Securing the Cloud with Policy-as-Code and Predeployment tools

Atlanta Cloud Conf. 2023

$ whoami

```
$ groups
ace gcpfellow sada
$ alias
tf=terraform
tfa='terraform apply'
k=’kubectl’
$ conferences
defcon kubecon bsidesatl bsidesnash
```

```
$ hobbies
    coffee homelab
```

$ whoami

```
$ groups

ace gcpfellow sada

$ alias

tf=terraform

tfa='terraform apply'

k=’kubectl’

$ conferences

defcon kubecon bsidesatl bsidesnash
```

```
$ hobbies

    coffee homelab
```

Tools and companies shown in this presentation are not endorsements

$ man policy-as-code

- SEE ALSO
  - risk-and-compliance-as-code
  - predeployment checks
  - DevSecOps
- My definition:
  - Policy-as-Code is a security pattern that focuses on preventing misconfigurations from being introduced into an environment.
- What tools implement this definition?
  - HashiCorp Sentinel
  - Bridgecrew Checkov
  - Terraform Validator
  - Open Policy Agent
- What infrastructure can be evaluated by Policy-as-Code?
  - Terraform
  - Cloudformation templates
  - Azure Bicep templates
  - Kubernetes Manifests
  - Anything you want!

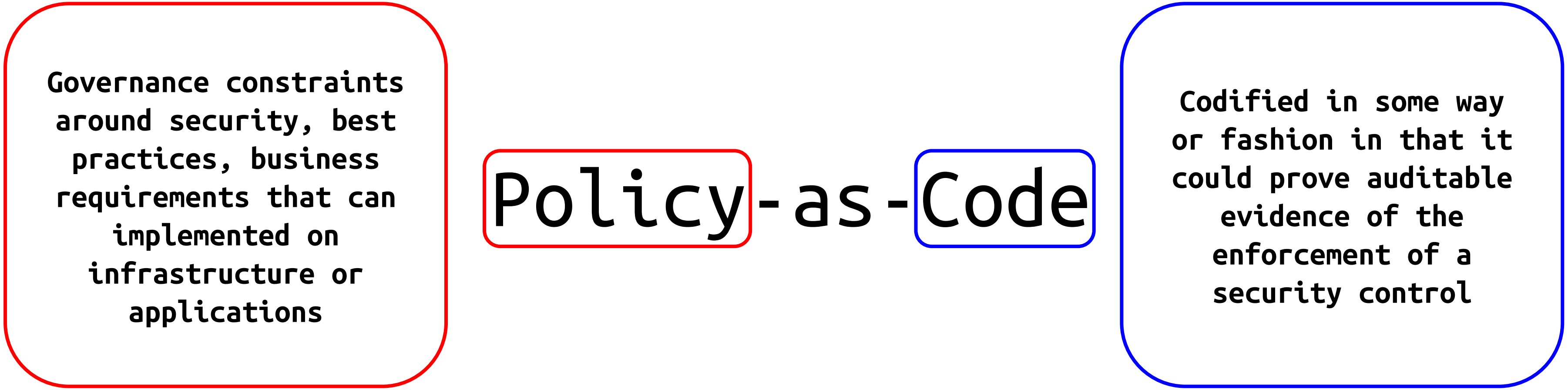
$ bard.google.com’s definition of Policy-as-Code

“Policy-as-code is a method of defining and managing security policies and compliance in a declarative way using code. It allows you to automate the process of defining and implementing policies, as well as the process of checking for compliance.

Policy-as-code is often used in conjunction with continuous integration/continuous delivery (CI/CD) pipelines. When you use policy-as-code, you can define your policies in a file and then use a tool to automatically apply those policies to your infrastructure. This can help you to ensure that your infrastructure is always compliant with your policies.

If you are looking for a way to improve the way you manage security policies and compliance, then policy-as-code is a great option.”

$ Policy-as-Code in a nutshell



Governance constraints around security, best practices, business requirements that can implemented on infrastructure or applications

Policy-as-Code

Codified in some way or fashion in that it could prove auditable evidence of the enforcement of a security control

$ Generic Policy-as-Code implementation

```
resource "azurerm_linux_virtual_machine" "atl_cloud_conf" {
  name                = "invalid-tf-dont-try-me"
  resource_group_name = azurerm_resource_group.example.name
  location            = azurerm_resource_group.example.location
  network_interface_ids = [
    azurerm_network_interface.example.id,
  ]
  source_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }
}
```

$ Generic Policy-as-Code implementation

```
resource "azurerm_linux_virtual_machine" "atl_cloud_conf" {
  name                = "invalid-tf-dont-try-me"
  resource_group_name = azurerm_resource_group.example.name
  location            = azurerm_resource_group.example.location
  network_interface_ids = [
    azurerm_network_interface.example.id,
  ]
  source_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }
}
```

1. Loop over all resources being created

$ Generic Policy-as-Code implementation

```
resource "azurerm_linux_virtual_machine" "atl_cloud_conf" {
  name                = "invalid-tf-dont-try-me"
  resource_group_name = azurerm_resource_group.example.name
  location            = azurerm_resource_group.example.location
  network_interface_ids = [
    azurerm_network_interface.example.id,
  ]
  source_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }
}
```

1. Loop over all resources being created
2. Find specific type of resource

$ Generic Policy-as-Code implementation

```
resource "azurerm_linux_virtual_machine" "atl_cloud_conf" {
  name                = "invalid-tf-dont-try-me"
  resource_group_name = azurerm_resource_group.example.name
  location            = azurerm_resource_group.example.location
  network_interface_ids = [
    azurerm_network_interface.example.id,
  ]
  source_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }
}
```

1. Loop over all resources being created
2. Find specific type of resource
3. Find the configurations we care about

$ Generic Policy-as-Code implementation

```
resource "azurerm_linux_virtual_machine" "atl_cloud_conf" {
  name                = "invalid-tf-dont-try-me"
  resource_group_name = azurerm_resource_group.example.name
  location            = azurerm_resource_group.example.location
  network_interface_ids = [
    azurerm_network_interface.example.id,
  ]
  source_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }
}
```

1. Loop over all resources being created
2. Find specific type of resource
3. Find the configurations we care about
4. Determine if that value is in compliance

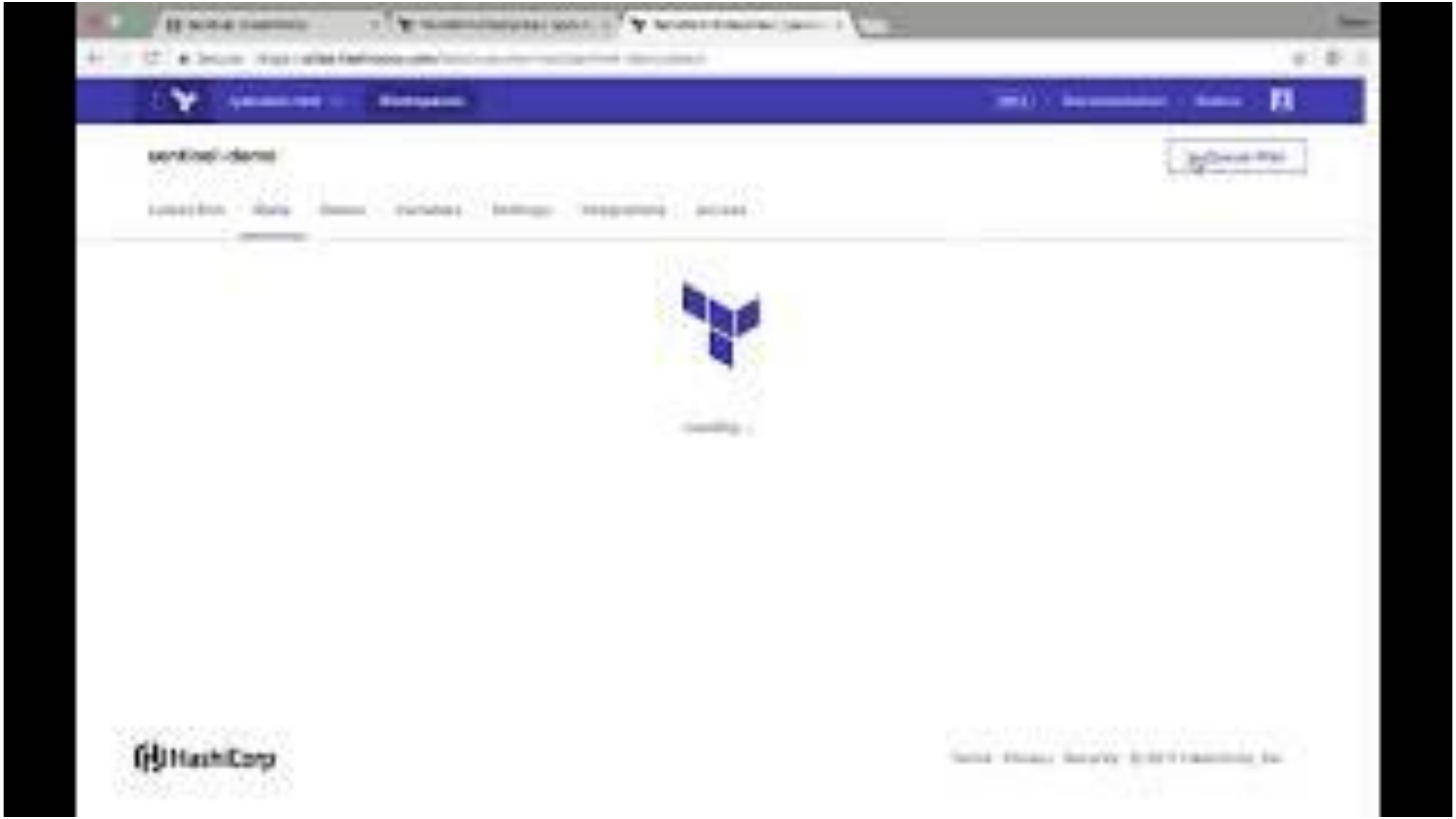
$ Generic Policy-as-Code implementation

```
resource "azurerm_linux_virtual_machine" "atl_cloud_conf" {
  name                = "invalid-tf-dont-try-me"
  resource_group_name = azurerm_resource_group.example.name
  location            = azurerm_resource_group.example.location
  network_interface_ids = [
    azurerm_network_interface.example.id,
  ]
  source_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }
}
```

1. Loop over all resources being created
2. Find specific type of resource
3. Find the configurations we care about
4. Determine if that value is in compliance
5. Exit code 1!
6. Glorified JSON parsing

[This deployment plan violates security control XYZ …]

Example Gifs!


Workspaces
sentinel-demo
Queue Plan
Latest Run
Runs
States
Variables
Settings
Integrations
Access
Loading...
HashiCorp

```
jacks-reid@atl-cloud-conf $ 
```

```
jacks-reid@atl-cloud-conf $
```

```
jacks-reid@atl-cloud-conf $ 
```

Why bother with Policy-as-Code?

$ A quick aside: Cloud Security Posture Management tools

- Cloud Security Posture Management tools are the primary tool for most cloud security analysts for scoping and understanding their cloud attack surface

- CSPMs are huge players within the cloud security ecosystem and augment, in some cases replace, the native cloud security capabilities of the various public cloud providers

- CSPMs work very similarly to PaC! They scan cloud asset metadata and tell you “what’s wrong”**

PRISMA® CLOUD
BY PALO ALTO NETWORKS

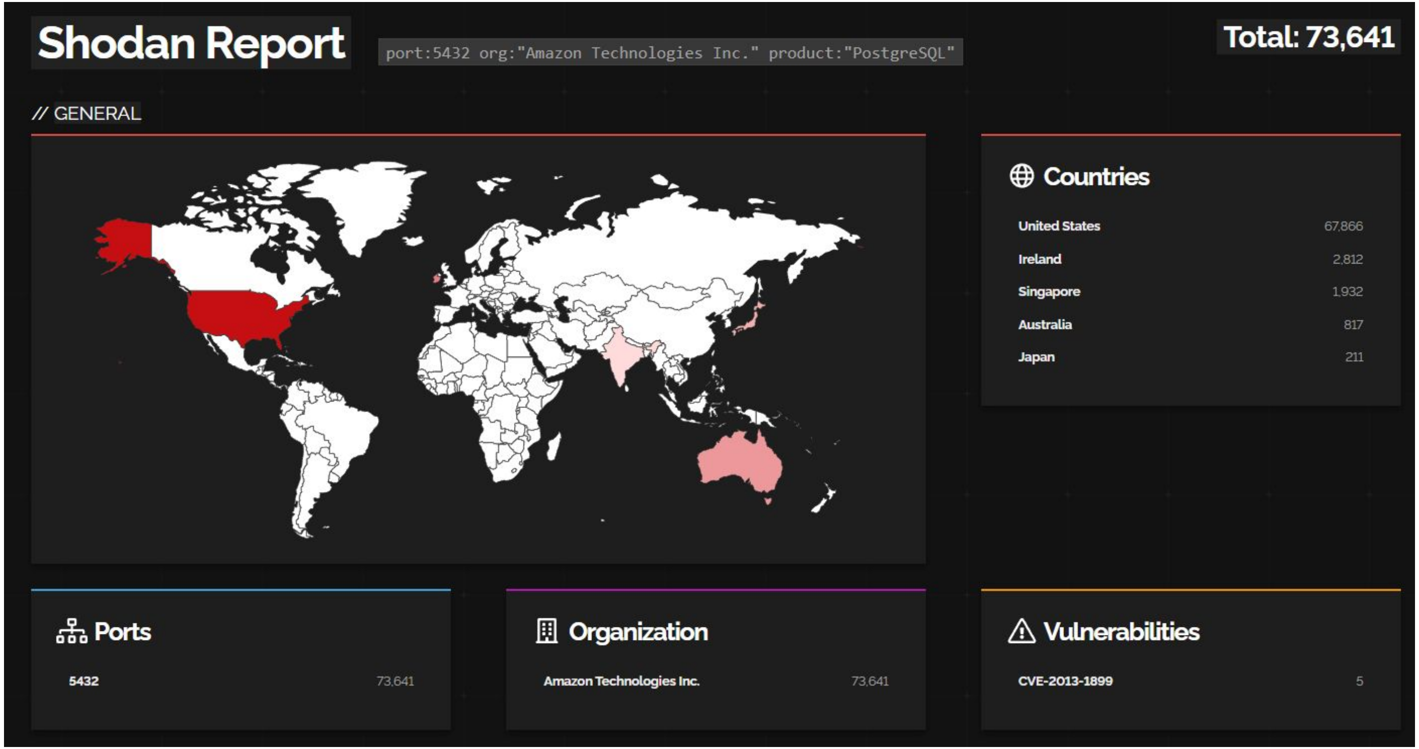
WIZ

orca security

$ ACME Corp’s Cloud Wonderland premise

- The majority of all application team leads have administrator access to most of our AWS accounts/Azure subscriptions/GCP projects.
- Our CSPM tool is constantly blowing up with alerts. The resident Cloud Security Operations Center doesn’t know what matters anymore. They tried taking down a public bucket but got burned once the application team proved that they are serving static site content and haven’t had the confidence to try again since.
- The organization is 24 months into our cloud journey. The environment is messy and sprawling, and the new acquisition on the horizon is expected to make things worse… and multi-cloud.
- Cleaning up a misconfiguration takes ages and new moles are popping up on our scanners faster than we can whack them.

$ Why do we need Policy-as-Code? (1/2)



Shodan Report

`port:5432 org:"Amazon Technologies Inc." product:"PostgreSQL"`

Total: 73,641

// GENERAL

Countries

United States	67,866
Ireland	2,812
Singapore	1,932
Australia	817
Japan	211

Ports

5432	73,641

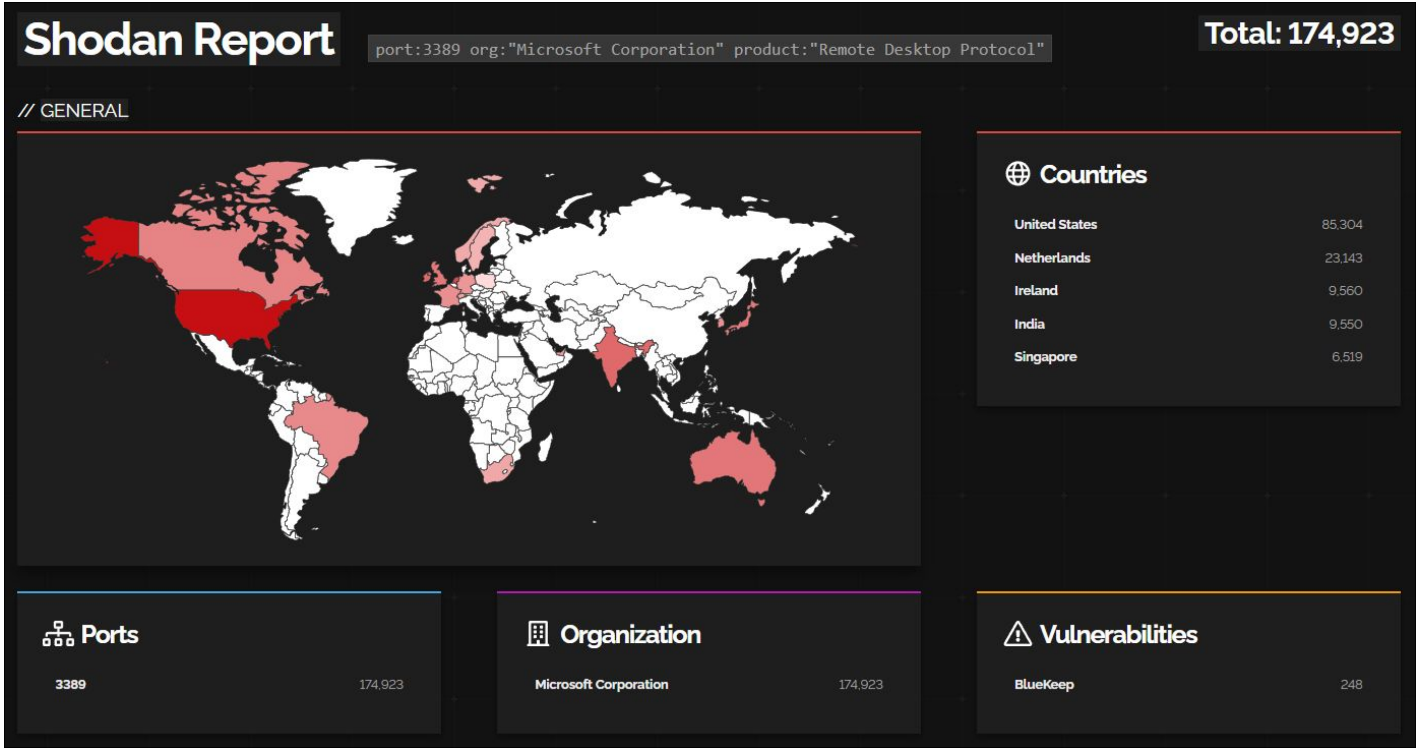
Organization

Amazon Technologies Inc.	73,641

Vulnerabilities

CVE-2013-1899	5

$ Why do we need Policy-as-Code? (2/2)



Shodan Report

`port:3389 org:"Microsoft Corporation" product:"Remote Desktop Protocol"`

Total: 174,923

// GENERAL

Countries

United States	85,304
Netherlands	23,143
Ireland	9,560
India	9,550
Singapore	6,519

Ports

3389	174,923

Organization

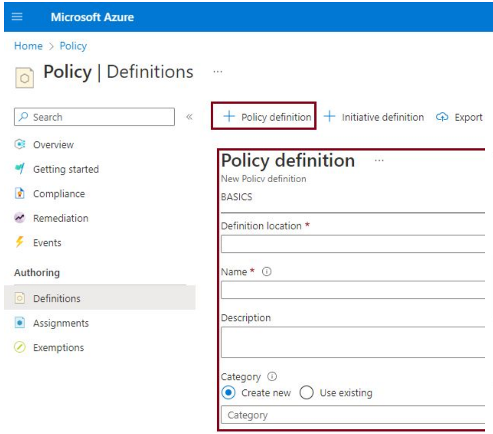
Microsoft Corporation	174,923

Vulnerabilities

BlueKeep	248

Different Policy-as-Code Implementations

$ PaC tooling ecosystem… and more!



Policy definition ···

New Policy definition

BASICS

Definition location *

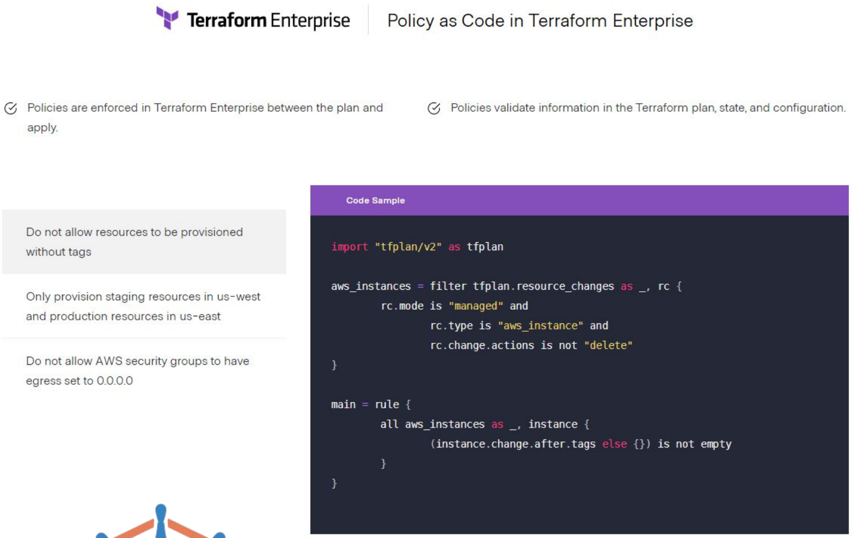
Name *

Description

Category

● Create new ○ Use existing

Category



Policies are enforced in Terraform Enterprise between the plan and apply.

Policies validate information in the Terraform plan, state, and configuration.

Do not allow resources to be provisioned without tags

Only provision staging resources in us-west and production resources in us-east

Do not allow AWS security groups to have egress set to 0.0.0.0

```
import "tfplan/v2" as tfplan

aws_instances = filter tfplan.resource_changes as _, rc {
        rc.mode is "managed" and
                rc.type is "aws_instance" and
                rc.change.actions is not "delete"
}

main = rule {
        all aws_instances as _, instance {
                (instance.change.after.tags else {}) is not empty
        }
}
```

Open Policy Agent

aws what's next

AWS CloudFormation Guard

Kyverno

checkov by bridgecrew

$ Infrastructure-as-Code Policy Languages

- HashiCorp Sentinel
- AWS Cloudformation Guard
- Open Policy Agent Rego languages
- Are these Policy-as-Code? Sure!

Policy languages are the “standard” Policy-as-Code implementation offering vast extensibility at the cost of learning a bespoke, and sometimes tricky, policy language

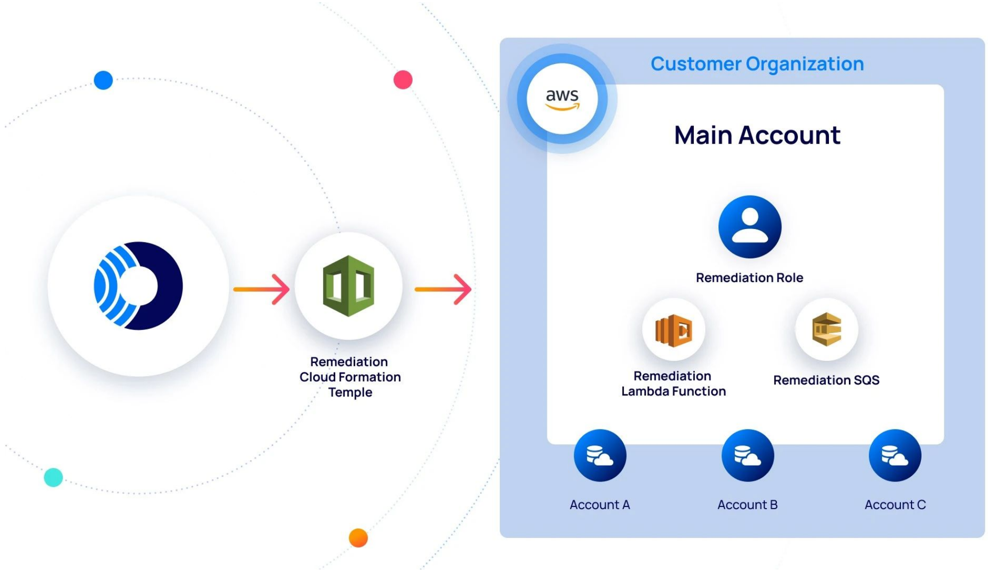
$ Native Cloud Guardrails

- Azure Policy
- Google Cloud Organization Policies
- AWS Service Control Policies
- Are these Policy-as-Code? Sure!

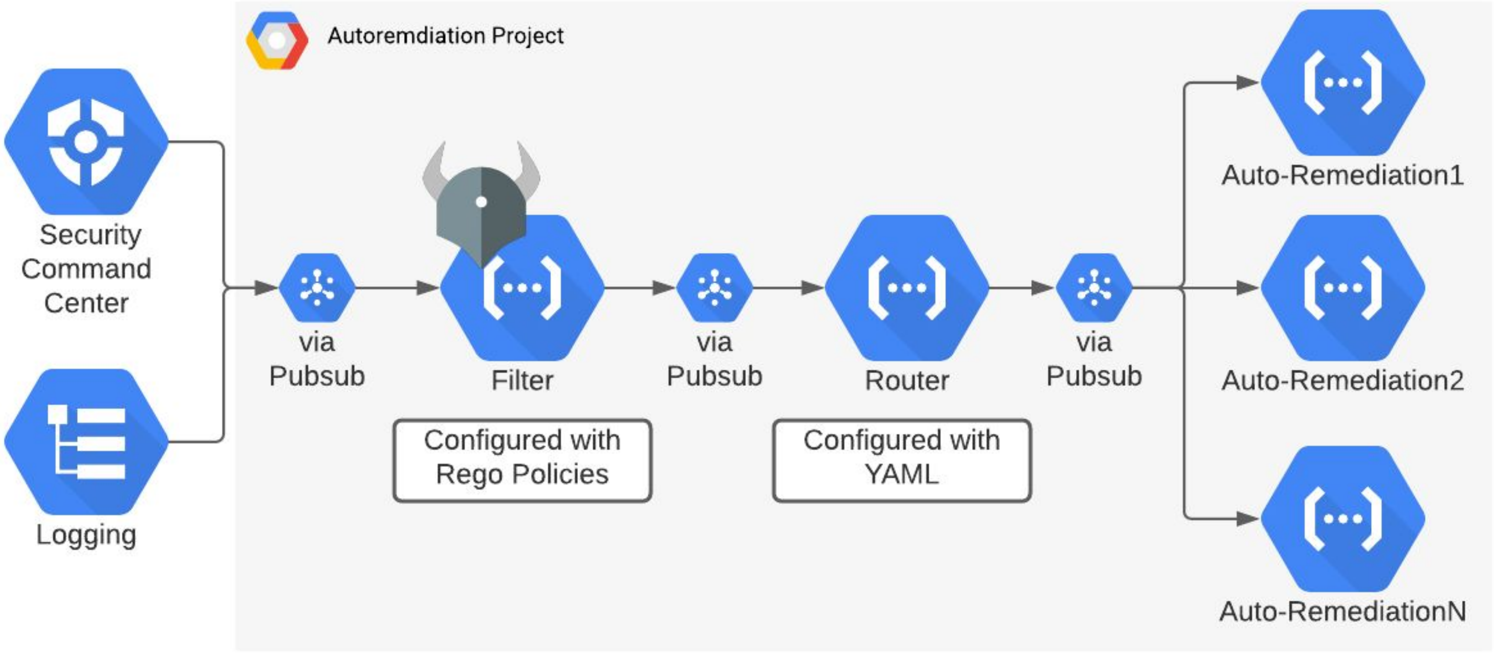
Cloud native mechanisms are fantastic guardrails, particularly for that specific cloud’s mechanisms. What they may lack in customizability they make up for in a managed, seamless experience.

$ Serverless based asset remediation streams (1/2)

- “Digital immune systems”
- Are these Policy-as-Code? Sure!



$ Serverless based asset remediation streams (2/2)

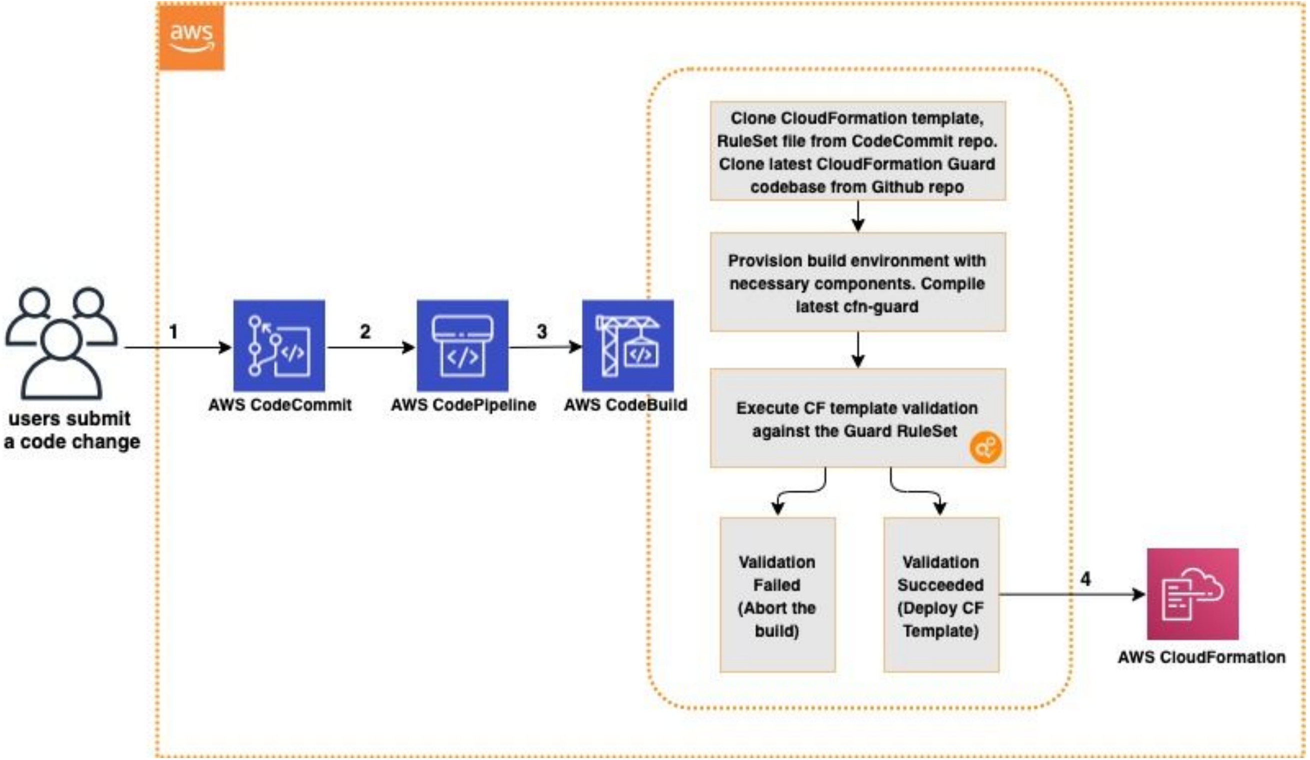


$ Pre-predeployment checks! (1/2)

- Many of tools we’ve talked about are “admission time controls”
- Right when a team is about to deploy is worst time to tell them that they are violating a security control
- What are you doing to embed security within regular development processes, even when a release isn’t being cut?
- Most tools have some form of …

```
if (deployment_problem) { print(‘feedback to try and fix your problem’) }
```

$ Pre-predeployment checks! (2/2)



Clone CloudFormation template, RuleSet file from CodeCommit repo. Clone latest CloudFormation Guard codebase from Github repo

Provision build environment with necessary components. Compile latest cfn-guard

Execute CF template validation against the Guard RuleSet

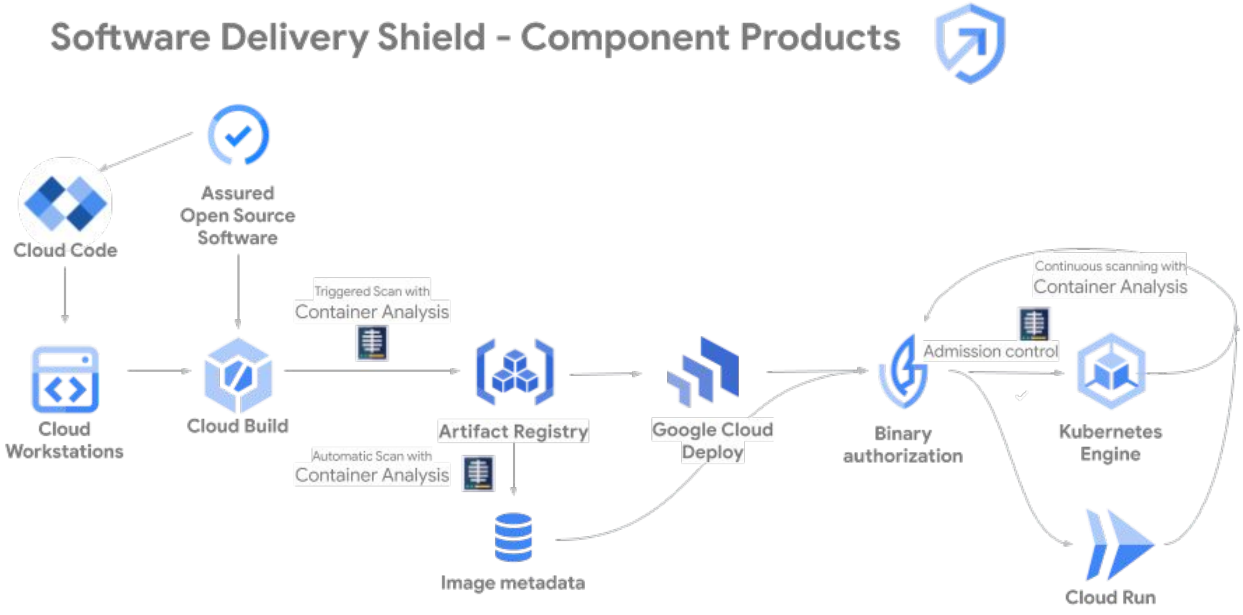
Validation Failed (Abort the build)

Validation Succeeded (Deploy CF Template)

$ Application based Policy-as-Code (1/2)

- The looming compliance requirements above our industry

Software Delivery Shield - Component Products

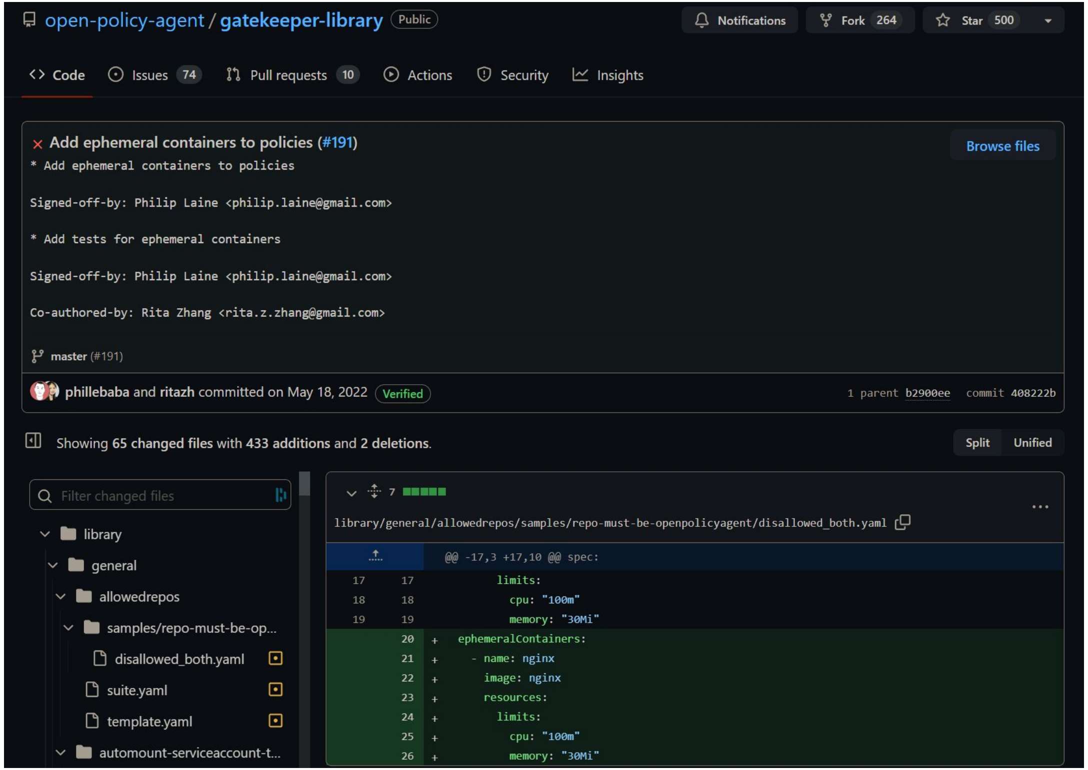


$ Application based Policy-as-Code (2/2)

“A call for cybersecurity liability—holding software providers responsible”

“This is historic – the U.S. government has never taken a stance on issues of product liability with regard to software. The intent is to remove the ability for organizations to disclaim all liability in an End-User License Agreement (EULA) that consumers have no choice but to accept. The Strategy demands we replace this long-held policy with one that puts the onus on the company producing the software. It enables those companies to earn back liability protections in the form of safe harbors, consisting of well-defined best practices and standards that will likely mature over time.” - Brian Fox, CTO Sonatype

$ Policy-as-Code is hard! (1/2)



Add ephemeral containers to policies (#191)

```
* Add ephemeral containers to policies

Signed-off-by: Philip Laine <philip.laine@gmail.com>

* Add tests for ephemeral containers

Signed-off-by: Philip Laine <philip.laine@gmail.com>

Co-authored-by: Rita Zhang <rita.z.zhang@gmail.com>
```

master (#191)

phillebaba and ritazh committed on May 18, 2022 Verified

1 parent b2900ee commit 408222b

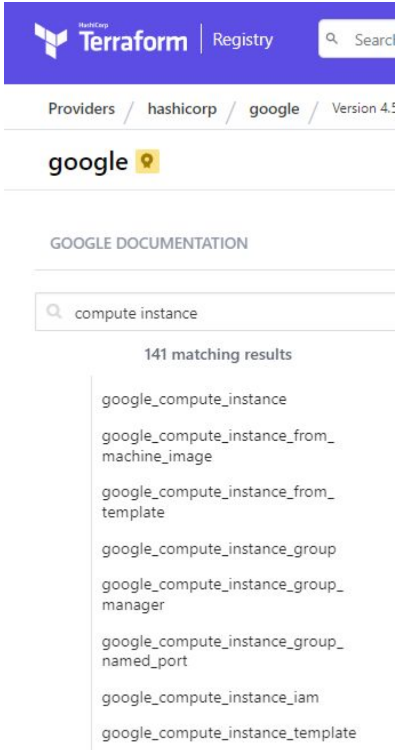
Showing 65 changed files with 433 additions and 2 deletions.

7

library/general/allowedrepos/samples/repo-must-be-openpolicyagent/disallowed_both.yaml

```
@@ -17,3 +17,10 @@ spec:
          limits:
            cpu: "100m"
            memory: "30Mi"
+   ephemeralContainers:
+     - name: nginx
+       image: nginx
+       resources:
+         limits:
+           cpu: "100m"
+           memory: "30Mi"
```

$ Policy-as-Code is hard! (2/2)

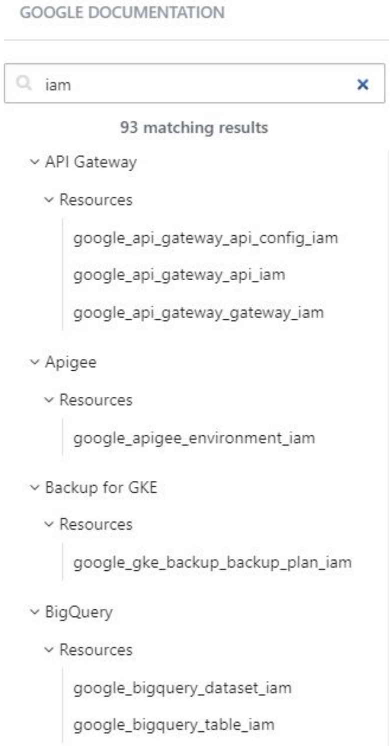


GOOGLE DOCUMENTATION

compute instance

141 matching results

- google_compute_instance
- google_compute_instance_from_machine_image
- google_compute_instance_from_template
- google_compute_instance_group
- google_compute_instance_group_manager
- google_compute_instance_group_named_port
- google_compute_instance_iam
- google_compute_instance_template

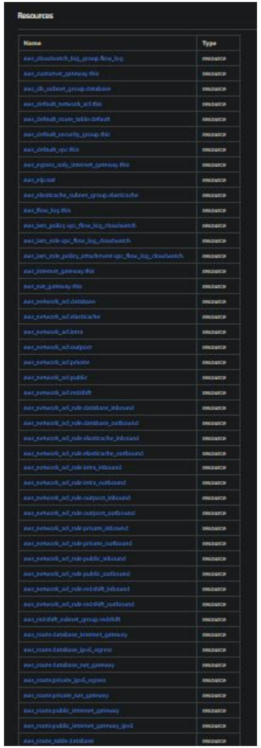


GOOGLE DOCUMENTATION

iam

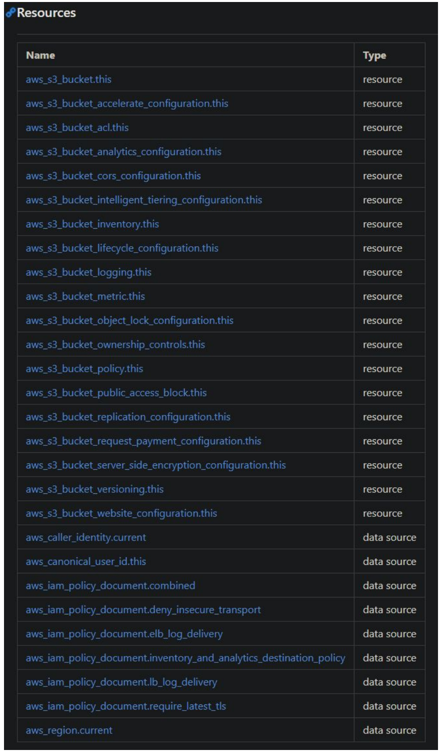
93 matching results

- API Gateway
  - Resources
    - google_api_gateway_api_config_iam
    - google_api_gateway_api_iam
    - google_api_gateway_gateway_iam
- Apigee
  - Resources
    - google_apigee_environment_iam
- Backup for GKE
  - Resources
    - google_gke_backup_backup_plan_iam
- BigQuery
  - Resources
    - google_bigquery_dataset_iam
    - google_bigquery_table_iam



Resources

Name	Type
aws_cloudwatch_log_group.flow_log	resource
aws_customer_gateway.this	resource
aws_db_subnet_group.database	resource
aws_default_network_acl.this	resource
aws_default_route_table.default	resource
aws_default_security_group.this	resource
aws_default_vpc.this	resource
aws_egress_only_internet_gateway.this	resource
aws_eip.nat	resource
aws_elasticache_subnet_group.elasticache	resource
aws_flow_log.this	resource
aws_iam_policy.vpc_flow_log_cloudwatch	resource
aws_iam_role.vpc_flow_log_cloudwatch	resource
aws_iam_role_policy_attachment.vpc_flow_log_cloudwatch	resource
aws_internet_gateway.this	resource
aws_nat_gateway.this	resource
aws_network_acl.database	resource
aws_network_acl.elasticache	resource
aws_network_acl.intra	resource
aws_network_acl.outpost	resource
aws_network_acl.private	resource
aws_network_acl.public	resource
aws_network_acl.redshift	resource
aws_network_acl_rule.database_inbound	resource
aws_network_acl_rule.database_outbound	resource
aws_network_acl_rule.elasticache_inbound	resource
aws_network_acl_rule.elasticache_outbound	resource
aws_network_acl_rule.intra_inbound	resource
aws_network_acl_rule.intra_outbound	resource
aws_network_acl_rule.outpost_inbound	resource
aws_network_acl_rule.outpost_outbound	resource
aws_network_acl_rule.private_inbound	resource
aws_network_acl_rule.private_outbound	resource
aws_network_acl_rule.public_inbound	resource
aws_network_acl_rule.public_outbound	resource
aws_network_acl_rule.redshift_inbound	resource
aws_network_acl_rule.redshift_outbound	resource
aws_redshift_subnet_group.redshift	resource
aws_route.database_internet_gateway	resource
aws_route.database_ipv6_egress	resource
aws_route.database_nat_gateway	resource
aws_route.private_ipv6_egress	resource
aws_route.private_nat_gateway	resource
aws_route.public_internet_gateway	resource
aws_route.public_internet_gateway_ipv6	resource
aws_route_table.database	resource



Resources

Name	Type
aws_s3_bucket.this	resource
aws_s3_bucket_accelerate_configuration.this	resource
aws_s3_bucket_acl.this	resource
aws_s3_bucket_analytics_configuration.this	resource
aws_s3_bucket_cors_configuration.this	resource
aws_s3_bucket_intelligent_tiering_configuration.this	resource
aws_s3_bucket_inventory.this	resource
aws_s3_bucket_lifecycle_configuration.this	resource
aws_s3_bucket_logging.this	resource
aws_s3_bucket_metric.this	resource
aws_s3_bucket_object_lock_configuration.this	resource
aws_s3_bucket_ownership_controls.this	resource
aws_s3_bucket_policy.this	resource
aws_s3_bucket_public_access_block.this	resource
aws_s3_bucket_replication_configuration.this	resource
aws_s3_bucket_request_payment_configuration.this	resource
aws_s3_bucket_server_side_encryption_configuration.this	resource
aws_s3_bucket_versioning.this	resource
aws_s3_bucket_website_configuration.this	resource
aws_caller_identity.current	data source
aws_canonical_user_id.this	data source
aws_iam_policy_document.combined	data source
aws_iam_policy_document.deny_insecure_transport	data source
aws_iam_policy_document.elb_log_delivery	data source
aws_iam_policy_document.inventory_and_analytics_destination_policy	data source
aws_iam_policy_document.lb_log_delivery	data source
aws_iam_policy_document.require_latest_tls	data source
aws_region.current	data source

How do I launch a policy program?

$ Convincing your boss to launch a PaC program

- ***What you are REALLY asking for…***
  - “I want to slow down deployment processes for application teams”
  - “Our already strained team should take on more responsibility”
  - “We will need to become experts in cloud infrastructure and this random policy language”
  - “Potential this could get big enough to require a full team time employee”
- ***Why should we do this?***
  - Bring numbers, how long does it take your team to remove a misconfigured resource today? How will using PaC help the team from a $$$ perspective?
  - Advocate for platform engineering - “this is the sort of steps we need to take to legitimize our need for a dedicated platform team”
  - What is our rate of misconfiguration propagations in the environment? This is where you bring in the line graph of doom based on current adoption and misconfiguration rates

$ Your boss agrees!

$ Now what? (1/2)

- ***Acknowledge that your job will be harder*** before it gets easier
- ***Set expectations with application teams*** - when will the gravy stop flowing?
- ***Pick your battles.*** What are your non-negotiable policies? What policies would you feel confident “causing a problem for”? Public databases, public instances, public buckets, etc.
- ***Write your policy. Test your policy. Test your policy again.*** Test in non-production. Put it into dry-run mode. Start testing it with a specific application team. Test it with… *snipped*

$ Now what? (2/2)

- ***When something goes wrong, how can teams reach you?*** You might not be as popular or well-known as we like to think. platform-predeployment@acme.org
- ***Provide examples. You are going to aggravate folks.*** You aren’t responsible for writing their code, but they’d sure like you better if you had hardened, preapproved templates for them to use.
- ***Do you have an on-call team?*** If so, this should probably be a part of their responsibilities.
- ***You are going to run into plenty of exceptions.*** Are you ready to handle that flood? Do you have a process for documenting them? Temporary exceptions?

$ Revisiting ACME Corp’s Cloud Wonderland w/ PaC

- The majority of all application team leads have administrator access to most of our AWS accounts/Azure subscriptions/GCP projects. They still have that admin access, but serverless based remediation streams keep them from goofing up.

- Our CSPM tool is slowly trending down on total number of misconfigurations. The Cloud Security Operations Center is empowered by the policy program and has a wealth of cloud specific knowledge encapsulated in policy documentation.

- The organization acquisition was successful. The culture of predeployment security built internally is now assimilating to the new team members.

- Cleaning up a misconfiguration still takes ages, but there are less moles to whack, and each manually whacking is quickly codified as a policy in our pipelines.

$ In summary…

- Short term pain for long term gain
- Culture eats strategy for breakfast
- Developer feedback, developer feedback
- Start small, dream big
- Any Policy-as-Code tool is better than nothing

Thank you!

https://linkedin.com/in/jackson-reid