Developer Experience on Continuous Delivery

Building a CD system for k8s that developers LOVE

Euccas Chen

Software Engineer

Tobi Ogunnaike

Software Engineer


Photo by Markus Spiske
Mission
Bring everyone the inspiration to create a life they love!

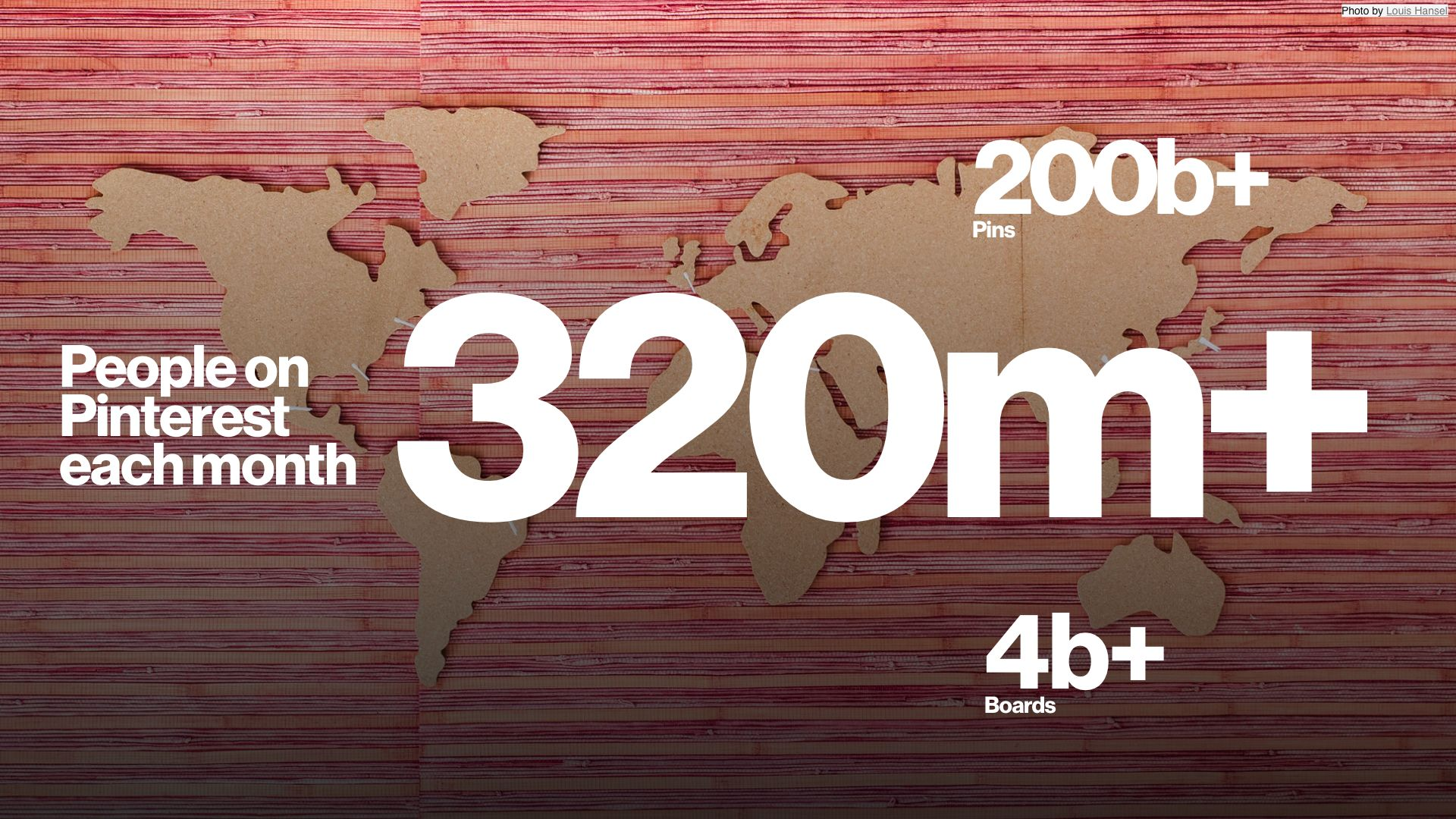


Photo by Louis Hansel
200b+
Pins
People on Pinterest each month
320m+
4b+
Boards

Infrastructure Footprint

$O(10^5)$
of servers

$O(10^4)$
of deploys / month

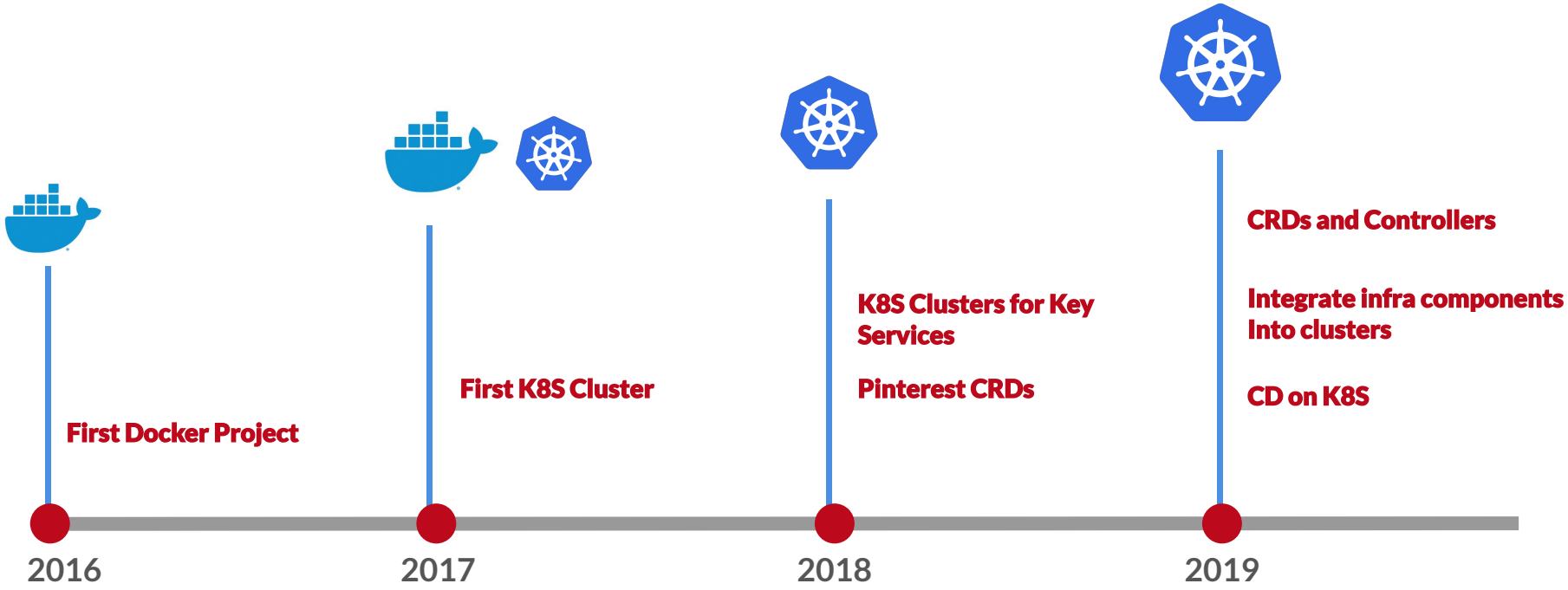
$O(10^3)$
of services


Engineering Productivity
Fast, safe & delightful path from an idea to production, without worrying about infrastructure
Photo by Markus Winkler

Agenda

1. Build a CD system for k8s
2. Adoption and migration
3. Lessons we learned

Kubernetes at Pinterest



Custom resources and controllers

Pinterest CRD

- Model unique workloads
- Inject runtime support
- Simplified config
- 6 CRD types
  - PinterestService
  - PinterestCronJob
  - PinterestJobSet
  - PinterestDaemon
  - PinterestTrainingJob
  - PinterestStatefulSet

```yaml
apiVersion: pinterest.com/v1
kind: PinterestService
metadata:
  name: exampleservice
  project: exampleproject
  namespace: default
spec:
  iamrole: role1
  loadbalancer:
      port: 8080
  replicas: 3  #Default 1
  sidecarconfig:
    sidecar1:
      deps:
      - example.dep
    sidecar2:
      log_level: info
  template:
    spec:
      initcontainers:
        - name: init
          image: gcr.io/kuar-demo/kuard-amd64:1
      containers:
        - name: exampleservice
          image: gcr.io/kuar-demo/kuard-amd64:1
```

CRD, 25 lines

Translated by controller

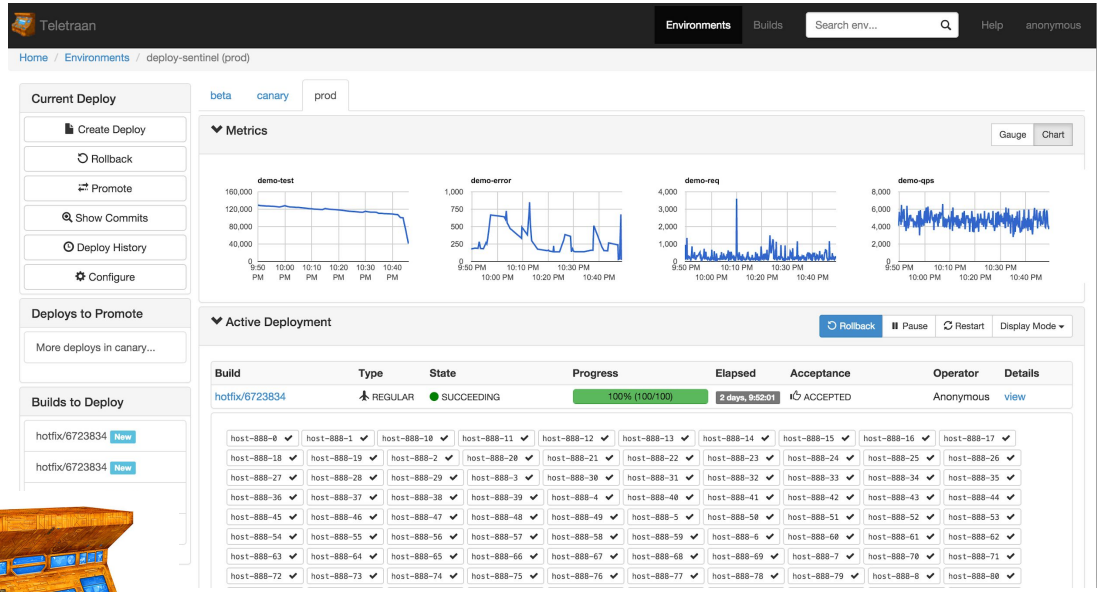
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    pinterest.com/identity: pinterest.exampleservice
  creationTimestamp: null
  labels:
    app: exampleservice
  name: exampleservice
  namespace: default
  ownerReferences:
  - apiVersion: pinterest.com/v1
    blockOwnerDeletion: true
    kind: PinterestService
    name: exampleservice
    uid: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
spec:
  replicas: 3
  selector:
    matchLabels:
      name: exampleservice
  strategy: {}
  template:
    metadata:
      annotations:
        pinterest.com/iamrole: role1
        pinterest.com/identity: pinterest.exampleservice
        pinterest.com/networkmode: dedicatedeni
        security.alpha.kubernetes.io/unsafe-sysctls: net.ipv4.conf.lo.
      creationTimestamp: null
      labels:
        app: exampleservice
        name: exampleservice
    spec:
      containers:
      - env:
        - name: KNOX_SERVICE_AUTH
          value: "1"
        - name: K8S_POD_NAME
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        - name: K8S_POD_ID
          valueFrom:
            fieldRef:
              fieldPath: metadata.uid
        envFrom:
        - configMapRef:
            name: exampleservice-configs
        image: gcr.io/kuar-demo/kuard-amd64:1
        name: exampleservice
        resources: {}
        volumeMounts:
        - mountPath: /usr/bin/knox
          name: system-knox
          readOnly: true
        - mountPath: /var/lib/knox
          name: pod-knox-lib
        - mountPath: /var/lib/normandie
          mountPropagation: HostToContainer
          name: system-normandie-lib
          readOnly: true
        - mountPath: /var/serverset
          name: system-serverset
          readOnly: true
        - mountPath: /var/config
          name: system-config
          readOnly: true
        - mountPath: /etc/zookeeper_hosts.conf
          name: system-zum-zk-hosts
          readOnly: true
        - mountPath: /etc/cell_zookeeper_hosts.conf
          name: system-zum-cell-zk-hosts
          readOnly: true
        - mountPath: /etc/pin
          name: system-pin
          readOnly: true
        - mountPath: /var/log
          name: system-pod-log
```

K8s resource, 380 lines

Current deployment system

Teletraan

- Deploy code to VMs
- Running since 2016
- 2.7K environments
- 7K deploys/day



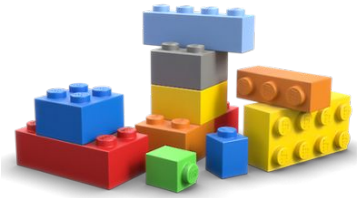
https://github.com/pinterest/teletraan

Hermez { Design, Build }

Tl;dr: We are building a new Continuous Delivery system for Kubernetes at Pinterest.

Deploying to k8s: Challenges

What problems are we solving?



Complexity

Operational toil

Pinterest specific

Deploying to k8s: What we want

Make it easy

Abstract away complexity

Minimal configs

Single interface

~~Complexity~~

End to end

From code commits to deployment

Visibility

Debuggability

~~Operational toil~~

Customization

Integrate with existing infra systems

Deployment pipelines

Migrate from Teletraan

~~Pinterest specific~~

Existing Solutions ?

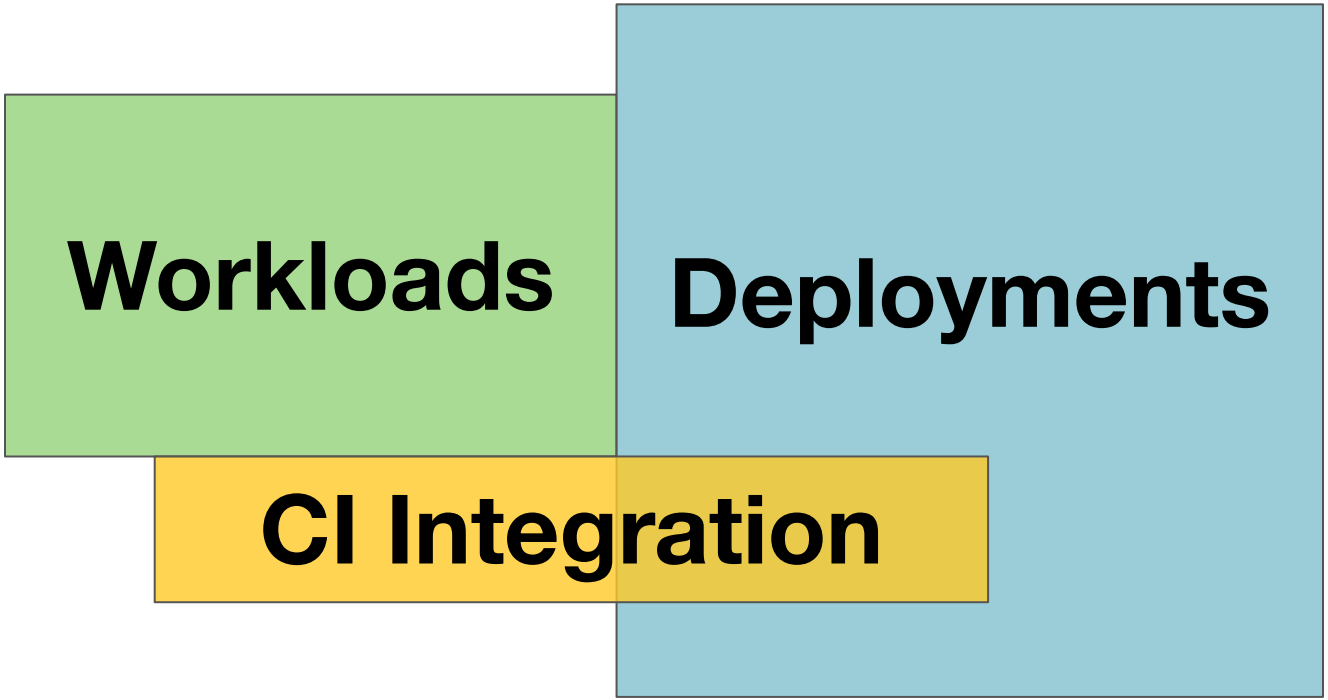
kubectl

pcloud


43

Introducing Hermez

1. The user-facing system for CD
2. Kubernetes first
3. Delightful developer experience


Workloads
Deployments
CI Integration

Workloads

Easy configuration

- Code repository
- K8s config file

Workload types

- K8s: workload types defined by Pinterest CRDs
- Data streaming, Teletraan

Operation support

- Workload healthiness, metrics, config change audit trail, authZ, notification

Create a workload

Category	Workload Name	Workload Type
Production	hermez-ui	PinterestService
Production	hermez-ui-testing	PinterestService
Production	kubecon-demo-cronjob	PinterestCronJob
Non-Production	kubecon-demo-jobsets	PinterestJobSets
Non-Production	kubecon-demo-service	PinterestService
Non-Production	kubecon-demo-statefulset	PinterestStatefulset
Production	kubecon-demo-streamingjob	PinterestStreamingJob
Non-Production	kubecon-demo-trainingjobs	PinterestTrainingJobs

Deployments

Deploy commits and PRs

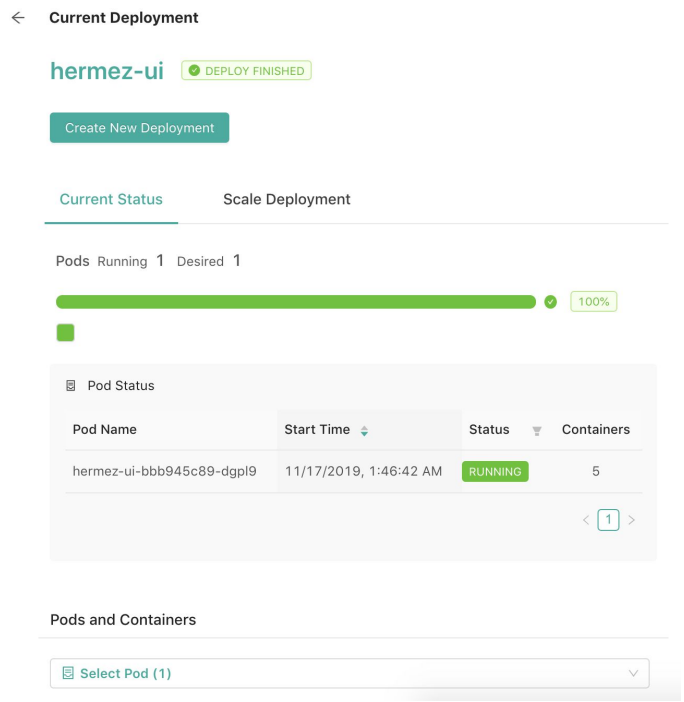
- Rollback, hotfix
- Scale a deployment: manual, auto-scaling

Continuous Delivery pipelines

- Standard deployment pipelines
- Integrate with Spinnaker to run pipelines

Visibility

- Current running version, deployment details
- K8s: Pod and container status, events, logs
- Deployment history



CI Integration

Build pipelines

- Support individual service's repo and monorepos
- Build container images
- Publish k8s artifacts

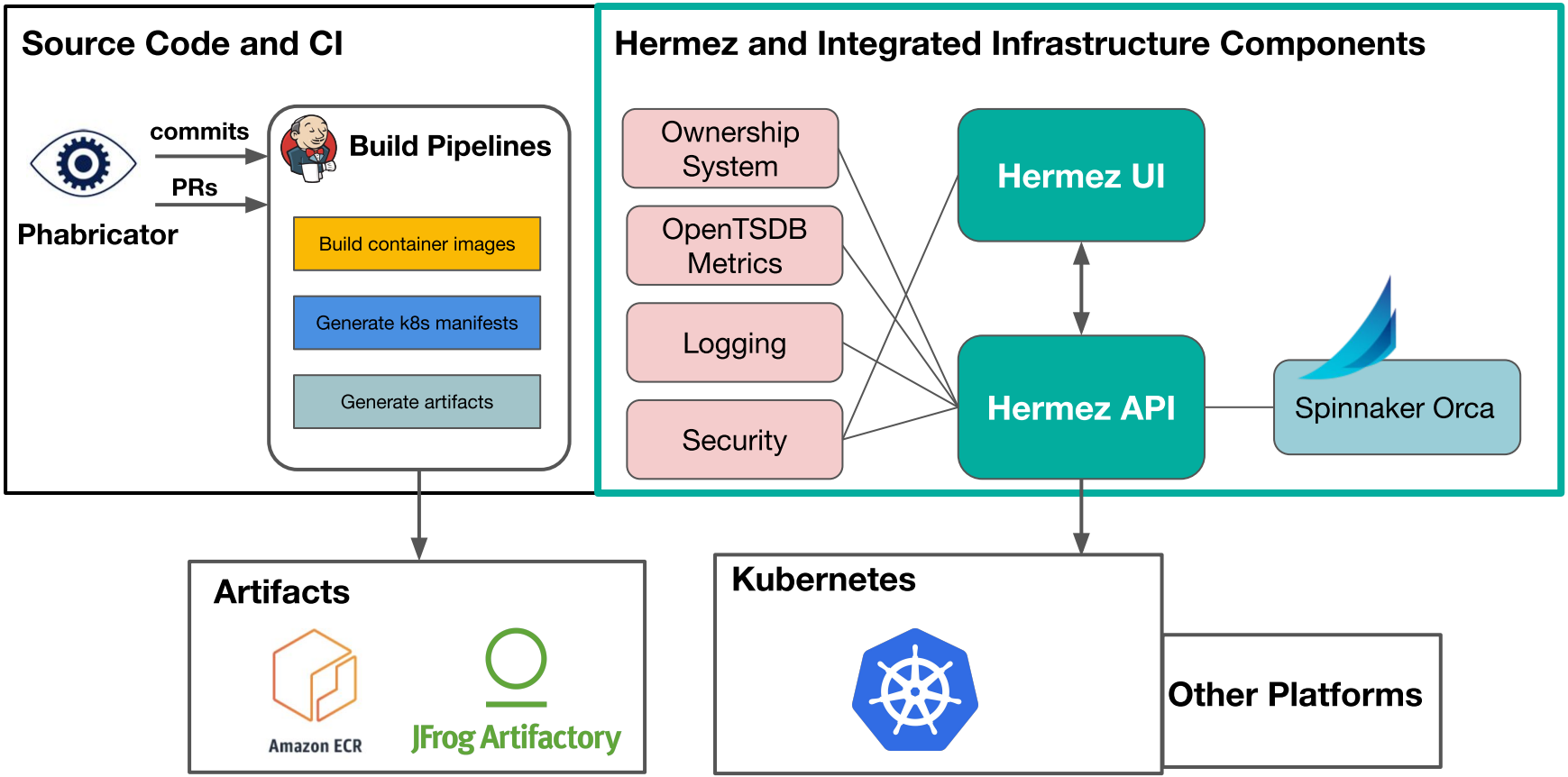
Bridge CI and CD

- Visualize the process of "from code commits to deployment"
- Logs for debugging
- Trigger build on-demand

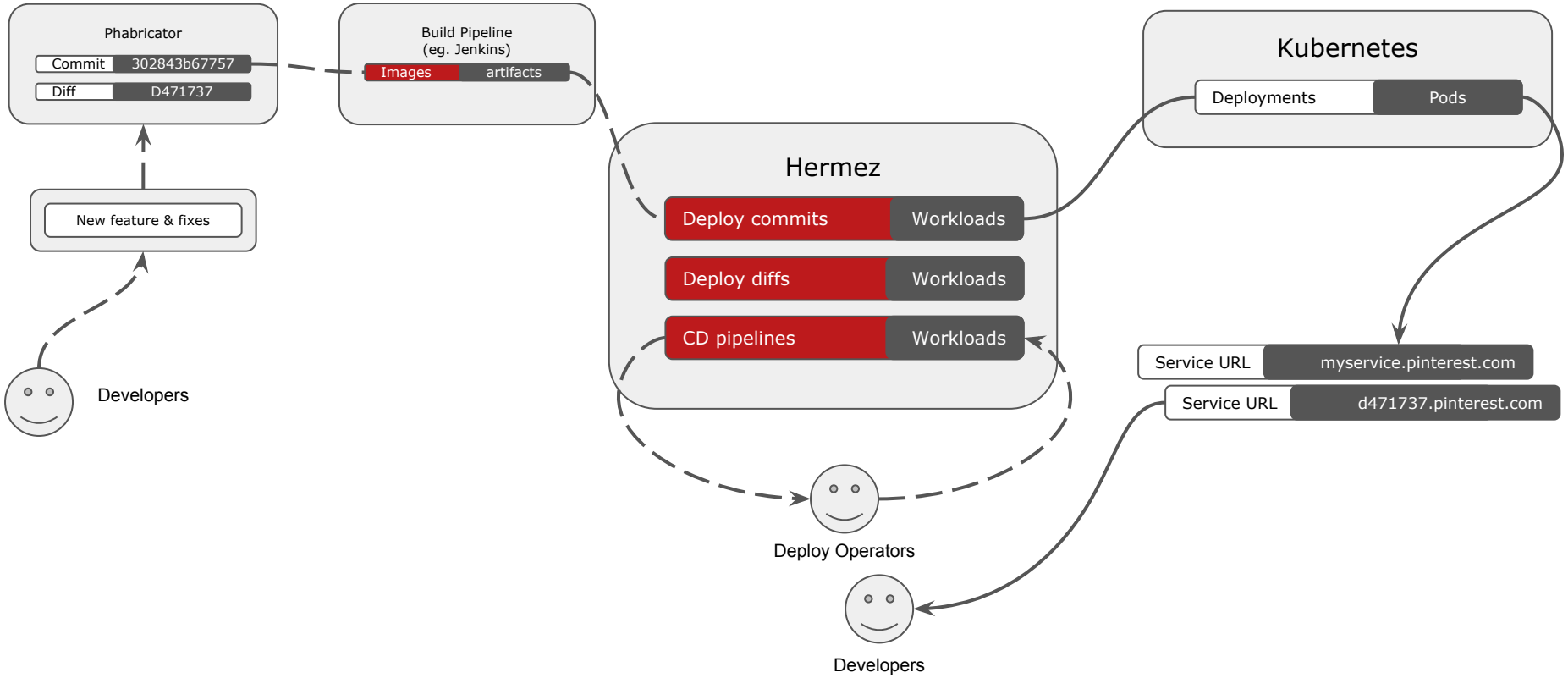
	Commit	Branch	Author	Commit Time	Artifacts Status
+	95f6178	master	@euccaschen	Fri Nov 15 2019 1:48:16 PM	Artifacts Available
+	8b360be	master	@oogunnaike	Fri Nov 15 2019 1:35:23 PM	Artifacts Available
+	063470d	master	@euccaschen	Fri Nov 15 2019 12:02:03 PM	Artifacts Available
-	8e9fc98	master	@euccaschen	Fri Nov 15 2019 10:21:46 AM	Artifacts Available

8e9fc98e872a9d279440e579dbcf3600eed5dd7e workloads summary table: persist user's searching

Deployment process



A developer’s experience



Hermez { Adoption, Migration }

Tl;dr: We are helping Pinterest engineering teams to deploy and migrate their services onto Kubernetes using the new CD system.

Customer adoption is not easy

“If you build it, they will come”

Said no successful product owner ever.

What can Hermez do for me?

And why should I care?

Demo time for a Cronjob!

Cronjob demo

Context - No single, recommended path for deploying cronjobs at Pinterest

Call outs

1. **First class support for cronjob operation** *#feature*
   view cronjob schedule, execution history, next scheduled run in the UI

2. **Easy integration with existing systems** *#minimal-config*
   build systems, artifact stores and docker registry

3. **Debuggability** *#dev-experience*
   container logs, pod status, workload metrics

Demo time for a Service!

Service demo

Call outs

1. **Deploy PRs with easily shareable URLs** #feature

2. **Easy integration with existing systems** *#minimal-config*
   build systems, artifact stores and docker registry

3. **Debuggability** *#dev-experience*
   container logs, pod status, workload metrics

How we prepared Hermez for adoption

1. Reduced scope
   Limited workload types (PinterestService, PinterestCronjob)

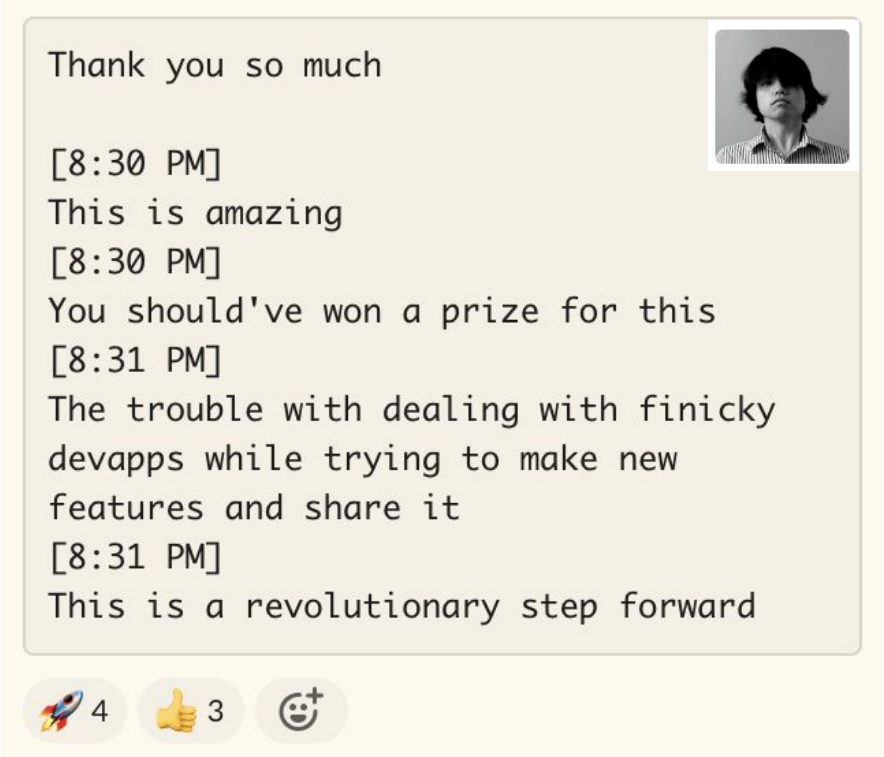
2. Partner with **early adopters**
   (SRE, Ads, Tools)

3. Evaluate feedback, iterate & improve

4. Knowledge sharing - onboard runbook, status updates, demos, brownbags

5. Self-service migration tools

Homefeed + Hermez: a fairytale adoption story



Hermez { Learning }

Tl;dr: We want to share our experience and collaborate with the community.

Our path to a new CD system

1

Design

User story

UI mockup

Feedback sessions with teams

2

MVP

Minimal set of features

Hackathon

Gather feedback

3

Dogfood

Use Hermez to deploy Hermez

Find early adopters

Iterate and learn

4

Production

Break into smaller scopes

Onboard new services

Migrate existing services

What have we learned ?

- Treat workloads as the first class objects
- Do not force users to know about Kubernetes
- Start security integration early
- Form and UX language matters

Reach out to us

#engineering-productivity-team

Euccas Chen

euccaschen@pinterest.com
https://www.linkedin.com/in/euccas

Tobi Ogunnaike

oogunnaike@pinterest.com
https://www.linkedin.com/in/tobiogunnaike