Monitor your Docker Containers

Presentation By:
   Amir Arsalan
   Behnam Loghmani

IRAN OpenStack Users Group   IRAN Docker Users Group   جامعه آزاد رایانش ابری ایران
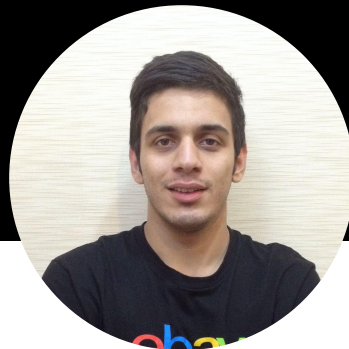
Behnam Loghmani

Cloud Architect, DevOps, System
Administrator, Platform Engineer

Email :
behnam@alcazar.cloud

Amir Arsalan

DevOps, Container Architect,
Software Developer

Email:
arsalan@alcazar.cloud

Agenda:

What is docker?
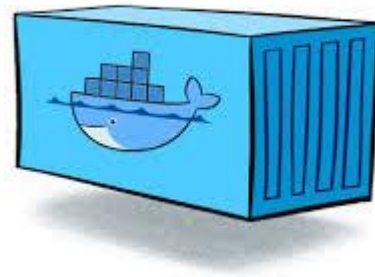Why monitoring?
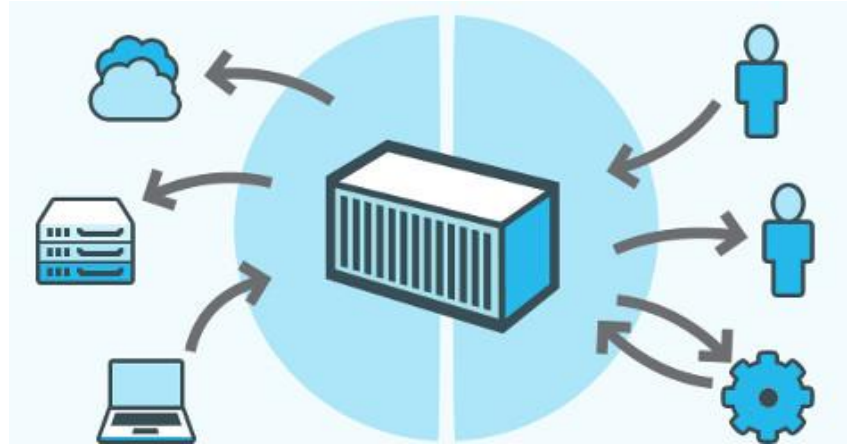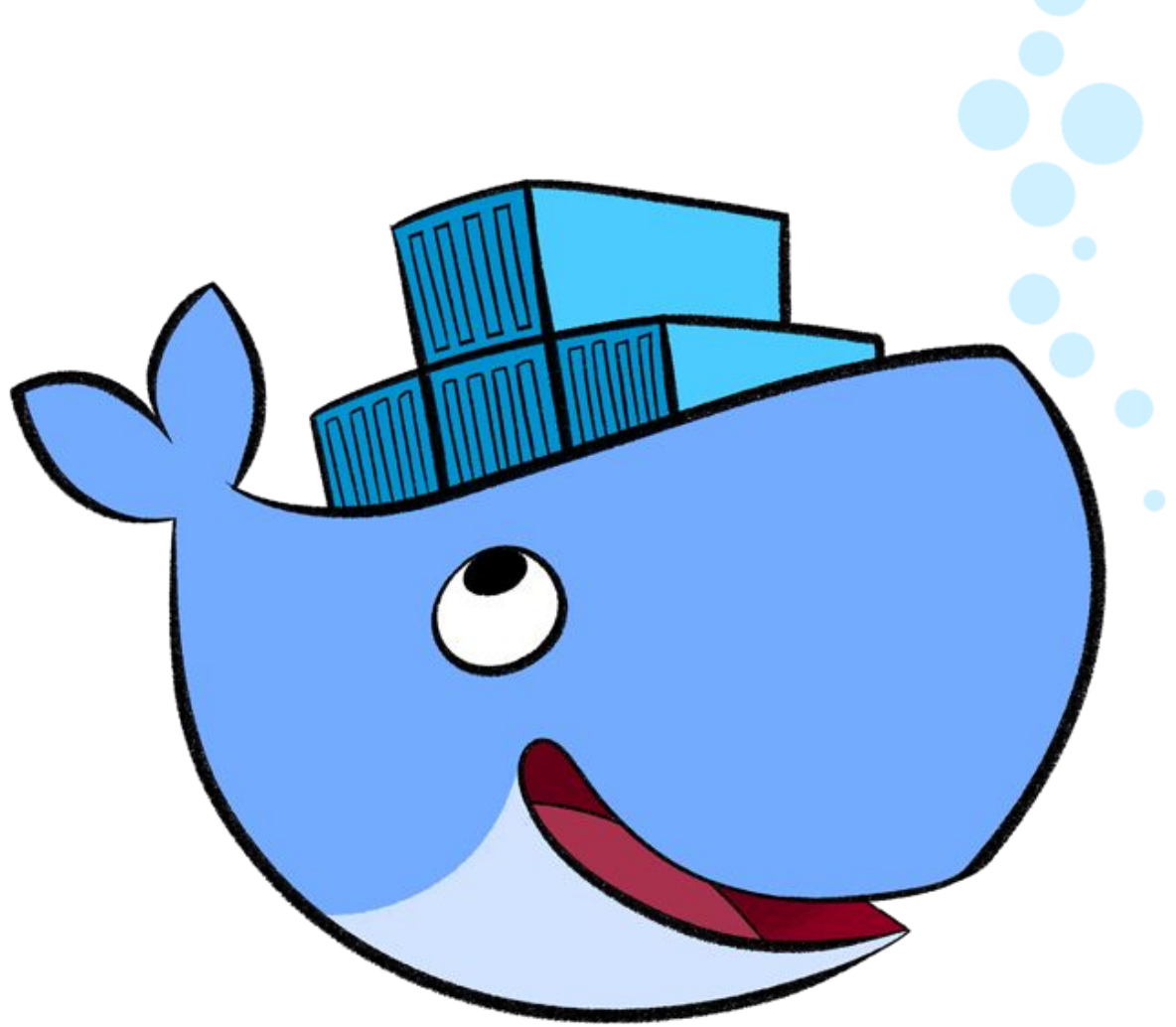Comparing Solutions
Conclusion

What is Docker ?

What is Container

A container is a lightweight **virtual runtime**. Its primary purpose is to provide software **isolation**.

What is docker

Docker is a technology that allows you to **Build**, **Run**, **Test** and **Deploy** distributed applications inside software containers.
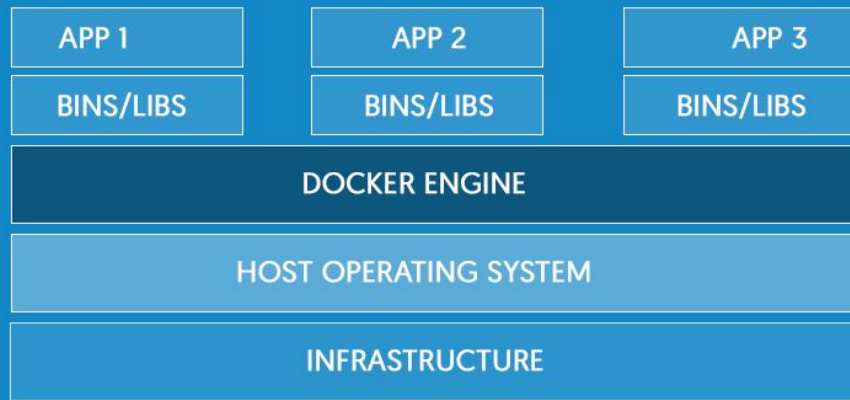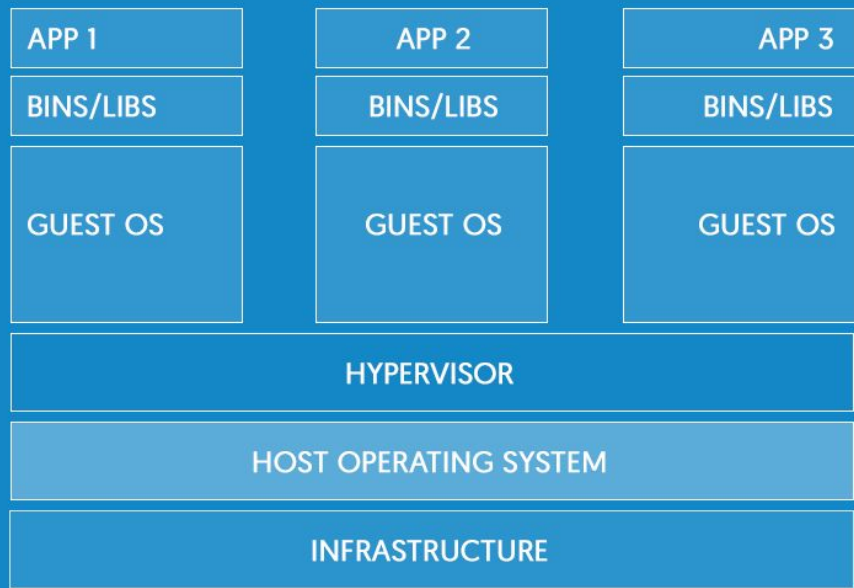
VMs vs Docker

Why Monitoring?

Monitoring is Crucial

Running software in production without monitoring is like driving without visibility: you have no idea if you're about to crash, or how to stay on the road.
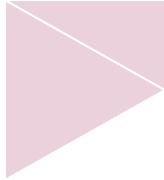
The need for monitoring is well understood, so monitoring solutions cover:

- application performance monitoring instruments your custom code to identify and pinpoint bottlenecks or errors

- infrastructure monitoring collects metrics about the host, such as CPU load and available memory

Comparing Solutions

Comparing based on the following criteria:

1) Ease of deployment

2) Level of detail of information presented

3) Level of aggregation of information from entire deployment

4) Ability to raise alerts from the data

5) Ability to monitor non-docker resources

6) Cost

Docker Stats

```
$ docker stats fervent_panini 5acfcb1b4fd1
CONTAINER          CPU %         MEM USAGE/LIMIT      MEM %        NET I/O
5acfcb1b4fd1       0.00%         115.2 MiB/1.045 GiB  11.03%       1.422 kB/648 B
fervent_panini     0.02%         11.08 MiB/1.045 GiB  1.06%        648 B/648 B
```

```
echo -e "GET /containers/[CONTAINER_NAME]/stats HTTP/1.0\r\n" | nc -U /var/run/docker.sock
```

OUTPUT

```
{
        "read": "2015-03-17T01:13:19.991994397Z",
        "network": {
        "rx_bytes": 1296,
        "rx_packets": 16,
        "rx_errors": 0,
        "rx_dropped": 0,
        "tx_bytes": 648,
        "tx_packets": 8,
        "tx_errors": 0,
        "tx_dropped": 0
        },
        "cpu_stats": {
        "cpu_usage": {
            "total_usage": 63893077,
            "percpu_usage": [
                63893077
            ],
            "usage_in_kernelmode": 40000000,
            "usage_in_usermode": 10000000
        },
```

Docker Stats

Score Card:

1. Easy of deployment: ☆☆☆☆☆
2. Level of detail: ☆☆☆☆☆
3. Level of aggregation: **none**
4. Ability to raise alerts: **none**
5. Ability to monitor non-docker resources: **none**
6. Cost: **Free**

cAdvisor

cAdvisor is a useful tool that is trivially easy to setup, it saves us from having to ssh into the server to look at resource consumption and also produces graphs for us.
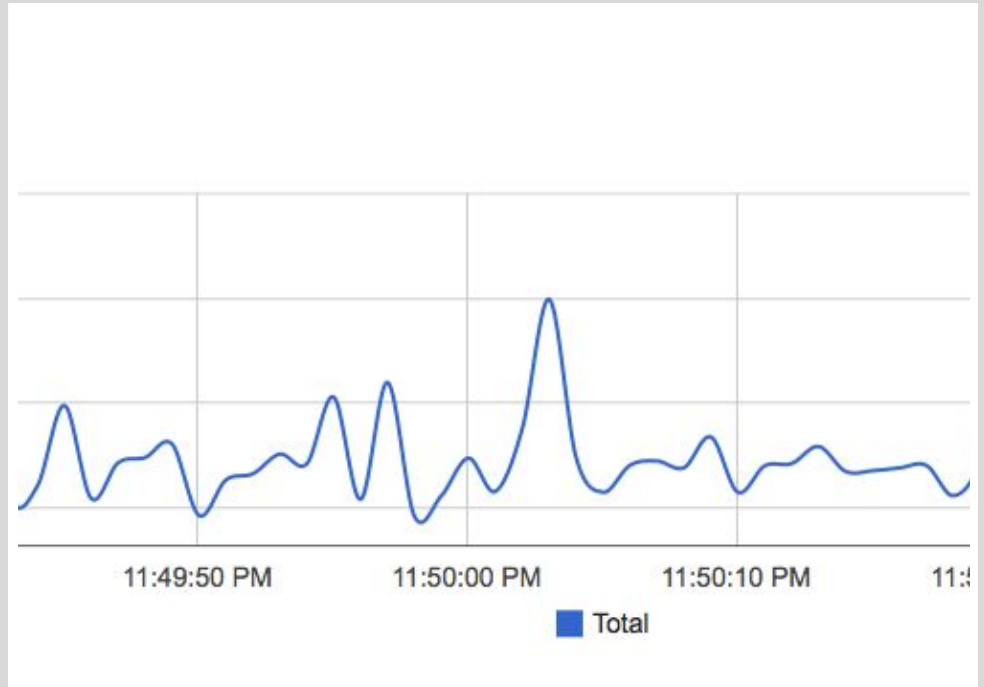
```
docker run                                             \
  --volume=/:/rootfs:ro                                \
  --volume=/var/run:/var/run:rw                        \
  --volume=/sys:/sys:ro                                \
  --volume=/var/lib/docker/:/var/lib/docker:ro  \
  --publish=8080:8080                                  \
  --detach=true                                        \
  --name=cadvisor                                      \
  google/cadvisor:latest
```

cAdvisor

Score Card:
1. Easy of deployment: ☆☆☆☆☆
2. Level of detail: ☆☆
3. Level of aggregation: ☆
4. Ability to raise alerts: **none**
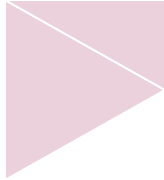5. Ability to monitor non-docker resources: **none**
6. Cost: **Free**

Scout is a **hosted** monitoring service which can aggregate metrics from many hosts and containers and present the data over longer time-scales.

advantage of using Scout over CAdvisor is that it has a large set of plugins which can pull in other data about your deployment in addition to docker information.

One drawback of Scout is that it does not present detailed information about individual containers on each host like CAdvisor can.

```
docker run -d  --name scout-agent                             \
    -v /proc:/host/proc:ro                                    \
    -v /etc/mtab:/host/etc/mtab:ro                             \
    -v /var/run/docker.sock:/host/var/run/docker.sock:ro      \
    -v `pwd`/scoutd.yml:/etc/scout/scoutd.yml                 \
    -v /sys/fs/cgroup/:/host/sys/fs/cgroup/                   \
    --net=host --privileged                                   \
    soutapp/docker-scout
```

Scout

Score Card:

1. Easy of deployment: ☆☆☆☆
2. Level of detail: ☆☆
3. Level of aggregation: ☆☆☆
4. Ability to raise alerts: ☆☆☆
5. Ability to monitor non-docker resources: **Supported**
6. Cost: **10$/host**

DATADOG

DataDog is a **hosted** monitoring service.

DataDog, which addresses several of the short-comings of Scout as well as all of the limitations of CAdvisor.

DataDog's alerting system is a lot more flexible and detailed then Scout's.

Datadog collects metrics about CPU usage, memory and I/O for all containers running in the system. In addition you get counts of running and stopped containers as well as counts of docker images.

```
docker run -d --privileged --name dd-agent              \
    -h `hostname`                                       \
    -v /var/run/docker.sock:/var/run/docker.sock        \
    -v /proc/mounts:/host/proc/mounts:ro                \
    -v /sys/fs/cgroup/:/host/sys/fs/cgroup:ro           \
    -e API_KEY=YOUR_API_KEY datadog/docker-dd-agent      \
```

Data Dog

Score Card:
1. Easy of deployment: ☆☆☆☆☆
2. Level of detail: ☆☆☆☆☆
3. Level of aggregation: ☆☆☆☆☆
4. Ability to raise alerts: **Supported**
5. Ability to monitor non-docker resources: ☆☆☆☆☆
6. Cost: **15$/host**

Scout and Datadog provide centralized monitoring and alerting however both are hosted services that can get expensive for large deployments.

Unfortunately sensu does not have any docker support out of the box.

The alerting is not as advanced as DataDog or Scout, as you are only able to alert on checks failing on individual hosts.

Sensu Monitoring Framework

Score Card:
1. Easy of deployment: ☆
2. Level of detail: ☆☆☆☆
3. Level of aggregation: ☆☆☆☆
4. Ability to raise alerts: **Supported but limited**
5. Ability to monitor non-docker resources: ☆☆☆☆☆
6. Cost: **Free**

Prometheus

It is a self-hosted set of tools which collectively provide metrics storage, aggregation, visualization and alerting.

Prometheus on the other hand is a pull based server which expects monitored servers to provide a web interface from which it can scrape data.

Prometheus uses a very simple format for input data and can ingest from any web endpoint which presents the data.

Where Prometheus is lacking is in level of polish and ease of deployment.

Prometheus

Score Card:

1. Easy of deployment: ☆☆
2. Level of detail: ☆☆☆☆☆
3. Level of aggregation: ☆☆☆☆☆
4. Ability to raise alerts: ☆☆☆☆
5. Ability to monitor non-docker resources: ☆☆☆☆☆
6. Cost: **Free**

Sysdig cloud

Sysdig cloud is a hosted service that provides metrics storage, aggregation, visualization and alerting.

be warned if you would like to use Sysdig cloud on non-mainstream kernels you may have to get your hands dirty with some system hacking.

The metrics are aggregated for the host as well as broken down per container.

The alerting system in the Sysdig cloud is among the best we have seen so far, however, the inability to target different email addresses for different alerts is problematic.

Sysdig cloud

Score Card:
1. Easy of deployment: ☆☆☆
2. Level of detail: ☆☆☆☆☆
3. Level of aggregation: ☆☆☆☆☆
4. Ability to raise alerts: ☆☆☆☆
5. Ability to monitor non-docker resources: **Supported**
6. Cost: **20$/host**

Conclusion

From our research so far DataDog seems to be the best-in-class system for monitoring docker deployments.

However, at $15 per host the system can get expensive for large deployments. For larger scale, self-hosted deployments Prometheus is able to fulfill most requirements however the complexity in setting up and managing may be prohibitive.
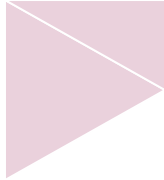
Open Source Mixed

Based on our experiences, monitoring systems must be fully customizable to fit different companies needs.
So with combining open source applications you can build whatever you want.

maintaining, updating, and keeping system uptime are problematic in these systems but our team do it in your company and giving full support to you.

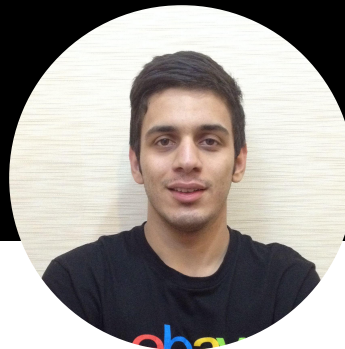Should you have any questions, please do not hesitate to contact us.

https://goo.gl/CSAR6r
https://telegram.me/alcazarteam

Behnam Loghmani

Cloud Architect, DevOps, System
Administrator, Platform Engineer

Email :
behnam@alcazar.cloud

Amir Arsalan

DevOps, Container Architect,
Software Developer

Email:
arsalan@alcazar.cloud

Thank You