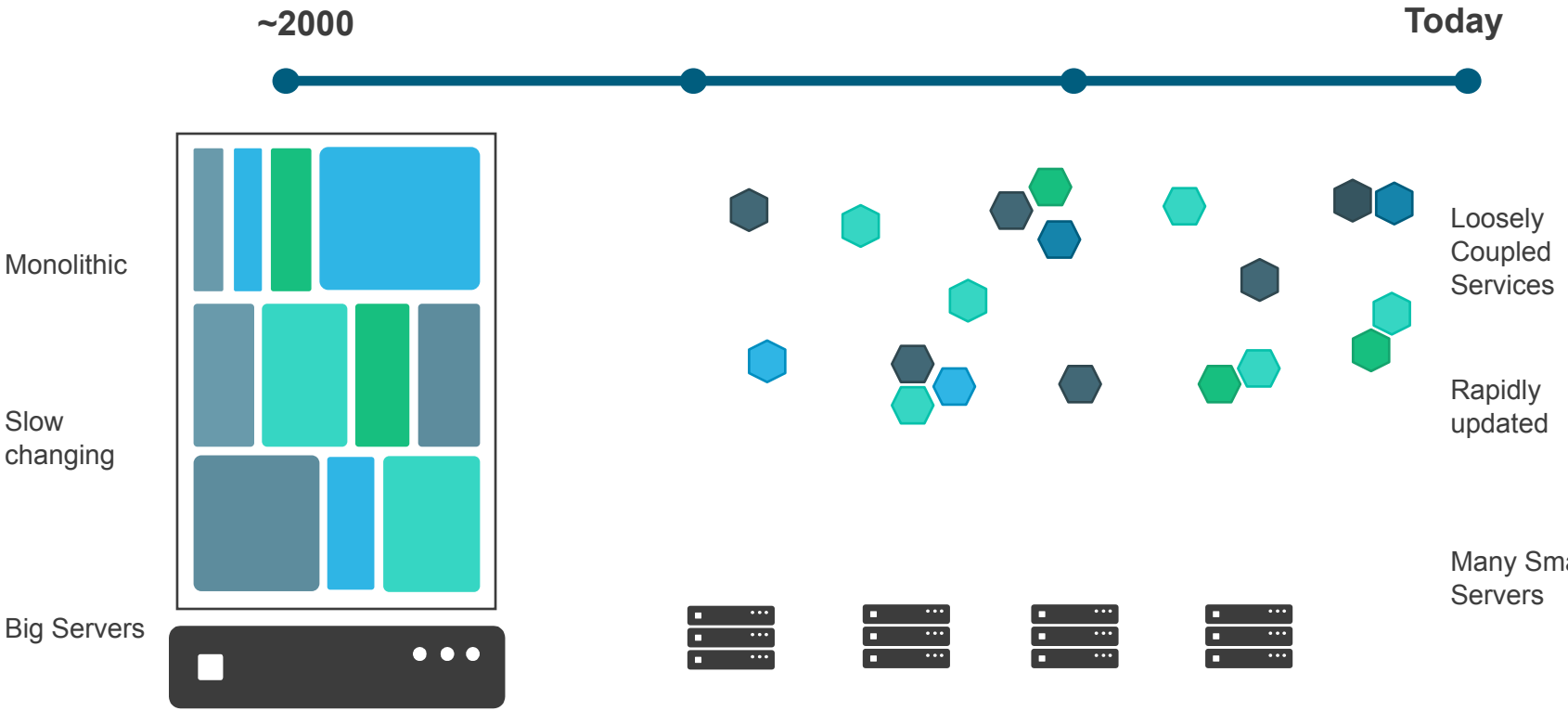
Kubernetes 101

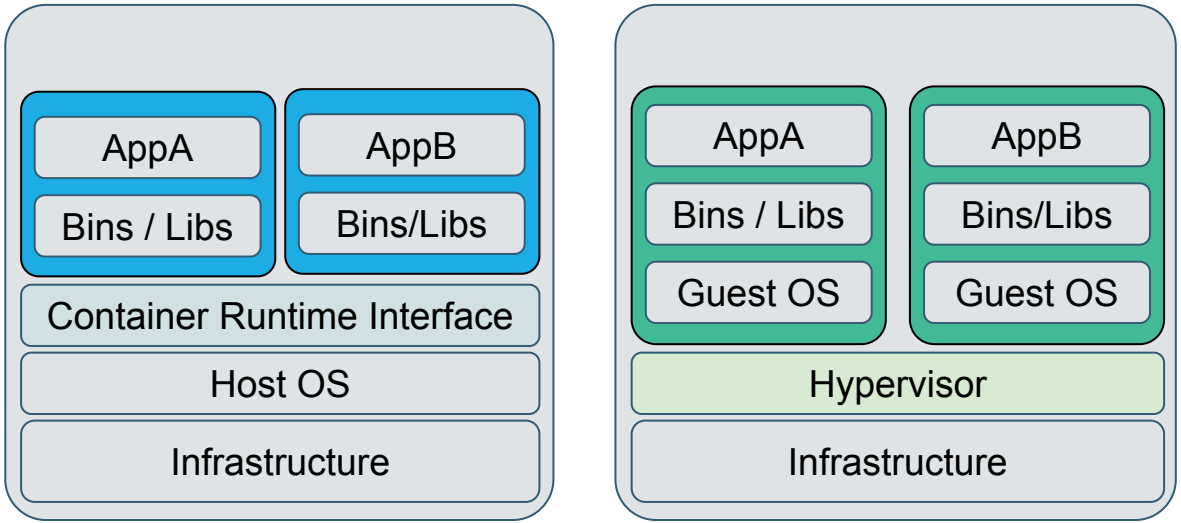
@vincentdesmet

Applications evolve



Reference: http://docker.com

Containers / VMs



Docker containers

Container

- Packages up software binaries and dependencies
- Isolates software from each other
- Container is a standard format
- Easily portable across environment
- Allows ecosystem to develop around its standard

Reference: http://docker.com

Docker Concepts

Docker Image

The basis of a Docker container

Docker Container

The standard unit in which the application service resides

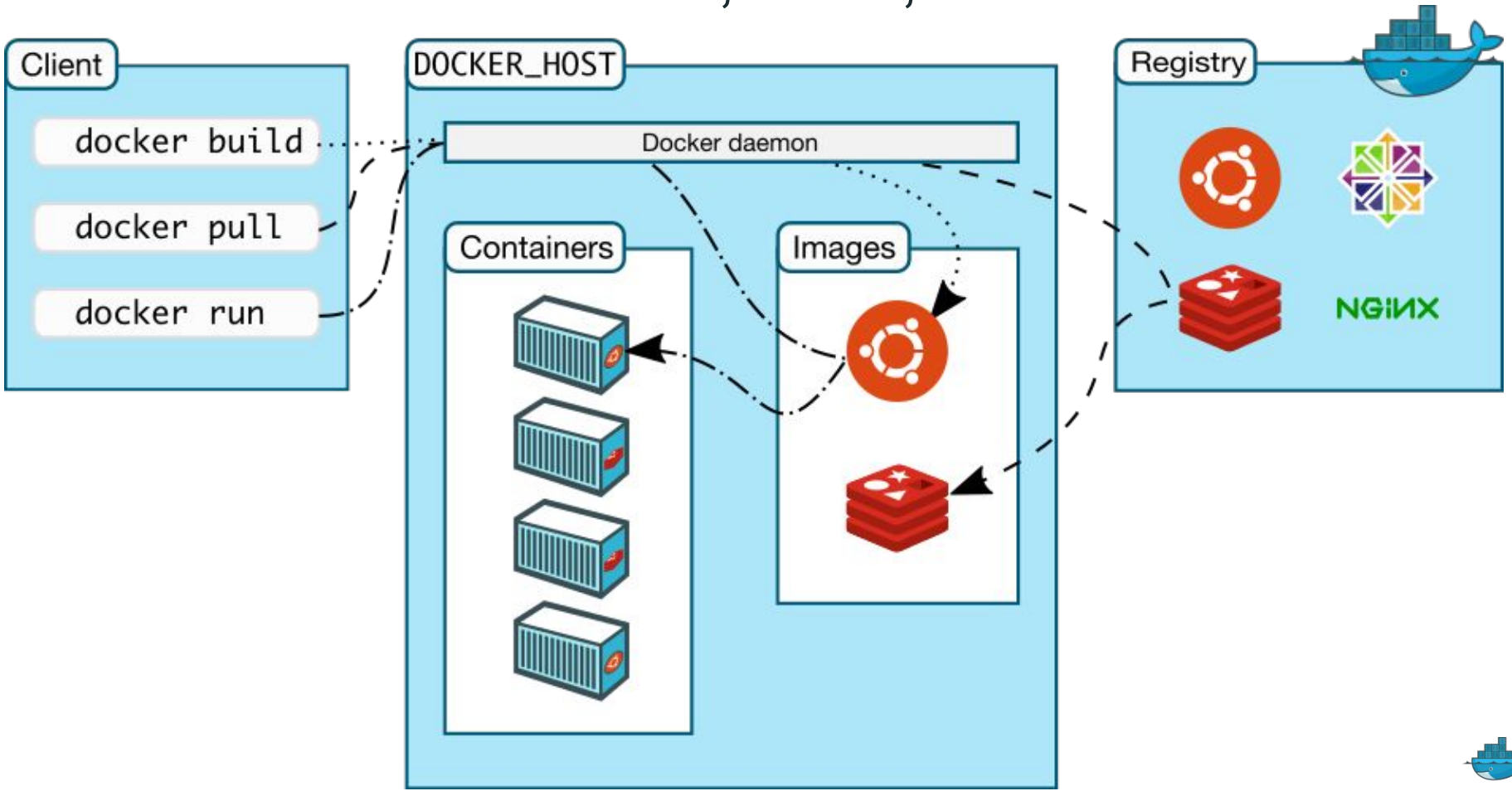
Docker Engine

Creates, ships and runs Docker containers deployable on physical or virtual host locally, in a datacenter or cloud service provider

Docker Registry

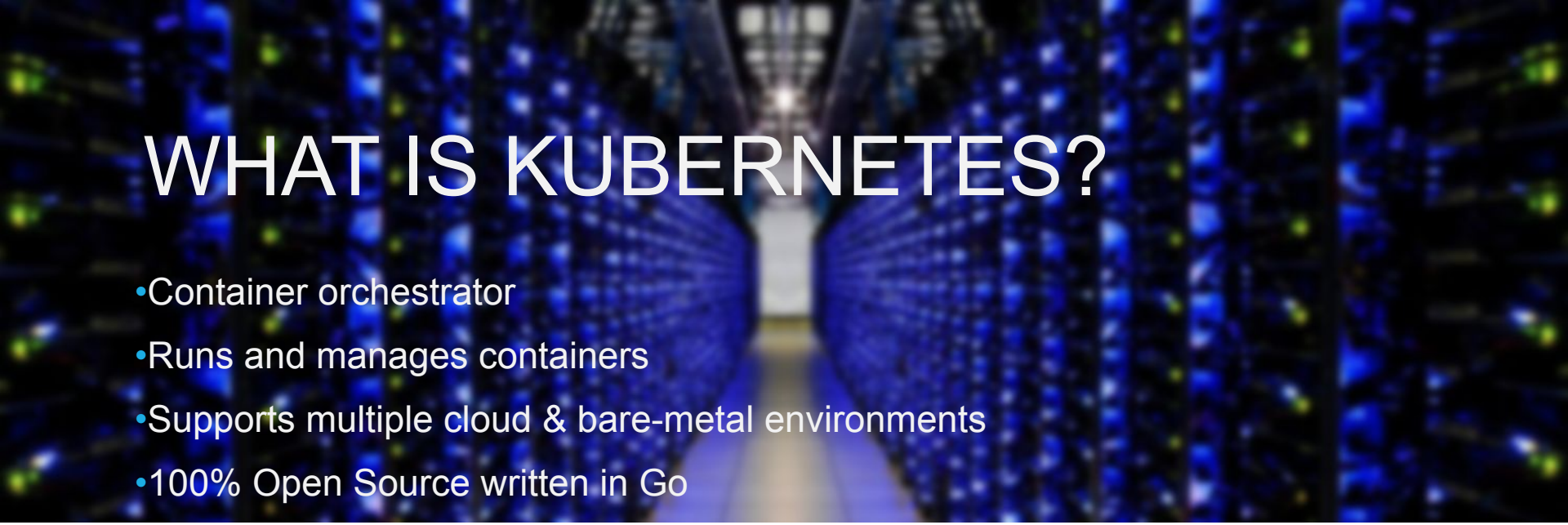
On-premises registry for image storing and collaboration

Docker Tools: BUILD, SHIP, RUN



KUBERNETES

"κυβερνήτης" (kubernetes) is Greek for "pilot" or "helmsman of ship"



WHAT IS KUBERNETES?

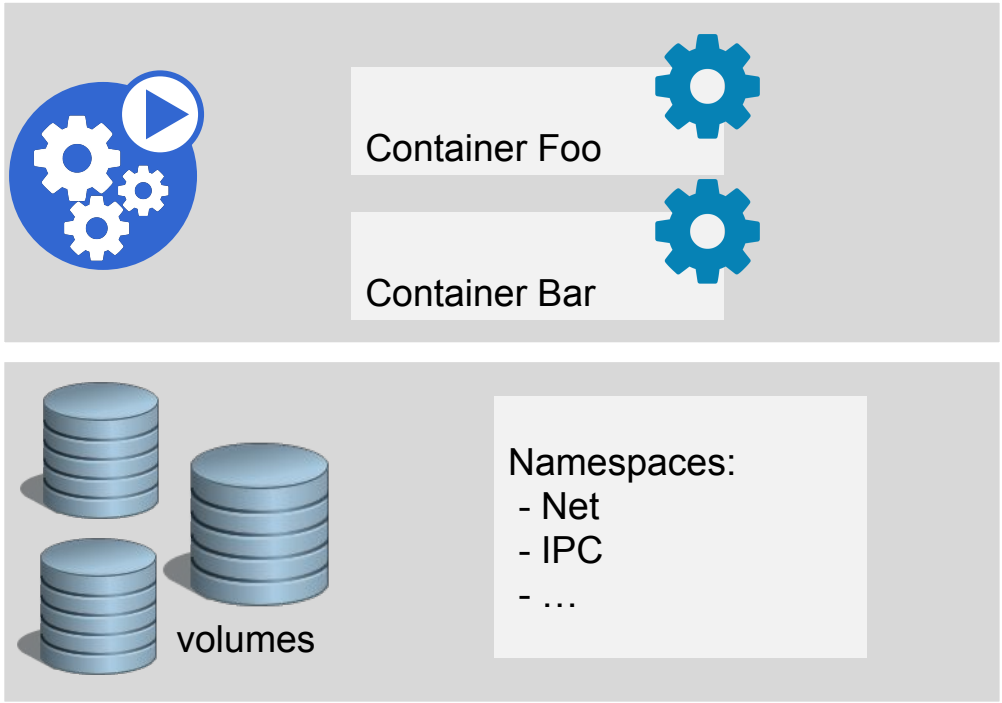
- Container orchestrator
- Runs and manages containers
- Supports multiple cloud & bare-metal environments
- 100% Open Source written in Go
- Built on decades of experience of running containers at Google
- First project hosted by CNCF (Accepted on Mar. 11 2016)

Container packaged

Dynamically scheduled

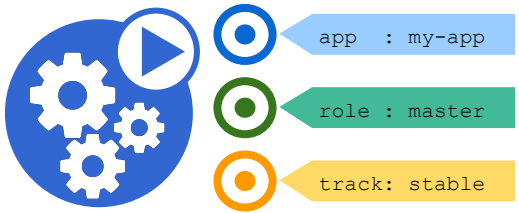
(Micro) Service oriented

PODS & VOLUMES

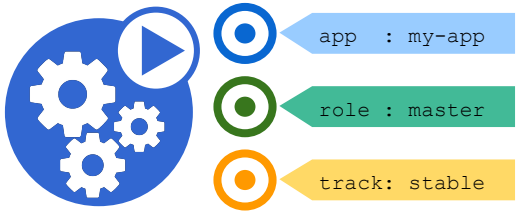


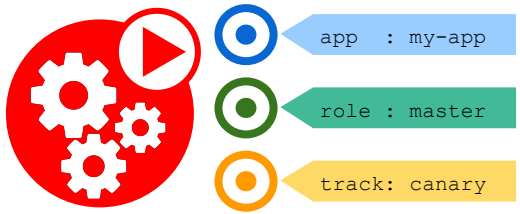
PODS & VOLUMES

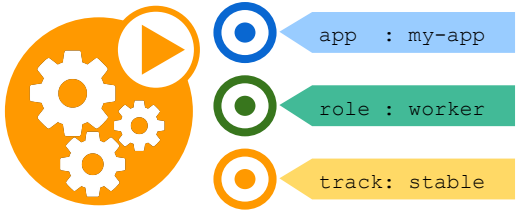
LABELS

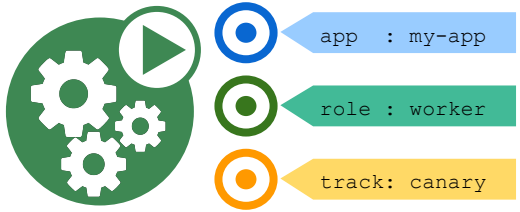


LABELS

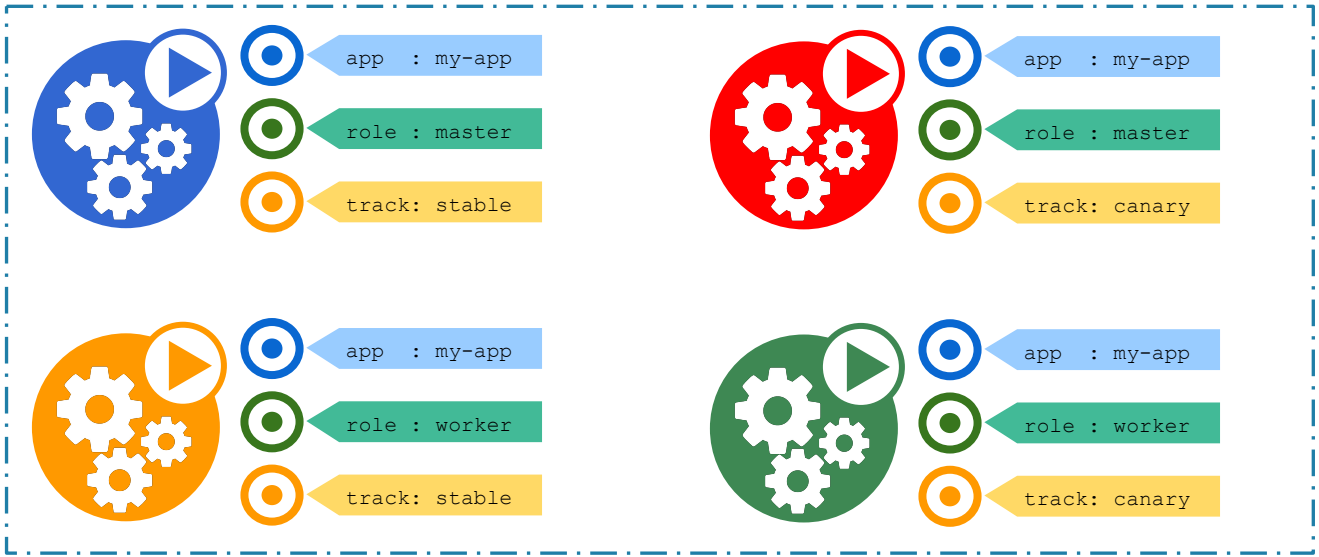

app : my-app
role : master
track: stable


app : my-app
role : master
track: canary


app : my-app
role : worker
track: stable

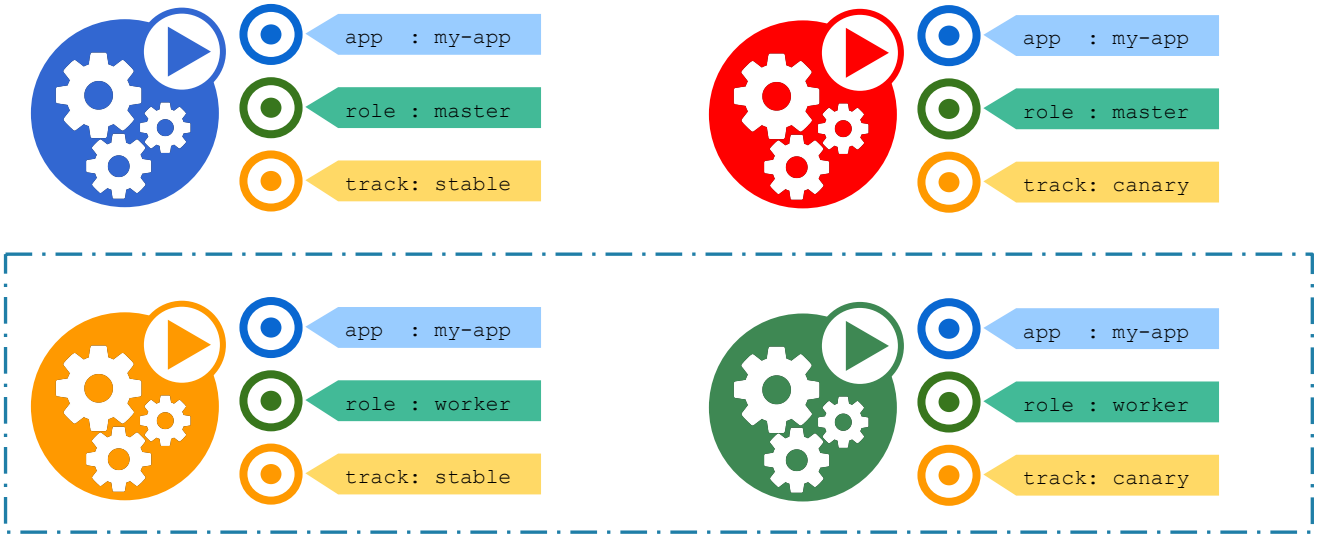

app : my-app
role : worker
track: canary

SELECTORS



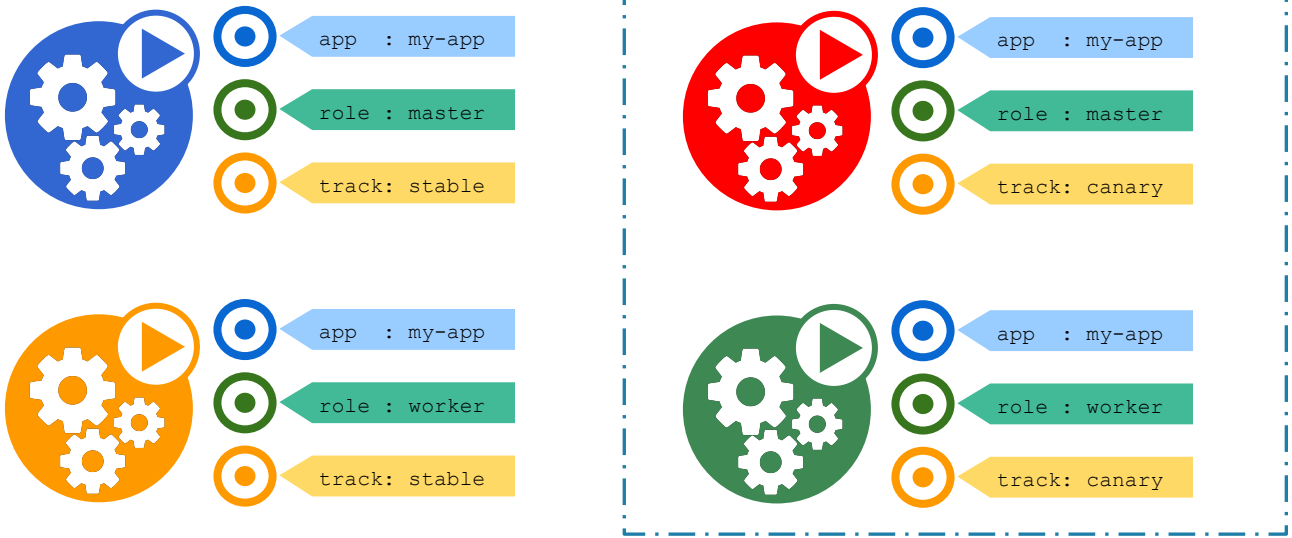

```
app  :  my-app
```

SELECTORS




```
app  :  my-app
role :  worker
```

SELECTORS

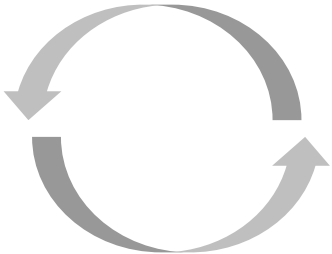



```
app   :  my-app
track:  canary
```

CONTROL LOOPS

Desired state

```
pods:
 - foo
 - bar
```



Actual state

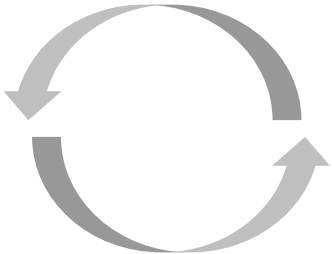
```
pods:
 - foo
```

CONTROL LOOPS

Create

Desired state

```
pods:
 - foo
 - bar
```



```
create
“bar”
```

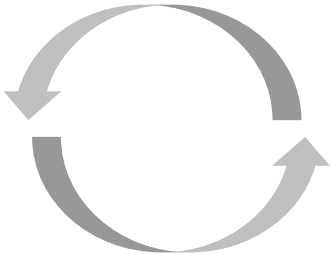
Actual state

```
pods:
 - foo
```

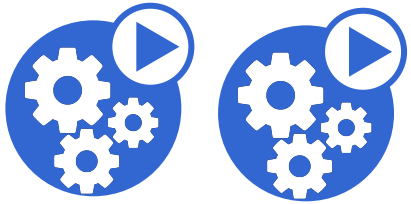
CONTROL LOOPS

Desired state

```
pods:
 - foo
 - bar
```



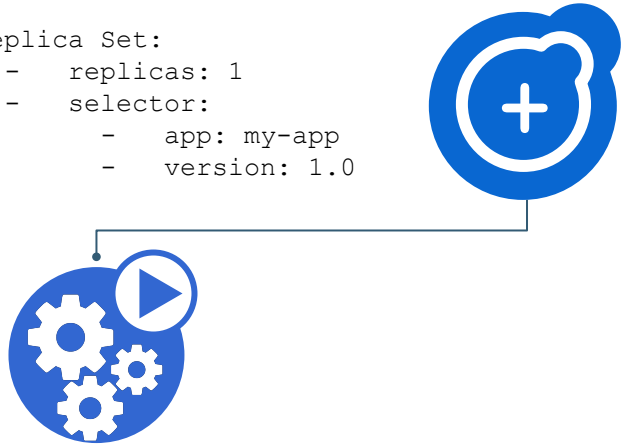
Actual state



```
pods:
 - foo
 - bar
```

REPLICA SETS

```
Replica Set:
  -   replicas: 1
  -   selector:
        -   app: my-app
        -   version: 1.0
```

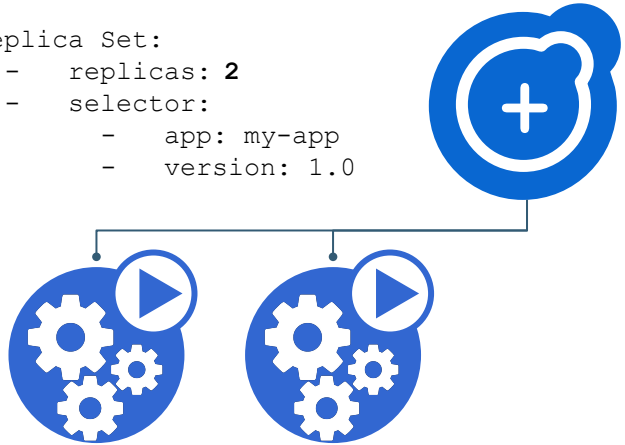


- Desired State:

  - Pod Specification
  - Replica Count
  - Label Selector

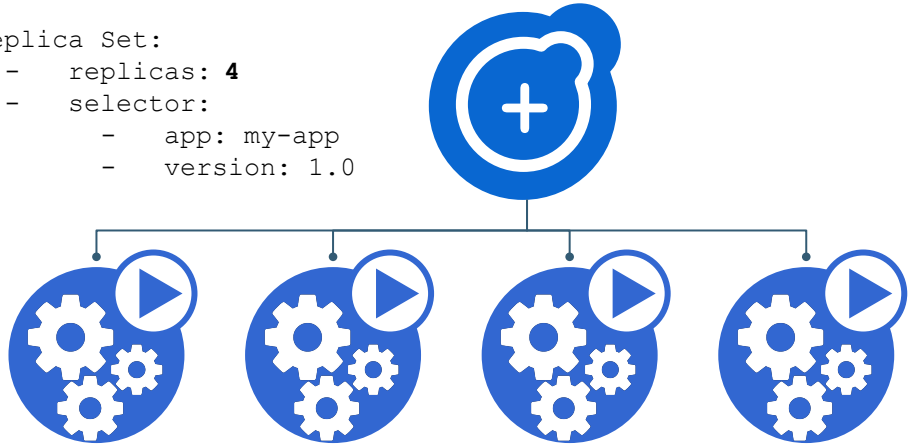
REPLICA SETS

```
Replica Set:
  -   replicas: 2
  -   selector:
        -   app: my-app
        -   version: 1.0
```



REPLICA SETS

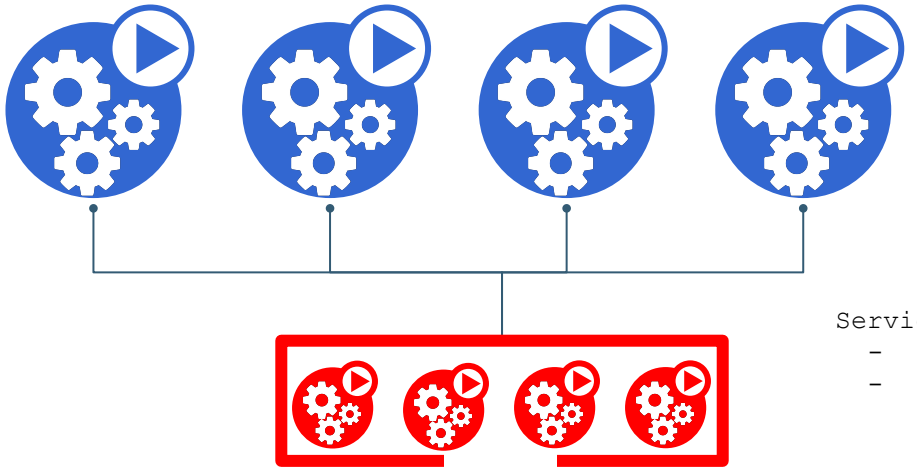
```
Replica Set:
  -   replicas: 4
  -   selector:
        -   app: my-app
        -   version: 1.0
```



SERVICES

- de-couple discovery from application
  - Define how to access pods
  - Act as a proxy (Virtual IP - stable for DNS)

Think of:
- Dynamic Routing Table



```
Service:
  -   nodePort: 30128
  -   selector:
        -   app: my-app
        -   version: 1.0
```

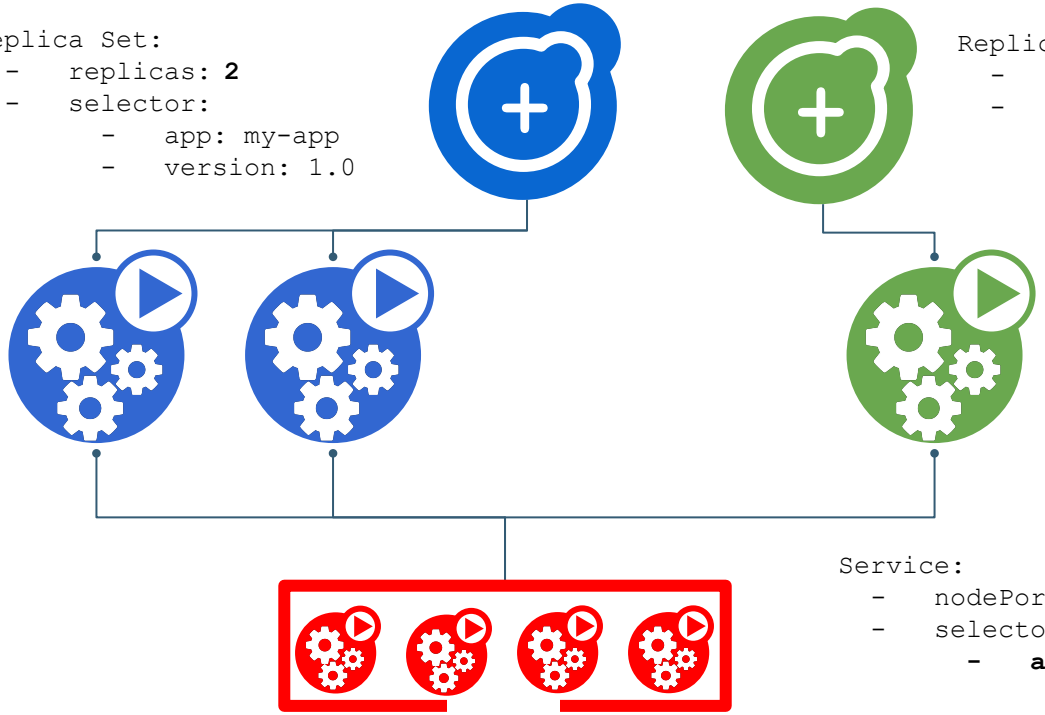
CANARIES

Think of:

- Partially live the new version

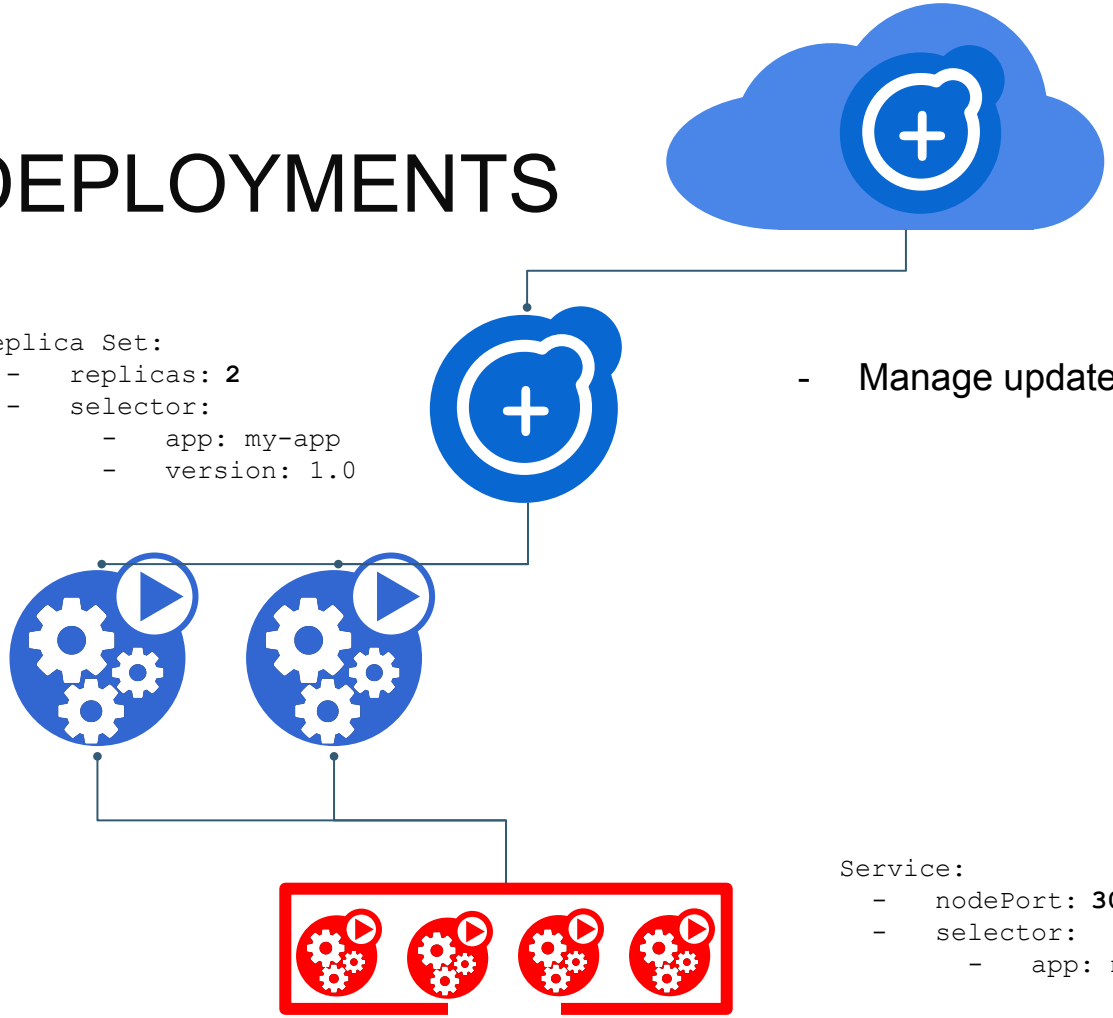
```
Replica Set:
  -   replicas: 2
  -   selector:
        -   app: my-app
        -   version: 1.0
```

```
Replica Set:
  -   replicas: 1
  -   selector:
        -   app: my-app
        -   version: canary
```



```
Service:
  -   nodePort: 30128
  -   selector:
        -   app: my-app
```

DEPLOYMENTS

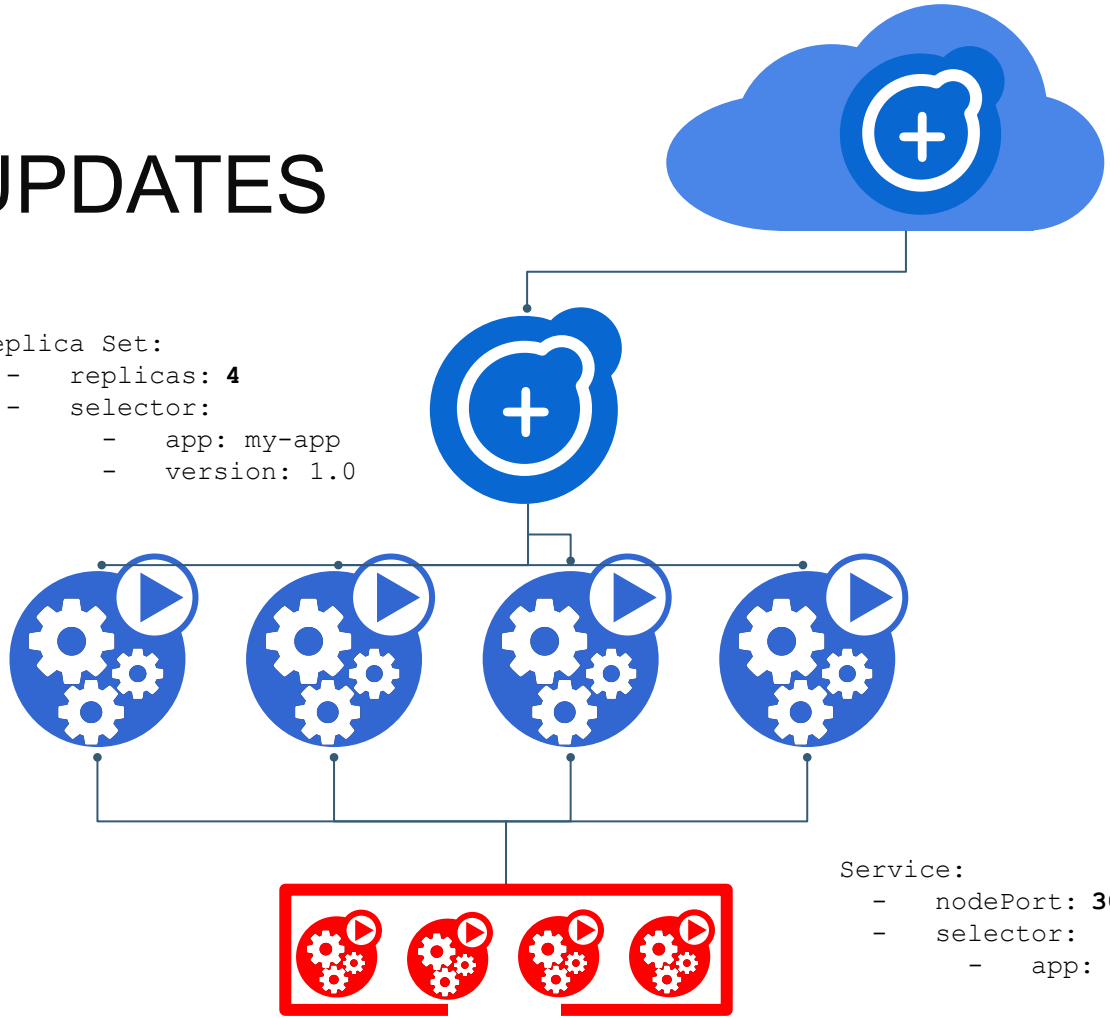


```
Replica Set:
  -   replicas: 2
  -   selector:
        -   app: my-app
        -   version: 1.0
```

- Manage updates with Deployment resources

```
Service:
  -   nodePort: 30128
  -   selector:
        -   app: my-app
```

UPDATES

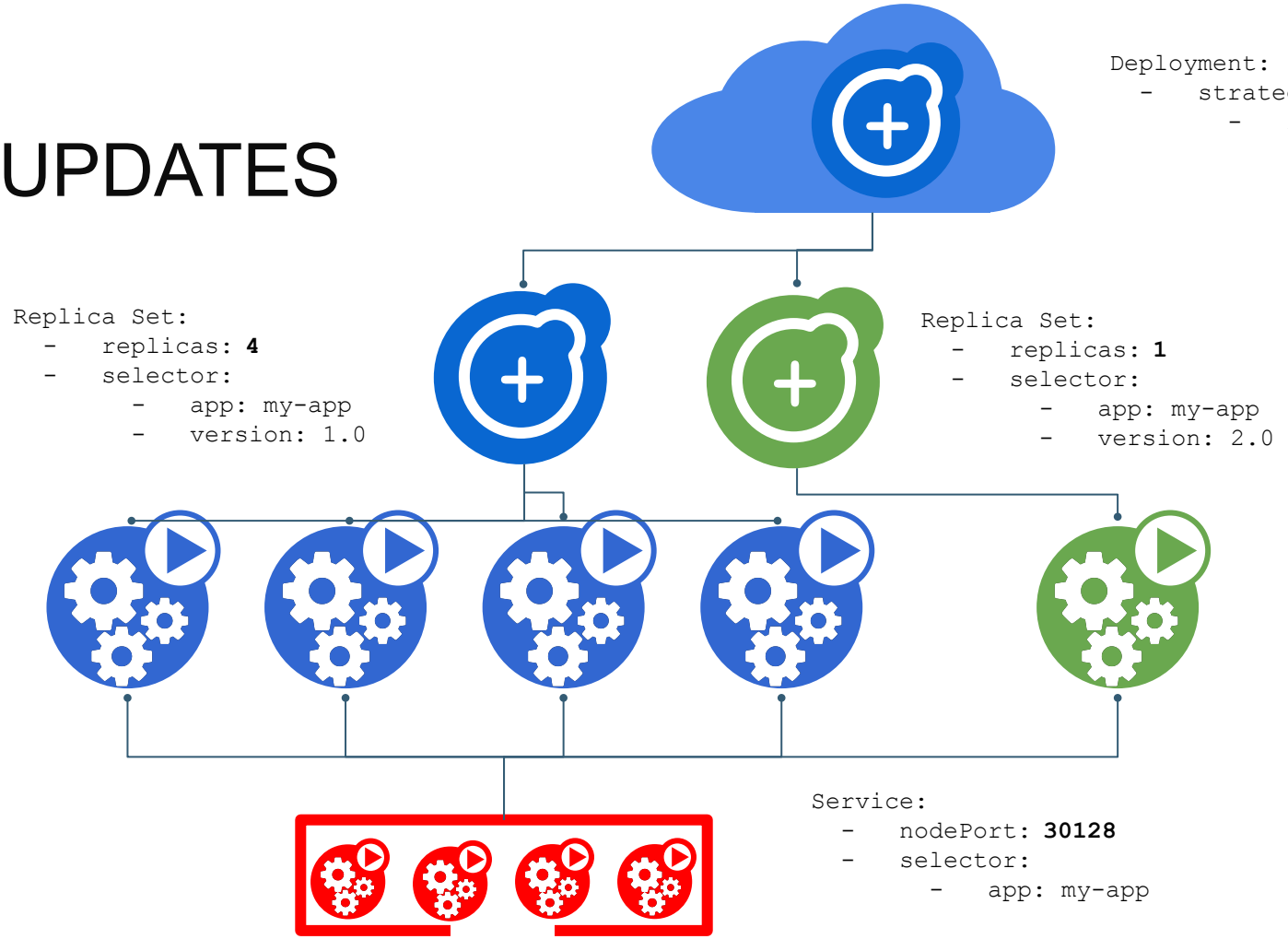


```
Deployment:
  -   strategy:
        -   rollingUpdate
```

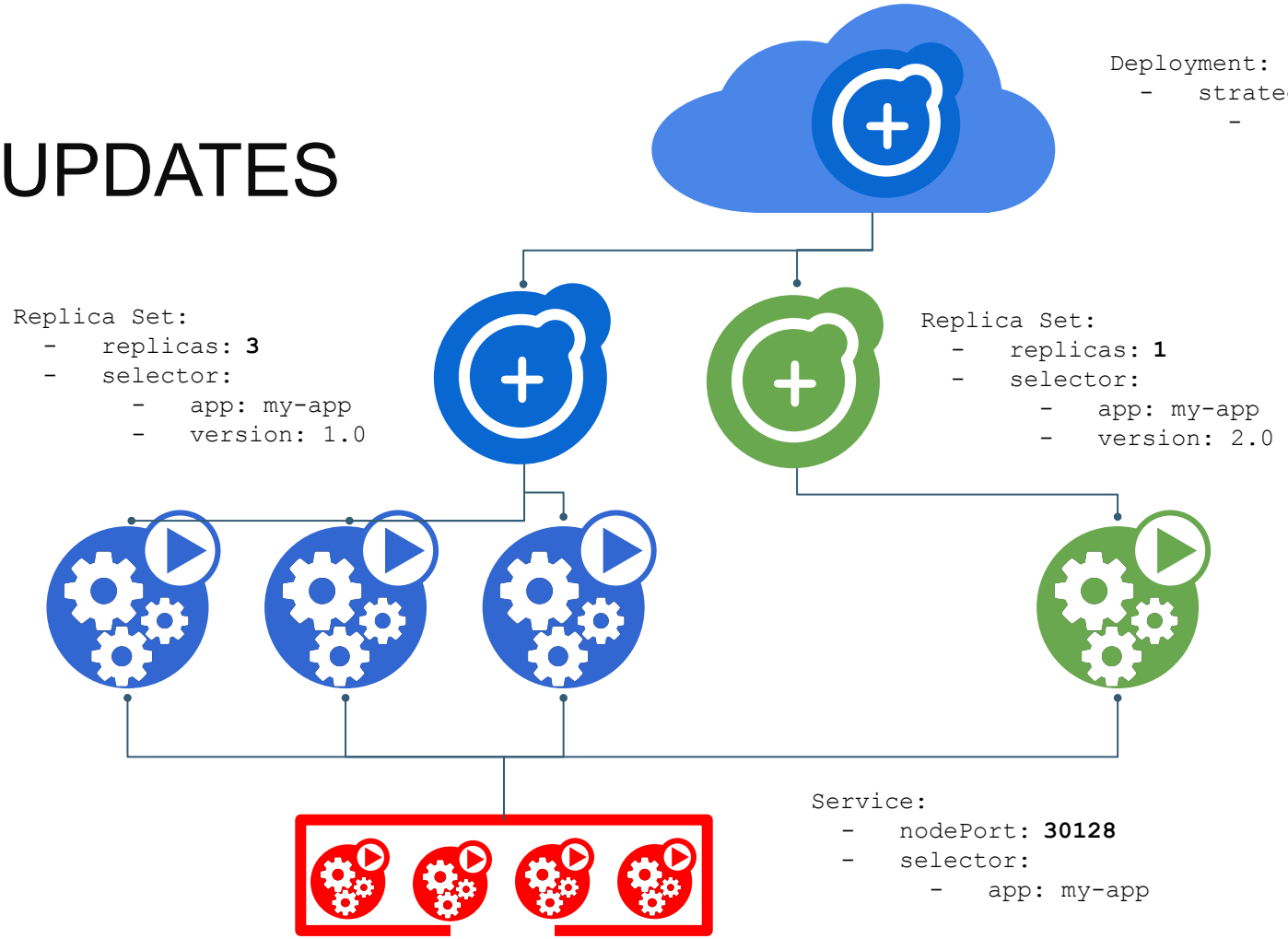
```
Replica Set:
  -   replicas: 4
  -   selector:
        -   app: my-app
        -   version: 1.0
```

```
Service:
  -   nodePort: 30128
  -   selector:
        -   app: my-app
```

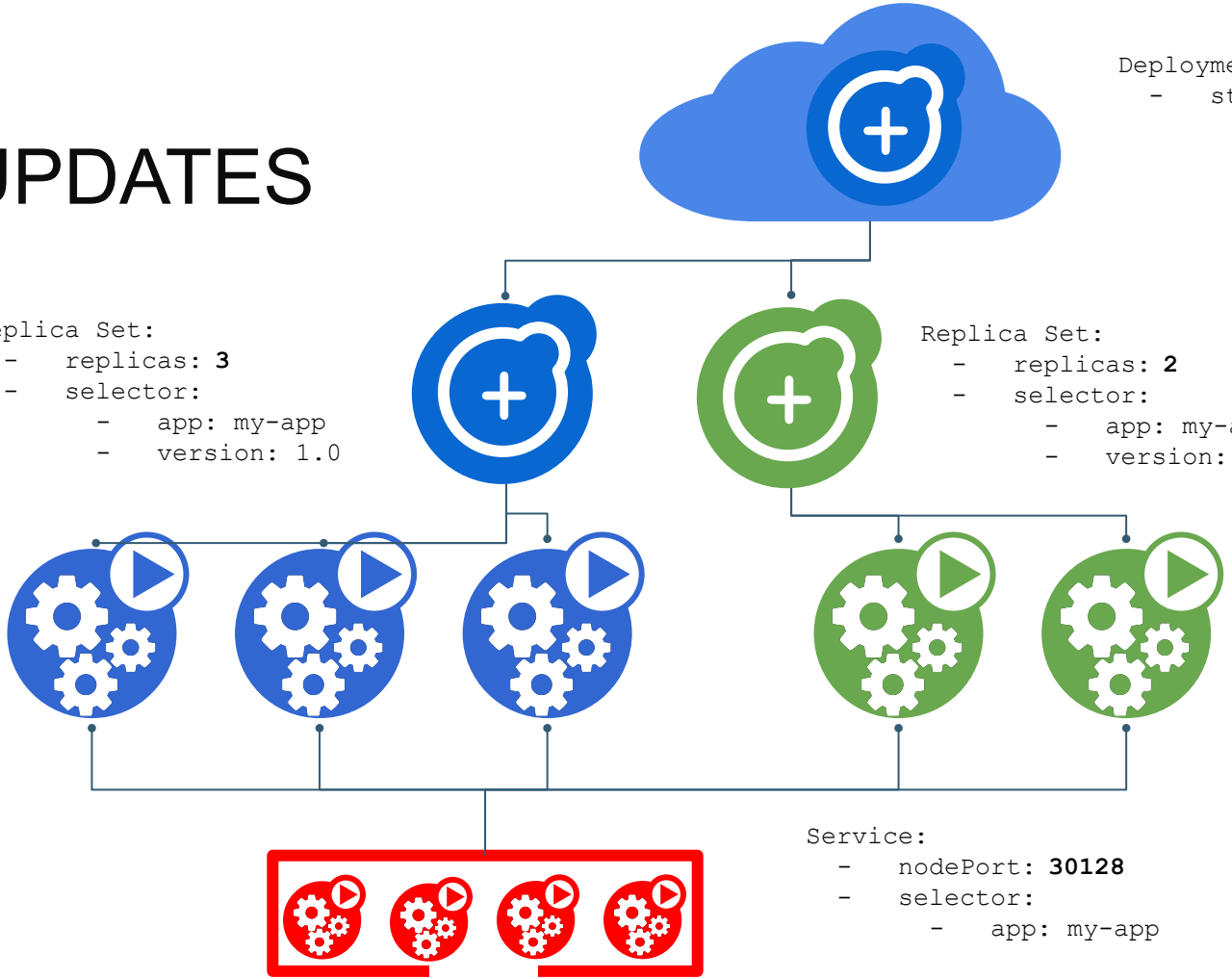
UPDATES


Deployment:
- strategy:
- rollingUpdate
Replica Set:
- replicas: 4
- selector:
- app: my-app
- version: 1.0
Replica Set:
- replicas: 1
- selector:
- app: my-app
- version: 2.0
Service:
- nodePort: 30128
- selector:
- app: my-app

UPDATES


Deployment:
- strategy:
- rollingUpdate
Replica Set:
- replicas: 3
- selector:
- app: my-app
- version: 1.0
Replica Set:
- replicas: 1
- selector:
- app: my-app
- version: 2.0
Service:
- nodePort: 30128
- selector:
- app: my-app

UPDATES



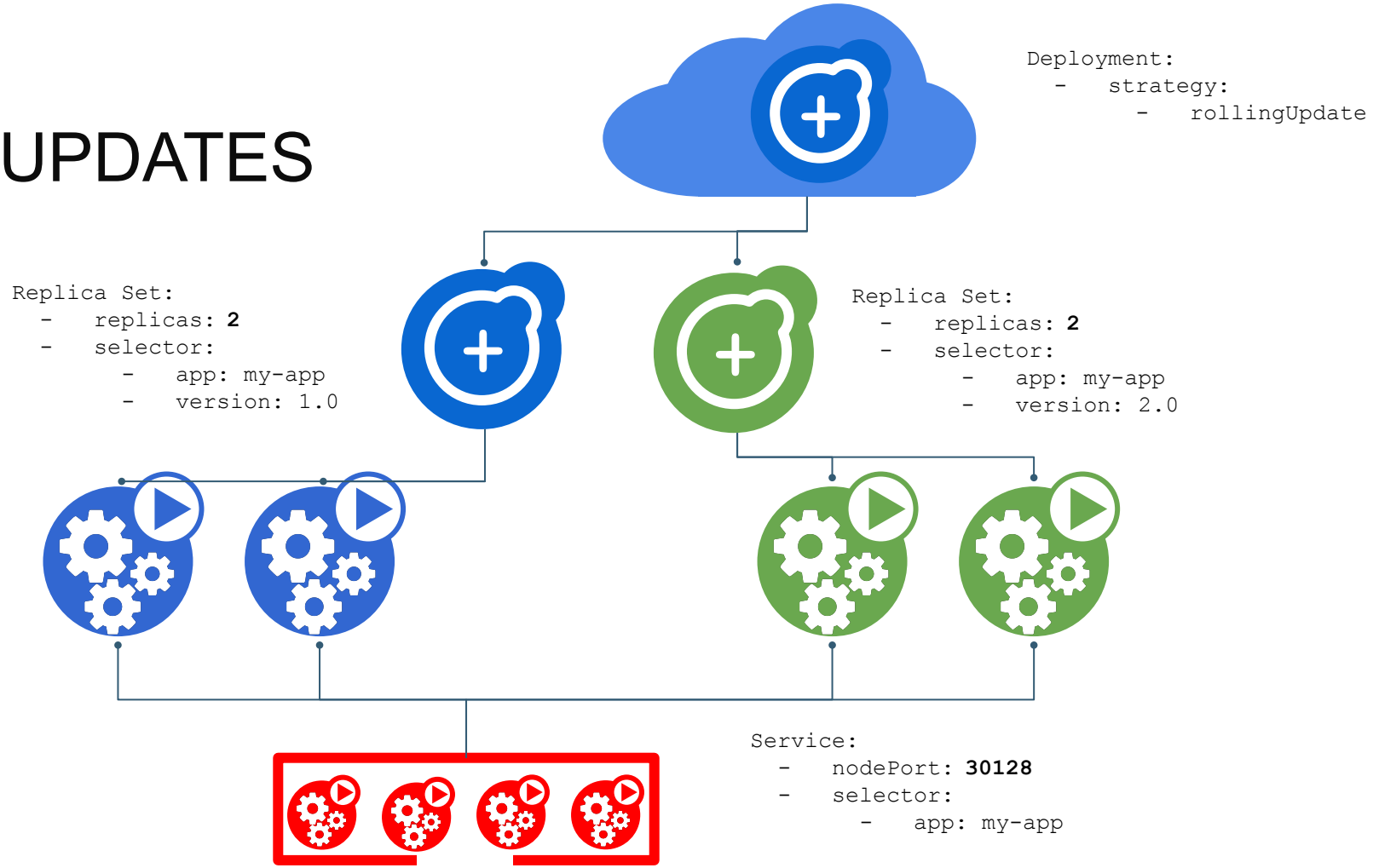
```
Deployment:
  -   strategy:
        -   rollingUpdate
```

```
Replica Set:
  -   replicas: 3
  -   selector:
        -   app: my-app
        -   version: 1.0
```

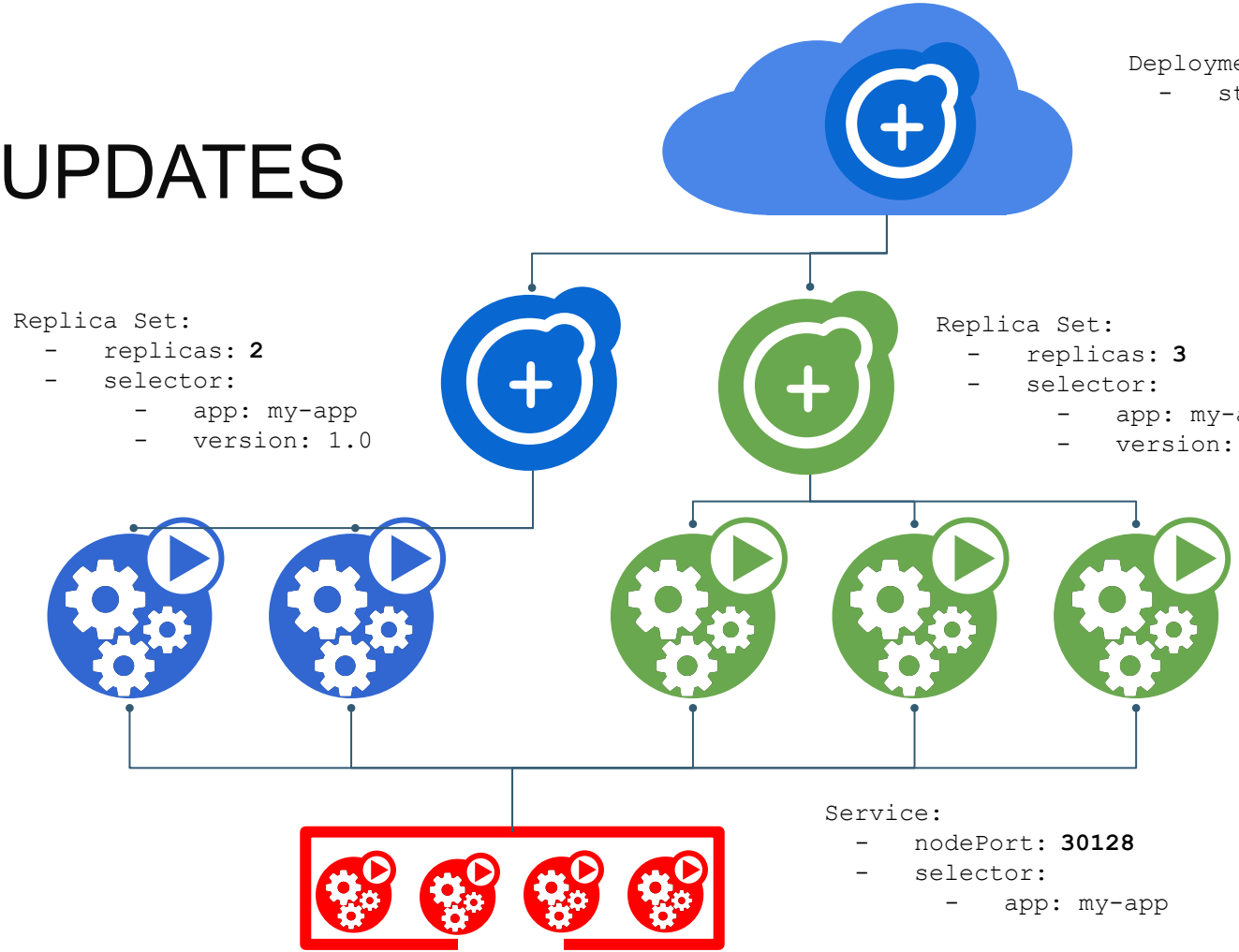
```
Replica Set:
  -   replicas: 2
  -   selector:
        -   app: my-app
        -   version: 2.0
```

```
Service:
  -   nodePort: 30128
  -   selector:
        -   app: my-app
```

UPDATES


Deployment:
- strategy:
- rollingUpdate
Replica Set:
- replicas: 2
- selector:
- app: my-app
- version: 1.0
Replica Set:
- replicas: 2
- selector:
- app: my-app
- version: 2.0
Service:
- nodePort: 30128
- selector:
- app: my-app

UPDATES



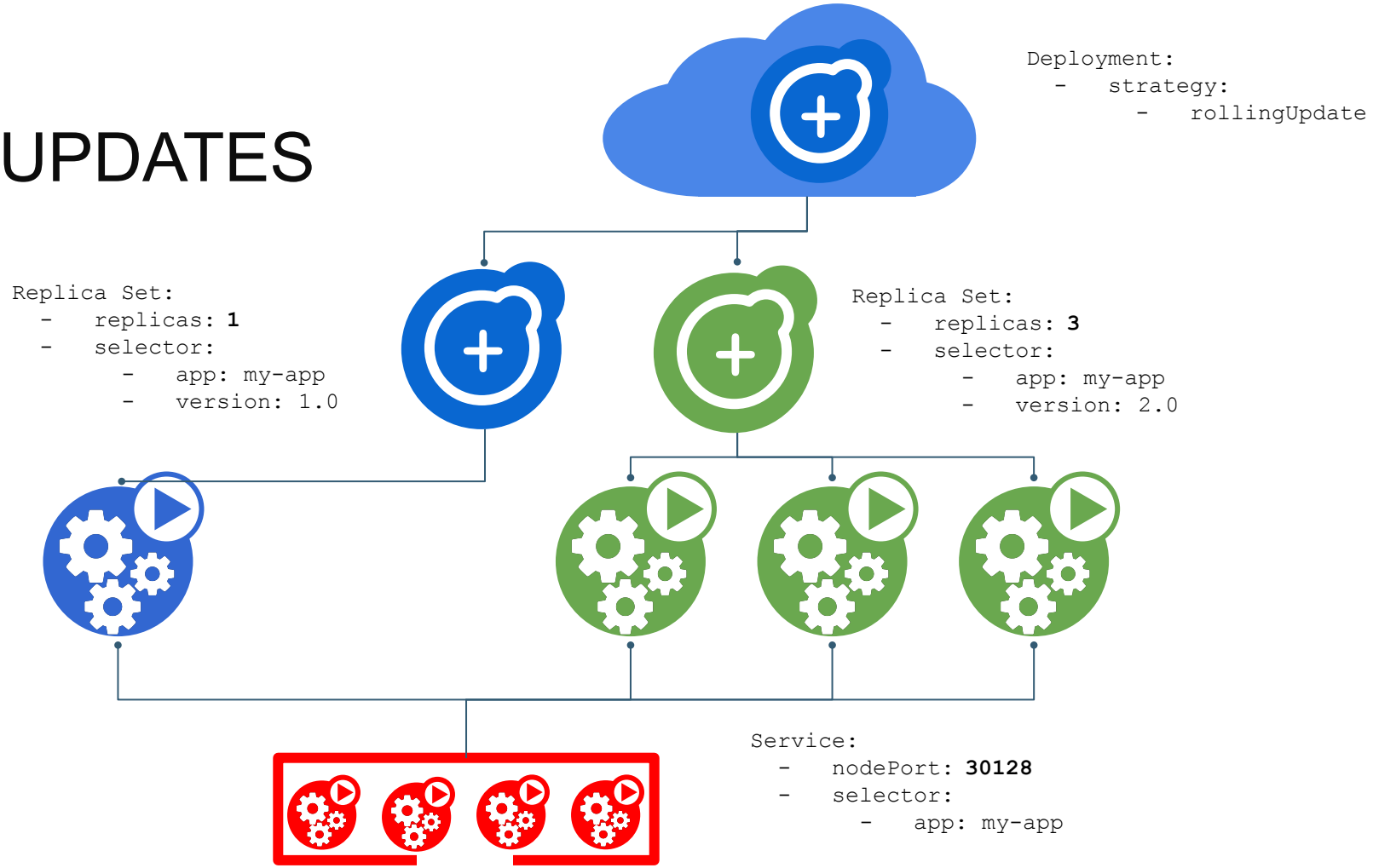
Deployment:
- strategy:
  - rollingUpdate

Replica Set:
- replicas: **2**
- selector:
  - app: my-app
  - version: 1.0

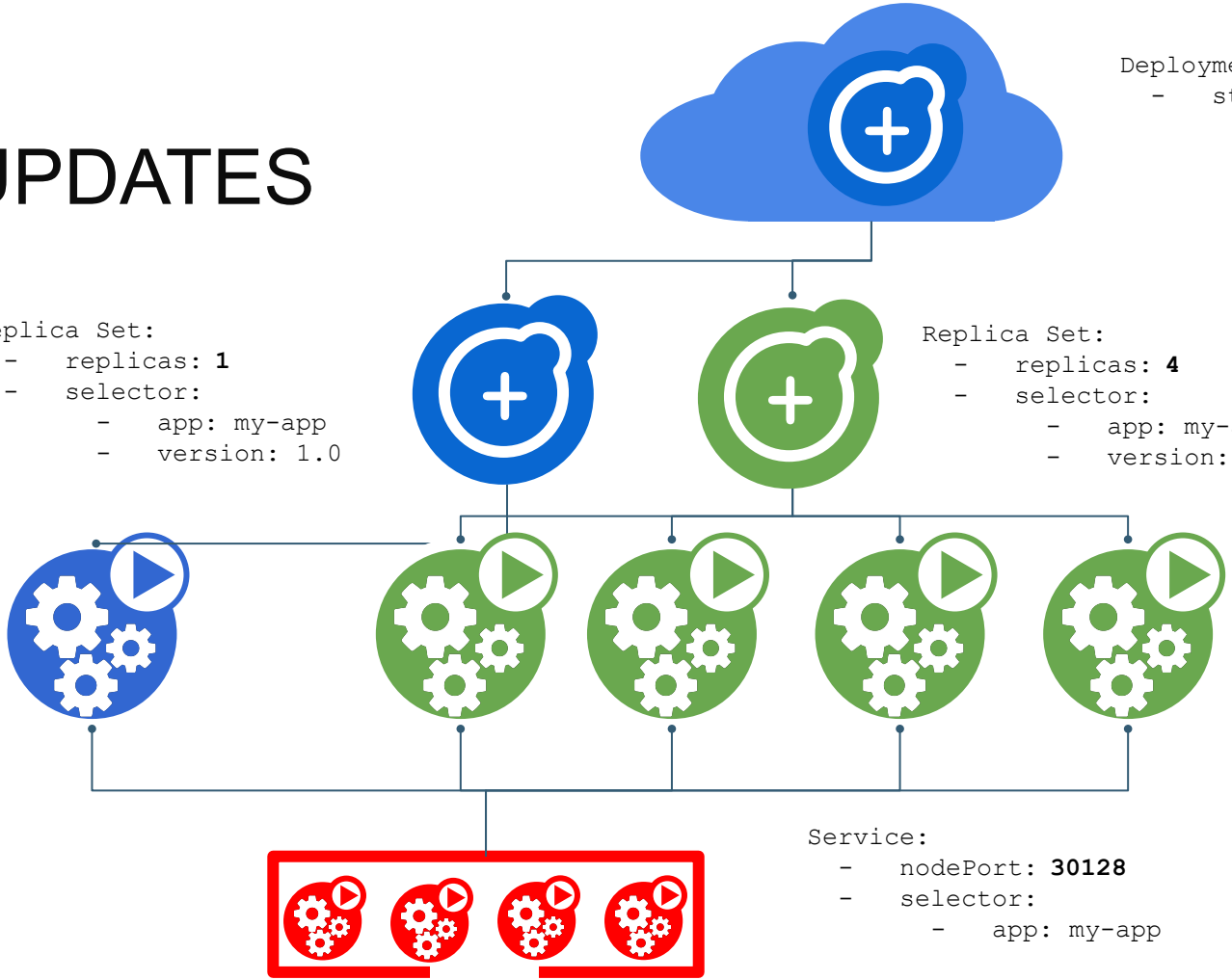
Replica Set:
- replicas: **3**
- selector:
  - app: my-app
  - version: 2.0

Service:
- nodePort: **30128**
- selector:
  - app: my-app

UPDATES


Deployment:
- strategy:
- rollingUpdate
Replica Set:
- replicas: 1
- selector:
- app: my-app
- version: 1.0
Replica Set:
- replicas: 3
- selector:
- app: my-app
- version: 2.0
Service:
- nodePort: 30128
- selector:
- app: my-app

UPDATES

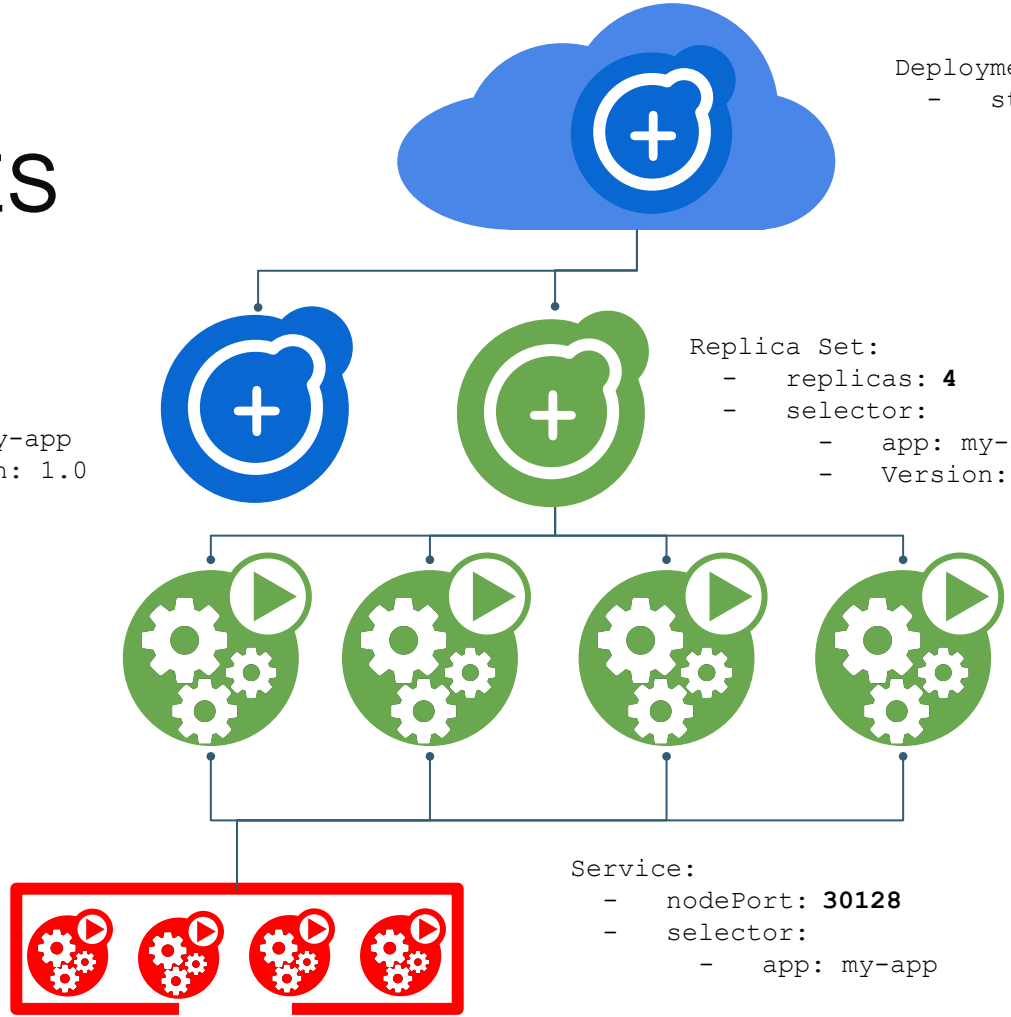

Deployment:
- strategy:
- rollingUpdate
Replica Set:
- replicas: 1
- selector:
- app: my-app
- version: 1.0
Replica Set:
- replicas: 4
- selector:
- app: my-app
- version: 2.0
Service:
- nodePort: 30128
- selector:
- app: my-app

UPDATES

```
Deployment:
  -   strategy:
        -   rollingUpdate
```

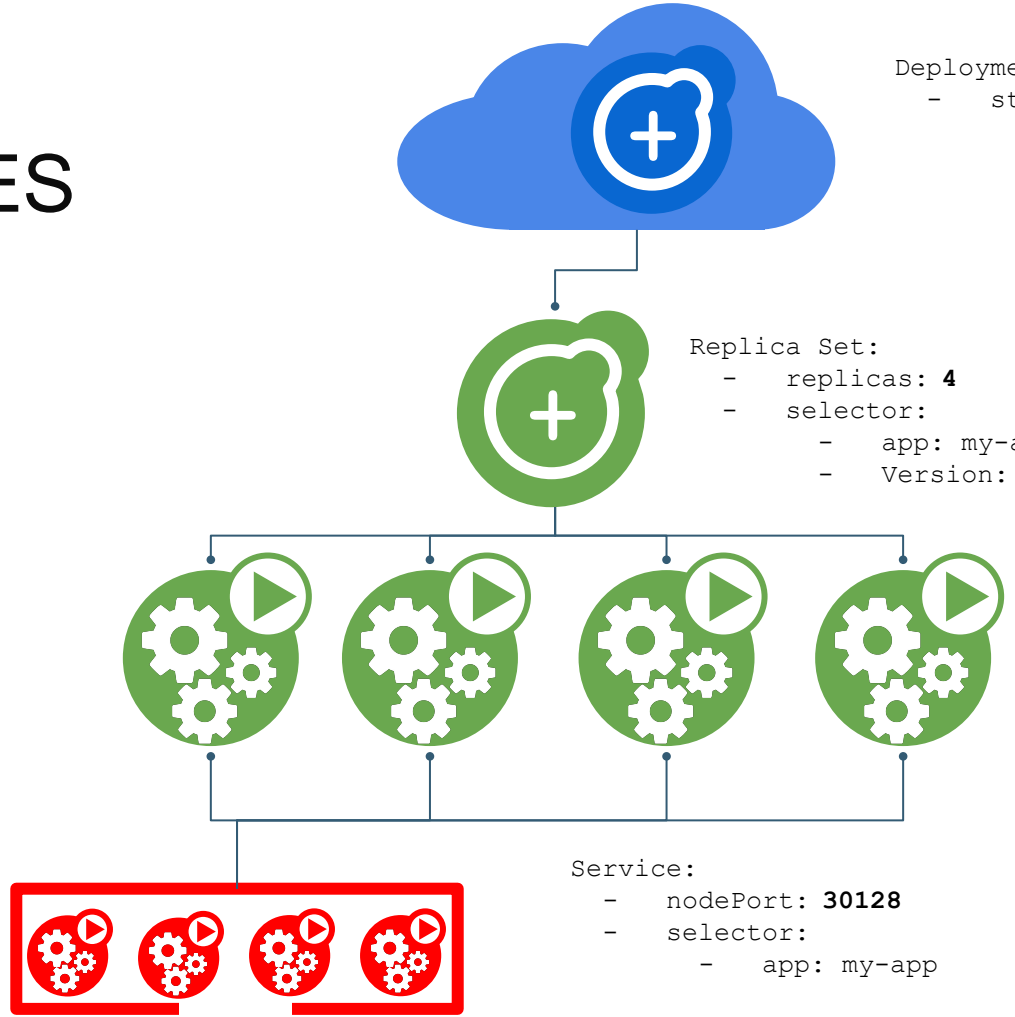
```
Replica Set:
  -   replicas: 0
  -   selector:
        -   app: my-app
        -   version: 1.0
```

```
Replica Set:
  -   replicas: 4
  -   selector:
        -   app: my-app
        -   Version: 2.0
```



```
Service:
  -   nodePort: 30128
  -   selector:
        -   app: my-app
```

UPDATES

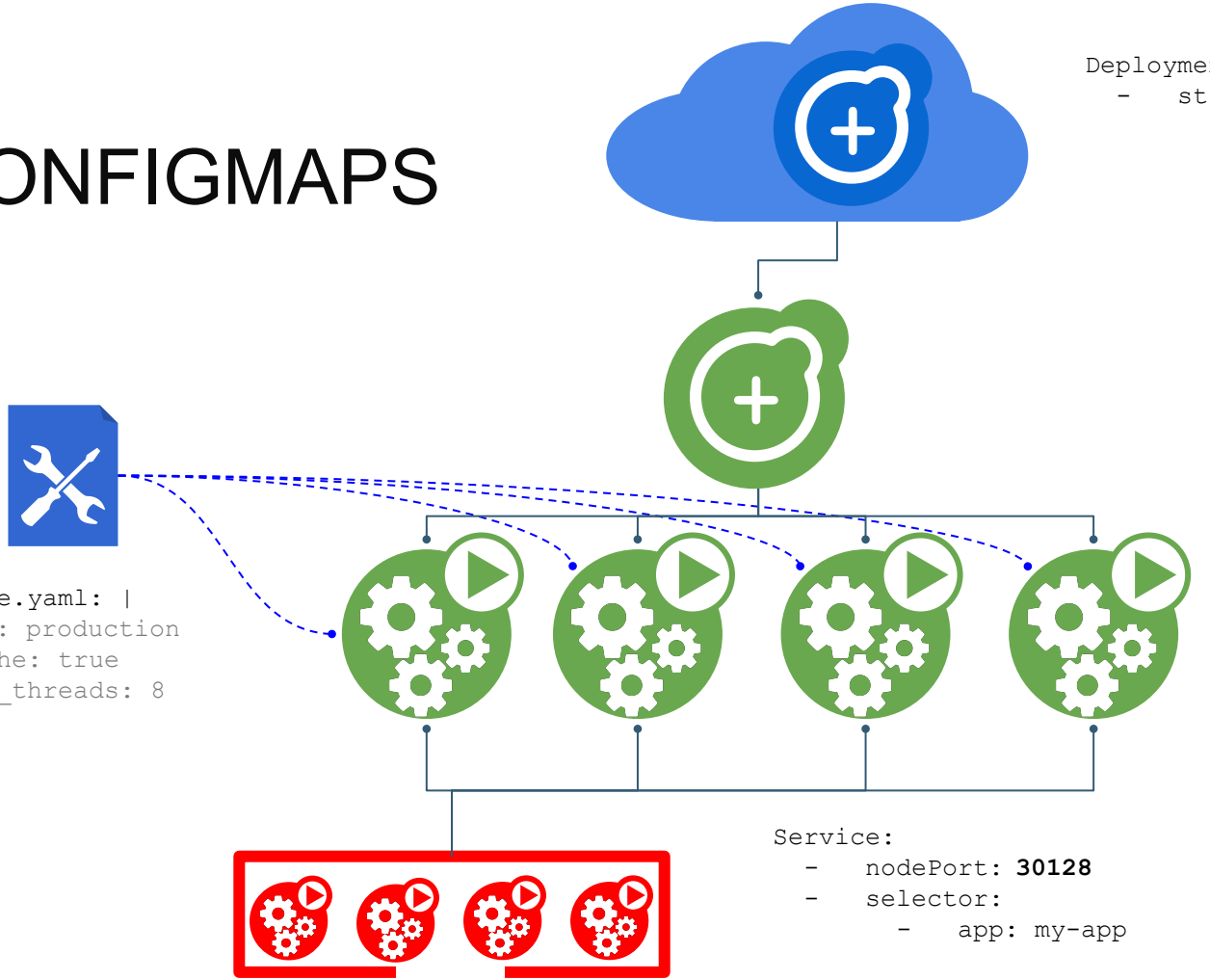


```
Deployment:
  -   strategy:
        -   rollingUpdate
```

```
Replica Set:
  -   replicas: 4
  -   selector:
        -   app: my-app
        -   Version: 2.0
```

```
Service:
  -   nodePort: 30128
  -   selector:
        -   app: my-app
```

CONFIGMAPS

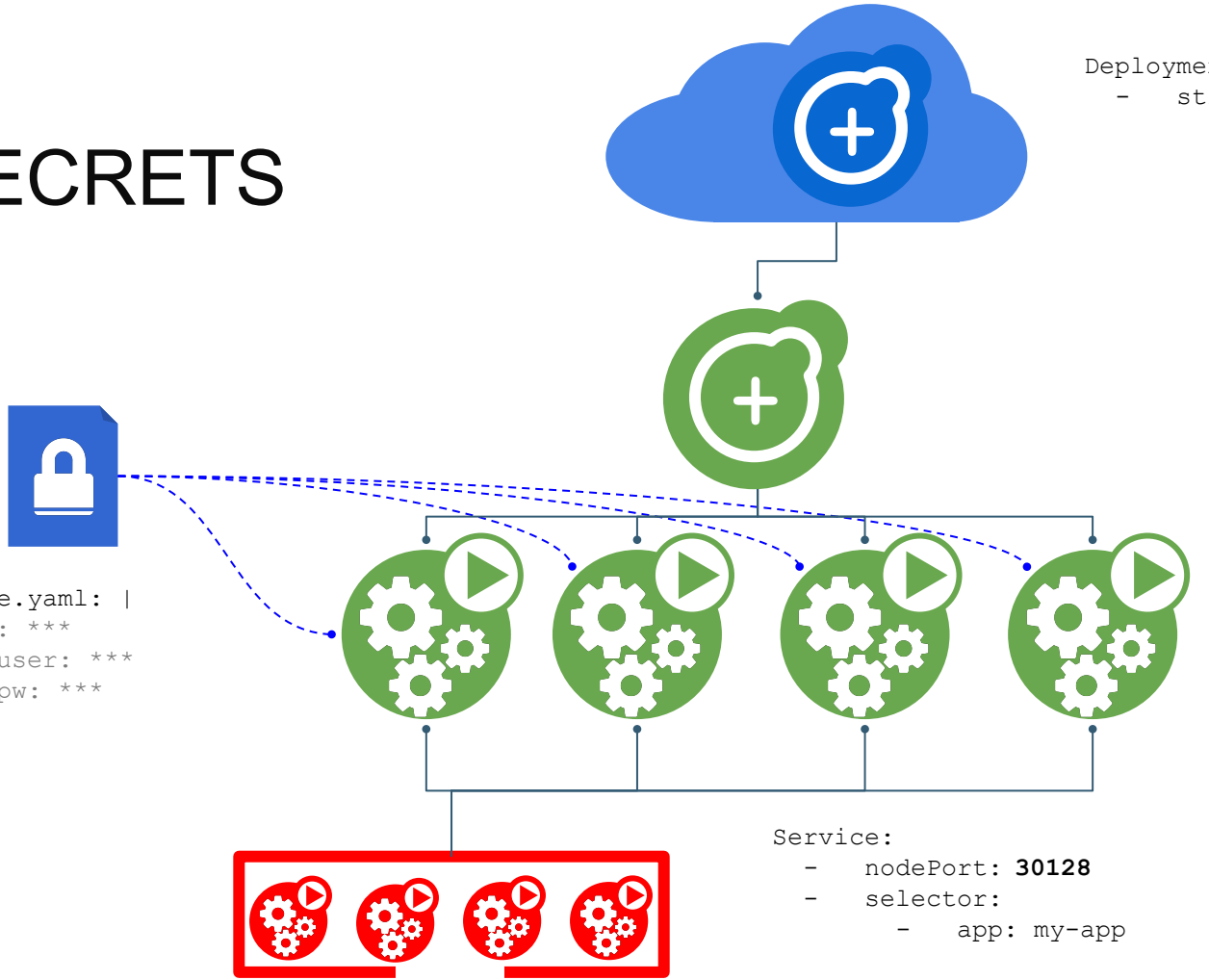


```
Deployment:
  -   strategy:
        -   rollingUpdate
```

```
ConfigMap:
  -   sample.yaml: |
        env: production
        cache: true
        max_threads: 8
```

```
Service:
  -   nodePort: 30128
  -   selector:
        -   app: my-app
```

SECRETS



```
Deployment:
  -   strategy:
        -   rollingUpdate
```

```
Secret:
  -   sample.yaml: |
        key: ***
        db_user: ***
        db_pw: ***
```

```
Service:
  -   nodePort: 30128
  -   selector:
        -   app: my-app
```

Demo

Minikube

k8s Recap

- Multiple resource types
  - Deployments
  - Services
  - ConfigMaps
  - Secrets
  - …

- Serializable Manifests: Infrastructure as code

Helm: Package Manager

- Collaboration with Google, Bitnami, Deis and others
- Deis/Helm -> Kubernetes/Helm
- Architecture:
  - Client: Helm
  - Server: Tiller

Helm core values

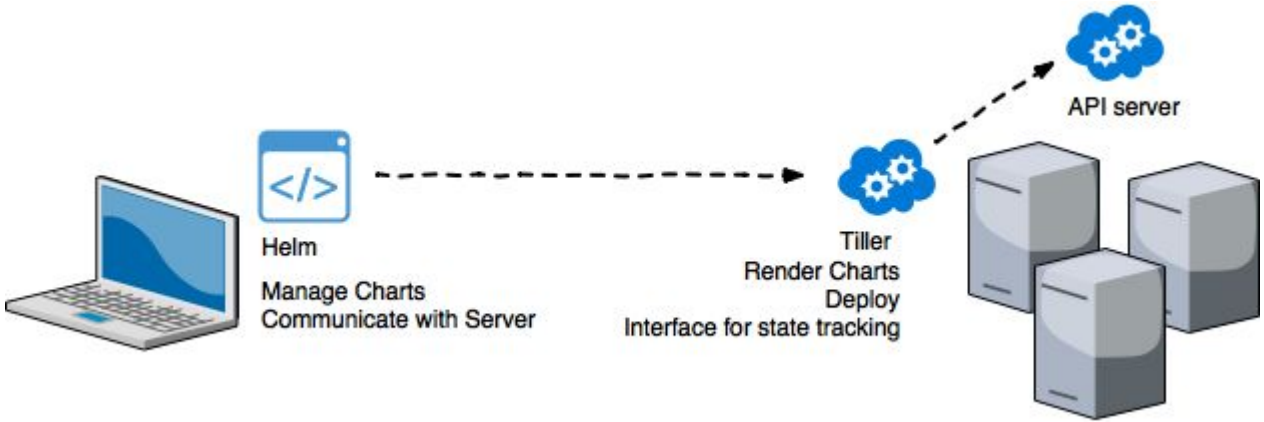
- Aim for the Apt/Yum/Homebrew UX
- Ensure collaboration
- Reproducible releases
- Shareable Packages

Chart, Repositories, Releases

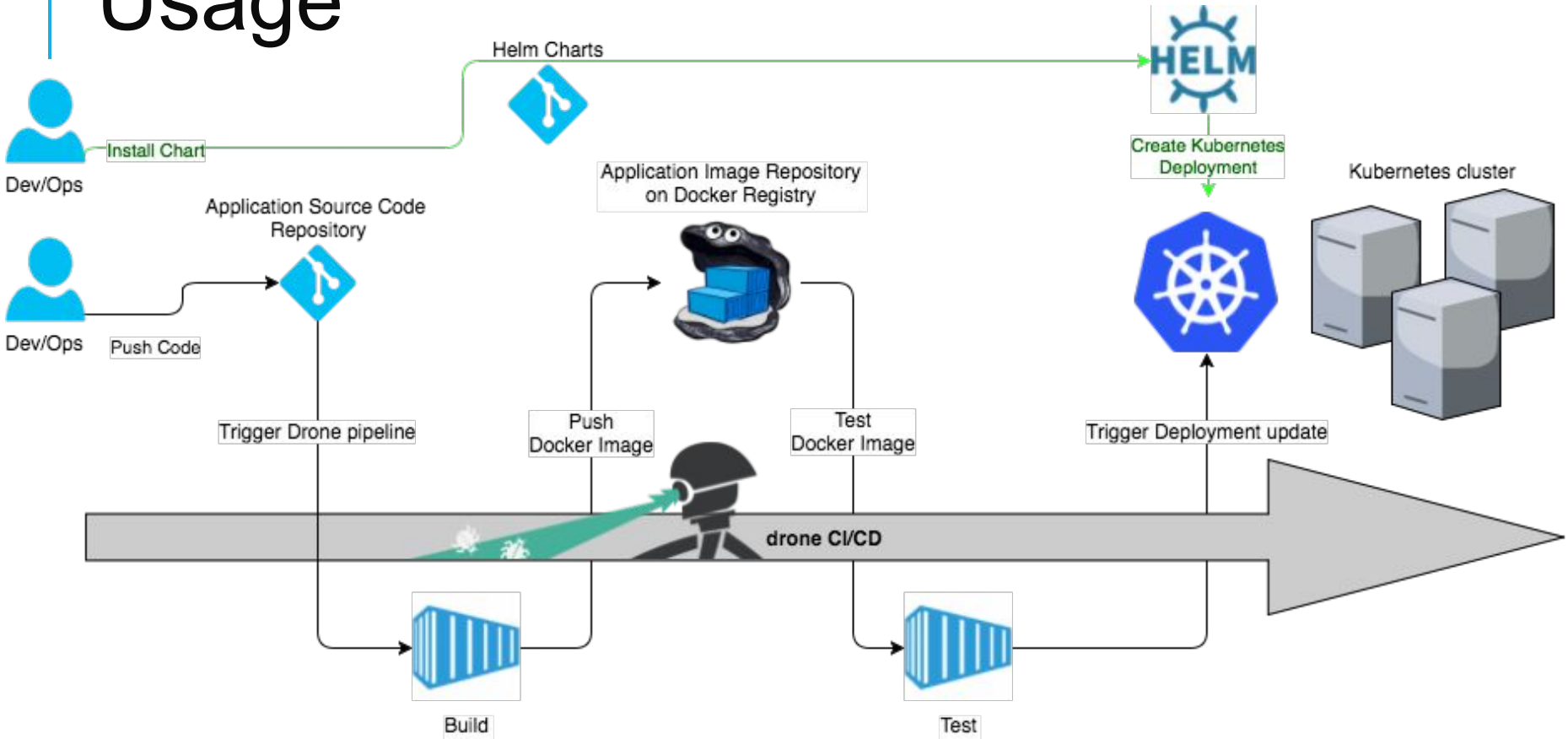
- **Chart**: “Package”, “Bundle”
- **Repository**: Package Repository
  Evolving towards a registry (CoreOS)
- **Release**: Installed Chart (same chart can be installed multiple times)

Basic Architecture

Client - Server



Usage


Helm Charts
HELM
Install Chart
Dev/Ops
Create Kubernetes Deployment
Kubernetes cluster
Application Image Repository on Docker Registry
Application Source Code Repository
Dev/Ops
Push Code
Trigger Drone pipeline
Push Docker Image
Test Docker Image
Trigger Deployment update
drone CI/CD
Build
Test

Review using helm

- Install charts (creates a release)
  - `helm install`
- List releases
  - `helm ls`
- Bootstrap charts
  - `helm create`

More features

- Upgrade Releases
  - `helm upgrade <release>`
- Search Charts
  - `helm search`
  - KubeApps.com (Monocular - https://github.com/helm/monocular)
- Manage Chart dependencies
  - `requirements.yaml`
  - `helm dep up`
- Helm Plugins
  - Keybase: `helm keybase sign/verify`
  - GitHub: `helm github push`