

What Developers want (and what platform people can do about it)

40 min



Your hosts today



Developer Advocate at Codefresh

Kostis Kapelonis

 kostis@codefresh.io



Lead Architect at Port

Mor Paz

 mor@getport.io

Agenda

- Intro - information overload
- Demo 1
- Give developers what they need
- Demo 2
- Deep dive into the mechanics
- Demo 3

People first

People



Developer

Focus is on writing code and shipping features



Operations/Ops

Focus is on running the systems and giving infrastructure to developers



DevEx

Focus is on day-2-day productivity, onboarding and removing obstacles

Developers

- Want to ship features
- Move fast
- Don't really want to learn K8s
- Want to follow commit to prod
- Want easy way to create env
- Like clicking buttons



SRE/Operations/DevOps/platform/infra

- Want healthy systems
- Move slow
- Hate open tickets
- Prefer automation
- Want to help developers
- Ideally everything is self-service



DevEX/Team lead/CTO etc.

- Care about development onboarding
- Monitor sprint velocity
- Responsible for day-to-day operations and blockers
- Hate “walls”, silos and lengthy drop-off procedures



Information overload

Simple booking example (5 options)

 Book a flight

Round trip

 Departing from *

 Arriving at *

Travel dates

 12 July – 29 July

Passengers



1 adult

Complex booking example (20 options)

 Book a flight

Round trip

 Departing from *

 Arriving at *

Travel dates

 12 July – 29 July 

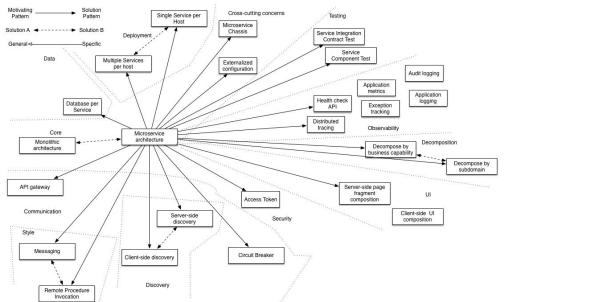
Passengers

1 adult

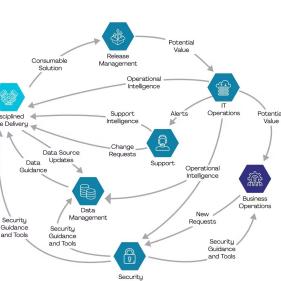
- Choose pilot
- Choose Airplane
- Terminal (depart)
- Terminal (arrive)
- Meal catering company
- Fuel company
- Fuel amount

- Choose Co-pilot
- Choose ground crew
- Choose air-hostess crew
- Air-control hand-off
- Cleaner crew
- Baggage handlers
- Insurance

Microservices



Environment types



Infra management



DevTools



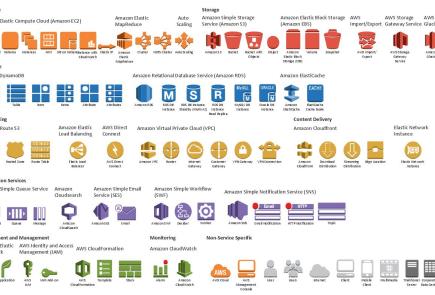
Databases & ETL



Permissions



Managed services



Clouds



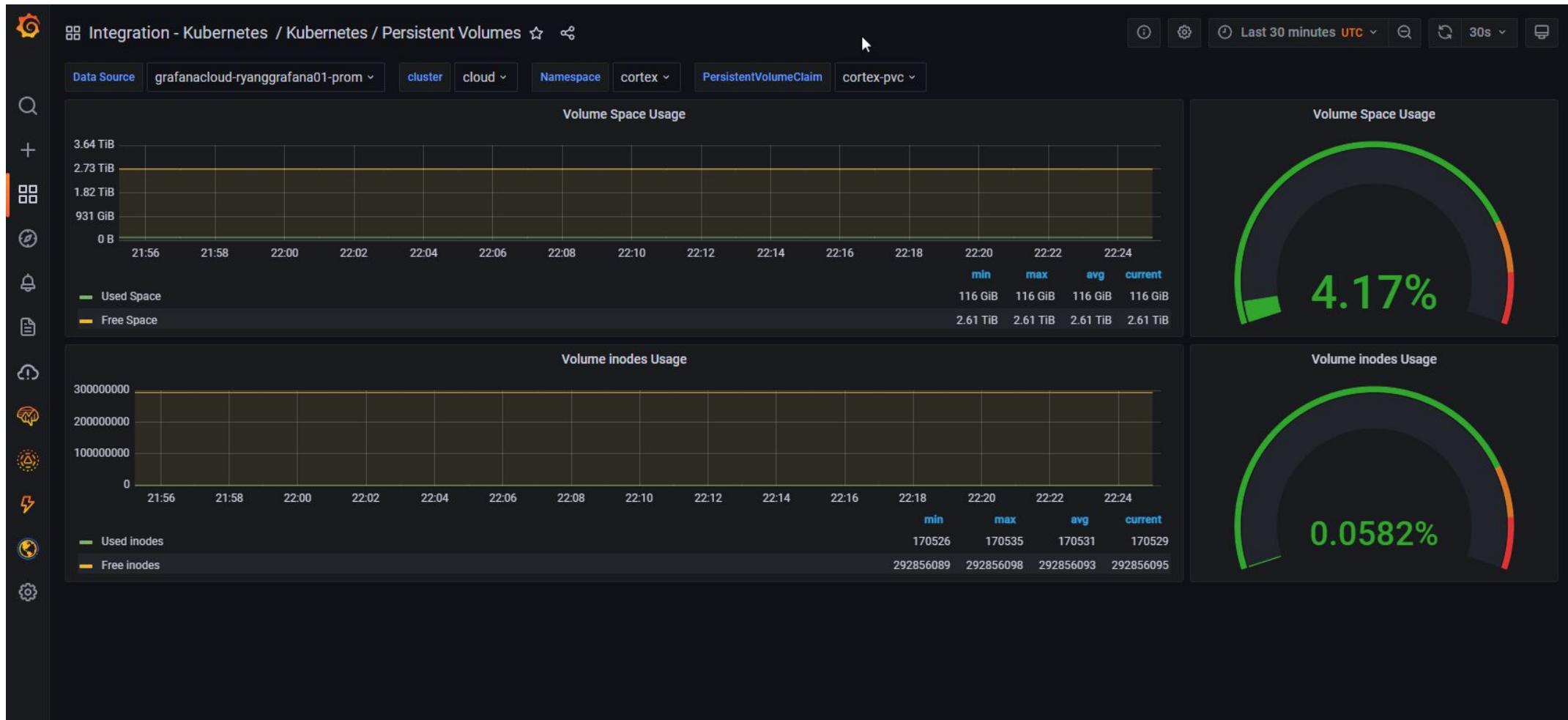
Code shipment



Configurations



Is this what developers want?



Is this what developers want?

Applications / guestbook

APPLICATION DETAILS

v2.1.5+ 

APP DETAILS **APP DIFF** **SYNC** **SYNC STATUS** **HISTORY AND ROLLBACK** **DELETE** **REFRESH** **Log out**

APP HEALTH  **Healthy**

CURRENT SYNC STATUS  **Synced** **MORE** **To HEAD (53e28ff)**

Author: May Zhang <may_zhang@intuit.com> -
Comment: feat: update helm samples to use helm3 (#78)

LAST SYNC RESULT  **Sync OK** **MORE** **To 53e28ff**

Succeeded 3 minutes ago (Fri Nov 05 2021 15:19:49
GMT-0500)
Author: May Zhang <may_zhang@intuit.com> -
Comment: feat: update helm samples to use helm3 (#78)

guestbook   **13 minutes**

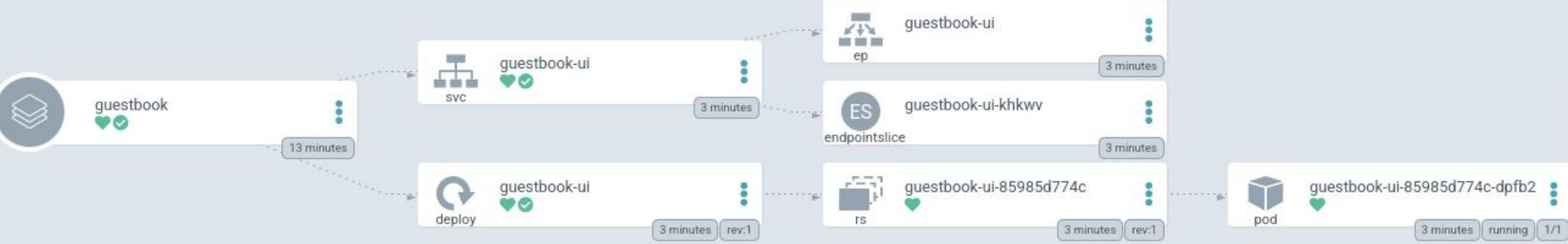
guestbook-ui   **3 minutes**

guestbook-ui-khkww   **3 minutes**

guestbook-ui-85985d774c-dpfb2   **3 minutes running 1/1**

deploy   **3 minutes rev:1**

guestbook-ui-85985d774c-dpfb2   **3 minutes rev:1**



Focusing on what matters with Port

An Open Internal Developer Portal

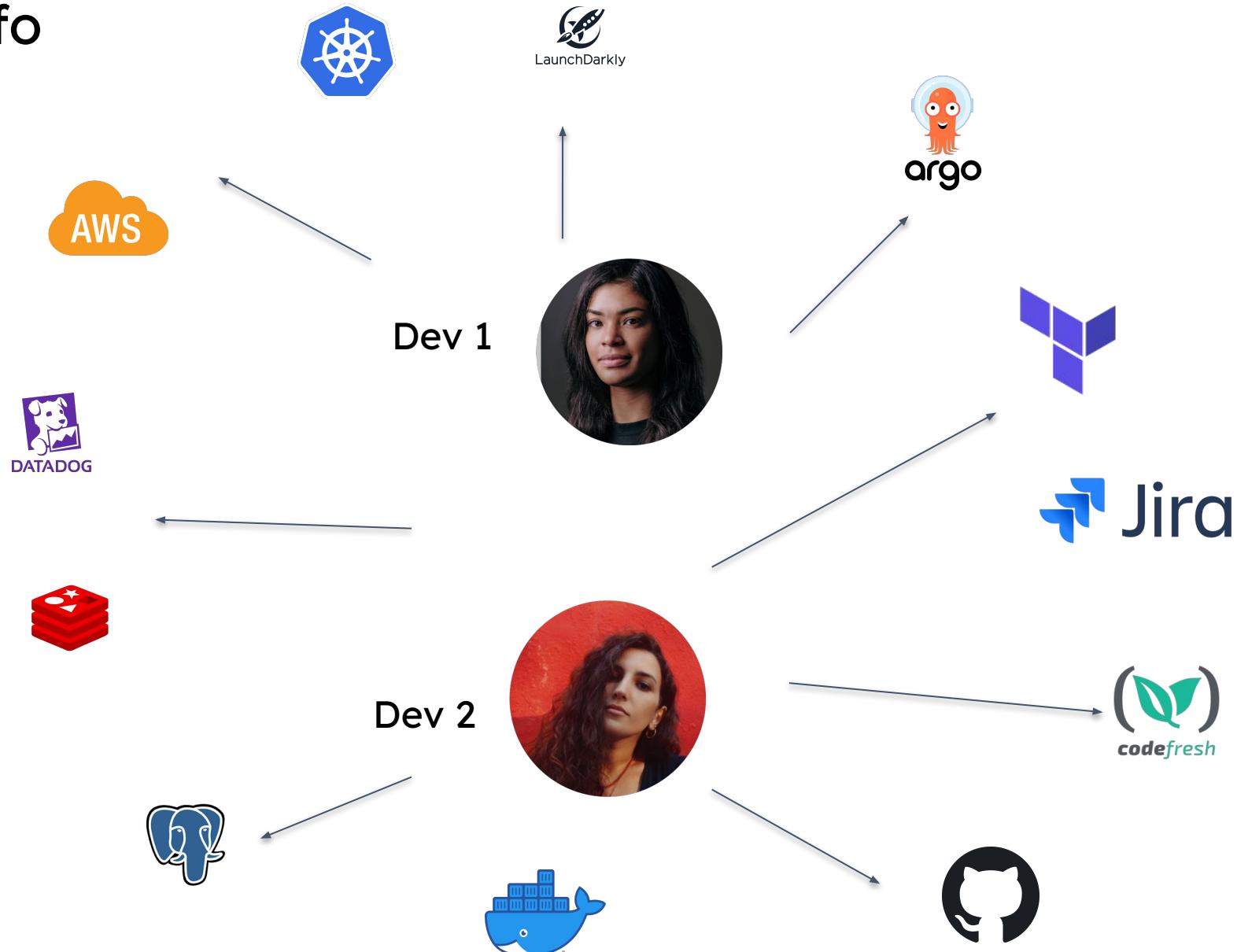
The screenshot shows the Port developer portal interface. The left sidebar contains a navigation menu with the following items:

- Services
- Running Services
- Deployments** (selected)
- Feature Flags
- K8s Clusters
- Cloud Permissions
- Cloud Resources
- Domains
- Developer Envs
- Packages
- Third Parties
- Alerts
- Vulnerabilities
- Misconfigurations

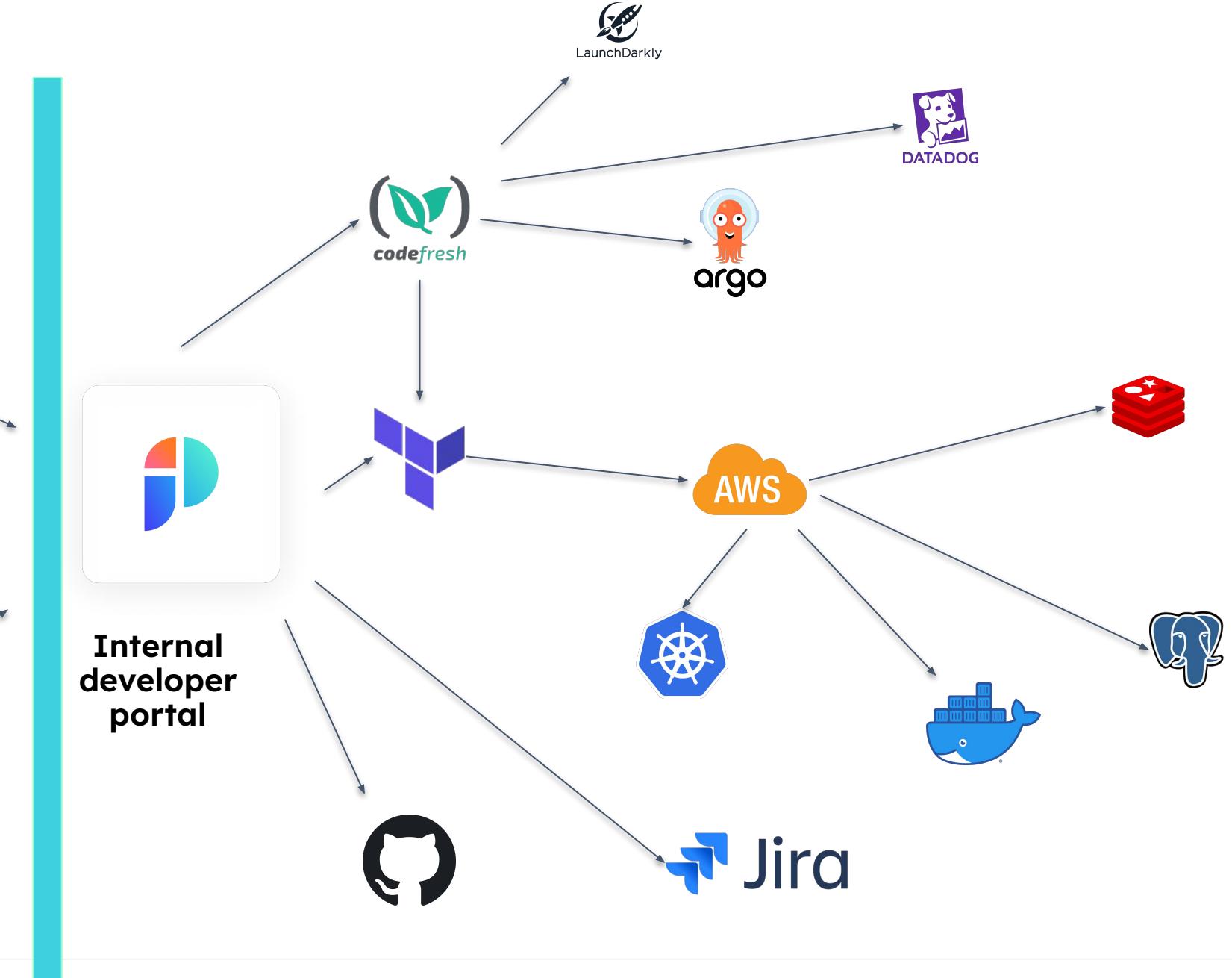
The main content area is titled "Deployments". It includes a search bar, filter, hide, sort, group by, and deployment buttons. A table lists ten deployment entries:

Domain	Title	Last Update	Deployment Status	User	Workflow URL	Branch URL	Commit SHA
Shipping	checkout-production	an hour ago	Failed	Michael M...	View	View	d47b9...
Subscript...	rating-production	an hour ago	In Progress	Ryoko De...	View	View	e6ebf...
Shipping	load-generator-security	an hour ago	Success	Michael M...	View	View	6f484...
Analytics	order-staging	an hour ago	Success	Francisca ...	View	View	af34e...
Inventory	shipping-production	an hour ago	Success	Michael M...	View	View	31b4a...
Subscript...	payment-staging	an hour ago	Success	Francisca ...	View	View	72dd0...
Shipping	ads-sandbox	an hour ago	Failed	Francisca ...	View	View	b3c80...
Shipping	checkout-staging	an hour ago	Failed	Michael M...	View	View	fddde...
Inventory	authorization-security	an hour ago	Success	Ryoko De...	View	View	c1f7d6...

Too much info



Developer portal



Use Cases

-  Cloud Resource Catalog
-  K8s & ArgoCD Visualization
-  Cloud Resource Permissions
-  Ephemeral Environments
-  Microservice Catalog
-  IaC for Developers
-  Package Management
-  On-call and Incident Management

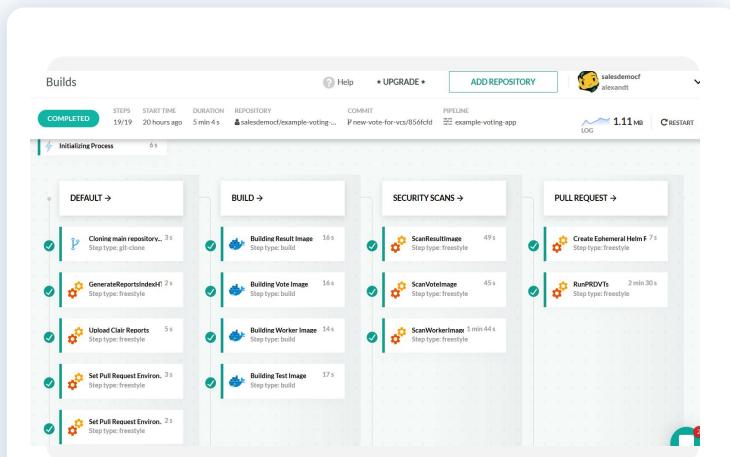
Demo 1

**Creating a model that
matches your team's needs**

Behind the scenes

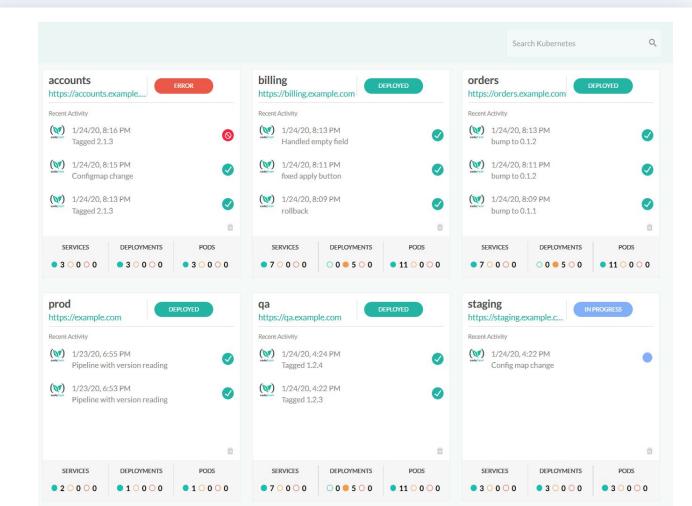
Expose to developers only what they need

What Codefresh offers



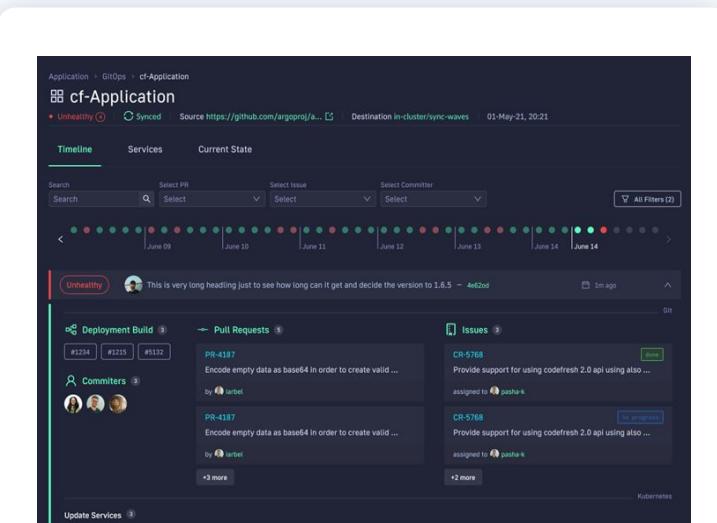
Pipelines

Implement Continuous Integration



Deployments

Implement Continuous Delivery

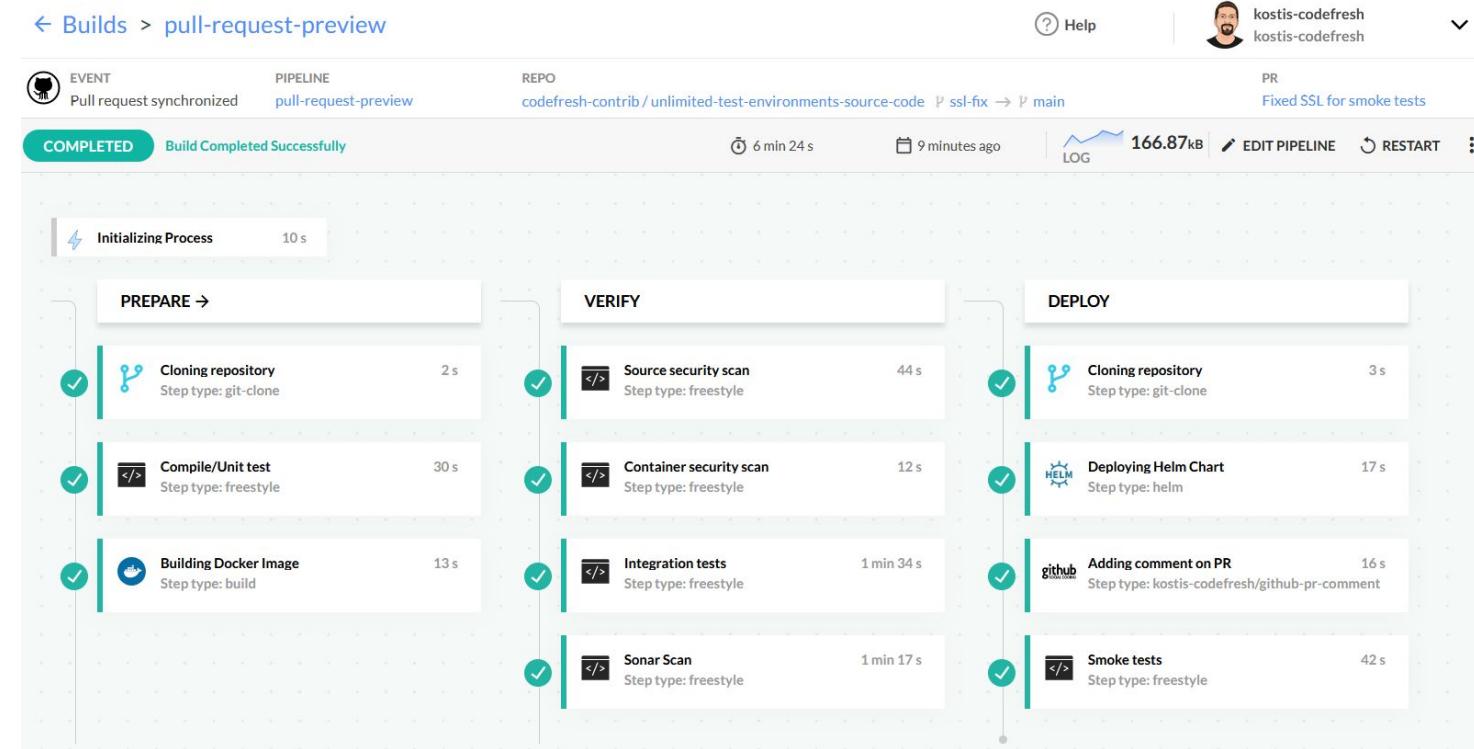


GitOps

Enterprise Argo CD

Codefresh pipelines

- Docker based pipelines
- Native Kubernetes integration
- Native Registry integration
- Argo CD/Rollouts/Workflows/events integration



Building/Pushing an image

GitHub Actions

```
- name: Log in to Docker Hub
  uses: docker/login-action@f4ef78c080cd8ba55a85445d5b36e214a81df20a
  with:
    username: ${{ secrets.DOCKER_USERNAME }}
    password: ${{ secrets.DOCKER_PASSWORD }}

- name: Extract metadata (tags, labels) for Docker
  id: meta
  uses: docker/metadata-action@9ec57ed1fcdbf14dcef7dfbe97b2010124a938b7
  with:
    images: my-docker-hub-namespace/my-docker-hub-repository

- name: Build and push Docker image
  uses: docker/build-push-action@3b5e8027fcad23fd98b2e3ac259d8d67585f671
  with:
    context: .
    file: ./Dockerfile
    push: true
    tags: ${{ steps.meta.outputs.tags }}
    labels: ${{ steps.meta.outputs.labels }}
```

Codefresh

```
BuildMyImage:
  title: Building My Docker image
  type: build
  image_name: my-app-image
  registry: dockerhub
  working_directory: '${{main_clone}}'
  tag: 1.0.1
```

Deploying to Kubernetes

GitHub Actions

```
- name: Configure AWS credentials
  uses: aws-actions/configure-aws-credentials@v1
  with:
    aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
    aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
    aws-region: us-east-2

- name: Login to Amazon ECR
  id: login-ecr
  uses: aws-actions/amazon-ecr-login@v1

- name: deploy to cluster
  uses: kodermax/kubectl-aws-eks@master
  env:
    KUBE_CONFIG_DATA: ${{ secrets.KUBE_CONFIG_DATA_STAGING }}
    ECR_REGISTRY: ${{ steps.login-ecr.outputs.registry }}
    ECR_REPOSITORY: my-app
    IMAGE_TAG: ${{ github.sha }}
  with:
    args: set image deployment/${ECR_REPOSITORY} $ECR_REGISTRY=$ECR_REPOSITORY
```

Codefresh

```
deploy_to_k8:
  title: deploying to cluster
  type: deploy
  kind: kubernetes
  cluster: myDemoAKSCluster
  namespace: demo
  service: my-python-app
```

Codefresh settings

- Attach clusters once
- Attach registries once
- Attach Helm repos once
- Attach X once (and then reference it by name)

The screenshot shows the 'INTEGRATIONS' section of the Codefresh UI. It lists several provider configurations with 'CONFIGURE' buttons:

- Datadog (NEW!) - Display Codefresh build analytics in your Datadog CI dashboard
- Git - Connect to GitHub, Bitbucket, or other Git providers for secure authentication to repositories
- Docker Registries - Integrate with Docker registries to make pulling and pushing images simple
- Kubernetes - Manage integrations with your Kubernetes clusters
- Helm - Manage integrations with Helm repositories and charts
- Slack - Send configurable notifications to Slack channels or users
- Cloud Storage - Connect cloud storage providers to make integrating your storage in a pipeline simple

Below this, the 'Kubernetes' provider configuration is expanded:

- Kubernetes**: Add or configure one of your existing clusters. New to Kubernetes? Follow these [simple instructions](#).
- ADD PROVIDER** button
- Amazon AWS**: 1 Clusters
 - myawscluster**: Show Cluster Nodes ▾
- ADD CLUSTER** button
- Custom Providers**: 1 Clusters
 - loft-real-cluster**: Show Cluster Nodes ▾
- ADD CLUSTER** button

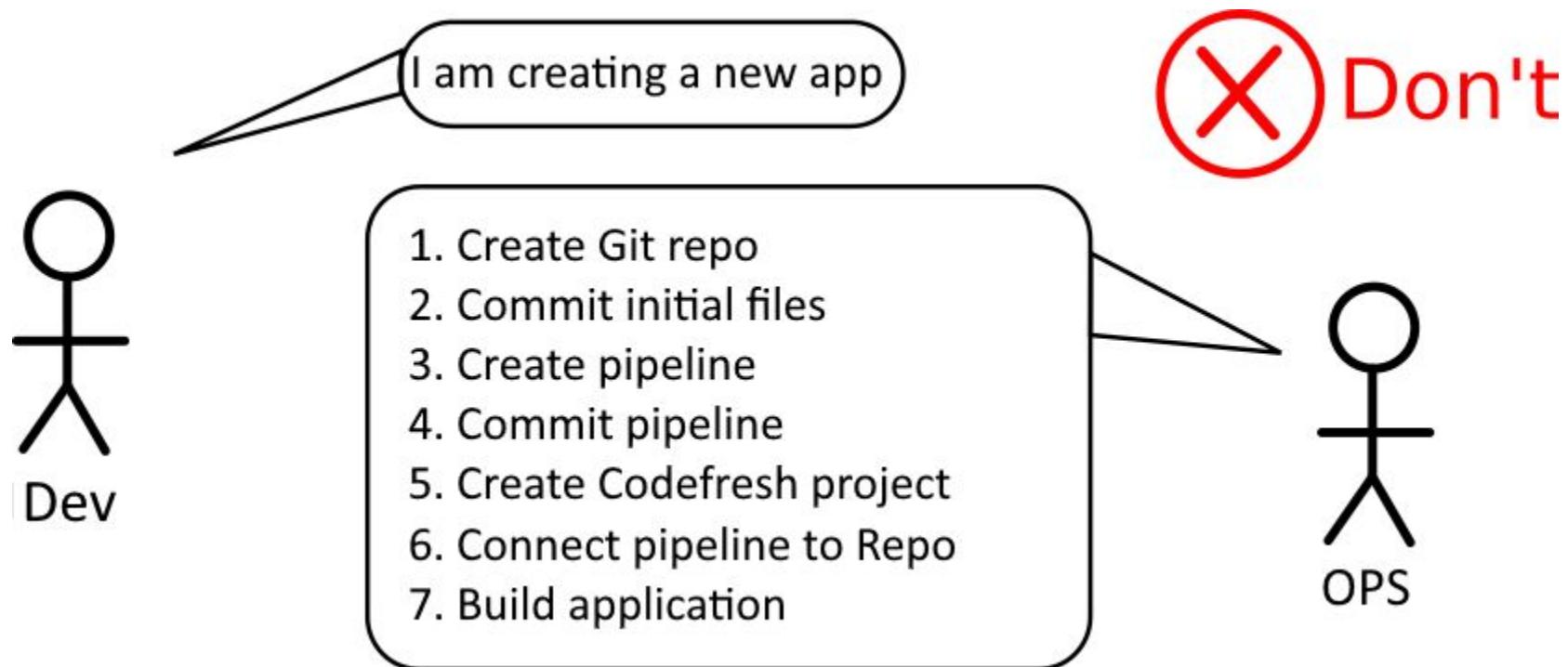
Demo 2

Creating a Port blueprint just
for developers

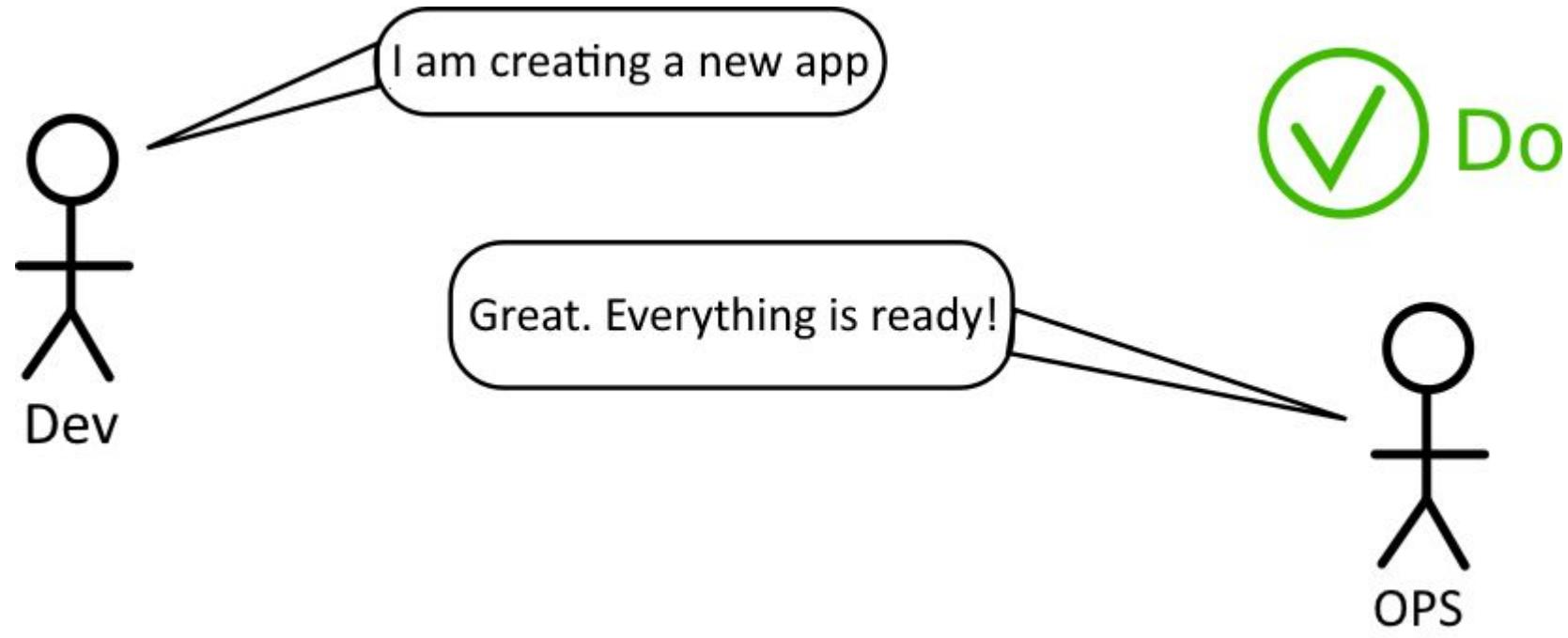
Real World example

Brand new application/micro-service

What usually happens



What should happen



Master pipeline

Create **Everything** a developer needs in a single step

01

Create Git Repo

Happens via GitHub
terraform provider

02

Create Clusters

Happens via Vcluster chart

03

Create Pipelines

Happens via Codefresh
Terraform provider

04

Connect Clusters

Happens via Codefresh

05

Ready!

Developer can start working

Terraform everywhere

- Codefresh can run Terraform natively
- Port can inject data with Terraform
- Terraform provider for GitHub
- Terraform provider for Virtual Clusters
- Terraform provider for Codefresh itself



Terraform provider (Codefresh)

Example Usage

```
resource "codefresh_project" "test" {
    name = "myproject"

    tags = [
        "production",
        "docker",
    ]

    variables = {
        go_version = "1.13"
    }
}
```

Terraform provider (GitHub)

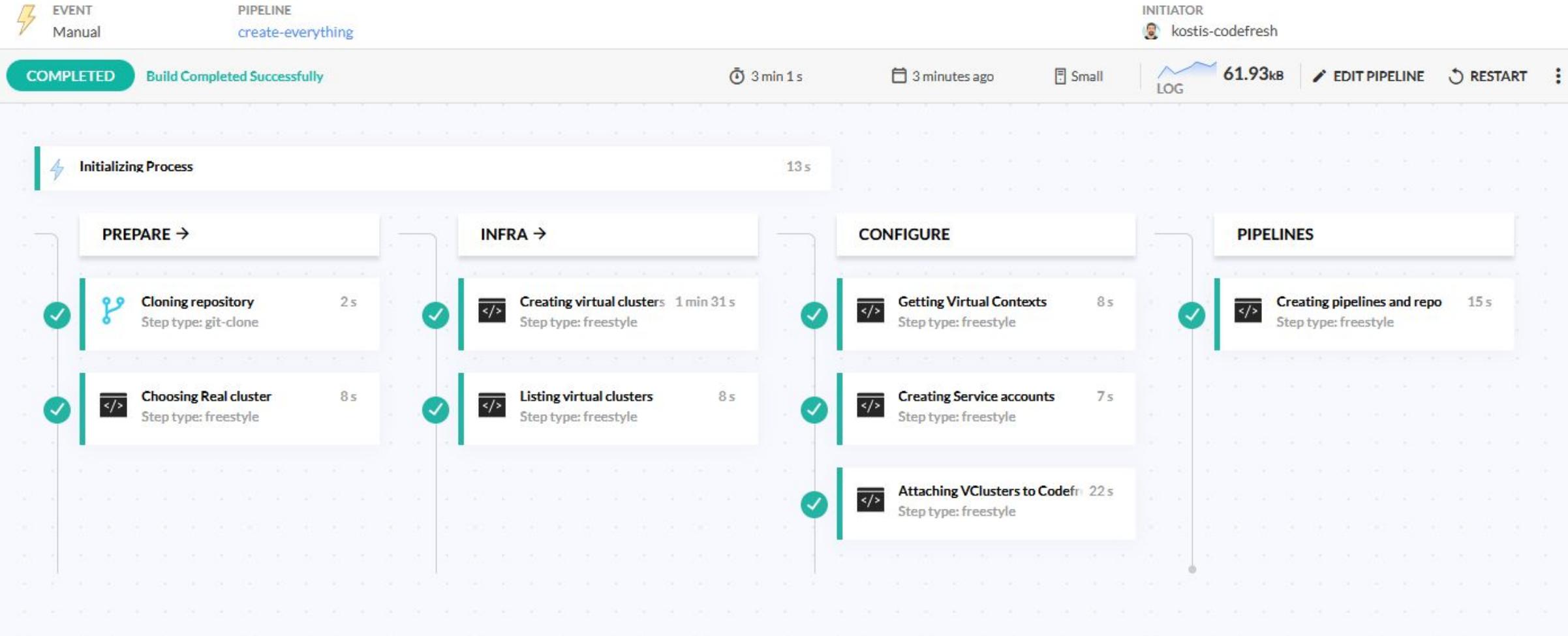
Example Usage

```
resource "github_repository" "example" {
    name      = "example"
    description = "My awesome codebase"

    visibility = "public"

    template {
        owner          = "github"
        repository     = "terraform-template-module"
        include_all_branches = true
    }
}
```

Master pipeline



Developer self-service

Port's internal developer portal lets developers provision, terminate and perform day-2 operations on any asset exposed in the catalog, within the policies and guardrails you've set.

The image displays two side-by-side screenshots of Port's developer self-service interface, illustrating the provisioning and creation of new services.

Spin up DevEnv

This screenshot shows the configuration for provisioning a development environment. It includes:

- Services:** Auth (release/0.2), Recommendation (release/0.4), Notification (release/0.5), Currency (release/0.6).
- TTL (Days):** 3, 5, 10, Manual Approval.
- A large button labeled **Spin up**.
- A callout box titled **Provision DevEnv** detailing the process:
 - Pending approval
 - Run Jenkins job
 - Self-terminate in 10 days
 - Update in Port

Scaffold new service

This screenshot shows the configuration for creating a new service. It includes:

- Repo name:** Input field.
- Language:** Python, Go, Java, Node.js.
- Product Unit:** Analytics, SaaS, Control Panel.
- Resources:** Icons for Docker, Kubernetes, Terraform.
- Description:** Input field.
- A large button labeled **Scaffold**.
- A callout box titled **Create new service** detailing the process:
 - Create new github repo
 - Apply template
 - Generate Terraform files
 - Update in Port

Demo 3

Creating everything a developer needs with a single click

Recap

- Create a Port model with Codefresh/Terraform
- Create Self-service actions for devs
- Actions call Codefresh pipelines/terraform/tools

- Developers are happy (self-serve)
- DevOps/SREs are happy (no involvement at all)
- DexEx are happy (no blockers, no tickets)

Thank you! 🚀

getport.io codefresh.io