Master Helm for Kubernetes

From Zero to Production Hero

By Salwan Mohamed

DevOps & Platform Engineer

What is Helm?

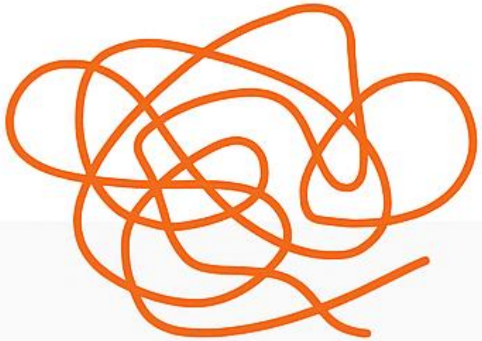
The package manager for Kubernetes

- Streamlines complex deployments
- Manages application packages (Charts)
- Enables version control & rollbacks
- Templating for configuration management

Why Helm is Essential

Without Helm

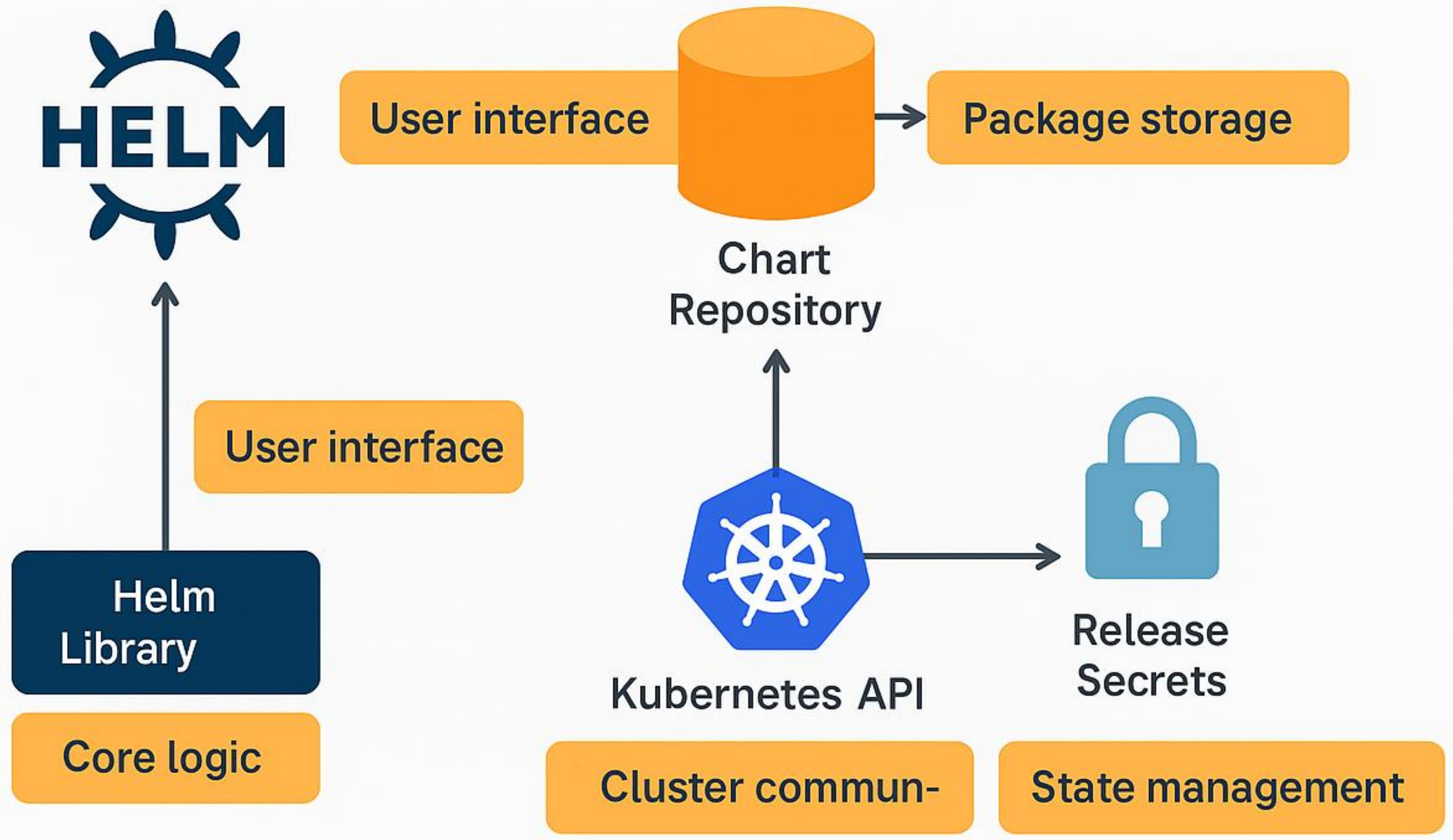
- ✗ Repetitive YAML files
- ✗ Manual deployment steps
- ✗ Configuration drift
- ✗ Difficult rollbacks



With Helm:

- ✓ Reusable templates
- ✓ One-command deployments
- ✓ Consistent configurations
- ✓ Easy version management

Helm 3 Architecture



Helm Charts Explained

What is a Chart?

- Collection of Kubernetes templates
- Configurable with values
- Versioned packages

Chart Structure:

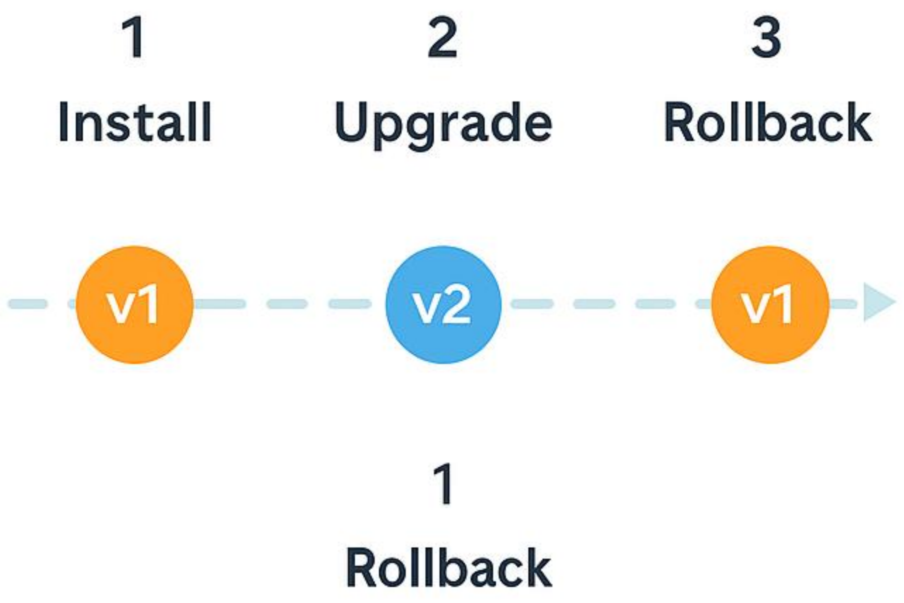
```
mychart/
├── Chart.yaml
└── values.yaml
── templates/
```

Understanding Releases

What is a Release?

- Deployed instance of a chart
- Has version history
- Tracks configuration changes

Release Lifecycle



Installing Helm 3

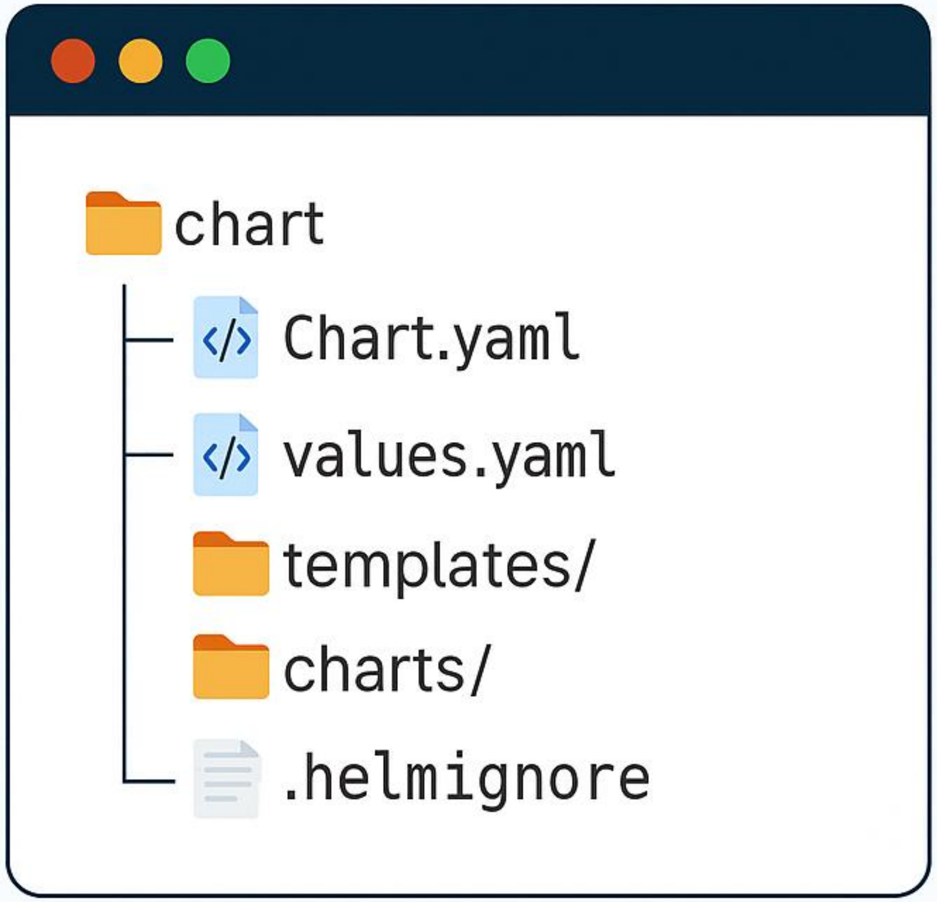
```
macOS: brew install helm
```

```
curl https://raw.gjithuburconten-t.com/helm/
helm/master/scripts/get-helm-3 | bash
```

```
Windows: choco install kubernetes-helm
```

```
Verify: helm version
```

Chart Directory Structure



```
chart
├─ Chart.yaml
├─ values.yaml
│  templates/
│  charts/
└─ .helmignore
```

Chart.yaml - Metadata

values.yaml - Default config

templates/ - Kubernetes manifests

charts/ - Dependencies

.helmignore - Exclude files

Chart.yaml Configuration

✓ **Required Fields:**

```
apiVersion: v2
name: myapp
version: 1.0.0
```

Optional Fields: "My application

```
description: opironal
maintainers: - name:'SalwanMohamed
dependencies: []
```

values.yaml Best Practices

✓ DO

- ✓ Provide sensible defaults
- ✓ Document all values
- ✓ Use nested structure

✗ DON’T

- ✗ Hardcode secrets
- ✗ Environment-specific values
- ✗ Complex logic

Create Your First Chart

```
helm create myapp
```

```
myapp/
├ Chart.yaml
├ values.yaml
├ templates/
│  ├ deployment.yaml
│  ├ service.yaml
└ _helpers.tpl
```

Next Steps: Customize templates & values

Helm Templating 101

Template Syntax:

```
metadata:
  name: {{ include 'myapp.fullname' .}
  labels:
    app: {{ .Values.app.name }
spec:
  replicas: {{ .Values.replicaCount }
```

Key Elements:
{{ }}, .Values, include

Installing Charts

From Repository:

```
helm repo add bitnami
https://charts.bitnami.com/om/bitnami
```

Local Chart:

```
helm install my-nginx
helm install myapp
./myapp
```

With Custom Values:

```
helm install myapp
./myapp -f values.yaň ->values.yaml
```

Managing Helm Repositories

1. **Add Repository:**

```
helm repo add stable
https://charts.helm.sh/stable
```

2. **Update Repositories:**

```
helm repo update
```

3. **Search Charts:**

```
helm search repo nginx
```

nginx

stable/nginx

0.14.3 1.15.0

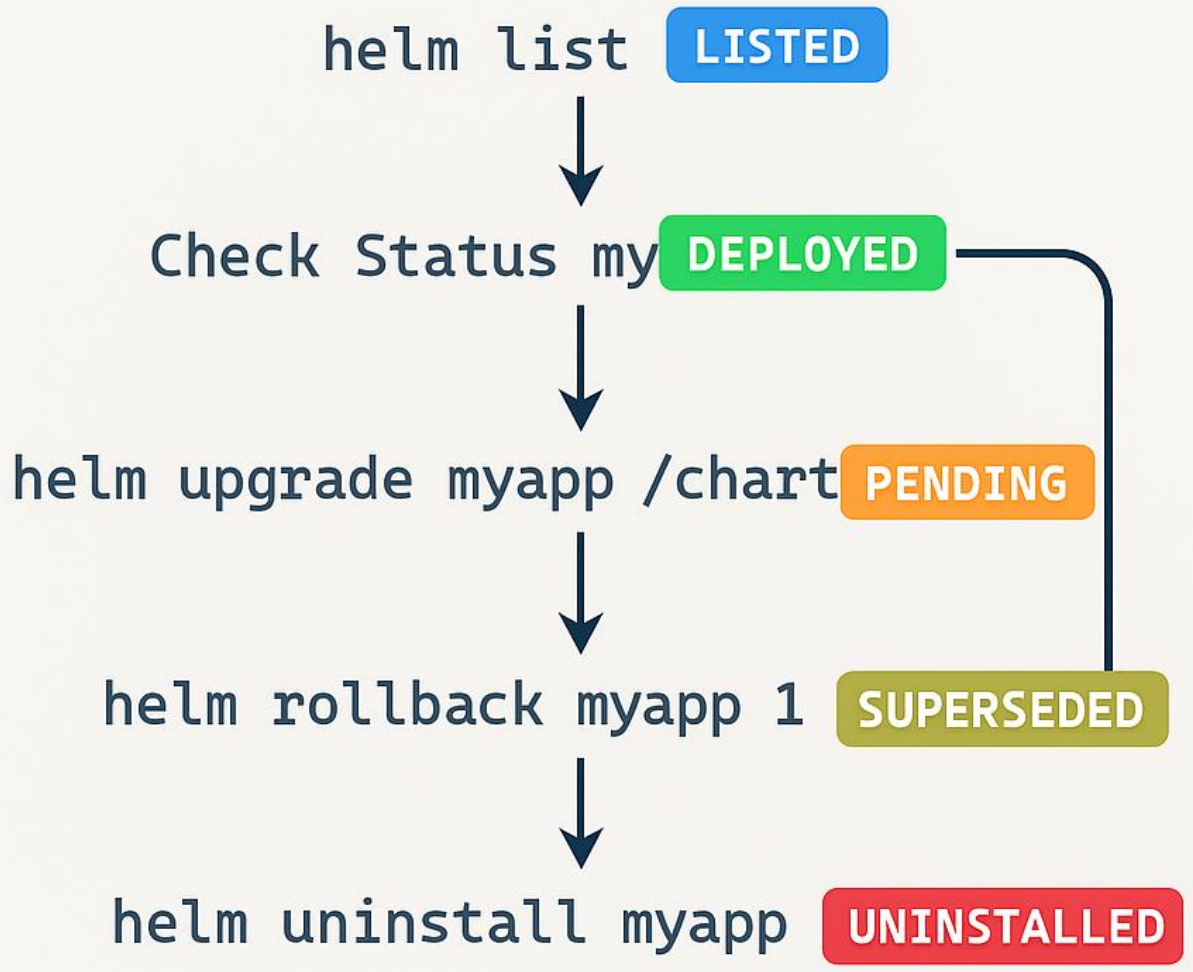
Popular lightweight web server

stable/nginx-ingress

1.6.4 0.21.0

An nginx Ingress controller that uses Config...

Managing Releases



```
helm list
Check Status my
helm upgrade myapp /chart
helm rollback myapp 1
helm uninstall myapp
```

Template Validation

Lint Chart

```
helm lint ./myapp
```

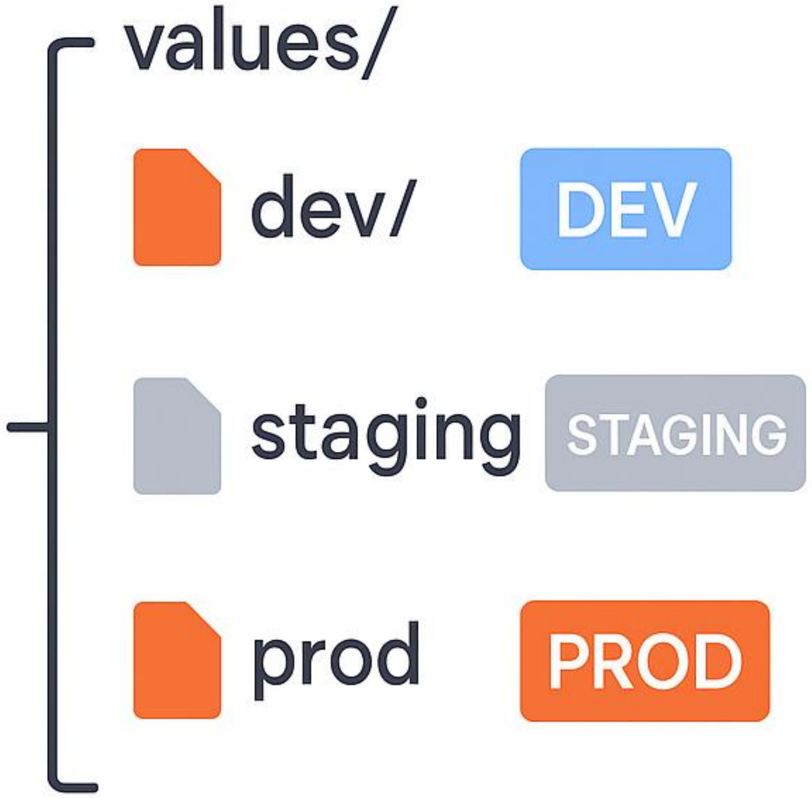
Dry Run

```
helm install
--dry-run -debug
myapp ./myapp
```

Template Rendering

```
helm template
myapp ./myapp
```

Multi-Environment Strategy

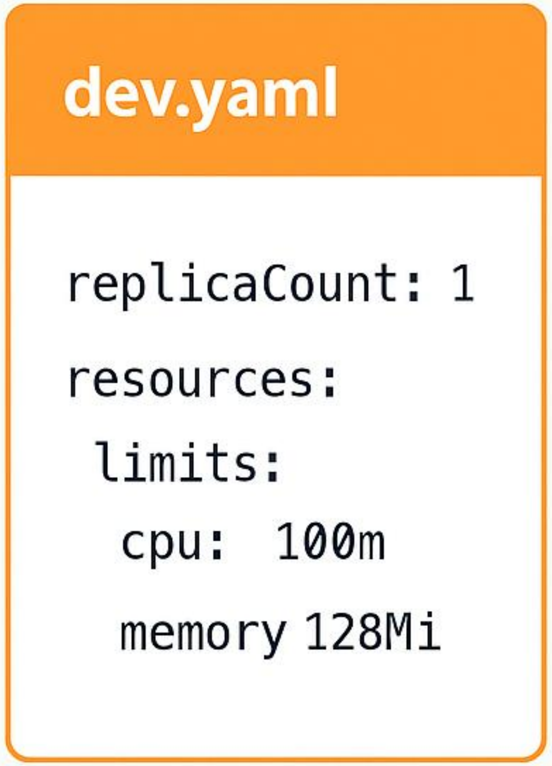


Deployment Examples:

```
helm install app-dev
./chart -f values/dev.yml
```

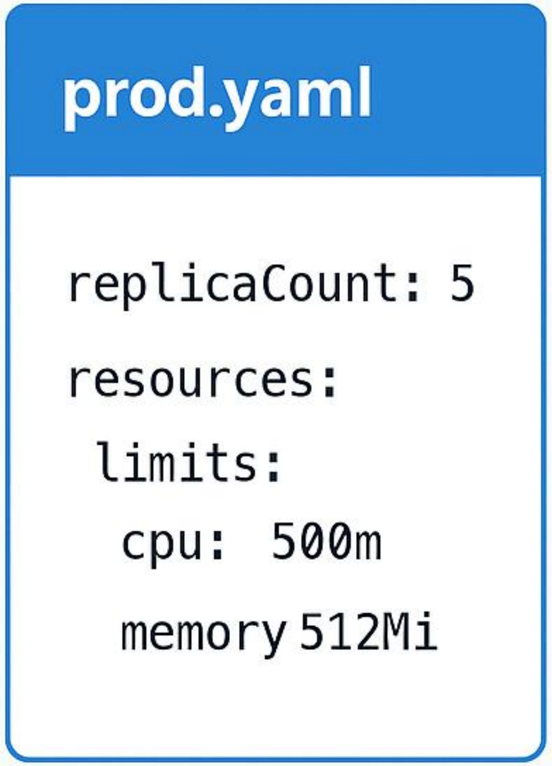
```
helm install app-prod
./chart -f values/prod.yml
```

Environment Configuration



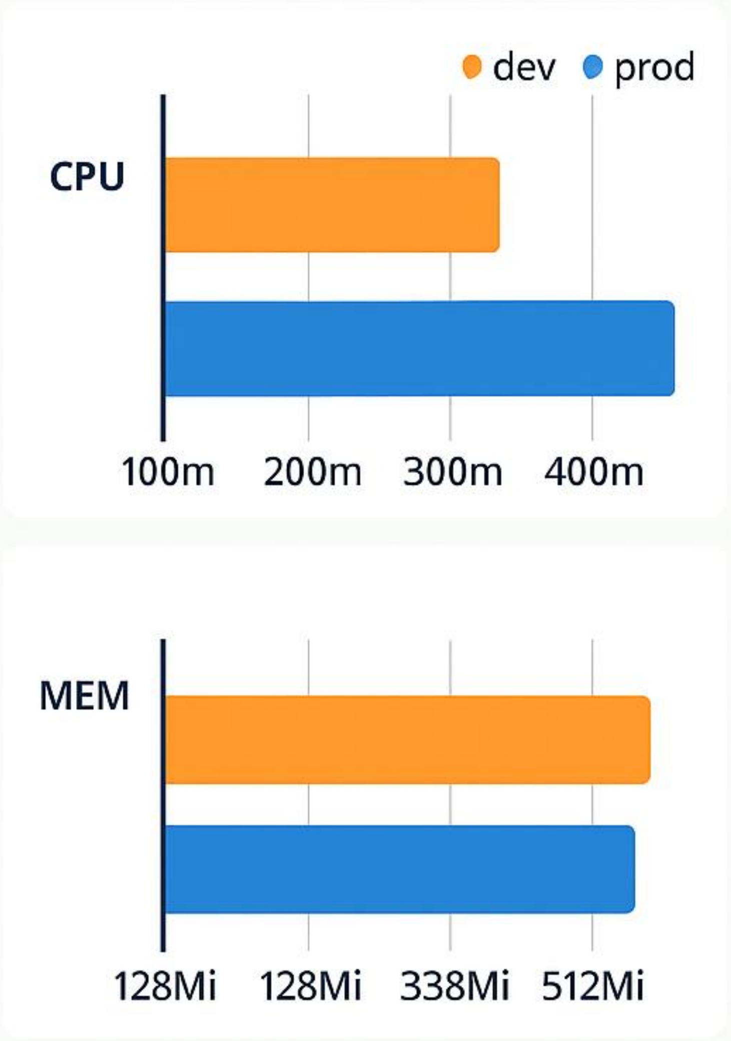
dev.yaml

```
replicaCount: 1
resources:
 limits:
  cpu:  100m
  memory 128Mi
```



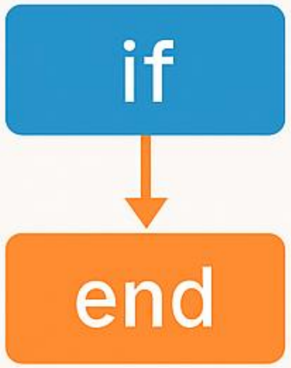
prod.yaml

```
replicaCount: 5
resources:
 limits:
  cpu:  500m
  memory 512Mi
```



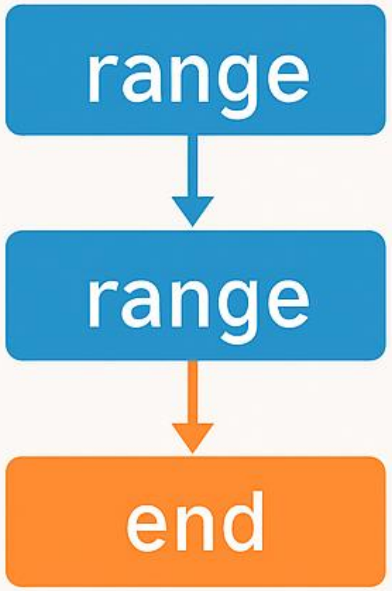
Advanced Template Functions

Conditionals:



```
{{- if .Values ingress.enabled }}
 # Ingress configuration
{{- end }}
```

Loops:



```
{{- range .Values.env }}
  - name: {{ .name }
  - value: {{ .value}}
{{- end }}
```

kubernetes

HELM

_helpers.tpl Usage

Define Helper:

```
{{- define "myapp.labels -}

app.kubernetes.io/name:
{{ include "myapp.name"  .}}
app.kubernetes.io/instance:
{{ .Release.Name }}
{{- end }
```

Use Helper:

```
metadata:
 labels:
  {- include "myapp.labels
   . | nindent 4 }
```

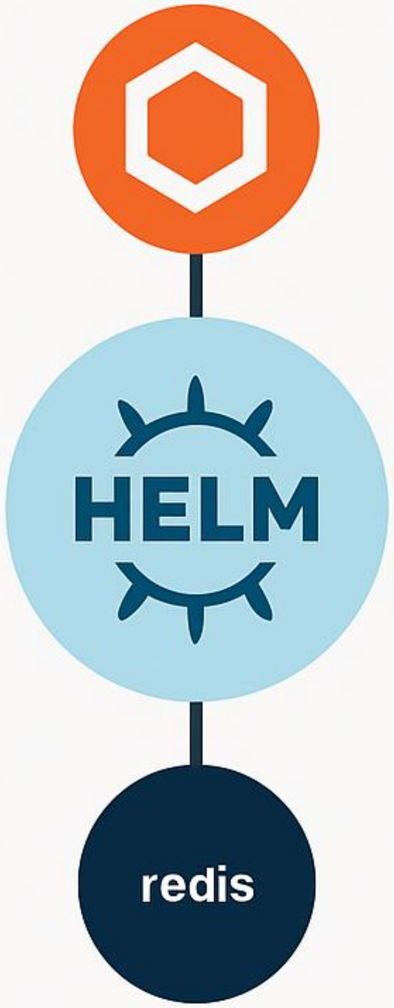
DRY

kubernetes HELM

Chart Dependencies

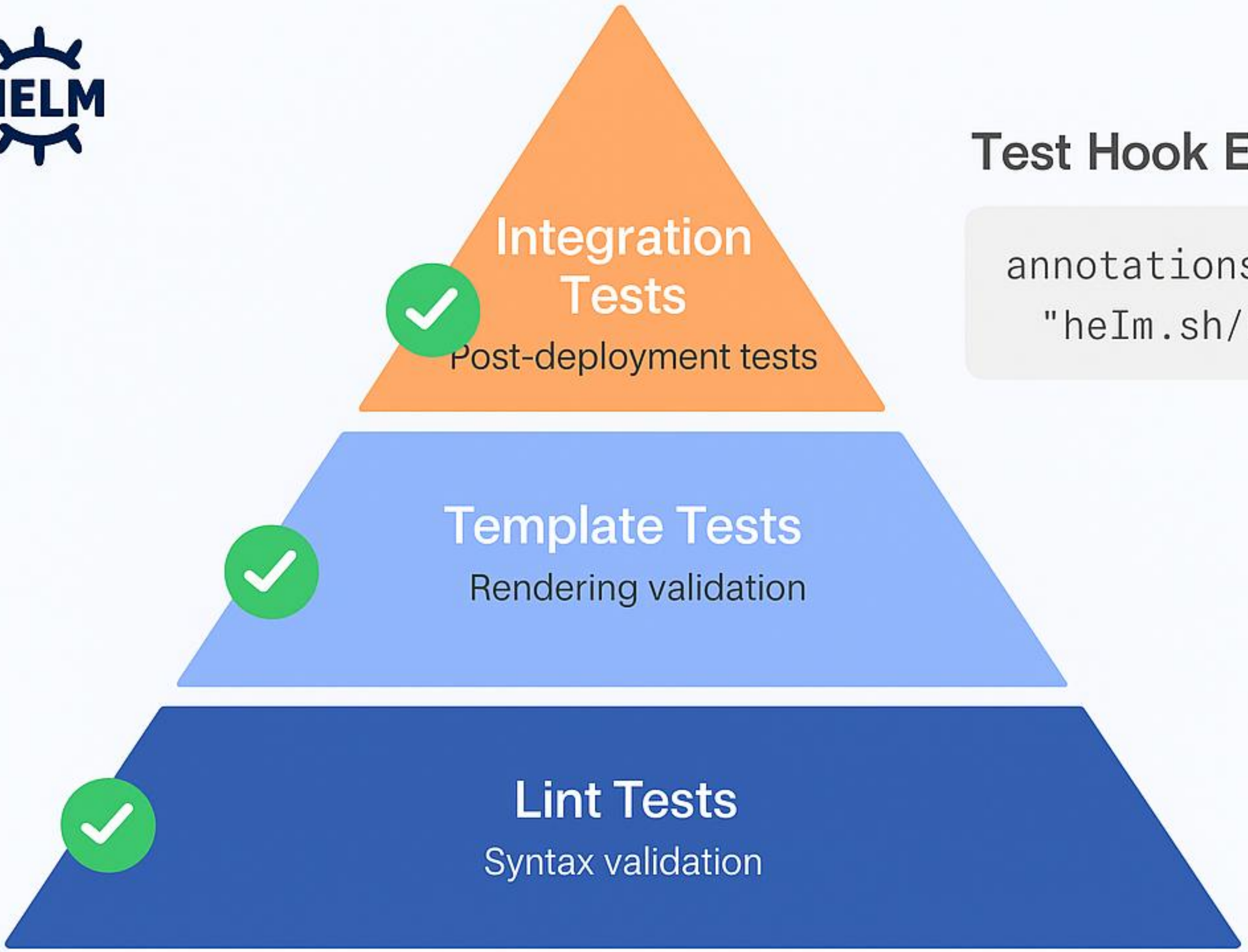
Chart.yaml:

```
dependencies:
  - name: redis
    version: "17.1.2"
  - repository:
    "https://charts.
    bitnami.com/
    bitnami
```



```
helm dependency update
helm dependency build
```

Chart Testing Strategy

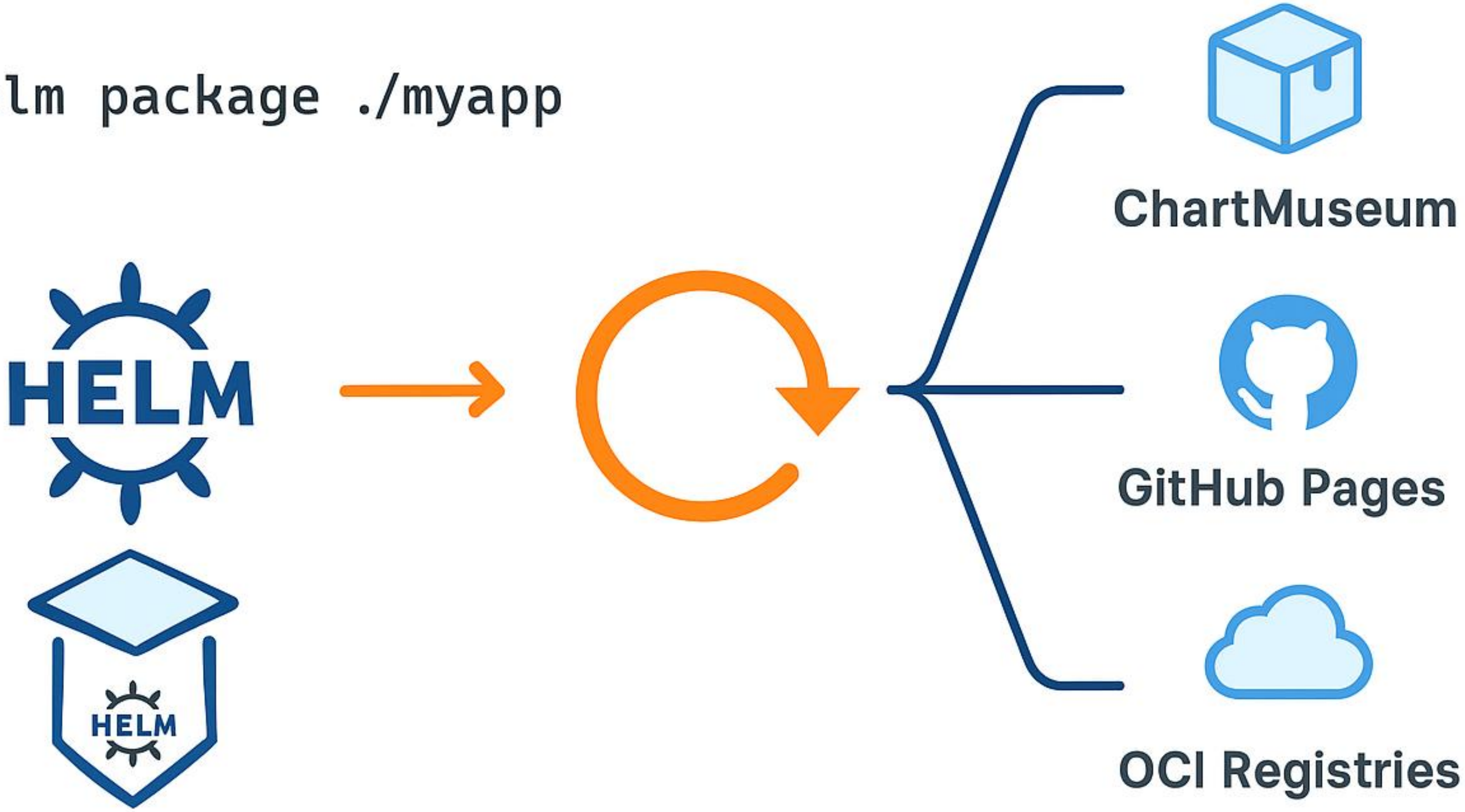


Test Hook Example:

```
annotations:
  "helm.sh/hook": test
```

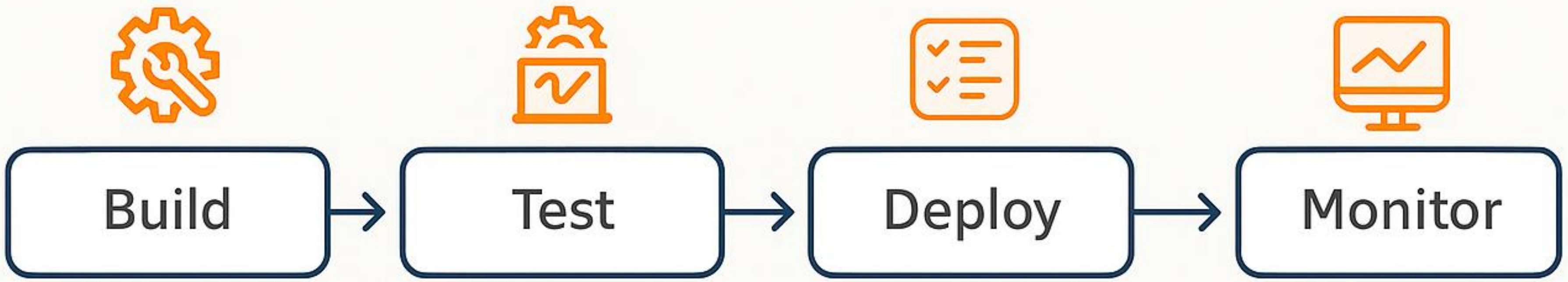
Chart Distribution

```
helm package ./myapp
```



Private repositories

Helm in CI/CD



argoCD

Jenkins

FLUX

GitLab CI

GitHub Actions Example:

```
-  name: Deploy
   run: |
   helm upgrade ---install
   myapp ./chart \
   --namespace prcoduction
   --values values/prod.yam
```

Helm Security Essentials

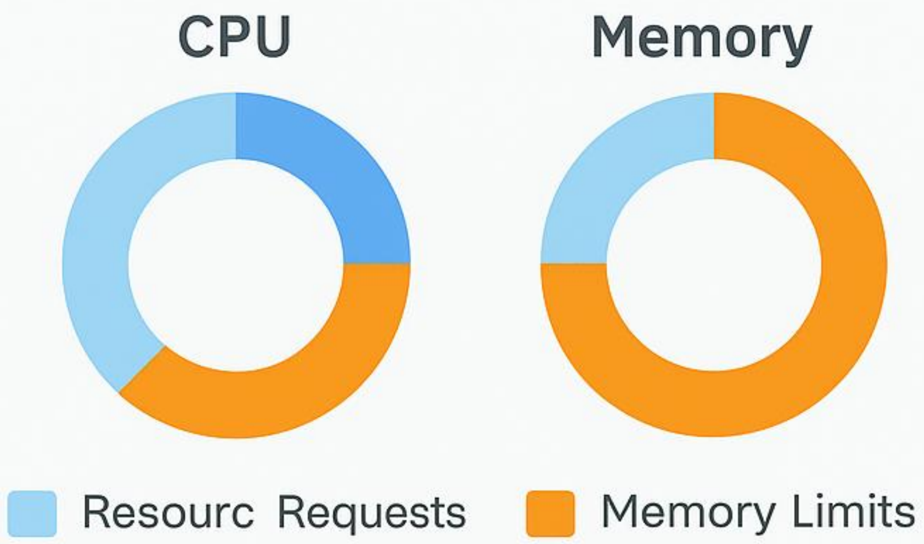
- Never store secrets in values.yaml
- Use external secret management
- Implement RBAC properly
- Regular security audits

Tools: Sealed Secrets, External Secrets Operator

Resource Best Practices

Always Define:

```
resources:
   requests:  100m
   memory:    128Mi
limits:
   requests:  500m
   memory:    512Mi
```



Include Probes:

- Liveness probes
- Readiness probes
- Startup probes

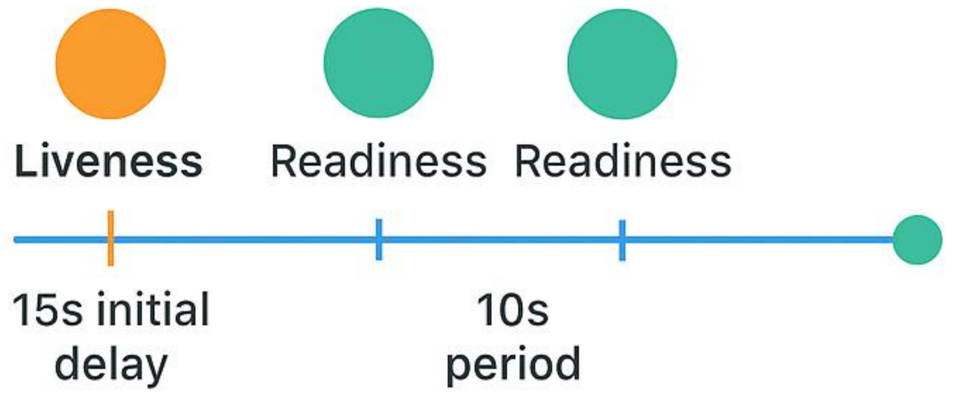
Health Checks Configuration

Liveness Probe:

```
livenessProbe:
  httpGet:
    path: /healtzz
    port: 8080
  initialDelaySeconds: 15
  periodSeconds: 10
```

Readiness Probe:

```
readinessProbe:
  httpGet:
    path: /ready
    port: 8080
```



Common Issues & Solutions

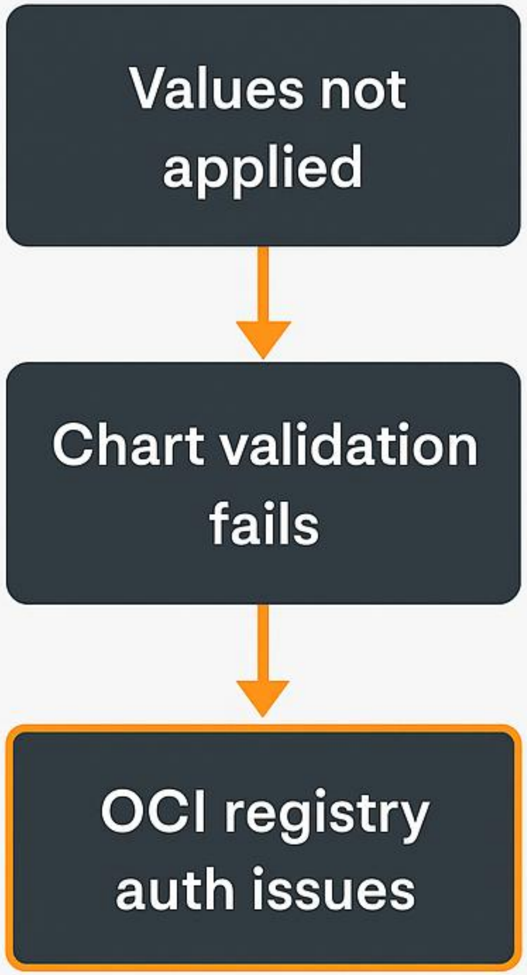
Template rendering errors

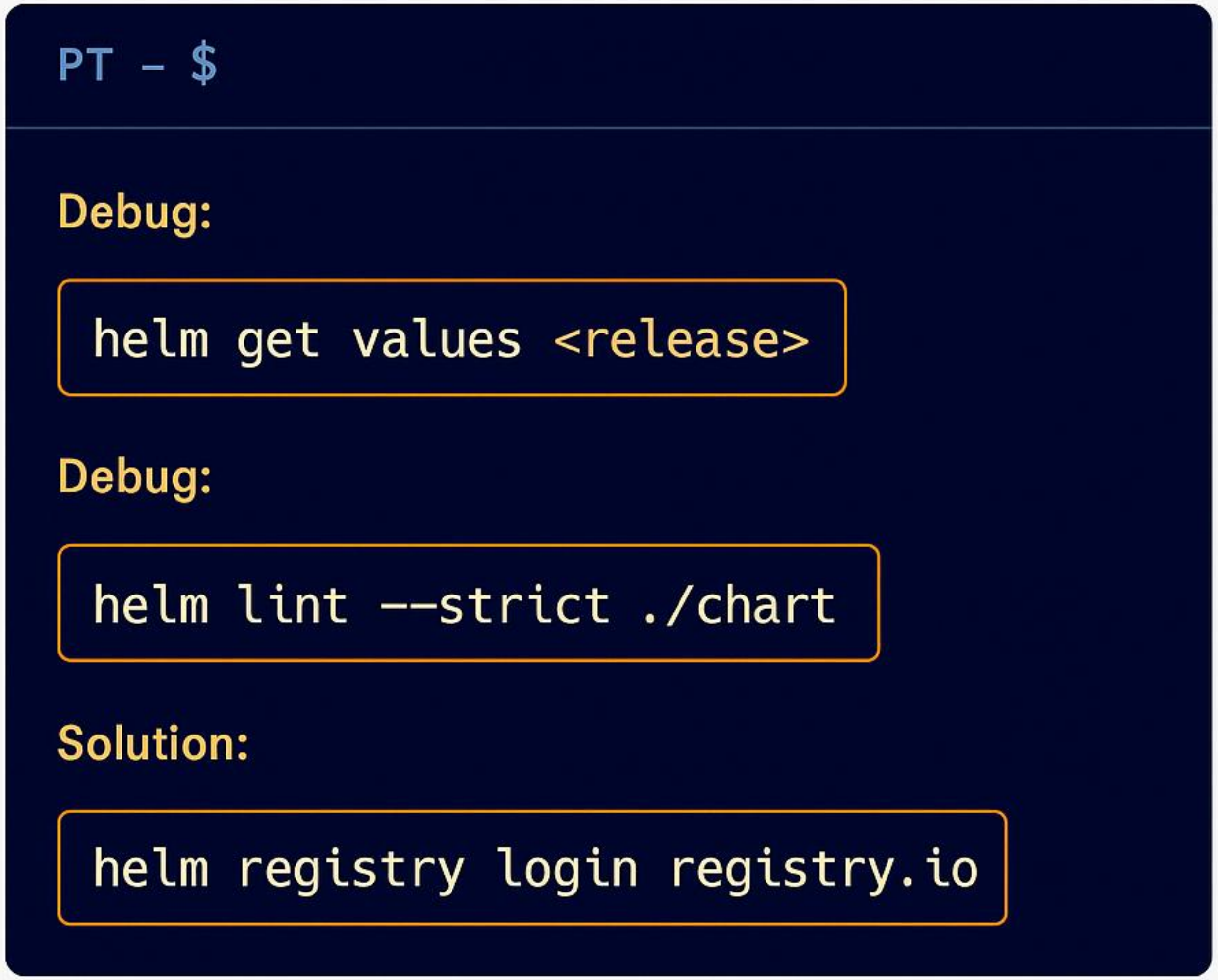
Use **helm** template for debugging

Release stuck in pending-upgrade

`helm rollback <release> <revision>`

Advanced Troubleshooting





```
PT - $
```

Debug:

```
helm get values <release>
```

Debug:

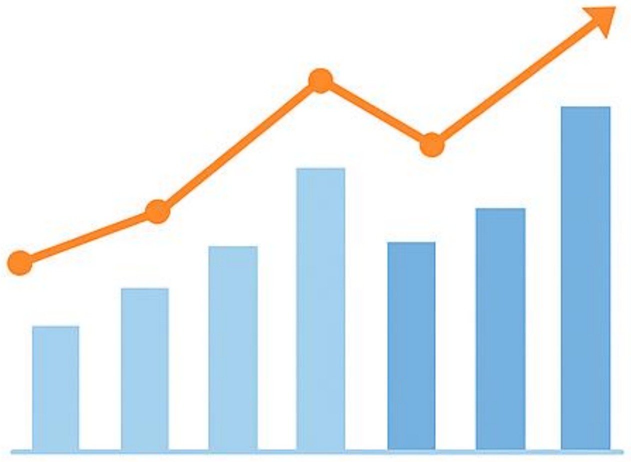
```
helm lint --strict ./chart
```

Solution:

```
helm registry login registry.io
```

Helm Performance Tips

- Use .helmignore effectively
- Minimize chart dependencies
- Keep templates simple
- Use atomic installations



Track deployment times

Helm vs Kustomize

VS

Helm Strengths

- ✓ Templating & packaging
- ✓ Release management
- ✓ Rollback capabilities

Kustomize Strengths

- ✓ Overlay-based approach
- ✓ Built into kubectl
- ✓ GitOps friendly

When to Use Each

Real Example: Web Application

Ingress
(SSL termination)

Deployment
(3 replicas)

Service
(LoadBalancer)

ConfigMap
(app config)

HELM

Values Structure

```
app:
  name: webapp
  image:nginx:1.21
```

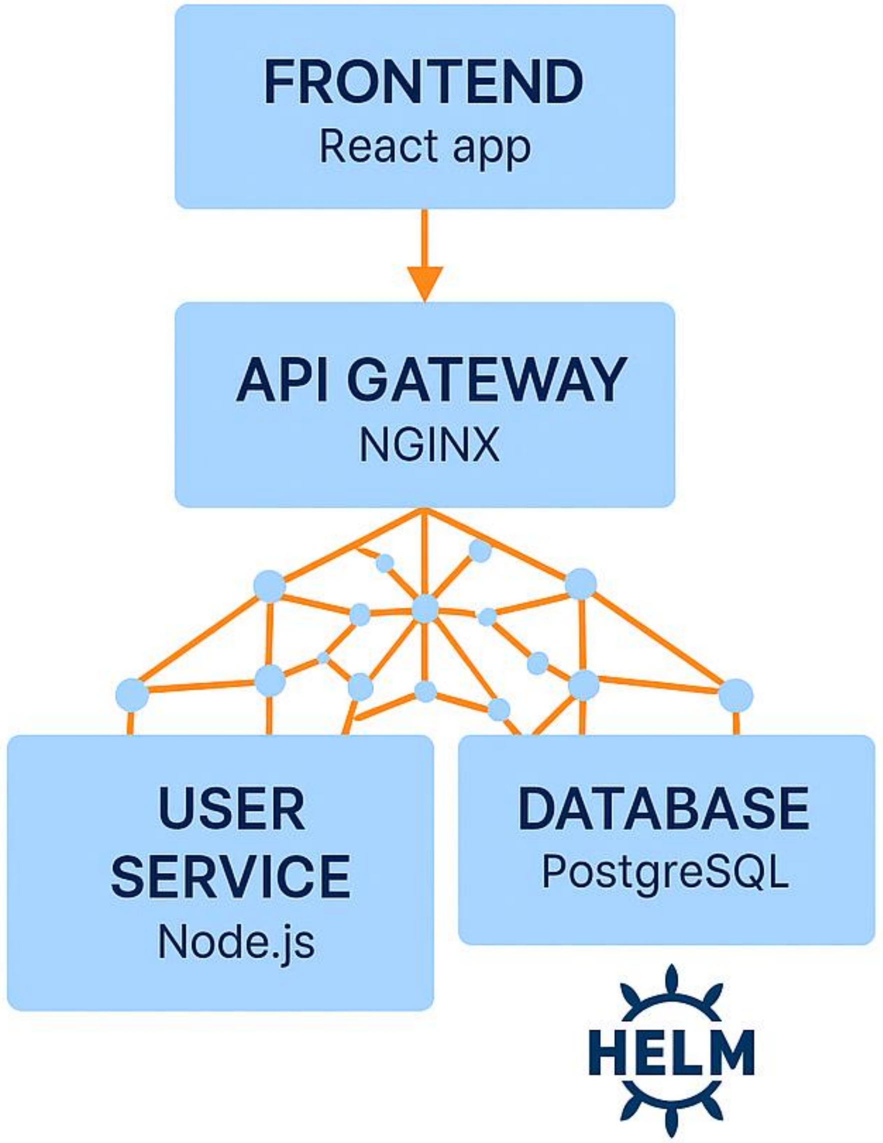
Example: Microservices Stack

SERVICES:

- **Frontend** (React app)
- **API Gateway** (NGINX)
- **User Service** (Node,js)
- **Database** (PostgreSQL)

DEPLOYMENT STRATEGY:

- Umbrella chart approach
- Shared configurations



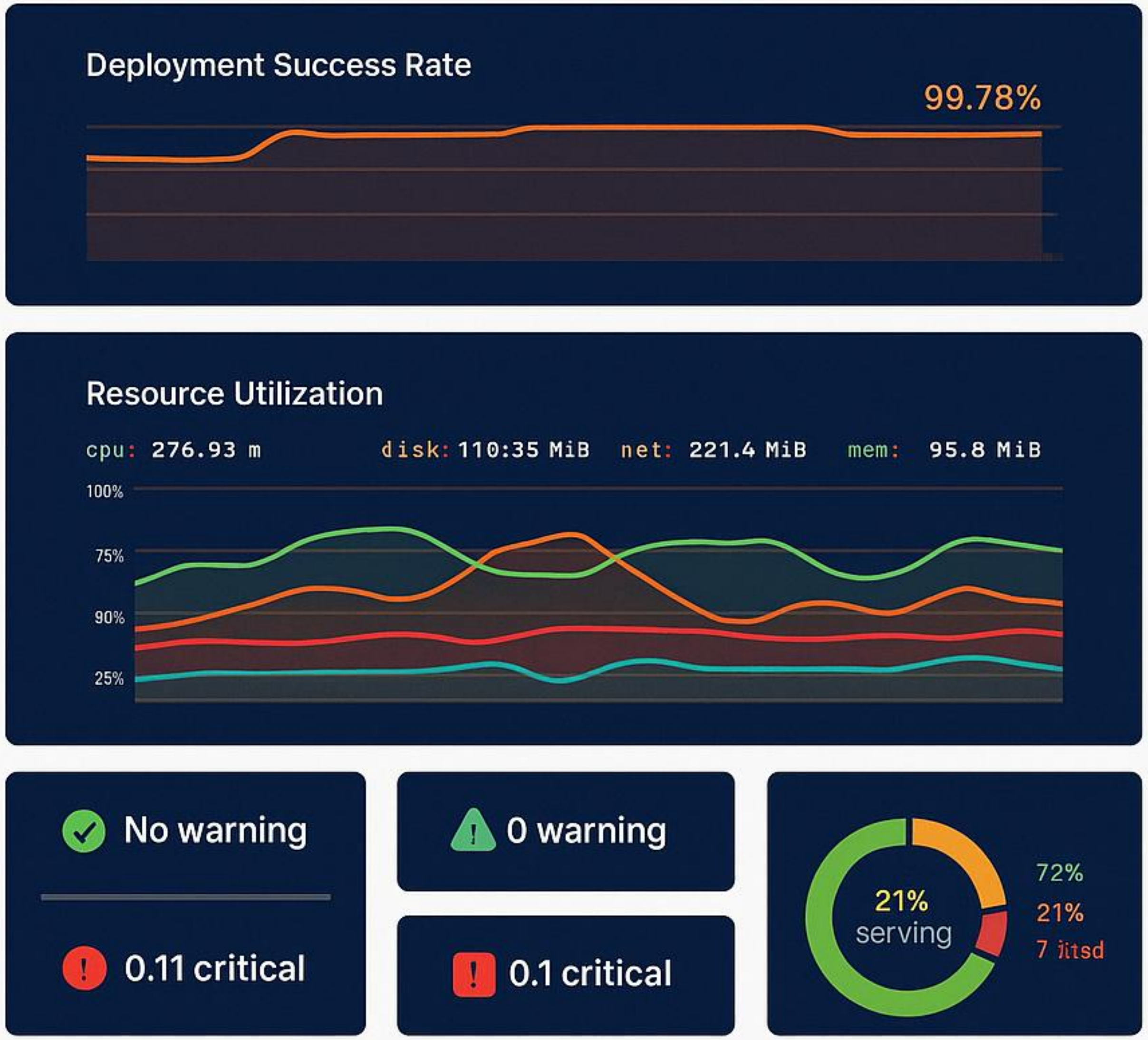
Monitoring Helm Deployments

Key Metrics:

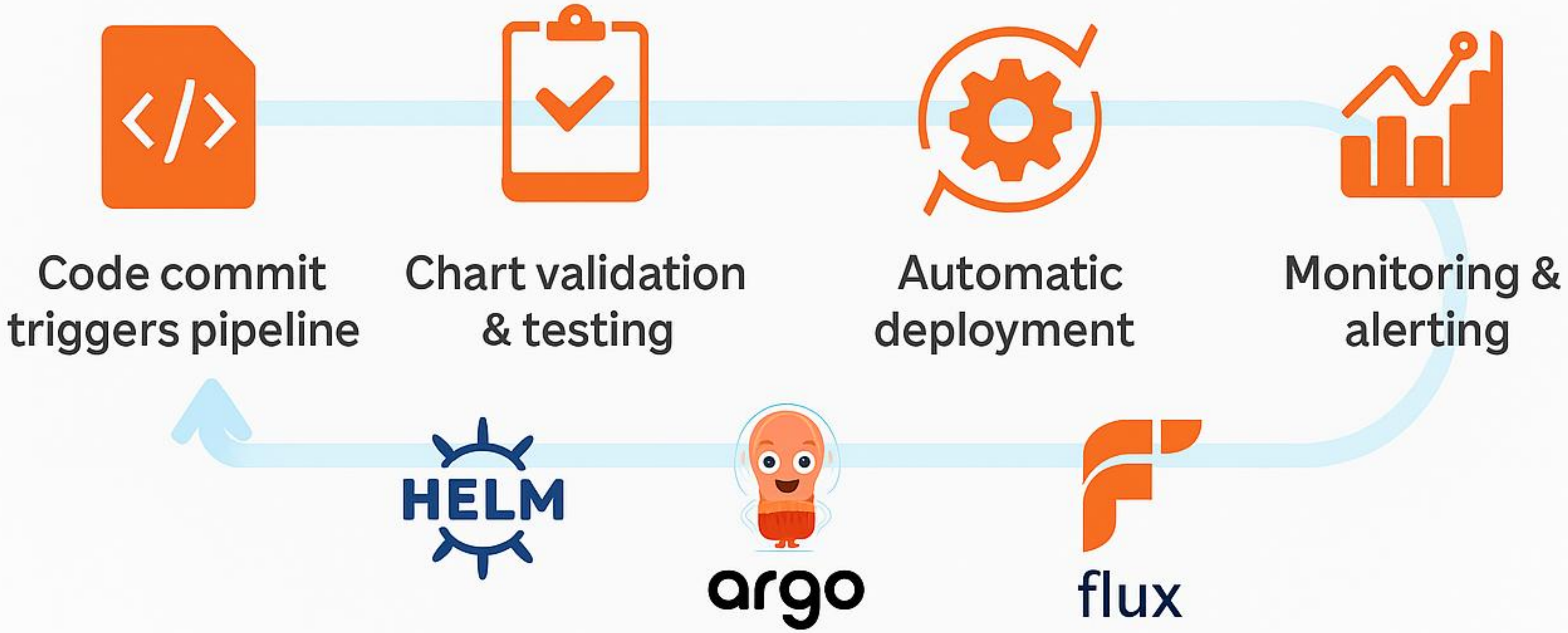
- Deployment success rate
- Release health status
- Resource utilization

Tools Integration:

- Prometheus alerts
- Grafana dashboards



GitOps + Helm Workflow

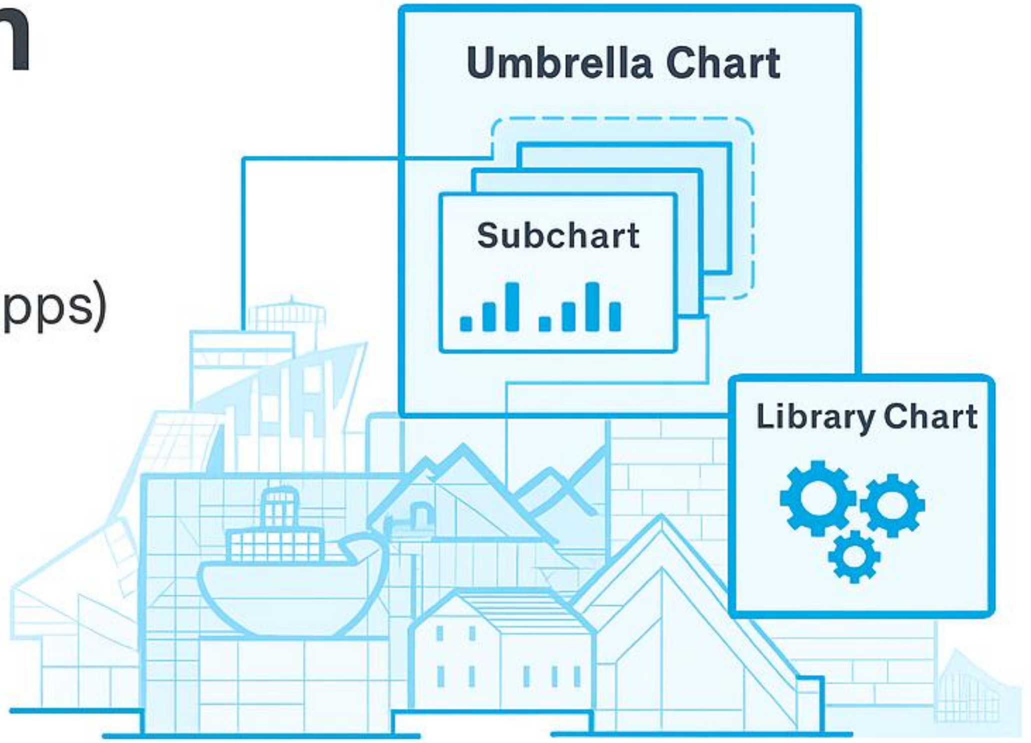


Tools: ArgoCD, **Flux, Flux**

- Declorative
- Auditable
- Rollback-friendly

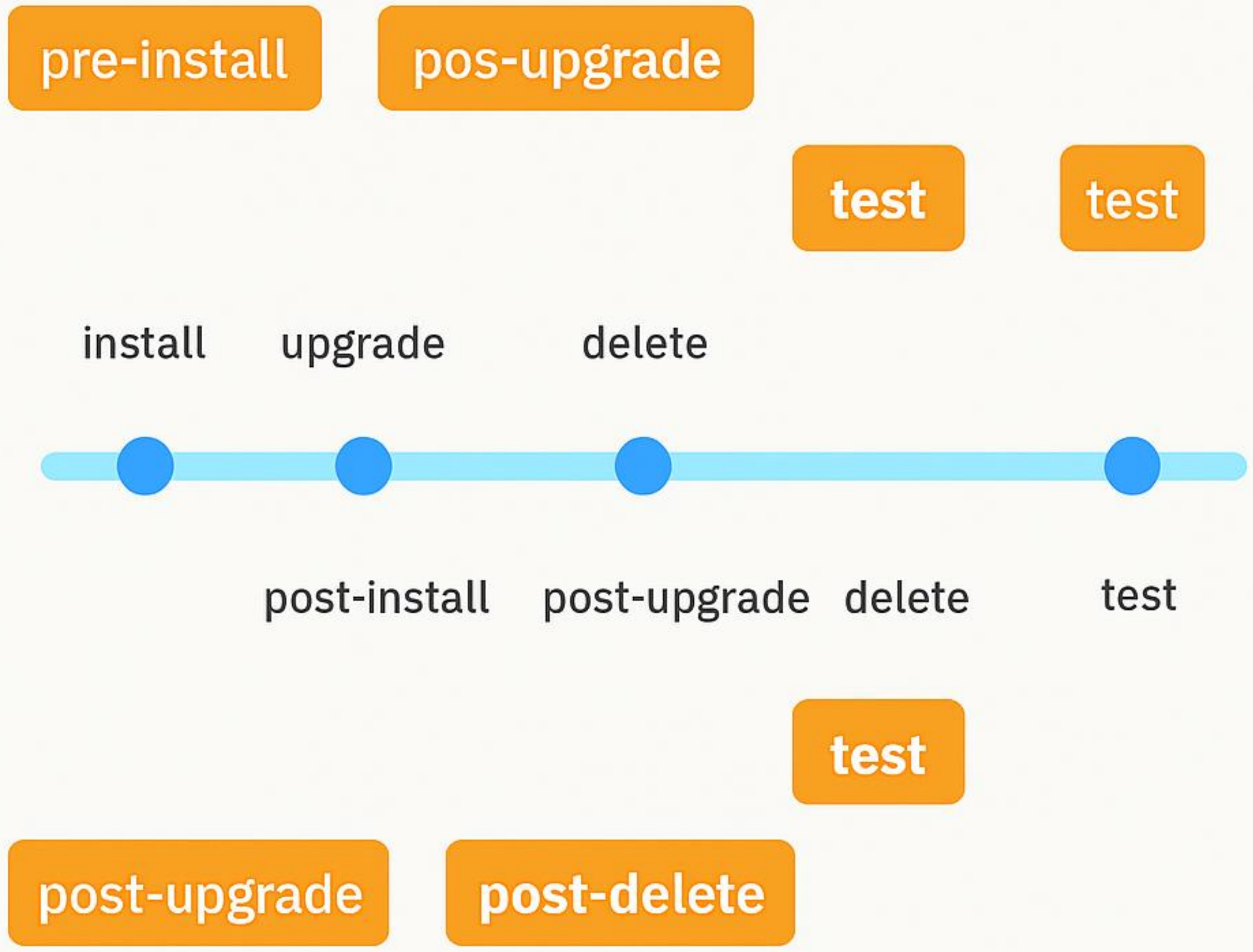
Advanced Helm Patterns

- **Umbrella Charts** (multiple apps)
- **Library Charts** (shared components)
- **Subchart Overrides**
- **Rolling Updates**



Use Cases: Complex appllications, shared services

Helm Hooks Explained



Example Use Cases

- Database migrations
- Cache warming
- Cleanup tasks

Production Readiness Checklist

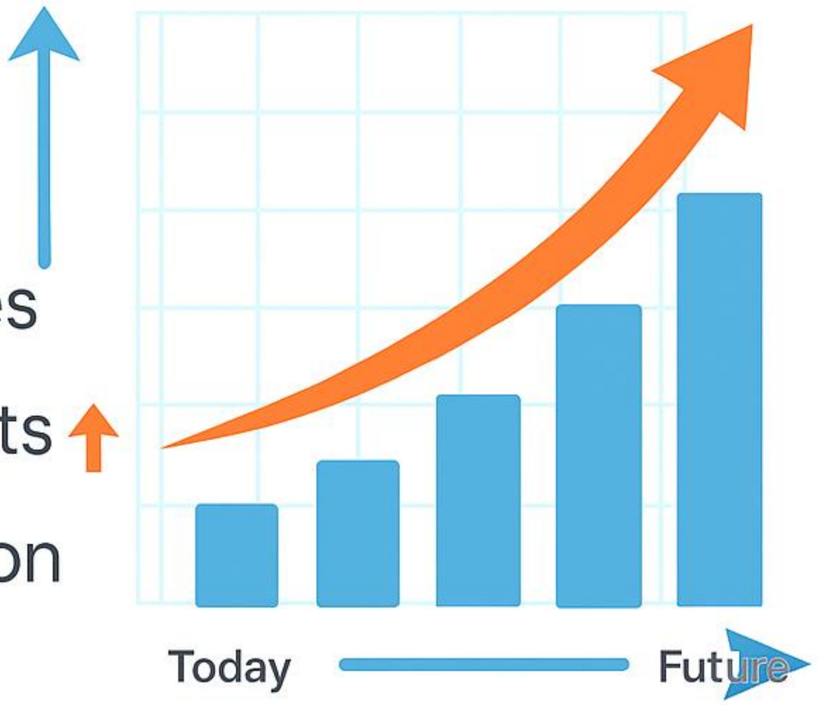
MUST-HAVE:

- ✓ Resource limits defined
- ✓ Health checks configured
- ✓ Security scanning completed
- ✓ Backup & recovery tested
- ✓ Monitoring enabled
- ✓ Documentation updated

Helm's Future & Trends

Emerging Trends:

- OCI registry adoption
- Enhanced security features
- Performance improvements
- AI-assisted chart generation



Community Growth:
50k+ stars on GitHub

Key Takeaways

Remember:

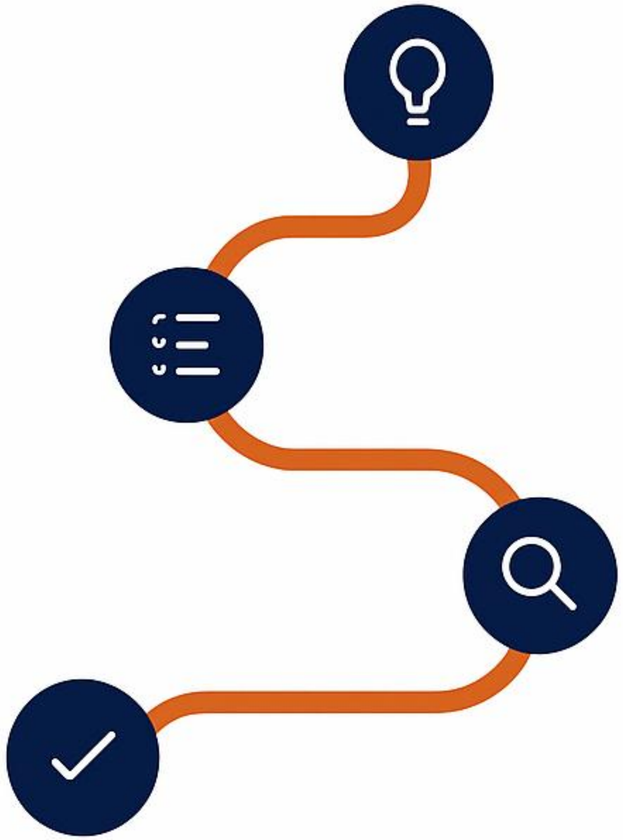
- 🎯 Start simple, iterate
- 📒 Always use version control
- 🔒 Security is paramount
- 🧪 Test before production
- 📚 Document everything

💡 **Success Metrics:** Faster deployments, fewer er

Continue Learning

Resources:

- Official Helm Documenatation
- CNCF Helm Training
- Helm Community Slack
- YouTube Tutorials



Practice Projects:

- Deploy a sample app
- Create custom charts

Salwan Mohamed

DevOps & Platform Engineer

www.linkedin.com/in/salwan-mohamed

Kubernetes • Helm • Platform Engineering

Always happy to help with your Helm journey!