


# Kubernetes Application Deployment Wizard

- Step 0: Login and Create/Load Application
  - Login
  - Load Existing Application
- Step 1: Define your New Application
- Step 2: Define Application Services/Components
- Step 3: Assign Ingress Policies to Services/Components
- Step 4: Define Volumes (if needed)
  - Secrets
  - Persistent Volume Claims
  - Configuration
- Step 5: Application Performance
- Step 6: Application Health
- Step 7: Container Specifications
- Step 8: Cost and Performance Optimization
- Step 9: Visualize and Submit
  - Passing Validation
  - Errors and/or Warnings

## Step 0: Login and Create/Load Application

### Login

 Kruise Application Deployment Wizard

---

Username

Password

Login

### Load Existing Application



## Kruise Application Deployment Wizard

Create a New Application

Import an Existing Application

Deployment Git Repo URL


Deployment Git Path

Deployment Git Revision


Import

Once the import button is pressed the tool will import the YAML files from the repository. If the import is successful the Wizard will start from the first step as in the case of a new application but data will be already loaded from the import. The user can apply any changes and these will be saved in the repository as a new commit.

## Step 1: Define your New Application



Kruise Application Deployment Wizard



Application DetailsServiceIngressVolumesPerformanceHealthContainerOptimizeSubmit

Create a New ApplicationImport an Existing Application

Application Name

Team Org

Target Environment

Dev | Stage | Prod

Deployment Git Repo URL

Target Region

STL | KCI | BEL


Deployment Git Path

Target Namespace


Deployment Git Revision

>

## Step 2: Define Application Services/Components



Kruise Application Deployment Wizard



Application DetailsServiceIngressVolumesPerformanceHealthContainerOptimizeSubmit

Service/Component Name

Type

ExternalName | ClusterIP | LoadBalancer

Name

Port

Target Port

Name

Port


Target Port

<


>

### Step 3: Assign Ingress Policies to Services/Components






Kruise Application Deployment Wizard



Application DetailsServiceIngressVolumesPerformanceHealthContainerOptimizeSubmit



Service/Component Name

Available List of Components


Volume Name


Mount Path

Sub Path

Type

secret





Secret Name

Key




Value


Key


Value


Key


Value
















## Persistent Volume Claims




## Kruise Application Deployment Wizard



Application Details
Service
Ingress
Volumes
Performance
Health
Container
Optimize
Submit

Volume Name
▼

<p><b>Service/Component Name</b></p> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 10px;">Available List of Components <span style="float: right;">✓</span></div> <p>Volume Name</p> <input style="width: 90%;" type="text"/> <p>Mount Path</p> <input style="width: 90%;" type="text"/> <p>Sub Path</p> <input style="width: 90%;" type="text"/> <p>Type</p> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 10px;">persistent volume <span style="float: right;">✓</span></div> <div style="text-align: center;"></div>	<p><b>Claim Name</b></p> <div style="border: 1px solid #ccc; height: 30px; width: 90%;"></div> <p>Access Type</p> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 10px;">RW Exclusive to App   RO any Apps   RW any Apps <span style="float: right;">✓</span></div> <p>Size (in GB)</p> <input style="width: 90%;" type="text"/> <p>Storage Class</p> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 10px;">host (single AZ)   nfs (multi AZ) <span style="float: right;">✓</span></div>
--	---

⏪
<
>
⏩

Application Details Service Ingress Volumes Performance Health Container Optimize Submit

Volume Name

ApplicationSecrets

Service/Component Name

Available List of Components

Volume Name

Mount Path

Sub Path

Type

persistent volume

Claim Name

Access Type


RW Exclusive to App | RO any Apps | RW any Apps

Size (in GB)


Storage Class

host (single AZ) | nfs (multi AZ)

## Configuration



# Kruise Application Deployment Wizard



Application Details

Service

Ingress

Volumes


Performance

Health

Container

Optimize

Submit



Volume Name

ApplicationSecrets

Volume Name

ApplicationPersistentVolumeClaim

Service/Component Name

Available List of Components


Volume Name

Mount Path

Sub Path


Type



config



Config Map Name


Config Content






## Step 5: Application Performance





## Kruise Application Deployment Wizard



Application Details

Service

Ingress

Volumes


Performance

Health

Container

Optimize

Submit



Service/Component Name


Available List of Components

Number of Instances

1 (non-HA), 3, 6, 9, 12, 15 ... 3\*N

Upgrade Strategy

Rolling Update (HA) | Re-Create (Non-HA)



Requests

Memory

16MB, 32MB, 64MB, 128MB, 256MB ... 64GB

CPU

0.1, 0.25, 0.5, 1, 2, 4, 6

Ephemeral Disk

100MB, 250MB, 500MB, 1GB, 2GB .... 1TB

Limits

Memory



16MB, 32MB, 64MB, 128MB, 256MB ... 64GB

CPU


0.1, 0.25, 0.5, 1, 2, 4, 6

Ephemeral Disk


100MB, 250MB, 500MB, 1GB, 2GB .... 1TB



## Step 6: Application Health



## Kruise Application Deployment Wizard



Application Details

Service

Ingress

Volumes


Performance

Health

Container

Optimize

Submit



Service/Component Name


Available List of Components


Type

Liveness

Endpoint

Port






Liveness HTTP Headers


Key


Value




Key

Value









Readiness HTTP Headers

Key

Value




Key

Value






## Step 7: Container Specifications



# Kruise Application Deployment Wizard



Application Details

Service

Ingress

Volumes

Performance

Health

Container

Optimize

Submit

Service/Component Name

Available List of Components

Image Location

Image Version

Image Pull Policy

Always | IfNotPresent

Command

Additional Ports

Name

Port

Protocol

TCP | UDP

Container Args

Arg

Arg

Arg

Environment Variables

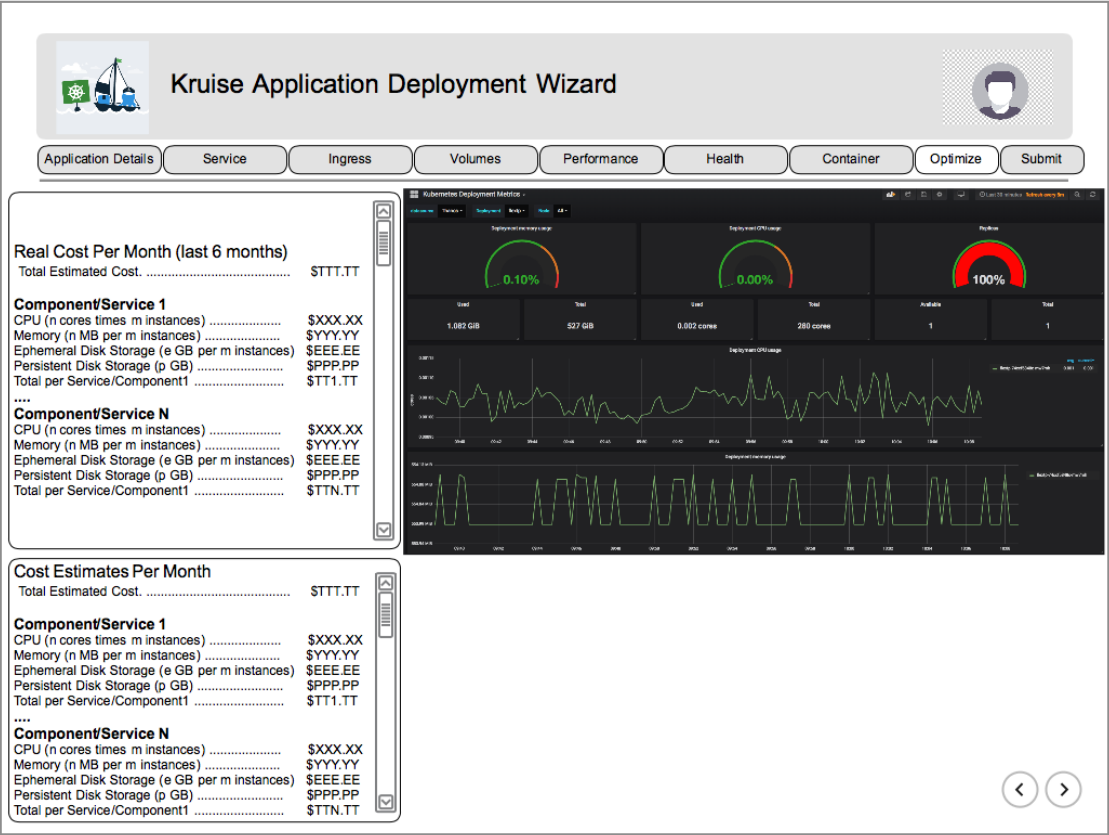
Key

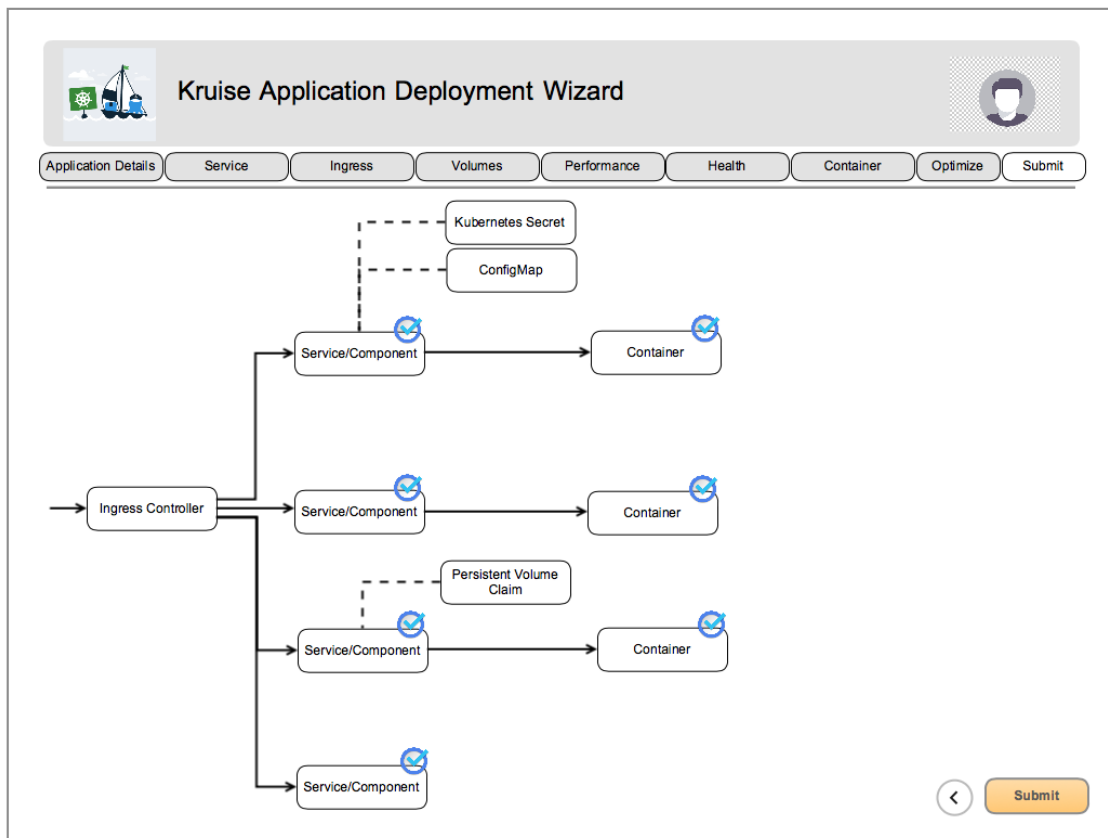
Value

Key

Value

## Step 8: Cost and Performance Optimization





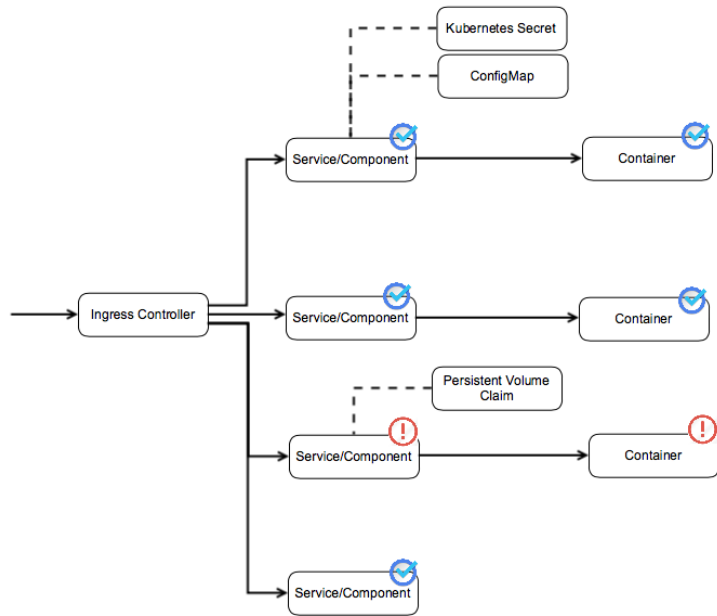
## Errors and/or Warnings



# Kruise Application Deployment Wizard



- Application Details
- Service
- Ingress
- Volumes
- Performance
- Health
- Container
- Optimize
- Submit



< Submit