



UniversaMente
una comunità per gli Atenei

UNIVERSAMENTE

CONTRATTO CRUI - RED HAT

PROFILO BLU


plurimedia

 **Red Hat**

 **CRUI**
Conferenza dei Rettori
delle Università Italiane

1. Il contratto CRUI - RED HAT

2. Agile, CI/CD

- a. Metodologia Agile
- b. Continuous Delivery/Continuous Integration
- c. OpenShift

3. Domande e risposte



Cosa contiene il contratto CRUI - RED HAT?

Sono presenti due package Red Hat: **Campus 1** e **Campus 2**

Oltre ad essi è possibile acquistare: **Ansible, RHEL add-on, TU** ed altro

Campus 1 contiene:

- Red Hat Enterprise Linux,
- Red Hat Enterprise Linux Workstation,
- Red Hat Enterprise Linux Desktop,
- Red Hat x86_64 Linux Server,
- Red Hat Linux HPC,
- Red Hat Linux for SAP Applications,
- Red Hat Linux Power Big Endian,
- Red hat Software Collections,
- Red Hat Developer Toolset,
- Red Hat Linux Atomic Host.

Campus 2 contiene:

- La distribuzione CAMPUS 1,
- Red Hat Enterprise Virtualization (RHEV),
- Red Hat OpenStack,
- Red Hat Gluster Storage,
- Red Hat Insight.

Quali i vantaggi per l'Ateneo o il Centro di ricerca?



Non ci sono limiti nel numero di sistemi Red Hat gestibili.

L'Ateneo può distribuire le sottoscrizioni Red Hat anche **a tutti gli studenti.**

Le distribuzioni acquisite tramite l'accordo CRUI - RED HAT possono essere utilizzate per fini **formativi e/o gestionali**, indifferentemente.

Per aderire al contratto CRUI - RED HAT

Iscrivere all'applicazione Plurimedia:
universamente.plurimedia.it

L'adesione al contratto CRUI - RED HAT da parte di un Ateneo può coincidere con **l'intero valore in FTE** dell'Ateneo stesso.

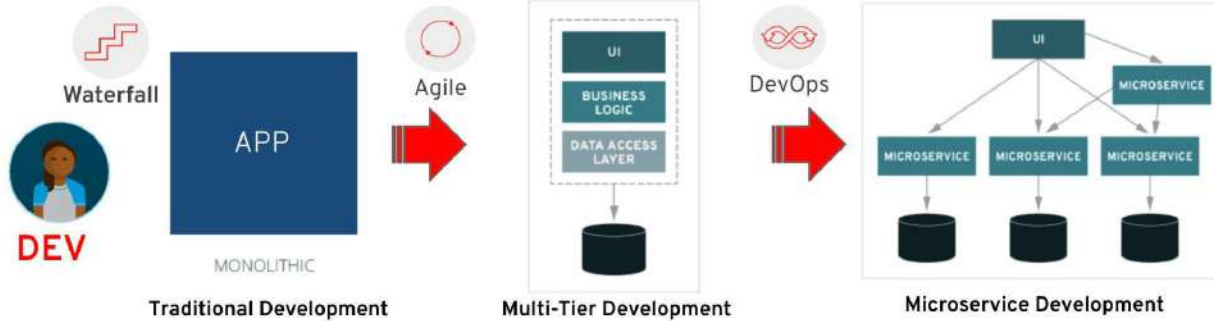
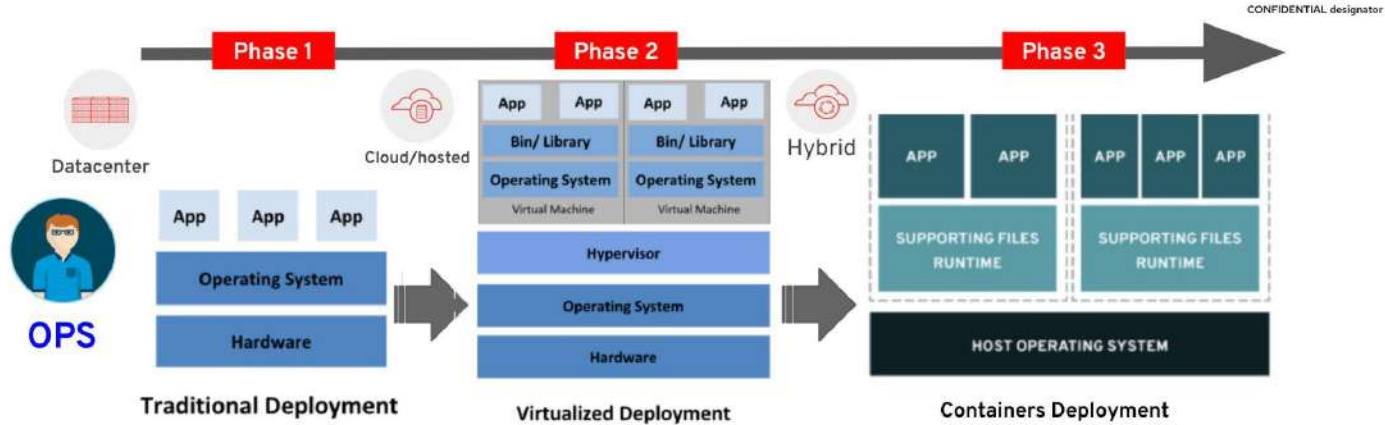
L'adesione al contratto CRUI - RED HAT può avvenire utilizzando la **soglia minima di 500 FTE**, indipendentemente dal valore in FTE complessivo dell'Ateneo.

Agile, CI/CD usando la tecnologia Red Hat

Questo seminario si pone l'obiettivo di affrontare e descrivere le nuove modalità di programmazione derivanti dall'uso spinto di piattaforme cloud enabled. Si discuterà di come diventi possibile e quali sono i prerequisiti per l'implementazione di una piattaforma di continuous delivery. Si discuterà della metodologia Agile e della piattaforma PaaS Openshift.

Giuseppe Romano
Red Hat Solution Architect

The new Agile way

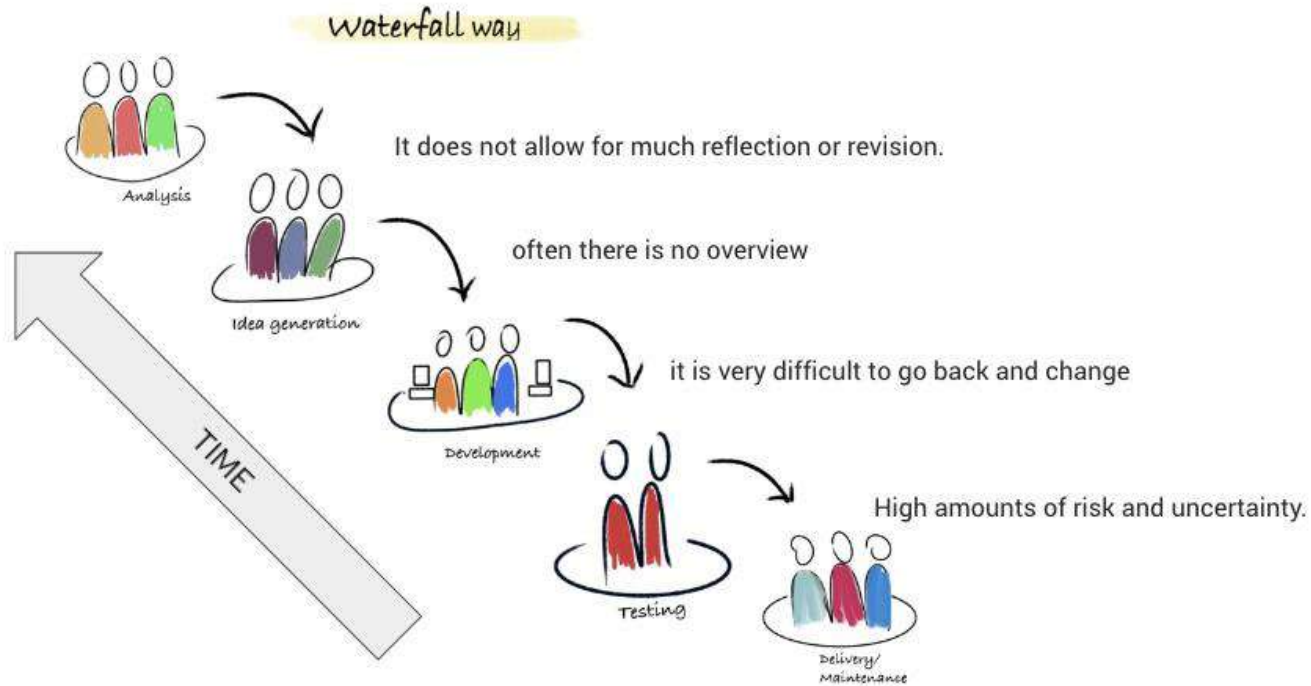




In waterfall methodology the biggest friction was the transition from the development team to the operations team.

One typical problem was: a software that was working correctly in the developer's environment was failing to start in the production server.

Why?





AGILE and Dev(*)Ops Methodology

DevOps is a methodology that collapse **development** team and **operations** team in a single entity.

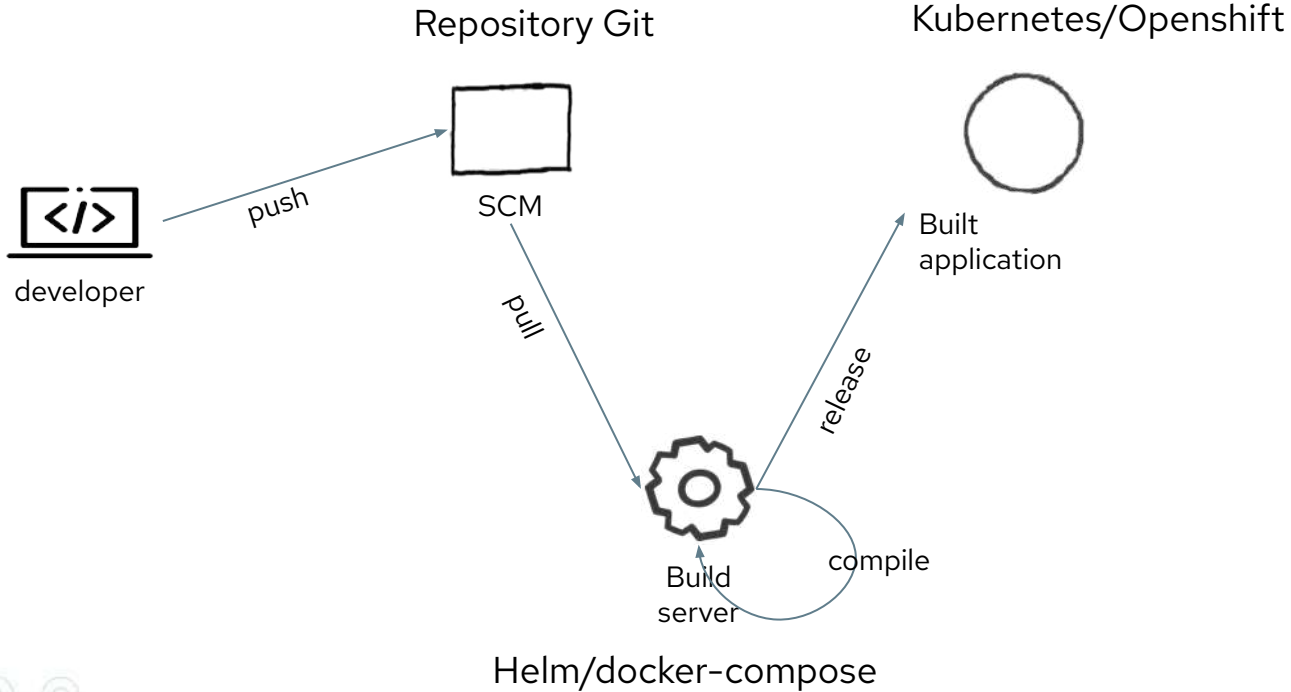
They need to collaborate during the whole software development life cycle instead only at the end of the development phase.

DevOps makes intensive use of **automation**. Automation allows to achieve speed, efficiency and reduce manual errors.



- Manage Change More Effectively
- Improved Customer Engagement
- Focus on the Highest Priorities First
- Increased Productivity
- Product Owner Feedback
- A Highly Collaborative Environment

DevOps pipeline





DevOps and CI/CD are not enough

We could enhance the CI/CD to manage also the “promotion to production” phase of built artifacts with **Continuous Release** (or **Continuous Deployment**)

This is a very **critical** aspect that must take into consideration a lot of intermediate steps (provision a machine, install operating system and/or runtime environment, install/upgrade software, keep operating system/software patched, etc..)

This process alone doesn't scale well when we want to adopt the (hybrid) cloud and have legacy monolithic applications.



The rise of the Containers

Containers are an **emerging technology** that allows decoupling applications from the environment where they will be executed in.

They can be seen as black boxes that include everything needed to run an application: code, runtime, system tools, system libraries and settings.

At runtime, containers, will run completely isolated from each other and in a reproducible manner.

Deployment process is the same for all the containers regardless of the application,



Container advantages

A containerized application will have the exact **same behaviour** in a dedicated datacenter, on a public cloud, in the developer's machine or on an embedded device.

Containers allows **developers** to concentrate on **business logic** while **operations** can concentrate on **deployment and management** without knowing the container's content or the application behaviour.

Containers are **more lightweight** than Virtual Machines, are **faster** and consumes **less resources**.

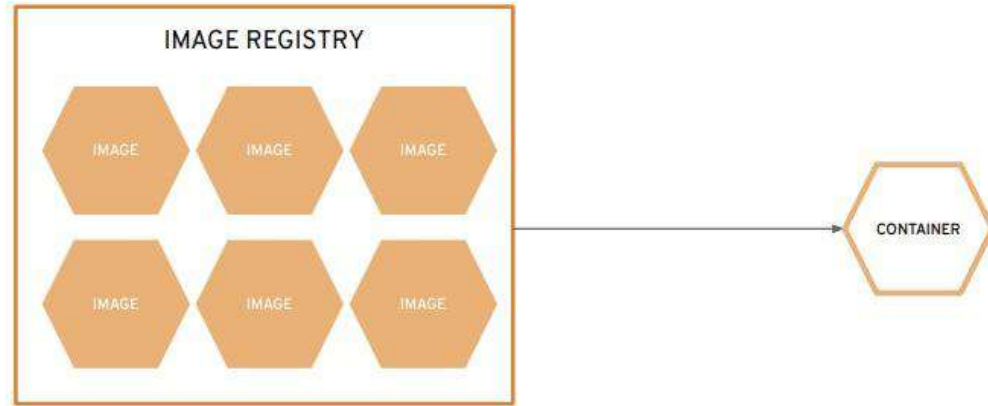
A container is the smallest compute unit



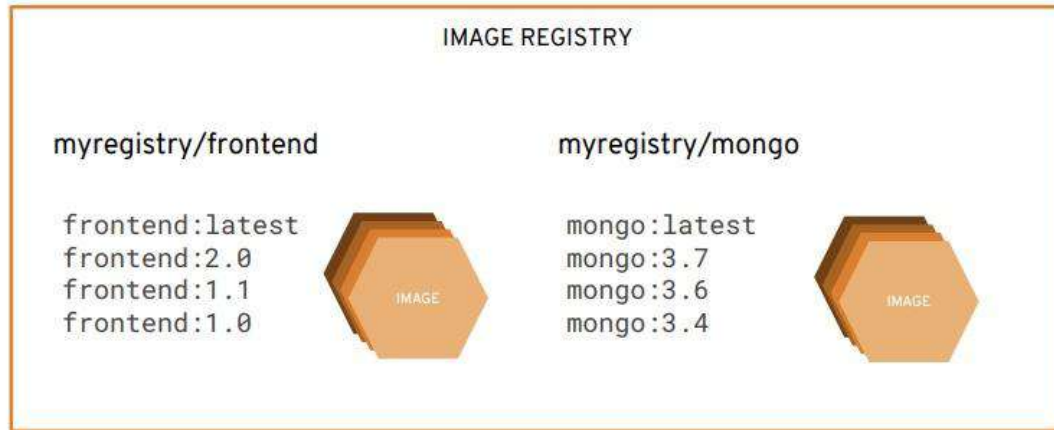
Containers are created from container images



Container images are stored in an image registry



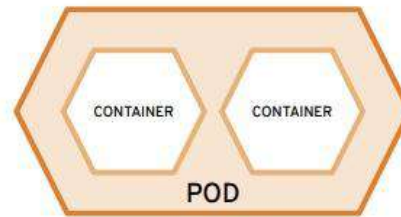
An image repository contains all versions of an image in the image registry



Containers are wrapped in pods which are units of deployment and management



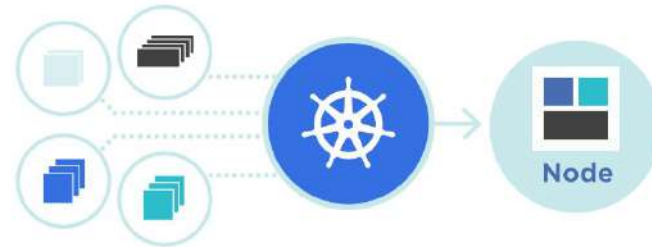
10.140.4.44



10.15.6.55

What Kubernetes is ?

Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications.

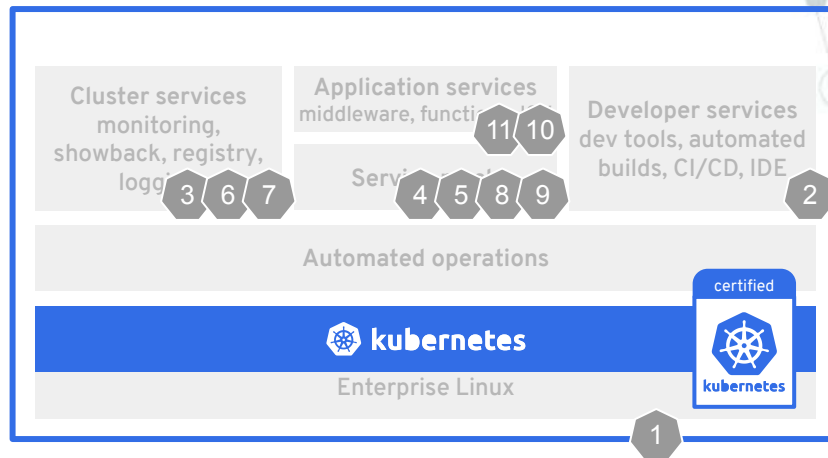


What's needed to put Kubernetes into production?

kubernetes

Lacks many essential components

1. Operating system
2. Container runtime (CRI-O, Containerd, Docker, etc).
3. Image registry
4. Software-defined networking
5. Load-balancer and routing
6. Log management
7. Container metrics and monitoring
8. DNS
9. Ingress
10. RBAC



The customer (or third-party) must configure, integrate, operate and support additional components to be fully operational.

Application Definition & Development

Databases	Data Warehouse	Streaming	Languages & Frameworks	SCM	Registry Services	Application Definition	CI / CD	Services as Code	API management

Orchestration & Management

Scheduling & Orchestration	Coordination & Service Discovery	Service Management

Runtime

OS	Cloud-Native Storage	Container Runtime	Cloud-Native Network

Provisioning

Infrastructure Automation	Host Management / Tooling	Secure Images

Infrastructure

Platforms

Paas / Container Service

Observability & Analysis

Monitoring

Logging

Event-based compute

Tracing

CNCF Projects

github.com/cncf/landscape

With OpenShift you decide the best place to run your applications

Traditional apps



Cloud-native apps



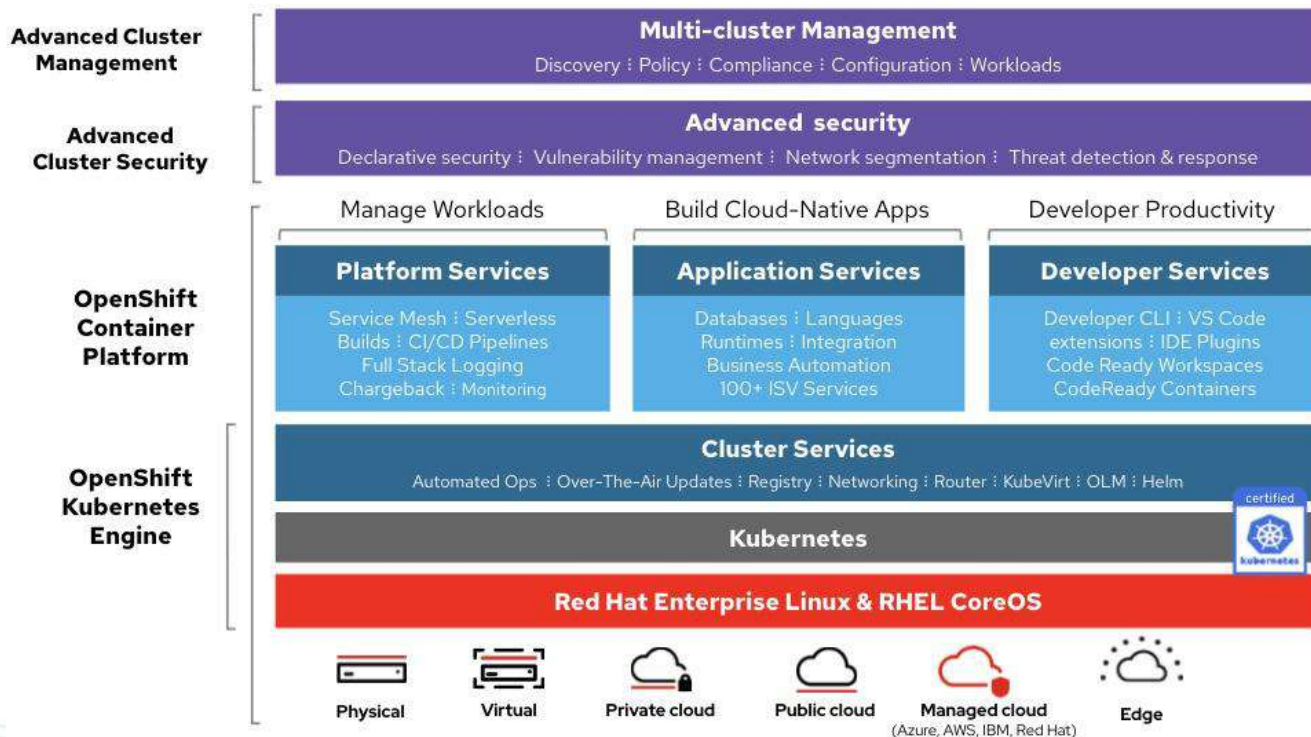
AI/ML, Functions...



Virtual Machines



OpenShift 4 - A smarter Kubernetes platform



CI/CD Pipelines

The goal of the continuous integration and continuous delivery (CI/CD) pipeline is to enable teams to release a constant flow of software updates into production to quicken release cycles, lower costs, and reduce the risks associated with development.



- The “CI” in CI/CD always refers to continuous integration, which is an automation process for developers.
- Continuous delivery means a developer’s changes to an application are automatically tested and uploaded to a repository (like a container registry), where they can then be deployed to a live environment
- Continuous deployment refer to automatically releasing a developer’s changes from the repository to a live environment

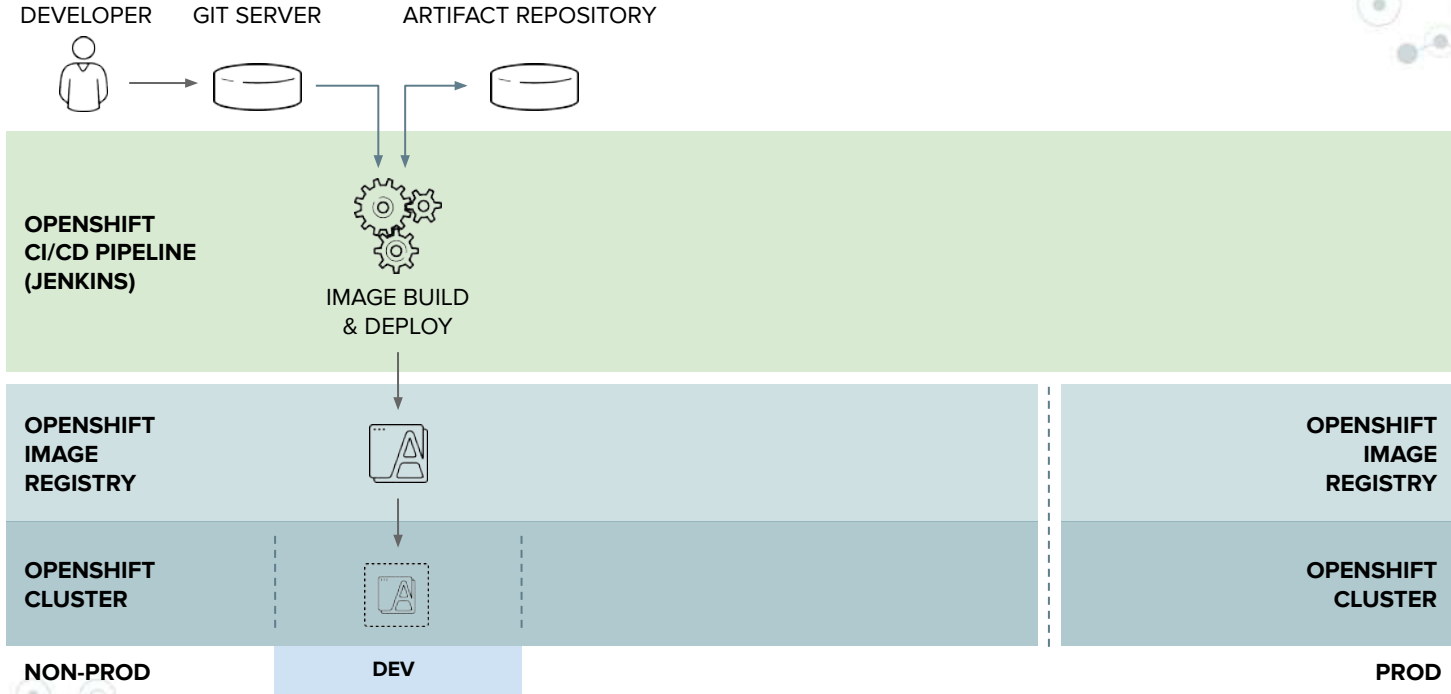
CI/CD Pipelines

Types of Pipeline Stages

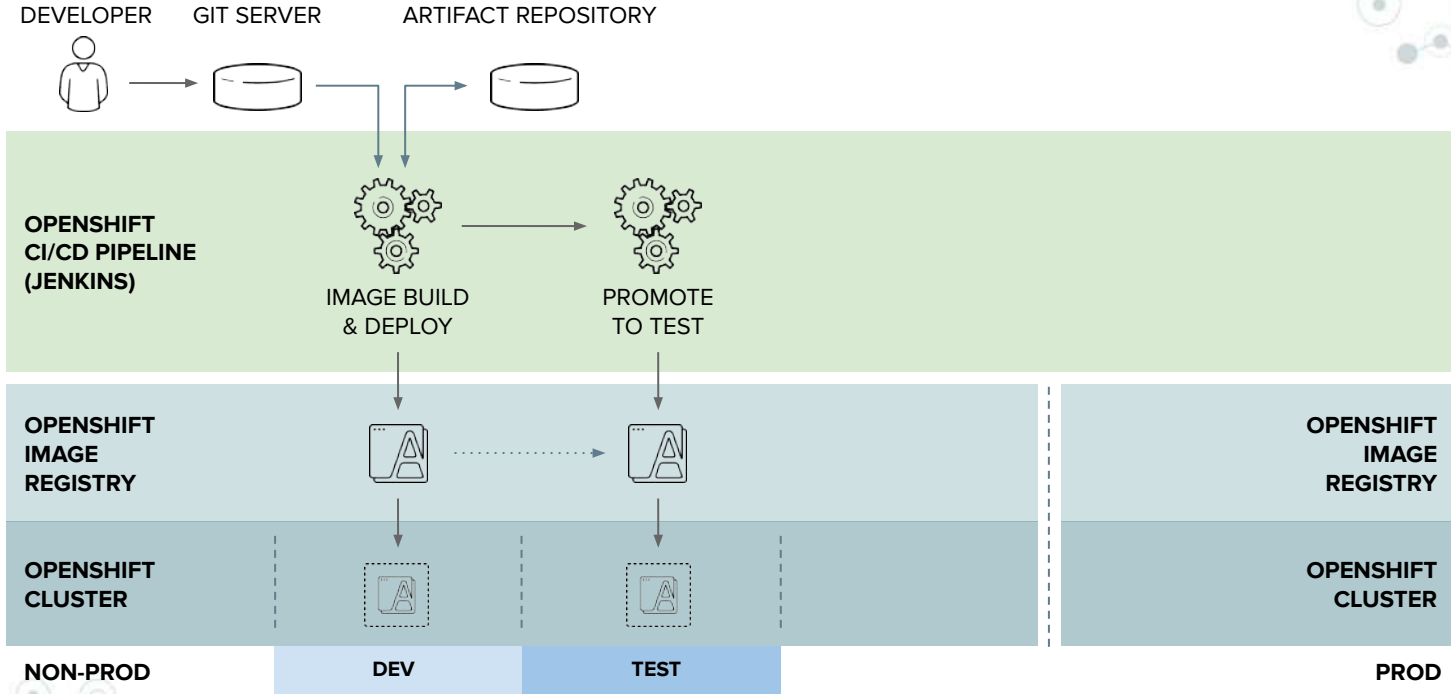
Typically, there are four stage in a pipeline:

Stage Type	Description
Build	Creates or builds artifacts used by other steps.
Promotion	Deploys artifacts created in a build step to a specified environment.
Validation	Validates actions performed in a previous step, typically build or deploy. If the validation step fails, the pipeline process might be aborted.
Compliance	These type of steps are sometimes needed to make the pipeline compliant with enterprise release management standards.

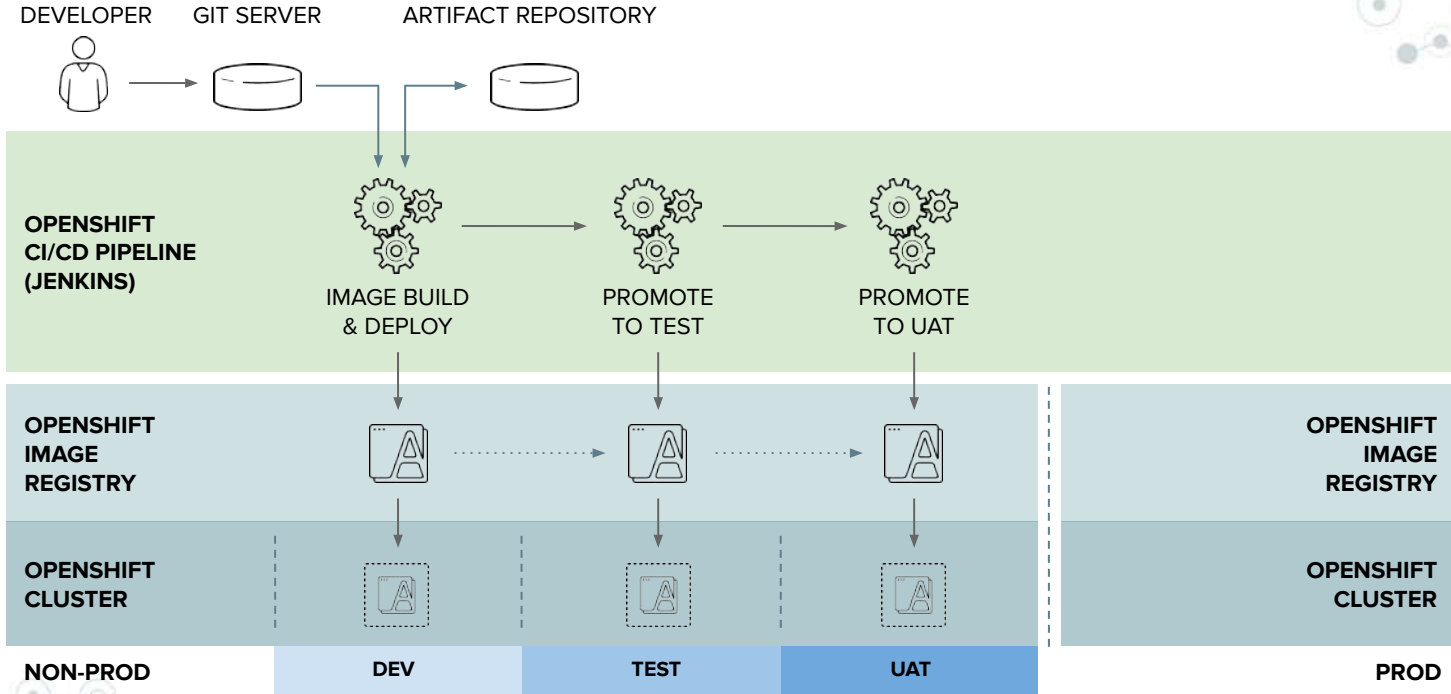
CI/CD Pipelines



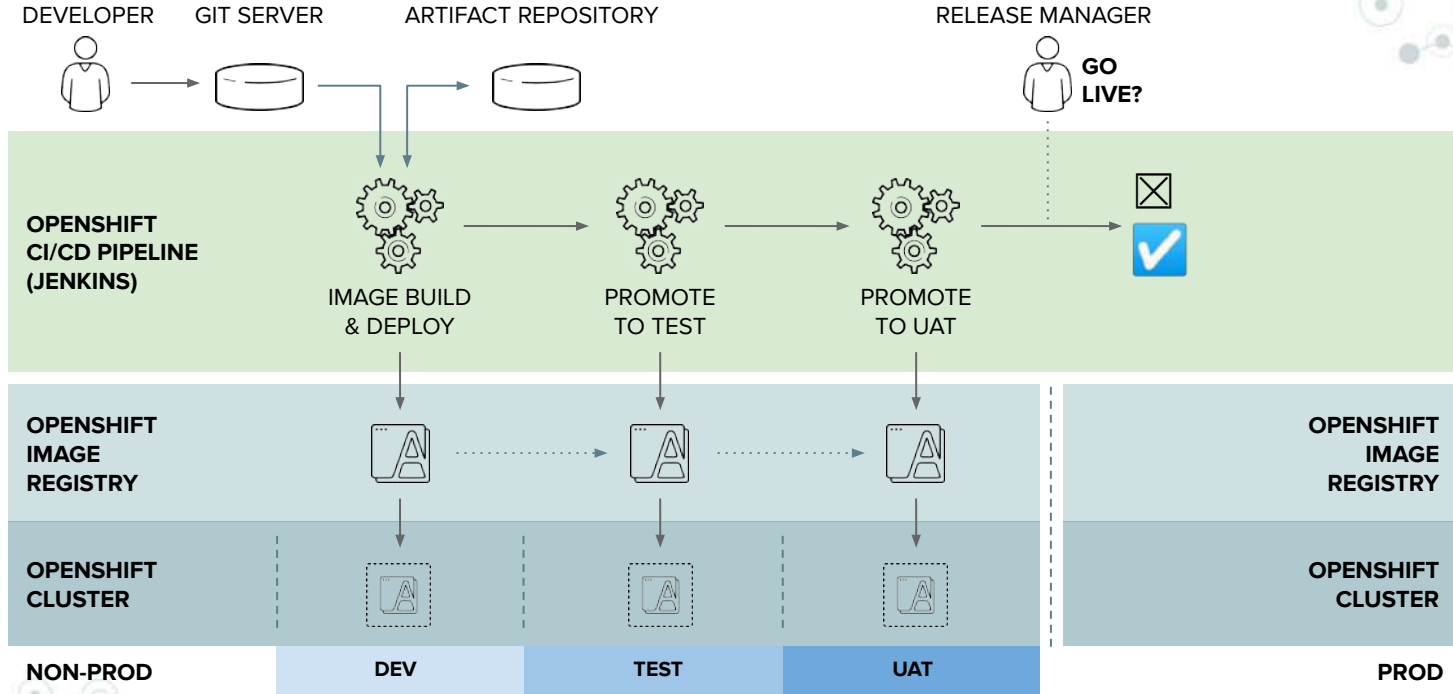
CI/CD Pipelines



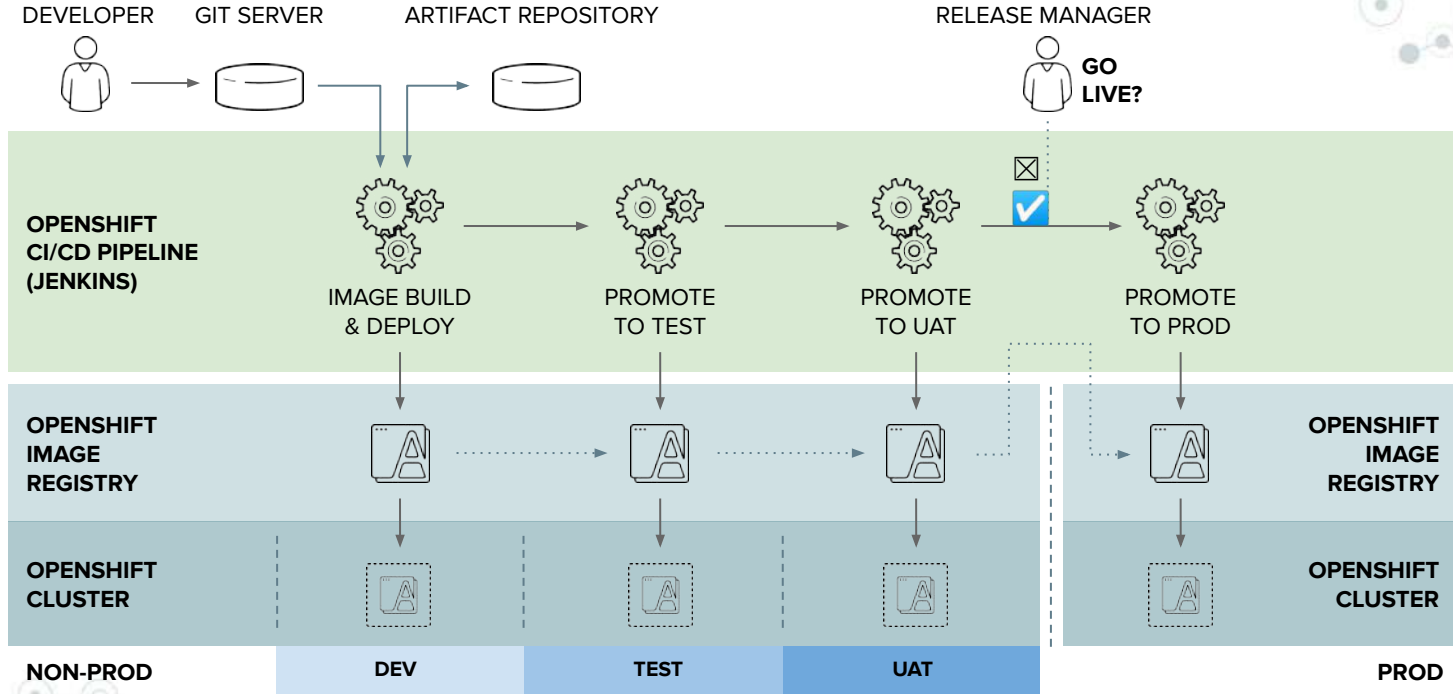
CI/CD Pipelines



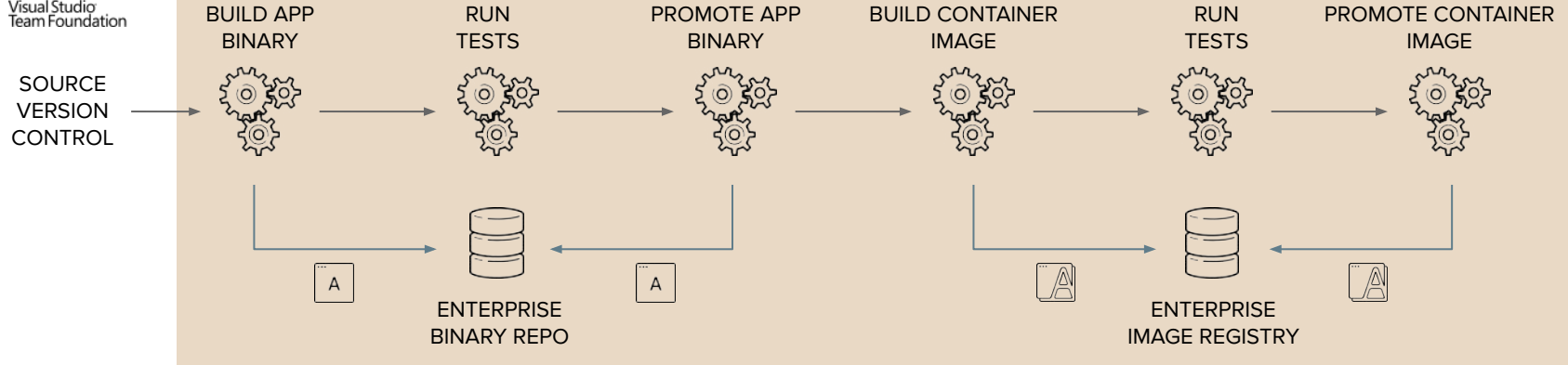
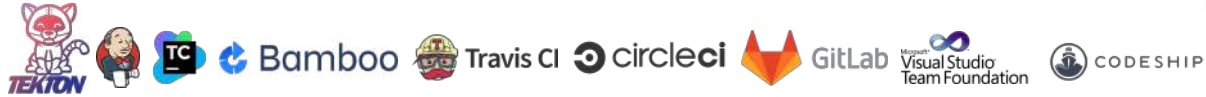
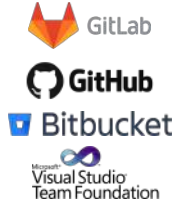
CI/CD Pipelines



CI/CD Pipelines



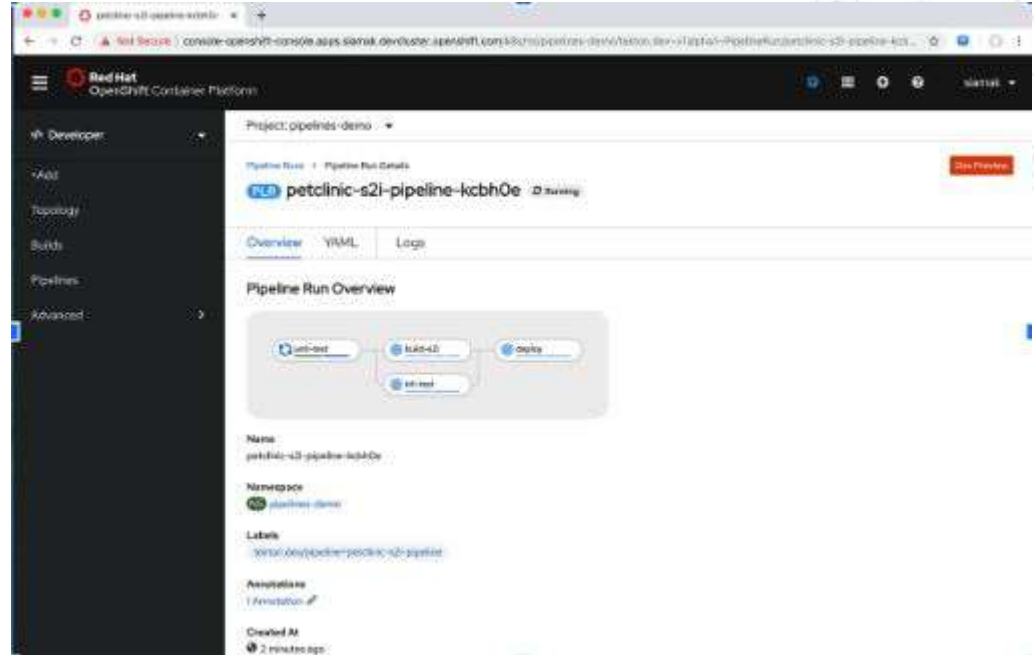
There is a full landscape of tools



OpenShift Pipelines



- OpenShift Pipelines 1.0 Tech Preview
- Authoring pipelines with Pipeline builder
- Share data between tasks in pipeline
 - Workspaces for sharing artifacts
 - Results for small data e.g. commit sha
- Webhooks
 - Default GitHub TriggerBinding
 - Filtering on payload (event, files, etc)
 - Add to payload with CEL expressions
 - Custom processing with interceptors
- Proxy support for git repositories



OpenShift - Pipeline Builder

Compose Pipelines from Task library

The screenshot displays the OpenShift Pipeline Builder interface. The top navigation bar shows the Red Hat logo and 'OpenShift Container Platform'. The left sidebar contains a menu with options: Developer, +Add, Topology, Monitoring, Builds, Pipelines (selected), More, Search, Health, Project Details, and Project Access. The main content area is titled 'Pipeline Builder' and shows a pipeline named 'demo-pipeline'. The 'Tasks' section contains a flow diagram with six tasks: 'codeps-01', 'codeps-02', 'codeps-03', 'k2d-nodeps', 'buildah', and 'openshift-client'. 'codeps-01' and 'codeps-02' are connected to 'codeps-03', which then connects to 'k2d-nodeps', 'buildah', and 'openshift-client'. Below the tasks, the 'Parameters' section indicates 'No parameters are associated with this pipeline' and includes an 'Add Parameters' button. The 'Resources' section also indicates 'No resources are associated with this pipeline' and includes an 'Add Resources' button.



“

Domande e risposte



“

**Qual è il vantaggio di uno sviluppo applicativo
Cloud Native?**



“

In quali casi bisognerebbe andare verso il cloud?



“

Quant'è usato Kubernetes?



“

Come ci si può avvicinare a questa tecnologia?

Grazie!

LINK UTILI

[universamente](#)

Il gruppo in linkedin

www.universamente.it

il portale della community

universamente.plurimedia.it

l'App di community

Maurizio Galotti

Email: maurizio.galotti@plurimedia.it

Giuseppe Romano

Email: gromano@redhat.com