



# Enabling Developers with Open Source

**Nicholas Parks, Cloud  
Architect, Kenzan**

@nparksnyc



# Agenda

## Introduction

Background

What is Developer Enablement?

Cloud, Containers, Canary, Continuous Delivery

## Hello Capstan

What is it?

What do I use it for?

How does it work?

## Demonstration

Use A Capstan Created Environment

How can I modify?

Questions

# Business Drivers for Developer Enablement

**Maximize Creative Hours to work on:**

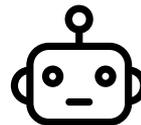
**User  
Experiences**



**Improved  
Features**



**Smarter  
AI/ML**



**Not on...**

**Why did my build break?**

**Which server did it deploy to?**

# Business Drivers for Developer Enablement

You pay for “Value Delivery”

**Customers**  
(External/Internal)



**Open Source  
Community**



**Greater Business  
Ecosystem**



**No one will accept...**

“Our deployment process is holding up releases to customers”

# Hurdles

**Getting Started with  
Containerized Apps?**

**Moving to the cloud?**

**Release all day, every  
day?**

Time to bootstrap  
containerized app  
experience

Existing Continuous  
Delivery Skill Deficit

How to “Solve”  
Operational complexity

Existing Automation =  
Rube Goldberg  
Fragility

Developers may not  
have extensive cloud  
experience

Where to start  
automating?

How do I know that  
software is “good” to  
release?

# What is Developer Enablement?

**Enable developers to execute as many facets of the creative process independently and on-demand.**

Cloud capable with the latest software packaging technology (containers) that can perform automated canary analysis in a continuous, repeatable, duplicatable way.

# Cloud and Containers

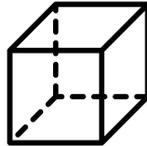


## Cloud

Any IaaS that provides elastic behavior transparently to the creator.

Provides a place to run what you created without needing to know all inherent machinations

The IaaS experience could be a PaaS experience.



## Containers (with orchestration)

Runtime environment where the lifecycle is managed without the developer

Packaging of the app/feature/service, independent of the IaaS

Packaging on the local development environment is the same as in the enterprise runtime

# Canary

## What are canary releases?

A change to a small subset of **production** users to gauge the fitness of the software before release to all customers.

Ability to execute *one last point of quality control* in **production** before fully committing to the new code/feature/hotfix/etc.

Canary is not new, but **Generic Automated Canary Analysis** is new(er) and uses statistical methods to **automatically** make this fitness determination.

# Continuous Delivery/Deployment

## Push the best, all day every day

*Continuous Delivery* is the orchestrated manifestation about how software is delivered within an organization regardless of whatever SDLC is followed.

With *Continuous Deployment*, this release occurs **without** human intervention



***...in a lean, post-agile, software company,  
Continuous Delivery offers a possibility to  
improve the development process, and  
adopting it is beneficial when aiming for a  
development culture that can fluently move  
to new business directions.***

[Marko Leppänen et Al. Towards post-agile development practices through productized development infrastructure.](#)

A decorative graphic on the left side of the slide. It consists of a large, semi-transparent blue circle that overlaps a smaller, solid blue circle. The background of the circles is a photograph of a forest with tall evergreen trees and a rocky path.

# Capstan

<https://github.com/kenzanlabs/capstan>

# Capstan

**Kenzan's new open source solution**

**Provision disposable containerized CI/CD environment in minutes**

**Github / Kenzan Labs / capstan**



## **Technologies**

Terraform

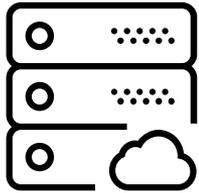
Cloud SDK

Spinnaker

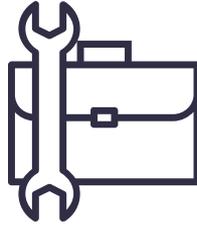
## **IAAS Provides**

Kubernetes

# Capstan



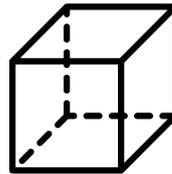
**Infrastructure as Code (IaC)**



**Explicit display of tool usage and configurations**



**Immutable Infrastructure**



**Container focus**

# Capstan



## Developer Acceleration

***Need to get your developers quickly aquantited to developing containerized applications?***

No longer burdened with training developers on how to deploy to Kubernetes.

If you can push your container into a supported repository and use Spinnaker than you can use Capstan.

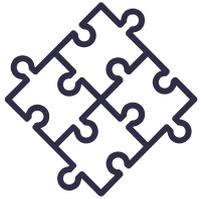


## Developer Sandbox

***Need a play place? Need to deliver apps that also interact with KubeFlow?***

With Spinnaker you can push apps into a Kubernetes Cluster that also has KubeFlow deployed.

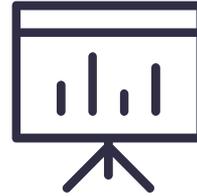
# Capstan



## Illustrate Bootstrapping

*Trying to put together infrastructure-as-code, a container platform, and software delivery platform?*

Capstan provides a reference implementation that your organization can bootstrap from.



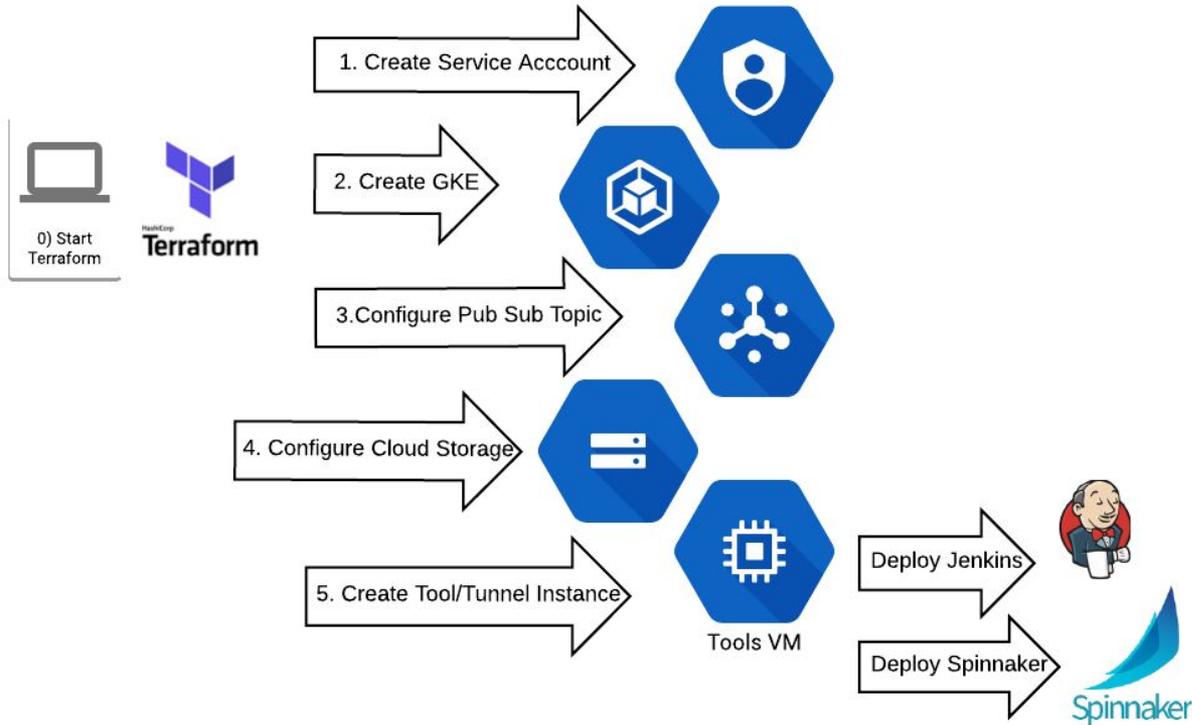
## Platform Ops Training

*How do I operate my platform in the cloud?*

*How do I perform IaaS changes or Kubernetes upgrades?*

Capstan provides a reference architecture for hosting containerized applications.

# Capstan



# Capstan

## How does it work?

### In Google Cloud...

1. Setup Trial Account
2. Create GCP Project
3. Create & Download Service Account JSON

### On Your Workstation

1. Install the Basics, SDK & Kube Components (optional)
2. Clone Github / Kenzan Labs / Github
3. Copy Service Account JSON as

```
$ git clone  
$ cd capstan  
$ git checkout 2018.q3.beta.gcp  
$ cp ~/Downloads/<service_account>*.json \  
gcp/terraform/gcp-account.json
```
4. ...other steps in GCP Readme

## Validate your GCP project

### Connectivity

To make sure we don't stumble into problems later, you need to perform the following:

1. Create a [Service account](#) with 'role/owner' for Terraform. Call it `terraform-admin` (or anything i. If you are presented with the option to generate a JSON key file do and save it for later.
2. Create a micro instance in `us-central1-a` with the service account `terraform-admin` (or what you named it).
3. Now leave your browser and open a terminal window
4. Perform a [gcloud init](#) if you have not done so as part of installing gcloud
  - i. Make sure your environment is referencing the current project (via `gcloud info`)
5. From your laptop perform a `gcloud ssh` into said instance. You can get the full `gcloud` command by clicking the arrow next to the SSH button for the instance.
  - i. This is to check connectivity between your laptop to GCP in a manner similar to what terraform does.

If everything happened without issue then we are good. You no longer need this test instance. You can delete it. You can also use the service account to set-up terraform.

### Enable Google Project Features

After verifying connectivity, we need to enable services/api endpoints for terraform.

Using the terminal window where you just attempted `gcloud ssh` perform the following commands

```
project
```

1. `gcloud services enable container.googleapis.com`
2. `gcloud services enable iam.googleapis.com`

Plan: 9 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

Terraform will perform the actions described above.

Only 'yes' will be accepted to approve.

Enter a value: yes

random\_integer.spin\_bucket: Creating...

max: "" => "99999"

min: "" => "1"

result: "" => "<computed>"

random\_integer.spin\_bucket: Creation complete after 0s (ID: 89144)

google\_service\_account.halyard\_toolsacct: Creating...

account\_id: "" => "halyard-tunnel-tools"

display\_name: "" => "halyard-tunnel-tools"

email: "" => "<computed>"

name: "" => "<computed>"

project: "" => "<computed>"

unique\_id: "" => "<computed>"

google\_service\_account.spinnaker: Creating...

account\_id: "" => "gcp-spinnaker"

display\_name: "" => "gcp-spinnaker"

email: "" => "<computed>"

name: "" => "<computed>"

project: "" => "<computed>"

unique\_id: "" => "<computed>"

google\_pubsub\_topic.gcr\_event\_stream: Creating...

name: "" => "gcr"

project: "" => "<computed>"

```
unnel (remote-exec): Success
unnel (remote-exec): + Run `hal deploy connect` to connect to Spinnaker.
unnel (remote-exec): =====
unnel (remote-exec): - Hopefully Spinnaker Deployed -
unnel (remote-exec): =====
unnel (remote-exec): *****
unnel: Creation complete after 16m0s (ID: halyard-tunnel)
```

ed, 0 changed, 0 destroyed.

```
te-exec): Success
te-exec): + Run `hal deploy connect` to connect to Spinnaker.
te-exec): =====
te-exec): - Hopefully Spinnaker Deployed -
te-exec): =====
te-exec): *****
ition complete after 14m45s (ID: halyard-tunnel)
```

Apply complete! Resources: 9 added, 0 changed, 0 destroyed.

parken01-mac:terraform parken\$ □



**Demo**

# Using Capstan

## Delivery Environments

Spinnaker

Jenkins

Kubernetes

## Using

Look at existing pipeline:

Simple

Canary

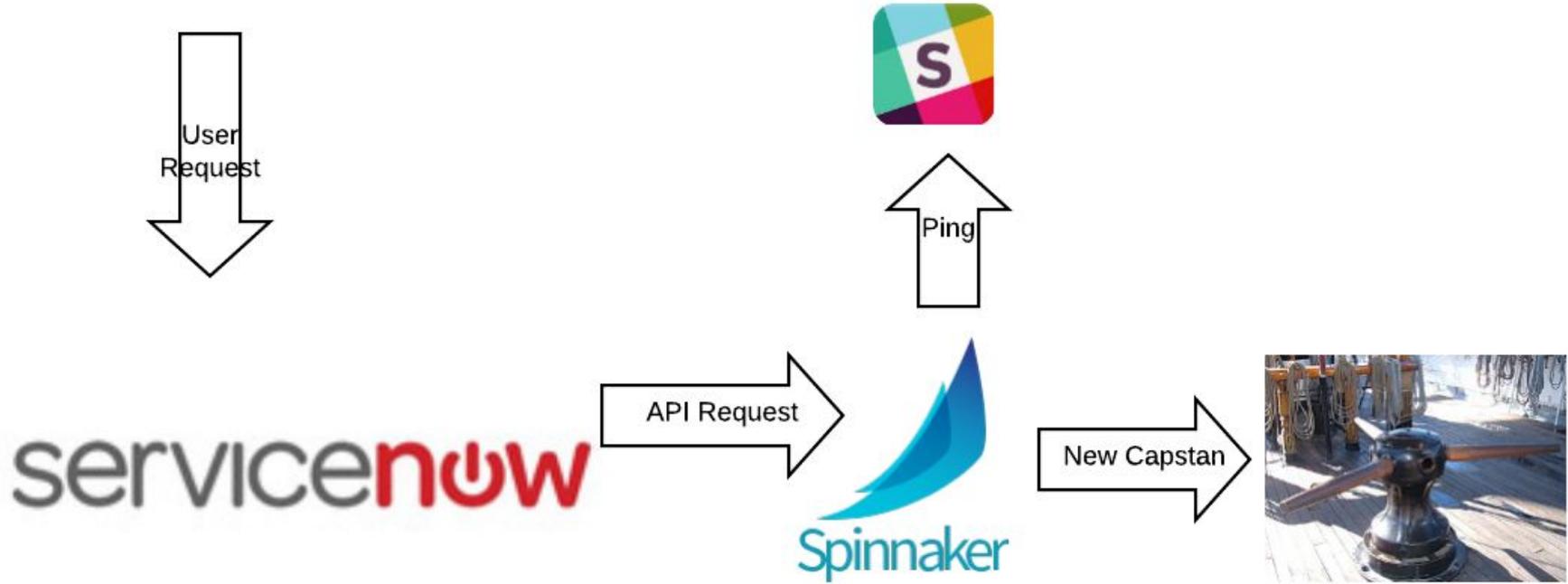
**Build One**

## Being Open Source

How can you change this?

**Service Delivery with ITSM**

# Open Source





THE LINUX FOUNDATION  
**OPEN SOURCE SUMMIT**