ITNW (IF THIS, NOW WHAT)

Orchestrating an Enterprise

We cannot solve our problems
with the same thinking we used
when we created them.

Albert Einstein

Education is what remains
after one has forgotten
everything he learned
in school.

Albert Einstein

Business Justification

Requirements

Build Pipelines

ThirdPartyResources

Ingress Controller SSL Integration

ChatOps

Michael Ward

Principal Systems Architect

Technical Lead on Project Bitesize

Pearson's Enterprise Platform-as-a-Service based on Kubernetes

Dev Oper and I

@devoperandi

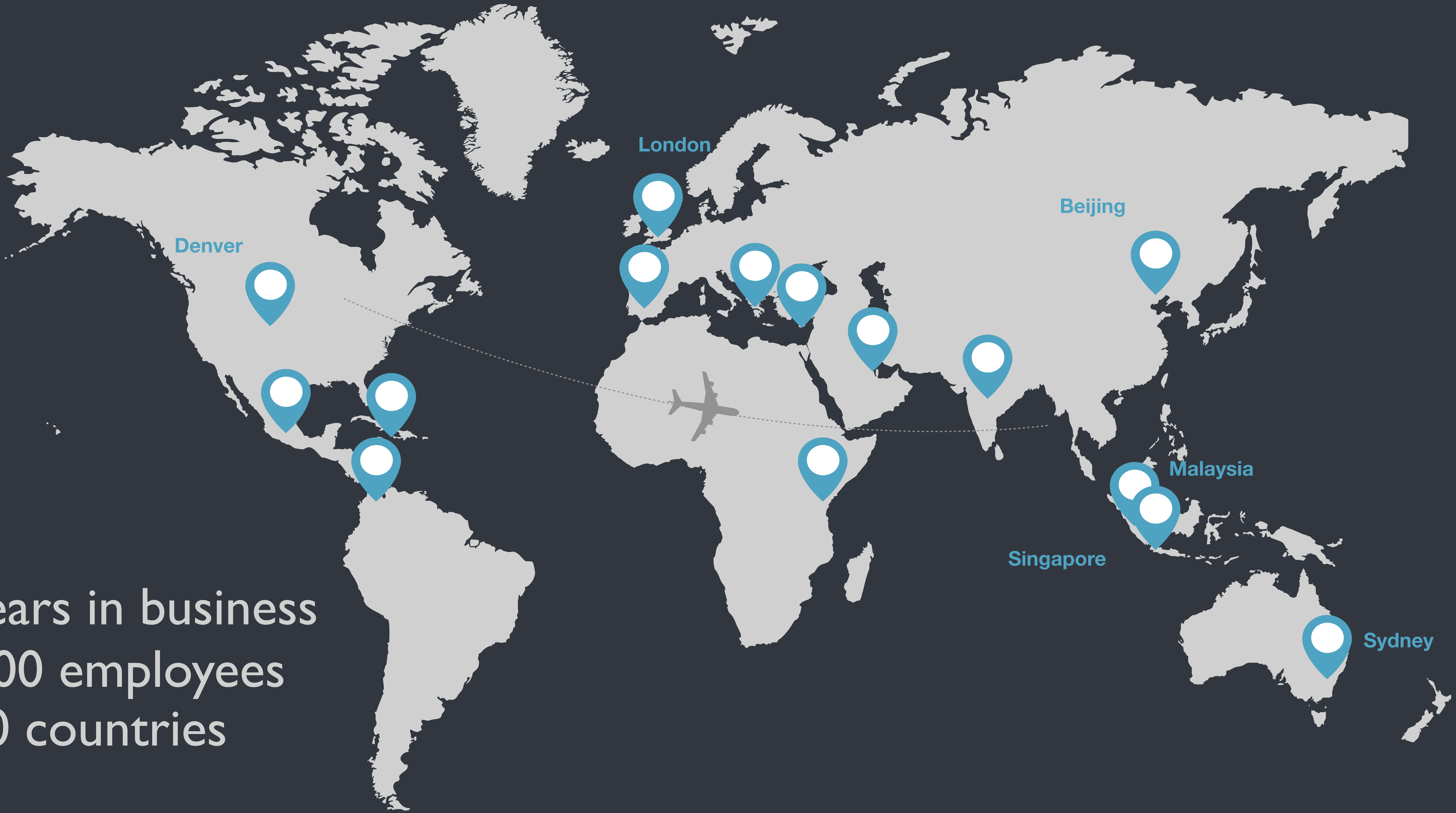
www.devoperandi.com

Pearson

Always Learning

Pearson

Always Learning

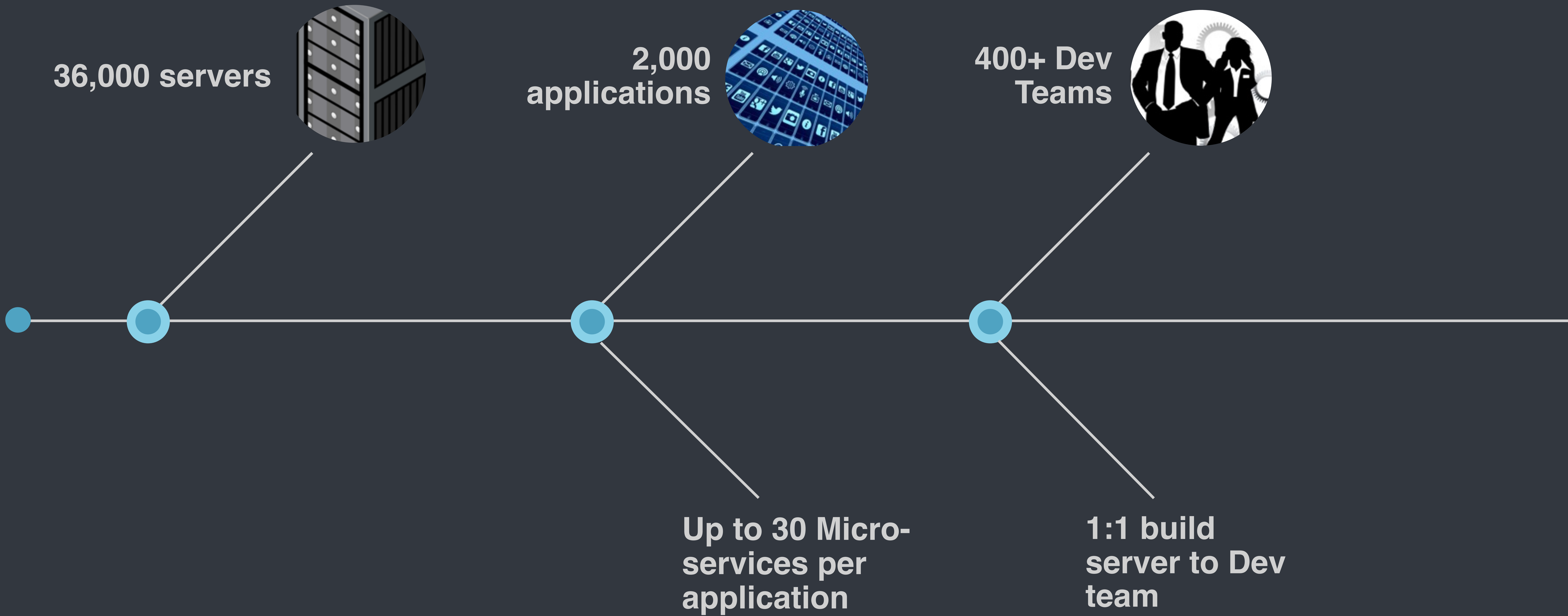


171 years in business

40,000 employees

70 countries

Pearson is no small potatoes:



36,000 servers

2,000 applications

400+ Dev Teams

Up to 30 Micro-services per application

1:1 build server to Dev team

OpenSource @pearson

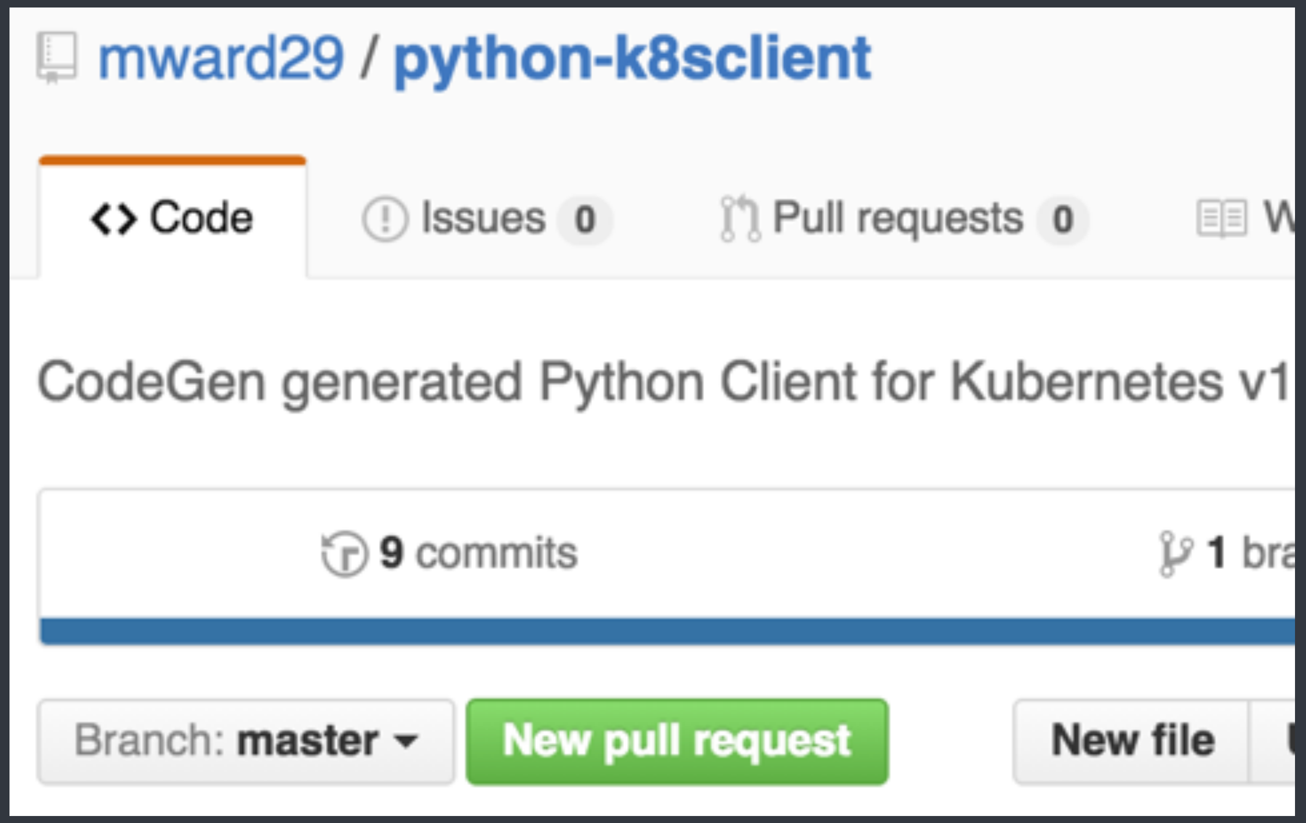
Education

Sharing of Information

Knowledge

OpenSource

Sharing of Code



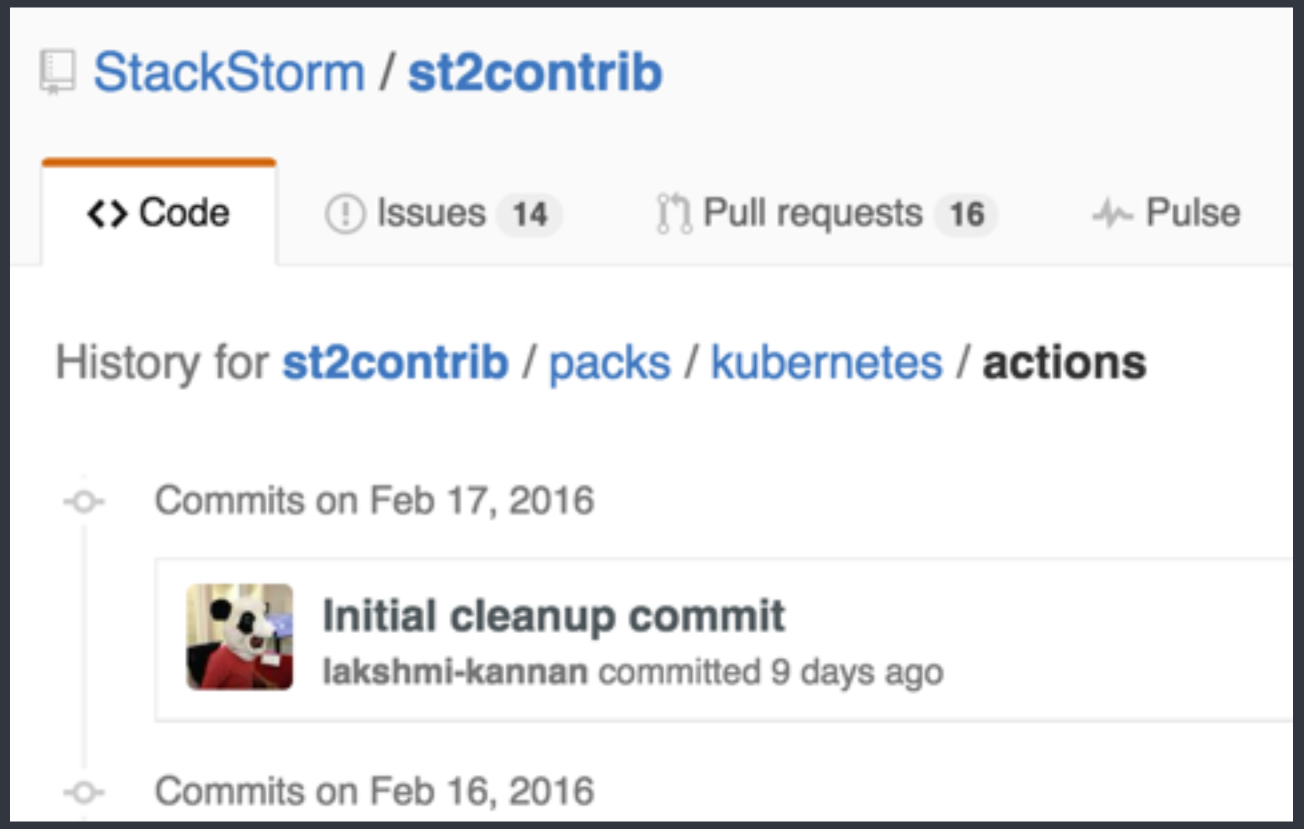
mward29 / **python-k8sclient**

CodeGen generated Python Client for Kubernetes v1

9 commits

1 bra

Branch: master



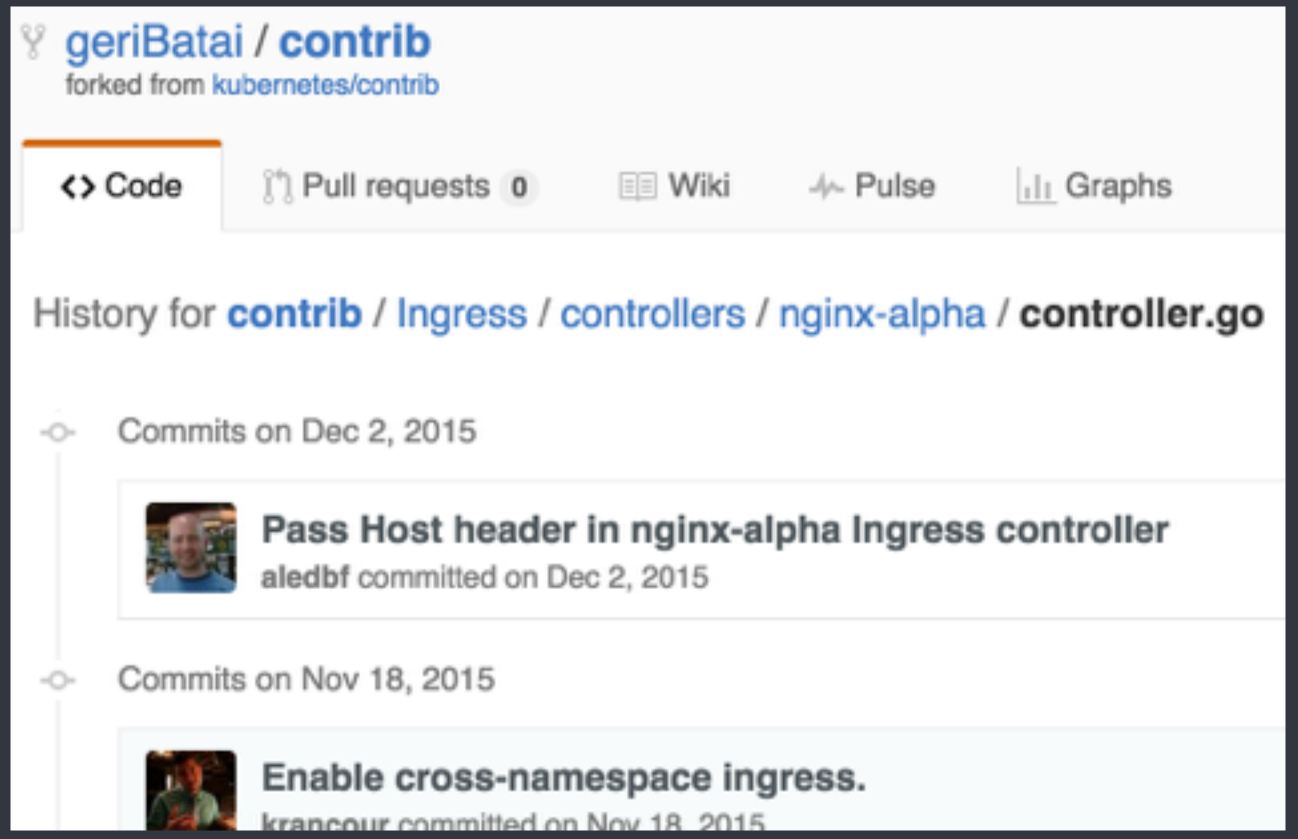
StackStorm / **st2contrib**

History for st2contrib / packs / kubernetes / **actions**

Commits on Feb 17, 2016

Initial cleanup commit
lakshmi-kannan committed 9 days ago

Commits on Feb 16, 2016



geriBatai / **contrib**
forked from kubernetes/contrib

History for contrib / Ingress / controllers / nginx-alpha / **controller.go**

Commits on Dec 2, 2015

Pass Host header in nginx-alpha Ingress controller
aledbf committed on Dec 2, 2015

Commits on Nov 18, 2015

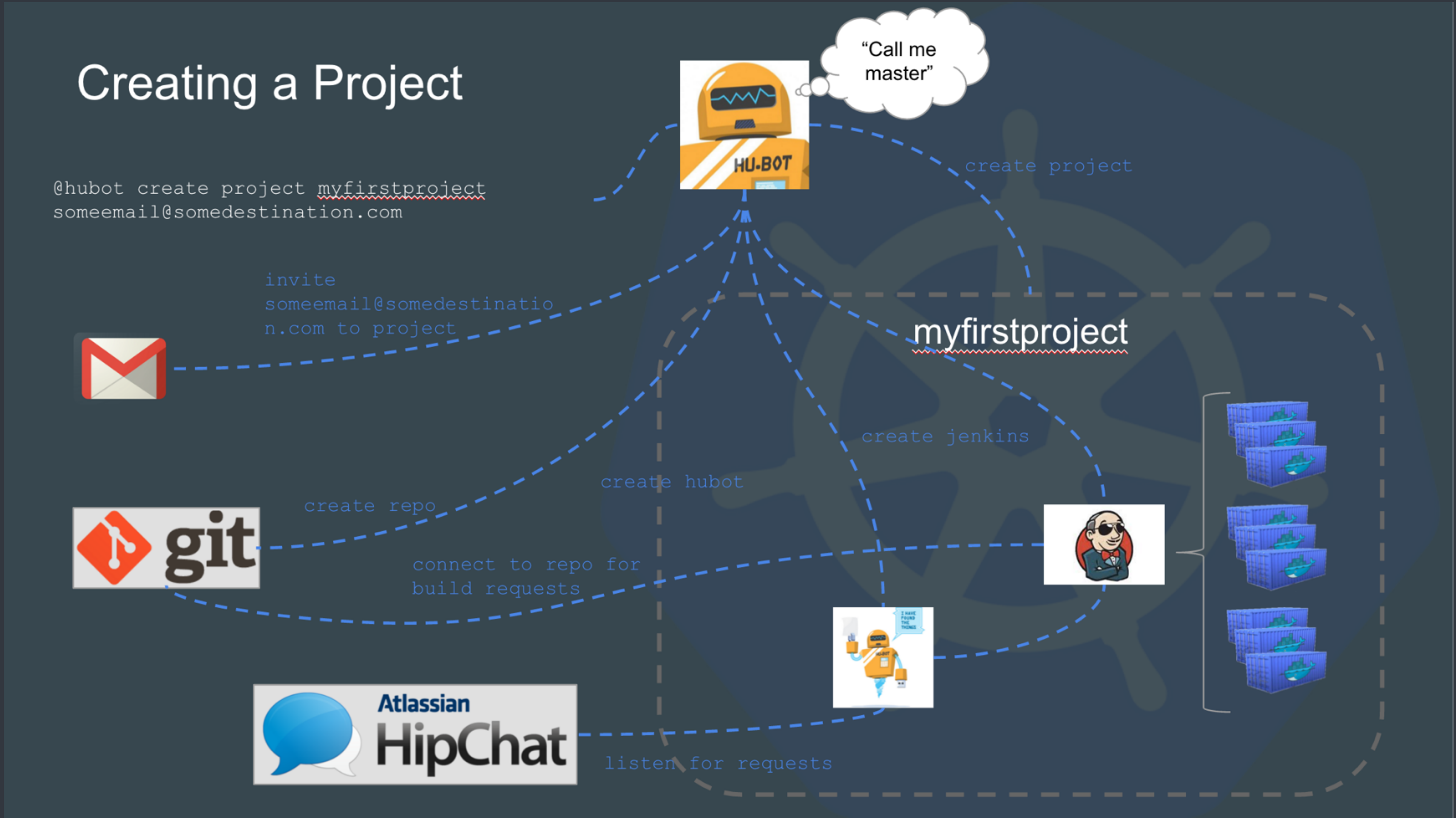
Enable cross-namespace ingress.
krancour committed on Nov 18, 2015

Python Client

Kubernetes Pack

Ingress Controller

ChatOps Integration



Creating a Project

```
@hubot create project myfirstproject
someemail@somedestination.com
```

create project

invite someemail@somedestination.com to project

myfirstproject

create jenkins

create hubot

create repo

connect to repo for build requests

listen for requests

DEMO?

Business Justification

Project Justification

Business
Security
Developers
Quality & Performance

🟦 **Business**

Cost

🟦 **Developers**

Management, Time,
Ease of use

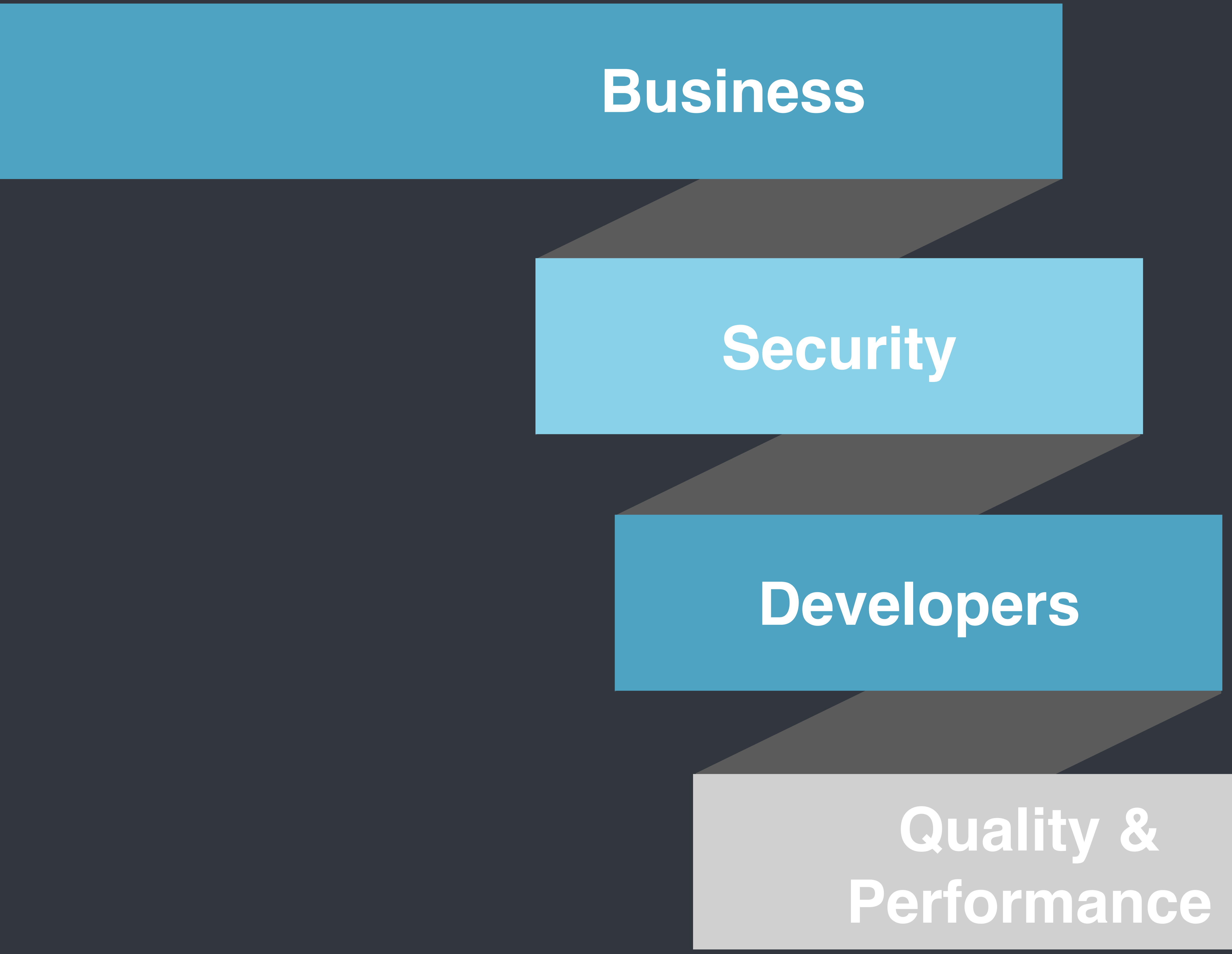
🟦 **Security**

Standardization,
Compliance and
Visibility

⬜ **QA & Perf**

Ease of Use, Agility,
Minimal disruption

Project Justification



Business

Security

Developers

Quality & Performance

Business

Cost

Build pipeline takes around 3 months for two Devs. Costing on average $50,000 just in development time PER DEV TEAM. (QA, Security, Performance not included)

400 Dev Teams x $50,000 = $20 Million

Project Justification

Business
Security
Developers
Quality & Performance

■ **Business**

Recurring Cost

10-20% of a Devs time on maintenance, changes, upkeep, troubleshooting

At 10%

400 X $10,000 = 2 Million/yr

Just because I can

"Quotes from Developers"

"Well we spent 6 months solid building a good starting point"

Pearson Developer

> "we never upgrade Jenkins once stable cause we can't get time for it"

Pearson Developer

Stakeholder Requirements

Reduce Migration Costs

Standardization
"same build process across whole Pearson estate"

Performance Testing
"automated perf testing"

< 5% Time required to Manage

Compliance
"pre-baked security"

Visibility

Ease of Use
"if you don't get this I can't help you"

Quality Testing

Our Requirements

Geographically Distributed

"Abstract Deployment but allow it anywhere"

Fast

"< 1hr"

Self Configuring

Cattle

"Jenkins as Cattle"

Automated Upgrades

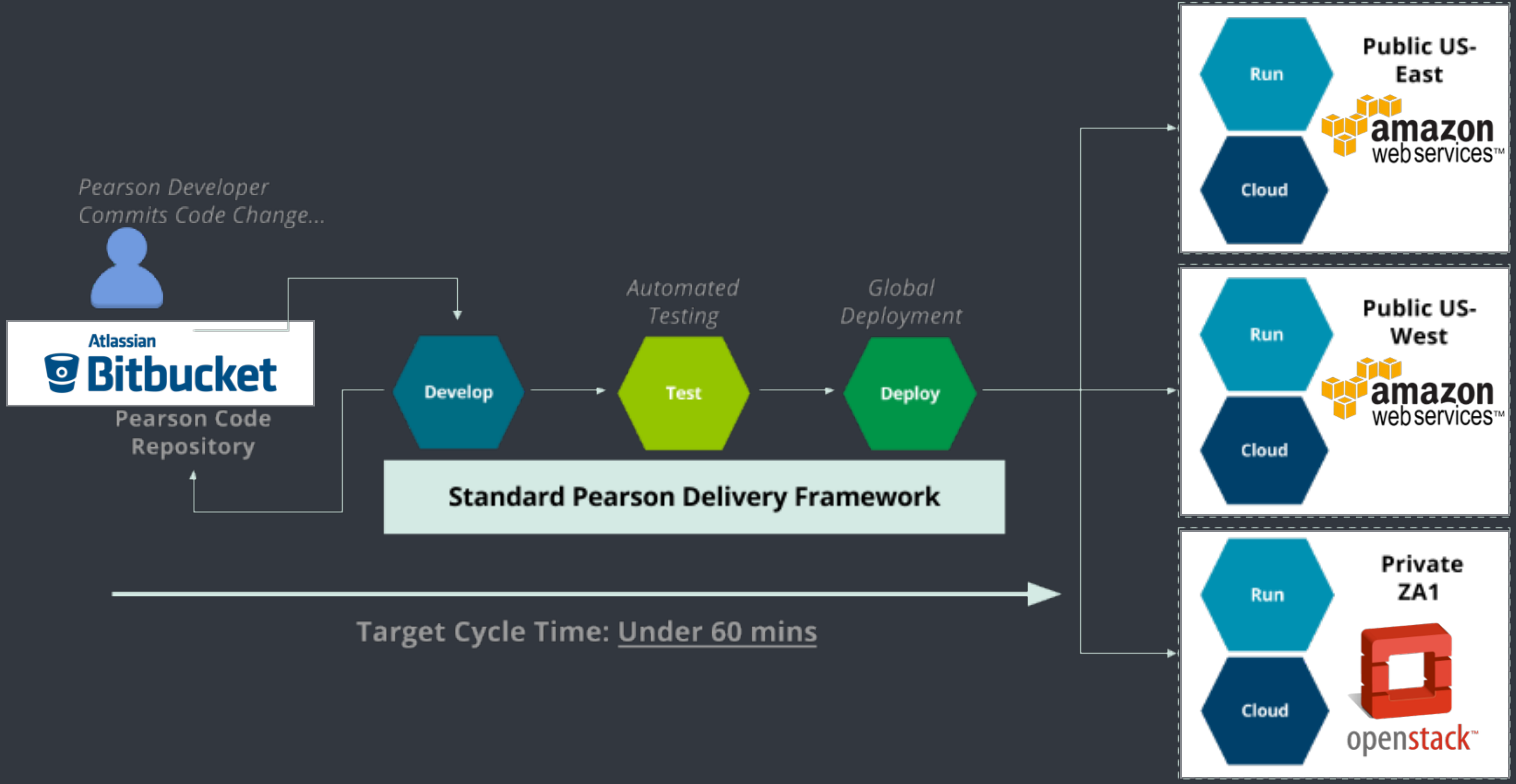
Scalability

"Horizontal"

Feedback Loops

"Clear communication across teams"

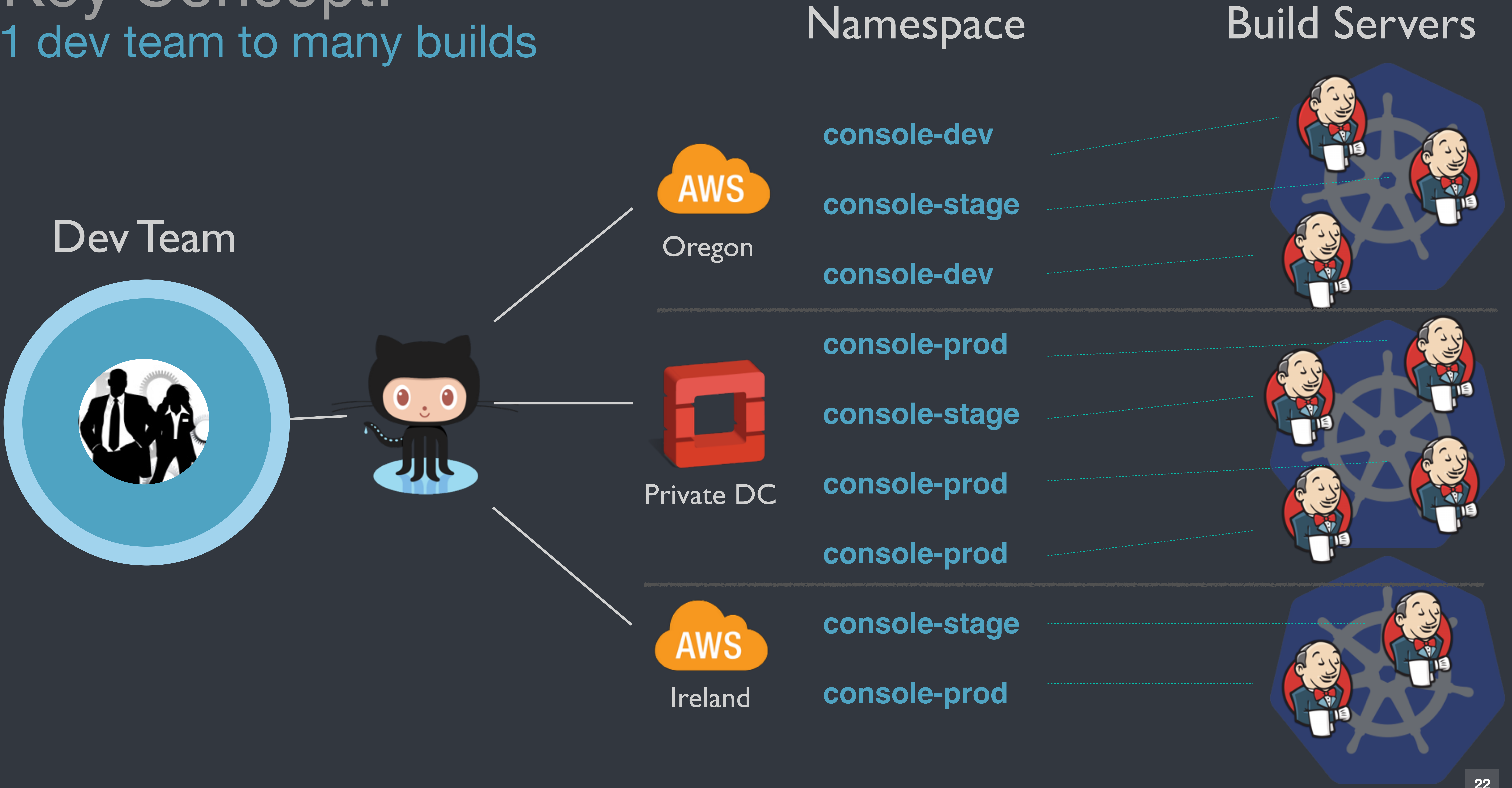
High Level Product Design



Build Pipelines

Key Concept:

1 dev team to many builds



Standard Process

Base_Builds

App_Builds

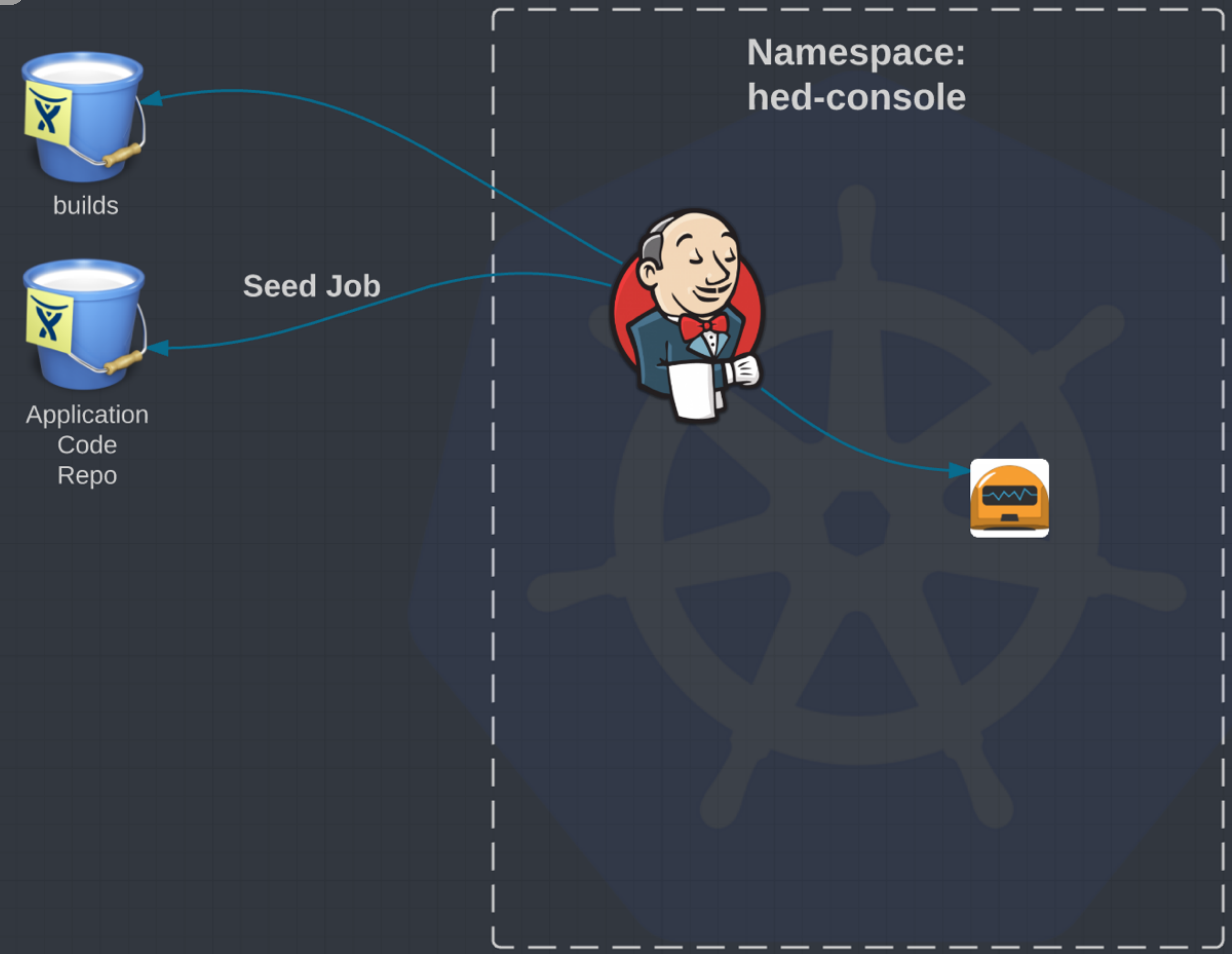
Standard Process

Standard Process

Standard Process

Base_Builds

App_Builds



Namespace:
hed-console

builds

Application
Code
Repo

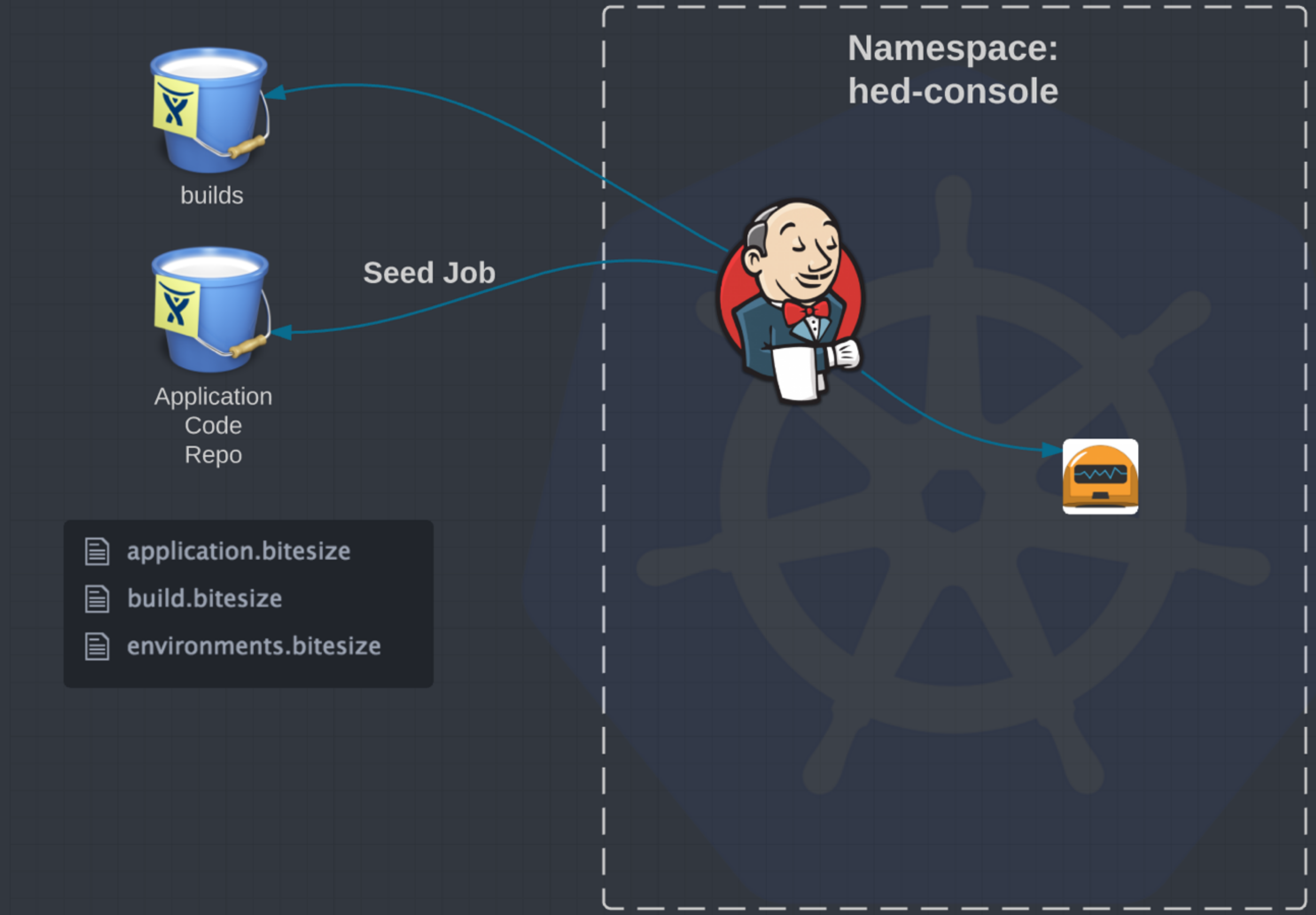
Seed Job

Standard Process

Standard Process

Base_Builds

App_Builds

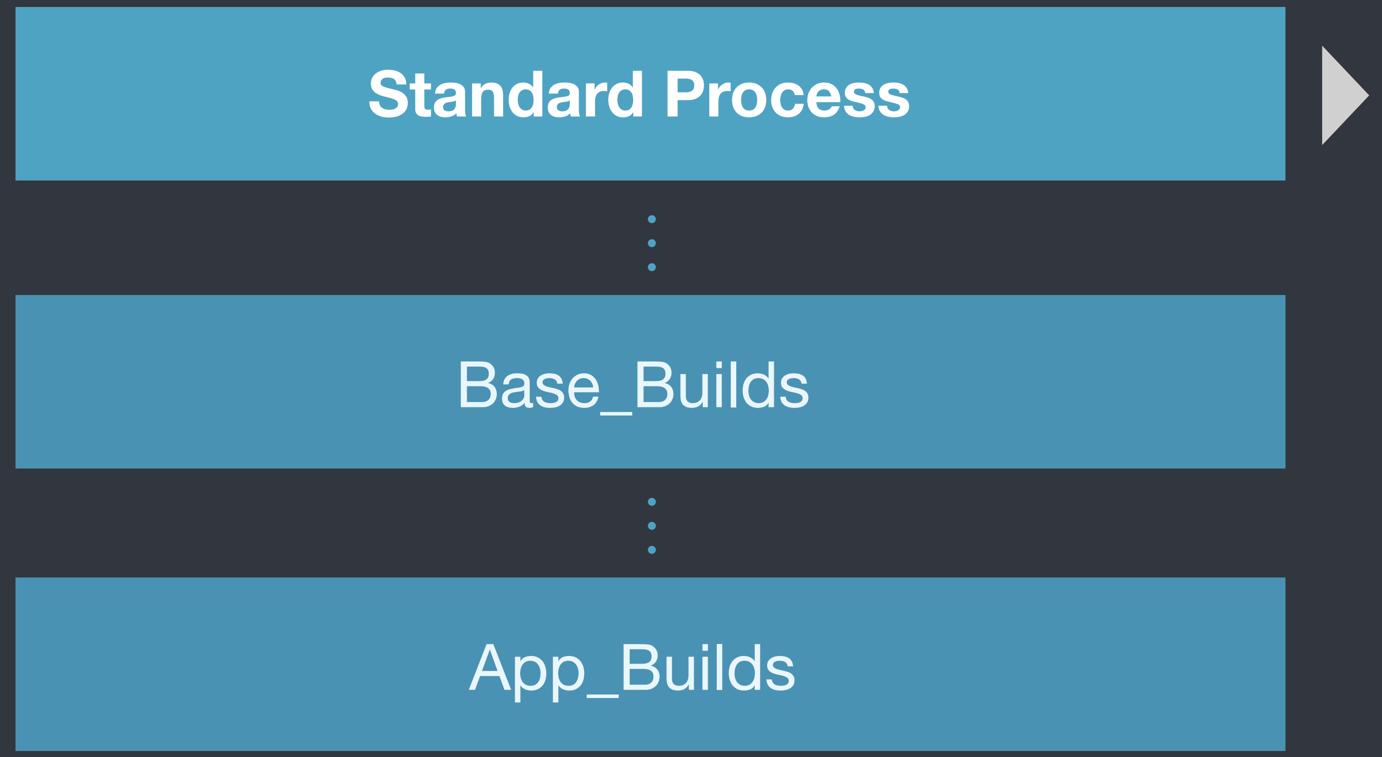


application.bitesize

build.bitesize

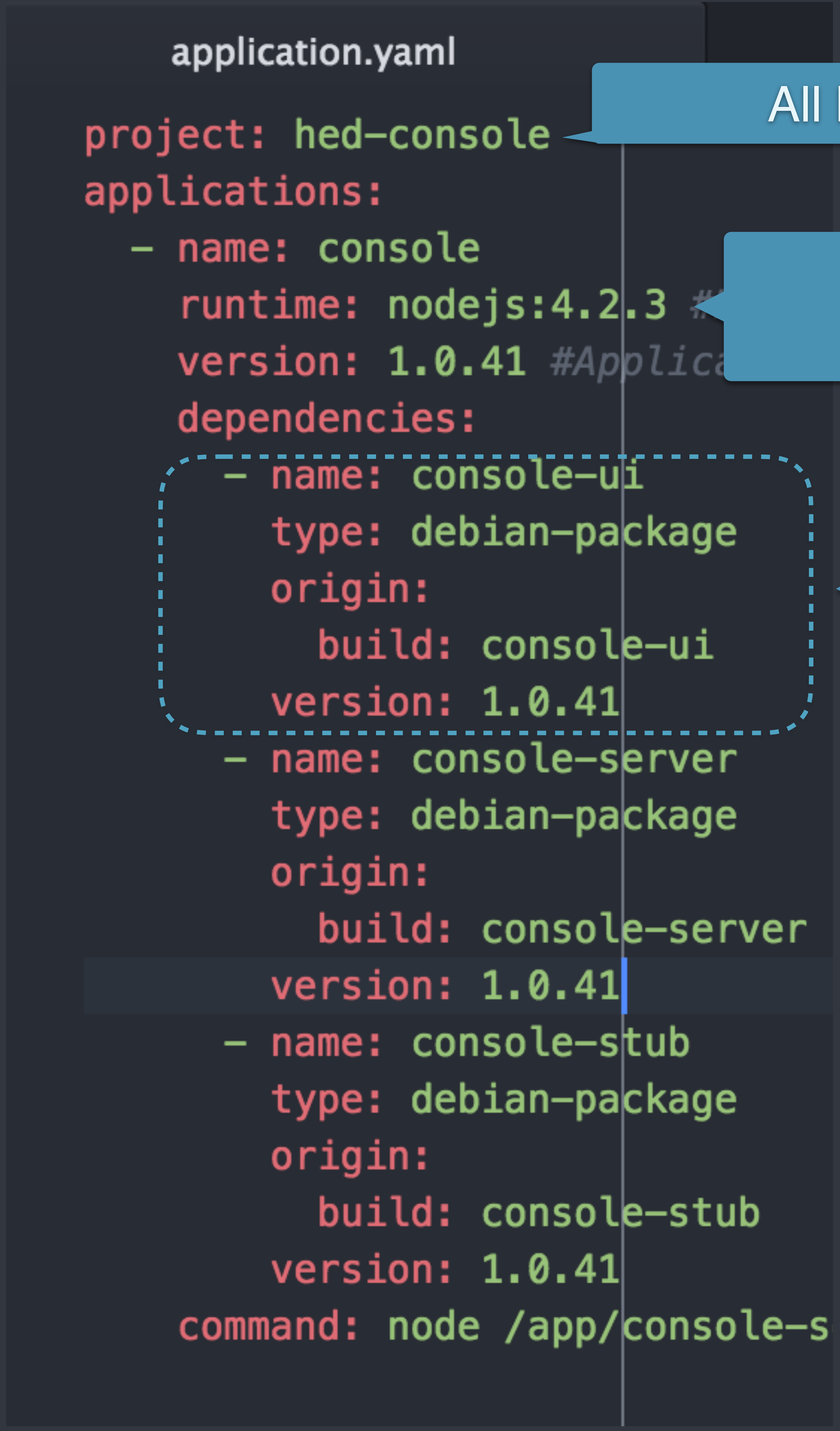
environments.bitesize

Standard Process


Standard Process

Base_Builds

App_Builds


application.yaml

```
project: hed-console
applications:
  - name: console
    runtime: nodejs:4.2.3 #
    version: 1.0.41 #Applica
    dependencies:
      - name: console-ui
        type: debian-package
        origin:
          build: console-ui
        version: 1.0.41
      - name: console-server
        type: debian-package
        origin:
          build: console-server
        version: 1.0.41
      - name: console-stub
        type: debian-package
        origin:
          build: console-stub
        version: 1.0.41
    command: node /app/console-s
```

All based on project name

base image

point to build component

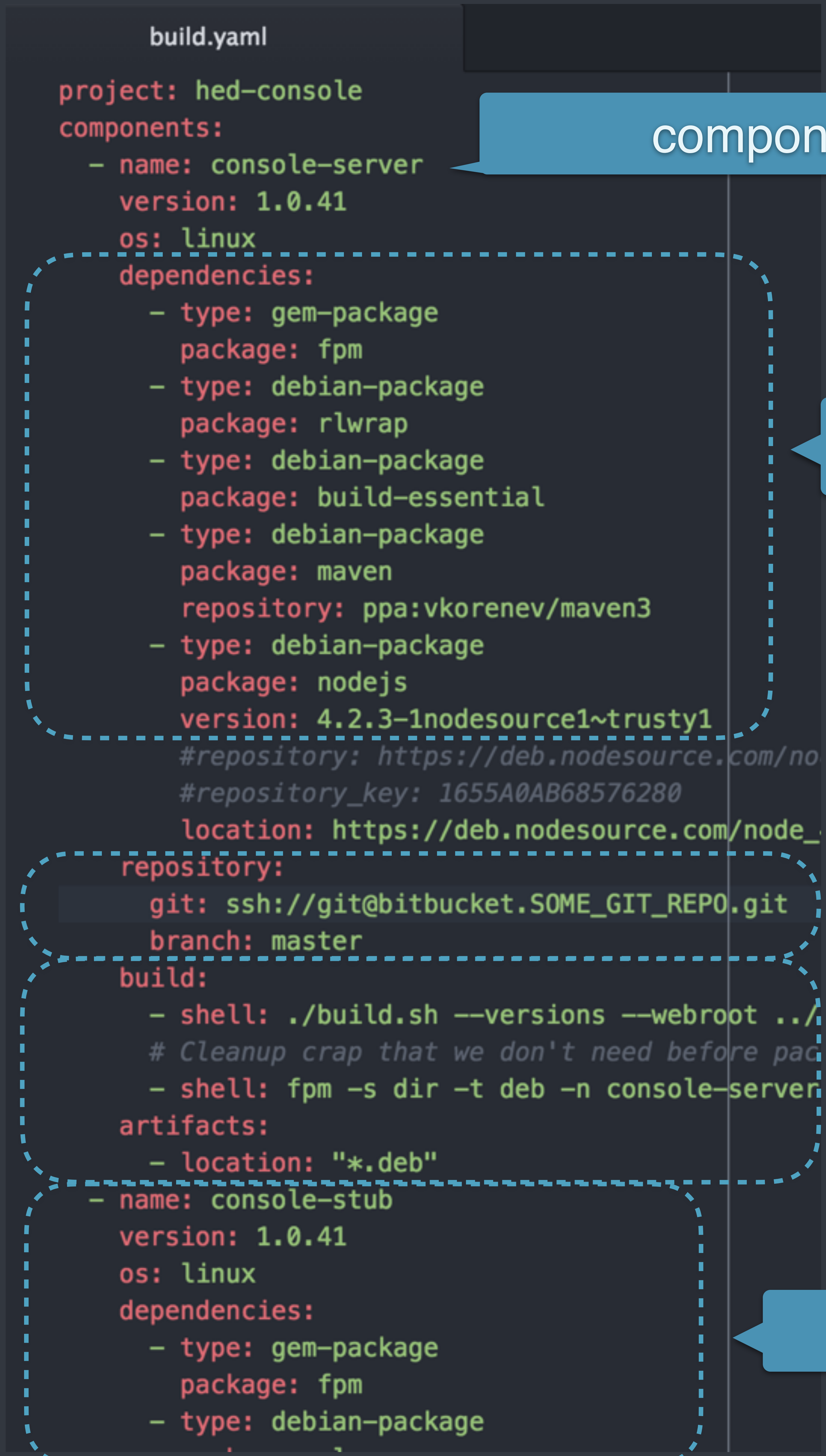
runtime command

Standard Process

Standard Process

Base_Builds

App_Builds



build.yaml

```yaml
project: hed-console
components:
  - name: console-server
    version: 1.0.41
    os: linux
    dependencies:
      - type: gem-package
        package: fpm
      - type: debian-package
        package: rlwrap
      - type: debian-package
        package: build-essential
      - type: debian-package
        package: maven
        repository: ppa:vkorenev/maven3
      - type: debian-package
        package: nodejs
        version: 4.2.3-1nodesource1~trusty1
        #repository: https://deb.nodesource.com/no
        #repository_key: 1655A0AB68576280
        location: https://deb.nodesource.com/node_
    repository:
      git: ssh://git@bitbucket.SOME_GIT_REPO.git
      branch: master
    build:
      - shell: ./build.sh --versions --webroot ../
      # Cleanup crap that we don't need before pac
      - shell: fpm -s dir -t deb -n console-server
    artifacts:
      - location: "*.deb"
  - name: console-stub
    version: 1.0.41
    os: linux
    dependencies:
      - type: gem-package
        package: fpm
      - type: debian-package
```

component: code segment

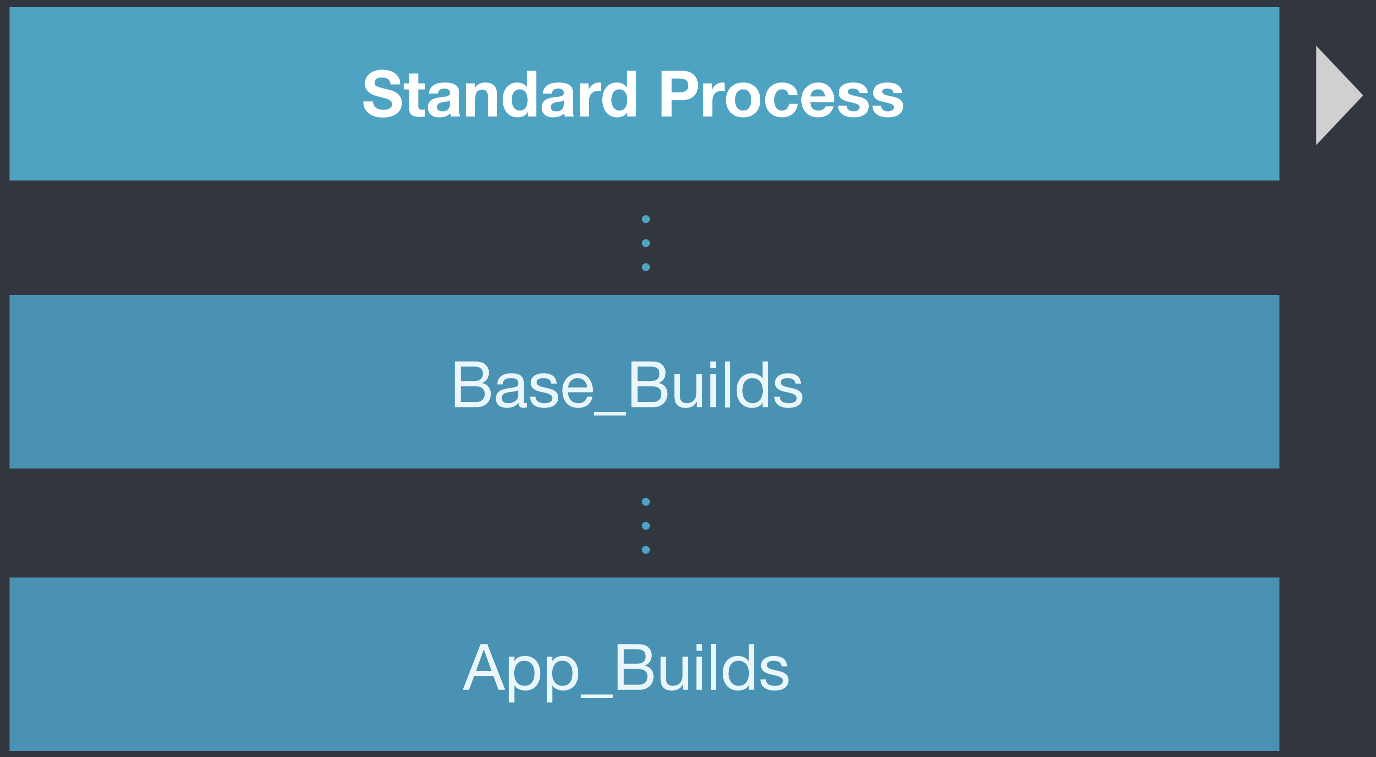
dependencies required for build

repo and branch location

build commands for component

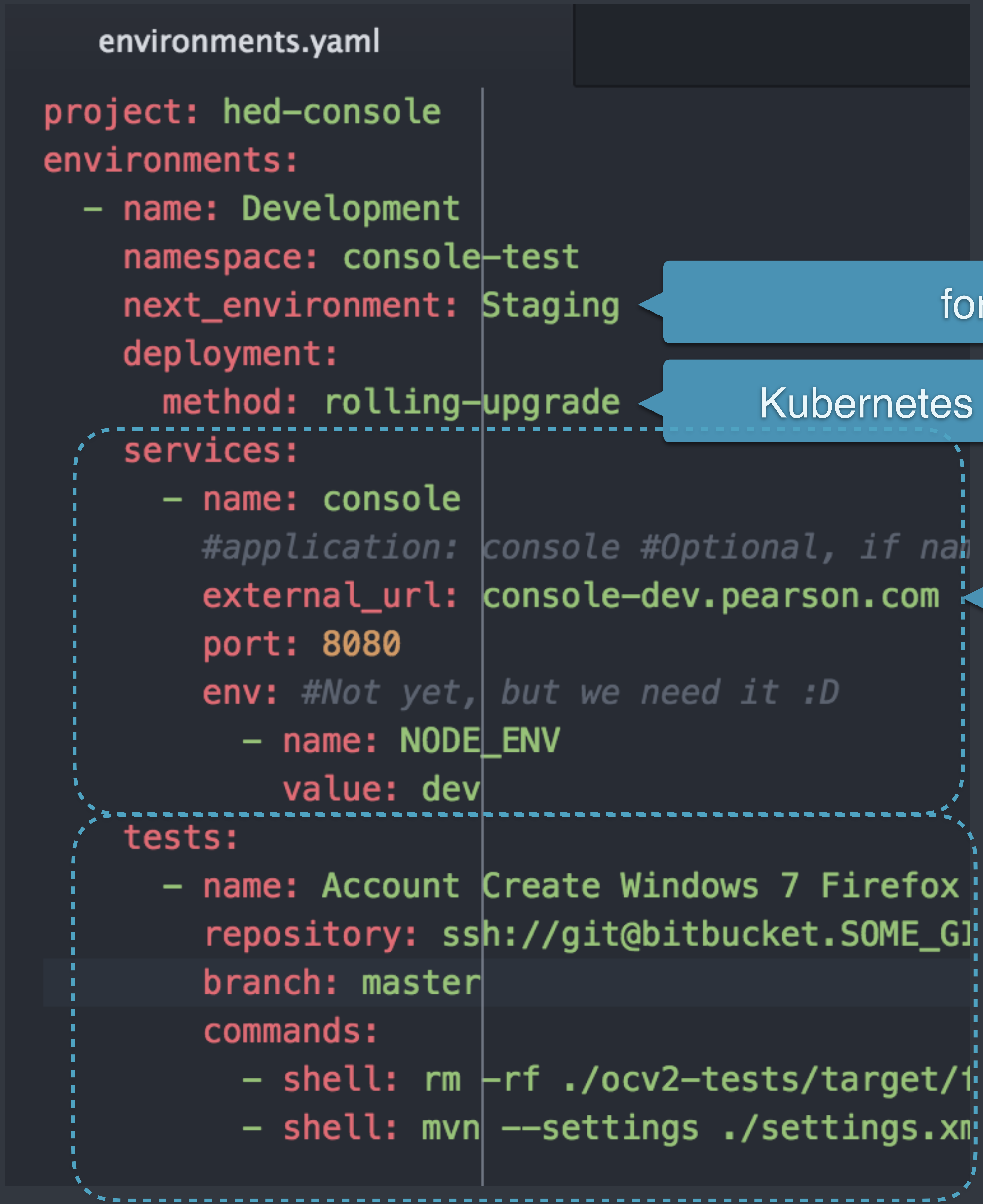
another piece of code for build

Standard Process


Standard Process

Base_Builds

App_Builds



```
project: hed-console
environments:
  - name: Development
    namespace: console-test
    next_environment: Staging
    deployment:
      method: rolling-upgrade
    services:
      - name: console
        #application: console #Optional, if nam
        external_url: console-dev.pearson.com
        port: 8080
        env: #Not yet, but we need it :D
          - name: NODE_ENV
            value: dev
    tests:
      - name: Account Create Windows 7 Firefox
        repository: ssh://git@bitbucket.SOME_GI
        branch: master
        commands:
          - shell: rm -rf ./ocv2-tests/target/t
          - shell: mvn --settings ./settings.xm
```

for CI/CD

Kubernetes Upgrade methon

Configures Service and Ingress

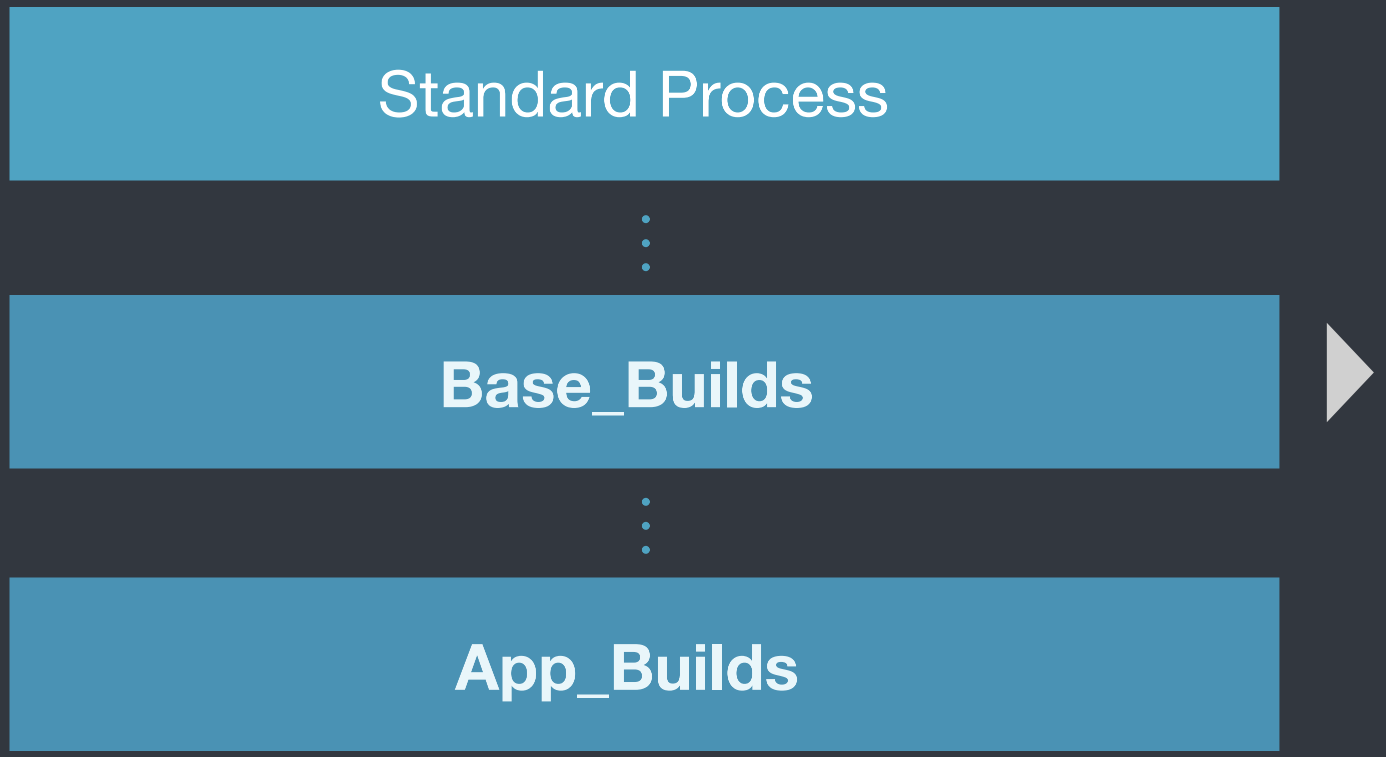
Tests to run in Build Process

Requirements

- ✓ Geographically Distributed
- Fast
- ✓ Self Configuring
- Feedback Loops
- ✓ Standardization
- Performance Testing
- < 5% Time Management required
- ✓ Ease of Use

- ✓ Cattle
- Automated Upgrades
- ✓ Scalability
- ✓ Reduce Migration Costs
- Compliance
- Visibility
- Quality Testing

Base Builds



Standard Process

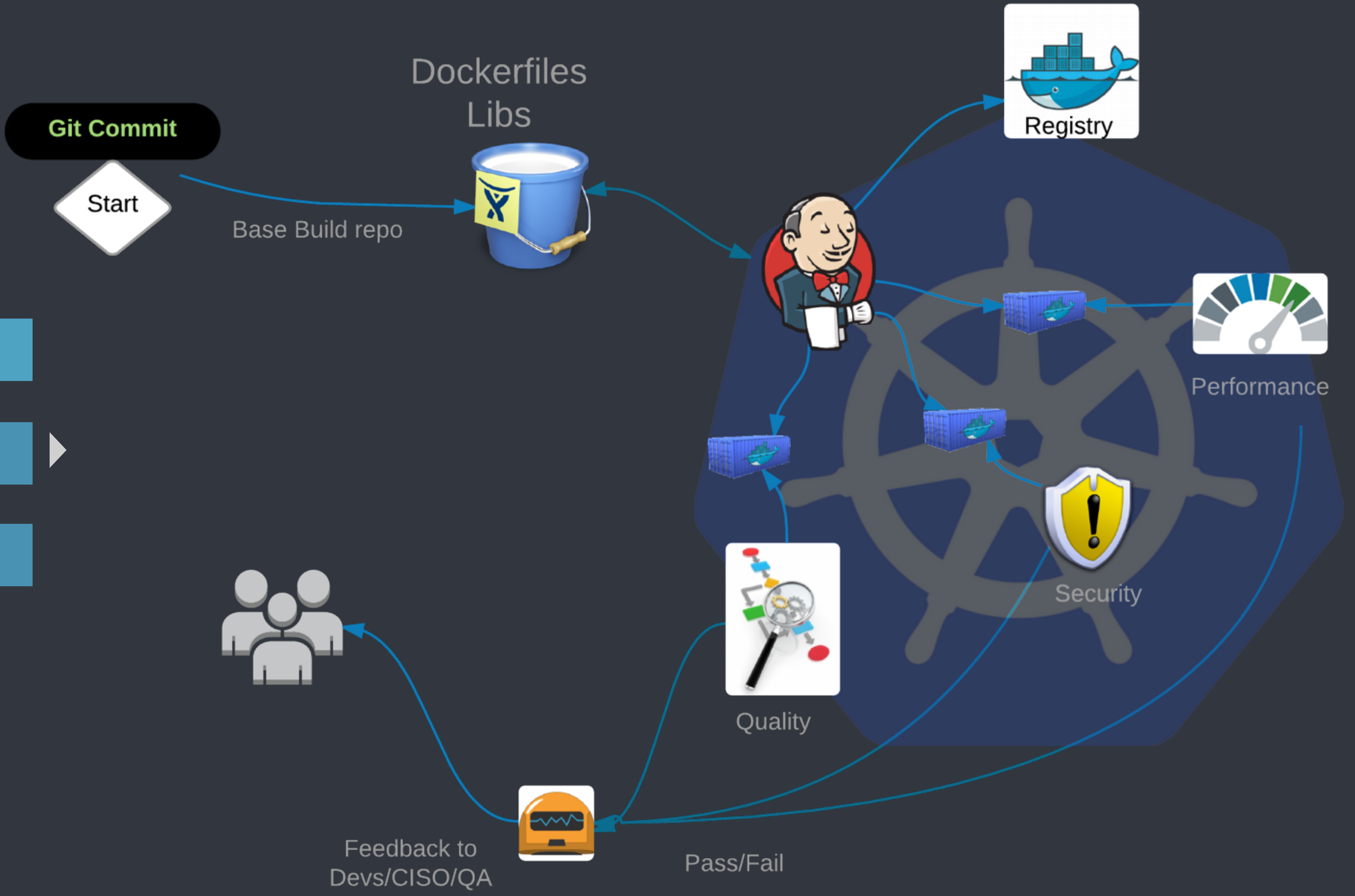
Base_Builds

App_Builds

Standard Process

Base_Builds

App_Builds

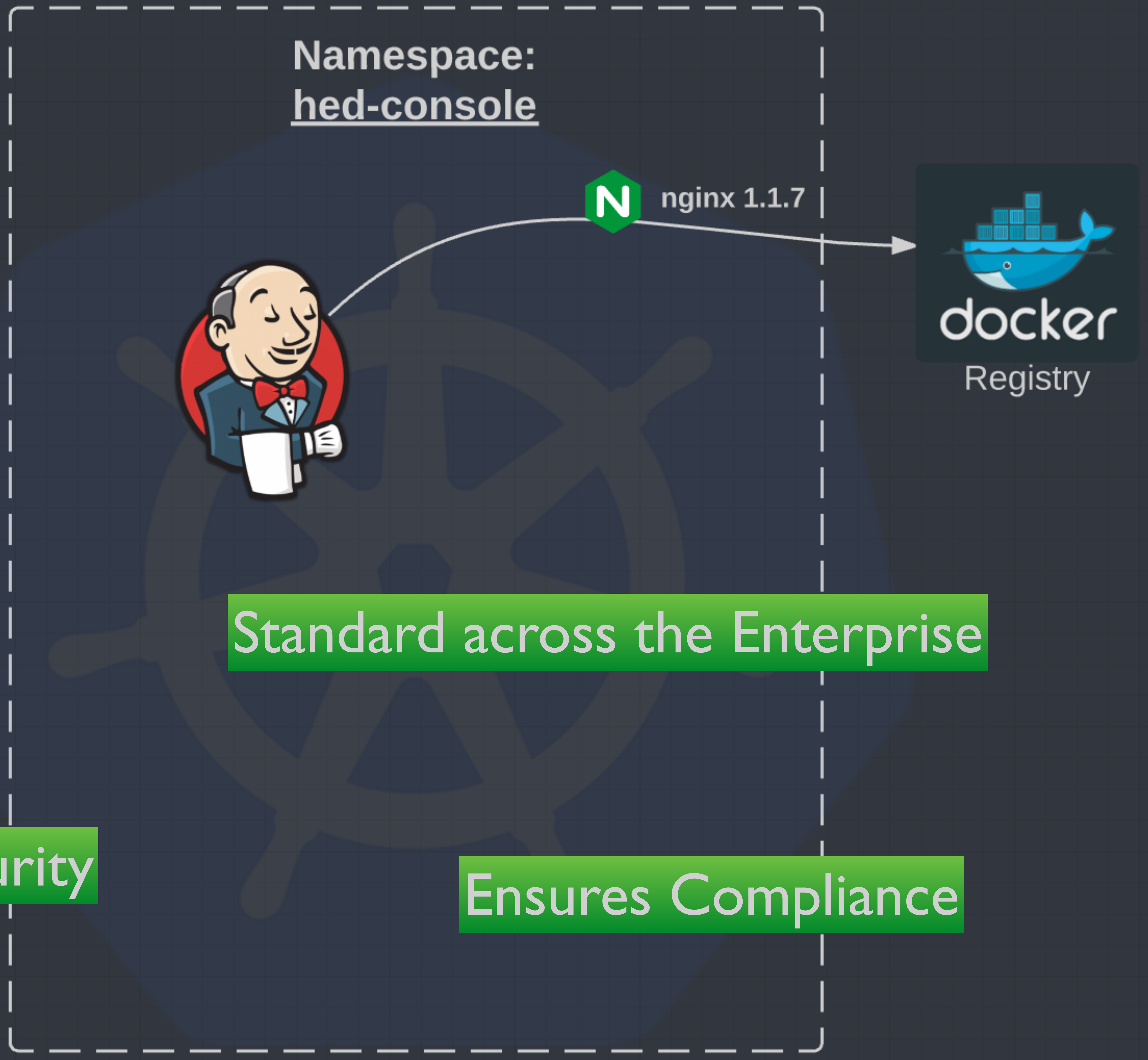


Base Build

Standard Process

Base_Builds

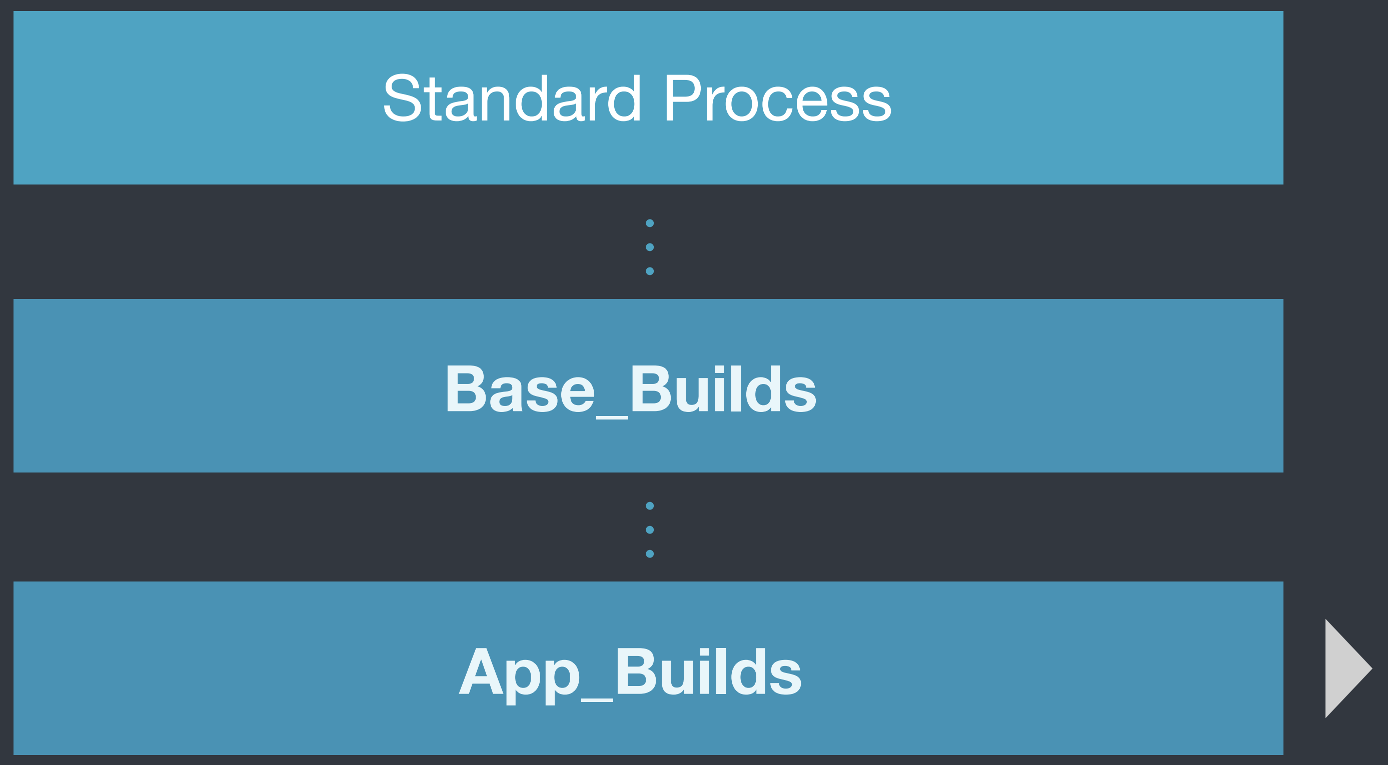
App_Builds



Standard across the Enterprise

Base level of Security

Ensures Compliance



Standard Process

Base_Builds

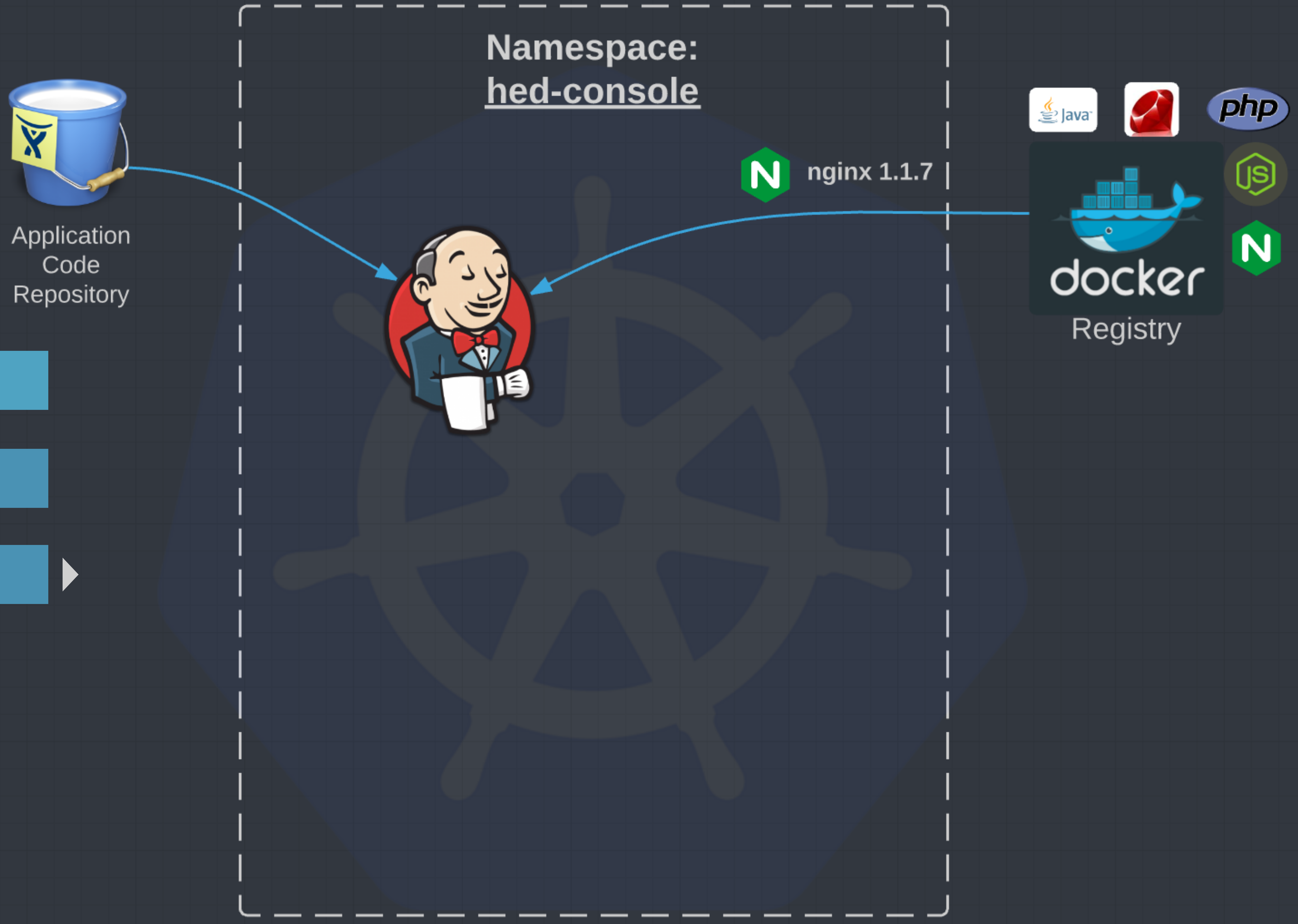
App_Builds

App Builds

Standard Process

Base_Builds

App_Builds



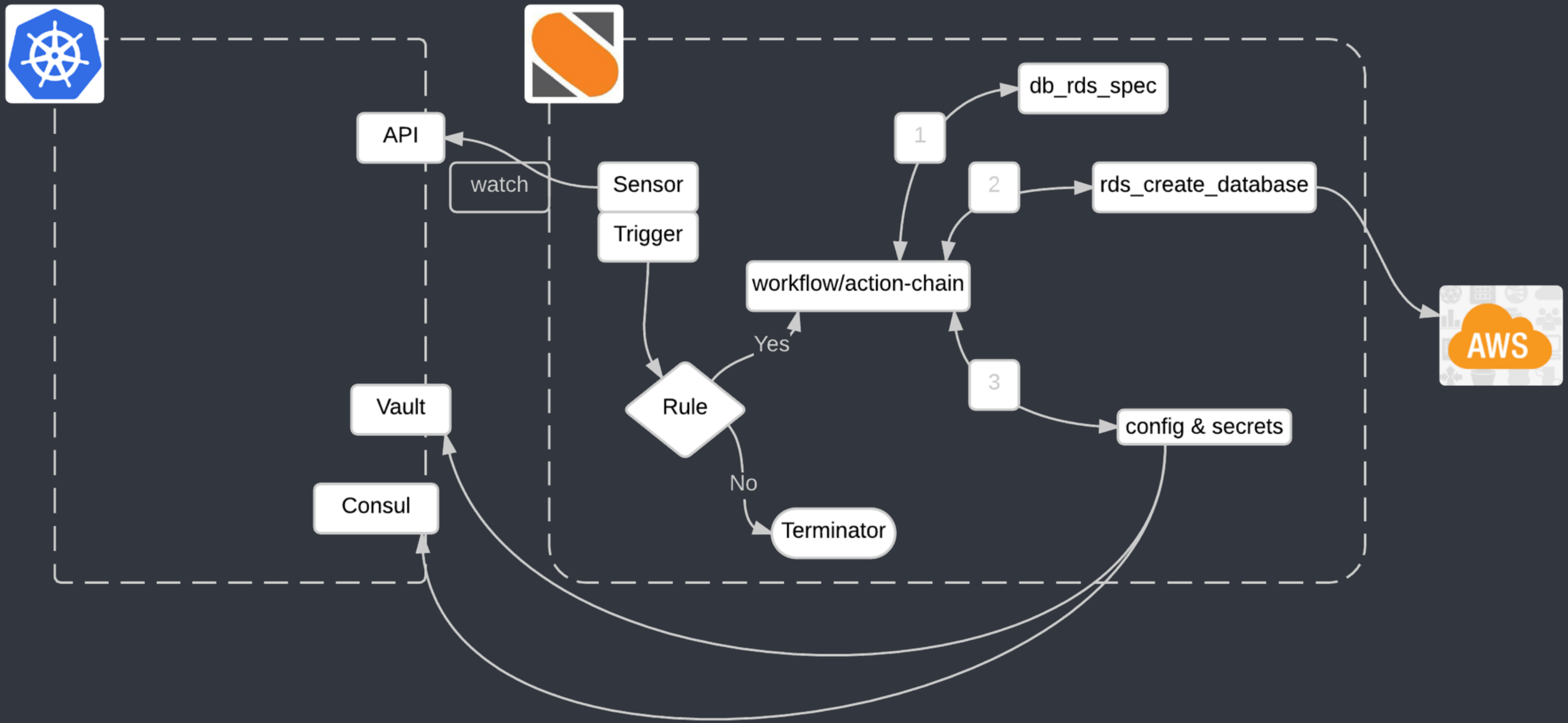
Requirements

- ✓ Geographically Distributed
- ✓ Fast
- ✓ Self Configuring
- Feedback Loops
- ✓ Standardization
- ✓ Performance Testing
- < 5% Time Management required
- ✓ Ease of Use

- ✓ Cattle
- Automated Upgrades
- ✓ Scalability
- ✓ Reduce Migration Costs
- ✓ Compliance
- Visibility
- ✓ Quality Testing

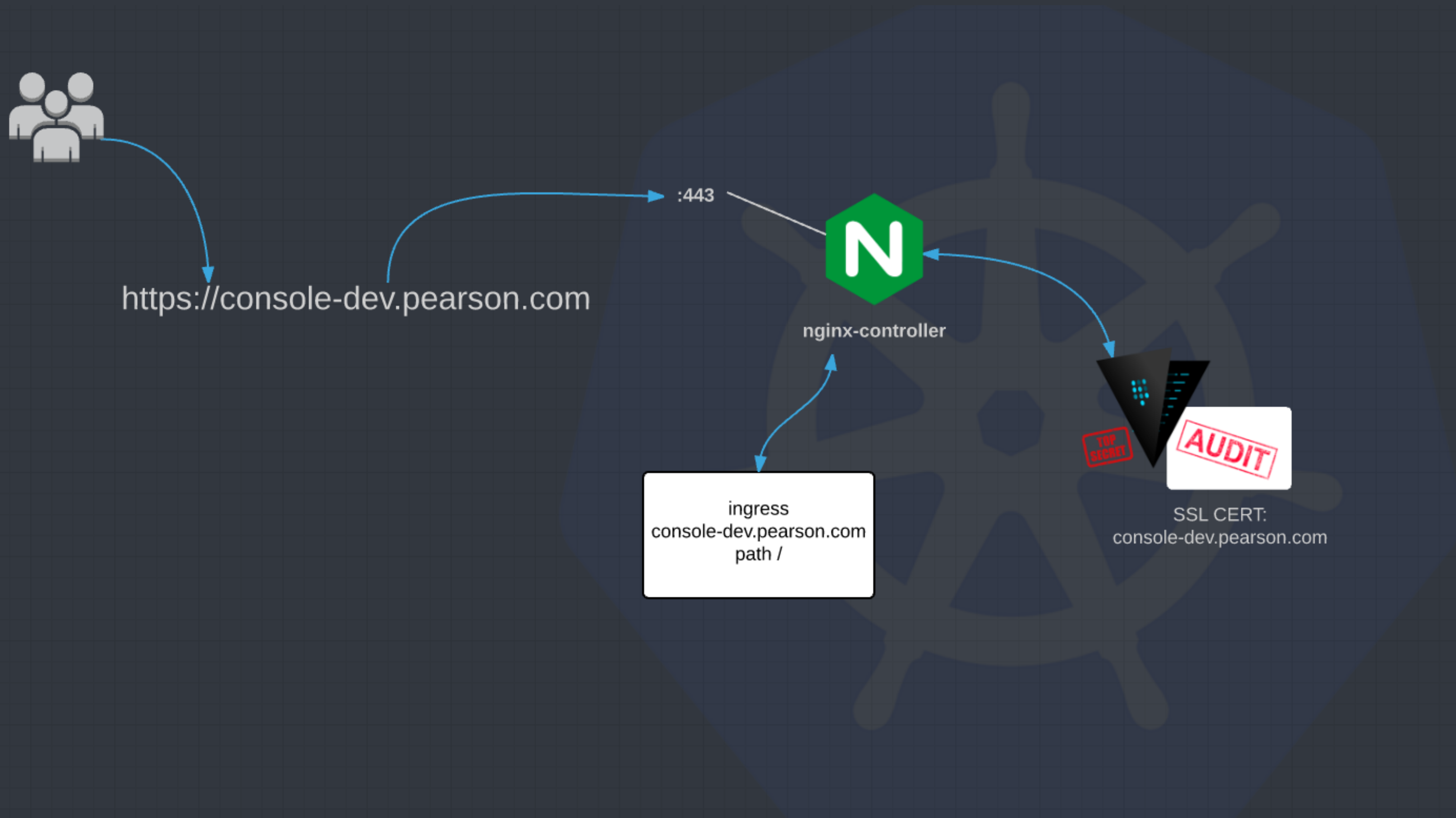
What else?

Manage External Resources with Kubernetes



DEMO?

HTTPS and automated integration with Vault for SSL Certs



https://console-dev.pearson.com

:443

nginx-controller

ingress
console-dev.pearson.com
path /

SSL CERT:
console-dev.pearson.com

StackStorm Integration - Michael Ward

https://github.com/StackStorm/st2contrib/tree/master/packs/kubernetes

Vault SSL Integration - Martin Devlin

https://github.com/devlinmr/contrib

Jenkins Build Pipelines - Simas Cepaitis

OpenSource Coming Soon

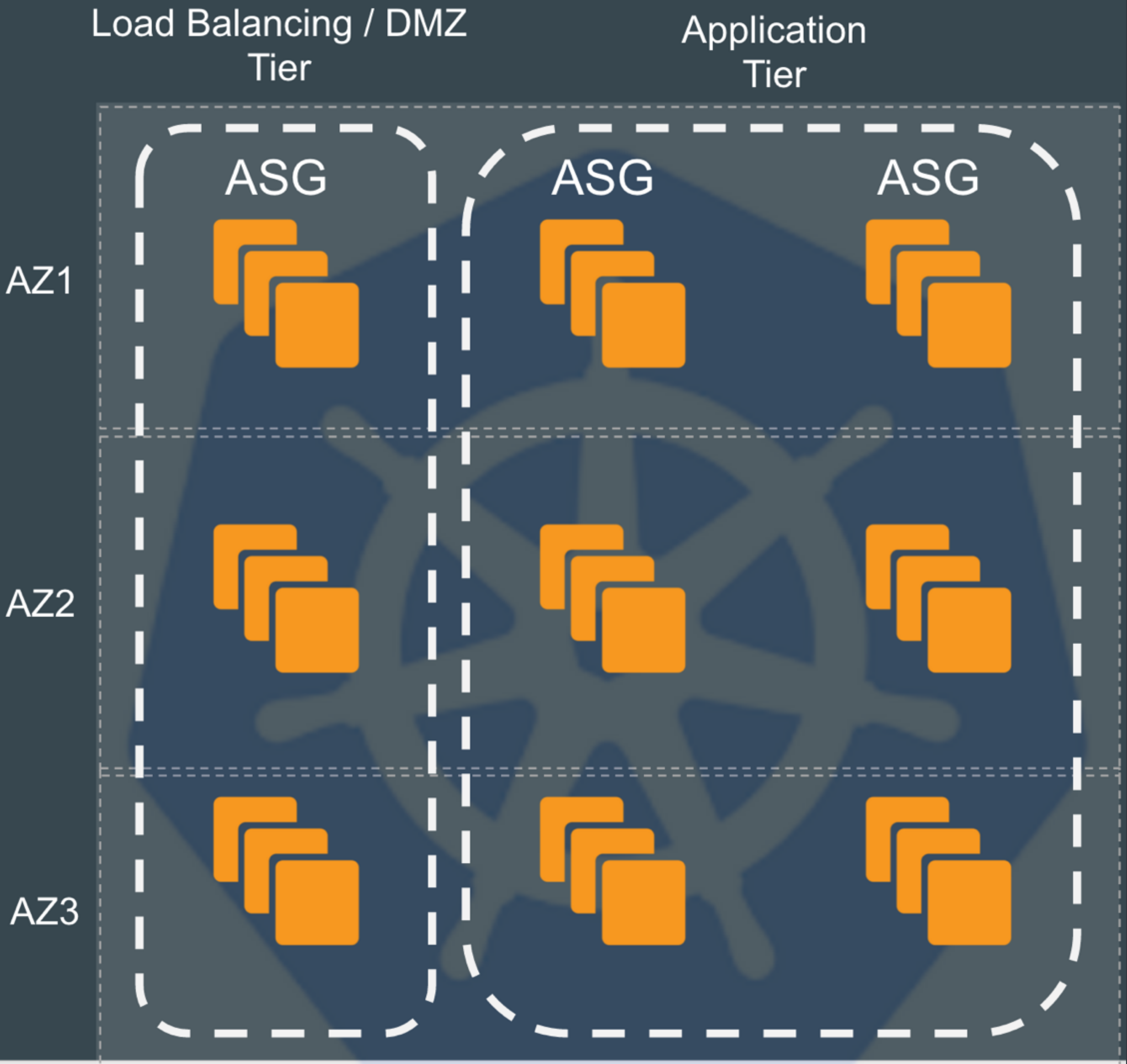
ChatOps - Simas Cepaitis & Michael Ward

OpenSource Coming Soon

ETCD restore
to alternate clusters

Highly Available Multi-AZ

- Distributed across multiple AZs for fault tolerance
- Autoscaling Groups (ASGs) for monitoring, scaling and desired state
- ASG per tier allows for resource differentiation and scaling for capacity requirements
- All within the Kubernetes cluster to simplify internal networking and orchestration



Meet the Team

Chris Jackson

Director

Mark Stocker

PM

Peter Idah

Engineer

Simas Cepaitis

Engineer

John Shirley

Engineer

Martin Devlin

Engineer

Bill Jorgensen

Engineer

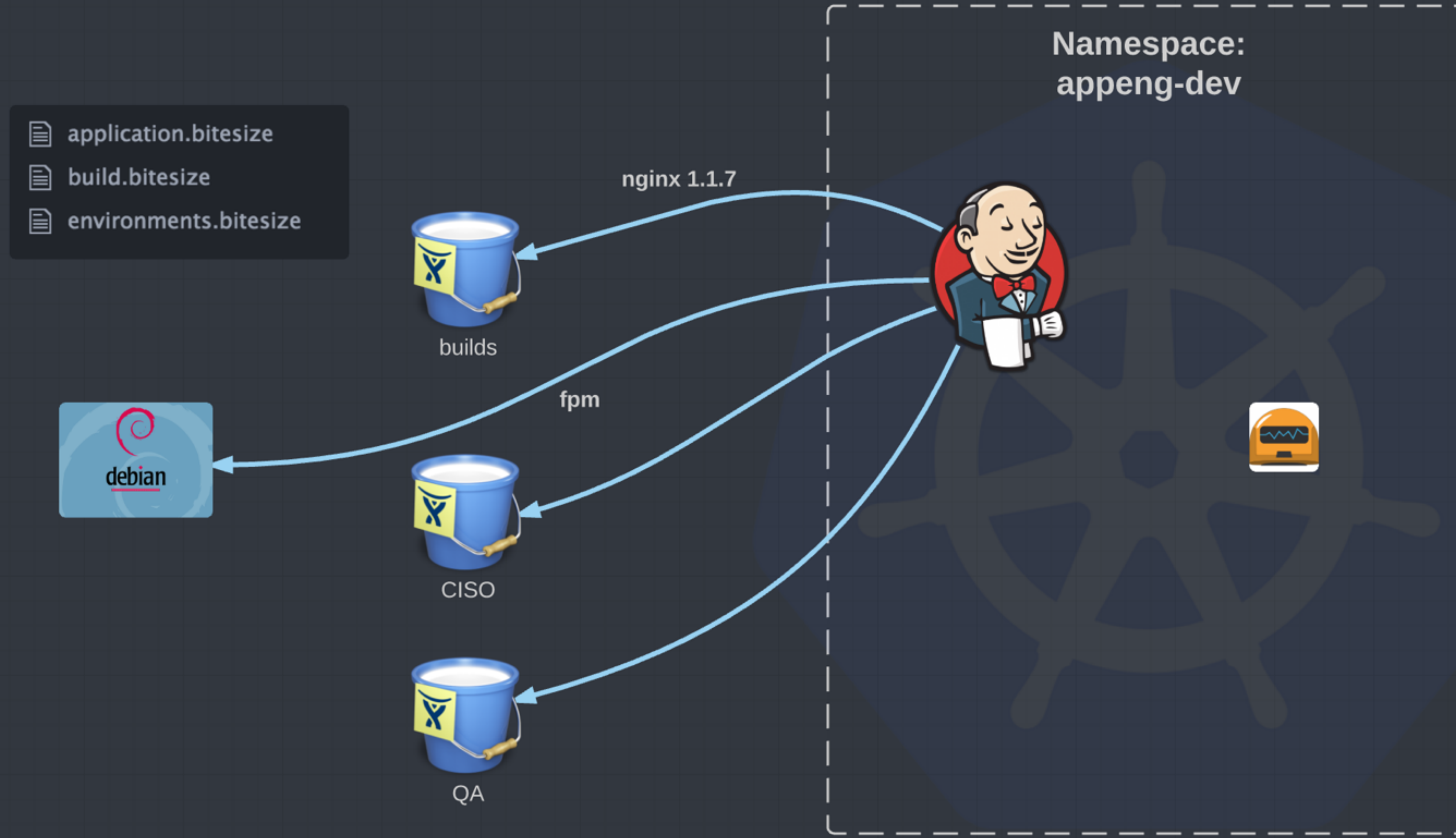
Questions

Standard Process

Standard Process

Base_Builds

App_Builds



application.bitesize

build.bitesize

environments.bitesize

PaaS as an ecosystem

Build

Event Driven Automation

Orchestration

Secrets

Runtime

App Config

API Management

Monitoring

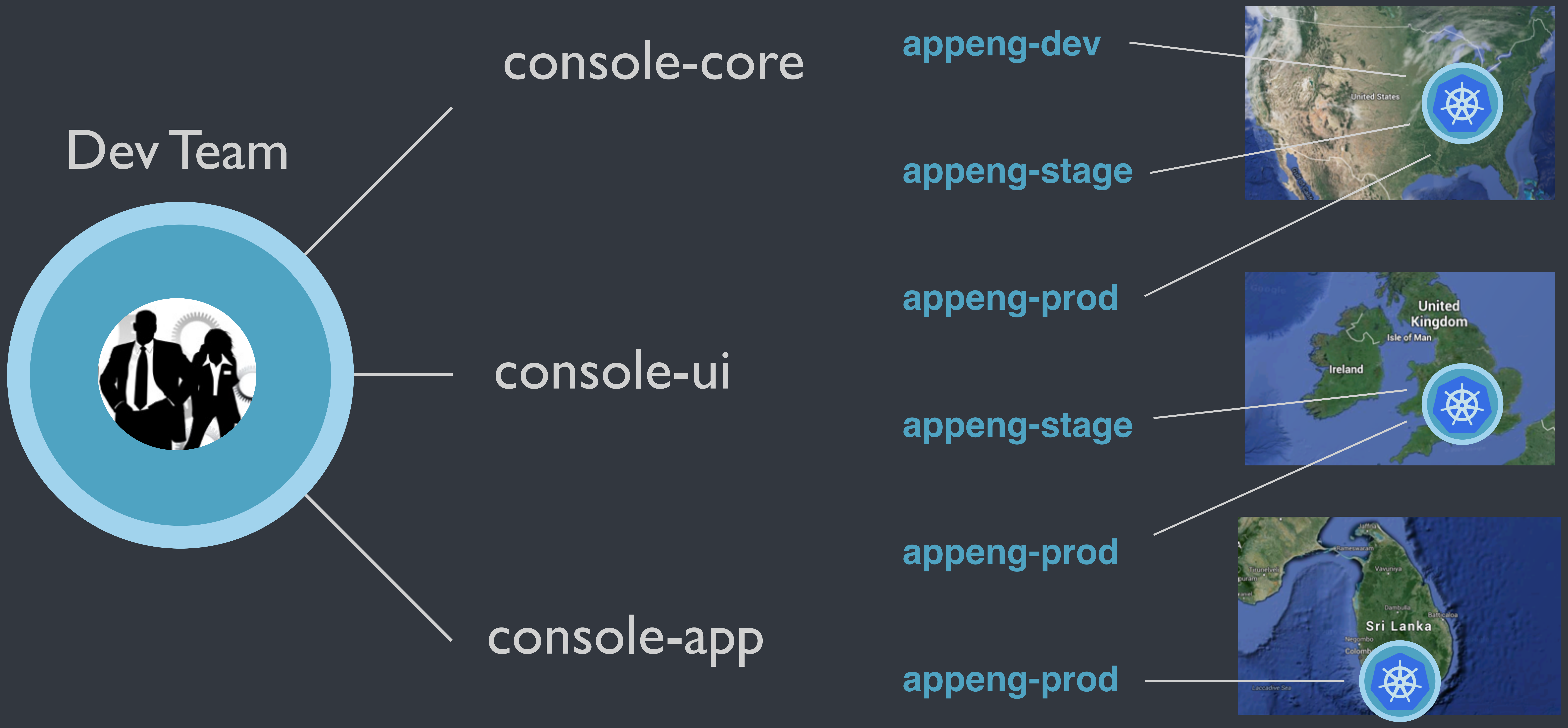
Key Concept:

1 dev team to many namespaces

Project

Namespace

Kubernetes Cluster



Dev Team

console-core

console-ui

console-app

appeng-dev

appeng-stage

appeng-prod

appeng-stage

appeng-prod

appeng-prod