attacks targeting the cloud infrastructure itself

To find the right balance between Agility and Compliance the security and governance processes must evolve to support modern Cloud infrastructures

A compliance program is a set of requirements that meet a specific standard (internal or external).

A requirement is expressed by a statement or assertion, the policy, that specifies the correct or expected behaviour of an entity

Transition to Compliance-as-Code embeds policy checks into the automation of infrastructure and software deployment, enhancing both speed and compliance


In planning
Piloting
In deployment
Already deployed
Cloud, edge and data center
Digital workplace
Network
Operations
AI cloud services
Cloud-tethered compute
BMaaS
Edge computing
Distributed cloud
Edge stream analytics
Edge functions
Cloud HPC
Edge servers
Serverless infrastructure
Consumption-based sourcing
Multicloud
LPWA
Edge PaaS
Object storage
Hybrid cloud computing
DaaS optimization tools (DOTs)
PC as a service
Modular devices
DEX tools
Unified workspaces
SMP
Desktop as a service
UEM
Unified endpoint security
Visual collaboration tools
Workplace analytics
Remote expert guidance solutions
Smart meeting room solution/system (Smart video meeting system)
APM
Container management
Network automation
SOAP
Observability
Automated incident response
Distributed tracing
Infrastructure automation
Digital experience monitoring
Serverle
cont
Native cloud monitoring
Open telemetry
Virtual support agents
Continuous compliance automation
AI-enabled log monitoring
Policy as code
Wi-Fi 6E
5G services
NFV
Multicloud networking
ZTNA
SASE
AI networking

Compliance As Code: shift-left and shift-right approach in a Cloud world

Paolo Latella

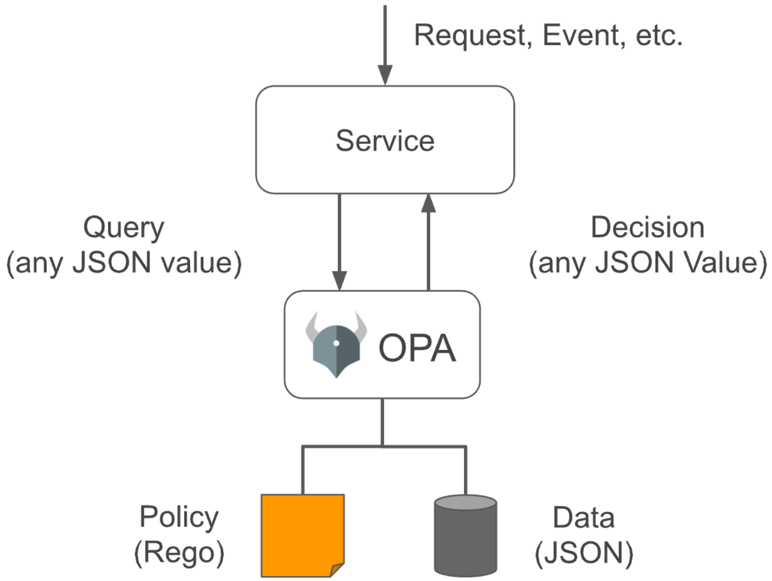
Cloud Advisor @ Recube
AWS Authorized Instructor
AWS Hero

in https://www.linkedin.com/in/paololatella/
@LatellaPaolo

reimagine skills to innovate
r3cube

We must define compliance policies in a way that allows them to be encoded, automated, and integrated into the cloud infrastructure

Encoded – Open Policy Agent



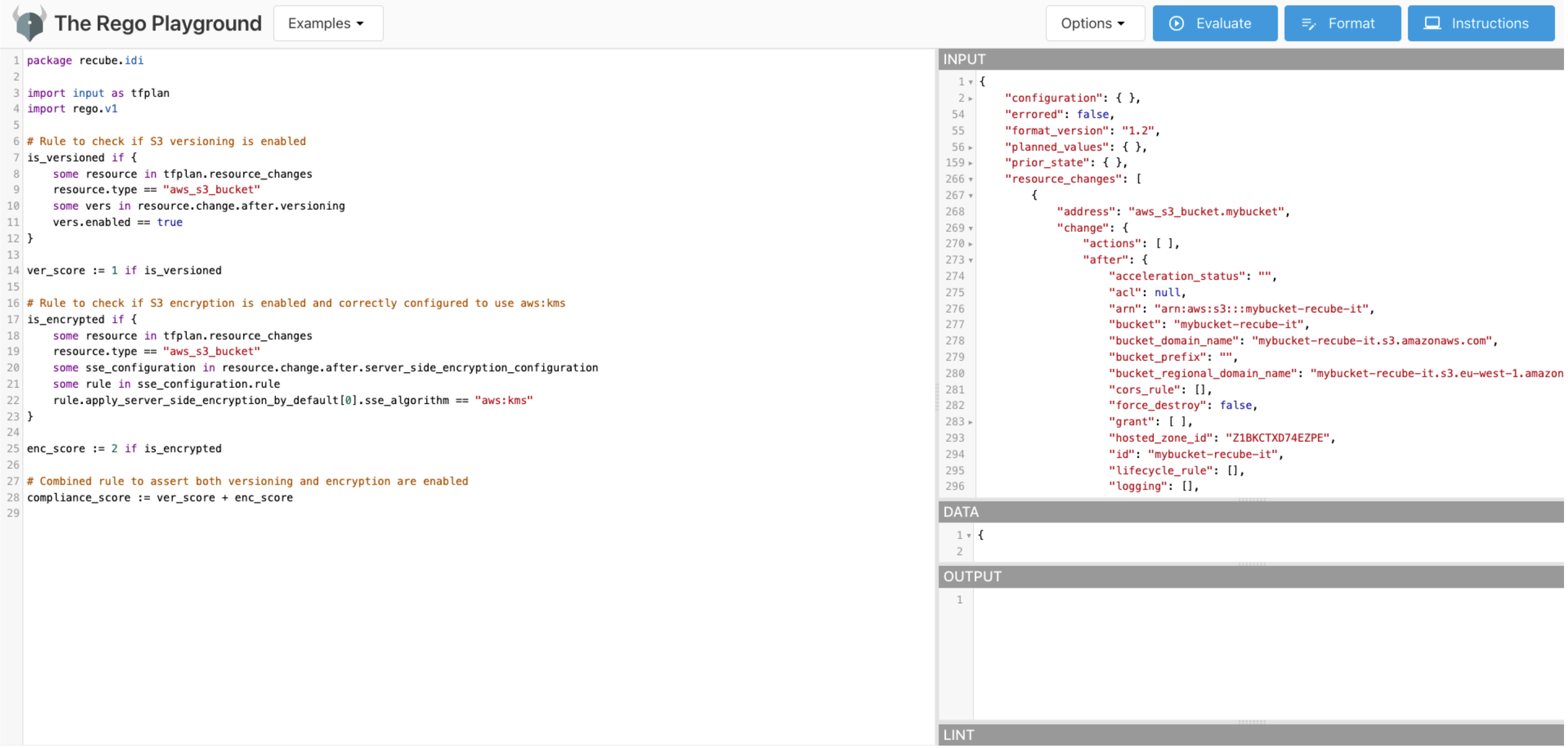
- Open source, general-purpose policy engine
  - OPA Policies written in Rego
  - Input in JSON
- A graduated project in the CNCF landscape
- Daemon

```
./opa run --server ./rules.rego
```

- Library
- CLI Tool

```
./opa eval -i template.json -d opa.rule 'data'
```

OPA - Terraform



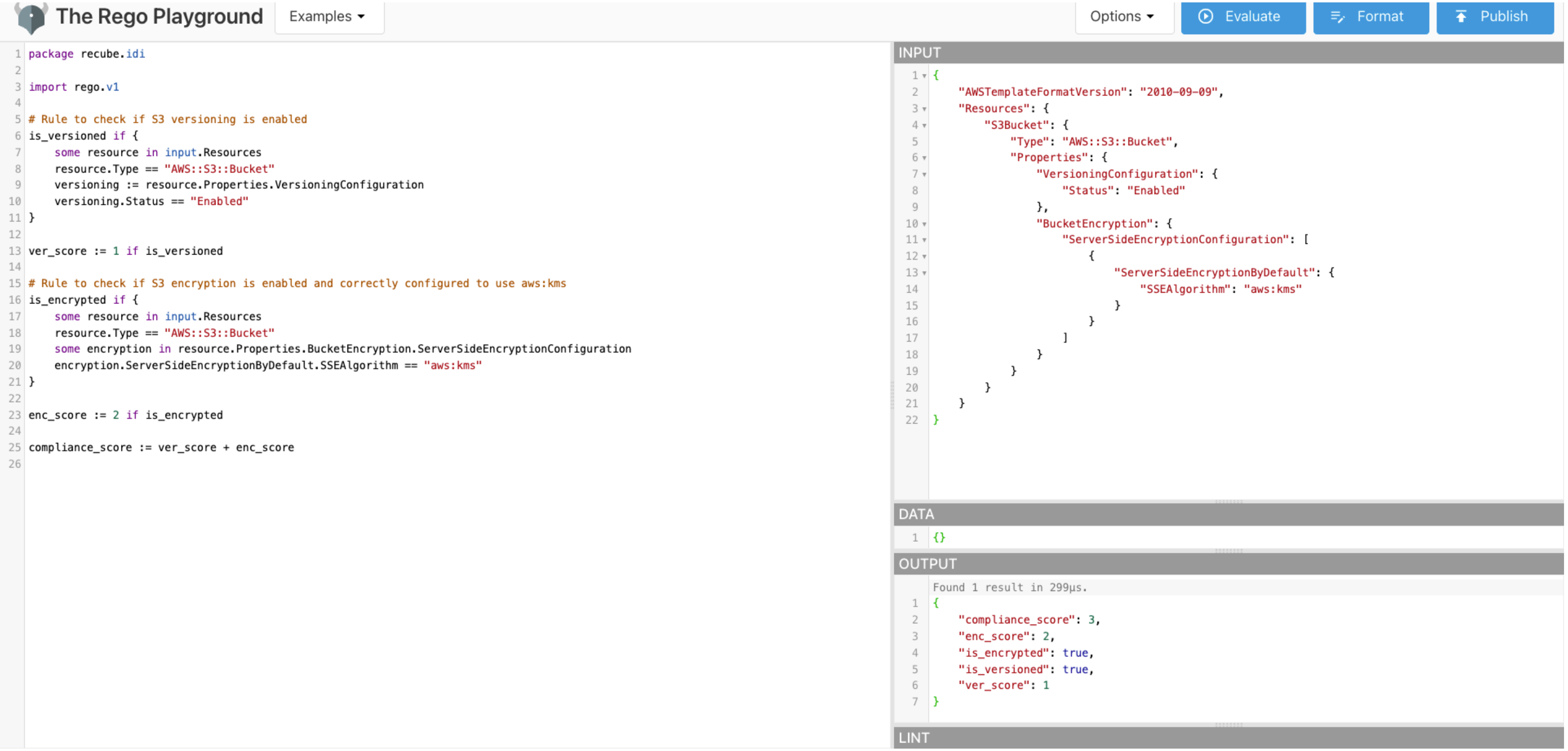
```
package recube.idi

import input as tfplan
import rego.v1

# Rule to check if S3 versioning is enabled
is_versioned if {
    some resource in tfplan.resource_changes
    resource.type == "aws_s3_bucket"
    some vers in resource.change.after.versioning
    vers.enabled == true
}

ver_score := 1 if is_versioned

# Rule to check if S3 encryption is enabled and correctly configured to use aws:kms
is_encrypted if {
    some resource in tfplan.resource_changes
    resource.type == "aws_s3_bucket"
    some sse_configuration in resource.change.after.server_side_encryption_configuration
    some rule in sse_configuration.rule
    rule.apply_server_side_encryption_by_default[0].sse_algorithm == "aws:kms"
}

enc_score := 2 if is_encrypted

# Combined rule to assert both versioning and encryption are enabled
compliance_score := ver_score + enc_score
```

```
{
    "configuration": { },
    "errored": false,
    "format_version": "1.2",
    "planned_values": { },
    "prior_state": { },
    "resource_changes": [
        {
            "address": "aws_s3_bucket.mybucket",
            "change": {
                "actions": [ ],
                "after": {
                    "acceleration_status": "",
                    "acl": null,
                    "arn": "arn:aws:s3:::mybucket-recube-it",
                    "bucket": "mybucket-recube-it",
                    "bucket_domain_name": "mybucket-recube-it.s3.amazonaws.com",
                    "bucket_prefix": "",
                    "bucket_regional_domain_name": "mybucket-recube-it.s3.eu-west-1.amazon
                    "cors_rule": [],
                    "force_destroy": false,
                    "grant": [ ],
                    "hosted_zone_id": "Z1BKCTXD74EZPE",
                    "id": "mybucket-recube-it",
                    "lifecycle_rule": [],
                    "logging": [],
```

```
{
```

```
terraform plan --out tfplan.binary

terraform show -json tfplan.binary > tfplan.json
```

OPA - Cloudformation



```
package recube.idi

import rego.v1

# Rule to check if S3 versioning is enabled
is_versioned if {
    some resource in input.Resources
    resource.Type == "AWS::S3::Bucket"
    versioning := resource.Properties.VersioningConfiguration
    versioning.Status == "Enabled"
}

ver_score := 1 if is_versioned

# Rule to check if S3 encryption is enabled and correctly configured to use aws:kms
is_encrypted if {
    some resource in input.Resources
    resource.Type == "AWS::S3::Bucket"
    some encryption in resource.Properties.BucketEncryption.ServerSideEncryptionConfiguration
    encryption.ServerSideEncryptionByDefault.SSEAlgorithm == "aws:kms"
}

enc_score := 2 if is_encrypted

compliance_score := ver_score + enc_score

```

INPUT

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "S3Bucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "VersioningConfiguration": {
                    "Status": "Enabled"
                },
                "BucketEncryption": {
                    "ServerSideEncryptionConfiguration": [
                        {
                            "ServerSideEncryptionByDefault": {
                                "SSEAlgorithm": "aws:kms"
                            }
                        }
                    ]
                }
            }
        }
    }
}
```

DATA

```
{}
```

OUTPUT

```
Found 1 result in 299µs.
{
    "compliance_score": 3,
    "enc_score": 2,
    "is_encrypted": true,
    "is_versioned": true,
    "ver_score": 1
}
```

LINT

Encoded – CFN Guard

- AWS CloudFormation Guard is an open-source, general-purpose, policy-as-code evaluation tool
- It provides a simple-to-use and expressive domain-specific language (DSL) to define policies to validate JSON- or YAML document

```
let variable = <resource type> <property name> < operator> <expression>

rule <rule name> [when <condition>] {
    <query> <operator> [query|value literal] [custom message] [or|OR]
    <query> <operator> [query|value literal] [custom message] [or|OR]
    ...
}
```

CFN Guard - Cloudformation

```
let s3_buckets = Resources.*[ Type == 'AWS::S3::Bucket' ]

rule s3_buckets_versioning when %s3_buckets !empty {
let versioning = %s3_buckets.Properties.VersioningConfiguration
%versioning exists
%versioning.Status == "Enabled"
}

rule s3_buckets_encrypted when %s3_buckets !empty {
let encryption =
%s3_buckets.Properties.BucketEncryption.ServerSideEncryptionConfiguration[*]
%encryption exists
%encryption.ServerSideEncryptionByDefault.SSEAlgorithm == "aws:kms"
}
```

```
cfn-guard validate --rules cfg_rules.guard --data cf_template_ko.json
cf_template_ko.json
cf_template_ko.json Status = FAIL
FAILED rules
cfg_rules.guard/s3_buckets_versioning    FAIL
---
Number of non-compliant resources 1
Resource = S3Bucket {
Type      = AWS::S3::Bucket
Rule = s3_buckets_versioning
```

Encoded – BDD in Compliance Testing

- In complex scenarios with multiple (non technical) stakeholder the BDD Compliance testing is more Effective

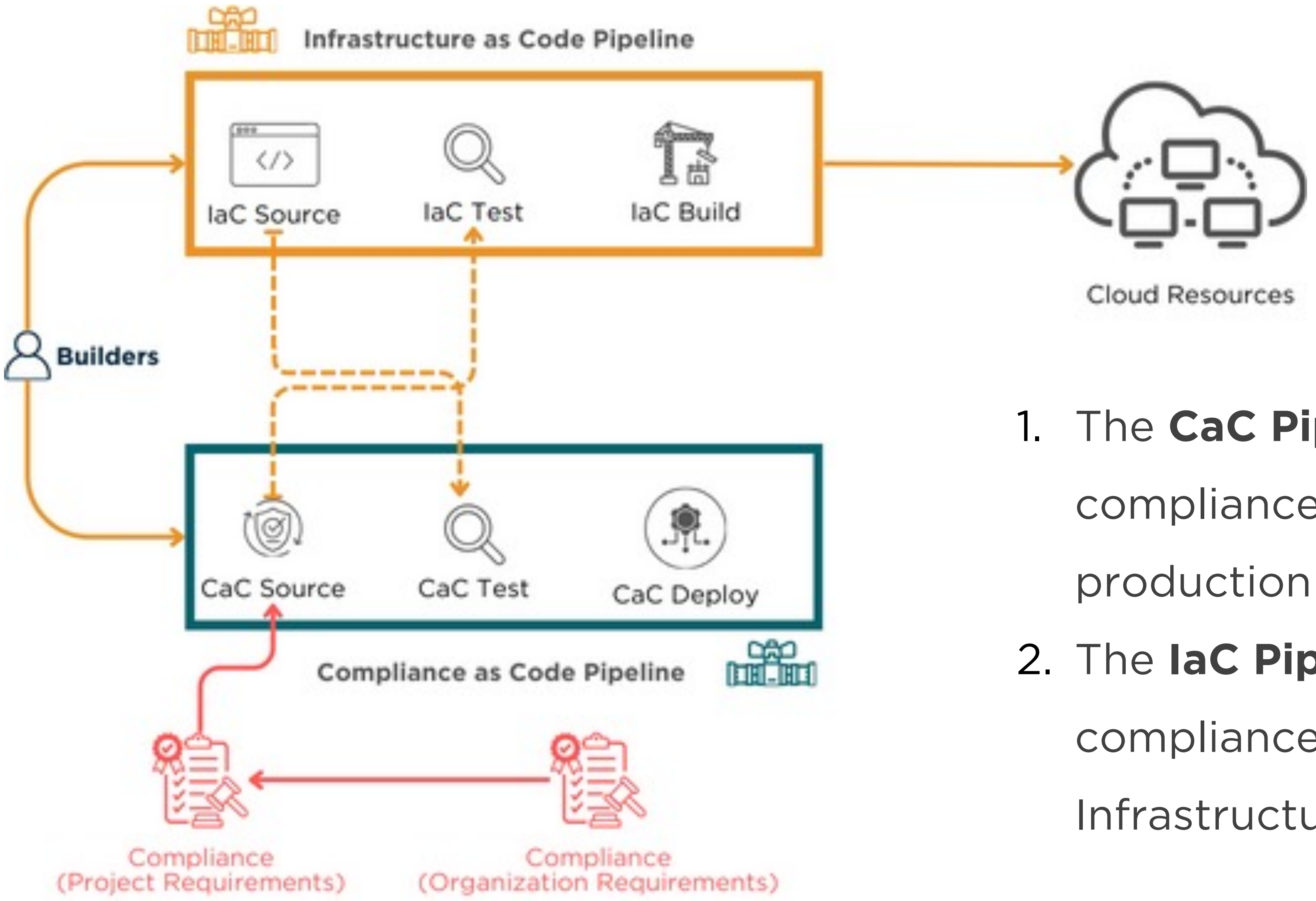
```
Feature: Ensure all S3 buckets are protected
  In order to enhance data security
  As an IT security specialist
  I want to ensure every S3 bucket is encrypted and versioned

  Scenario: S3 buckets must have versioning enabled
    Given I have aws_s3_bucket defined
    Then it must contain versioning
    Then it must contain enabled
    And its value must be true

  Scenario: S3 buckets must have server-side encryption enabled
    Given I have aws_s3_bucket defined
    Then it must contain server_side_encryption_configuration
    Then it must contain rule
    Then it must contain apply_server_side_encryption_by_default
    Then it must contain sse_algorithm
    And its value must be "aws:kms"
```

Encoded – BDD in Compliance Testing

- In a compliance testing scenarios, negative testing it’s more robust and efficient

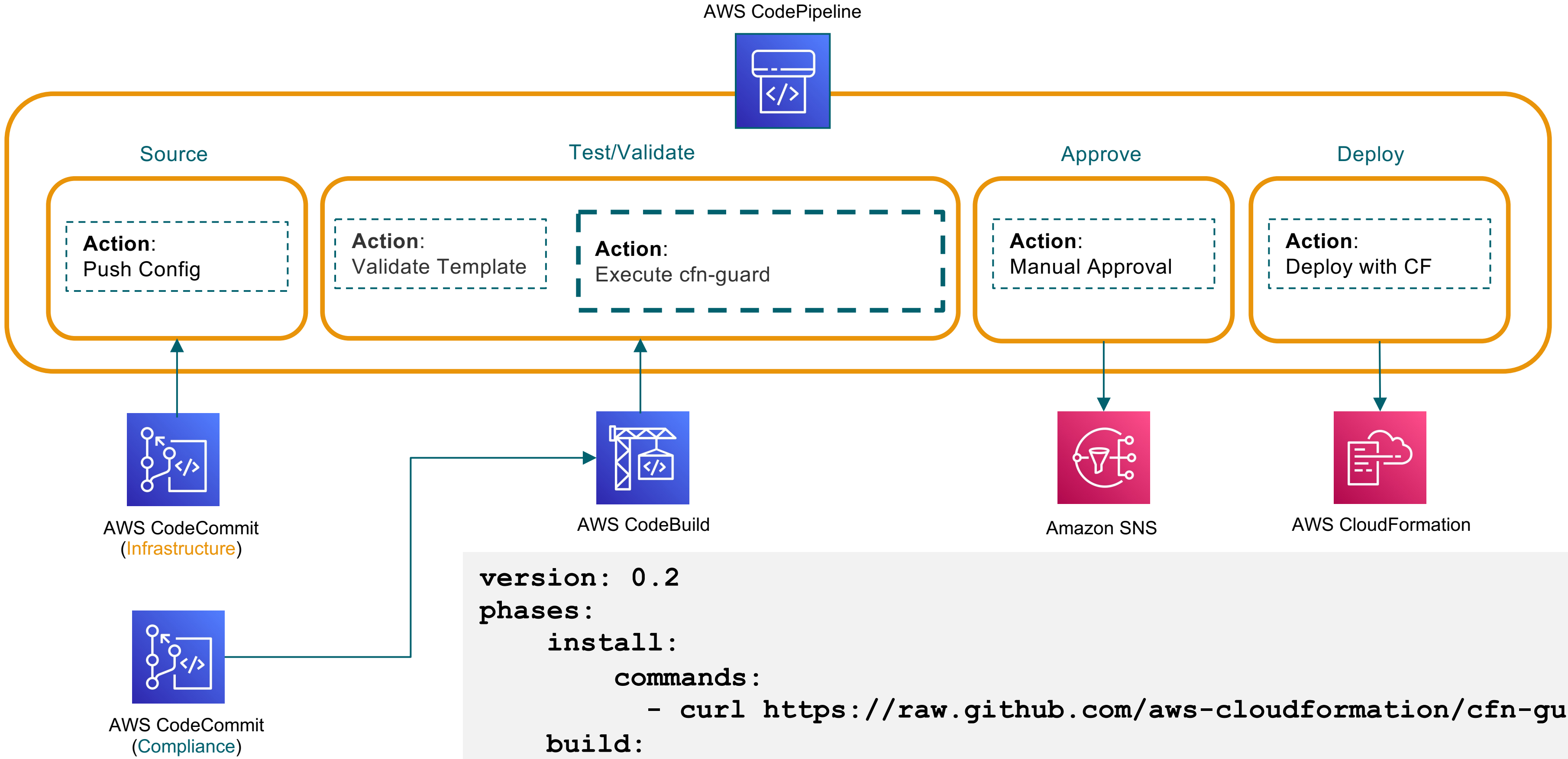
```
Feature: Ensure all S3 buckets are protected
  In order to enhance data security
  As an IT security specialist
  I want to ensure every S3 bucket is encrypted and versioned

  Scenario: S3 buckets must not be unversioned
    Given I have aws_s3_bucket defined
    Then it must contain versioning
    Then it must contain enabled
    And its value must not be false

  Scenario: S3 buckets must not be unencrypted
    Given I have aws_s3_bucket defined
    Then it must contain server_side_encryption_configuration
    Then it must contain rule
    And its value must not be null
```

Automated



1. The **CaC Pipeline** test new compliance check on IaC production ready template
2. The **IaC Pipeline** execute compliance check on new Infrastructure template

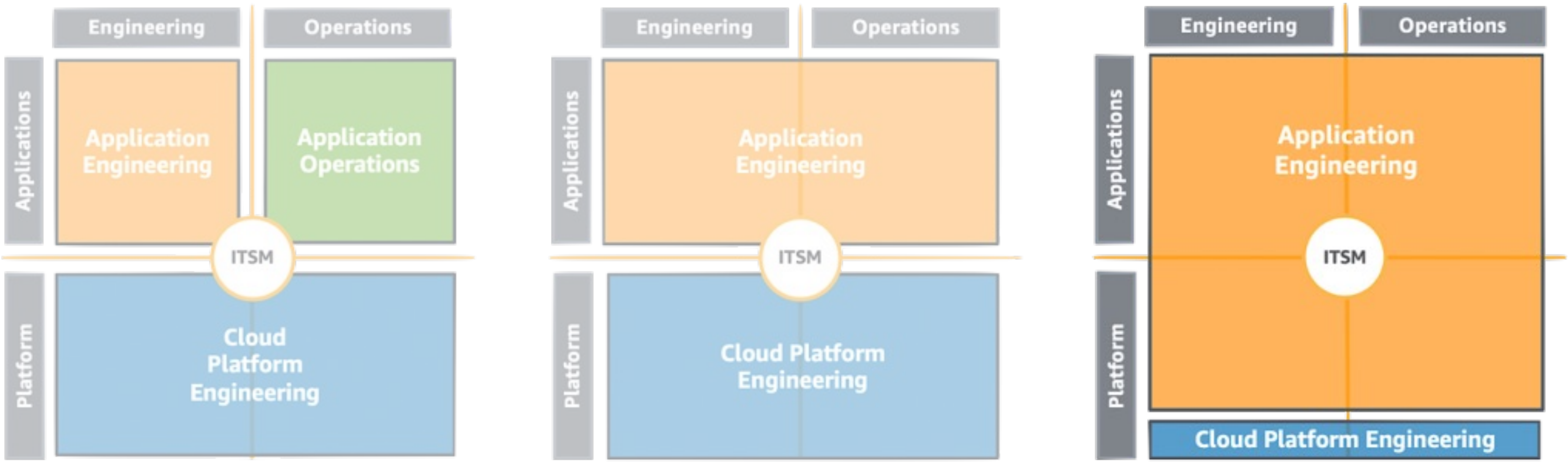
Automated – Example – AWS



```
version: 0.2
phases:
    install:
        commands:
          - curl https://raw.github.com/aws-cloudformation/cfn-guard/main/install-guard.sh | sh
    build:
        commands:
          - git pull git://github.com/recube/idi2024.git
          - cfnlint template.json
          - cfn-guard validate --rules rules --data template.json
```

We must adopt a federated governance model and encourage both Shif-Left and Shift-Right approach

Governance Model



Teams autonomously manage their resources within organizational compliance and security frameworks.

Shift LEFT

Development cycle

We can detect compliance issues very early in the process, but tests are limited in scope and related to a specific workload.

Tests are planned and executed by the project team

Tools can be agnostic to Cloud provider capabilities

Shift Right

Running Cycle

We can assess the resources against compliance requirements defined at the organization level

Tests are planned and executed by the platform team (under CCoE supervisor)

Tools (services) are often provided directly by Cloud provider

Shift LEFT

Static Code Analysis Tools

Shift Right

Static and Dynamic Code Analysis Tools

Shift LEFT

Static Code Analysis Tools

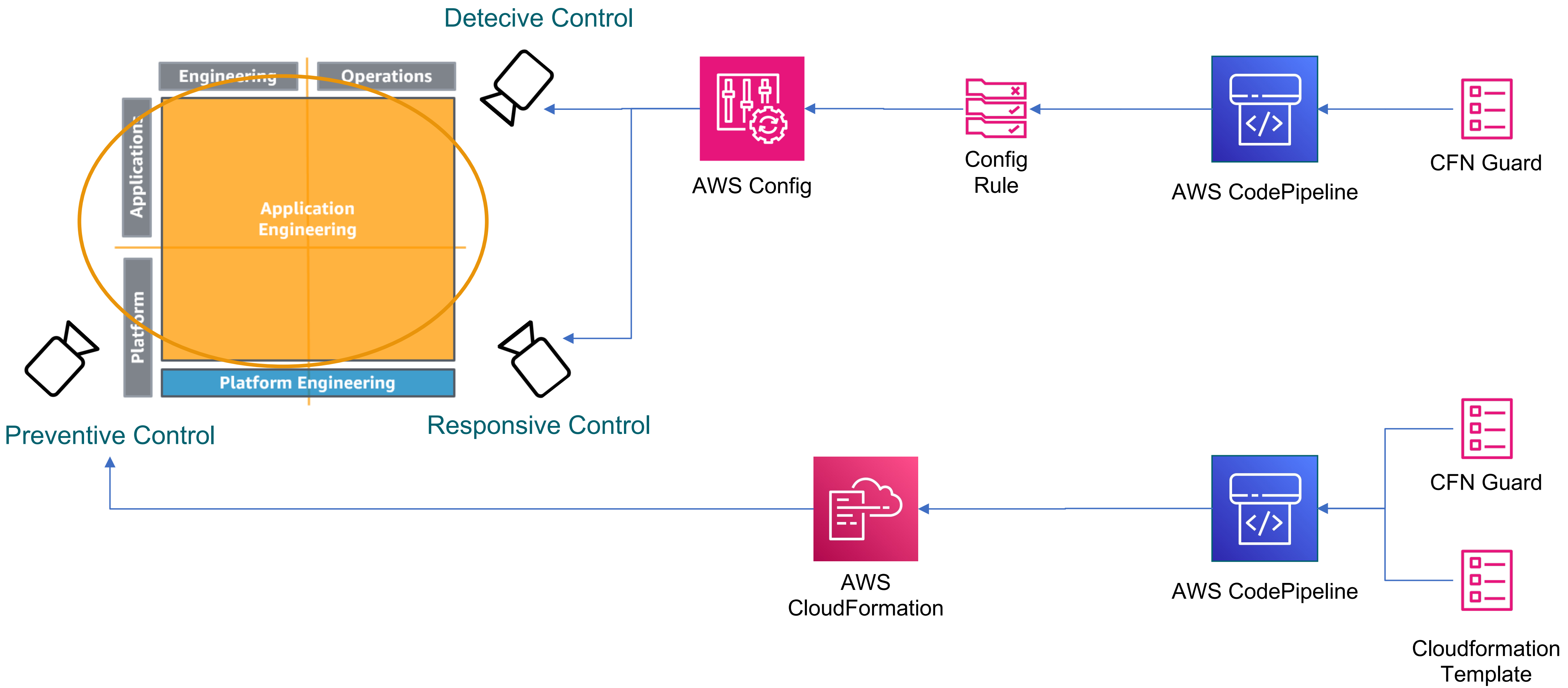
OPA
Cloudformation Guard
Terraform Compliance
Chef InSpec
Hashicorp Sentinel
Checkov
Regula

Shift Right

Static and Dynamic Code Analysis Tools

OPA *
Cloudformation Guard
AWS Config / Azure Policy
Cloud Custodian

Shift-Right – AWS Config and …



Detective Control

AWS Config

Config Rule

AWS CodePipeline

CFN Guard

Preventive Control

Responsive Control

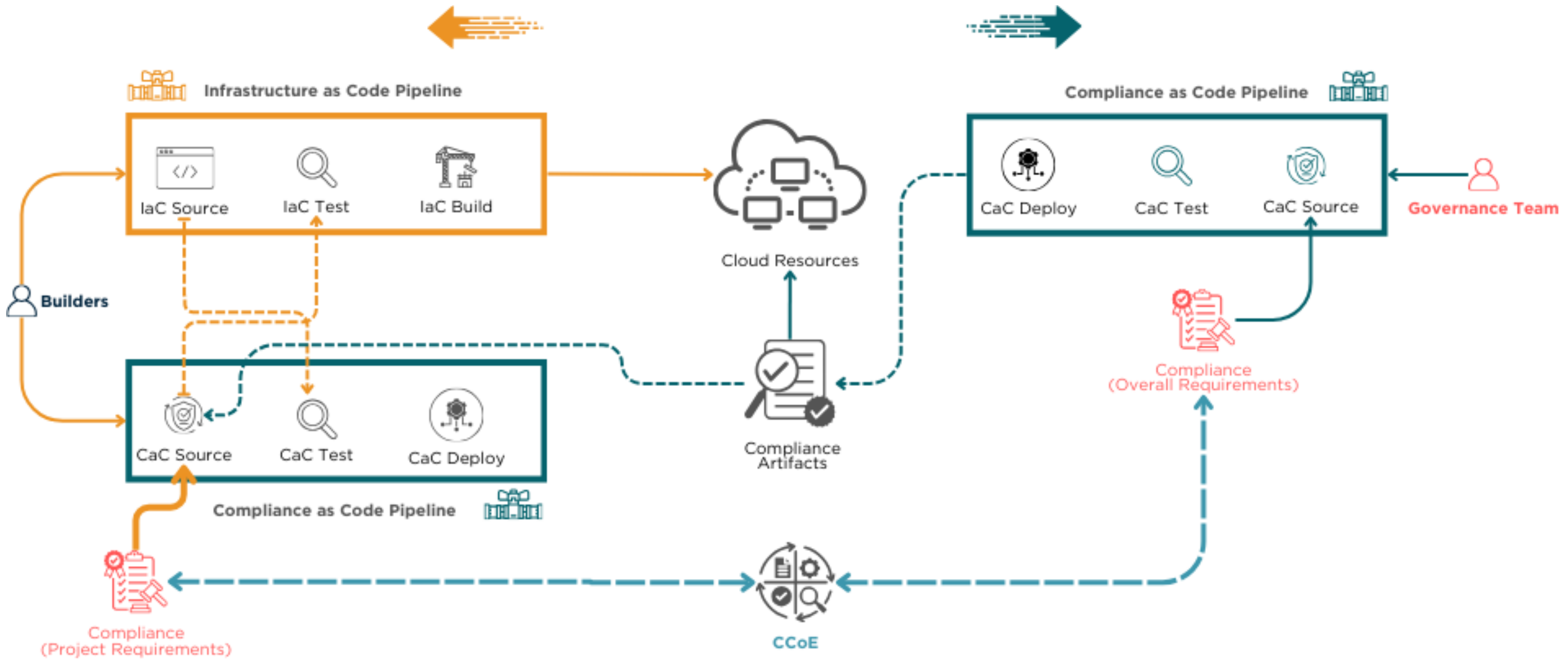
AWS CloudFormation

AWS CodePipeline

CFN Guard

Cloudformation Template

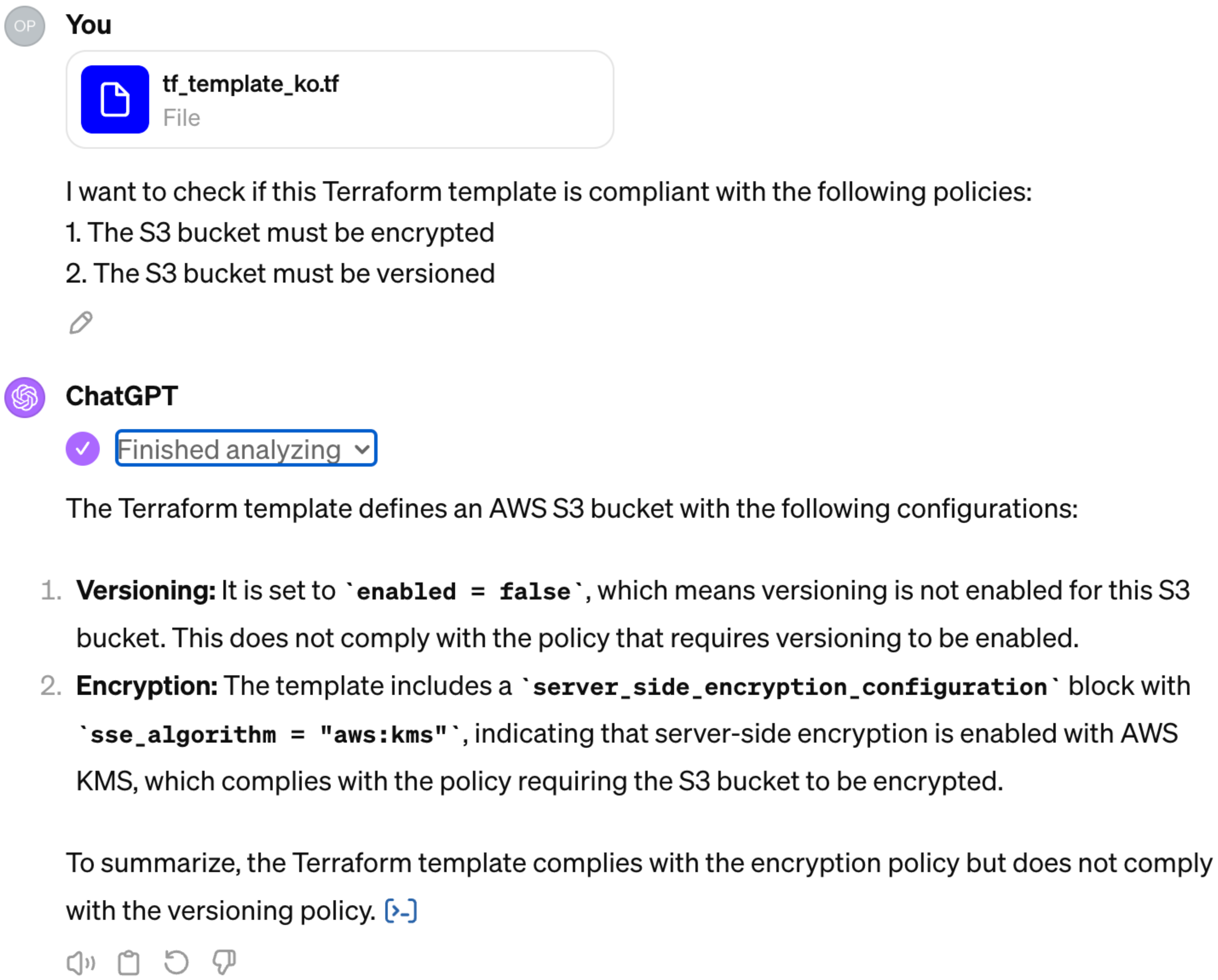
Shift-Left and Shift-Right - Automation



Thank You!

https://www.linkedin.com/in/paololatella/

@LatellaPaolo



tf_template_ko.tf

I want to check if this Terraform template is compliant with the following policies:
1. The S3 bucket must be encrypted
2. The S3 bucket must be versioned

The Terraform template defines an AWS S3 bucket with the following configurations:

1. **Versioning:** It is set to `enabled = false`, which means versioning is not enabled for this S3 bucket. This does not comply with the policy that requires versioning to be enabled.
2. **Encryption:** The template includes a `server_side_encryption_configuration` block with `sse_algorithm = "aws:kms"`, indicating that server-side encryption is enabled with AWS KMS, which complies with the policy requiring the S3 bucket to be encrypted.

To summarize, the Terraform template complies with the encryption policy but does not comply with the versioning policy.