



Certified Kubernetes Application Developer

Webinar Series 3/3



Session duration: 4:00 PM to 5:50 PM

Online Meeting Rules

- If not muted, mute yourself.
- Ask your questions in the chat window.
- Use mic if you are explicitly asked.
- If you want to show something, we will make you a presenter
- If you like you can activate your camera. We love to see you all 😊
- Please do not spam the chat window.
- Do not post inappropriate content.
- Have a fun. 😊





Azure Meetup

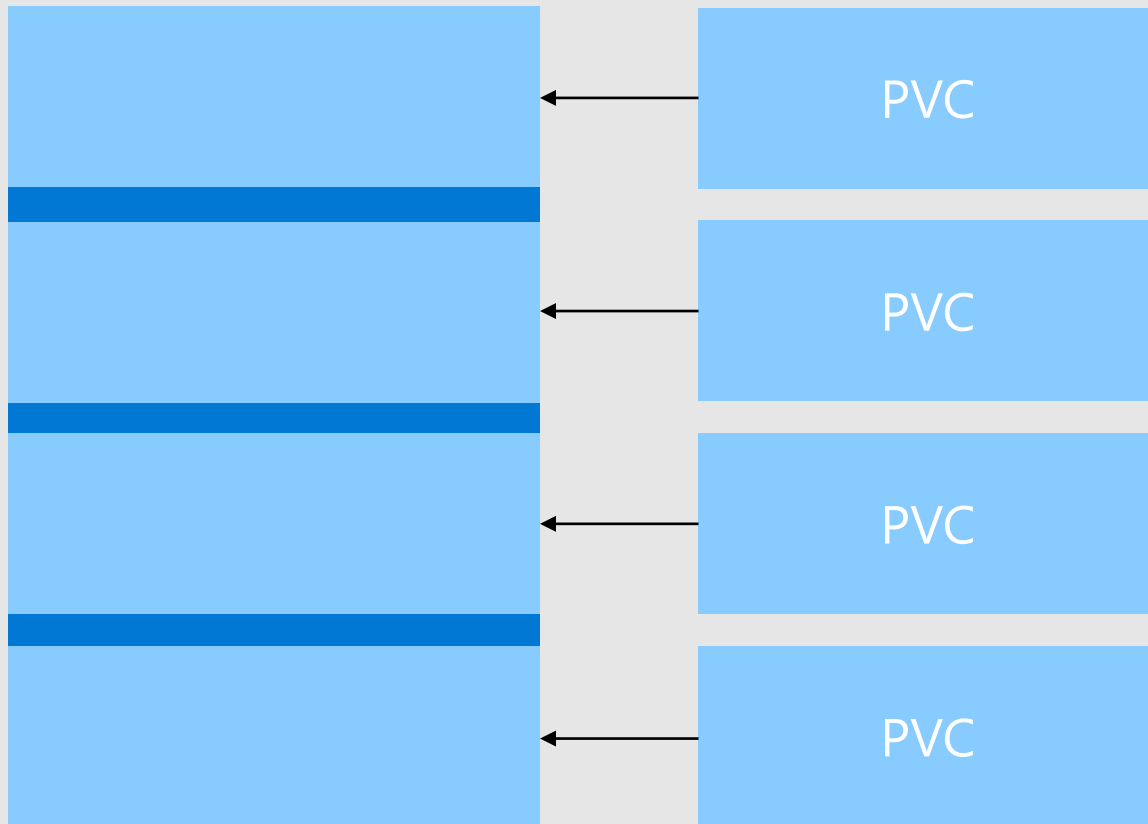
FRANKFURT



DEVELOPERS.DE

Storage

Persistent Volume



Storage Class

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard
provisioner: kubernetes.io/aws-ebs
parameters:
  type: gp2
reclaimPolicy: Retain | Remove
allowVolumeExpansion: true
mountOptions:
  - debug
volumeBindingMode: Immediate
```

Provisioners

Volume Plugin	Internal Provisioner	Config Example
AWSElasticBlockStore	✓	AWS EBS
AzureFile	✓	Azure File
AzureDisk	✓	Azure Disk
CephFS	-	-
Cinder	✓	OpenStack Cinder
FC	-	-
FlexVolume	-	-
Flocker	✓	-
GCEPersistentDisk	✓	GCE PD
Glusterfs	✓	Glusterfs
iSCSI	-	-
Quobyte	✓	Quobyte

Volume Plugin	Internal Provisioner	Config Example
NFS	-	-
RBD	✓	Ceph RBD
VsphereVolume	✓	vSphere
PortworxVolume	✓	Portworx Volume
ScaleIO	✓	ScaleIO
StorageOS	✓	StorageOS
Local	-	Local

Re-visit Persistent Volume

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0003
spec:
  capacity:
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: slow
  mountOptions:
    - hard
    - nfsvers=4.1
  nfs:
    path: /tmp
    server: 172.17.0.2
```

Cluster-wide managed by Administrator

Declare:

- Capacity
- VolumeMode
 - Filesystem
 - Block
- AccessModes:
 - ReadWriteOnce
 - ReadOnlyMany
 - ReadWriteMany

ConfigMaps & Secrets

ConfigMap

```
kubectl create configmap <name>
```

Sources:

```
--from-file=/path/to/file
```

```
--from-env-file=/path
```

```
--from-literal=my.config.value=foobar
```

Create Secret

```
kubectl create secret type name sources
```

Secret Types:

- docker-registry
- Generic (file, directory, literal)
- tls

RBAC

ClusterRole

```
kubectl create clusterrole <name>
```

```
--verb=get,list,watch
```

```
--resource=pods
```

```
--resource-name=podname
```

RoleBinding

```
kubectl create rolebinding <binding-name>
```

```
--clusterrole=admin  
--user=niko  
--group=developers  
--namespace=default  
--serviceaccount=ns:name
```

ClusterRoleBinding

```
kubectl create clusterrolebinding <cluster-rb-name>
```

```
--clusterrole=cluster-admin
```

```
--user=root
```



HELM

Kubernetes Applications

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: nginx-deployment
```

```
  labels:
```

```
    app: nginx
```

```
spec:
```

```
  replicas: 3
```

```
  selector:
```

```
    matchLabels:
```

```
      app: nginx
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: nginx
```

```
  spec:
```

```
    containers:
```

```
      - name: nginx
```

```
        image: nginx:1.14
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
  name: my-service
```

```
spec:
```

```
apiVersion: v1
```

```
kind: PersistentVolume
```

```
metadata:
```

```
  name: foo-pv
```

```
spec:
```

```
  storageClassName: ""
```

```
  claimRef:
```

```
    name: foo-pvc
```

```
  name:
```

```
  ...
```

```
apiVersion: v1
```

```
kind: Secret
```

```
metadata:
```

```
  name: secret-sa-sample
```

```
  annotations:
```

```
    kubernetes.io/service-account.name: "sa-name"
```

```
  type: kubernetes.io/service-account-token
```

```
apiVersion: networking.k8s.io/v1
```

```
kind: Ingress
```

```
metadata:
```

```
  name: minimal-ingress
```

```
  annotations:
```

```
    nginx
```

```
apiVersion: v1
```

```
kind: ConfigMap
```

```
metadata:
```

```
  name: game-demo
```

```
data:
```

```
  # property-like keys; each key maps to a simple
```

```
  player_initial_lives: "3"
```

```
  ui_properties_file_name: "user-interface.properties"
```

```
  # file-like keys
```

```
  ns,monsters
```

```
  lives=5
```

```
  erties: |
```

```
  e
```

```
  rue
```

Helm Concept

Chart

A Helm package which contains all of the resource definitions necessary to run an application in Kubernetes

Repository

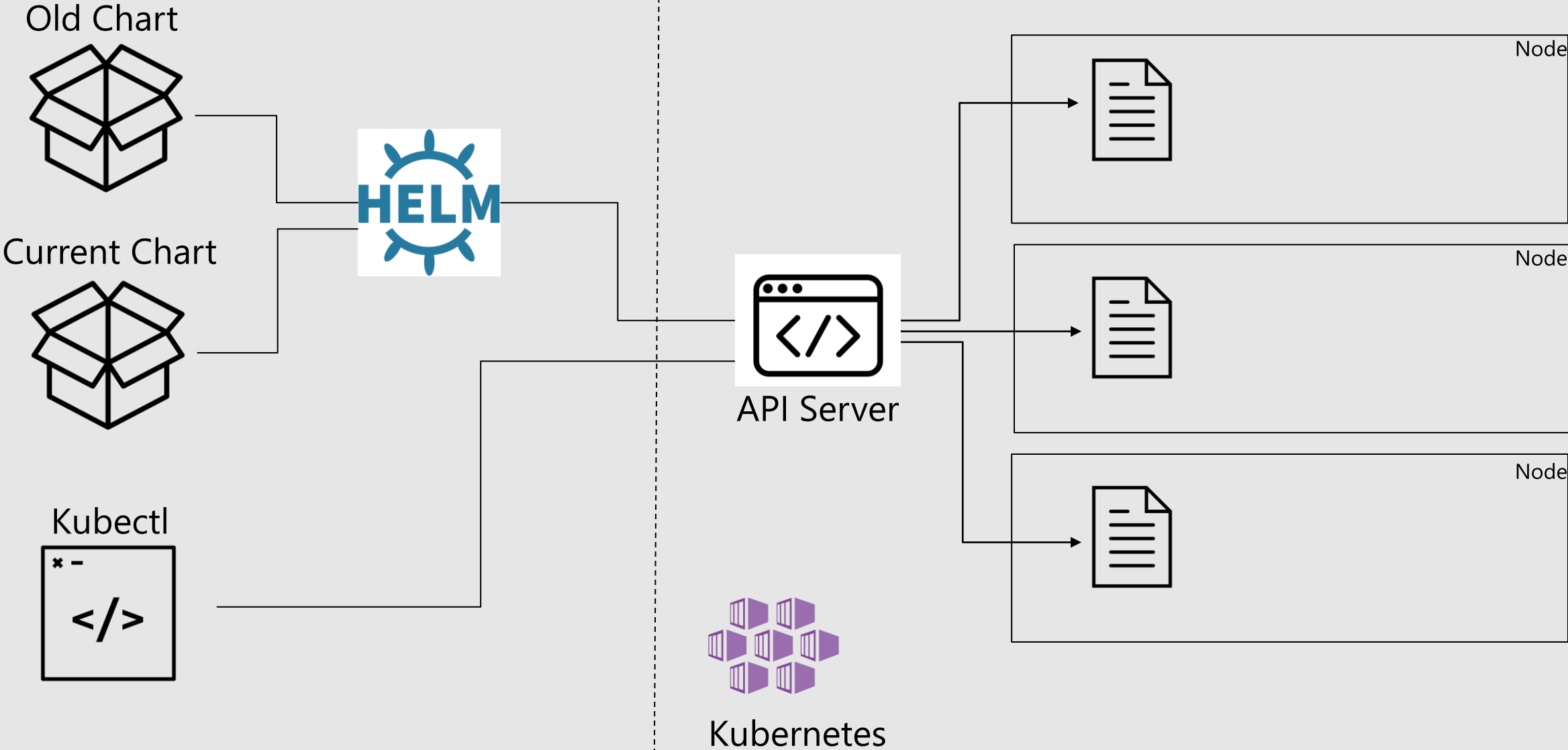
Place where charts can be collected and shared. It's like NPM, but for Kubernetes

Release

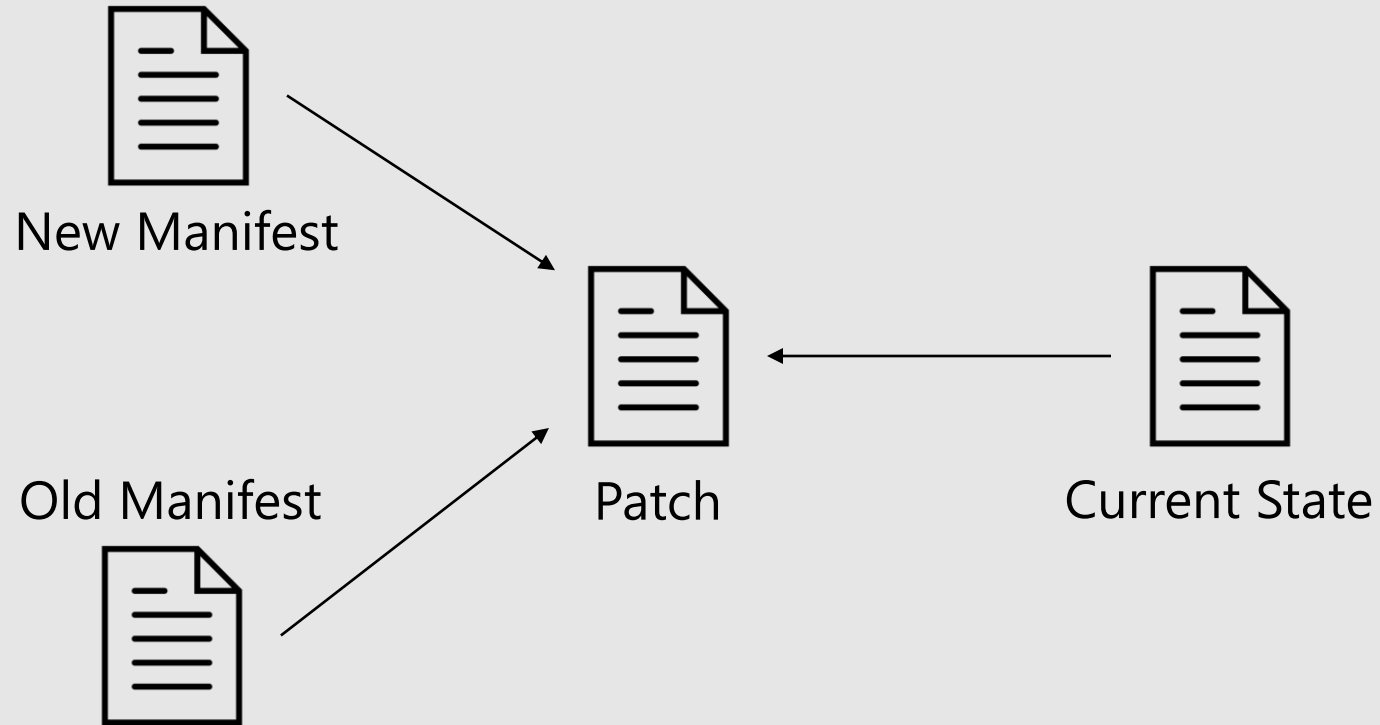
Instance of a chart running in a Kubernetes cluster.

Helm installs ***charts*** into Kubernetes, creating a new ***release*** for each installation. You can store/get charts from Helm **chart repositories**.

Helm 3



3-way Strategic Merge Patch



Working with Repositories

```
helm search [hub|repo] {chartname}
```

```
helm search repo {name} {chartname}
```

```
helm repo add {name} {uri}
```

```
helm repo update {flags}
```

Working with Packages

```
helm install {release_name} {helm chart}
```

```
helm upgrade {release_name} {helm chart}
```

```
helm status {release_name}
```

```
Helm history {release_name}
```

```
helm rollback {release_name} {revision:1}
```

```
helm uninstall {release_name}
```

Customize Charts

```
helm show values stable/elastic-stack
```

Displays values.yaml of Chart

--set CLI Override

--values (-f) config.yaml override

```
helm install -f config.yaml stable/elastic-stack
```

Install Package - Options

```
helm install <release-name> <chart_origin>
```

```
helm upgrade <release-name> <chart_origin> --install
```

```
helm upgrade -f override.yaml {release_name} {helm  
chart}
```


Creating Charts

Create new Chart

```
helm create {chart_name}
```

```
chart_name
```

```
    Chart.yaml
```

```
    templates/
```

```
        deployment.yaml
```

```
        _helpers.tpl
```

```
        resources.yaml
```

```
    .helmignore
```

```
    values.yaml
```

Chart Documentation

chart_name

Chart.yaml

LICENSE

README.md

templates

NOTES.txt

values.schema.json

values.yaml

.helmignore



Chart Setup

DEMO

Q &A



Thank You!

ευχαριστώ Salamat Po متشكراً شكراً Grazie

благодаря ありがとうございます Kiitos Teşekkürler 谢谢

ໂພນດຸຣຸນດຳ Obrigado شكریه Terima Kasih Dziękuję

Hvala Köszönöm Tak Dank u Wel ДЯКУЮ Tack

Mulțumesc спасибо Danke Cám ơn Gracias

多謝晒 Ďakujem תודה நன்றி Děkuji 감사합니다





