styra

Terraform Secured By Open Policy Agent

Adding Policy-as-Code to Infrastructure-as-Code

Peter ONeill

6.9.22

About Me

1

Peter ONeill

2

Community Advocate

Advocating for the Open Policy Agent Community, building the project around the CNCF ecosystem.

3

Digital Nomad

I've worked in 25+ countries and still have only seen a tiny fraction of the planet.

4

Open Source Contributor

Open Source Software is changing how we develop applications and increasing the rate of innovation. We can all go further if we work together.

@peteroneilljr

Agenda

Securing your Terraform deployments

1

Terraform and IaC

A quick recap of Terraform and IaC.

2

GitOps and CI/CD

Deployment patterns for Terraform

3

Authorization

Discuss how authorization is currently used in IaC and how it can be optimized.

4

Decoupled Policies

How to decouple your policies with Open Policy Agent

5

Secure pipelines

Showing what points you should implement policies throughout your pipeline.

6

Demo

Terraform deployment with Open Policy Agent validations

Terraform, Infrastructure-as-Code

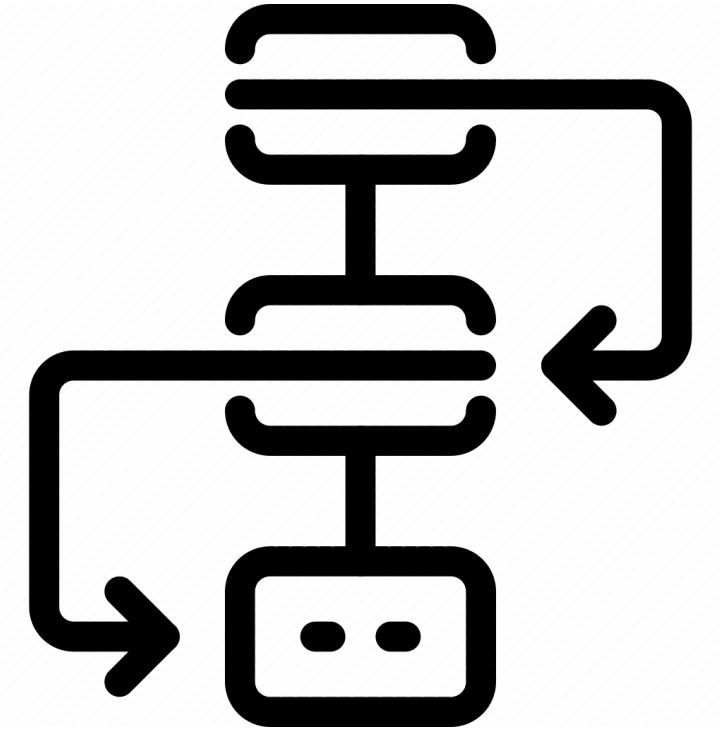
Declarative

Moving from Imperative to Declarative

Imperative provides the step by step instructions

Declarative states what you want to the end state to be

Terraform, Infrastructure-as-Code



Code

- Writing the Terraform Manifest Files

Plan

- Understanding what actions are going to happen

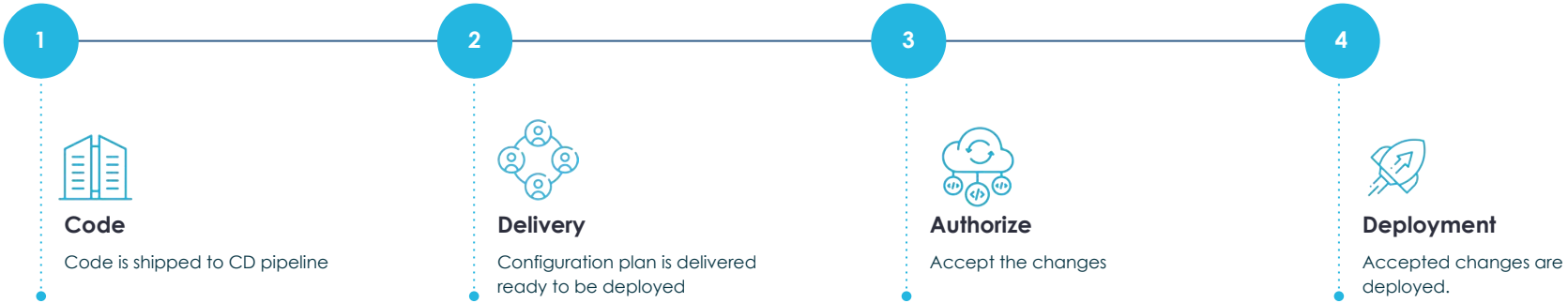
Apply

- Creating the infrastructure

Continuous Delivery or Continuous Deployment

Using GitOps for infrastructure

- **Single Source of Truth**
- **Declarative**
- **Versioned**
- **Automated**



Authorization

Not as simple as protect me from the malicious actors

01. Secrets

This one is kind of obvious, don't store keys in the code.

02. Malicious Actors

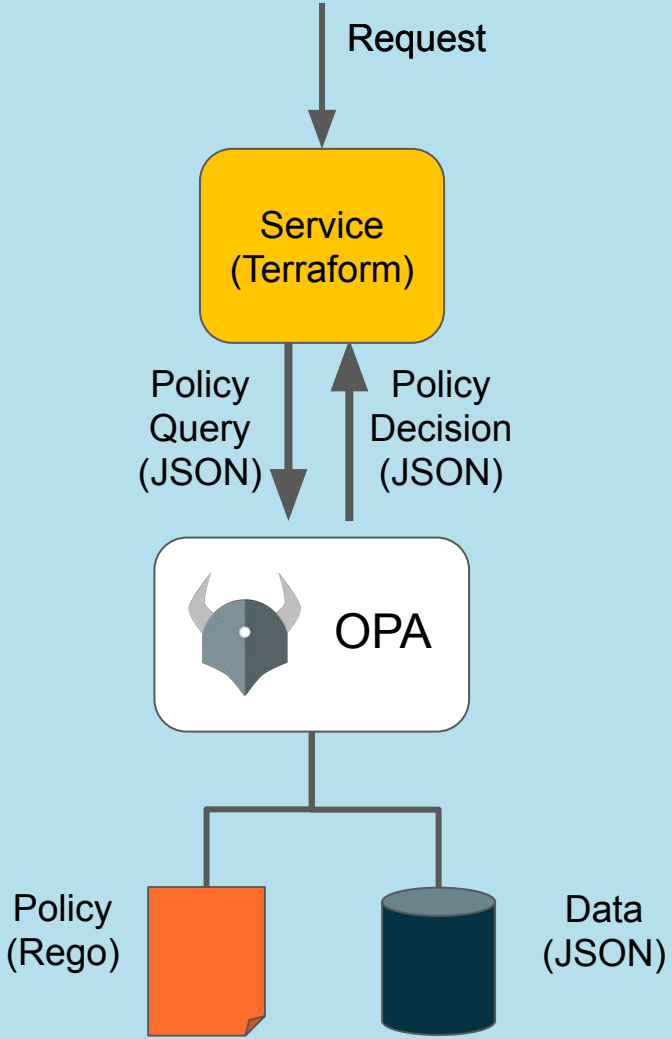
These can be known and unknown actors.

03. Unintentional Changes

One of the most common security risks.

04. Radical Changes

Changes typically fall into an ordinary set of parameters.



1

General Purpose

As a policy engine, OPA is designed to work with any of your services.

2

Policy-as-Code

Defined in Rego, a purpose-built policy language used write your policies.

3

Decoupled Policies

OPA sits next to your services, removing the policy logic from your service.

4

Policy Development Life Cycle

Authoring, testing, and deployment of your policies.

Secure your infrastructure with OPA

Let's put it all together and secure our Terraform deployment with Open Policy Agent

1

Create TF Manifest

Author your Terraform manifest files as you would normally.

2

Local Validation

Use OPA to double-check the manifest files.

3

Commit to Git

Create a PR and submit your changes.

4

Automated Testing

Use an automated testing tool like GitHub actions to authorize the changes using your Rego policies.

5

Deploy

Allow Terraform to create an authorized deployment.

styra | HashiCorp Terraform

Terraform and OPA Demo

Evaluating your local manifest files

&

Authorizing your changes with GitHub Actions

Helpful Links

Options for using OPA with IaC

1. **opa exec**
   https://www.openpolicyagent.org/docs/latest/cli/#opa-exec
2. **Conftest**
   https://www.conftest.dev/
3. **CloudFormation**
   https://www.openpolicyagent.org/docs/edge/aws-cloudformation-hooks/

Learn More

1. **Styra Academy**
   https://academy.styra.com/
2. **Open Policy Agent Docs**
   https://www.openpolicyagent.org/docs/latest/
3. **Styra DAS Free**
   https://signup.styra.com/
4. **This Demo**
   https://github.com/peteroneilljr/Conf42-and-open-policy-agent

Thank You