ALL DAY DEVOPS

caffeinated by sonatype

NOVEMBER 11, 2022

Secure Your Prometheus Server From Indiscreet Eyes or Die by Metrics

Miguel Hernández, *@MiguelHzBz*
David de Torres, @maellyssa@mastodon.social

sysdig

MODERN INFRASTRUCTURE

Agenda

- What is…
- History keeps repeating
- Scenarios
  - Worst
  - Recommend / Best practices
- Conclusion

What is

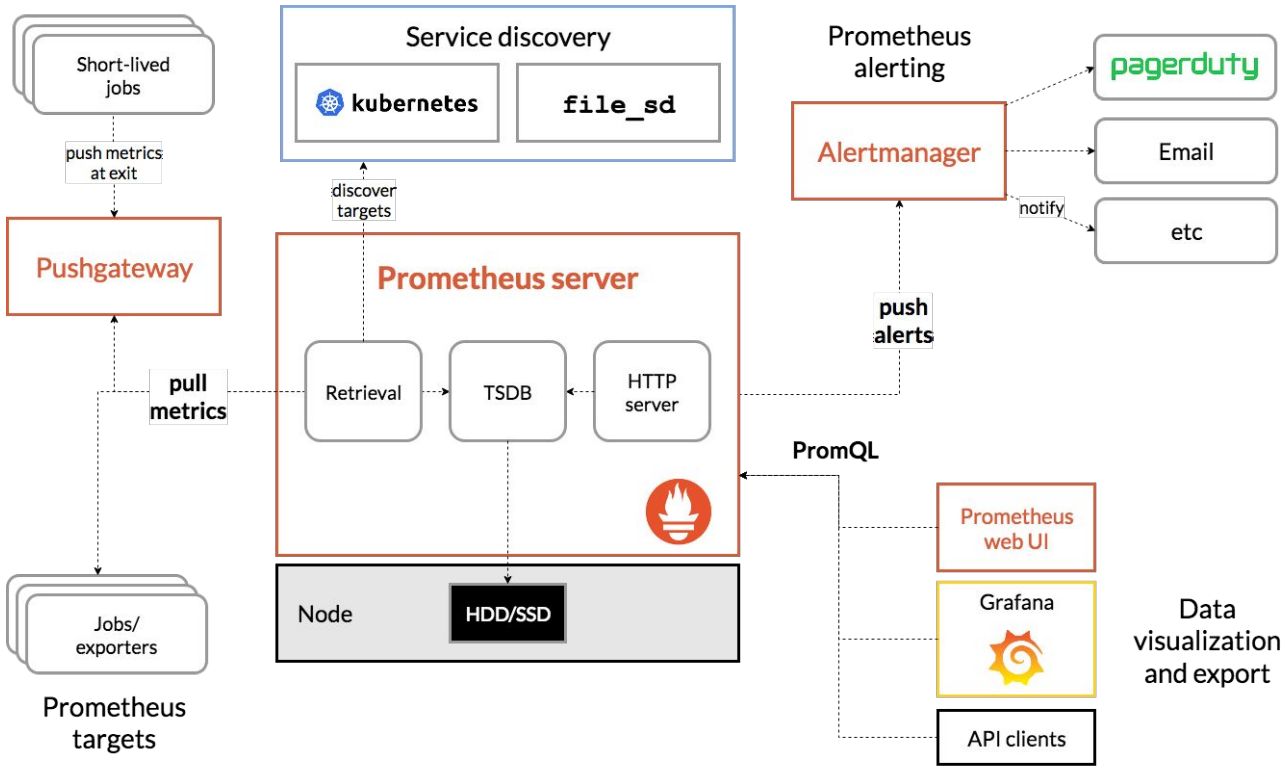
ADDO
ALL DAY DEVOPS
caffeinated by sonatype

MODERN
INFRASTRUCTURE

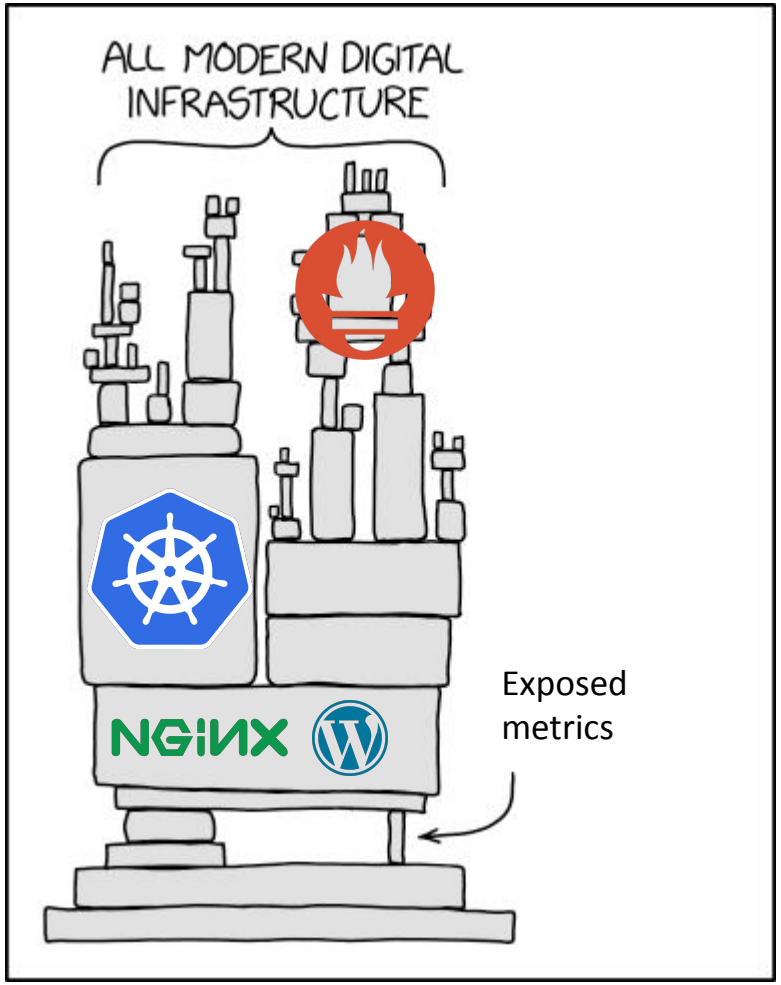
Kubernetes

- Open-source and open vendor-neutral governance & ownership, CNCF project to orchestrate containers at scale

Prometheus


Short-lived jobs
push metrics at exit
Pushgateway
Service discovery
kubernetes
file_sd
discover targets
Prometheus server
Retrieval
TSDB
HTTP server
pull metrics
Node
HDD/SSD
Jobs/ exporters
Prometheus targets
Prometheus alerting
Alertmanager
pagerduty
Email
etc
notify
push alerts
PromQL
Prometheus web UI
Grafana
API clients
Data visualization and export

Single point of failure



https://xkcd.com/2347/

Aquel que no conoce la historia, está condenado a repetirla

Kubernetes in the wild

Accessing the Dashboard UI

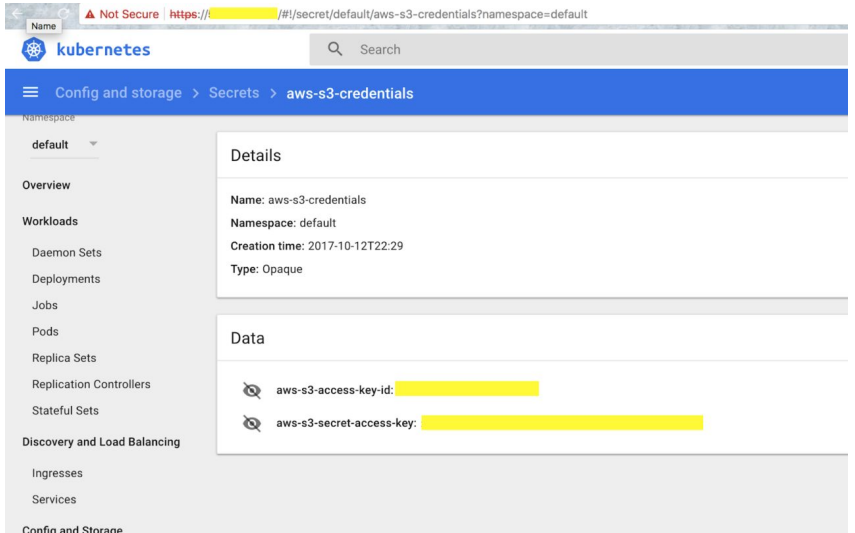
To protect your cluster data, Dashboard deploys with a minimal RBAC configuration by default. Currently, Dashboard only supports logging in with a Bearer Token. To create a token for this demo, you can follow our guide on creating a sample user.

Warning: The sample user created in the tutorial will have administrative privileges and is for educational purposes only.

Tesla cloud resources are hacked to run cryptocurrency-mining malware

Crooks find poorly secured access credentials, use them to install stealth miner.

DAN GOODIN - 2/20/2018, 8:21 PM



https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/

But Prometheus is only metrics...

juice-shop / juice-shop Public

Code | Issues 3 | Pull requests 1 | Actions | Security | Insights

[⭐] Exposed Prometheus Metrics Endpoint #1275

Closed J12934 opened this issue on 28 Dec 2019 · 9 comments

https://github.com/juice-shop/juice-shop/issues/1275

BLOG HOME >

Don't let Prometheus Steal your Fire

Real world secrets exposed by unsafe defaults

By Andrey Polkovnychenko and Shachar Menashe | October 12, 2021

12 min read

SHARE:

https://jfrog.com/blog/dont-let-prometheus-steal-your-fire/

https://www.cncf.io/online-programs/a-look-at-how-hackers-exploit-prometheus-grafana-fluentd-jaeger-more/

Worst scenario

Worst scenario

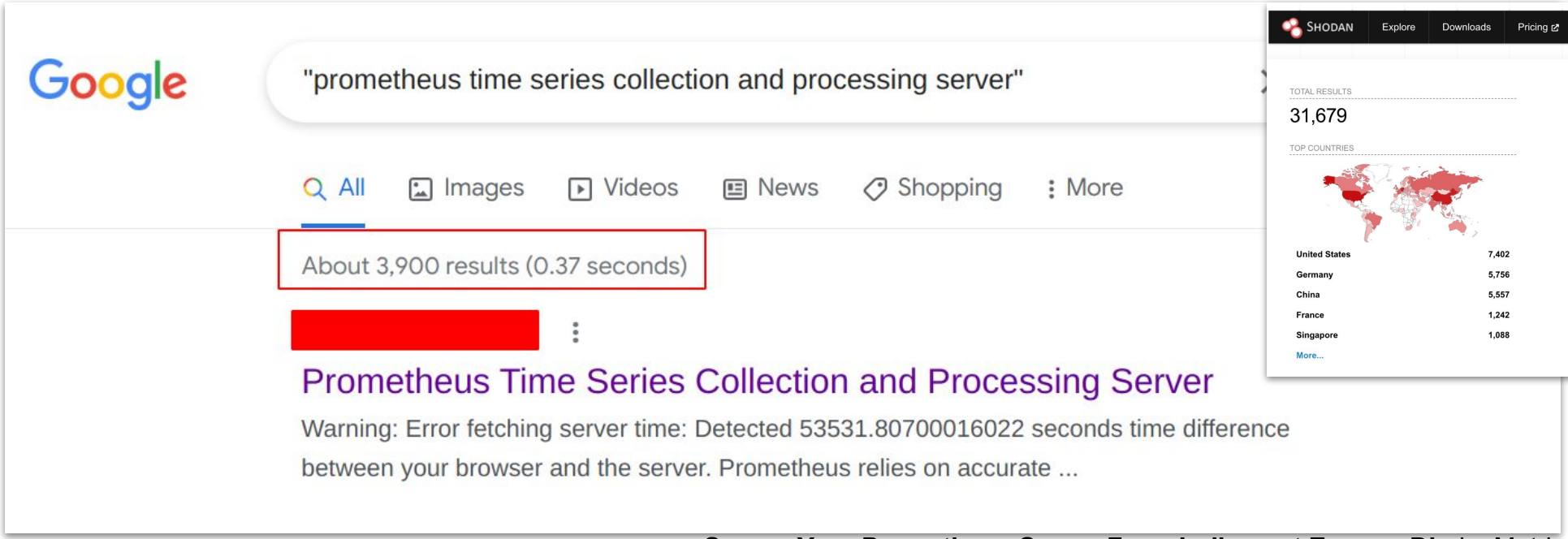
HackK8s Cluster Any%		3
	1:58.92	
Gathering info - Prometheus	-1:23	0:32.9
Initial access - T1195	-1:24	0:50.0
Level Up - Elevation Privileges	-1:23	1:06.9
Gain Persistence	-1:58	1:18.4
Leak Secrets	-2:10	1:26.9
Remove evidences	-2:08	1:42.6
$$$$$$$	-2:11	1:58.9

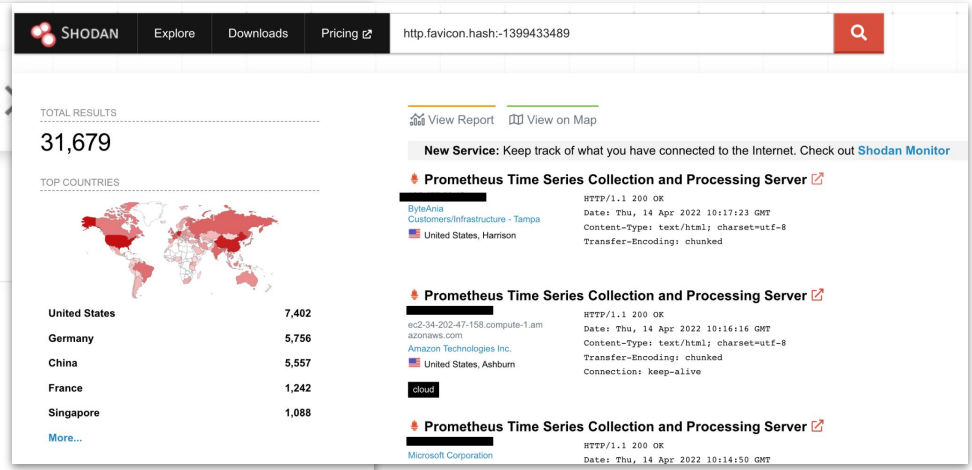
- Prometheus exposed to internet
- No authentication

Real or fiction?

Prometheus allows (and recommends) using basic authentication, but not enabled by default: https://prometheus.io/docs/operating/security/

Exposing open Prometheus endpoints to the Internet is a bad idea... and as every bad idea, it's highly adopted:





https://github.com/sansatart/scrapts/blob/master/shodan-favicon-hashes.csv

What will we us to fingerprint Kubernetes?

Node Exporter

- Physical infrastructure
- Network interfaces

Kube State Metrics

- Host OS & kernel
- Kubernetes components
- Hostnames and network topology
- Logical hierarchy
- Secrets location
- Applications (and versions) deployed

Node Exporter:
`node_dmi_info`

bios_vendor:
- SeaBIOS
- Amazon EC2

bios_version:
- seabios-1.9.1-qemu-project.org
- 8f19b21
- 1.0

bios_release:
- 1.0

bios_date:
- 10/16/2017
- 04/01/2014

chassis_asset_tag:
- Amazon EC2

chassis_vendor:
- Amazon EC2
- Alibaba Cloud

system_vendor:
- Tencent Cloud
- Amazon EC2
- Alibaba Cloud

product_name:
- m5.xlarge
- Alibaba Cloud ECS

product_version:
- pc-i440fx-2.1

board_vendor:
- Amazon EC2

board_asset_tag:
- i-00280f617XXXXX

board_vendor:
- Smdbmds
- Amazon EC2

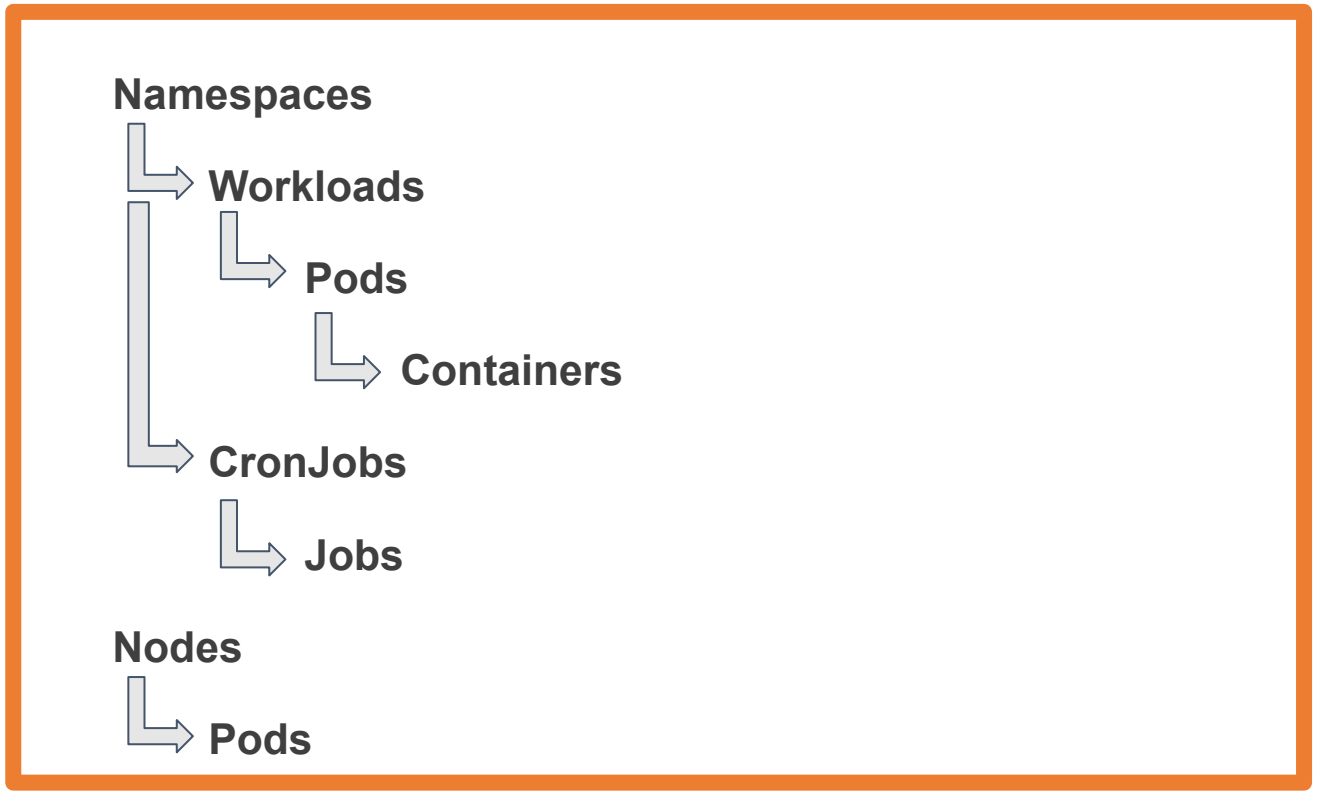
Node Exporter:

```
node_network_info{device=~'eth.+'}
```

```
{
    address="06:d5:XX:XX:XX:XX",
    broadcast="ff:ff:ff:ff:ff:ff",
    device="eth0",
    instance="172.31.XX.XX:9100",
    instance_az="us-west-2a",
    instance_id="i-XXXXX",
    instance_name="XXX-XXX",
    instance_type="c5.xlarge",
    instance_vpc="vpc-XXXXXXX",
    job="ec2_instances",
    operstate="up"
}
```

KSM:

```
kube_namespace_status_phase

kube_deployment_spec_replicas
kube_daemonset_status_desired_number_scheduled
kube_statefulset_replicas
kube_replicaset_spec_replicas

kube_pod_info

kube_pod_container_info

kube_cronjob_info

kube_job_info
```



Kubernetes:

`kubernetes_build_info`

Component
- API-server
- controller-manager
- kube-proxy…

Major, minor version

git version

git commit

build_date

go_version

KSM Exporter:

`kube_node_info`

os_image:

- Ubuntu 18.04.4 LTS
- Ubuntu 20.04.3 LTS
- CentOS Linux 7 (Core)
- Tencent Linux 2.4

kernel_version:

- 5.11.0-1027-aws
- 4.15.0-142-generic
- 4.14.105-19-0020.1
- 3.10.0-1160.59.1.el7.x86_64

KSM:

`kube_pod_container_info`

Custom:

`prometheus_build_info`

pod (app name)

image name + tag + sha256

- docker.io/library/cassandra:3.11.6
- sha256:5aa8400b4b3b794b5eba85f79b75a9ed9326e41428ae3a9d6b91cd731f2cf768

Prometheus version

KSM:

```
kube_secret_info
kube_secret_type

kube_secret_annotations
```

Namespace

Secret name

Type
- Opaque
- service-account-token…

Kubectl last applied info (leak)

Application (application that uses the secret can be usually guessed by the name of secret/namespace)

```
kube_secret_annotations{kubectl_kubernetes_io_last_applied_configuration != ""}
```

ADDO ALL DAY DEVOPS caffeinated by sonatype

Cloud provider

Credentials:

- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking

- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:

- Kube-proxy
- Kube-admin
- Kubelet

Topology

- Cluster IP
- Namespaces
- Nodes

Known Vulnerabilities:

- CVE-2020-8554
- CVE-2020-8558
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

Node

- Kernel
- OS
- Go version
- Git version
- Docker

Known Vulnerabilities:

- CVE-2022-0847 - dirty pipe (Kernel Linux)
- CVE-2022-0185
- USN-3833-1: Linux kernel (AWS) vulnerabilities
  - CVE-2018-18955
- CVE-2021-3156

Pod / Container

Registry:

- docker.io

Image:

- Image-id

Service

- Service-example
  - Website
  - API
  - …
  - https://example.com

Known Vulnerabilities:

- CVE-2021-44521 - Cassandra
- https://mariadb.com/kb/en/security/ - RCE
- CVE-2020-28035
- Wordpress
- CVE-2018-16850 - PostgreSQL
- CVE-2019-11043 - PHP
- CVE-2021-44228 - Log4j
- CVE-2022-22963 - Spring Cloud
- CVE-2020-13942 - Apache unomi

Kubernetes Secrets

- Service auth tokens

Path Attacker

Leak data scenario - Attacker Path (Supply Chain)

Cloud provider

Credentials:
- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking
- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:
- Kube-proxy
- Kube-admin
- Kubelet

Topology
- Cluster IP
- Namespaces
- Nodes

Known Vulnerabilities:
- CVE-2020-8554
- CVE-2020-8558
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

Node

- Kernel
- OS
- Go version
- Git version
- Docker

Known Vulnerabilities:
- CVE-2022-0847 - dirty pipe (Kernel Linux)
- CVE-2022-0185
- USN-3833-1: Linux kernel (AWS) vulnerabilities
  - CVE-2018-18955
- CVE-2021-3156

Pod / Container

Registry:
- docker.io

Image:
- Image-id

Service
- Service-example
  - Website
  - API

Known Vulne
- CVE-2021-44521 - Cassandra
- https://mariadb.com/kb/en/security/ - RCE
- CVE-2020-28035
- Wordpress
- CVE-2018-16850 - PostgreSQL
- CVE-2019-11043 - PHP
- CVE-2021-44228 - Log4j
- CVE-2022-22963 - Spring Cloud
- CVE-2020-13942 - Apache unomi

Kubernetes Secrets

- Service auth tokens

Cryptomining scenario - Attacker Path

Cloud provider

Credentials:
- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking
- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:
- Kube-proxy
- Kube-admin
- Kubelet

Topology
- Cluster IP
- Namespaces
- Nodes

Known Vulnerabilities:
- CVE-2020-8554
- CVE-2020-8558
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

Node

- Kernel
- OS
- Go version
- Git version
- Docker

Known Vulnerabilities:
- CVE-2022-0847 - Dirty pipe (Kernel Linux)
- CVE-2022-0185
- USN-5333-1: Linux kernel (AWS) vulnerabilities
  - CVE-2018-18955
- CVE-2021-3156

https://dirtypipe.cm4all.com/
https://github.com/arget13/DDexec

Pod / Container

Registry:
- docker.io

Image:
- Image-id

Service
- Service-example
  - Website
  - API
  - …
  - https://example.com

Known Vulnerabilities:
- CVE-2021-44521 - Cassandra
- https://mariadb.com/kb/en/security/ - RCE
- CVE-2020-28035
- Wordpress
- CVE-2018-16850 - PostgreSQL
- CVE-2019-11043 - PHP
- CVE-2021-44228 - Log4j
- CVE-2022-22963 - Spring Cloud
- CVE-2020-13942 - Apache unomi

https://github.com/kozmer/log4j-shell-poc

Kubernetes Secrets

- Service auth tokens

Ransomware scenario - Attacker Path

Cloud provider

Credentials:
- AWS Keys
- Tencent Keys
- Alibaba Keys

Networking
- Load Balancer
- Region & AZ
- VPC
- Instance IP & ID

K8s Cluster

Components:
- Kube-proxy
- Kube-admin
- Kubelet

Topology
- Cluster IP
- Namespaces
- Nodes

Known Vulnerabilities:
- CVE-2020-8554
- CVE-2020-8559
- CVE-2020-8559
- CVE-2021-25735
- CVE-2021-25737
- CVE-2021-25741

2

Node

- Kernel
- OS
- Go version
- Git version
- Docker

Known Vulnerabilities:
- CVE-2022-0847 - dirty pipe (Kernel Linux)
- CVE-2022-0185
- USN-3833-1: Linux kernel (AWS) vulnerabilities
  - CVE-2018-18955
- CVE-2021-3156

Pod / Container

Registry:
- docker.io

Image:
- Image-id

Service
- Service-example
  - Website
  - API
  - …
  - https://example.com

Known Vulnerabilities:
- CVE-2021-44521 - Cassandra
- https://mariadb.com/kb/en/security/ - RCE
- CVE-2020-28035
- Wordpress
- CVE-2018-16850 - PostgreSQL
- CVE-2019-11043 - PHP
- CVE-2021-44228 - Log4j
- CVE-2022-22963 - Spring Cloud
- CVE-2020-13942 - Apache Unomi
- …

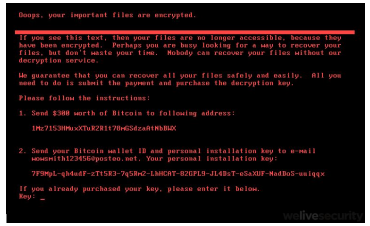
1

https://github.com/hktalent/spring-spel-0day-poc

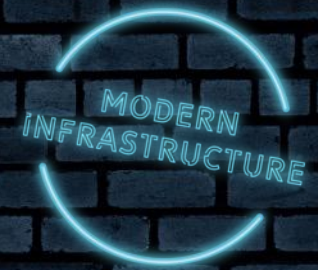
Kubernetes Secrets

3

- Service auth tokens



Recommend scenario

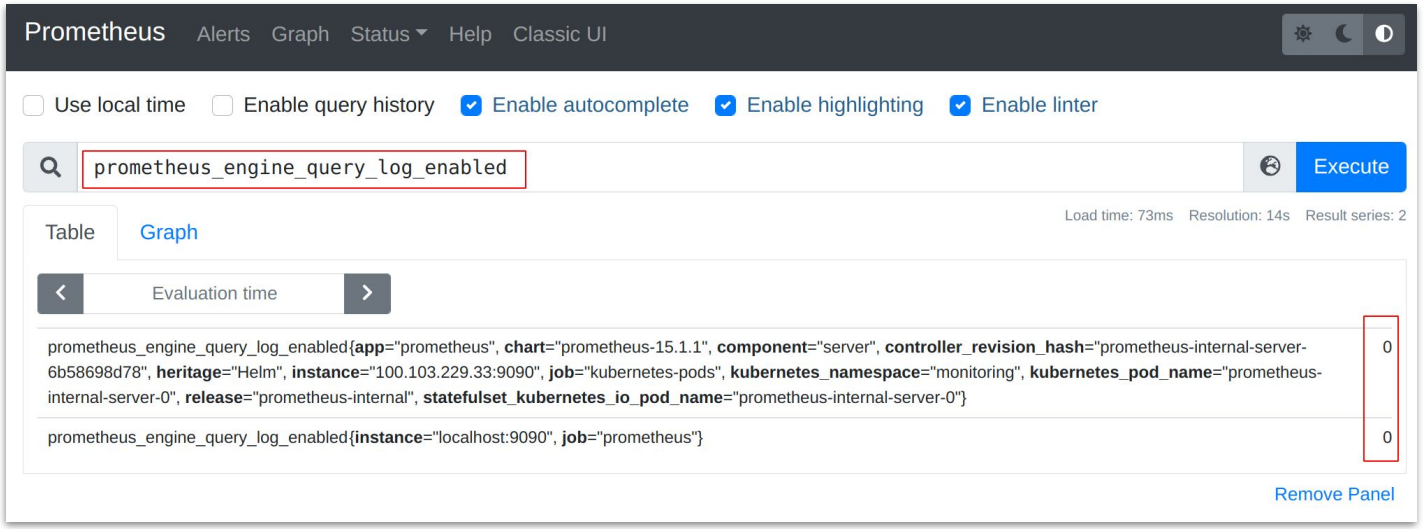


Logging Queries

Prometheus allows query logging... but it's not enabled by default.

You can check if loggin is enabled by querying this metric

prometheus_engine_query_log_enabled



USING THE PROMETHEUS QUERY LOG

Prometheus has the ability to log all the queries run by the engine to a log file, as of 2.16.0. This guide demonstrates how to use that log file, which fields it contains, and provides advanced tips about how to operate the log file.

Enable the query log

The query log can be toggled at runtime. It can therefore be activated when you want to investigate slownesses or high load on your Prometheus instance.

To enable or disable the query log, two steps are needed:

1. Adapt the configuration to add or remove the query log configuration.
2. Reload the Prometheus server configuration.

Logging all the queries to a file

This example demonstrates how to log all the queries to a file called `/prometheus/query.log`. We will assume that `/prometheus` is the data directory and that Prometheus has write access to it.

First, adapt the `prometheus.yml` configuration file:

```
global:
  scrape_interval:     15s
  evaluation_interval: 15s
  query_log_file: /prometheus/query.log
scrape_configs:
- job_name: 'prometheus'
  static_configs:
  - targets: ['localhost:9090']
```

Prometheus Secrets

Secrets

Non-secret information or fields may be available via the HTTP API and/or logs.

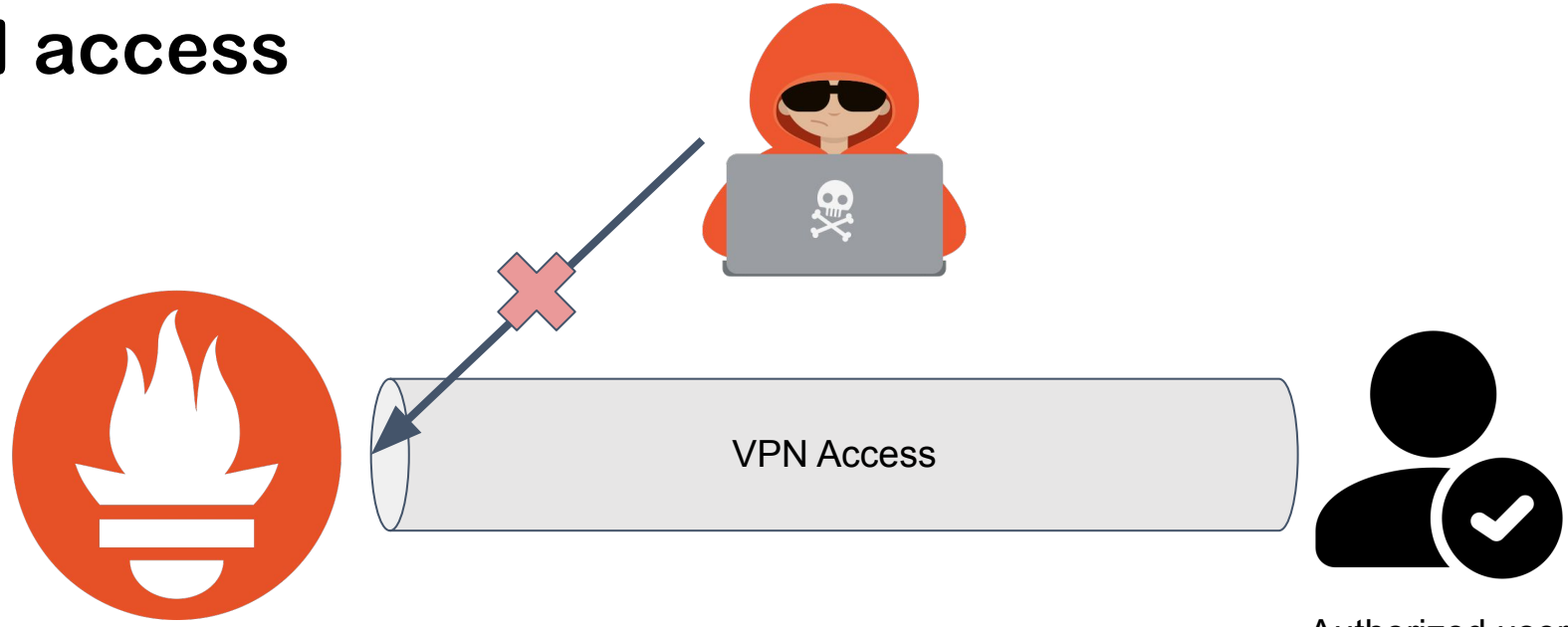
In Prometheus, metadata retrieved from service discovery is not considered secret. Throughout the Prometheus system, metrics are not considered secret.

Fields containing secrets in configuration files (marked explicitly as such in the documentation) will not be exposed in logs or via the HTTP API. Secrets should not be placed in other configuration fields, as it is common for components to expose their configuration over their HTTP endpoint. It is the responsibility of the user to protect files on disk from unwanted reads and writes.

Secrets from other sources used by dependencies (e.g. the `AWS_SECRET_KEY` environment variable as used by EC2 service discovery) may end up exposed due to code outside of our control or due to functionality that happens to expose wherever it is stored.

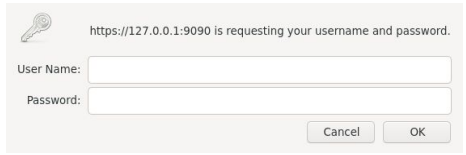
Securing Exporters and Prometheus from outsiders

- VPN access



Securing Exporters and Prometheus from outsiders

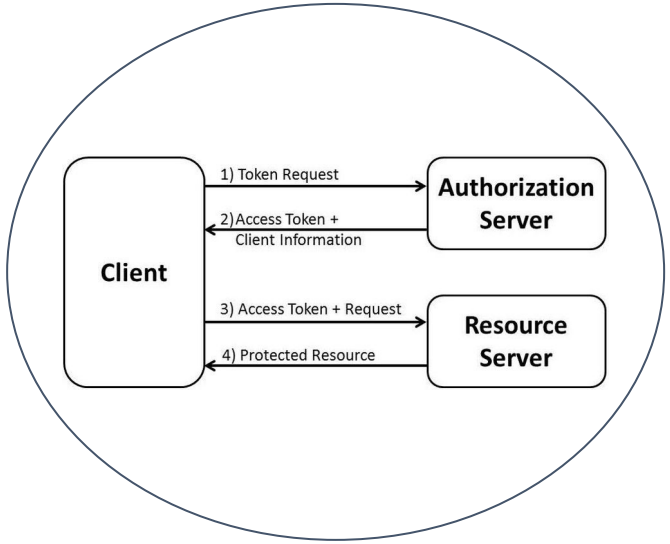
- Basic Authentication in Prometheus
- https://prometheus.io/docs/guides/basic-auth/

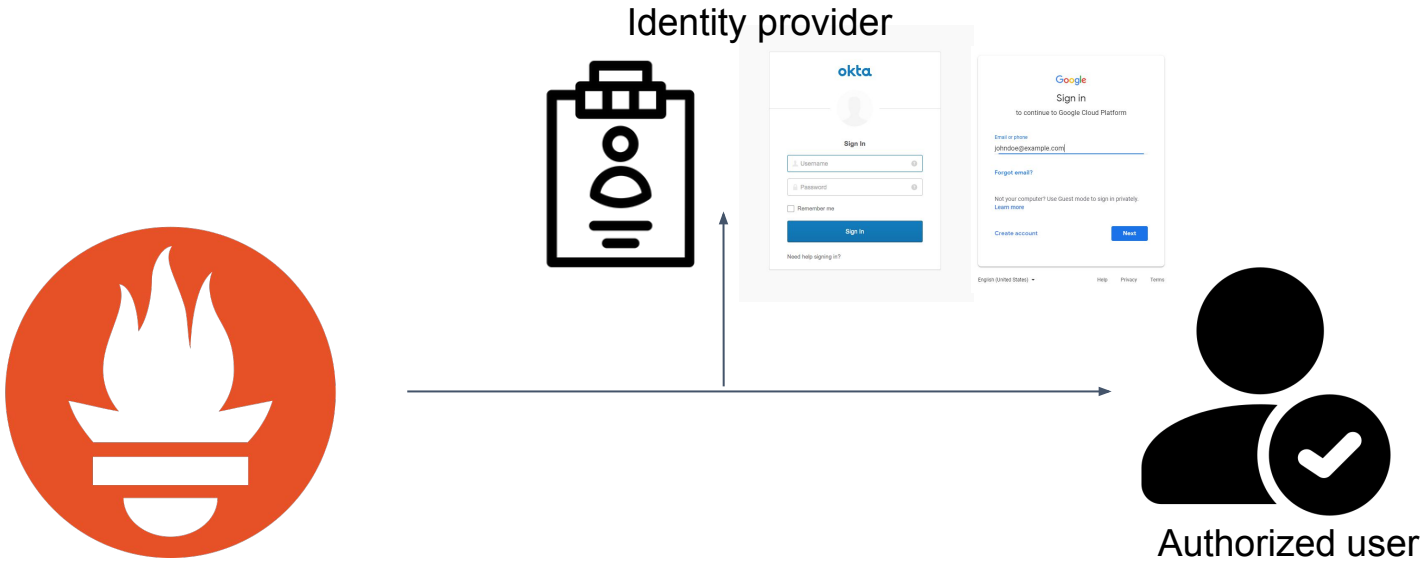


Authorized user

Securing Exporters and Prometheus from outsiders

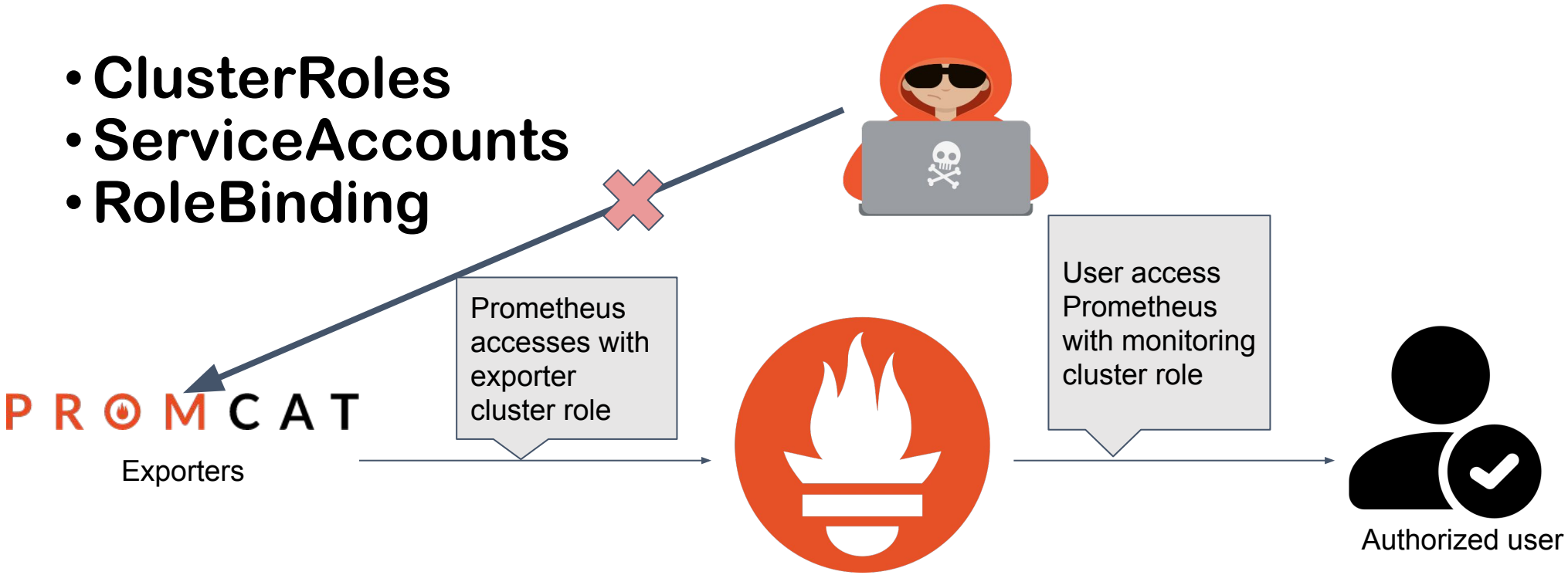
- Ingress Authentication with Oauth





Securing Exporters and Prometheus from insiders

- ClusterRoles
- ServiceAccounts
- RoleBinding



```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: prometheus
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: prometheus
subjects:
- kind: ServiceAccount
  name: prometheus
  namespace: monitoring
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: prometheus
rules:
- apiGroups: [""]
  resources:
  - nodes
  - nodes/metrics
  - services
  - endpoints
  - pods
  verbs: ["get", "list", "watch"]
- apiGroups: [""]
  resources:
  - configmaps
  verbs: ["get"]
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses
  verbs: ["get", "list", "watch"]
- nonResourceURLs: ["/metrics"]
    verbs: ["get"]
```

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: prometheus
  namespace: monitoring
```

Securing Prometheus. Lessons learned from OpenShift

Manuel Hernandez
Integrations Engineer

Jesus Angel Samitier
Integrations Engineer

Integration Engineers
PromCat.io maintainers
at Sysdig

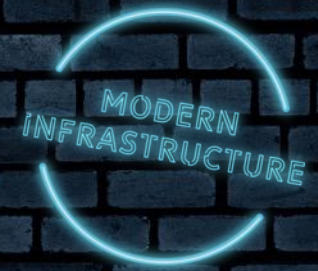
PromCon 2022 EU - Munich 8-10 Nov

sysdig

For a full description of how to use RBAC in Kubernetes to secure Prometheus and exporters, see the this talk at PromCon 2022 from Manuel Hernandez and Jesus Angel Samitier and the lighting talk at KubeCon 2018 presenting kube-rbac-proxy

Conclusion


ADDO
ALL DAY DEVOPS
caffeinated by sonatype


MODERN
INFRASTRUCTURE

Conclusion

We could think that metrics are not important in a security perspective, but we show that's not true.

It's also important to mention that the proper services Kubernetes or Prometheus advise of the problems to expose their data to the world.

- **Secure your Cloud provider with Principle of least privilege.**
  - **https://www.cisa.gov/uscert/ncas/current-activity/2020/01/24/nsa-releases-guidance-mitigating-cloud-vulnerabilities**
- **Secure your Cluster Kubernetes**
  - **https://media.defense.gov/2021/Aug/03/2002820425/-1/-1/0/CTR_Kubernetes_Hardening_Guidance_1.1_20220315.PDF**
- **Secure the Host / OS**
  - **https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-123.pdf**
- **Secure the containers**
  - **https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf**
- **Secure your code**
  - **https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-95.pdf**
- **Secure your Prometheus Metrics!**
  - **https://prometheus.io/docs/operating/security/#prometheus**

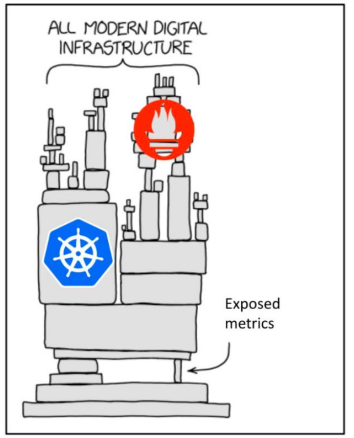
ADDO ALL DAY DEVOPS caffeinated by sonatype

Miguel Hernandez
Security Researcher
Sysdig
@MiguelHzBz

David de Torres
Manager of Engineering
Sysdig
@maellyssa

Kubernetes fingerprinting with Prometheus

KubeCon CloudNativeCon Europe 2022



Blog:
https://sysdig.com/blog/exposed-prometheus-exploit-kubernetes-kubeconeu/

Video KubeCon EU 2022:
https://www.youtube.com/watch?v=5cbbm_L6n7w

sysdig


ADDO
ALL DAY DEVOPS
caffeinated by sonatype