IBM Cloud private 2.1.0.1 Proof of Technology

Index

Lab 1: Deploy and expose an Application using IBM Cloud Private console	4
1. Login to the console	4
2. Deploy a new application (K8s Deployment)	5
3. Expose the application (K8s Service)	9
3.1 ClusterIP	9
3.2 NodePort	13
3.3 Ingress	17
Lab 2: Deploy and expose an Application using K8s CLI	21
1. Configure Kubectl	21
2. Deploy an application	23
3. Expose the application	25
Lab 3. Storage	
Create Persistent Volume (PV) and Persistent Volume Claim (PVC)	
Use the volume in an application	34
Lab 4. Image repository	39
1. Create a docker image	
Prepare the environment	39
Create the image	40
2. Publish the image to IBM Cloud Private	41
3. Create an application	43
Lab 5. Helm Catalog	46
Deploy MQ in IBM Cloud Private	46
Adding HELM Repositories	50

Custom Chart 53
Lab 6. Configure Kubernetes to manage the application
Automatic restarts
Query application health
AutoScale
Rollouts. Update without down time65
Lab 7. Logging
Command line73
ICP Console
ELK74
Lab 8. Monitoring
Lab 9. Alerts
APPENDIX
Lab3 App.js
Lab 3. package.json

Lab 1: Deploy and expose an Application using IBM Cloud Private console.

IBM Cloud Private host: 192.168.142.100.

Master, Worker and Proxy are in the same computer.

1. Login to the console

Using your browser access to https://192.168.142.100:8443 and login with admin / admin

	Platform - IBM Cloud Priv ×								_		X
ſ	← → C ▲ Not secure https://	/ 192.168.142.100 :8443/oidc/login.jsp	Q	☆	2	Θ	ABP	∾^+	÷	Ţ	:
		IBM Cloud Private									
		Open, Enterprise-grade,									
		Log in to your account.									
		Usemame									
		Password									
			Log i	in							

IBM Cloud Private ×		_	
← → C ▲ Not secure https://192.3	168.142.100:8443/console/ G	2 ★ 🗊 😝 🐠	0 🗠 🐥 💻 :
∎ ^{Bees} Ξ IBM Cloud Private			admin 🔘 Create resource Support
Dashboard			
System Overview			
Nodes 1	Shared Storage 42 GB 0.5 Available	Deployments 20	
100% 1 Active Active 0 Inactive	0 % 2 Q B Used Available 0 S Released 0 S Failed	100% 20 Healthy 0	Healthy Unhealthy
Resource Overview			
CPU 4	Memory 7.79 GiB	GPU 0	
Utilization 0.84 CPU 21% Allocation 0 CPU 0%	Utilization 6.29 GiB 81% Allocation 0 B 0%	Utilization 0 GPU 0%	Allocation 0 GPU 0%
100%	100%	100%	
75%	75%	75%	
25%	25%	25%	
9:57 AM 9:57 AM	9:57 AM 9:57 AM	9:57 AM	9:57 AM

2. Deploy a new application (K8s Deployment)

https://kubernetes.io/docs/concepts/workloads/controllers/deployment/

We are going to deploy an NGinx official image.

Click on "Deployments"



Select namespace default

Docs							admi	• ©
IBM Cloud Private							Create resource	Suppo
Deployments							All namespaces	•
							All namespaces	
O. Search items							default	
20 - itame par parta 1-20 of 20 its					1.	of 1 marine	kube-public	
NAME *	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATI	kube-system	
calico-policy-controller	kube- system	1	1	1	1	Feb 5th AM	services	

And click "Create deployment"

-
No Deployments found.
default namespace does not contain any Deployments.
Create Deployment

Enter the following information

General: Name: nginx (lower letters) Replicas: 1 Container Settings: Name: nginx Image: nginx:1.13.1

deployment Create Deploy	nent	×
	Name	
General	nginx	
Labels		
Container settings	Replicas	
Volumes	1	
Environment variables		
Node selector		
Node selector		

deployment Create Deploy	vment	×
General Labels	Name nginx	
Container settings	Image nginx:1.13.1	
volumes Environment variables	Image pull policy	
Node selector	If not present	▼
	Cancel Creat	e

Click "Create".

The first time the image nginx must be pulled from docker hub and it takes a bit of time to deploy the Application (Deployment)

20 🔻 items	per page 1-1 of 1 items						1 of 1 pages	<	1	>
NAME -	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME		ACT	ION	
nginx	default	1	1	-	-	Feb 5th 2018 at 10:20 AM		1		

Deploymer NginX Overview	nts / nginx / nginx-7d -7d9f6dc5d5-d Containers Ever	9f6dc5d5- pft∨ nts	dpftv / Logs									
O, Sear 20 ✔	rch items items per page 1-3 of 3 items								1 of 1 pages	<	1	>
ТҮРЕ	SOURCE	COUNT	REASON	MESSAGE				FIRST SEEN *	LAST S	EEN		
Normal	default-scheduler	1	Scheduled	Successfully	assigned nginx-7d9	916dc5d5-dpttv to 192.168	3.142.100	Feb 5th 2018 at 10:20 AM	1 Feb 5th	2018 8	at 10:20	D AM
Normal	kubelet 192.168.142.100	1	SuccessfulMountVolun	ne MountVolume	e.SetUp succeeded	for volume "default-token	-qmg7t"	Feb 5th 2018 at 10:20 AM	1 Feb 5th	2018 a	at 10:20	D AM
Normal	kubelet 192.168.142.100	1	Pulling	pulling image	"nginx:1.13.1"			Feb 5th 2018 at 10:20 AM	1 Feb 5th	2018 a	at 10:20	D AM
20 🔻	items per page 1-1 of 1 items								1 of 1 pages	<	1	>
NAME +	NAMESPACE		DESIRED C	CURRENT	READY	AVAILABLE	CREATION TI	МЕ		ACT	TION	
nginx	default		1 1	1	1	1	Feb 5th 2018	at 10:20 AM				

Clicking in the application name nginx the console will show you information about the application, replica set and pods

Deployment details				ReplicaSets					
Туре	Detail			Туре		desired		Current	
Name	nginx			nginx-7d9f6dc	5d5	1		1	
Namespace	default			Feb Sth 2018 at	10:20 AM				
Creation time	Feb 5th 2018 at 10:2	0 AM							
Labels	app=nginx								
Selector	app=nginx								
Replicas	1 desired 1 total 1	updated 1 a	available						
RollingUpdateStrategy	25% max unavailable	e, 25% max s	urge						
MinReadySeconds	0								
Pods									
Q Search items									
20 🔻 items per page	1-1 of 1 items					1 of 1 pages	<	1 >	
NAME -	NAMESPACE	STATUS	HOST IP	POD IP	READY	START TIME		ACTION	
nginx-7d9f6dc5d5-dpftv	default	Running	192.168.142.100	10.1.90.220	1/1	Feb 5th 2018 at 10:20 A	М	:	

We will use latter the label "app=nginx", also take note of the pod IP list at the bottom, in this case 10.1.90.220

3. Expose the application (K8s Service)

3.1 ClusterIP

https://kubernetes.io/docs/concepts/services-networking/service/

This application is not exposed yet, now we are going to expose it. Go to Services:

×	IBM Cloud Private	
I	Dashboard	
•	Catalog	
•	Workloads	
~	Network Access	
	Services	1
•	Configuration	0
→ 1	Platform	
•	Manage	
	Command Line Tools	

First we are going to expose the application using a kubernetes Service of type "ClusterIP".

Click on "Create Service".

Services default Services Ingress Create Service 📀 Q Search items 20 👻 items per page | 1-1 of 1 items < 1 > 1 of 1 pages NAME 🔺 CREATION TIME ACTION NAMESPACE : default Feb 5th 2018 at 9:06 AM kubernetes

Enter the following information

General:

Name: nginxsrv Type: ClusterIP

Ports:

Name: http Port: 80 TargetPort: 80 Selectors:

> Selector: app Value: nginx

service Create Service						×
General	•	Name				
Ports		Туре				
Selectors		ClusterIP				•
		Cluster IP				
		Session affinity				•
	-					
					Cancel Crea	ite
service Create Service						×
	*	Protocol	Name	Port	TargetPort	

Conoral	Protocol	Name	Port	TargetPort
General	тср 🗸	http	80	80
Ports				
Selectors	Add port 🔂			
SERVICE				×
Create Service				
Ganaral	Selector		Value	
General	арр		nginx	•
Ports				
Selectors	Add selector	0		

Services

Services	Ingress					
O Search	items			Cr	eate Serv	vice 🛨
20 - it	tems per page 1-2 of 2 items		1 of 1 pages	<	1	>
NAME 🔶	NAMESPACE	CREATION TIME		A	CTION	
kubernetes	default	Feb 5th 2018 at 9:06 AM			:	
nginxsrv	default	Feb 5th 2018 at 11:45 AM			:	

default

If we click on the service "nginxsrv" we get the details:

Service details	
Туре	Detail
Name	nginxsrv
Namespace	default
Creation time	Feb 5th 2018 at 11:45 AM
Туре	ClusterIP
Labels	app=nginxsrv
Selector	app=nginx
IP	10.0.80
Port	http 80/TCP
Node port	None
Session affinity	None

We will work with the command line latter. If we get the details using the command line we get more information as for example the IPs exposed by the service.

kubectl describe service nginxsrv

Name:	nginxsrv
Selector:	app=nginx
Type:	ClusterIP
IP:	10.0.0.80
Port:	http 80/TCP

 TargetPort:
 80/TCP

 Endpoints:
 10.1.90.220:80

 ...
 ...

As you can see the service exposes by a unique IP 10.0.0.80 or hostname "nginxsrv" the application/deployment "app=nginx" in the IP 10.1.90.220, in case we scale or restart the application the service ip/hostame will be the same and will automatically update the Endpoint IPs.

The type of service "ClusterIP" is used for communication between applications inside the cluster so our application is not public yet. We have two options, create a service of type NodePort instead of type ClusterIP or create an Ingress kubernetes object (https://kubernetes.io/docs/concepts/services-networking/ingress/)

3.2 NodePort

Go to Services:



Now we are going to expose the application/deployment as a service of type "NodePort".

Click on "Create Service".

Services default Services Ingress O Search items Create Service 📀 1 > 20 👻 items per page | 1-2 of 2 items 1 of 1 pages < NAME 🔺 NAMESPACE CREATION TIME ACTION • kubernetes default Feb 5th 2018 at 9:06 AM default ÷ Feb 5th 2018 at 11:45 AM <u>nginxsrv</u>

Enter the following information

General:

Name: nginxsrv2 Type: NodePort Ports: Name: http Port: 80 TargetPort: 80 Selectors:

> Selector: app Value: nginx

service Create Service		×
General	Name	
Ports	15moltz	
Selectors	Type NodePort	-

service Create Service					×
General Ports	•	Protocol TCP -	Name http	Port 80	TargetPort 80
Selectors		Add port 9			
service Create Service					×
General	•	Selector app		Value nginxį	•
Selectors		Add selector 🚱			
Services Services Ingress					
Q Search items					
20 vitems per page 1-3 of 3 items		NAMESPAC	E		CREATION TIME
kubernetes		default			Feb 5th 2018 at 9:06 AM
nginxsrv		default			Feb 5th 2018 at 11:45 AM

If we click on the service "nginxsrv2" we get the details:

default

nginxsrv2

Feb 5th 2018 at 2:12 PM

Service details	
Туре	Detail
Name	nginxsrv2
Namespace	default
Creation time	Feb 5th 2018 at 2:12 PM
Туре	NodePort
Labels	app=nginxsrv2
Selector	app=nginx
IP	10.0.0.227
Port	http 80/TCP
Node port	http 31687/TCP
Session affinity	None

A service of type NodePort is a ClusterIP service plus a NodePort. Now the property "Node port" is configured. With NodePort kubernetes enables on the worker and proxy nodes the port "NodePort" to access the Application/Deployment from outside the cluster.

Now you can open a browser and set the URL http://<worker/proxy node>:<node port> and kubernetes will balance the traffic to the internal application/deployment ip:port.





With NodePort we get TCP balancing but no additional features.

In the next section we will use the kubernetes object "Ingress" that enables a kubernetes internal nginx HTTP server to balance the traffic and provides additional features like SSL management

3.3 Ingress

Once we have a service we can create an Ingress object. As explained an Ingress object is use to expose an application to consumers outside the kubernetes cluster. It uses a service to know the IPs of the application/deployment.

The Ingress Controller used by default in ICP is "NGINX Ingress Controller" (<u>https://github.com/kubernetes/ingress-nginx</u>) but there are more ingress controllers available, as for example F5 Ingress Controller

Go to "Services".

×	IBM Cloud Private	
I	Dashboard	
» (Catalog	
÷ ۱	Workloads	
+ I	Network Access	
	Services	1
÷ (Configuration	0
) I	Platform	
→ 1	Manage	
→ 0	Command Line Tools	

And select Ingress Tab.



Click on "Create Ingress"

No ingress found. default namespace does not contain any ingress.
Create Ingress 🗲
General: Name: nginxingress

Rules:

Hostname: nginxhost Service name: nginxsrv Service port: 80

INGRESS Create Ingress		×
General Annotations	Name nginxingress Service name	
Rules	Service port	

ingress Create	Ingress				×
General		Hostname nginxhost	Service nan	ne Service port 80	Path
Annotations Rules		Add rules 🕒			
20 - item	ns per page 1-1 of 1	items		1 of 1 pages	1 >
NAME 🗢	NAMESPACE	HOSTNAMES	ADDRESS	CREATION TIME	ACTION
nginxingress	default	nginxhost	-	Feb 6th 2018 at 9:59 AN	1

With this Ingress object kubernetes has configured the ingress controller (an internal nginx) to route traffic from proxy nodes to the application/deployment in the cluster.

The "rule" is that all the requests made for the hostname "nginxost" must be routed to the service "nginxsrv". There are other options to configure the rules, not only by hostname, it is also possible to use the path.

It is also possible to use annotations to configure how the ingress controller manages the requests.

https://github.com/kubernetes/ingress-nginx/blob/master/docs/user-guide/annotations.md

Now if we add the hostname "nginxhost" to the hosts file of the operating system, pointing to the proxy node ip "192.168.142.100", and access to the url <u>http://nginxhost</u> we will access to our application/deployment.



You can also invoke the application using curl:

```
osboxes@osboxes:~$ curl -H "Host:nginxhost" 192.168.142.100
<!DOCTYPE html>
<html>
....
<body>
<h1>Welcome to nginx!</h1>
If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.
```

```
For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.
```

```
<em>Thank you for using nginx.</em></body></html>
```

Lab 2: Deploy and expose an Application using K8s CLI

1. Configure Kubectl

Access to IBM Cloud Private console and login with admin / admin

https://192.168.142.100:8443

Platform - IBM Cloud Priv ×	
$\leftarrow \rightarrow \mathbf{C}$ A Not secure https://	/192.168.142.100:8443/oidc/login.jsp 🔍 🛠 🔝 🚱 🐵 🞑 🔤 🐺 📜
	IBM Cloud Private Fast. Flexible. Intelligent. Open. Enterprise-grade. Log in to your account.
	Password Log in

Show the client configuration

BM Clou	ud Private	×	-		Ŀ			x
← → C	A Not secu	re https ://19	ର୍ ★	٣	Θ	ABP		:
Docs						admin	0	Â
🔳 ІВМ С	loud Private		Config	gure clie	ent			
Dashboard	d		About					ПΙ
			Log ou	ut				
System Ove	erview							
Nodes :: https://192.168.1	1 .42.100:8443/cor	nsole/#			_			-

Copy the configuration:

		×
	Configure Kubectl	
	Before you run commands in the kubectl command line interface for this cluster, you must configure the client.	
	Prerequisites: Install the kubectl CLI: kubectl	
	To configure the CLI, paste the displayed configuration commands into your terminal window and run them.:	
	kubectl config set-cluster mycluster.icpserver=https://192.168.142.100:8001insecure-skip-tls kubectl config set-context mycluster.icp-contextcluster=mycluster.icp kubectl config set-credentials admintoken=eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhZG1pbiIsImF0> kubectl config set-context mycluster.icp-contextuser=adminnamespace=default kubectl config use-context mycluster.icp-context	
	4 F	
1		

This configuration is used by the k8s CLI (kubectl) to interact with correct kubernetes provider. In this case the configuration points to the IBM Cloud Private installation.

Paste it in the command line terminal. (open a putty for 192.168.142.100 with user osboxes / osboxes.org



It is also possible to get this configuration through command line in case of system integrations.

2. Deploy an application

Now we can execute kubectl commands in name of user "admin".

List the application deployed in Lab1:

osboxes@osboxes:~\$ kubectl get deployments NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE nginx 1 1 1 1 1h

List the service deployed in Lab1:

osboxes@osboxes:/\$ kubectl get service NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 1d nginxsrv ClusterIP 10.0.0.80 <none> 80/TCP 22h nginxsrv2 NodePort 10.0.0.227 <none> 80:31687/TCP 20h

osboxes@osboxes:/\$ kubectl get ingress NAME HOSTS ADDRESS PORTS AGE nginxingress nginxhost 192.168.142.100 80 20m

Although you can create an application directly from command line we are going to create a descriptor file. Create a file "nginx-deployment.yml" with the content:

osboxes@osboxes:~\$ vi nginx-deployment.yml

osboxes@osboxes:~\$ cat nginx-deployment.yml

apiVersion: extensions/v1beta1 kind: Deployment metadata: name: nginx-deployment labels: app: nginx spec: replicas: 1 template: metadata: labels: app: nginx-deployment spec: containers: - name: nginx image: nginx:1.13.1 resources: requests: cpu: 100m memory: 100Mi ports: - containerPort: 80

Create the deployment using k8s CLI:

osboxes@osboxes:~\$ kubectl create -f nginx-deployment.yml deployment "nginx-deployment" created

osboxes@osboxes:~\$ kubectl get deployments NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE nginx 1 1 1 1 1h nginx-deployment 1 1 1 0 9s

osboxes@osboxes:~\$ kubectl get pods NAME READY STATUS RESTARTS AGE nginx-1769497579-80krq 1/1 Running 0 1h nginx-deployment-1290044638-30937 1/1 Running 0 23m

If we go the IBM Cloud Private console we will see the new nginx-deployment.

IBM Cloud Pri	vate					Cr	eate resource	Supp
Deploymer	nts					d	efault	•
O Search items						Crea	ate Deploymen	t 🕀
20 🔻 items per	page 1-2 of 2 iten	ns				1 of 1 pages	< 1	>
NAME -	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME	ACTI	ION
nginx	default	1	1	1	1	Feb 5th 2018 at 10:20	D AM	
nginx-deployment	default	1	1	1	1	Feb 6th 2018 at 10:22	2 AM	

3. Expose the application

Create a file "nginx-deployment.yml" with the content:

osboxes@osboxes:~\$ vi nginx-service.yml

osboxes@osboxes:~\$ cat nginx-service.yml

```
apiVersion: v1
kind: Service
metadata:
name: nginx-service
labels:
app: nginx-service
spec:
type: NodePort
ports:
- port: 80
targetPort: 80
# nodePort: 31320
selector:
app: nginx-deployment
Comments about this file:
```

We are going to expose the service as NodePort with the property "type: NodePort". Although it is possible to specify a nodePort it is recommended to let K8s to assign it dynamically. With the selector property we are specifying that the deployment that we are going to expose with this service is the one we created before, it has the property "selector: app: nginxdeployment"

Create the service using K8s CLI:

osboxes@osboxes:~\$ kubectl create -f nginx-service.yml service "nginx-service" created

osboxes@osboxes:~\$ kubectl get services CLUSTER-IP EXTERNAL-IP PORT(S) NAME TYPE AGE ClusterIP 10.0.0.1 <none> kubernetes 443/TCP 1d nginx-service NodePort 10.0.0.68 <none> 80:30062/TCP 16s ClusterIP 10.0.0.80 <none> nginxsrv 80/TCP 22h nginxsrv2 NodePort 10.0.0.227 <none> 80:31687/TCP 20h

Take note of the port "30062", it is the port assigned by k8s to our service in the worker nodes.

In the IBM Cloud private console:

Services				default		•
Services Ingress						
O Search items				Create	e Servic	e 🔂
20 - items per page 1-4	of 4 items		1 of 1 pages	<	1	>
NAME -	NAMESPACE	CREATION TIME		ACTIO	N	
<u>kubernetes</u>	default	Feb 5th 2018 at 9:06 AM		:		
nginx-service	default	Feb 6th 2018 at 10:34 AM		:		
nginxsrv	default	Feb 5th 2018 at 11:45 AM		:		
nginxsrv2	default	Feb 5th 2018 at 2:12 PM		:		

Now we can invoke the application (Deployment), this time we have exposed it as NodePort so to invoke it we need to point to <K8s worker IP>:<NodePort>, in our case 192.168.142.100:30062.

IBM Cloud Private also expose that port through the proxy component balancing the load between the worker nodes, in our installation the worker node and the proxy node are in the same computer so the url will be the same 192.168.142.100:30062.



Lab 3. Storage

Create Persistent Volume (PV) and Persistent Volume Claim (PVC)

https://kubernetes.io/docs/concepts/storage/volumes/

During this lab we will configure Storage in IBM Cloud private, this storage will allow to our application to access to a persistent storage.

We will use nginx and will load a page we have created.

There are several options for the Storage, like for example NFS, but for this lab we will use the local file system.

Connect to the IBM Cloud private host and in the command line (putty terminal):

sudo mkdir -p /aStorage/nginx sudo chmod 777 -R /aStorage/ vi /aStorage/nginx/index.html cat /aStorage/nginx/index.html

<html> <body> Hello IBM Cloud private </body> </html>

Now go to the IBM Cloud private console, login as admin / admin and go to storage.

×	IBM Cloud Private	
1	Dashboard	
. (Catalog	
	Workloade	
,		
•	Network Access	
) (Configuration	
•	Platform	
	Alerting	
	Logging	
	Metering	
	Monitoring	
	Network	%
	Nodes	
	Storage	

Create a new Persistent Volume:

Storage								
PersistentVolume	Persis	tentVolume	Claim					
O Search items						Cre	eate PersistentVo	lume 🕂
20 👻 items per page	e 1-3 o	f 3 items				1 of 1 pag	es 🔇 1	>
NAME -	түре	CAPACITY	ACCESS MODE	RECLAIM POLICY	STATUS	CLAIM	CREATION TIME	ACTION
<u>cloudant-</u> 192.168.142.100		2Gi	RWO	Delete	Bound	kube-system/icp-ds-icp-ds-0	Feb 5th 2018 at 9:08 AM	:
<u>image-manager-</u> 192.168.142.100		20Gi	RWO	Retain	Bound	kube-system/image-manager- image-manager-0	Feb 5th 2018 at 9:12 AM	:
logging-es-pv- 192.168.142.100		20Gi	RWO	Delete	Bound	kube-system/elasticsearch-data- elasticsearch-data-0	Feb 5th 2018 at 9:12 AM	:

PERSISTENTVOLUME

Create PersistentVolume

Labels Parameters Storage class name Capacity Unit 10 Mi Access mode Read write many Reclaim policy Retain Storage type Host path	General	Name	
Parameters Capacity Unit 10 Unit Mi Access mode Read write many Reclaim policy Retain Storage type Host path	Labels	Starade class name	
Capacity Unit 10 Mi • • • • • • • • • • • • • • • • • •	Parameters		
10 Mi Access mode Read write many Reclaim policy Retain Storage type Host path		Capacity Unit	
Access mode Read write many Reclaim policy Retain Storage type Host path		10 Mi	•
Reclaim policy Retain Retain Reclaim policy Retain		Access mode	
Reclaim policy Retain Storage type Host path		Read write many	•
Retain Storage type Host path		Reclaim policy	
Storage type Host path		Retain	•
Host path	•	Storage type	
		Host path	•

PERSISTENTVOLUME

Create PersistentVolume

Conoral	•	Кеу	Value	
Labele		path	/aStorage/nginx	•
Parameters		Add parameter 📀		

JSON file:

```
{
    "kind": "PersistentVolume",
    "apiVersion": "v1",
    "metadata": {
        "name": "nginxhtml",
        "labels": {}
},
```

×

×

```
"spec": {
   "capacity": {
    "storage": "10Mi"
   },
   "accessModes": [
    "ReadOnlyMany"
   ],
   "persistentVolumeReclaimPolicy": "Retain",
   "hostPath": {
     "path": "/aStorage/nginx"
   }
 }
}
  20 - items per page | 1-4 of 4 items
NAME -
                                    түре
                                                CAPACITY
                                                            ACCESS MODE
                                                                            RECLAIM POLICY
                                                                                               STATUS
                                                                                                          CLAIM
cloudant-192.168.142.100
                                                2Gi
                                                            RWO
                                                                            Delete
                                                                                               Bound
                                                                                                          kube-system/icp-ds-icp-ds-0
image-manager-192.168.142.100
                                                20Gi
                                                            RWO
                                                                            Retain
                                                                                               Bound
                                                                                                           kube-system/image-manager-image-manager-0
logging-es-pv-192.168.142.100
                                                            RWO
                                                                            Delete
                                                                                                          kube-system/elasticsearch-data-elasticsearch-data-0
                                                20Gi
                                                                                               Bound
<u>nginxhtml</u>
                                                            RWX
                                                                            Retain
                                                                                               Available
                                    Hostpath
                                                10Mi
```

Now select the tab PersistentVolumeClaim and create a new Persistent Volume Claim



PERSISTENTVOLUMECLAIM Create PersistentVolumeClaim		×
Name nginxvolume		
Storage class name		l
Storage requests Unit 10 Mi	•	1
Access mode Read write many	•	-
	Cancel Create	

JSON File:

```
{
 "kind": "PersistentVolumeClaim",
 "apiVersion": "v1",
 "metadata": {
  "name": "nginxvolume"
 },
 "spec": {
  "resources": {
   "requests": {
   "storage": "10Mi"
   }
  },
  "accessModes": [
   "ReadOnlyMany"
 ]
}
}
```

It is possible to use Labels and Storage Classes to specify which Persistent Volume will be bound by the Persistent Volume Claim.

20 👻 items per	page 1-1 of 1 items									1 of 1 pages	<	1	>
NAME 🗢	NAMESPACE	STATUS	PERSISTENTV	OLUME	REQUEST	rs	ACCESS MODE		CREATION TIME			AC	TION
nginxvolume	default	Pending	-		10Mi		RWX		Feb 6th 2018 at	11:29 AM			:
20 👻 items per	page 1-1 of 1 items									1 of 1 pages	<	1	>
NAME 🔺	NAMESPACE	STATUS	PERSISTENTV	OLUME	REQUEST	ſS	ACCESS MODE		CREATION TIME			AC	TION
nginxvolume	default	Bound	nginxhtml		10Mi		RWX		Feb 6th 2018 at	11:29 AM		*	
Storage													
PersistentVolume	PersistentVol	umeClaim											
20 👻 items per	page 1-4 of 4 items									1 of 1 pages	<	1	>
NAME 🛎	ТҮРЕ	CAPACITY	ACCESS MODE	RECLAIM POLICY	STATUS	CLAIM				CREATION	TIME		ACTION
cloudant-192.168.1	42.100	2Gi	RWO	Delete	Bound	kube-sy	stem/icp-ds-icp-	ds-0		Feb 5th 201 AM	.8 at 9:0	8	:
<u>nginxhtml</u>	Hostpa	ath 10Mi	RWX	Retain	Bound	default/	nginxvolume			Feb 6th 201 AM	.8 at 11:	15	:

Use the volume in an application.

Creating a new deployment in the UI we can specify the volume claim in the "Volumes" section of a new deployment window:

DEPLOYMENT × Create Deployment Name Volume Path General nginxvolume nginxvolume /usr/share/nginx/htm 0 Labels Container settings Add volume 🕀 Environment variables Node selector Cancel Create

YAML file:

kind: Deployment apiVersion: extensions/v1beta1 metadata: name: nginx-volume spec: replicas: 1 template: metadata: labels: app: nginx-volume spec: hostNetwork: false volumes: - persistentVolumeClaim: claimName: nginxvolume name: nginxvolume containers: - name: nginx image: nginx imagePullPolicy: IfNotPresent

ports:
- protocol: TCP
containerPort: 80
resources:
limits:
cpu: 100m
memory: 100Mi
volumeMounts:
- name: nginxvolume
mountPath: "/usr/share/nginx/html"

Using the console with the property "Path" or in the yaml file with the property "mountPath" we specify in which path inside the container we are going to mount our storage. In this case is "/usr/share/nginx/html" that is where nginx loads its default web page.

Doc	C8		admin	0
	IBM Cloud Private		Create resource	
	Dashboard			
	System Overview	RESOURCE X Create resource		
	100% Active	1 1 1 22 0 14 15 16 1 130: nginx-volume 12 12 1 130: nginx-volume 12 1 13: nostHetwork: false 0 14 volume2: 16 15 - persistentvolumeClaim: 16 - claimine: nginxvolume 17 container: 18 - container: 19 - container: 10 - container:	Healthy Unhealthy	
	Resource Overview	21 immgePullPolicy: IfNotPresent 22 ports: 23 - protocol: TCP 24 containerPort: 80 25 - resources: 26 - limits:		
	CPU 4	27 Cpui Jeem 28 menory: 100vii 29 volumekounts: 30 name: nginxvolume 11 CPUI - tation 0 G	PU 0%	
	Allocation	0.1 CPU Cancel Create ation 0 G	PU 0%	
	100%			
	75%	75% 75%		

In the console, in the top menu, select create resource, copy the yaml and click "Create"
Deployments

O, Search items						c	reate Deployment 📀
20 🔻 items per page	1-3 of 3 items					1 of 1 pages	< 1 >
NAME -	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME	ACTION
nginx	default	1	1	1	1	Feb 5th 2018 at 10:20 AM	*
nginx-deployment	default	1	1	1	1	Feb 6th 2018 at 10:40 AM	:
nginx-volume	default	1	1	1	1	Feb 6th 2018 at 11:44 AM	:

Expose the application:

As before click on "Create resource" and create a servive:

Services				defau	lt	•
Services Ingress						
O, Search items				Cre	ate Servi	ice 🛨
20 - items per page 1-5 of 5 items			1 of 1 pages	<	1	>
NAME -	NAMESPACE	CREATION TIME	AC	TION		
kubernetes	default	Feb 5th 2018 at 9:06 AM		* *		
nginx-service	default	Feb 6th 2018 at 10:34 AM		:		
nginx-volume	default	Feb 6th 2018 at 11:51 AM		:		

Check the node port exposed.

default

-

Services / nginx-volume /

nginx-volume

Overview

Service details	
Туре	Detail
Name	nginx-volume
Namespace	default
Creation time	Feb 6th 2018 at 11:51 AM
Туре	NodePort
Labels	app=nginx-volume
Selector	app=nginx-volume
IP	10.0.057
Port	80/TCP
Node port	<u>30741/TCP</u>
Session affinity	None

Invoke the application from the command line:

osboxes@bootnode:~\$ curl 192.168.142.100:30741 <html> <body> Hello IBM Cloud private </body> </html>

Make a change in the file and execute it again, you will see your updated file.

Lab 4. Image repository

During these labs we have been using docker hub as image repository, in this lab we are going to create our own docker image and we are going to push it to IBM Cloud Private image repository, then we will use this image to create an application.

This image is a node.js application that we will use in the next lab to demonstrate the out of the box features of Kubernetes as a "docker container manager".

1. Create a docker image

Prepare the environment

For this lab we provide two files, app.js with the node.js code and package.json to automate the build and execution of the node.js with npm.

In case you don't have app.js and pacakage.json files in your environment they are in the Appendix section of this document.

Copy / create them in a folder with name "node".

osboxes@osboxes:~\$ pwd /home/osboxes

osboxes@osboxes:~\$ mkdir node

osboxes@osboxes:~\$ vi node/app.js osboxes@osboxes:~\$ vi node/package.json

osboxes@osboxes:~\$ ls node app.js package.json

Create the dockerfile to create our image.

osboxes@osboxes:~\$ vi hellonode.dockerfile

osboxes@osboxes:~\$ cat hellonode.dockerfile

FROM debian:8.7

RUN apt-get -y update && apt-get install -y curl

RUN curl -sL https://deb.nodesource.com/setup_6.x | bash -RUN apt-get install -y nodejs

RUN mkdir /app ADD ./node/app.js /app ADD ./node/package.json /app RUN cd /app && npm install

ENV PORT 8080 EXPOSE 8080 WORKDIR "/app" CMD ["npm", "start"]

With this docker file what we do is to create a new docker image from a from a debian docker image in dockerhub. This is the line "FROM debian:8:1".

Then we install node and copy our files inside the image.

Finally we say to the image that it has be started with the command /app/npm start

Create the image

At this moment we have these images in our local repository:

osboxes@osboxes:~\$ sudo docker images grep -v ibmcom								
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE				
nginx	latest	3f8a4339aadd	5 weeks ago	108MB				
hello-world	latest	f2a91732366c	2 months ago	1.85kB				
nginx	1.13.1	c246cd3dd41d	7 months ago	107MB				

Create our hellonode docker image.

osboxes@osboxes:~\$ pwd /home/osboxes osboxes@osboxes:~\$ sudo docker build -t hellonode:1.0 -f hellonode.dockerfile .

•••

Successfully built 84ed3945077c

If we execute again docker images we will see debian:8.7 and hellonode:1.0

osboxes@osboxes:~\$ sudo docker images grep -v ibmcom							
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE			
hellonode	1.0	595463fa00bb	22 seconds ago	227MB			
nginx	latest	3f8a4339aadd	5 weeks ago	108MB			
hello-world	latest	f2a91732366c	2 months ago	1.85kB			
nginx	1.13.1	c246cd3dd41d	7 months ago	107MB			
debian	8.7	054abe38b1e6	9 months ago	123MB			

2. Publish the image to IBM Cloud Private

In the IBM Cloud private list the images.

Login in IBM Cloud private.

https://192.168.142.100:8443

List the images:



As this is a fresh installation the user does not have any image yet, all the images used in previous labs where in dockerhub repository.

Using docker push command we will publish hellonode image to the IBM Cloud Private repository.

Tag the image

First we need to tag the docker image, with the tag name docker will know where to push the image.

osboxes@osboxes:~\$ sudo docker tag hellonode:1.0 mycluster.icp:8500/default/hellonode:1.0

osboxes@osboxes:~\$ sudo docker images grep -v ibmcom							
REPOSITORY	TAG	IMAGE ID	CREA	TED	SIZE		
hellonode	1.0	595463fa00l	ob 4 min	utes ago	227MB		
mycluster.icp:8500/defaul	t/hellonode	1.0	595463fa00)bb 4 r	minutes ago		
227MB							
nginx	latest	3f8a4339aadd	5 week	s ago	108MB		
hello-world	latest	f2a9173236	56c 2 mo	onths ago	1.85kB		
nginx	1.13.1	c246cd3dd41	d 7 mon	ths ago	107MB		
debian	8.7	054abe38b1e	6 9 mont	ths ago	123MB		

mycluster.icp:8500/ is the hostname pointing to IBM Cloud Private image repository. default is the namespace hellonode is the image name 1:0 is the version tag

Login to the repository and push the image

osboxes@osboxes:~\$ sudo docker login mycluster.icp:8500 Username (devuser): admin Password: Login Succeeded

osboxes@osboxes:~\$ sudo docker push mycluster.icp:8500/default/hellonode:1.0

Now in the IBM Cloud Private console you can see the image.

Images

Q Search items						
20 🔹 items per page 1-1 of 1 items			1 of 1 pages	<	1	>
NAME -	OWNER	SCOPE				
default/hellonode	default	namespace				

The Scope determines if this image can be seen in any other namespace or just in the current one. By default it is configured with scope "namespace" what means that it can only be used from the current namespace.

3. Create an application

As we did in the previous create a new deployment and service using the "Create Resource" option.

```
kind: Deployment

apiVersion: extensions/v1beta1

metadata:

name: hellonode

spec:

replicas: 1

template:

metadata:

labels:

app: hellonode

spec:

containers:

- name: hellonode

image: mycluster.icp:8500/default/hellonode:1.0

imagePullPolicy: IfNotPresent
```

Pay attention to the image property "image: mycluster.icp:8500/default/hellonode:1.0", now we are using the internal docker image repository. Also the property "replicas: 1", we will use it latter.

Deployments

default 👻

O _s Search items							Create I	Deployme	nt 🔂
20 - items per page 1-4	of 4 items					1 of 1 pages	<	1	>
NAME 🛎	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME		ACTION	
hellonode	default	1	1	1	1	Feb 6th 2018 at 12:27 PM		:	
nginx	default	1	1	1	1	Feb 5th 2018 at 10:20 AM		:	
nginx-deployment	default	1	1	1	1	Feb 6th 2018 at 10:40 AM		:	
nginx-volume	default	1	1	1	1	Feb 6th 2018 at 11:44 AM		:	

kind: Service apiVersion: v1 metadata: name: hellonode labels: app: hellonode spec: type: NodePort ports: - protocol: TCP name: node port: 8080 targetPort: 8080 selector: app: hellonode clusterIP: " sessionAffinity: None

Services

Services	Ingress		
O Search it	ems		
20 🕶 ite	ms per page 1-6 of 6 items		
NAME 🛎		NAMESPACE	CREATION TIME
hellonode		default	Feb 6th 2018 at 12:29 PM

Services / hellonode /

hellonode

Overview

Service details	
Туре	Detail
Name	hellonode
Namespace	default
Creation time	Feb 6th 2018 at 12:38 PM
Туре	NodePort
Labels	app=hellonode
Selector	app=hellonode
IP	10.0.0.128
Port	node 8080/TCP
Node port	node 31491/TCP
Session affinity	None

In this script the port is 31491 but it will be different in each case. It is recommended you take note of your port to use it in the rest of the labs.

Í	NAME AND ADDRESS OF TAXABLE PARTY.			
	□ IBM Cloud Private × □ 192.168.142.100:31491	×		
	$\leftarrow \rightarrow \mathbf{C}$ (1) 192.168.142.100:31491	. ☆	<u>اگ</u>	Θ
	I'm alive!!!			

Lab 5. Helm Catalog

Helm is a tool that streamlines installing and managing Kubernetes applications. Think of it like apt/yum/homebrew for Kubernetes.

Use Helm to:

Find and use popular software packaged as Kubernetes charts Share your own applications as Kubernetes charts Create reproducible builds of your Kubernetes applications Intelligently manage your Kubernetes manifest files Manage releases of Helm packages

Helm uses a packaging format called charts. A chart is a collection of files that describe a related set of Kubernetes resources. A single chart might be used to deploy something simple, like a memcached pod, or something complex, like a full web app stack with HTTP servers, databases, caches, and so on.

Charts are created as files laid out in a particular directory tree, then they can be packaged into versioned archives to be deployed.

https://github.com/kubernetes/helm/blob/master/README.md https://github.com/kubernetes/helm/blob/master/docs/charts.md

By default ICP comes with a HELM repository to access and Install IBM software, but it is possible to add external repositories with helm charts for third party software.

Deploy MQ in IBM Cloud Private

Access to ICP HELM Catalog



Helm ch	larts				
Deploy yo	ur applications and install software packages				
	ibm-cloudant-dev Cloudant for Linux. ibm-charts		ibm-datapower-dev IBM DataPower Gateway ibm-charts		ibm-db2oltp-dev IBM Db2 Developer-C Edition 11.1.2.2 iFix002 ibm-charts
	ibm-db2warehouse-dev Db2 Warehouse Developer-C for Non- Production v2.1.0 ibm-oharte		ibm-dsm-dev IBM Data Server Manager Developer-C Edition. Note that there can only be one ibm-charts	\bigotimes	ibm-dsx-dev IBM Data Science Experience (DSX) Developer Edition brings together best of ibm-charts
٢	ibm-icplogging Log storage and search management solution ibm-charts		ibm-icplogging-kibana Installs Kibana, a web UI to query and visualize data in existing Elasticsearch ibm-charte		ibm-icpmonitoring IBM monitoring service in private cloud ibm-charts
٢	ibm-iisee-eval IBM InfoSphere Information Server for Evaluation v11.7 (Evaluation) ibm-charts	۲	ibm-integration-bus-dev IBM Integration Bus for application integration, routing and transformation ibm-charts	*	ibm-lsfce-dev IBM LSF Community Edition cluster with Application Center GUI ibm-charts

Look for mq and click on the chart

Catalog	
O _o mq	
Helm charts Deploy your applications and install software packages	
ibm-mqadvanced-server-dev IBM MQ queue manager	rabbitmq Open source message broker software that implements the Advanced Message
ibm-charts	ibm-charts

Take a look at the documentation and click on configure

ibm-mqadvanced-server-dev V 1.1.0

IBM MQ queue manager	IBM MQ						
ibm-charts View Licenses	IBM® MQ is messaging middleware that simplifies and accelerates the integration of diverse applications and business data across multiple platforms. It uses message queues to facilitate the exchanges of information and offers a single messaging solution for cloud, mobile, Internet of Things (IoT) and on-premises environments.						
VERSION 1.1.0 -	Introduction						
PUBLISHED Jan 2nd 2018 TYPE Helm Chart	This chart deploys a single IBM MQ Advanced for Developers server (queue manager) into an IBM Cloud private or other Kubernetes environment.						
	Prerequisites						
	Kubernetes 1.6 or greater, with beta APIs enabled						
	If persistence is enabled (see <u>configuration</u>), then you either need to create a PersistentVolume, or specify a Storage Class i classes are defined in your cluster.						
	Configure						

Set this configuration:

Release name: mymq

Target namespace: default

Check the box "I have read and agreed to the license agreements"

Uncheck the box "Enable persistence"

Service type: NodePort

Queue manager name: qmgr

Admin password: admin

Release name 🕕	Target namespace 🕕
mymq	default
✓ I have read and agreed to the <u>license agreements</u>	
Persistence	
Enable persistence 1	
Service	
Service name 🕕	Service type 🚯
qmgr	NodePort

Queue manager	
Queue manager name 🚯	Admin password 🕕
qmgr	

Click Install and when finished click on "View Hel Release"



Click on "mymq"

Helm releases

O, Search	items					
20 🔻	items per page 1-1 of 1 it	ema			1 of 1 pages 🔍 🗸	1 >
NAME*	NAMESPACE	STATUSA	UPDATEDA	CHART NAME*	CHART VERSION*	ACTION
mymq	default	DEPLOYED	Feb 8th 2018	ibm-mqadvanced-server-dev	1.1.0	:

In the installation sever kubernetes objects where created. Take note of the port assigned to 9443, in this case 32178

Secret				
NAME		TYPE	DATA	AGE
mymq-ibm-mq		Opaque	1	3m
Service				
NAME	CLUSTER IP	EXTERNAL IP	PORT(S)	AGE
mymq-ibm-mq	10.0.0.13	<nodes></nodes>	1414:32134/TCP,9443 32178/ ¹ CP	3m
StatefulSet				
NAME		KIND		
mymq-ibm-mq		StatefulSet.v1	beta2.apps	

Access to the admin console and login with admin / admin

<u> </u>			A No	+		h#		1016	01/17	100.22	170 <i>1</i> :L		oncolo	,								- :
				t see	Lure	mu	ps .//1:	92.10	0.142	100.52	1/0/10	ninq/c	onsole							~ r	()	:
IBM MO	סC ב	nsole																				
IBM	MQ	Cont	ainer	•	+																	
																				0	Add v	vidget
Que	ue Ma	anage	r							S	¢ ×	Char	nels on o	qmgr							C 🗘	×
+	Ê		•	I N	lore	Sea	rch					+	Î II	•		More	Sea	rch				
			▲Name						Status				▲Nan	пе			Туре	Э	Ove	rall cha	nnel sta	atus
qmgr						1	Running	1				DEV.AI	MIN.SVR	CONN		Server-co	onnectio	n	🕂 Inacti	ve		
												DEV.AF	P.SVRCC	NN		Server-co	onnectio	n	🕂 Inacti	ve		
Total:	1 Se	lected:	0						Up	dated: 2:5	58:49 PM	Total:	2 Selecte	d: 0						Update	d: 2:58:	48 PM
Que	ues o	n qmg	r							្ទ	¢ ×	Торіс	s on qm	gr							្រ 🗘	×
+	Ê			N	lore	Sea	rch					+	İ	\sim		More	Sea	rch				
		Name			(Queue	type		QL	leue depti	h			▲Na	me				Topic S	String		
DEV.D	EAD.L	ETTER	R.QUEUE	E Lo	cal			0				DEV.B/	ASE.TOPIC	0			d	ev/				
DEV.Q	UEUE	.1		Lo	cal			0														
DEV.Q	UEUE	.2		Lo	cal			0														
DEV.Q	UEUE	.3		Lo	cal			0														
Total:	4 Se	lected:	0						Up	dated: 2:5	58:49 PM	Total:	1 Selecte	d: 0						Update	d: 2:58:	47 PM

https://192.168.142.100:32178/ibmmq/console

Adding HELM Repositories

Manage HELM Repositories:



By default there are two repositories.

Ibm-charts is the repository we have just used where are all the carts to install IBM software

Local-charts is an internal repository where ICP users can register their own charts.

Click on "Add repository" to add an extern repository

Repositories

O ₆ Search items		Sync repositories 🔿	Add reposite	ory 🕂
20 🔻 items per page	1-2 of 2 items	1 of 1 pages	< 1	>
NAME	URLA		ACTION	
ibm-charts	https://raw.githubusercontent.com/IBM/charts/master/repo/stable		:	
local-charts	https://192.168.142.100:8443/helm-repo/charts		:	

Add the repository https://kubernetes-charts.storage.googleapis.com/

× Add repository Name public-charts URL https://kubernetes-charts.storage.googleapis.com/



Repositories

O ₄ Search items	
20 🔻 items per page 3	L-3 of 3 items
NAME	URL-
ibm-charts	https://raw.githubusercontent.com/IBM/charts/master/repo/stable
local-charts	https://192.168.142.100:8443/helm-repo/charts
public-charts	https://kubernetes-charts.storage.googleapis.com

Go to the HELM Catalog and check the new charts.



Catalog



Custom Chart

https://www.ibm.com/support/knowledgecenter/en/SSBS6K_2.1.0/app_center/add_package.htm

A chart is a zip file of kubernetes objects as the files we used in previous labs to create deployment and service objects.

Access to the url https://github.com/jxadro/ICP_PoT/tree/master/custom-chart-app

Branch: master 🔻	ICP_PoT / custom-chart-app /
jxadro chart	
templates	
.helmignore	
Chart.yaml	
README.md	
🖹 values.yaml	

Branch: master ICP_PoT / custom-chart-app / templates /
jxadro chart
■_helpers.tpl
■ deployment.yaml
🖹 hpa.yaml
ingress.yaml
■ service.yaml

This is the structure of a chart.

Inside the folder "templates" resides all the objects that are going to be created during the char installation. It can be as many objects as needed. Although in this case there is only one deployment if we would install a microservice application there would be several deployments.

These files use external parameters so the user can set the desired values during the chart installation:

```
Branch: master •
ICP_PoT / custom-chart-app / templates / service.yaml

jxadro chart
1 contributor
```

```
19 lines (18 sloc) 494 Bytes
      apiVersion: v1
   1
   2
       kind: Service
       metadata:
   4
       name: {{ .Release.Name }}
        labels:
   6
           app: {{ .Release.Name }}
          chart: {{ .Chart.Name }}-{{ .Chart.Version | replace "+" "_" }}
   8
          release: {{ .Release.Name }}
           heritage: {{ .Release.Service }}
   9
       spec:
        type: {{ .Values.service.type }}
         ports:
           - port: {{ .Values.service.externalPort }}
             targetPort: {{ .Values.service.internalPort }}
  15
             protocol: TCP
             name: {{ .Values.service.name }}
  17
        selector:
           app: {{ .Release.Name }}
```

The default values are taken from the file values.yaml

Branch: master ▼ ICP_PoT / custom-chart-app / values.yaml

ightarrow jxadro chart

contributor

30 lines (29 sloc) 879 Bytes

1 # Default values for chartapp.

2 # This is a YAML-formatted file.

3 # Declare variables to be passed into your templates.

- 4 replicaCount: 1
 5 image:
 6 repository: mycluster.icp:8500/default/demonodeapp
- 7 tag: latest
- 8 pullPolicy: IfNotPresent
- 9 service: 10 type: NodePort
- 11 externalPort: 8080
- 12 internalPort: 8080
- 13 replicaCount: 1
- 14 autoscaling:
- 15 enabled: true
- 16 minReplicas: 1
- 17 maxReplicas: 10
- 18 targetCPUUtilizationPercentage: 50

Lab 6. Configure Kubernetes to manage the application

List the pods.

kubectl get pods NAME READY STATUS RESTARTS AGE hellonode-546947b56f-hvq77 1/1 Running 0 18m

See that the hellonode pod has not been restarted anytime yet. RESTARTS = 0

osboxes@osboxes:~\$ kubectl get pods				
NAME	READY	STATUS	RESTARTS	AGE
hellonode-546947b56f-hvq77	1/1	Running	0	18m

Automatic restarts

Invoke /kill , The /kill method executes the sentence "process.exit();" what makes the node.js process quit and finish the container.



After invoking /kill, if you execute in the command line kubectl get pods, you will see that the hellonode pod has status completed and 0/1 Ready. But as when we deployed the Application we set that we wanted 1 replica active, when kubernetes detects that there are less replicas active than the number configured it automatically restarts the pods.

So if you execute again get pods you will see that the hellonode pod has ready 1/1, status running and 1 restart.

osboxes@osboxes:~\$ kubectl get pods NAME hellonode-546947b56f-hvq77	READY 0/1	STATUS Completed	RESTARTS 0	AGE 22m
osboxes@osboxes:~\$ kubectl get pods				
NAME	READY	STATUS	RESTARTS	AGE
hellonode-546947b56f-hvq77	1/1	Running	1	22m

In case of deleting the pod, kubernetes will create a new one:

osboxes@osboxes:~\$ kubectl	get pods									
NAME	READY	STATUS	RESTARTS	AGE						
hellonode-546947b56f-hvq77	1/1	Running	1	2 6m						
osboxes@osboxes:~\$ kubectl	delete pod 1	hellonode-S	546947b56f-	-hvq77						
pod "hellonode-546947b56f-hvq77" deleted										
osboxes@osboxes:~\$ kubectl	get pods									
NAME	READY	STATUS		RESTARTS	AGE					
hellonode-546947b56f-tmsc7	0/1	Container(Creating	0	5s					
osboxes@osboxes:~\$ kubectl	get pods									
NAME	READY	STATUS	RESTARTS	AGE						
hellonode-546947b56f-tmsc7	1/1	Running	0	12s						

Query application health

We want kubernetes to ask the application its status, if it does not return a 200 OK, Kubernetes will restart the Application.

In the node.js application we used to create our hellonode docker images we have two methods:

```
app.get("/infect", function(req, res, next){
    isHealthy = false;
app.get("/health", function(req, res, next){
    if(isHealthy)
        res.send("GREEN");
    else
        res.status(500).send("RED");
```

So after invoking the infect method, the app will start returning HTTP 500 code.

First we need to configure our Application to poll the application status.

Edit the hellonode application in the IBM Cloud Private console:

Deploy	ments					defau	llt 👻
O, Search	h items					Create [Deployment 🕂
20 🔻	items per page 1	1-1 of 1 items				1 of 1 pages 💙	1 >
NAME 🔺	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME	ACTION
<u>hellonode</u>	default	1	1	1	1	Feb 6th 2018 at 12:27 PM	
						Edit	
						Scale	
						Remove	

Modify the deployment descriptor to add the element livenessProbe, this element is used by kubernetes to poll the application asking for the status, in case it does not return 200 OK the Application will be restarted.



```
"livenessProbe": {
    "httpGet": {
        "path": "/health",
        "port": 8080
    },
    "initialDelaySeconds": 5,
    "periodSeconds": 10
},
```

Wit this configuration kubernetes will invoke the operation /heath in periods of 10 seconds, if three consecutives times the application does not returns 200 OK, kubernetes will restart the Application.

In the command line execute kubectl get pods and see the logs

osboxes@osboxes:~\$ kubectl	. get pods				
NAME	READY	STATUS	RESTARTS	AGE	
hellonode-c9576c598-gb917	1/1	Running	0	54s	
osboxes@osboxes:~\$ kubectl	logs -f he	- llonode-c9	576c598-ab9	17	
> HelloKube@1.0.0 start /a	ממו				
> node app is					
, node approb					
[2018-02-06 11:58:40.566]	[INFO] Hell	oKube – St	arting		
[2018-02-06 11:58:40.582]	[INFO] Hell	oKube – He	lloKube list	tening on port 8080	
[2018-02-06 11:58:43.573]	[INFO] Hell	oKube – <mark>la</mark>	lala		
[2018-02-06 11:58:44.325]	[INFO] Hell	oKube – <mark>op</mark>	eration /hea	alth invoked returning (GREEN
[2018-02-06 11:58:46.574]	[INFO] Hell	oKube – <mark>la</mark>	lala		
[2018-02-06 11:58:49.575]	[INFO] Hell	oKube – la	lala		

Take note of the number of restarts.

Invoke /infect method.

BM Clou	d Private >	× 🗋 192.10	58.142	2.100:	31491/	fir ×		
$\leftrightarrow \rightarrow G$	(i) 192.168.142	2.100 :31491	e,	☆	2	Θ	ABP	
I don't feel that go	od							

(See that the port does not change after restarting the pods, that is the objective of the kubernetes service objects, they automatically reflect and balance the traffic to the application configured in the service selector)

Look at the logs. After the method /health is invoked three times returning RED the Application will be restarted.

[2018-02-06	12:01:29.368]	[INFO]	HelloKube	_	operation	/infect	invoked		
[2018-02-06	12:01:31.855]	[INFO]	HelloKube		cough				
[2018-02-06	12:01:34.317]	[INFO]	HelloKube		operation	/health	invoked	returning	RED
[2018-02-06	12:01:34.855]	[INFO]	HelloKube		cough				
[2018-02-06	12:01:37.856]	[INFO]	HelloKube		cough				
[2018-02-06	12:01:40.859]	[INFO]	HelloKube		cough				
[2018-02-06	12:01:43.861]	[INFO]	HelloKube		cough				
[2018-02-06	12:01:44.318]	[INFO]	HelloKube		operation	/health	invoked	returning	RED
[2018-02-06	12:01:46.863]	[INFO]	HelloKube		cough				
[2018-02-06	12:01:49.867]	[INFO]	HelloKube		cough				
[2018-02-06	12:01:52.869]	[INFO]	HelloKube		cough				
[2018-02-06	12:01:54.319]	[INFO]	HelloKube		operation	/health	invoked	returning	RED
osboxes@osbo	oxes:~\$								

osboxes@osboxes:~\$ kubectl NAME	get pods READY	STATUS	RESTARTS	AGE	
osboxes@osboxes:~\$ kubectl	logs -f hel	llonode-c95	576c598-gb91	on L7	
> HelloKube@1.0.0 start /ap > node app.js	p				
[2018-02-06 12:02:04.270] [[2018-02-06 12:02:04.289] [INFO] Hello	oKube - Sta Kube - Hel	arting lloKube list	cening on port 8080	
[2018-02-06 12:02:04.326] [[2018-02-06 12:02:07.276] [INFO] Hello INFO] Hello	oKube – ope oKube – lal	eration /hea Lala	alth invoked returning	GREEN

AutoScale

Create a policy to scale your application, you can configure IBM Cloud private to scale the Application based on CPU Usage.

Before creating a new policy take note of the current number of replicas:

20 - it	20 🗸 items per page 1-1 of 1 items 1 of 1 pages <								
NAME 🗢	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME	ACTION		
hellonode	default	1	1	1	1	Feb 6th 2018 at 12:27 PM	• • •		

Go to Workload -> Scaling Policies



Create a new policy:



POLICY Create Policy			×
Name			•
hellonodescale			
Scale target			
hellonode			
Minimum replications			
2			
Maximum replications			
5			
Target CPU			
20			
			•
	Cancel	Create	

YAML:

apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
name: hellonodescale
namespace: default
spec:
scaleTargetRef:
kind: Deployment
name: hellonode
apiVersion: extensions/v1beta1
minReplicas: 2
maxReplicas: 5
targetCPUUtilizationPercentage: 20

If you go to the deployments you will see that kubernetes is scaling the application to 2 pods as it is the minimum specified in the new policy

Deployments

20 🔻 items	s per page 1-1 of 1 items							1 of 1 pag	(es 🔇	1	>
NAME 🛧	NAMES	SPACE DE	SIRED	CURRENT	READY	AVAILABLE	CREATION TIME	,	CTION		
hellonode	default	t 2	:	1	1	1	Feb 6th 2018 at 12:27 PM		:		
20 🔻 ite	ems per page 1-1 of 1 if	tems						1 of 1 pages	<	1	>
NAME -	NAMESPACE	DESIRED	CURRE	NT R	READY	AVAILABLE	CREATION TIME			ACTIO	N
hellonode	default	2	2	2	2	2	Feb 6th 2018 at	12:27 PM		:	

Now we are going to query the application in an infinity loop to generate load and make that the CPU consumed by the application grows beyond 20%. But first we need to set the CPU required by the application/deployment.

Deployı	ments						default		•
O, Search i	tems					с	reate Dep	loyme	nt C
20 🔻 ite	ems per page 1-1 of 1	. items				1 of 1 pages	<	1	>
NAME 🔺	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME		АСТІО	N
<u>hellonode</u>	default	2	2	2	2	Feb 6th 2018 at 12:27 PM		:	
						Edit			٦
						Scale			
					Remov	e .			



```
Query the application.
```

for ((i=1;10<=100;i++)); do curl -H "Connection: close" --connect-timeout 1 --connect-timeout 1 http://192.168.142.100:31491/; done

SBORESGESDAVES:-S for (1=1;10<=100;1+7); do curi - connection: close --connect-imedut i --connect-imedut i nctp://192.106.142.100;1431; ("m alive!!!I'm alive!!!

With the commands "kubectl describe horizontalpodautoscaler hellonodescale" you can get the status of the policy. You can see the current percentage of CPU used and the actions taken.

osboxes@osboxes:~	\$ kubect	l describe horizon	talpodautos	scaler hellonodescale	
Name:				hellonodescale	
Namespace:				default	
Labels:				<none></none>	
Annotations:				<none></none>	
CreationTimestamp				Wed, 07 Feb 2018 14:48:46 +0000	
Reference:				Deployment/hellonode	
Metrics:				(current / target)	
resource cpu on	pods (as a percentage of	request):	46% (46m) / 20%	
Min replicas:					
Max replicas:					
Conditions:					
туре		Reason	Message		
AbleToScale		SucceededRescale	the HPA co	ontroller was able to update the target scale to 4	
ScalingActive		ValidMetricFound	the HPA wa	as able to succesfully calculate a replica count from cpu resource utilization	(percentage of request)

20 🔻 it	ems per page 1-1 of 1	1 of 1 pages 🔸	1 >						
NAME 🔶	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME	ACTION		
hellonode	default	4	4	4	4	Feb 6th 2018 at 12:27 PM	:		
20 🔻 ite	ems per page 1-1 of 1	items				1 of 1 pages 💙	1 >		
NAME 🔶	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME	ACTION		
hellonode	default	5	5	5	5	Feb 6th 2018 at 12:27 PM	:		

From Kubernetes 1.6 it is possible to user more metrics than CPU % usage.

Rollouts. Update without down time.

https://kubernetes.io/docs/concepts/workloads/controllers/deployment/#updating-adeployment

In this section we are going to change the docker image version used by an application/deployment. Kubernetes does it progressively without downtime.

Remove the policy we created in the previous lab and set the number of replicas to 5. The biggest number of replicas the smoother the rollout will be.

Scaling F	Policies							defaul	t	•
O, Search ite	ms							Cre	eate Poli	cy 🔂
20 🔻 item	ns per page 1-1 of 1 it	ems				1	of 1 pages	<	1	>
NAME +	NAMESF	PACE REF	ERENCE	CPU%	MIN REPLICAS	MAX REP	PLICAS		ACTION	
hellonodescale	default	hell	onode	20	2	5			:	
							Edit Remove			
Deploym	nents							defaul	lt	•
O, Search ite	ms						(Create D	eployme	ent 🕈
20 🔻 item	ns per page 1-1 of 1 it	ems				1	of 1 pages	<	1	>
NAME 🛎	NAMESPACE	DESIRED	URRENT	READY	AVAILABLE	CREATION TIME			ACTIO	N
<u>hellonode</u>	default	2	2	2	2	Feb 6th 2018 at 12	:27 PM		:	
							Edit Scale Remov	/e		
20 🔻 iten	ns per page 1-1 of 1 it	ems				1	of 1 pages	<	1	>
NAME 🔺	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME			ACTI	ON
hellonode	default	5	5	5	5	Feb 6th 2018 at 12	2:27 PM		:	

We are going to create a new version of our hellonode docker image used in Lab3.

Go to the command line and edit the file ./node/app.js

```
osboxes@osboxes:~$ pwd
/home/osboxes
osboxes@osboxes:~$ vi ./node/app.js
osboxes@osboxes:~$ cat ./node/app.js
...
app.get("/", function(req, res, next){
logger.info("operation / invoked...");
```

```
res.send("I'm alive - 2 !!! ");
});
```

•••

```
Create the docker image with version tag: 2.0
```

osboxes@osboxes:~\$ sudo docker build -t mycluster.icp:8500/default/hellonode:2.0 -f hellonode.dockerfile .

Push the image to IBM Cloud private image repository:

osboxes@osboxes:~\$ sudo docker push mycluster.icp:8500/default/hellonode:2.0

Check the available images in our repository:

Images / default/hellonode /	
default/hellonode	
Overview	
Image details	
Туре	Detail
Type Name	Detail default/hellonode
Type Name Owner	Detail default/hellonode default
Type Name Owner Scope	Detail default/hellonode default namespace
Type Name Owner Scope Tags	Detail default/hellonode default namespace 1.0, 2.0

To be able to rollout the application without downtime we need to add the property "readinessprobe" to our application/deployment.

This property tells kubernetes how to check when the application is ready so kubernetes will enable the new pod IP in the service object only when the app is ready and not when just the container is ready.

Edit the Application:

Deployments default -O_ Search items Create Deployment 🛨 1 of 1 pages 🛛 🔨 20 🔻 items per page | 1-1 of 1 items 1 > NAME 🔺 NAMESPACE DESIRED CURRENT READY AVAILABLE CREATION TIME ACTION : Feb 6th 2018 at 12:27 PM <u>hellonode</u> default 5 5 5 5 Scale Remove

Add the following element:

```
"readinessProbe": {
    "failureThreshold": 3,
    "httpGet": {
        "path": "/",
        "port": 8080,
        "scheme": "HTTP"
    },
    "initialDelaySeconds": 5,
    "periodSeconds": 2,
    "successThreshold": 1,
    "timeoutSeconds": 1
}
```

DE	EPLOYMENT	×
_		
E	dit Deployment	
	"template": {	
	"metadata": {	
24	"creationTimestamp": null,	
	"labels": {	
	"app": "hellonode"	
27	}	
	},	
	"spec": {	1.00
	"containers": [
	 {	
	"name": "hellonode",	
33	"image": "mycluster.icp:8500/default/hellonode:1.0",	- 11
34 -	"readinessProbe": {	
	"failureThreshold": 3,	
36 -	"httpGet": {	
	"path": "/",	
	"port": 8080,	
	"scheme": "HTTP"	
	Ъ	
	"initialDelaySeconds": 5,	
	"periodSeconds": 2,	
	"successThreshold": 1,	
44	"timeoutSeconds": 1	
	Ъ	
46 -	"resources": {	
47 -	"requests": {	
	"cpu": "100m",	_
	"memory": "100Mi"	- ×
		_
	Cancel Submit	
	Cancer Sublint	

Double check you removed the scaling policy and you scale the application to 5 instances.

Open one terminal and execute the command: kubectl rollout status deployment/hellonode

```
osboxes@osboxes:~$ kubectl rollout status deployment/hellonode
Waiting for rollout to finish: 4 of 5 updated replicas are available...
deployment "hellonode" successfully rolled out
```

You see the result of our last update (rollout).

Open other terminal and query the application:

for ((i=1;10<=100;i++)); do curl -H "Connection: close" --connect-timeout 1 --connect-timeout 1 http://192.168.142.100:31491/; done

osboxe8posboxes:~\$ for ((i=1:10<=100;i+1); do curl -H "Connection: close" --connect-timeout 1 --connect-timeout 1 http://192.168.142.100:31491/; done I'm alive!!!I'm
I'm alive!!! is the only message corresponding to the version 1.

Edit the application/deployment and change the docker image version.

Deployn	nents					c	lefault 👻
						_	
O _v Search ite	ems					Cre	ate Deployment +
20 🔻 iter	ms per page 1-1 of 1 i	items				1 of 1 pages	(1 >
NAME 🔺	NAMESPACE	DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME	ACTION
hellonode	default	5	5	5	5	Feb 6th 2018 at 12:27 PM	*
						Edit	
						Edit Scale	



Execute again the command kubectl rollout status deployment/hellonode, you will see how kubernetes rollout your application

osboxes	osbo	oxes:~\$]	cube	ectl roll	Loi	it st	tatı	lS	depl	loyment/he	ellon	ode	
Waiting	for	rollout	to	finish:	2	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	2	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	2	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	3	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	3	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	4	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	4	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	4	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	4	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	4	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	4	out	of	5	new	replicas	have	been	updated
Waiting	for	rollout	to	finish:	1	old	rep	pli	icas	are pendi	ing te	ermina	ation
Waiting	for	rollout	to	finish:	1	old	rep	pli	icas	are pendi	ing te	ermina	ation
Waiting	for	rollout	to	finish:	4	of S	5 ur	pda	ated	replicas	are a	availa	able
Waiting	for	rollout	to	finish:	3	of S	5 up	pda	ated	replicas	are a	availa	able
Waiting	for	rollout	to	finish:	4	of S	5 up	pda	ated	replicas	are a	availa	able
deployme	ent '	'hellonod	de"	successf	Eul	lly i	roll	Leo	d out				

And in the logs you will see first a mix of messages and then only I'm alive -2 !!! messages.

", alive!!!I'm alive - 2 !!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive! 'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive - 2 !!!I'm a I'm alive!!!I'm alive!!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!!I'm alive!!!!I'm alive!!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!I'm alive!!!I'm alive!!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!!I'm alive!!!I'm alive!!!!I'm alive!!!! live - 2 !!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive alive!!!I'm alive - 2 !!!I'm alive!!!I'm alive!!!I'm alive - 2 !!!I'm alive - 2 !!!I'm alive ve!!!I'm alive - 2 !!!I'm alive!!!I'm alive!!!I'm alive - 2 !!!I'm alive!!!I'm alive!!!I'm al alive!!!I'm alive!!!I'm alive - 2 !!!I'm alive!!!I'm alive!!!I'm alive - 2 !!!I'm !!!I'm alive!!!I'm alive - 2 !!!I'm alive!!!I'm alive!!!I'm alive - 2 !!!I'm alive!!!I'm alive live!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive!!!I' alive!!!I'm alive - 2 !!!I'm alive!!!I'm alive - 2 !!!I'm alive!!!I'm alive - 2 !!!I'm alive ve - 2 !!!I'm alive!!!I'm alive!!!I'm alive!!!I'm alive - 2 !!!I'm alive!!!I'm alive!!!I'm al !!!I'm alive!!!I'm alive - 2 !!!I'm alive - 2 !!!I'm alive!!!I'm alive!!!I'm alive - 2 !!!I'm !!!I'm alive - 2 !!!I'm alive!!!I'm alive!!!I'm alive - 2 !!!I'm alive!!!I'm alive!!!I'm aliv I'm alive - 2 !!!I'm alive - 2 !!!I'm alive!!!I'm alive - 2 !!!I'm alive - 2 !!!I'm alive -2 alive - 2 !!!I'm alive -

!!!I'm alive - 2 !!!!I'm alive - 2 !!!I'm alive - 2 !!!!I'm a

We can control how the rollout update is done with the property "strategy". For example:

"strategy": {

```
"type": "RollingUpdate",
"rollingUpdate": {
    "maxUnavailable": 1,
    "maxSurge": 1
}
```

This element means that during a rollout as maximum (maxSurge) there can only be 1 more pod than the replicas configured. And that as maximum (maxUnavailabe) there can only be 1 pod less available than the replicas configured.

So in the Application status we will never see more than 6 as current and less than 4 as available.

Q Search items 20 • items per page 1-16 of 16 items 1 of 1 pages < 1 TYPE SOURCE COUNT REASON MESSAGE FIRST SEEN - LAST SEEN Normal deployment- controller 7 ScalingRepticaSet (combined from similar events): Scaled down replica set hellonode-5699dc65879 to 0 Feb 7th 2018 at 4:02 PM Feb 7th 2018 at 4:22 PM Feb 7th 2018 at 4:22 PM Normal deployment- controller 2 ScalingRepticaSet Scaled up replica set hellonode-85d98c545f to 5 Feb 7th 2018 at 4:22 PM Feb 7th 2018 at 4:22 PM Normal deployment- controller 2 ScalingRepticaSet Scaled down replica set hellonode-85d98c545f to 2 Feb 7th 2018 at 4:29 PM Feb 7th 2018 4:29 PM Normal deployment- controller 1 ScalingRepticaSet Scaled down replica set hellonode-5d9dc65879 to 1 Feb 7th 2018 at 4:29 PM Feb 7th 2018 4:29 PM Normal deployment- controller 1 ScalingRepticaSet Scaled down replica set hellonode-6d9dc65879 to 2 Feb 7th 2018 4:29 PM Feb 7th 2018 4:29 PM Normal deployment- controller 1 ScalingRepticaSet Scaled up replica set hellonode-5d9dc65879 to 2 Feb 7th 2018 4:29 PM Feb 7th 2018 4:29 PM	
Q_ Search items 20 items per page 1-16 of 16 items 1 of 1 pages 1 TYPE SOURCE COUNT REASON MESSAGE FIRST SEEN * LAST SEEN * Normal deployment- controller 7 ScalingReplicaSet (combined from similar events): Scaled down replica set hellonode-5d9dc65879 to 0 Feb 7th 2018 at 4:05 PM Feb 7th 2018 at 4:25 PM Feb 7th 2018 at 4:24 PM Normal deployment- controller 2 ScalingReplicaSet Scaled up replica set hellonode-85d98c545f to 5 Feb 7th 2018 at 4:28 PM Feb 7th 2018 4:28 PM Normal deployment- controller 2 ScalingReplicaSet Scaled up replica set hellonode-85d98c545f to 2 Feb 7th 2018 at 4:29 PM Feb 7th 2018 at 4:29 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-65d9dc65879 to 1 Feb 7th 2018 at 4:29 PM Feb 7th 2018 at 4:29 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-65d9dc65879 to 2 Feb 7th 2018 at 4:29 PM Feb 7th 2018 at 4:29 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-65d9dc65879 to 2 Feb 7th 2018 at 4:29 PM Feb 7th 2018 at	
20 • items per page 1-16 of 16 items 1 of 1 pages < 1	
TYPESOURCECOUNTREASONMESSAGEFIRST SEEN *LAST SEENNormaldeployment- controller7ScalingReplicaSet(combined from similar events): Scaled down replica set hellonode-5d9dc65879 to 0Feb 7th 2018 at 4/22 PMFeb 7th 2018 at 4/22 PMFeb 7th 2018 at 4/22 PMNormaldeployment- controller2ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 5Feb 7th 2018 at 4/23 PMFeb 7th 2018 4/24 PMNormaldeployment- controller2ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 2Feb 7th 2018 at 4/23 PMFeb 7th 2018 4/23 PMNormaldeployment- controller1ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 1Feb 7th 2018 at 4/29 PMNormaldeployment- controller1ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 1Feb 7th 2018 at 4/29 PMNormaldeployment- controller1ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 1Feb 7th 2018 at 4/29 PMNormaldeployment- controller1ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4/29 PMNormaldeployment- controller1ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4/29 PMNormaldeployment- controller1ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4/29 PMNormaldeployment- controlle	
Normal controllerdeployment- controller7ScalingReplicaSet hellonode-5d9dc65879 to 0Feb 7th 2018 at 4:05 PMFeb 7th 2018 at 4:25 PMNormal deployment- controller2ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 5Feb 7th 2018 at 4:23 PMFeb 7th 2018 at 4:24 PMNormal deployment- controller2ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 2Feb 7th 2018 at 4:23 PMFeb 7th 2018 at 4:24 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-65d9dc65879 to 1Feb 7th 2018 at 4:29 PMFeb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 1Feb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-65d9dc65879 to 2Feb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:29 PMNormal deployment- controller1Sca	
Normal controller2ScalingReplicaSet Scaled up replica set hellonode-85d98c545f to 5Feb 7th 2018 at 4:23 PMFeb 7th 2018 4:24 PMNormal deployment- controller2ScalingReplicaSet Scaled down replica set hellonode-85d98c545f to 2Feb 7th 2018 at 4:23 PMFeb 7th 2018 4:28 PMNormal deployment- controller1ScalingReplicaSet Scaled down replica set hellonode-5d9dc65879 to 1Feb 7th 2018 at 4:29 PMFeb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSet Scaled down replica set hellonode-85d98c545f to 1Feb 7th 2018 at 4:29 PMFeb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSet Scaled down replica set hellonode-85d98c545f to 1Feb 7th 2018 at 4:29 PMFeb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSet Scaled down replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:29 PMFeb 7th 2018 4:29 PMNormal deployment- controller1ScalingReplicaSet Scaled down replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:29 PMFeb 7th 2018 4:29 PMNormal deployment- controller1ScalingReplicaSet Scaled up replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:43 PMFeb 7th 2018 4:43 PMNormal deployment- controller1ScalingReplicaSet Scaled up replica set hellonode-5d9dc65879 to 4Feb 7th 2018 at 4:43 PMNormal deployment- controller1ScalingReplicaSet Scaled up replica set hellonode-65d9dc65879 to 5Feb 7th 2018 at	at
Normal controller2ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 2Feb 7th 2018 at 4:23 PMFeb 7th 2018 4:28 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-5d9dc65879 to 1Feb 7th 2018 at 4:29 PMFeb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled down replica set hellonode-5d9dc65879 to 1Feb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 1Feb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-5d9dc65879 to 2Feb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-65d9dc65879 to 4Feb 7th 2018 at 4:43 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-65d9dc65879 to 5Feb 7th 2018 at 4:43 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-665cff556 to 1Feb 7th 2018 at 4:46 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-665cff556 to 1Feb	at
Normal controllerScalingReplicaSetScaled up replica set hellonode-5d9dc65879 to 1Feb 7th 2018 at 4:29 PMFeb 7th 2018 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 1Feb 7th 2018 at 4:29 PMFeb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-85d98c545f to 1Feb 7th 2018 at 4:29 PMFeb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:29 PMFeb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled down replica set hellonode-85d98c545f to 0Feb 7th 2018 at 4:29 PMFeb 7th 2018 at 4:29 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-5d9dc65879 to 4Feb 7th 2018 at 4:43 PMFeb 7th 2018 4:43 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-5d9dc65879 to 5Feb 7th 2018 at 4:46 PMFeb 7th 2018 at 4:46 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-665cff556 to 1Feb 7th 2018 at 4:54 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-65d9dc65879 to 4Feb 7th 2018 at 4:54 PMNormal deployment- controller1ScalingReplicaSetScaled up replica set hellonode-5d9dc65879 to 4Feb 7th 2018 at <td>at</td>	at
Normal controller1ScalingReplicaSet Scaled down replica set hellonode-85d98c545f to 1Feb 7th 2018 at 4:29 PMFeb 7th 2018 at 	at
Normal controller deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-5d9dc65879 to 2 Feb 7th 2018 at 4:29 PM Feb 7th 2018 4:29 PM Normal deployment- controller 1 ScalingReplicaSet Scaled down replica set hellonode-85d98c545f to 0 Feb 7th 2018 at 4:29 PM Feb 7th 2018 at 4:29 PM Feb 7th 2018 at 4:29 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-5d9dc65879 to 4 Feb 7th 2018 at 4:43 PM Feb 7th 2018 4:43 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-5d9dc65879 to 5 Feb 7th 2018 at 4:46 PM Feb 7th 2018 at 4:46 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-665cff556 to 1 Feb 7th 2018 at 4:54 PM Feb 7th 2018 at 4:54 PM Normal deployment- controller 1 ScalingReplicaSet Scaled down replica set hellonode-65cff556 to 1 Feb 7th 2018 at 4:54 PM Feb 7th 2018 at 4:54 PM	at
Normal controller deployment- controller 1 ScalingReplicaSet Scaled down replica set hellonode-85d98c545f to 0 Feb 7th 2018 at 4:29 PM Feb 7th 2018 at 4:29 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-5d9dc65879 to 4 Feb 7th 2018 at 4:43 PM Feb 7th 2018 at 4:43 PM Feb 7th 2018 at 4:43 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-5d9dc65879 to 5 Feb 7th 2018 at 4:46 PM Feb 7th 2018 at 4:46 PM Normal controller 1 ScalingReplicaSet Scaled up replica set hellonode-665cff556 to 1 Feb 7th 2018 at 4:54 PM Feb 7th 2018 at 4:54 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-665cff556 to 1 Feb 7th 2018 at 4:54 PM Feb 7th 2018 at 4:54 PM	at
Normal controller 1 ScalingReplicaSet Scaled up replica set hellonode-5d9dc65879 to 4 Feb 7th 2018 at 4/43 PM Feb 7th 2018 at 4/43 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-5d9dc65879 to 5 Feb 7th 2018 at 4/46 PM Feb 7th 2018 4/46 PM Feb 7th 2018 4/46 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-665cff556 to 1 Feb 7th 2018 at 4/54 PM Feb 7th 2018 4/54 PM Normal deployment- controller 1 ScalingReplicaSet Scaled down replica set hellonode-665cff556 to 1 Feb 7th 2018 at 4/54 PM Feb 7th 2018 4/54 PM Normal deployment- controller 1 ScalingReplicaSet Scaled down replica set hellonode-5d9dc65879 to 4 Feb 7th 2018 at 4/54 PM Feb 7th 2018 4/54 PM	at
Normal controller deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-5d9dc65879 to 5 Feb 7th 2018 at 4:46 PM Feb 7th 2018 at 4:46 PM Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-665cff556 to 1 Feb 7th 2018 at 4:54 PM Feb 7th 2018 at 4:54 PM Normal deployment- controller 1 ScalingReplicaSet Scaled down replica set hellonode-5d9dc65879 to 4 Feb 7th 2018 at 4:54 PM Feb 7th 2018 at 4:54 PM	at
Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-665cff556 to 1 Feb 7th 2018 at 4:54 PM Feb 7th 2018 at 4:54 PM Normal deployment- controller 1 ScalingReplicaSet Scaled down replica set hellonode-5d9dc65879 to 4 Feb 7th 2018 at 4:54 PM Feb 7th 2018 at 4:54 PM	at
Normal deployment- controller 1 ScalingReplicaSet Scaled down replica set hellonode-5d9dc65879 to 4 Feb 7th 2018 at 4:54 PM 4:54 PM	at
	at
Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-665cff556 to 2 Feb 7th 2018 at 4:54 PM 4:54 PM	at
Normal deployment- controller 1 ScalingReplicaSet Scaled down replica set hellonode-5d9dc65879 to 3 Feb 7th 2018 at 4:54 PM 4:54 PM	at
Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-665cff556 to 3 Feb 7th 2018 at 4:54 PM 4:54 PM	at
Normal deployment- controller 1 ScalingReplicaSet Scaled down replica set hellonode-5d9dc65879 to 2 Feb 7th 2018 at 4:55 PM 4:55 PM	at
Normal deployment- controller 1 ScalingReplicaSet Scaled up replica set hellonode-665cff556 to 4 Feb 7th 2018 at 4:55 PM 4:55 PM	at

In the application details:

Deployments / hellonode /

hellonode

We see the scale up and scale down events.
Lab 7. Logging

By default docker and kubernetes can access to the logs that a container prints to its STDOUT. There are also options to access to internal or custom containers logs.

https://www.ibm.com/support/knowledgecenter/en/SSBS6K_2.1.0/manage_metrics/logging_elk. html

Command line

List the pods.

kubectl get pods

osboxes@osboxes:~\$ kubectl	get pods			
NAME	READY	STATUS	RESTARTS	AGE
hellonode-665cff556-bs4ds	1/1	Running	0	17h
hellonode-665cff556-f8b7b	1/1	Running	1	17h
hellonode-665cff556-pmn7s	1/1	Running	0	17h
hellonode-665cff556-pv7g2	1/1	Running	0	1m
hellonode-665cff556-qdjhr	1/1	Running	1	17h

Tail the logs.

kubectl logs -f kubectl logs -f hellonode-665cff556-pv7g2

oshoves(loshoves •~\$	kubectl logs	-f hellonode.	-665cff556-pt	792	
0500x0500500x05. 9	KUDCCCI 1095	I IICIIOIIOUC	003CI1330 þ	1192	
Nucleowybed1 0 0	atant /ann				
> HelloRube@1.0.0	Start /app				
> node app.js					
[2018-02-08 09:09:	49.846] [INFO]	HelloKube -	Starting		
[2018-02-08 09:09:	50.189] [INFO]	HelloKube -	HelloKube li	stening on port	8080
[2018-02-08 09:09:	51.698] [INFO]	HelloKube -	operation /	invoked	
[2018-02-08 09:09:	52.980] [INFO]	HelloKube -	lalala		
[2018-02-08 09:09:	53.693] [INFO]	HelloKube -	operation /	invoked	
[2018-02-08 09:09:	55.709] [INFO]	HelloKube -	operation /	invoked	
[2018-02-08 09:09:	55.981] [INFO]	HelloKube -	lalala		
[2018-02-08 09:09:	57.692] [INFO]	HelloKube -	operation /	invoked	
[2018-02-08 09:09:	58.983] [INFO]	HelloKube -	lalala		
[2018-02-08 09:09:	59.691] [INFO]	HelloKube -	operation /	invoked	
[2018-02-08 09:10:	01.690] [INFO]	HelloKube -	operation /	invoked	
[2018-02-08 09:10:	01.983] [INFO]	HelloKube -	lalala		

ICP Console

Go to deployments and select your application/deployment.

Deployments					default	•
O ₆ Search items					Create Deployn	nent 🕂
20 🔻 items per page 1-1 of 1 items				1 of 1 pages	< 1	>
NAME A NAMESPACE DESIRED	CURRENT	READY	AVAILABLE	CREATION TIME	ACT	ION
hellonode default 5	5	5	5	Feb 6th 2018 at 12:27 PM		

Select the tab "Logs" and review the logs of the pods of your deployment.

Deployments / hellonode /	
hellonode	
Overview Events Logs	s
	Q, Search hellonode-665cff556-bs4ds → hellonode →
-[32m[2018-02-08 09:12:21.420] -[32m[2018-02-08 09:12:22.127] -[32m[2018-02-08 09:12:23.375] -[32m[2018-02-08 09:12:23.419] -[32m[2018-02-08 09:12:25.419] -[32m[2018-02-08 09:12:25.419] -[32m[2018-02-08 09:12:29.378] -[32m[2018-02-08 09:12:29.378] -[32m[2018-02-08 09:12:29.378] -[32m[2018-02-08 09:12:31.420] -[32m[2018-02-08 09:12:32.127] -[32m[2018-02-08 09:12:33.419] -[32m[2018-02-08 09:12:35.380] -[32m[2018-02-08 09:12:35.419] -[32m[2018-02-08 09:12:35.419] -[32m[2018-02-08 09:12:35.419] -[32m[2018-02-08 09:12:35.419] -[32m[2018-02-08 09:12:35.419] -[32m[2018-02-08 09:12:33.419] -[32m[2018-02-08 09:12:33.419] -[32m[2018-02-08 09:12:33.419] -[32m[2018-02-08 09:12:33.419] -[32m[2018-02-08 09:12:34.420] -[32m[2018-02-08 09:12:41.420] -[32m[2018-02-08 09:12:41.384] -[32m[2018-02-08 09:12:41.420] -[32m[2018-02-08 09:12:41.420] -[32m[2018-02-08 09:12:41.420] -[32m[2018-02-08 09:12:41.420]	<pre>[INFO] HEITOKODC = [JSMIDIGIT::: [INFO] HEITOKODC = [JSMIDIGIT::: [INFO] HEITOKODC = [JSMODeration / invoked returning GREEN [INFO] HEITOKUDE = [JSMODeration / invoked [INFO] HEITOKUDE = [JSMODERATION / invoked returning GREEN</pre>

ELK

With ICP a you have out of the box an ELK installation to manage your application logs.



Open Kibana UI

×	IBM Cloud Private
1	Dashboard
)	Catalog
+ 1	Workloads
• 1	Network Access
→ (Configuration
÷ 1	Platform
	Alerting
	Logging

The Kibana console is open. The first action is to configure an index:



Click on "Discover" and you will see the logs of all the containers in the cluster



Create a filter



Add filter			×
Filter	is 🔻	hellonode	Edit Query DSL
Label			
Optional			Cancel Save

And click "Add" to the log label

t kubernetes.pod	
t log add	
# offset	

Now you see all the logs of you application.



From the helm catalog in ICP you can deploy your own ELK instances with the custom scope and configuration needed in your infrastructure or projects

Lab 8. Monitoring

ICP uses out of the box Prometheus and Grafana to monitor the state of your Kubernetes cluster and the containers running on it

https://prometheus.io/

https://grafana.com/

Basically you visualize with Grafana the metrics collected in Prometheus.

Go to Monitoring

×	IBM Cloud Private
	Dashboard
•	Catalog
•	Workloads
•	Network Access
•	Configuration
-	Platform
	Alerting
	Logging
	Metering
	Monitoring

By default there are some dashboards created, access to "ICP 2.1 Performance" dashboard.



IBM Cloud Private × Image: Grafana - ICP 2.1 Perform × ← C A Not secure https://192.168.142.100:8443/grafana/dashboard/file/ICP2.1-Performance.js Q A (interval	: C ards
← → C ▲ Not secure https://192.168.142.100:8443/grafana/dashboard/file/ICP2.1-Performance.js Q ☆ ☆ ■ ICP 2.1 Performance IBM Provided 2.5 • C ♦<	2 ards
Image: Sime state ICP 2.1 Performance IBM Provided 2.5 • Image: Sime state Image: Sime state	C ards
interval 5m • C Kibana = DashBo	ards
Voungest Node Lintime Total memory Available Memory Memory Free ICP Total CPU 15 Minute Ava	
Available wennory mennory rece internet of sminute wennory mennory rece Average Machine Visit f	the
3.13 day 7.79 GiB 1.76 GiB 23% 35%	D- iit
4 Page	a
Oldest Node Uptime Total Disk Space Disk Space Available Disk Space Used Active Containers Last 5 Active Pods ICP N min Cour	ode int
3.13 day 389.87 GiB 172.11 GiB 57.0% 48 112 1	
	al
Memory by node Top 5 Containers by CPU Top 5 Container by Memory	emory
192.168.142.100:9100 7 k8s_jcp-ds_jcp-ds_0_kube-system_b5901ed4- 0a4b-11e8-8c84-000c29951ca0_0 kube- system 46.46 k8s_prometheus_monitoring-prometheus- 77d4df9dd6-nhv6z_kube-system_a93c1357- 04cs11a8-8c84-000c29051ca0_0 kube- system 1.	10 GE
k8s_logstash_logstash-5ccb9849d6- kube- system k4.89 k8s_logstash_logstash-5ccb9849d6- kube- system k8s_logstash_logstash-5ccb9849d6- kube- system k1.89 <	1.32 B
k8s_es-data_elasticsearch-data-0_kube- system_4f829173-0a4c-11e8-8c84- 000c29951ca0_0 kube- system 9.38 k8s_icp-ds_icp-ds-0_kube-system_b5901ed4- 0a4b-11e8-8c84-000c29951ca0_0 kube- system 82	5.34 B
k8s_apiserver_k8s-master- kube- system_1ed4bf7511f8789dfb50607f7e16b471_0 k0e- system_24829173-0a4c-11e8-8c84- kube- system	14 GE
Container CPU Utilization	
100	
80	
60 + + + + + + + + + + + + + + + + + + +	5
40 k8_calico-node-amd64_calico-node-amd64_gc462_kUbe-system_7ef62cc8/0a4b-11e8-8c84-000c29951ca0_0	
42	

Import a new dashboard.



In a browser load the page <u>https://github.com/jxadro/ICP_PoT/blob/master/App%20Monitoring-1511965974320.json</u> and copy the json.

📮 jxadro /	ICP_PoT			
<> Code	() Issues 0	1) Pull requests 0	Projects 0	🔳 Wiki
Branch: mast	ter ▼ ICP_PoT	/ App Monitoring-1	511965974320.js	on
jxadro 1	L			
1 contributo)r			
768 lines	(768 sloc) 19.6	КВ		
1 {				
2 "	inputs": [
3	{			
4	"name": "DS_PF	ROMETHEUS",		
5	"label": "prom	netheus",		
6	"description":	:"",		
7	"type": "datas	source",		
8	"pluginId": "p	prometheus",		
9	"pluginName":	"Prometheus"		
10	}			
1 11 1				

Paste the JSON in the dashboard import section

1 Import Dashboard

🖹 Import

🌲 Upload .json File	2		
Grafana.com Dashł	board		
Or paste ISON			
"inputs": [{ "name": "DS_PR "label": "promet "description": ""	OMETHEUS theus",	n,	
"type": "datasou "elugiold": "ereg	irce", mothous"		5
"type": "datasou "alwriald": "avor	hboard		2
"type": "datasou "elugiald": "arag Load	hboard		2
"type": "datasou "elugiald": "orag Load Import Dask Options Name	hboard	App Monitoring	

Back

Cancel

×



It is also possible to add datasources to Grafana additionally to the default "prometheus" data source.

Lab 9. Alerts

https://www.ibm.com/support/knowledgecenter/en/SSBS6K_2.1.0/manage_metrics/monitoring_service.html

Access to the alerting console.

×	IBM Cloud Private
	Dashboard
,	Catalog
,	Workloads
,	Network Access
,	Configuration
•	Platform
	Alerting
Alerti	manager Alerts Silences Statu
Filt	Group
	en metekon en opur"production"
Cust	om matcher, e.g. env="production"

No alerts found

By default there are no alerts defined. You can define alerts using any of the metrics collected by Prometheus.

First we are going to create new alerts and then we will explore how to visualize the available metrics.

To create new rules we have do it through ConfigMap objects. Edit the rules ConfigMap.

×	IBM Cloud Private
I	Dashboard
•	Catalog
+	Workloads
•	Network Access
~ (Configuration
	ConfigMaps
Config	Maps

O Search items				Create	ConfigMa	ap 🕁
20 🔻 items per page 1-20 of 30 items			1 of 2 pages	<	1	>
NAME -	NAMESPACE	CREATION TIME		ļ	ACTION	
alert-rules	kube-system	Feb 5th 2018 at 9:14 AM			:	
alertmanager-router-nginx-config	kube-system	Feb 5th 2018 at 9:14 AM	Edit			
<u>calico-config</u>	kube-system	Feb 5th 2018 at 9:06 AM	Remove			

Add the following rules:

```
"data": {
```

```
"sample.rules": "ALERT NodeMemoryUsage\n IF (((node_memory_MemTotal-
node_memory_MemFree-node_memory_Cached)/(node_memory_MemTotal)*100)) >
25\n FOR 1m\n LABELS {\n severity=\"page\"\n }\n ANNOTATIONS {\n
SUMMARY = \"{{$labels.instance}}: High memory usage detected\",\n
DESCRIPTION = \"{{$labels.instance}}: Memory usage is above 75% (current
value is: {{ $value }})\"\n }\nALERT HighCPUUsage\n IF
((sum(node_cpu{mode=~\"user|nice|system|irq|softirq|steal|idle|iowait\"})
by (instance, job)) - ( sum(node_cpu{mode=~\"idle|iowait\"}) by
(instance, job)))/(sum(node_cpu{mode=~\"user|nice|system|irq|softirq|steal
|idle|iowait\"}) by (instance, job)) * 100 > 2\n FOR 1m\n LABELS { \n
service = \"backend\" \n }\n ANNOTATIONS {\n summary = \"High CPU
Usage\",\n description = \"This computer has really high CPU usage for
over 10m\",\n }"
```



Now give it some minutes and refresh the alerting console.

Alertmanager Alerts Silences Status	New Silence
Filter Group Receiver: All S	how Silenced
	Add
Custom matcher, e.g. env="production"	
alertname="HighCPUUsage"	
12:14:37, 2018-02-08 + Info 🛃 Source 🎘 Silence	
service-backend Job-kubernetes-service-endpoints Instance-192, 100, 142, 100, 9100	
alertname="NodeMemoryUsage"	
12:14:37, 2018-02-08 🕂 Info 🛃 Source 🔏 Silence	
severity="page" kubernetes_namespace="kube-system" kubernetes_name="monitoring-prometheus-nodeexporter"	
job="kubernetes-service-endpoints" instance="192.168.142.100:9100" component="nodeexporter" app="monitoring-pro	ometheus"

To view the metrics you have to access to Prometheus console. By default it is not exposed so we have to create a service to expose it.

Click on "Create resource" and create the following service:



Review the NodePort assigned:

kubectl get service monitoring-prometheus-nodeport -n kube-system

osboxes@osboxes:~\$ kubectl get	service moni	toring-promet	heus-nodeport	-n kube-system	
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT (S)	AGE
monitoring-prometheus-nodeport	NodePort	10.0.0.78	<none></none>	9090: <mark>32038</mark> TCP	11m

Access to the Prometheus console:

👼 🗅 IBM Cloud Private 🛛 🖌 🌖 Prometheus Time Series 🗙 🕒 Alertmanager 🛛 🗙					
$\leftarrow \rightarrow G$	③ Not secure 192.168.142	. 100 :32038/graph?g0.range	_input=1h&g0.stacked=0&g0.e	expr=node_cp 🍳 🛧 🔊	
Prometheus	Alerts Graph Status - F	lelp			
node_cpu					
Execute Graph C 150K 100K	node_cpu kubelet_pod_worker_start_latency_micro kubelet_running_container_count kubelet_running_pod_count kubelet_runtime_operations kubelet_runtime_operations_latency_mic kubelet_runtime_operations_latency_mic kubelet_runtime_operations_latency_mic kubelet_runtime_operations_latency_mic kubelet_volume_stats_available_bytes kubelet_volume_stats_inodes kubelet_volume_stats_inodes_free kubelet_	pseconds_sum	s) O stacked	Load time: 3090ms Resolution: 14s Total time series: 36	
	11:15	11:30	11:45	12:00	

Alerts can be pushed to different systems or generally to an HTTP Service.

https://prometheus.io/docs/alerting/configuration/

APPENDIX

Lab3 App.js

```
var express = require('express');
var log4js = require('log4js');
var http = require('http');
var app = express();
var logger = log4js.getLogger('HelloKube');
var isHealthy = true;
```

logger.info("Starting...");

setInterval(function(){

```
if(isHealthy)
        {
                logger.info("lalala....");
        }
        else
        {
                 logger.info("cough...");
        }
},3000);
app.get("/", function(req, res, next){
        logger.info("operation / invoked...");
        res.send("I'm alive!!!");
});
app.get("/env", function(req, res, next){
        res.json(process.env);
});
app.get("/health", function(req, res, next){
        if(isHealthy)
        {
                logger.info("operation /health invoked... returning GREEN");
                res.send("GREEN");
        }
        else
        {
                 logger.info("operation /health invoked... returning RED");
                 res.status(500).send("RED");
        }
});
app.get("/infect", function(req, res, next){
        logger.info("operation /infect invoked...");
        isHealthy = false;
        res.send("I don't feel that good...");
});
app.get("/kill", function(req, res, next){
        res.send("You are dead...");
        process.exit();
});
```

```
var port = process.env.PORT || 8080;
```

```
app.listen(port, function(){
```

```
logger.info('HelloKube listening on port ' + port);
```

});

Lab 3. package.json

```
{
    "name": "HelloKube",
    "main": "app.js",
    "description": "Listeneint API Connect API Events",
    "version": "1.0.0",
    "private": false,
    "scripts": {
        "start": "node app.js"
    },
    "dependencies": {
            "express": "~4.2.0",
            "log4js": "~0.6.15"
    }
}
```