Cloud & container monitoring

04.05.2018, Lars Michelsen
Check_MK Conference #4
Some cloud definitions

- **Software-as-a-Service (SaaS)**
  - ‘Applications’
  - Office 365, Salesforce, Netflix

- **Platform-as-a-Service (PaaS)**
  - ‘Runtime environment for applications’
  - Azure App Services, AWS Elastic Beanstalk

- **Infrastructure-as-a-Service (IaaS)**
  - ‘Server in the cloud’
  - Amazon EC2, Azure VM, Google CE, Amazon S3, Azure Storage
Containers – what is it?

Virtual machines

- VM 1
  - App 1
  - App 2
  - Guest OS*

  Hypervisor

- VM 2
  - App 1
  - App 2
  - Guest OS*

  Infrastructure

Containers

- Container 1
  - App 1
  - bins | libs

- Container 2
  - App 2
  - bins | libs

  Containerization

  OS

  Infrastructure

Virtualization runs on a ‘hypervisor’

- Enables portability, increases utilization of infrastructure – but resource burden

Containers use the nodes OS kernel

- Enables portability at high-infrastructure efficiency

* incl. Binaries and libraries
What is docker on my host?

Node

Containers

Images

Docker registry

Daemon
From one to many, many containers

Kubernetes Master
Container orchestration
Implications for monitoring

1. Cloud-APIs (for PaaS / SaaS) → Plugins
2. Containers as additional layer → Plugins
3. Fast changing environments → Dynamic configuration
4. Single metrics become less relevant → Aggregated metrics
0) Intro

1) Cloud monitoring

2) Container monitoring

3) Dealing with dynamics

4) Metrics
## IaaS is standard business

<table>
<thead>
<tr>
<th></th>
<th>Public Cloud</th>
<th>Private Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SaaS</strong></td>
<td>Planned via APIs</td>
<td>Planned via APIs</td>
</tr>
<tr>
<td><strong>PaaS</strong></td>
<td>Planned via APIs</td>
<td>Planned via APIs</td>
</tr>
<tr>
<td><strong>IaaS</strong></td>
<td>![Check_MK Agent]</td>
<td>![Check_MK Agent]</td>
</tr>
</tbody>
</table>

- **IaaS**: Standard business via Check_MK Agent.
Monitoring PaaS & SaaS via APIs

We build what is needed

Special-Agent and Checks
- Working on Azure (e.g. SQL database)
- Involved in several migrations (e.g. multi region Azure)
- What do you need? AWS, Azure services, OpenStack, ...?
0) Intro

1) Cloud monitoring

2) **Container monitoring**

3) Dealing with dynamics

4) Metrics
Basics are already possible

Existing options...

- Checks (Docker on Check_MK Exchange)
- Process checks and other resources of docker nodes
- Agent in docker container

... but

- Need to care about configuration
... but currently developing much broader feature set

- Container specific metrics
- Dynamic configuration
- Aggregated metrics
## Container monitoring future

<table>
<thead>
<tr>
<th>Phase I</th>
<th>Phase II</th>
<th>Phase III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native container support</td>
<td>Container orchestration</td>
<td>Mgmt. for container &amp; container orchestration</td>
</tr>
</tbody>
</table>

![Docker](docker.png)  ![Kubernetes](kubernetes.png)  ![Rancher](rancher.png)
Docker native Check- & Inventory Plug-Ins

**Phase I**

**Node checks:** System status, #images, #containers, disk usage

**Node inventory:** Version, labels, networks

**Phase II**

**Container checks:** CPU, memory, disk IO, traffic, uptime, health

**Container inventory:** Node running on, labels, networks

**Phase III**

**Image inventory:** Time created, labels, size, #containers (state)
Agent with batteries included

Phase I

- `ps check` is aware of namespaces
- `veth interfaces` are totally ignored

Phase II

- `container mounts` are totally ignored

Phase III

- `docker ext.` included in agent
How Check_MK gets docker data

Phase I

How to get the data?

Node agent?

Phase II

Cont. agent?

Yes

Phase III

No

Execute Nodes Agent
Contact via Node
How Check_MK gets docker data

Phase I

Phase II

Phase III

How to get the data?

Node agent?

Cont. agent?

Yes

Yes

Install Agent in Image Contact via Node

No

Execute Nodes Agent Contact via Node
How Check_MK gets docker data

Phase I

How to get the data?

Yes

Phase II

Cont. agent?

Yes

Install Agent in Image
Contact via Node

No

Phase III

Execute Nodes Agent
Contact via Node

No

Install agent in Image
Contact agent via network
Short demo!

Phase I

Phase II

Phase III

CENTER FOR THE STUDY OF "MURPHY'S LAW"

CLOSED TODAY BECAUSE EVERYTHING THAT COULD GO WRONG, DID GO WRONG.

SCHWADRON
Monitoring of container orchestration tool Kubernetes

<table>
<thead>
<tr>
<th>Phase I</th>
<th>Phase II</th>
<th>Phase III</th>
</tr>
</thead>
</table>

- **Pod deployments**: Current vs. available Pods
- **Running pods**: Per node, per replica set
- **Node**: Resource request vs. limits
Monitoring of management tools for container orchestration

Only idea stage yet – your feedback is welcome
... and then ...

Continuously improve Phase I-III

More plugins

Phase I / II / III

Support tools, e.g. Docker Swarm
0) Intro

1) Cloud monitoring

2) Container monitoring

3) Dealing with dynamics

4) Metrics
Requires highly dynamic configuration

Why?

- Volatile environment
- Monitoring configuration thus needs to adapt very dynamically

Dynamic configuration (DCD)

- Focus on containers
- Nodes & Kubernetes later
- Enterprise Edition only
Dynamic Configuration Daemon (DCD) architecture

Sources
- Check_MK Agent with Docker Plug-in
- Kubernetes API
- Cloud APIs (e.g. AWS)

Central site
- Check_MK
- DCD
- Connector: Docker “unmanaged”
- Connector: Kubernetes
- Connector: AWS
- WATO-API
- WATO
- Check_MK Base
- Core
- ...

Remote sites
- Check_MK
- WATO-API
- WATO
- Check_MK Base
- Core
- ...

WATO
- Sources
- Central site
- Remote sites
Also useful beyond containers

- **Virtual machines**: Ask vCenter for VMs
- **Network scan**: Ask the network for hosts
- **LDAP**: Ask for users or hosts
0) Intro

1) Cloud monitoring

2) Container monitoring

3) Dealing with dynamics

4) Metrics
Static containers: Standard metrics

Static containers

“well known” hosts

Business as usual
Dynamic containers: Which metrics are relevant?

Dynamic pool of containers

Aggregated metrics
Averages or total values over all containers

Total CPU load

Understand performance & trends
Dynamic containers: Which metrics are relevant?

Dynamic pool of containers

Total CPU load

Individual CPU load

Individual metrics
Value per container

Understand faults
What we will do with these metrics

### Aggregated metrics
- Collect with regular monitoring
- Independent of individual hosts
- Not limited to containers
- Make configurable via GUI

### Container metrics
- Volatility per container
- Defines resolution
- Defines lifetime of metrics
Getting ready for a new world.

Join us!