acend[

Argo CD

Tools

experience knowledge

Supported Tools

- ¬Kubernetes manifests can be spcified in several formats:
  - ¬helm charts
  - ¬kustomize applications
  - ¬jsonnet files
  - ¬Plain directory of YAML/json manifests
  - ¬Any custom config management tool configured as a config management plugin
    - ¬e.g. kustomize with OpenShift Manifest support

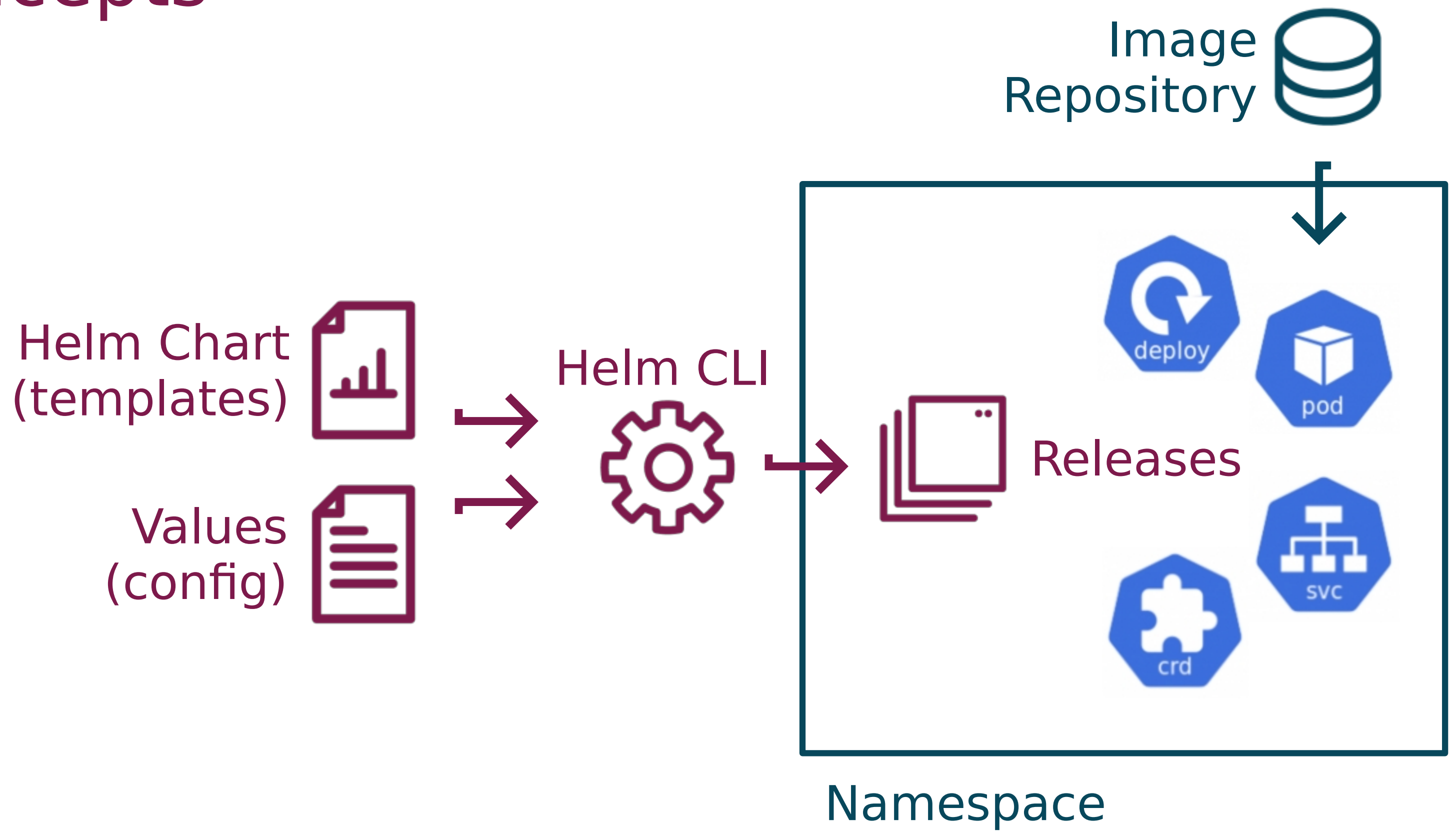
ArgoCD Tools

Helm - Package manager for Kubernetes

Argo CD Tools - Helm

Architecture and concepts

- ¬Charts
  - ¬Metadata
  - ¬Values
  - ¬Templates
- ¬Releases
- ¬Repositories



Argo CD Tools - Helm

Templates

¬No configuration duplication

¬Go template language

```
$ cat myapp/templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ include "myapp.fullname" . }}
  labels:
{{- include "myapp.labels" . | nindent 4 }}
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      {{- include "myapp.selectorLabels" \
    . | nindent 6 }}
  template:
    metadata:
      labels:
        {{- include "myapp.selectorLabels"\
        . | nindent 8 }}
    spec:
    [...]
```

Basic concepts

Values

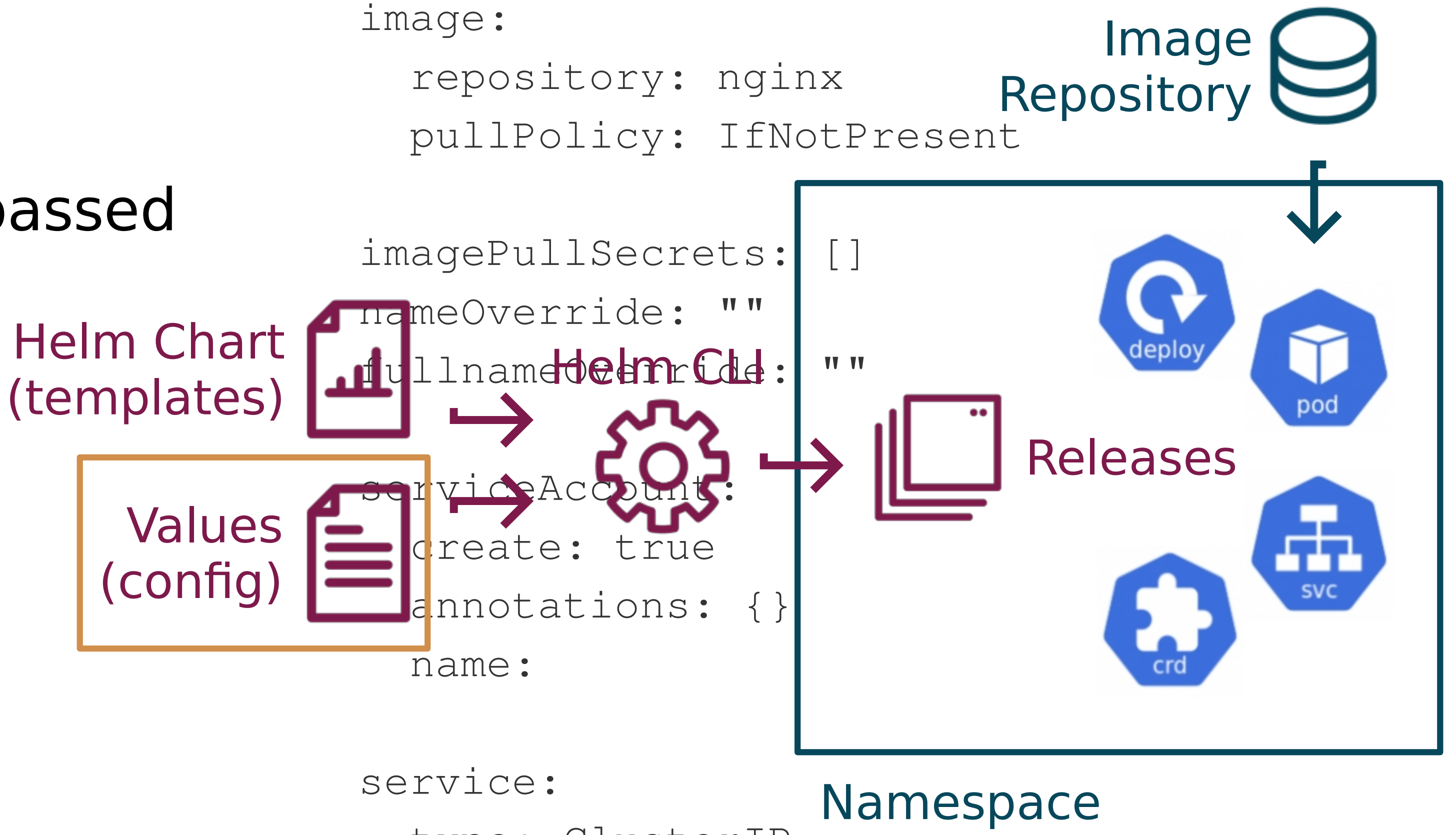
- Declare variables to be passed into your templates
- Default values
- YAML-formatted

```
$ cat myapp/values.yaml
replicaCount: 1
image:
  repository: nginx
  pullPolicy: IfNotPresent

imagePullSecrets: []
nameOverride: ""
fullnameOverride: ""

serviceAccount:
  create: true
  annotations: {}
  name:

service:
  type: ClusterIP
  port: 80
```



Specifing the values

Using a values file

```
$ argocd app set <app> --values values-production.yaml
```

Using individual key-value pair

```
$ argocd app set <app> --parameter replicaCount=2
```

Parameters are a GitOps Anti Pattern!

ArgoCD Tools

Kustomize

Alternatives

Kustomize.io

Kubernetes native configuration management

¬Template-free customization

¬No new language to learn

¬Everything is plain YAML

¬Integrated into kubectl

```
mykustomize/
├── base
│   ├── deployment.yaml
│   └── kustomization.yaml
└── overlays
    ├── production
    │   ├── kustomization.yaml
    │   └── replica_count.yaml
    └── staging
        ├── kustomization.yaml
        └── replica_count.yaml
```

Alternatives

kustomize.io

When to use Kustomize

¬Lots of duplicate configuration code across environments

¬Hate using templating engines

¬Don’t want to cover all possible use cases/parameters

¬As post-rendering tool in Helm
(https://helm.sh/docs/topics/advanced/#post-rendering)

Specifing the overlay

¬Path must point to the overlay

```
$ argocd app create <app> --path
'kustomize-app/overlays/production' ...
```

```
kustomize-app/
├── base/
├── overlays/
│   ├── acceptance/
│   ├── development/
│   ├── production/
```

Specifing the parameters

¬Use the parameter option

```
$ argocd app set <app> --parameter namePrefix=<namePrefix>
```

Parameters are a GitOps Anti Pattern!

ArgoCD Tools

Custom Tools - Plugins

Supported Tools

- Argo CD allows to integrate more config management tools to generate K8S Resources
  - Tool must be available in the argocd-repo-server pod
  - Registered as new plugin in the argocd-cm Configmap

Argo CD Tools

Supported Tools

```
data:
  configManagementPlugins: |
    - name: pluginName
      init:                          # Optional command to
initialize application source directory
        command: ["sample command"]
        args: ["sample args"]
      generate:                      # Command to generate
manifests YAML
        command: ["sample command"]
        args: ["sample args"]
```

Argo CD Tools - Helm

Use the custom tool

```
$ argocd app create <app>
    --config-management-plugin <pluginName>
```

THX
experience knowledge