KubeCon | CloudNativeCon

Europe 2019

Container Forensics :: When your cluster becomes a cluster

Maya Kaczorowski & Ann Wallace, Google Cloud

Maya Kaczorowski

Security PM, Google Cloud

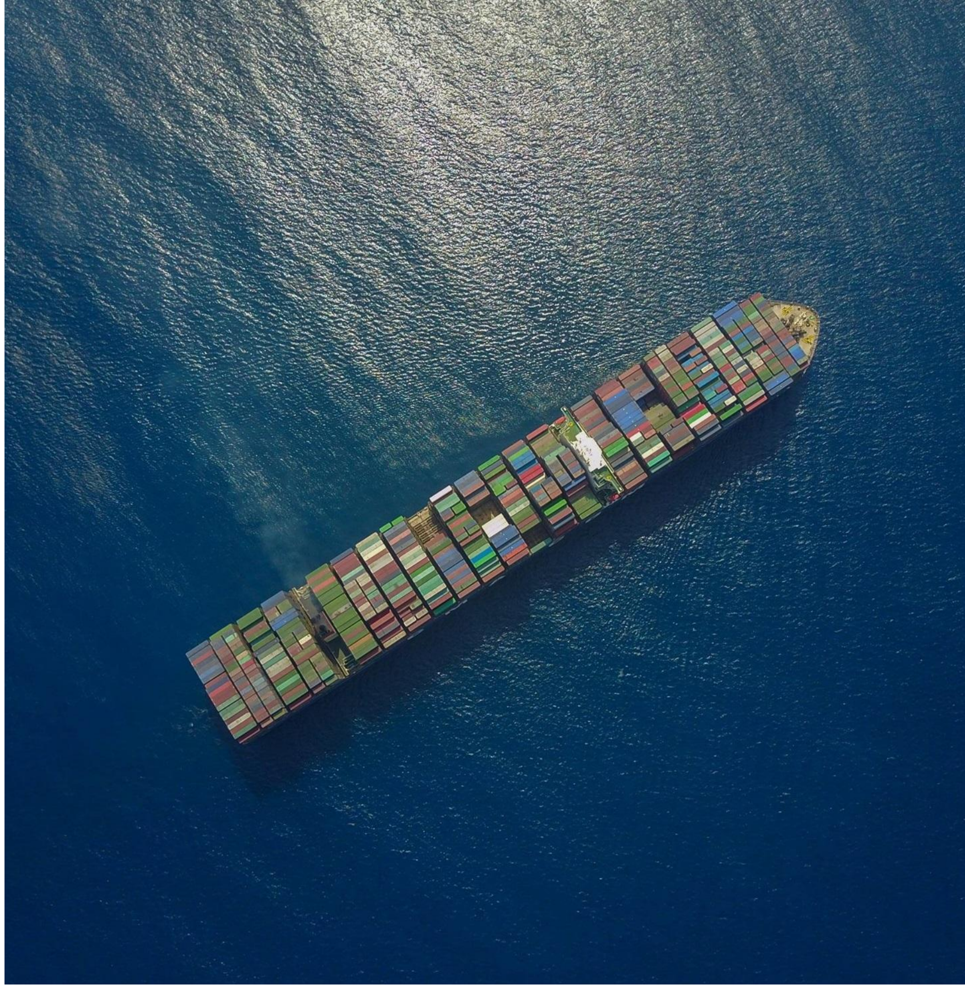
@MayaKaczorowski

Ann Wallace

Security Global Practice Lead, Google Cloud

@AnnNWallace

Container attacks happen



Google Cloud

Threats seen in the wild

2018 February May June **2019** April

Tesla

Unsecured Kubernetes dashboard with cloud account credentials

Used to mine cryptocurrency

Shopify

Researcher could access and replay kubelet credentials

Not exploited

Weight Watchers

Unsecured Kubernetes dashboard with sensitive data, including credentials

Not exploited

Docker Hub

Public images with embedded cryptocurrency mining malware

Used to mine cryptocurrency

Docker Hub

Database with 190k+ Docker Hub accounts exposed

Unknown impact

Google Cloud

GKE honeypot

100 popular apps from Docker Hub in GKE for 6 mos

“the project infrastructure largely did not attract container- or Kubernetes-specific attacks, but did attract a number of exploitation attempts.”

Security forensics 101

Google Cloud

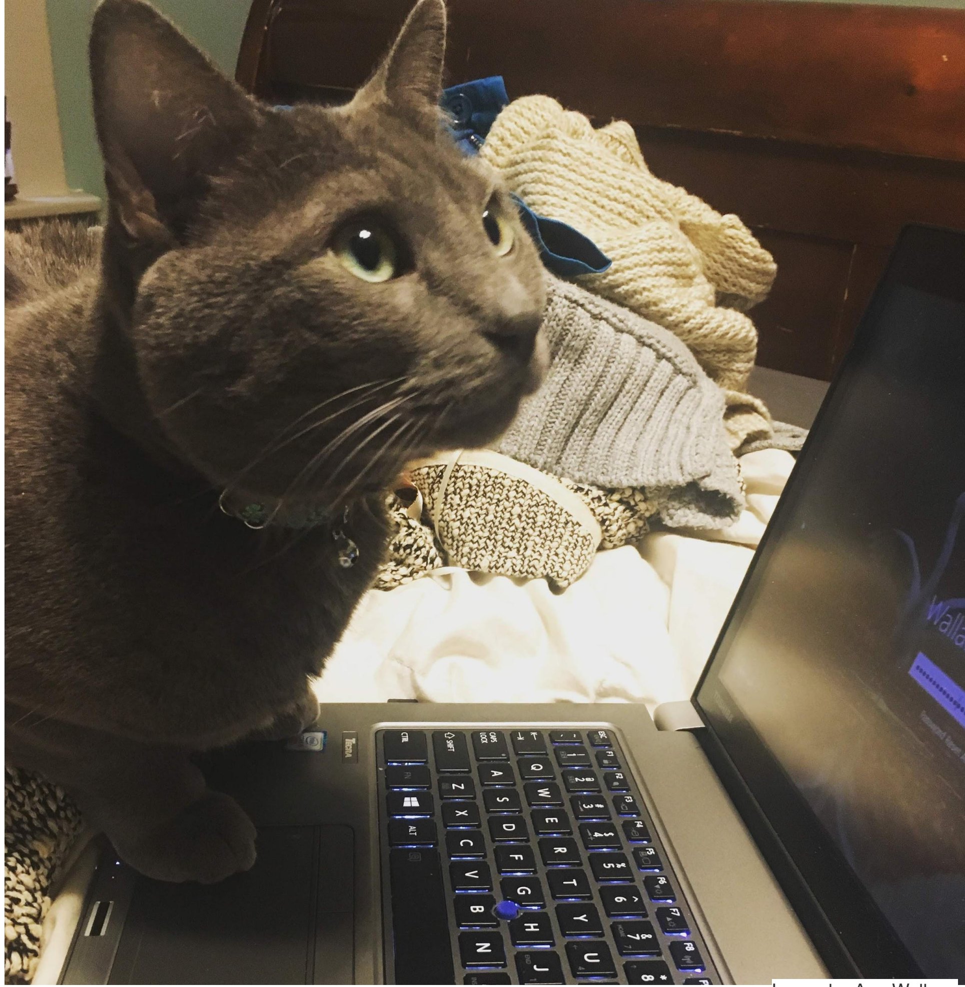
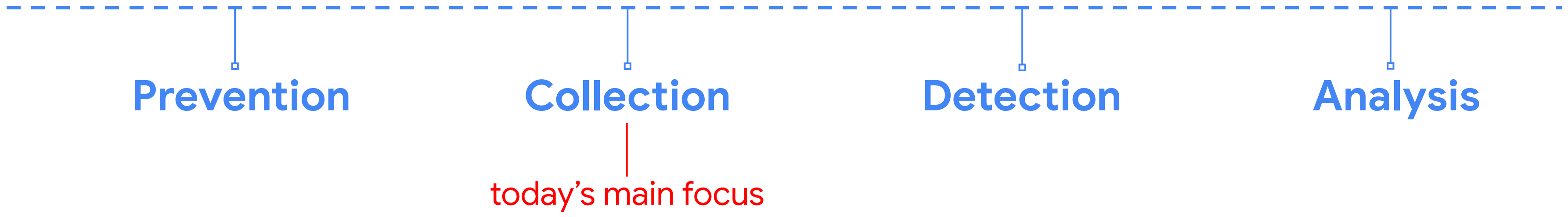


Image by Ann Wallace

Incident preparedness



Prevention

Set up a cluster

- Restrict access to kubectl
- Use RBAC
- Use a Network Policy
- Use namespaces
- Bootstrap TLS

Follow security hygiene

- Keep Kubernetes updated
- Use a minimal OS
- Use minimal IAM roles
- Use private IPs on your nodes
- Monitor access with audit logging
- Verify binaries that are deployed

Prevent known attacks

- Disable dashboard
- Disable default service account token
- Protect node metadata
- Scan images for known vulnerabilities

Prevent/limit impact of microservice compromise

- Set a Pod Security Policy
- Protect secrets
- Consider sandboxing
- Limit the identity used by pods
- Use a service mesh for authentication & encryption

Google Cloud

Don’t Panic

DO NOT!

(immediately) terminate and delete all nodes, containers & disks

DO NOT!

login to the server / container to see if you can 'track it down'

Google Cloud

Collection

How do you build a story?

Start by gathering **artifacts**

Logs

Disks

Live info

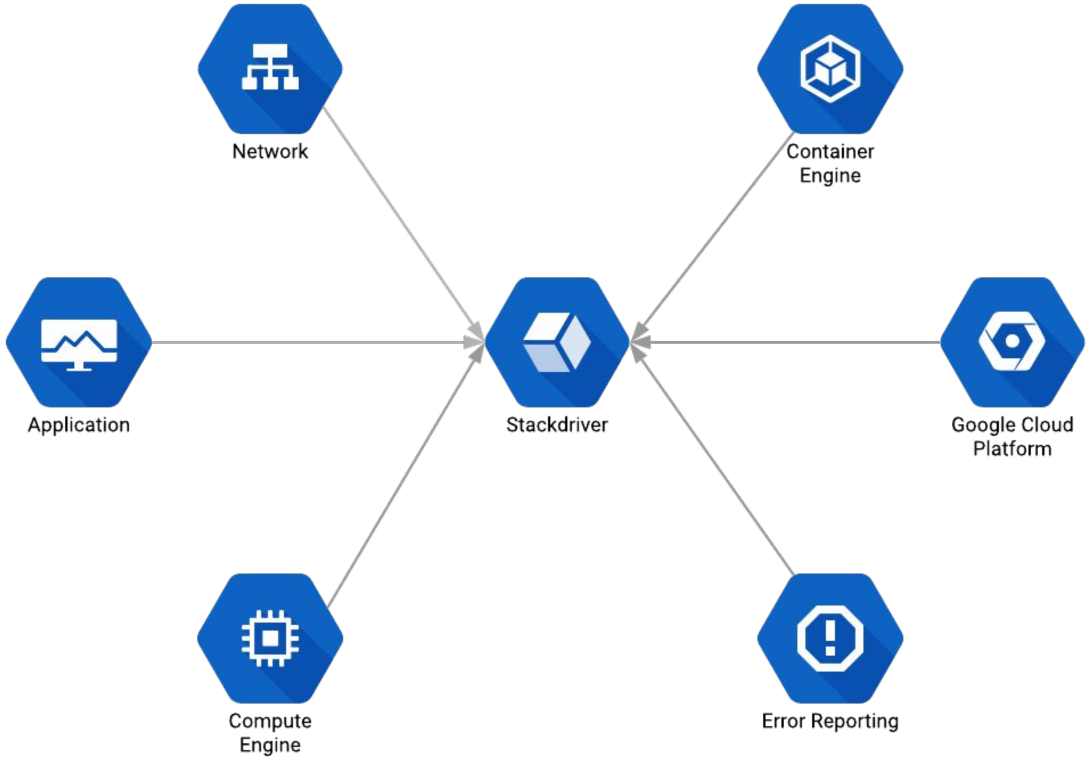
Google Cloud

Logs

Who did what, when and where?

System
Application
Network
Deployment
Cloud
Container

Google Cloud



Disks

Traditional	'Grab the disks' for offline analysis Takes machine off the network
Cloud	Use cloud APIs to make a snapshot Can be done transparently
Containers	There is no container snapshot mechanism

Live and Recorded Info

Client agents

Container sidecar

What is happening on the system?

How do you get real time info without logging in?

How do you gather information remotely from multiple systems?

Hope for the best but plan for the worst

Create an incident response plan

Who to contact

What actions to take

How to collect data

Critical systems to keep the business running

Communication plan

Google Cloud

Applying forensics to containers

Image by Ann Wallace

Google Cloud

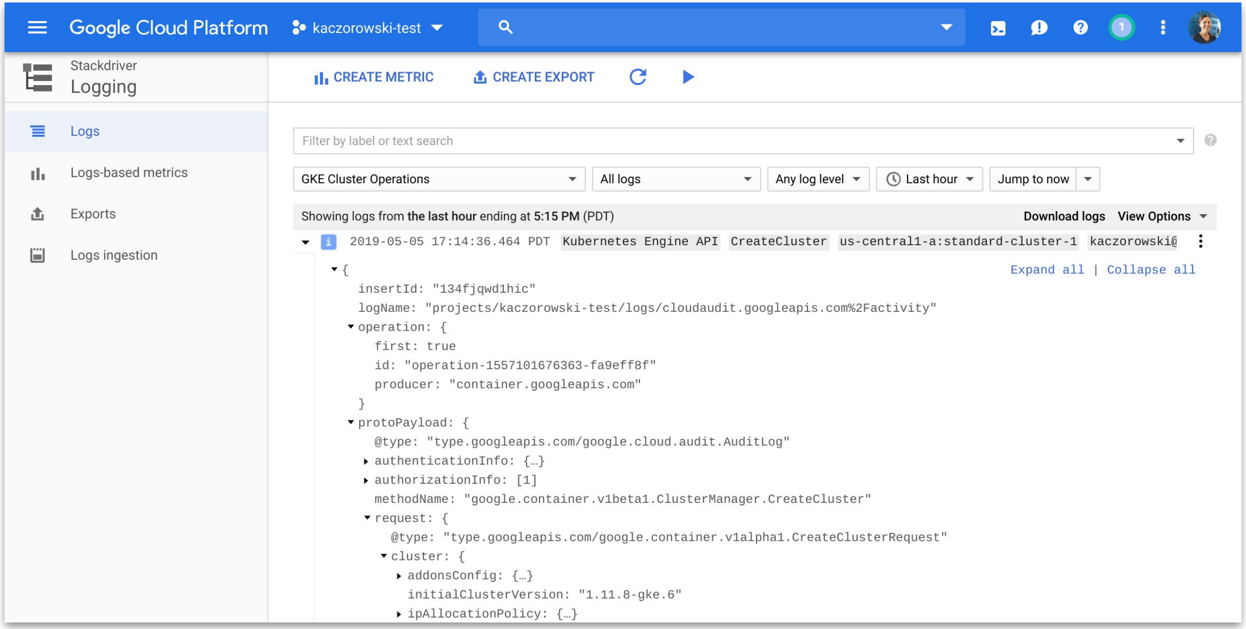
Logs

1. **Infrastructure logs**: what the infrastructure does, and what a human does to the infrastructure
2. **Kubernetes logs**: what the control plane does, what a container does to the control plane, and what a human does to the control plane
3. **Operating system logs**: what a container does to the node
4. **Application logs**: what an application does (in a container)

Google Cloud

1. Infrastructure logs

Sample
Cloud
Audit Log



Showing logs from **the last hour** ending at **5:15 PM** (PDT)

```
2019-05-05 17:14:36.464 PDT  Kubernetes Engine API  CreateCluster  us-central1-a:standard-cluster-1  kaczorowski@
  {
   insertId: "134fjqwd1hic"
   logName: "projects/kaczorowski-test/logs/cloudaudit.googleapis.com%2Factivity"
  operation: {
    first: true
    id: "operation-1557101676363-fa9eff8f"
    producer: "container.googleapis.com"
   }
  protoPayload: {
    @type: "type.googleapis.com/google.cloud.audit.AuditLog"
   authenticationInfo: {…}
   authorizationInfo: [1]
    methodName: "google.container.v1beta1.ClusterManager.CreateCluster"
   request: {
      @type: "type.googleapis.com/google.container.v1alpha1.CreateClusterRequest"
     cluster: {
       addonsConfig: {…}
        initialClusterVersion: "1.11.8-gke.6"
       ipAllocationPolicy: {…}
```

2. Kubernetes logs

Kubernetes audit policy

None <
Metadata <
Request <
RequestResponse

```
- level: Request
  verbs: ["get", "list", "watch"]
  resources: ${known_apis}
  omitStages:
    - "RequestReceived"
- level: RequestResponse
  resources: ${known_apis}
  omitStages:
    - "RequestReceived"
- level: Metadata
  omitStages:
    - "RequestReceived"
```

‘get’ responses can be large

‘RequestResponse’ default for known APIs

‘Metadata’ default for all other requests

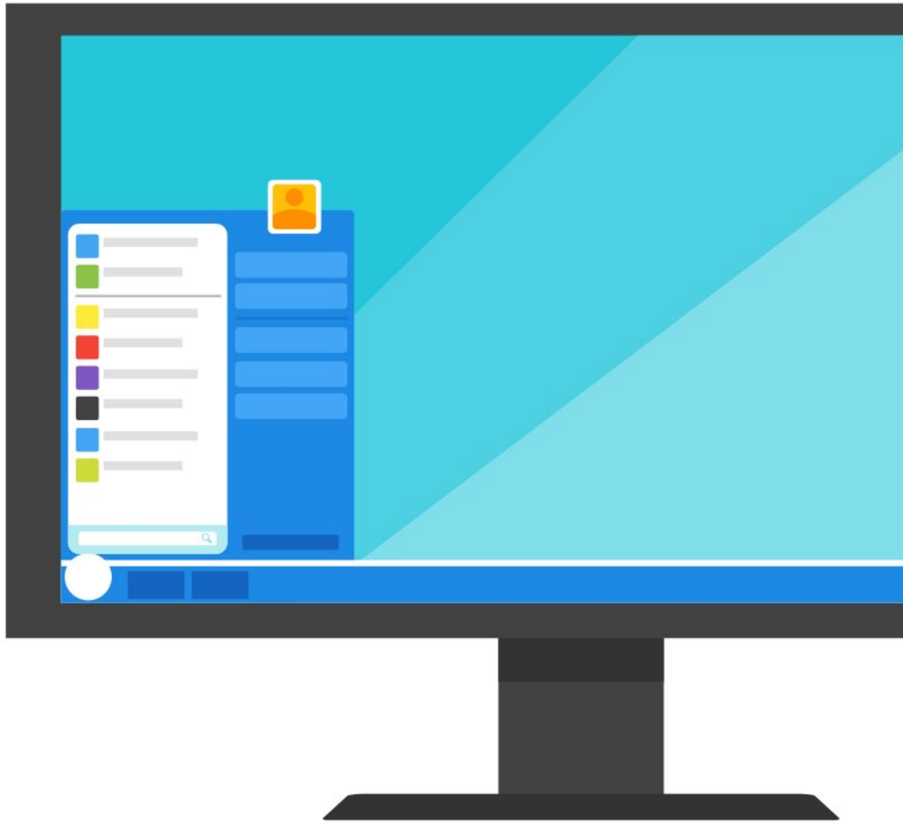
Google Cloud

Audit profile used by GKE:
https://github.com/kubernetes/kubernetes/blob/master/cluster/gce/gci/configure-helper.sh#L735

3. Operating system logs

- Network connections
- User logins
- SSH sessions
- Executions like execve()

See recommended auditd fluentd config for COS logs on GKE



Google Cloud

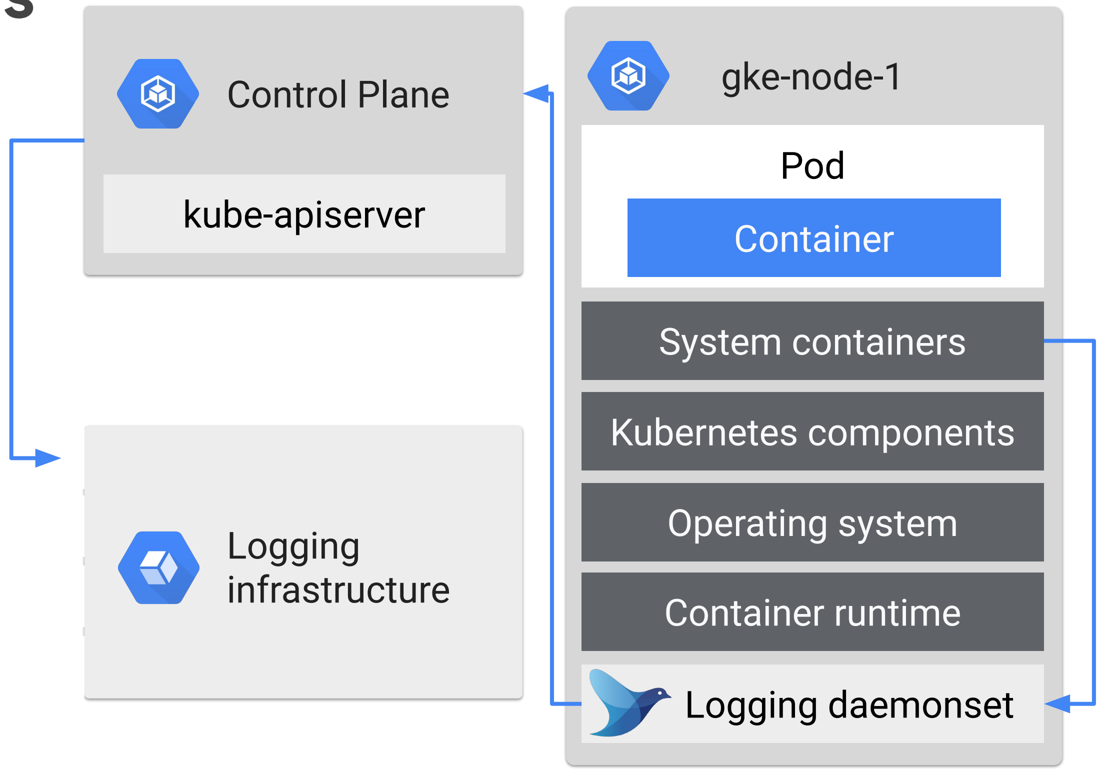
4. Application logs

- Errors
- Warnings
- Operations and other events

Google Cloud

Collecting all the logs

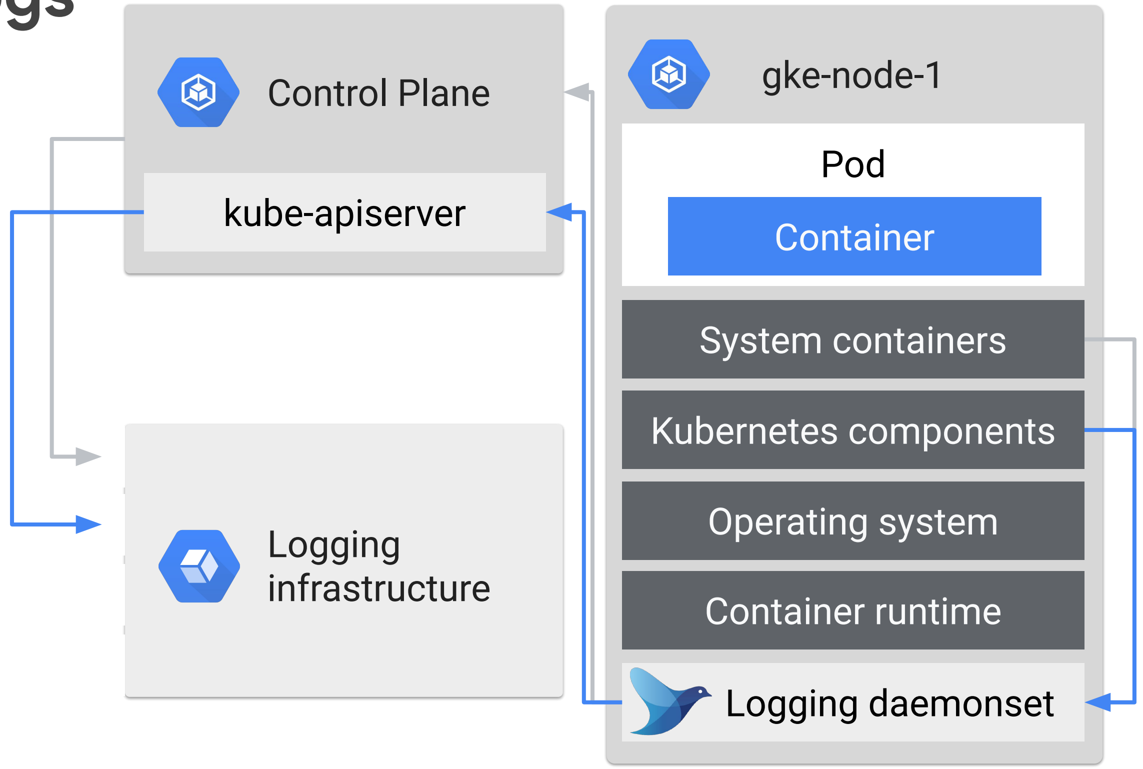
Infrastructure logs



Collecting all the logs

Infrastructure logs

Kubernetes logs



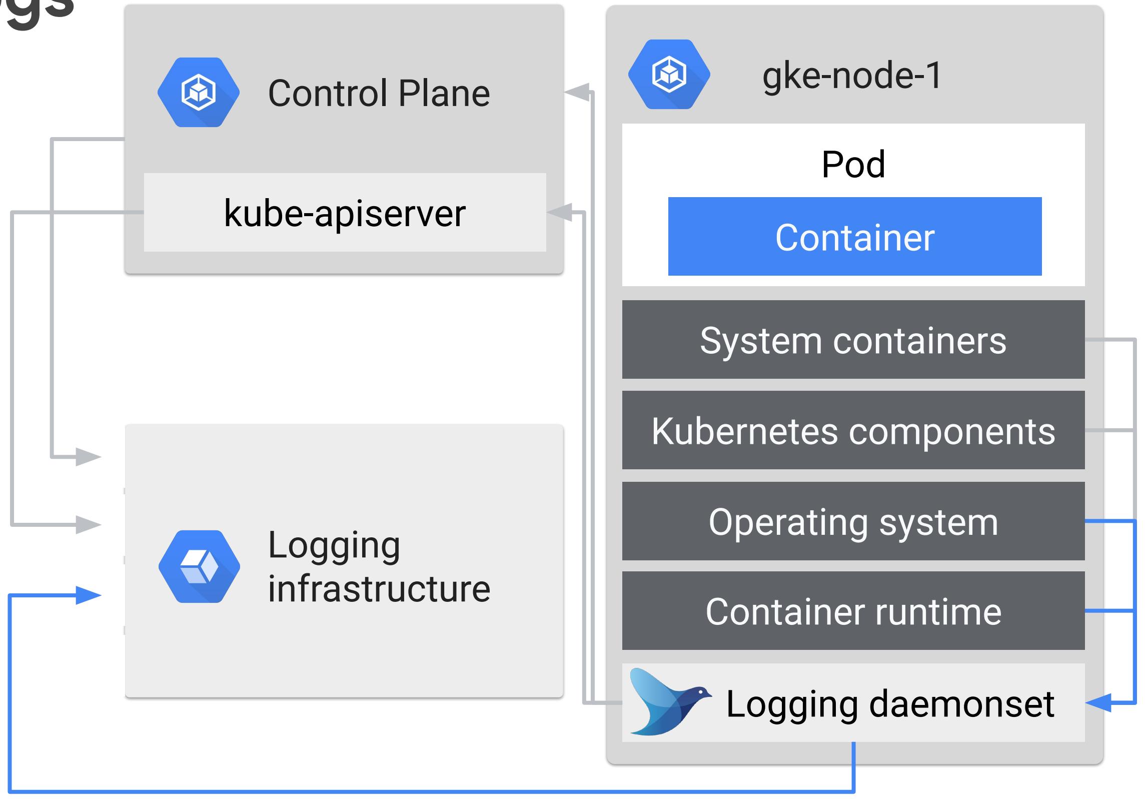
Google Cloud

Collecting all the logs

Infrastructure logs

Kubernetes logs

OS logs



Google Cloud

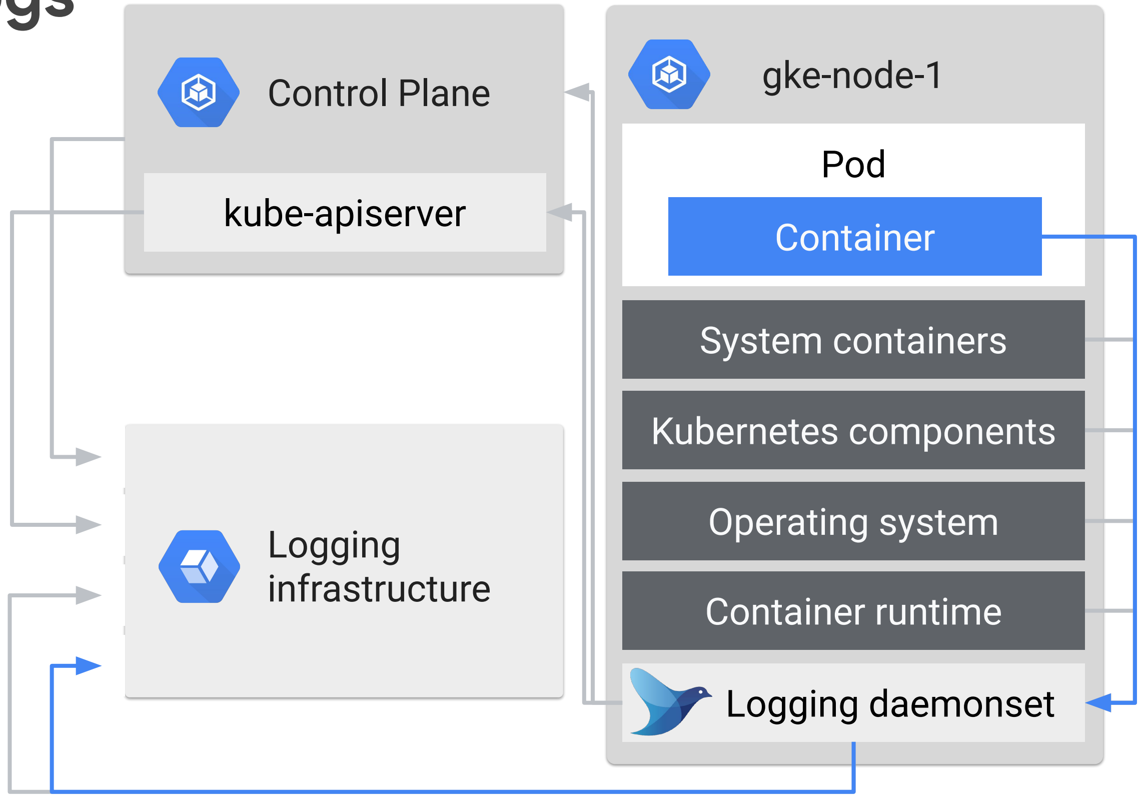
Collecting all the logs

Infrastructure logs

Kubernetes logs

OS logs

Application logs

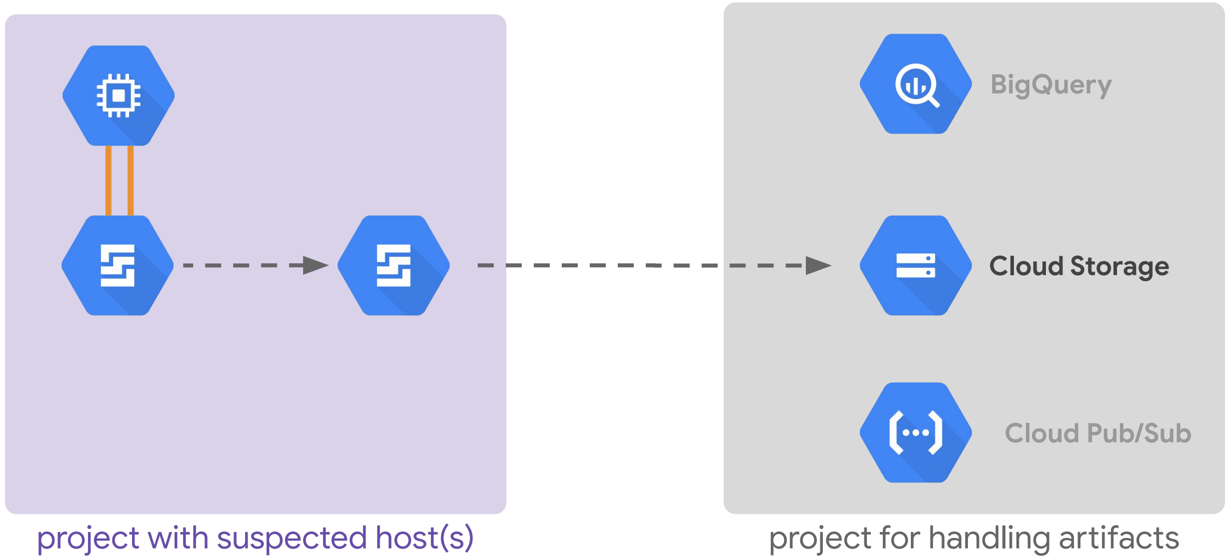


Snapshot the node

Identify affected node(s) and all attached disks

Create an duplicate of the disk(s) while online

Send the duplicated disk image for analysis



BigQuery
Cloud Storage
Cloud Pub/Sub
project with suspected host(s)
project for handling artifacts

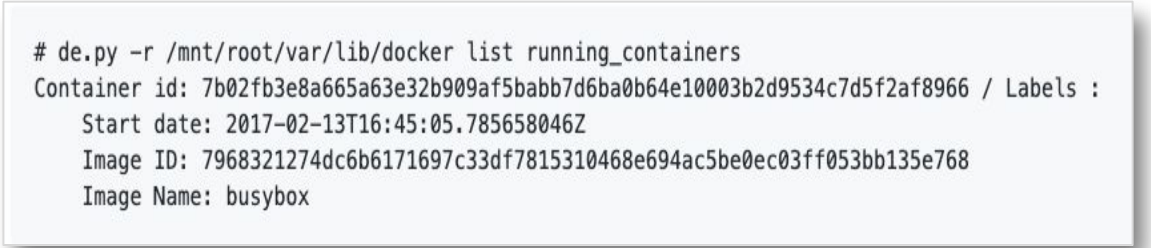
Google Cloud

docker-explorer

attach and mount the snapshot

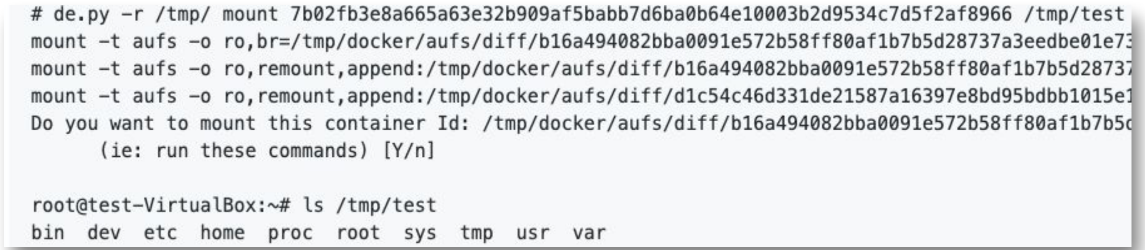
```
# mount /dev/sda1 /mnt/root
```

find the container id



```
# de.py -r /mnt/root/var/lib/docker list running_containers
Container id: 7b02fb3e8a665a63e32b909af5babb7d6ba0b64e10003b2d9534c7d5f2af8966 / Labels :
    Start date: 2017-02-13T16:45:05.785658046Z
    Image ID: 7968321274dc6b6171697c33df7815310468e694ac5be0ec03ff053bb135e768
    Image Name: busybox
```

mount the container filesystem



```
# de.py -r /tmp/ mount 7b02fb3e8a665a63e32b909af5babb7d6ba0b64e10003b2d9534c7d5f2af8966 /tmp/test
mount -t aufs -o ro,br=/tmp/docker/aufs/diff/b16a494082bba0091e572b58ff80af1b7b5d28737a3eedbe01e73
mount -t aufs -o ro,remount,append:/tmp/docker/aufs/diff/b16a494082bba0091e572b58ff80af1b7b5d28737
mount -t aufs -o ro,remount,append:/tmp/docker/aufs/diff/d1c54c46d331de21587a16397e8bd95bdbb1015e1
Do you want to mount this container Id: /tmp/docker/aufs/diff/b16a494082bba0091e572b58ff80af1b7b5d
      (ie: run these commands) [Y/n]

root@test-VirtualBox:~# ls /tmp/test
bin  dev  etc  home  proc  root  sys  tmp  usr  var
```

Google Cloud

Live and Recorded Info

GRR (GRR Rapid Response)

Sysdig Inspect & Capture

What is happening on the system?

How do you get real time info without logging in?

How do you gather information remotely from multiple systems?

Google Cloud

GRR know-before-you-go

With great power comes....

...extensive forensic capabilities that can aid in uncovering issues throughout your environment

Secure access to the GRR server

- root privileges
- admin interface
- GRR raw datastore

https://grr-doc.readthedocs.io/en/latest/installing-grr-server/**securing-access**.html

Google Cloud

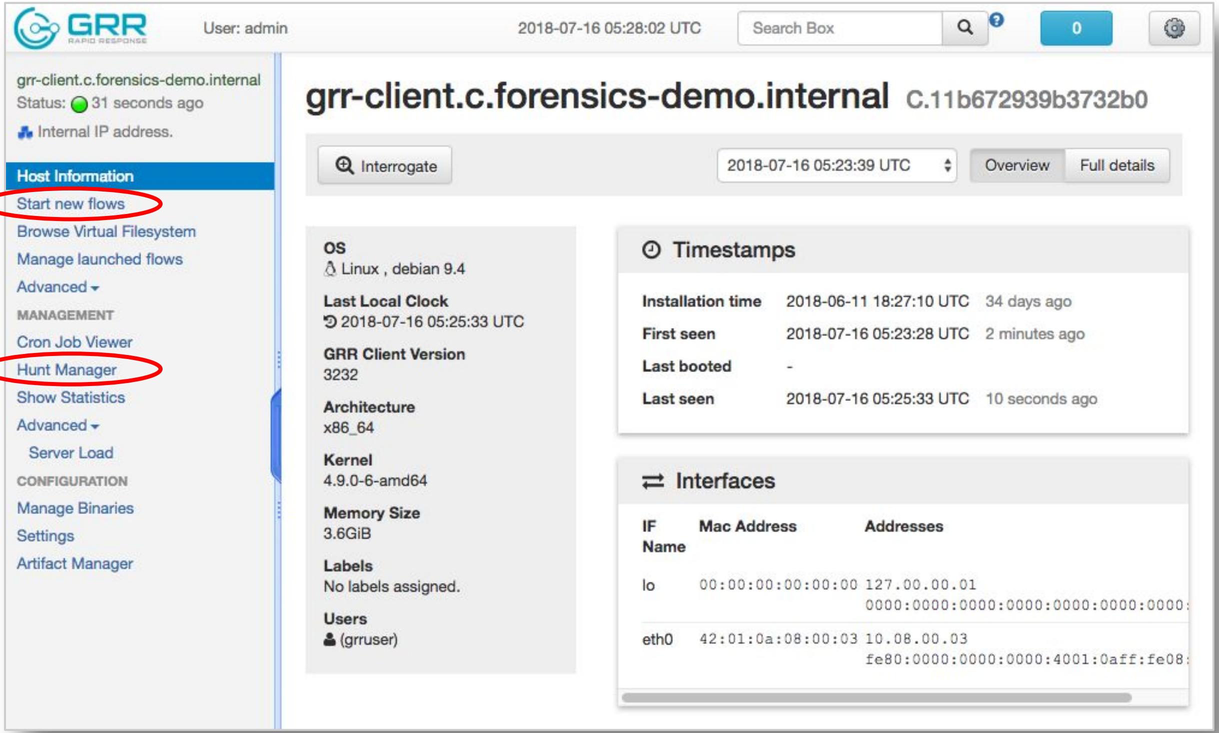
GRR admin console

Flows

Activities related to something that you've asked GRR to find out on the target machine:
- download browser history
- get details about a file
- dump memory for a process

Hunt

Running flows on a (large) set of instances looking for something specific, i.e. searching for a bad JAR or malware signature. You can monitor the progress of a hunt.



grr-client.c.forensics-demo.internal
Status: 31 seconds ago
Internal IP address.

grr-client.c.forensics-demo.internal C.11b672939b3732b0

OS
Linux , debian 9.4
Last Local Clock
2018-07-16 05:25:33 UTC
GRR Client Version
3232
Architecture
x86_64
Kernel
4.9.0-6-amd64
Memory Size
3.6GiB
Labels
No labels assigned.
Users
(grruser)

Timestamps

Installation time	2018-06-11 18:27:10 UTC	34 days ago
First seen	2018-07-16 05:23:28 UTC	2 minutes ago
Last booted	-	
Last seen	2018-07-16 05:25:33 UTC	10 seconds ago

Interfaces

IF Name	Mac Address	Addresses
lo	00:00:00:00:00:00	127.00.00.01 0000:0000:0000:0000:0000:0000:0000:
eth0	42:01:0a:08:00:03	10.08.00.03 fe80:0000:0000:0000:4001:0aff:fe08:

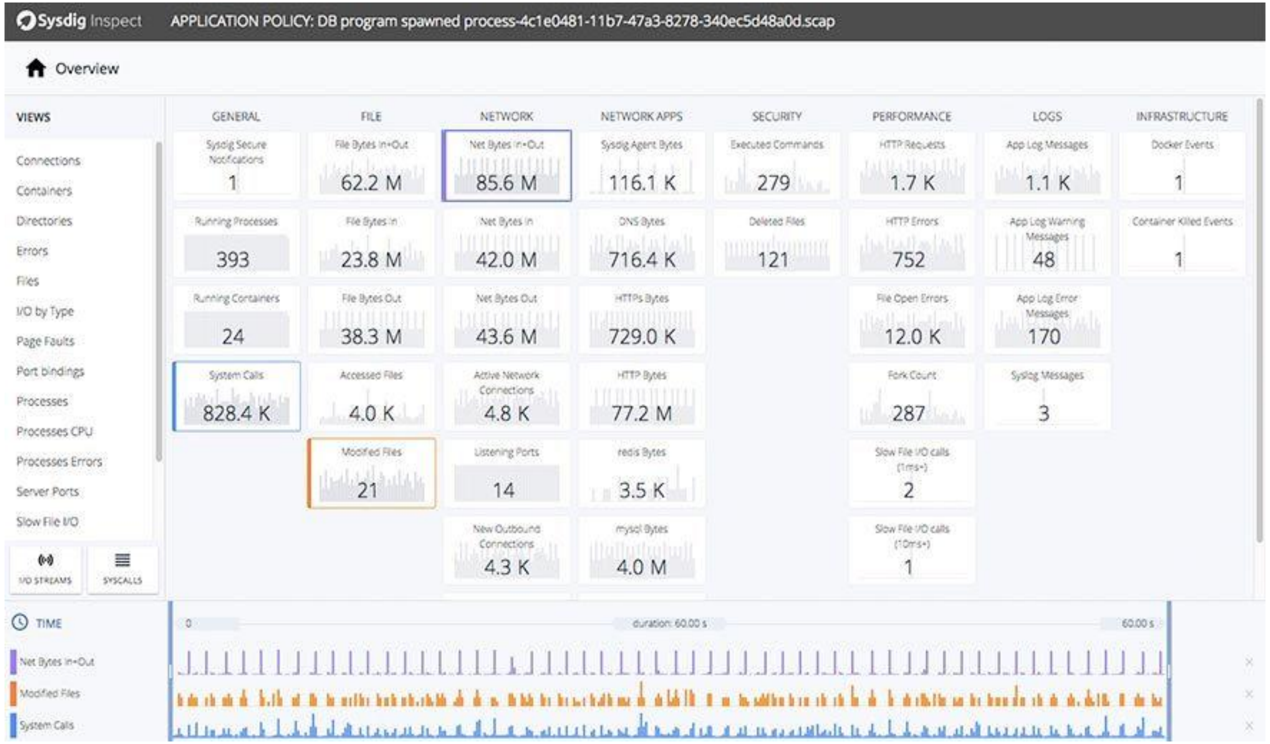
Google Cloud

Sysdig Inspect & Capture

observability

investigation

container history



Sysdig Inspect APPLICATION POLICY: DB program spawned process-4c1e0481-11b7-47a3-8278-340ec5d48a0d.scap

Overview

VIEWS

Connections
Containers
Directories
Errors
Files
I/O by Type
Page Faults
Port bindings
Processes
Processes CPU
Processes Errors
Server Ports
Slow File I/O

I/O STREAMS
SYSCALLS

GENERAL	FILE	NETWORK	NETWORK APPS	SECURITY	PERFORMANCE	LOGS	INFRASTRUCTURE
Sysdig Secure Notifications 1	File Bytes In+Out 62.2 M	Net Bytes In+Out 85.6 M	Sysdig Agent Bytes 116.1 K	Executed Commands 279	HTTP Requests 1.7 K	App Log Messages 1.1 K	Docker Events 1
Running Processes 393	File Bytes In 23.8 M	Net Bytes In 42.0 M	DNS Bytes 716.4 K	Deleted Files 121	HTTP Errors 752	App Log Warning Messages 48	Container Killed Events 1
Running Containers 24	File Bytes Out 38.3 M	Net Bytes Out 43.6 M	HTTPs Bytes 729.0 K		File Open Errors 12.0 K	App Log Error Messages 170	
System Calls 828.4 K	Accessed Files 4.0 K	Active Network Connections 4.8 K	HTTP Bytes 77.2 M		Fork Count 287	Syslog Messages 3	
	Modified Files 21	Listening Ports 14	redis Bytes 3.5 K		Slow File I/O calls (1ms+) 2		
		New Outbound Connections 4.3 K	mysql Bytes 4.0 M		Slow File I/O calls (10ms+) 1		

TIME 0 duration: 60.00 s 60.00 s

Net Bytes In+Out
Modified Files
System Calls

Google Cloud

Common mitigation options

Google Cloud

Mitigation options

Alert	Send an alert
Isolate	Restrict from other workloads
Pause	Stop running processes
Restart	Kill and restart running processes
Kill	Kill running processes but not restart

Mitigation options

Alert

Isolate

Pause

Restart

Kill

What it is:

- Alert your security response team to investigate

When you'd do it:

- Initial triage
  - Large SecOps team with container expertise
  - New environment not yet fine-tuned

How you would do it:

- Trigger on specific metrics or specific actions
- Metrics on centralized logs, to SMS/ email/ Slack/ etc.

Mitigation options

Alert

Isolate

Pause

Restart

Kill

What it is:

- Quarantine the container to watch what it does

When you'd do it:

- Get more info to know what's going on

How you would do it:

- Get on its own node
  - `kubectl cordon`
- Restrict connectivity, e.g., Network Policy
- Monitor with live forensics, agent, or filtering

Google Cloud

Mitigation options

Alert

Isolate

Pause

Restart

Kill

What it is:

- Suspend running processes

When you'd do it:

- Get further data for forensics
  - Auditing
  - Confirm the issue

How you would do it:

- `docker pause`

Google Cloud

Mitigation options

Alert

Isolate

Pause

Restart

Kill

What it is:

- Kill and restart a running container

When you'd do it:

- Roll out a fix

How you would do it:

- `docker restart`
- `kubectl delete pod`
- Roll out a new image!

Google Cloud

Mitigation options

Alert

Isolate

Pause

Restart

Kill

What it is:

- Stop running processes, without restart

When you'd do it:

- As a last resort (sh*t's on fire, yo)

How you would do it:

- `docker stop` = `SIGTERM`, and `SIGKILL` after 10 sec or `crictl stop`
- `docker kill` = `SIGKILL`
- `docker rm -f` = `SIGKILL or crictl rm -f`

Tying it all together

Image by Ann Wallace

Google Cloud

Privilege escalation

TL;DR - an attacker is able to break out of the container and effectively becoming root on the node.

Google Cloud

Gather some evidence

1. What do you already know?
2. What do you have in place to help you determine: Who, What, How, When, Where?

Google Cloud

Tying it all together :: logs

Deployment or OS logs	How was the container launched?
Container logs	Are there unexpected commands being ran? In, mv, cp, cat, *.sh, tar wget Are files in /dev or /proc being touched?
Network logs	Is there unexpected network traffic or increased egress traffic from a particular node?

Google Cloud

Tying it all together :: disks

Container & Nodes:

Have any binaries changed?

Are there any unexpected files?

Google Cloud

Tying it all together :: live & recorded info

What interesting things happened on the system?

- ➔ Processes
- ➔ System Calls
- ➔ Files
- ➔ Network
- ➔ I/O
- ➔ Users

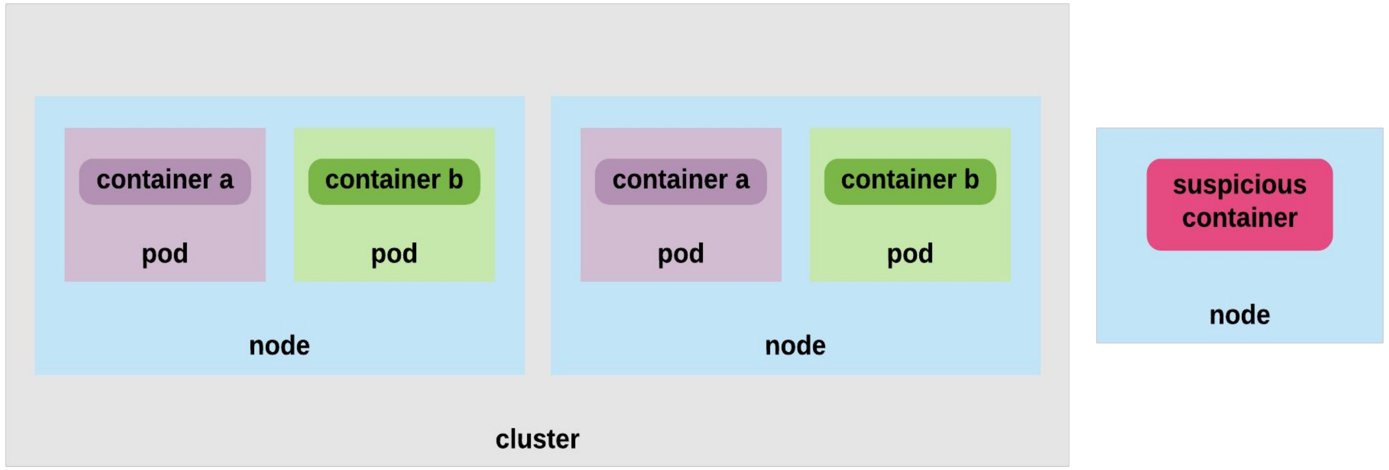
Google Cloud

Tying it all together :: mitigation options

Alert

Isolate

Deploy



Google Cloud

Tying it all together :: prevention

Preventing privilege escalation

Scan your images for vulnerabilities

Only allowed signed images to be deployed

Don’t run containers with the root user

Use user namespace isolation

Google Cloud

Steps to take today

Google Cloud

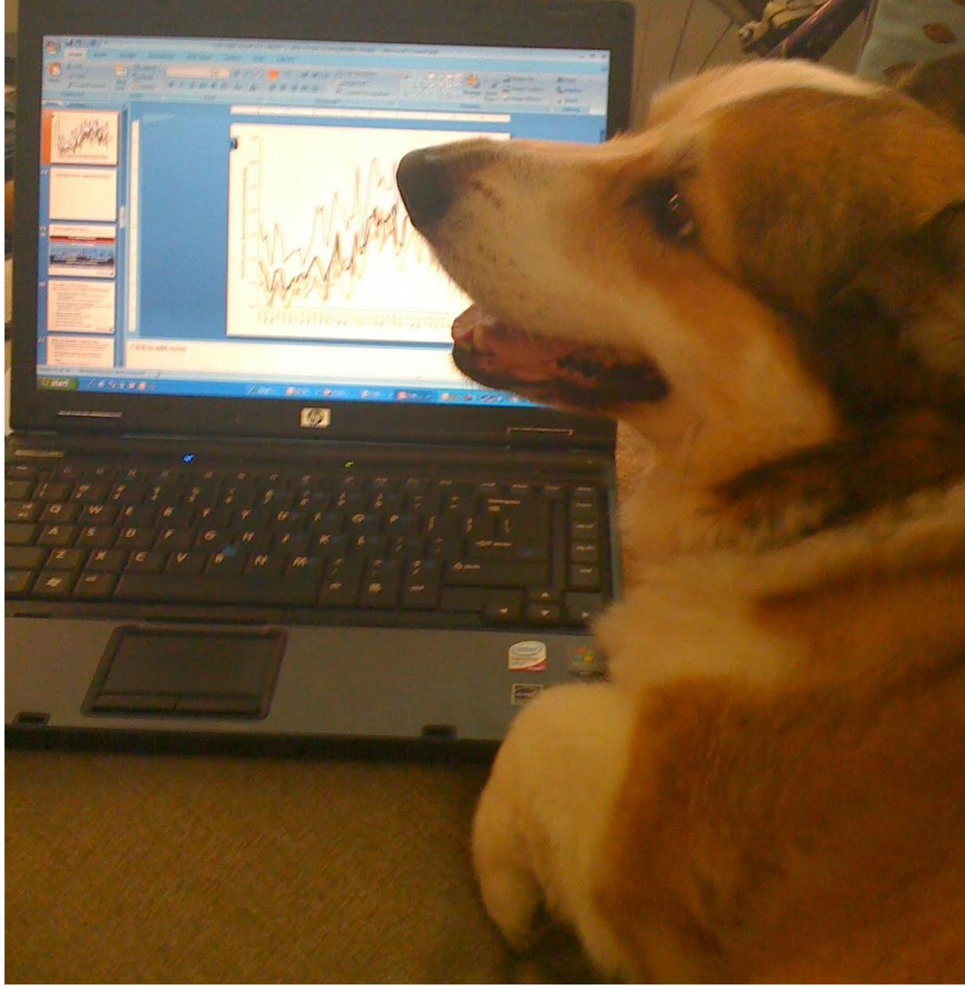


Image by Ann Wallace

You’ve got this!

- Create an incident response plan
- Follow container security best practices
- Sync all your logs to a central location
- Invest in container specific security tools (OSS or off the shelf)
- Rehearse the process with a fake event
- Don’t panic - Sh*t happens

Google Cloud

Image by Ann Wallace

Read

cloud.google.com/containers/security
sysdig.com/blog/gke-security-using-falco/

Watch

“Cloud Forensics 101” on YouTube

Clone

github.com/google/grr
github.com/spotify/terraform-google-grr
github.com/google/docker-explorer
github.com/sysdiglabs/kubectl-capture
github.com/draios/sysdig-inspect

Google Cloud

Questions?

Google Cloud