Cloud Native CI/CD

at Lunar Way

lunar way®

$ whoami

Emil Krog Ingerslev

- Cloud Architect / Site Reliability Engineer @lunarway
- Full stack experience in lots of languages including Node.js, C#, Elixir, Go, PHP, ASP
- Used CI/CD with TeamCity, Buildkite, Octopus, Jenkins, Powershell, bash, fish, CircleCI, Travis and more
- Previous: Lix, Cetrea, d60, Ennova, Energy Systems
- started blogging at https://ingerslev.io/
- home automation / food / gadget geek

@emilingerslev emilingerslev @emilingerslev

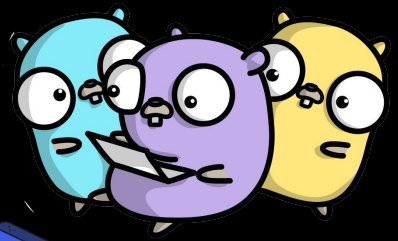
What is Lunar Way?

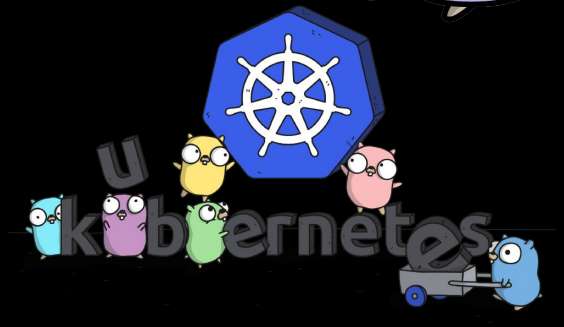
We're living in the era of mobile/digital only – we believe banking and commerce should to.

Our vision is to rethink the interaction with money and defining a completely new category - by introducing a new money app.

Lunar Ways infrastructure

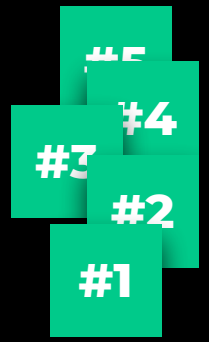
- microservices
  - +50 go services
  - +15 node.js services
  - a small bunch of java & other language services
- several internal libraries
- running (mostly) everything on k8s
  - nodes on AWS
  - provisioned with kops
- a bunch of cloud native tools
  - postgres, rabbitmq, humio, nginx-ingress-controller, and a lot more





Let’s go through some iterations that …

- … we have already done
- … we’re working on right now
- … we’re planning



lunar way

#1 Migrating from hundreds of build scripts

lunar way

#1

The old project structure

```
├── kubernetes
│   ├── dev
│   │   ├── configmap.yaml
│   │   ├── ingress.yaml
│   │   ├── db-configmap.yaml
│   │   └── ingress.yaml
│   ├── staging
│   │   ├── configmap.yaml
│   │   ├── ingress.yaml
│   │   ├── db-configmap.yaml
│   │   └── ingress.yaml
│   ├── prod
│   │   ├── configmap.yaml
│   │   ├── ingress.yaml
│   │   ├── db-configmap.yaml
│   │   └── ingress.yaml
│   ├── deployment.sh
│   ├── env.sh
│   ├── secrets.sh
│   └── verify.sh
├── Gopkg.lock
├── Gopkg.toml
├── Jenkinsfile
├── Makefile
└── ... project files etc. ...
```

lunar way

#1

- `Makefile`'s are pretty great but
  - ✗ not nice to have 100's of them to manage
- Local kubernetes resources are great too, but
  - ✗ not nice to improve them in 100's of repo's when general infrastructure needs change
- Being able to add custom build targets and customize each part of build and deploy is awesome
  - ✗ but doing subtle changes in the middle of otherwise shared code is confusing and hard to spot

```
├── kubernetes
│   ├── dev
│   │   ├── configmap.yaml
│   │   ├── ingress.yaml
│   │   ├── db-configmap.yaml
│   │   └── ingress.yaml
│   ├── staging
│   │   ├── configmap.yaml
│   │   ├── ingress.yaml
│   │   ├── db-configmap.yaml
│   │   └── ingress.yaml
│   ├── prod
│   │   ├── configmap.yaml
│   │   ├── ingress.yaml
│   │   ├── db-configmap.yaml
│   │   └── ingress.yaml
│   ├── deployment.sh
│   ├── env.sh
│   ├── secrets.sh
│   └── verify.sh
├── Gopkg.lock
├── Gopkg.toml
├── Jenkinsfile
├── Makefile
└── ... project files etc. ...
```

lunar way

#1

Our new project structure

```
├── Gopkg.lock
├── Gopkg.toml
├── Jenkinsfile
├── shuttle.yaml
└── ... project files etc. ...
```

lunar way

#1

shuttle

lunar way

#1

It’s open source!

lunarway/shuttle: A CLI for han

GitHub, Inc. [US] | https://github.com/lunarway/shuttle

Search or jump to... / Pull requests Issues Marketplace Explore

lunarway / shuttle

Watch 3 Unstar 13 Fork 0

Code Issues 9 Pull requests 0 Actions Projects 1 Wiki Insights Settings

A CLI for handling shared build and deploy tools between many projects no matter what technologies the project is using. Edit

Manage topics

84 commits 2 branches 17 releases 4 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

README.md

shuttle

A CLI for handling shared build and deploy tools between many projects no matter what technologies the project is using.

Report bug · Request feature · Releases · Latest release

Table of contents

- What is shuttle?
- Status
- How?
- Features
- Documentation
- Installing

lunar way

#1

shuttle.yaml

Purpose

- point to plan
- specify project specific vars
- specify project specific scripts

```yaml
plan: git://git@bitbucket.org:LunarWay/lw-shuttle-go-plan.git

vars:
  service: subscription
  squad: nasa
  docker:
    builder-image: golang:1.10.2
    cmds:
    - duplicate_subscriptions
  db: true
  rabbitmq: true
  s3:
    public: true
    private: true
  ingress: true
  tests:
    db: true
  k8s:
    local:
      replicas: 1
      db:
        base64Password: c3Vic2NyaXB0aW9u
      env:
        authentication.middleware.mocked.user.id: 'mockedUserId'
        # ...
    dev:
      replicas: 1
      env:
        execute.plan.switch.job: 'true'
        # ...
```

lunar way

#1

shuttle.yaml

Purpose

- point to plan
- specify project specific vars
- specify project specific scripts

```yaml
  - duplicate_subscriptions
db: true
rabbitmq: true
s3:
  public: true
  private: true
ingress: true
tests:
  db: true
k8s:
  local:
    replicas: 1
    db:
      base64Password: c3Vic2NyaXB0aW9u
    env:
      authentication.middleware.mocked.user.id: 'mockedUserId'
      # ...
  dev:
    replicas: 1
    env:
      execute.plan.switch.job: 'true'
      # ...
  staging:
    replicas: 1
    env:
      execute.plan.switch.job: 'false'
      # ...
  prod:
    replicas: 2
    env:
      execute.plan.switch.job: 'true'
      # ...
```

lunar way

#1

plan.yaml

located at
`.shuttle/plan/plan.yaml`

Purpose

- shared scripts spec
- help and descriptions

```yaml
scripts:
  build:
    description: |
      builds a docker image.
      'shuttle run build tag=latest'.
      tag is optional
    args:
    - name: tag
    actions:
    - shell: $plan/scripts/build.sh
  push:
    description: |
      pushes the docker image to quay.
      'shuttle run push tag=latest env=dev'.
      tag and env is required.
    args:
    - name: tag
      required: true
    - name: env
      required: true
    actions:
    - shell: $plan/scripts/push.sh
  test:
    args:
    - name: report
      description: output the status in reports dir
    - name: tags
      description: comma-seperated list of tags
```

lunar way

#1

Building...

Shell

Shell

11:37:58 lunar-way-subscription-service docker-for-desktop master 34

```
shuttle run build
Pulling latest plan changes
shell: $plan/scripts/build.sh
Using SSH key /Users/Emil/.ssh/id_rsa
writing RSA key
Sending build context to Docker daemon   1.34MB
Step 1/27 : FR[illegible]g:1.10.2 as builder
 ---> 3f30f1fc[illegible]
Step 2/27 : WO[illegible]o/src/bitbucket.org/LunarWay/lunar-way-subscription-service
[illegible]
[illegible]ABLED
[illegible] Using ca[illegible]
[illegible] b200ab47[illegible]
[illegible]4/27 : EN[illegible]inux
[illegible]Using c[illegible]
S[illegible]H_KEY
 ---> Using cache
 ---> 1dd5bc987185
Step 6/27 : RUN apt-get update && apt-get install -y tzdata zip ca-certificates
 ---> Using cache
 ---> f6d04e2258ac
Step 7/27 : RUN curl -fsSL -o /usr/local/bin/dep https://github.com/golang/dep/releases/download/v0.5.0/dep-linux-am
d64 && chmod +x /usr/local/bin/dep
 ---> Using cache
```

demo

lunar way

#1

Jenkins

lunar way

#1

Jenkins

lunar way

#1

Jenkins

lunar way

#1

Shuttle❤️Jenkins

lunar way

#1

Shuttle❤️Jenkins

```
// jenkins-server/scripts/shuttlePipeline.groovy

def call(body) {
  properties([[$class: 'BuildDiscarderProperty', strategy: [$class: 'LogRotator', numToKeepStr: '25']],
pipelineTriggers([[$class: 'BitBucketTrigger']])])

  node {
    // Checkout the project
    checkout scm

    // Init shuttle
    sh "curl -LO https://github.com/lunarway/shuttle/releases/download/\$(curl -s
https://api.github.com/repos/lunarway/shuttle/releases/latest | grep tag_name | cut -d '\"' -f 4)/shuttle-linux-
amd64"
    sh "chmod +x shuttle-linux-amd64"
    sh "mv shuttle-linux-amd64 shuttle"
    sh "./shuttle prepare -vc"

    // Load Pipeline defined in the plan
    def pipeline = load pwd() + '/.shuttle/plan/pipeline.groovy'
    pipeline.init()
  }
}
```

lunar way

#1

Shuttle❤️Jenkins

```
// lunar-way-subscription-service/Jenkinsfile

@Library("jenkins-pipeline-scripts") _

// Use the shuttle pipeline defined in lunar-way-jenkins-pipeline-scripts
shuttlePipeline {}
```

lunar way

#1

Shuttle❤️Jenkins

```
// lunar-way-subscription-service/.shuttle/plan/pipeline.groovy

def init() {
    try {
      stage('Generate Build Info'){
        project.buildInfo()
      }

      withCredentials([string(credentialsId: 'jenkins_ssh_key', variable: 'SSH_KEY')]) {
        stage('Build Image') {
          sh "./shuttle run build tag=${project.imageTag()}"
        }

        stage('Tests') {
          try {
            sh './shuttle -v run integration_test report=true'
          }
          finally {
            step([$class: 'JUnitResultArchiver', testResults: 'reports/*.xml', allowEmptyResults: true])
          }
        }
      }

      if (project.runCD()) {

        stage('Push Image') {
          sh "./shuttle run push tag=${project.imageTag()} env=$env.BRANCH_NAME"
        }
```

lunar way

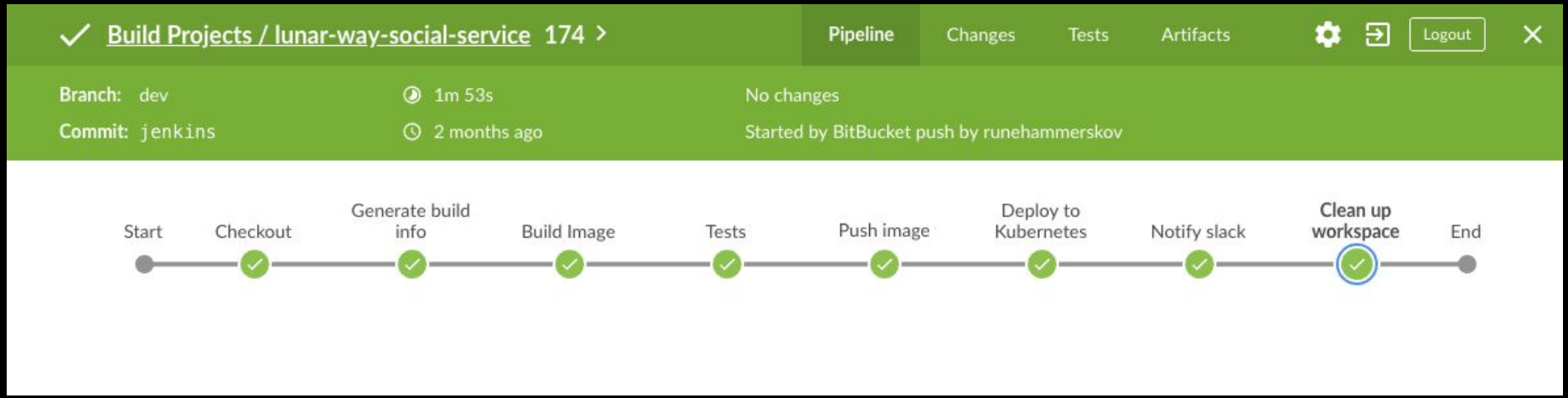
#1

```
          finally {
            step([$class: 'JUnitResultArchiver', testResults: 'reports/*.xml', allowEmptyResults: true])
          }
        }
      }

      if (project.runCD()) {

        stage('Push Image') {
          sh "./shuttle run push tag=${project.imageTag()} env=$env.BRANCH_NAME"
        }

        stage('Deploy to Kubernetes') {
          sh "./shuttle run deploy env=$env.BRANCH_NAME tag=${project.imageTag()}"
        }
      }

      stage('Notify slack') {
        slackslim.sendSuccessful("${shuttle.service()}", "${project.repo()}", "${shuttle.squad()}",
"${git.author()}", "${git.hash()}", "${git.message()}", "${project.imageTag()}")
      }
      stage('Clean up workspace') {
        step([$class: 'WsCleanup'])
      }
    }
    catch(exc){
      slackslim.sendFailure("${shuttle.service()}", "${project.repo()}", "${shuttle.squad()}", "${git.author()}",
"${git.hash()}", "${git.message()}", "${project.imageTag()}", "$exc")
      step([$class: 'WsCleanup'])
      throw exc
    }
}

return this
```

lunar way

#1

Shuttle❤️Jenkins



#2 Improving our developer tools

lunar way

#2

Improving our shell tools

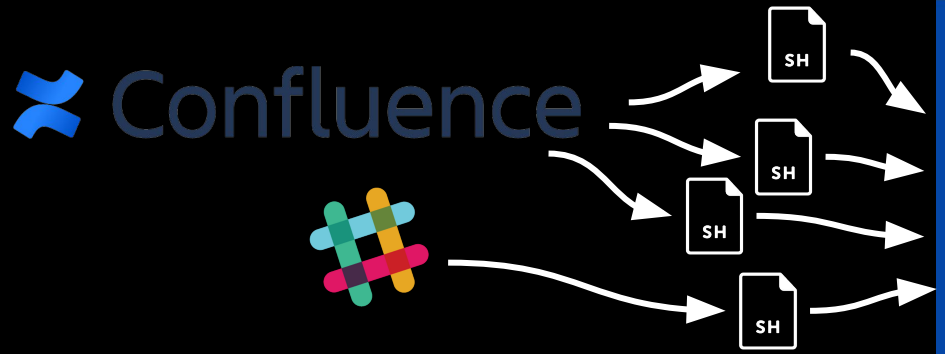
- all developers are on Mac's
- (oh-my-)zsh was the most used shell
- tapping into that

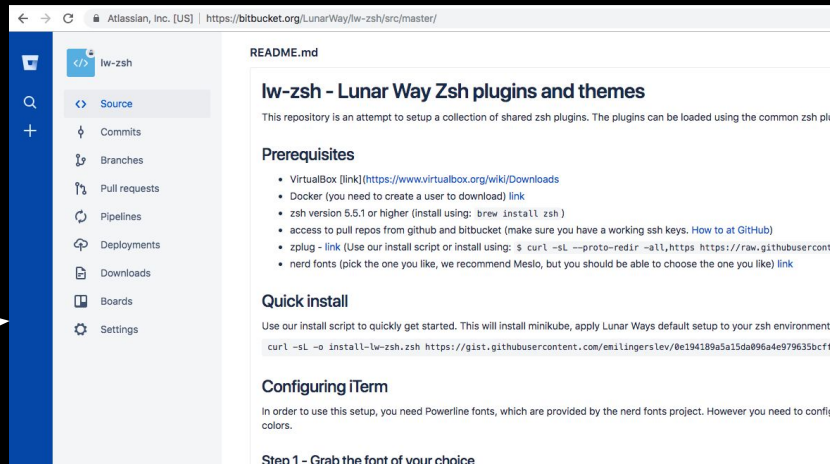
lunar way

#2

Aligning our shell tools

- Scripts shared on slack & confluence
- Made a zsh plugin
- Refined & added scripts

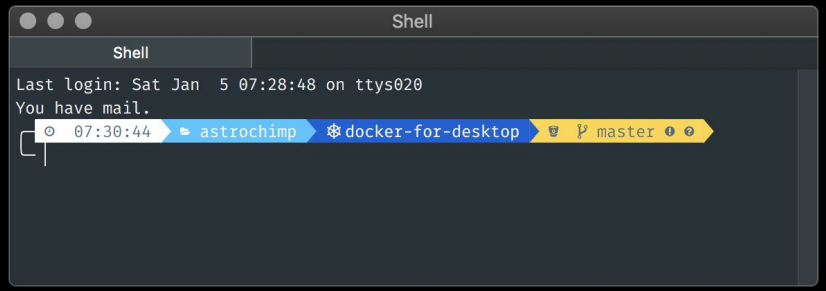




#2

Using our shell tools

- Choose a zsh plugin manager - zplug
- Made a shared list of good plugins
- Gave it a theme
- Made install guide



lunar way

#2

Demo our shell tools

lunar way

#3 Improving our security

#3

Finding a secure(-ity) entrypoint

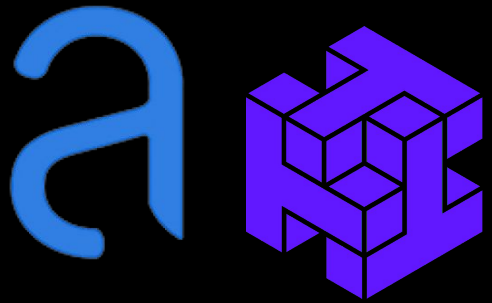
- Not security experts
- Tasked with improving security
- Iterative improvement
- Least but most valuable first move
- CI compatible

lunar way

#3

Investigating options

[:] SourceClear



Twistlock

lunar way

#3

Implementing POC's

- shuttle powered testing
- implemented once for all (per plan though)
- testing on projects live

lunar way

#3

```
// lunar-way-subscription-service/.shuttle/plan/pipeline.groovy
// ...
stage('Tests') {
  try {
    sh './shuttle -v run integration_test report=true'
  }
  finally {
    step([$class: 'JUnitResultArchiver', testResults: 'reports/*.xml', allowEmptyResults: true])
  }
}
// ...
```

lunar way

#3

```
// lunar-way-subscription-service/.shuttle/plan/pipeline.groovy
// ...
stage('Tests') {
  try {
    parallel 'project tests': { /* project test spec */ },
    'snyk security scan': { try {
        sh "./shuttle -v run security_scan:code tag=${project.imageTag()} upload=true"
      } catch(err) {
        echo "Failed snyk security scan - continuing during evaluation phase. Error: $err"
      } finally {
        publishHTML([allowMissing: true, alwaysLinkToLastBuild: false, keepAll: false, reportDir: 'reports',
reportFiles: 'snyk-code.html', reportName: 'snyk: code', reportTitles: '']) 
      }
    },
    'sourceclear scan': {
      try {
        sh "./shuttle -v run sourceclear_scan tag=${project.imageTag()}"
      } catch(err) {
        echo "Failed sourceclear scan - continuing during evaluation phase. Error: $err"
      }
    },
    'aqua scan': {
      try {
        sh "./shuttle -v run aqua_scan tag=${project.imageTag()}"
      } catch(err) {
        echo "Failed aqua scan - continuing during evaluation phase. Error: $err"
      }
    }
  } finally { /* ... */ }
}
```

[:] SourceClear

#3

```
// lunar-way-subscription-service/.shuttle/plan/pipeline.groovy
// ...
stage('Tests') {
  try {
    parallel 'project tests': {
      try {
        sh './shuttle -v run integration_test report=true'
      } finally {
        step([$class: 'JUnitResultArchiver', testResults: 'reports/*.xml', allowEmptyResults: true])
      }
    },
    'snyk security scan - code': {
      try {
        if(env.BRANCH_NAME == "dev"){
          sh "./shuttle -v run security_scan:code tag=${project.imageTag()} upload=true"
        } else {
          sh "./shuttle -v run security_scan:code tag=${project.imageTag()}"
        }
      } catch(err) {
        echo "Failed snyk security scan - continuing during evaluation phase. Error: $err"
      } finally {
        publishHTML([allowMissing: true, alwaysLinkToLastBuild: false, keepAll: false, reportDir: 'reports',
reportFiles: 'snyk-code.html', reportName: 'snyk: code', reportTitles: ''])
      }
    },
    'snyk security scan - docker': {
      try {
        if(env.BRANCH_NAME == "dev"){
          sh "./shuttle -v run security_scan:docker tag=${project.imageTag()} upload=true"
        } else {
          sh "./shuttle -v run security_scan:docker tag=${project.imageTag()}"
        }
      } catch(err) {
        echo "Failed snyk security scan - continuing during evaluation phase. Error: $err"
      } finally {
        publishHTML([allowMissing: true, alwaysLinkToLastBuild: false, keepAll: false, reportDir: 'reports',
reportFiles: 'snyk-docker.html', reportName: 'snyk: docker', reportTitles: ''])
      }
    }
  } finally { /* ... */ }
}
```

lunar way

#3

jenkins / bitbucket/lunar-way-s

https://jenkins.dev.lunarway.com/blue/organizations/jenkins...

Lunar Way / lunar-way-subscription-service < 38

Pipeline | Changes | Tests | Artifacts | Logout

Branch: dev
Commit: –

3m 41s
a day ago

No changes
Branch event

Start → Generate Build Info → Build Image → Tests → Push Image → Deploy to Kubernetes → Notify slack → Clean up workspace → End

Tests:
- project tests
- snyk security scan - code
- snyk security scan - docker

Clean up workspace - <1s

> Delete workspace when build is done <1s

lunar way

#4 Less Jenkins, more auto

lunar way

#4

lunar way

#4

Convention over configuration step 1

lunar way

#4

We had a long guide in confluence

How to setup a new service? -

https://lunarway.atlassian.net/wiki/pages/viewpage....

Dashboard / ... / Guides

Gem feller senere · Følger · Del

How to setup a new service?

Oprettet af Kasper Nissen, senest ændret af Bjørn Sørensen på jun. 07, 2018

Du ser en gammel version af denne side. Se den nuværende version.

Sammenlign med nuværende · Gendan denne version · Vis sidehistorik

« Forrige **Version 14** Næste »

This page will provide a guide for setting up a new service, pipeline job and deployments. This involves a couple of steps that will be described throughout this tutorial.

Create a Bitbucket repository

Go to https://bitbucket.org/repo/create and create a new repository. Set the owner to LunarWay. Set Project to Microservices and specify a name. By convention Lunar Way services are named `lunar-way-servicename-service`, e.g. `lunar-way-authentication-service`.

Scaffold a service

Use the Lunar Way service generator https://bitbucket.org/LunarWay/generator-lw to create a scaffold of a typical Lunar Way service.

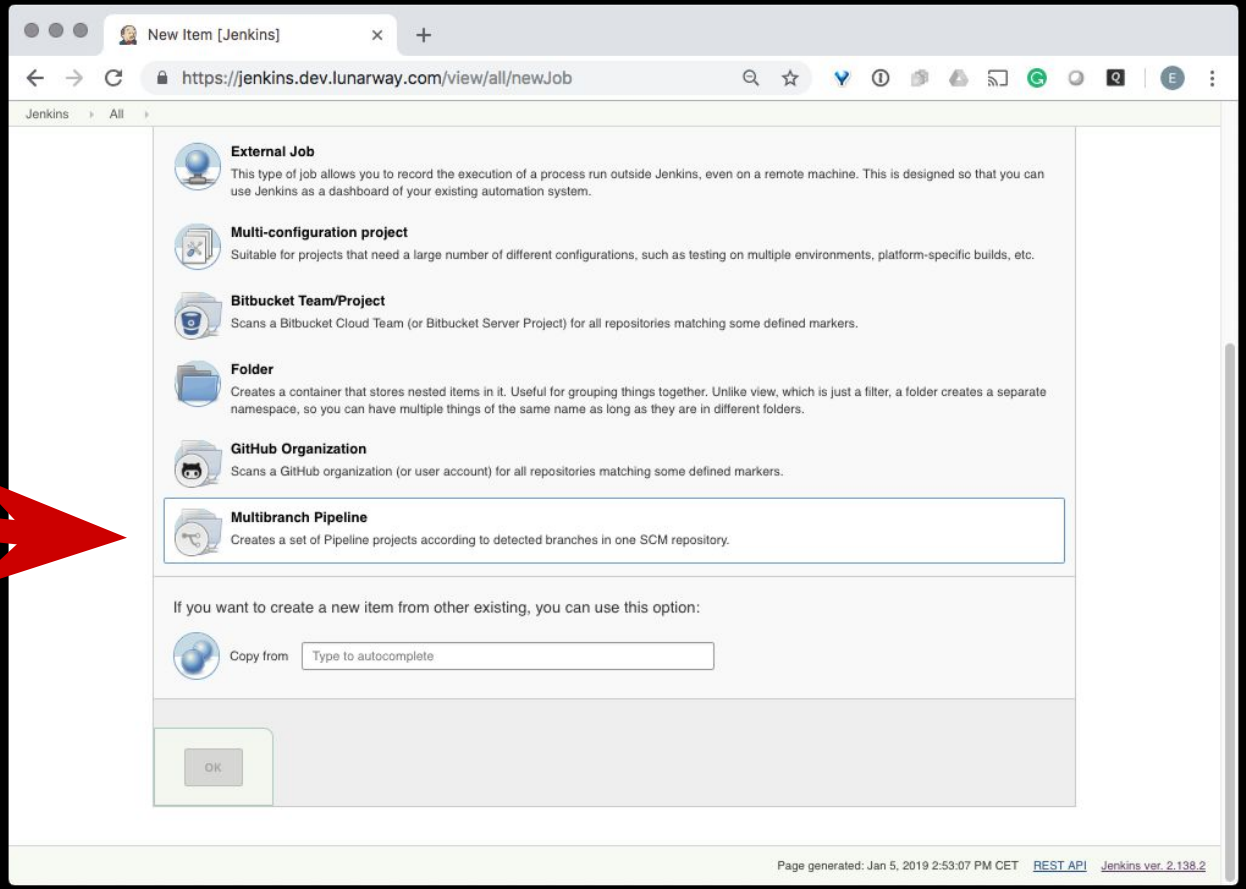
Once generated, initialise the directory as a git repo and hook it up to the Bitbucket repository.

```
cd lunar-way-<name>-service
git init
git remote add origin git@bitbucket.org:LunarWay/lunar-way-<name>-service.git
git add .
git commit -m "Generated service"
git push -u origin master
```

lunar way

#4

We had a lot of these multibranch pipelines



lunar way

#4

Configured

for each repository

for each branch

Branches (2) [Old Build Projec

https://jenkins.dev.lunarway.com/job/Old%20Build%20Projects/job...

Jenkins 3 search Emil Krog Ingerslev | log out

Jenkins › Old Build Projects › lunar-way-backend › ENABLE AUTO REFRESH

- Up
- Status
- Configure
- Scan Multibranch Pipeline Log
- Multibranch Pipeline Events
- Delete Multibranch Pipeline
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Move
- Open Blue Ocean
- Rename
- Pipeline Syntax
- Credentials

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle
3 Idle
4 Idle

lunar-way-backend

Branches (2)

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
		demo	8 mo 7 days - #6	1 mo 21 days - #17	1 min 51 sec	
		dev	1 mo 10 days - #1272	1 mo 20 days - #1250	2 min 4 sec	

Icon: S M L

Legend RSS for all RSS for failures RSS for just latest builds

https://jenkins.dev.lunarway.com/job/Old Build Projects/job/lunar-way-backend/#

lunar way

#4

Configured

for each repository

for each branch

lunar-way-backend Config [Je

https://jenkins.dev.lunarway.com/job/Old%20Build%20Projects/job...

Jenkins 3 search Emil Krog Ingerslev | log out

Jenkins › Old Build Projects › lunar-way-backend

General | Branch Sources | Build Configuration | Scan Multibranch Pipeline Triggers | Orphaned Item Strategy | Health metrics | Properties | JIRA | Pipeline Libraries | Pipeline Model Definition

Display Name

Description

[Plain text] Preview

Branch Sources

Single repository & branch

Allows a single, fixed branch of some repository to be configured. Not normally used, as it does not allow branches to be detected automatically. Might be used, for example, if you want to build all branches of one repository, *and* the `master` branch of a special clone.

Name: dev

Source Code Management: Git

Repositories

Repository URL: git@bitbucket.org:LunarWay/lunar-way-backe

Credentials: jenkins

Add

Advanced...

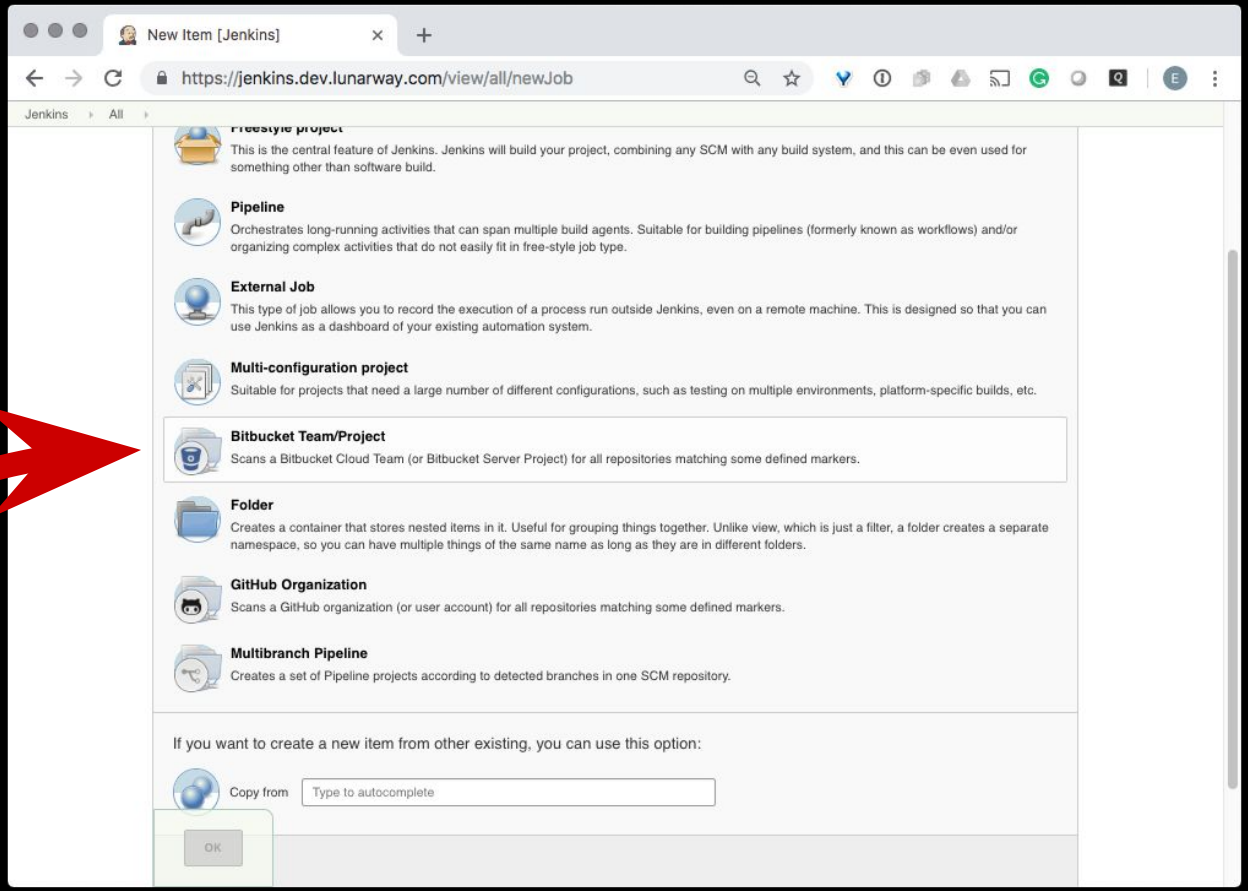
Add Repository

Save | Apply

lunar way

#4

Switching that out to this



lunar way

#4

Configuring it once for <u>all</u> projects and branches

Lunar Way Config [Jenkins]

https://jenkins.dev.lunarway.com/job/bitbucket/configure

Jenkins › Lunar Way ›

General | **Projects** | Scan Organization Folder Triggers | Orphaned Item Strategy | Health metrics | Properties | JIRA | Pipeline Libraries | Pipeline Model Definition

Bitbucket Team/Project

Credentials: lunarway-ci/****** (bitbucket-lunarway-ci) — Add

Owner: LunarWay

Behaviors

Repositories

Filter by project name (with regular expression)

Regular expression: ^((?!Archived).)*$

Within repository

Discover branches

Strategy: All branches

Discover pull requests from origin

Strategy: Merging the pull request with the current target branch revision

Discover pull requests from forks

Strategy: Merging the pull request with the current target branch revision

Trust: Forks in the same account

Filter by name (with regular expression)

Regular expression: ^((?!prod|staging).)*$

Save | Apply

lunar way

#4

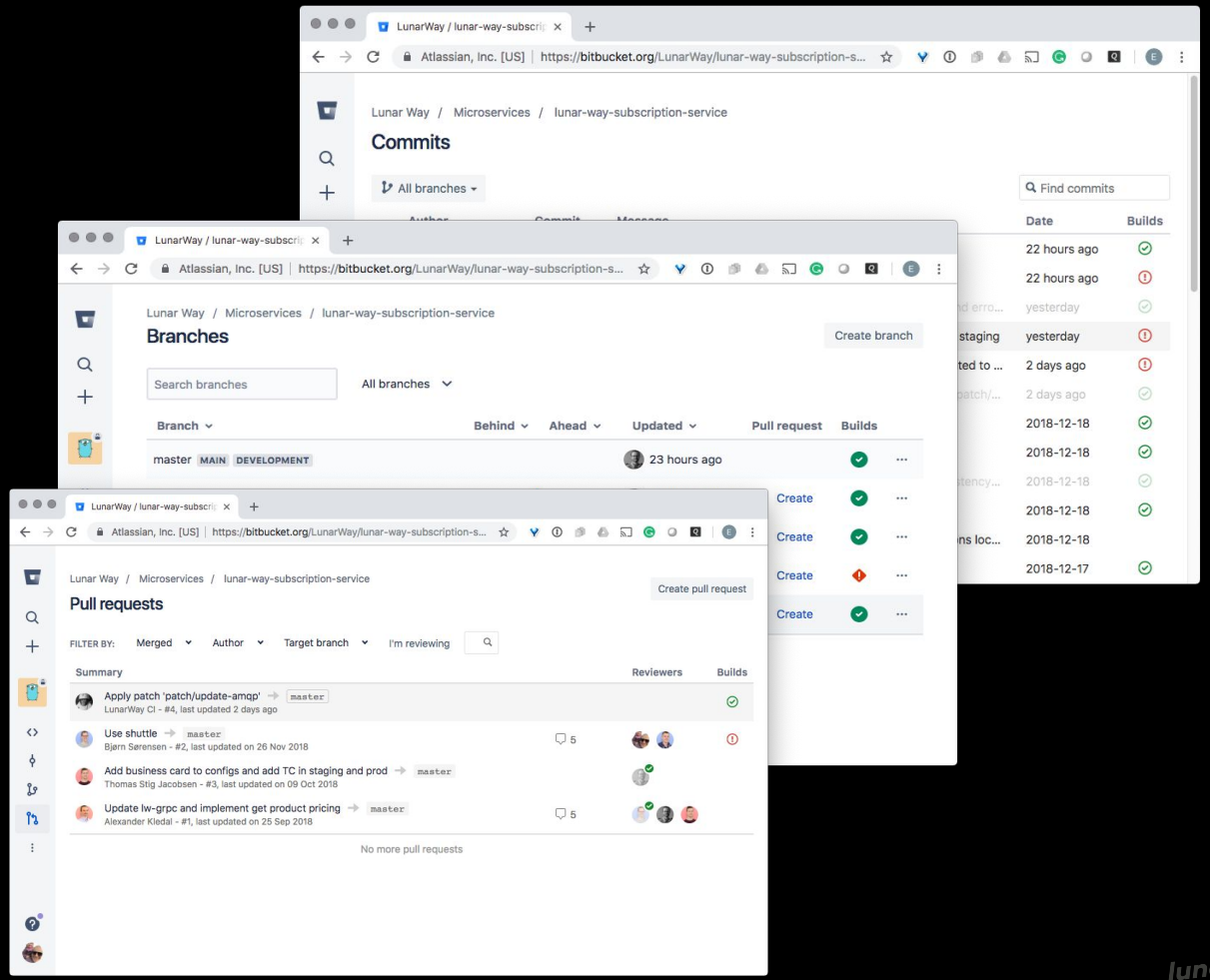
We even got

pull request support

and

status updates in bitbucket

out of the box



lunar way

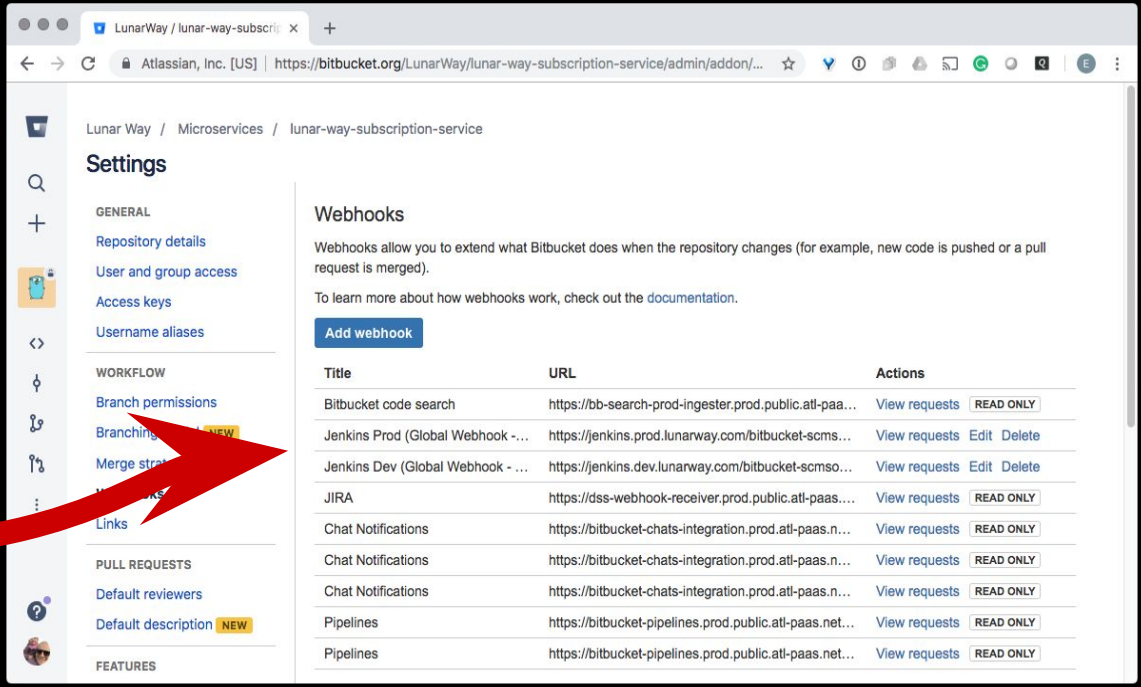
#4

Convention over configuration step 2

lunar way

#4

In bitbucket <u>every one</u> of our projects requires a pair of these webhooks



lunar way

#4

Requires several steps during service creation

Updating the webhooks requires changes to all projects

lunar way

#4

LunarWay / lunar-way-subscrip | site / master / issues / #13940

Atlassian, Inc. [US] | https://bitbucket.org/site/master/issues/13940/add-team-repository-webhooks

Bitbucket / Public Issue Tracker / master / Issues

Create issue

Add Team repository webhooks

Issue #13940 OPEN

Attach

Mikael Svensson created an issue 2017-03-03

We use the Bitbucket source plugin in Jenkins which works great and it is very popular.

The problem we have is that when scanning the Bitbucket team repos for branches containing Jenkinsfiles we hit the rate limit. We have over 300 repos in our team and we want new repos to be added to Jenkins as fast as possible. The plugin works by pulling the teams all pull requests and branches to search for Jenkinsfiles.

This could be handles more efficiently if Bitbucket teams have webhooks for repo/branch creation and commits to repos in the team.

Assignee	–
Type	enhancement
Priority	major
Status	open
Component	Hooks
Votes	16 Vote for this issue
Watchers	19 Watch this issue

lunar way

#4

LunarWay / lunar-way-subscrip | site / master / issues / #13940

Atlassian, Inc. [US] | https://bitbucket.org/site/master/issues/13940/add-team-repository-webhooks

Daniel Tao STAFF

We already have this, believe it or not. It is not documented as well as it should be, but you can register webhooks for all of a team's repositories pretty much the same way you can register them for a single repository. To see the events you can create webhooks for, see:

https://api.bitbucket.org/2.0/hook_events/team

The API endpoint itself is listed here:

https://developer.atlassian.com/bitbucket/api/2/reference/resource/teams/%7Busername%7D/hooks#post

I acknowledge the documentation could be made a bit more readable. Fortunately, as stated, it works the same way the API for registering repository webhooks works; just instead of:

https://api.bitbucket.org/2.0/repositories/{username}/{repo_slug}/hooks

You would make a call to:

https://api.bitbucket.org/2.0/teams/{username}/hooks

Let me know if this meets your needs! If you have any questions about usage, I can try to answer them.

Join the team!

#4

So basic'ly is just one (or two)
POST to the team
instead of each project

Result in project:

- project autodiscovery
- no manual creation steps
- easier editing for all projects

Catch:

- don't edit it in the UI

```
curl -X POST \
  -u "<username>:<token>" \
  https://api.bitbucket.org/2.0/teams/LunarWay/hooks \
  --data-binary '{
  "description": "Jenkins Dev (Global Webhook - Do not edit or delete)",
  "url": "https://jenkins.dev.lunarway.com/bitbucket-scmsource-hook/notify",
  "type": "webhook_subscription",
  "active": true,
  "events": [
    "repo:push",
    "repo:fork",
    "repo:updated",
    "repo:commit_comment_created",
    "repo:commit_status_created",
    "repo:commit_status_updated",
    "pullrequest:created",
    "pullrequest:updated",
    "pullrequest:approved",
    "pullrequest:unapproved",
    "pullrequest:fulfilled",
    "pullrequest:rejected",
    "pullrequest:comment_created",
    "pullrequest:comment_updated",
    "pullrequest:comment_deleted",
    "issue:created",
    "issue:updated",
    "issue:comment_created",
    "project:updated",
    "repo:created",
    "repo:deleted",
    "repo:imported",
    "repo:transfer"
  ]
}'
```

lunar way

#5 Greenkeeping

lunar way

#5

Proposal

- ***Imagine*** patching all microservices with a simple fix or update
- Make a PR with it and run all the tests
- Auto-merge the PR on test success
  - Optional manual PR approval first

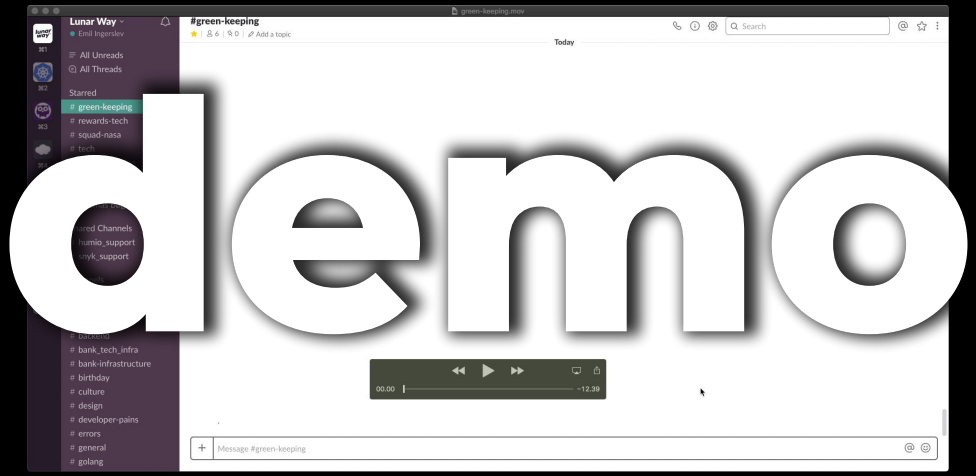
lunar way

#5

#5

Project Astrochimp

An **experiment** in implementing greenkeeping at Lunar Way

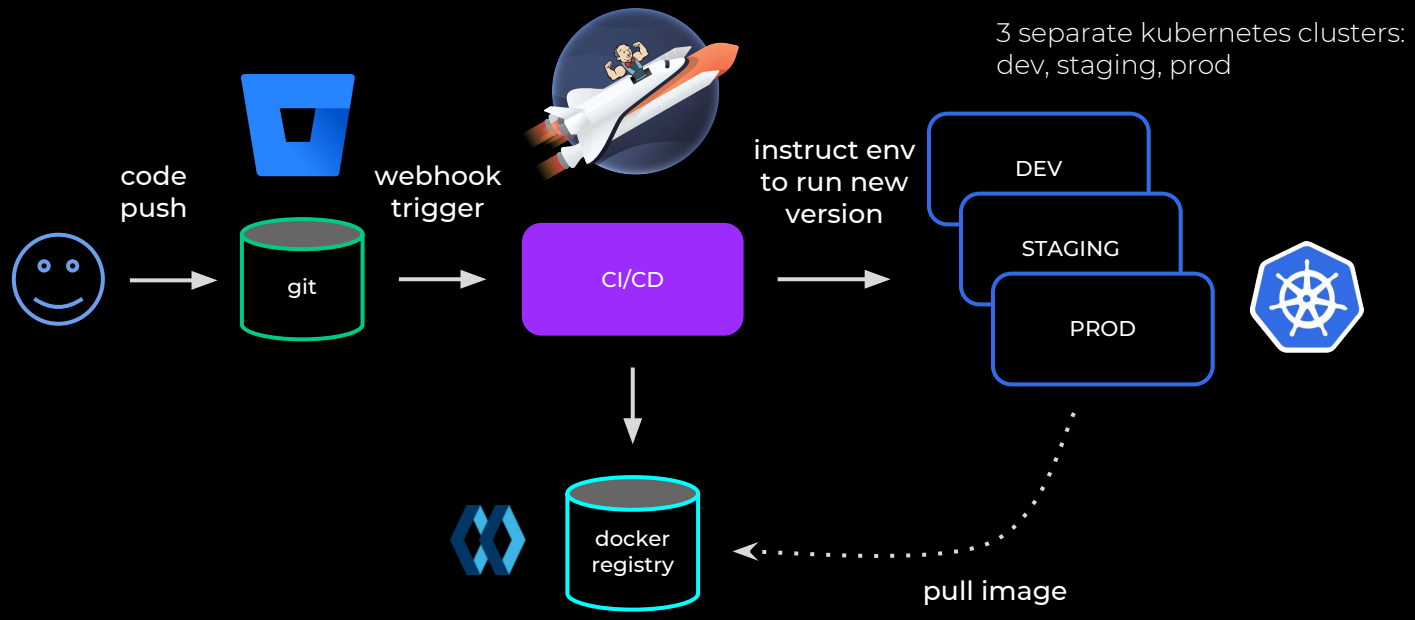


#6 Separating CD from CI

lunar way

#6

This is where we are



lunar way

#6

- CI systems should not instruct environments.
- Keeping track of changes to the environments becomes hard.
- Building artifacts for each environment is not feasible.

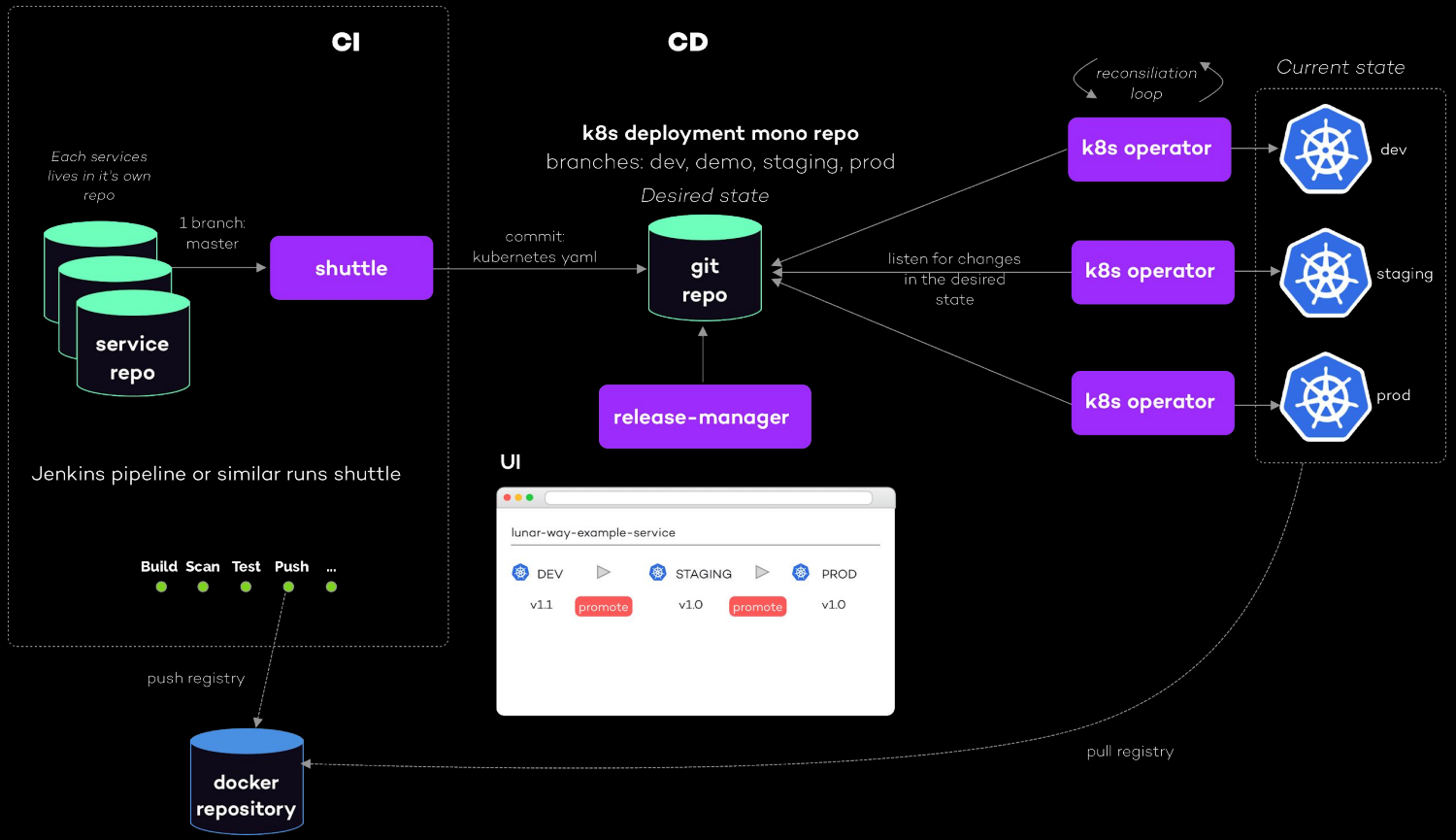
lunar way

#6

What to do instead?

lunar way

#6



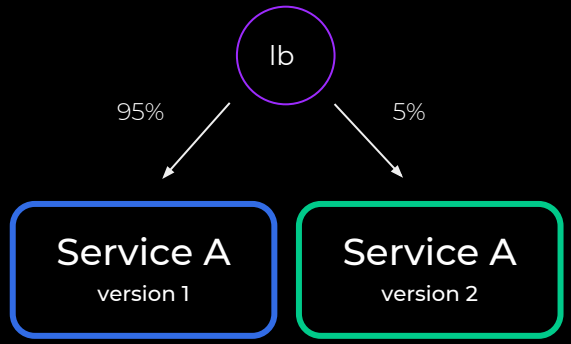
The grand vision

lunar way

#6

Why separate CD?

- Build and test releases only once
- Promote (validated) releases - not code
- More than normal release validation can be
  - manual tests
  - automated system tests across services
- A release is versioned to be easier to refer to and depend on
- More options for doing other types of deployments



lunar way

#6

Weavework Flux

Jenkins X

Rolling our own operator

lunar way

Questions?

lunar way

lunar way®