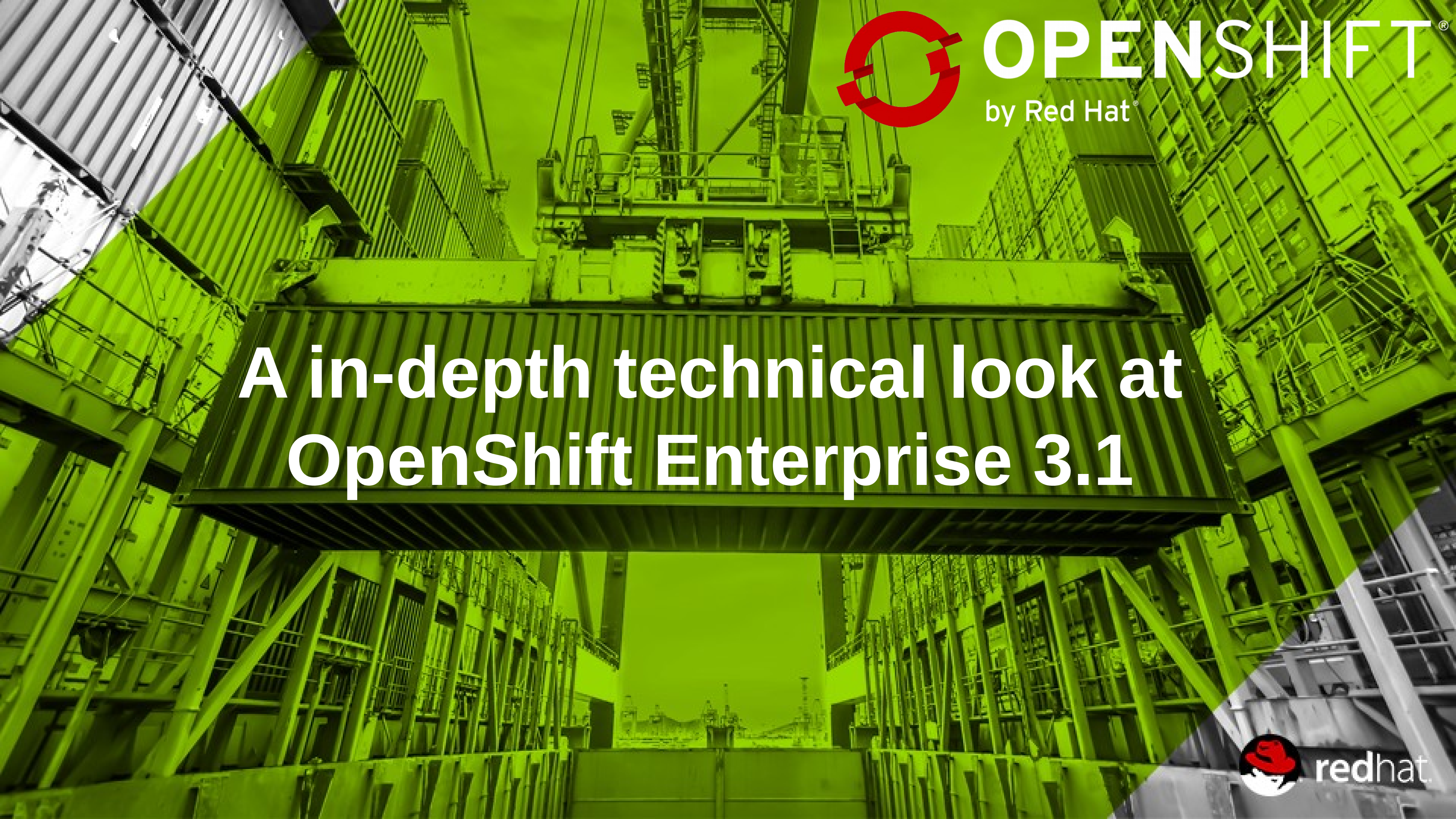
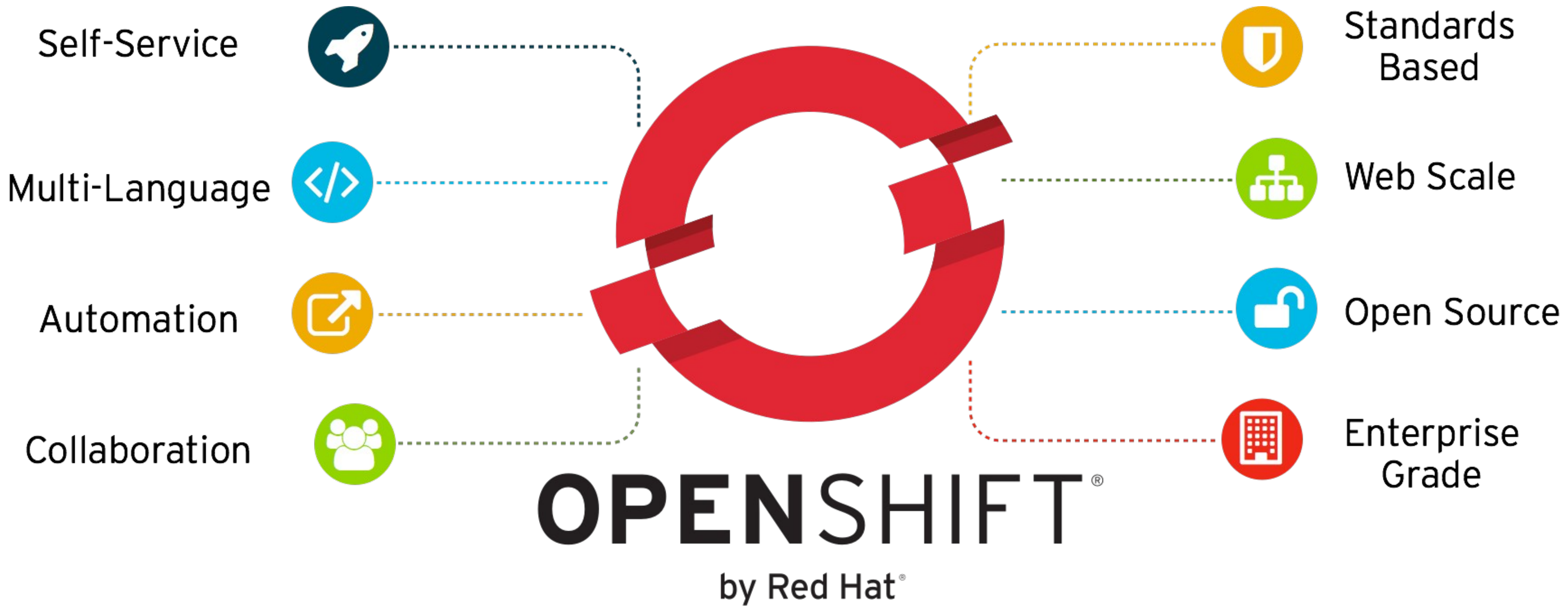
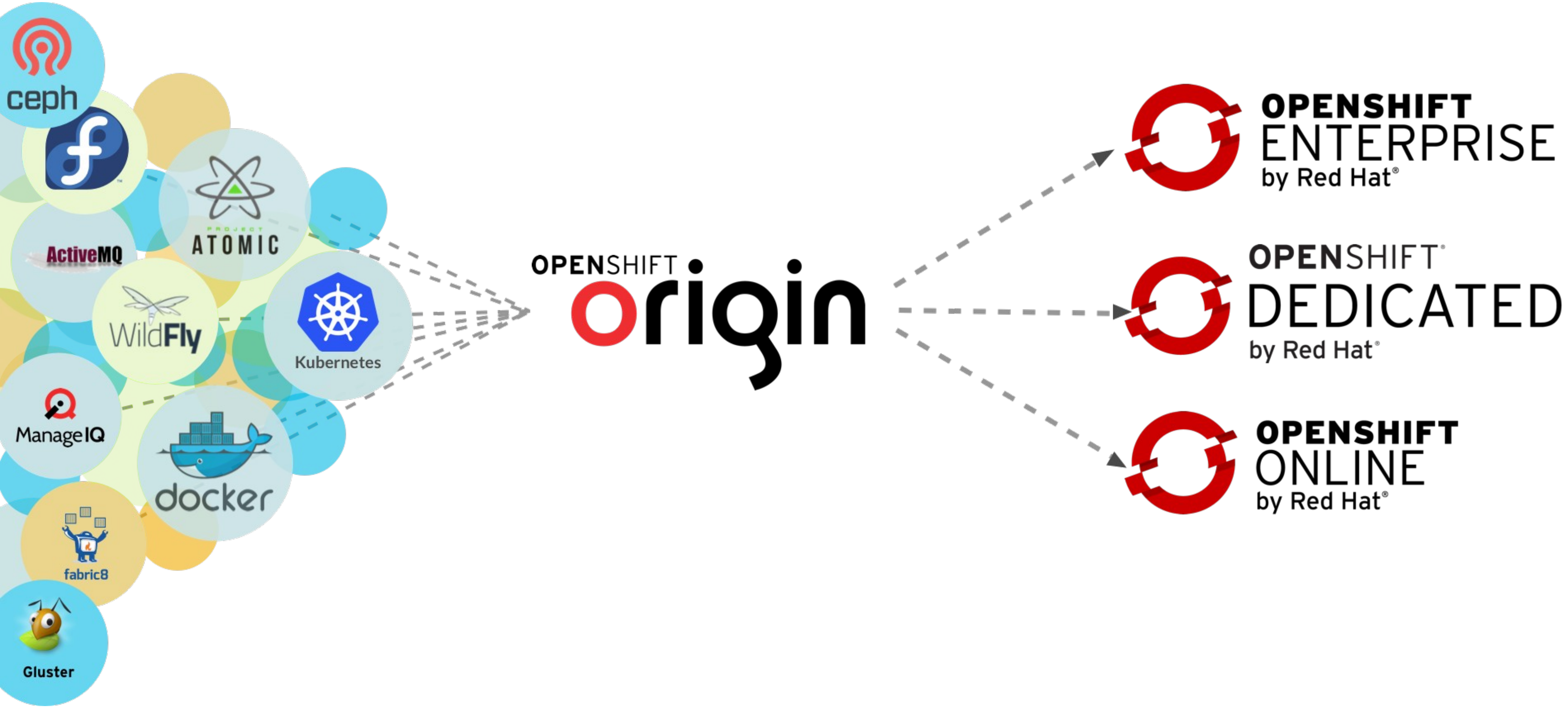


A in-depth technical look at OpenShift Enterprise 3.1

OpenShift Enables Both Dev and Ops



Community Powered Innovation


ceph
ATOMIC
ActiveMQ
WildFly
Kubernetes
ManageIQ
docker
fabric8
Gluster
OPENSHIFT origin
OPENSHIFT ENTERPRISE by Red Hat®
OPENSHIFT DEDICATED by Red Hat®
OPENSHIFT ONLINE by Red Hat®

Openshift 3.1 - Open Source Components

docker

kubernetes

etcd

Metrics

Installation

Logging with EFK

elastic

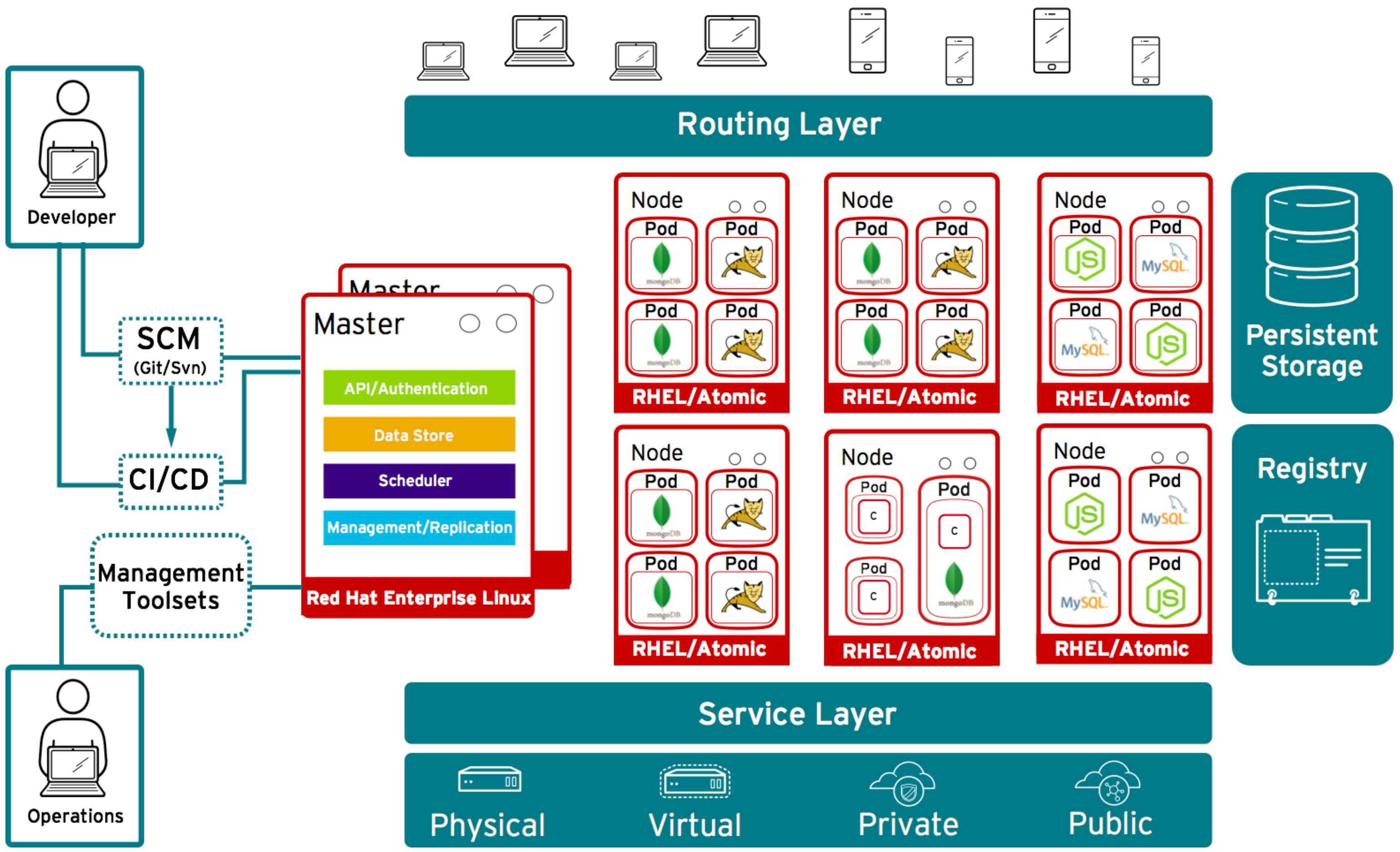
fluentd

kibana

Maven™

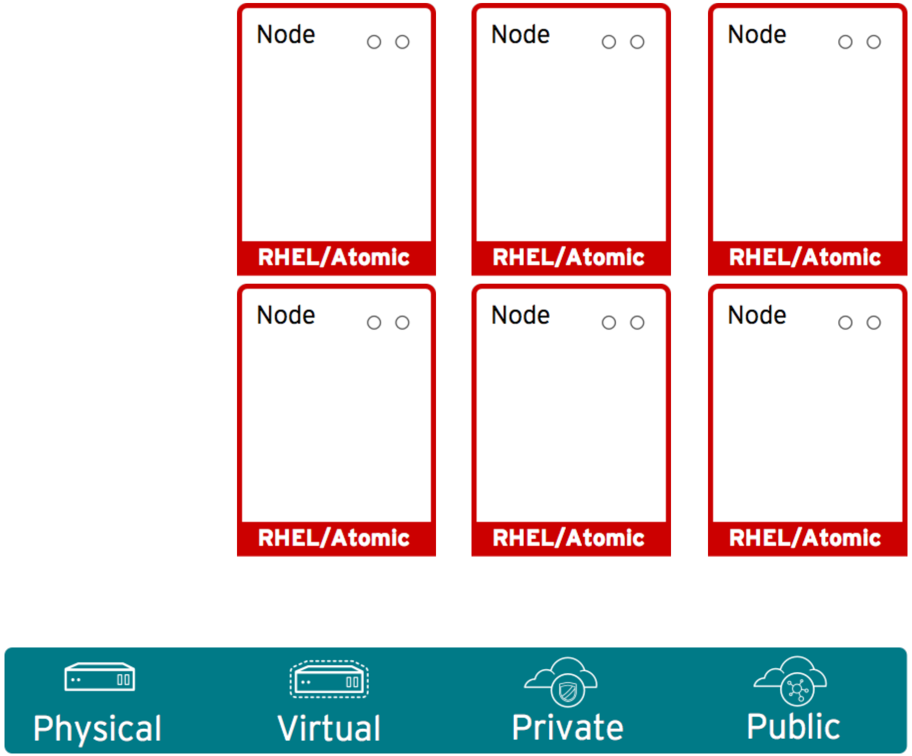
Architecture

10,000ft View

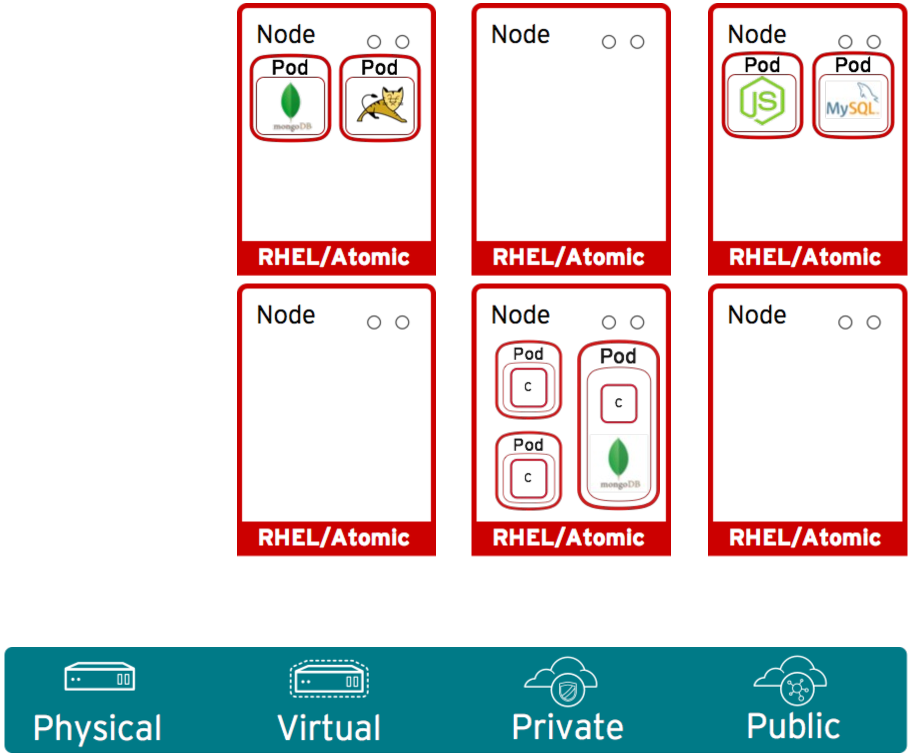


OpenShift runs on your choice of infrastructure

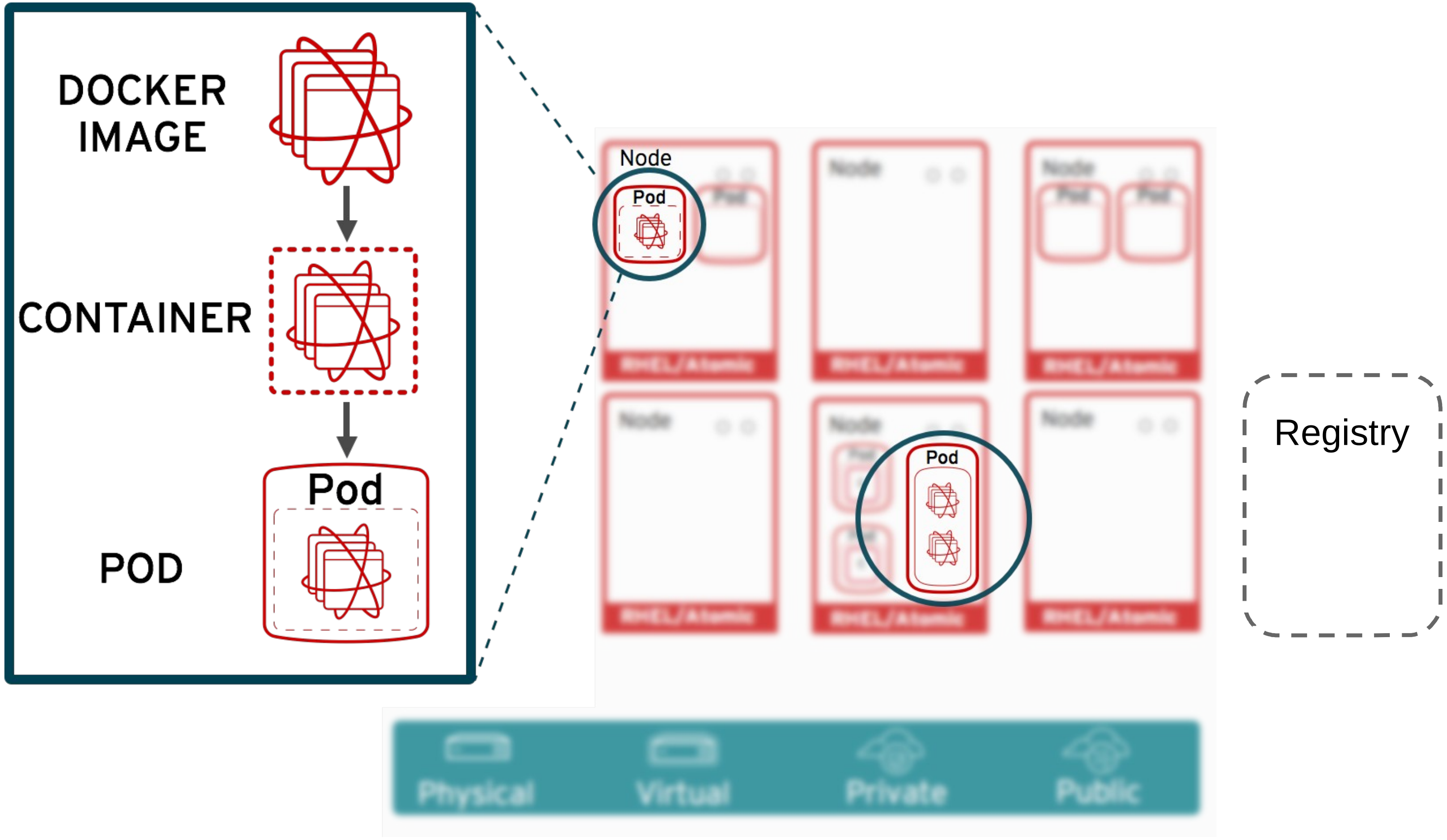
Nodes are instances of RHEL where apps will run



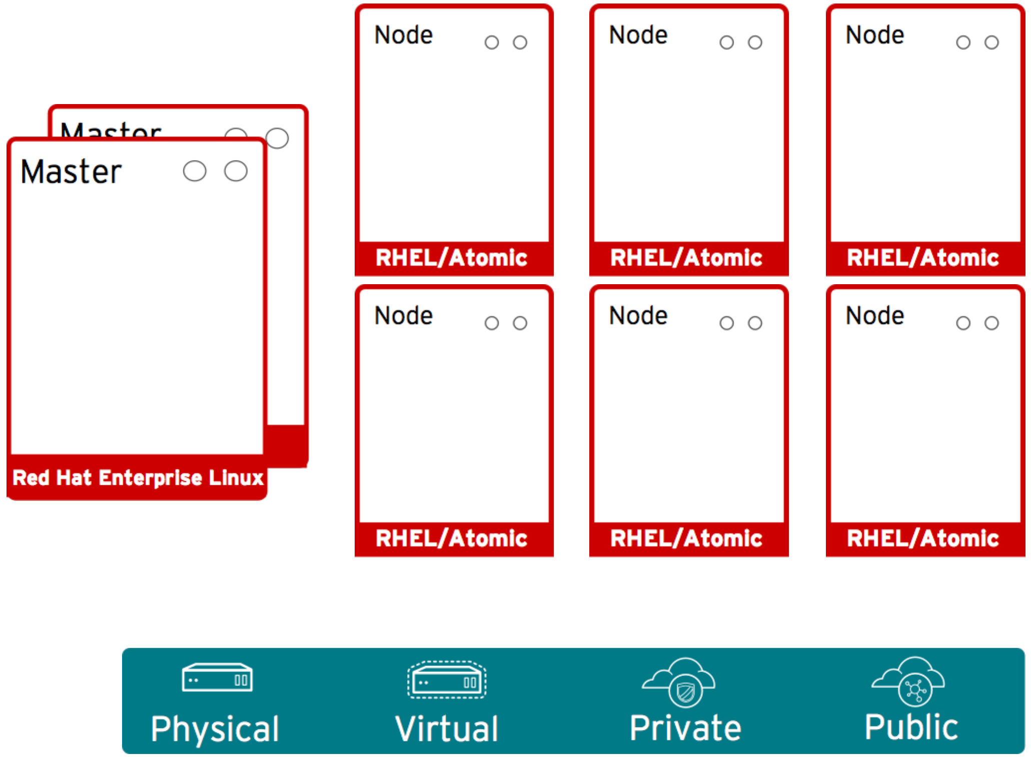
App services run in docker containers on each node



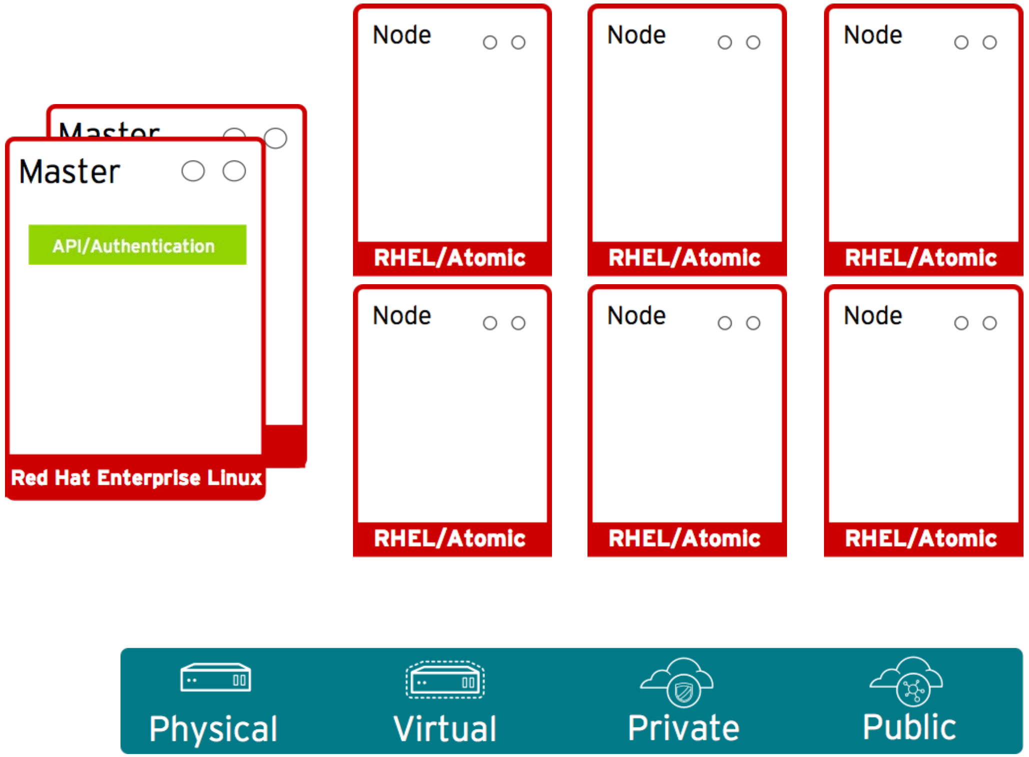
Pods run one or more docker containers as a unit



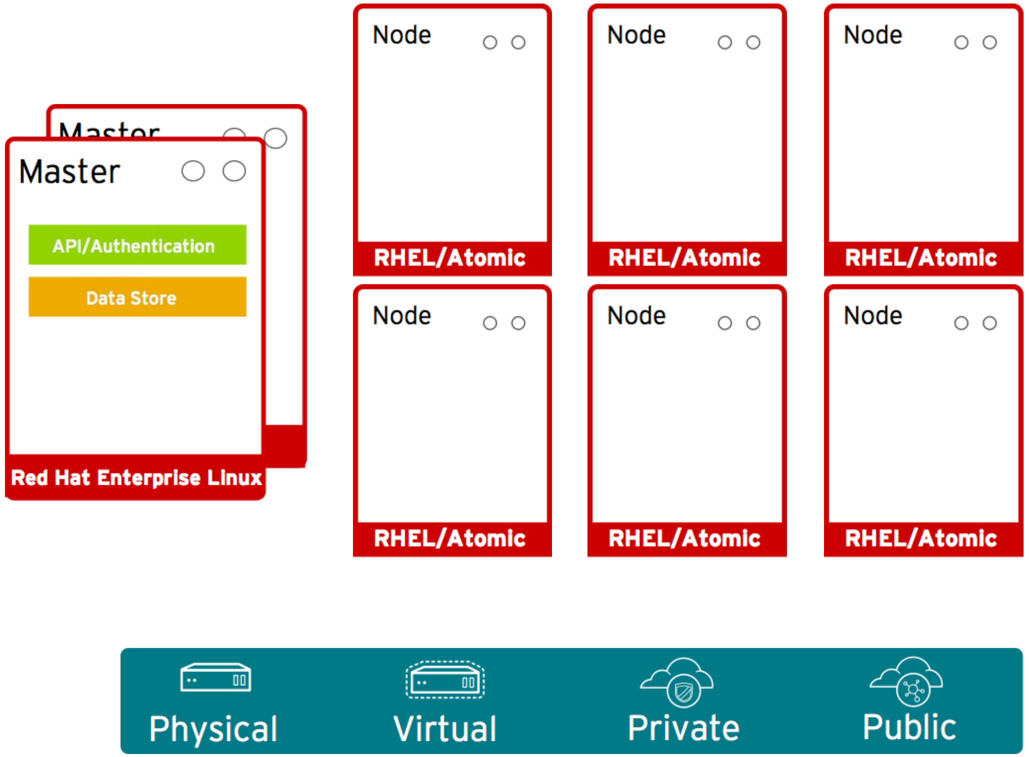
Masters leverage kubernetes to orchestrate nodes / apps



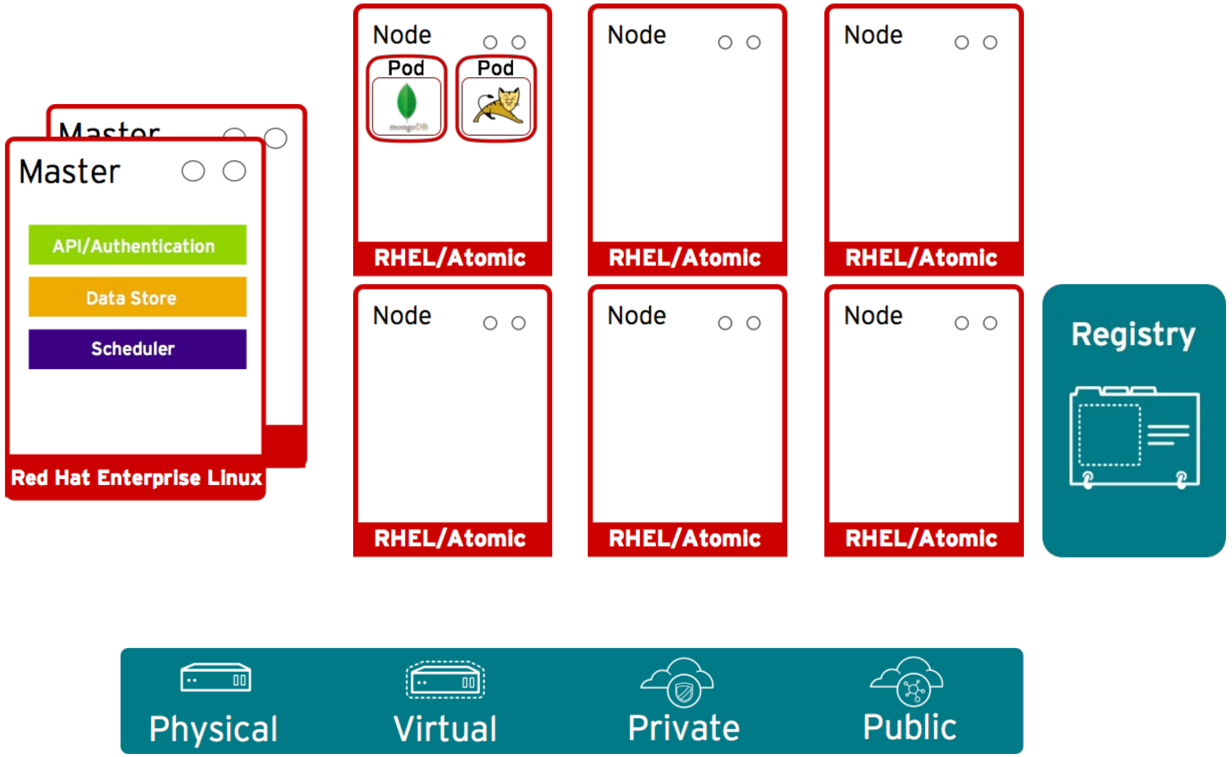
Master provides authenticated API for users & clients



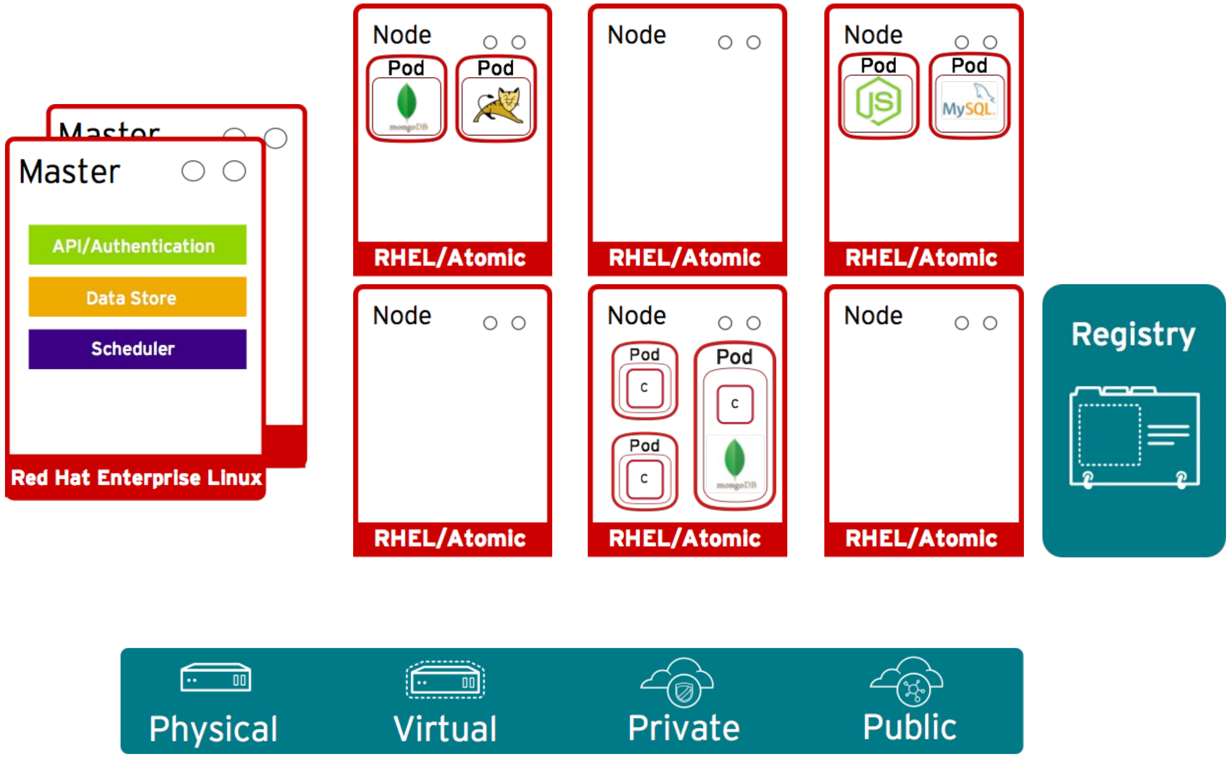
Master uses etcd key-value data store for persistence



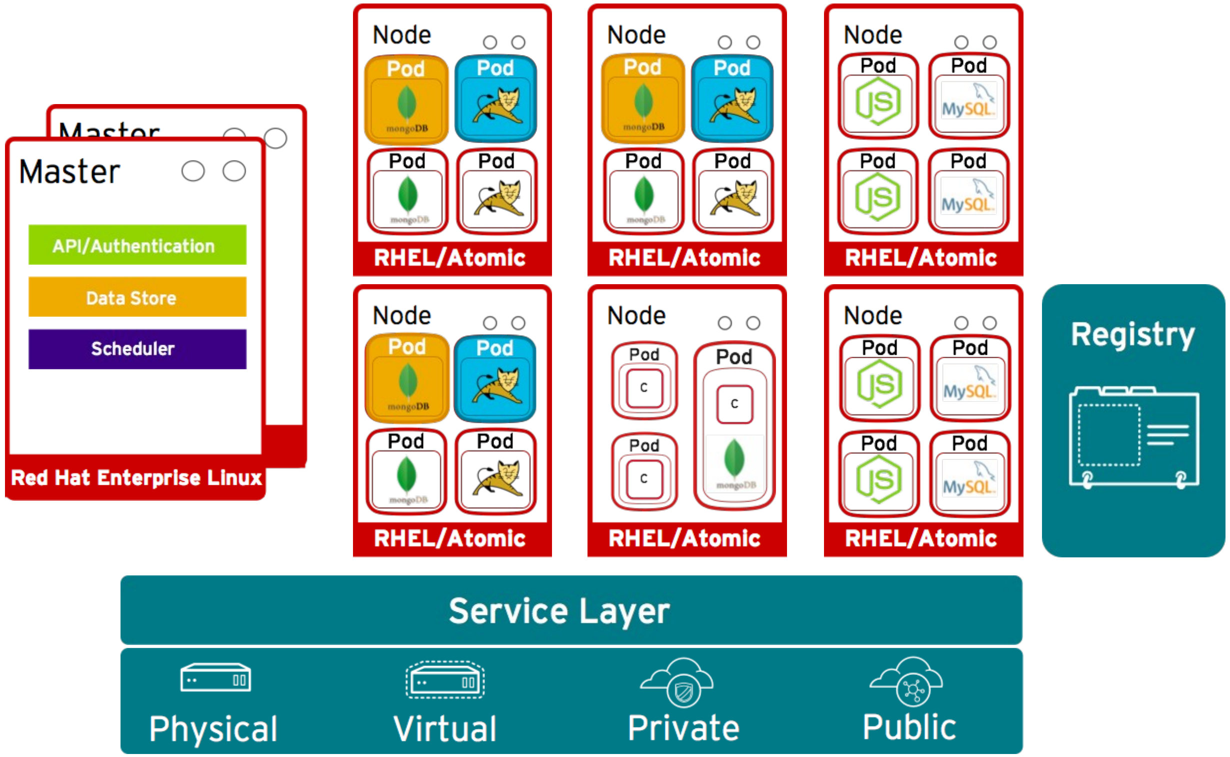
Master provides scheduler for pod placement on nodes



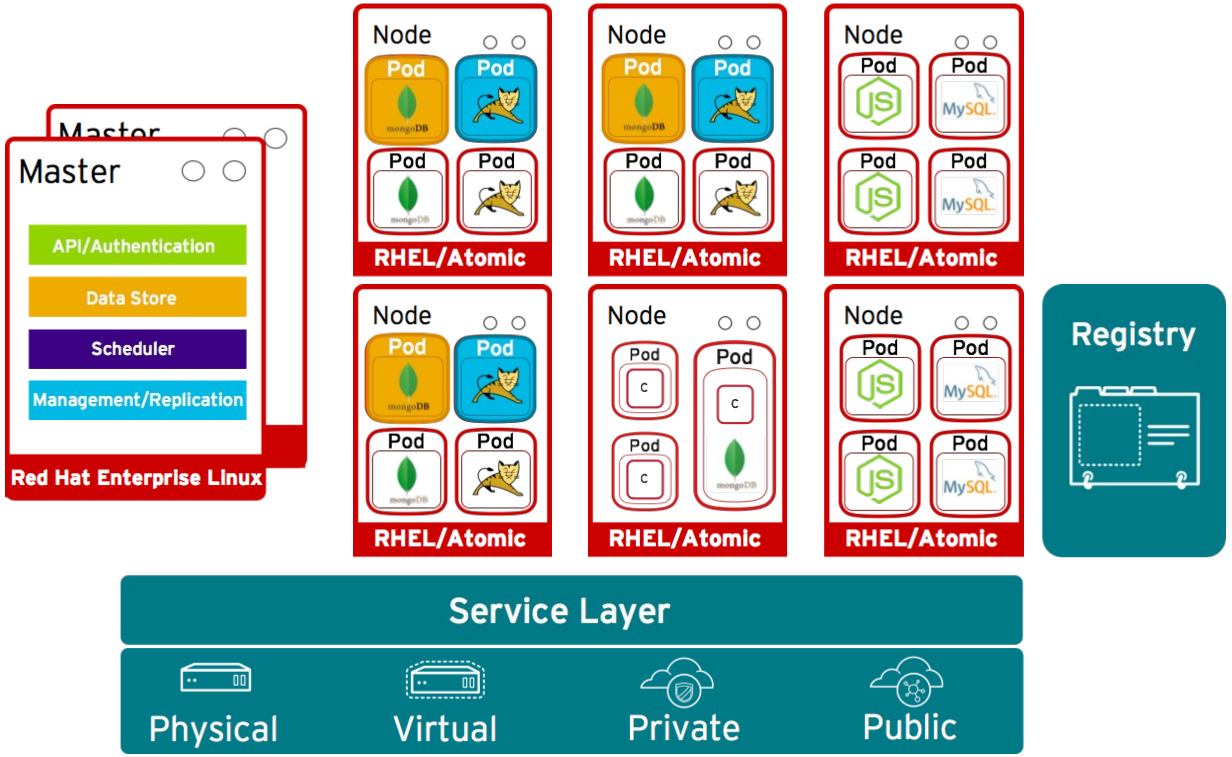
Pod placement is determined based on defined policy



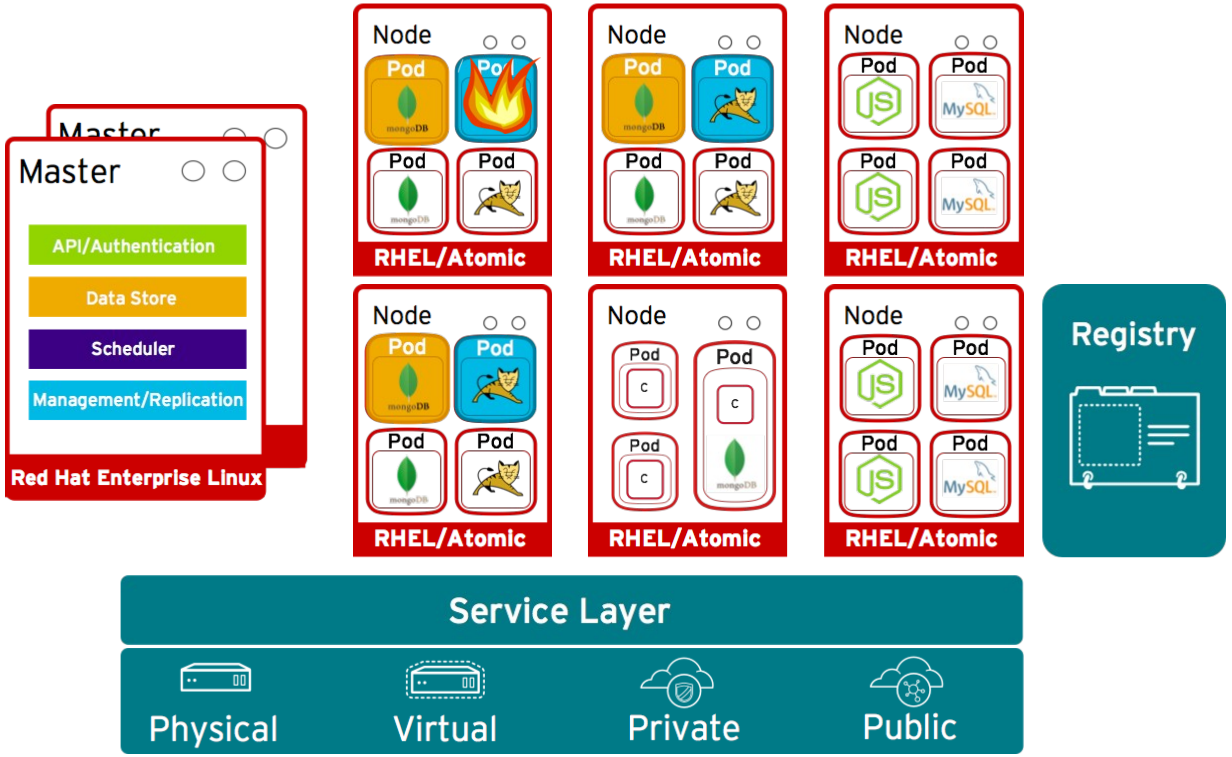
Services allow related pods to connect to each other



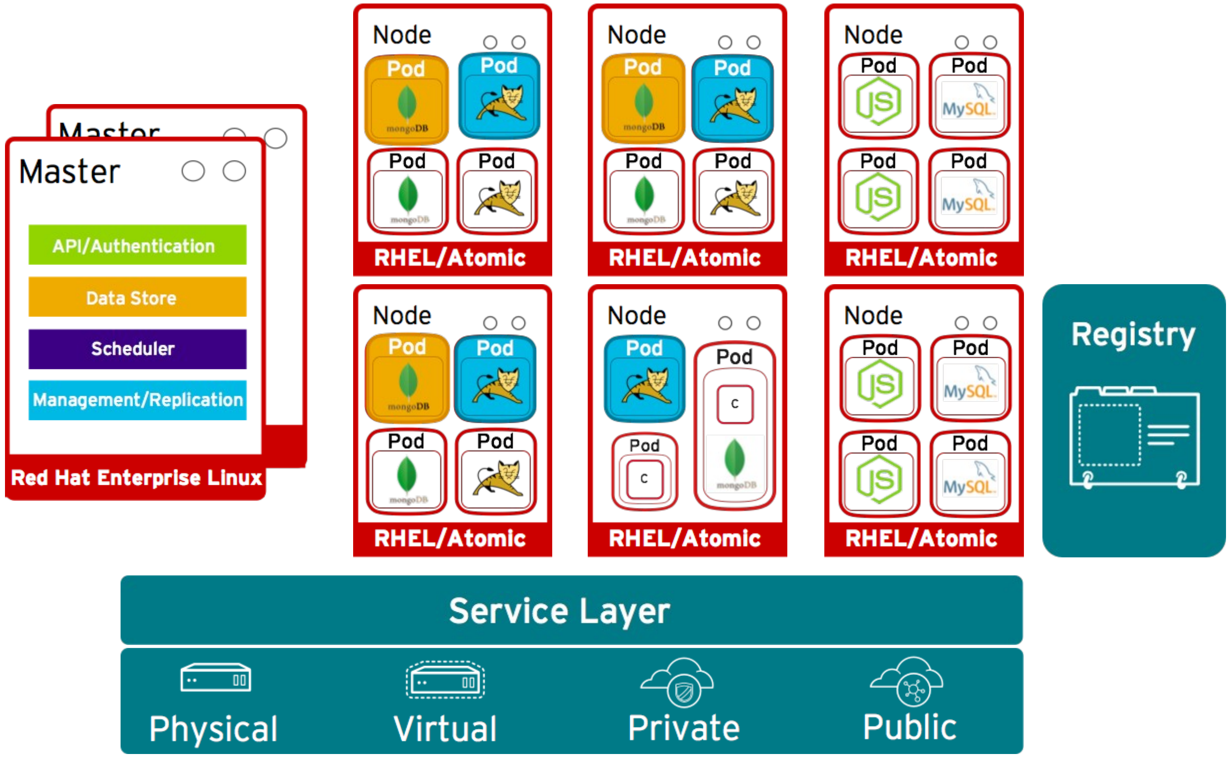
Management/Replication controller manages the pod lifecycle



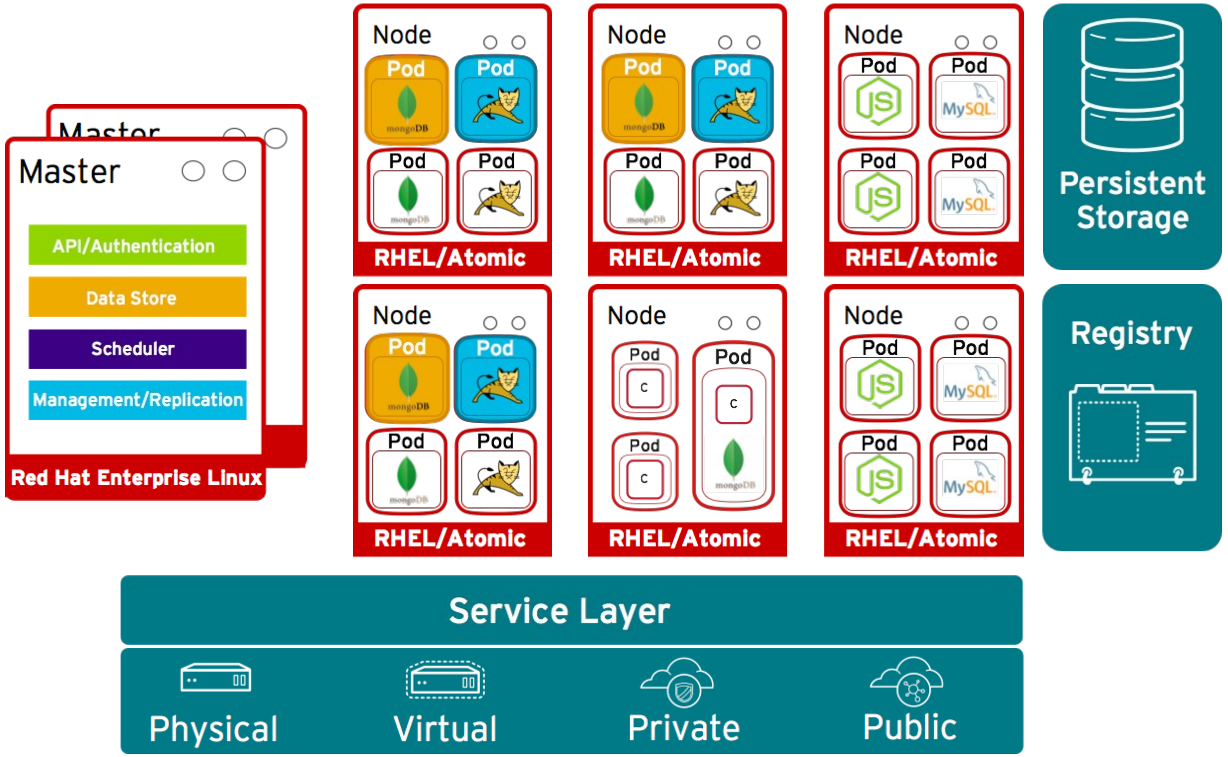
What if a pod goes down?



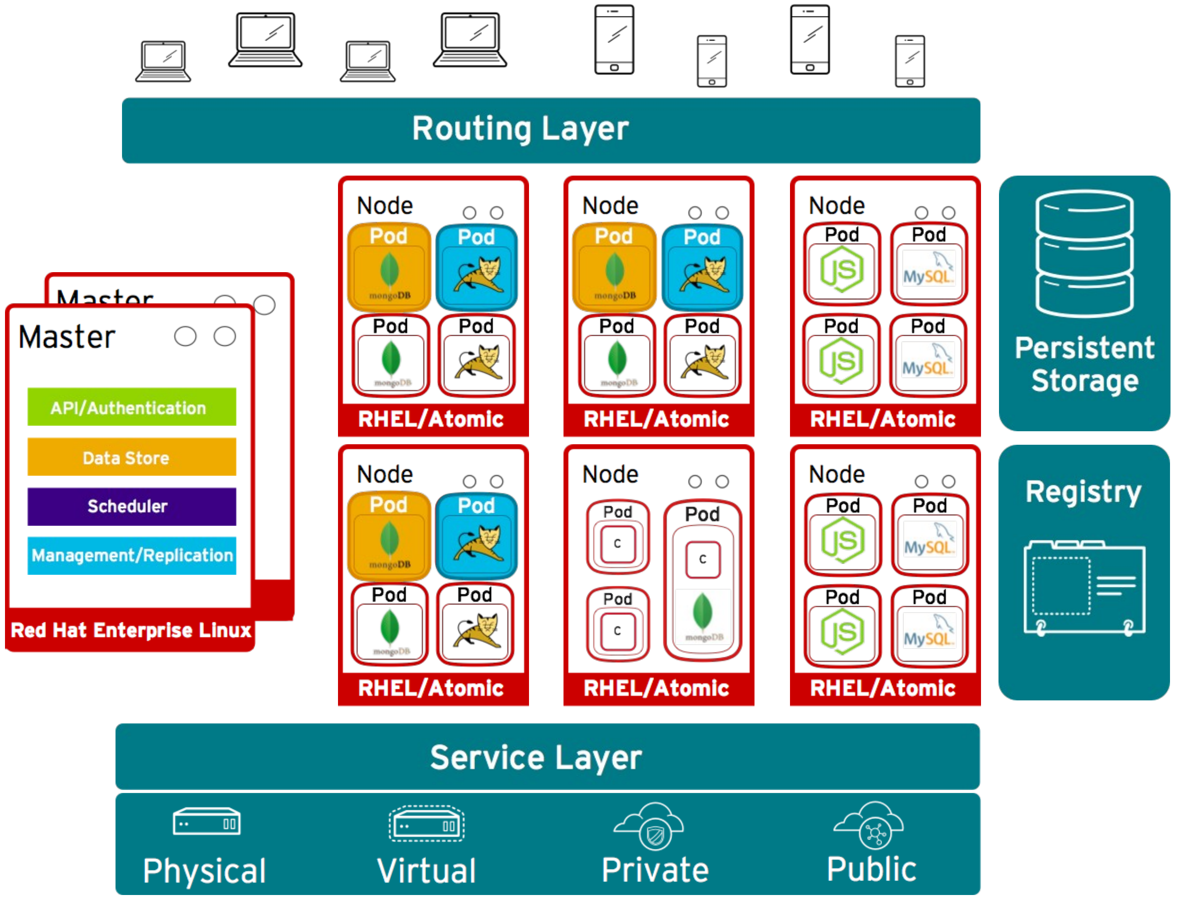
OpenShift automatically recovers and deploys a new Pod



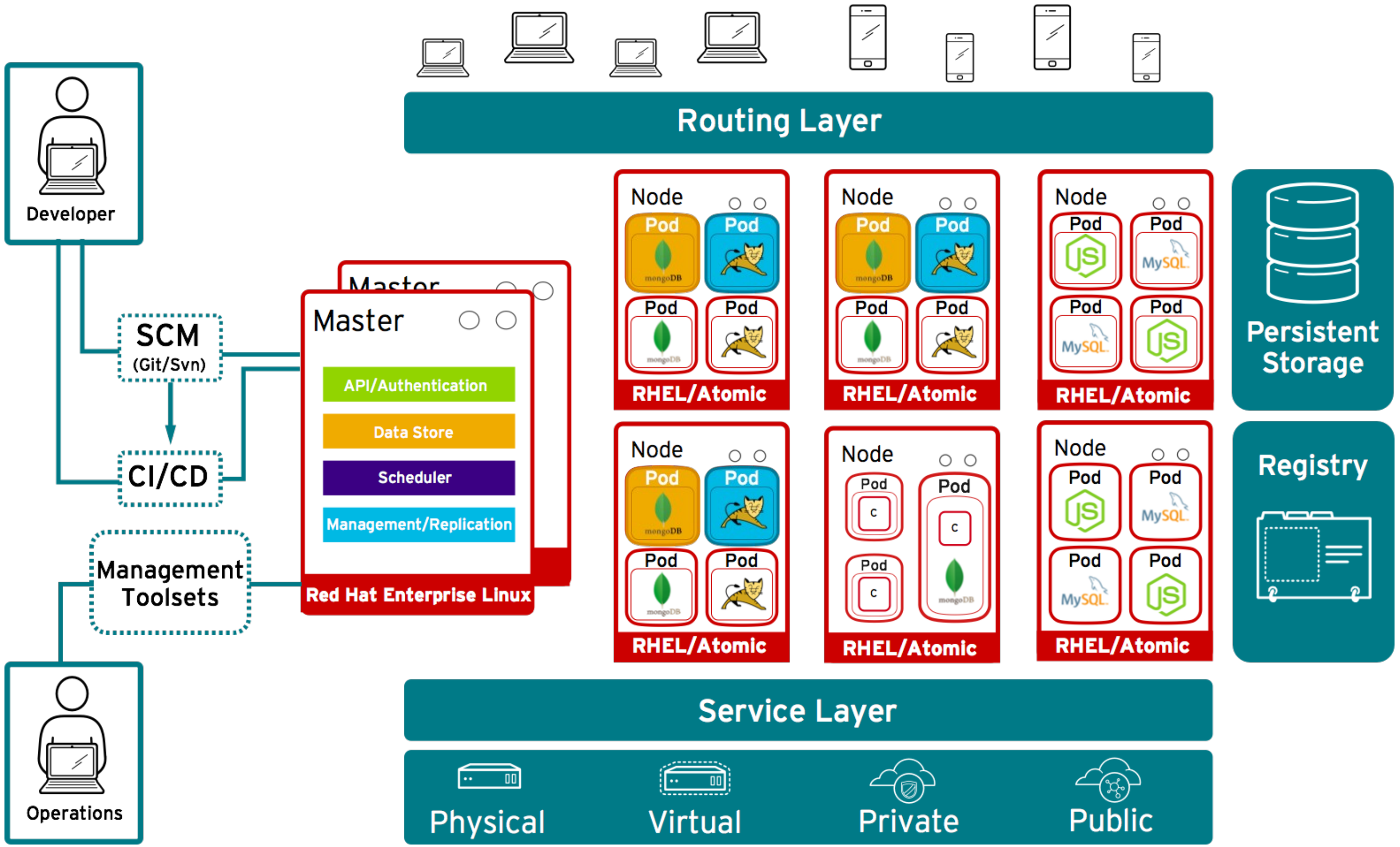
Pods can attach to shared storage for stateful services



Routing layer routes external app requests to pods



Developers access openShift via web, CLI or IDE



Storage Capabilities for stateful applications

- AWS
- Google Cloud Storage
- Ceph
- Gluster
- iSCSI
- NFS
- FibreChannel

New!

Storage Plugins

Attach persistent storage to your containers from a wide range of storage solutions.

Installation and first contact

Installation Openshift Enterprise 3.1

****** NAME RESOLUTION**
Use IDM
A A record is needed (*.app.ose.dom)

REPOSITORIES REQUIRED
rhel-7-server-rpms
rhel-7-server-extras-rpms
rhel-7-server-ose-3.1-rpms

```
# yum install -y git net-tools bind-utils iptables-services bridge-utils
# yum -y install atomic-openshift-utils
# yum -y install docker
```

Other steps : Configure Docker + Docker storage, install Openshift, configure the authentication, configure a registry then configure the router
(https://access.redhat.com/documentation/en/openshift-enterprise/3.1/installation-and-configuration/chapter-2-installing)

```
# oc login
```

To access the web console : https://openshift.dom:8443

First contact with OpenShift 3

CREATE A NEW USER

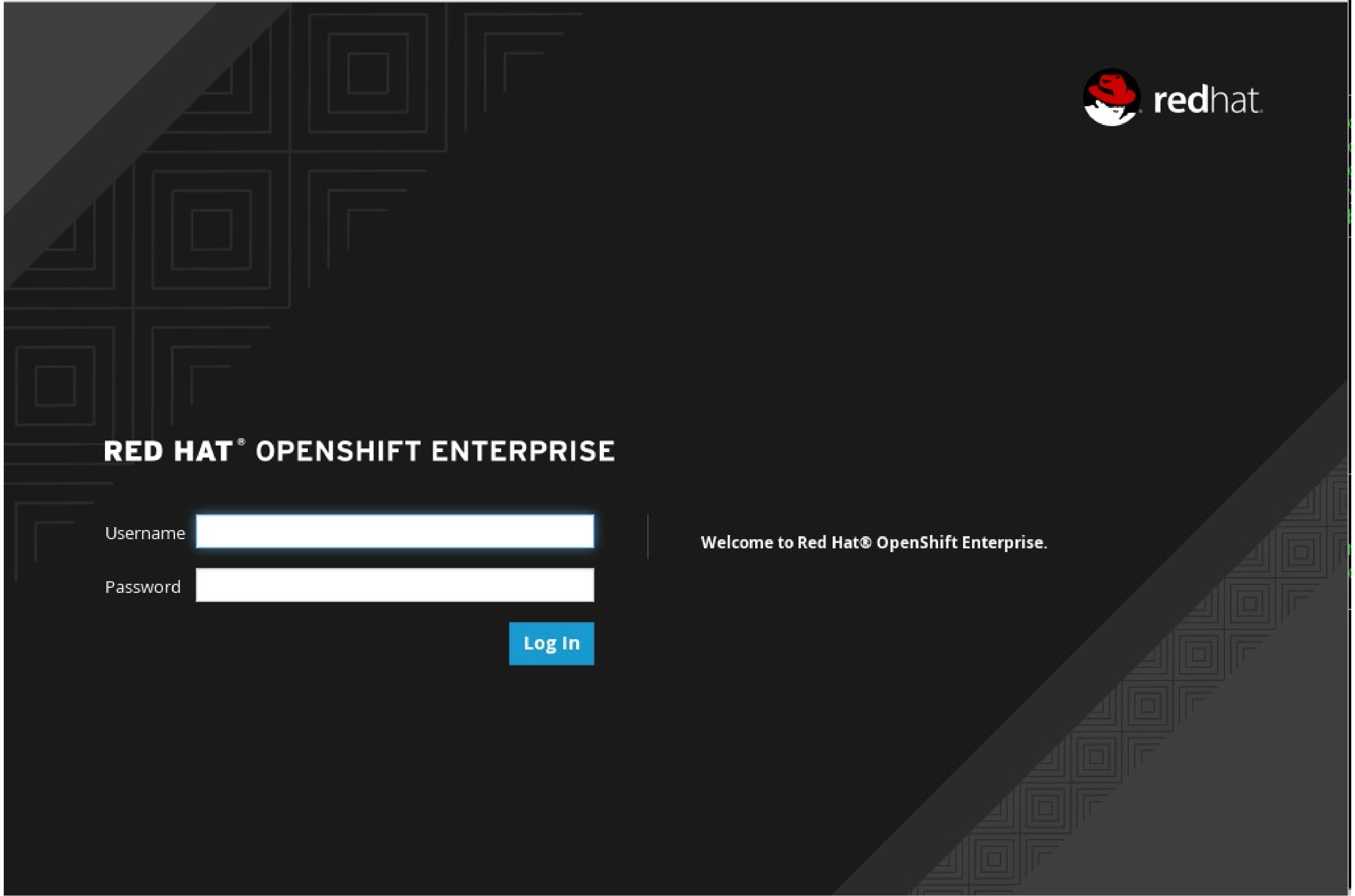
```
[root@os3 ~]# htpasswd -c /etc/origin/users.htpasswd georges
```

LOGIN AS A USER

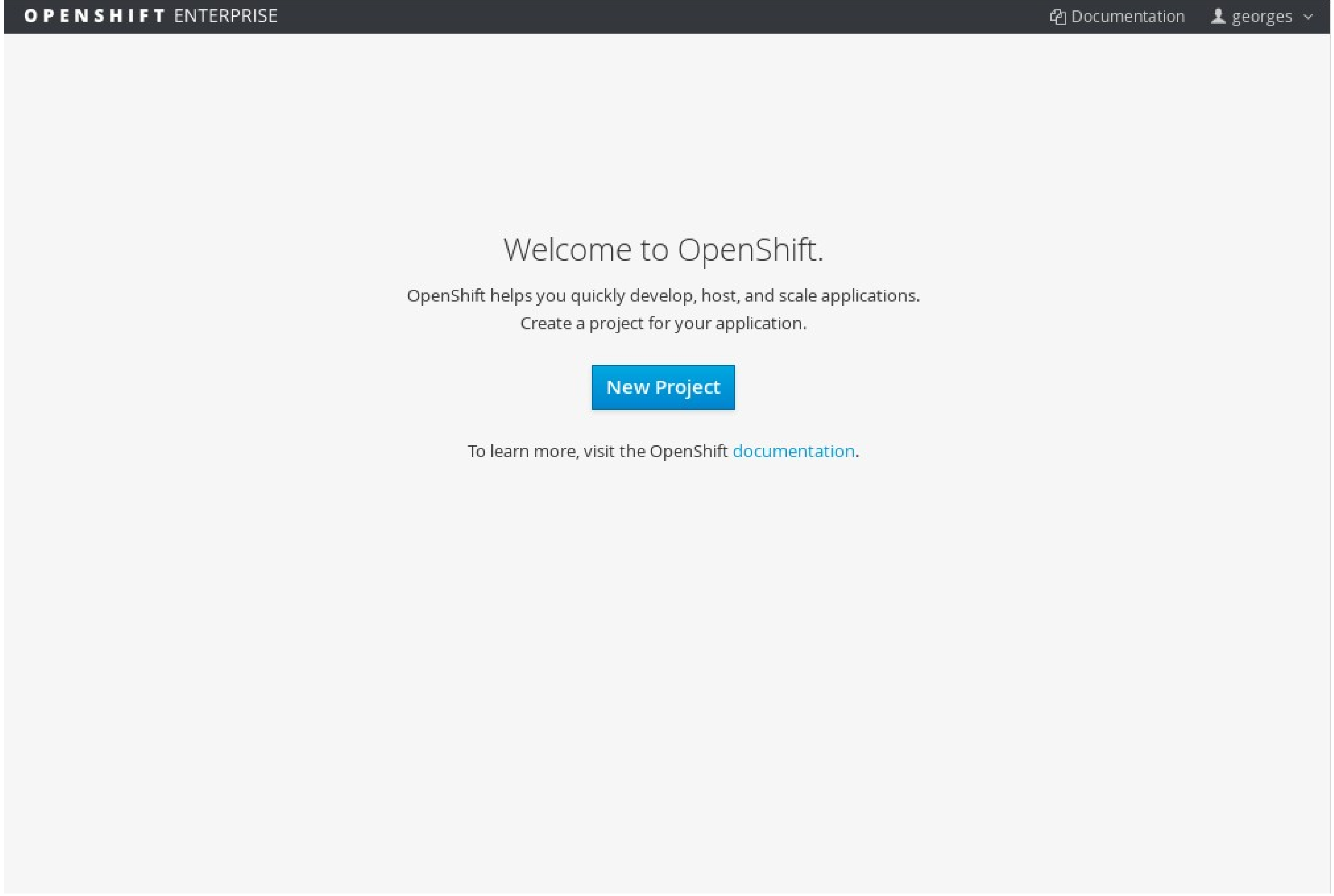
```
[root@os3 ~]# oc login
Authentication required for https://os3.mlc.dom:8443 (openshift)
Username: georges
Password:
Login successful.

You don't have any projects. You can try to create a new project, by running

    $ oc new-project <projectname>
```

LOGIN AS AN ADMIN

```
[root@os3 ~]# oc login -u system:admin -n default
 You have access to the following projects and can switch between them with 'oc project <projectname>':

  * default (current)
  * openshift
  * openshift-infra
  * template2

Using project "default".
```



Welcome to Red Hat® OpenShift Enterprise.



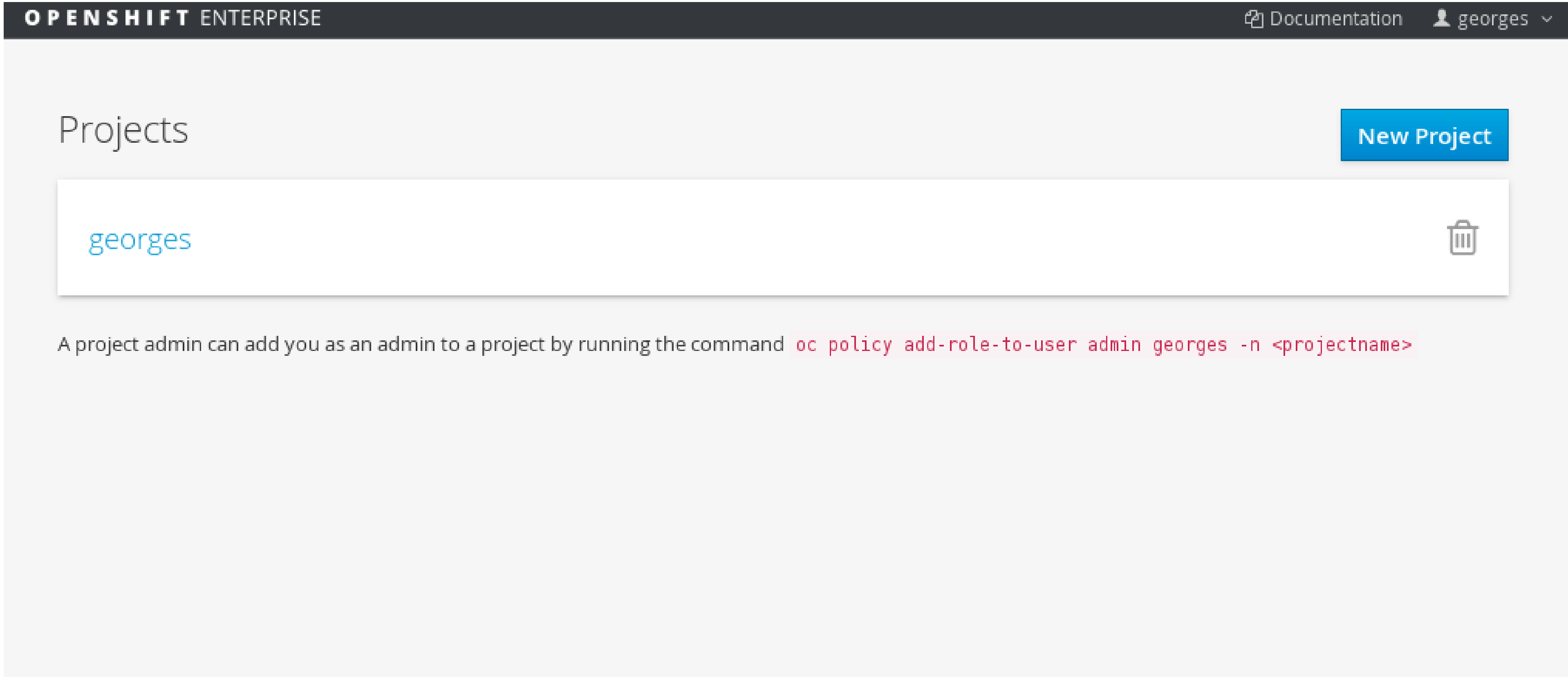
Welcome to OpenShift.

OpenShift helps you quickly develop, host, and scale applications.
Create a project for your application.

To learn more, visit the OpenShift documentation.

First contact with Openshift 3

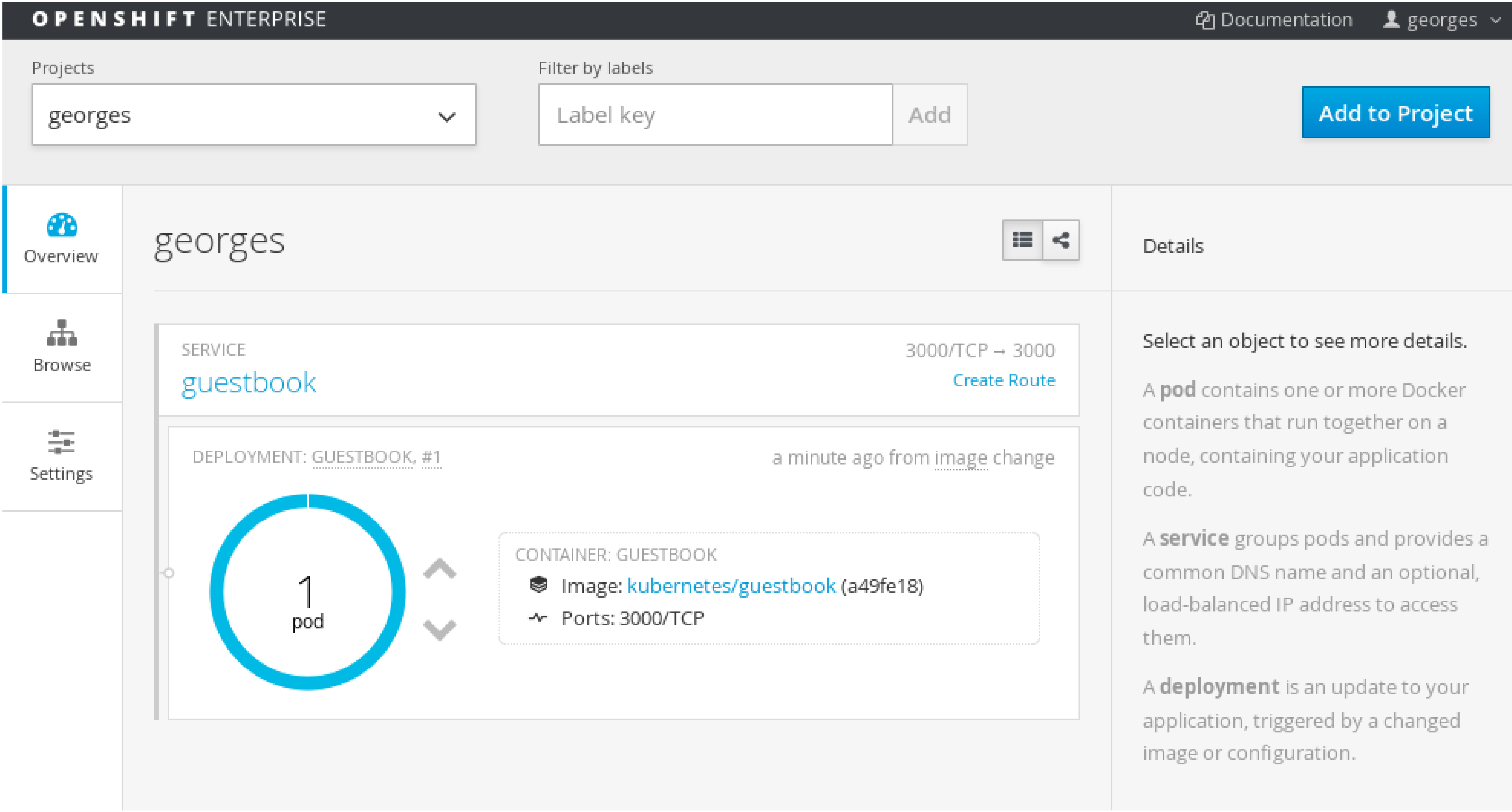
```
LOGIN AS A USER AND CREATE A NEW PROJECT
[root@os3 ~]# oc new-project georges
Now using project "georges" on server "https://os3.mlc.dom:8443".

CREATE AN APPLICATION (using a container from Docker Hub)
[root@os3 ~]# oc new-app kubernetes/guestbook
--> Found Docker image a49fe18 (15 months old) from Docker Hub for "kubernetes/guestbook"
    * An image stream will be created as "guestbook:latest" that will track this image
    * This image will be deployed in deployment config "guestbook"
    * Port 3000/tcp will be load balanced by service "guestbook"
--> Creating resources with label app=guestbook ...
    ImageStream "guestbook" created
    DeploymentConfig "guestbook" created
    Service "guestbook" created
--> Success
    Run 'oc status' to view your app.

[root@os3 ~]# oc get  pods
NAME                 READY     STATUS    RESTARTS   AGE
guestbook-1-deploy   1/1       Running   0          4s
guestbook-1-t2xos    1/1       Running   0          2s

[root@os3 ~]# oc get service
NAME        CLUSTER_IP       EXTERNAL_IP   PORT(S)    SELECTOR                                     AGE
guestbook   172.30.166.174   <none>        3000/TCP   app=guestbook,deploymentconfig=guestbook    22m
```



Projects

georges

A project admin can add you as an admin to a project by running the command `oc policy add-role-to-user admin georges -n <projectname>`



georges

SERVICE
guestbook
3000/TCP → 3000
Create Route

DEPLOYMENT: GUESTBOOK, #1
a minute ago from image change

1
pod

CONTAINER: GUESTBOOK
Image: kubernetes/guestbook (a49fe18)
Ports: 3000/TCP

Details

Select an object to see more details.

A **pod** contains one or more Docker containers that run together on a node, containing your application code.

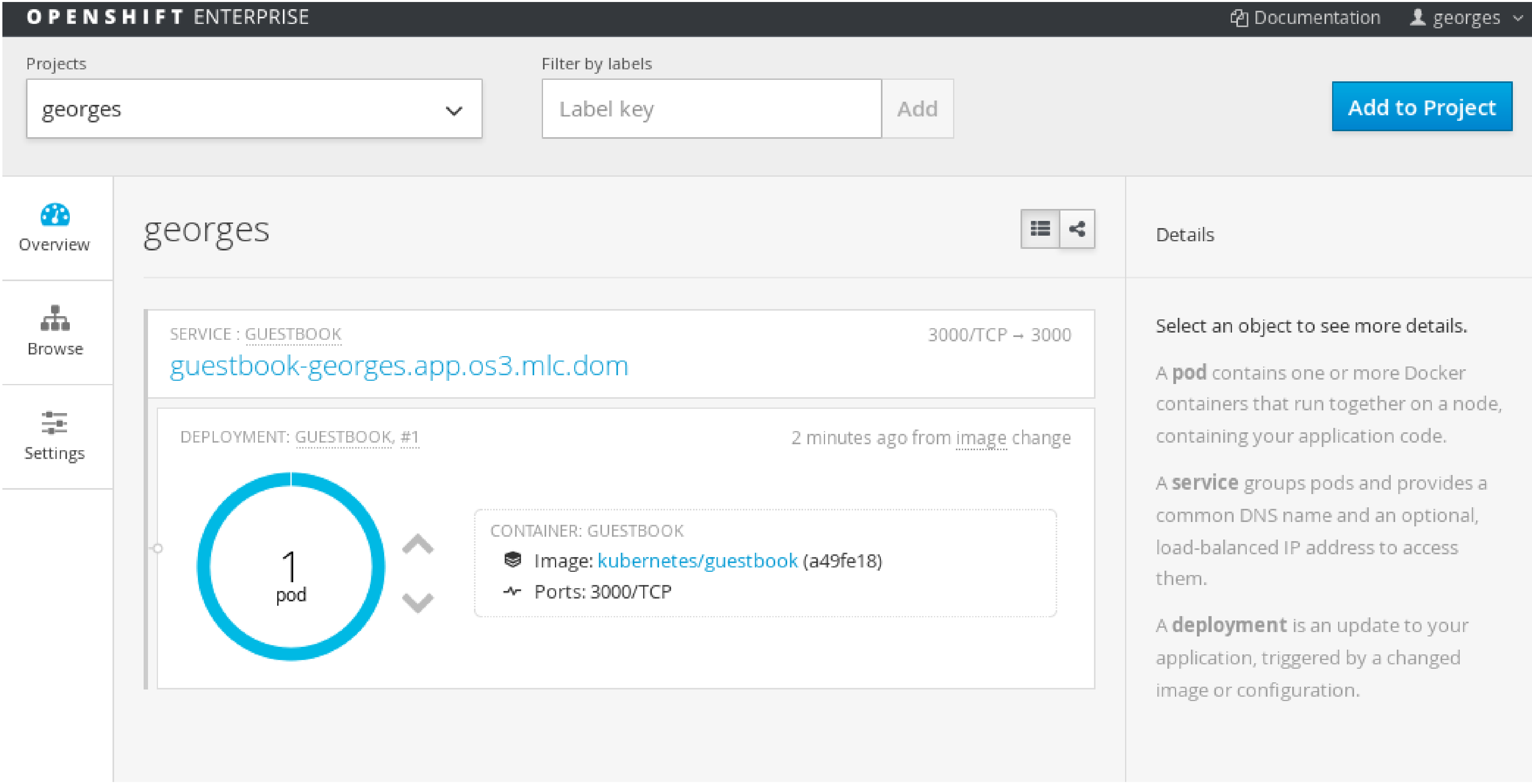
A **service** groups pods and provides a common DNS name and an optional, load-balanced IP address to access them.

A **deployment** is an update to your application, triggered by a changed image or configuration.

First contact with Openshift 3

EXPOSE A SERVICE

```
[root@os3 ~]# oc expose service guestbook
route "guestbook" exposed

[root@os3 ~]# oc get route
NAME        HOST/PORT                               PATH      SERVICE     LABELS          INSECURE POLICY   TLS TERMINATION
guestbook   guestbook-georges.app.os3.mlc.dom                 guestbook   app=guestbook
```



SERVICE : GUESTBOOK 3000/TCP → 3000
guestbook-georges.app.os3.mlc.dom

DEPLOYMENT: GUESTBOOK, #1 2 minutes ago from image change

1 pod

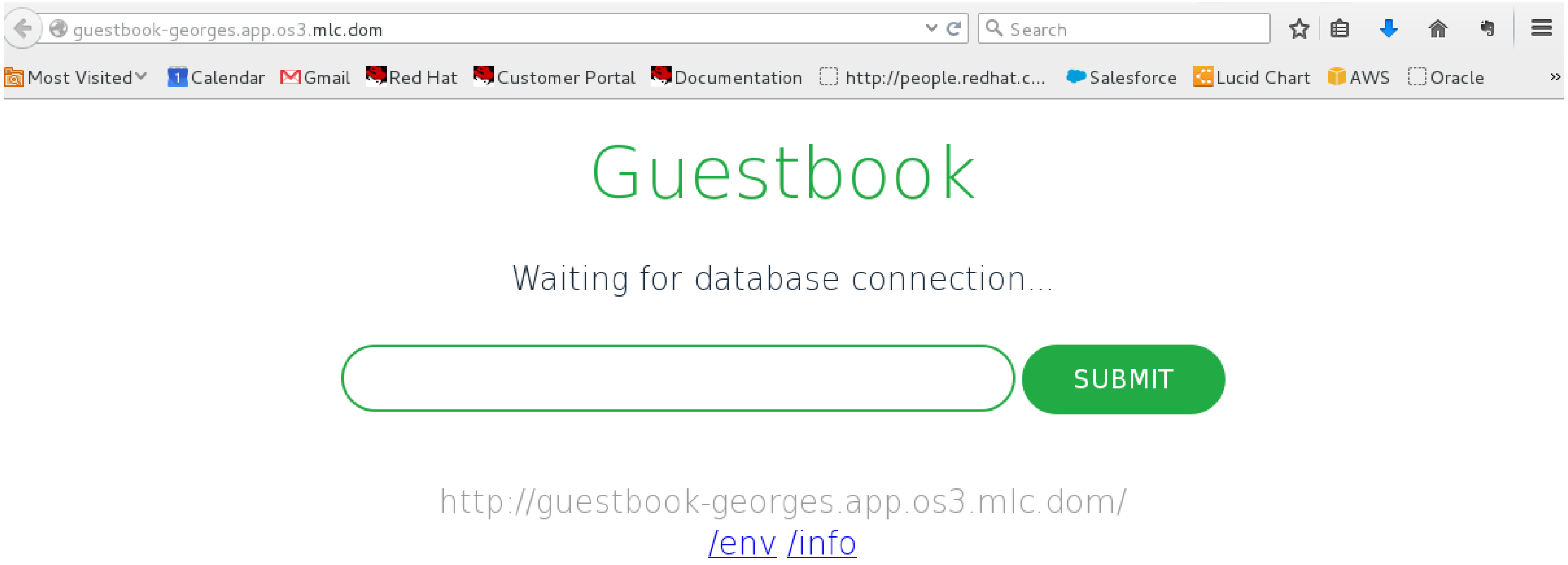
CONTAINER: GUESTBOOK
Image: kubernetes/guestbook (a49fe18)
Ports: 3000/TCP

Select an object to see more details.

A **pod** contains one or more Docker containers that run together on a node, containing your application code.

A **service** groups pods and provides a common DNS name and an optional, load-balanced IP address to access them.

A **deployment** is an update to your application, triggered by a changed image or configuration.


guestbook-georges.app.os3.mlc.dom

Most Visited Calendar Gmail Red Hat Customer Portal Documentation http://people.redhat.c... Salesforce Lucid Chart AWS Oracle

Guestbook

Waiting for database connection...

SUBMIT

http://guestbook-georges.app.os3.mlc.dom/
/env /info

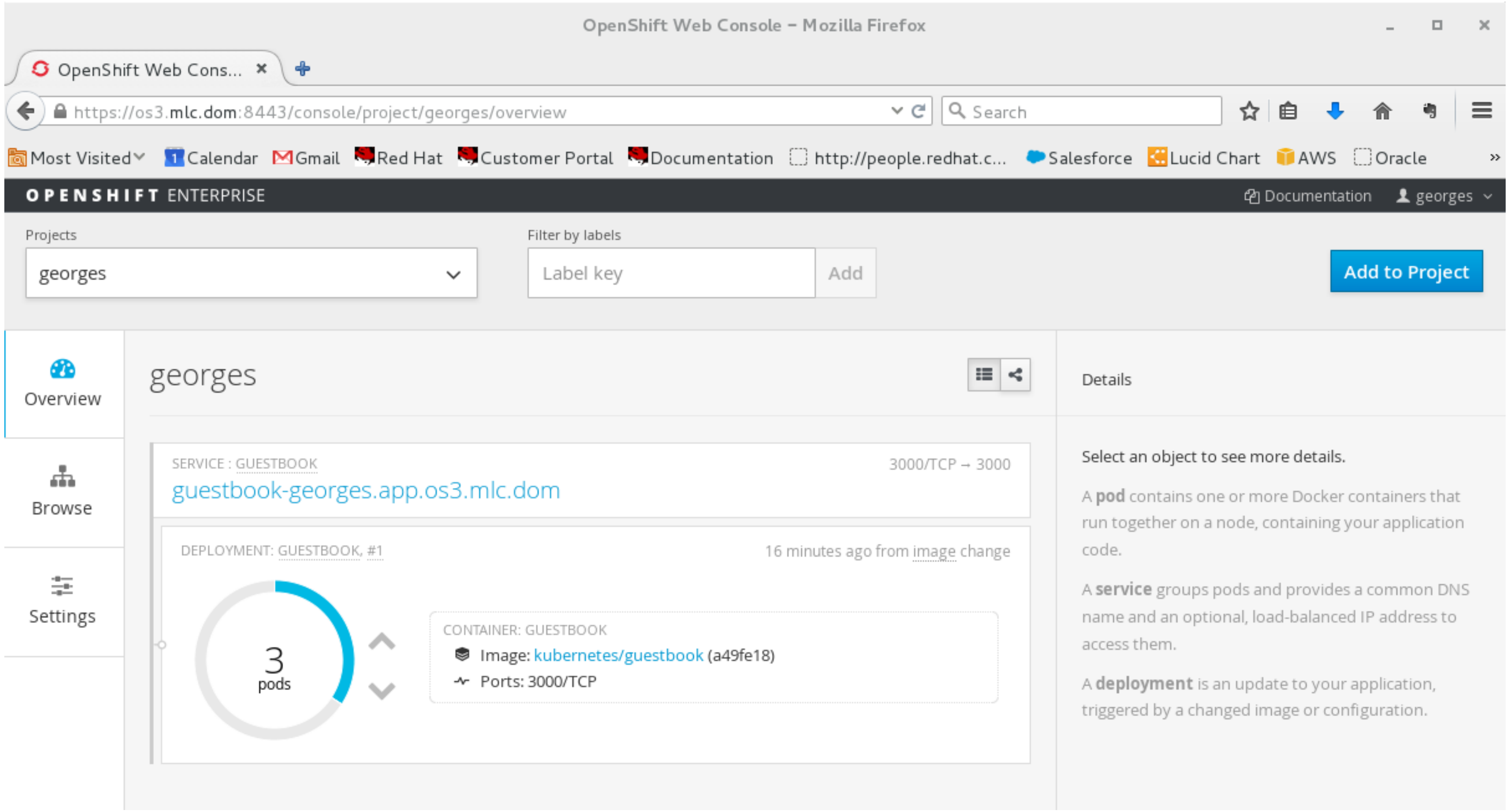
Replication Controllers (RC)
Used to specify and then ensure the desired number of Pods (replicas) are in existence

DeploymentConfiguration(DC)
Defines how something in Openshift should be deployed

```
[root@os3] # oc get rc
CONTROLLER    CONTAINER(S)   IMAGE(S)                                                                                                 SELECTOR
REPLICAS   AGE
guestbook-1   guestbook      kubernetes/guestbook@sha256:a49fe18bb57c8eee16e2002987e041f5ae9b5b70ae7b3d49eb60e5c26b9c6bd0
app=guestbook,deployment=guestbook-1,deploymentconfig=guestbook   1          10m


[root@os3 key]# oc get dc
NAME        TRIGGERS                    LATEST
guestbook   ConfigChange, ImageChange   1
```

SCALING UP

```
[root@os3]# oc scale --replicas=3 rc guestbook-1
replicationcontroller "guestbook-1" scaled
```



OpenShift Web Console – Mozilla Firefox

https://os3.mlc.dom:8443/console/project/georges/overview

OPENSHIFT ENTERPRISE

Projects: georges

Filter by labels: Label key

georges

SERVICE : GUESTBOOK 3000/TCP → 3000

guestbook-georges.app.os3.mlc.dom

DEPLOYMENT: GUESTBOOK, #1 16 minutes ago from image change

3 pods

CONTAINER: GUESTBOOK

Image: kubernetes/guestbook (a49fe18)

Ports: 3000/TCP

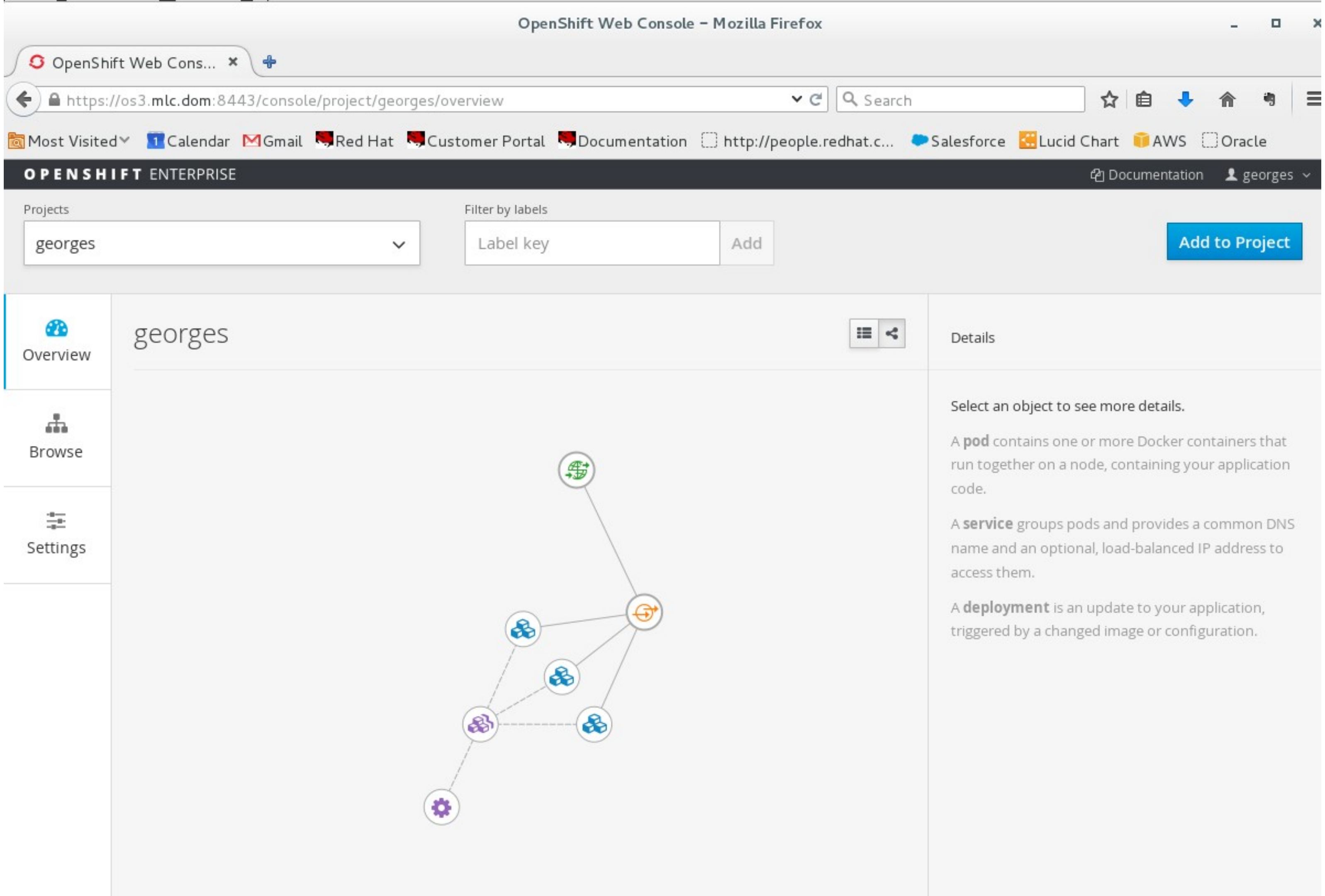
Details

Select an object to see more details.

A **pod** contains one or more Docker containers that run together on a node, containing your application code.

A **service** groups pods and provides a common DNS name and an optional, load-balanced IP address to access them.

A **deployment** is an update to your application, triggered by a changed image or configuration.



Select an object to see more details.

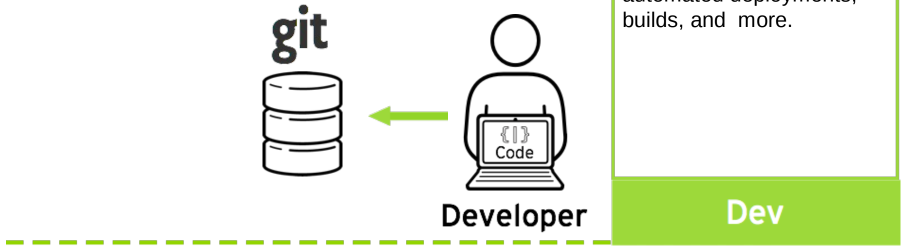
A **pod** contains one or more Docker containers that run together on a node, containing your application code.

A **service** groups pods and provides a common DNS name and an optional, load-balanced IP address to access them.

A **deployment** is an update to your application, triggered by a changed image or configuration.

Source to image

Source 2 Image Walk Through



Can configure triggers for automated deployments, builds, and more.

Code

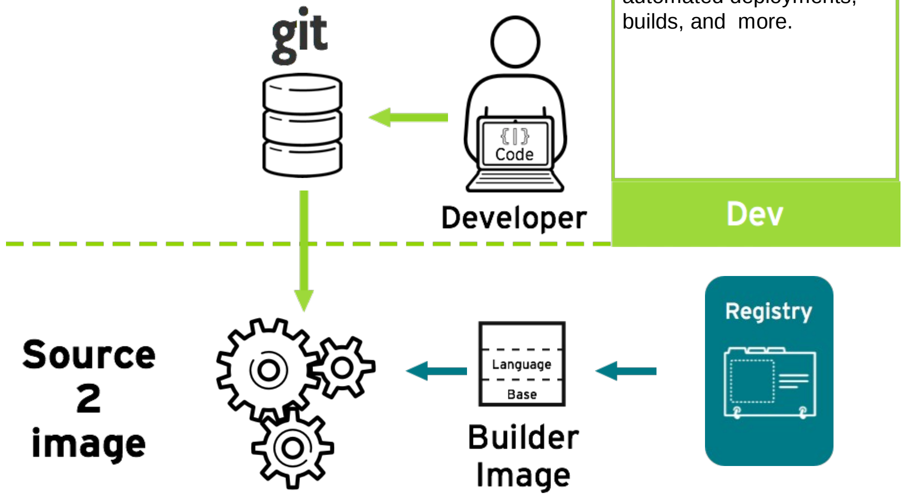
Developers can leverage existing development tools and then access the OpenShift Web, CLI or IDE interfaces to create new application services and push source code via GIT. OpenShift can also accept binary deployments or be fully integrated with a customer’s existing CI/CD environment.

Source 2 Image Walk Through

Build

OpenShift automates the Docker image build process with Source-to-Image (S2I). S2I combines source code with a corresponding Builder image from the integrated Docker registry. Builds can also be triggered manually or automatically by setting a Git webhook.

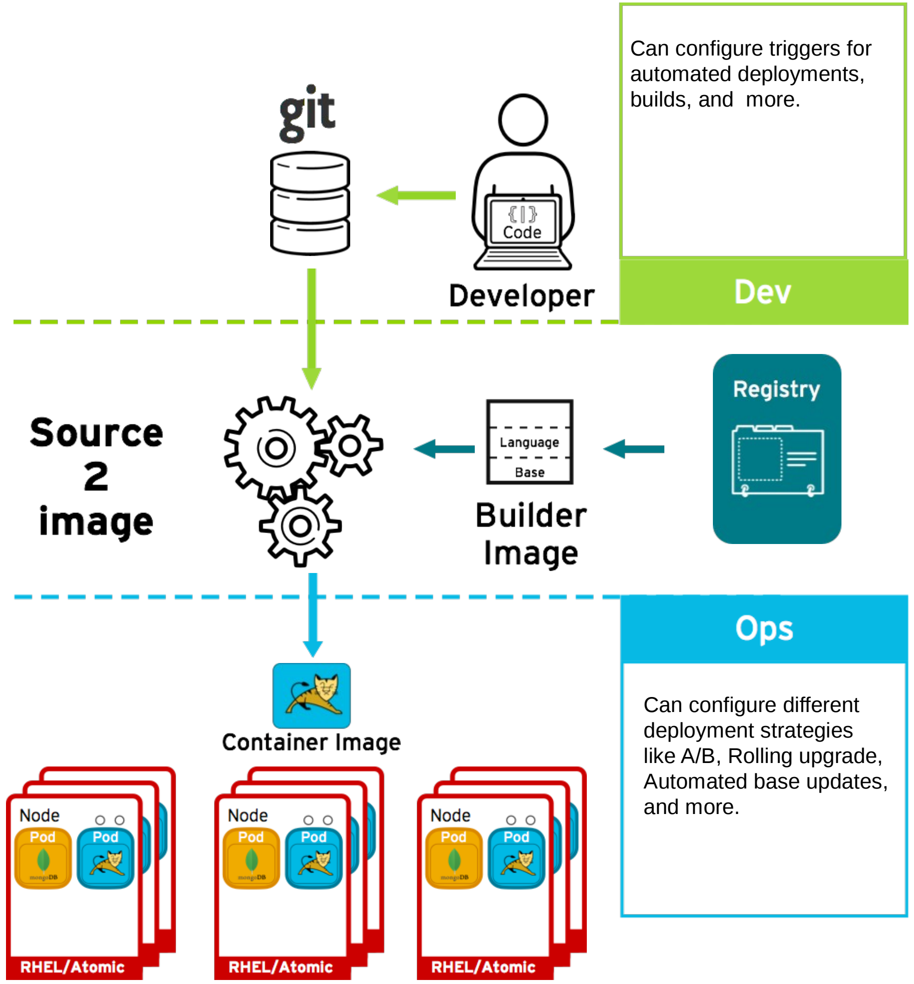
Can configure triggers for automated deployments, builds, and more.



Source 2 Image Walk Through

Deploy

OpenShift automates the deployment of application containers across multiple Node hosts via the Kubernetes scheduler. Users can automatically trigger deployments on application changes and do rollbacks, configure A/B deployments & other custom deployment types.



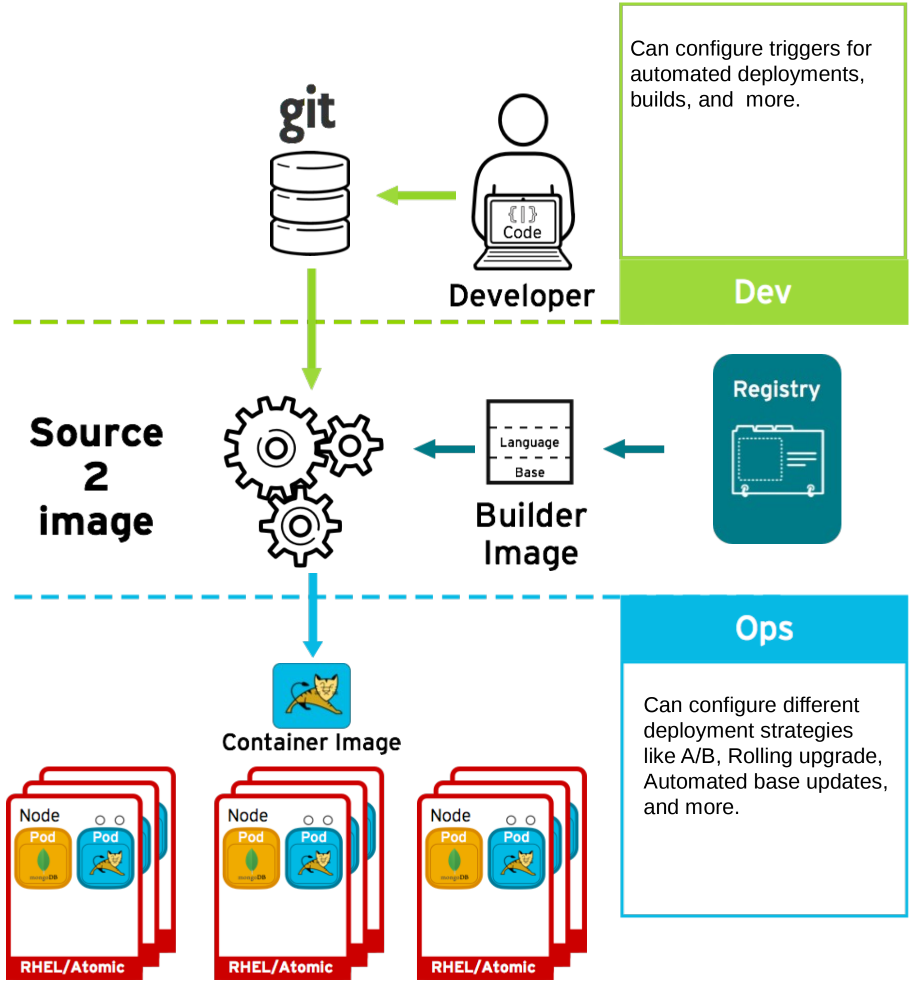
Can configure triggers for automated deployments, builds, and more.
git
{ | }
Code
Developer
Dev
Source 2 image
Language
Base
Builder Image
Registry
Ops
Can configure different deployment strategies like A/B, Rolling upgrade, Automated base updates, and more.
Container Image
Node
Pod
Pod
RHEL/Atomic
Node
Pod
Pod
RHEL/Atomic
Node
Pod
Pod
RHEL/Atomic

Source 2 Image Walk Through

Code

Build

Deploy

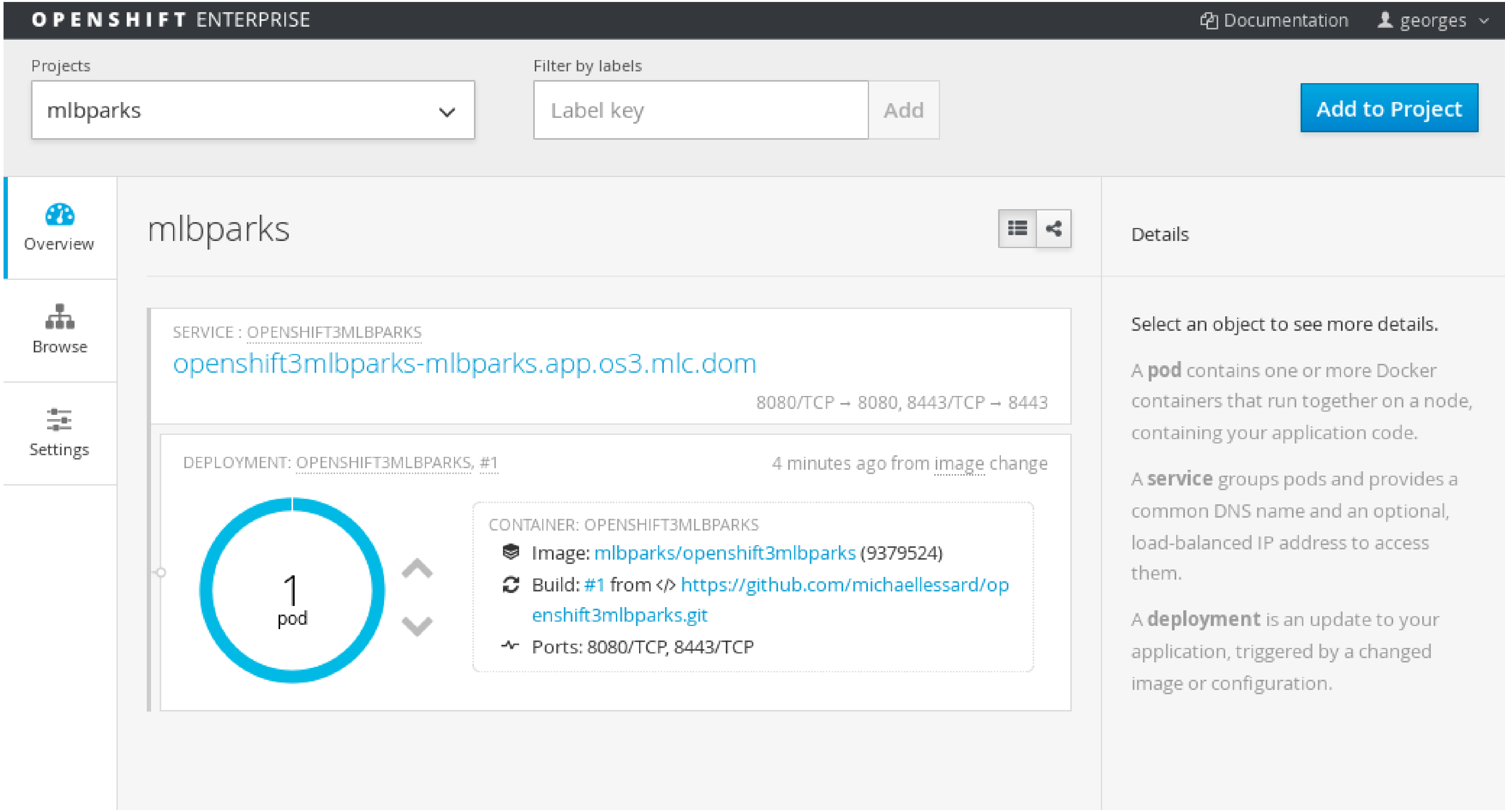


Can configure triggers for automated deployments, builds, and more.

Can configure different deployment strategies like A/B, Rolling upgrade, Automated base updates, and more.

SOURCE TO IMAGE EXAMPLE

```
[root@os3 ~]# oc new-project mlbparks
SOURCE TO IMAGE
[root@os3 ~]# oc new-app registry.access.redhat.com/jboss-eap-6/eap-openshift~https://github.com/michaellessard/openshift3mlbparks.git
[root@os3 ~]# oc get builds
NAME                  TYPE      FROM      STATUS    STARTED          DURATION
openshift3mlbparks-1   Source    Git       Running   12 seconds ago   12s

# oc build-logs openshft3mlbparks-1
…....
Downloading: https://repo1.maven.org/maven2/org/apache/commons/commons-compress/1.5/commons-compress-1.5.jar
Downloaded: https://repo1.maven.org/maven2/org/codehaus/plexus/plexus-archiver/2.4.1/plexus-archiver-2.4.1.jar (161 KB at
1417.2 KB/sec)
Downloading: https://repo1.maven.org/maven2/org/tukaani/xz/1.2/xz-1.2.jar
Downloaded: https://repo1.maven.org/maven2/org/apache/maven/maven-archiver/2.5/maven-archiver-2.5.jar (22 KB at 150.0
KB/sec)
…..

[root@os3 ~]# oc get pods
NAME                        READY     STATUS      RESTARTS   AGE
openshift3mlbparks-1-build   0/1       Completed   0          2m
openshift3mlbparks-1-ntig9   1/1       Running     0          27s

[root@os3 ~]# oc expose service openshift3mlbparks
```



mlbparks

Details

SERVICE : OPENSHIFT3MLBPARKS

openshift3mlbparks-mlbparks.app.os3.mlc.dom

8080/TCP → 8080, 8443/TCP → 8443

DEPLOYMENT: OPENSHIFT3MLBPARKS, #1

4 minutes ago from image change

1 pod

CONTAINER: OPENSHIFT3MLBPARKS

Image: mlbparks/openshift3mlbparks (9379524)

Build: #1 from </> https://github.com/michaellessard/openshift3mlbparks.git

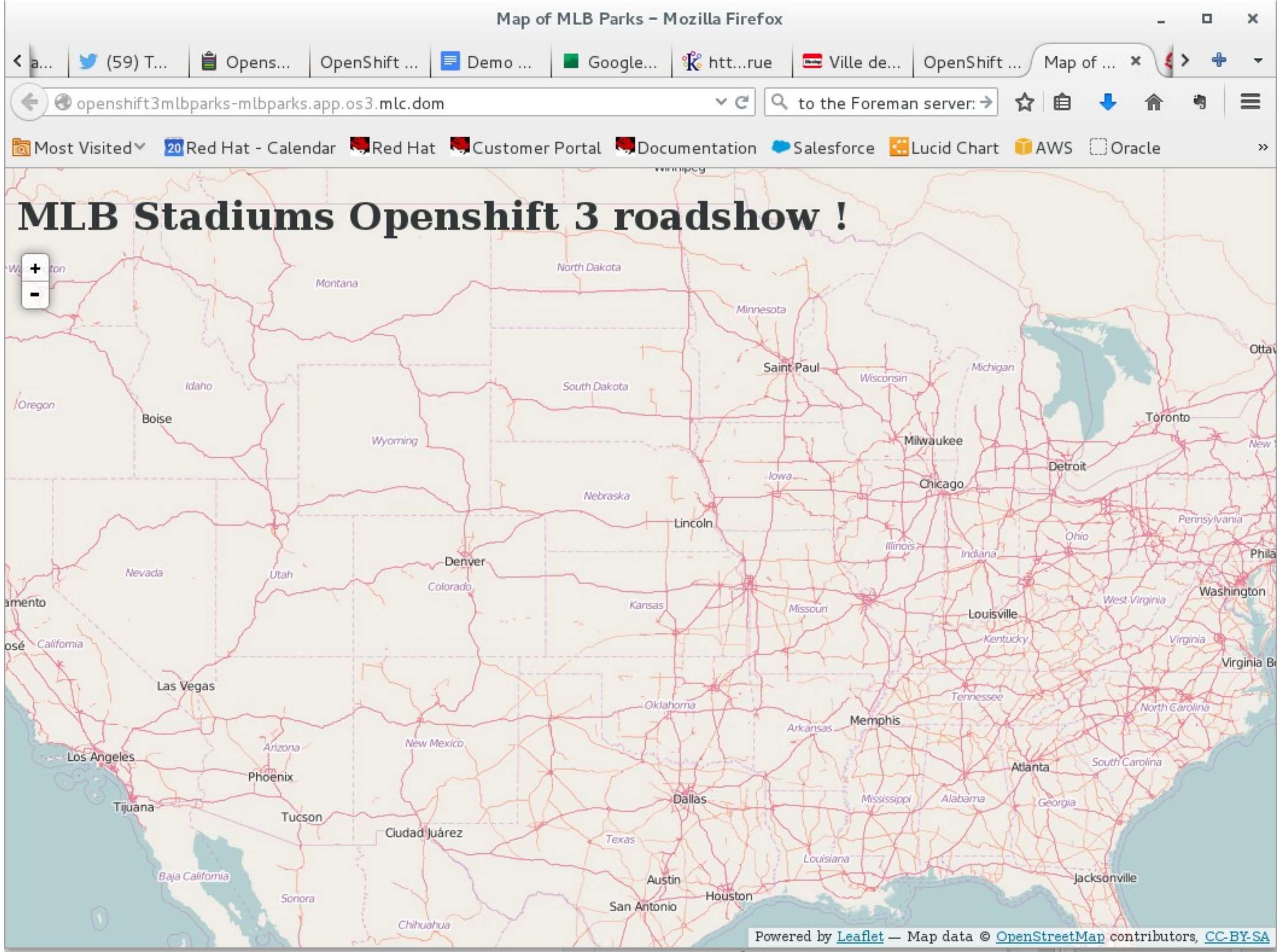
Ports: 8080/TCP, 8443/TCP

Select an object to see more details.

A **pod** contains one or more Docker containers that run together on a node, containing your application code.

A **service** groups pods and provides a common DNS name and an optional, load-balanced IP address to access them.

A **deployment** is an update to your application, triggered by a changed image or configuration.


Map of MLB Parks – Mozilla Firefox

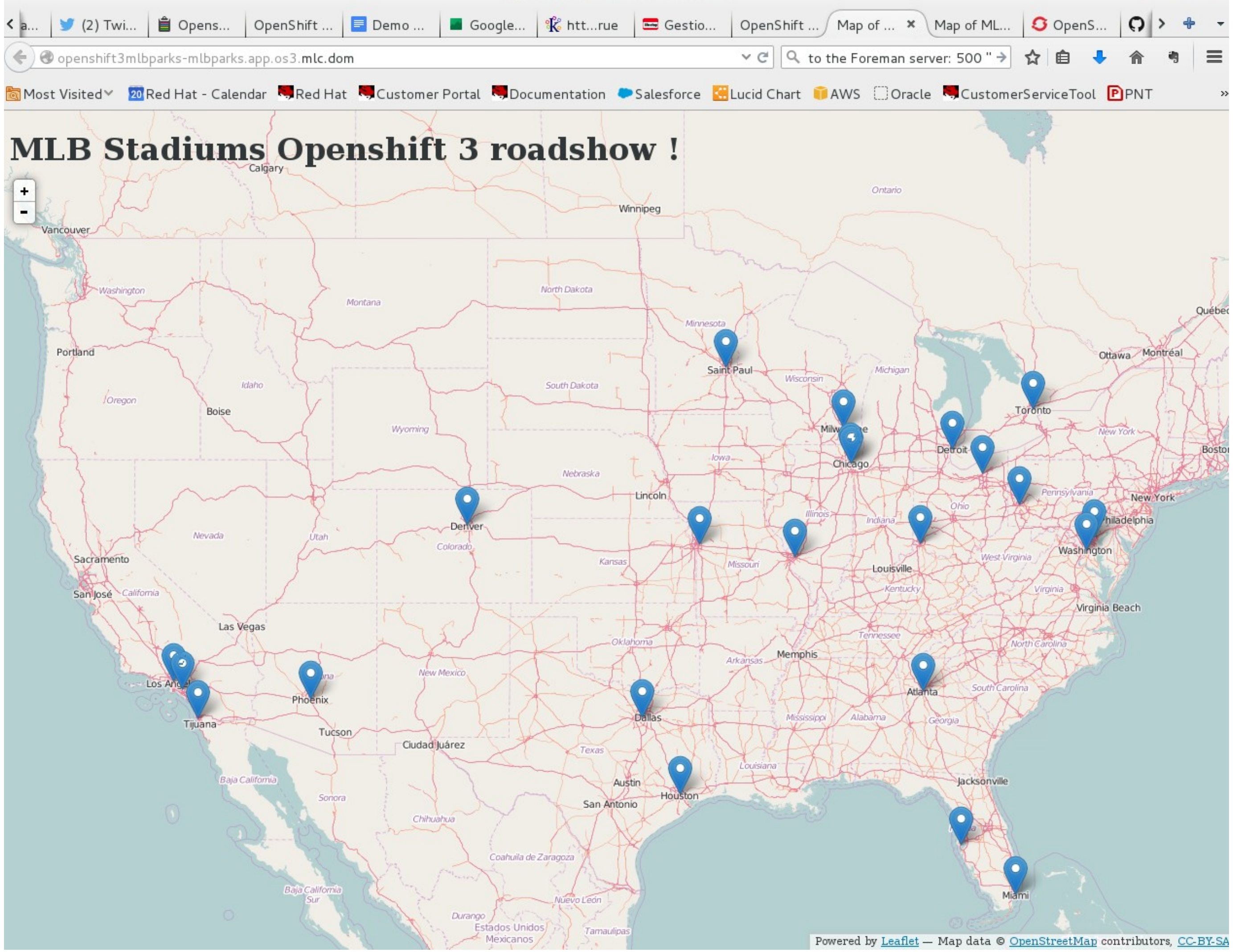
openshift3mlbparks-mlbparks.app.os3.mlc.dom

MLB Stadiums Openshift 3 roadshow !

Powered by Leaflet — Map data © OpenStreetMap contributors, CC-BY-SA

ENVIRONMENT VARIABLES + DC

```
[root@os3 ~]# oc new-app mongodb -e MONGODB_USER=mlbparks -e MONGODB_PASSWORD=mlbparks -e
MONGODB_DATABASE=mlbparks -e MONGODB_ADMIN_PASSWORD=mlbparks

[root@os3 ~]#  oc get dc
NAME                TRIGGERS                        LATEST
mongodb             ConfigChange, ImageChange       1
Openshift3mlbparks ConfigChange, ImageChange        1

# oc env dc openshift3mlbparks -e MONGODB_USER=mlbparks -e MONGODB_PASSWORD=mlbparks -e
MONGODB_DATABASE=mlbparks
deploymentconfig "openshift3mlbparks" updated

[root@os3 ~]# oc get dc
NAME                TRIGGERS                        LATEST
mongodb              ConfigChange, ImageChange   1
openshift3mlbparks   ConfigChange, ImageChange   2

DOCKER IMAGES IS NOW AVAILABLE
[root@os3 ~]# docker images
REPOSITORY                                          TAG             IMAGE ID          CREATED            VIRTUAL SIZE
172.30.177.161:5000/mlbparks/openshift3mlbparks              latest          80e9485fd5bb       30 minutes ago     958.9 MB
172.30.177.161:5000/mlbparks/openshift3mlbparks              <none>            80e9485fd5bb       30 minutes ago      958.9
MB
```

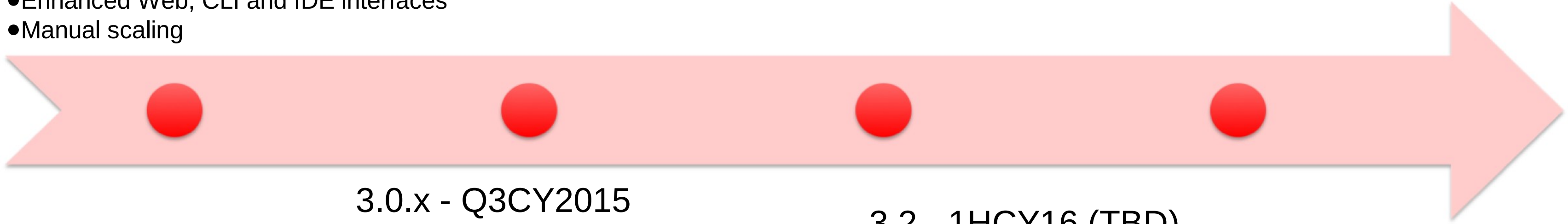


MLB Stadiums Openshift 3 roadshow !

OpenShift Product Roadmap Plan

3.0 - June 2015

- Docker container runtime & image packaging format
- Kubernetes orchestration & mgt.
- Source-to-Image & Docker builds
- JBoss EAP 6.4, JWS 3.0, A-MQ 6.2
- SCL images (Node, Python, PHP, Ruby...)
- Shared storage volumes for stateful apps
- Projects & team collaboration
- OAuth & enterprise auth integration (LDAP)
- Enhanced Web, CLI and IDE interfaces
- Manual scaling



3.0.x - Q3CY2015

- F5 & External Routing Examples
- Reference architectures
- Bug fixes

3.1 - Q4CY15

- CPU autoscaling *
- Integration Service / Fuse 6.x
- Decision Service / BRMS
- Cache Service / JDG
- Eclipse IDE completion
- Web/CLI UX enhancements
- SCL 2 image updates
- CloudForms 4.0 OSE Provider
- CPU/Memory Metrics Aggregation

- Additional storage plugins
- Networking enhancements
- ELK Log Aggregation
- CPU/Memory Overcommit
- HA Ref Arch/Enhancements
- Job Controller
- LDAP teams integration
- Jenkins Image / CI integration

3.2 - 1HCY16 (TBD)

- Mobile Service/Red Hat Mobile
- Autoscaling Enhancements
- CI/CD Pipelines
- Build Automation / Binary Deployment & ALM Integration
- Service Catalog
- Dev UX enhancements

- Idling
- Non-SNI routing
- OpenStack Neutron
- CloudForms Active Management
- Enterprise Registry
- Storage Enhancement
- Routing Enhancements

Merci !