



המוסד לביטוח לאומי

Log in IBM cloud

Welcome to
IBM Cloud

Start building immediately using
190+ unique services.

Create an IBM Cloud account

Get a \$200 credit when you upgrade

After you upgrade to a Pay-As-You-Go account, you can use the credit to try new services or scale your projects. The credit is valid for one month and can be used with any of our IBM Cloud offerings.

Learn more:

[Pricing](#) [Catalog](#) [Docs](#) [Status](#)

Log in to IBM Cloud

ID

IBMid ▼

☐ Remember me

[Forgot ID?](#)
[Forgot password?](#)

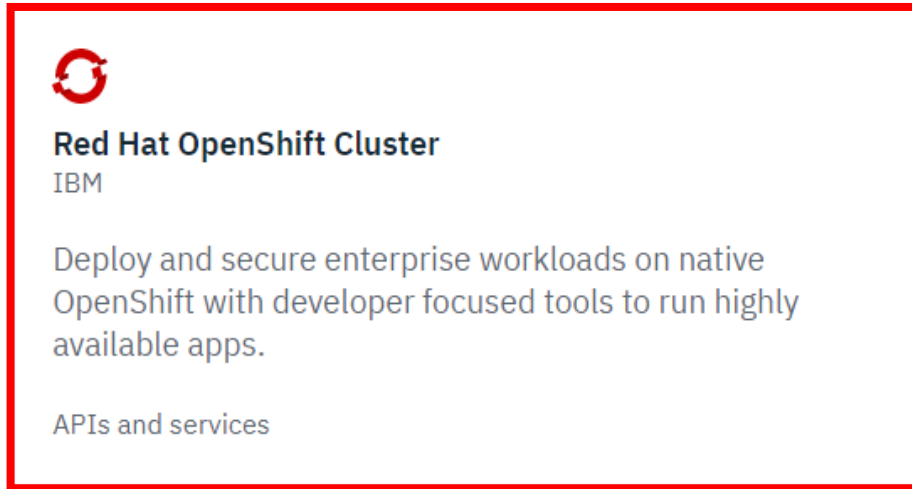
Continue

Create your OpenShift cluster

1. Click the Catalog menu



2. Click the Red Hat OpenShift cluster



3. Click the OpenShift cluster create button



Red Hat OpenShift Cluster

IBM

Create

Helpful links

[Creating a cluster and setting up your environment](#)

[View docs](#)

[Terms and conditions](#)

Overview

With Red Hat OpenShift on IBM Cloud, OpenShift developers have a fast and secure way to containerize and deploy enterprise workloads in Kubernetes clusters. OpenShift clusters build on Kubernetes container orchestration that offers consistency and flexibility in operations. Because IBM manages OpenShift Container Platform (OCP), you'll have more time to focus on your core tasks.

Features

- **OpenShift experience built on Kubernetes**

Use the OpenShift tools and APIs you already know for a single, consistent experience, even when working across hybrid environments or different cloud providers.

- **Heightened cluster and app security**

IBM provides security features to protect your cluster infrastructure, isolate your compute resources, encrypt data, and ensure security compliance in your container deployments. Further, OpenShift sets up strict Security Context Constraints for greater pod security by default.

- **Worldwide, continuous availability**

Deploy and scale workloads across the globe in all IBM Cloud multizone regions. OpenShift clusters include a managed master that is automatically spread across zones within the region for high availability.



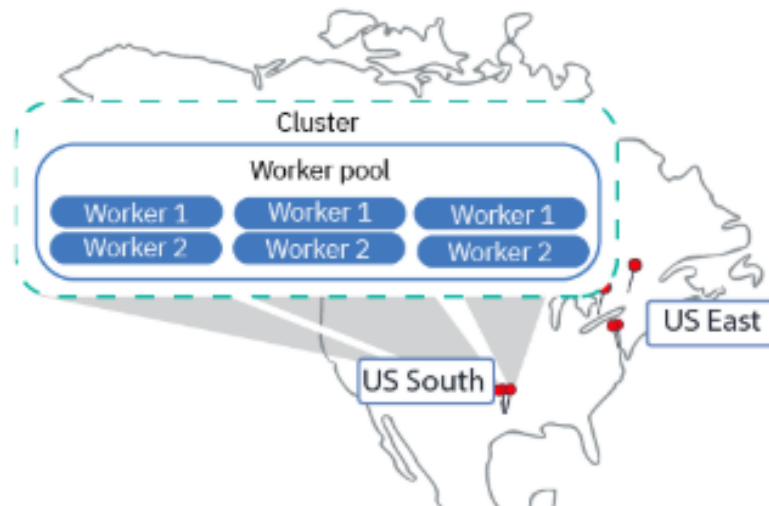
Select your Cluster Location

- A **single zone** cluster with multiple worker nodes in a worker pool.
- A **multizone** cluster that spreads worker nodes across zones within one region.
- Clusters with public network connectivity: Multiple clusters that are set up across zones or regions and that are connected via a global load balancer.

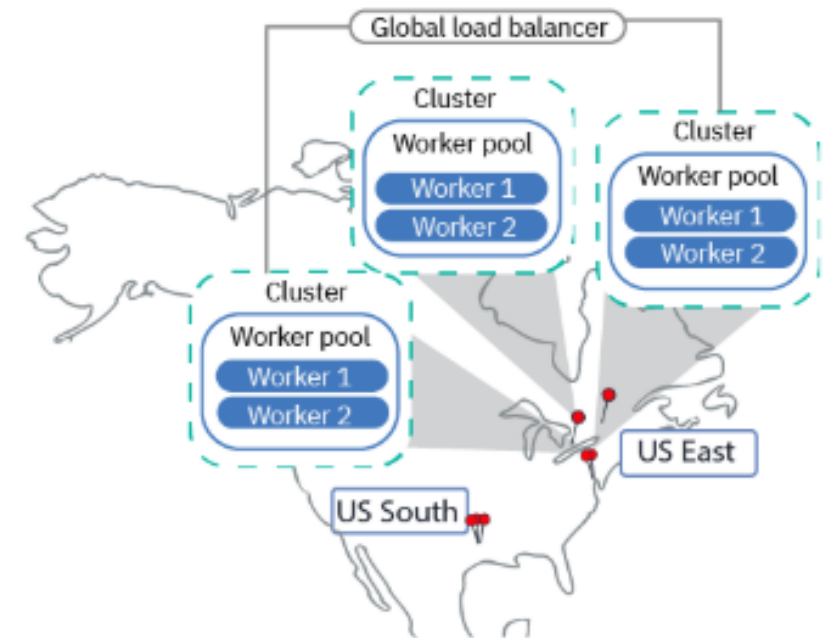
① Single zone cluster



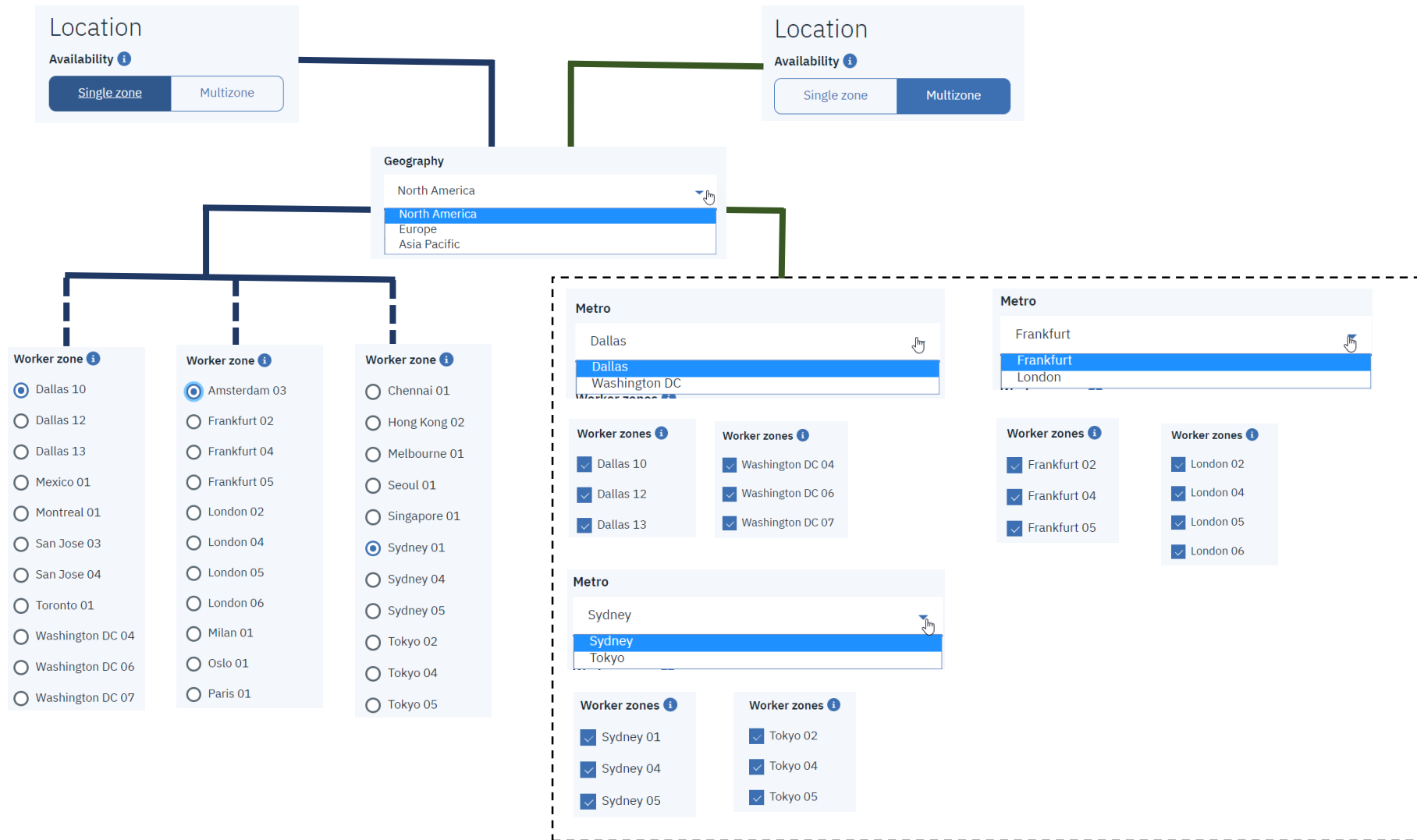
② Multizone cluster



③ Multiple clusters with global load balancer




Cluster high availability



Enable VLAN Spanning
To add multiple zones, you must [enable VLAN spanning](#). This allows worker nodes to communicate between zones. If you don't have the required permissions, contact your system administrator.

Select Cluster service endpoint

Master service endpoint 

Public endpoint only

Both private & public endpoints

Public endpoint only

Private endpoint only

Create Cluster metadata


Cluster Metadata

Cluster name

nii.dev1

Resource group


Default

Tags 

env:dev x

Default worker pool

Configure a set of worker nodes of the same flavor to create a default worker pool. Later, you can resize your worker pool to add or remove worker nodes. If you want a different flavor of worker node, you can create a new worker pool.

OpenShift version 

3.11.154

Latest, Stable, Default



Filter

Machine

- ☐ Bare Metal (11)
- ☐ Virtual - shared (15)
- ☐ Virtual - dedicated (9)

Use cases

- ☐ Balanced Cores and RAM (10)
- ☐ Extra local storage for SDS (4)
- ☐ GPU (2)
- ☐ RAM-intensive (15)
- ☐ Data-intensive (11)

Size

- ☐ Medium (18)
- ☐ Large (17)

Flavor 

4 vCPUs 16GB RAM

Virtual - shared
b3c.4x16
RHEL
25GB SSD primary disk
100GB SSD secondary disk
1Gbps network speed

\$0.27 / hr



8 vCPUs 32GB RAM

Virtual - shared
b3c.8x32
RHEL
25GB SSD primary disk
100GB SSD secondary disk
1Gbps network speed

\$0.50 / hr

8 vCPUs 64GB RAM

Virtual - shared
m3c.8x64
RHEL
25GB SSD primary disk
100GB SSD secondary disk
1Gbps network speed

\$0.54 / hr

16 vCPUs 16GB RAM

Virtual - shared
c3c.16x16
RHEL
25GB SSD primary disk
100GB SSD secondary disk
1Gbps network speed

\$0.66 / hr

16 vCPUs 32GB RAM

Virtual - shared
c3c.16x32
RHEL
25GB SSD primary disk
100GB SSD secondary disk
1Gbps network speed

\$0.77 / hr

16 vCPUs 64GB RAM

Virtual - shared
b3c.16x64
RHEL
25GB SSD primary disk
100GB SSD secondary disk
1Gbps network speed

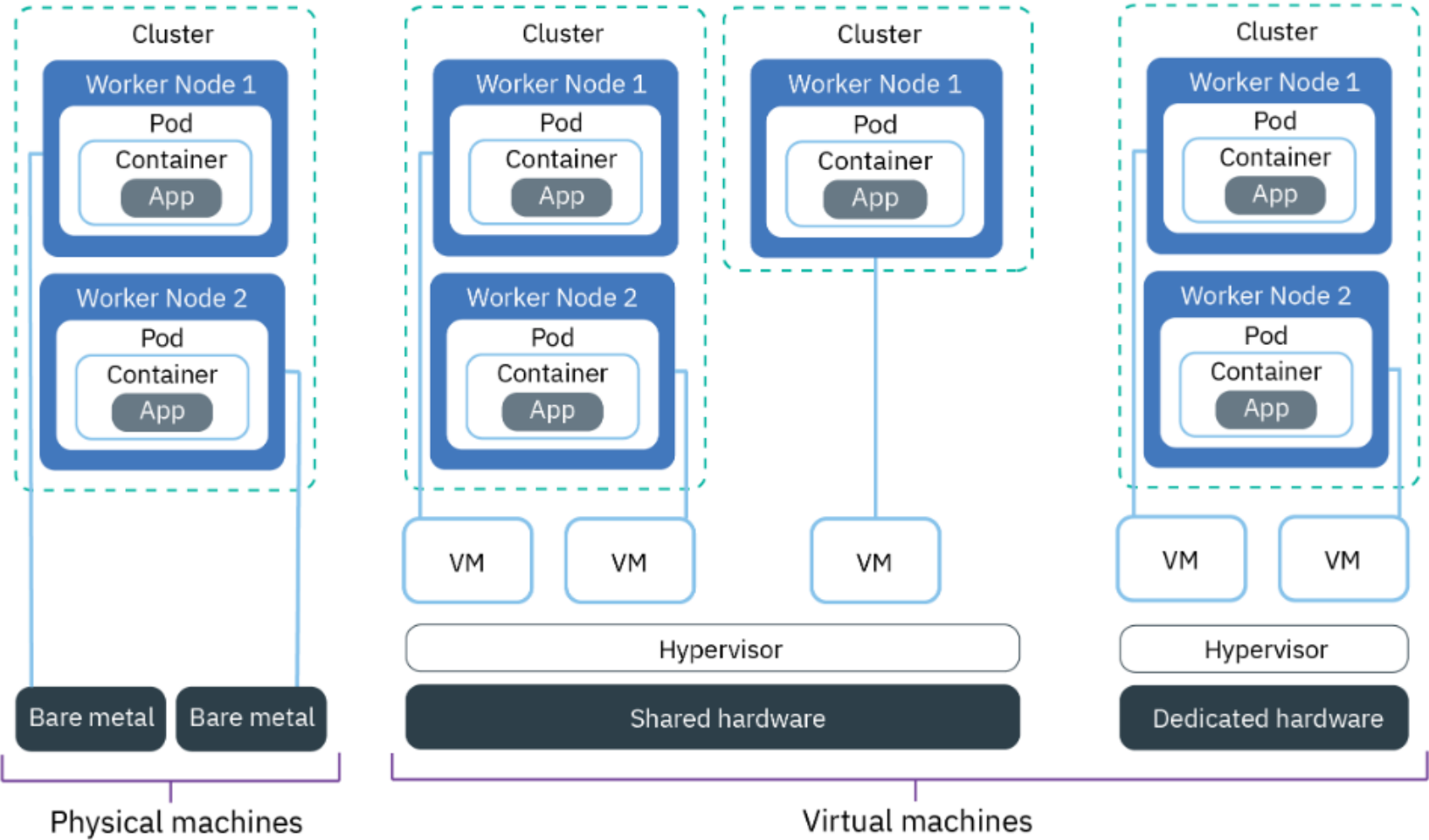
\$0.95 / hr

32 vCPUs 32GB RAM

32 vCPUs 64GB RAM

☒ Encrypt local disk

Cluster worker nodes



Cluster worker nodes

Worker nodes

2



Infrastructure permissions checker

✓ Permission requirements and suggestions satisfied ✓

- ✓ You have all the required and suggested virtual server permissions.
- ✓ You have all the required and suggested physical server permissions.
- ✓ You have all the required and suggested network permissions.
- ✓ You have all the required and suggested storage permissions.

For more information about assigning infrastructure permissions, see [the docs](#).

Create cluster

Order summary

b3c.4x16 - 4 vCPUs 16GB RAM

2 worker nodes \$0.70 / hr

IP allocation ⓘ \$16.00 / month

OCP license fee ⓘ \$800.00 / month

Total* **\$1,318.56 / month**
estimated

*Actual monthly total will vary with [tiered pricing](#) for the hourly worker nodes and the 30-day fixed [OCP license fee](#).

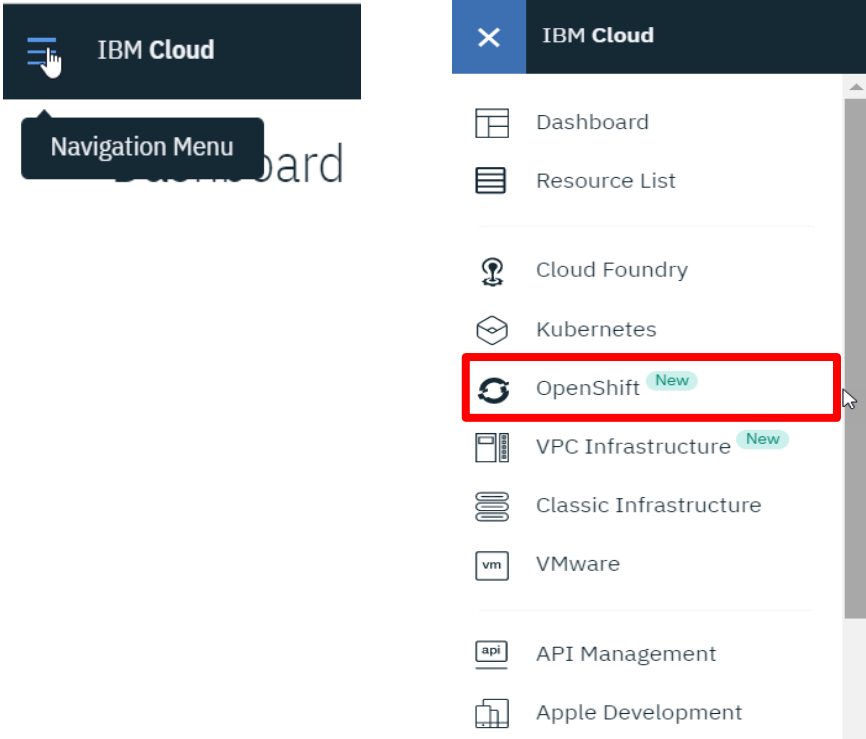
Additional charges for bandwidth might apply. [Learn more](#).

Create cluster

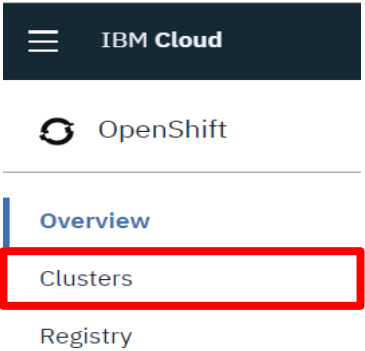
Add to estimate



Work with RHOCP



View your RHOCP clusters




Clusters

RESOURCE GROUP
All Resources ▼

LOCATION
All Locations ▼


Filter

Create cluster

Name	State	Location	Worker Count	Created ▼	Version	
nii.dev01	 Normal	Milan 01	2	10/27/2019, 10:34 AM	3.11.146_1528	...
Items per page: 10 ▼ 1–1 of 1 items				1 of 1 pages	<	1 ▼ >

Work with RHOCF clusters

Clusters / .dev01

 nii.dev01 Normal

[OpenShift web console](#) [Connect via CLI](#)

[Access](#) [Overview](#) [Worker Nodes](#) [Worker Pools](#) [DevOps](#) New

Summary

Cluster ID

bmqlcglf04h218clcqj0

Master status

Ready

Version

3.11.146_1528

Zones

mil01

Creator

yair_shaked@il.ibm.com

Created

10/27/2019, 10:34 AM

Ingress subdomain

nii.dev01.mil01.containers.appdomain.cloud

Resource group

Default

Key protect (Beta)

[Enable](#)

Image pull secrets

Enabled

Public service endpoint URL

<https://c103-eu-de.containers.cloud.ibm.com:30739> [Disable](#)

Private service endpoint URL

<https://c103-private.eu-de.containers.cloud.ibm.com:30739>

Worker Nodes 2

100%

Normal

2

Normal

0

Warning

0

Critical

0

Pending

Cluster Insights

22.0%

Total CPU Usage

36.3%

Total Memory Usage

User

System

Free

Used

Free

[Access](#)[Overview](#)[Worker Nodes](#)[Worker Pools](#)[DevOps](#) New

Worker Nodes


[Add worker pool](#) +

<input type="checkbox"/>	Name ▲	Status	Worker Pool	Zone	Private IP	Public IP	Version
> <input type="checkbox"/>	000002ca	● Normal	default	mil01	10.144.68.118	159.122.156.147	3.11.146_1527
> <input type="checkbox"/>	00000184	● Normal	default	mil01	10.144.68.112	159.122.156.152	3.11.146_1527
Items per page: 10 ▼ 1–2 of 2 items						1 of 1 pages	< 1 ▼ >

[Access](#)[Overview](#)[Worker Nodes](#)[Worker Pools](#)[DevOps](#) New

Worker Pools

[Add worker pool](#) +

Name ▲	Zones	Workers Per Zone	Actual / Declared Workers	Flavor	Labels
default	mil01	2	2 / 2	b3c.4x16	 ...
Items per page: 10 ▼ 1–1 of 1 items				1 of 1 pages	< 1 ▼ >

OC Developer Linux Cluster Login

OpenShift web console 



To deploy and manage containers,
launch the OpenShift web console.

OPENSIFT CONTAINER PLATFORM

Service Catalog

IAM#zlvd@il.ibm.com

Search Catalog

Browse Catalog

Deploy Image Import YAML / JSON Select from Project

All

Languages

Databases

Middleware

CI/CD

Other

Filter 40 Items

.NET

.NET Core

.NET

.NET Core + PostgreSQL (Persistent)

.NET

.NET Core Example

.NET

.NET Core Runtime Example

3scale-gateway

My Projects

+ Create Project

5 of 15 Projects

View All

default

created an hour ago

ibm-cert-store

created an hour ago

ibm-system

created an hour ago


kube-proxy-and-dns

created an hour ago

kube-public

created an hour ago

OC Developer Linux terminal

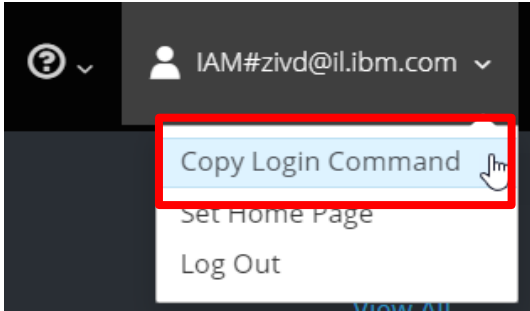


The screenshot displays the IBM Cloud Shell for Workshop interface. At the top, a dark blue header bar contains the text "IBM Cloud Shell for Workshop" on the left. On the right side of this bar, there is a red-outlined button with a document icon and a red-outlined dropdown menu labeled "Choose an account" with a downward arrow. To the right of the dropdown is a user profile icon. Below the header bar, a light gray area shows a user profile for "nii user" with the email "niiuser@mail.com" and a "Log out →" button. A large green arrow points downwards from this area to a terminal window below. The terminal window has a dark blue header bar with "IBM Cloud Shell for Workshop" on the left and "Ziv Dai's Account" with a dropdown arrow and a user icon on the right. Below the header bar, the terminal content shows a "Terminal" tab, a plus sign for a new tab, and a scroll bar on the right. The terminal text includes: "Welcome to IBM Cloud Shell!", "Image: 0.4.28-darklaunch.0.787", "Note: Only data stored in '/home/zivd' will be persisted.", "Type 'ibmcloud' to use IBM Cloud CLI.", and the prompt "zivd@cloudshell:~\$".

Password : ikslab

<https://workshop.shell.cloud.ibm.com>

Copy OC login key



Paste Login key

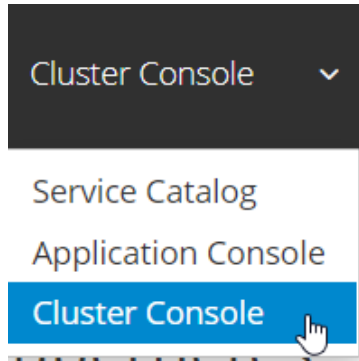
```
zivd@cloudshell:~$ oc login https://c103-e.eu-de.containers.cloud.ibm.com:30739 --token=91btTyrIfLfg6b6qbopubFHR_ZLe
Logged into "https://c103-e.eu-de.containers.cloud.ibm.com:30739" as "IAM#zivd@il.ibm.com" using the token provided.

You have access to the following projects and can switch between them with 'oc project <projectname>':

* default
  ibm-cert-store
  ibm-system
  jenkins-master
  kube-proxy-and-dns
  kube-public
  kube-service-catalog
  kube-system
  openshift
  openshift-ansible-service-broker
  openshift-console
  openshift-infra
  openshift-monitoring
  openshift-node
  openshift-template-service-broker
  openshift-web-console

Using project "default".
Welcome! See 'oc help' to get started.
zivd@cloudshell:~$
```

As a developer you can view your application configuration for OCP, enter your OCP dashboard and select cluster console



A deployment is an object in Kubernetes that lets you manage a set of identical pods.

Everyone running applications on Kubernetes cluster uses a deployment.

It's what you use to scale, roll out, and roll back versions of your applications.

With a deployment, you tell Kubernetes how many copies of a Pod you want running. The deployment takes care of everything else.

Deployment

"I want three of my
Node.js app Pods running"

kubectl commands

<https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands>

GETTING STARTED

create

get

run

expose

delete

diff

APP MANAGEMENT

apply

annotate

autoscale

convert

edit

label

patch

replace

GETTING STARTED

This section contains the most basic commands for getting a workload running on your cluster.

- `run` will start running 1 or more instances of a container image on your cluster.
- `expose` will load balance traffic across the running instances, and can create a HA proxy for accessing the containers from outside the cluster.

Once your workloads are running, you can use the commands in the [WORKING WITH APPS](#) section to inspect them.

create

Create a resource from a file or from stdin.

example

Create a pod using the data in pod.json.



Lab 1 : Create Hello-World pod

run will start running 1 or more instances of a container image on your cluster.

```
$ kubectl run NAME --image=image [--env="key=value"] [--port=port] [--replicas=replicas] [--dry-run=bool] [--overrides=inline-json] [--command] -- [COMMAND] [args...]
```

1. Run hello world

`docker image ls hello`

```
ibm@icpl:~$ docker image ls hello
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
------------	-----	----------	---------	------

```
ibm@icpl:~$
```

`oc get deploy`

```
root@icpl:~# kubectl get deploy
```

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
nginx	1	1	1	1	2d
nginx-demo-nginxref-nginx	1	1	1	1	1d

```
root@icpl:~#
```

`oc run hello-world --image=paulbouwer/hello-kubernetes:1.4 --port=80`

```
zivd@cloudshell:~$ oc run hello-world --image=paulbouwer/hello-kubernetes:1.4 --port=80
deploymentconfig.apps.openshift.io/hello-world created
zivd@cloudshell:~$
```

```
oc run hello-world --image=paulbouwer/hello-kubernetes:1.4 --port=80
```

Pod name



Image name on docker hub

Pod port



<https://github.com/paulbouwer/hello-kubernetes>

Branch: master hello-kubernetes / server.js

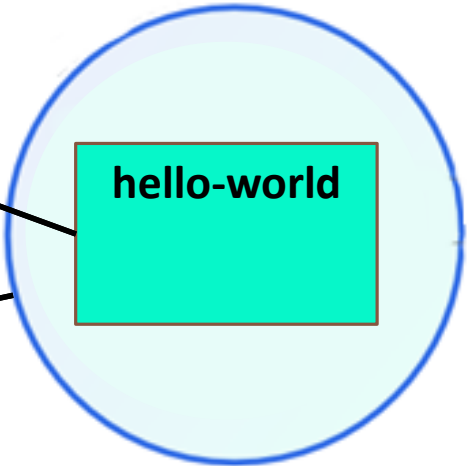
```
// Configuration
var port = process.env.PORT || 8080;
var message = process.env.MESSAGE || "Hello world!";
```

IBM Cloud Private view

```
npm info using node@v8.1.0
npm info using npm@5.0.3
npm info it worked if it ends with ok
npm info lifecycle hello-kubernetes@1.4.0~prestart: hello-kubernetes@1.4.0
npm info lifecycle hello-kubernetes@1.4.0~start: hello-kubernetes@1.4.0
> node server.js
> hello-kubernetes@1.4.0 start /usr/src/app
Listening on port 8080 - http://hello-world-7ff856b5d5-4f546:8080
```

Container:8080

Pod:80



NAME	IMAGE	PORT	STATE
hello-world	paulbouwer/hello-kubernetes:1.4	80/TCP	Running



OCP – View Hello-World Pod

OPENSIFT CONTAINER PLATFORM

Cluster Console

IAM#zivd@il.ibm.com

Home

Workloads

Pods

Deployments

Deployment Cor

Stateful Sets

Secrets

Config Maps

Cron Jobs

Jobs

Daemon Sets

Replica Sets

Project: default

Pods

Create Pod

hello

6 Running

0 Pending

0 Terminating

0 CrashLoopBackOff

1 Completed

0 Failed

0 Unknown

NAME ↑	NAMESPACE	POD LABELS	NODE	STATUS	READINESS
<div>⚙️ P hello-v</div>	cn6c <div>NS default</div>	<div>deploym...=hello-wor...</div> <div>deploymen... =hello-...</div> <div>run=hello-world</div>	<div>N 10.144.68.112</div>	Running	Ready

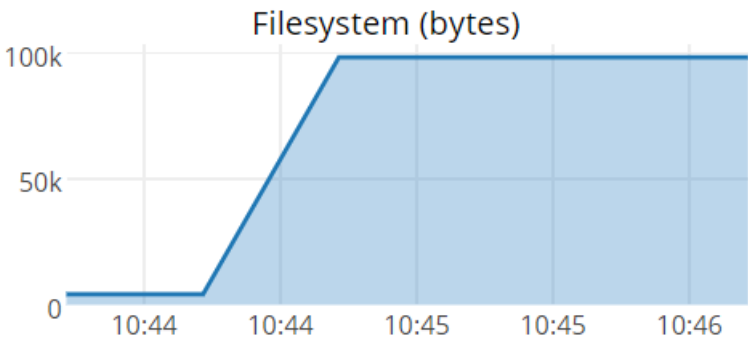
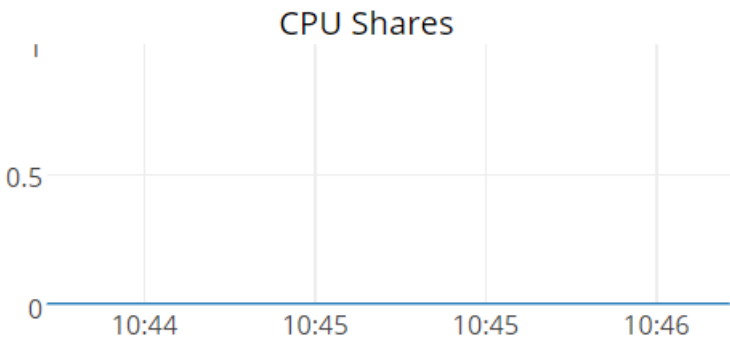
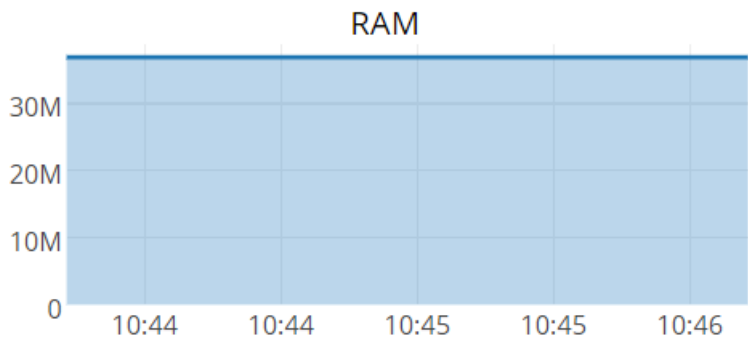
P

hello-world-1-8cn6c

Actions ▾

- Overview
- YAML
- Environment
- Logs
- Events
- Terminal

Pod Overview



NAME

hello-world-1-8cn6c

STATUS

Running

NAMESPACE

NS

 default

RESTART POLICY

Always Restart



View resources

TYPE

oc get pods -o wide

```
zivd@cloudshell:~$ oc get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE
docker-registry-69c6bdcdb5-2k5lf	1/1	Running	0	9d	172.30.197.181	10.144.68.112	<none>	
docker-registry-69c6bdcdb5-2s5w2	1/1	Running	0	9d	172.30.197.183	10.144.68.112	<none>	
hello-world-1-8cn6c	1/1	Running	0	7m	172.30.197.191	10.144.68.112	<none>	
ibm-content-mgmt-script-pod-38f6f48e-21c4-45c3-8c85-0d011b63cb21-global-preinstall	0/1	Completed	0	63d	172.30.18.81	10.144.68.118	<none>	
registry-console-6b57d65b57-n8pzz	1/1	Running	0	51d	172.30.18.84	10.144.68.118	<none>	
router-697ff67754-72jnn	1/1	Running	0	9d	172.30.197.180	10.144.68.112	<none>	
router-697ff67754-7b44r	1/1	Running	0	9d	172.30.197.182	10.144.68.112	<none>	

```
zivd@cloudshell:~$
```

NAME

oc get pods -l run=hello-world

```
zivd@cloudshell:~$ oc get pods -l run=hello-world
```

NAME	READY	STATUS	RESTARTS	AGE
hello-world-1-8cn6c	1/1	Running	0	8m

```
zivd@cloudshell:~$
```

oc describe pod/hello-world-1-8cn6c

```
zivd@cloudshell:~$ oc describe pod/hello-world-1-8cn6c
```

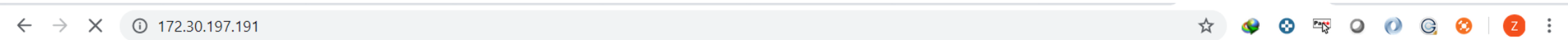
```
Name:          hello-world-1-8cn6c
Namespace:     default
Priority:       0
PriorityClassName: <none>
Node:          10.144.68.112/10.144.68.112
Start Time:    Thu, 16 Jan 2020 08:43:07 +0000
Labels:        deployment=hello-world-1
                deploymentconfig=hello-world
                run=hello-world
Annotations:   openshift.io/deployment-config.latest-version=1
                openshift.io/deployment-config.name=hello-world
                openshift.io/deployment.name=hello-world-1
                openshift.io/scc=anyuid
Status:        Running
IP:            172.30.197.191
Controlled By: ReplicationController/hello-world-1
Containers:
  hello-world:
    Container ID:  cri-o://7c55bb2df9d93ee46d1149ed24b23d26b5946b6dc602c72f503f88be63bf15ba
    Image:         paulbouwer/hello-kubernetes:1.4
    Image ID:      docker.io/paulbouwer/hello-kubernetes@sha256:a9fc93acfb734827a72107bf7f759745a66ea61758863c094c36e5f4f4b810b
    Port:          80/TCP
    Host Port:     0/TCP
    State:         Running
      Started:    Thu, 16 Jan 2020 08:43:09 +0000
    Ready:        True
```

Events:

Type	Reason	Age	From	Message
Normal	Scheduled	10m	default-scheduler	Successfully assigned default/hello-world-1-8cn6c to 10.144.68.112
Normal	Pulled	10m	kubelet, 10.144.68.112	Container image "paulbouwer/hello-kubernetes:1.4" already present on machine
Normal	Created	10m	kubelet, 10.144.68.112	Created container
Normal	Started	10m	kubelet, 10.144.68.112	Started container

```
zivd@cloudshell:~$ oc describe pod/hello-world-1-8cn6c
```

The created POD is NOT connected to the outside world yet, therefore there is no access



This site can't be reached

172.30.197.191 took too long to respond.

Try:

- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_TIMED_OUT

Reload

Details



Lab 2 : expose Hello-world

Take a replication controller, service or pod and expose it as a new Kubernetes Service

```
$ kubectl expose (-f FILENAME | TYPE NAME) [--port=port] [--protocol=TCP|UDP] [--target-port=number-or-name] [--name=name] [--external-ip=external-ip-of-service] [--type=type]
```

```
oc expose pod/hello-world-1-8cn6c --type=LoadBalancer --port=80 --target-port=8080
```

```
zivd@cloudshell:~$ oc expose pod/hello-world-1-8cn6c --type=LoadBalancer --port=80 --target-port=8080
service/hello-world-1-8cn6c exposed
zivd@cloudshell:~$
```

oc get services

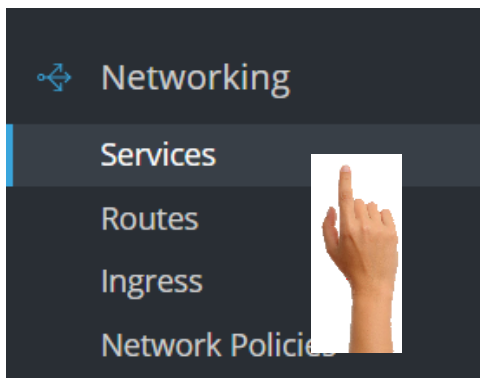
```
zivd@cloudshell:~$ oc get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
docker-registry	ClusterIP	172.21.223.28	<none>	5000/TCP	81d
hello-kubernetes	LoadBalancer	172.21.214.99	159.122.163.36	80:32267/TCP	6d
hello-world-1-8cn6c	LoadBalancer	172.21.254.36	159.122.163.37	80:32466/TCP	2m
kubernetes	ClusterIP	172.21.0.1	<none>	443/TCP,53/UDP,53/TCP	81d
registry-console	ClusterIP	172.21.201.236	<none>	9000/TCP	81d
router	LoadBalancer	172.21.160.128	159.122.163.34	80:32485/TCP,443:32193/TCP	81d

```
zivd@cloudshell:~$
```



ICP – View Hello-World Service



Deployment Configs
Stateful Sets
Secrets
Config Maps
Cron Jobs
Jobs
Daemon Sets
Replica Sets
Replication Controllers
HPAs
Networking
Services
Routes

Project: default ▾

Services

Create Service

Filter Services by name...

NAME ↑	NAMESPACE	LABELS	POD SELECTOR	LOCATION
docker-registry	default	docker-registry=default	docker-registry=default	172.21.223.28:5000
hello-kubernetes	default	No labels	app=hello-kubernetes	172.21.214.99:80
hello-world-1-8cr	default	deployment=hello-world-1 deploymentconfig=hello-world run=hello-world	deployment=hello-world-1, deploymentconfig=hello-world, run=hello-world	172.21.254.36:80

```
oc expose deployment hello-world --type=LoadBalancer --port=80 --target-port=8080
```



Deploy name

Service type

Pod port

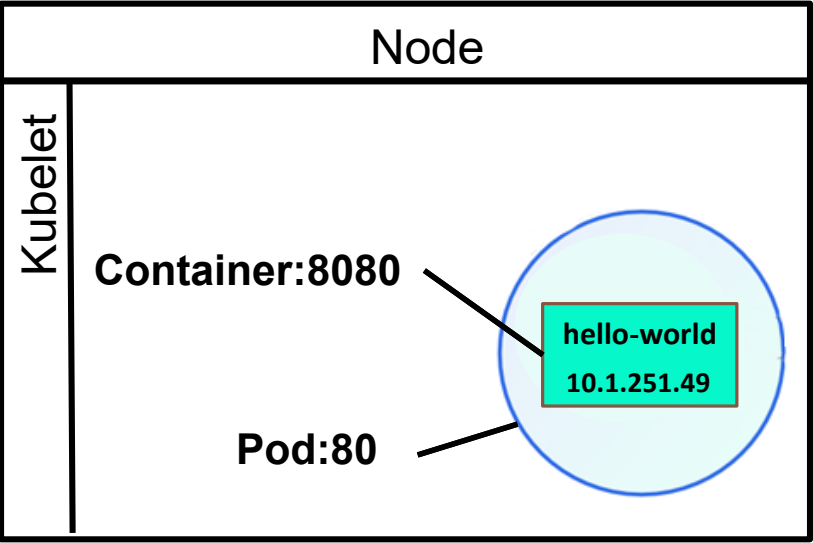
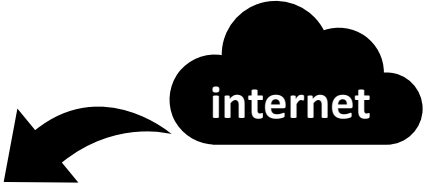
Container port

OCP view

Type	Location
External Load Balancer	159.122.163.37
Ingress point(s) of load balancer	

SERVICE PORT MAPPING			
Name	Port	Protocol	Pod Port or Name
-	S 80	TCP	P 8080
Node Port			

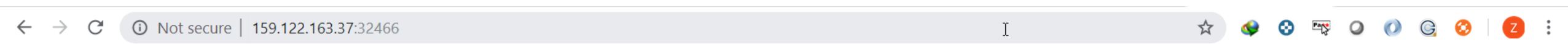
Service:NodePort 32422/NodePort



oc describe service hello-world

```
zivd@cloudshell:~$ oc describe service hello-world
Name:             hello-world-1-8cn6c
Namespace:        default
Labels:           deployment=hello-world-1
                  deploymentconfig=hello-world
                  run=hello-world
Annotations:      <none>
Selector:         deployment=hello-world-1,deploymentconfig=hello-world,run=hello-world
Type:             LoadBalancer
IP:              172.21.254.36
LoadBalancer Ingress: 159.122.163.37
Port:             <unset> 80/TCP
TargetPort:       8080/TCP
NodePort:         <unset> 32466/TCP
Endpoints:        172.30.197.191:8080
Session Affinity: None
External Traffic Policy: Cluster
Events:
  Type    Reason              Age   From                Message
  ----    -
  Normal  EnsuringLoadBalancer 9m    service-controller  Ensuring load balancer
  Normal  EnsuredLoadBalancer  9m    service-controller  Ensured load balancer

zivd@cloudshell:~$
```



Hello world!

pod: hello-world-1-8cn6c
node: Linux (3.10.0-1062.1.2.el7.x86_64)

```
oc get all -l run=hello-world
```


oc get pod/hello-world -o yaml

or

oc get pod/hello-world -o json

```
zivd@cloudshell:~$ oc get pod/hello-world-1-8cn6c -o yaml
apiVersion: v1
kind: Pod
metadata:
  annotations:
    openshift.io/deployment-config.latest-version: "1"
    openshift.io/deployment-config.name: hello-world
    openshift.io/deployment.name: hello-world-1
    openshift.io/scc: anyuid
  creationTimestamp: 2020-01-16T08:43:07Z
  generateName: hello-world-1-
  labels:
    deployment: hello-world-1
    deploymentconfig: hello-world
    run: hello-world
  name: hello-world-1-8cn6c
  namespace: default
  ownerReferences:
  - apiVersion: v1
    blockOwnerDeletion: true
    controller: true
    kind: ReplicationController
    name: hello-world-1
    uid: 364d00a8-383c-11ea-b2f1-365a6ef74f50
  resourceVersion: "19426055"
  selfLink: /api/v1/namespaces/default/pods/hello-world-1-8cn6c
  uid: 3804c9c0-383c-11ea-a0b7-8abcd7cd5f32
spec:
  containers:
```



Lab 4 : Create deployment using yaml

Copy and create new directory in your local environment and create the following YAML files:

deploy.yml

```
Terminal x +
GNU nano 2.5.3
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: hello-kubernetes
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: hello-kubernetes
    spec:
      containers:
      - name: hello-kubernetes
        image: paulbouwer/hello-kubernetes:1.4
        ports:
        - containerPort: 8080
```

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: hello-kubernetes
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: hello-kubernetes
    spec:
      containers:
      - name: hello-kubernetes
        image: paulbouwer/hello-kubernetes:1.4
        ports:
        - containerPort: 8080
```



deploy.yaml

```
apiVersion: extensions/v1beta1
```

Version of the Kubernetes API

```
kind: Deployment
```

object you want to create

```
metadata:
```

```
  name: hello-kubernetes
```

Pod name

```
spec:
```

```
  replicas: 1
```

Container specification

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: hello-kubernetes
```

```
    spec:
```

```
      containers:
```

```
        - name: hello-kubernetes
```

```
          image: paulbouwer/hello-kubernetes:1.4
```

```
          ports:
```

```
            - containerPort: 8080
```

oc apply -f deploy.yml

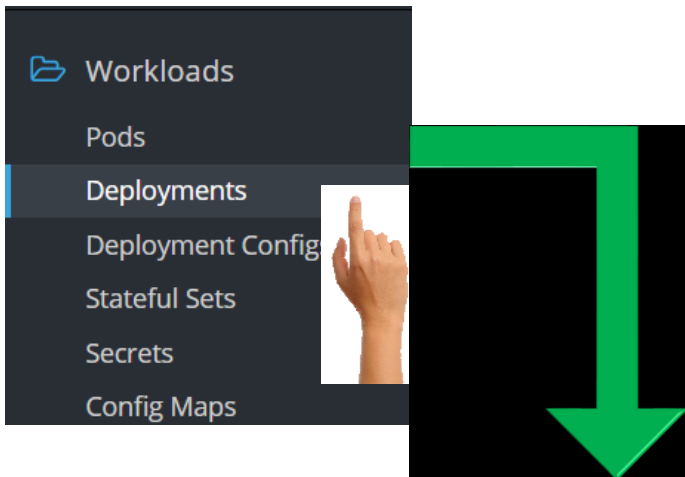
```
zivd@cloudshell:~/lab$ oc apply -f deploy.yml
deployment.extensions/hello-kubernetes created
zivd@cloudshell:~/lab$
```

oc get deploy -l app=hello-kubernetes

```
zivd@cloudshell:~$ oc get deploy -l app=hello-kubernetes
NAME           DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
hello-kubernetes 3         3         3            3           4m
zivd@cloudshell:~$
```



ICP – View hello-kubernetes Deploy



Deployments

Create Deployment

Filter Deployments by name...

NAME ↑	NAMESPACE	LABELS	STATUS	POD SELECTOR
docker-registry	default	docker-registry=default	2 of 2 pods	docker-registry=default
hello-kubernet	default	app=hello-kubernet	3 of 3 pods	app=hello-kubernet
registry-console	default	name=registry-console	1 of 1 pods	name=registry-console
router	default	router=router	2 of 2 pods	router=router

Deployment Overview

DESIRED COUNT 3 pods >	UP-TO-DATE COUNT 3 pods	MATCHING PODS 3 pods	{ 3 available 0 unavailable
---------------------------	----------------------------	-------------------------	-----------------------------------

NAME

hello-kubernetes

UPDATE STRATEGY

RollingUpdate

NAMESPACE

NS default

MAX UNAVAILABLE

1 of 3 pods

LABELS

app=hello-kubernetes

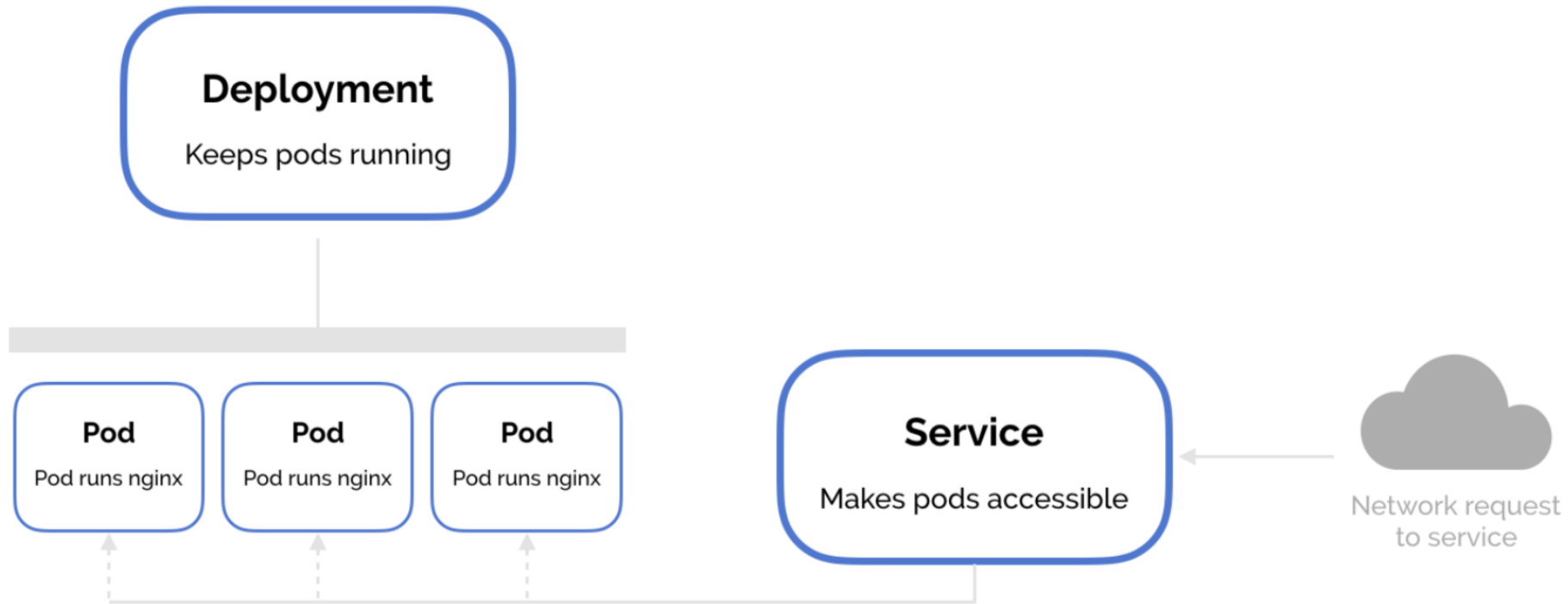
MAX SURGE

1 greater than 3 pods

POD SELECTOR

PROGRESS DEADLINE

A service is used to allow network access to a set of pods.



service.yml

Terminal × +

GNU nano 2.5.3

```
apiVersion: v1
kind: Service
metadata:
  name: hello-kubernetes
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 8080
  selector:
    app: hello-kubernetes
```

```
apiVersion: v1
kind: Service
metadata:
  name: hello-kubernetes
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 8080
  selector:
    app: hello-kubernetes
```

```
drwxrwxr-x 2 zivd zivd 4096 Jan  9 13:46 ./
drwxrwxrwx 6 root root 4096 Jan  9 13:43 ../
-rw-rw-r-- 1 zivd zivd  341 Jan  9 13:44 deploy.yml
-rw-rw-r-- 1 zivd zivd  173 Jan  9 13:46 service.yml
zivd@cloudshell:~/lab$
```



```
oc apply -f service.yml
```

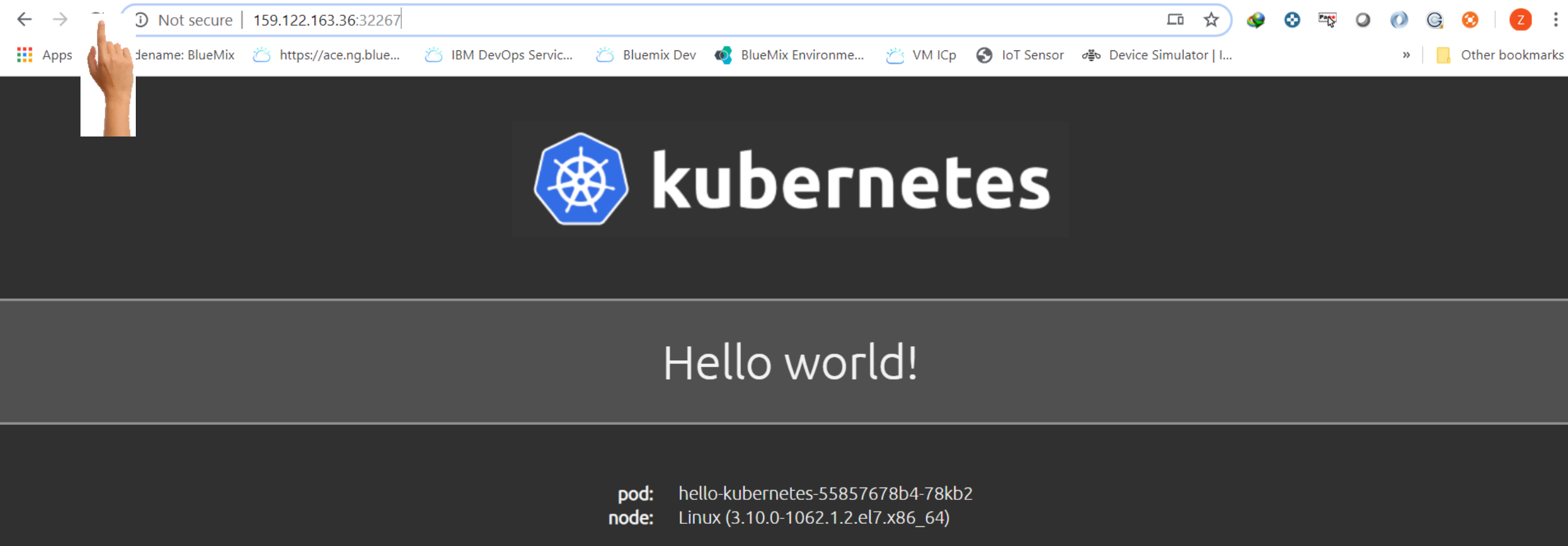
```
zivd@cloudshell:~/lab$ oc apply -f service.yml
service/hello-kubernetes unchanged
zivd@cloudshell:~/lab$
```

```
oc get service -l app=hello-kubernetes
```

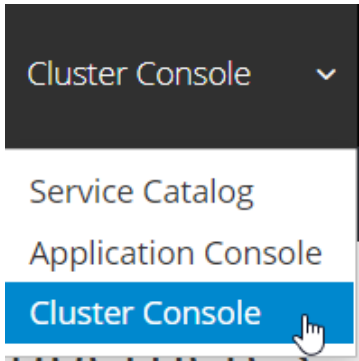
```
zivd@cloudshell:~$ oc get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
docker-registry	ClusterIP	172.21.223.28	<none>	5000/TCP	81d
hello-kubernetes	LoadBalancer	172.21.214.99	159.122.163.36	80:32267/TCP	6d
hello-world-1-8cn6c	LoadBalancer	172.21.254.36	159.122.163.37	80:32466/TCP	30m
kubernetes	ClusterIP	172.21.0.1	<none>	443/TCP,53/UDP,53/TCP	81d
registry-console	ClusterIP	172.21.201.236	<none>	9000/TCP	81d
router	LoadBalancer	172.21.160.128	159.122.163.34	80:32485/TCP,443:32193/TCP	81d

Open your browser and copy service external-ip with port (for my example it is) <http://159.122.163.36:32267/>



You can view your application configuration for OCP, enter your OCP dashboard and select cluster console



Open workloads→deployments

Workloads

Pods

Deployments

Deployment Configs

Stateful Sets

Secrets

Config Maps

Cron Jobs

Create Deployment

Filter Deployments by name...

NAME ↑	NAMESPACE	LABELS	STATUS	POD SELECTOR
docker-registry	default	docker-registry=default	2 of 2 pods	docker-registry=default
hello-kubernetes	default	app=hello-kubernetes	3 of 3 pods	app=hello-kubernetes
registry-console	default	name=registry-console	1 of 1 pods	name=registry-console
router	default	router=router	2 of 2 pods	router=router

Click hello-kubernetes

Project: default ▾

D

hello-kubernetes

Actions ▾

OverviewYAMLPodsEnvironmentEvents

Deployment Overview

DESIRED COUNT

3 pods >

UP-TO-DATE COUNT

3 pods

MATCHING PODS

3 pods

{

3 available

0 unavailable

}

NAME

hello-kubernetes

UPDATE STRATEGY

RollingUpdate

NAMESPACE

NS

default

MAX UNAVAILABLE

1 of 3 pods

LABELS

app=hello-kubernetes

MAX SURGE

1 greater than 3 pods

POD SELECTOR

Q

app=hello-kubernetes

PROGRESS DEADLINE

10m 0s

NODE SELECTOR

No selector

MIN READY SECONDS

Not Configured

ANNOTATIONS

2 Annotations >

STATUS

Active

CREATED AT

🕒

Oct 28, 10:57 am

Containers

NAME	IMAGE	RESOURCE LIMITS	PORTS
hello-kubernetes	paulbouwer/hello-kubernetes:1.4	-	8080/TCP

Conditions

TYPE	STATUS	UPDATED	REASON	MESSAGE
Available	True	🕒 Oct 28, 10:57 am	MinimumReplicasAvailable	Deployment has minimum availability.
Progressing	True	🕒 Oct 28, 10:57 am	NewReplicaSetAvailable	ReplicaSet "hello-kubernetes-55857678b4" has successfully progressed.

View YAML created at OCP by clicking the YAML TAB

OverviewYAMLPodsEnvironmentEvents

```
1 kind: Deployment
2 apiVersion: apps/v1
3 metadata:
4   annotations:
5     deployment.kubernetes.io/revision: '1'
6   kubectrl.kubernetes.io/last-applied-configuration: >
7     ([{"apiVersion":"extensions/v1beta1","kind":"Deployment","metadata":{"annotations":{},"name":"hello-kubernetes","namespace":"default"},"spec":{"replicas":3,"template":{"metadata":{"labels":{"app":"hello-kubernetes"}},"spec":{"containers":[{"image":"paulbouwer
8     /hello-kubernetes:1.4","name":"hello-kubernetes","ports":[{"containerPort":8080}]}}}}])
9 selfLink: /apis/apps/v1/namespaces/default/deployments/hello-kubernetes
10 resourceVersion: '245827'
11 name: hello-kubernetes
12 uid: f41169c2-f968-11e9-b486-a6476c637043
13 creationTimestamp: '2019-10-28T08:57:21Z'
14 generation: 1
15 namespace: default
16 labels:
17   app: hello-kubernetes
18 spec:
19   replicas: 3
20   selector:
21     matchLabels:
22       app: hello-kubernetes
23   template:
24     metadata:
25       creationTimestamp: null
26     labels:
27       app: hello-kubernetes
28     spec:
29       containers:
30         - name: hello-kubernetes
31           image: paulbouwer/hello-kubernetes:1.4
32           ports:
33             - containerPort: 8080
34               protocol: TCP
35           resources: {}
36           terminationMessagePath: /dev/termination-log
37           terminationMessagePolicy: File
38           imagePullPolicy: IfNotPresent
39 restartPolicy: Always
40 terminationGracePeriodSeconds: 30
41 dnsPolicy: ClusterFirst
42 securityContext: {}
43 schedulerName: default-scheduler
44 strategy:
45   type: RollingUpdate
```

View Deployments running PODs on OCP by clicking the Pods TAB

hello-kubernetesActions

OverviewYAMLPodsEnvironmentEvents

Filter Pods by name...

3 Running0 Pending0 Terminating0 CrashLoopBackOff0 Completed0 Failed0 Unknown

NAME ↑	NAMESPACE	POD LABELS	NODE	STATUS	READINESS
hello-kubernetes-55857678b4-2k9m9	default	app=hello-kubernetes pod-template-hash=1141323460	10.144.68.112	Running	Ready
hello-kubernetes-55857678b4-4xgbb	default	app=hello-kubernetes pod-template-hash=1141323460	10.144.68.118	Running	Ready
hello-kubernetes-55857678b4-tbtrh	default	app=hello-kubernetes pod-template-hash=1141323460	10.144.68.118	Running	Ready

View Pod info

hello-kubernetes-55857678b4 > Pod Details

P hello-kubernetes-55857678b4-2k9m9

Actions ▾

Overview

YAML

Environment

Logs

Events

Terminal

Pod Overview

RAM

CPU Shares

Filesystem (bytes)

NAME

hello-kubernetes-55857678b4-2k9m9

NAMESPACE

NS

 default

LABELS

app=hello-kubernetes

pod-template-hash=1141323460

ANNOTATIONS

1 Annotation >

NODE SELECTOR

Q

 node-role.kubernetes.io/compute=true

CREATED AT

2023-08-08 11:00:00

STATUS

Running

RESTART POLICY

Always Restart

POD IP

172.30.197.161

NODE

N

 10.144.68.112

View container info

C hello-kubernetes

Container Overview

STATE

running

ID

cri-oi//f6a835682aae29fa2497018610418107889ce0486a6bb1e236c557b5a8

RESTARTS

0

RESOURCE REQUESTS

-

RESOURCE LIMITS

-

Image Details

IMAGE

paulbouwer/hello-kubernetes

IMAGE VERSION/TAG

1.4

COMMAND

-

ARGS

-

PULL POLICY

Pull If Needed

Network

NODE

N

 10.144.68.112

POD IP

172.30.197.161