StackRox

CNCF Webinar Series

Kubernetes Security Best Practices

Connor Gorman, Principal Engineer, StackRox

11 March 2020

CLOUD NATIVE
COMPUTING FOUNDATION

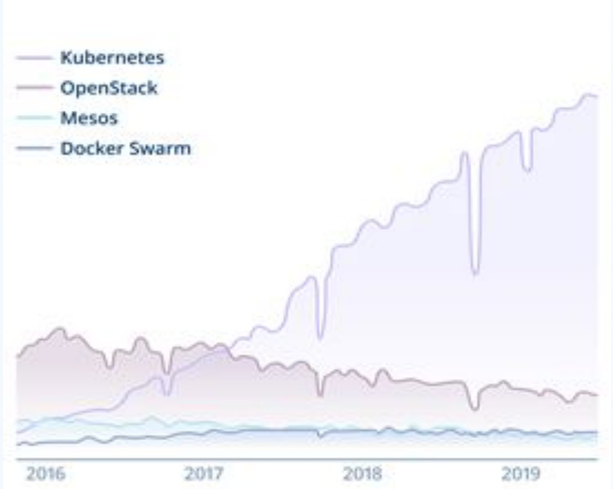
What we'll cover

- General Kubernetes hygiene
- Workload best practices
- Demo
- Questions?

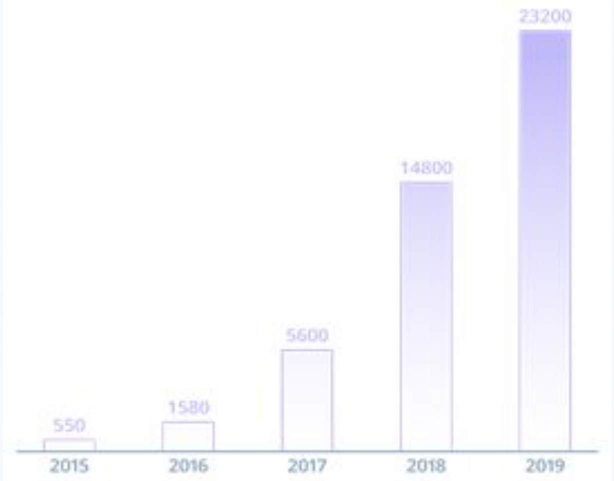
CLOUD NATIVE COMPUTING FOUNDATION StackRox

What are we doing here?

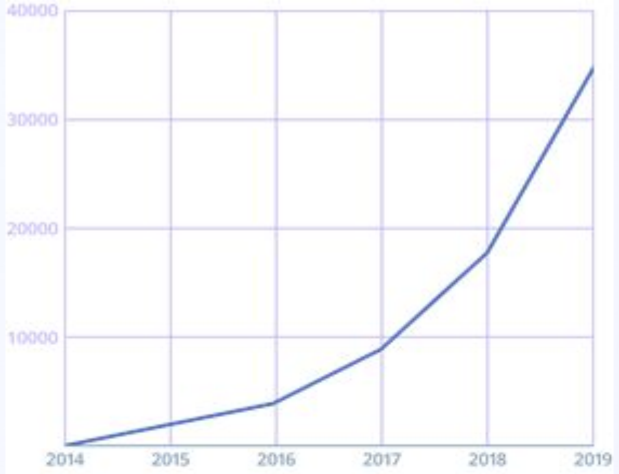
Scratch that … Kubernetes is here!



Google Trends



KubeCon Attendees

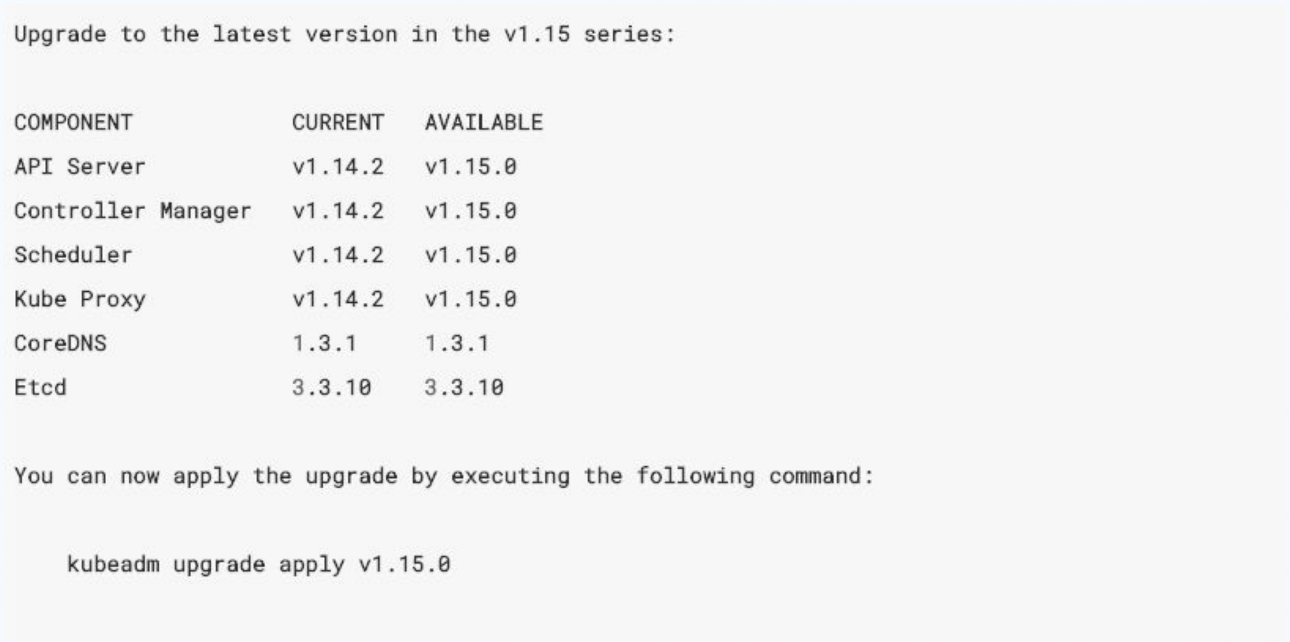


Cumulative Growth of Contributors

Kubernetes Hygiene

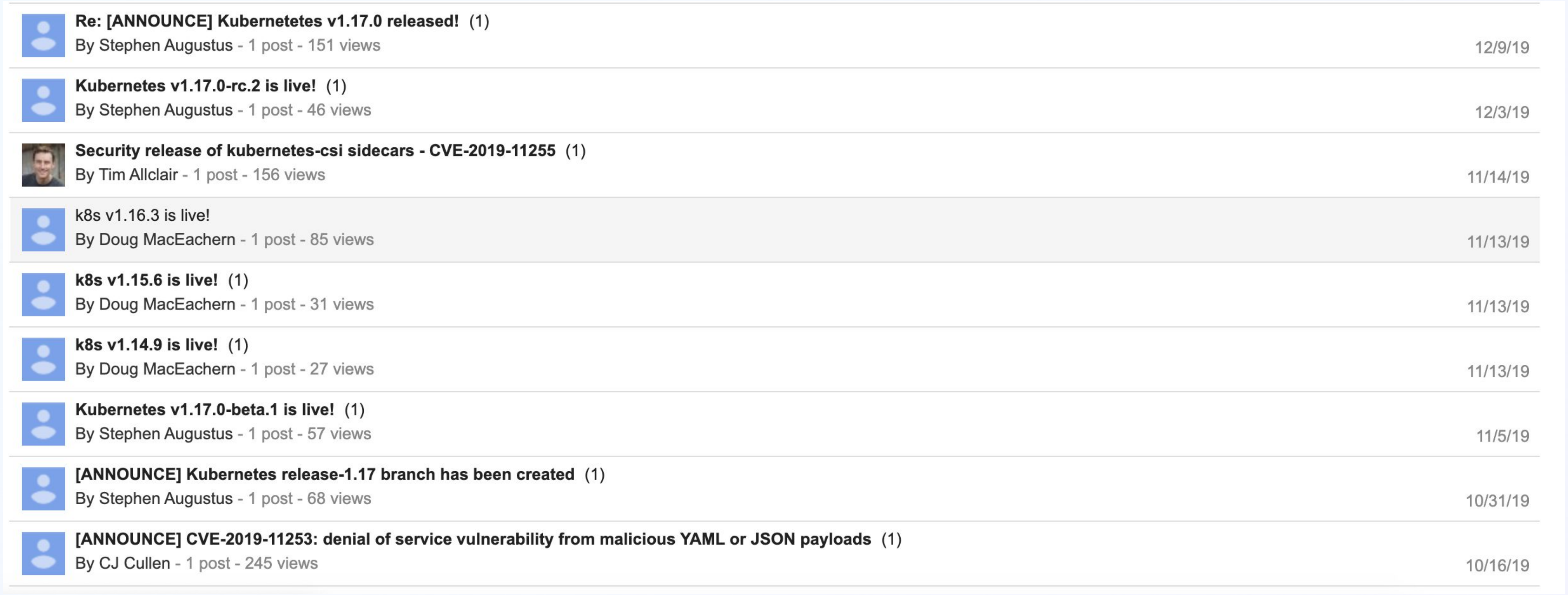
CLOUD NATIVE COMPUTING FOUNDATION StackRox

Upgrade to a current version!



```
Upgrade to the latest version in the v1.15 series:

COMPONENT            CURRENT   AVAILABLE
API Server           v1.14.2   v1.15.0
Controller Manager   v1.14.2   v1.15.0
Scheduler            v1.14.2   v1.15.0
Kube Proxy           v1.14.2   v1.15.0
CoreDNS              1.3.1     1.3.1
Etcd                 3.3.10    3.3.10

You can now apply the upgrade by executing the following command:

    kubeadm upgrade apply v1.15.0
```

Kubernetes-Announce Google Group



Re: [ANNOUNCE] Kubernetetes v1.17.0 released! (1)
By Stephen Augustus - 1 post - 151 views 12/9/19

Kubernetes v1.17.0-rc.2 is live! (1)
By Stephen Augustus - 1 post - 46 views 12/3/19

Security release of kubernetes-csi sidecars - CVE-2019-11255 (1)
By Tim Allclair - 1 post - 156 views 11/14/19

k8s v1.16.3 is live!
By Doug MacEachern - 1 post - 85 views 11/13/19

k8s v1.15.6 is live! (1)
By Doug MacEachern - 1 post - 31 views 11/13/19

k8s v1.14.9 is live! (1)
By Doug MacEachern - 1 post - 27 views 11/13/19

Kubernetes v1.17.0-beta.1 is live! (1)
By Stephen Augustus - 1 post - 57 views 11/5/19

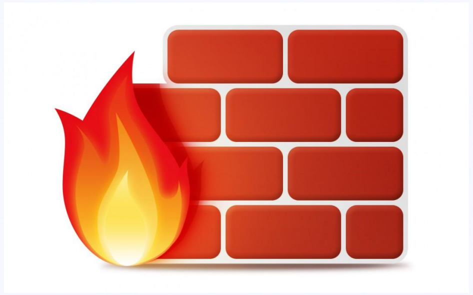
[ANNOUNCE] Kubernetes release-1.17 branch has been created (1)
By Stephen Augustus - 1 post - 68 views 10/31/19

[ANNOUNCE] CVE-2019-11253: denial of service vulnerability from malicious YAML or JSON payloads (1)
By CJ Cullen - 1 post - 245 views 10/16/19

Harden Node Security

Control network access to sensitive ports.

Make sure that your network restricts access to ports used by kubelet, including 10250 and 10255. Consider limiting access to the Kubernetes API server except from trusted networks.



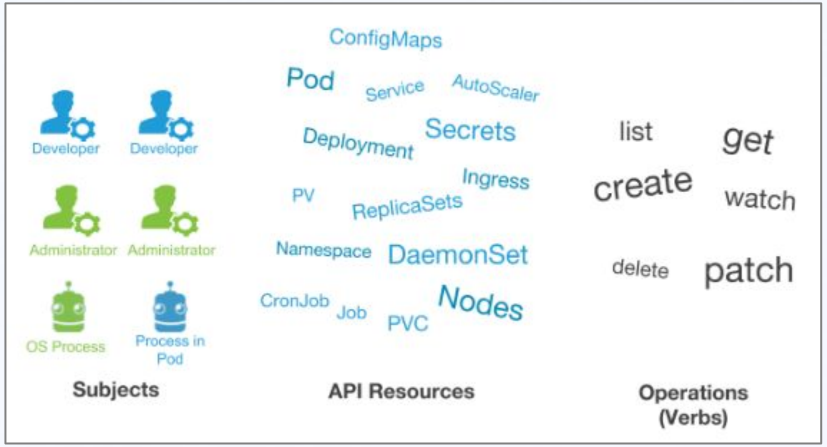
Harden Node Security

Minimize administrative access to Kubernetes nodes.

Access to the nodes in your cluster should generally be restricted — debugging and other tasks can usually be handled without direct access to the node.

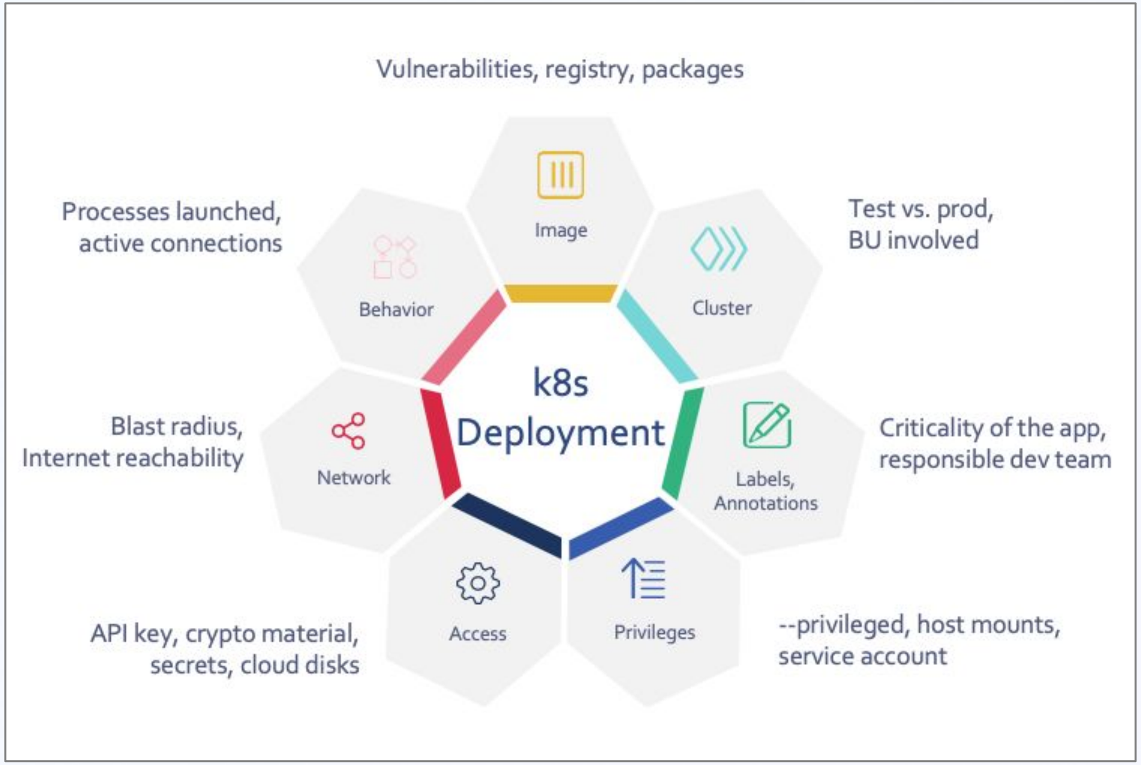
Enable Role-Based Access Control

Control who can access the Kubernetes API and what permissions they have.

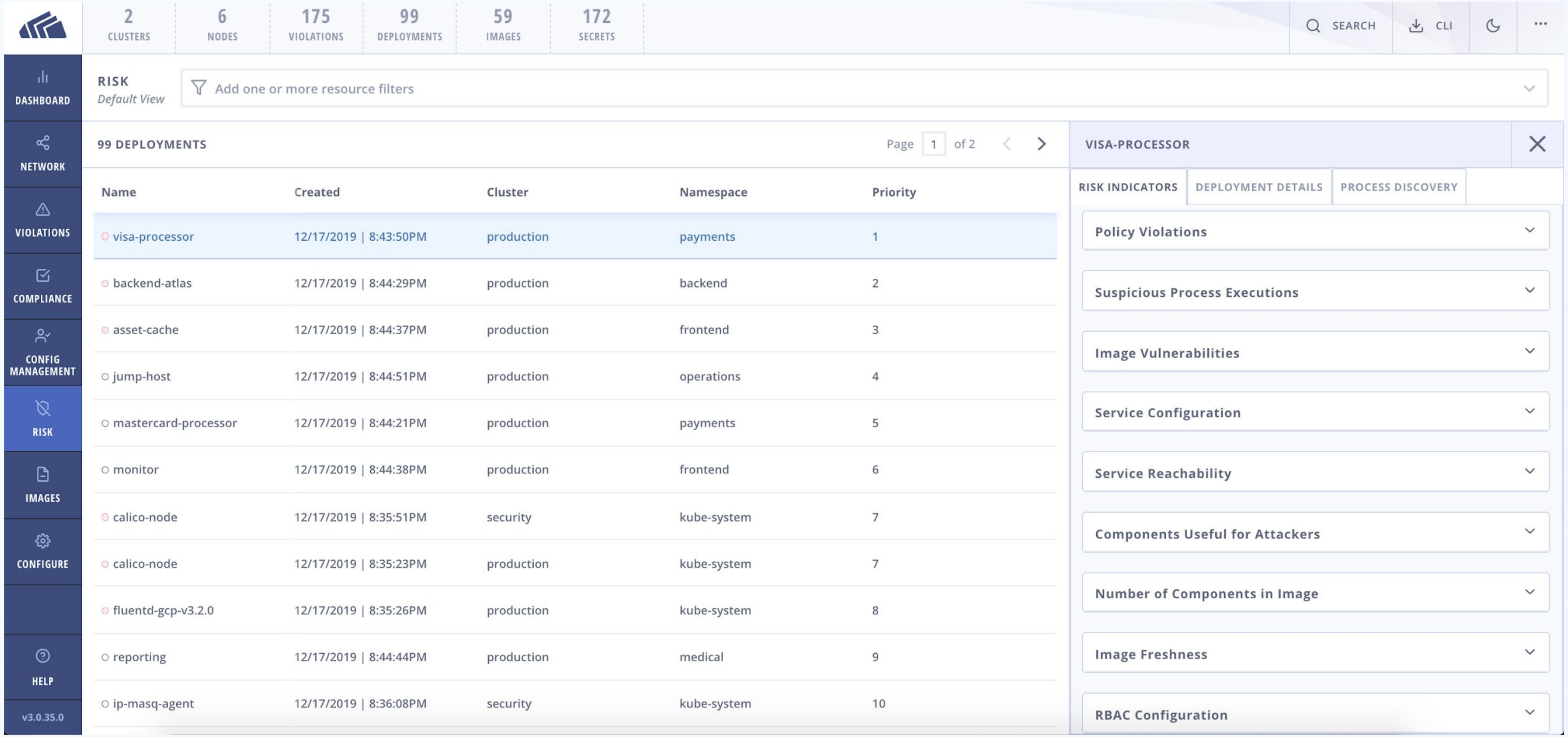


Workload Best Practices

Contextualizing Risk


Vulnerabilities, registry, packages
Image
Processes launched, active connections
Behavior
Test vs. prod, BU involved
Cluster
k8s Deployment
Blast radius, Internet reachability
Network
Criticality of the app, responsible dev team
Labels, Annotations
API key, crypto material, secrets, cloud disks
Access
Privileges
--privileged, host mounts, service account

How can we think about Risk?



2 CLUSTERS | 6 NODES | 175 VIOLATIONS | 99 DEPLOYMENTS | 59 IMAGES | 172 SECRETS

RISK
Default View

Add one or more resource filters

99 DEPLOYMENTS

Page 1 of 2

Name	Created	Cluster	Namespace	Priority
visa-processor	12/17/2019 | 8:43:50PM	production	payments	1
backend-atlas	12/17/2019 | 8:44:29PM	production	backend	2
asset-cache	12/17/2019 | 8:44:37PM	production	frontend	3
jump-host	12/17/2019 | 8:44:51PM	production	operations	4
mastercard-processor	12/17/2019 | 8:44:21PM	production	payments	5
monitor	12/17/2019 | 8:44:38PM	production	frontend	6
calico-node	12/17/2019 | 8:35:51PM	security	kube-system	7
calico-node	12/17/2019 | 8:35:23PM	production	kube-system	7
fluentd-gcp-v3.2.0	12/17/2019 | 8:35:26PM	production	kube-system	8
reporting	12/17/2019 | 8:44:44PM	production	medical	9
ip-masq-agent	12/17/2019 | 8:36:08PM	security	kube-system	10

VISA-PROCESSOR

RISK INDICATORS | DEPLOYMENT DETAILS | PROCESS DISCOVERY

- Policy Violations
- Suspicious Process Executions
- Image Vulnerabilities
- Service Configuration
- Service Reachability
- Components Useful for Attackers
- Number of Components in Image
- Image Freshness
- RBAC Configuration

Leverage Namespaces

- Great for resource usage tracking
- Allows RBAC to be finely-tuned
- Allows for generic network policies and network segmentation
- Makes kubectl results more sane

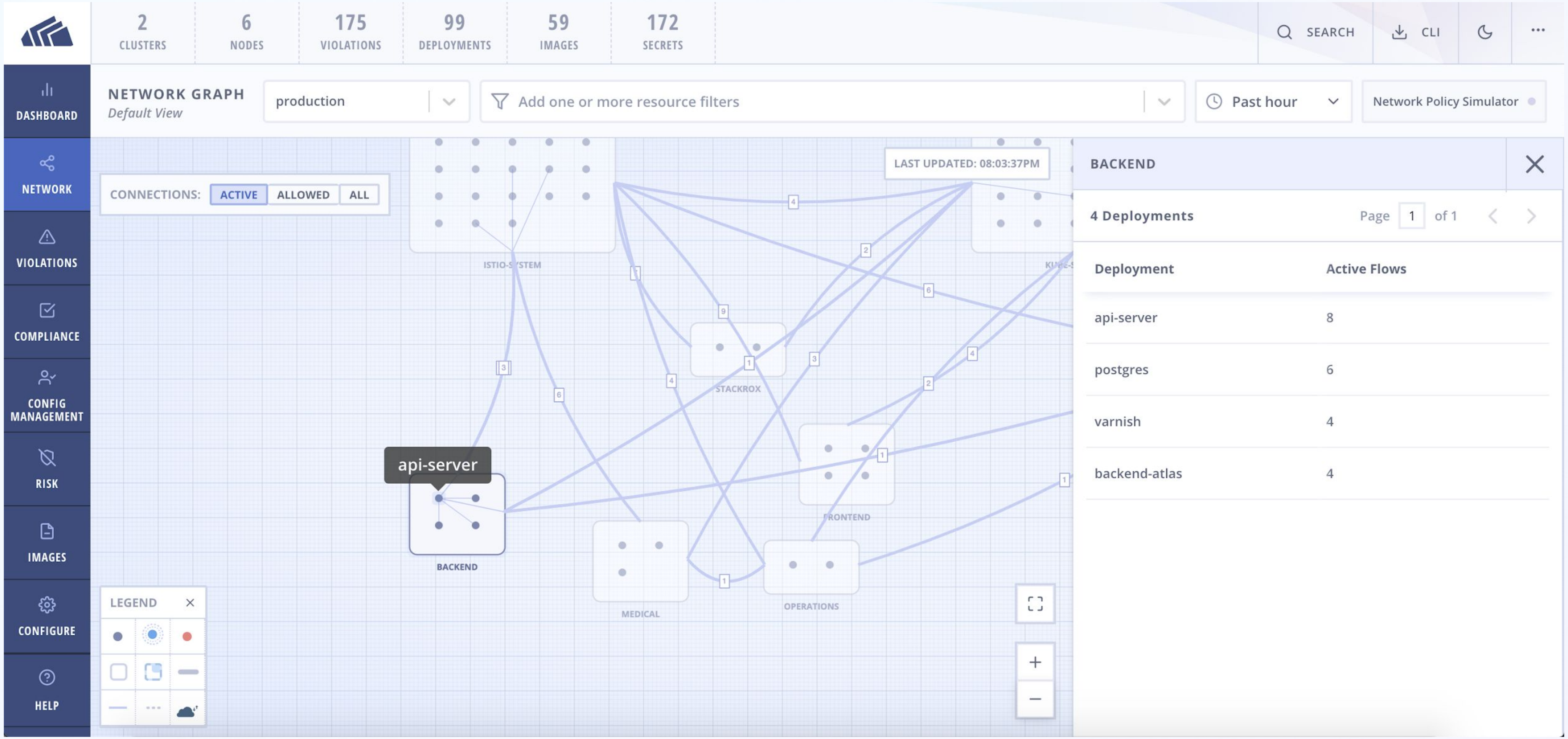
Leverage Network Policies

- Pod-centric firewalling - Pod A can/can't talk to Pod B
- Generic policies on Ingress/Egress can help ensure fine-grained connections
- Namespace isolation helps ensure compliance especially in multi-tenant environments

Challenges

- What if my environment already exists?
- How can I scale network policies at my organization?
- How do I make sure that developers are enabled to build their own network policies?

Visualize Network Traffic and Policies



BACKEND

4 Deployments — Page 1 of 1

Deployment	Active Flows
api-server	8
postgres	6
varnish	4
backend-atlas	4

Slim down your images

- Go distroless or use lightweight base images
- Remove package managers and network utilities
- Remove filesystem modification utilities (chmod, chown)
- Scan and enforce to prevent them from entering your environment again

...how do I debug now?

Looking ahead to Ephemeral Containers!

- Alpha as of 1.16! So use with caution
- Allows binding of a new container to an existing Pod to facilitate the execution of debugging commands, network utilities, etc
- Images no longer have to include: curl, apt, bash, or other utilities

Demo

Configurations to explore

- **Read-only root file system**
- **Linux capabilities**
- **Network policies**
- Host mounts
- Disable service account auto-mount
- Environment
- Resource requirements

Read-only filesystem

```
securityContext:
  readOnlyRootFilesystem: true

volumes:
  - emptyDir: {}
    name: varlog
```

$\longrightarrow$ Specifies read-only FS

$\longrightarrow$ Creates RAM based empty-dir

Example: Stopping a Struts exploit

Deploying a vulnerable container (with R/W root FS)

```
$ ./1-deploy.sh
Creating Struts-vulnerable deployment...
namespace "api" created
deployment.apps "api-server" created
NAME                          READY     STATUS               RESTARTS    AGE
api-server-7c98c55d4d-7wbl5   0/1       ContainerCreating    0            0s
api-server-7c98c55d4d-7wbl5   1/1       Running              0           2s
```

Example: Stopping a Struts exploit

The exploit works — we can download and run minerd.

```
$ ./2-exploit.sh
Forwarding traffic to Struts-vulnerable deployment...

Using Struts to try to download and run a cryptominer...

Processes running:
  PID COMMAND
    1 java
   67 minerd
```

Can my app be read-only?

```
$ docker diff k8s_nginx_nginx-7db9fccd9b-xyz
C /run
A /run/nginx.pid
A /run/secrets
A /run/secrets/kubernetes.io
A /run/secrets/kubernetes.io/serviceaccount
C /var
C /var/cache
C /var/cache/nginx
A /var/cache/nginx/client_temp
A /var/cache/nginx/fastcgi_temp
A /var/cache/nginx/proxy_temp
A /var/cache/nginx/scgi_temp
A /var/cache/nginx/uwsgi_temp
```

CLOUD NATIVE COMPUTING FOUNDATION StackRox

Example: Stopping a Struts exploit

After declaring a VOLUME for `/usr/local/tomcat`, and opting-in for a read-only root FS:

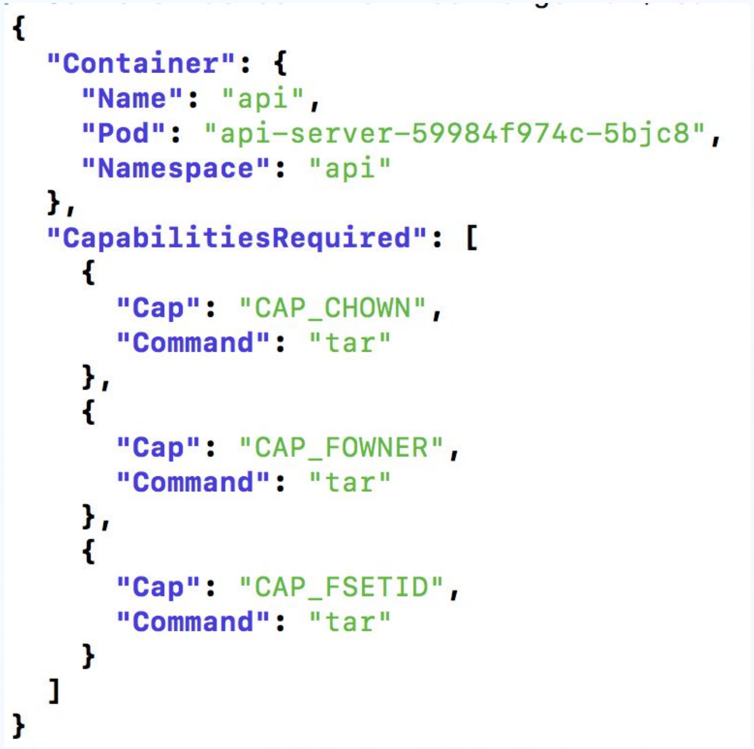
```
$ ./2-exploit.sh
Forwarding traffic to Struts-vulnerable deployment...

Using Struts to try to download and run a cryptominer...
/miner.tgz: Read-only file system

Processes running:
  PID COMMAND
    1 java
```

Linux Capabilities

Split root superpowers into a series of capabilities such as

- CAP_FOWNER (used by chmod)
- CAP_CHOWN (used by chown)
- CAP_NET_RAW (used by ping)

Linux Capabilities



```
{
  "Container": {
    "Name": "api",
    "Pod": "api-server-59984f974c-5bjc8",
    "Namespace": "api"
  },
  "CapabilitiesRequired": [
    {
      "Cap": "CAP_CHOWN",
      "Command": "tar"
    },
    {
      "Cap": "CAP_FOWNER",
      "Command": "tar"
    },
    {
      "Cap": "CAP_FSETID",
      "Command": "tar"
    }
  ]
}
```

Example: Capabilities dropped

```
securityContext:
  capabilities:
    drop:
      - all
```

```
minerd
tar: minerd: Cannot change ownership to uid 1000, gid 1000: Operation not permitted
tar: Exiting with failure status due to previous errors
```

Network Policies

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: web-allow-all-ns-monitoring
spec:
  podSelector:
    matchLabels:
      app: web
  ingress:
    - from:
      - namespaceSelector:
          matchLabels:
            team: operations
        podSelector:
          matchLabels:
              type: monitoring
```

Security is Hard!

Let’s chat

Think of a question later?

cgorman@stackrox.com

Want to learn more?

https://www.stackrox.com/

We’re hiring!