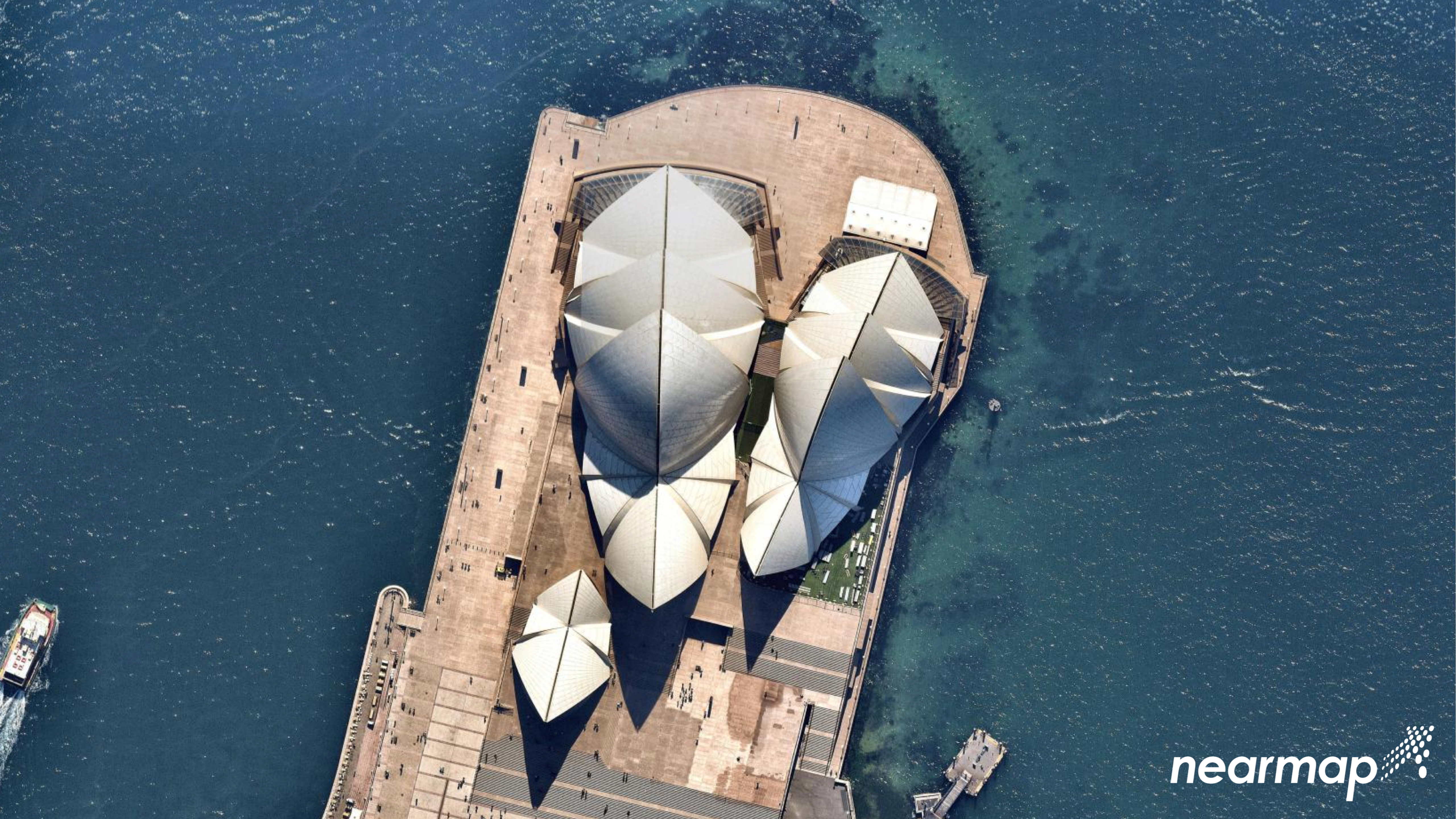
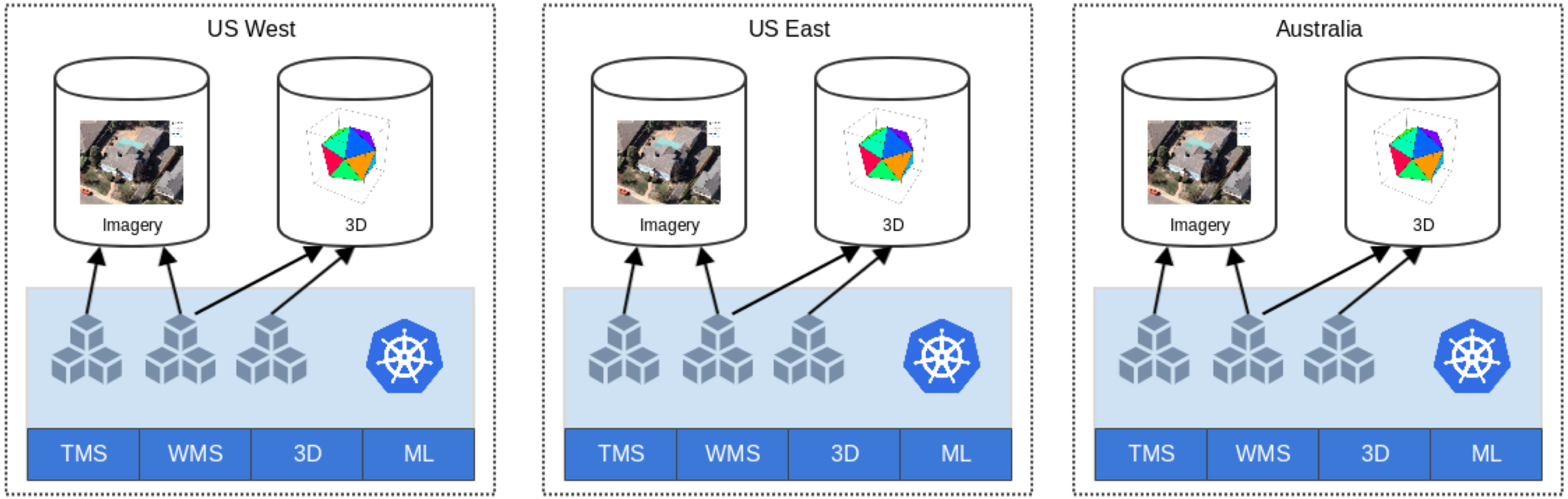
CONTINUOUS DELIVERY MEETS CUSTOM KUBERNETES CONTROLLER.

A DECLARATIVE CONFIGURATION APPROACH TO CI/CD

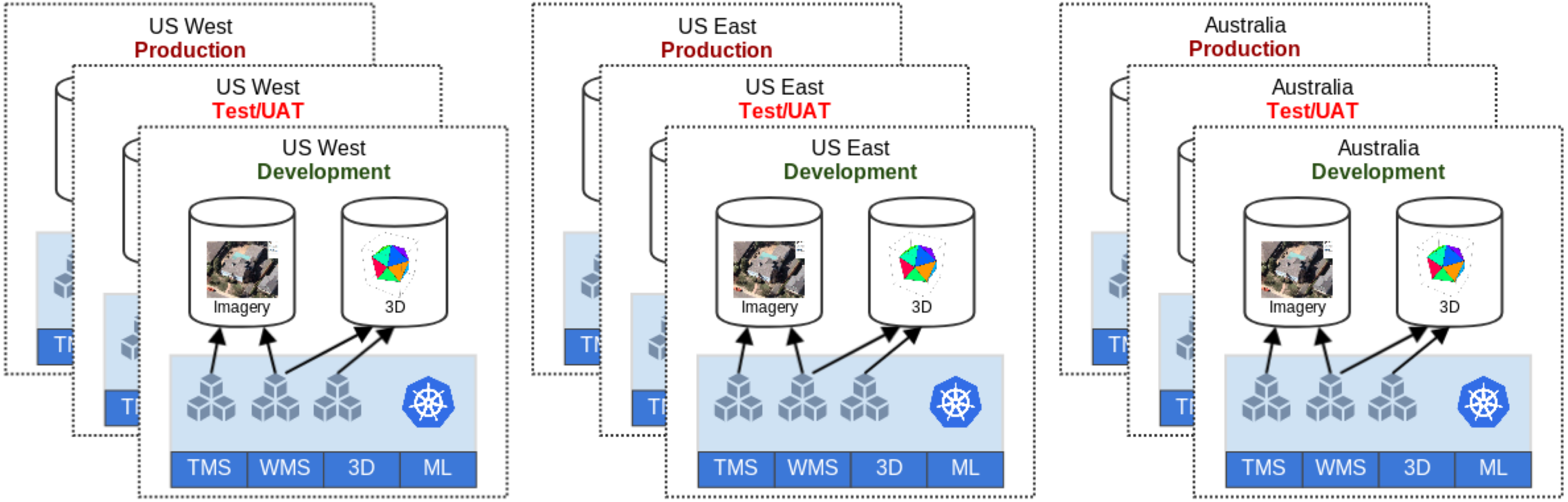
Presenters:
Simon Cochrane, Director of Engineering (API)
Suneeta Mall, Software Engineer



NEARMAP ENVIRONMENTS.



NEARMAP ENVIRONMENTS.



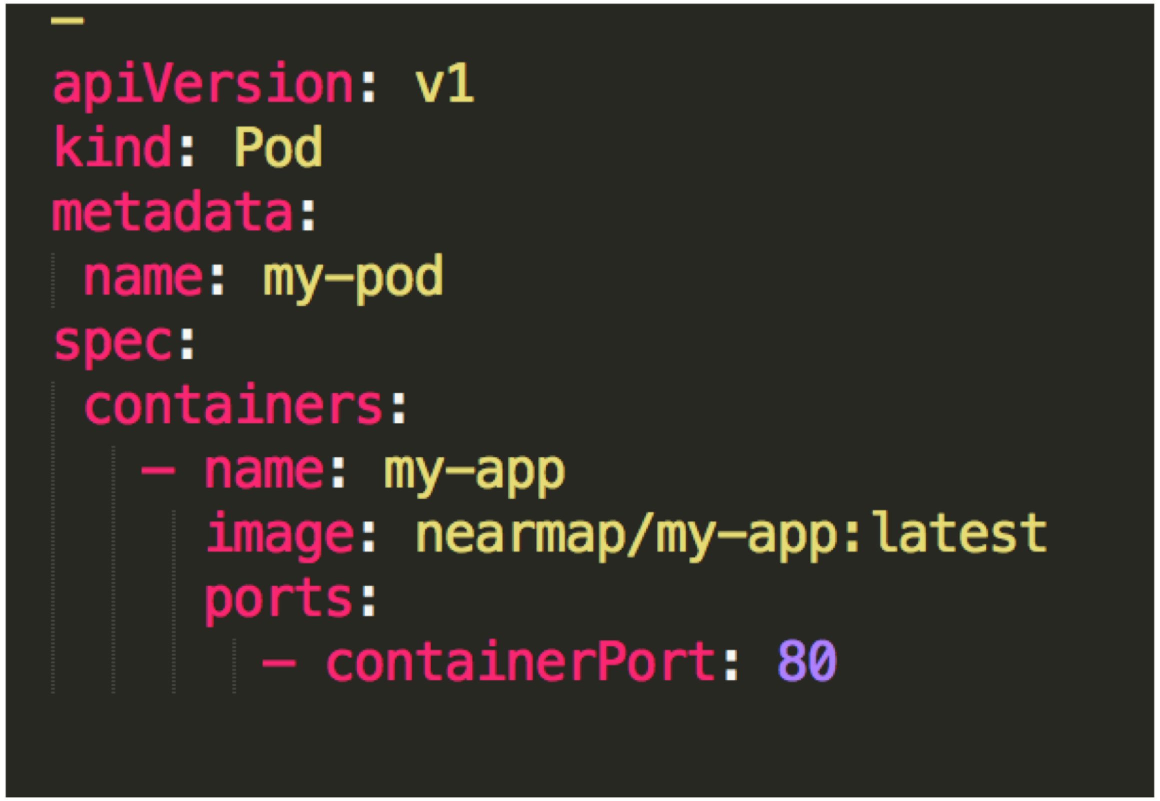
CI/CD ON KUBERNETES.

Kubernetes specifically states that it

"Does not deploy source code and does not build your application. Continuous Integration, Delivery, and Deployment (CI/CD) workflows are determined by organization cultures and preferences as well as technical requirements."

CONFIGURATION FILES.

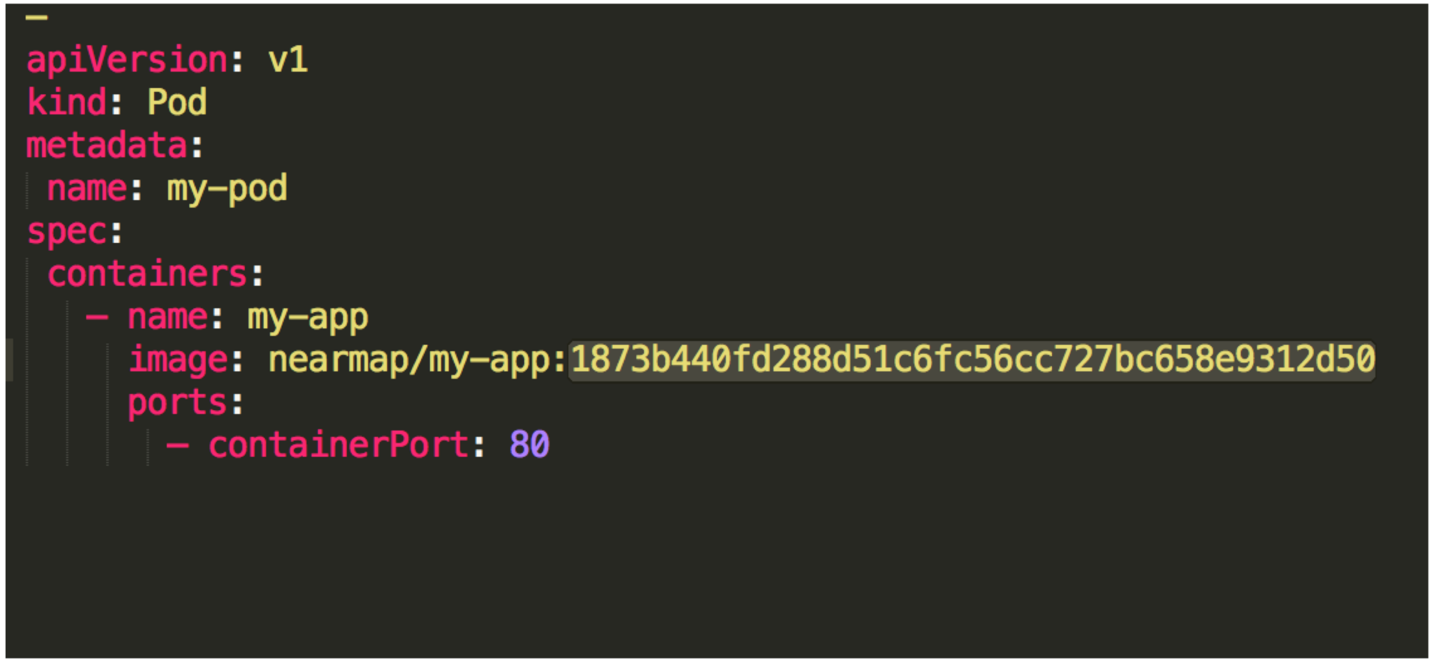
- Recommended not to use the *:latest* tag



```yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
    - name: my-app
      image: nearmap/my-app:latest
      ports:
        - containerPort: 80
```

CONFIGURATION FILES.

- Specify a version number (digest or git hash)
- Should be in source control
- How to manage multiple environments?

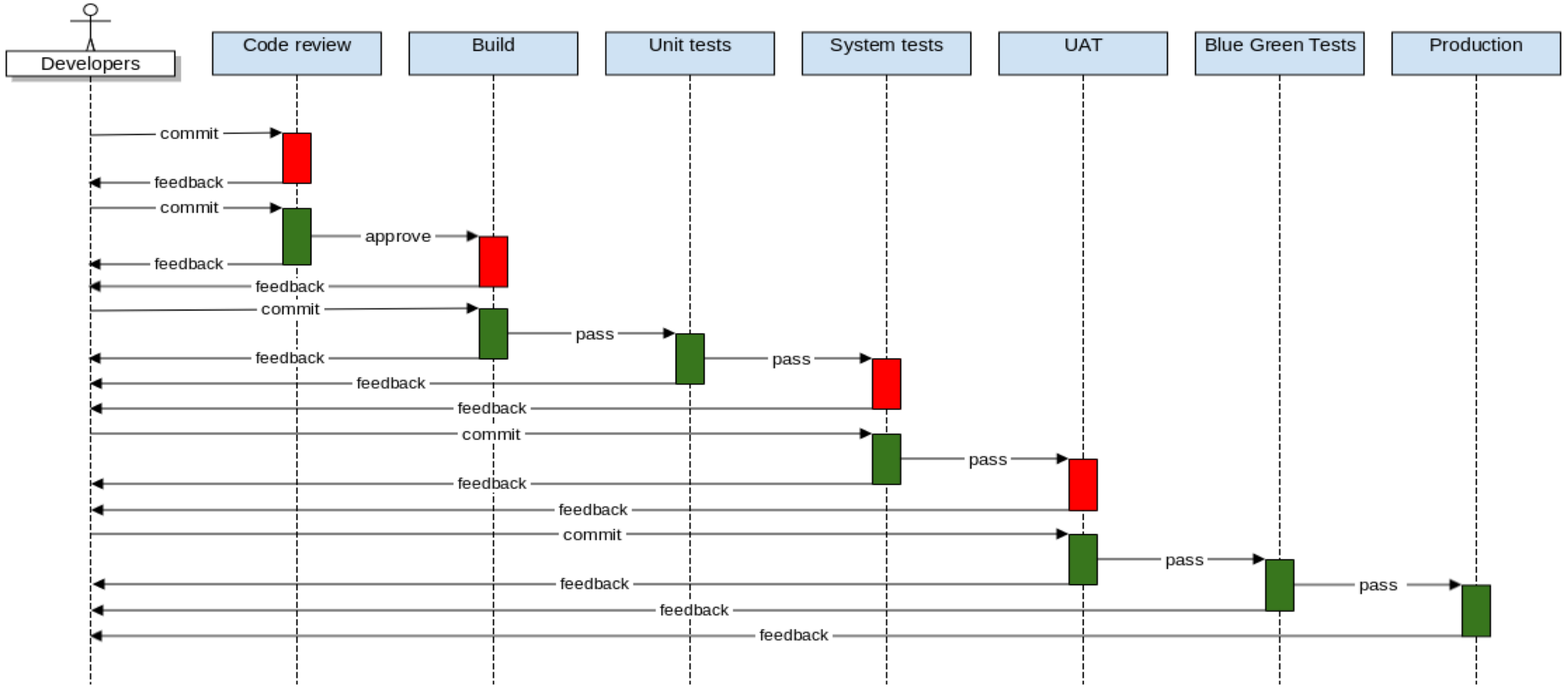


```yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
    - name: my-app
      image: nearmap/my-app:1873b440fd288d51c6fc56cc727bc658e9312d50
      ports:
        - containerPort: 80
```

nearmap

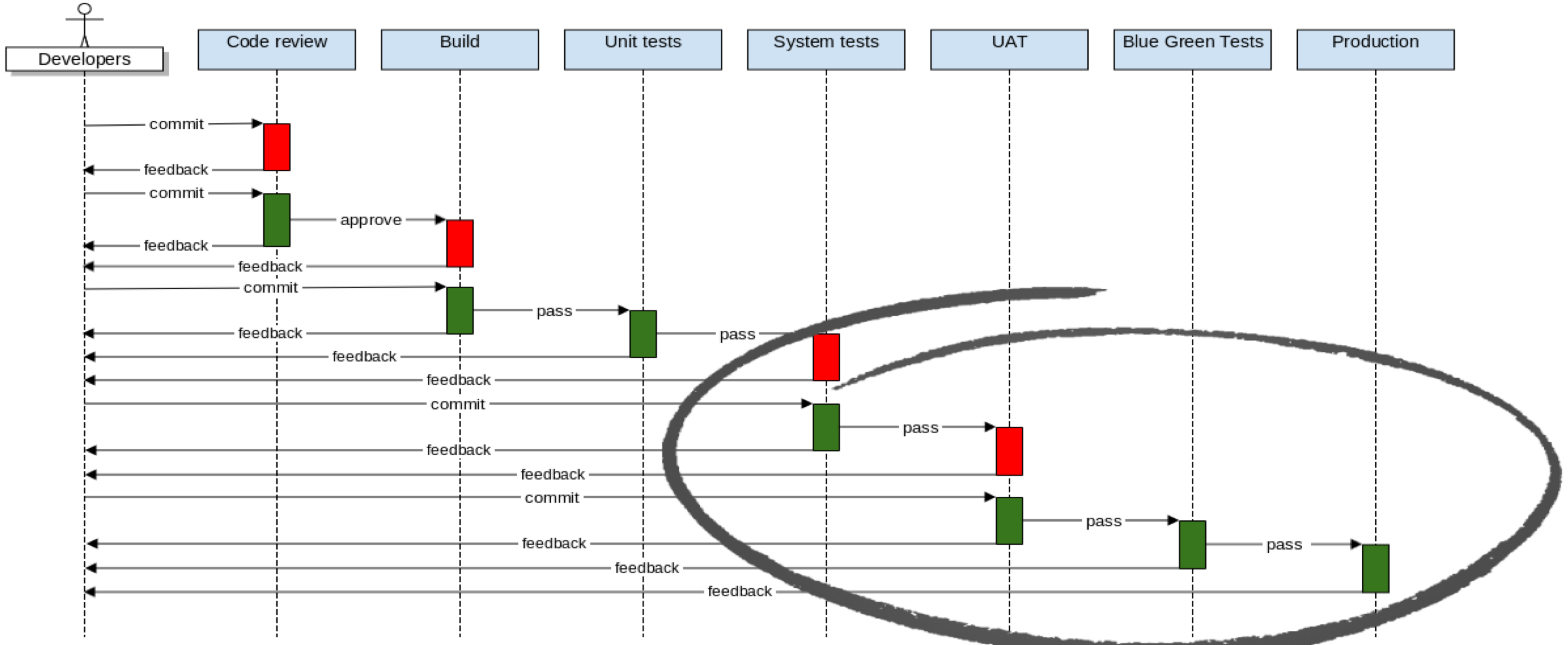
CONTINUOUS DELIVERY.

Set of workflows and validations that provide a reliable process for releasing software.



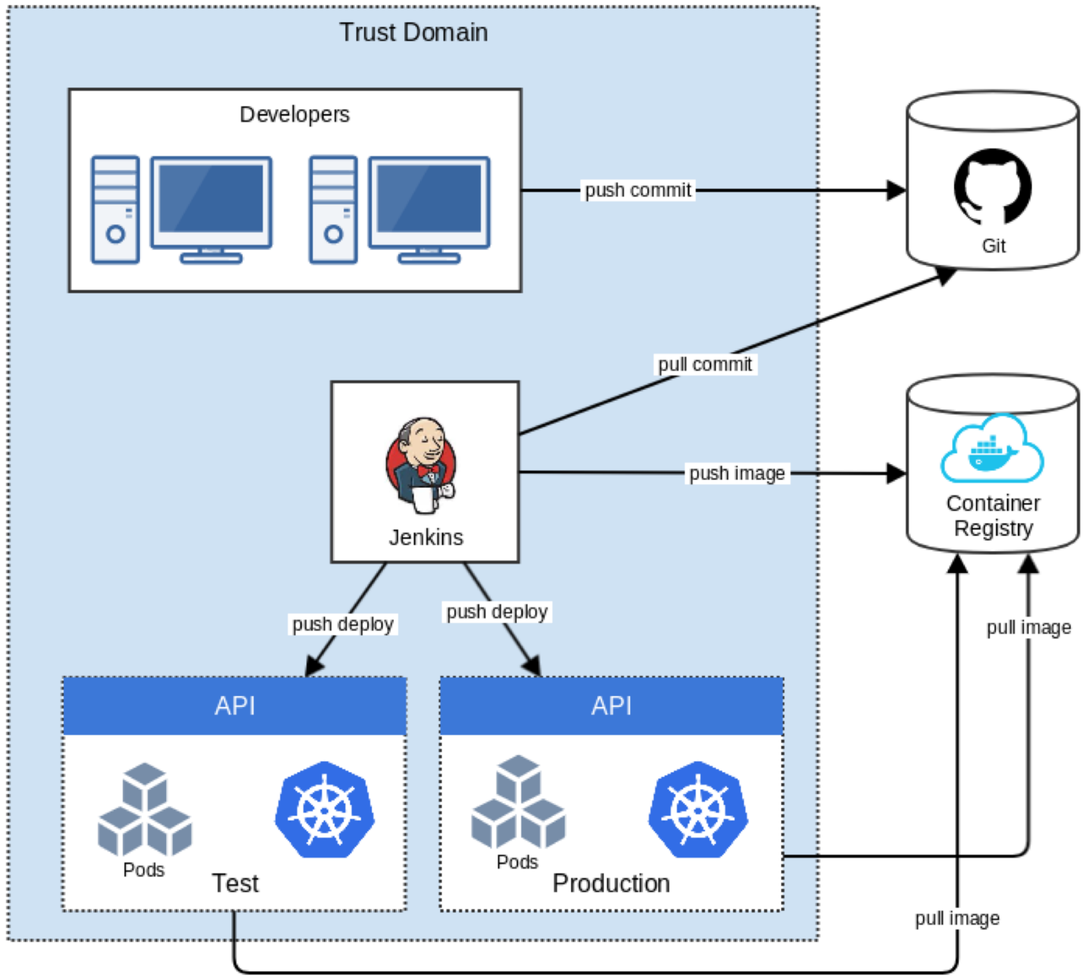
CONTINUOUS DELIVERY.

Set of workflows and validations that provide a reliable process for releasing software.



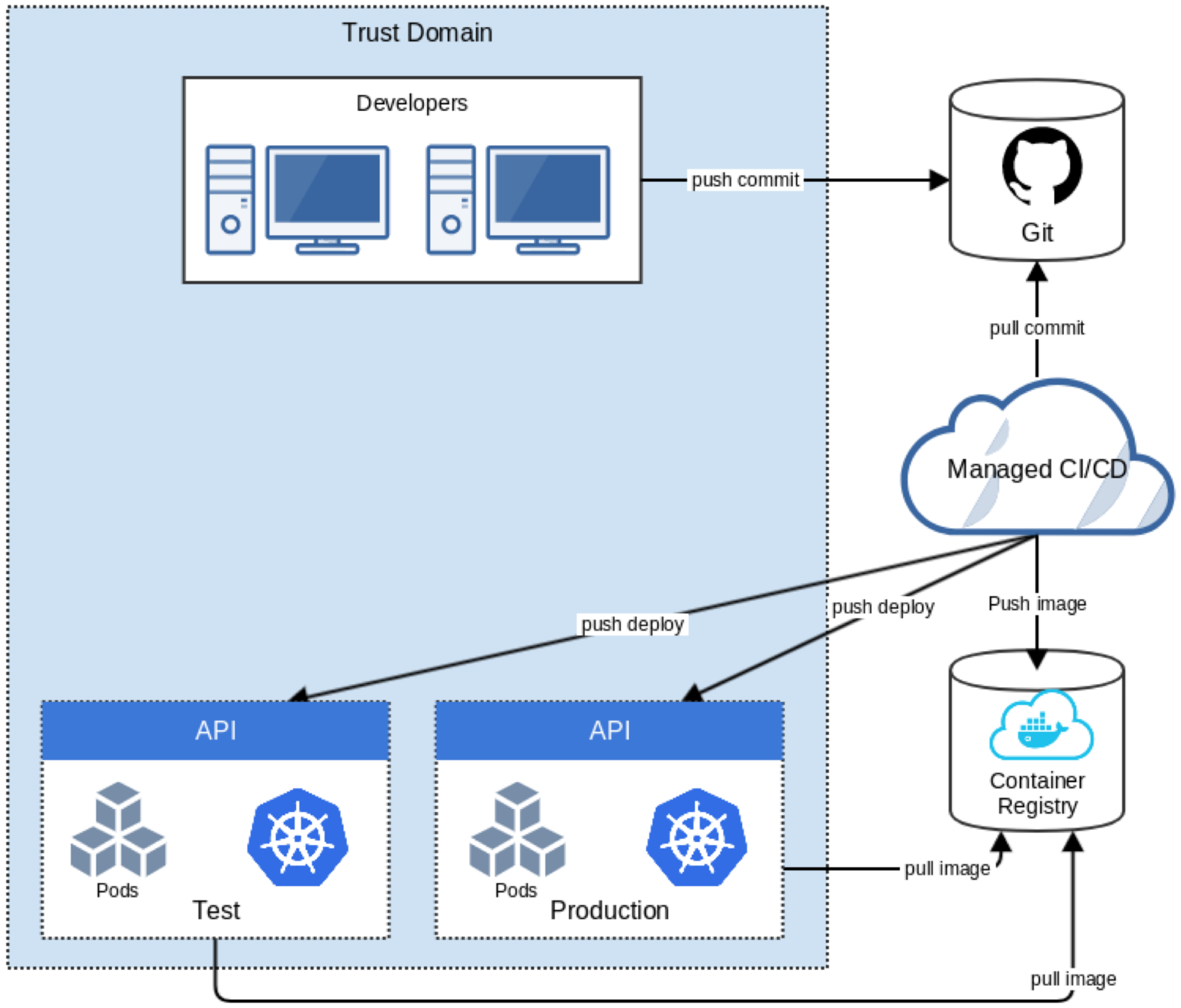
SELF-HOSTED CD.

e.g. Jenkins, TeamCity



MANAGED CD.

E.g. Circle CI, Shippable, AWS CodePipeline



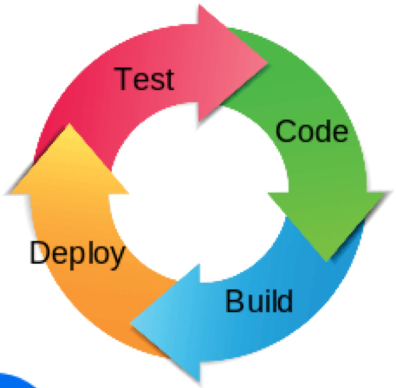
EXISTING SOLUTIONS?

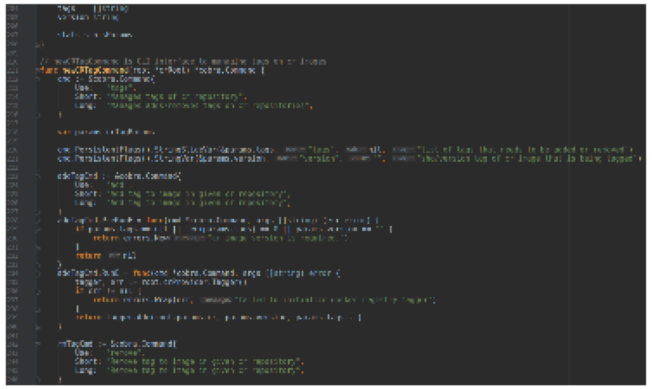
HELM

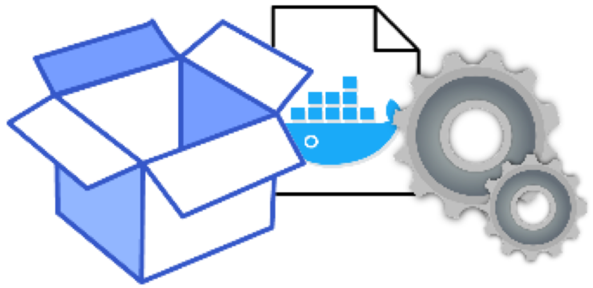
Spinnaker

SKAFFOLD

Hello World!







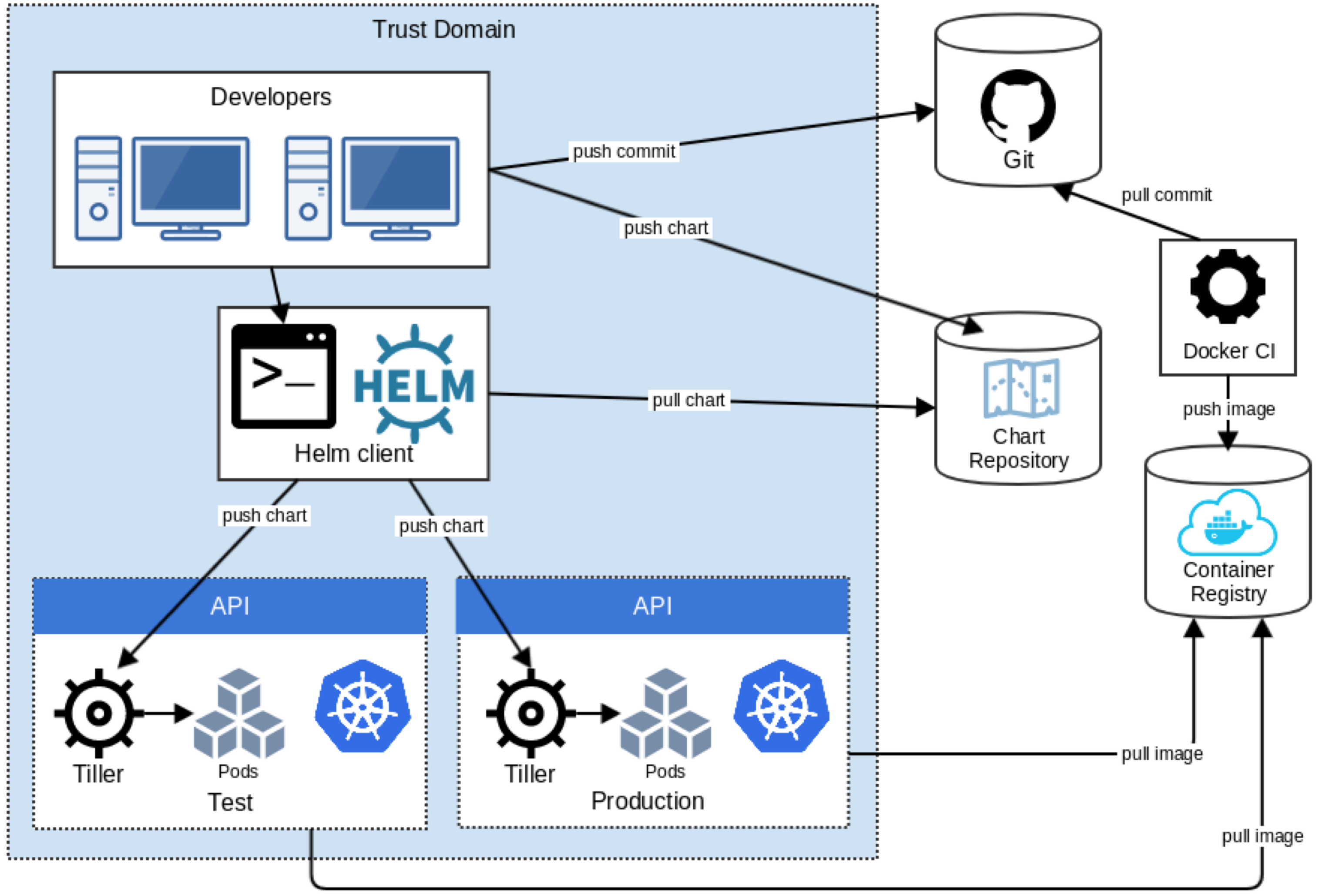
CONCLUSION:

Best used during development cycle for fast feedback loops. Once development is complete, another CI/CD tool should take over.

HELM.

CONCLUSION:

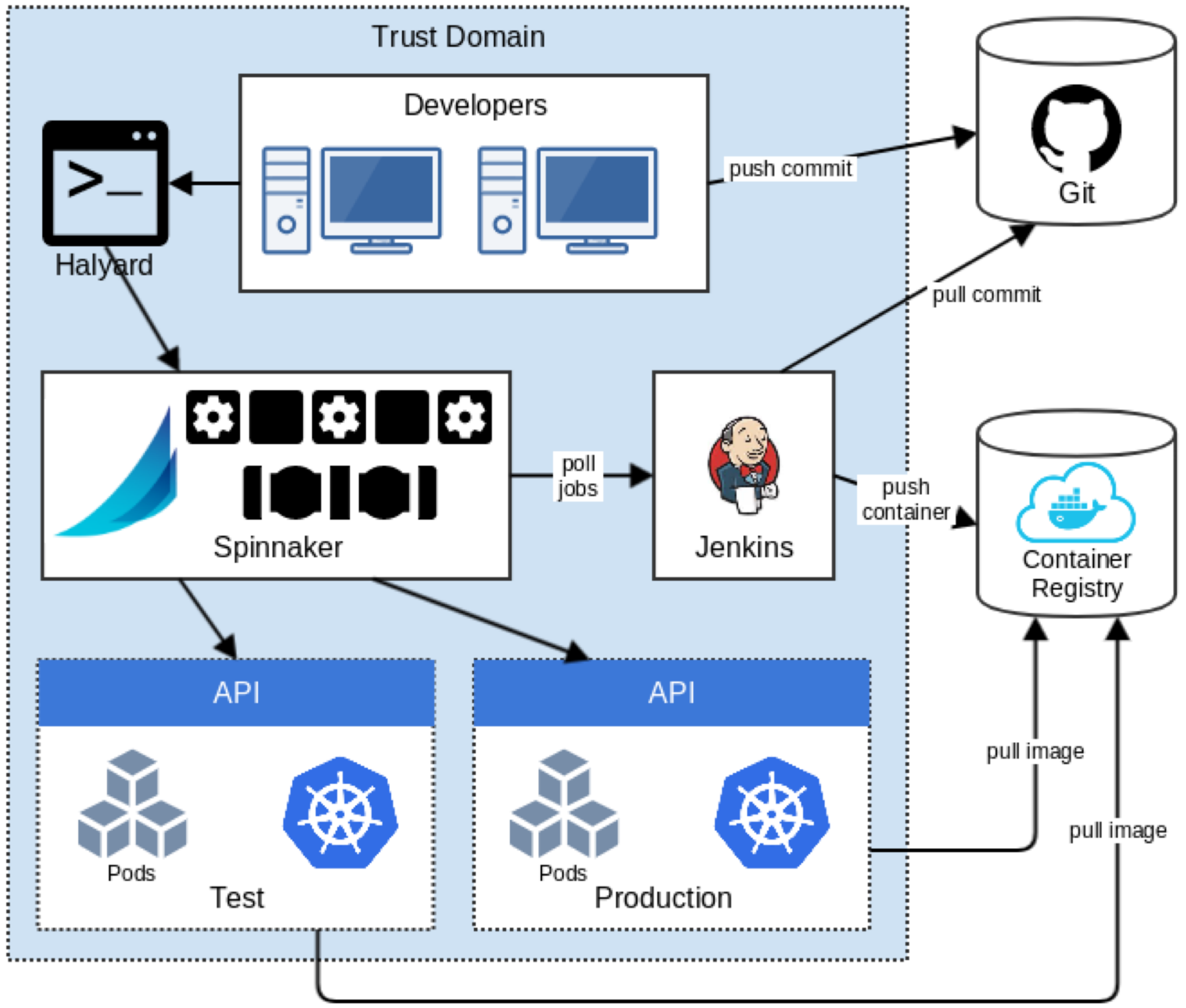
Use when distributing software to other parties.



SPINNAKER.

CONCLUSION:

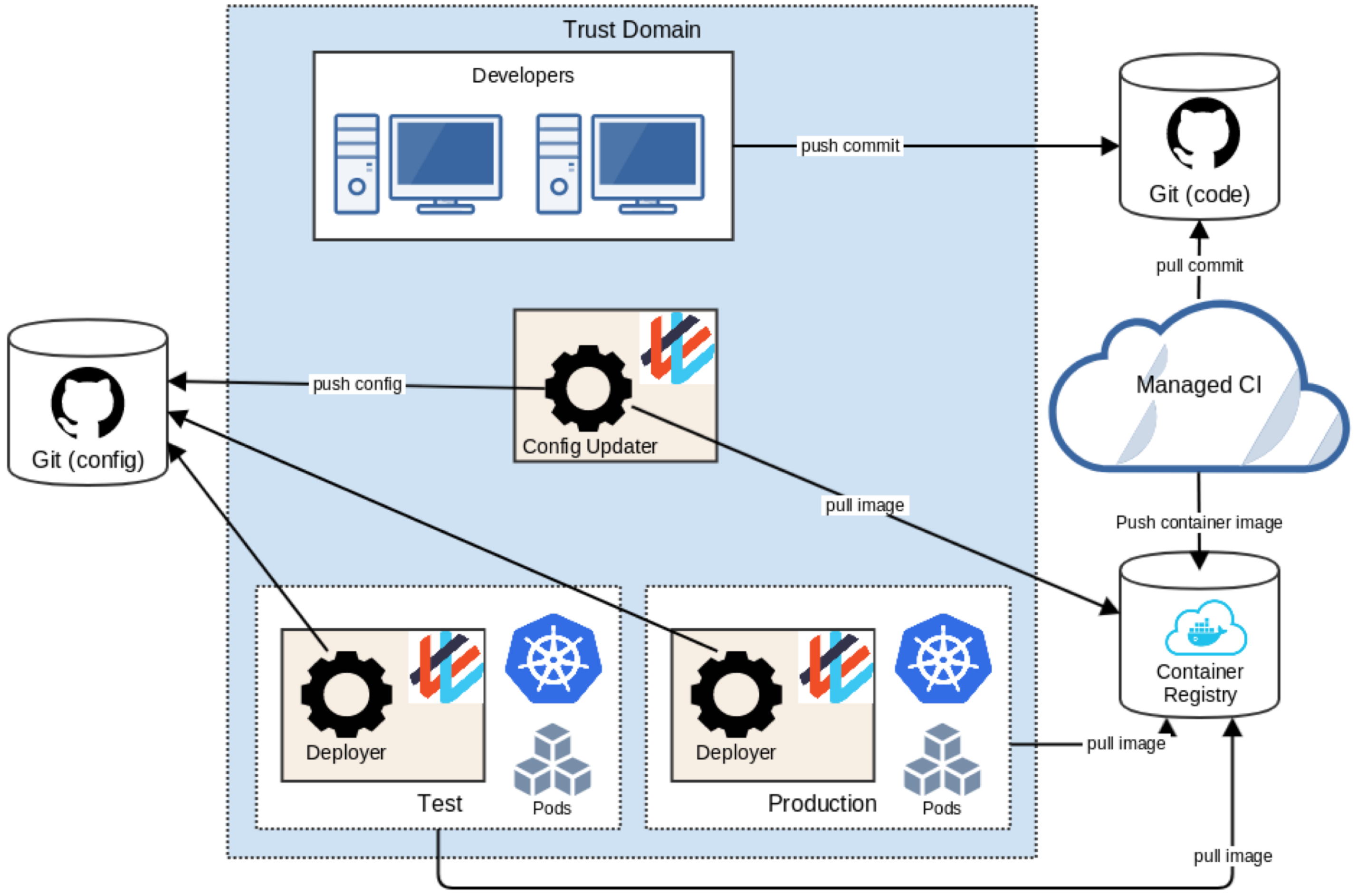
Use if you have specialised deployment requirements



WEAVE FLUX.

CONCLUSION:

Use if Gitops approach is important to you, or you are invested in the Weave Cloud platform.



CONFIGURATION IN GIT?

- Source of truth for application configuration
- Version numbers?
  - When releasing frequently?
- History and rollback

WHAT'S NEXT?

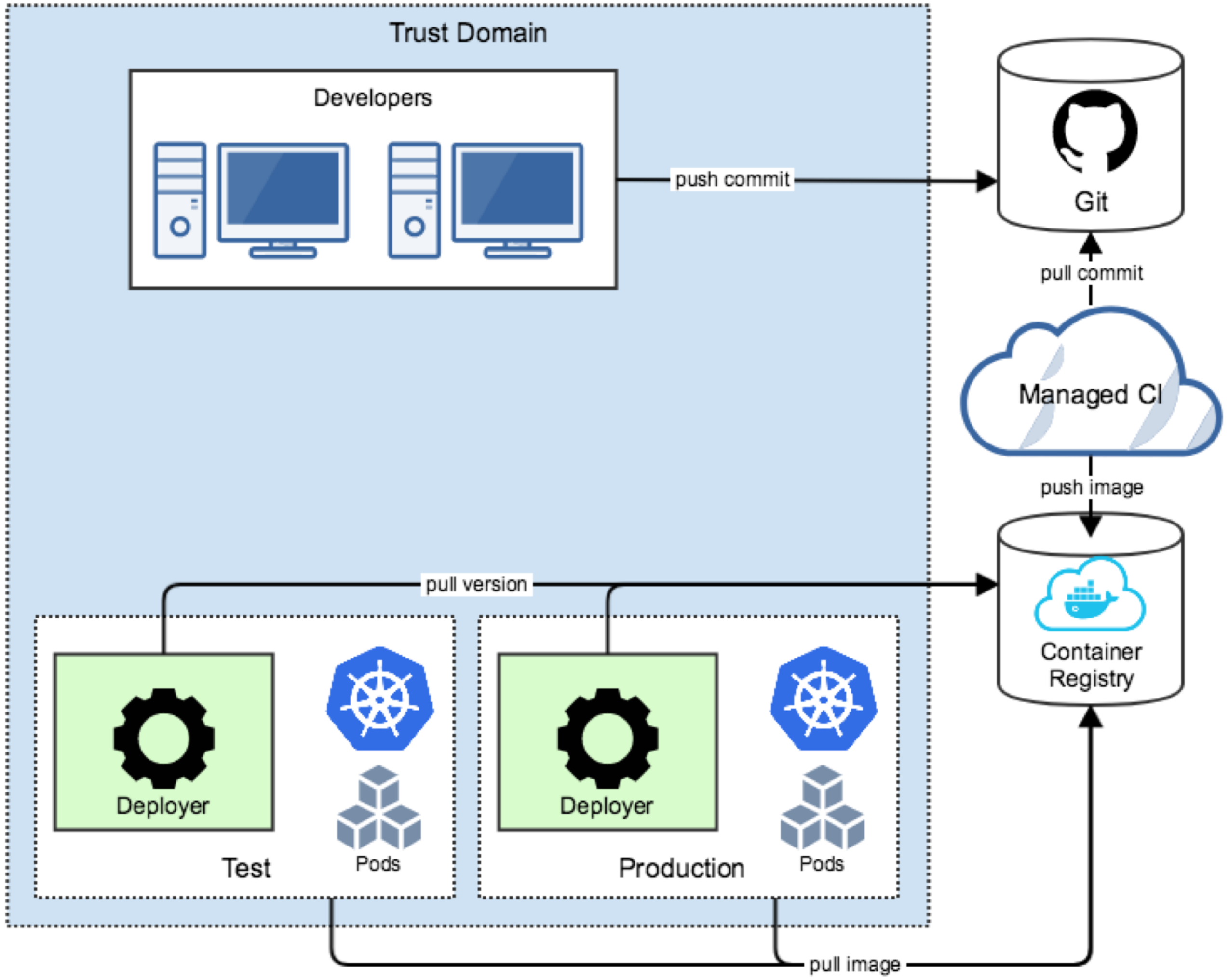
nearmap

CD-LITE.

A simplified approach to Continuous Delivery

- BYO continuous integration tool
- No additional infrastructure
- Build on existing Kubernetes concepts
- Support full automation
- Support best practices
  - Secure environments
  - Blue-green deployments
  - Version history and rollback
  - Instrumentation/visibility

SIMPLER PIPELINE.



ADVANTAGES.

- **Don’t need access to additional resources**
  - e.g. don’t need git access
- **Doesn’t require a separate config repo**
- **Simplified configuration**
  - Exists alongside application code
- **Easy to setup and manage**

CD-LITE: CONTAINER VERSION MANAGER.

Solve these challenges by using intrinsic Kubernetes principles and native abstractions

+ ContainerVersion
Custom Resource Definition (CRD)

+ Custom Controllers

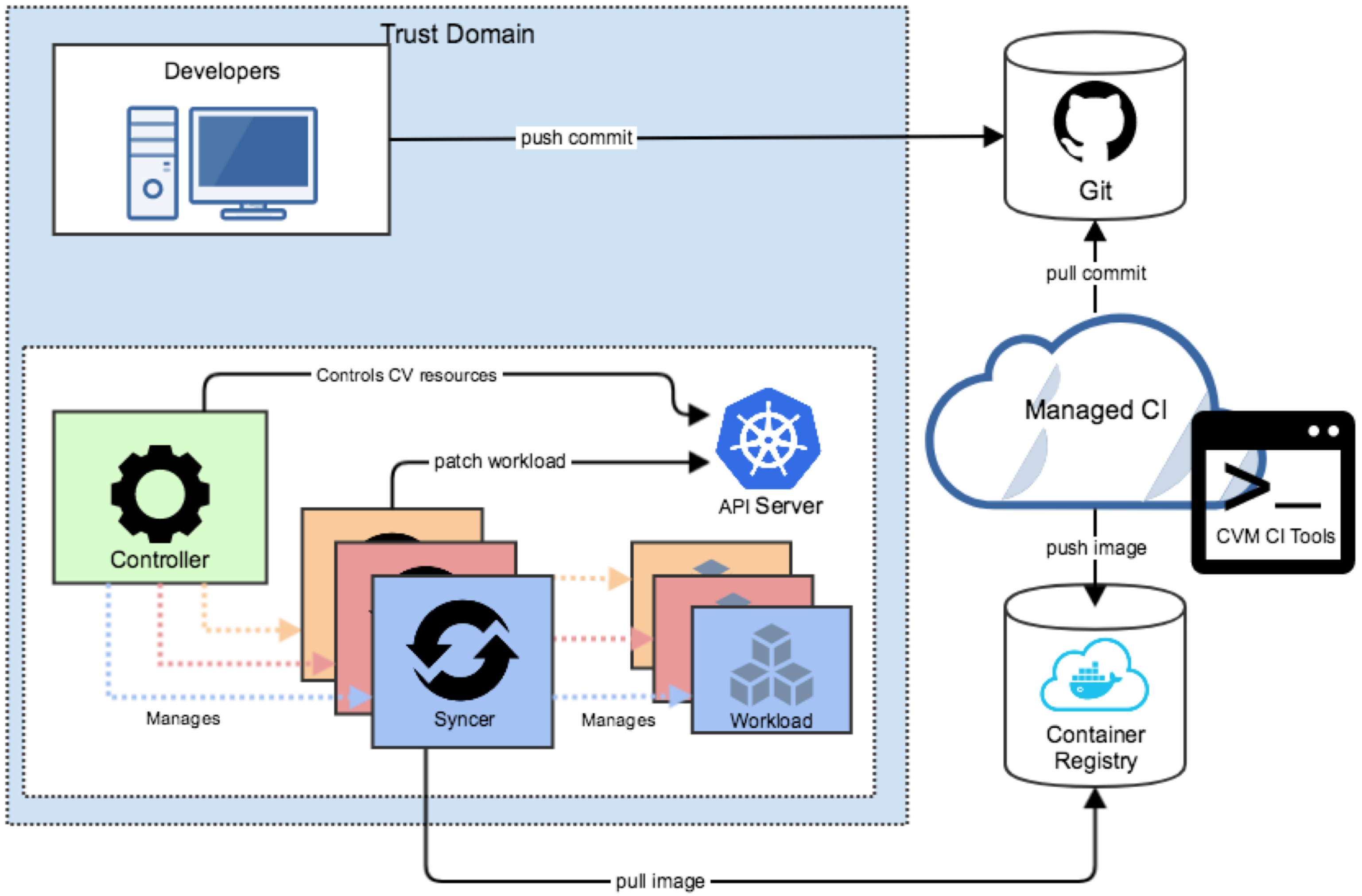
CONTAINER VERSION DECLARATION.

Defines rules for managing container versions

```yaml
kind: ContainerVersion
apiVersion: custom.k8s.io/v1
metadata:
  name: myappcv
spec:
  imageRepo: <AWS_ACC_ID>.dkr.ecr.us-east-1.amazonaws.com/nearmap/cvm-example
  tag: demo
  pollIntervalSeconds: 300
  selector:
    cvapp: myapp
  container:
    name: myapp
```

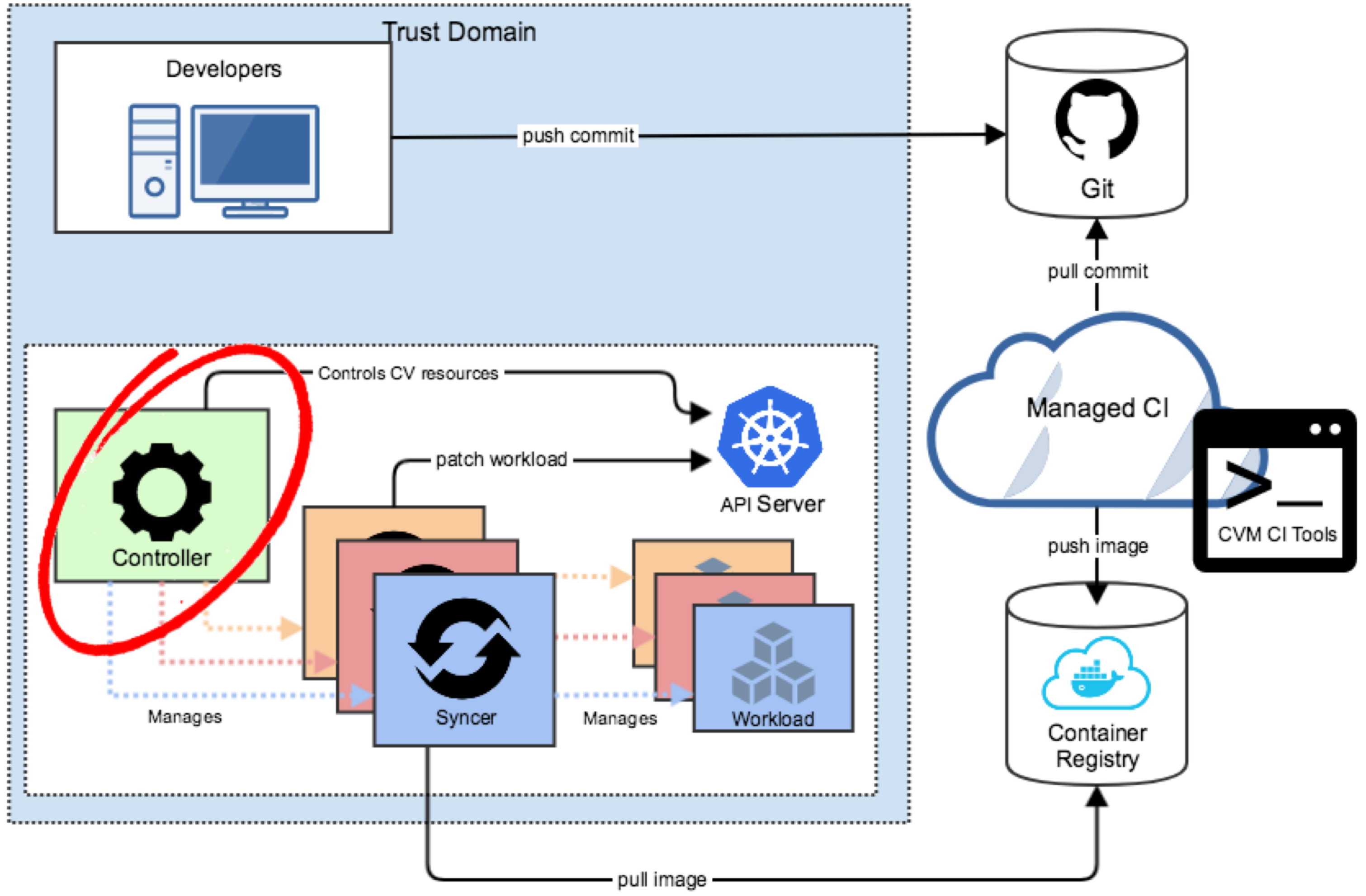
ARCHITECTURE.

- CV Controller
- CR Syncer
- CI Tools



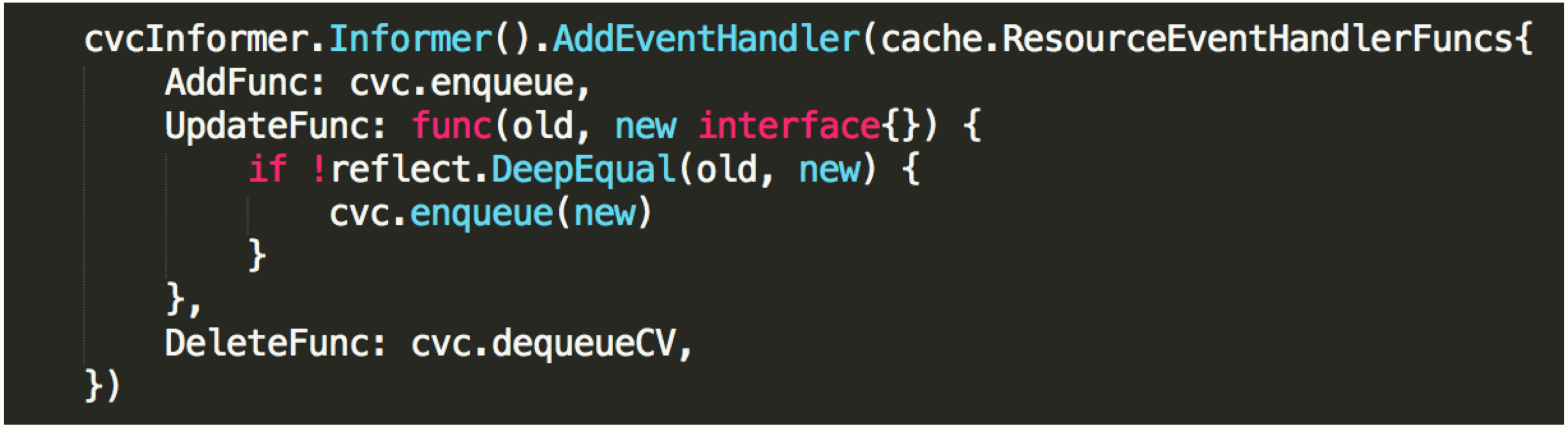
CV CONTROLLER.

ROLE: Reacts to changes in CV resources



CV CONTROLLER.

ROLE: Reacts to changes in CV resources

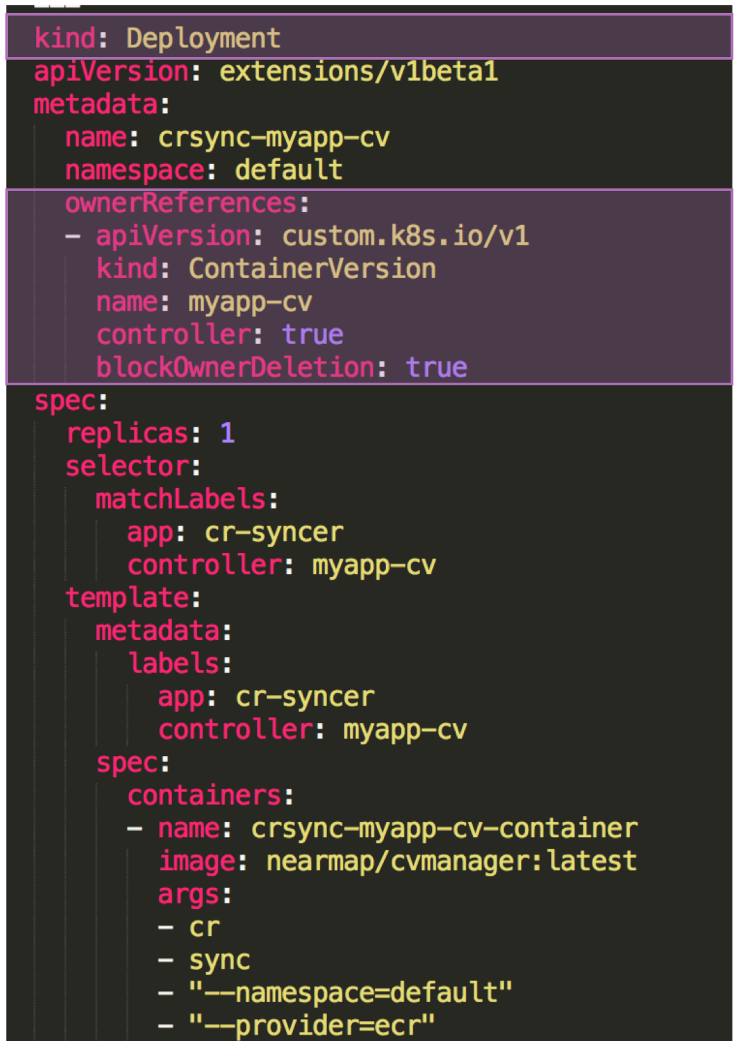


```
cvcInformer.Informer().AddEventHandler(cache.ResourceEventHandlerFuncs{
	AddFunc: cvc.enqueue,
	UpdateFunc: func(old, new interface{}) {
		if !reflect.DeepEqual(old, new) {
			cvc.enqueue(new)
		}
	},
	DeleteFunc: cvc.dequeueCV,
})
```

CV CONTROLLER.

ROLE: Reacts to changes in CV resources

- Creates and updates CR Syncers

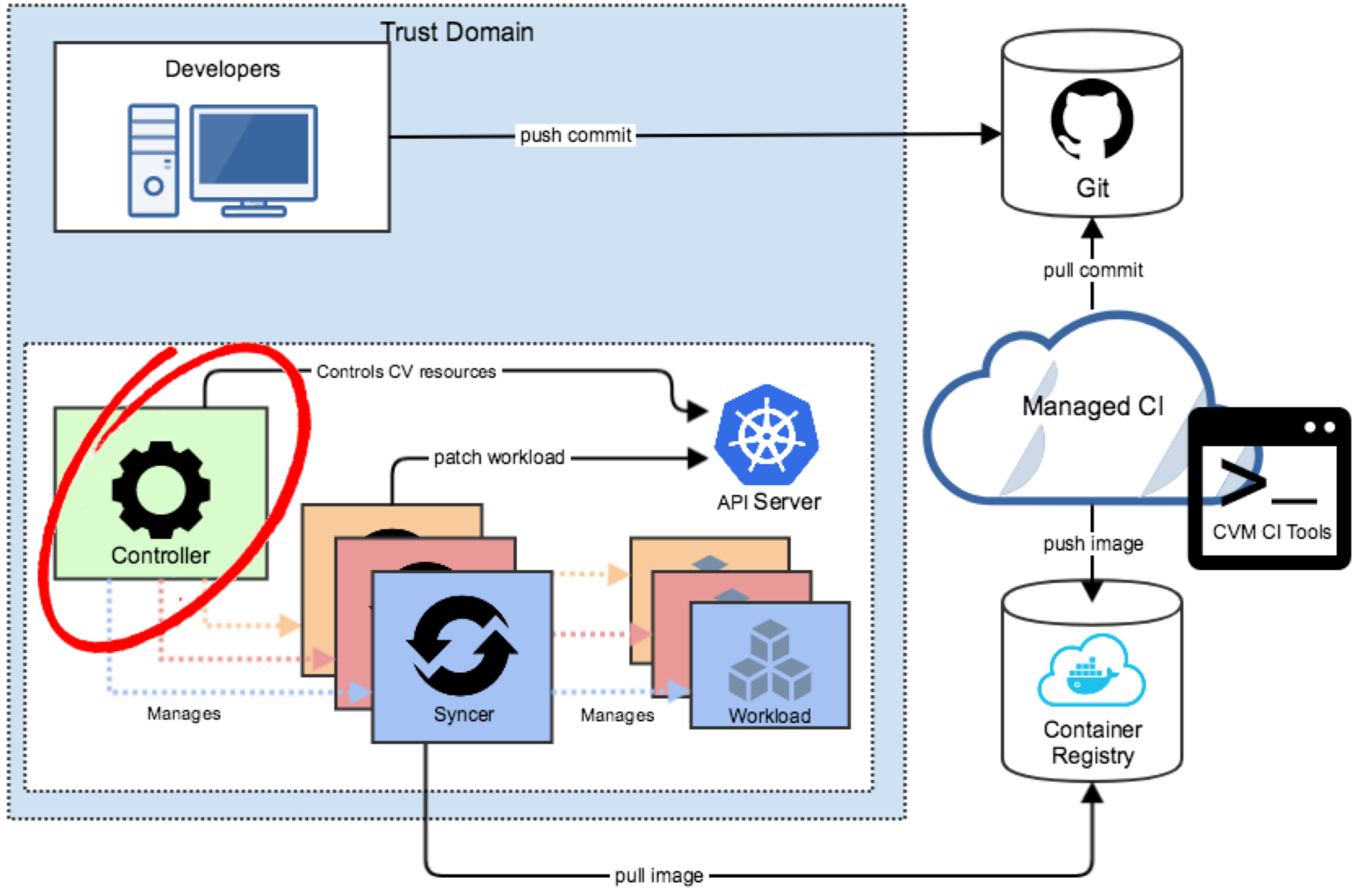


```yaml
kind: Deployment
apiVersion: extensions/v1beta1
metadata:
  name: crsync-myapp-cv
  namespace: default
  ownerReferences:
  - apiVersion: custom.k8s.io/v1
    kind: ContainerVersion
    name: myapp-cv
    controller: true
    blockOwnerDeletion: true
spec:
  replicas: 1
  selector:
    matchLabels:
      app: cr-syncer
      controller: myapp-cv
  template:
    metadata:
      labels:
        app: cr-syncer
        controller: myapp-cv
    spec:
      containers:
      - name: crsync-myapp-cv-container
        image: nearmap/cvmanager:latest
        args:
        - cr
        - sync
        - "--namespace=default"
        - "--provider=ecr"
```

CV CONTROLLER.

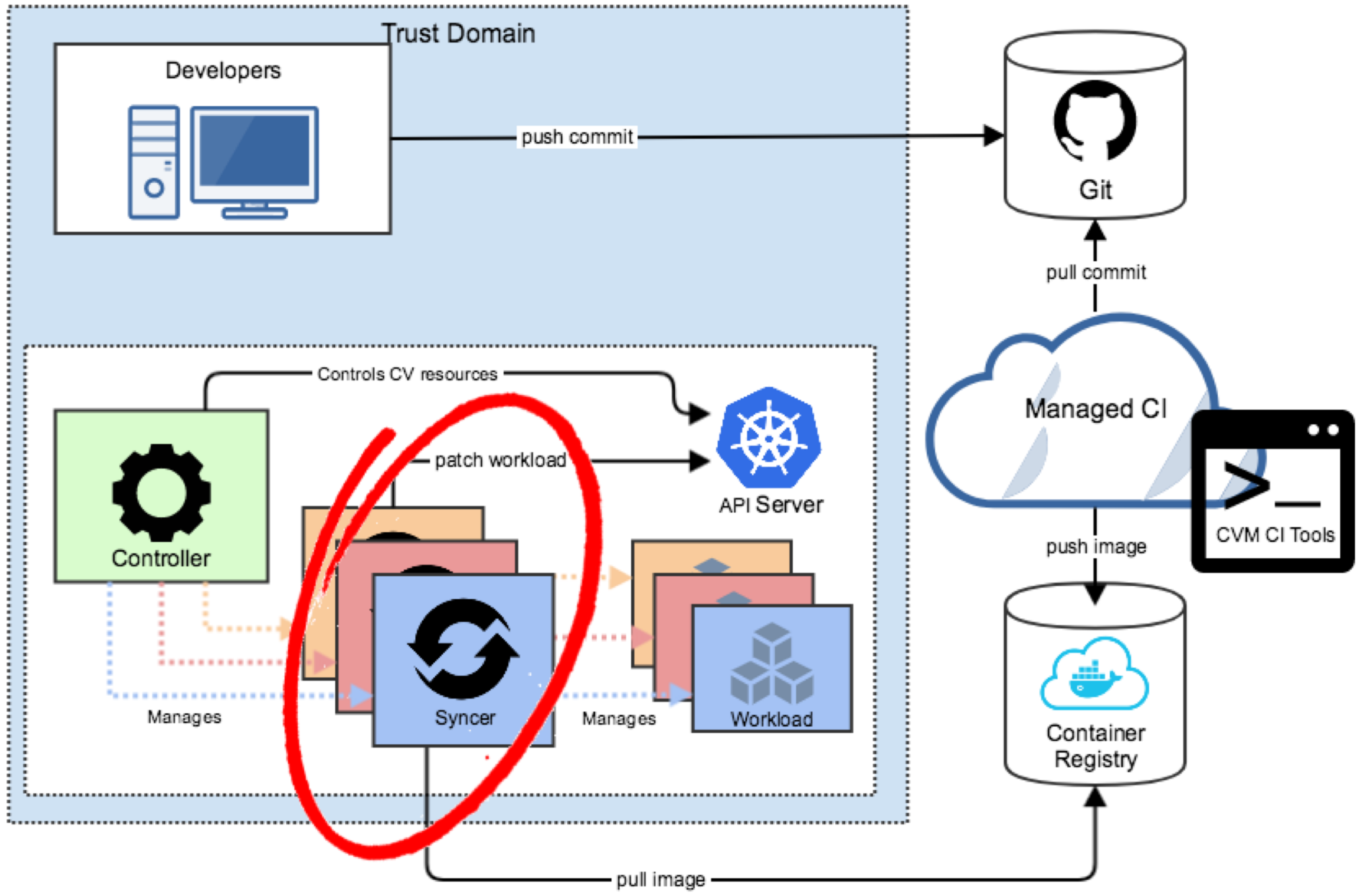
ROLE: Reacts to changes in CV resources

- Creates and updates CR Syncer per CV resources
- Provide visibility on version updates



CR SYNCERS.

ROLE: To ensure container state, as declared by the CV resource, is met.



CR SYNCERS.

Periodically syncs with registry to check for changes in desired container version

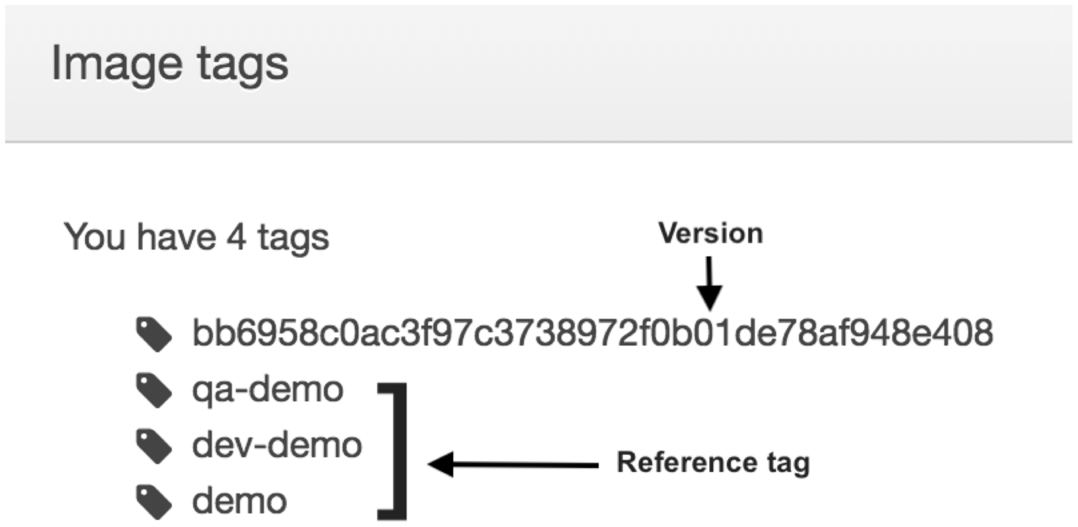
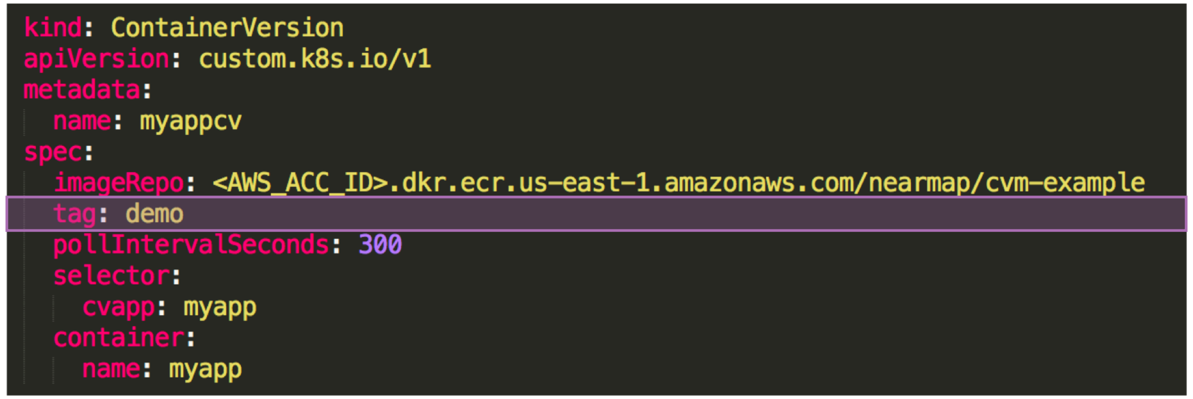


Image tags

You have 4 tags

- bb6958c0ac3f97c3738972f0b01de78af948e408
- qa-demo
- dev-demo
- demo



```
kind: ContainerVersion
apiVersion: custom.k8s.io/v1
metadata:
  name: myappcv
spec:
  imageRepo: <AWS_ACC_ID>.dkr.ecr.us-east-1.amazonaws.com/nearmap/cvm-example
  tag: demo
  pollIntervalSeconds: 300
  selector:
    cvapp: myapp
  container:
    name: myapp
```

CR SYNCER: VALIDATION

- Regression check
- Quality checks
- Container vulnerability scan
- Image signature

CR SYNCER: ROLLOUTS

- **Patches PodSpec of matched workload to trigger the rollout**
  - StrategicMerge

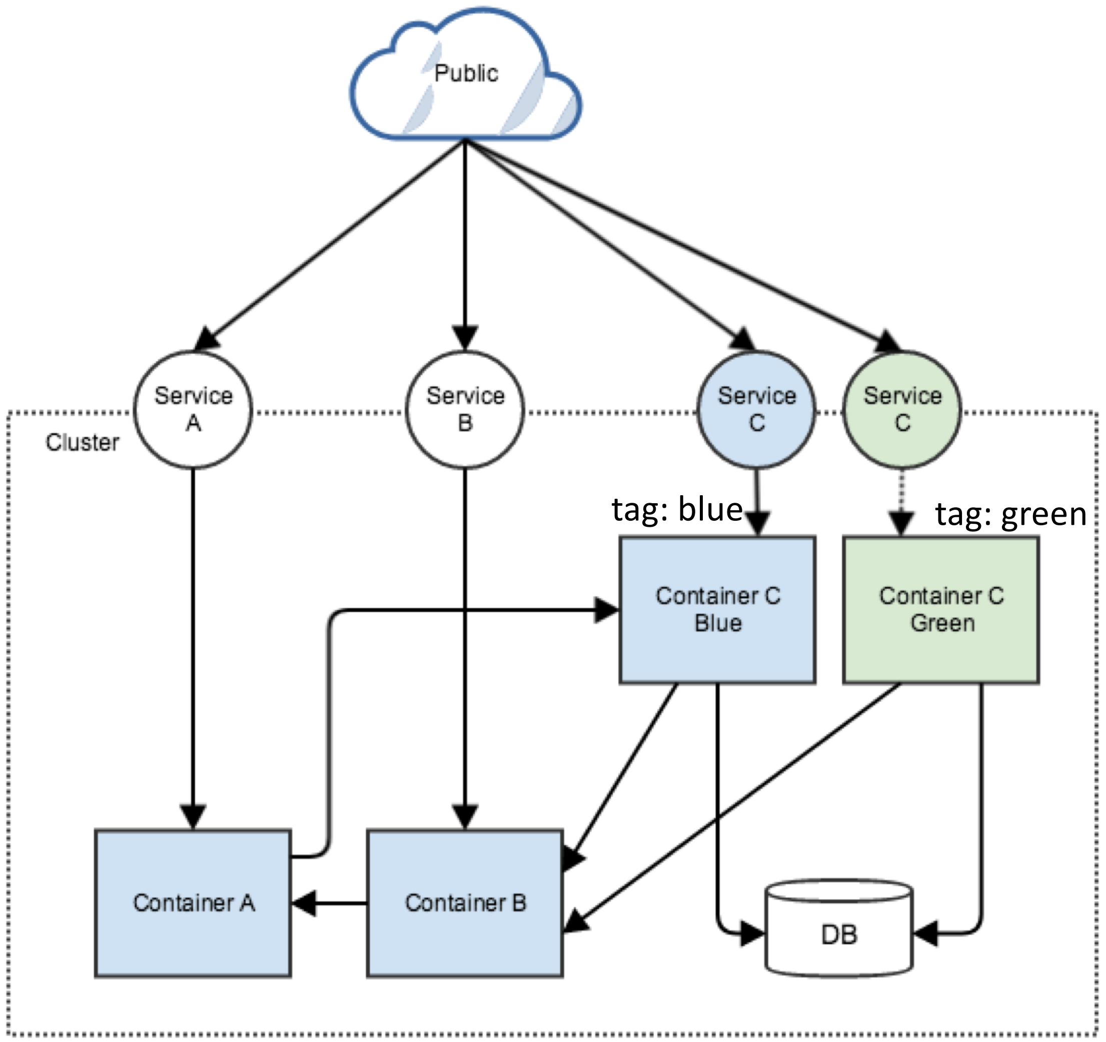
- **Using native strategy**
  - RollingUpdate
  - Recreate
  - OnDelete

CR SYNCER: COMPLEX ROLLOUTS

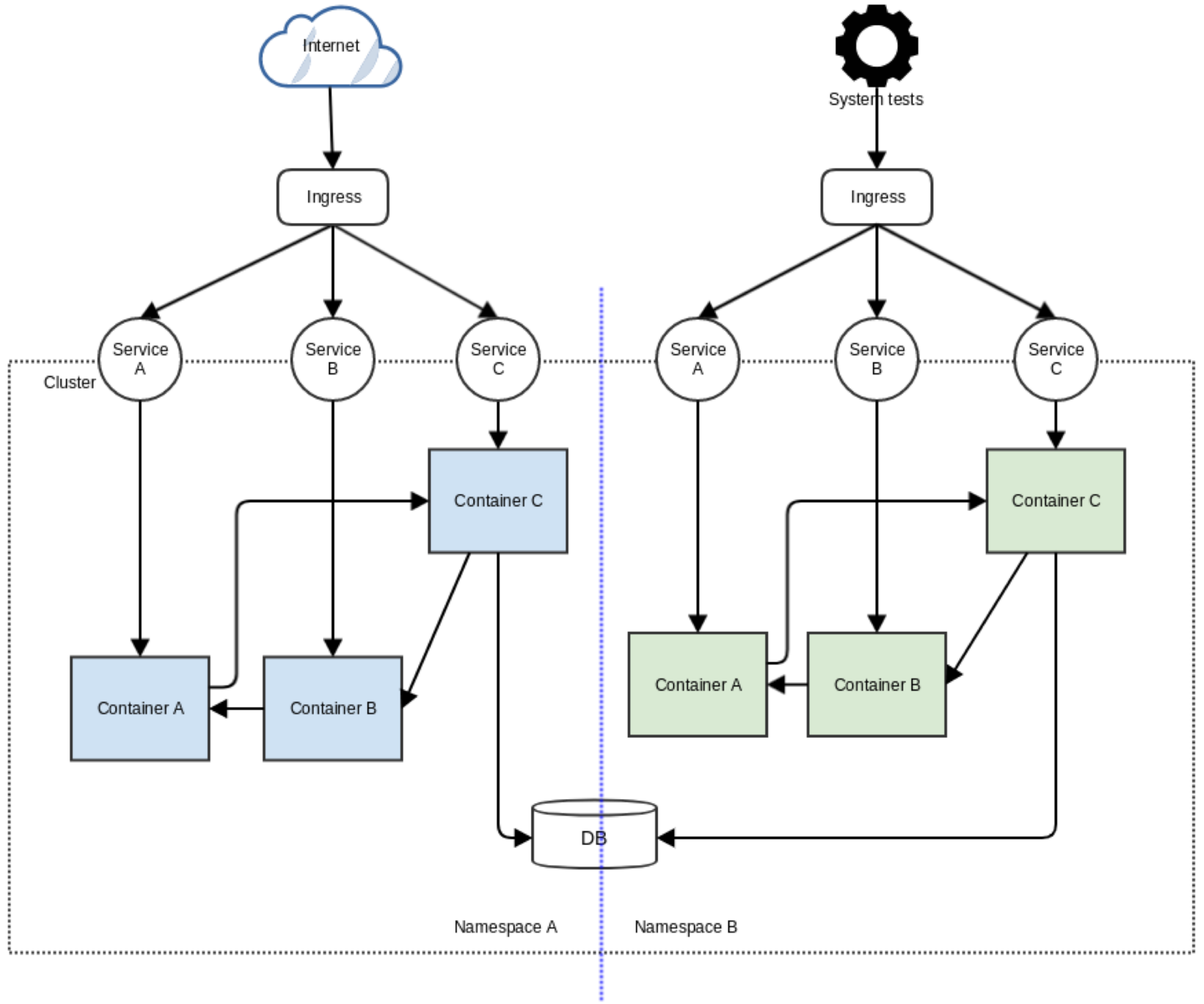
- **Using non-native strategy**
  - Canary
  - Blue Green deploys

- **May requires manual intervention**
  - WIP to allow automatic blue green rollouts

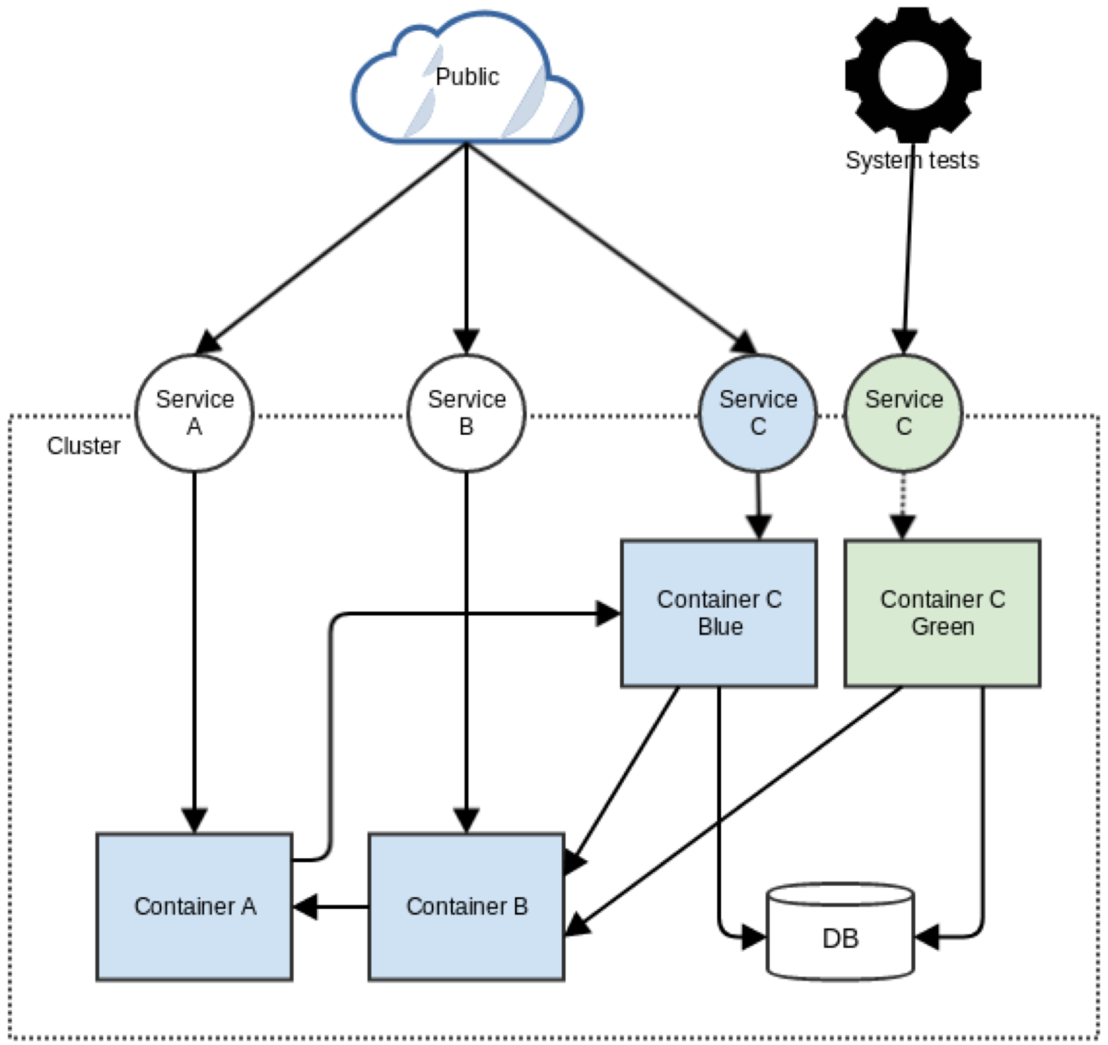
CANARY DEPLOYMENT.



BLUE GREEN NAMESPACE DEPLOYMENT.



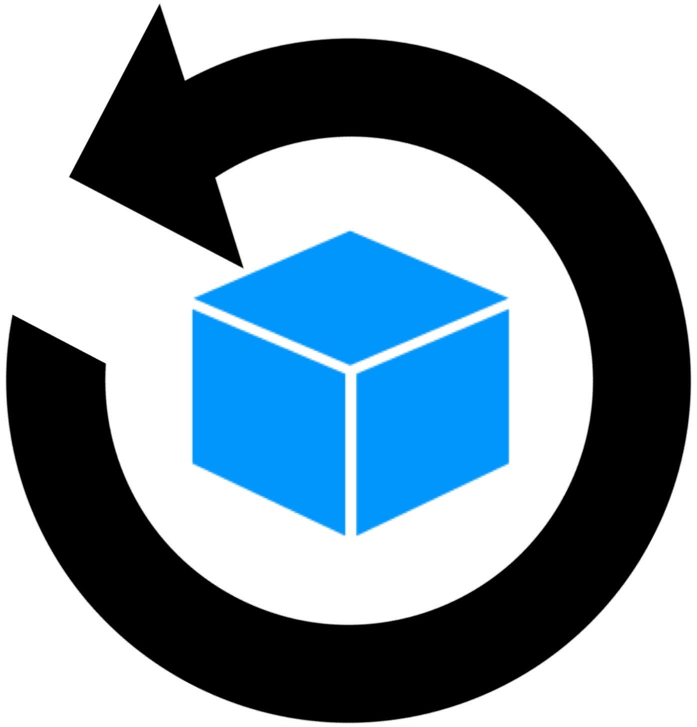
BLUE GREEN SERVICE DEPLOYMENT.



nearmap

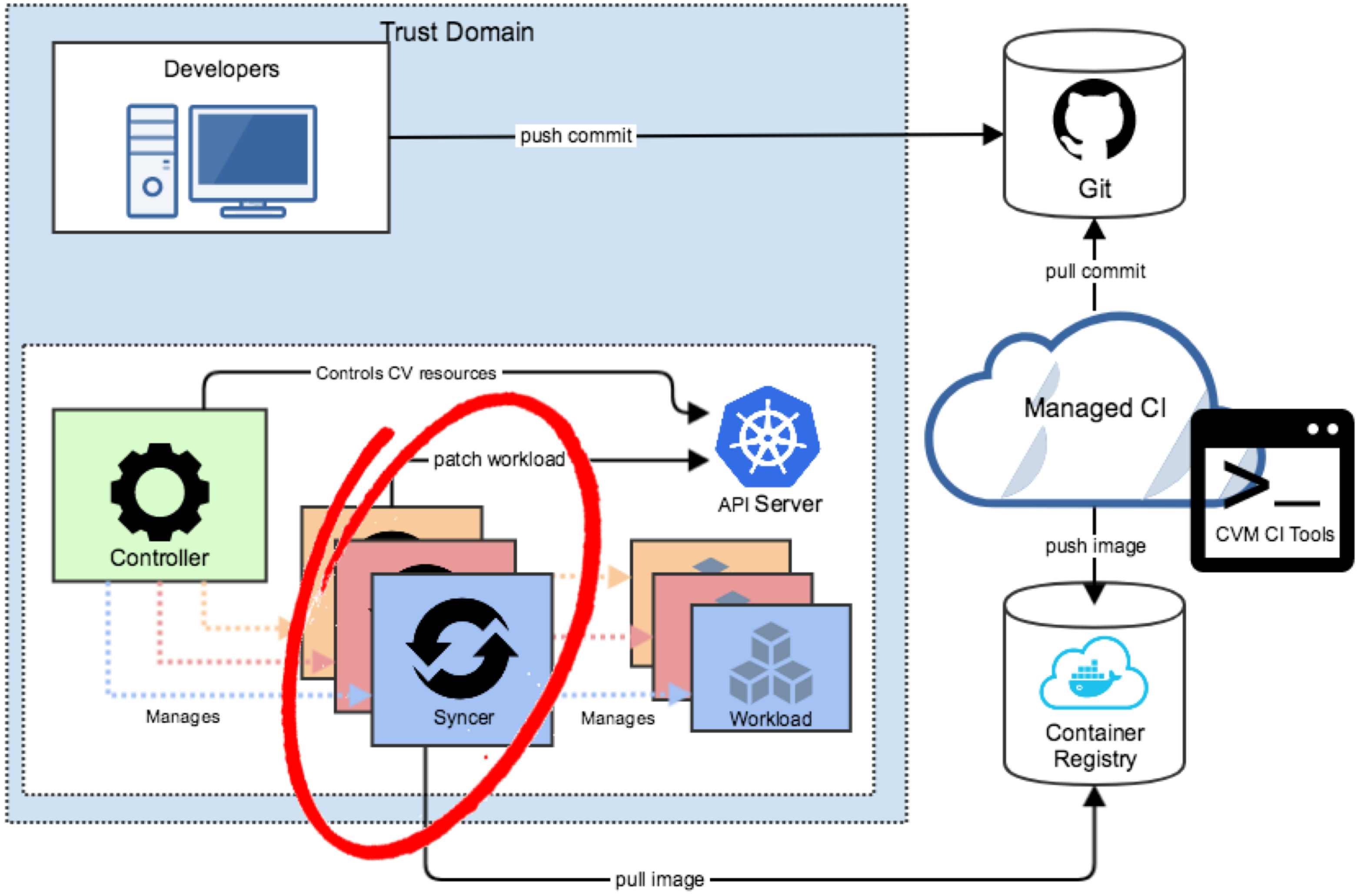
CR SYNCER: ROLLBACK

- **Roll-forward is encouraged**
  - Proportional scaling
- **Optional rollback**
- **Failed deployment triggers event/notification**

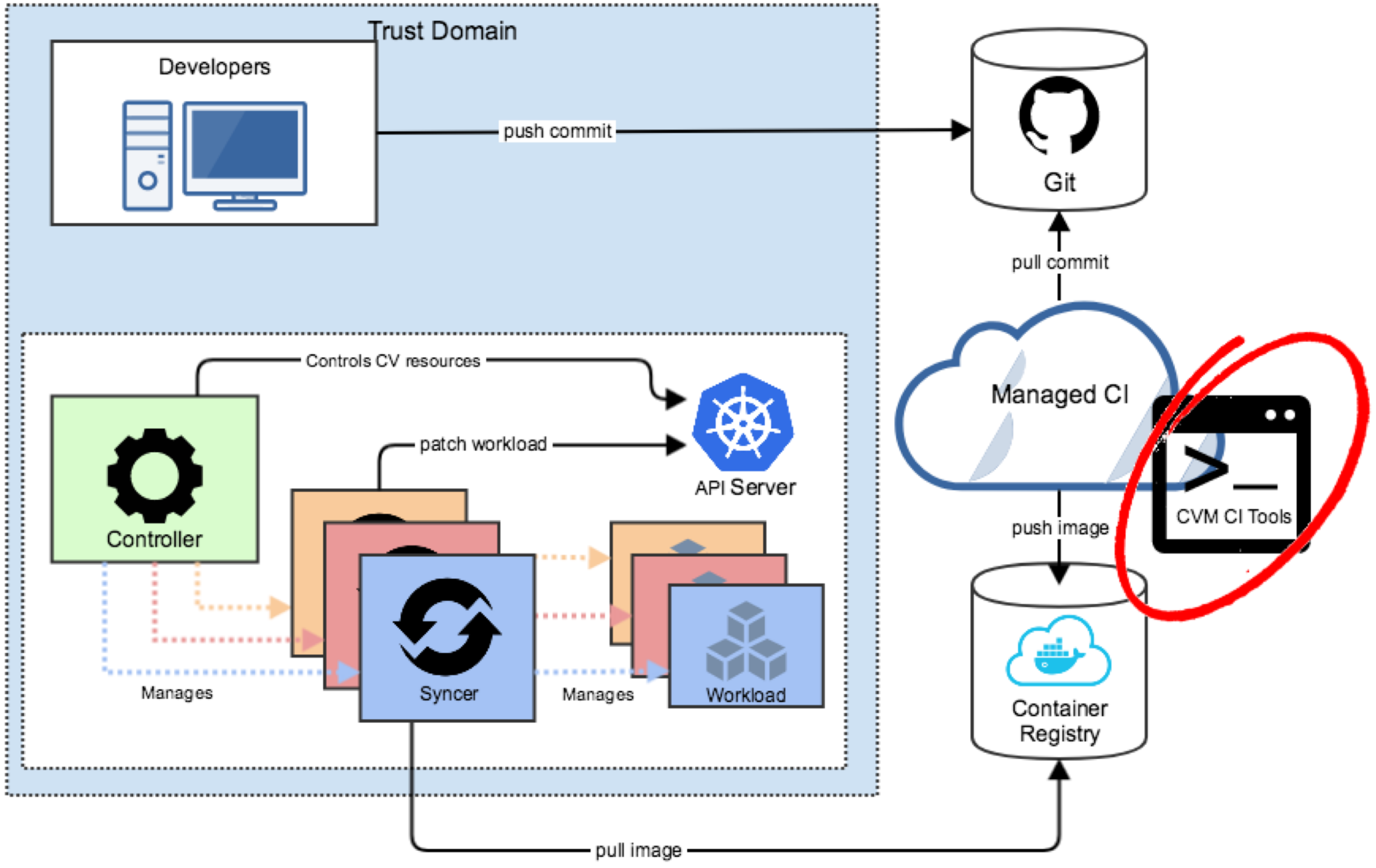


nearmap

CR SYNCERS.

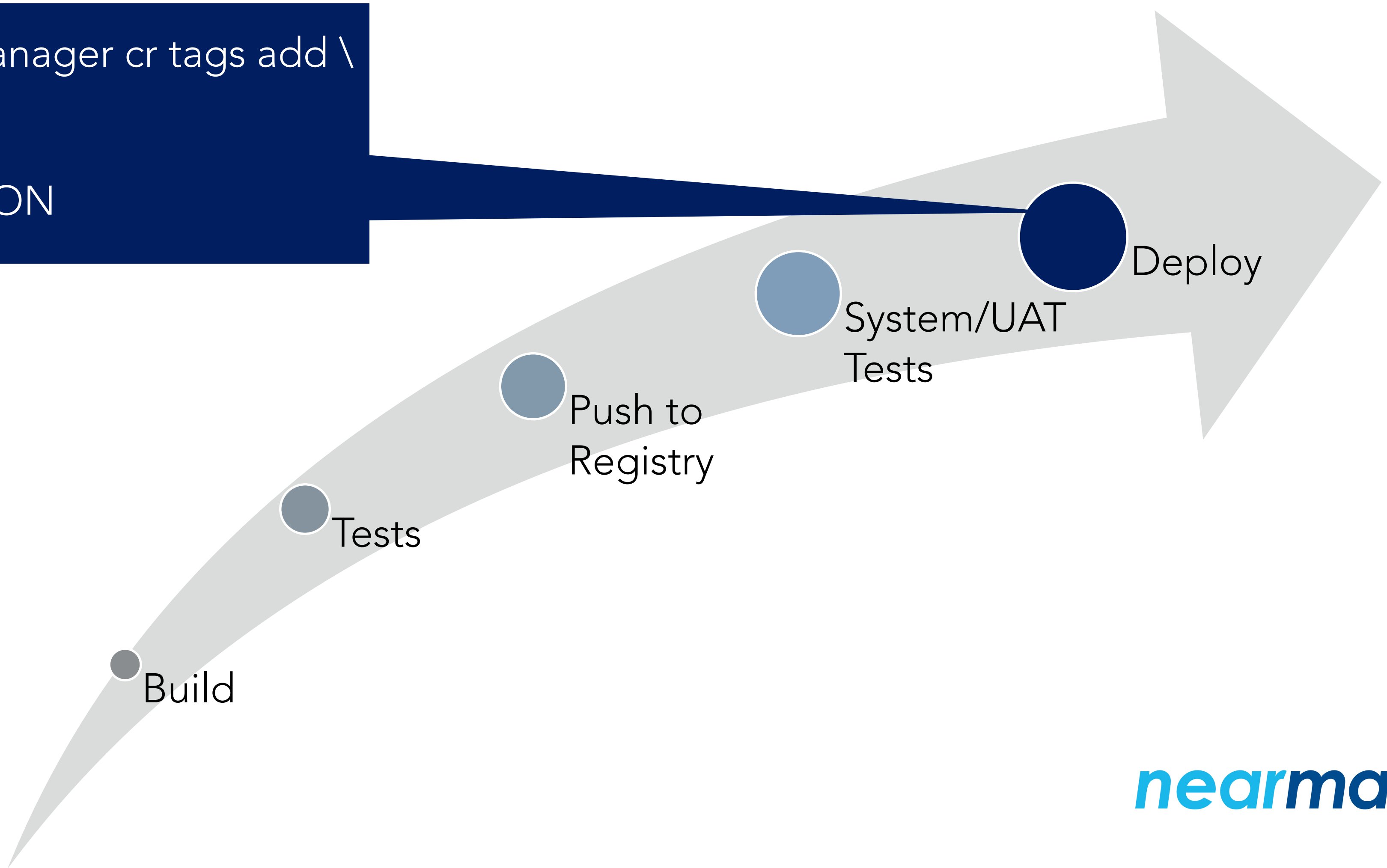

Trust Domain
Developers
push commit
Git
pull commit
Managed CI
CVM CI Tools
push image
Container Registry
Controls CV resources
patch workload
API Server
Controller
Syncer
Workload
Manages
Manages
pull image

CI TOOLS.


Trust Domain
Developers
push commit
Git
pull commit
Managed CI
CVM CI Tools
push image
Container Registry
Controls CV resources
patch workload
API Server
Controller
Syncer
Workload
Manages
Manages
pull image

CI TOOLS.

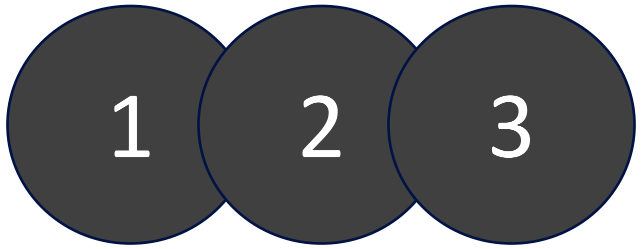
```
docker run nearmap/cvmanager cr tags add \
            --repo $REPO \
            --tags $TAG \
            --version $VERSION
```



EASE OF USE.

- **One time install only**
  - In 3 easy steps

1. Install CVManager
2. Define CV resource for workload
3. Integrate with CI



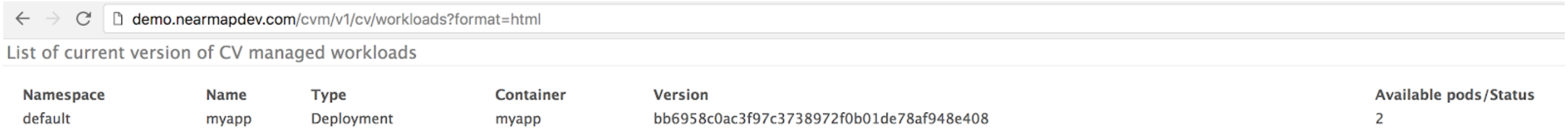
BENEFITS OF CVM.

"Self managed self healing CI/CD pipeline"

BENEFITS OF CVM.

Separation Of Concerns
Instrumentation
Kubernetes
Open Source
Native Solution
Blue Green
Instrumentation
Simple Configuration
Best Practices
Low Maintenance
Simple Configuration
Native Solution
Roll Forward
Simple Configuration
Visibility
Reduce Security Risk
Roll Forward
Reduce Security Risk
Release History
Visibility
Cluster Is Responsible For Managing Its Own Versions
Roll Forward
Separation Of Concerns
Cluster Is Responsible For Managing Its Own Versions
Rollback
Rollback
No Additional Infrastructure
Self Managed
Self Managed
Version History
Best Practices
No Additional Infrastructure
Monitoring
Blue Green
Dashboard And Metrics
Cluster Is Responsible For Managing Its Own Versions
Best Practices
Reduce Security Risk
Open Source
Monitoring
Dashboard And Metrics
Native Solution
Security
Self Managed
Visibility
Release History
Low Maintenance
Monitoring
Reduce Security Risk
Version History
No Additional Infrastructure
Security
Low Maintenance
Kubernetes
Rollback
Kubernetes
Blue Green
Low Maintenance
Open Source
Dashboard And Metrics
Self Managed
Separation Of Concerns
Release History
Security
Instrumentation
Version History

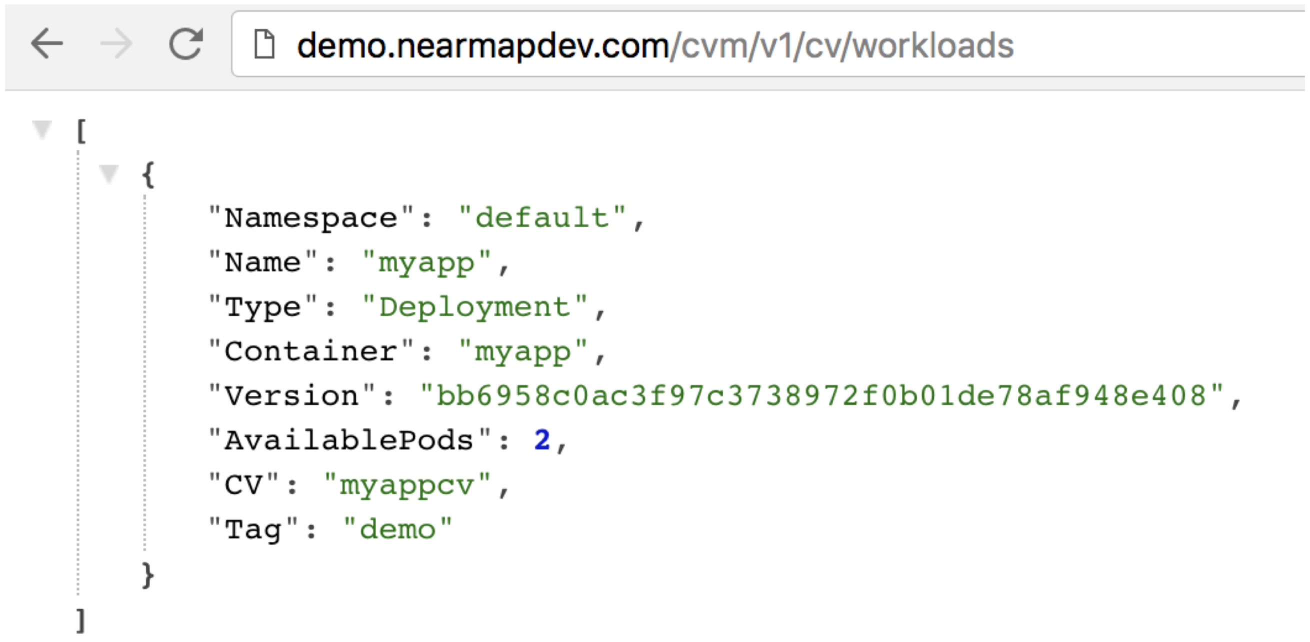
VISIBILITY: CURRENT VERSION



demo.nearmapdev.com/cvm/v1/cv/workloads?format=html

List of current version of CV managed workloads

Namespace	Name	Type	Container	Version	Available pods/Status
default	myapp	Deployment	myapp	bb6958c0ac3f97c3738972f0b01de78af948e408	2

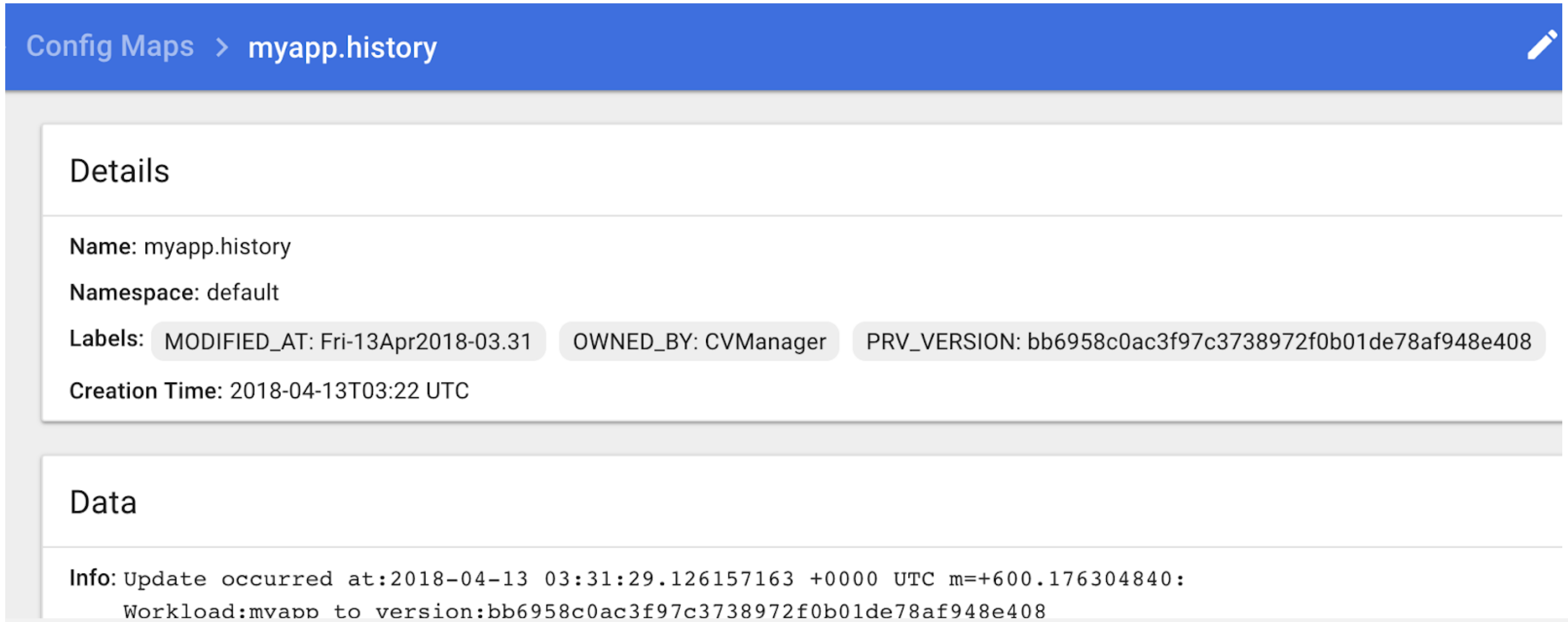


demo.nearmapdev.com/cvm/v1/cv/workloads

```
[
  {
    "Namespace": "default",
    "Name": "myapp",
    "Type": "Deployment",
    "Container": "myapp",
    "Version": "bb6958c0ac3f97c3738972f0b01de78af948e408",
    "AvailablePods": 2,
    "CV": "myappcv",
    "Tag": "demo"
  }
]
```

VISIBILITY: RELEASE HISTORY

- Opt-in
- Captured in configmap
- Exposed on REST



Config Maps > myapp.history

Details

Name: myapp.history

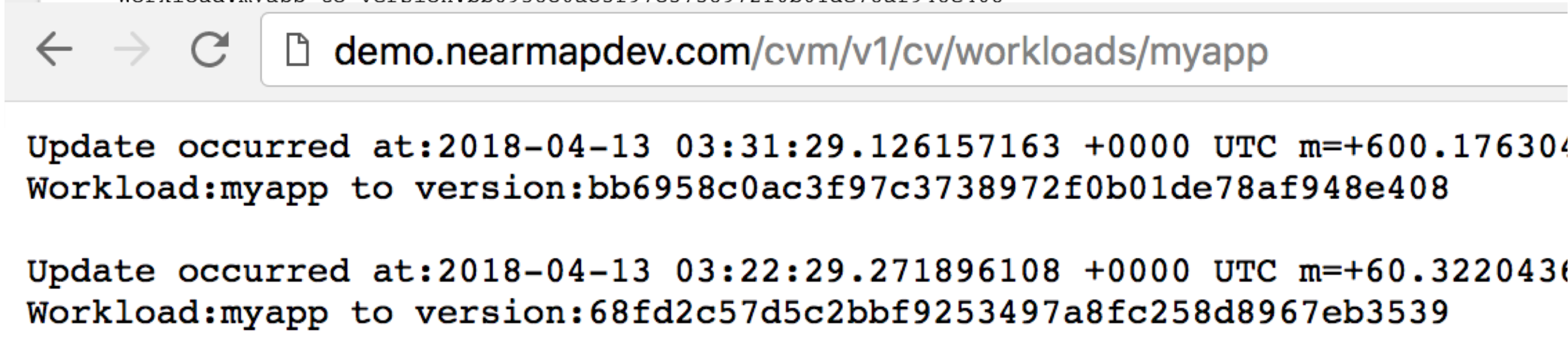
Namespace: default

Labels: MODIFIED_AT: Fri-13Apr2018-03.31 OWNED_BY: CVManager PRV_VERSION: bb6958c0ac3f97c3738972f0b01de78af948e408

Creation Time: 2018-04-13T03:22 UTC

Data

```
Info: Update occurred at:2018-04-13 03:31:29.126157163 +0000 UTC m=+600.176304840:
    Workload:myapp to version:bb6958c0ac3f97c3738972f0b01de78af948e408
```



demo.nearmapdev.com/cvm/v1/cv/workloads/myapp

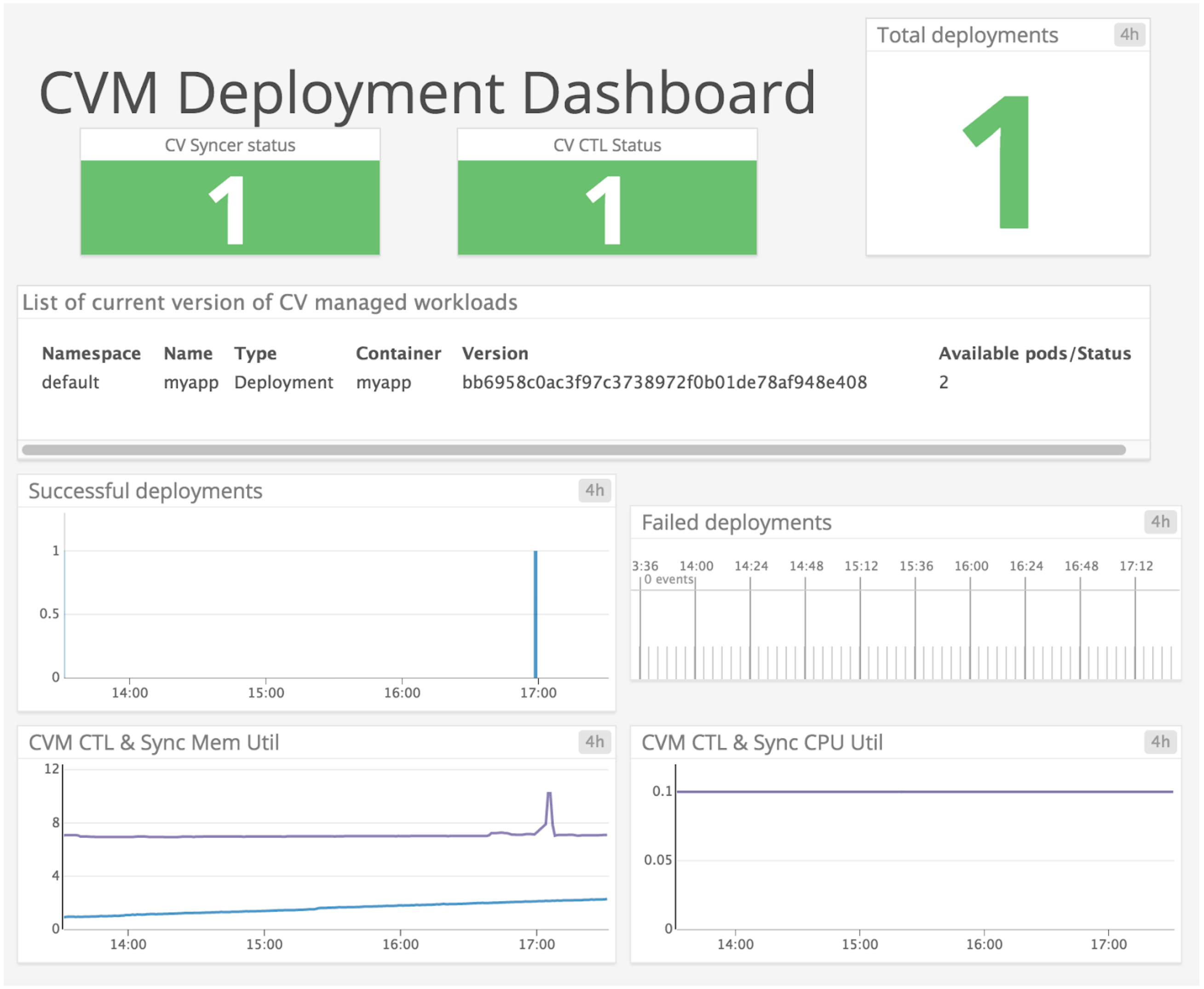
```
Update occurred at:2018-04-13 03:31:29.126157163 +0000 UTC m=+600.17630
Workload:myapp to version:bb6958c0ac3f97c3738972f0b01de78af948e408

Update occurred at:2018-04-13 03:22:29.271896108 +0000 UTC m=+60.322043
Workload:myapp to version:68fd2c57d5c2bbf9253497a8fc258d8967eb3539
```

MONITORING.

- **Captures stats**
  - Success
  - Failure
    - Registry
    - Bad config
    - Container not found
    - Rollback
- **Supports event notification and service check**

DEPLOYMENT DASHBOARD.



CVM Deployment Dashboard

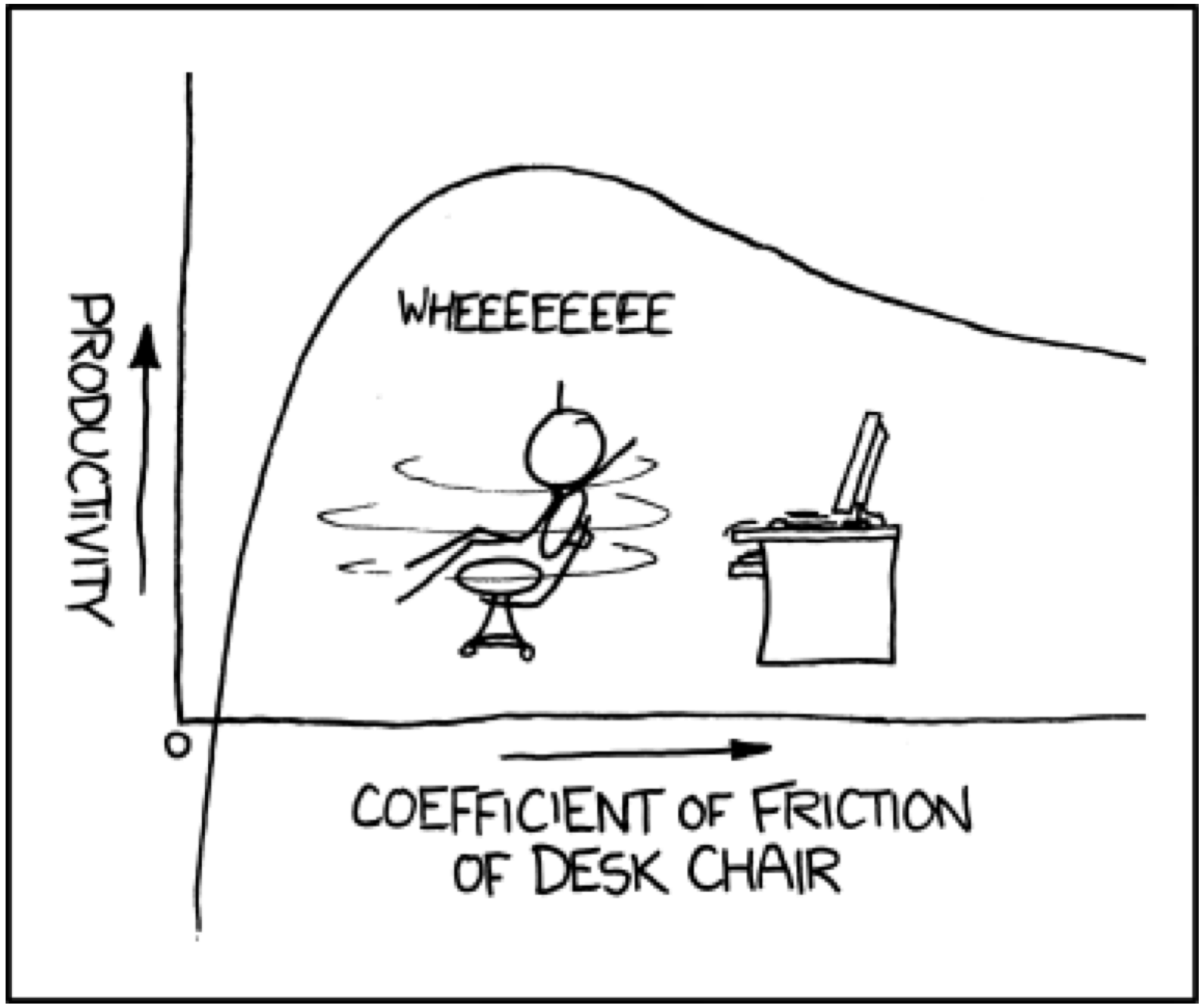
CV Syncer status: 1

CV CTL Status: 1

Total deployments (4h): 1

List of current version of CV managed workloads

Namespace	Name	Type	Container	Version	Available pods/Status
default	myapp	Deployment	myapp	bb6958c0ac3f97c3738972f0b01de78af948e408	2

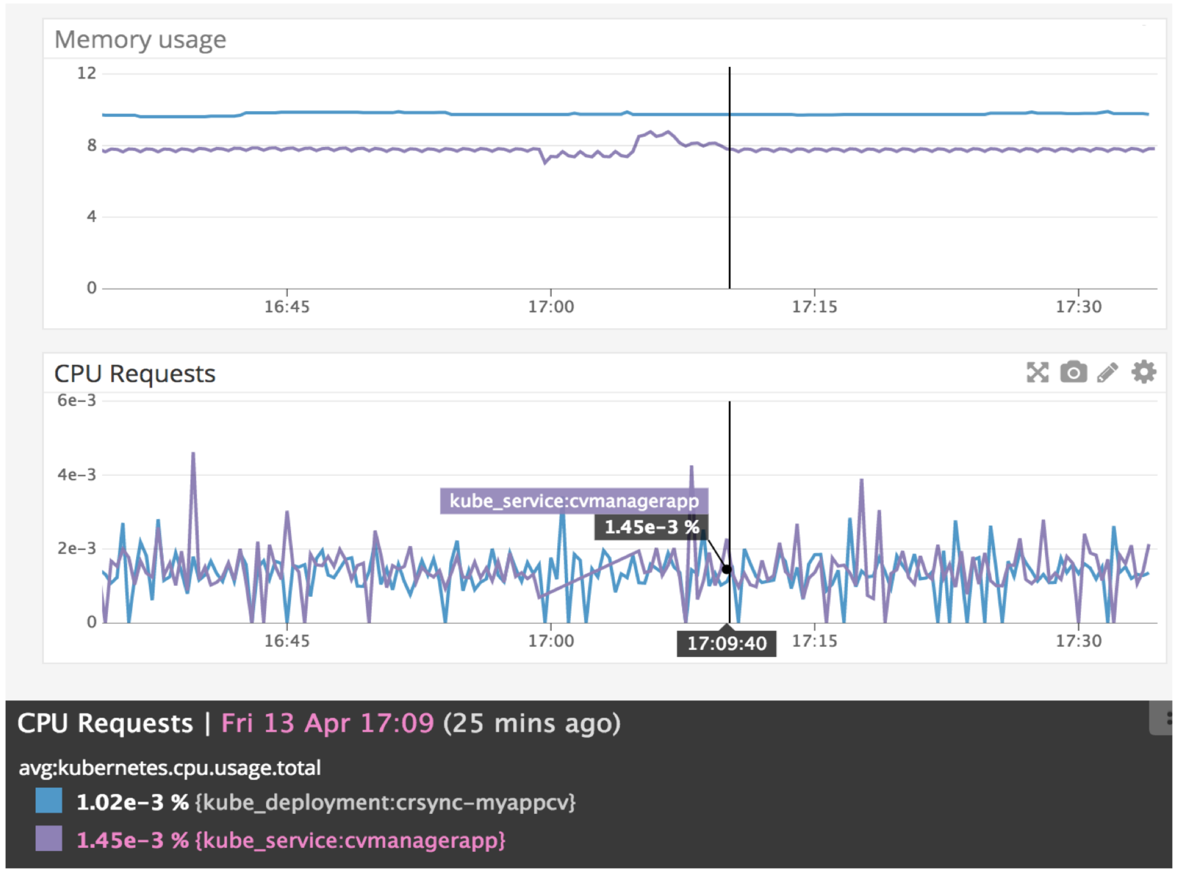


https://xkcd.com/815/

RESOURCE UTILIZATION.

Will this approach scale to demand?

CPU	≅ **0.001** Core
Memory	≅ **10** MiB

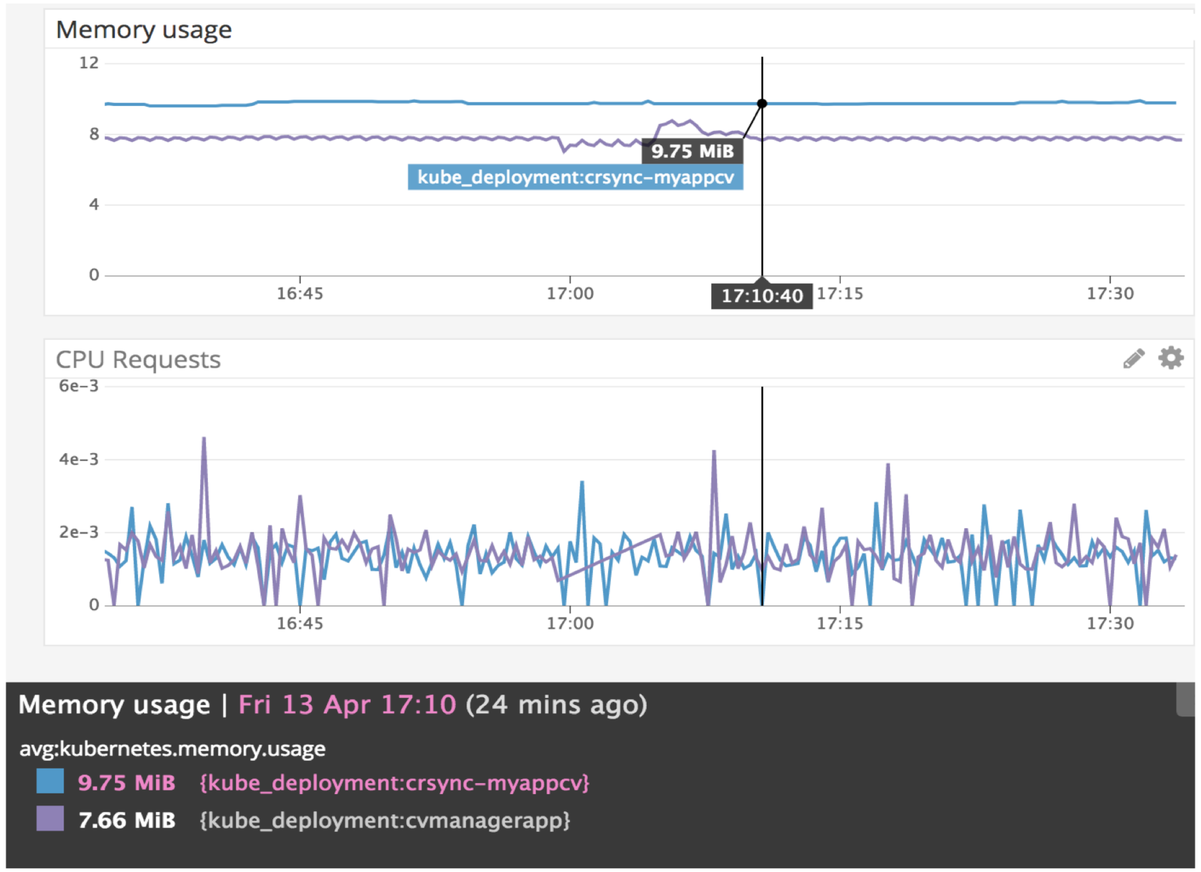


CPU Requests | Fri 13 Apr 17:09 (25 mins ago)

avg:kubernetes.cpu.usage.total

🟦 **1.02e-3 %** {kube_deployment:crsync-myappcv}

🟪 **1.45e-3 %** {kube_service:cvmanagerapp}



Memory usage | Fri 13 Apr 17:10 (24 mins ago)

avg:kubernetes.memory.usage

🟦 **9.75 MiB** {kube_deployment:crsync-myappcv}

🟪 **7.66 MiB** {kube_deployment:cvmanagerapp}

CVM BEST PRACTICES.

- Merge to master initiates deployments
- Use git hashes as version numbers
- Deployment Dashboard
- Automate tests

SIMPLE ENOUGH.

That CV-Manager updates itself

DEMO

CVManager is open-source, available under MIT license

https://github.com/nearmap/cvmanager/

Blog

https://nearmap.io/2018/04/cvmanager-intro/

Sample application

https://github.com/nearmap/cvm-example

THANK YOU!

Questions?