



MONOLITHS TO MICROSERVICES: APP TRANSFORMATION

Hands-on Technical Workshop

Thomas Qvarnström
Sr. Technical Marketing
Manager
Middleware BU

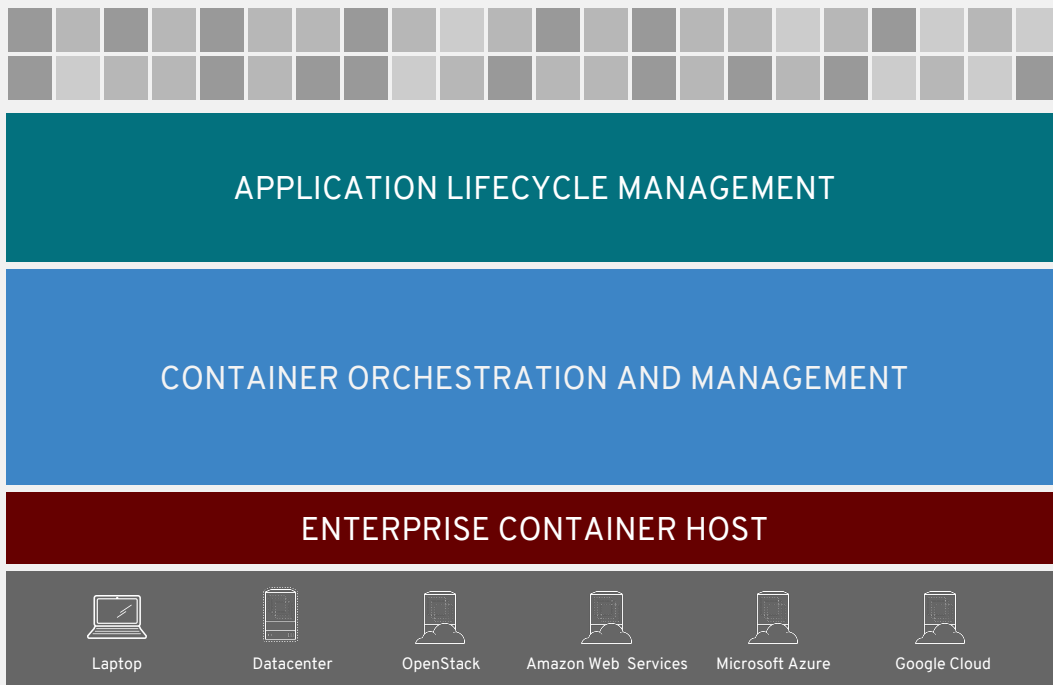
James Falkner
Sr. Technical Marketing
Manager
Middleware BU

A DEVELOPER INTRODUCTION TO OPENSHIFT

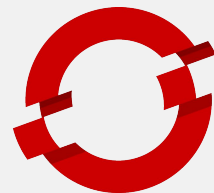


A secure and enterprise-grade container application platform based on Kubernetes for traditional and cloud-native applications

CLOUD-NATIVE CAPABILITIES WITH RED HAT OPENSIFT



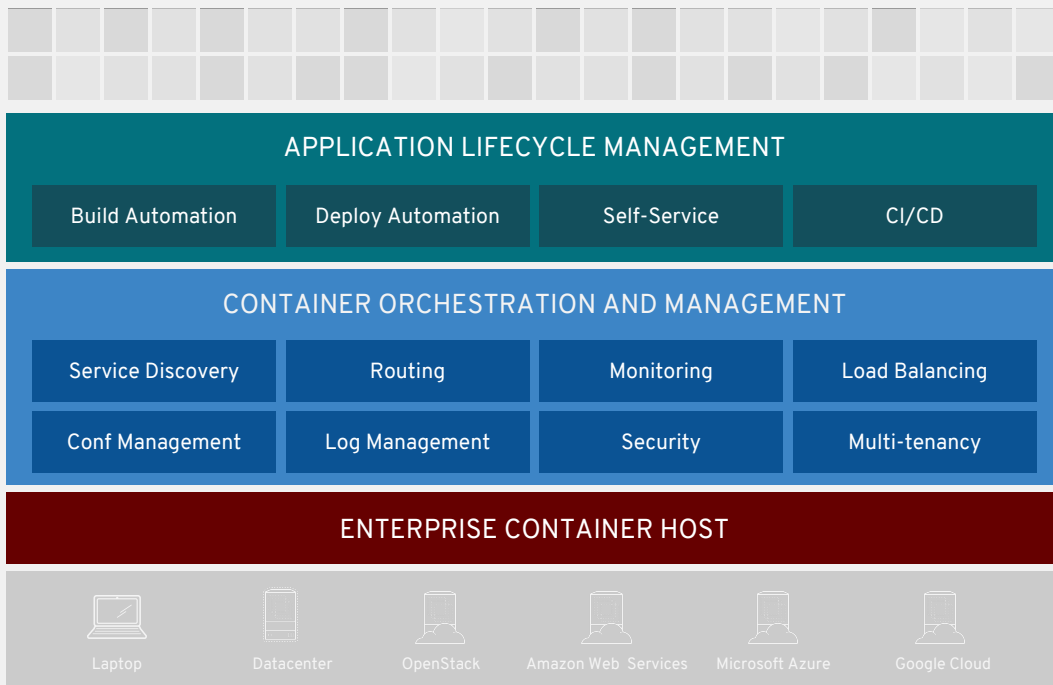
ANY
CONTAINER



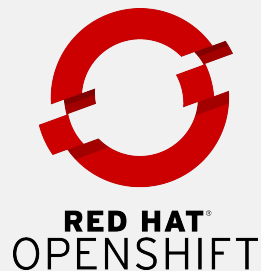
RED HAT
OPENSIFT

ANY
INFRASTRUCTURE

CLOUD-NATIVE CAPABILITIES WITH RED HAT OPENSIFT



ANY
CONTAINER



ANY
INFRASTRUCTURE

A container is the smallest compute unit

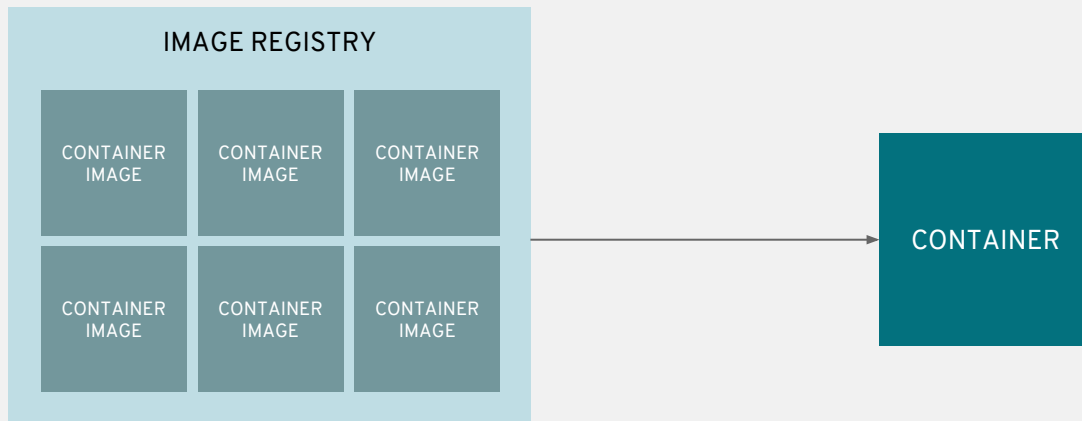


CONTAINER

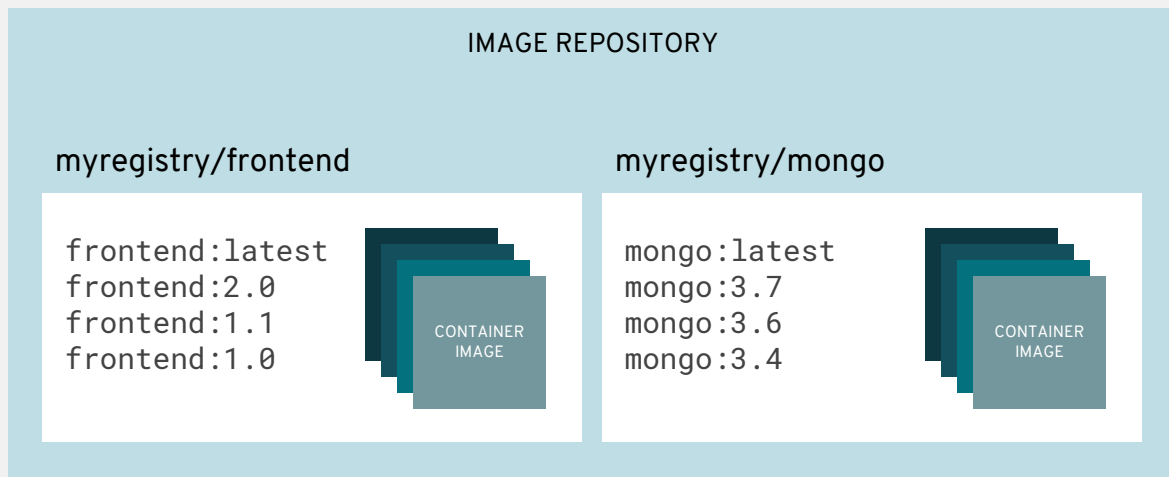
containers are created from
container images during a build



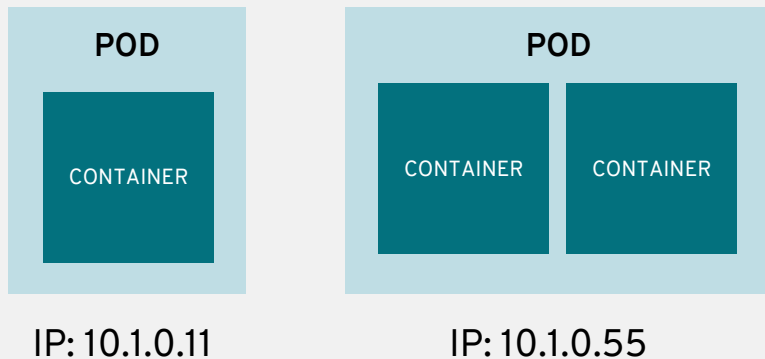
container images are stored in
an image registry



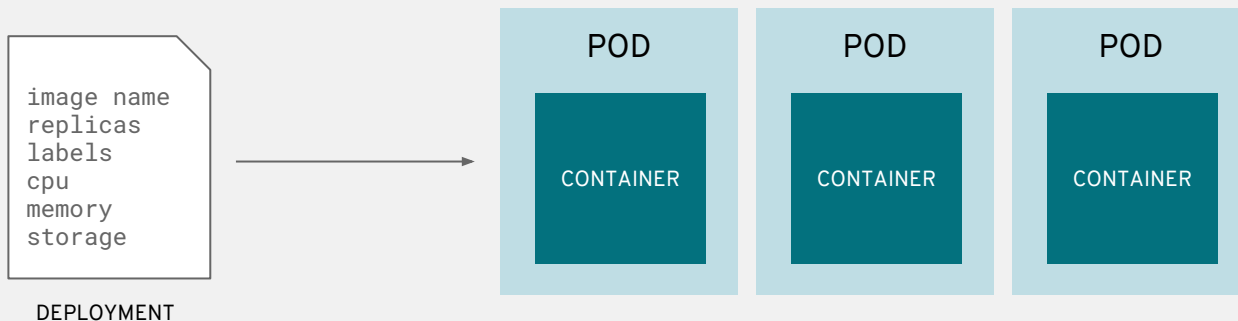
an image repository contains all versions of an image in the image registry



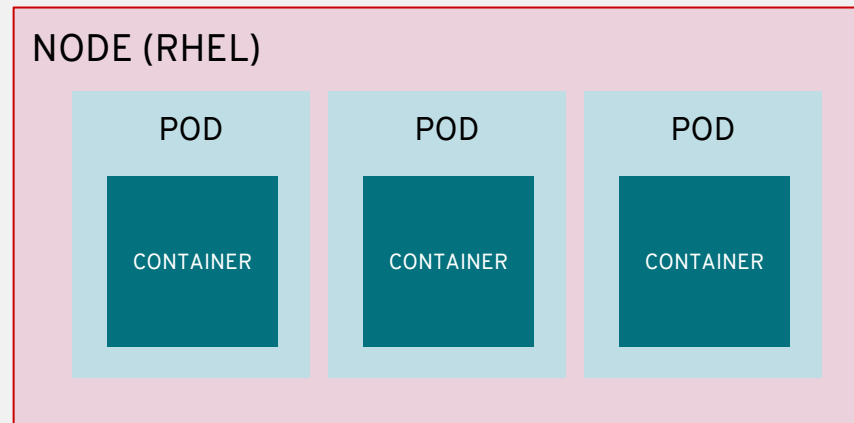
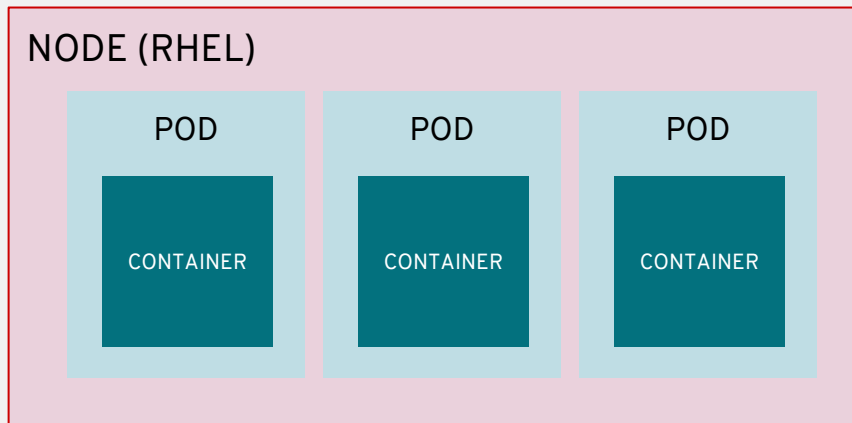
containers are wrapped in pods which are units of deployment and management, and share a common network address



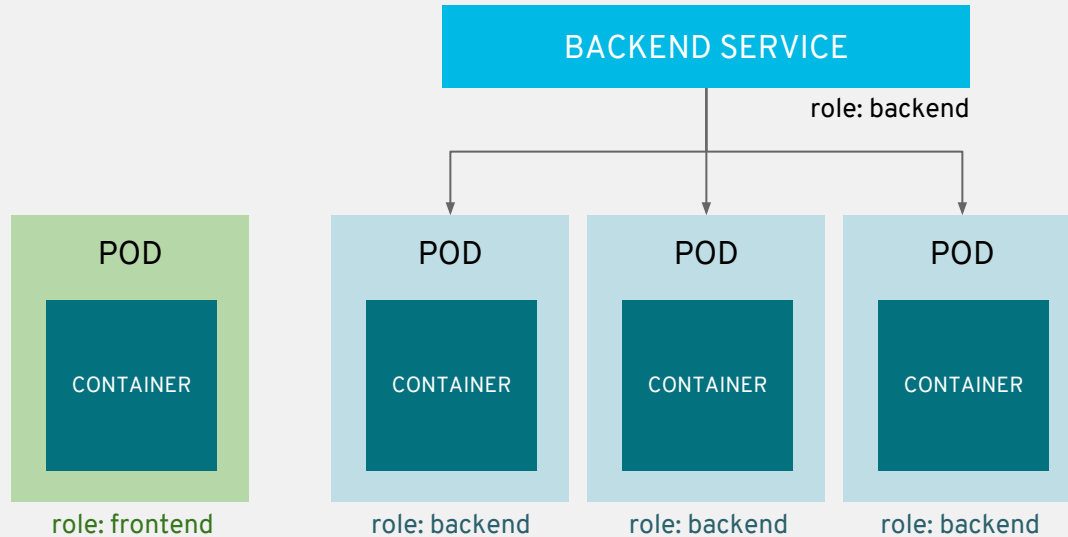
Pods configuration is defined in a deployment



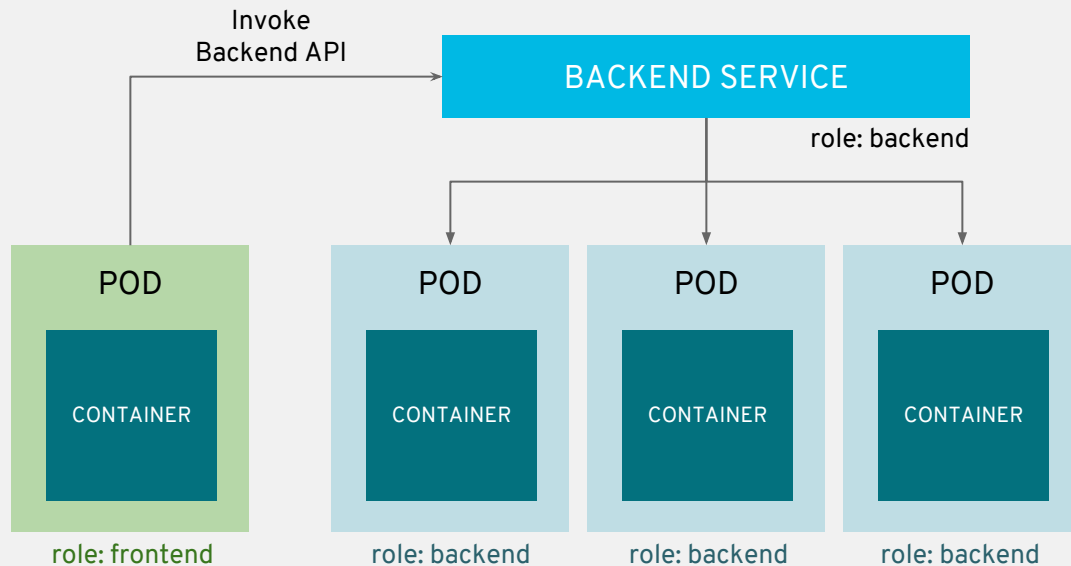
pods are deployed to and run on nodes



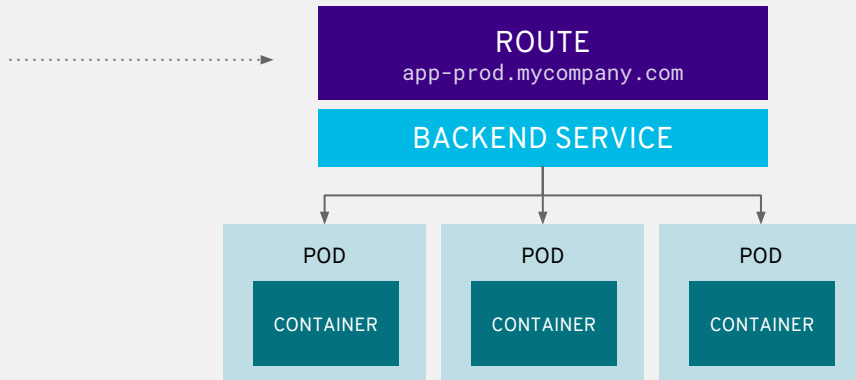
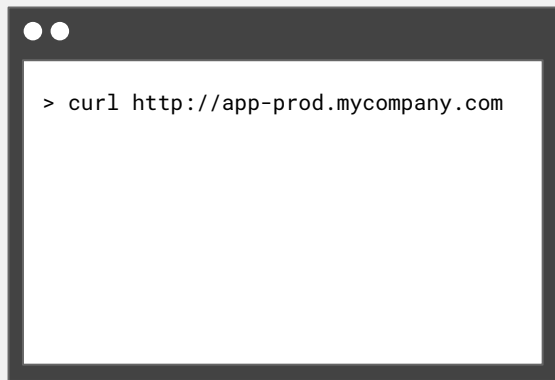
services provide internal load-balancing and service discovery across pods



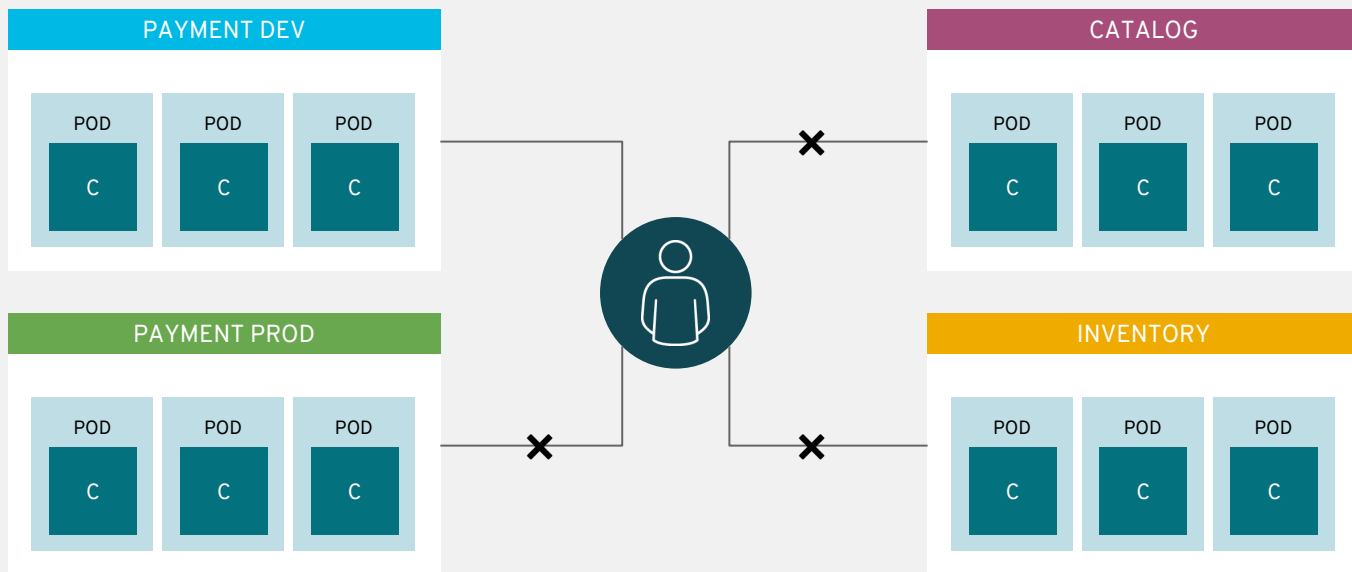
apps can talk to each other via services



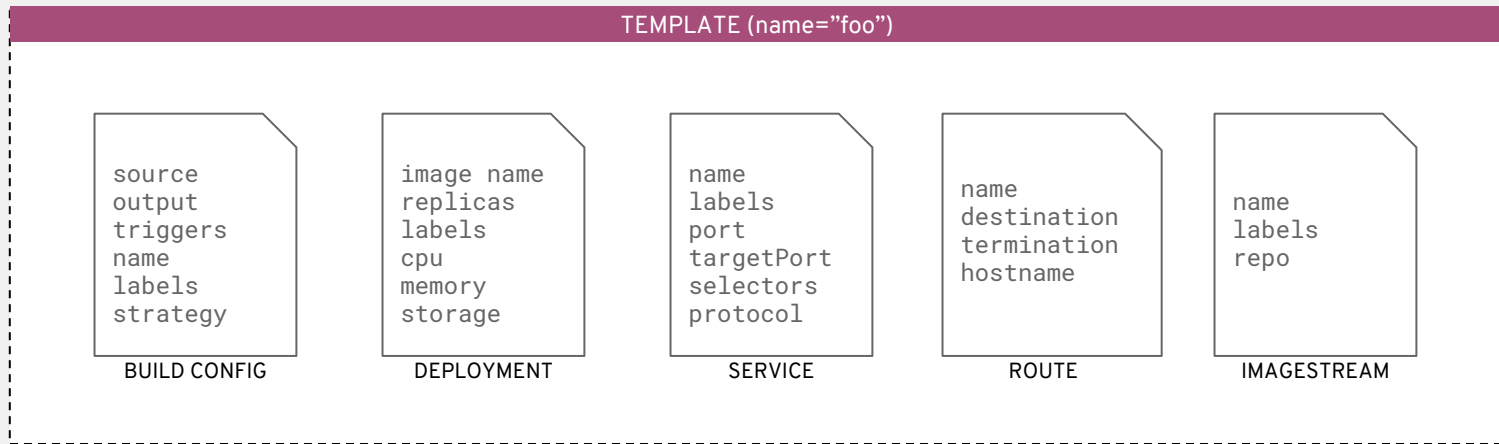
routes add services to the external load-balancer and provide readable urls for the app



projects isolate apps across environments,
teams, groups and departments



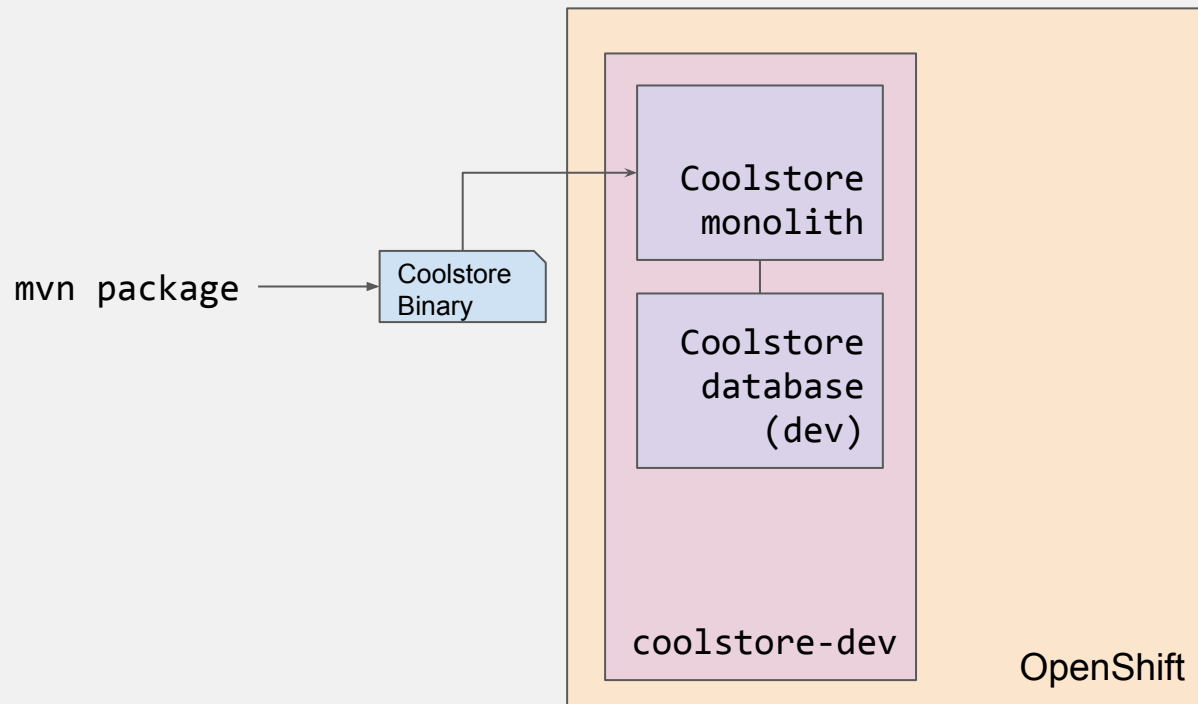
templates define a blueprint for an application that can be instantiated within a project



\$ oc new-app foo

LAB: DEVELOPER INTRODUCTION TO OPENSHIFT

CURRENT STATE



GOAL FOR LAB

In this lab you will learn:

- Important OpenShift concepts for developers
- How OpenShift makes developers and architects happier
- How to do efficient round-trip development:
 - Separate **dev** from **prod** environments
 - Quick deployments using **rsync** / port-forwarding
 - Promoting apps using **CI/CD Pipelines**

LAB: DEVELOPER INTRO TO OPENSIFT

A man with wild, light-colored hair, wearing a white lab coat over a plaid shirt, and large green safety goggles. He is holding two metal tools, possibly pliers or wire cutters, one in each hand, and looking intently at them. He is wearing a watch on his left wrist. The background is a workshop or lab with various equipment and a blue poster on the wall.

SCENARIO 3

A DEVELOPER INTRODUCTION TO OPENSIFT

WRAP-UP AND DISCUSSION

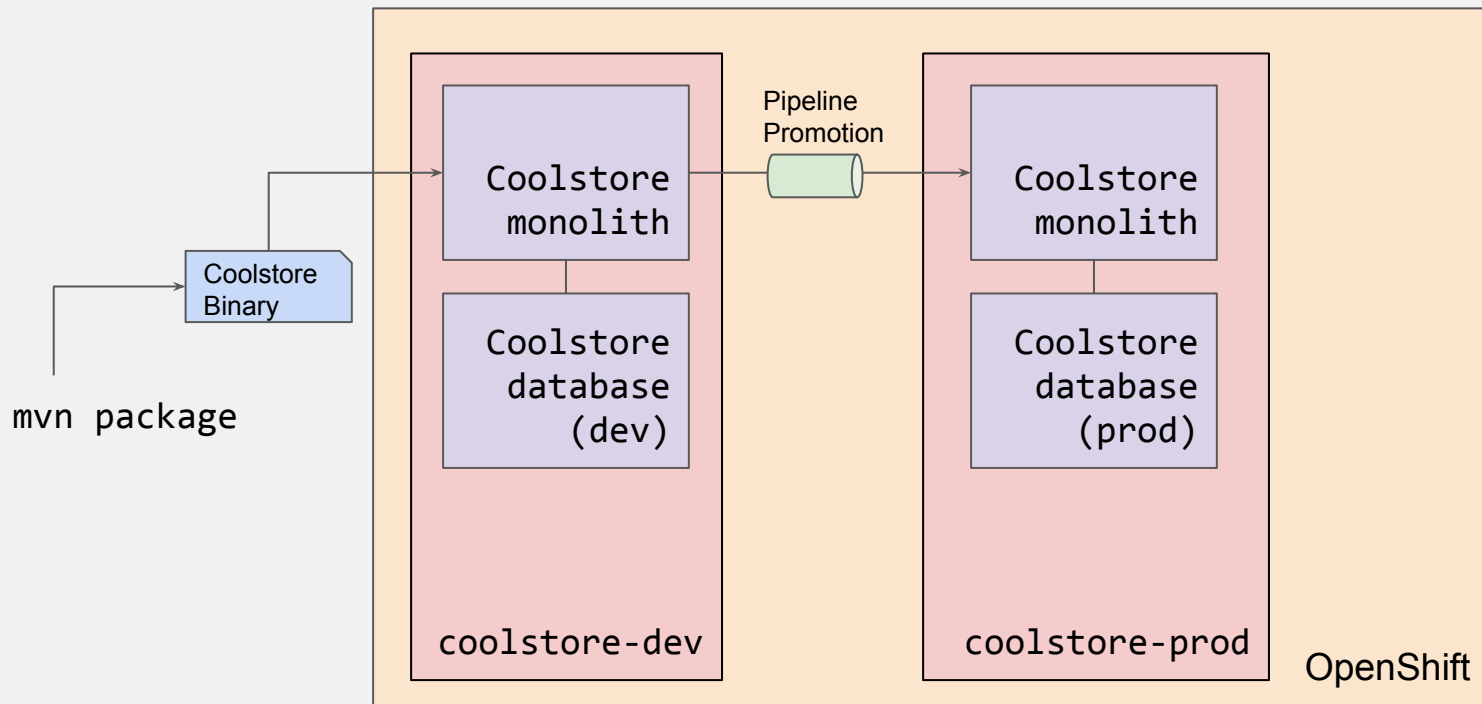
RESULT OF LAB

In this lab you learned how to:

- Do quick deployments with `oc rsync`
- Create a production environment separate from dev
- Promote tested/verified builds between environments using OpenShift pipeline builds

You should now have two projects (dev and prod) running the same CoolStore app! In the next lab we will begin the process of breaking the monolith up into microservices.

DESIRED RESULT OF SCENARIO 3



LEARN MORE: learn.openshift.com



Interactive Learning Portal

Our Interactive Learning Scenarios provide you with a pre-configured OpenShift instance, accessible from your browser without any downloads or configuration. Use it to experiment, learn OpenShift and see how we can help solve real-world problems.

Getting Started
with OpenShift for
Developers

START SCENARIO

Logging in to an
OpenShift Cluster

START SCENARIO

Deploying
Applications From
Images

START SCENARIO

Deploying
Applications From
Source

START SCENARIO

Using the CLI to
Manage Resource
Objects

START SCENARIO

Connecting to a
Database Using
Port Forwarding

START SCENARIO

Transferring Files
in and out of
Containers

START SCENARIO



THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos