@BiradarSangam

we use different cloud providers like aws , GCP & Azure also adopted famous project like k8s & helm ! looking for open source tool to secure infrastructure misconfiguration

https://github.com/tenable/terrascan

terrascan by tenable

tell me more about it !

In short
1. its contain pre-defined 500+ Policies for security best practices
2. its support scanning of terraform , AWS's cloudFormation Templates (CFT) Azure Resource Manager (ARM) also support kubernetes , Helm charts & kustomize as well as DockerFile
3. integrated with docker images vulnerabilities scanning
for AWS/Azure/GCP ,harbor container registeris

Quick Start Your Terrascan Journey
Secure your cloud native infrastructure

Install terrascan on mac / linux

```
> curl -L "$(curl -s https://api.github.com/repos/tenable/terrascan/releases/latest
  |grep -o -E 'https://.+?_Darwin_x86_64.tar.gz")" > terrascan.tar.gz
> tar -xf terrascan.tar.gz terrascan && rm terrascan.tar.gz
> install terrascan /usr/local/bin && rm terrascan
> terrascan
```

Homebrew users can install by

```
> brew install terrascan
```

Some of most useful commands

```
> terrascan init ----> download the latest rego policies into ./terrascan
> terrascan [command] --help ----> provide more info about a command
> terrascan scan -t aws -----> scan terraform AWS Resources
> terrascan scan -t azure ----> scan terraform Azure Resources
> terrascan scan -t gcp -----> scan terraform GCP resources
> terrascan scan -i cft -----> scan AWS's CloudFormation
> terrascan scan -i docker -----> scan dockerfile
> terrascan scan -i kustomize -----> scan Kustomize
> terrascan scan -i Kustomize --iac-version v2
> terrascan scan -t aws -r git -u <github-url> ----> scan remote repository
> terrascan scan -i helm  ----> scan helm charts
```

Integrations

Use as Kubernetes Admission Controller
Argo CD pre-sync hook to scan github repo
Github and Gitlab
Pre-Commit Hook
Write your own Custom Policies with Terrascan Rego editor

run Terrascan as server mode

```
> terrascan server
```

default terrascan server listen on 9010 and support following route

API Routes
GET -/Health ---- check health of server

Scan IaC File

POST -/v1{iac}/{iacVersion}/{cloud}/local/file/scan

example curl -i -F "file=@aws_cloudfront_distribution.tf" localhost:9010/v1/terraform/v14/aws/local/file/scan

Scan Remote IaC

POST - /v1/{iac}/{iacVersion}/{cloud}/remote/dir/scan

Terraform AWS

```
curl --location --request POST 'http://localhost:9010/v1/terraform/v14/aws/local/file/scan' \
--form 'file=@"main.tf"'
```

Terraform Azure

```
curl --location --request POST 'http://localhost:9010/v1/terraform/v14/azure/local/file/scan' \
--form 'file=@"main.tf"'
```

Terraform GCP

```
curl --location --request POST 'http://localhost:9010/v1/terraform/v14/gcp/local/file/scan' \
--form 'file=@"main.tf"'
```

Terraform cloudFormation

```
curl --location --request POST 'http://localhost:9010/v1/cft/v1/aws/local/file/scan' \
--form 'file=@"incorrectTypesInParamsCftTemplate.yml"'
```

kubernetes

```
curl --location --request POST 'http://localhost:9010/v1/k8s/v1/k8s/local/file/scan' \
--form 'file=@"test_pod.yml"'
```

Docker

```
curl --location --request POST 'http://localhost:9010/v1/docker/v1/Docker/local/file/scan' \
--form 'file=@"Dockerfile"'
```

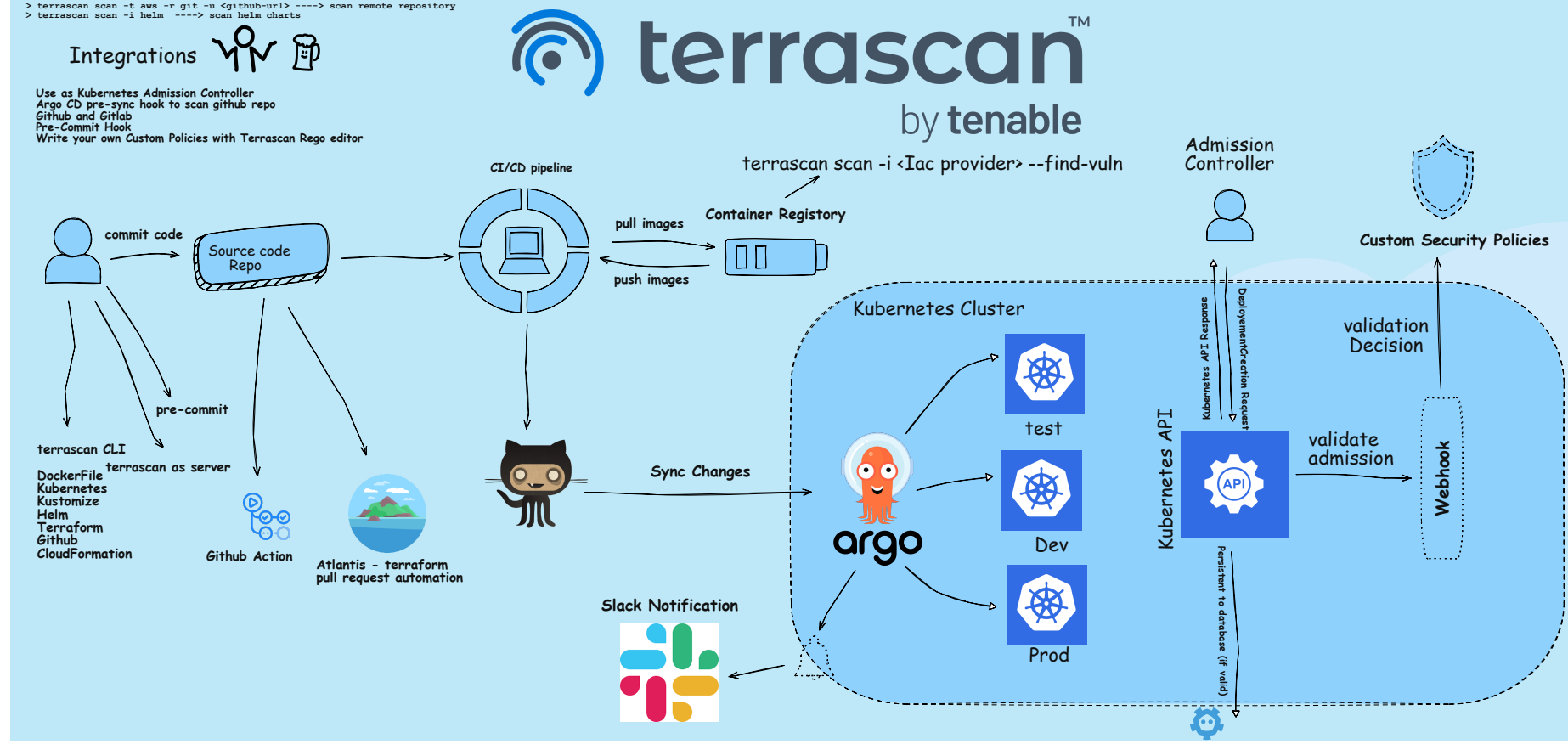
you can check above examples really useful

Terrascan Scan flags

```
terrascan scan [flags]

Flags:
      --categories strings list of categories of violations to be reported by terrascan (
        example: --categories="category1,category2")
      --config-only will output resource config (should only be used for debugging purposes)
      --find-vuln fetches vulnerabilities identified in Docker images
  -h, --help help for scan
  -d, --iac-dir string path to a directory containing one or more IaC files (default ".")
  -f, --iac-file string path to a single IaC file
  -i, --iac-type string iac type (arm, cft, docker, helm, k8s, kustomize, terraform, tfplan)
      --iac-version string iac version (arm: v1, cft: v1, docker: v1, helm: v3, k8s: v1, kustomize: v2, v3, v4,
         terraform: v12, v13, v14, v15, tfplan: v1)
      --non-recursive do not scan directories and modules recursively
      --webhook-token string the auth token to call the notification webhook URL
      --webhook-url string the URL where terrascan will send the scan report and normalized config json
  -p, --policy-path stringArray policy path directory
  -t, --policy-type strings policy type (all, aws, azure, docker, gcp, github, k8s) (default [all])
  -r, --remote-type string type of remote backend (git, s3, gcs, http, terraform-registry)
  -u, --remote-url string url pointing to remote IaC repository
      --repo-ref string branch of the repo being scanned
      --repo-url string URL of the repo being scanned, will be reflected in scan summary
      --scan-rules strings one or more rules to scan (example: --scan-rules="ruleID1,ruleID2")
      --severity string minimum severity level of the policy violations to be reported by terrascan
      --show-passed display passed rules, along with violations
      --skip-rules strings one or more rules to skip while scanning (example: --skip-rules="ruleID1,ruleID2")
      --use-colors string color output (auto, t, f) (default "auto")
      --use-terraform-cache use terraform init cache for remote modules (when used directory scan will be
           non recursive, flag applicable only with terraform IaC provider)
  -v, --verbose will show violations with details (applicable for default output)

Global Flags:
  -c, --config-path string config file path
  -l, --log-level string log level (debug, info, warn, error, panic, fatal) (default "info")
      --log-output-dir string directory path to write the log and output files
  -x, --log-type string log output type (console, json) (default "console")
  -o, --output string output type (human, json, yaml, xml, junit-xml, sarif, github-sarif) (default "human")
      --temp-dir string temporary directory path to download remote repository,module and templates
```



CloudNativeFolks Community
https://discord.com/invite/9ERSnT7