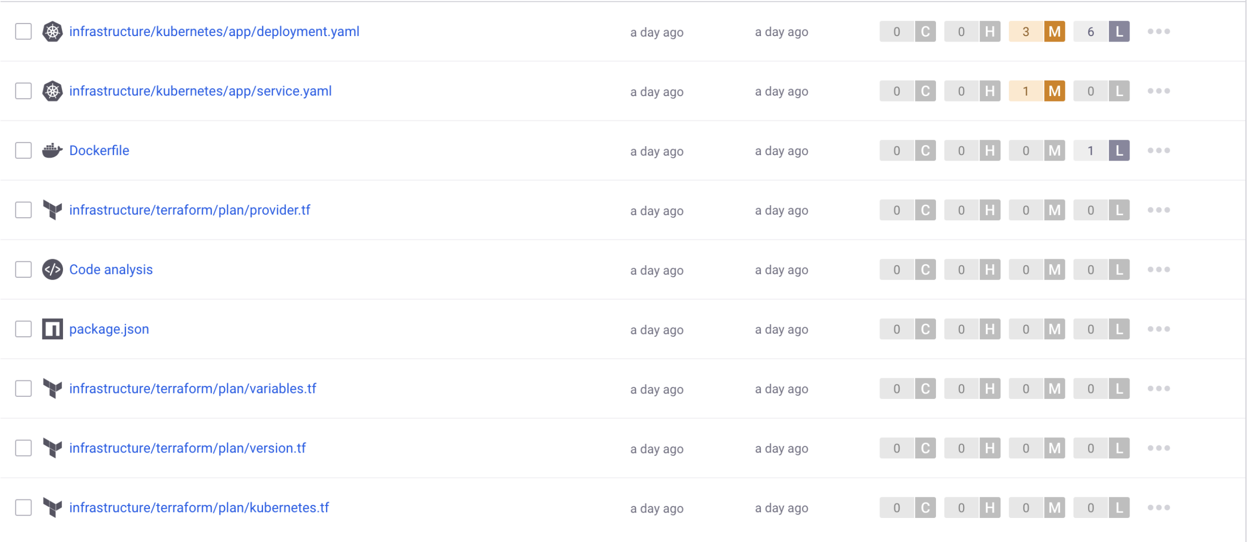
Security Audit - Hackathon 2024

Code Analysis

We've audited the full repository using Snyk code.

The tool found several vulnerabilities in the code with mostly medium severity.

Here is the result:



□	infrastructure/kubernetes/app/deployment.yaml	a day ago	a day ago	0 C	0 H	3 M	6 L	•••
□	infrastructure/kubernetes/app/service.yaml	a day ago	a day ago	0 C	0 H	1 M	0 L	•••
□	Dockerfile	a day ago	a day ago	0 C	0 H	0 M	1 L	•••
□	infrastructure/terraform/plan/provider.tf	a day ago	a day ago	0 C	0 H	0 M	0 L	•••
□	Code analysis	a day ago	a day ago	0 C	0 H	0 M	0 L	•••
□	package.json	a day ago	a day ago	0 C	0 H	0 M	0 L	•••
□	infrastructure/terraform/plan/variables.tf	a day ago	a day ago	0 C	0 H	0 M	0 L	•••
□	infrastructure/terraform/plan/version.tf	a day ago	a day ago	0 C	0 H	0 M	0 L	•••
□	infrastructure/terraform/plan/kubernetes.tf	a day ago	a day ago	0 C	0 H	0 M	0 L	•••

TypeScript Code

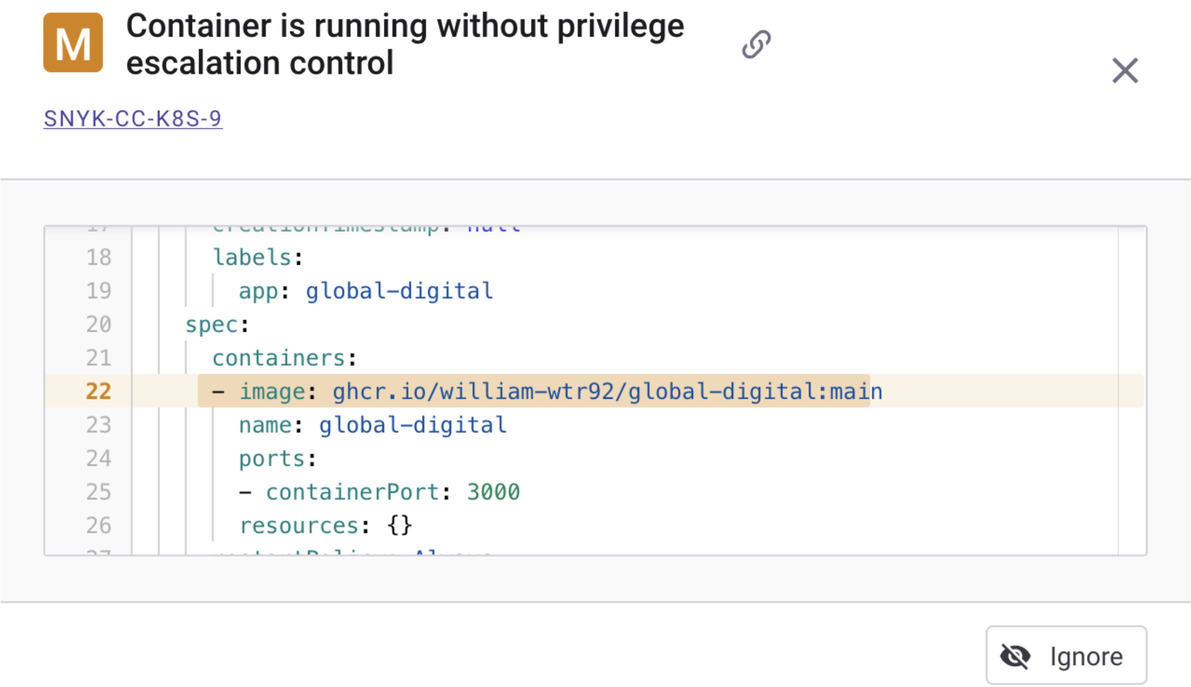
No vulnerabilities have been found in the code during the test.

Config & Package files

All packages are up-to-date and no vulnerabilities have been found in the package.json file.

Three medium vulnerabilities have been detected in the Kubernetes deployment file (`infrastructure/kubernetes/app/service.yaml`):

- The first one means that processes could elevate current privileges via known vectors, for example SUID binaries

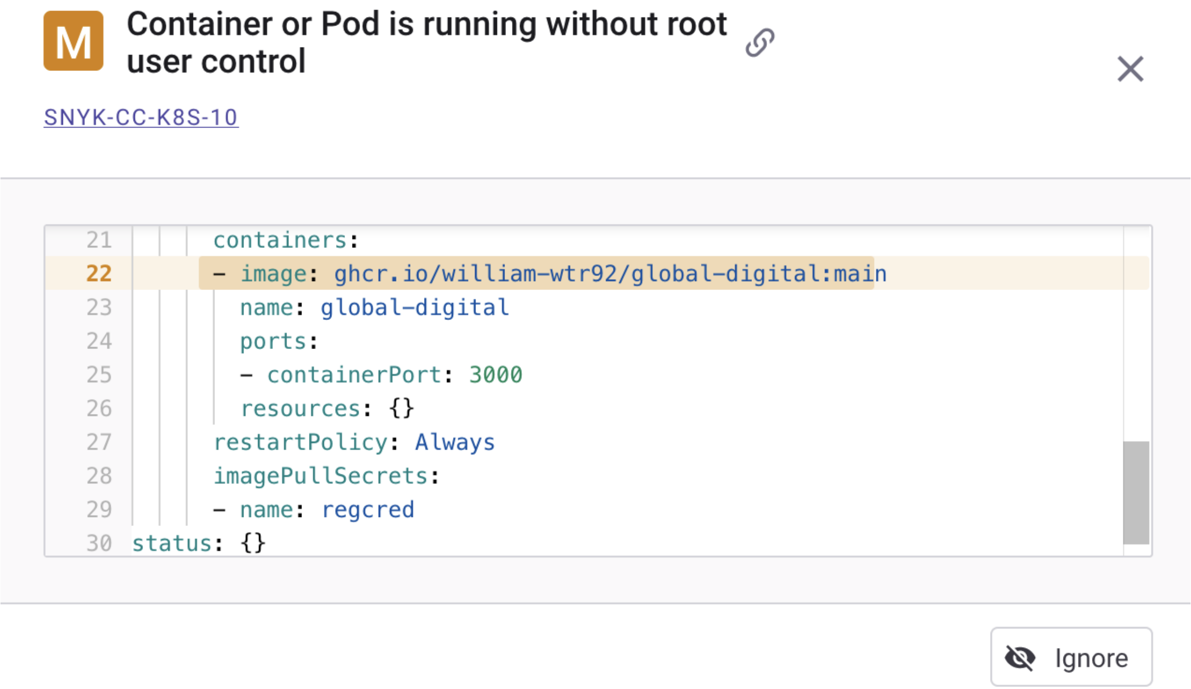


Container is running without privilege escalation control

SNYK-CC-K8S-9

```
      creationTimestamp: null
      labels:
        app: global-digital
    spec:
      containers:
      - image: ghcr.io/william-wtr92/global-digital:main
        name: global-digital
        ports:
        - containerPort: 3000
        resources: {}
      restartPolicy: Always
```

- The second one shows us that container could be running with full administrative privileges

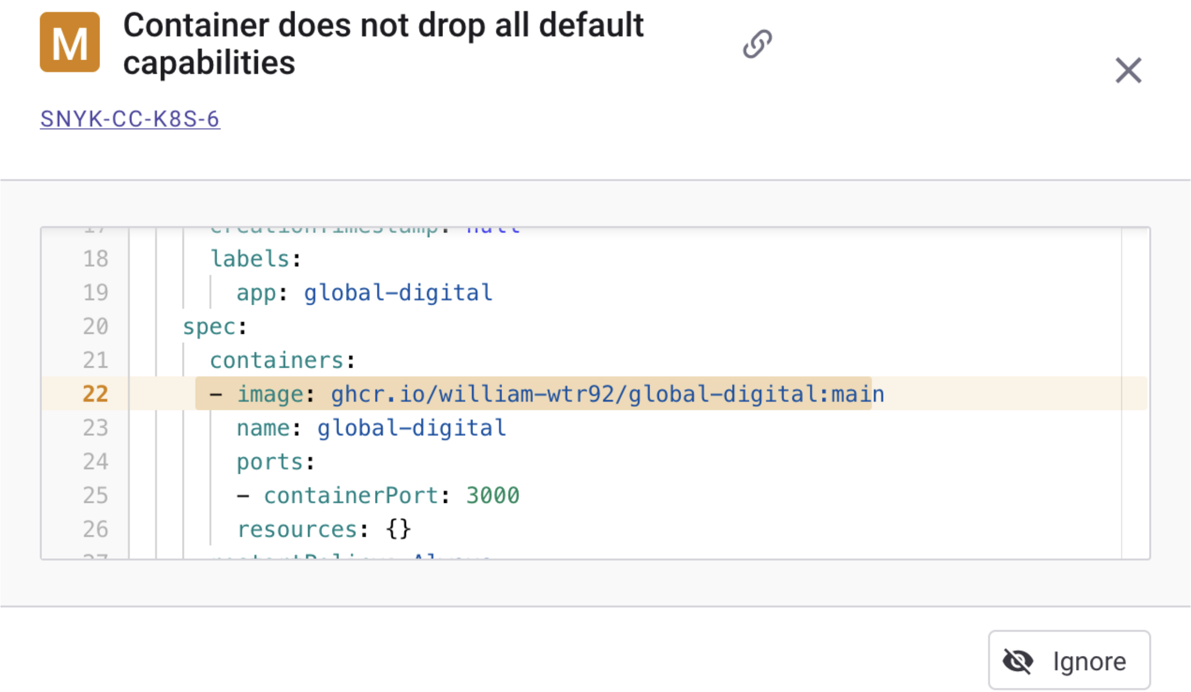


Container or Pod is running without root user control

SNYK-CC-K8S-10

```
      containers:
      - image: ghcr.io/william-wtr92/global-digital:main
        name: global-digital
        ports:
        - containerPort: 3000
        resources: {}
      restartPolicy: Always
      imagePullSecrets:
      - name: regcred
status: {}
```

- The third one informs us that containers are running with potentially unnecessary privileges



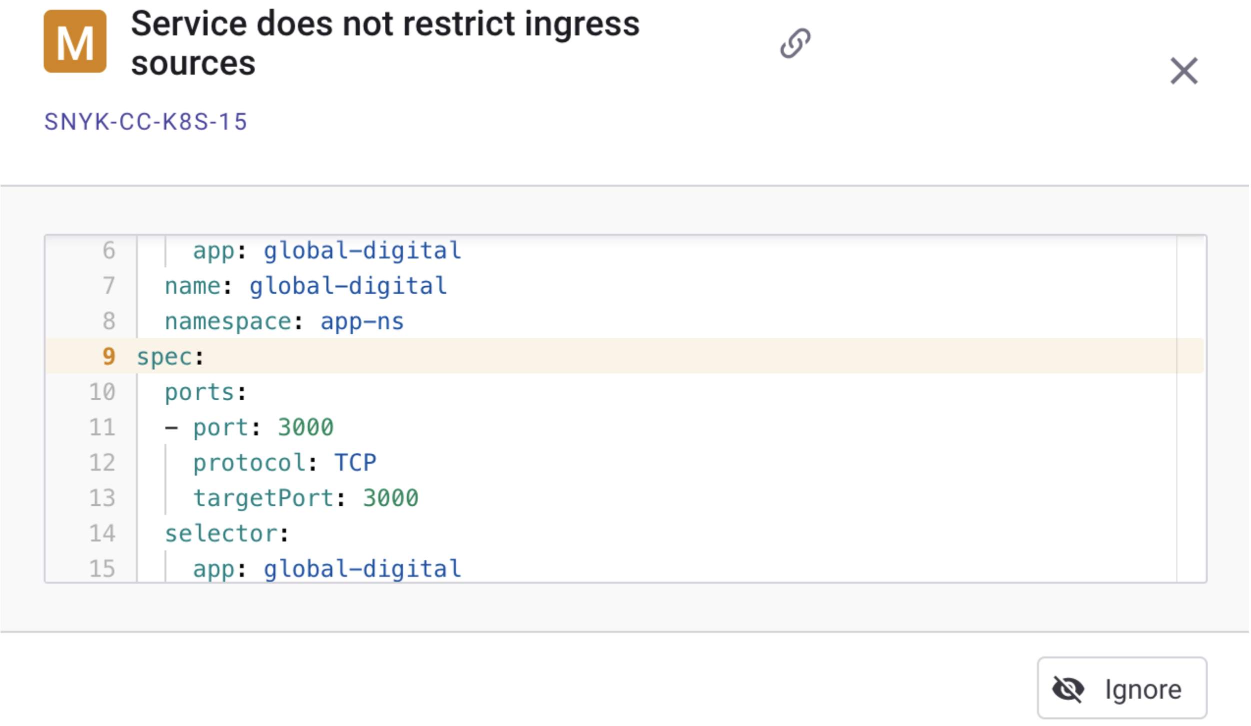
Container does not drop all default capabilities

SNYK-CC-K8S-6

```
  creationTimestamp: null
  labels:
    app: global-digital
spec:
  containers:
  - image: ghcr.io/william-wtr92/global-digital:main
    name: global-digital
    ports:
    - containerPort: 3000
    resources: {}
  restartPolicy: Always
```

Finally, one medium vulnerability was found in the Kubernetes service file (`infrastructure/kubernetes/app/service.yaml`)**:**

It warns us that defining a Load balancer Service without setting the **loadBalancerSourceRanges** property will use the default value of `0.0.0.0/0`. Therefore, this allows access to any traffic to the Node Security Group(s), potentially meaning everyone can access your service.



Service does not restrict ingress sources

SNYK-CC-K8S-15

```yaml
    app: global-digital
  name: global-digital
  namespace: app-ns
spec:
  ports:
  - port: 3000
    protocol: TCP
    targetPort: 3000
  selector:
    app: global-digital
```