
AWS
re:Invent

NETFLIX aws

NFX302

Simplifying delivery as code with Spinnaker and Kubernetes

Nima Kaviani
Principal Solutions Architect
AWS

Rob Hilton
Principal Solutions Architect
AWS

Luis Pollo
Senior Software Engineer
Netflix

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Evolution of Kubernetes support in Spinnaker

Intro to Managed Delivery

Kubernetes integration into Managed Delivery

Opportunities and future work

Q&A



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Kubernetes and Spinnaker: the paradigms



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

k8s: declare your intentions

Kubernetes is unique in its positioning to operators

- Declarative in nature
- Developer-friendly syntax
- Define your end state explicitly, k8s complies

Kubernetes tooling is extensible

- Easily interface with cloud providers
- “Operators” allow declarative → imperative translation



Intro to Spinnaker



What is it?

Multi-cloud continuous delivery platform
Open-source: github.com/spinnaker
Focused on fast, safe and reliable software delivery
At Netflix:

- ~11k applications
- ~28k pipelines
- ~200k pipeline executions per day

Two core sets of features:

- Application management
  - Applications/microservices → infrastructure (clusters, firewalls, load balancers, etc.)
- Application deployment
  - Continuous delivery workflows

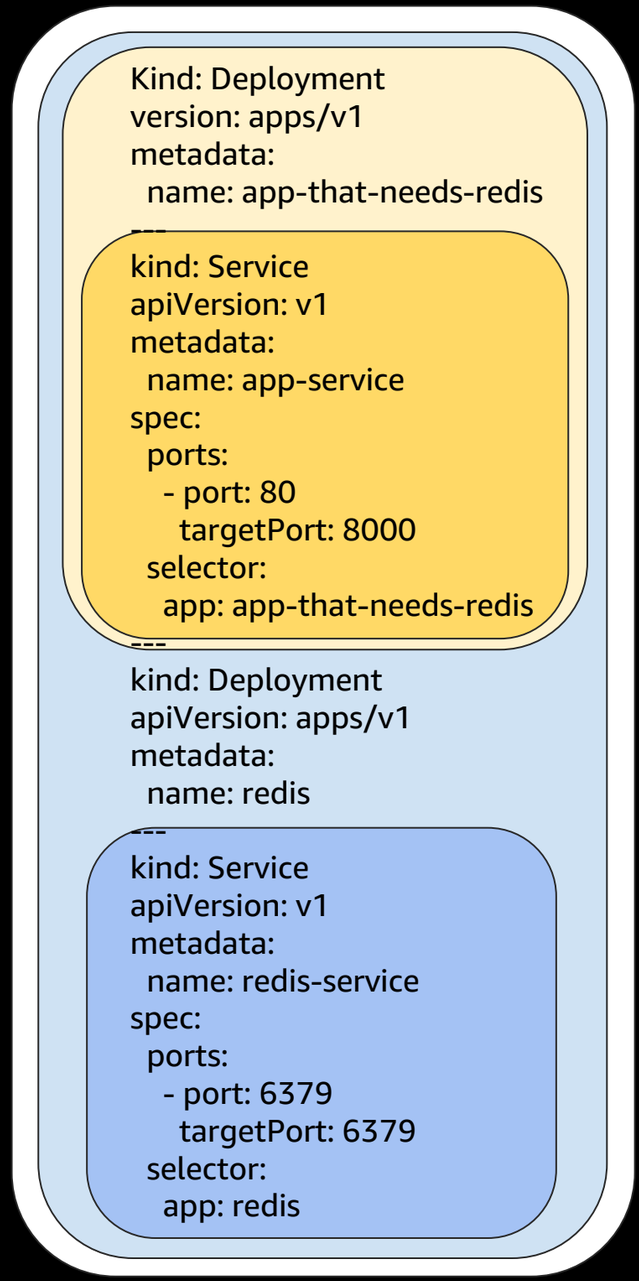
Spinnaker/k8s: a history lesson

Traditional cloud infrastructure orchestration was/is very sequential

- SGs → launch configs → ASGs → load balancers
- Early providers were a perfect fit for simplification
- Pipelines allowed for easy sequencing of those events

Spinnaker Meets Kubernetes

- Infrastructure definitions became “declarative”
- V1: Very similar to imperative infrastructure providers
- V2: Fully adopts k8s declarative design: provide manifests



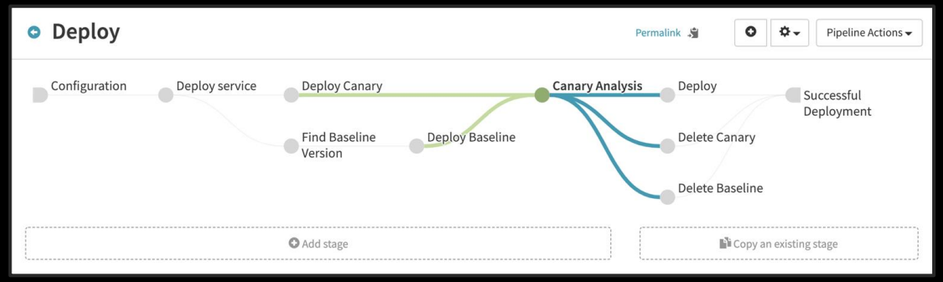
Spinnaker/k8s: the current state

Kubernetes is declarative “in isolation.” With multiple clusters, external services, or complex dependencies, Kubernetes begins to benefit from external tooling like Spinnaker.

- **Spinnaker k8s-v2 provider is fully declarative, accepting manifests as is**
- **Spinnaker pipelines are still “v1,” or inherently imperative**



+



=



Declarative YAML

Imperative pipelines

WAT?!

Spinnaker's Managed Delivery

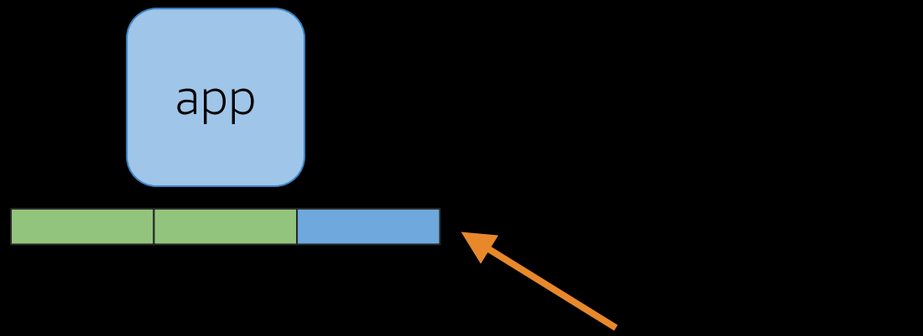


AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5 Years of Pipelines





(a simple pipeline)

5 Years of Pipelines



app



It's hard to keep my similar apps updated.

What's the current best practice?

Why is this so complicated?

app app app



Many unique pipelines

What are you trying to do?



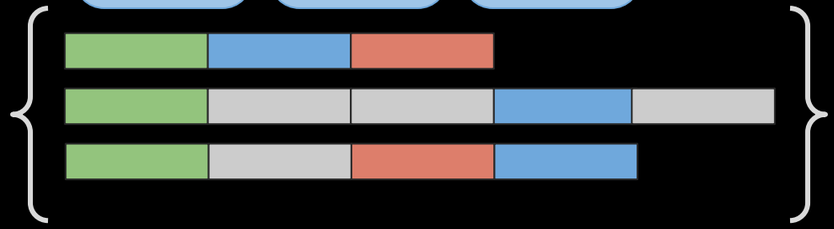
It's hard to keep my similar apps updated.

What's the current best practice?

Why is this so complicated?



app app app



Many unique pipelines

What's your intent?



"I'm trying to deploy a debian package to my test environment."

What’s your intent?



“I’m just trying to figure out why my pipeline is failing.”

What's your intent?



"The person who made this pipeline left,
and I don't know how to change it."

What’s your intent?



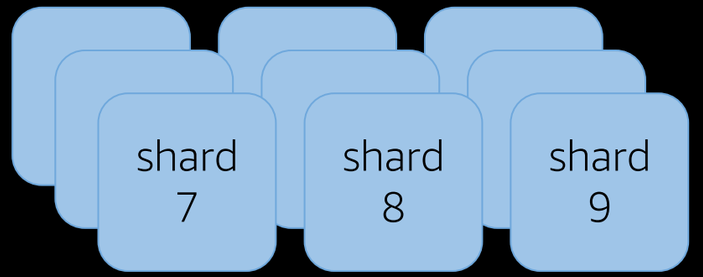
“I want things to be easier”



I want to store config as code

- Simple app
- Prod app
- Highly available app

I want someone else to decide the basics, and keep them up to date



I want to easily manage many similar configs

Enter Managed Delivery



Define **infrastructure** requirements **as code**

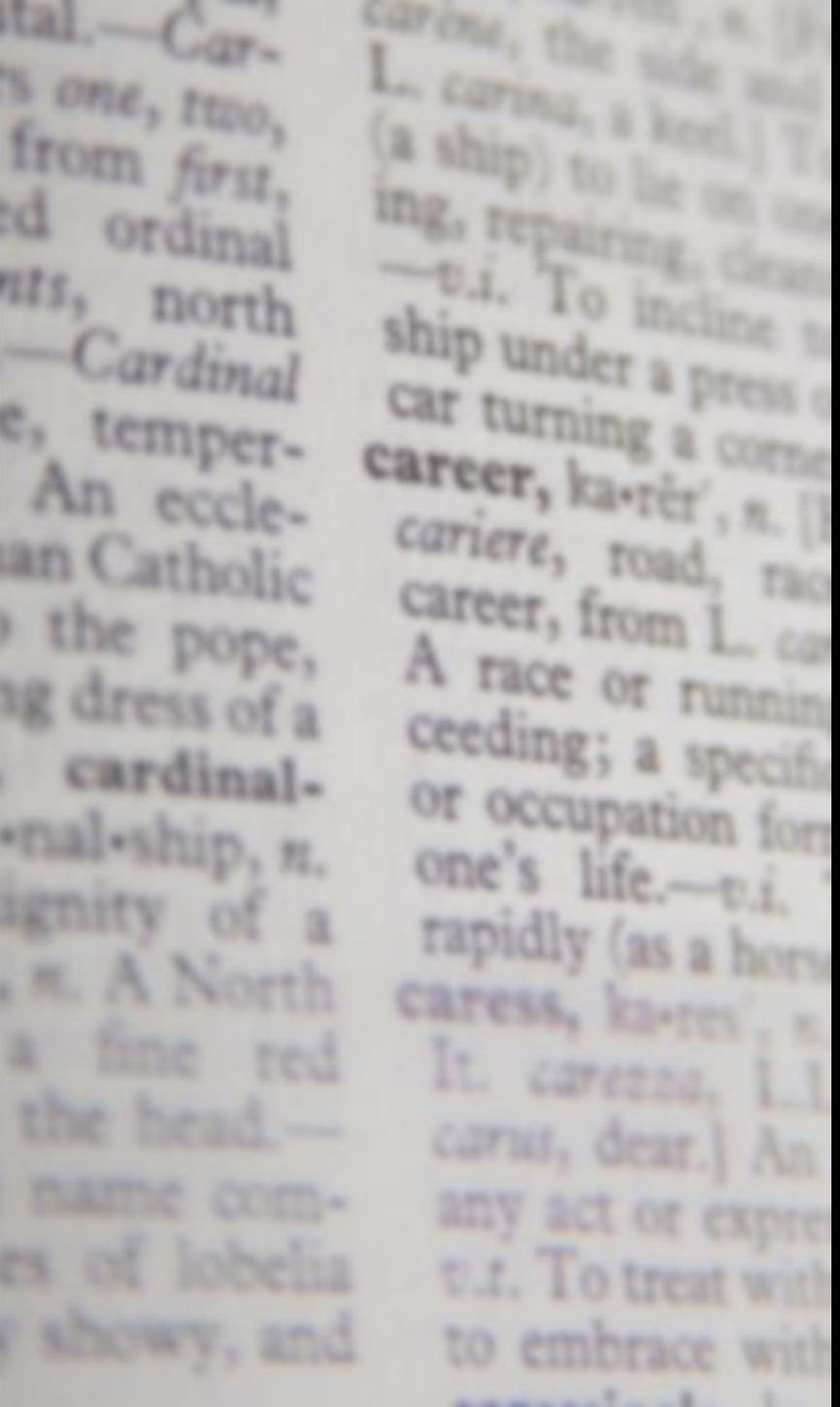
Define **delivery** requirements **as code**

Share configs as best practices

Update best practices

Safely **roll out updates** of best practices

Provide **pluggable** spots for others to inject opinions



Definitions

Artifact
A versioned software package that you deploy

Resource
A piece of infrastructure you manage declaratively

Spec
User-supplied definition of a resource

Current state
The actual state of a resource in the cloud (if it exists)

Desired state
The state you'd like a resource to be in at all times

Environment
Logical grouping of resources (test, staging, prod, etc.)

Constraint
A gate to promote an artifact version into an environment

Declarative infrastructure

Natively supported resource kinds:

- EC2 Security groups
- EC2 Load balancers (classic and application)
- EC2 Clusters
- Titus Clusters

More resources can be supported via plugins

```yaml
kind: ec2/cluster@v1
metadata:
  serviceAccount: delivery-engineering@netflix.com
spec:
  moniker:
    app: keeldemo
    stack: test
  imageProvider:
    reference: keeldemo
  locations:
    account: test
    regions:
    - name: us-west-2
    - name: us-east-1
  launchConfiguration:
    instanceType: m5.large
  capacity:
    min: 1
    max: 2
    desired: 2
  dependencies:
    loadBalancerNames:
    - keeldemo-test-clb
    securityGroupNames:
    - keeldemo
```

Resource spec

Declarative delivery

Artifacts describe your deployable software packages

Environments logically group the resources needed to run your software. For example, “test” and “prod” with compute and networking resources

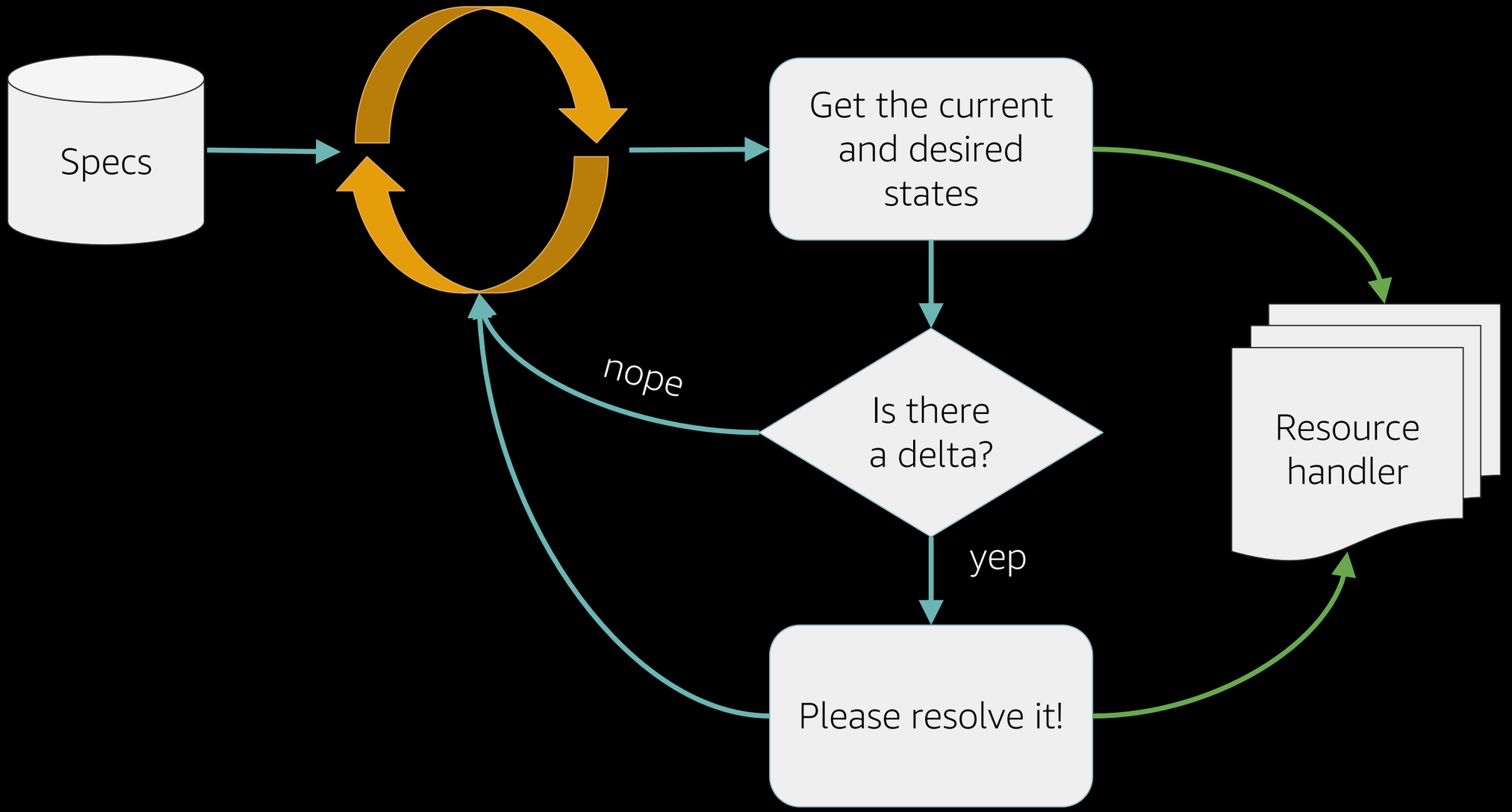
Constraints gate whether an artifact can be promoted to an environment:

- deployed in prior environment
- deployment windows
- manual approval
- …

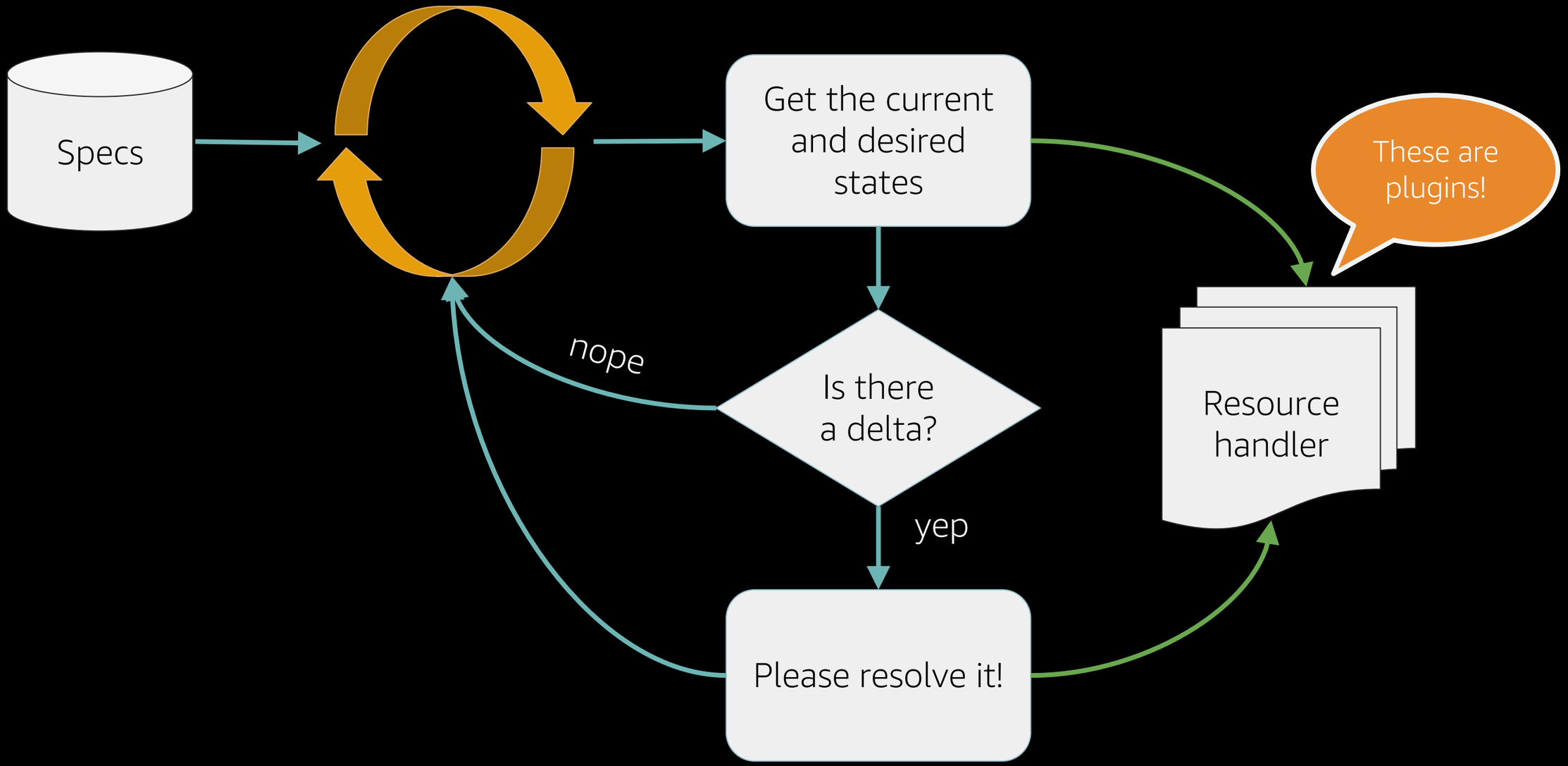
```yaml
name: sample-delivery-config
application: keeldemo
artifacts:
- name: keeldemo
  type: deb
environments:
- name: test
  constraints: []
  resources:
  - kind: ec2/classic-load-balancer@v1
    # details
  - kind: ec2/cluster@v1
    # details
- name: prod
  notifications: # omitted for brevity
  constraints:
  - type: depends-on
    environment: test
  resources:
  - kind: ec2/cluster@v1
    # details
  - kind: ec2/classic-load-balancer@v1
    # details
```

Delivery config

Resource monitoring & actuation loop



Resource monitoring & actuation loop

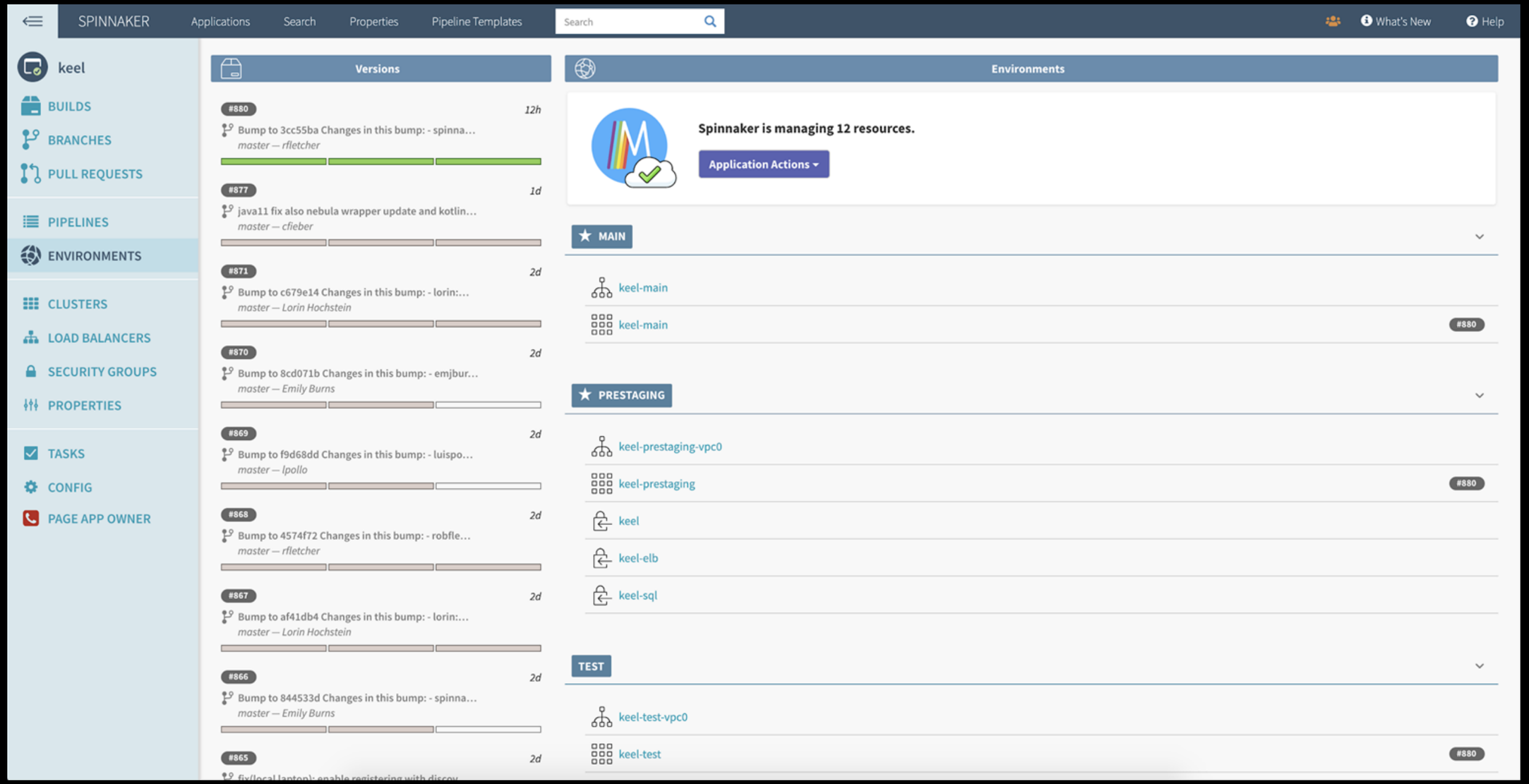


Managed delivery: current state

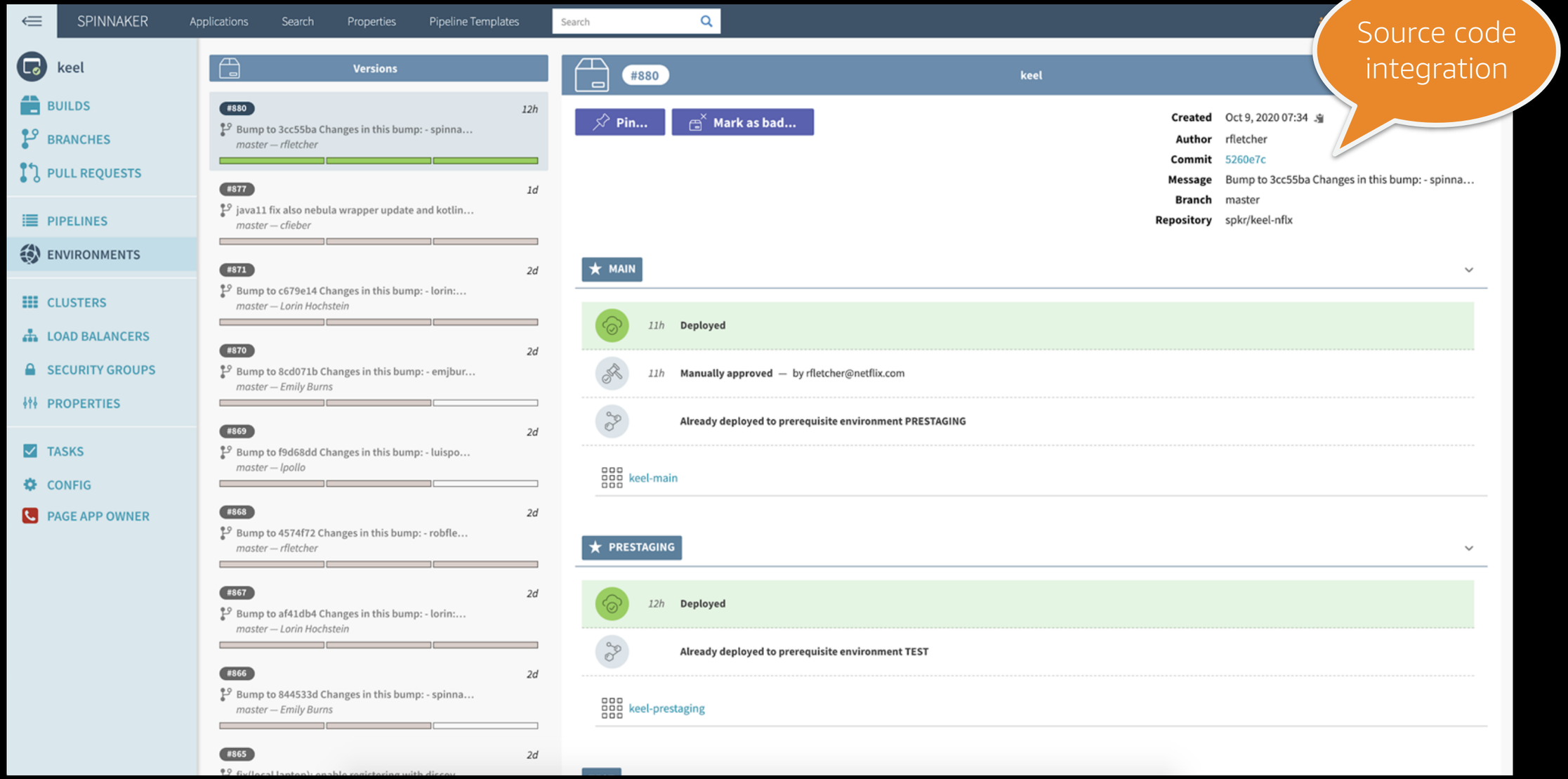


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

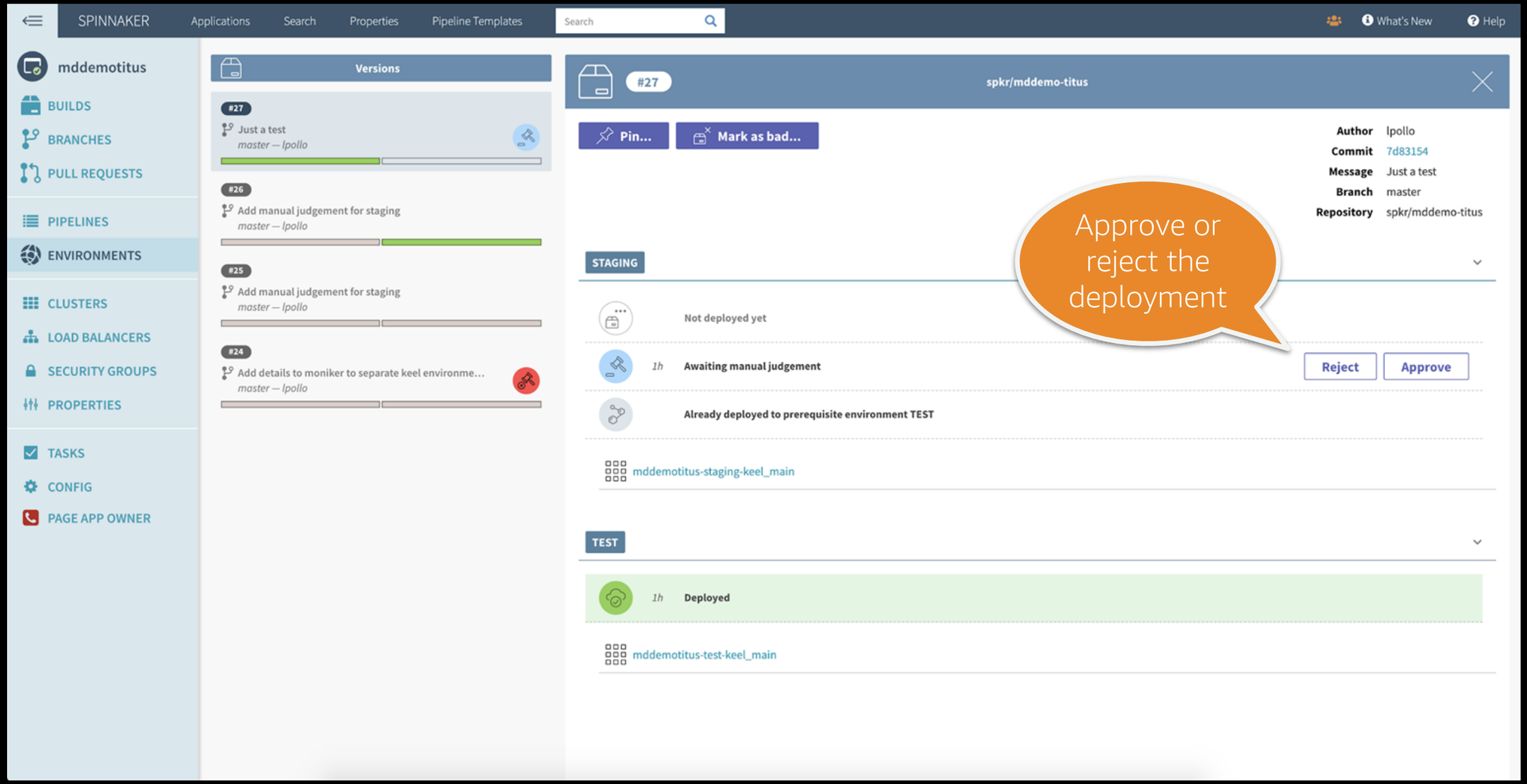
Environments view



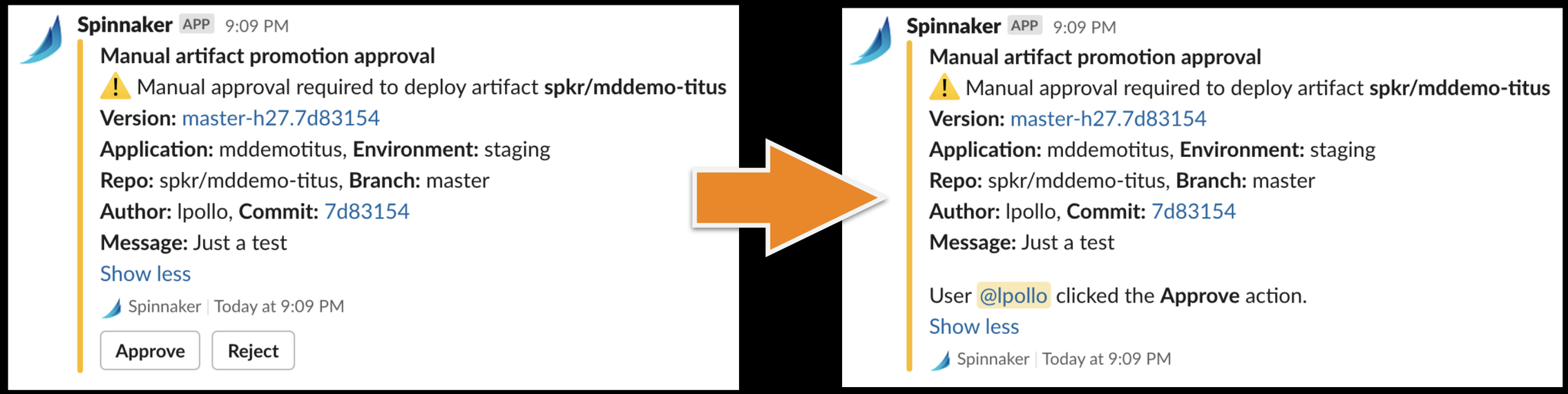
Artifact details view



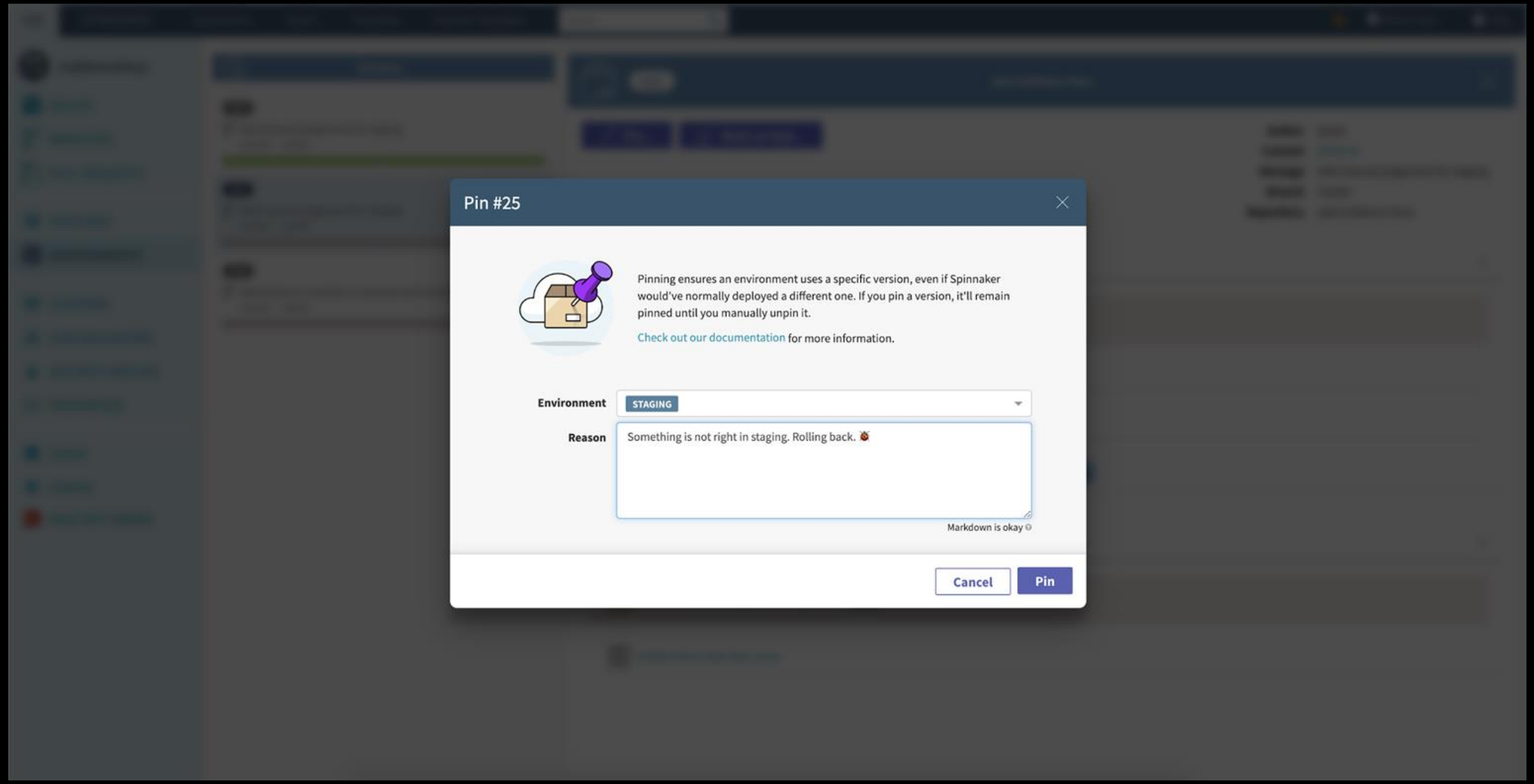
Manual promotion approval...



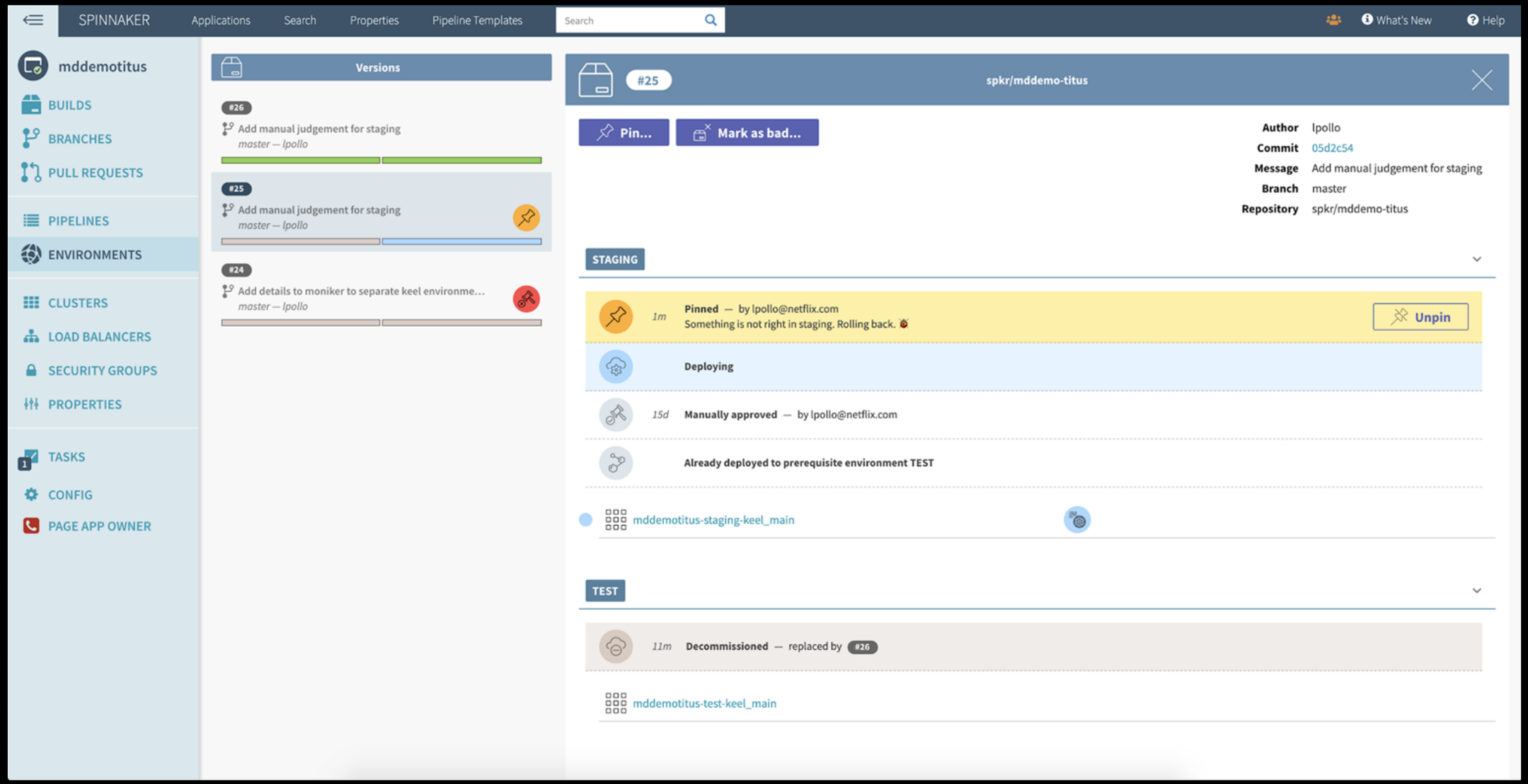
...with Slack integration!



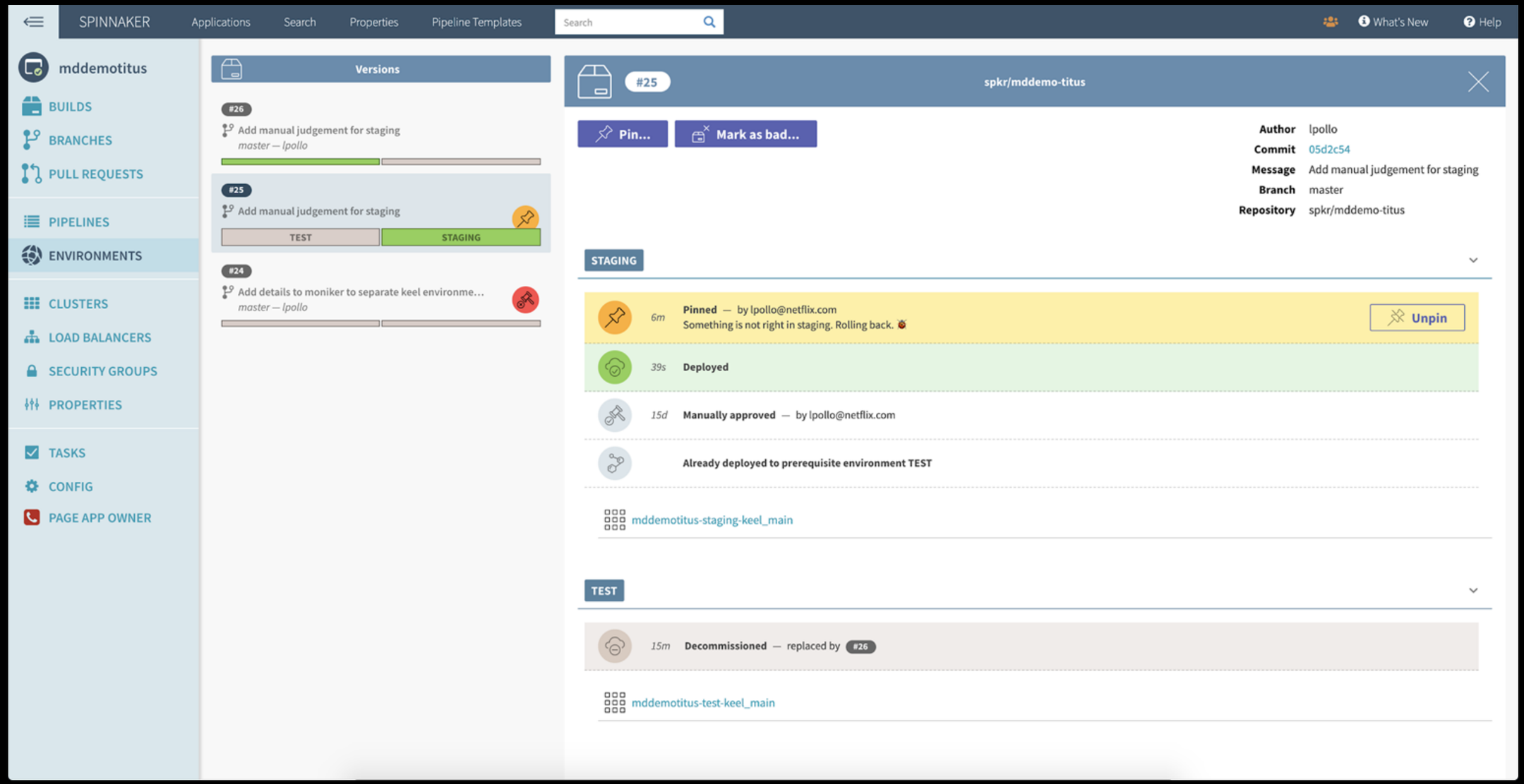
Pinning (rollback, hotfixes)



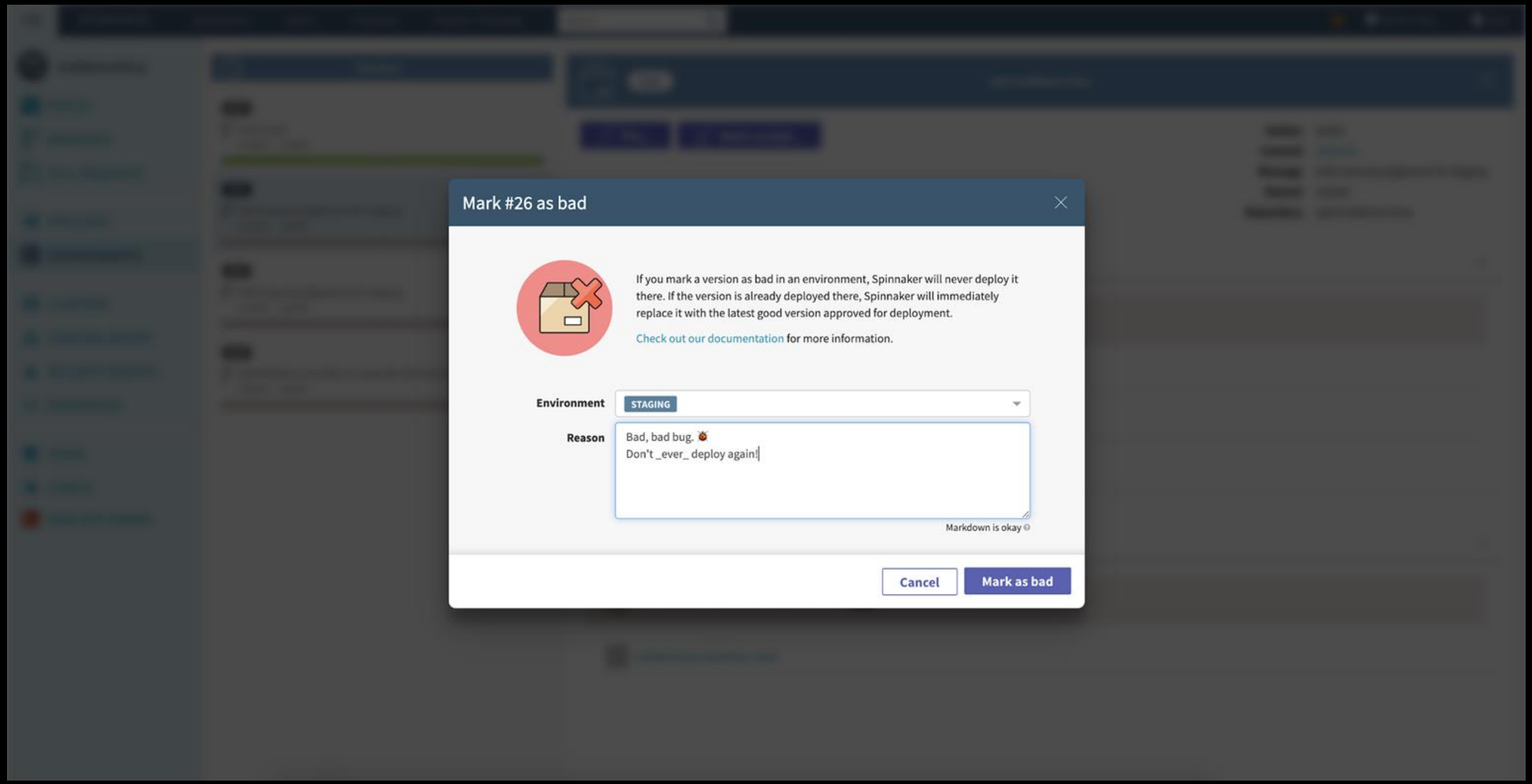
Pinning (rollback, hotfixes)



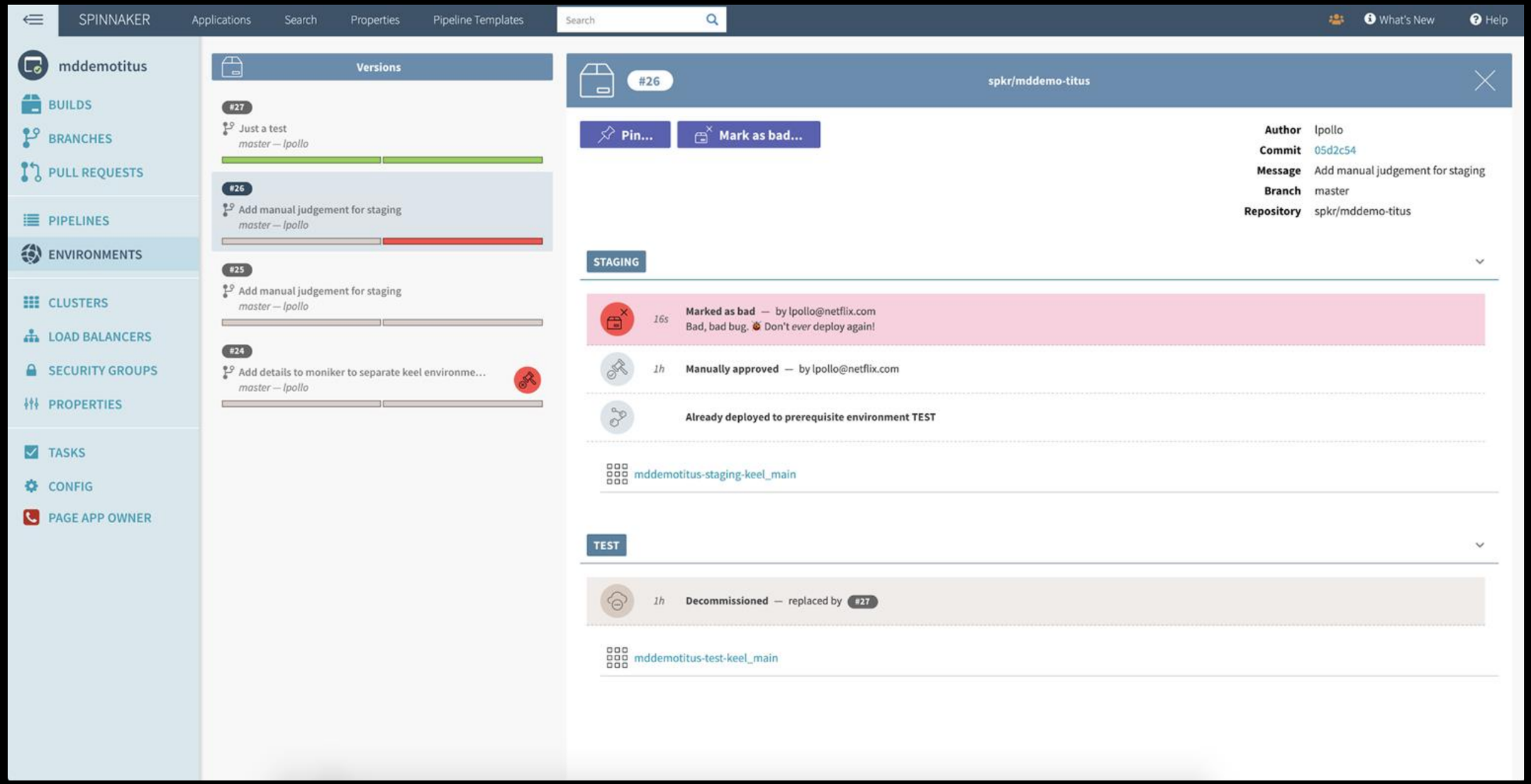
Pinning (rollback, hotfixes)



Marking a version “bad”



Marking a version “bad”



Keel plugins

ResourceHandler

Implement your own declarative resources

Resolver

Add your own opinions or smart defaults into resource specs

ConstraintEvaluator

Implement your own promotion constraints

ArtifactSupplier

Implement your own artifact types

Kubernetes support in Managed Delivery

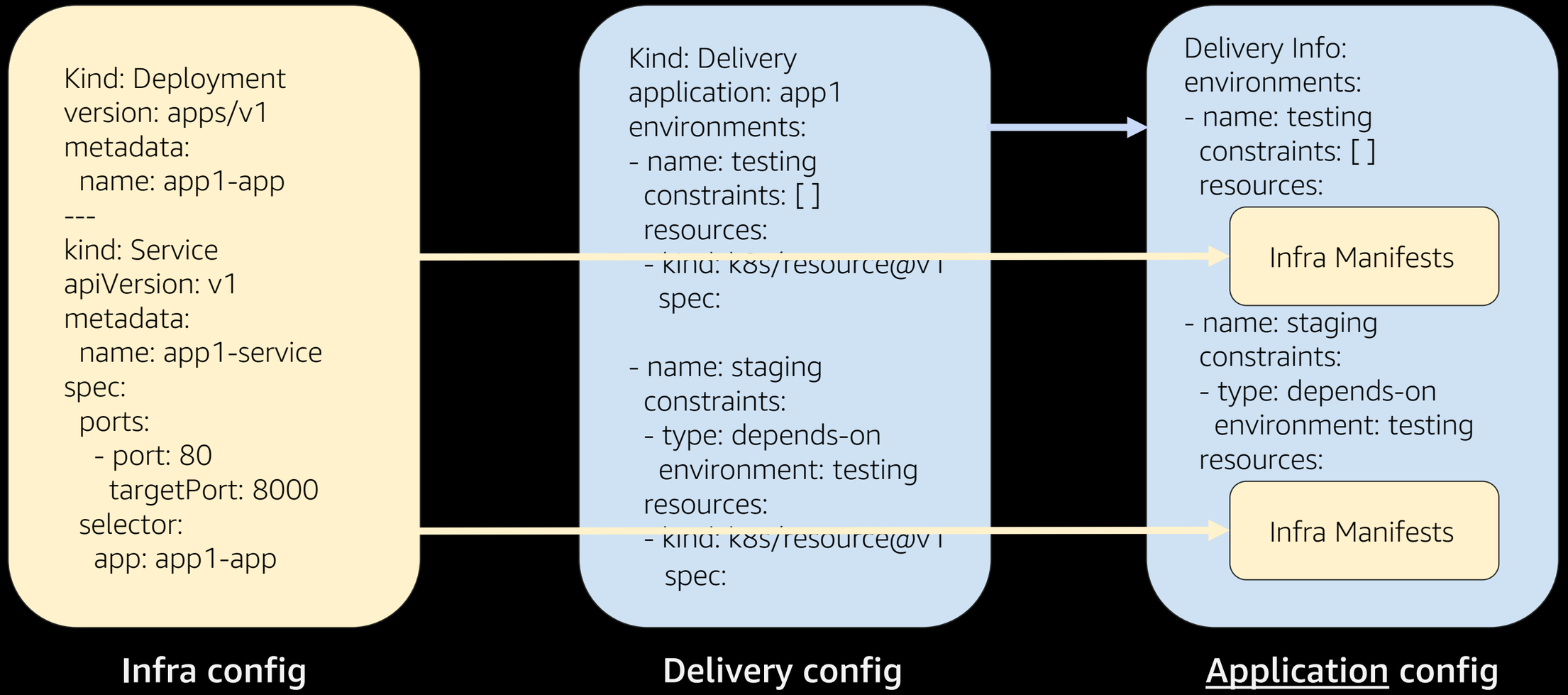


AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Spinnaker/k8s: Managed Delivery

But wait, isn't Managed Delivery declarative? Doesn't it have a similar API structure?



Managed Delivery: Keel Plugin

*github.com/nimakaviani/**managed-delivery-k8s-plugin***

README.md

Managed Delivery K8s Plugin

Usage

Add the following to `keel.yml` in the necessary profile to load plugin.

```
spinnaker:
    extensibility:
      plugins:
        aws.ManagedDeliveryK8sPlugin:
          id: aws.ManagedDeliveryK8sPlugin
          enabled: true
          version: <<plugin release version>>
      repositories:
        awsManagedDelvieryK8sPluginRepo:
          id: awsManagedDelvieryK8sPluginRepo
          url: https://raw.githubusercontent.com/nimakaviani/managed-delivery-k8s-plugin/master/plug
```

Managed Delivery: Deployment

- Kleat for Config Generation (https://github.com/spinnaker/kleat)

```
> kleat generate halconfig.yml dest-dir/
```

```yaml
persistentStorage:
  sql:
    enabled: true
    default:
      jdbcUrl: jdbc:mysql://localhost:3306/keel
      user: user
      password: password
    migration:
      jdbcUrl: jdbc:mysql://localhost:3306/keel
      user: user
      password: password

managedDelivery:
  enabled: true
  keel:
    plugins:
      ec2:
        enabled: false
      k8s:
        enabled: true
```

Managed Delivery: Deployment

- Kleat for Config Generation (https://github.com/spinnaker/kleat)

```
> kleat generate halconfig.yml dest-dir/
```

- Add plugin info to `keel.yml`

```
❯ cat <<EOF >> keel.yml
spinnaker:
    extensibility:
      plugins:
        aws.ManagedDeliveryK8sPlugin:
          id: aws.ManagedDeliveryK8sPlugin
          enabled: true
          version: << plugin release version >>
      repositories:
        awsManagedDelvieryK8sPluginRepo:
          id: awsManagedDelvieryK8sPluginRepo
          url:  https://[...]/managed-delivery-k8s-plugin/master/plugins.json
EOF
```

Managed Delivery: Deployment

- Kleat for Config Generation
  (https://github.com/spinnaker/kleat)

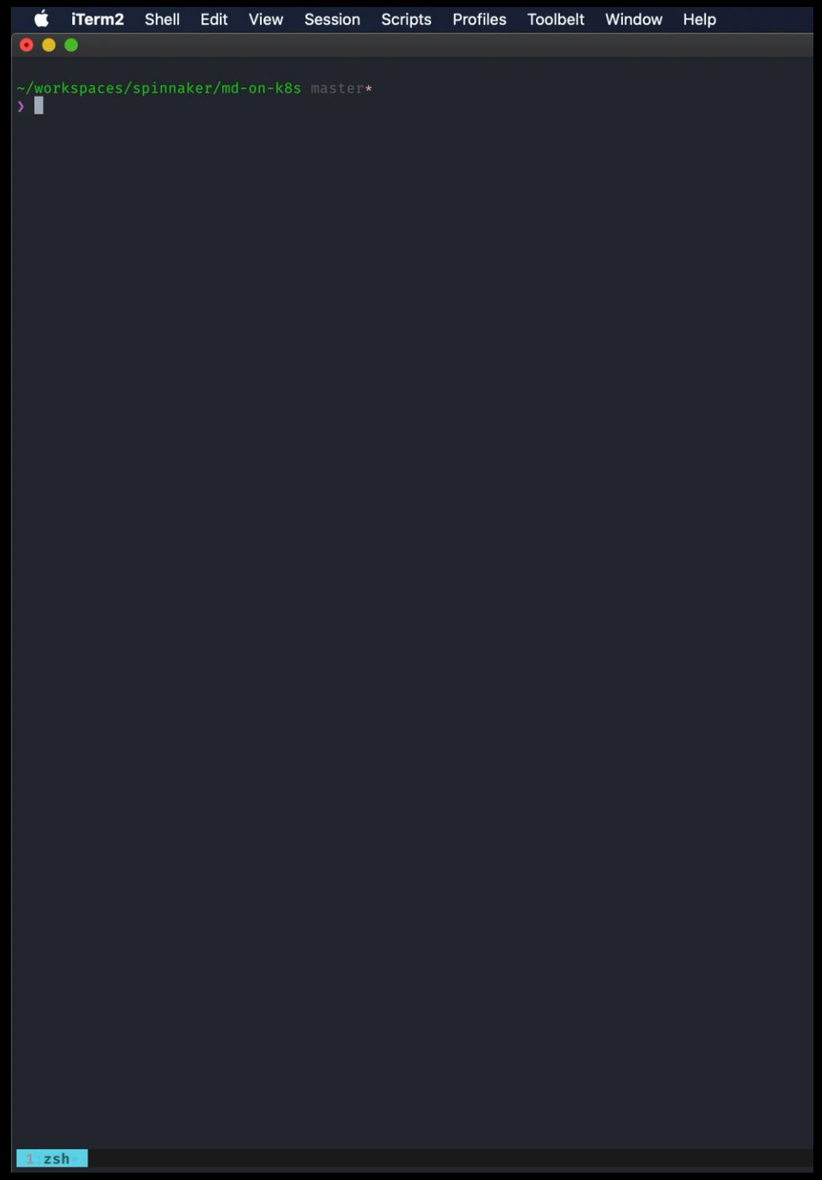
```
> kleat generate halconfig.yml dest-dir/
```

- Add plugin info to `keel.yml`

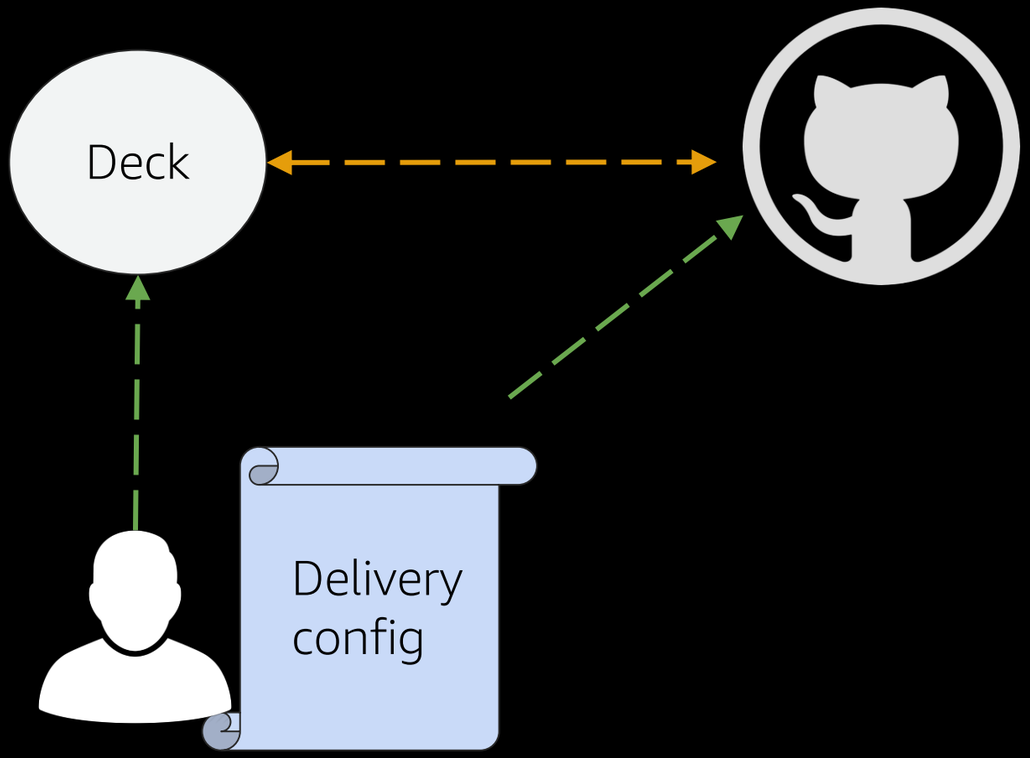
- Define Custom Overlays
  and Apply

```
> kustomize build base/
```

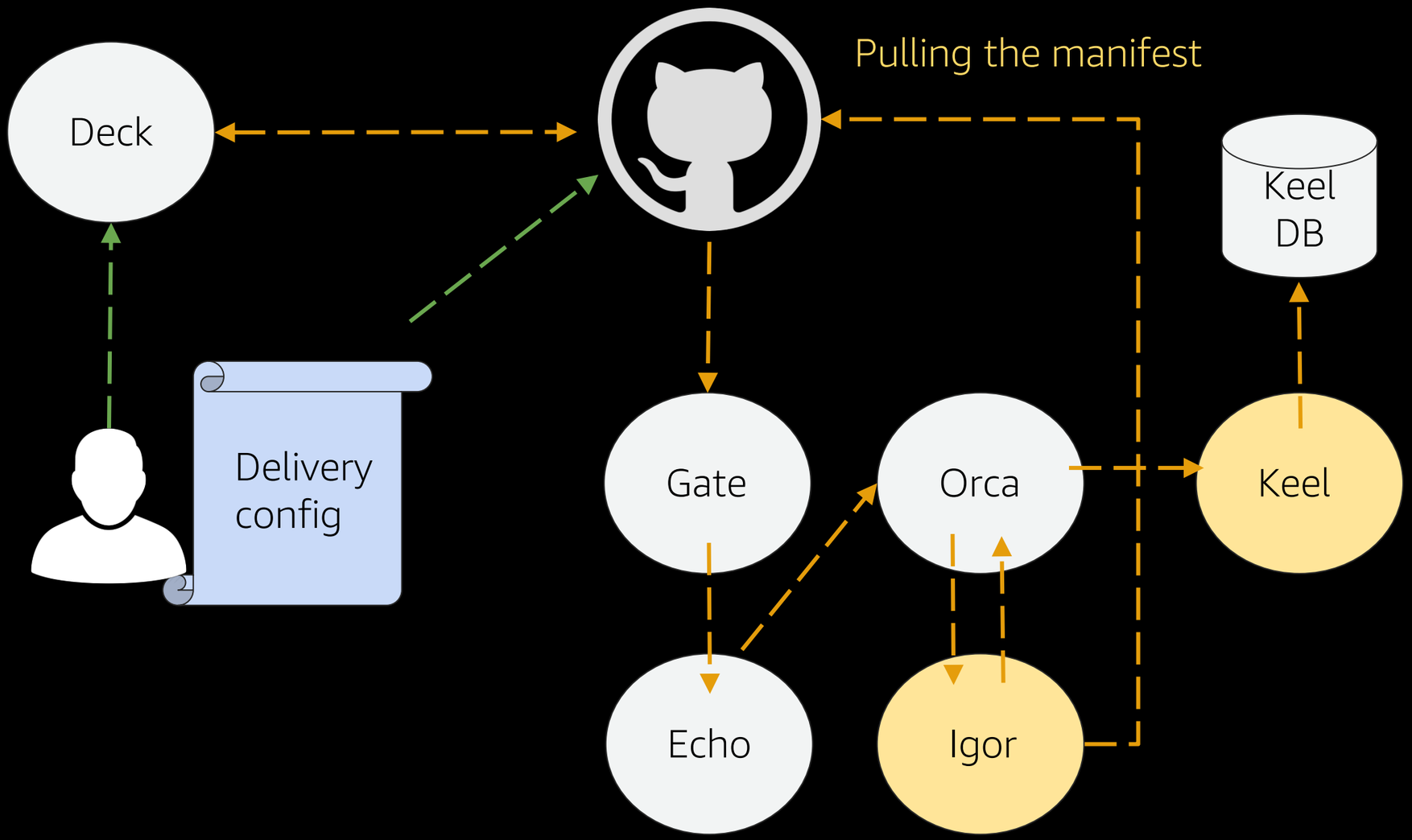
- Have your cluster going ...



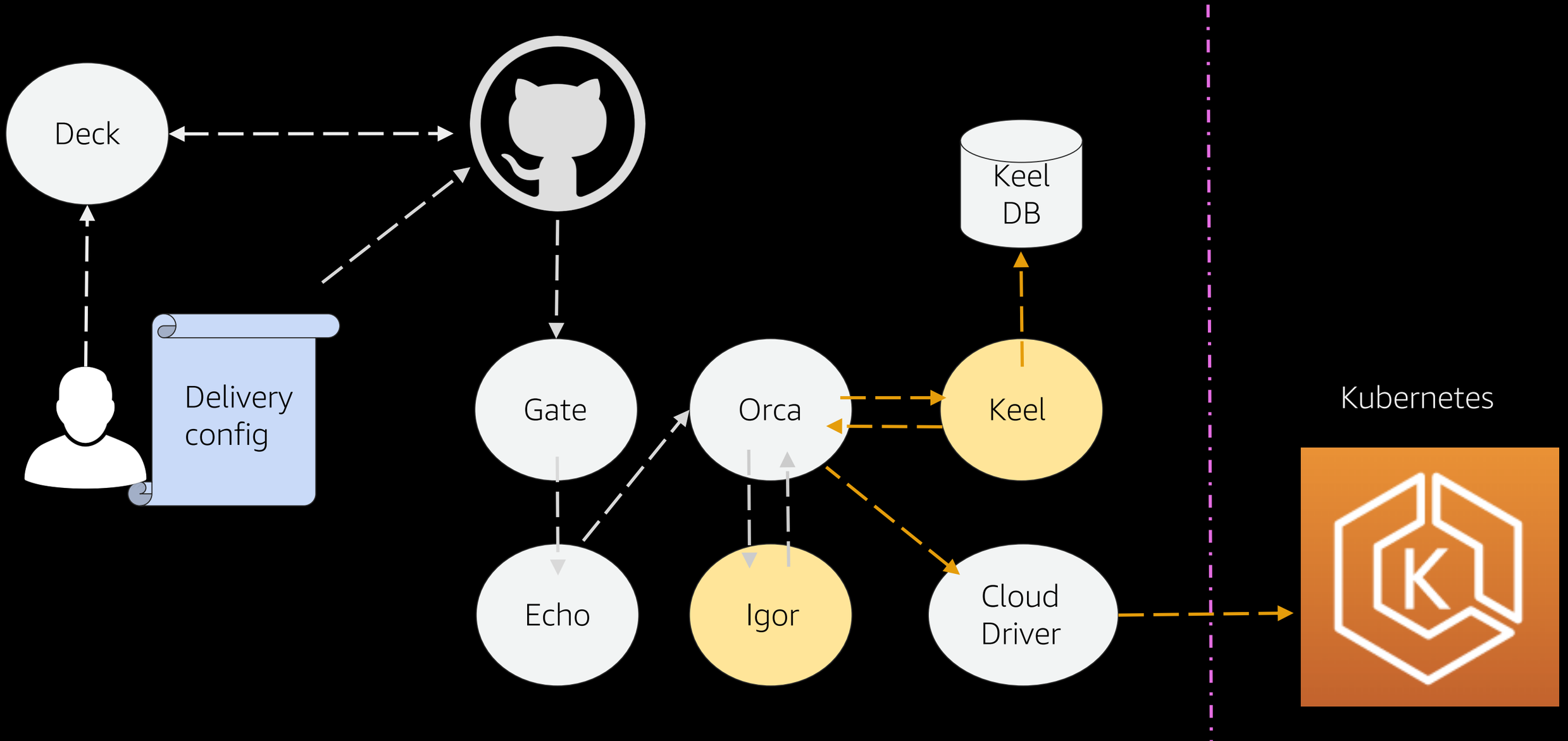
Managed Delivery: GitOps trigger



Managed Delivery: GitOps trigger



Managed Delivery: GitOps deploy



Aren’t there other solutions already?



ArgoCD

\+

(Argo Flux)



Flux

Plus all the usual suspects:





...

Existing solutions:

- require piecing together support for K8S, or support only Kubernetes
- require piecing together support for GitOps workflows, or support only GitOps workflows
- require piecing together support for infrastructure outside Kubernetes (AWS, etc.)

The Jenkins image is a trademark of the Jenkins project (jenkins.io)

The Gitlab logo is a trademark of Gitlab and used subject to the terms of the Creative Commons license

Aren’t there other solutions already?



ArgoCD

\+

(Argo Flux)



Flux

Plus all the usual suspects:





...

Existing solutions:

- require piecing together support for K8S, or support *only* Kubernetes
- require piecing together support for GitOps workflows, or support *only* GitOps workflows
- require piecing together support for any infrastructure outside Kubernetes (AWS, etc.)



Managed Delivery addresses these gaps

The Jenkins image is a trademark of the Jenkins project (jenkins.io)

The Gitlab logo is a trademark of Gitlab and used subject to the terms of the Creative Commons license

Coming Soon ...

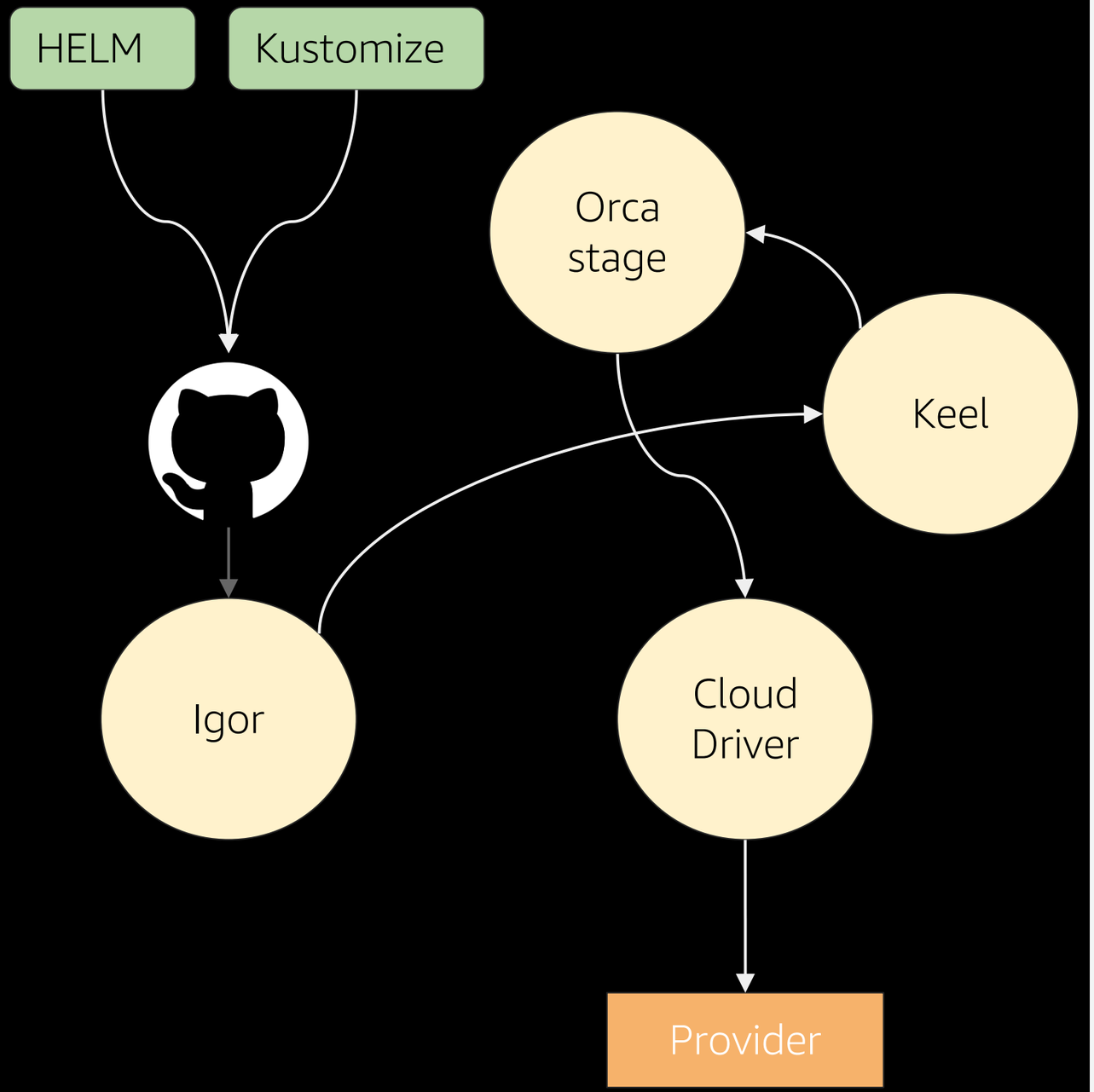
Community deployment of keel & Managed delivery using Kleat

Coming Soon ...

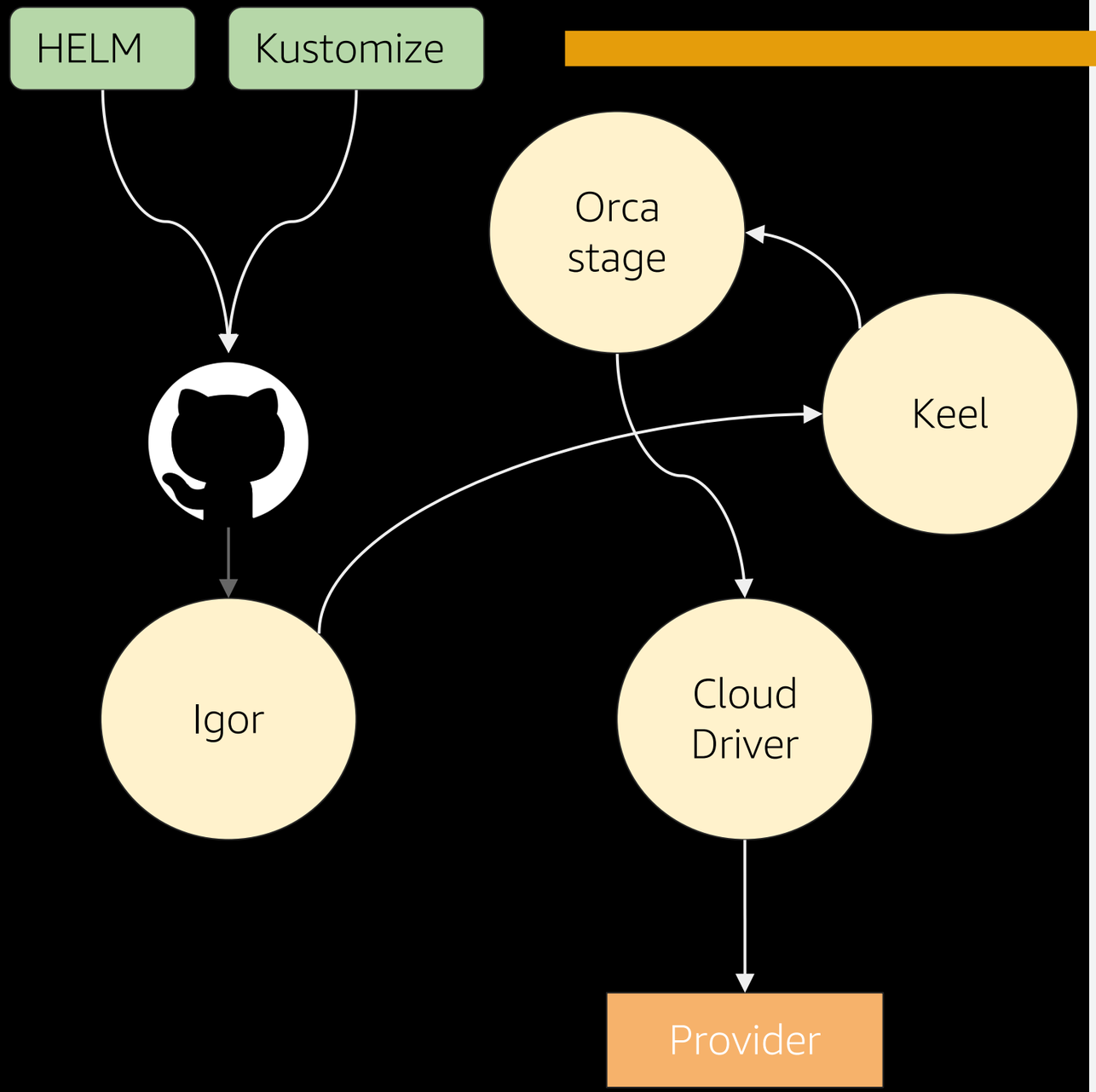
Community deployment of keel & Managed delivery using Kleat

Robust and reliable Kubernetes Keel Plugin

MD Templating Model



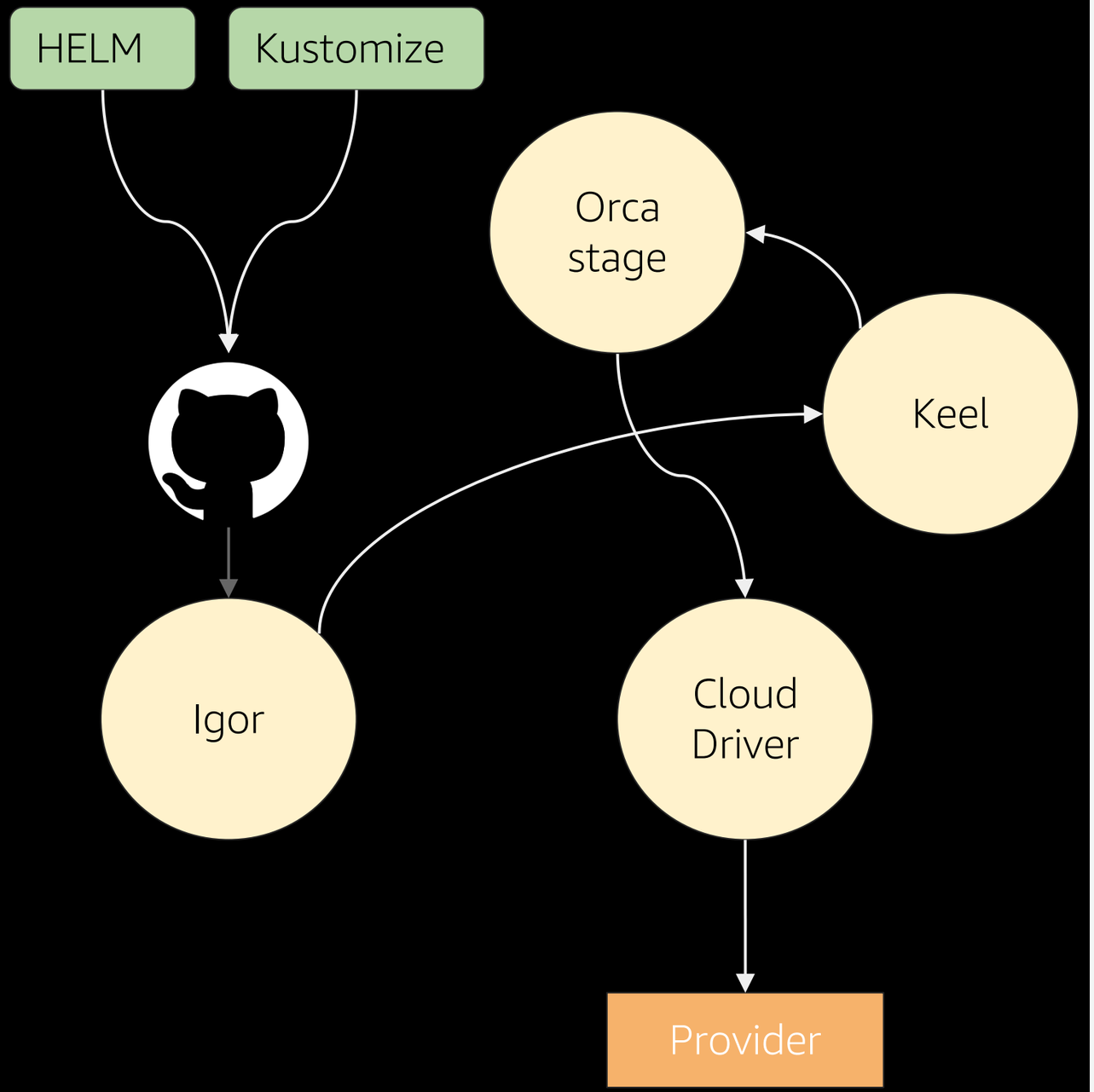
MD Templating Model



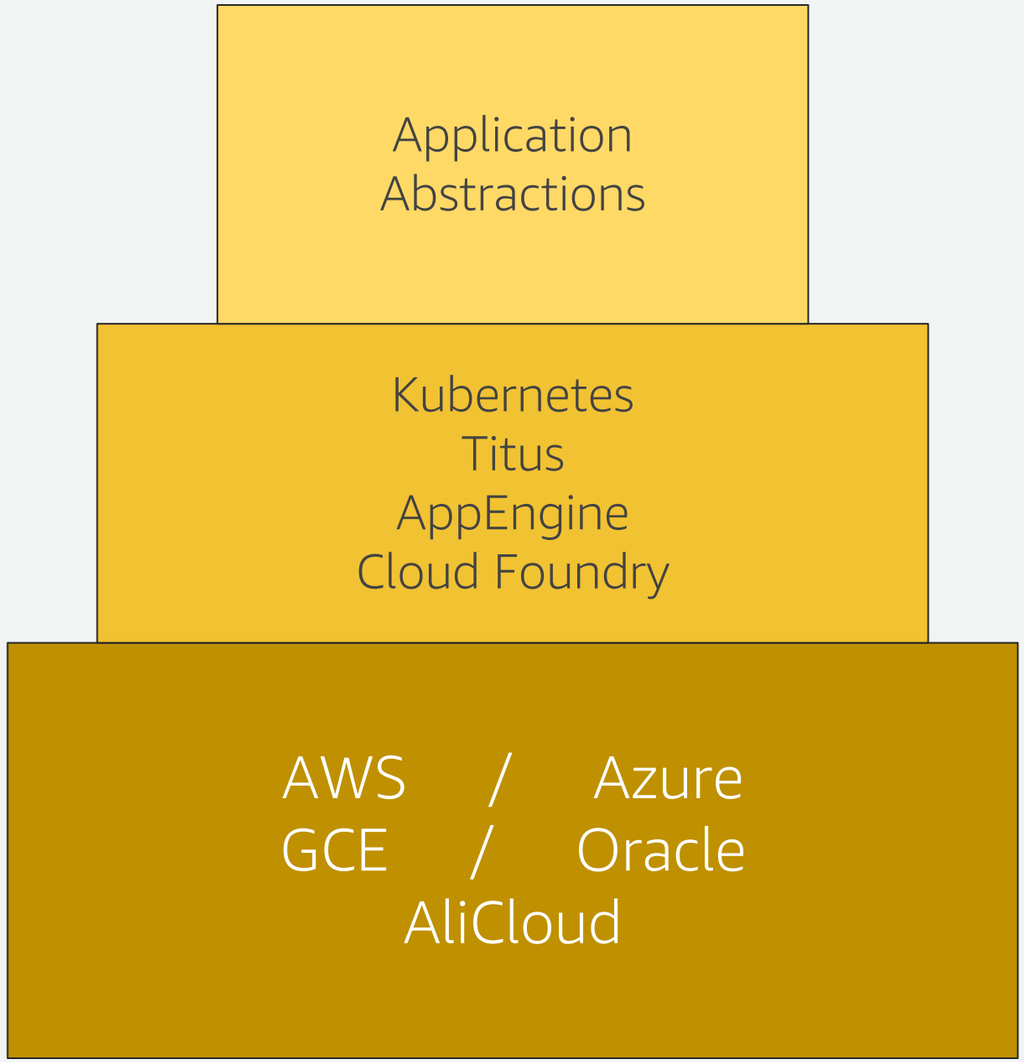

```
- kind: k8s/kustomize@v1
  metadata:
        account: my-k8s-account
            name: kustomization.yml
  spec:
            apiVersion: kustomize.config.k8s.io/v1beta1
            namePrefix: prod-
            configMapGenerator:
            - files:
              - application.properties
              name: demo-configmap
            patchesStrategicMerge:
            - patch.yml

- kind: k8s/kustomize@v1
  metadata:
        account: my-k8s-account
        name: patch.yml
  spec:
            apiVersion: apps/v1
        kind: Deployment
        metadata:
          name: the-deployment
            ...
```

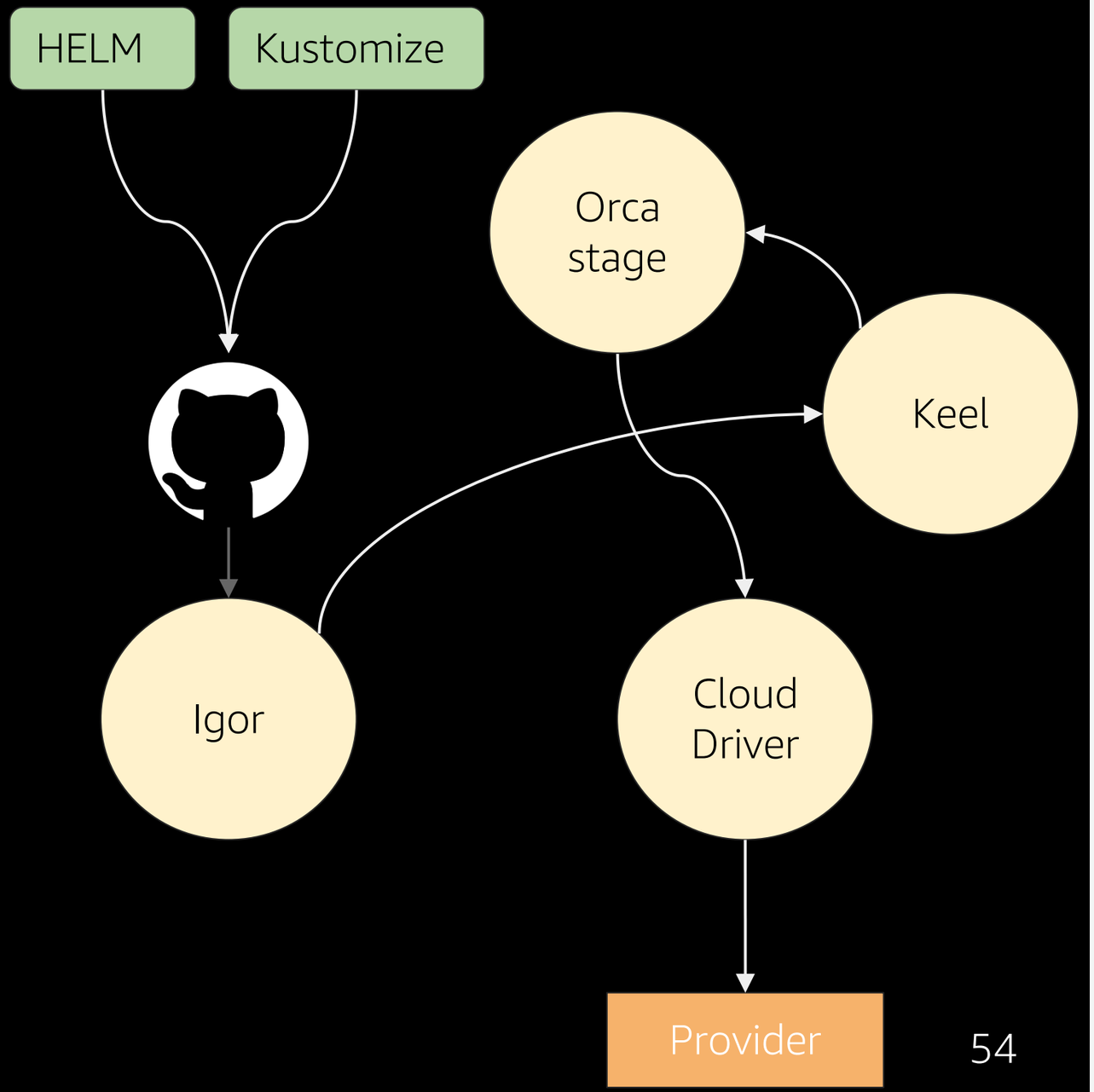
MD Templating Model



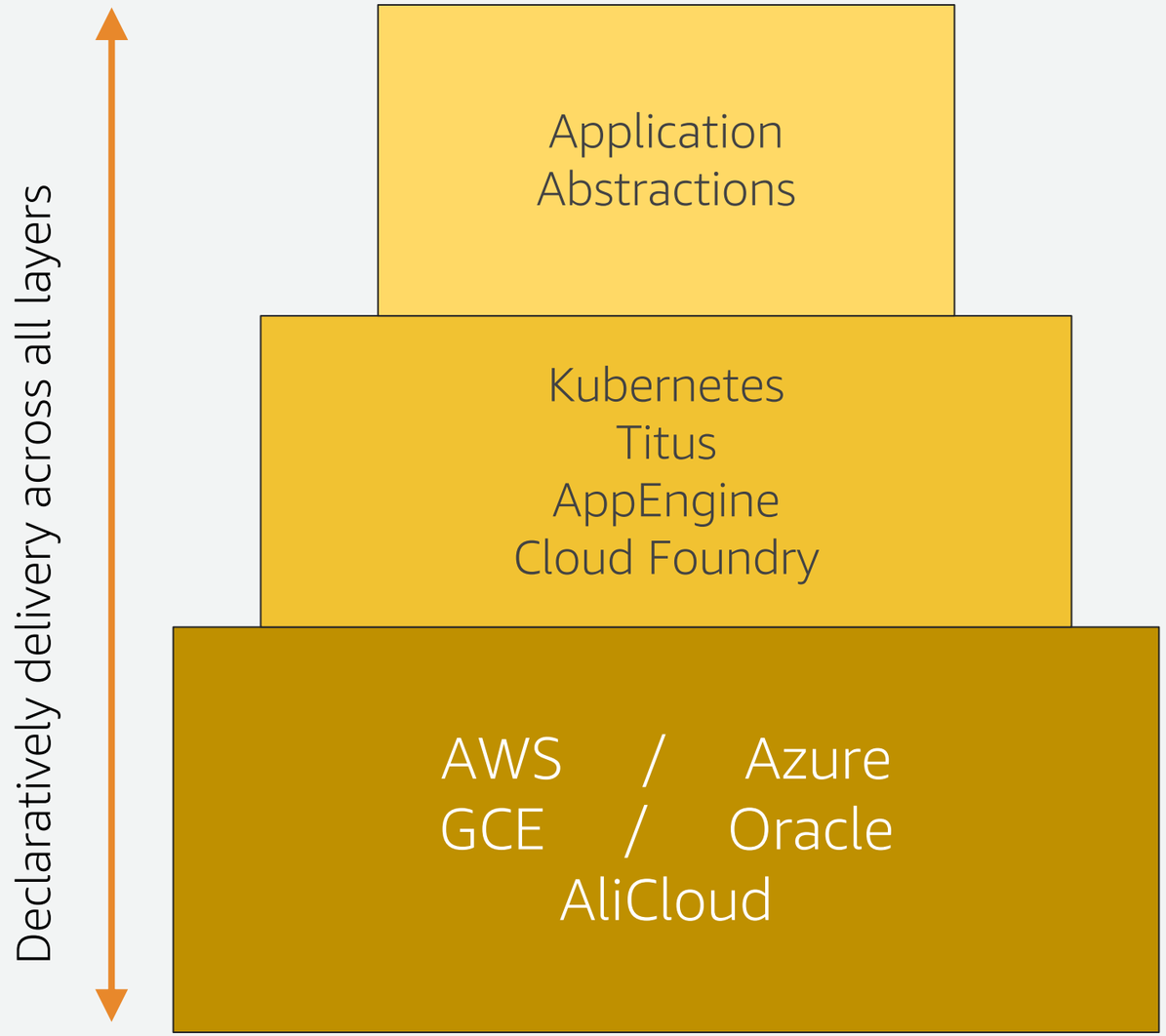
Delivery across Stacks



MD Templating Model



Delivery across Stacks



Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.