Kubernetes Monitoring Revisited

Checkmk Conference #7 – June 15 - 16 2021, online

Martin Hirschvogel
Head of Product Marketing & Product Owner Kubernetes
tribe29 GmbH
Kubernetes has come to stay. Probably.

Please indicate if your company/organization currently uses, or has future plans to use, containers for any of the below options.
Poll: Are you / your company using Kubernetes?

- Nope
- Yes, Vanilla Kubernetes
- Yes, OpenShift
- Yes, ...
- ...
- Yes, but my distribution is not listed in here
01 Status quo

02 Revisiting the experience from start to finish

03 How to monitor Kubernetes properly

04 Timeline
Status quo: K8s Monitoring with Checkmk

Native Kubernetes Monitoring

- Easy to set-up, requires only access to API
- Combined with Checkmk agents on the nodes, good for cluster health monitoring
- Not optimal for detailed performance analysis (e.g. CPU load on pod x)

Prometheus Integration (*new*)

- Ideal if Prometheus is already set up and configured within the cluster
- Minimized “monitoring load” in cluster - no double monitoring
- Provides detailed insights into containers and pods for performance analysis
Our vision: A helpful, easy-to-use K8s monitoring

Issues of most monitoring tools (incl. us)

- (Useless) Information overload
- No good structure
- Cumbersome installation + configuration
- No good alerting

Our vision

- Shows you the information you need in an easy-to-understand manner
- Alerts you when something in your cluster needs to be fixed
- Is easy to install and configure
- Follows Kubernetes-native principles
Terminology explained: Pods & Deployments

My_Deployment
Replicas = 4

IP: 10.60.1.52
IP: 10.60.2.25
IP: 10.60.0.5
IP: 10.60.1.14
Terminology explained: Nodes & Cluster

My_Kube_Cluster
CPU: 40000
RAM: 820 GB

Node_001
CPU: 3000
RAM: 64 GB
Terminology explained: Persistent volumes
01 | Status quo

02 | Revisiting the experience from start to finish

03 | How to monitor Kubernetes properly

04 | Timeline
Revisiting the exp from start to finish

Install Checkmk  Prepare K8s  Configure  Monitor  Alert
Today ... in K8s

- Make Docker image available in own registry
- Write own manifest / helm chart etc.

The wish

- Infrastructure as Code - reproducible, one command installation
- Simple update procedure

Our plan

- All Checkmk editions available in Docker registry (not only Checkmk Raw)
- Pre-defined manifests & helm-charts enabling a one click deployment and update

Available today: registry.checkmk.com + github.com/tribe29/kubernetes
Install Checkmk
Prepare K8s
Configure
Monitor
Alert

Available today: registry.checkmk.com + github.com/tribe29/kubernetes

registry.checkmk.com/enterprise/
- check-mk-enterprise:2.0.0p5
- check-mk-enterprise:2.0.0-2021.05.27
  =
- check-mk-enterprise:2.0.0-latest
- check-mk-enterprise:2021.05.27
  =
- check-mk-enterprise:master-latest

registry.checkmk.com/managed/
- check-mk-managed:2.0.0p5
  =
- check-mk-managed:2.0.0-latest

registry.checkmk.com/free/
- check-mk-free:2.0.0p5
  =
- check-mk-free:2.0.0-latest
Install Checkmk

Available today: registry.checkmk.com + github.com/tribe29/kubernetes
Install Checkmk

Available today: registry.checkmk.com + github.com/tribe29/kubernetes

```yaml
spec:
  replicas: 1
  selector:
    matchLabels:
      app: checkmk
  strategy:
    type: Recreate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: checkmk
    spec:
      serviceAccount: checkmk
      containers:
      - image: registry.checkmk.com/enterprise/check-mk-enterprise:2.6.0p2
        ImagePullPolicy: IfNotPresent
        name: checkmk
        ports:
          - containerPort: 5080
            name: webport
            protocol: TCP
```
Today ...

- Set-up ClusterRole, ClusterRoleBinding & ServiceAccount
  or
- Set-up Prometheus and open it to Checkmk

The wish

- Keep it simple
- Infrastructure as Code - reproducible, one command set-up

Our plan

- Checkmk K8s agent
- Deployment with pre-built manifests / helm charts
Today ...

- Create dummy host
- Set up special agent
- Get token / certificate
- Set-up dynamic config
- ... discover services!
  + (optional)
- Piggyback renaming
- Granular dynamic config rules & folder concept

The wish

- Straight-forward configuration with few steps
- Don’t have to be an advanced user to use it

Our plan

- Simple guided workflow incl pre-configured dynamic config set-up with folders
- Naming scheme for cluster, objects & namespaces
- Consistent labels for filters / visualization and alerting
Today ...
- Flat view: Each object = host
- Many, many hosts
- No structure & relationships

The wish
- Immediately see the health and performance of the cluster and its components
- Drill-down for debugging

Our plan
- Structure information
- Dashboards
Today ...
- Adapt rules for alerts

The wish
- Get an alert automatically, if & when there is a problem
- Have K8s expert knowledge already in the monitoring

Our plan
- Opinionated monitoring with K8s-specific alerting
01 | Status quo
02 | Revisiting the experience from start to finish
03 | How to monitor Kubernetes properly
04 | Timeline
Kubernetes isn’t hard, they said

- It is complicated, and it gets even more complicated
- In production more complex concepts need to be handled
- Kubernetes is like a datacenter - computing, networking, storage, applications - all in one (person)
- Understanding everything is challenging
- On top, things are changing quickly:
  If you read about it, it’s already deprecated!
- For most K8s monitoring set-ups: you need to know each object, the relevant metrics, good thresholds for alarming

Sources:
https://www.reddit.com/r/kubernetes/comments/nfbqag/kubernetes_isnt_that_hard_they_said_youll_have_no/
But... I have Prometheus!
Different roles, different monitoring needs

- Application
- Application
- Application

- Application infrastructure
- Network infrastructure
- Cluster infrastructure
## Monitoring the cluster infrastructure

<table>
<thead>
<tr>
<th>Control plane</th>
<th>Latency? All components healthy? Certificates ok?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volumes</td>
<td>Persistent volume running out of space? Persistent volume claims pending?</td>
</tr>
<tr>
<td>Nodes</td>
<td>Kubelet running? All nodes healthy? Auto-scaling working?</td>
</tr>
<tr>
<td></td>
<td>CPU &amp; memory highly utilized? Capacity remaining for new pods?</td>
</tr>
<tr>
<td></td>
<td>Enough space in filesystem? Network and Disk I/O ok?</td>
</tr>
<tr>
<td></td>
<td><strong>Pods evenly distributed among nodes? Enough room for more pods?</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Nodes pool configured to needs (balance between free mem &amp; CPU)?</strong></td>
</tr>
</tbody>
</table>

*Only for self-managed*
Monitoring the application infrastructure

**Pods**
- Any pods in bad phase?
- CPU or memory usage in relation to limits ok?
- Restart rate too high?

Any pods consuming excessive CPU / mem?
Horizontal pod auto-scaling ok?
Are pods utilizing all cores?
Monitoring the networking

**ServiceMesh**  
*Success rate, error rates, latency, requests*

**Service**  
*Monitor corresponding pods.*

**Ingress**  
*Working? Accessible from the outside? Too many requests? Error rate? Latency?*
The architecture

- **Master Node**: ETCD, Scheduler, Controller Manager, K8s API Server

- **Worker Nodes**:
  - **Checkmk K8s Cluster Agent**
  - **Checkmk K8s Agent**
  - **Checkmk K8s Agent**
  - **Custom cAdvisor Container**
  - **Checkmk K8s Agent Container**

- **Applications**
| 01 | Status quo |
| 02 | Revisiting the experience from start to finish |
| 03 | How to monitor Kubernetes properly |
| 04 | Timeline |
Auto-set-up of K8s monitoring

Well-structured host & services in Checkmk Cluster dashboard

Prepare K8s

Checkmk K8s agent with support for vanilla K8s

Configure

Install Checkmk

Checkmk Registry

Checkmk Registry

Checkmk K8s agent with support for vanilla K8s

Monitor

Alert

Opinionated monitoring with expert thresholds

Easy-to-use event monitoring

Dashboards for deployments, nodes etc.

Support for further K8s distros, e.g. EKS, OpenShift

Auto-set-up of K8s monitoring

Checkmk Registry

Install Checkmk

Checkmk 2.1

Checkmk Registry

Checkmk K8s agent with support for vanilla K8s

Monitor

Alert

Opinionated monitoring with expert thresholds

Easy-to-use event monitoring

Dashboards for deployments, nodes etc.

Support for further K8s distros, e.g. EKS, OpenShift

Auto-set-up of K8s monitoring

Checkmk K8s agent with support for vanilla K8s

Prepare K8s

Checkmk Registry

Install Checkmk

Checkmk 2.1

Checkmk Registry

Checkmk K8s agent with support for vanilla K8s

Prepare K8s

Checkmk K8s agent with support for vanilla K8s

Configur
Questions?

Join me in our #cmk7-speakers-corner after this session for further questions.
Thank You!

tribe29 GmbH
Kellerstraße 29
81667 München
Deutschland

Web — tribe29.com
E-Mail — info@tribe29.com