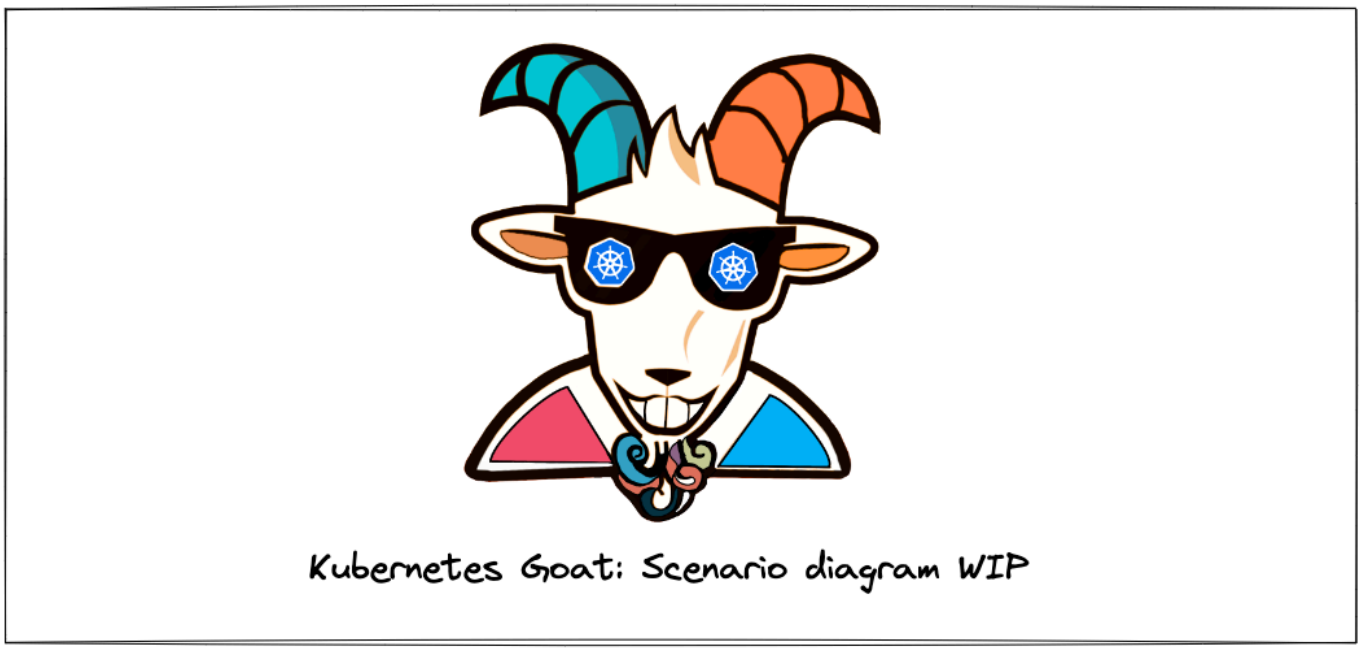
# ⎈ Gaining environment information

## 🙌 Overview

This is no different from the traditional workloads. Most of the compute instances while running the applications store sensitive information like secrets, api_keys, etc. in the environment variables. Similarly, in Kubernetes, most of the people store sensitive information like Kubernetes Secrets and the Config values in the environment variables and if an attacker can find application vulnerabilities like RCE(remote code execution) or command injection then it's game over for that secret.




By the end of the scenario, we will understand and learn the following

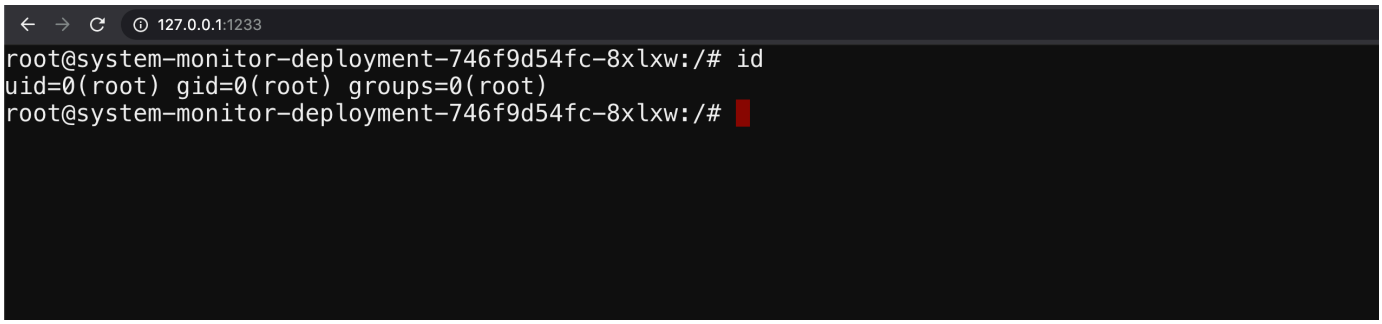
1. How to explore the environment variables and analyze
2. Gaining access to sensitive information in the container

## ⚡ The story

Each environment in Kubernetes will have a lot of information to share. Some of the key things include secrets, API keys, configs, services, and a lot more. So let's go ahead and find the vault

key!

> INFO
>
> To get started with the scenario, navigate to http://127.0.0.1:1233




## 🎯 Goal

> TIP
>
> Get the `k8s_goat_flag` flag value then you have completed this scenario. This can be found in multiple ways!

## ☐ Hints & Spoilers

<details>
<summary>✨ Not sure where the environment variables get stored?</summary>
</details>

# 🎉 Solution & Walkthrough

## 🎲 Method 1

- We can explore the container by running different commands and understand the system much better. This is more like an exploration scenario to explore the container and understand it better

- We can get the container runtime information by running the below command

```
cat /proc/self/cgroup
```

```
root@gke-kubernetes-goat-default-pool-e2db1114-k0fg:/# cat /proc/self/cgroup
12:rdma:/
11:pids:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
10:cpu,cpuacct:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
9:memory:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
8:cpuset:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
7:blkio:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
6:net_cls,net_prio:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
5:devices:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
4:hugetlb:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
3:freezer:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
2:perf_event:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
1:name=systemd:/kubepods/pod64038cec-4832-46c6-9063-2bd06d715f0c/2b10d1893e045648588dd469e5f2946bb46e05bd0dce369cb0753a3e04adc5ff
0::/system.slice/containerd.service
```

- We can get the information of the container host's information

```
cat /etc/hosts
```

- We can get the mount information

```
mount
```

- We can also look and explore the file system

```
ls -la /home/
```

- We can access the environment variables, including Kubernetes secrets mounted and service names, ports, etc.

```
printenv
```

```
root@gke-kubernetes-goat-default-pool-e2db1114-k0fg:/# printenv
LS_COLORS=
BUILD_CODE_SERVICE_PORT_80_TCP=tcp://10.0.5.168:80
POOR_REGISTRY_SERVICE_PORT_5000_TCP_PORT=5000
METADATA_DB_SERVICE_PORT=80
INTERNAL_PROXY_INFO_APP_SERVICE_PORT_5000_TCP_ADDR=10.0.1.98
SYSTEM_MONITOR_SERVICE_PORT=tcp://10.0.5.198:8080
BUILD_CODE_SERVICE_SERVICE_PORT=80
HEALTH_CHECK_SERVICE_PORT_3000_TCP_ADDR=10.0.5.201
HOSTNAME=gke-kubernetes-goat-default-pool-e2db1114-k0fg
BUILD_CODE_SERVICE_SERVICE_HOST=10.0.5.168
POOR_REGISTRY_SERVICE_PORT_5000_TCP_PROTO=tcp
METADATA_DB_SERVICE_PORT_HTTP=80
INTERNAL_PROXY_API_SERVICE_PORT_3000_TCP=tcp://10.0.1.42:3000
METADATA_DB_SERVICE_HOST=10.0.13.21
KUBERNETES_GOAT_HOME_SERVICE_SERVICE_PORT=80
HUNGER_CHECK_SERVICE_PORT_8080_TCP_ADDR=10.0.10.239
HUNGER_CHECK_SERVICE_PORT_8080_TCP_PORT=8080
INTERNAL_PROXY_INFO_APP_SERVICE_PORT=tcp://10.0.1.98:5000
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_ADDR=10.0.0.1
BUILD_CODE_SERVICE_PORT_80_TCP_ADDR=10.0.5.168
INTERNAL_PROXY_INFO_APP_SERVICE_PORT_5000_TCP_PORT=5000
SYSTEM_MONITOR_SERVICE_SERVICE_HOST=10.0.5.198
POOR_REGISTRY_SERVICE_PORT_5000_TCP_ADDR=10.0.3.5
INTERNAL_PROXY_API_SERVICE_SERVICE_PORT=3000
KUBERNETES_PORT=tcp://10.0.0.1:443
HUNGER_CHECK_SERVICE_PORT_8080_TCP=tcp://10.0.10.239:8080
PWD=/
METADATA_DB_PORT_80_TCP_ADDR=10.0.13.21
HOME=/root
SYSTEM_MONITOR_SERVICE_PORT_8080_TCP=tcp://10.0.5.198:8080
K8S_GOAT_VAULT_KEY=k8s-goat-cd2da27224591da2b48ef83826a8a6c3
SYSTEM_MONITOR_SERVICE_PORT_8080_TCP_ADDR=10.0.5.198
INTERNAL_PROXY_API_SERVICE_PORT_3000_TCP_PORT=3000
BUILD_CODE_SERVICE_PORT_80_TCP_PROTO=tcp
INTERNAL_PROXY_API_SERVICE_PORT_3000_TCP_ADDR=10.0.1.42
HEALTH_CHECK_SERVICE_PORT_3000_TCP_PORT=3000
HEALTH_CHECK_SERVICE_SERVICE_HOST=10.0.5.201
KUBERNETES_SERVICE_PORT_HTTPS=443
HUNGER_CHECK_SERVICE_PORT=tcp://10.0.10.239:8080
POOR_REGISTRY_SERVICE_SERVICE_PORT=5000
KUBERNETES_GOAT_HOME_SERVICE_SERVICE_HOST=10.0.1.69
KUBERNETES_PORT_443_TCP_PORT=443
HEALTH_CHECK_SERVICE_SERVICE_PORT=3000
BUILD_CODE_SERVICE_PORT_80_TCP_PORT=80
HEALTH_CHECK_SERVICE_PORT_3000_TCP_PROTO=tcp
KUBERNETES_GOAT_HOME_SERVICE_PORT_80_TCP=tcp://10.0.1.69:80
POOR_REGISTRY_SERVICE_SERVICE_HOST=10.0.3.5
```

- Hooray 🥳 , now we can see that it contains the Kubernetes Goat flag which is mounted as a Kubernetes secret inside the pod/container

## References

- Kubernetes Secrets
- Injecting Secrets into Kubernetes Pods via Vault Agent Containers

Edit this page