ADDO

ALL DAY DEVOPS

NOVEMBER 12, 2020

Barak Schoster @bridgecrew

Embedding security in your Terraform and Cloudformation code

bridgecrew

Agenda

- Talk about the process of resolving misconfigs
- View the state of misconfigs in open source IaC modules
- Write Policy as Code
- Automate in our CI Pipeline
- Runtime analysis of cloud infrastcture

That’s me

Barak

@BarakSchoster
Schosterbarak

Slides at:
bridgecrew.io/blog/

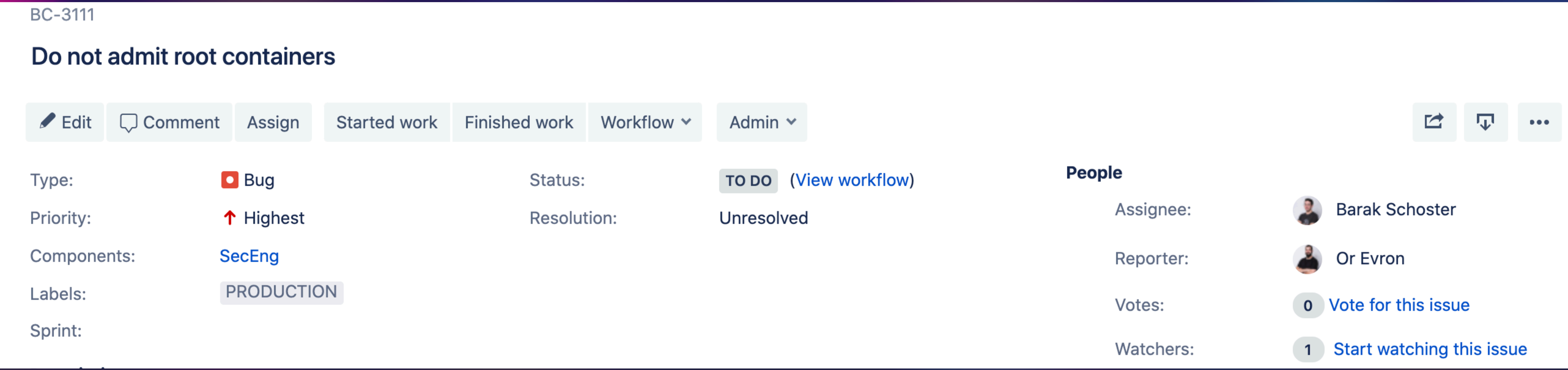
As an engineer

I want to move fast

I DO NOT want to break things

And the thing I have
love & hate relationship
with is…

Jira



BC-3111

Do not admit root containers

Type: Bug

Priority: Highest

Components: SecEng

Labels: PRODUCTION

Sprint:

Status: TO DO (View workflow)

Resolution: Unresolved

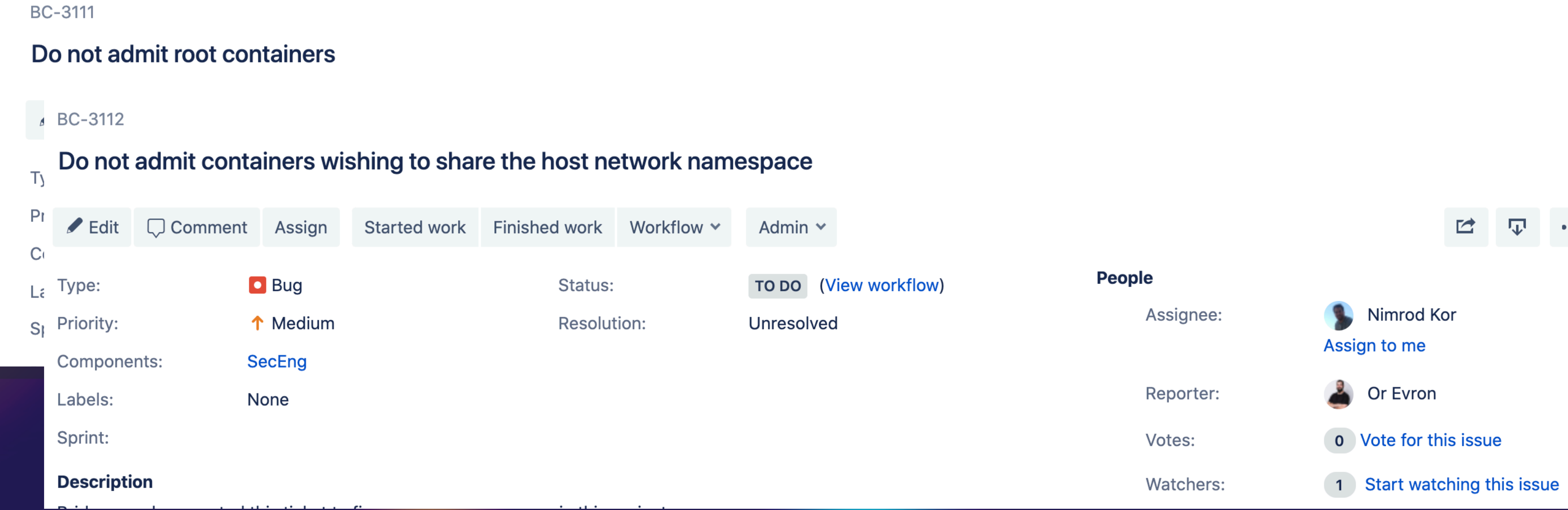
People

Assignee: Barak Schoster

Reporter: Or Evron

Votes: 0 Vote for this issue

Watchers: 1 Start watching this issue



BC-3111

Do not admit root containers

BC-3112

Do not admit containers wishing to share the host network namespace

Type: Bug

Priority: Medium

Components: SecEng

Labels: None

Sprint:

Status: TO DO (View workflow)

Resolution: Unresolved

People

Assignee: Nimrod Kor

Assign to me

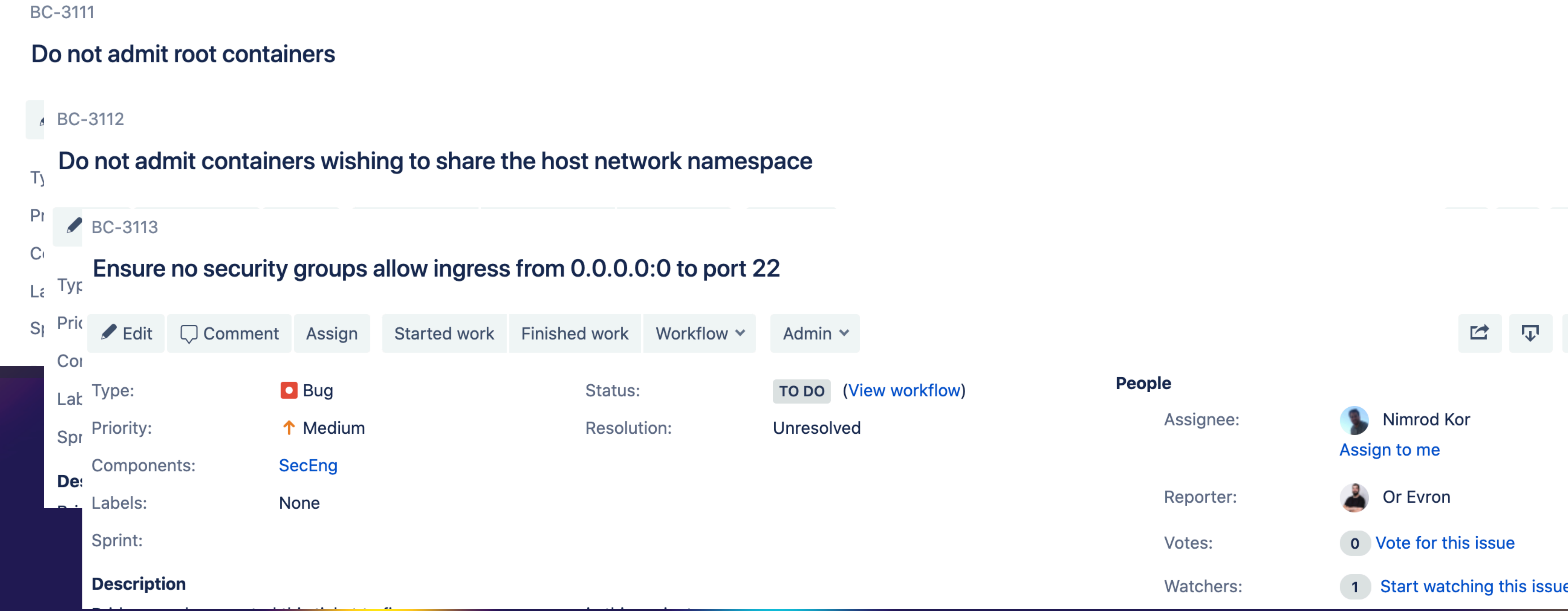
Reporter: Or Evron

Votes: 0 Vote for this issue

Watchers: 1 Start watching this issue

Description

bridgecrew



BC-3111

Do not admit root containers

BC-3112

Do not admit containers wishing to share the host network namespace

BC-3113

Ensure no security groups allow ingress from 0.0.0.0:0 to port 22

Type: Bug

Priority: Medium

Components: SecEng

Labels: None

Sprint:

Status: TO DO (View workflow)

Resolution: Unresolved

People

Assignee: Nimrod Kor

Assign to me

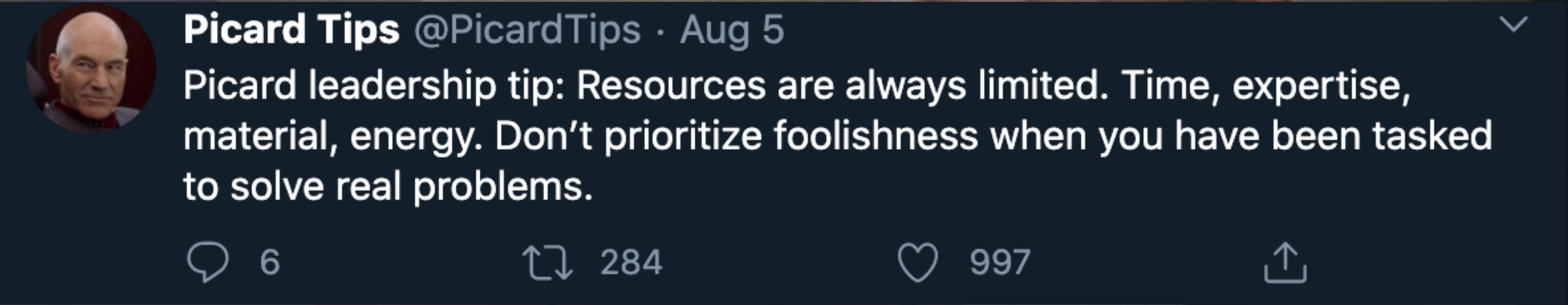
Reporter: Or Evron

Votes: 0 Vote for this issue

Watchers: 1 Start watching this issue

Description

And this is where our story begins…



Picard Tips @PicardTips · Aug 5

Picard leadership tip: Resources are always limited. Time, expertise, material, energy. Don’t prioritize foolishness when you have been tasked to solve real problems.

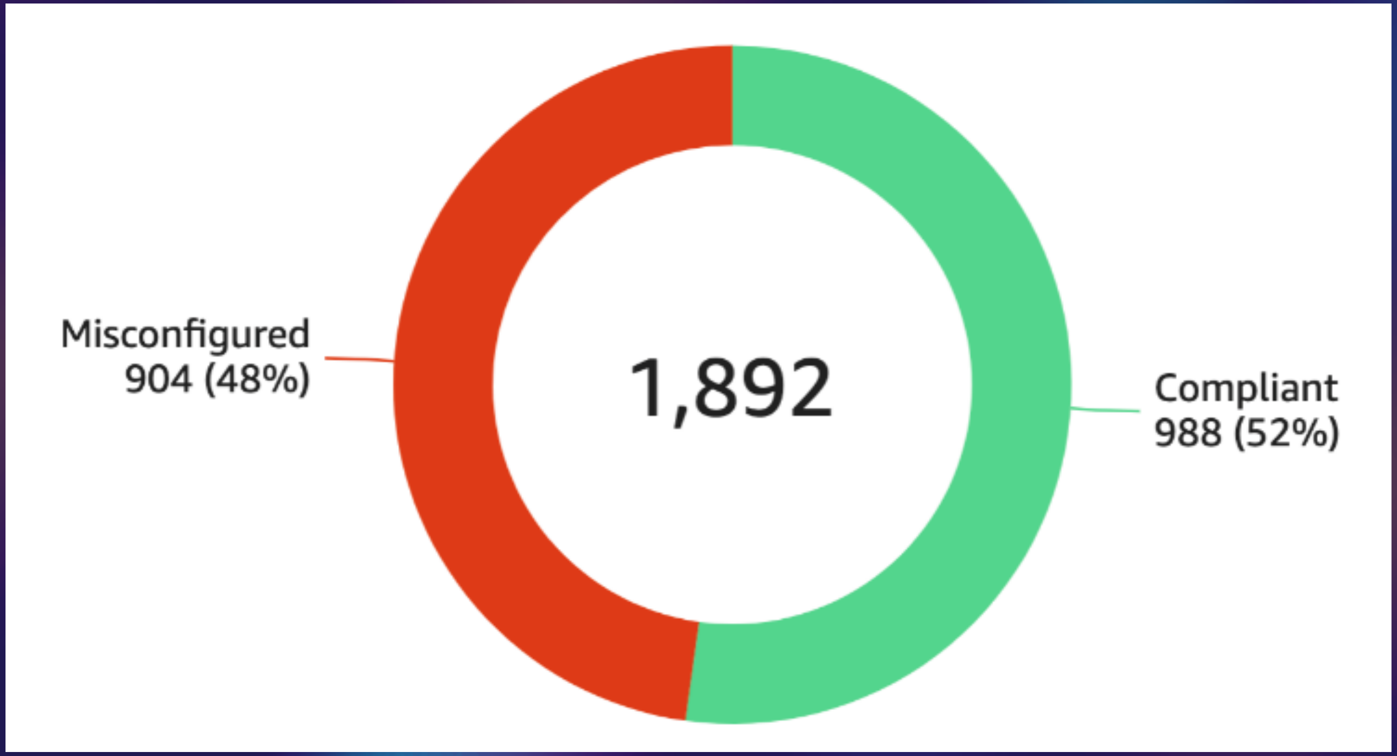
So let’s open our eyes
and look at some…

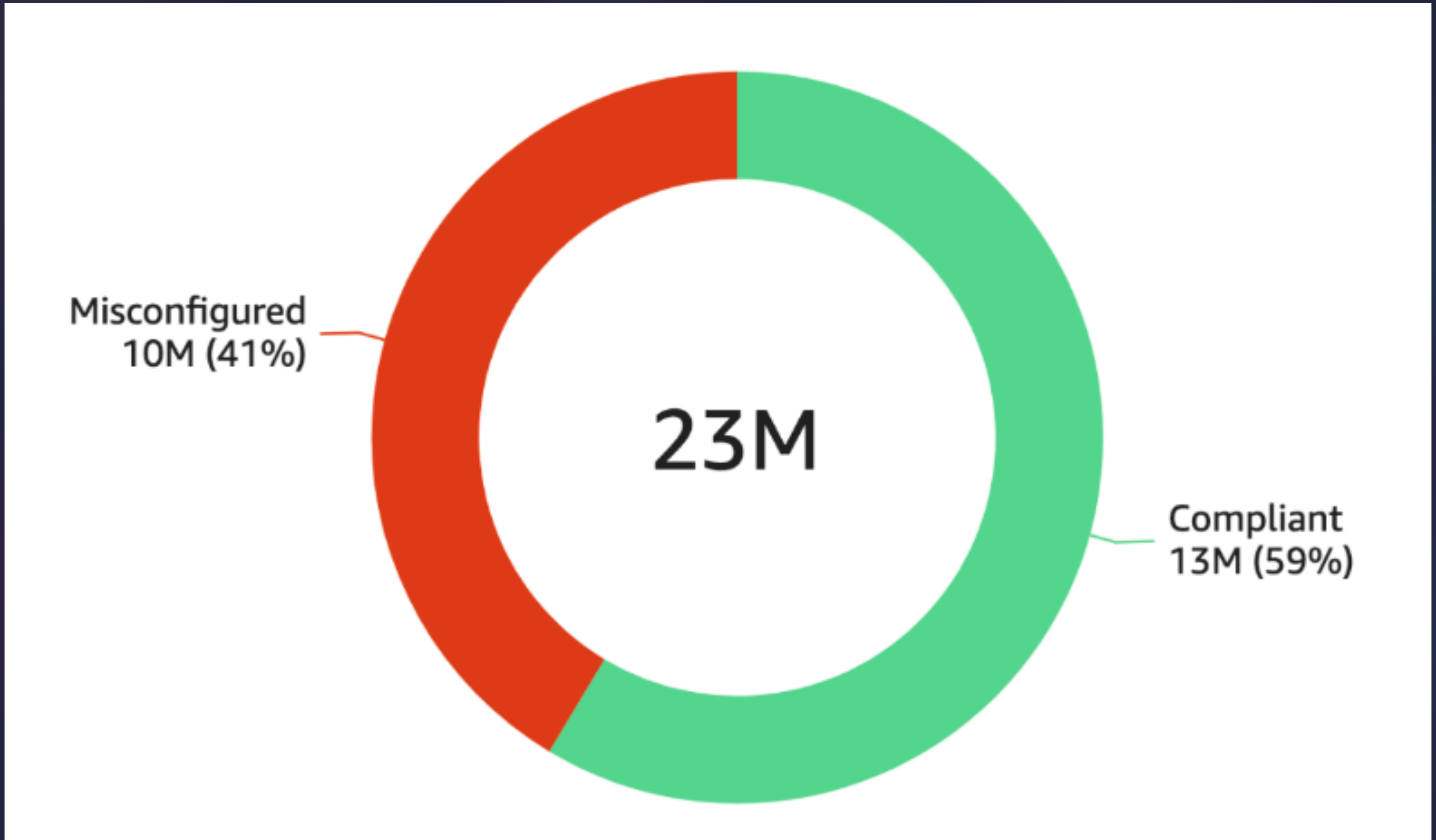
...data

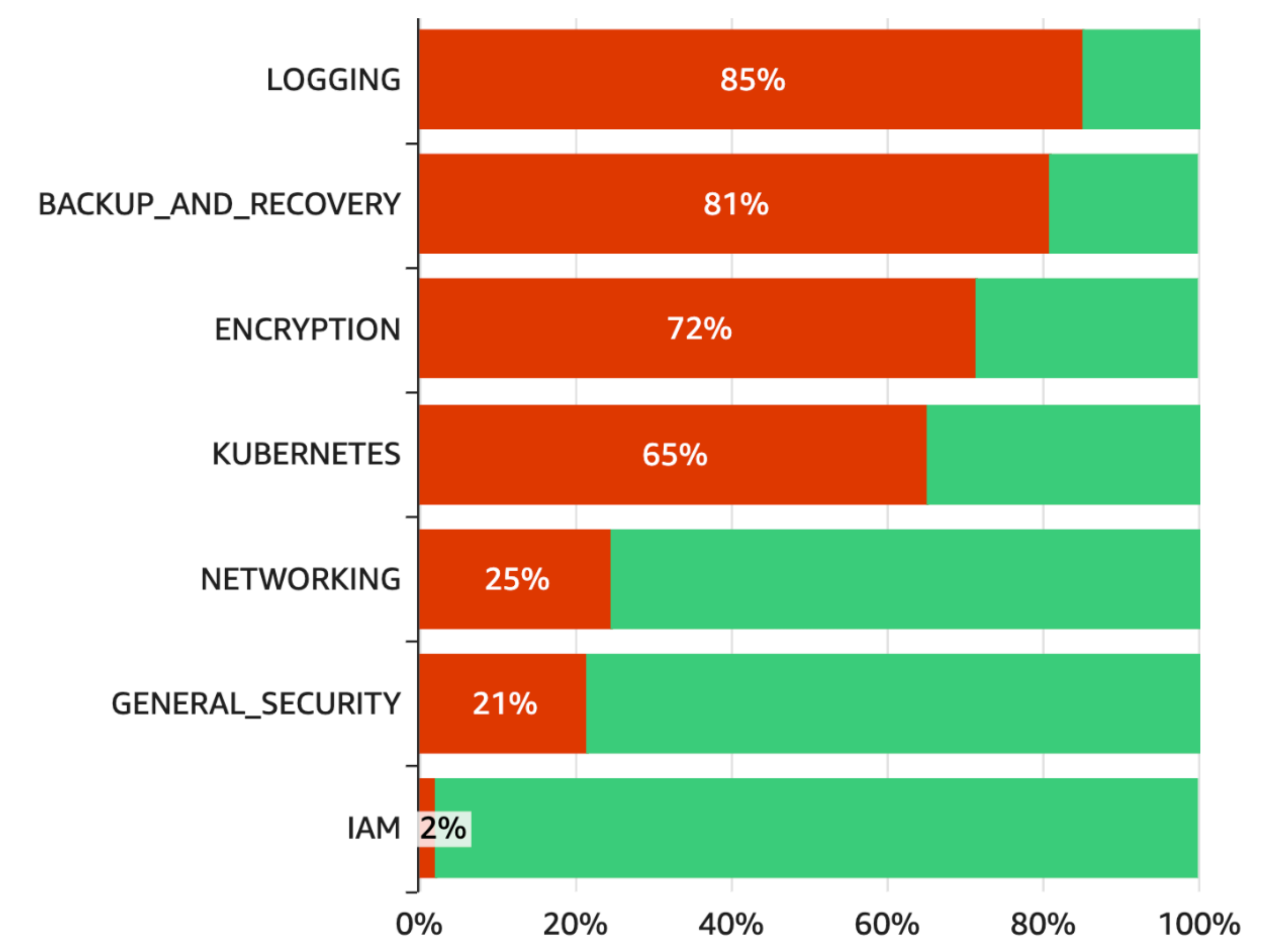
bridgecrew

15,749 IAC FILES

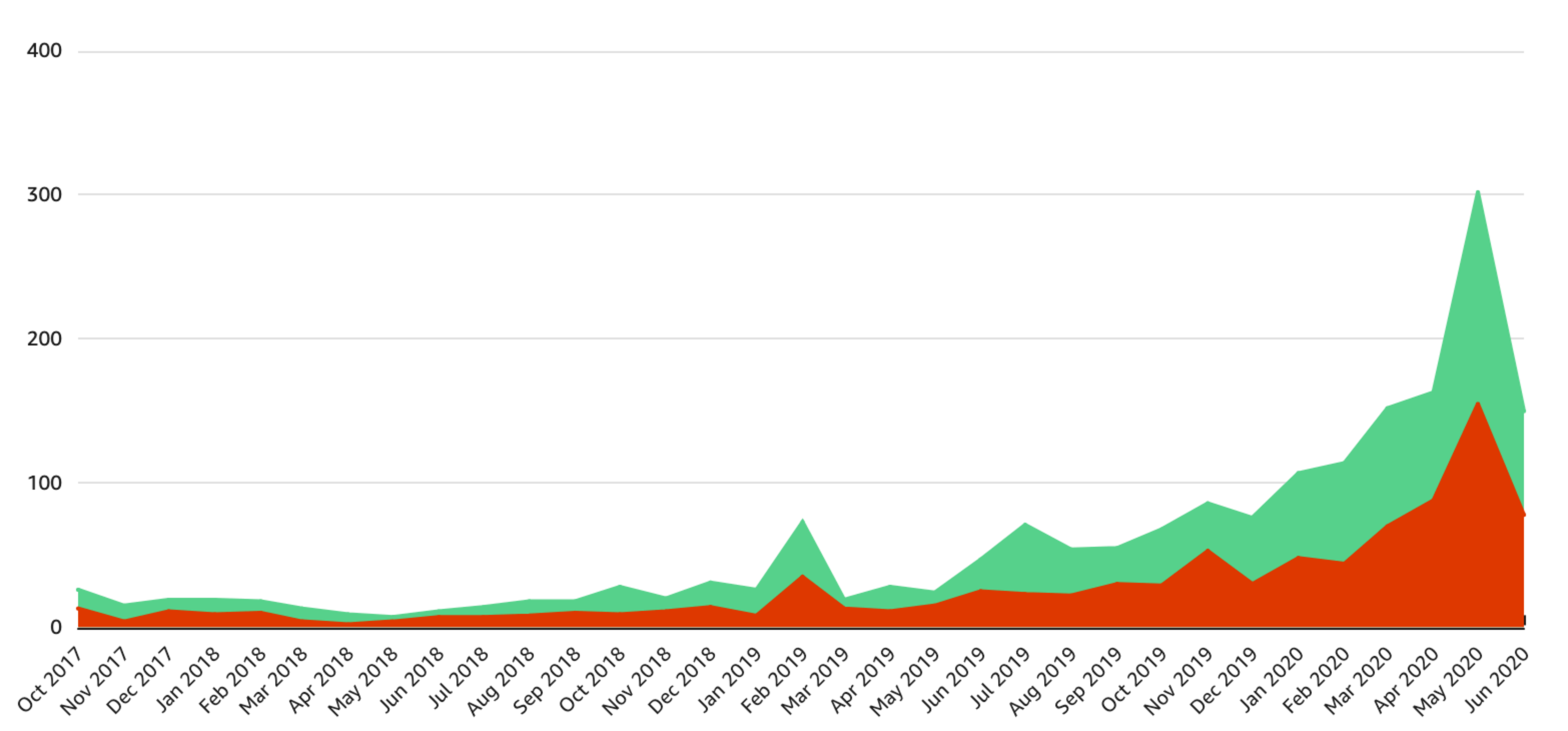
1892 IAC AWS MODULES




Misconfigured
10M (41%)
23M
Compliant
13M (59%)

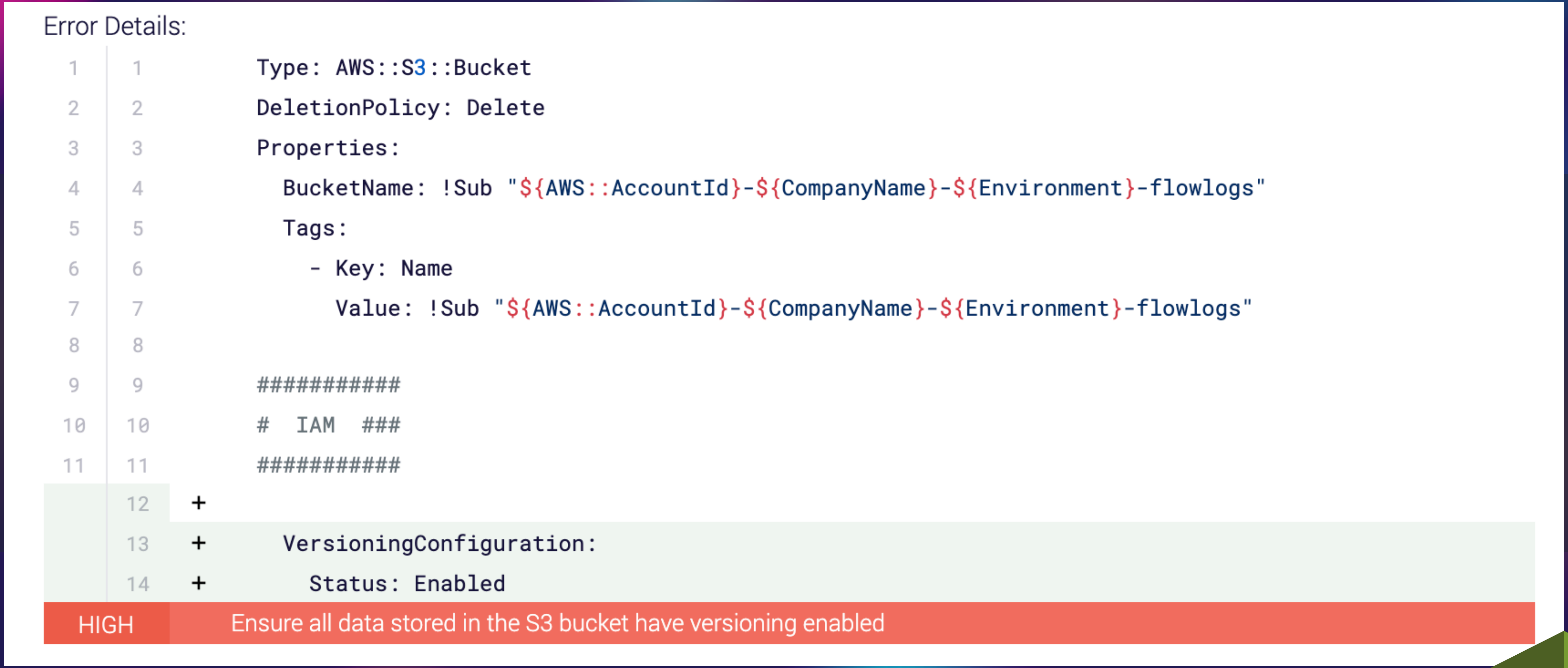

LOGGING
85%
BACKUP_AND_RECOVERY
81%
ENCRYPTION
72%
KUBERNETES
65%
NETWORKING
25%
GENERAL_SECURITY
21%
IAM
2%
0%
20%
40%
60%
80%
100%

bridgecrew



400
300
200
100
0
Oct 2017
Nov 2017
Dec 2017
Jan 2018
Feb 2018
Mar 2018
Apr 2018
May 2018
Jun 2018
Jul 2018
Aug 2018
Sep 2018
Oct 2018
Nov 2018
Dec 2018
Jan 2019
Feb 2019
Mar 2019
Apr 2019
May 2019
Jun 2019
Jul 2019
Aug 2019
Sep 2019
Oct 2019
Nov 2019
Dec 2019
Jan 2020
Feb 2020
Mar 2020
Apr 2020
May 2020
Jun 2020

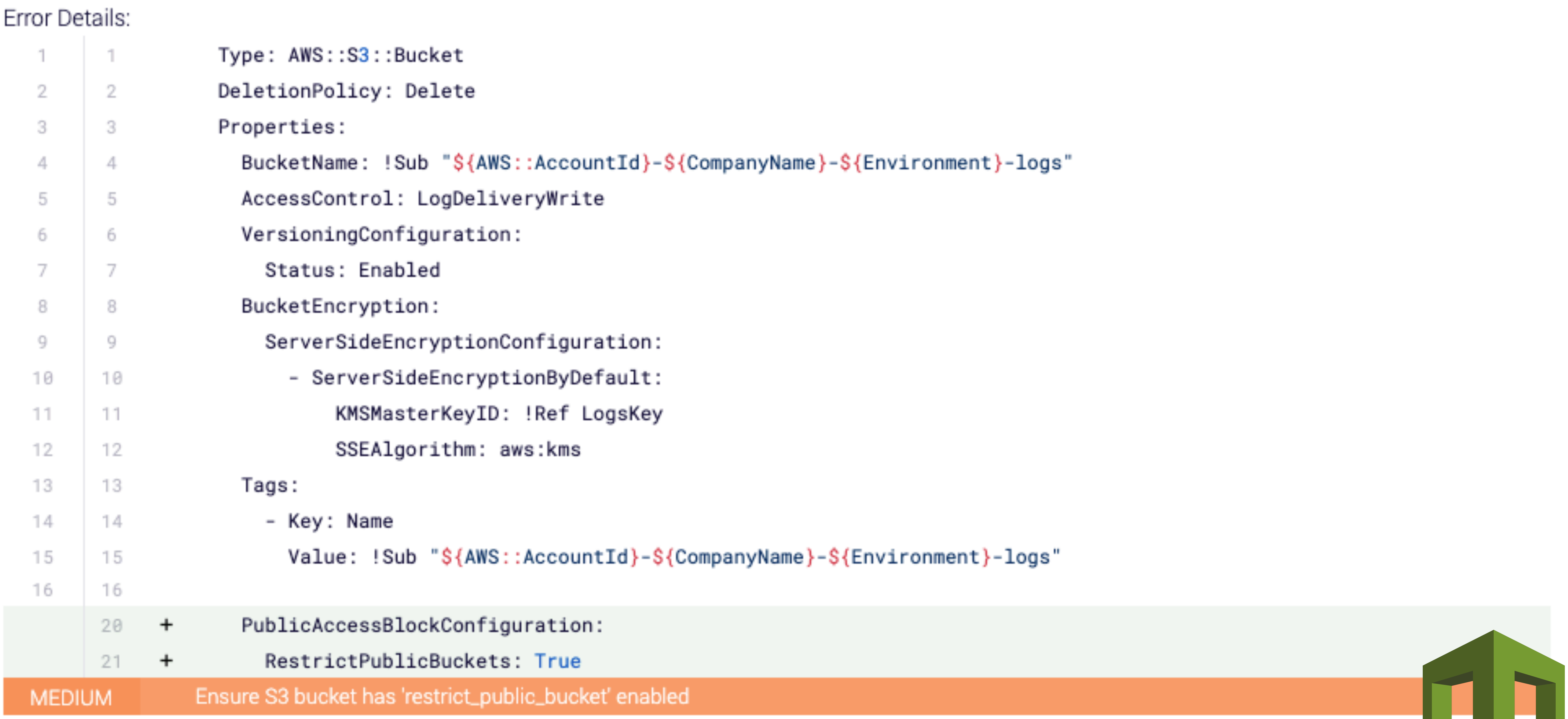
Infrastructure as code (IaC) presents
a new **risk** and a new **opportunity**



Error Details:

```
    Type: AWS::S3::Bucket
    DeletionPolicy: Delete
    Properties:
      BucketName: !Sub "${AWS::AccountId}-${CompanyName}-${Environment}-flowlogs"
      Tags:
        - Key: Name
          Value: !Sub "${AWS::AccountId}-${CompanyName}-${Environment}-flowlogs"

    ###########
    #  IAM  ###
    ###########
+
+     VersioningConfiguration:
+       Status: Enabled
```

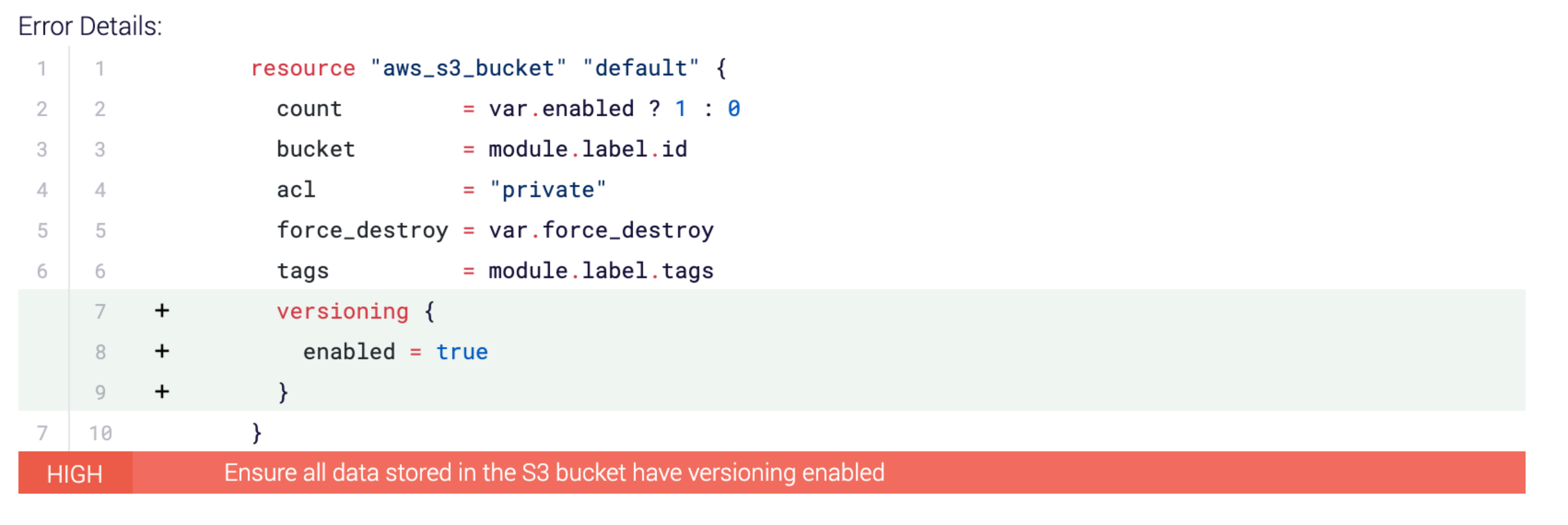
HIGH   Ensure all data stored in the S3 bucket have versioning enabled



Error Details:

```
    Type: AWS::S3::Bucket
    DeletionPolicy: Delete
    Properties:
      BucketName: !Sub "${AWS::AccountId}-${CompanyName}-${Environment}-logs"
      AccessControl: LogDeliveryWrite
      VersioningConfiguration:
        Status: Enabled
      BucketEncryption:
        ServerSideEncryptionConfiguration:
          - ServerSideEncryptionByDefault:
              KMSMasterKeyID: !Ref LogsKey
              SSEAlgorithm: aws:kms
      Tags:
        - Key: Name
          Value: !Sub "${AWS::AccountId}-${CompanyName}-${Environment}-logs"

+     PublicAccessBlockConfiguration:
+       RestrictPublicBuckets: True
```

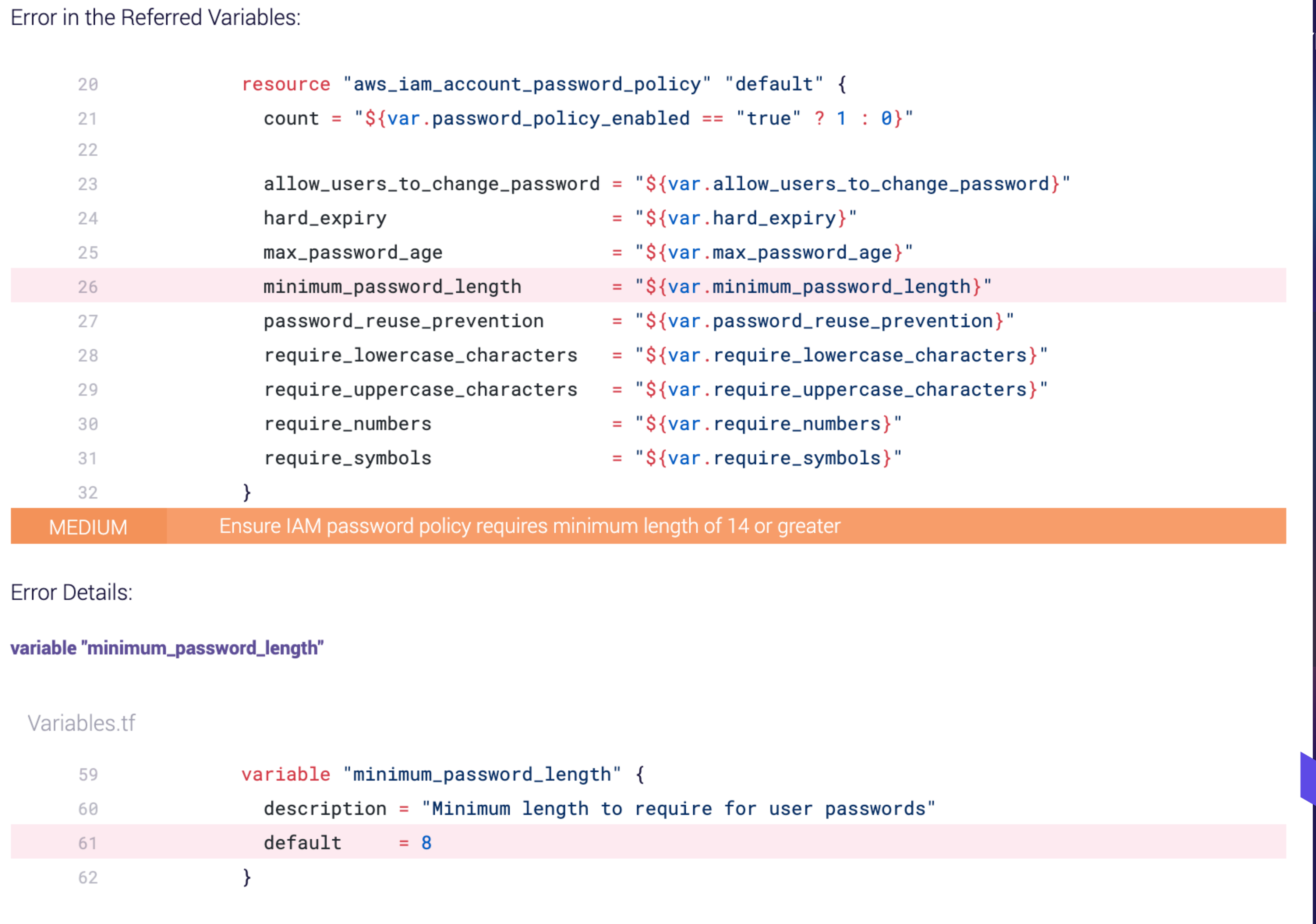
MEDIUM   Ensure S3 bucket has 'restrict_public_bucket' enabled



Error Details:

```
      resource "aws_s3_bucket" "default" {
        count         = var.enabled ? 1 : 0
        bucket        = module.label.id
        acl           = "private"
        force_destroy = var.force_destroy
        tags          = module.label.tags
+       versioning {
+         enabled = true
+       }
      }
```

HIGH Ensure all data stored in the S3 bucket have versioning enabled



Error in the Referred Variables:

```
resource "aws_iam_account_password_policy" "default" {
  count = "${var.password_policy_enabled == "true" ? 1 : 0}"

  allow_users_to_change_password = "${var.allow_users_to_change_password}"
  hard_expiry                    = "${var.hard_expiry}"
  max_password_age               = "${var.max_password_age}"
  minimum_password_length        = "${var.minimum_password_length}"
  password_reuse_prevention      = "${var.password_reuse_prevention}"
  require_lowercase_characters   = "${var.require_lowercase_characters}"
  require_uppercase_characters   = "${var.require_uppercase_characters}"
  require_numbers                = "${var.require_numbers}"
  require_symbols                = "${var.require_symbols}"
}
```

MEDIUM Ensure IAM password policy requires minimum length of 14 or greater

Error Details:

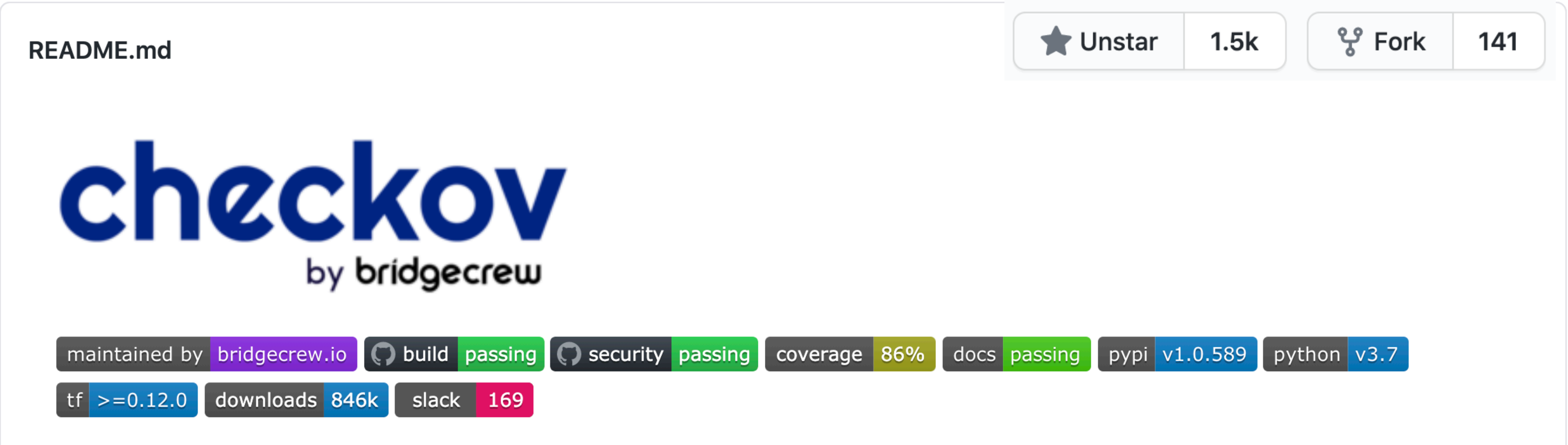
variable "minimum_password_length"

Variables.tf

```
variable "minimum_password_length" {
  description = "Minimum length to require for user passwords"
  default     = 8
}
```

bridgecrew

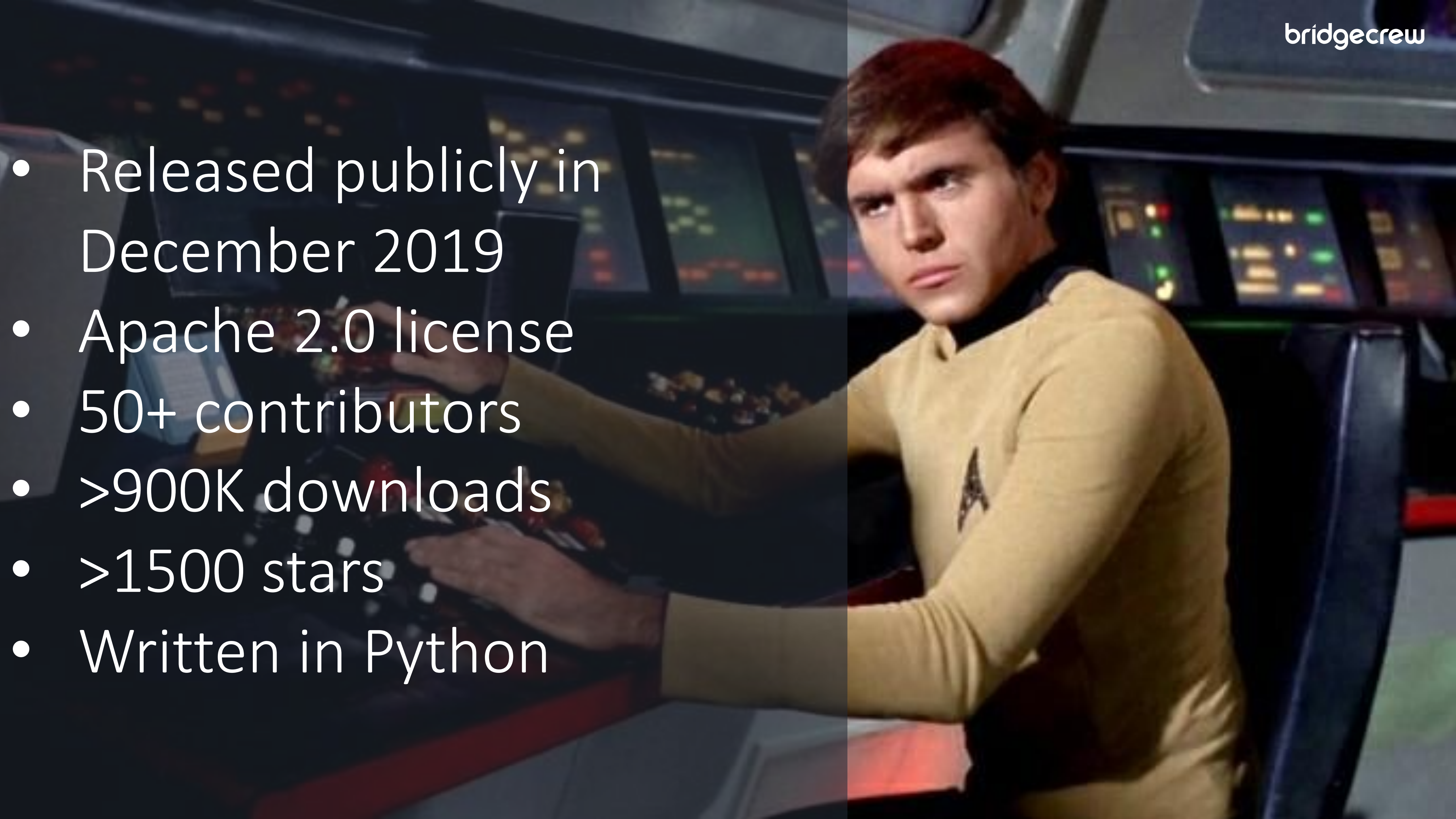
https://github.com/bridgecrewio/checkov



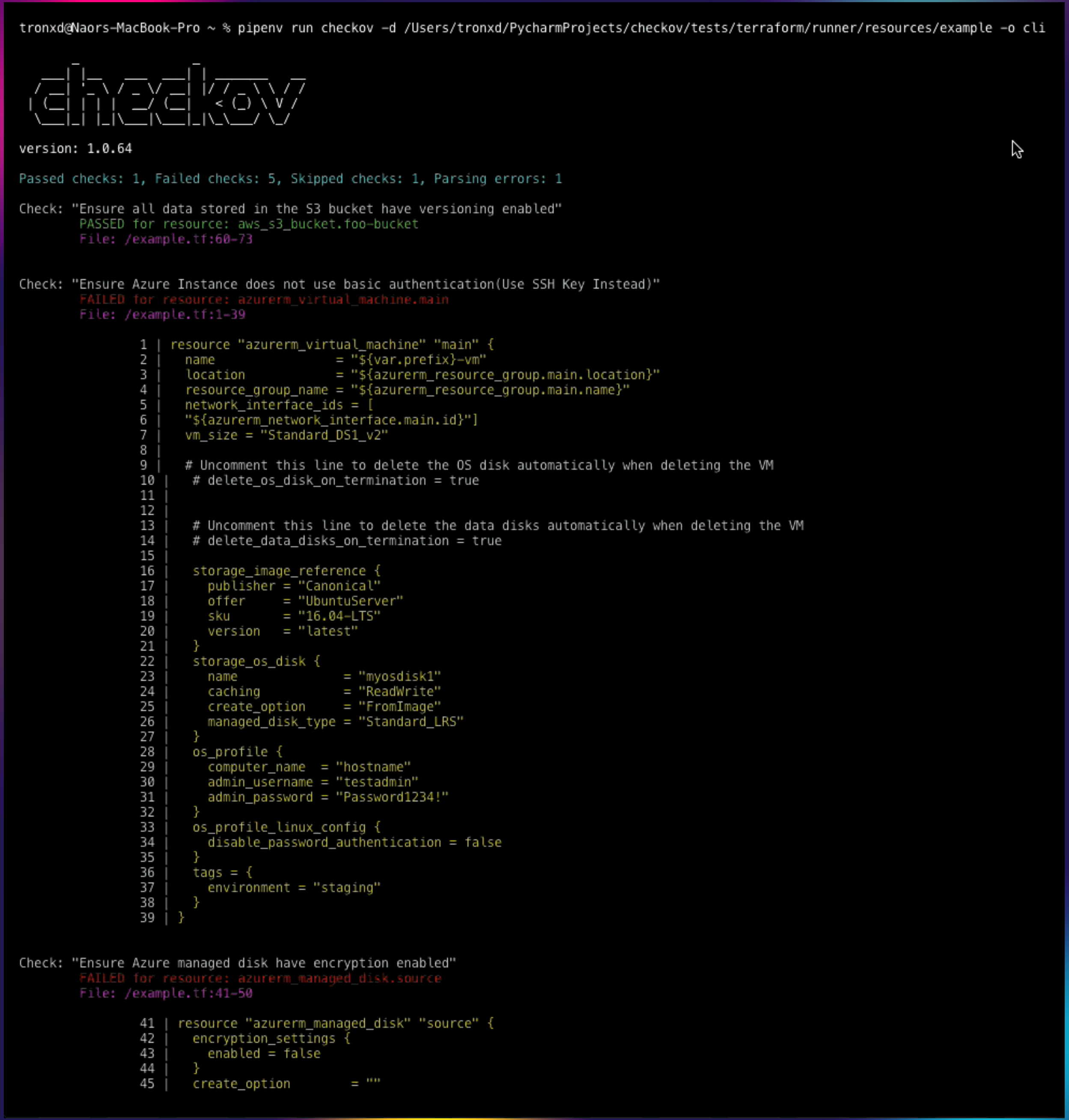
README.md

checkov

by bridgecrew



- Released publicly in December 2019
- Apache 2.0 license
- 50+ contributors
- >900K downloads
- >1500 stars
- Written in Python

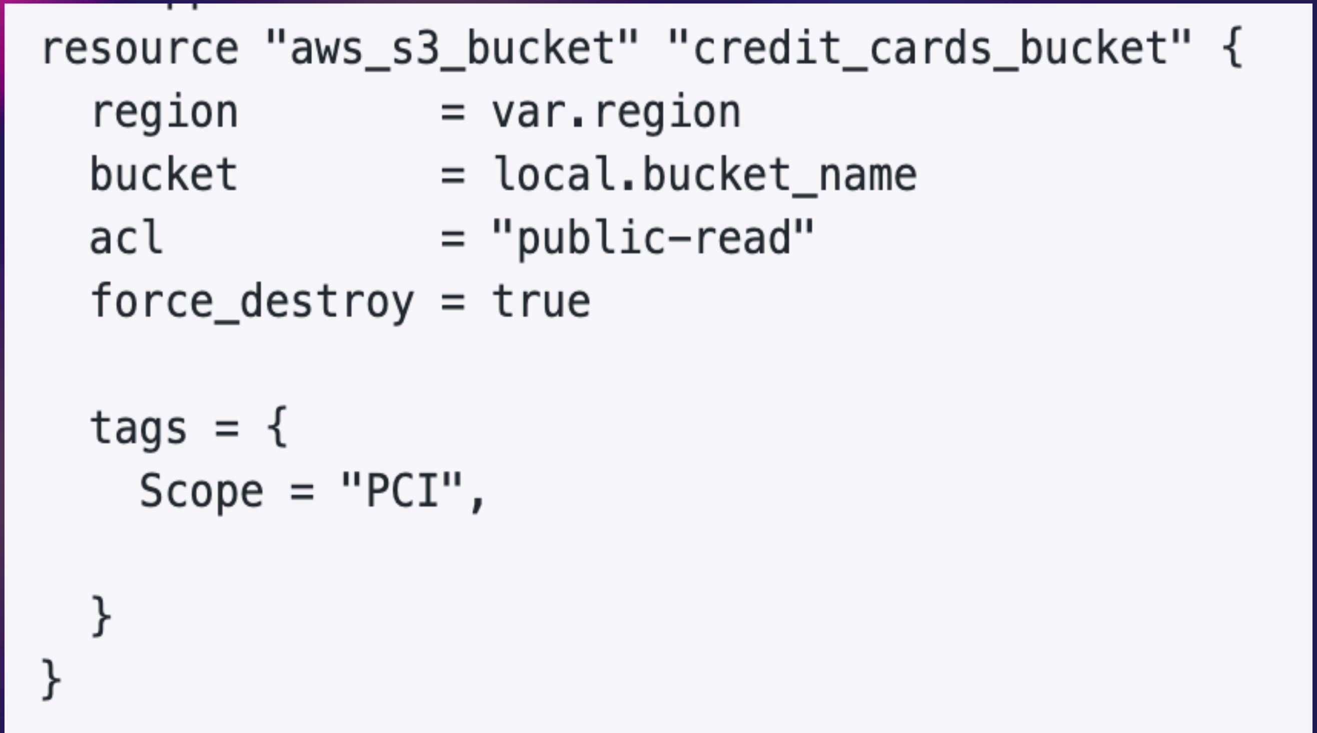


```
tronxd@Naors-MacBook-Pro ~ % pipenv run checkov -d /Users/tronxd/PycharmProjects/checkov/tests/terraform/runner/resources/example -o cli

       _               _
   ___| |__   ___  ___| | _______   __
  / __| '_ \ / _ \/ __| |/ / _ \ \ / /
 | (__| | | |  __/ (__|   < (_) \ V /
  \___|_| |_|\___|\___|_|\_\___/ \_/

version: 1.0.64

Passed checks: 1, Failed checks: 5, Skipped checks: 1, Parsing errors: 1

Check: "Ensure all data stored in the S3 bucket have versioning enabled"
        PASSED for resource: aws_s3_bucket.foo-bucket
        File: /example.tf:60-73


Check: "Ensure Azure Instance does not use basic authentication(Use SSH Key Instead)"
        FAILED for resource: azurerm_virtual_machine.main
        File: /example.tf:1-39

                1 | resource "azurerm_virtual_machine" "main" {
                2 |   name                  = "${var.prefix}-vm"
                3 |   location              = "${azurerm_resource_group.main.location}"
                4 |   resource_group_name = "${azurerm_resource_group.main.name}"
                5 |   network_interface_ids = [
                6 |   "${azurerm_network_interface.main.id}"]
                7 |   vm_size = "Standard_DS1_v2"
                8 |
                9 |   # Uncomment this line to delete the OS disk automatically when deleting the VM
               10 |    # delete_os_disk_on_termination = true
               11 |
               12 |
               13 |    # Uncomment this line to delete the data disks automatically when deleting the VM
               14 |    # delete_data_disks_on_termination = true
               15 |
               16 |    storage_image_reference {
               17 |      publisher = "Canonical"
               18 |      offer     = "UbuntuServer"
               19 |      sku       = "16.04-LTS"
               20 |      version   = "latest"
               21 |    }
               22 |    storage_os_disk {
               23 |      name              = "myosdisk1"
               24 |      caching           = "ReadWrite"
               25 |      create_option     = "FromImage"
               26 |      managed_disk_type = "Standard_LRS"
               27 |    }
               28 |    os_profile {
               29 |      computer_name  = "hostname"
               30 |      admin_username = "testadmin"
               31 |      admin_password = "Password1234!"
               32 |    }
               33 |    os_profile_linux_config {
               34 |      disable_password_authentication = false
               35 |    }
               36 |    tags = {
               37 |      environment = "staging"
               38 |    }
               39 | }


Check: "Ensure Azure managed disk have encryption enabled"
        FAILED for resource: azurerm_managed_disk.source
        File: /example.tf:41-50

               41 | resource "azurerm_managed_disk" "source" {
               42 |    encryption_settings {
               43 |      enabled = false
               44 |    }
               45 |    create_option         = ""
```

Checkov statically analyzes for known best practices implemented in IaC manifests like Terraform and Cloudformation templates

Policy as code

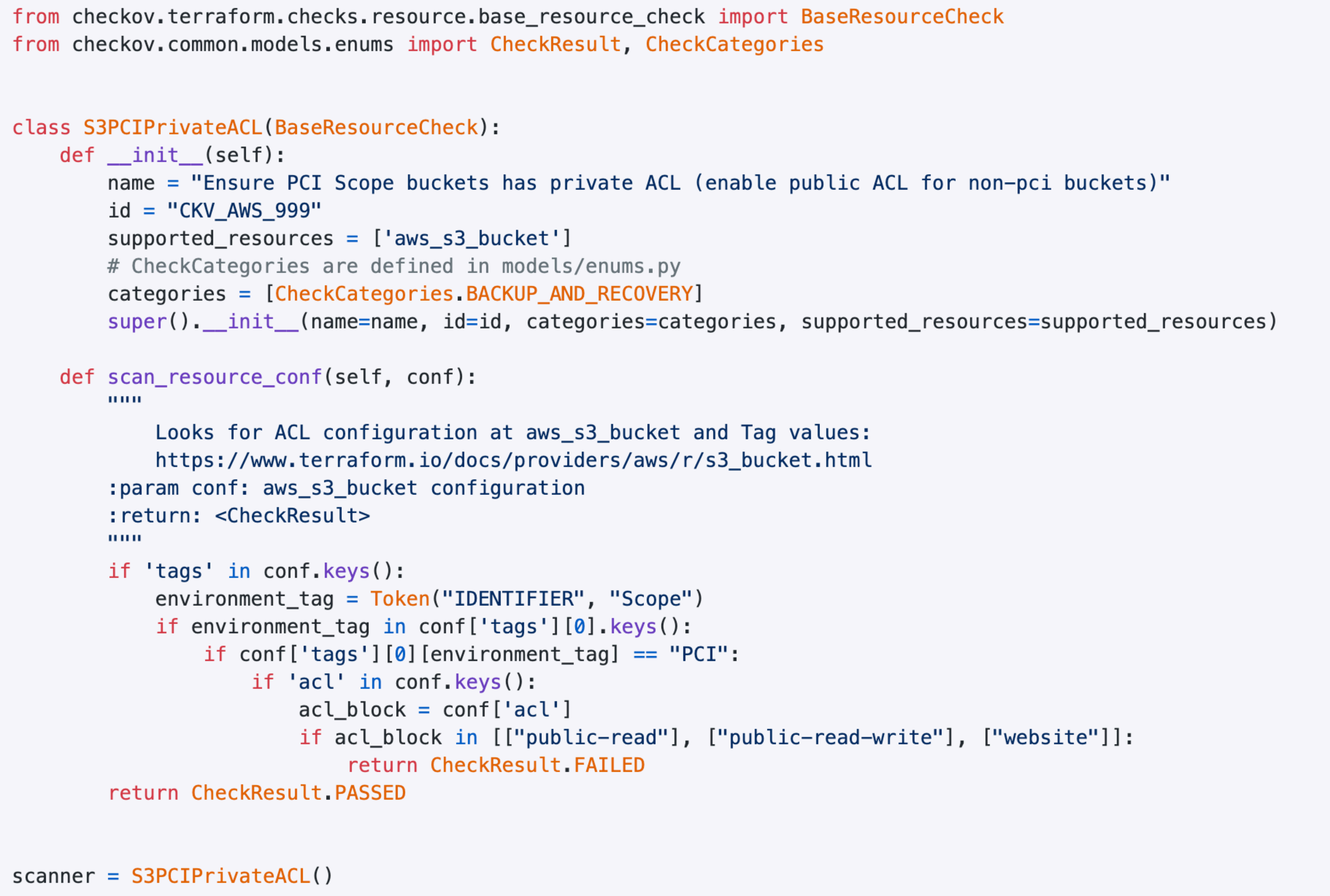
- Version controlled
- Peer reviewed
- Can utilize inheritance and have code reuse (python)
- Part of SDLC
- Continuous integration



```
resource "aws_s3_bucket" "credit_cards_bucket" {
  region        = var.region
  bucket        = local.bucket_name
  acl           = "public-read"
  force_destroy = true

  tags = {
    Scope = "PCI",

  }
}
```

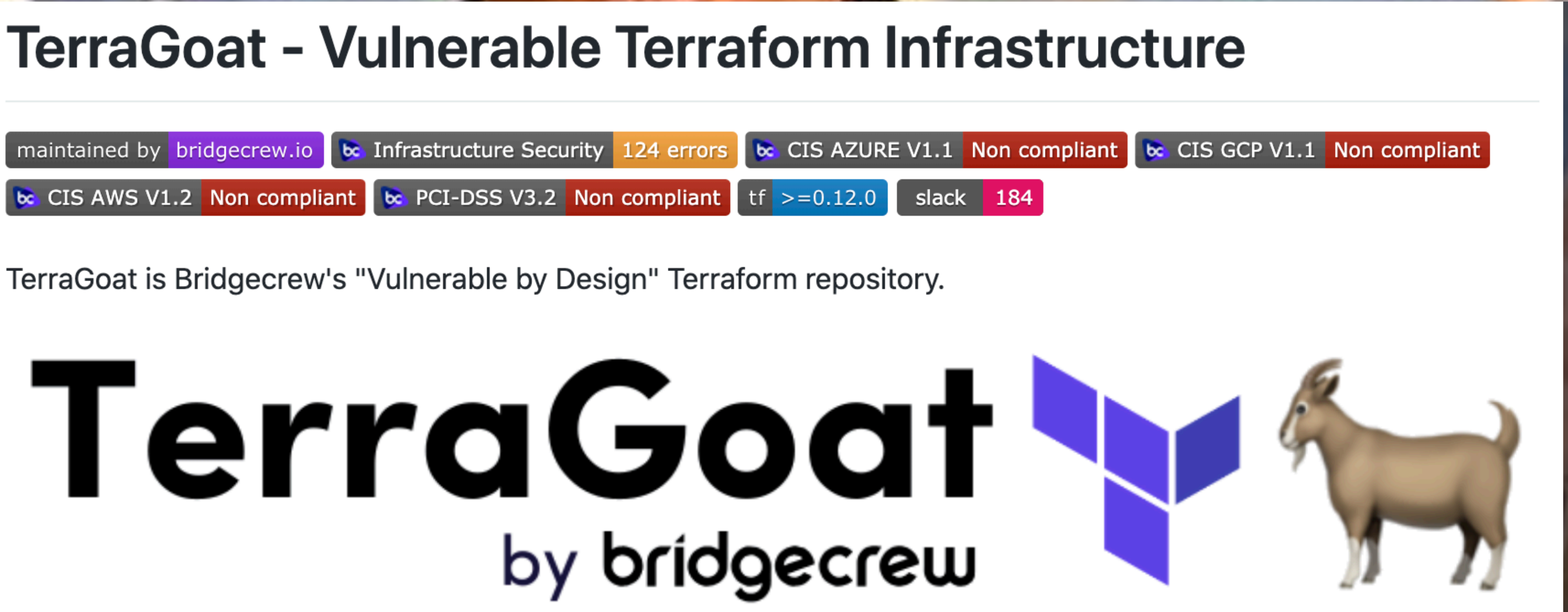
Policy:

Ensure PCI Scope buckets has private ACL (enable public ACL for non-pci buckets)"



```python
from checkov.terraform.checks.resource.base_resource_check import BaseResourceCheck
from checkov.common.models.enums import CheckResult, CheckCategories


class S3PCIPrivateACL(BaseResourceCheck):
    def __init__(self):
        name = "Ensure PCI Scope buckets has private ACL (enable public ACL for non-pci buckets)"
        id = "CKV_AWS_999"
        supported_resources = ['aws_s3_bucket']
        # CheckCategories are defined in models/enums.py
        categories = [CheckCategories.BACKUP_AND_RECOVERY]
        super().__init__(name=name, id=id, categories=categories, supported_resources=supported_resources)

    def scan_resource_conf(self, conf):
        """
            Looks for ACL configuration at aws_s3_bucket and Tag values:
            https://www.terraform.io/docs/providers/aws/r/s3_bucket.html
        :param conf: aws_s3_bucket configuration
        :return: <CheckResult>
        """
        if 'tags' in conf.keys():
            environment_tag = Token("IDENTIFIER", "Scope")
            if environment_tag in conf['tags'][0].keys():
                if conf['tags'][0][environment_tag] == "PCI":
                    if 'acl' in conf.keys():
                        acl_block = conf['acl']
                        if acl_block in [["public-read"], ["public-read-write"], ["website"]]:
                            return CheckResult.FAILED
        return CheckResult.PASSED


scanner = S3PCIPrivateACL()
```

Brace for live demo

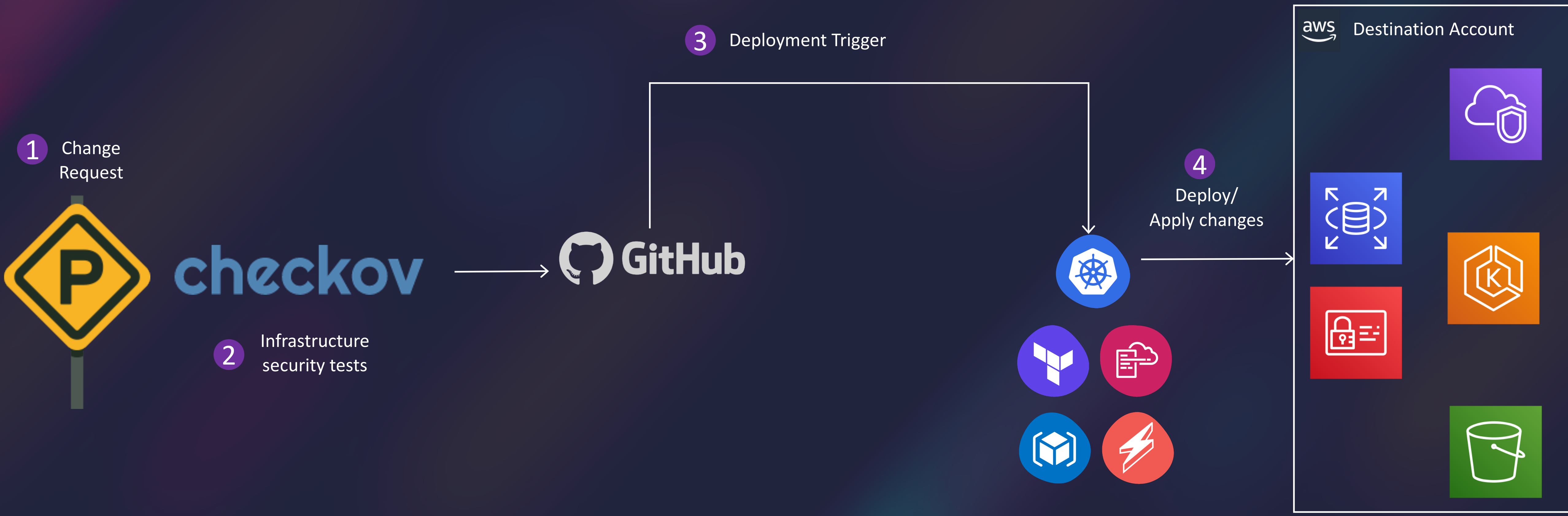


TerraGoat - Vulnerable Terraform Infrastructure

maintained by bridgecrew.io | Infrastructure Security 124 errors | CIS AZURE V1.1 Non compliant | CIS GCP V1.1 Non compliant | CIS AWS V1.2 Non compliant | PCI-DSS V3.2 Non compliant | tf >=0.12.0 | slack 184

TerraGoat is Bridgecrew's "Vulnerable by Design" Terraform repository.

bridgecrew



1 Change Request

checkov

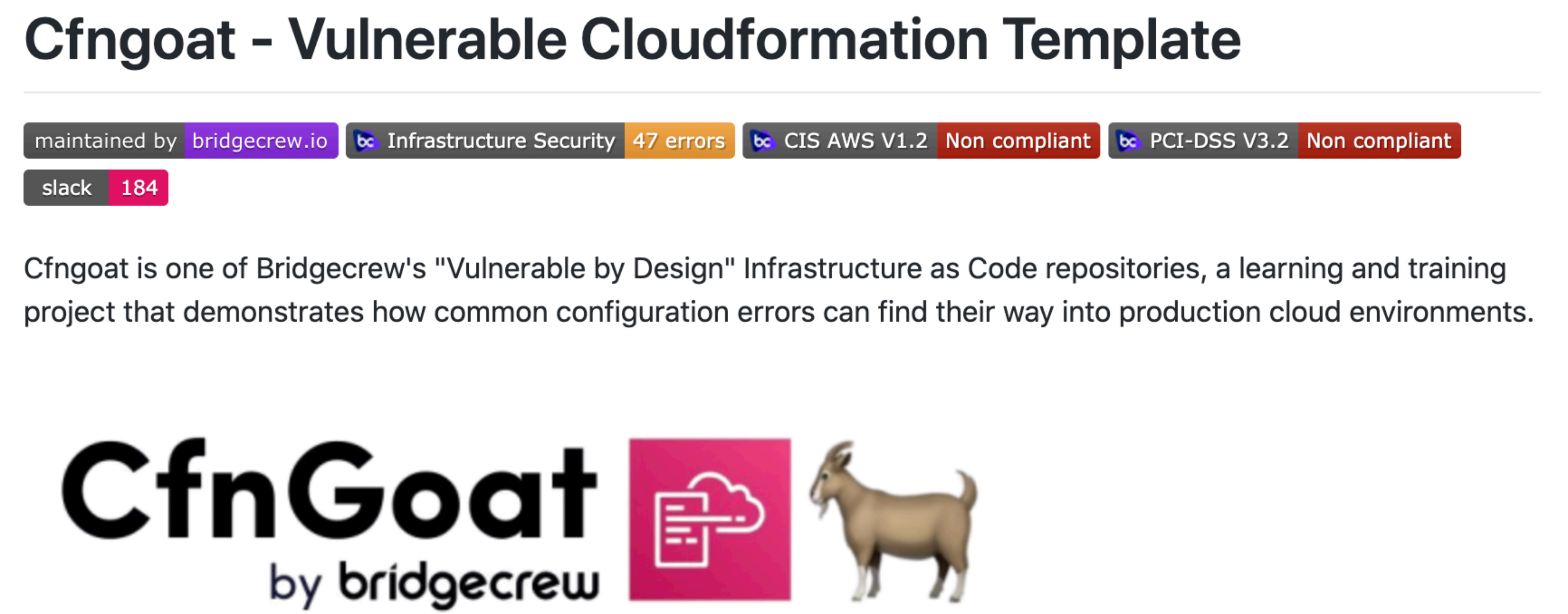
2 Infrastructure security tests

GitHub

3 Deployment Trigger

4 Deploy/ Apply changes

aws Destination Account

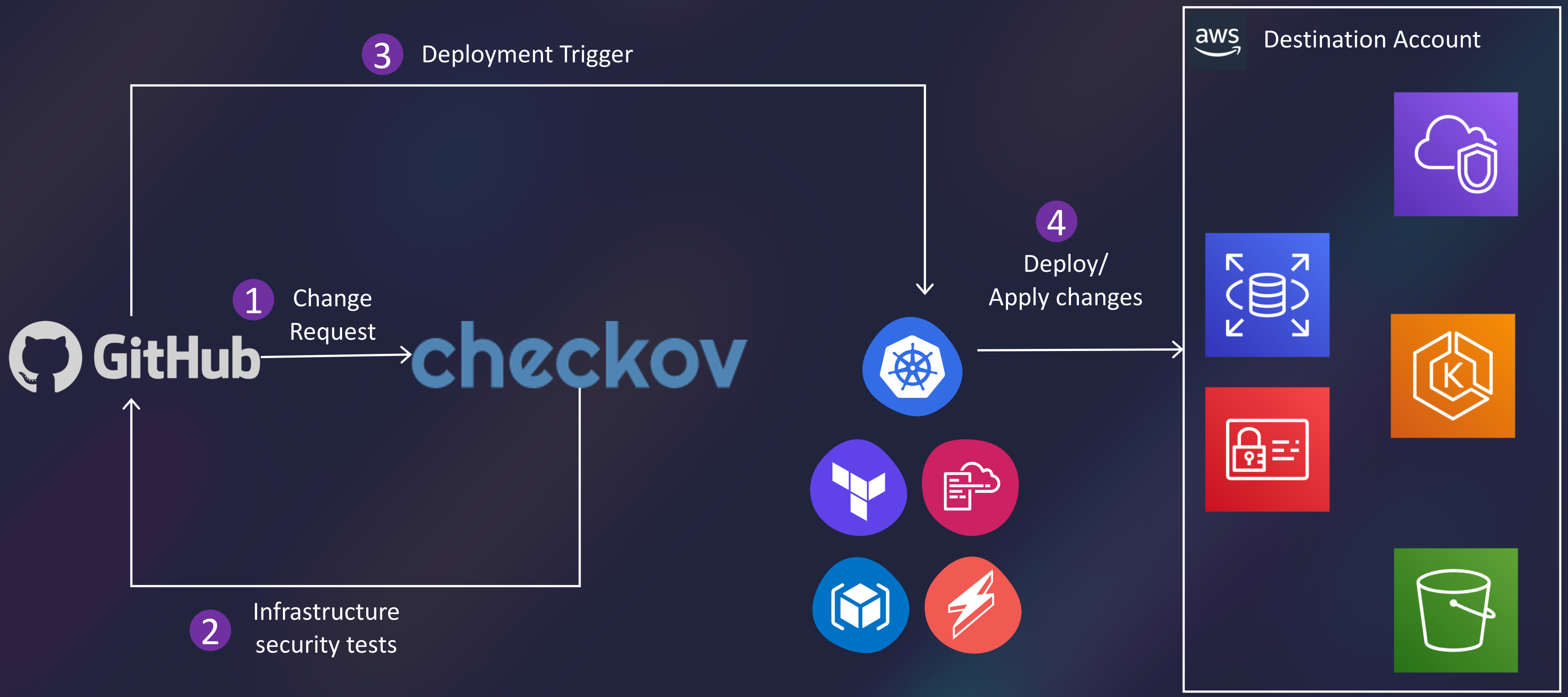


Cfngoat - Vulnerable Cloudformation Template

maintained by bridgecrew.io | Infrastructure Security 47 errors | CIS AWS V1.2 Non compliant | PCI-DSS V3.2 Non compliant | slack 184

Cfngoat is one of Bridgecrew's "Vulnerable by Design" Infrastructure as Code repositories, a learning and training project that demonstrates how common configuration errors can find their way into production cloud environments.

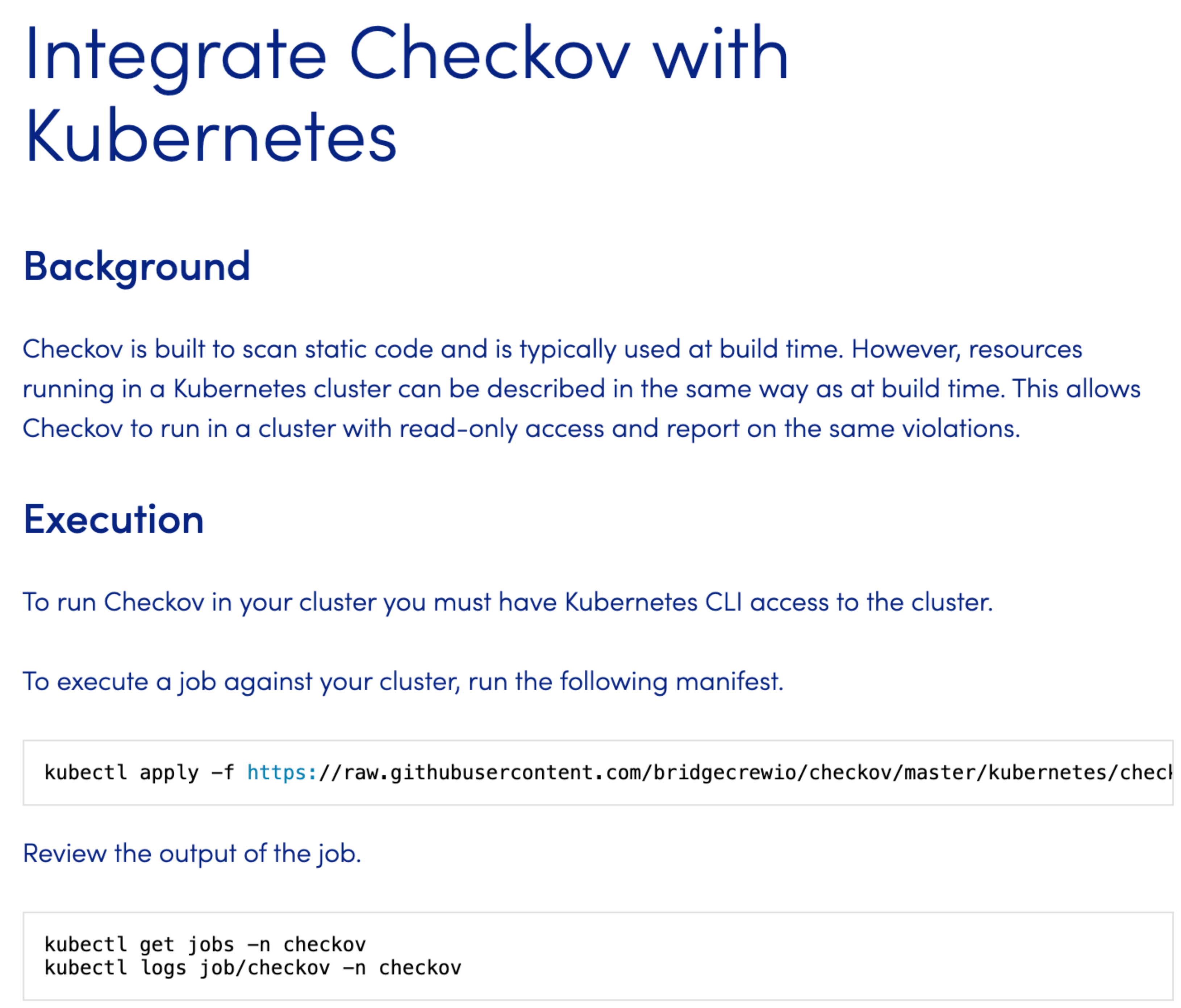
Brace for live demo



3 Deployment Trigger
1 Change Request
GitHub
checkov
2 Infrastructure security tests
4 Deploy/ Apply changes
aws Destination Account

Another one!

Runtime analysis of K8s cluster



Integrate Checkov with Kubernetes

Background

Checkov is built to scan static code and is typically used at build time. However, resources running in a Kubernetes cluster can be described in the same way as at build time. This allows Checkov to run in a cluster with read-only access and report on the same violations.

Execution

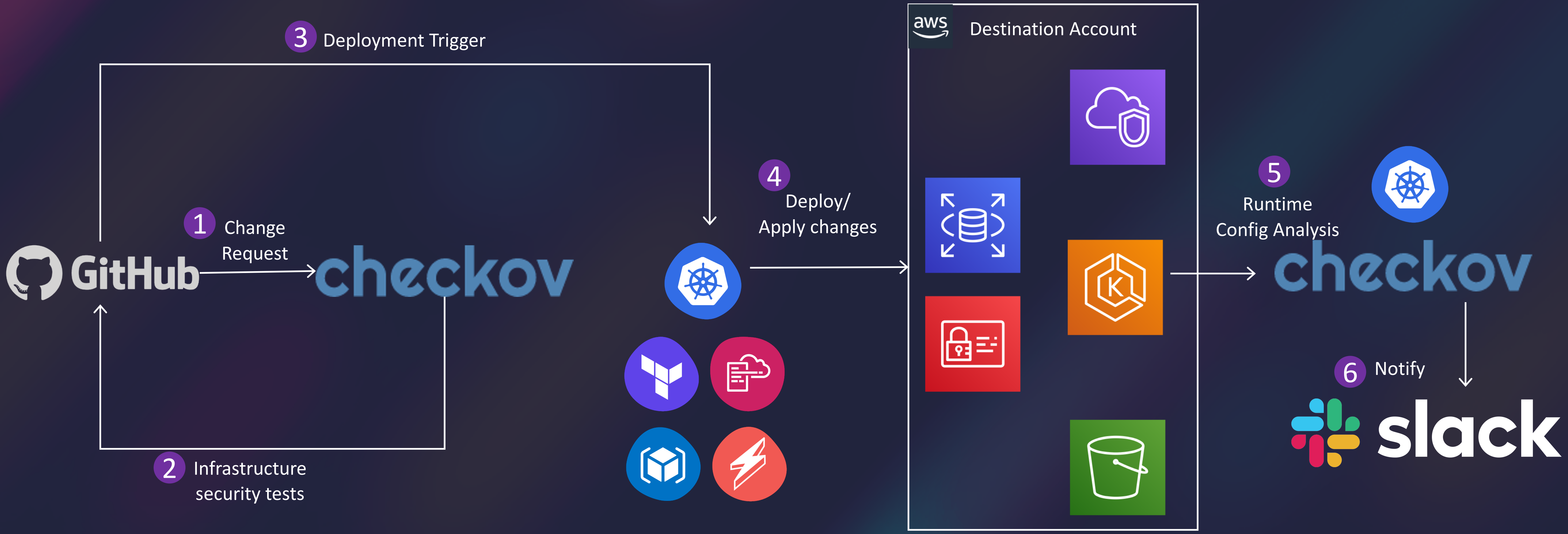
To run Checkov in your cluster you must have Kubernetes CLI access to the cluster.

To execute a job against your cluster, run the following manifest.

```
kubectl apply -f https://raw.githubusercontent.com/bridgecrewio/checkov/master/kubernetes/check
```

Review the output of the job.

```
kubectl get jobs -n checkov
kubectl logs job/checkov -n checkov
```



3 Deployment Trigger
1 Change Request
GitHub
checkov
2 Infrastructure security tests
4 Deploy/ Apply changes
aws Destination Account
5 Runtime Config Analysis
checkov
6 Notify
slack

Misconfig Analysis

Pre-commit

GitHub

Continuous Integration

Running Cluster

A WORLD WHERE:

Infrastructure is developed and secured in the same place

Issues are automatically prevented from being deployed

Security is a business enabler rather than a hindrance

TAKEAWAYS

Keep your manifests secure

Have a fast feedback loop on configuration changes

Monitor both build-time and runtime

Version control your policies

bridgecrew

Try Checkov and join the slack slack.bridgecrew.io

CONTACT ME barak@bridgecrew.io

THANK YOU TO OUR SPONSORS

NORTHROP GRUMMAN

sonatype

State Farm™

CloudBees®

Google Cloud

NowSecure™

D2IQ

ConnectALL
An Orasi Company

moogsoft®

NS1.

Carnegie Mellon University
Software Engineering Institute

CONTINO

ATARC

MACQUARIE

DZone®
A DEVADA MEDIA PROPERTY

MediaOps

TechBeacon

SD Times
SOFTWARE DEVELOPMENT

Forcepoint