KubeCon | CloudNativeCon

North America 2021

RESILIENCE

REALIZED

How NOT to start with K8s

Christian Heckelmann

2018 - How everything started...

- We want to use containers!
- PoC with K8s and Docker Swarm
- Needs to run on-prem (CentOS on VMWare and Xen)
- Kubernetes 1.9
- Don‘t start by your own – You are building a little Datacenter within a Datacenter!

KubeCon | CloudNativeCon

North America 2021

RESILIENCE

REALIZED

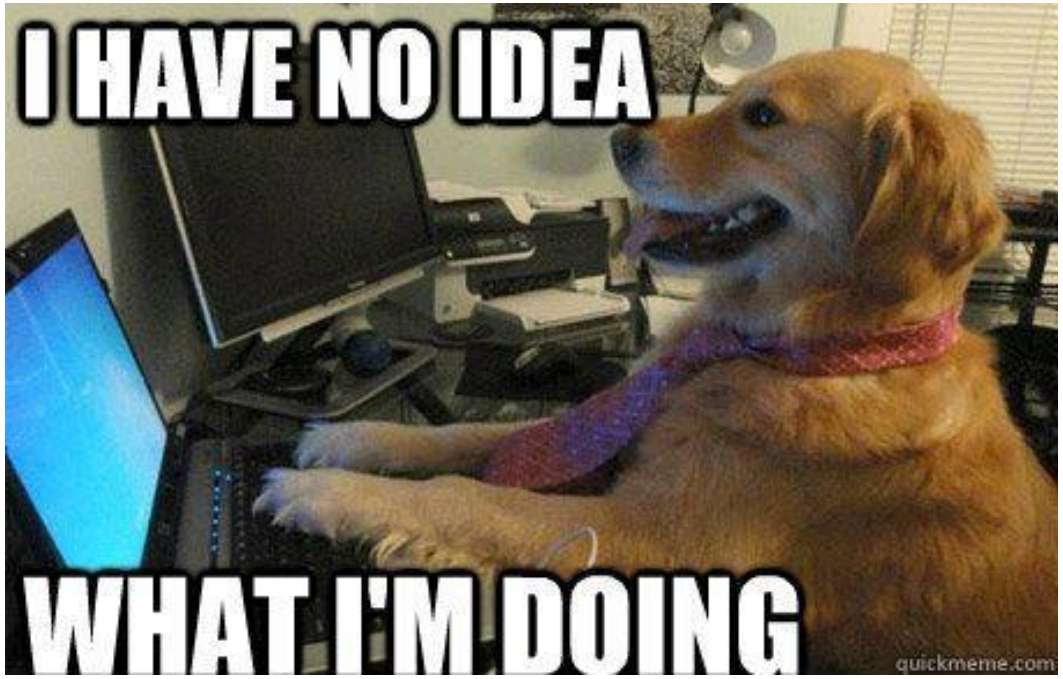
Infrastructure and Operations

Do not install K8s from scratch

- Failed hard with my first cluster while upgrading
- Switched to Rancher

Other options (for on-prem)

- VMWare Tanzu
- Kubespray



Automation everywhere

- Try to automate provisioning of clusters
- Provision a cluster with a UI is easy but what if you need to deploy 10 - 20 - 100 Clusters?
- You don‘t want to run „kubectl apply –f“ X-Times to get your cluster ready

Network CNI and Configuration

flannel

weave net

cilium

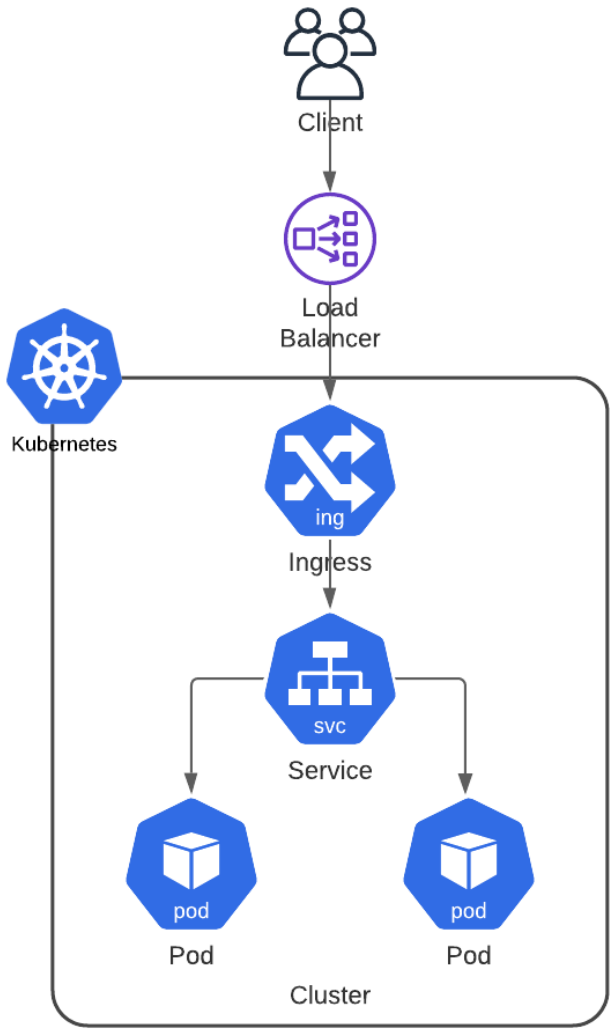
canal

- The right CIDR Range for pod and service network
- Network Policies
- Encrypt data in transit
- Know at least the basics (like DNS in K8s)

https://kubernetes.io/docs/concepts/cluster-administration/networking/

How we could reach our service?

- Loadlbalancer (F5, MetalLB)
- Which protocols (https/tcp)?
- Do you want to split your traffic (internal/external)?
- Security (Web Application Firewall)
- SSL Certificates
  - Certmanager with LetsEncrypt



We need a way to store some files...

- Keep it simple!
- Backup
- What kind of workload do you expect?
- What is your Infrastructure already able to provide? – Ask the experts!

https://kubernetes.io/docs/concepts/storage/storage-classes/

https://medium.com/volterra-io/kubernetes-storage-performance-comparison-v2-2020-updated-1c0b69f0dcf4

RBAC

- Use Service Accounts from the beginning
- Use separate accounts for deployment/monitoring and operation tasks
- Don’t share credentials across teams
- Don’t use “kubeconfig” files in your Pipelines

Logging and Monitoring

- Infrastructure
- Cluster services like etcd, API-Server, DNS
- Kubernetes events like CrashLoopBackOff, ImagePullBackoff, OOMKilled
- Requested and used resources
- Logs (where should I write my application log?)

All ingresses are down!?

Check for common mistakes and reject changes in K8s before it’s getting worse like with Policies:

- Ingress validation
- Pod Security (e.g. Disallow Privileged Containers)
- Required Resource Limits/Requests
- Health checks

kubectl

Operators (and Developers) should be familiar with the common kubectl commands.

KubeCon | CloudNativeCon

North America 2021

RESILIENCE

REALIZED

Development & Deployment

Wrong motivation

“I run it in K8s cause I don’t want to request a VM”

Local Development

Dev: Kubernetes is down, my deployment is not working!!!!!

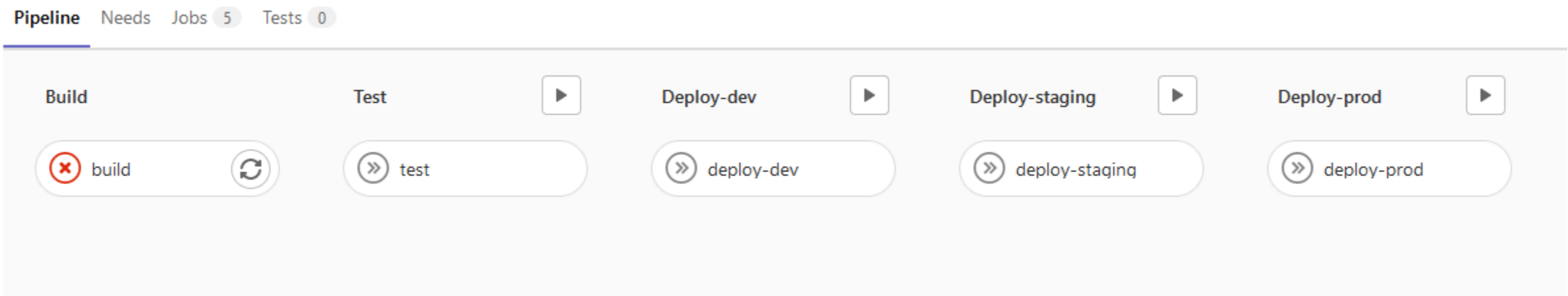
Ops: Ok your Container is not working at all

- Install Docker on your local machine and build the image
- Use KinD, MiniKube, K3s to test your application and deployment

:latest - why isn't it working anymore?

Never ever use the latest tag either in your CI/CD Pipeline or in your Kubernetes deployments.

If a new version on an image is getting pulled, it could break the container or workload.



Private Registry & Base images

Create a library of own base images (private registry) for the tools you are using, to have full control

- Check the images for vulnerabilities - Harbor
- Add additional configurations to the image (e.g. Java8 cgroup memory bug)
- No need to pull from docker hub and hit the rate limit (and don't use ImagePullPolicy: Always)
- Block/Allow lists if possible

```
~ docker pull untrusted/withacryptominer:latest
```

Not using the „power“ of Kubernetes

“Why Autoscaling? I have two Pods!”

Health checks

Make sure your application is configured with a proper health check.

If you get ever asked to “restart” a container within a cluster to mitigate an issue, you must ask them why this is necessary.

Use a policy engine like OPA or Kyverno to block those deployments

+IT-Ops - Please restart the ECG Algorithm service in Kubernetes.

Thanks,

Resource Limits and Requests

- Kubernetes decides based on your resource requests where to run the pod in your cluster
- If limits are not defined, the pod can “explode” and affect another workload on the worker node
- Add default memory requests and limits on your cluster or namespaces
- Prevent requests like CPU: 12 – memory 128GB
- Dev: But I don’t know how many resources my container is consuming? → Local Development, kubectl top pod

Env Variables for Config and Secrets?!

- Do not put Secrets in Environment Variables
- Create a standard how your application configuration should be managed
- Use a secret store like vault or sealed secrets
- K8s secrets are not encrypted!
- Use configmaps for configuration

helm

- Over 90% of deployments are the same
- Provide Templates which can be adopted easily
- Preventing common mistakes “forgotten health checks, accidently exposed services…”

```
apps/v1
yment

i-service

ubernetes.io/name: api-service
ubernetes.io/instance: {{ .Release.Name }}
ubernetes.io/managed-by: {{ .Release.Service }}
kubernetes.io/part-of: keptn-{{ .Release.Namespace }}
kubernetes.io/component: {{ include "control-plane.name" .
kubernetes.io/version: {{ .Values.apiService.image.tag | d
m.sh/chart: {{ include "control-plane.chart" . }}

tor:
tchLabels:
app.kubernetes.io/name: api-service
app.kubernetes.io/instance: {{ .Release.Name }}
licas: 1
plate:
etadata:
  labels:
    app.kubernetes.io/name: api-service
    app.kubernetes.io/instance: {{ .Release.Name }}
    app.kubernetes.io/managed-by: {{ .Release.Service }}
    app.kubernetes.io/part-of: keptn-{{ .Release.Namespace }}
    app.kubernetes.io/component: {{ include "control-plane.nam
```

Summary

- Invest in Training
- Document how to use the platform
- Provide templates to avoid common mistakes
- Reject everything which does not meet the required standards
- Involve SecOps from the beginning

Ping me!

https://www.linkedin.com/in/christian-heckelmann-82375a25/

@wurstsalat

checkelmann@gmail.com