A Developer's Guide to Kubernetes Security

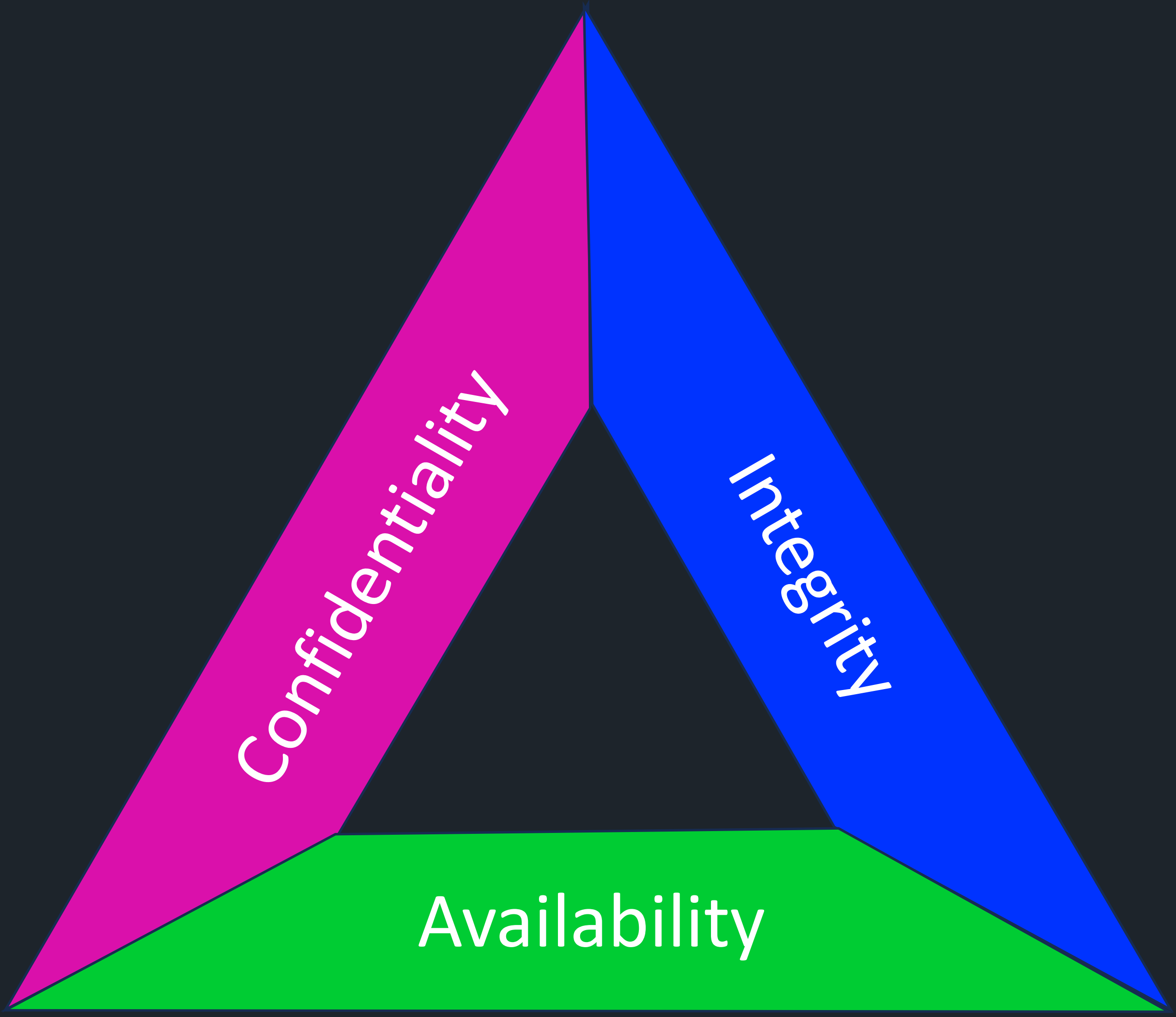
Gene Gotimer

Principal DevOps Engineer at Praeses, LLC

@OtherDevOpsGene

SECURITY

CIA



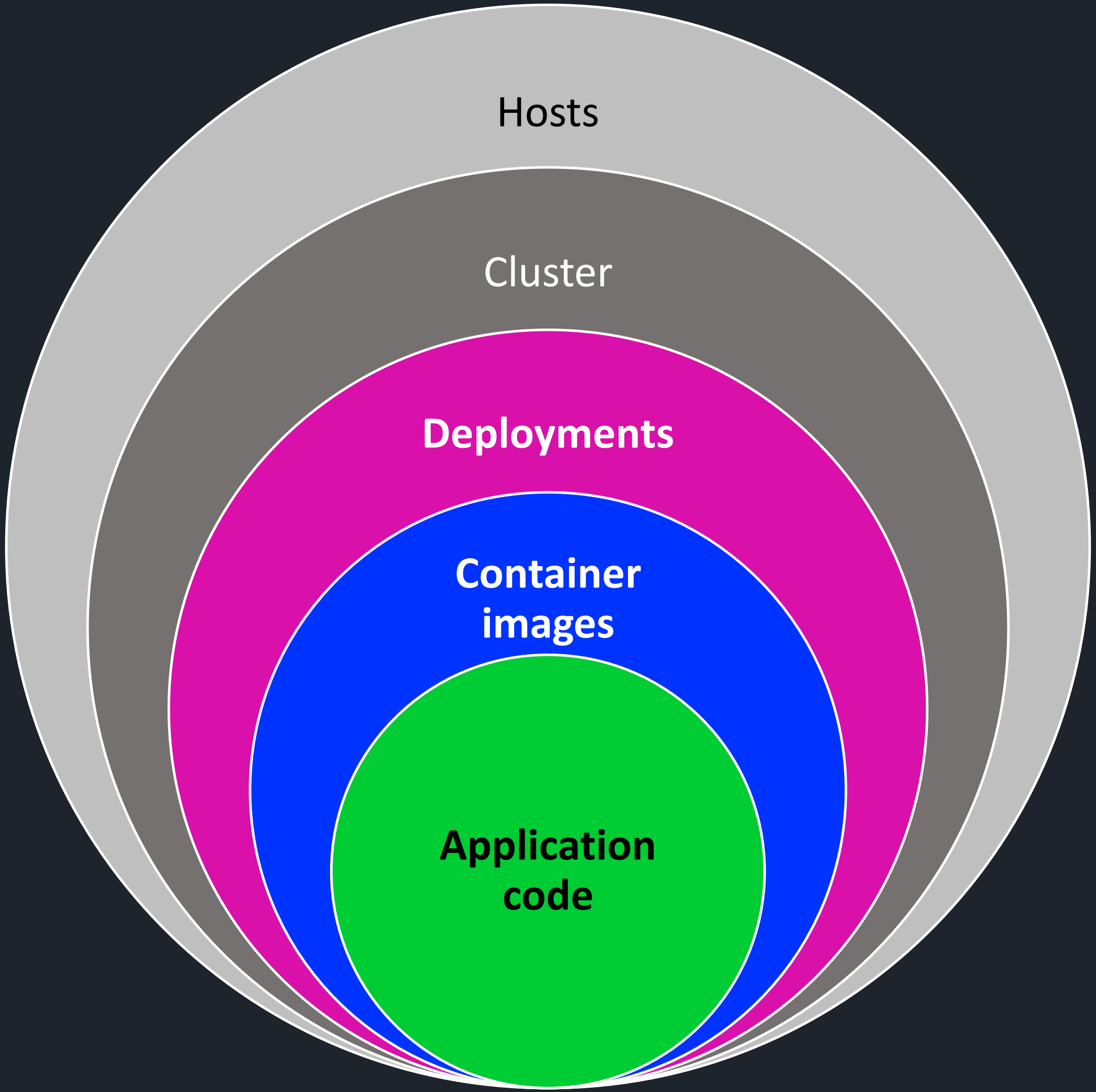
SECURITY

Least privilege

- Don’t grant privileges unless needed
- Reduce blast radius

KUBERNETES

Layers



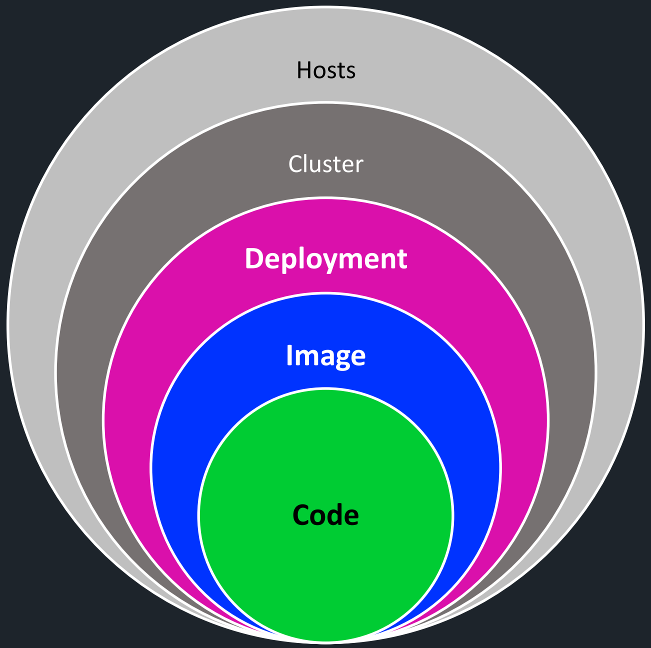
Code

Images

Deployment

Maintenance

Wrap-up



CODE

Threat modeling

- What are we protecting?
- Why are we protecting it?
- How might it be compromised?
- What happens if we fail to protect it?
- How will we react/respond and move on?

SCHLAGE
READY
Recognition Systems
HandKey II

Master
Master

CODE

Threat modeling

- STRIDE
- OWASP Threat Dragon
- PASTA
- CAIRIS
- Threagile

CODE

SAST

Static application security testing

Scan our source code

- Look for risky/dangerous practices
- Memory leaks
- SQL injections
- Race conditions
- Untrusted inputs
- Unfiltered outputs

SAST

Semgrep

Semgrep

- Supports 30+ languages
- Python, Docker, and cloud versions
- Code stays local in all three

```
$ pip install -U semgrep
$ semgrep scan --config auto
```

```
$ docker pull returntocorp/semgrep
$ docker run --rm –v "$(pwd):/src" \
    returntocorp/semgrep \
    semgrep scan --config auto
```

Static code analysis

Semgrep

```
$ semgrep scan --config auto
…
┌──────────────────┐
│ 29 Code Findings │
└──────────────────┘

    app/routes/contributions.js
       javascript.browser.security.eval-detected.eval-detected
          Detected the use of eval(). eval() can be dangerous if used to evaluate dynamic content. If
          this content can be input from outside the program, this may be a code injection
          vulnerability. Ensure evaluated content is not definable by external sources.
          Details: https://sg.run/7ope

           32┆ const preTax = eval(req.body.preTax);
            ⋮┆----------------------------------------
           33┆ const afterTax = eval(req.body.afterTax);
            ⋮┆----------------------------------------
           34┆ const roth = eval(req.body.roth);
            ⋮┆----------------------------------------
       javascript.lang.security.audit.code-string-concat.code-string-concat
          Found data from an Express or Next web request flowing to `eval`. If this data is user-
          controllable this can lead to execution of arbitrary system commands in the context of your
          application process. Avoid `eval` whenever possible.
          Details: https://sg.run/96Yk
```

CODE

SCA

Software composition analysis

Scan our dependencies

- and their transitive dependencies
- 6/7 vulns come from transitive dependencies

SCA

Trivy

- Filesystems
- Git repos
- Container images

```
$ docker pull aquasec/trivy
$ docker run --rm \
    -v "$(pwd):/work" \
    -workdir /work \
    aquasec/trivy \
    filesystem .
```

SCA

```
$ trivy filesystem .
…
package-lock.json (npm)
=======================
Total: 39 (UNKNOWN: 0, LOW: 2, MEDIUM: 9, HIGH: 21, CRITICAL: 7)

┌──────────────────────┬─────────────────────┬──────────┬────────┬───────────────────┬───────────────
│       Library        │    Vulnerability    │ Severity │ Status │ Installed Version │ Fixed Version
├──────────────────────┼─────────────────────┼──────────┼────────┼───────────────────┼───────────────
│ bson                 │ CVE-2020-7610       │ CRITICAL │ fixed  │ 1.0.9             │ 1.1.4
│                      │                     │          │        │                   │
│                      │                     │          │        │                   │
│                      ├─────────────────────┼──────────┤        │                   │
│                      │ CVE-2019-2391       │ MEDIUM   │        │                   │
│                      │                     │          │        │                   │
│                      │                     │          │        │                   │
├──────────────────────┼─────────────────────┼──────────┤        ├───────────────────┼───────────────
│ decode-uri-component │ CVE-2022-38900      │ HIGH     │        │ 0.2.0             │ 0.2.1
│                      │                     │          │        │                   │
├──────────────────────┼─────────────────────┤          │        ├───────────────────┼───────────────
│ glob-parent          │ CVE-2020-28469      │          │        │ 3.1.0             │ 5.1.2
│                      │                     │          │        │                   │
├──────────────────────┼─────────────────────┼──────────┤        ├───────────────────┼───────────────
│ helmet-csp           │ GHSA-c3m8-x3cg-qm2c │ MEDIUM   │        │ 1.2.2             │ 2.9.1
```

SCA

Grype

- Filesystems
- Container images
- Finds some different vulns than Trivy

```
$ docker pull anchore/grype
$ docker run --rm \
    -v "$(pwd):/work" \
    -workdir /work \
    anchore/grype \
    dir:.
```

SCA

```
$ grype dir:.
NAME                  INSTALLED  FIXED-IN  TYPE  VULNERABILITY        SEVERITY
adm-zip               0.4.4      0.4.11    npm   GHSA-3v6h-hqm4-2rg6  Medium
ajv                   6.10.0     6.12.3    npm   GHSA-v88g-cgmw-v5xw  Medium
ansi-regex            3.0.0      3.0.1     npm   GHSA-93q8-gq69-wqmw  High
async                 2.6.1      2.6.4     npm   GHSA-fwr7-v2mv-hh25  High
bl                    1.0.3      1.2.3     npm   GHSA-pp7h-53gx-mx7r  Medium
bl                    1.1.2      1.2.3     npm   GHSA-pp7h-53gx-mx7r  Medium
brace-expansion       1.1.6      1.1.7     npm   GHSA-832h-xg76-4gv6  High
braces                1.8.5      2.3.1     npm   GHSA-g95f-p29q-9xw4  Low
braces                1.8.5      2.3.1     npm   GHSA-cwfw-4gq5-mrqx  Low
bson                  1.0.9      1.1.4     npm   GHSA-v8w9-2789-6hhr  Critical
bson                  1.0.9      1.1.4     npm   GHSA-4jwp-vfvf-657p  Medium
chownr                1.0.1      1.1.0     npm   GHSA-c6rq-rjc2-86v2  Low
cryptiles             0.2.2      4.1.2     npm   GHSA-rq8g-5pc5-wrhr  Critical
cryptiles             2.0.5      4.1.2     npm   GHSA-rq8g-5pc5-wrhr  Critical
debug                 2.2.0      2.6.9     npm   GHSA-9vvw-cc9w-f27h  High
debug                 2.2.0      2.6.9     npm   GHSA-gxpj-cx7g-858c  Medium
decode-uri-component  0.2.0      0.2.1     npm   GHSA-w573-4hg7-7wgq  High
diff                  1.4.0      3.5.0     npm   GHSA-h6ch-v84p-w6p9  High
dot-prop              4.2.0      4.2.1     npm   GHSA-ff7x-qrg7-qggm  High
extend                3.0.0      3.0.2     npm   GHSA-qrmc-fj45-qfc2  Medium
fsevents              1.2.9      1.2.11    npm   GHSA-xv2f-5jw4-v95m  Critical
fstream               1.0.10     1.0.12    npm   GHSA-xf7w-r453-m56c  High
```

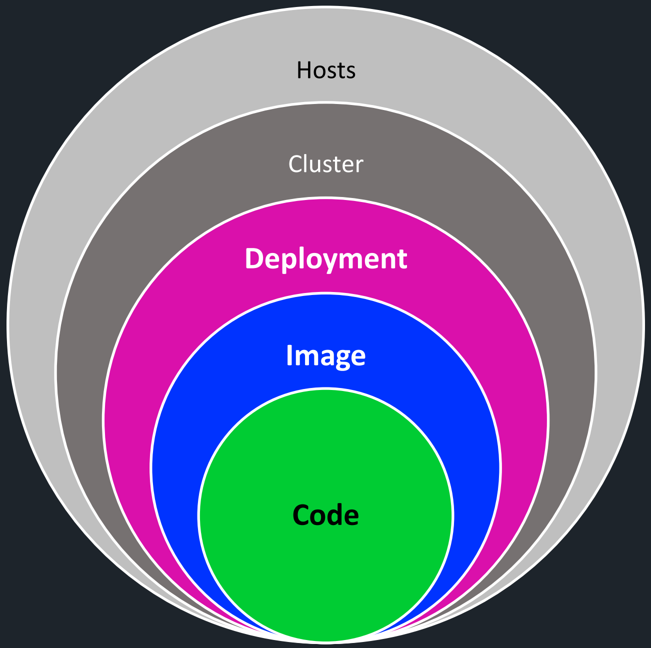
Code

Image

Deployment

Maintenance

Wrap-up



IMAGE

Base images

- Include the minimal supporting software
- Reduce the blast radius

Base image choices

- scratch (nothing but the application)
- Distroless (minimal supporting files)
- Small image (Alpine or BusyBox)
- Minimal Linux (Slim or UBI Micro)
- Anything else (*you have made a mistake*)

IMAGE

Automated builds

- Builds should be repeatable and reliable
- That means automated
- Dockerfile and/or pipeline
- GitHub Actions
- GitLab CI/CD
- Infrastructure-as-code (IaC)

IaC analysis

Checkov

- Dockerfiles
- Kubernetes manifests

```
$ pip install -U checkov
$ checkov -d .
```

```
$ docker pull bridgecrew/checkov
$ docker run --rm --tty \
    -v "$(pwd):/work" \
    -workdir /work \
    bridgecrew/checkov \
    checkov -d .
```

IaC analysis

checkov
by bridgecrew

```
$ checkov -d . --quiet --compact

dockerfile scan results:

Passed checks: 57, Failed checks: 1, Skipped checks: 0

Check: CKV_DOCKER_2: "Ensure that HEALTHCHECK instructions have been added to container images"
        FAILED for resource: /Dockerfile.
        File: /Dockerfile:1-18
        Guide: https://docs.paloaltonetworks.com/content/techdocs/en_US/prisma/prisma-cloud/prisma-cloud-code-security-policy-reference/docker-policies/docker-policy-index/ensure-that-healthcheck-instructions-have-been-added-to-container-images.html

github_actions scan results:

Passed checks: 56, Failed checks: 2, Skipped checks: 0

Check: CKV2_GHA_1: "Ensure top-level permissions are not set to write-all"
        FAILED for resource: on(E2E Test)
        File: /.github/workflows/e2e-test.yml:0-1
Check: CKV2_GHA_1: "Ensure top-level permissions are not set to write-all"
        FAILED for resource: on(Lint)
        File: /.github/workflows/lint.yml:0-1
```

SCA

aqua trivy

```
$ trivy image nodegoat:dev
…
nodegoat:dev (alpine 3.15.4)
============================
Total: 21 (UNKNOWN: 0, LOW: 0, MEDIUM: 12, HIGH: 8, CRITICAL: 1)

┌──────────────┬────────────────┬──────────┬────────┬───────────────────┬───────────────┬───────────
│   Library    │ Vulnerability  │ Severity │ Status │ Installed Version │ Fixed Version │  Title
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼───────────
│ libcrypto1.1 │ CVE-2022-4450  │ HIGH     │ fixed  │ 1.1.1n-r0         │ 1.1.1t-r0     │ double free af
│              │                │          │        │                   │               │ https://avd.aq
│              ├────────────────┤          │        │                   │               ├───────────
│              │ CVE-2023-0215  │          │        │                   │               │ use-after-free
│              │                │          │        │                   │               │ https://avd.aq
│              ├────────────────┤          │        │                   │               ├───────────
│              │ CVE-2023-0286  │          │        │                   │               │ X.400 address
│              │                │          │        │                   │               │ https://avd.aq
│              ├────────────────┤          │        │                   ├───────────────┼───────────
│              │ CVE-2023-0464  │          │        │                   │ 1.1.1t-r2     │ Denial of serv
│              │                │          │        │                   │               │ X509 policy co
│              │                │          │        │                   │               │ https://avd.aq
│              ├────────────────┼──────────┤        │                   ├───────────────┼───────────
│              │ CVE-2022-2097  │ MEDIUM   │        │                   │ 1.1.1q-r0     │ AES OCB fails
│              │                │          │        │                   │               │ https://avd.aq
```

SCA

```
$ grype docker:nodegoat:dev
NAME                    INSTALLED   FIXED-IN    TYPE  VULNERABILITY        SEVERITY
ansi-regex              3.0.0       3.0.1       npm   GHSA-93q8-gq69-wqmw  High
ansi-regex              4.1.0       4.1.1       npm   GHSA-93q8-gq69-wqmw  High
bson                    1.0.9       1.1.4       npm   GHSA-v8w9-2789-6hhr  Critical
bson                    1.0.9       1.1.4       npm   GHSA-4jwp-vfvf-657p  Medium
busybox                 1.34.1-r5               apk   CVE-2022-48174       Critical
debug                   2.2.0       2.6.9       npm   GHSA-9vvw-cc9w-f27h  High
debug                   2.2.0       2.6.9       npm   GHSA-gxpj-cx7g-858c  Medium
decode-uri-component    0.2.0       0.2.1       npm   GHSA-w573-4hg7-7wgq  High
glob-parent             3.1.0       5.1.2       npm   GHSA-ww39-953v-wcq6  High
got                     6.7.1       11.8.5      npm   GHSA-pfrx-2q88-qq97  Medium
helmet-csp              1.2.2       2.9.1       npm   GHSA-c3m8-x3cg-qm2c  Medium
http-cache-semantics    3.8.1       4.1.1       npm   GHSA-rc47-6667-2j5j  High
i                       0.3.6       0.3.7       npm   GHSA-x55w-vjjp-222r  High
ini                     1.3.5       1.3.6       npm   GHSA-qqgx-2p2h-9c37  High
kind-of                 6.0.2       6.0.3       npm   GHSA-6c8f-qphg-qjgp  High
libcrypto1.1            1.1.1n-r0   1.1.1t-r2   apk   CVE-2023-0464        High
libcrypto1.1            1.1.1n-r0   1.1.1t-r0   apk   CVE-2023-0286        High
libcrypto1.1            1.1.1n-r0   1.1.1t-r0   apk   CVE-2023-0215        High
libcrypto1.1            1.1.1n-r0   1.1.1t-r0   apk   CVE-2022-4450        High
libcrypto1.1            1.1.1n-r0   1.1.1v-r0   apk   CVE-2023-3817        Medium
libcrypto1.1            1.1.1n-r0   1.1.1u-r2   apk   CVE-2023-3446        Medium
libcrypto1.1            1.1.1n-r0   1.1.1u-r0   apk   CVE-2023-2650        Medium
```

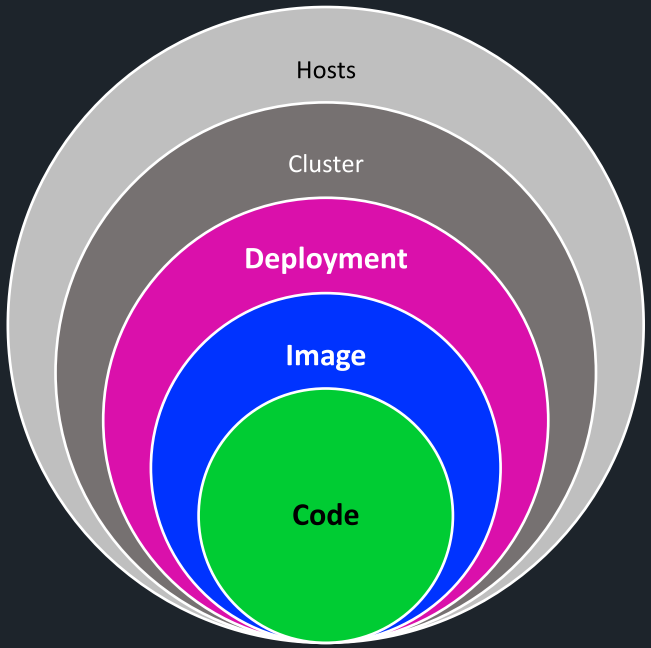
Code

Image

Deployment

Maintenance

Wrap-up



DEPLOYMENT

Recommended practices

Principle of least privilege

- Prevent privileged containers
- Require the file system to be read-only

Protect the image supply chain

- Use a specific version of an image

Ensure availability

- Set memory and CPU requests/limits
- Liveness and readiness probes

IaC analysis

checkov

by bridgecrew

```
$ checkov -d . --quiet --compact

kubernetes scan results:

Passed checks: 1066, Failed checks: 180, Skipped checks: 0

Check: CKV_K8S_20: "Containers should not run with allowPrivilegeEscalation"
        FAILED for resource: Deployment.sock-shop.front-end
        File: /09-front-end-dep.yaml:2-52
        Guide: https://docs.paloaltonetworks.com/content/techdocs/en_US/prisma/prisma-cloud/prisma-cloud-
code-security-policy-reference/kubernetes-policies/kubernetes-policy-index/bc-k8s-19.html
Check: CKV_K8S_43: "Image should use digest"
        FAILED for resource: Deployment.sock-shop.front-end
        File: /09-front-end-dep.yaml:2-52
        Guide: https://docs.paloaltonetworks.com/content/techdocs/en_US/prisma/prisma-cloud/prisma-cloud-
code-security-policy-reference/kubernetes-policies/kubernetes-policy-index/bc-k8s-39.html
Check: CKV_K8S_38: "Ensure that Service Account Tokens are only mounted where necessary"
        FAILED for resource: Deployment.sock-shop.front-end
        File: /09-front-end-dep.yaml:2-52
        Guide: https://docs.paloaltonetworks.com/content/techdocs/en_US/prisma/prisma-cloud/prisma-cloud-
code-security-policy-reference/kubernetes-policies/kubernetes-policy-index/bc-k8s-35.html
Check: CKV_K8S_29: "Apply security context to your pods and containers"
        FAILED for resource: Deployment.sock-shop.front-end
        File: /09-front-end-dep.yaml:2-52
```

Code

Image

Deployment

Maintenance

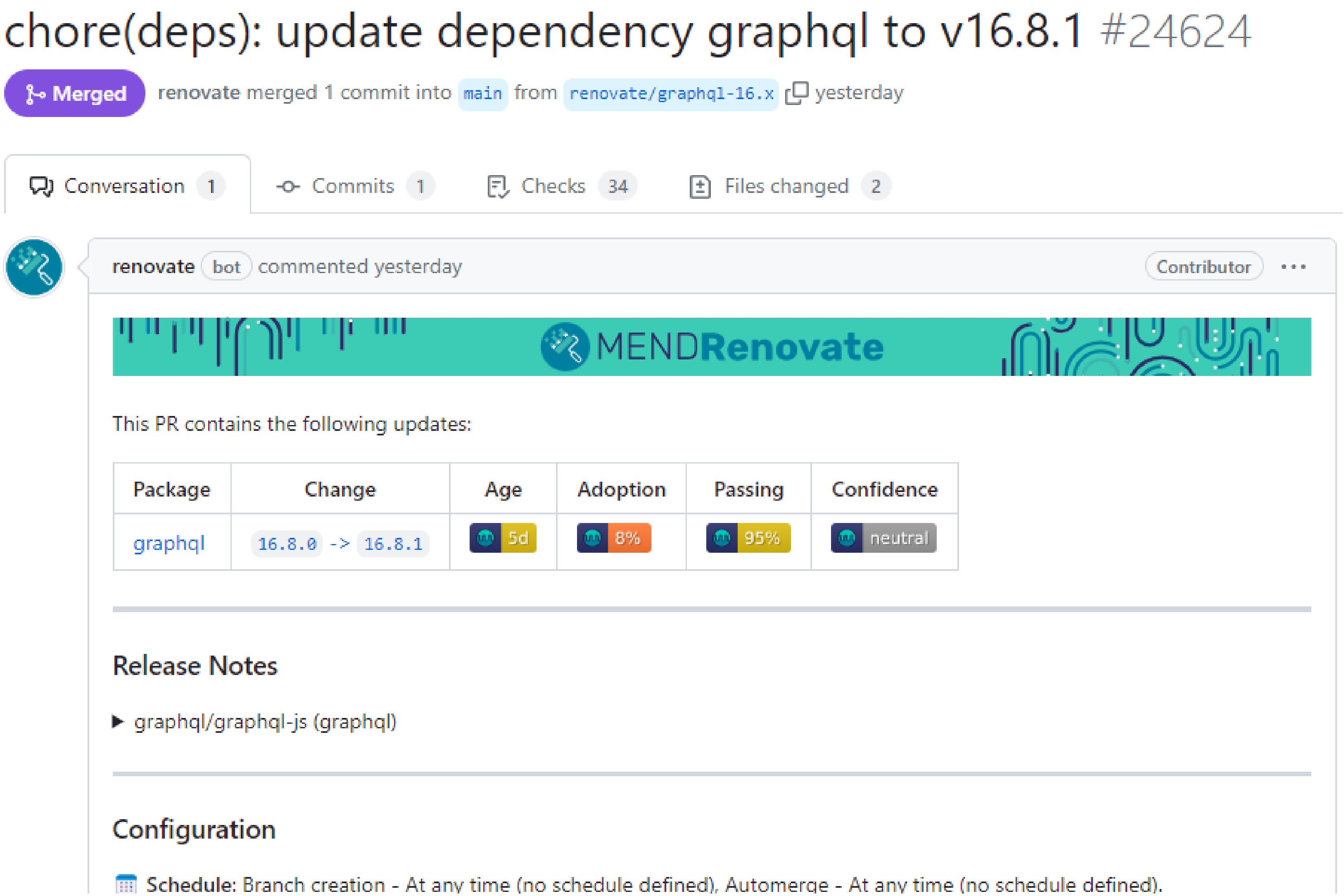
Wrap-up

Dependency updates

Renovate

- Checks dependencies and transitive deps
- Checks base images
- Runs as GitHub Action
- Runs in GitLab CI/CD
- Creates PRs for available updates
- Can automerge (e.g., high test coverage)

Dependency updates



chore(deps): update dependency graphql to v16.8.1 #24624

Merged renovate merged 1 commit into `main` from `renovate/graphql-16.x` yesterday

renovate bot commented yesterday

This PR contains the following updates:

Package	Change	Age	Adoption	Passing	Confidence
graphql	`16.8.0` -> `16.8.1`	5d	8%	95%	neutral

Release Notes

▶ graphql/graphql-js (graphql)

Configuration

📅 **Schedule**: Branch creation - At any time (no schedule defined), Automerge - At any time (no schedule defined).

MAINTENANCE

Frequent builds

- Latest patches
- Latest base images
- Frequent pipeline scans for vulnerabilities
- Repeated testing

- It's automated anyway, so why not?

MAINTENANCE

Clean code

- Keep code quality high
  - You are scanning anyway
- Use a consistent style
  - Fewer mistakes
  - Fewer misunderstandings
- Easier code reviews
  - Can focus on content, not style

Code

Image

Deployment

Maintenance

Wrap-up

WRAP-UP

Key takeaways

- Scan your code.
- Scan your dependencies and keep them updated.
- Use the smallest base image you can.
- Scan your images and keep them updated.
- Use automation and scan your IaC.
- Rebuild frequently and keep everything updated.

WRAP-UP

Single biggest win

Keep everything up-to-date.

WRAP-UP

Tools

OWASP NodeGoat: https://github.com/OWASP/NodeGoat

Semgrep: https://github.com/returntocorp/semgrep

Aqua Security Trivy: https://github.com/aquasecurity/trivy

Anchore Grype: https://github.com/anchore/grype

Checkov by Bridgecrew: https://github.com/bridgecrewio/checkov

Google Distroless:
https://github.com/GoogleContainerTools/distroless

Chainguard Distroless: https://github.com/chainguard-images

Sock Shop:
https://github.com/microservices-demo/microservices-demo

Renovate: https://github.com/renovatebot/renovate

WRAP-UP

Threat modeling

STRIDE

https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats

OWASP Threat Dragon

https://www.threatdragon.com/

PASTA

https://versprite.com/blog/what-is-pasta-threat-modeling/

CAIRIS

https://cairis.org/

Threagile

https://threagile.io/

WRAP-UP

More talks and info

Keeping Your Kubernetes Cluster Secure
Trivy and Grype demos
https://www.youtube.com/@otherdevopsgene

Kubernetes tool wrappers
https://github.com/OtherDevOpsGene/k8s-tool-wrappers

GitGuardian Blog: Always Be Updating
https://blog.gitguardian.com/always-be-updating/

WRAP-UP

Next talk

Castle Defense 101 (aka Threat Modeling)

Thursday at 2:35 pm in Aloeswood

Questions?

Gene Gotimer

Principal DevOps Engineer at Praeses, LLC

@OtherDevOpsGene