Preventing Kubernetes Misconfiguration: Static Analysis and Beyond

Matt Johnson
Developer Advocate Lead
bridgecrew

bridgecrew

AGENDA

- Misconfiguration challenges
- Write policy as code
- Automate in our CI Pipeline
- Helm chart analysis
- Runtime analysis of k8 cluster

Matt Johnson

🐦 @Metahertz

metahertz

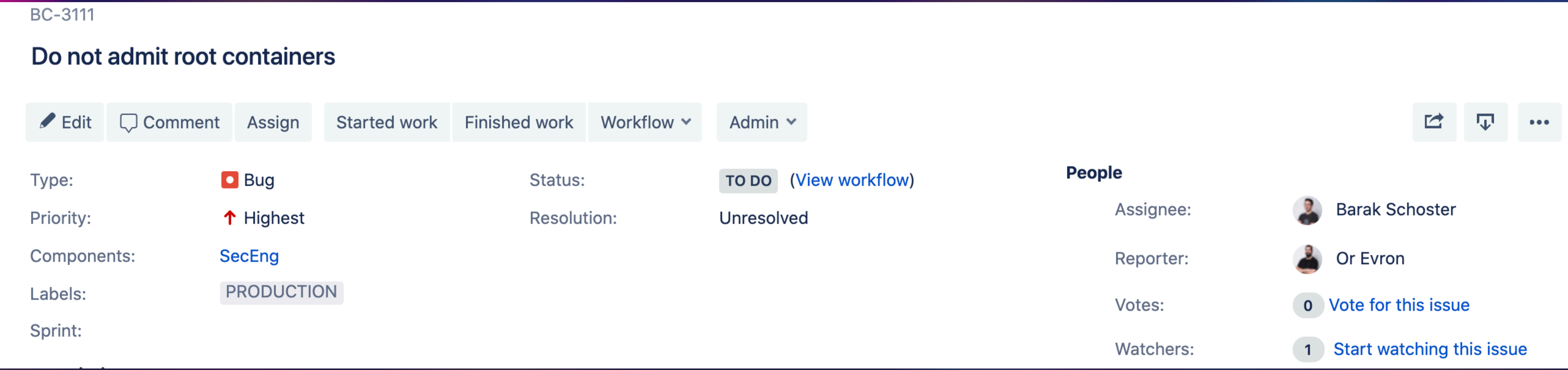
As an engineer
I want to move fast

I <u>DO NOT</u>

want to break things

The thing I have
love/hate relationship with is…

Jira



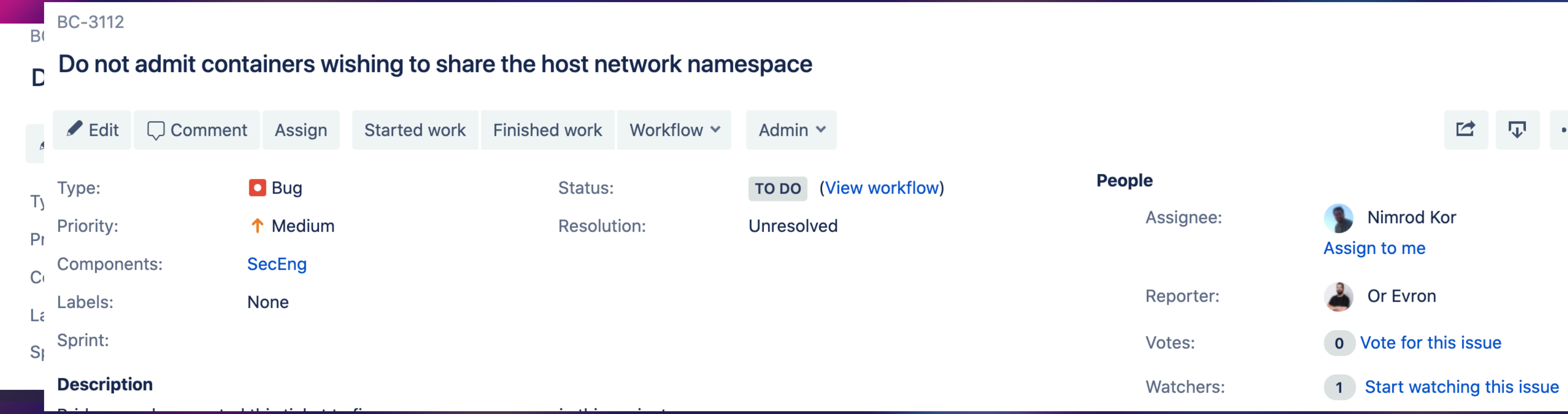
BC-3111

Do not admit root containers

Type:	Bug	Status:	TO DO (View workflow)	Assignee:	Barak Schoster
Priority:	↑ Highest	Resolution:	Unresolved	Reporter:	Or Evron
Components:	SecEng			Votes:	0 Vote for this issue
Labels:	PRODUCTION			Watchers:	1 Start watching this issue
Sprint:					

People

bridgecrew



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

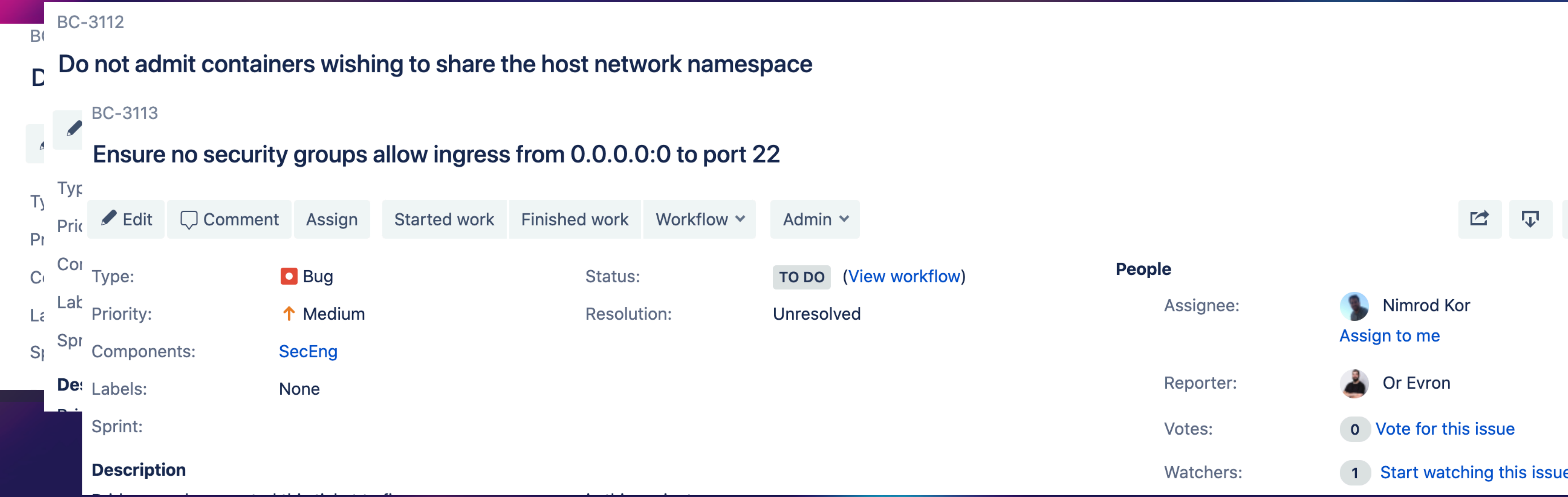
Assign to me

Reporter: Or Evron

Votes: 0 Vote for this issue

Watchers: 1 Start watching this issue

Description



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

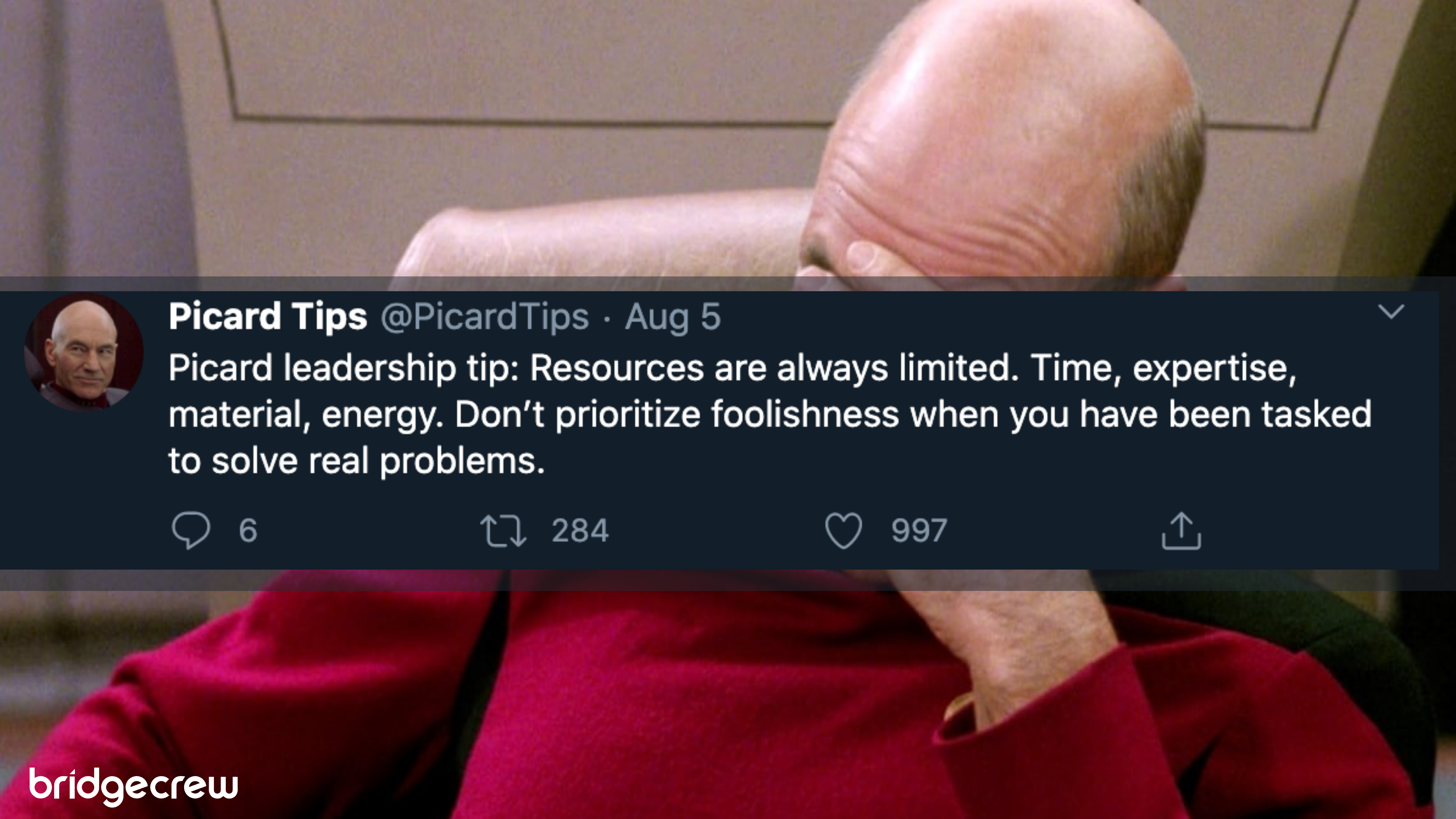
Type:	Bug	Status:	TO DO (View workflow)
Priority:	Medium	Resolution:	Unresolved
Components:	SecEng		
Labels:	None		
Sprint:			

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

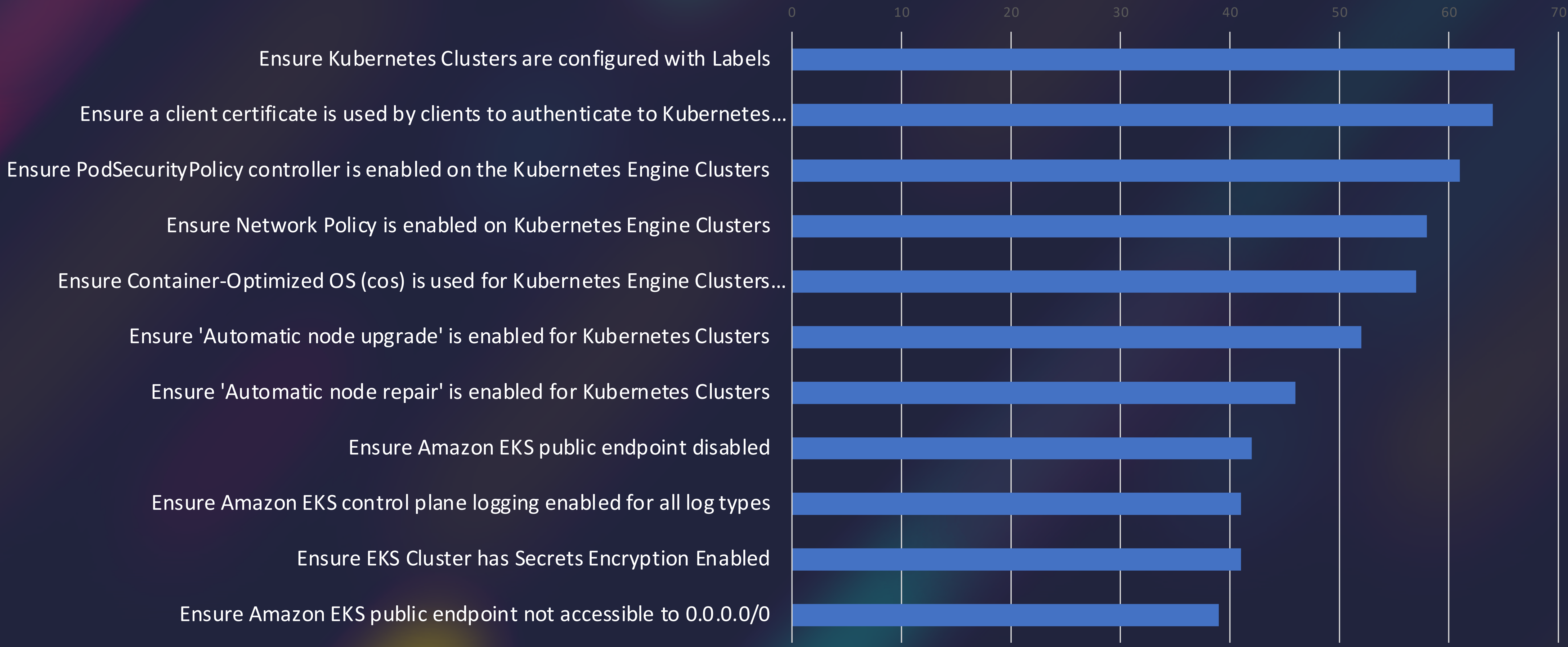
Picard leadership tip: Resources are always limited. Time, expertise, material, energy. Don’t prioritize foolishness when you have been tasked to solve real problems.

bridgecrew

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

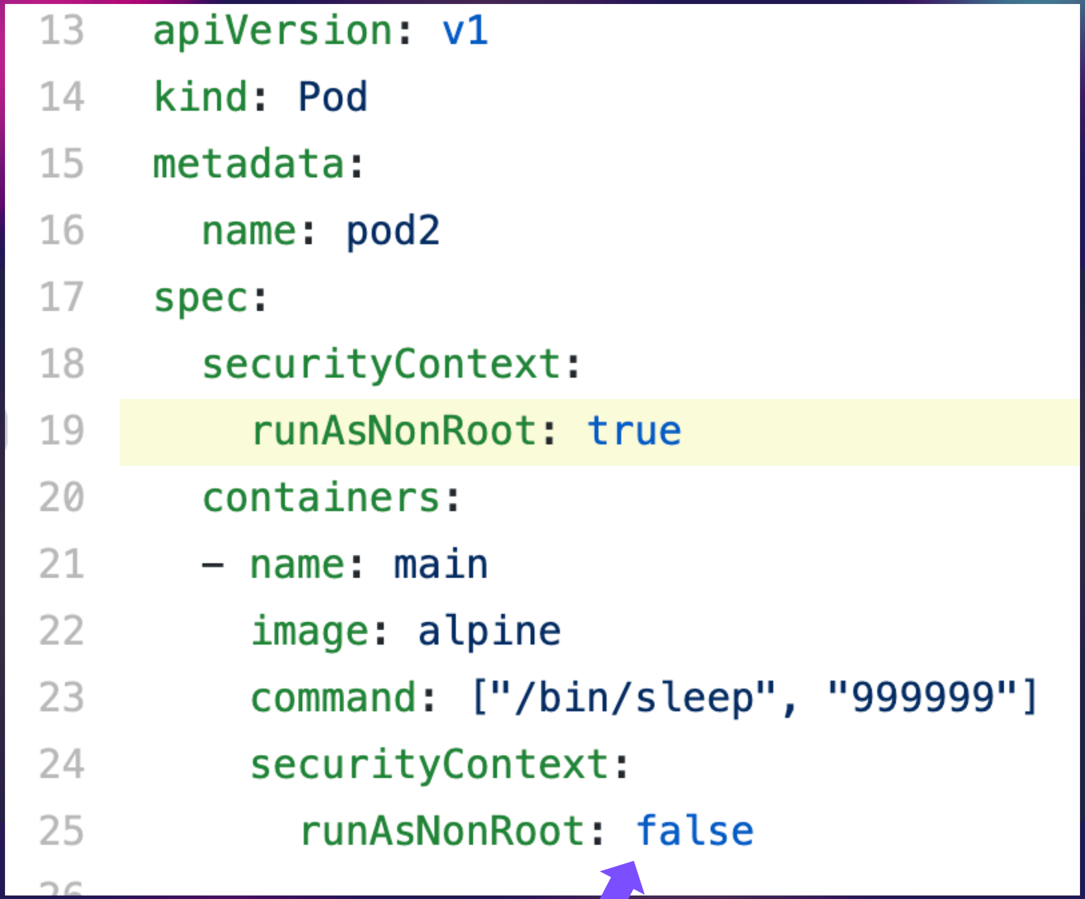
…data

Top Failing Kubernetes checks

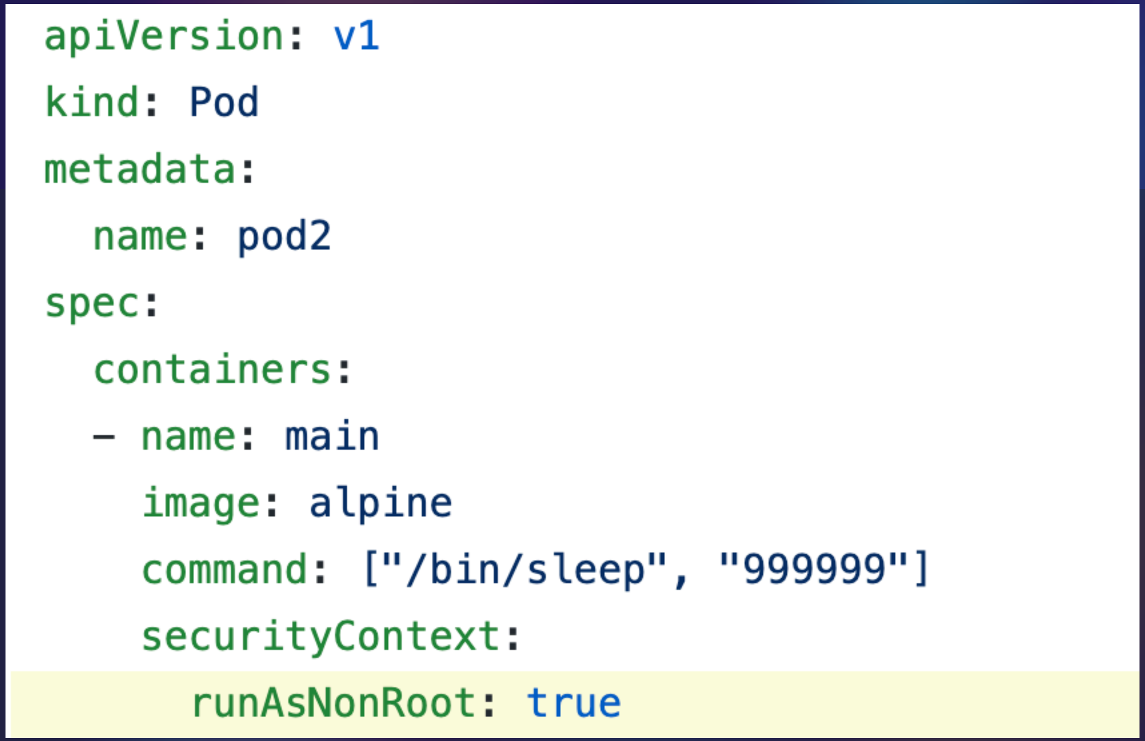


Infrastructure as code (IaC) presents
a new **risk** and a new **opportunity**

bridgecrew

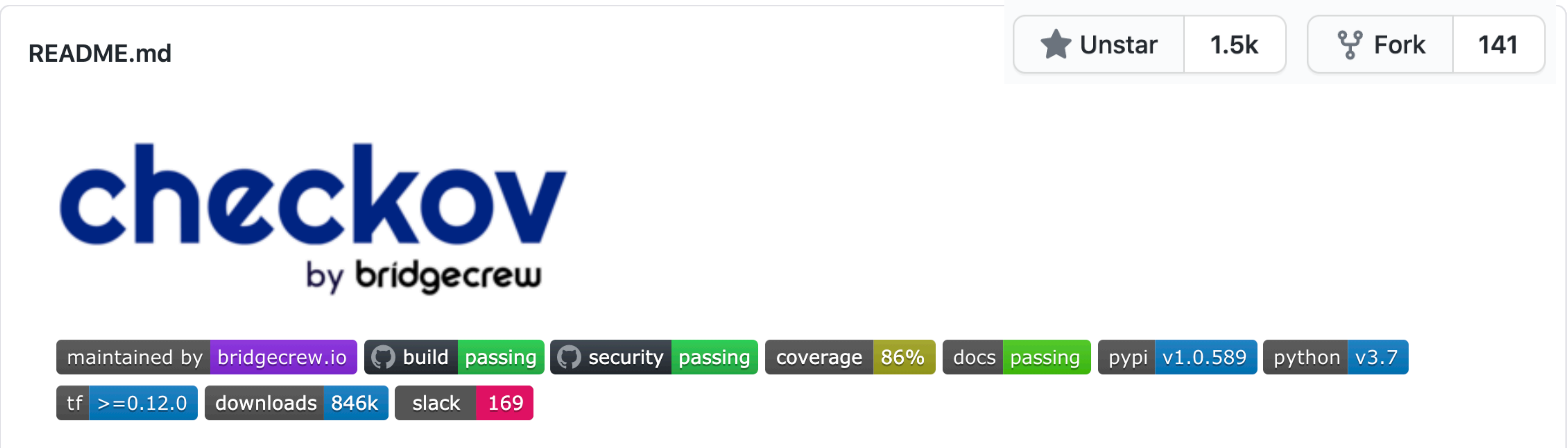


```yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod2
spec:
  securityContext:
    runAsNonRoot: true
  containers:
  - name: main
    image: alpine
    command: ["/bin/sleep", "999999"]
    securityContext:
      runAsNonRoot: false
```



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod2
spec:
  containers:
  - name: main
    image: alpine
    command: ["/bin/sleep", "999999"]
    securityContext:
      runAsNonRoot: true
```

https://github.com/bridgecrewio/checkov



README.md

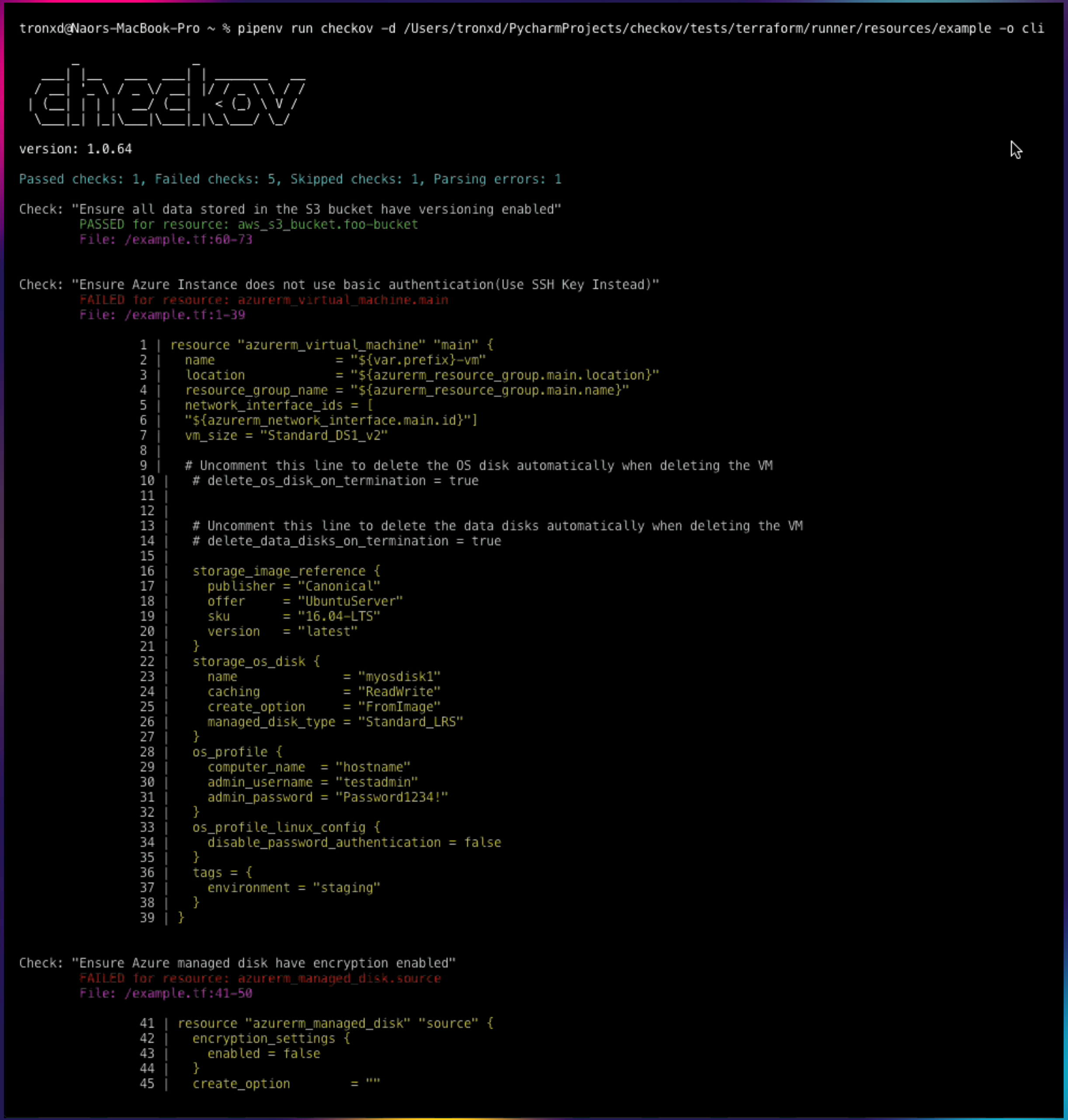
checkov

by bridgecrew

maintained by bridgecrew.io build passing security passing coverage 86% docs passing pypi v1.0.589 python v3.7

tf >=0.12.0 downloads 846k slack 169

- Released publicly in December 2019
- Apache 2.0 license
- 50+ contributors
- >800K downloads
- >1400 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        = ""
```

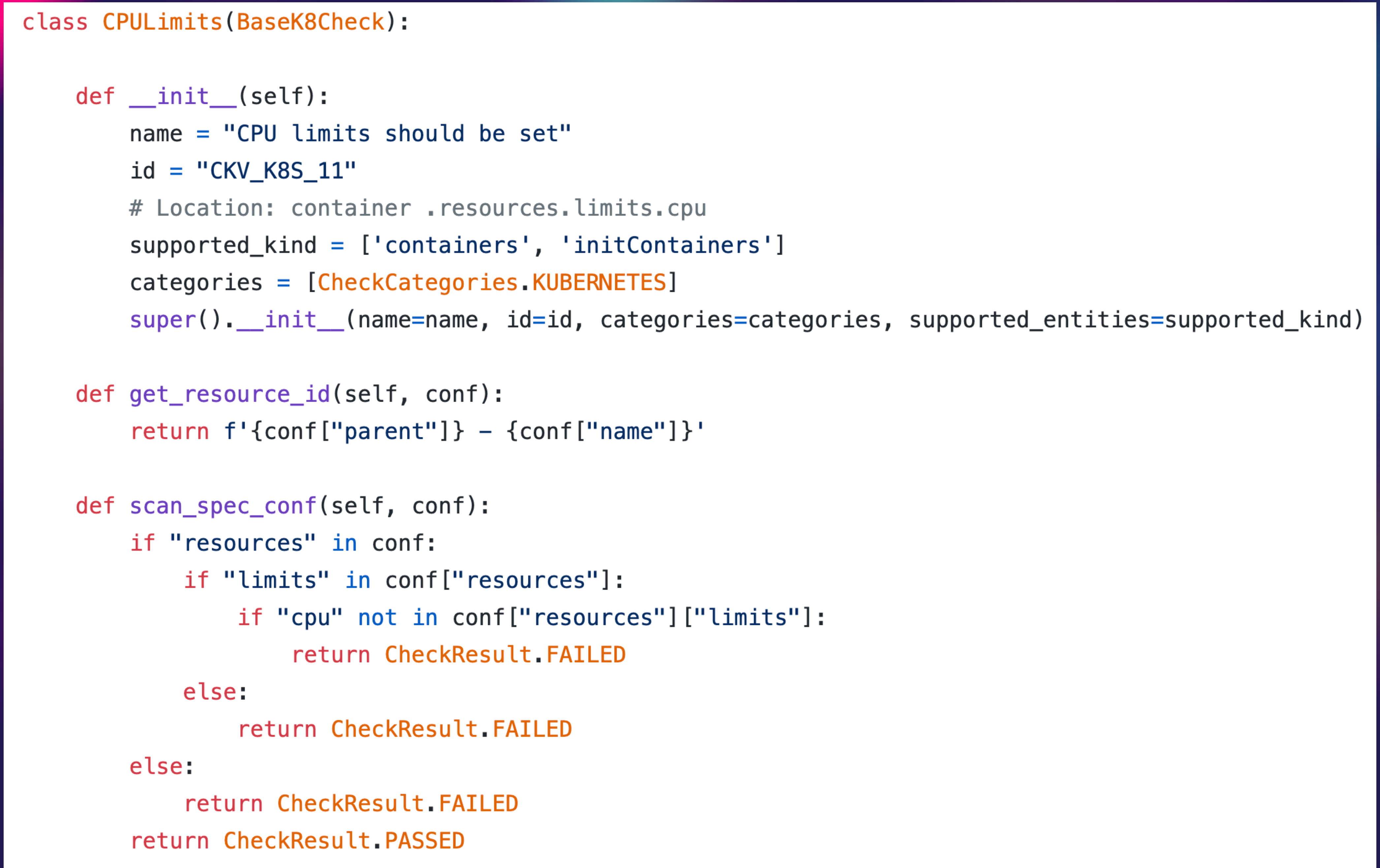
Checkov statically analyzes for known best practices implemented in IaC manifests like k8s YAML

bridgecrew

Policy as code

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

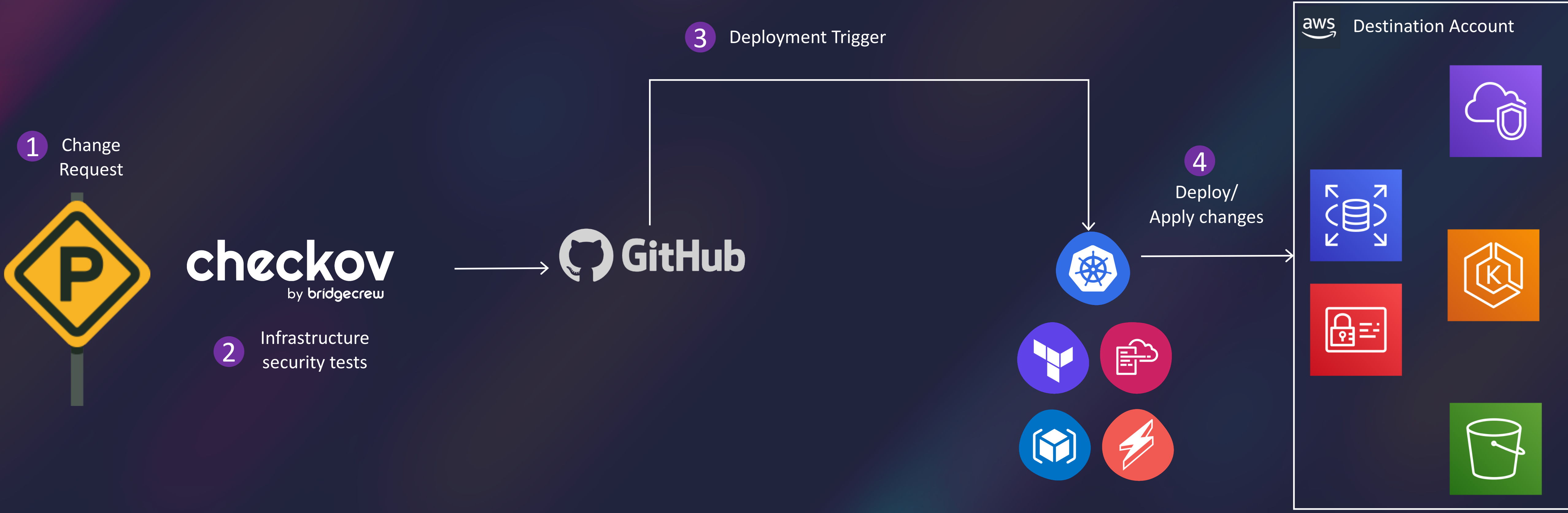
bridgecrew



```python
class CPULimits(BaseK8Check):

    def __init__(self):
        name = "CPU limits should be set"
        id = "CKV_K8S_11"
        # Location: container .resources.limits.cpu
        supported_kind = ['containers', 'initContainers']
        categories = [CheckCategories.KUBERNETES]
        super().__init__(name=name, id=id, categories=categories, supported_entities=supported_kind)

    def get_resource_id(self, conf):
        return f'{conf["parent"]} - {conf["name"]}'

    def scan_spec_conf(self, conf):
        if "resources" in conf:
            if "limits" in conf["resources"]:
                if "cpu" not in conf["resources"]["limits"]:
                    return CheckResult.FAILED
            else:
                return CheckResult.FAILED
        else:
            return CheckResult.FAILED
        return CheckResult.PASSED
```

Brace for live demo

bridgecrew



1 Change Request

2 Infrastructure security tests

3 Deployment Trigger

4 Deploy/ Apply changes

Destination Account

bridgecrew

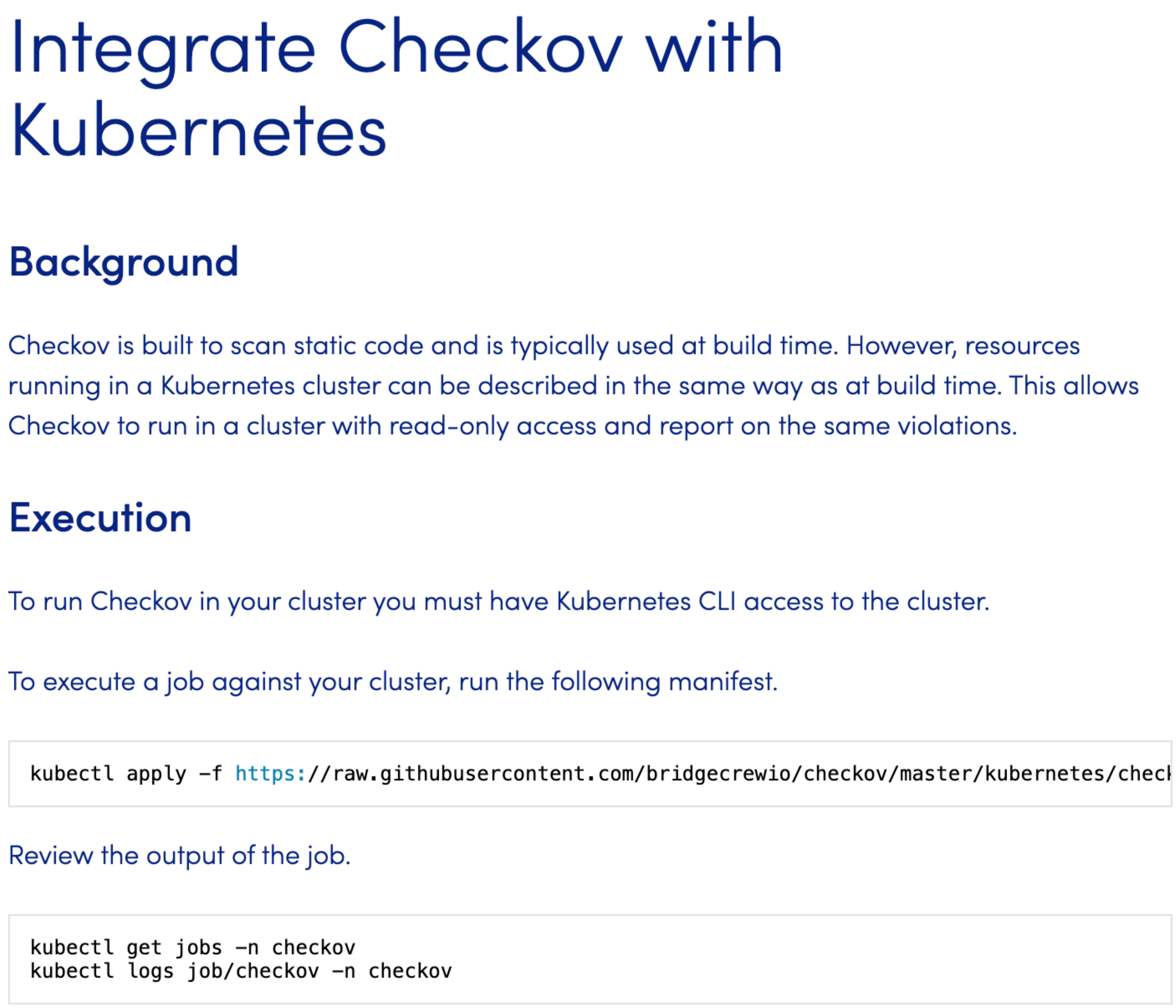
Brace for live demo

checkov
by bridgecrew

HELM

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

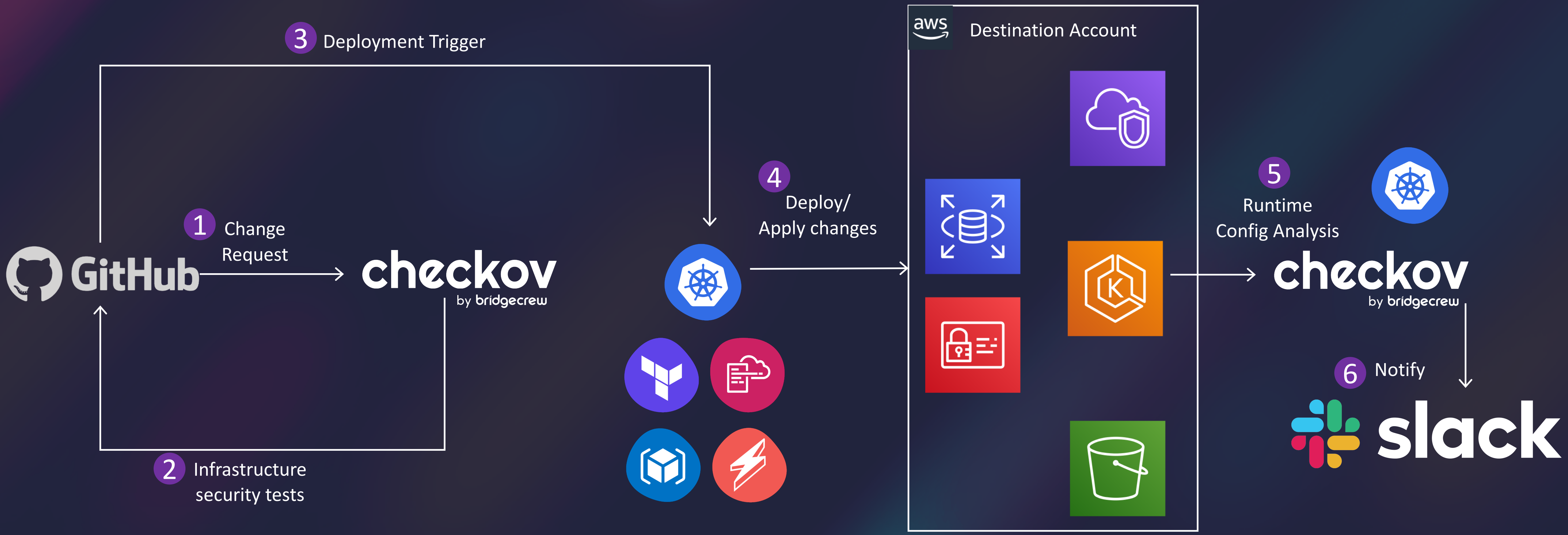
To run Checkov in your cluster you must have Kubernetes CLI access to the cluster.

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 by bridgecrew
2 Infrastructure security tests
4 Deploy/ Apply changes
aws Destination Account
5 Runtime Config Analysis
checkov by bridgecrew
6 Notify
slack

Misconfig Analysis

Pre-commit

Continuous Integration

Running Cluster

bridgecrew

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 Kubernetes manifests and Helm charts secure

Have a fast feedback loop on configuration changes

Monitor both build-time and runtime

Version control your policies

bridgecrew

Try Checkov and join our Slack slack.bridgecrew.io

CONTACT ME matt@bridgecrew.io