

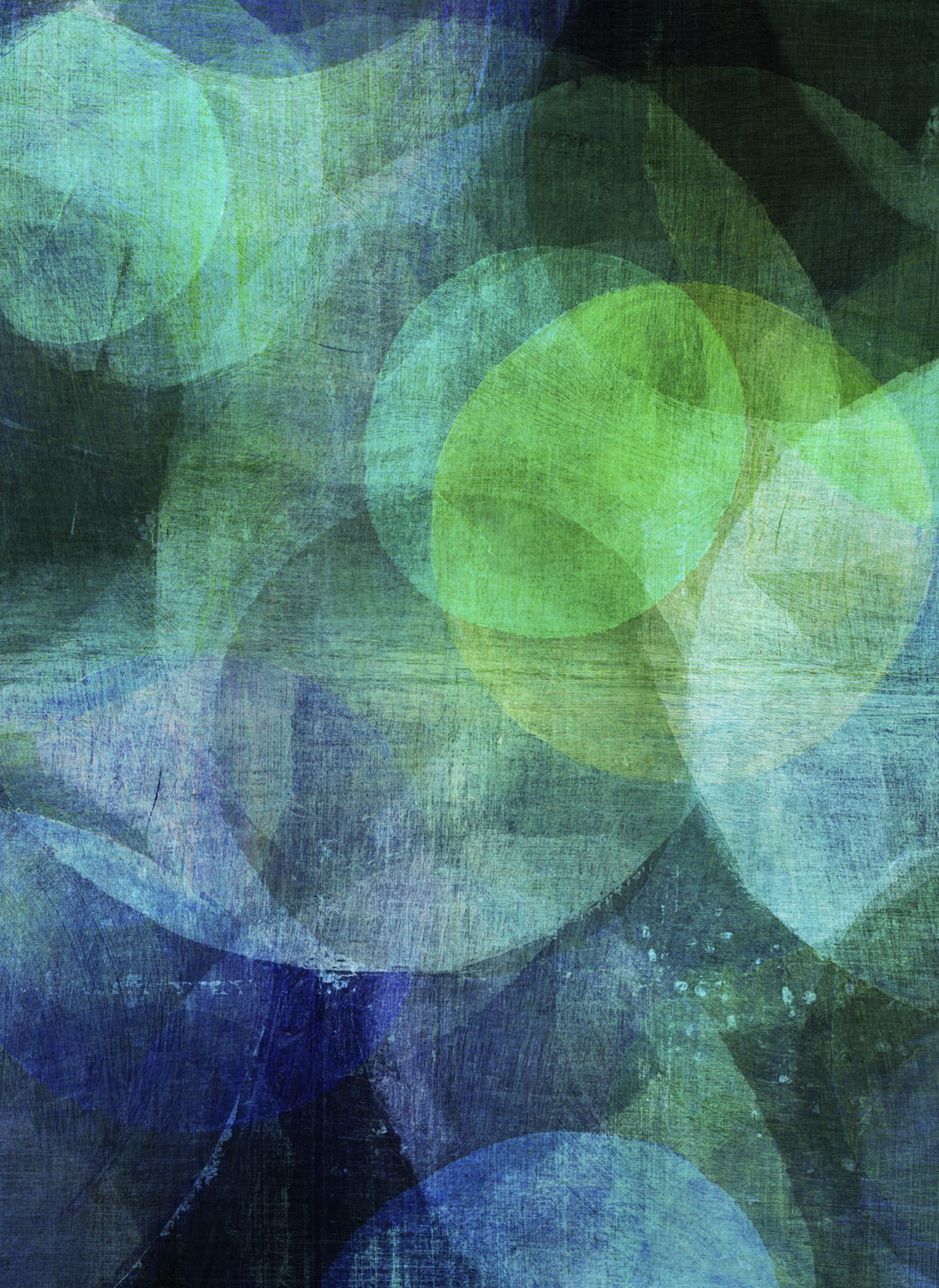
INTRODUCTION TO KUBERNETES



INTRODUCTION TO KUBERNETES

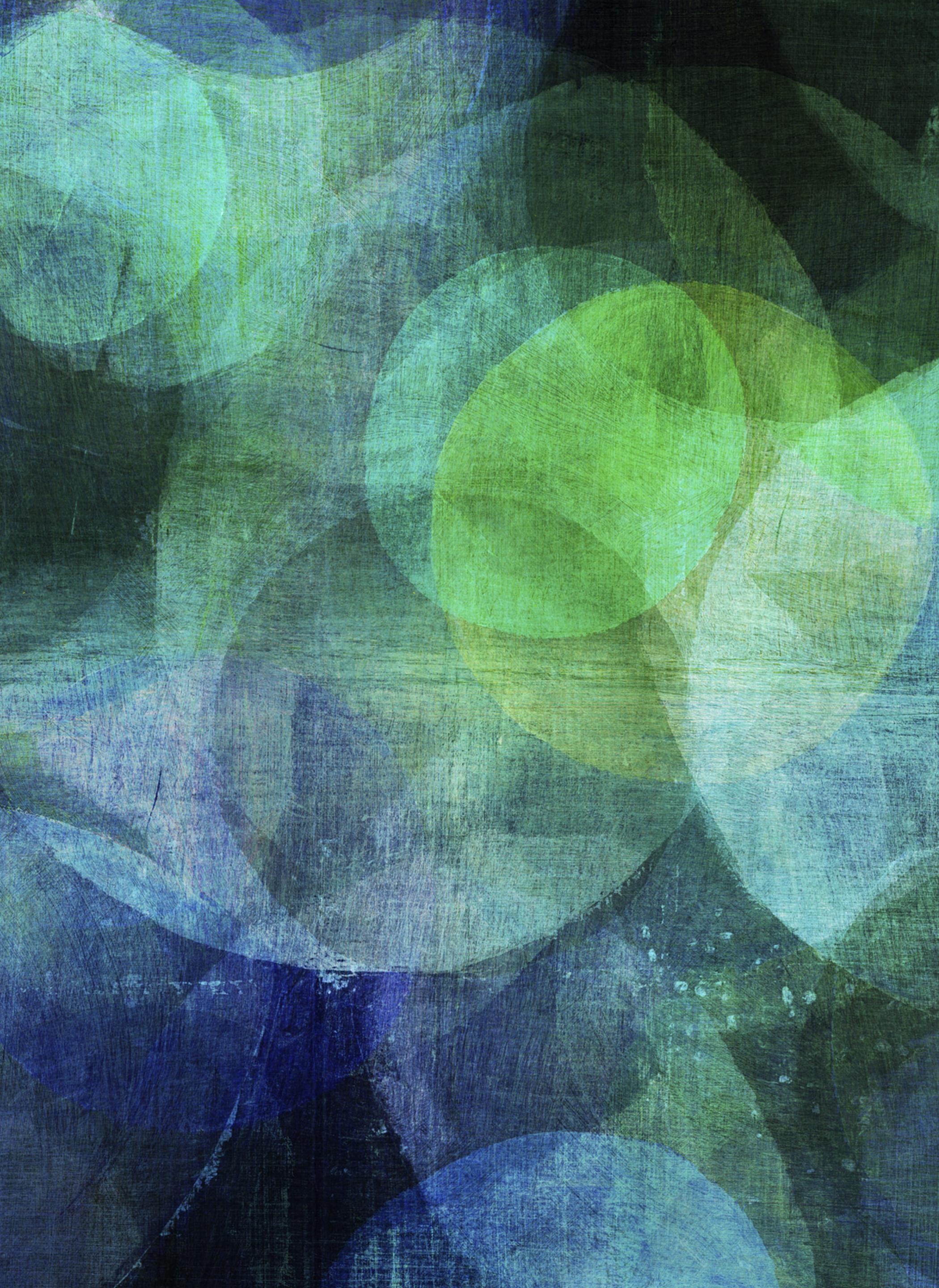
Continuous Deployment and Helm

AGENDA



AGENDA

- K8s Deployment Pipeline



AGENDA

- K8s Deployment Pipeline
 - Helm



AGENDA

- K8s Deployment Pipeline
 - Helm
 - Helm and Helm Chart

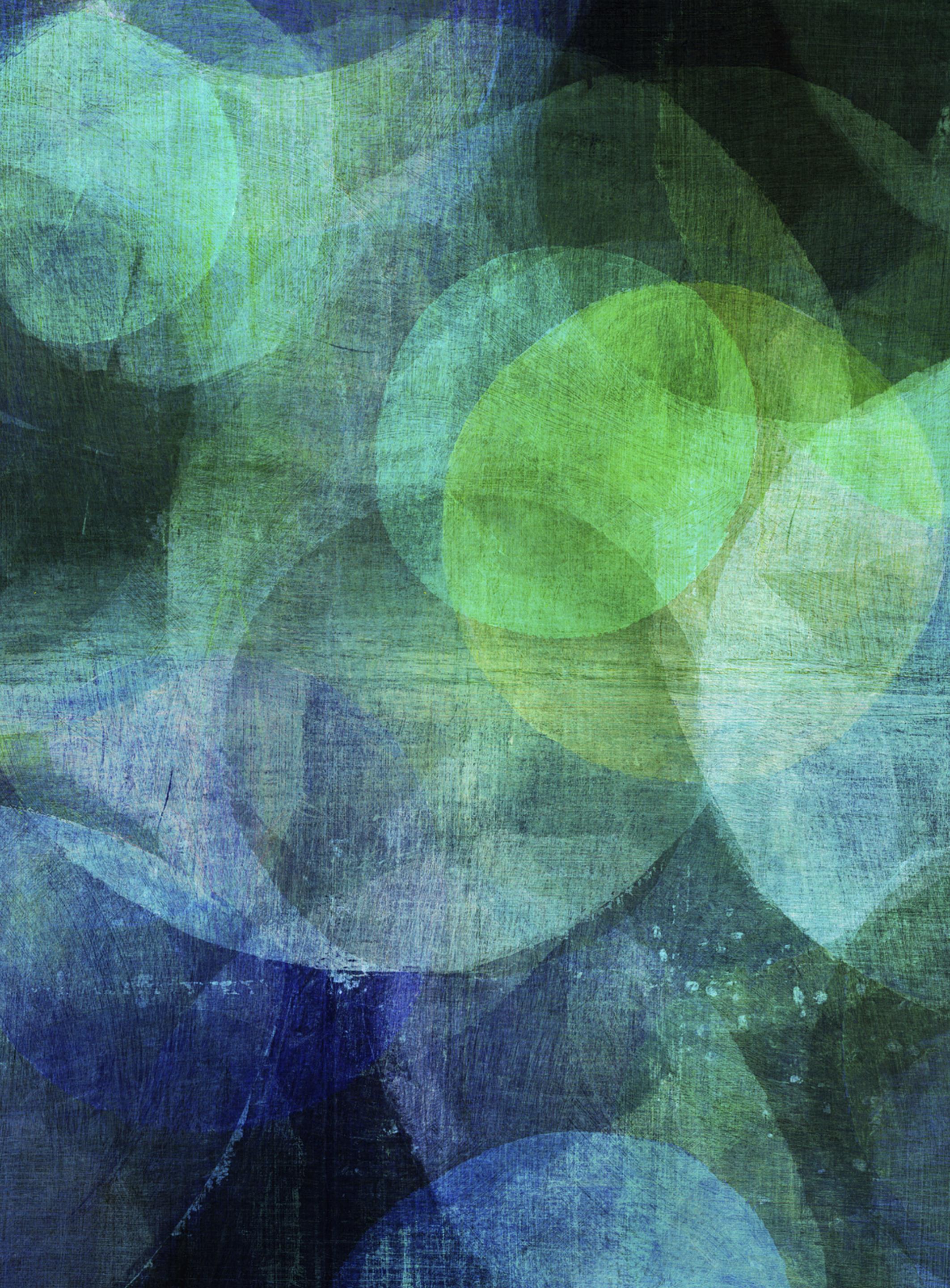


AGENDA

- K8s Deployment Pipeline

- Helm

- Helm and Helm Chart
 - Package Management

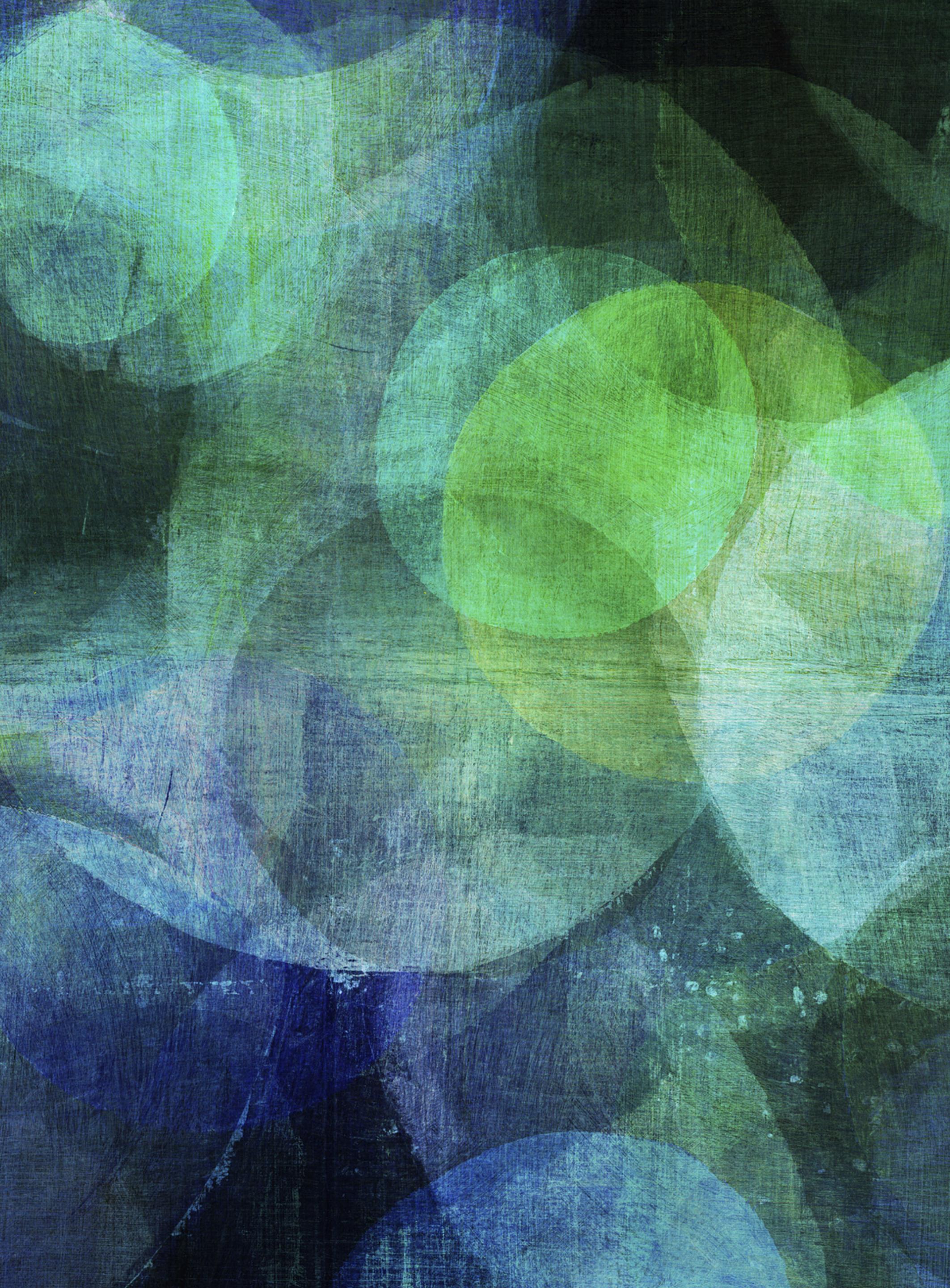


AGENDA

- K8s Deployment Pipeline

- Helm

- Helm and Helm Chart
 - Package Management
 - Template Engine

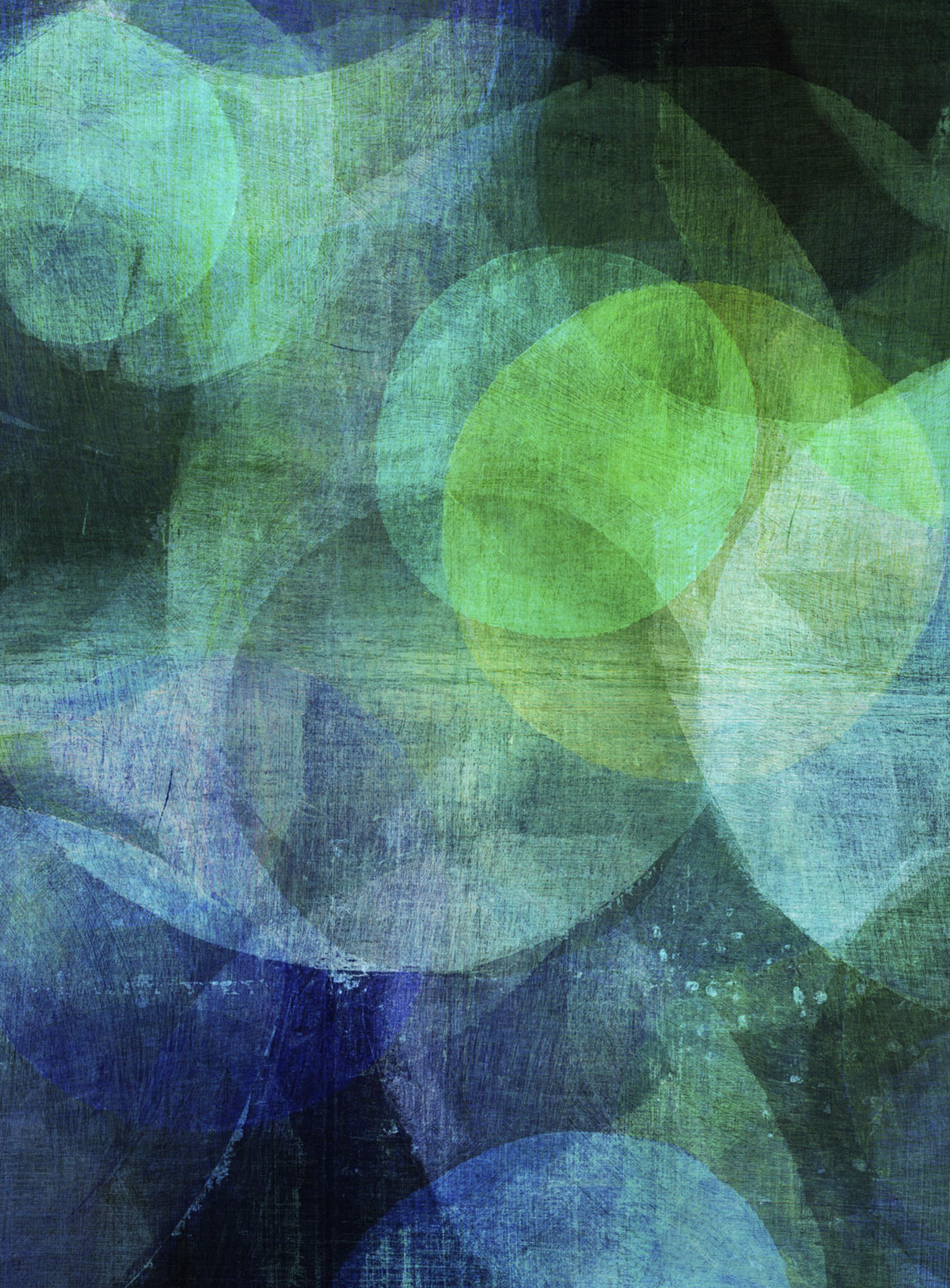


AGENDA

- K8s Deployment Pipeline

- Helm

- Helm and Helm Chart
 - Package Management
 - Template Engine
 - Release Management



K8S DEPLOYMENT PIPELINE

DEPLOYMENT PIPELINE

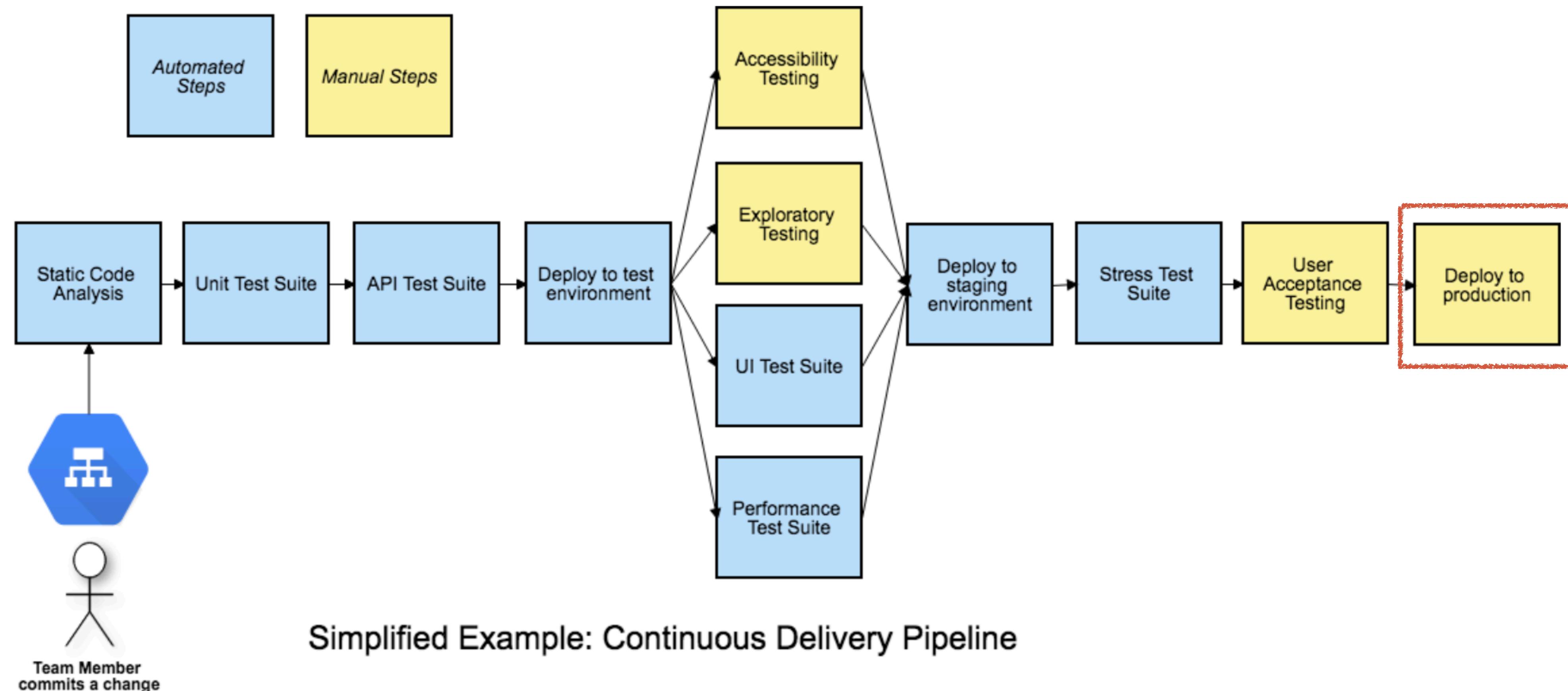
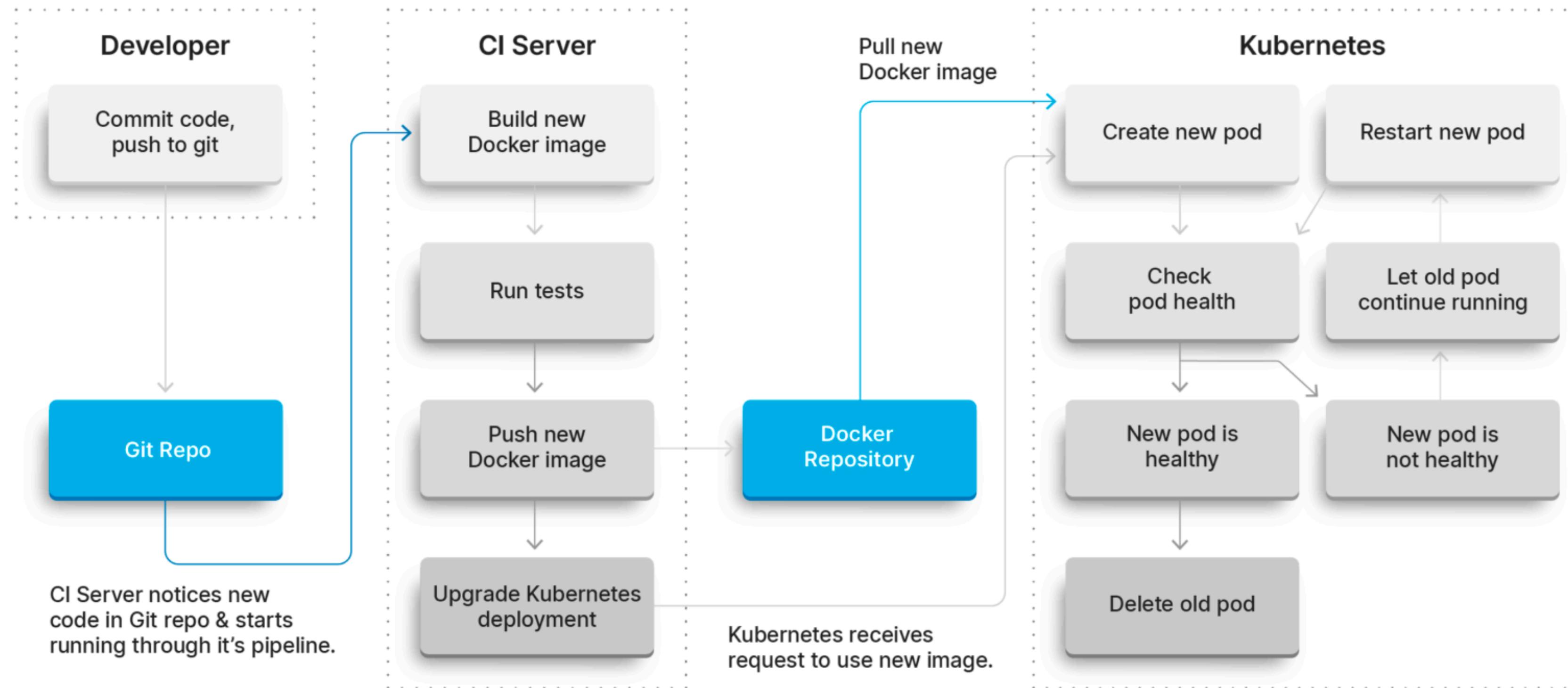


Figure 1: A Continuous Delivery Pipeline (my diagrams are inspired by Katrina Clokie's pipeline diagrams in "A Practical Guide to Testing in DevOps")

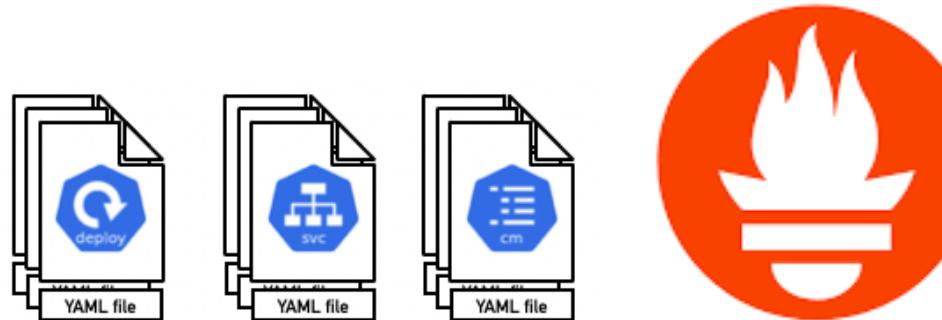
DEPLOYMENT PIPELINE WITH K8S



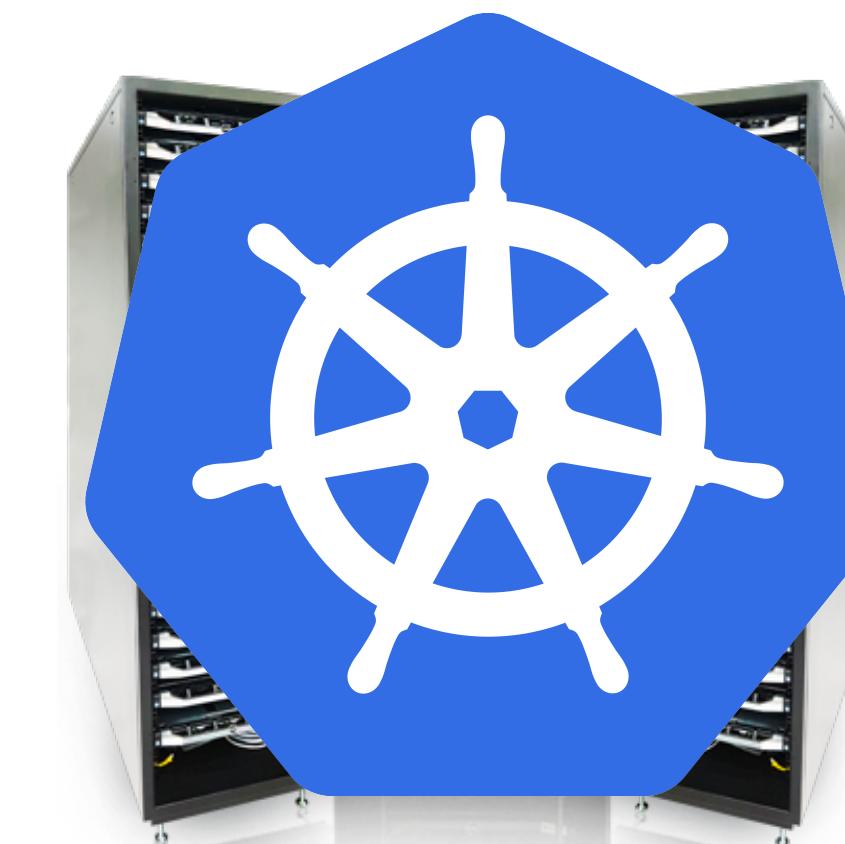
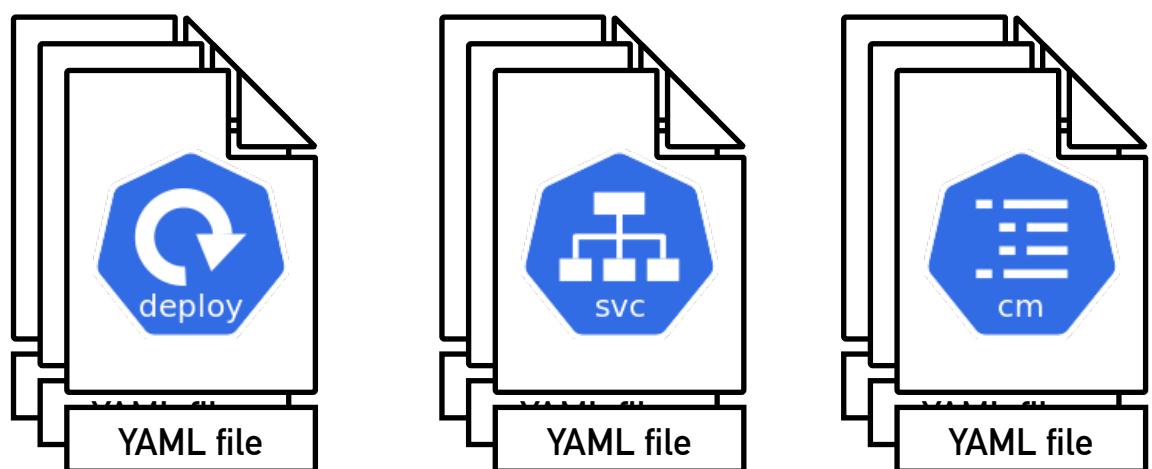
The goal of Continuous Delivery (CD) is to get your changes into production in a safe manner. Kubernetes has the ability to deploy very quickly, especially if using a recreate strategy where all the Pods are killed and replaced vs incrementally with a rolling strategy; but this causes downtime.

PACKAGE MANAGE FOR K8S

WHAT'S HELM - PACKAGE MANAGER FOR K8S



Prometheus



kubernetes



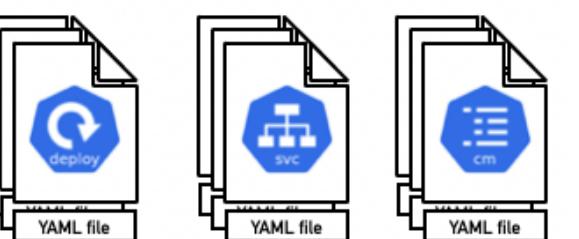
elasticsearch



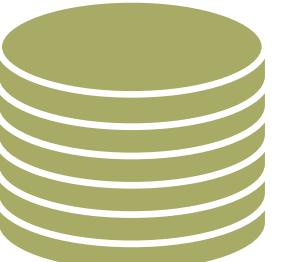
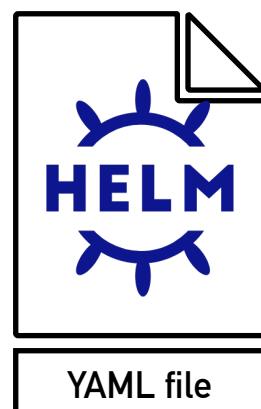
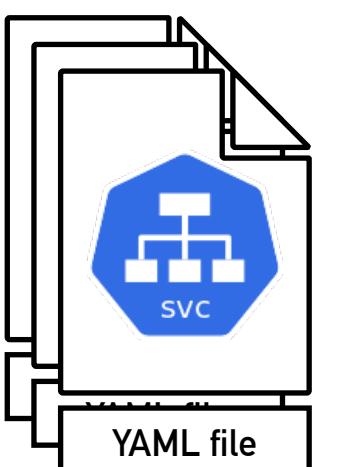
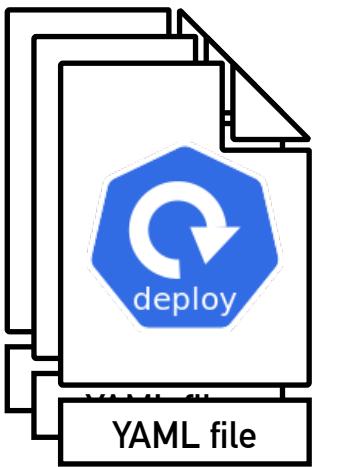
logstash



kibana

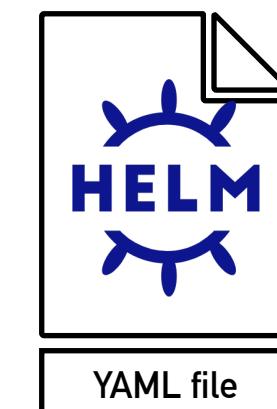
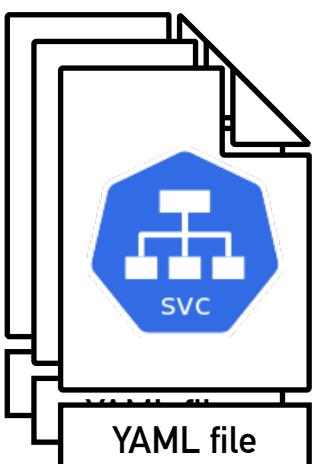
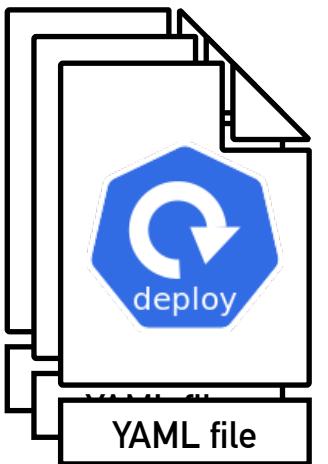


HELM CHARTS



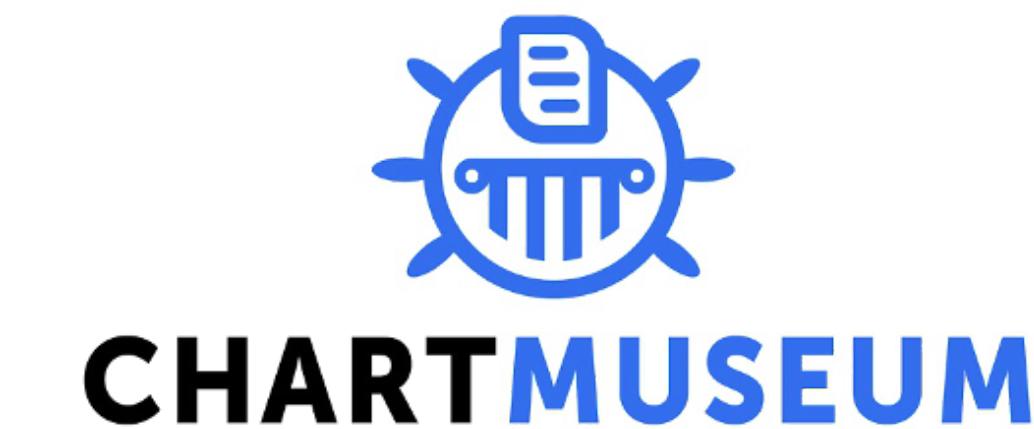
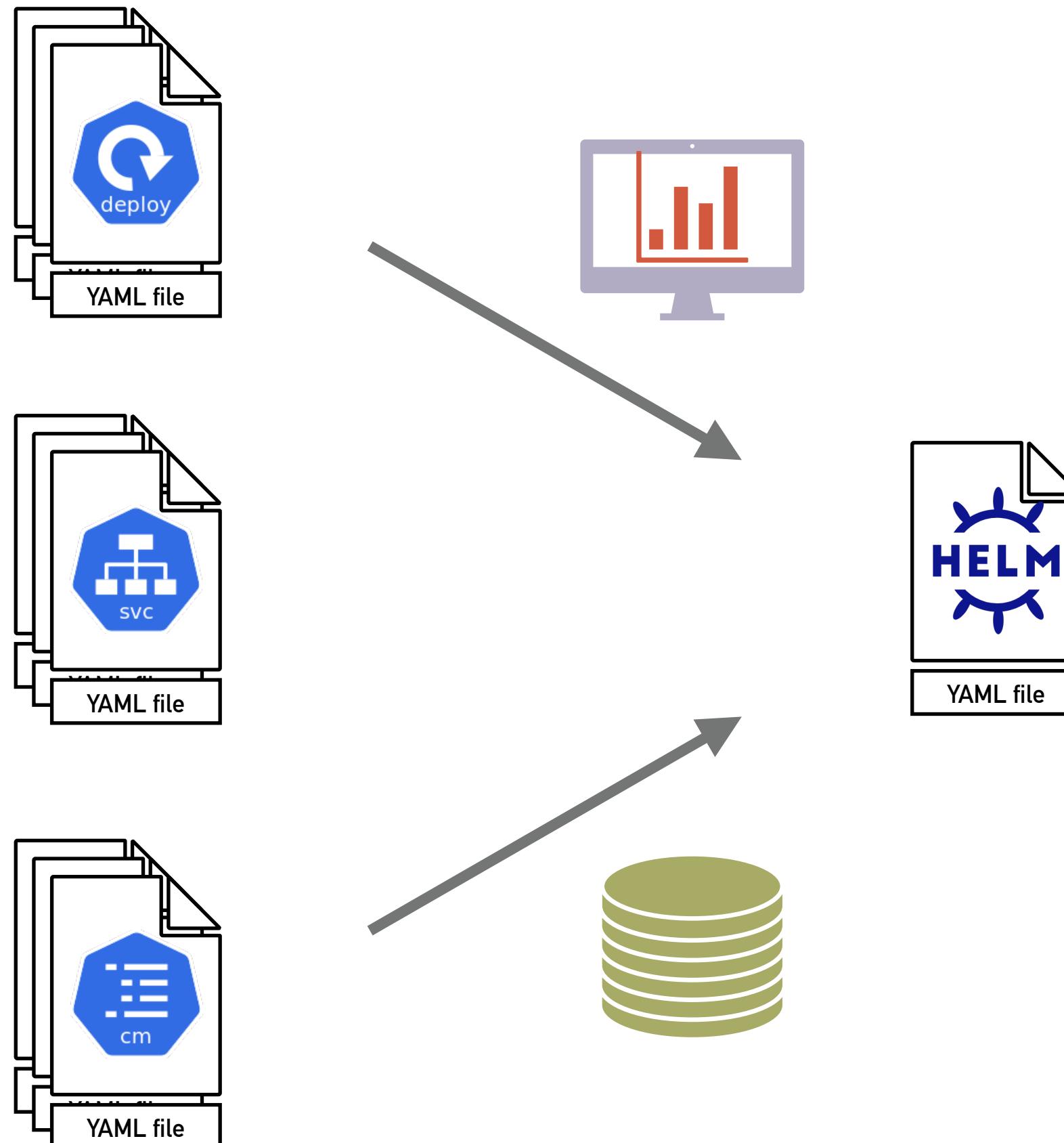
CHARTMUSEUM

HELM CHARTS

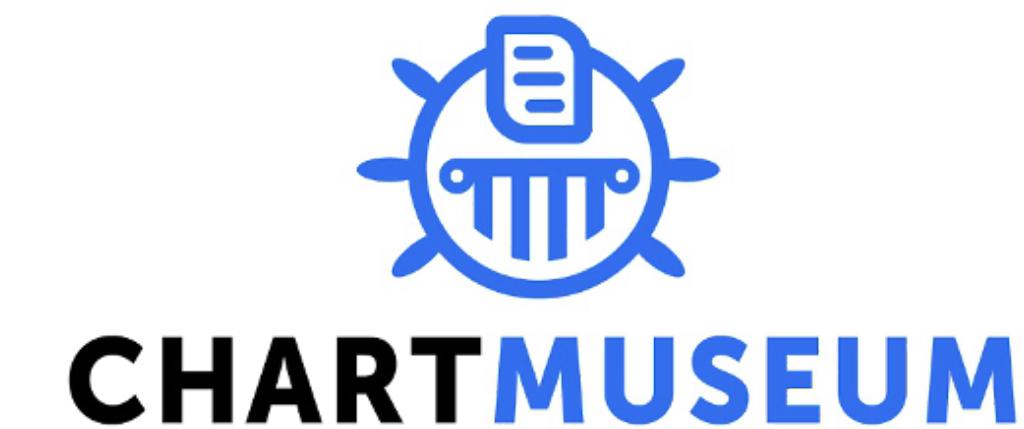
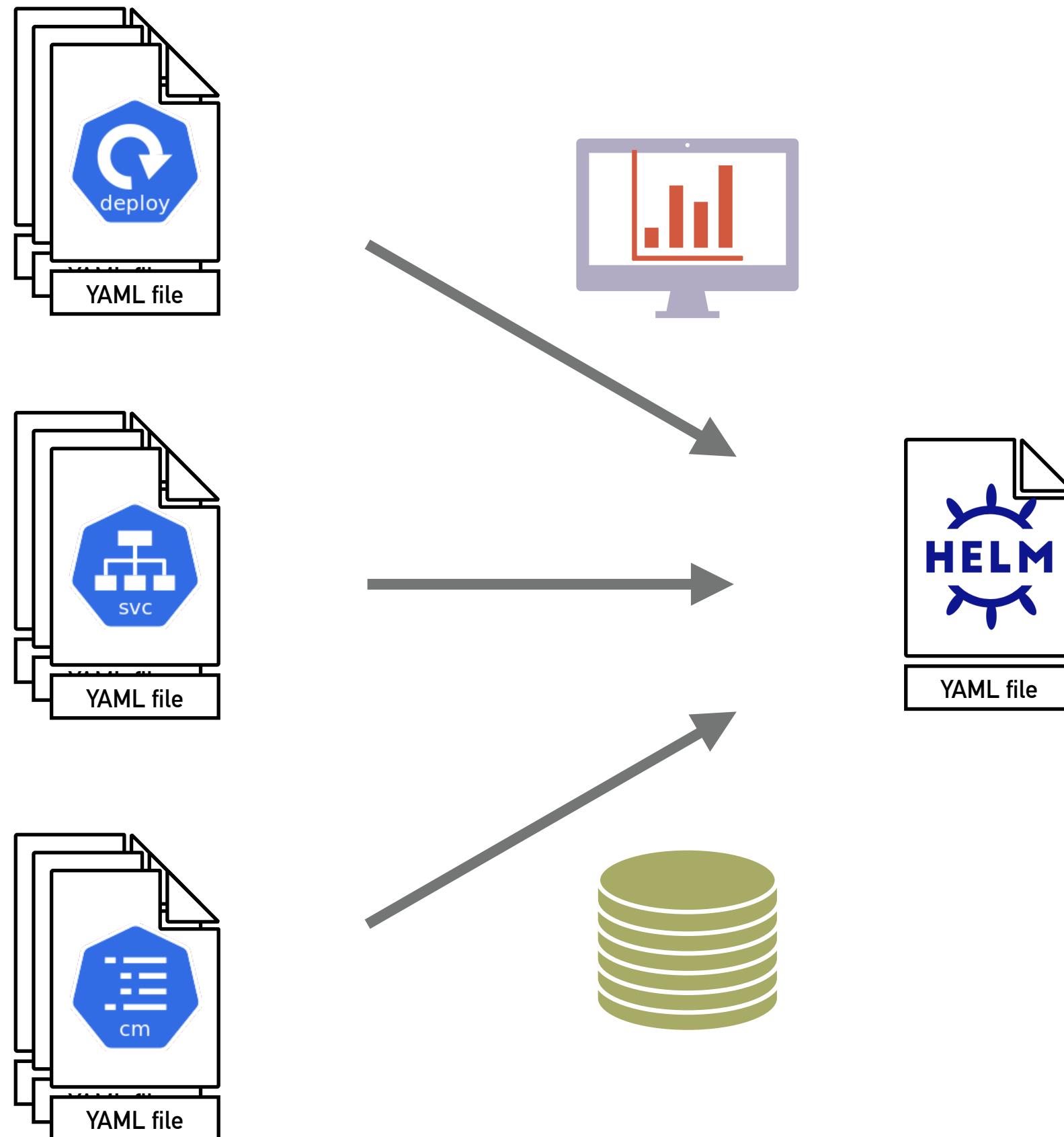


CHARTMUSEUM

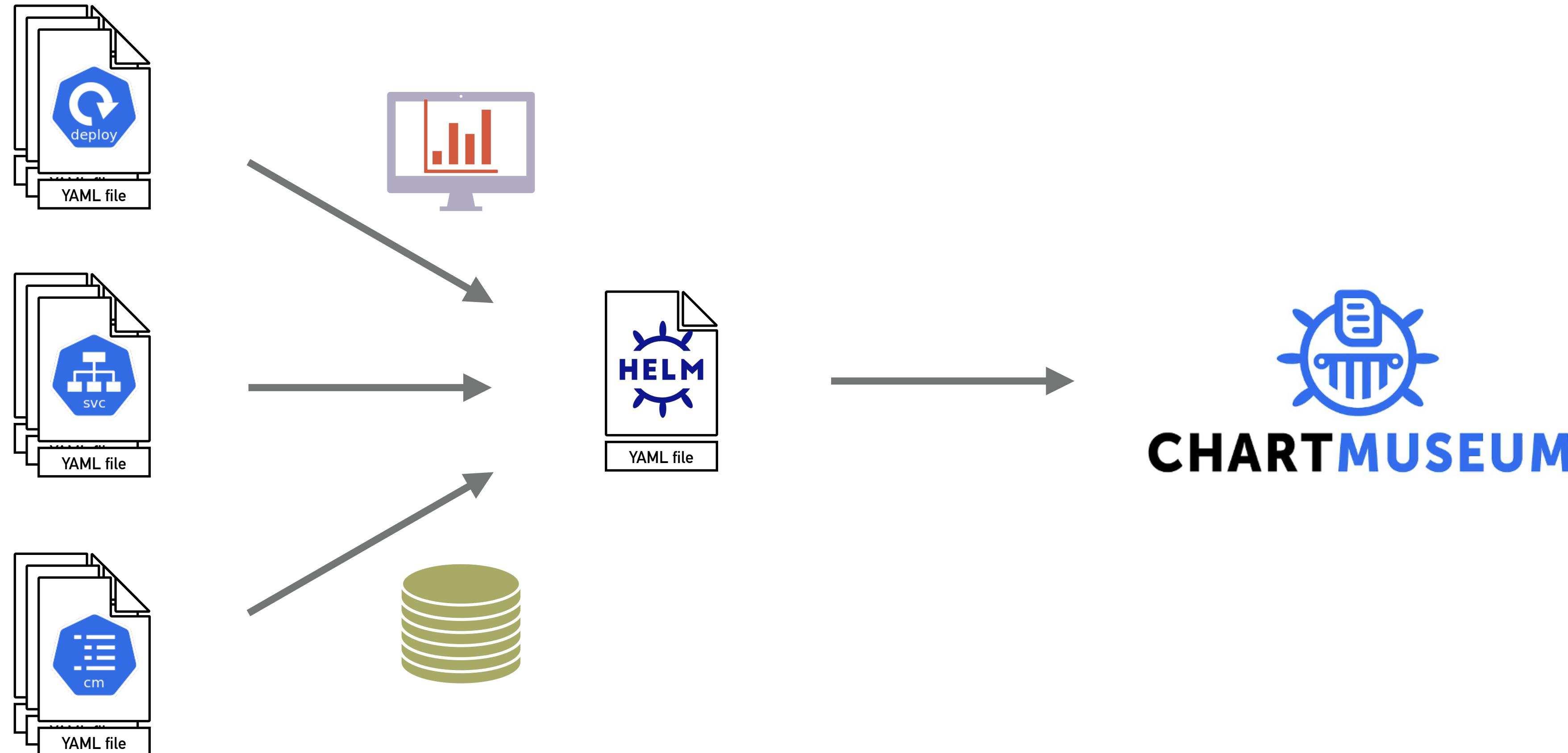
HELM CHARTS



HELM CHARTS



HELM CHARTS



HELM HUB (ARTIFACTHUB.IO)

ArtifactHUB

STATS SIGN UP SIGN IN

Find, install and publish Kubernetes packages

Mysql Cluster

- mysql-cluster**
USER: kfirfer REPO: kfirfer 2
- mysql-cluster**
ORG: Bitpoke REPO: bitpoke 0
- mysql-cluster**
ORG: Helm REPO: presslabs 0
- mysql-innodbcluster**
ORG: MySQL REPO: mysql-operator 1
- ndb-operator**
ORG: MySQL REPO: mysql-ndb-operator 0

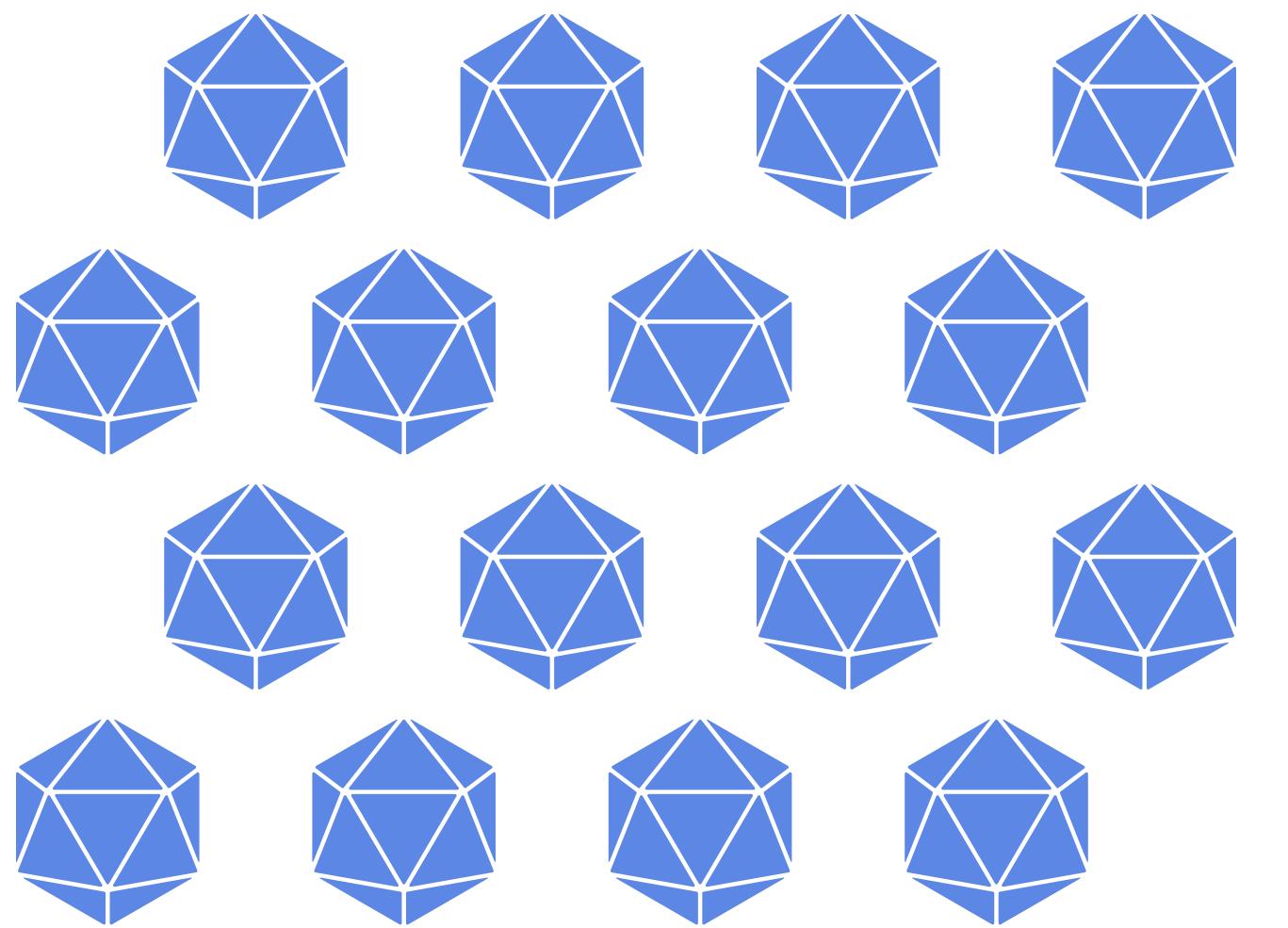
See all results (27)

Please see the [repositories guide](#) for more information about how to list your content on Artifact Hub.

TEMPLATE ENGINE

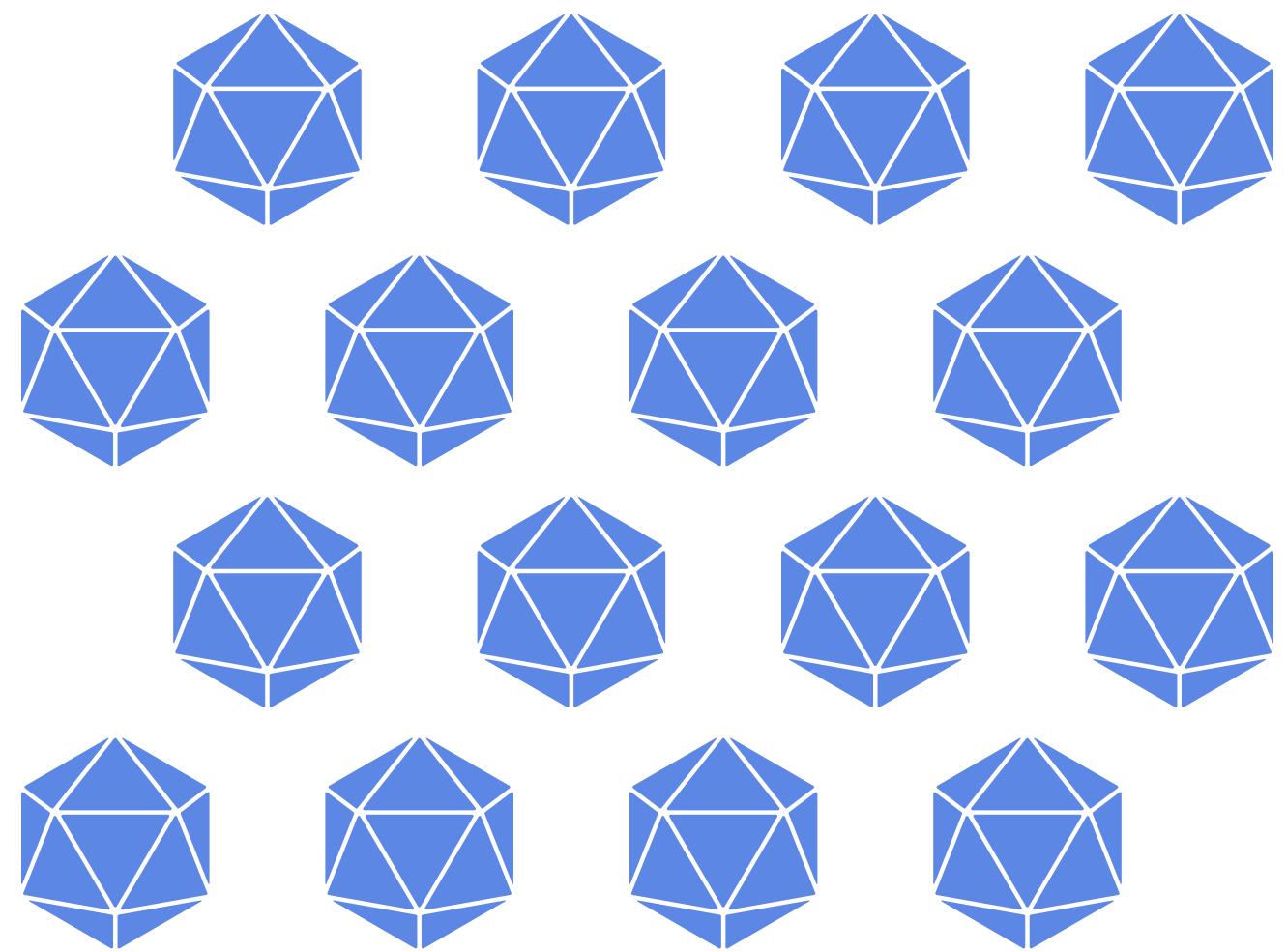
RISING OF MICROSERVICES

RISING OF MICROSERVICES



Microservices

RISING OF MICROSERVICES

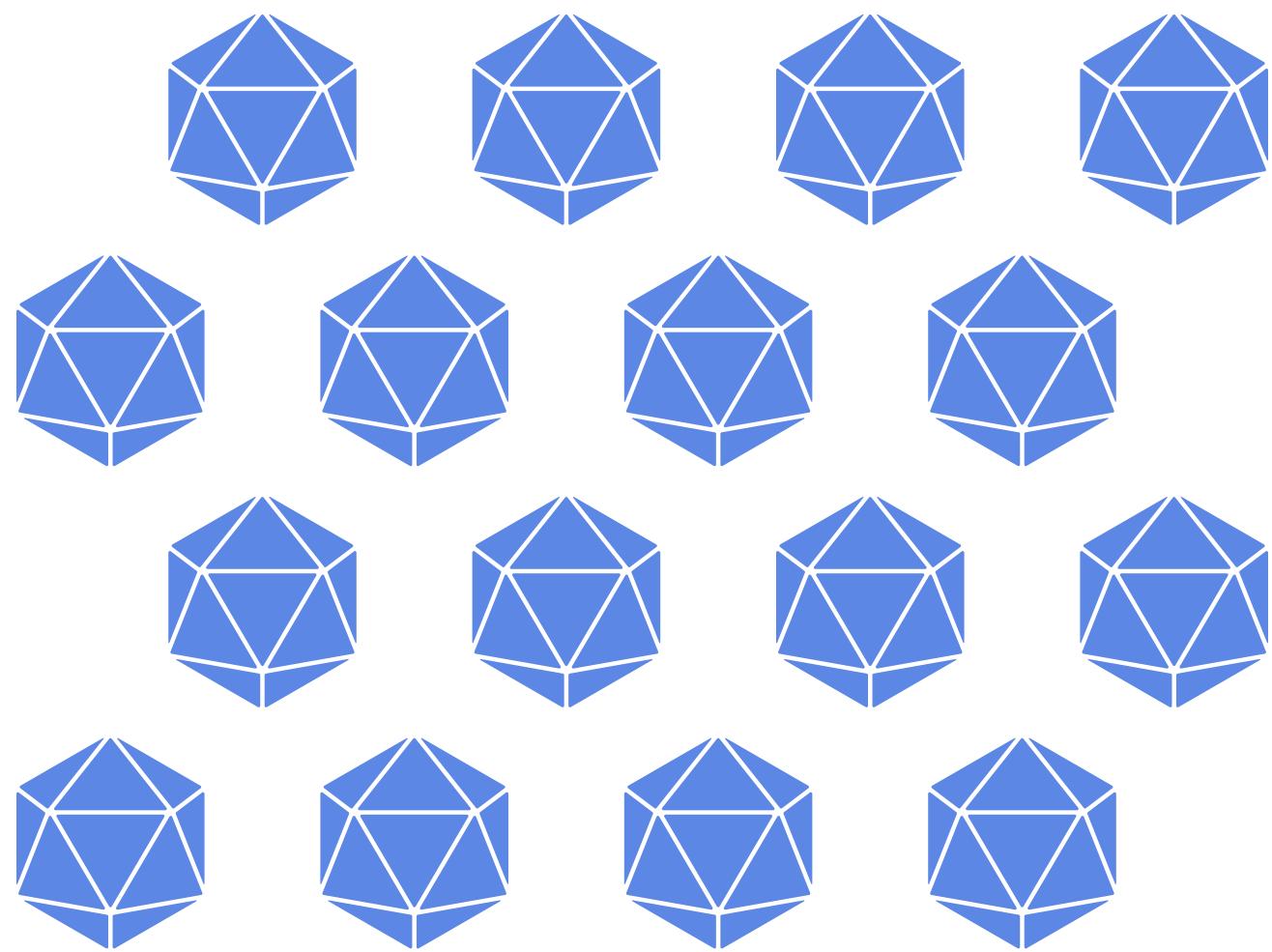


Microservices

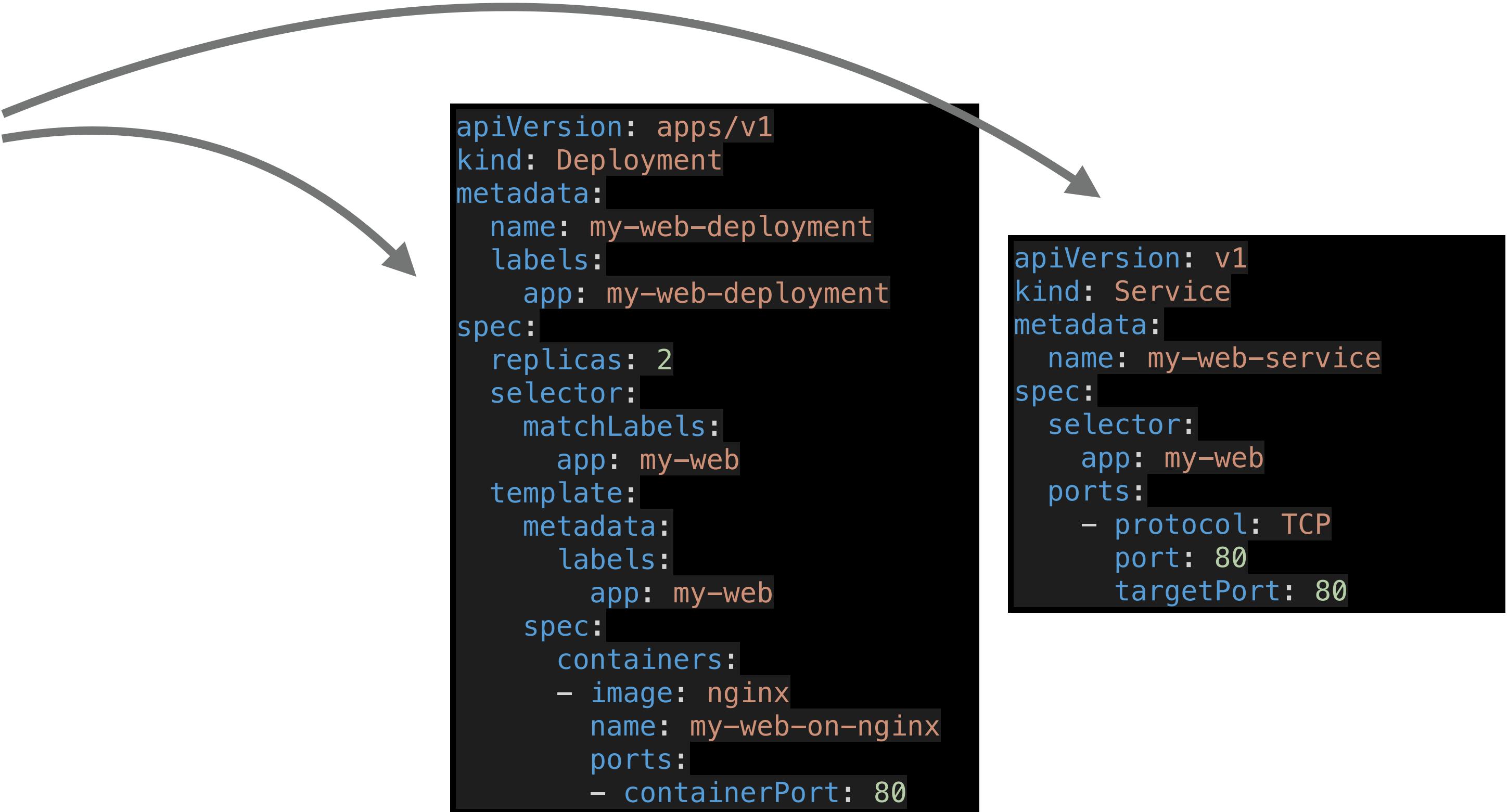
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
        ports:
          - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

RISING OF MICROSERVICES



Microservices



YAML (DEPLOYMENT AND SERVICE) FILE ALMOST THE SAME

YAML (DEPLOYMENT AND SERVICE) FILE ALMOST THE SAME

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
  spec:
    containers:
      - image: nginx
        name: my-web-on-nginx
        ports:
          - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

YAML (DEPLOYMENT AND SERVICE) FILE ALMOST THE SAME

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
  spec:
    containers:
      - image: nginx
        name: my-web-on-nginx
        ports:
          - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
  spec:
    containers:
      - image: nginx
        name: my-web-on-nginx
        ports:
          - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

YAML (DEPLOYMENT AND SERVICE) FILE ALMOST THE SAME

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

YAML (DEPLOYMENT AND SERVICE) FILE ALMOST THE SAME

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
  spec:
    containers:
      - image: nginx
        name: my-web-on-nginx
        ports:
          - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
  spec:
    containers:
      - image: nginx
        name: my-web-on-nginx
        ports:
          - containerPort: 80
```

```
sion: v1
Service
iVersion: v1
nd: Service
taapiVersion: v1
nækind: Service
edmetadata:
se  name: my-web-service
spec:
pc  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

YAML (DEPLOYMENT AND SERVICE) FILE ALMOST THE SAME

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

YAML (DEPLOYMENT AND SERVICE) FILE ALMOST THE SAME

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web-deployment
  template:
    metadata:
      labels:
        app: my-web-deployment
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

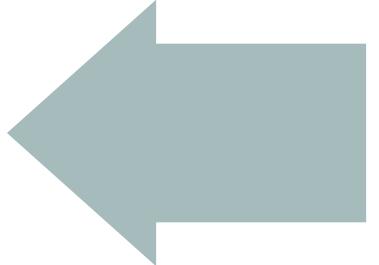
TEMPLATE ENGINE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

TEMPLATE ENGINE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.comtainer.image}}
          name: {{.Value.comtainer.name}}
        ports:
          - containerPort: {{.Value.port}}
```

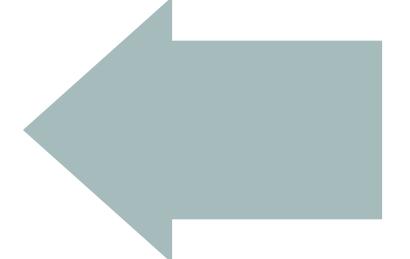


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
        ports:
          - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

TEMPLATE ENGINE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.comtainer.image}}
          name: {{.Value.comtainer.name}}
          ports:
            - containerPort: {{.Value.port}}
```



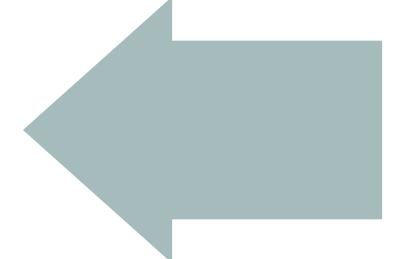
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

Common Blueprint

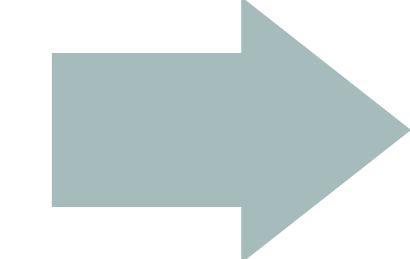
TEMPLATE ENGINE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.comtainer.image}}
          name: {{.Value.comtainer.name}}
          ports:
            - containerPort: {{.Value.port}}
```



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

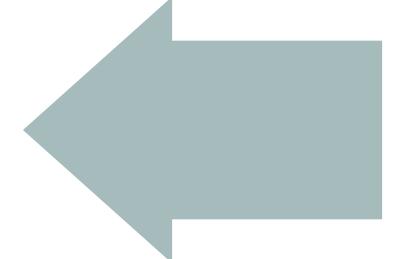


```
name: my-app
containers:
  image: my-api:1.3.0
  name: my-api
  ports: 9000
```

Common Blueprint

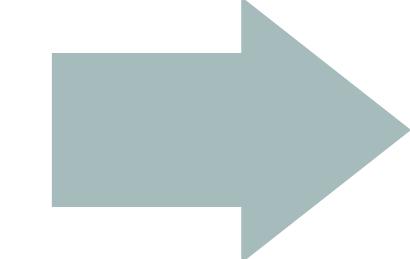
TEMPLATE ENGINE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.comtainer.image}}
          name: {{.Value.comtainer.name}}
          ports:
            - containerPort: {{.Value.port}}
```



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

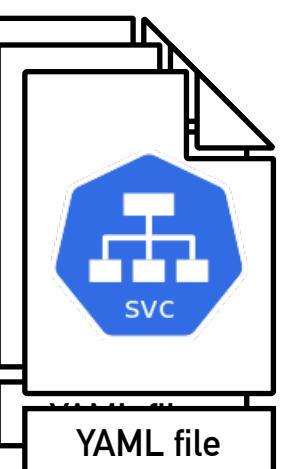
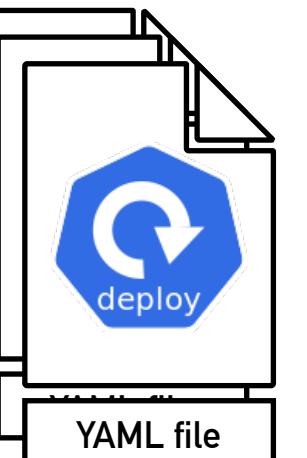


Dynamic Value

Common Blueprint

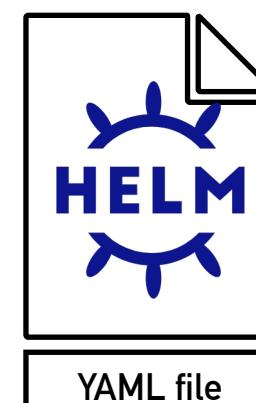
TEMPLATE ENGINE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```



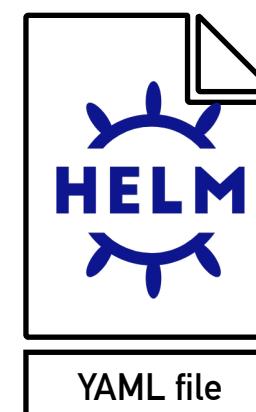
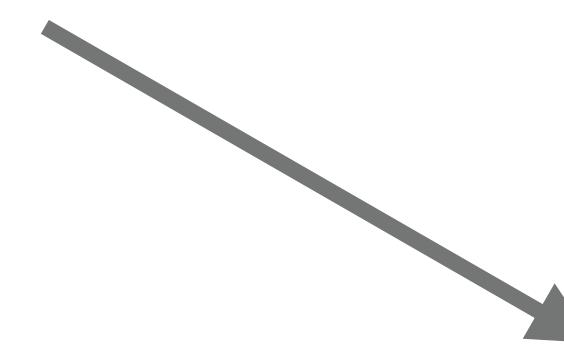
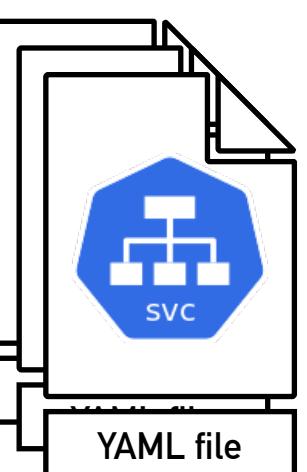
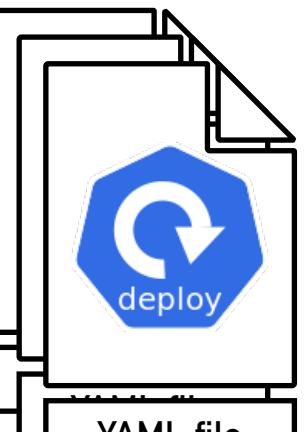
```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.comtainer.image}}
          name: {{.Value.comtainer.name}}
          ports:
            - containerPort: {{.Value.port}}
```



TEMPLATE ENGINE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

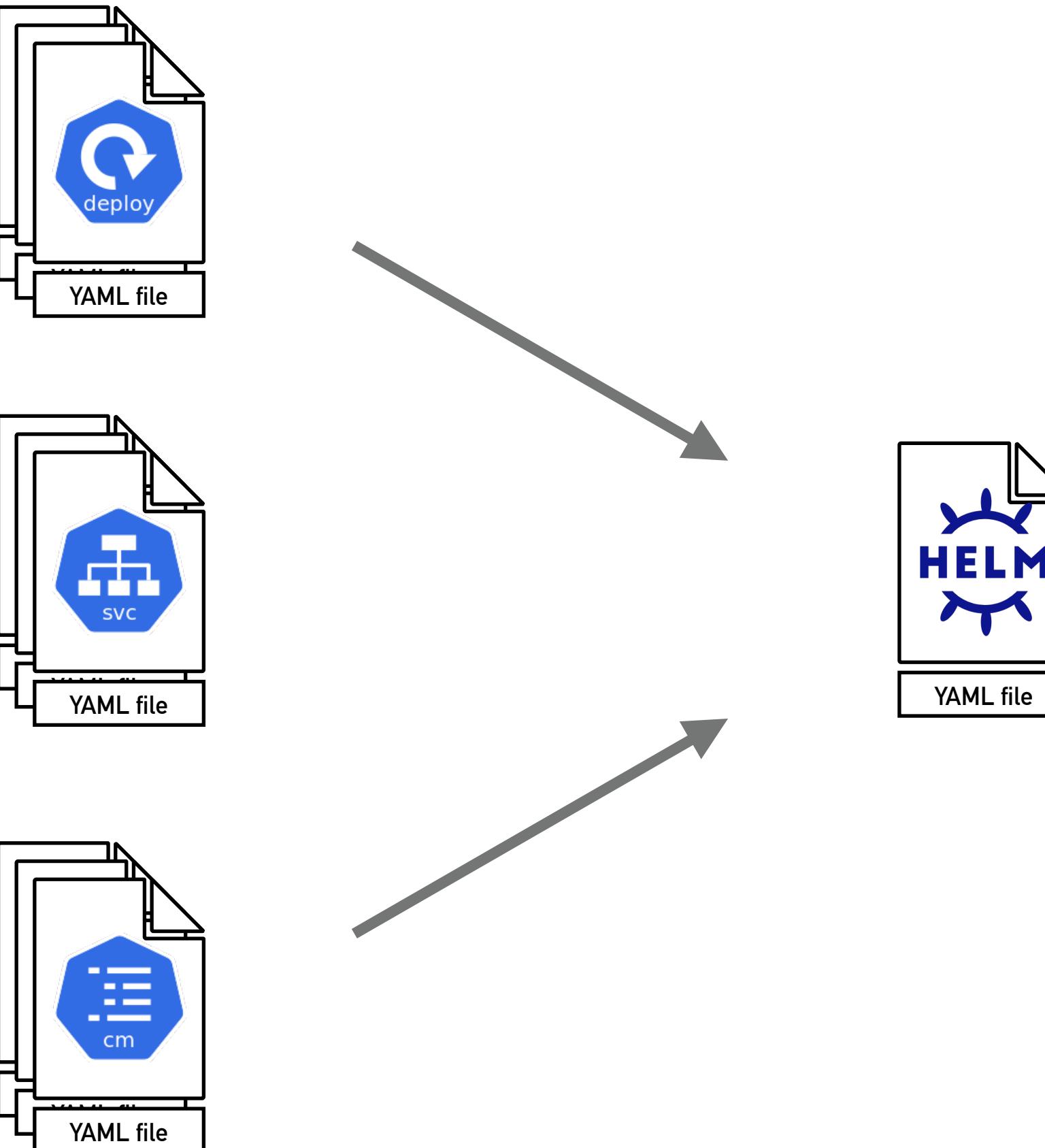


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.comtainer.image}}
          name: {{.Value.comtainer.name}}
          ports:
            - containerPort: {{.Value.port}}
```

TEMPLATE ENGINE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

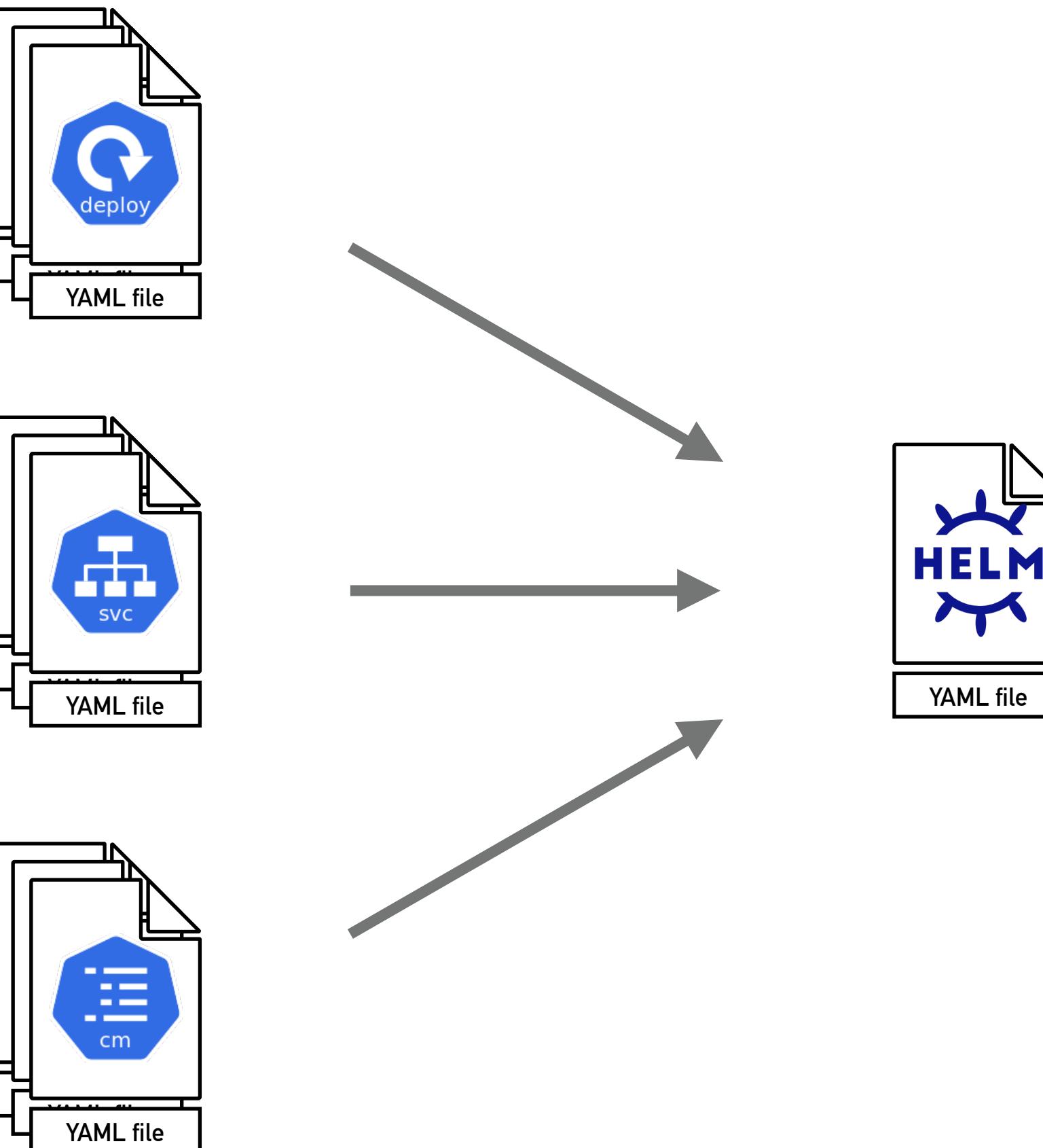


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.comtainer.image}}
          name: {{.Value.comtainer.name}}
          ports:
            - containerPort: {{.Value.port}}
```

TEMPLATE ENGINE

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
        - image: nginx
          name: my-web-on-nginx
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

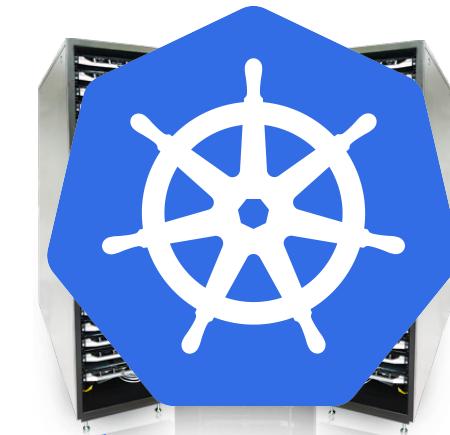


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.comtainer.image}}
          name: {{.Value.comtainer.name}}
          ports:
            - containerPort: {{.Value.port}}
```

TEMPLATE FOR ENVIRONMENT



kubernetes



kubernetes



kubernetes



kubernetes



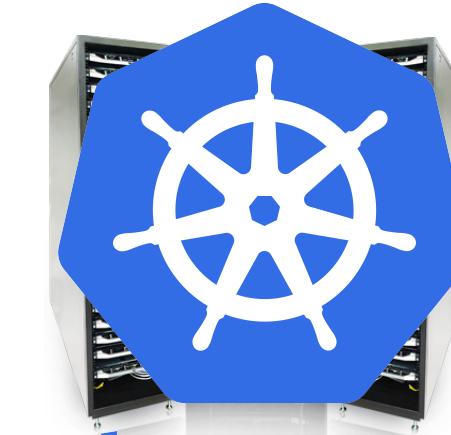
YAML file

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.container.image}}
          name: {{.Value.container.name}}
          ports:
            - containerPort: {{.Value.port}}
```

TEMPLATE FOR ENVIRONMENT



kubernetes



kubernetes



kubernetes

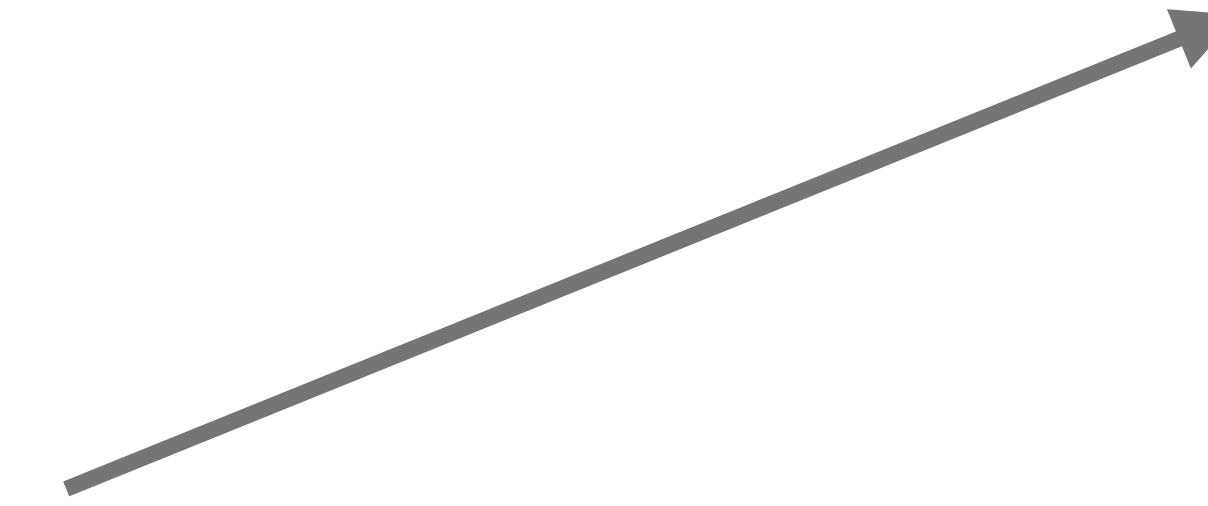


kubernetes

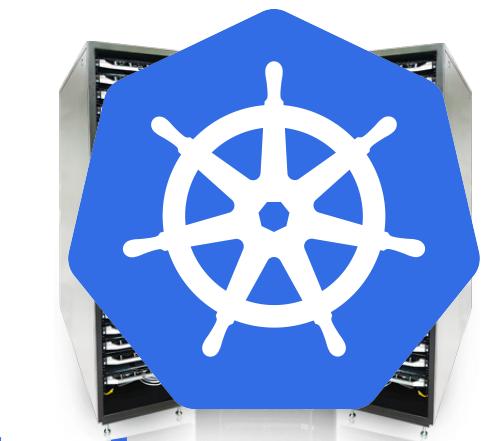


YAML file

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.container.image}}
          name: {{.Value.container.name}}
          ports:
            - containerPort: {{.Value.port}}
```



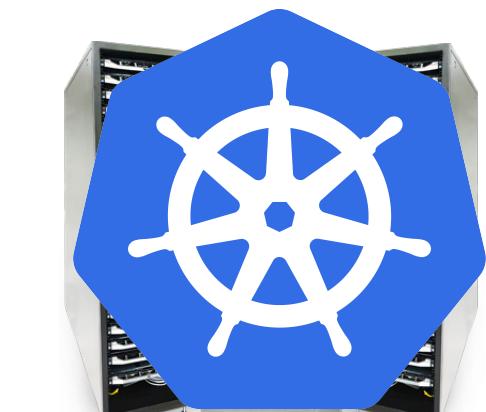
TEMPLATE FOR ENVIRONMENT



kubernetes



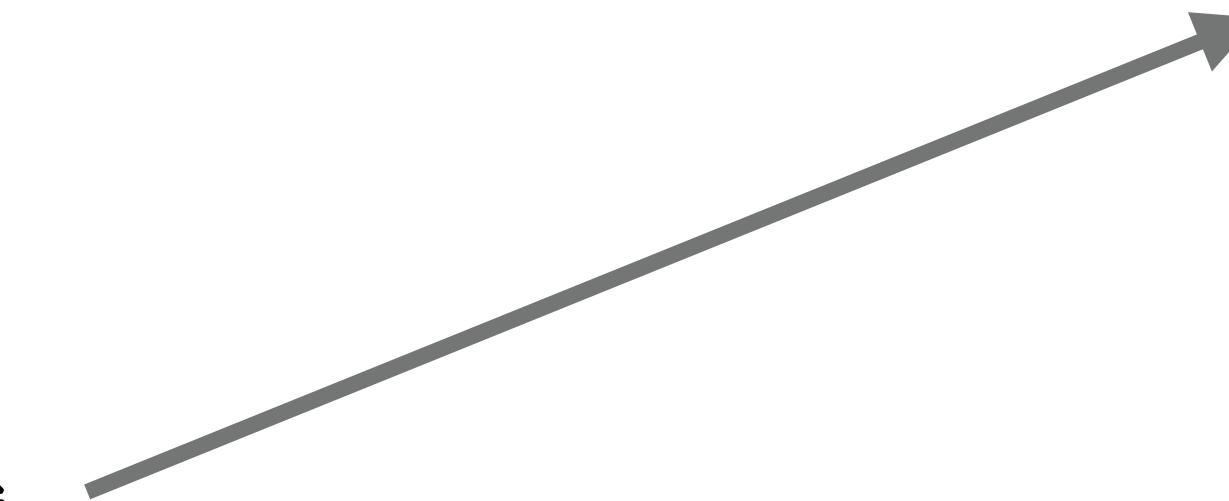
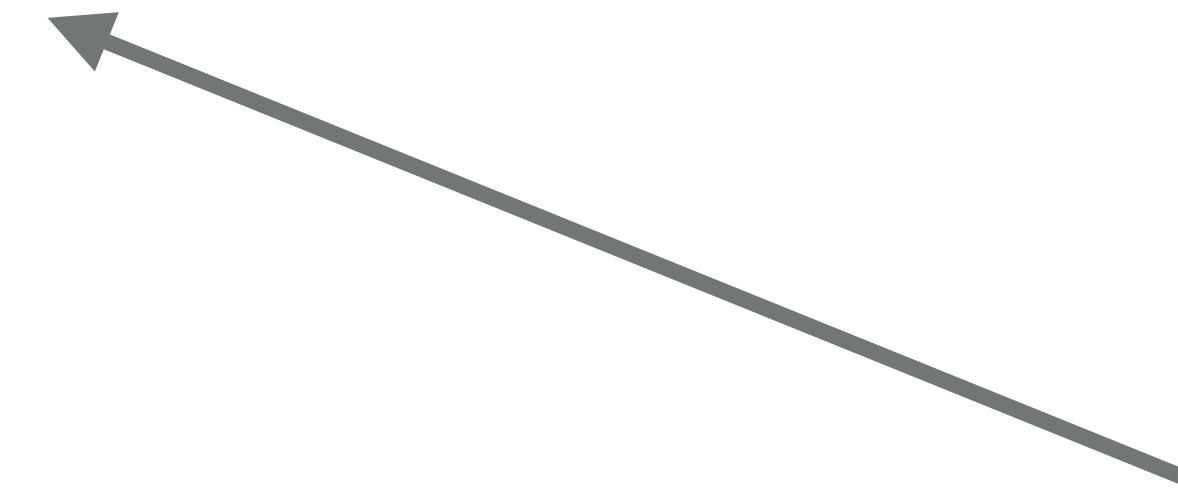
kubernetes



kubernetes



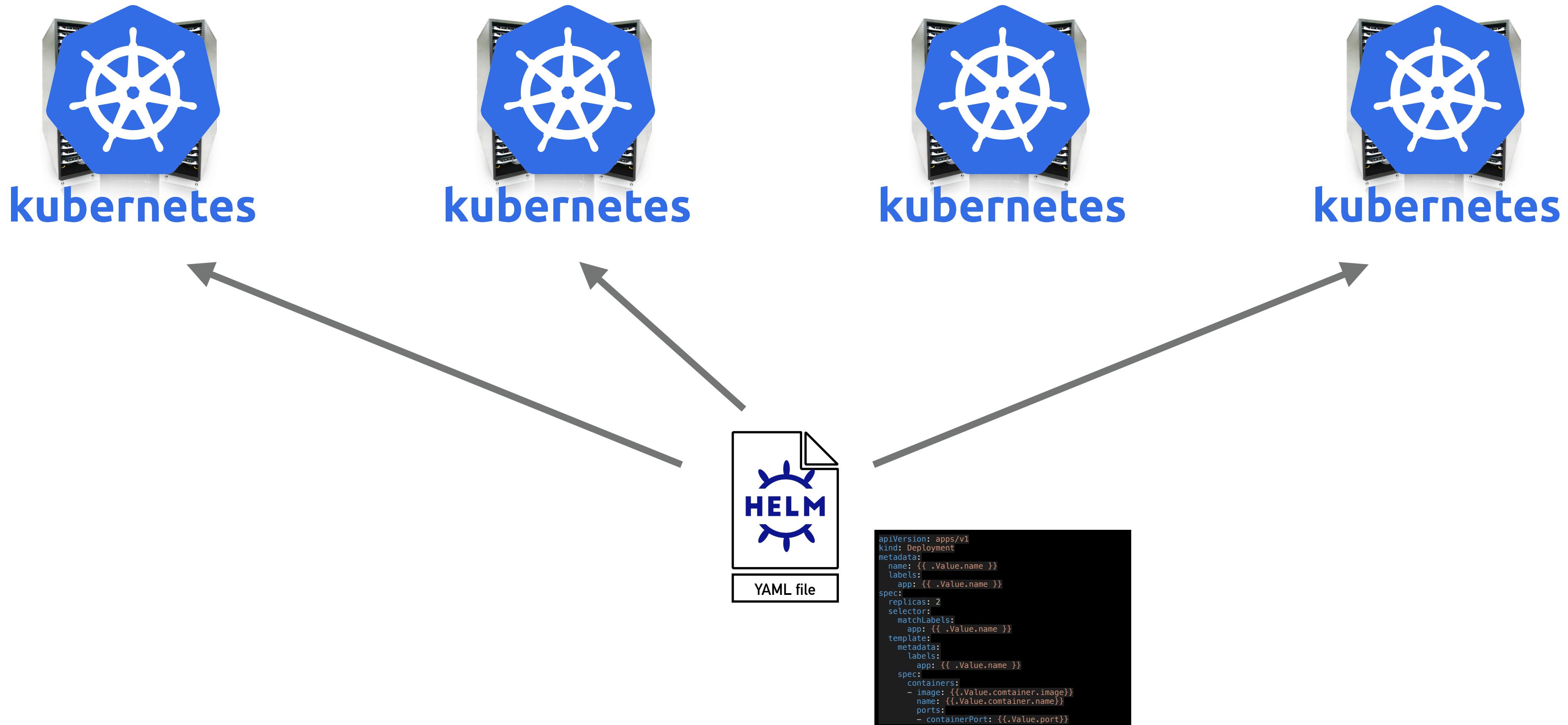
kubernetes



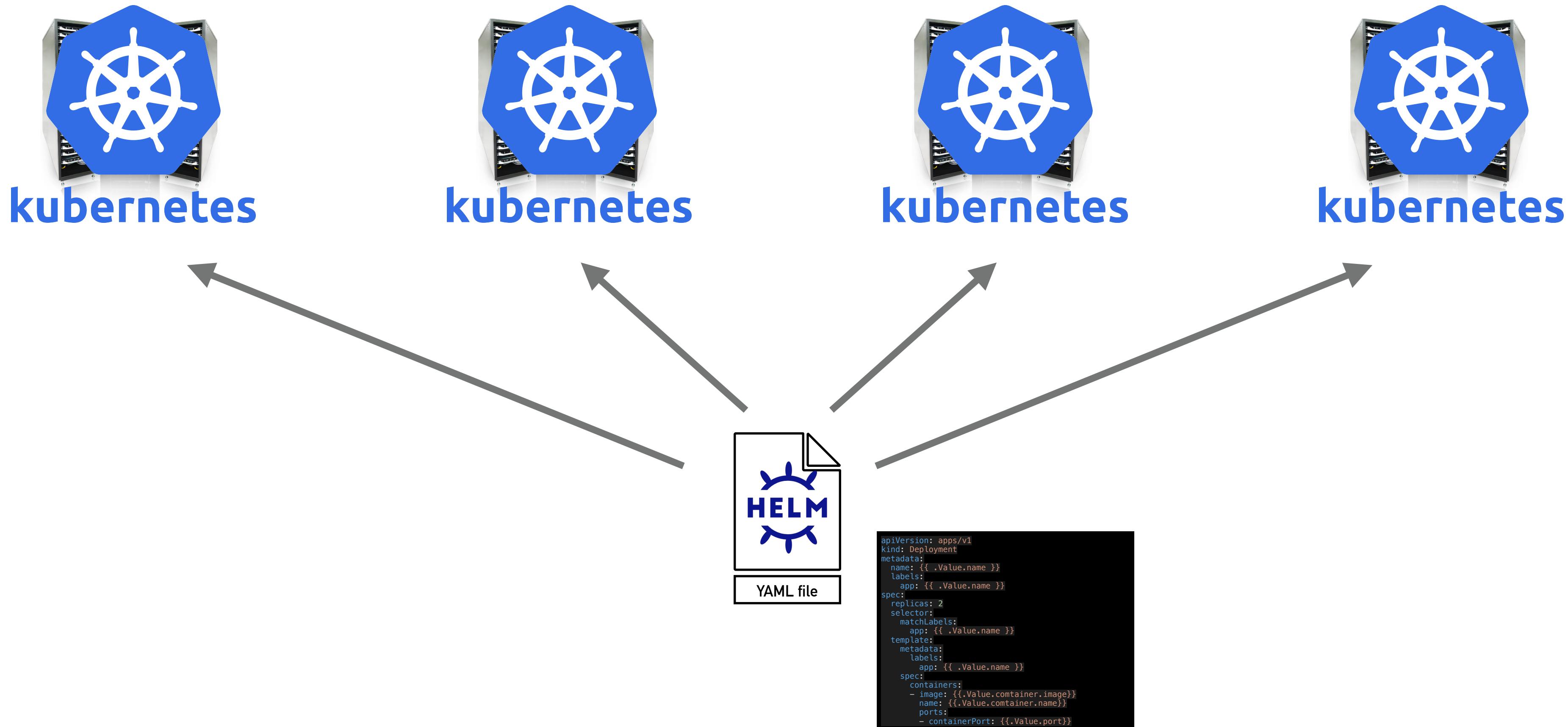
YAML file

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Value.name }}
  labels:
    app: {{ .Value.name }}
spec:
  replicas: 2
  selector:
    matchLabels:
      app: {{ .Value.name }}
  template:
    metadata:
      labels:
        app: {{ .Value.name }}
    spec:
      containers:
        - image: {{.Value.container.image}}
          name: {{.Value.container.name}}
          ports:
            - containerPort: {{.Value.port}}
```

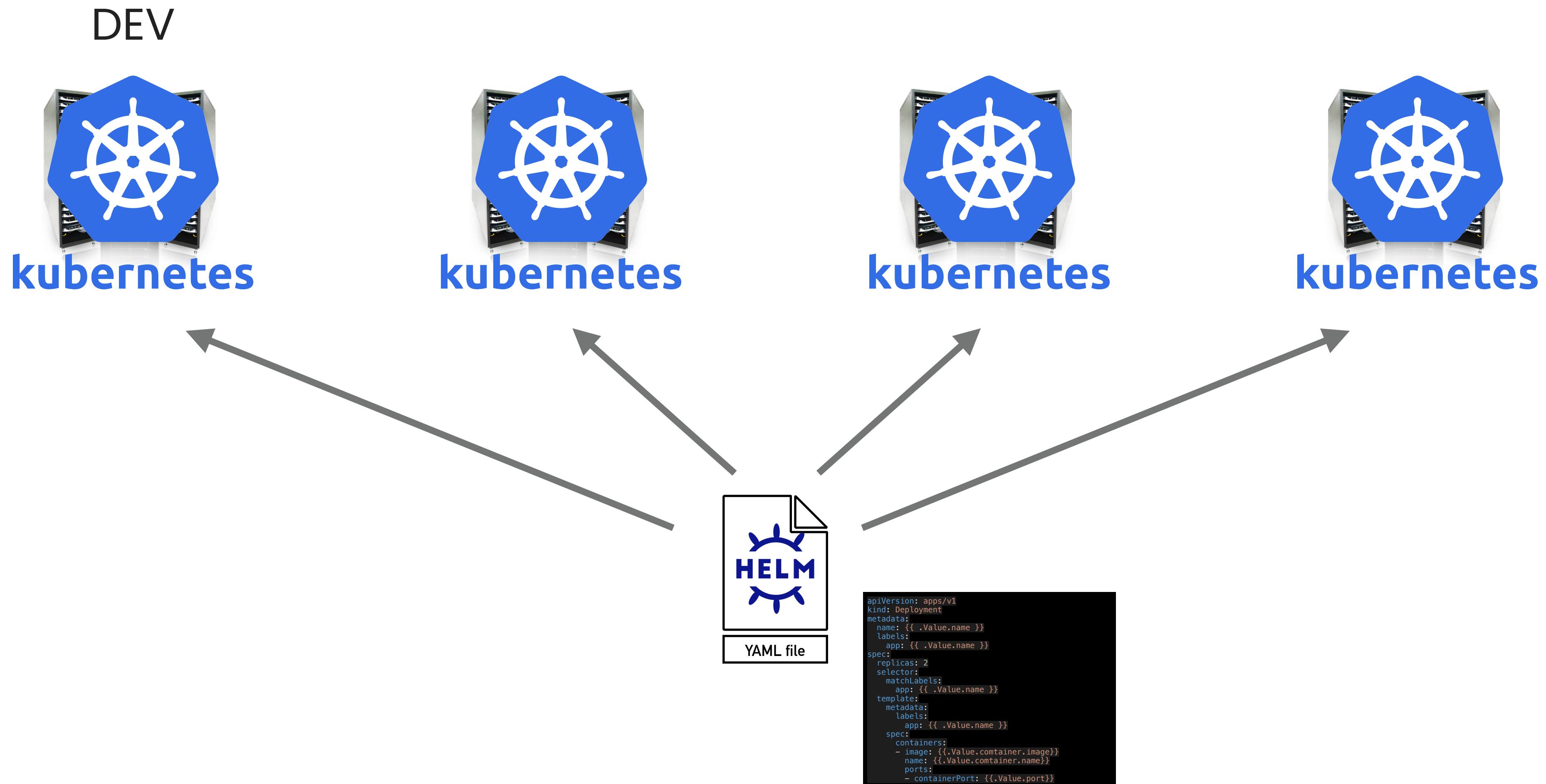
TEMPLATE FOR ENVIRONMENT



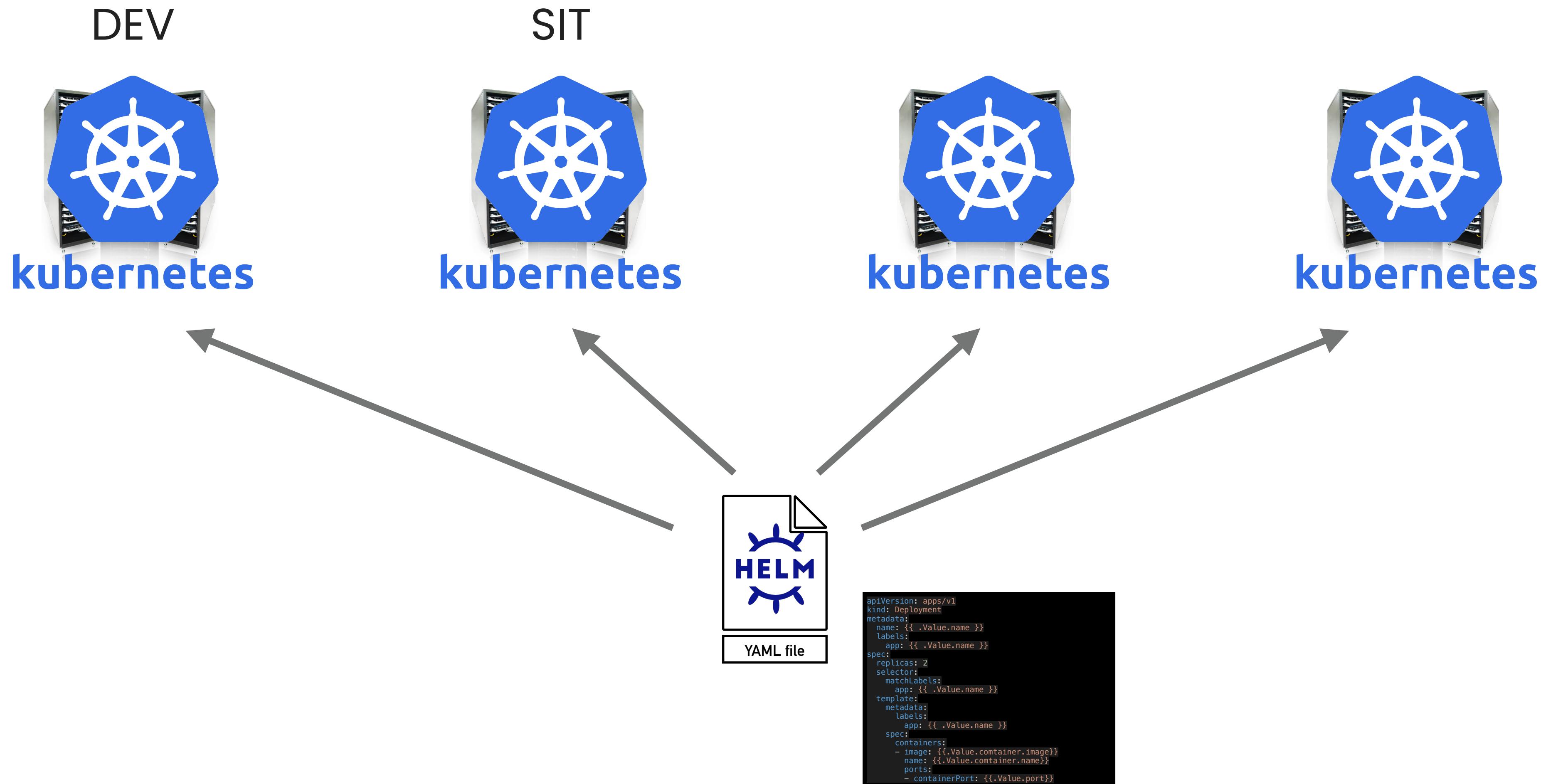
TEMPLATE FOR ENVIRONMENT



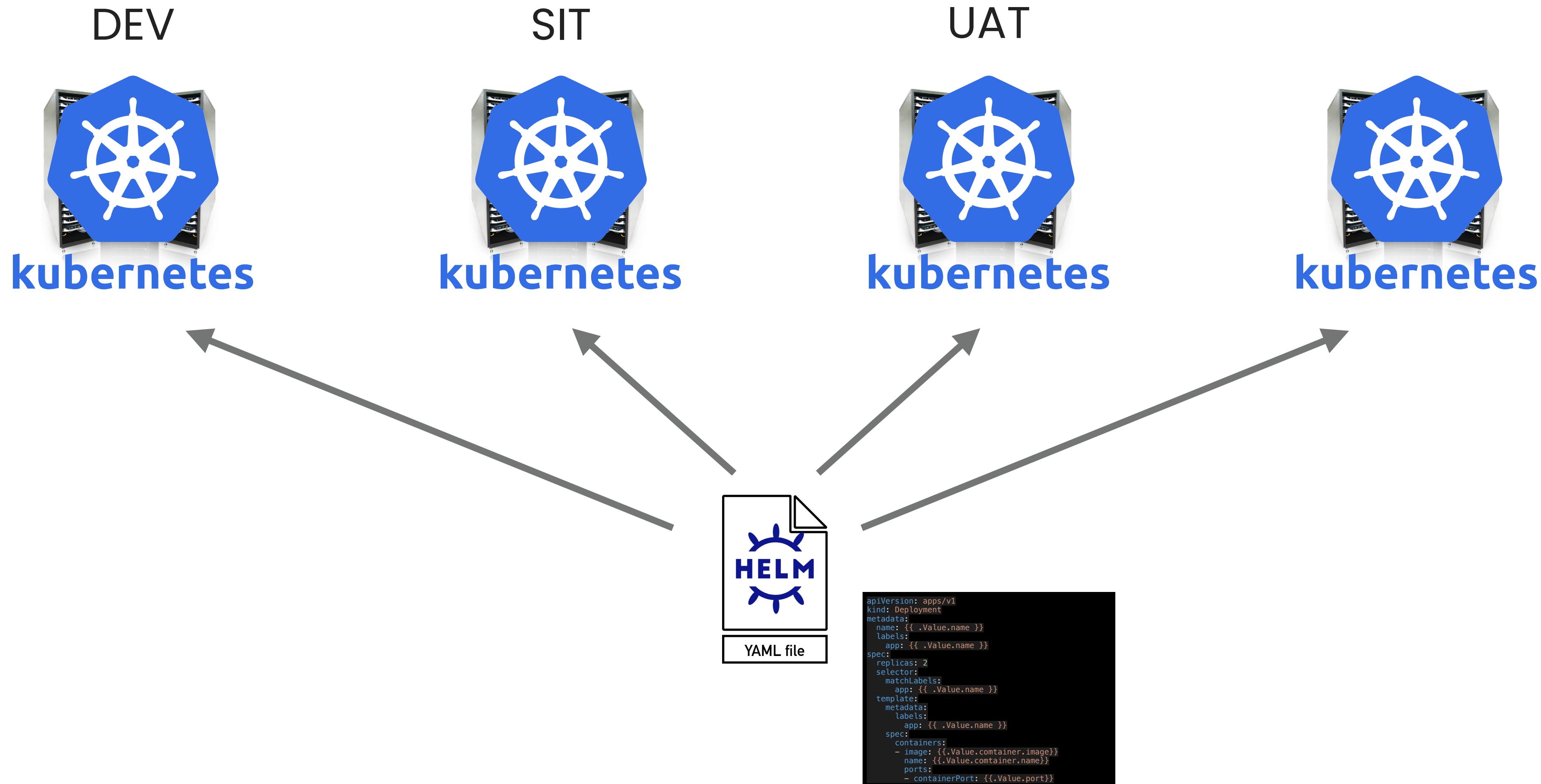
TEMPLATE FOR ENVIRONMENT



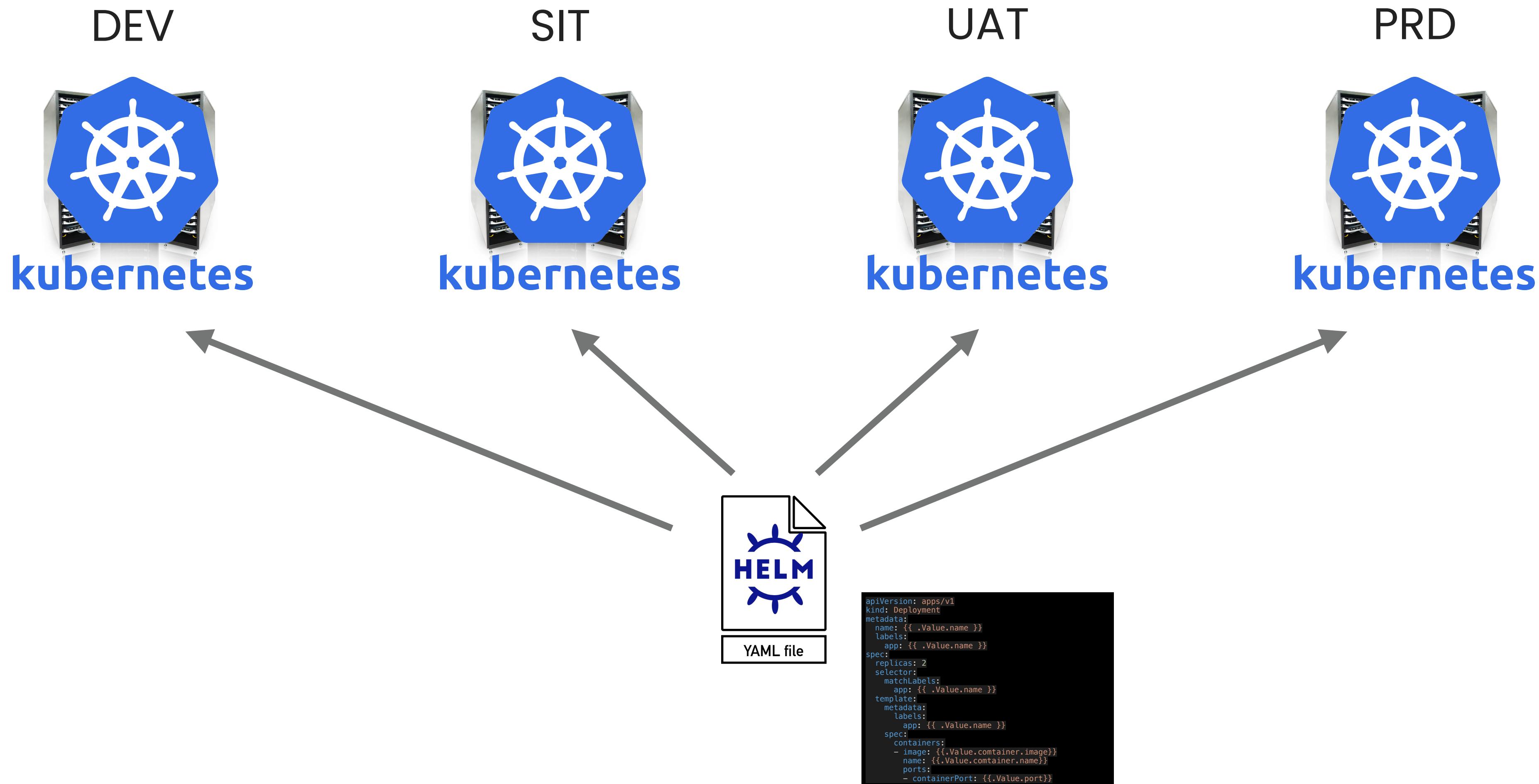
TEMPLATE FOR ENVIRONMENT



TEMPLATE FOR ENVIRONMENT



TEMPLATE FOR ENVIRONMENT

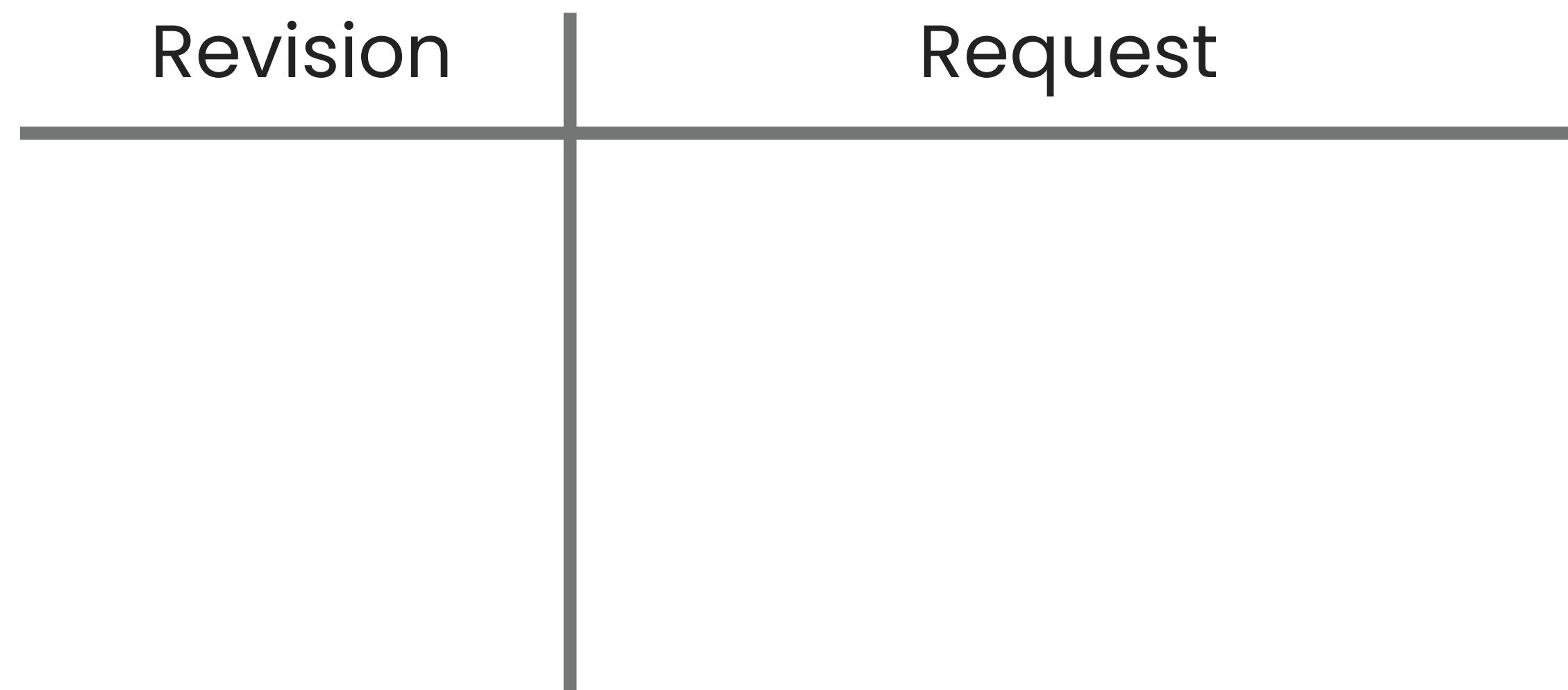


RELEASE MANAGEMENT

RELEASE MANAGEMENT

Keep track of all chart executions

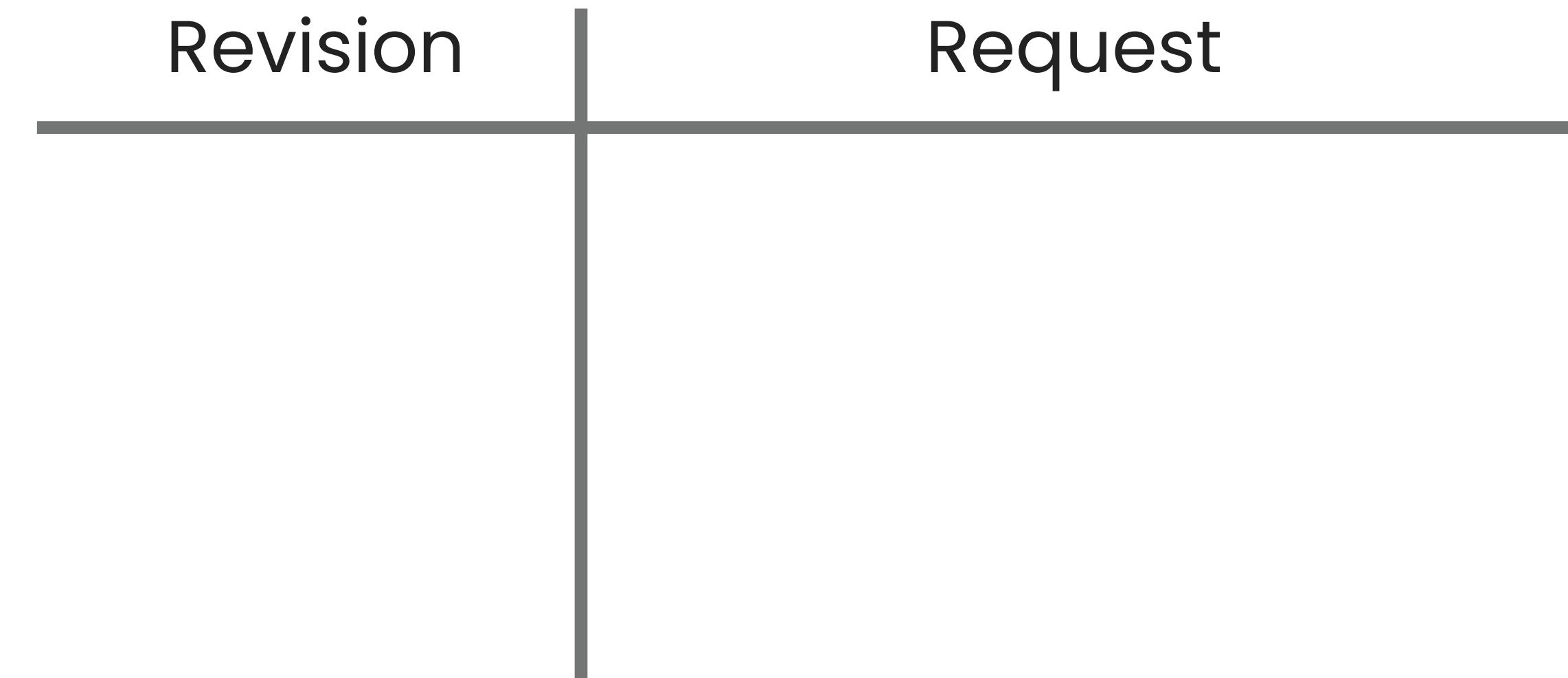
Changes are applied to existing deployment instead of creating new one



RELEASE MANAGEMENT

Keep track of all chart executions

Changes are applied to existing deployment instead of creating new one



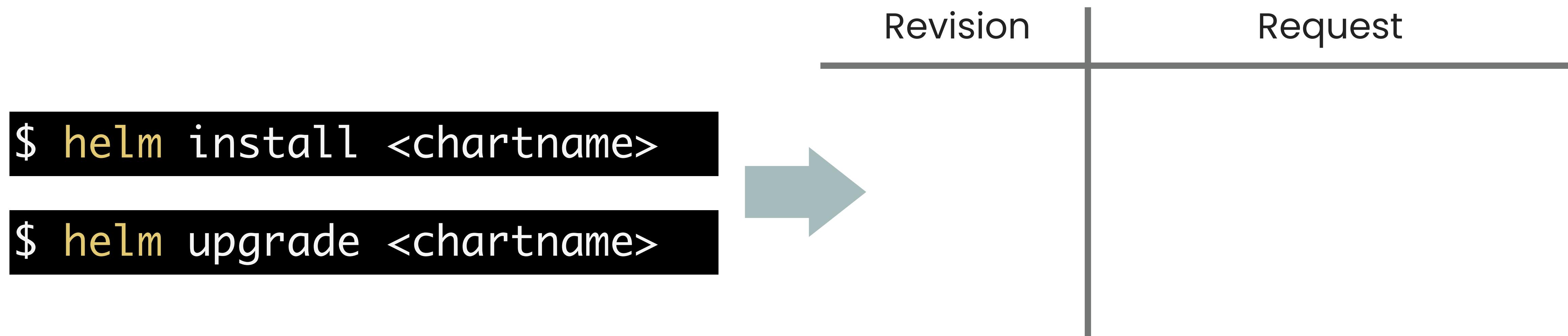
```
$ helm install <chartname>
```

```
$ helm upgrade <chartname>
```

RELEASE MANAGEMENT

Keep track of all chart executions

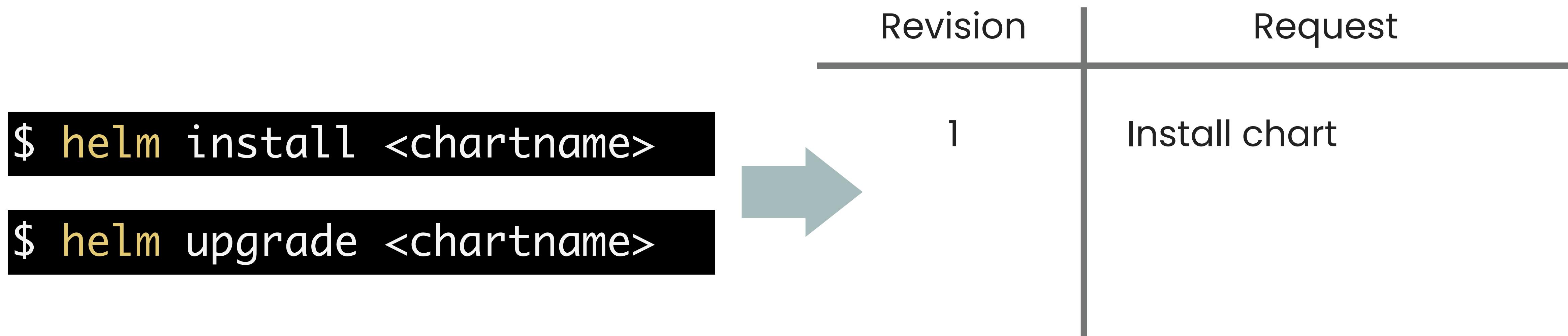
Changes are applied to existing deployment instead of creating new one



RELEASE MANAGEMENT

Keep track of all chart executions

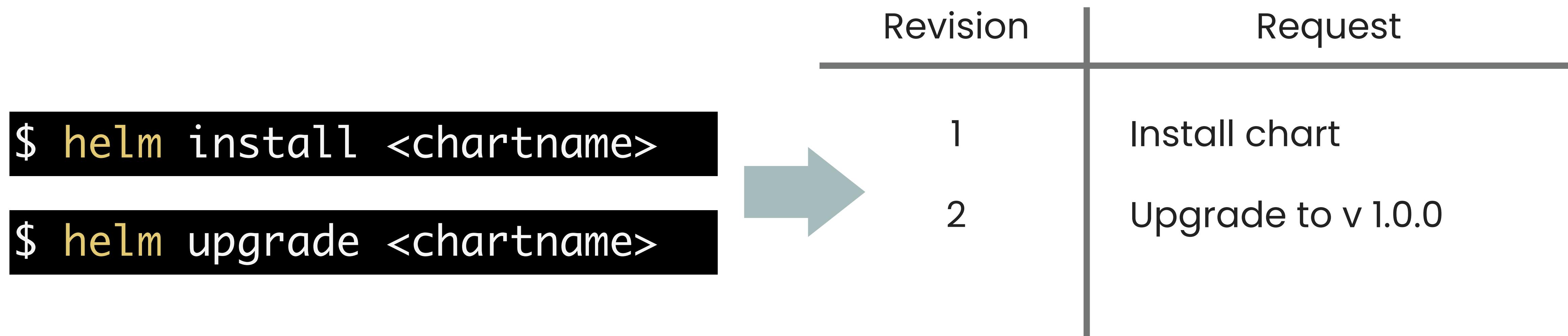
Changes are applied to existing deployment instead of creating new one



RELEASE MANAGEMENT

Keep track of all chart executions

Changes are applied to existing deployment instead of creating new one

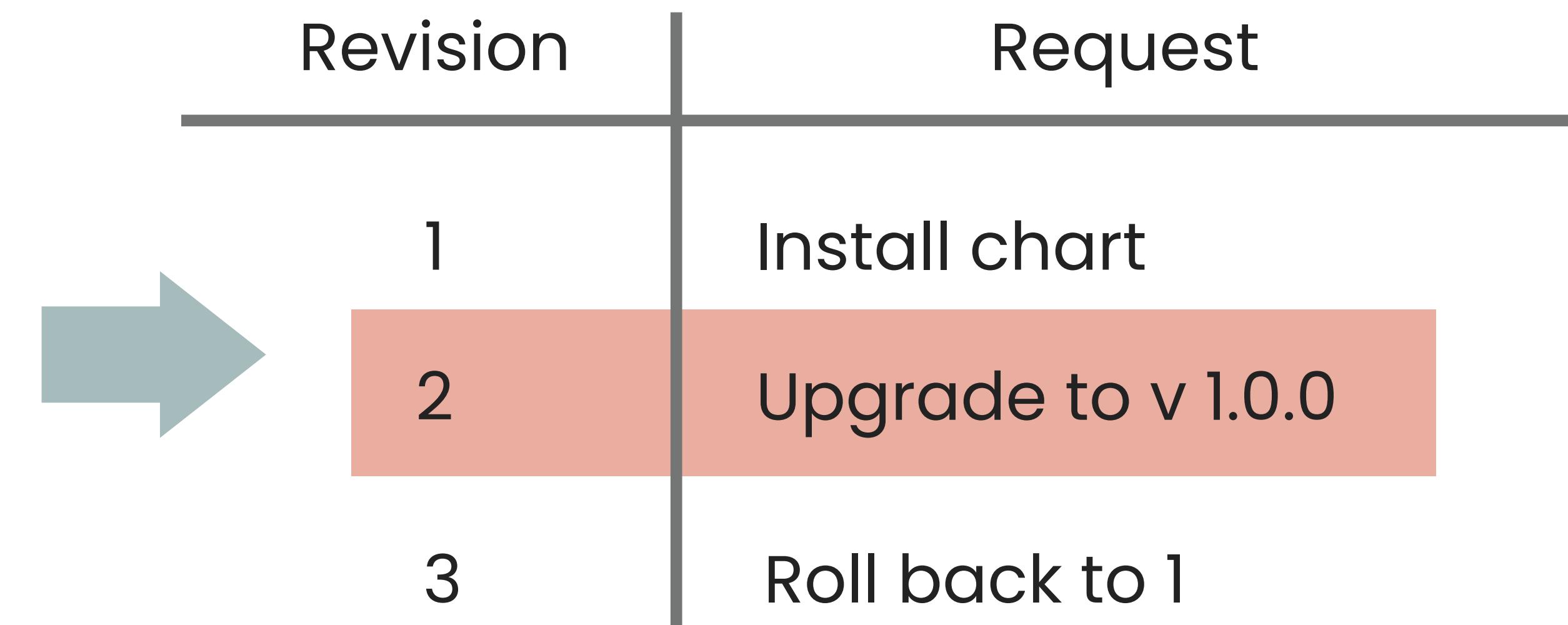


HANDLING ROLLBACK

```
$ helm install <chartname>
```

```
$ helm upgrade <chartname>
```

```
$ helm rollback 1
```



THANK YOU