

IBM Code Day 2018

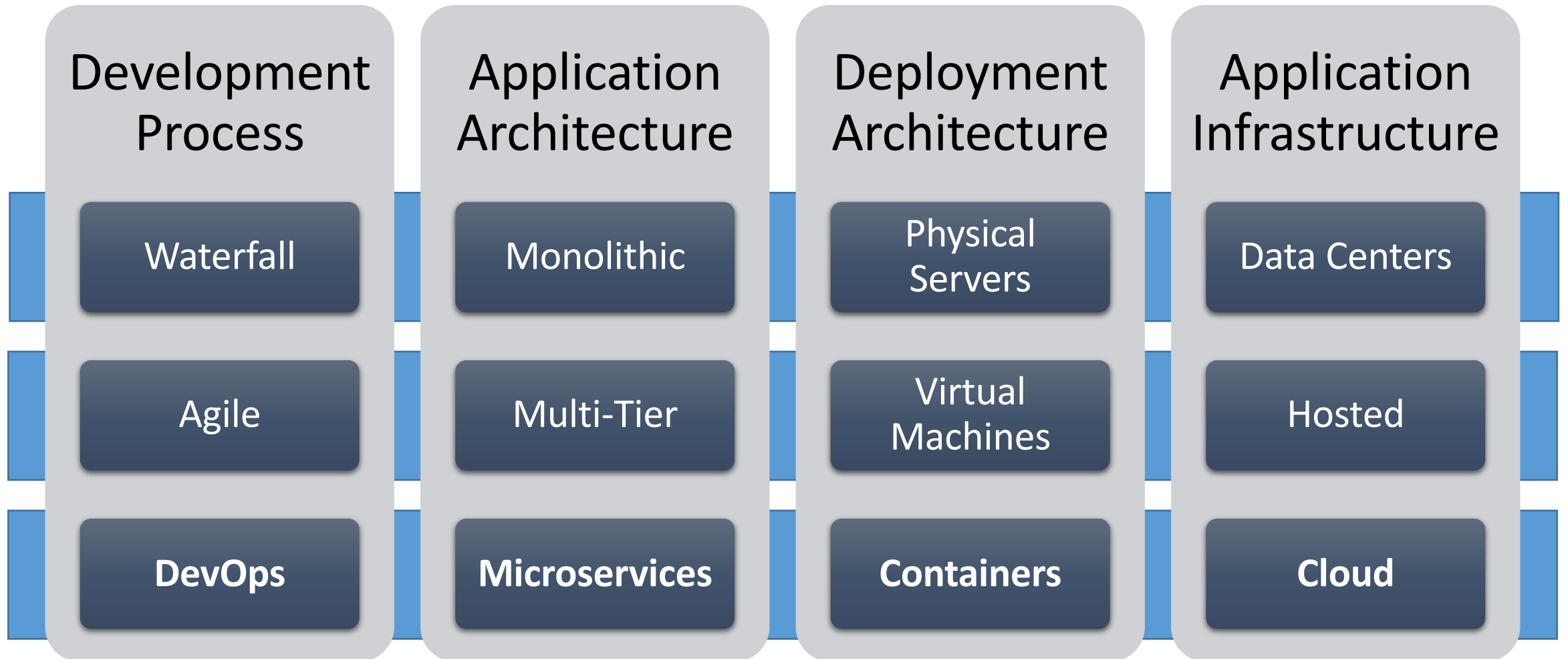
CODE. CONTENT. COMMUNITY.

Enterprise Kubernetes Cluster with IBM Private Cloud

#IBMCodeDay

Shajeer Mohammed
STSM - IBM Cloud Private
Twitter : @mshajeer

Our IT World Morphs



Containers bring Big Wins for developers

Highly Portable
Packaging solution -
for microservices

Helps you quickly
create fast, stable,
reliable, scalable,
fault-tolerant apps.

Lightweight,
Encapsulated OS
abstraction

Dev Environments
that match OTHER
Dev and Test
Environments

No more waiting for
weeks for a VM just
so you can run a
series of tests

Kubernetes has made
the Ops experience
much easier for the
developer

During development you
get to be as close as
possible to production
environment with
minimal changes

With micro-service
based applications,
every service can be
developed
independently

Access to a non-
public environment
to do continuous
integration

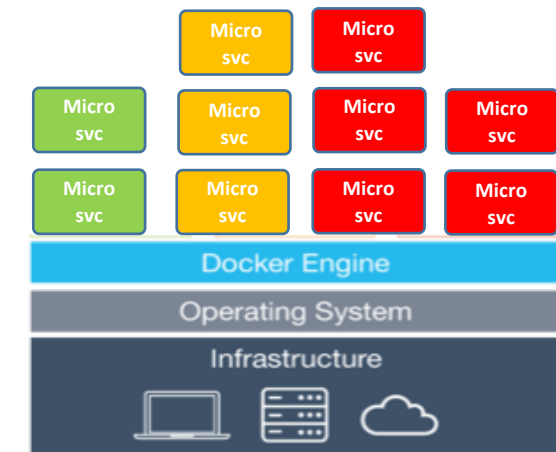
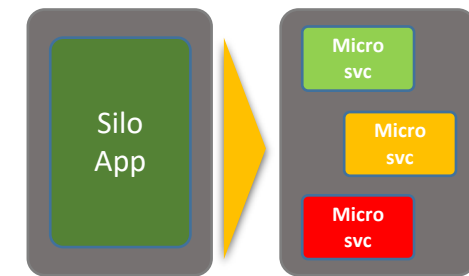
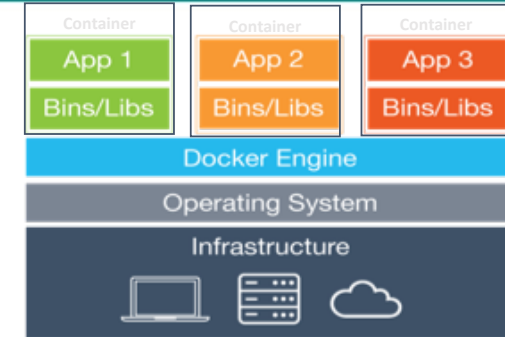
Cloud native building blocks

Container

Container allows you to **package an application** with all of its dependencies into a standardized unit for software development.

Microservice

Microservice architecture is about **breaking down** large **silos applications** into more manage-able fully decoupled pieces that will provide **agility, scalability** and **resilience**.



Where to Use Containers

Best suited for a Cloud-native Application.

The properties/requirements for a Cloud-native Application are formulated by the Cloud Native Computing Foundation (CNCF)

You will benefit mostly from IBM Private Cloud, if your applications are micro-service based and every service has a separate container.

Your microservices are stateless

The Twelve Factors

I. Codebase

One codebase tracked in revision control, many deploys

II. Dependencies

Explicitly declare and isolate dependencies

III. Config

Store config in the environment

IV. Backing services

Treat backing services as attached resources

V. Build, release, run

Strictly separate build and run stages

VI. Processes

Execute the app as one or more stateless processes

VII. Port binding

Export services via port binding

VIII. Concurrency

Scale out via the process model

IX. Disposability

Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity

Keep development, staging, and production as similar as possible

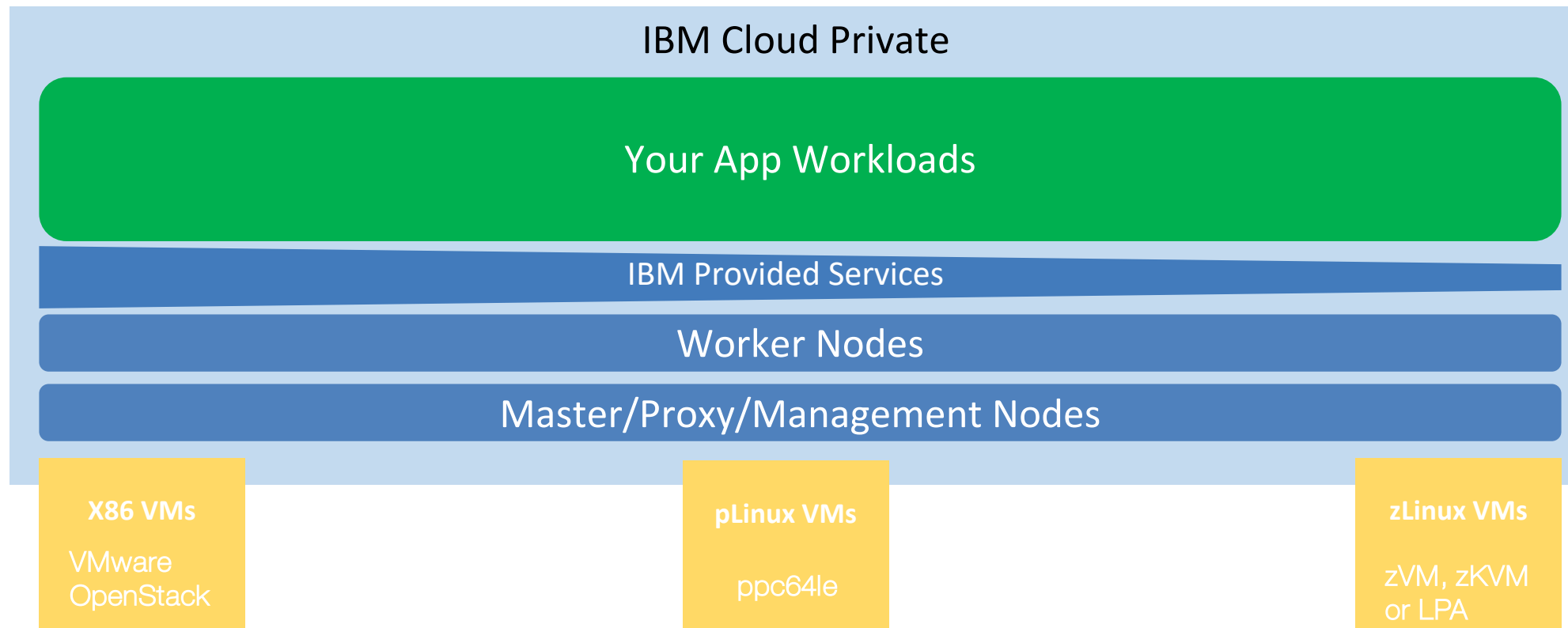
XI. Logs

Treat logs as event streams

XII. Admin processes

Run admin/management tasks as one-off processes

Your Workloads, Your Infrastructure



Mix and match worker nodes to run Kubernetes cloud apps you need on the infrastructure you have. Manage from the same master node.

An integrated platform with enterprise content

BYOL



IBM
WebSphere



Enterprise Content

Next Generation IBM Middleware, Data, Integration & Analytics



DevOps



Hybrid Cloud Mgmt

Core Operational Services

Application Development, Integration, Operations & Management

IBM Cloud Private Platform

- Kubernetes-based platform
- Multiple open compute models
- Software and policy-driven network storage
- Auto-scaling
- Multi-site HA/DR features

- Stateful and stateless application support
- Built-in monitoring and logging
- Integrated enterprise-grade security
- Vulnerability Advisor to prevent risk



Kubernetes



CLOUDFOUNDRY



Apache OpenWhisk



Runs on Enterprise Infrastructure



IBM Code Day



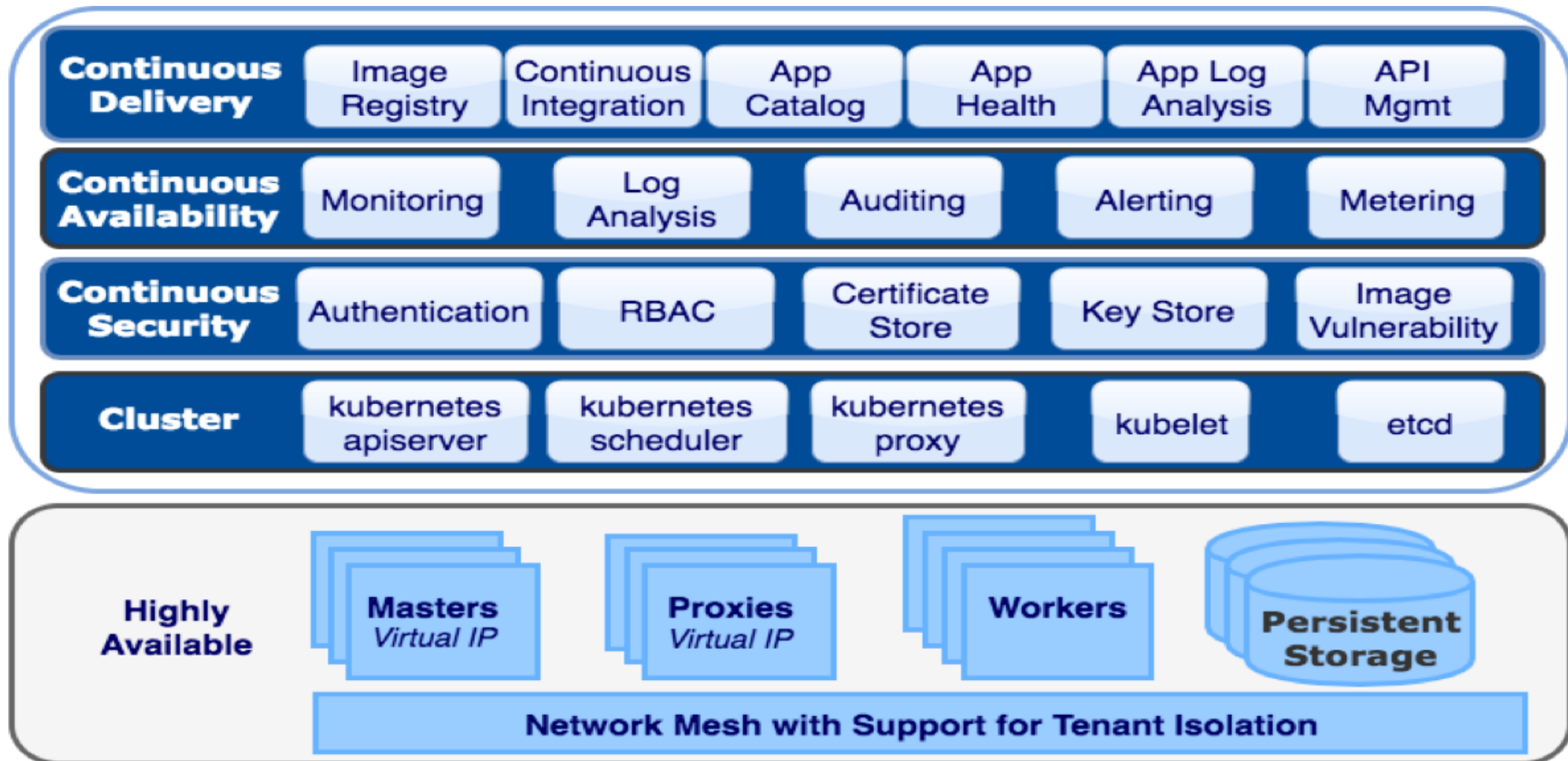
z Systems

vmware®

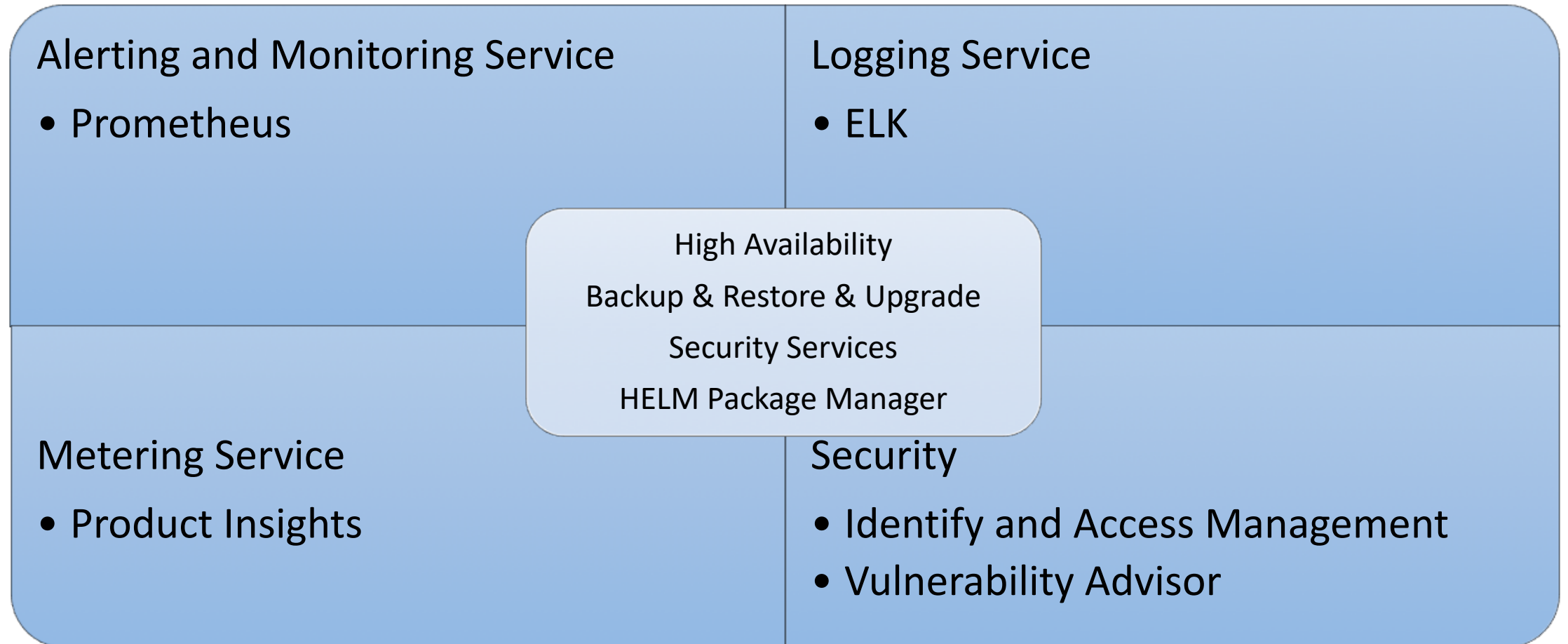


openstack®

IBM Cloud Private Architecture



IBM Cloud Private Enterprise Features



IBM Cloud Private Editions

Community

Platform

- Kubernetes
- Core services
- Content catalog

**Freely Available
in Docker Hub**

Cloud Native

Platform

- Kubernetes
- Core services
- Content catalog

Cloud Foundry (Optional)

IBM Enterprise Software

- Microservice Builder
- WebSphere Liberty
- IBM SDK for node.js
- Cloud Automation Manager

Enterprise

Platform

- Kubernetes
- Core services
- Content catalog

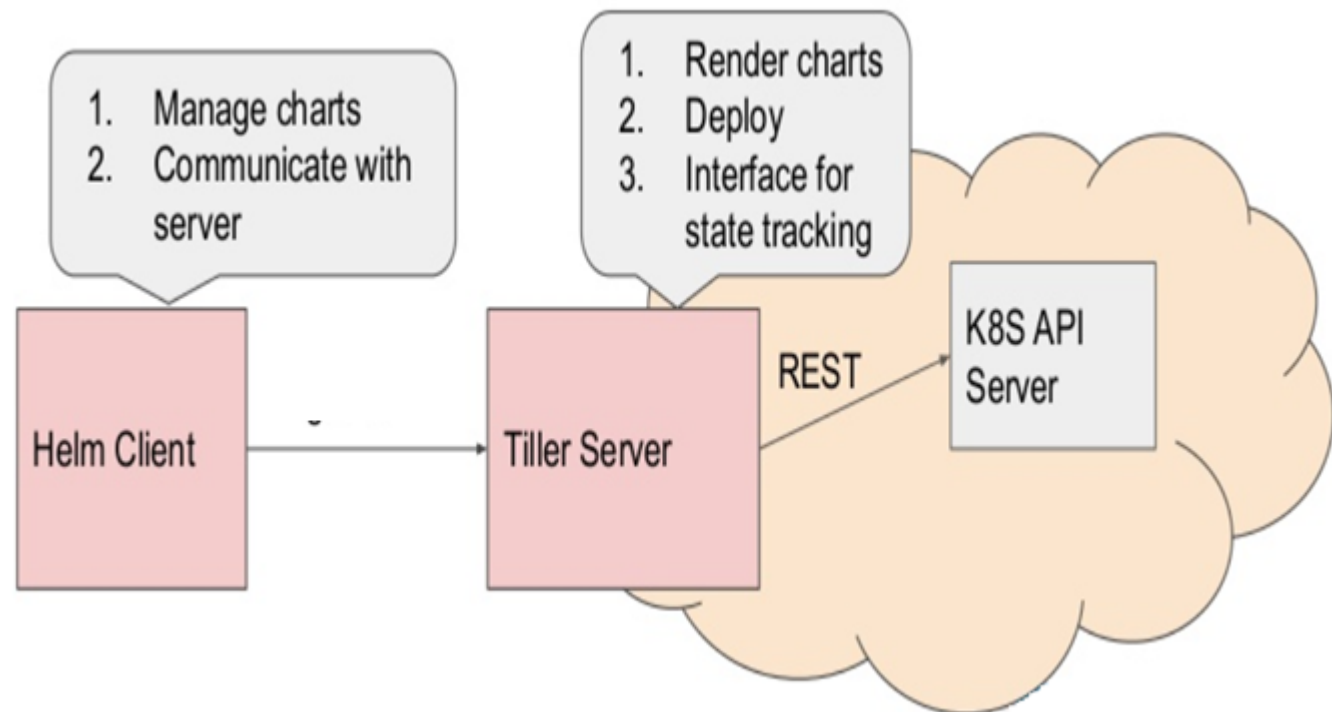
Cloud Foundry (Optional)

IBM Enterprise Software

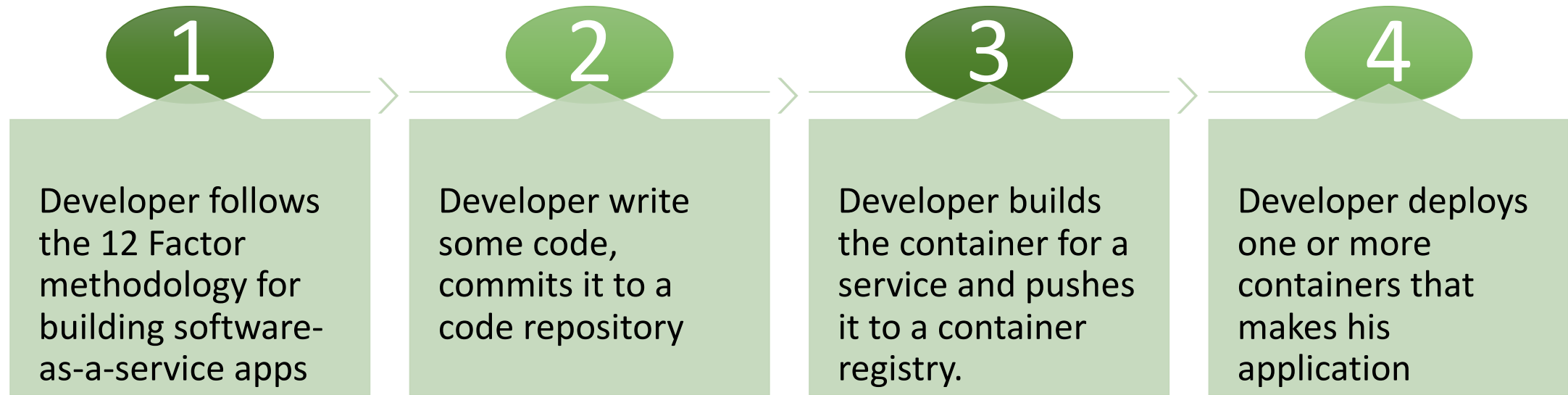
- Cloud Native Edition, plus:
- + WAS ND
 - + MQ Advanced
 - + API Connect Professional
 - + Db2 Direct Advanced
 - + UrbanCode Deploy

HELM Charts

- Helm helps you manage Kubernetes applications
- Charts are easy to version, share, and host on public or private servers.
- helm rollback can be use to roll back to an older version of a release with ease.
- **Chart** is a package helps you define, install, and upgrade complex Kubernetes applications
- **Repository** is the place where charts can be collected and shared
- **Release** A chart instance is loaded into Kubernetes, it then becomes a release. (Same chart can be installed several times into the same cluster. Each will have its own release.)



A basic development workflow for Kubernetes



- Describe the micro-service deployment ([worker nodes](#), [pod/deployment](#), [ingress mapping](#), [load balancer](#), [storage](#), etc.)
- [Deploy the micro-services](#) in Kubernetes (IBM Cloud Private) in [dev/test](#), [stage](#) or [production environment](#)
- Incrementally, [modify the deployment description](#) and deploy

Demo

- Quick walk thru of ICP
- Demonstrate how to customize an existing Jenkins HELM chart and how ICP can be used to deploy Jenkins and create a new POD for every new build request and destroy the POD after the build is complete.
 - Deploy a Jenkins HELM Chart
 - Configure Jenkins with Kubernetes Plugin
 - Configure a simple build job in Jenkins
 - Start a build

Learn More

- Deploy Cassandra Database -
<https://developer.ibm.com/code/patterns/deploy-a-scalable-apache-cassandra-database-on-kubernetes/>
- Deploy a cloud-native microservice -
<https://www.ibm.com/cloud/garage/tutorials/cloudnative-microservices-cloud-private>
- Set up a Jenkins pipeline -
<https://www.ibm.com/cloud/garage/tutorials/cloud-private-jenkins-pipeline>
- Transform traditional WebSphere apps to WebSphere Liberty -
<https://www.ibm.com/cloud/garage/content/course/websphere-on-cloud-private/0>



<https://developer.ibm.com/code/>



Signup for IBM Cloud
<https://bluemix.net>



<https://www.ibm.com/watson/products-services/>



Code & instructions

<https://github.com/IBMDevConnect>
<https://github.com/IBM>
<https://github.com/IBM-Cloud>
<https://ibm-cloud.github.io/#/>
<http://ibm.github.io>
<https://github.com/watson-developer-cloud>
<https://github.com/ibm-bluemix-mobile-services>



developerWorks

<https://developer.ibm.com/in/>
<https://developer.ibm.com/tv/>



Recipes

<https://developer.ibm.com/recipes/>



Join our Slack team and stay in touch with the experts

<https://ibmdevconnect.slack.com>

Send in your request

<http://ibm.biz/slackrequest>



Data Science Experience

<https://datascience.ibm.com>



Apply for IBM Global Entrepreneur Program

<https://developer.ibm.com/startups>

Join our Meetup groups



Bangalore : <https://www.meetup.com/IBMDevConnect-Bangalore>

Delhi / Gurugram / Noida :
<https://www.meetup.com/ibmcloudecosystem/>

Mumbai / Pune : <https://www.meetup.com/Cloud-Mumbai-Meetup/>

Hyderabad / Vishakapatnam:
<https://www.meetup.com/Hyderabad-Cognitive-with-Cloud>

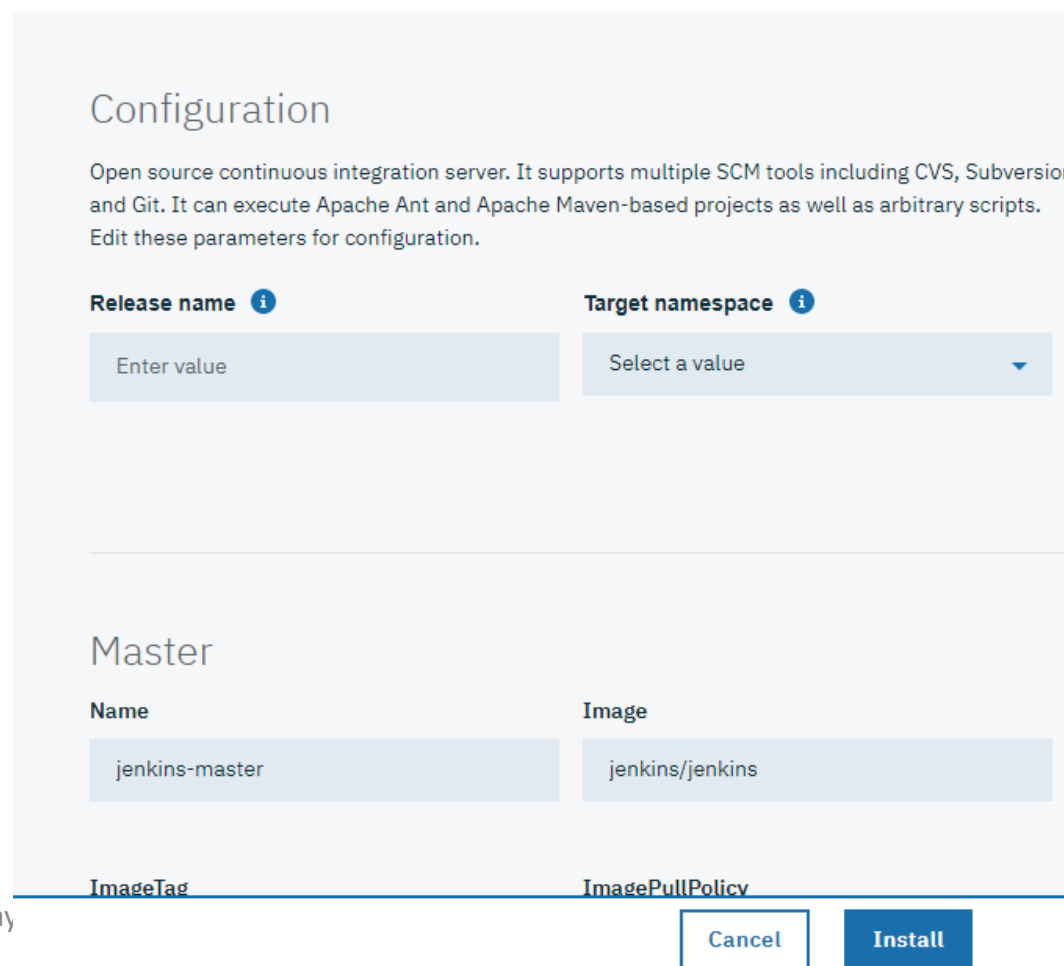
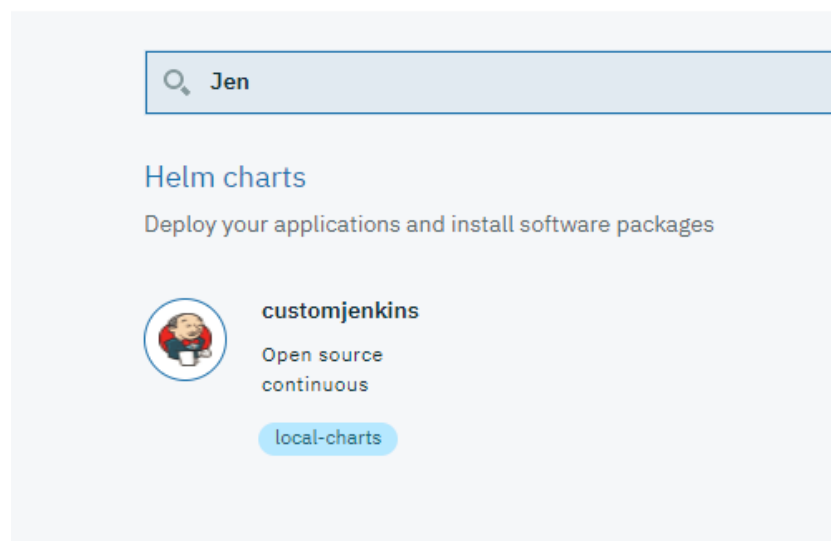
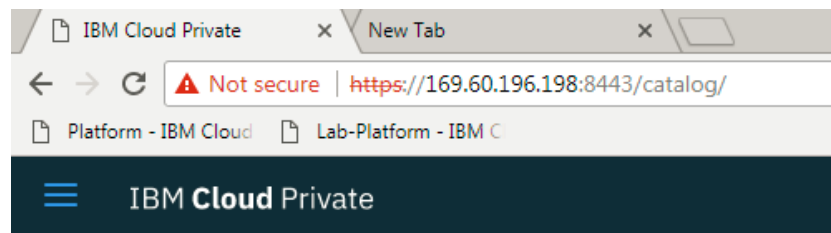
Thank You

BACKUP

Customizing an HELM Chart

- Clone from
 - <https://github.com/kubernetes/charts/tree/master/stable/jenkins> or
 - `git clone https://github.com/ibm-cloud-architecture/icp-openldap.git`
- Edit the files (charts.yaml, values.yaml etc..)
- Check with `helm lint --strict Jenkins`
- Package using `helm package customjenkins`
- Login to ICP - `bx pr login -a https://9.199.144.37:8443 --skip-ssl-validation`
- Upload to ICP using `bx pr load-helm-chart --archive customjenkins-0.13.1.tgz`
- In ICP UI Sync Repositories

Deploying Jenkins HELM Chart



Configure Jenkins

Jenkins » configuration

Kubernetes

Name:

Kubernetes URL:

Kubernetes server certificate key:

Disable https certificate check: ☐

Kubernetes Namespace:

Credentials:

Jenkins URL:

Jenkins tunnel:

Connection Timeout:

Read Timeout:

Container Cap:

Max connections to Kubernetes API:

- Find the admin password
- Login to Jenkins
- Configure Proxy (Optional)
- Install Plugins
- Configure Plugins

Kubernetes Pod Template

Name:

Namespace:

Labels:

Usage:

The name of the pod template to inherit from:

Containers

Container Template

Name:

Docker image:

Always pull image: ☐

Working directory:

Command to run:

Arguments to pass to the command:

Allocate pseudo-TTY: ☒

Docker Image [cloudbees/jnlp-slave-with-java-build-tools-dockerfile](https://cloudbees.github.io/jenkins-slave-with-java-build-tools-dockerfile/)

Configure and Run a build

General Source Code Management Build Tr

Post-build Actions

Project name

Description

[Plain text] [Preview](#)

- ☐ Discard old builds
- ☐ GitHub project
- ☐ This project is parameterized
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary
- ☒ Restrict where this project can be run

Label Expression

[Label autoscale](#) is serviced by n

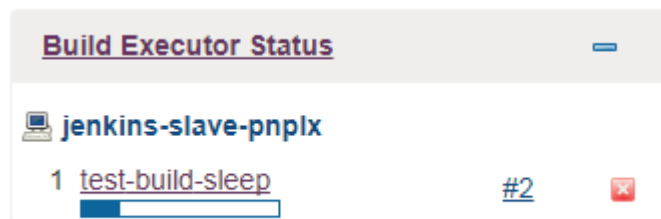
Build

Execute shell

Command

[See the list of available environment variables](#)

Jenkins Build Running on Slave



```
root@shaj-master:~# kubectl get po
NAME                                READY    STATUS    RESTARTS   AGE
customjenkins-customjenkins-66468d6b-zsc95  1/1      Running   0           1d
root@shaj-master:~# kubectl get po
NAME                                READY    STATUS    RESTARTS   AGE
customjenkins-customjenkins-66468d6b-zsc95  1/1      Running   0           1d
jenkins-slave-pnplx                      2/2      Running   0           9s
root@shaj-master:~# kubectl get po
NAME                                READY    STATUS    RESTARTS   AGE
customjenkins-customjenkins-66468d6b-zsc95  1/1      Running   0           1d
jenkins-slave-pnplx                      2/2      Terminating   0           44s
```