IBM Cloud 用戶實作研習營

IBM Cloud Kubernetes Service (IKS)
使用教學

2019/8/7

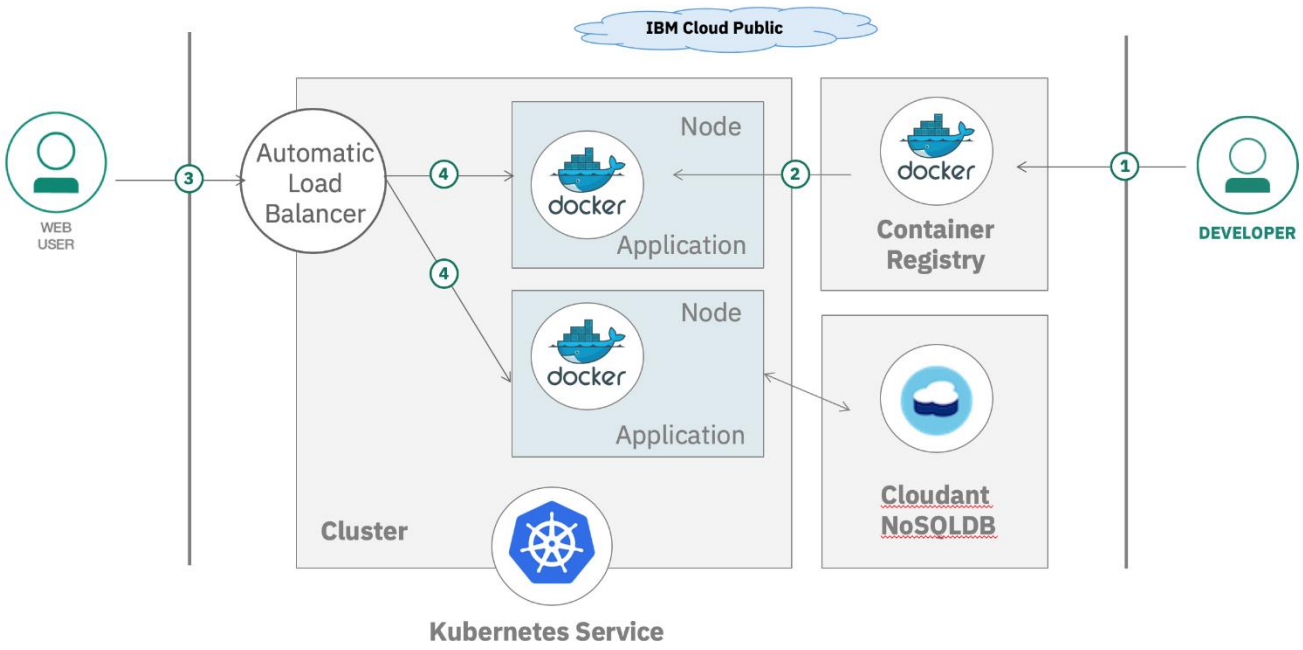
Introduction

- In this lab, you act as a developer. You configure a custom Kubernetes cluster that is used to deploy and test a web application for managing todos. The front end is written in Angular and the reminders are being stored in a Cloudant NoSQL dababase. All run on Node.js, in Docker container managed by Kubernetes.

Objectives

- Create a cluster with 1 worker pool that has 2 worker node.
- Install the CLIs for running Kubernetes commands and managing Docker images in IBM Cloud Container Registry.
- Create a private image repository in IBM Cloud Container Registry to store your images.
- Add the Cloudant NoSQL DB service to the cluster so that any app in the cluster can use that service.

Architecture



- A developer clone an existing web application from Github.
- Run the application locally and produce a Docker container image.
- The image is pushed to a namespace in the Container Registry Service.
- The application is deployed to a Kubernetes cluster.
- Users access the application.

Pre-Requisites

- Get a IBM Cloud account
- Install the IBM Cloud CLI(Reference Appendix)
- Install docker for Mac or Windows
- Install Kubectl
- Install a Git client
- Install Node.js

Verify the plugins installation

To verify that the plug-in has been installed properly, run the following command with administrator privilege.

Ibmcloud plugin list

```
C:\windows\system32>ibmcloud plugin  list
正在列出已安裝的外掛程式...

外掛程式名稱                                版本      狀態
cloud-functions/wsk/functions/fn            1.0.32
container-registry                          0.1.395   有更新可用
container-service/kubernetes-service        0.3.58    有更新可用
dev                                         2.3.0     有更新可用
sdk-gen                                     0.1.12
vpc-infrastructure/infrastructure-service   0.4.5     有更新可用
```

Create account and get cluster

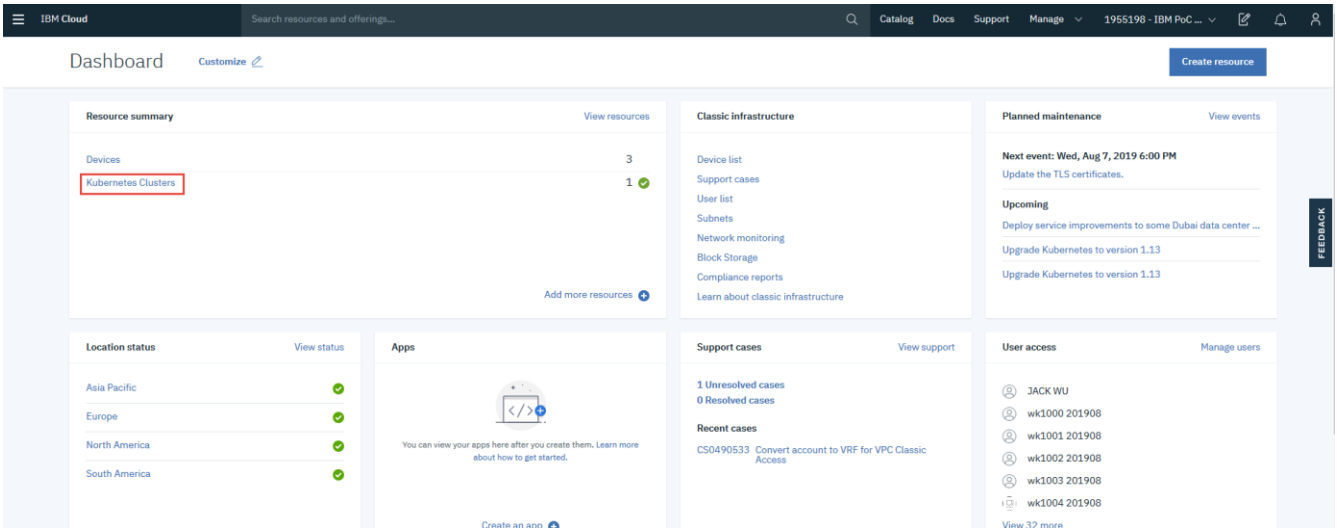
In this section, you will create your own IBM Cloud account, and then get access to a IBM Cloud Lab account which contains pre-provisioned clusters. Each lab attendee will be granted access to one cluster.

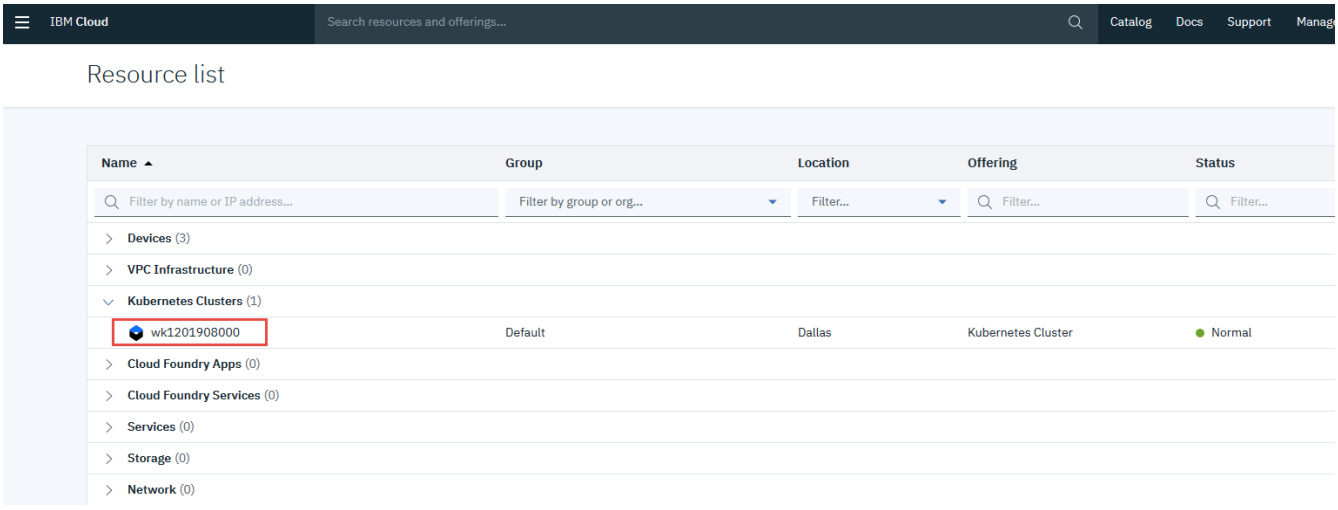
Login to IBM Cloud

1. Create your own IBM Cloud account. Ex: wk1201908000@dayrep.com
2. After the email verification, confirm by logging in to https://cloud.ibm.com

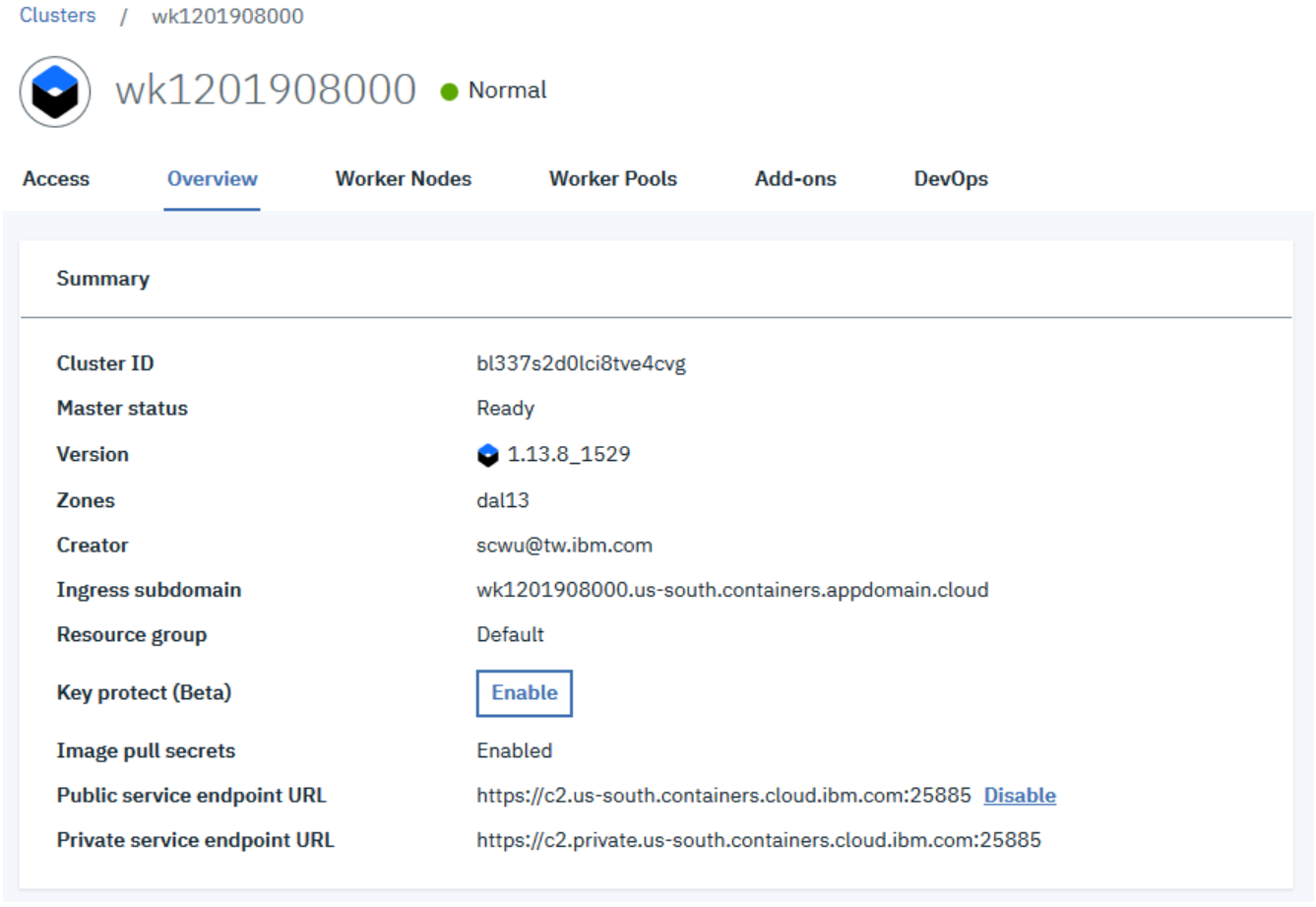
Get a Kubernetes cluster

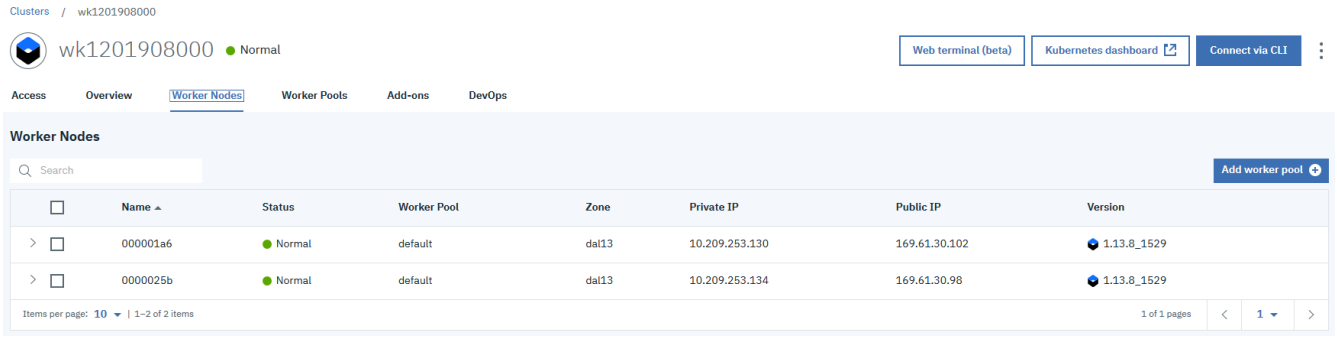
1. You can see the Kubernetes Clusters item in Resource summary on the Dashboard page. Click Kubernetes Clusters item and find your kubernetes cluster, ex: wk1201908000

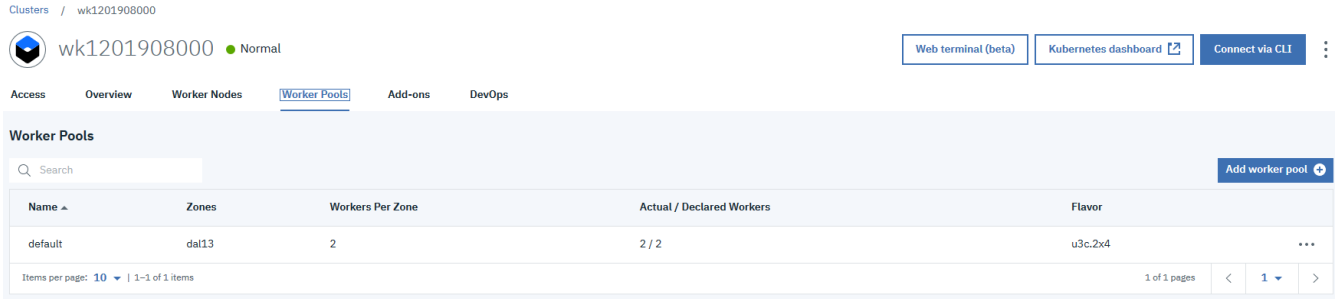




2. Click your kubernetes cluster, ex: wk1201908000 to see the detail.







3. Use below command to login IBM Cloud in the command prompt window.

ibmcloud login -a cloud.ibm.com -r us-south -g Default

```
已設定資源群組 Default 的目標

已設定地區 us-south 的目標


API 端點：      https://cloud.ibm.com
地區：          us-south
使用者：        scwu@tw.ibm.com
帳戶：          IBM PoC - WK1_201908 (85afb2e3d6c545c4b08e3901707cab07) <-> 1955198
資源群組：      Default
CF API 端點：
組織：
空間：

提示：如果您正在管理 Cloud Foundry 應用程式及服務
- 請使用 'ibmcloud target --cf' 以互動方式設定目標 Cloud Foundry 組織/空間，或使用 'ibmcloud target --cf-api ENDPOINT -o
 ORG -s SPACE' 設定目標組織/空間。
- 如果您想要執行 Cloud Foundry CLI 搭配現行 IBM Cloud CLI 環境定義，請使用 'ibmcloud cf'。
```

4. Use below command to check kubernetes cluster status in the command prompt window. You should see your cluster.

ibmcloud ks clusters

```
C:\windows\system32>ibmcloud ks clusters
OK
名稱            ID                   狀態     建立時間      工作程式   位置     版本          資源群組名稱   Provider
wk1201908000    b1337s2d0lci8tve4cvg   normal   6 hours ago   2          Dallas   1.13.8_1529   Default        classic
```

5. Use below command to get the cluster config from IBM Cloud and set up the environment variable in the command prompt window.

ibmcloud ks cluster-config --cluster wk1201908000

```
C:\windows\system32>ibmcloud ks cluster-config --cluster wk1201908000
OK
已順利下載 wk1201908000 的配置。

匯出環境變數以開始使用 Kubernetes。

PowerShell
$env:KUBECONFIG = "C:\Users\JACKWU\.bluemix\plugins\container-service\clusters\wk1201908000\kube-config-dal13-wk12019080
00.yml"

Command Prompt
SET KUBECONFIG=C:\Users\JACKWU\.bluemix\plugins\container-service\clusters\wk1201908000\kube-config-dal13-wk1201908000.y
ml
```

SET KUBECONFIG=C:\Users\JACKWU\.bluemix\plugins\container-service\clusters\wk1201908000\kube-config-dal13-wk1201908000.yml

```
C:\windows\system32>SET KUBECONFIG=C:\Users\JACKWU\.bluemix\plugins\container-service\clusters\wk1201908000\kube-config
-dal13-wk1201908000.yml

C:\windows\system32>
```

6. Use below command to test the client environment in the command prompt window.

kubectl get nodes

```
C:\windows\system32>kubectl get nodes
NAME             STATUS    ROLES     AGE       VERSION
10.209.253.130   Ready     <none>    5h        v1.13.8+IKS
10.209.253.134   Ready     <none>    5h        v1.13.8+IKS
```

kubectl cluster-info

```
C:\windows\system32>kubectl cluster-info
Kubernetes master is running at https://c2.us-south.containers.cloud.ibm.com:25885
CoreDNS is running at https://c2.us-south.containers.cloud.ibm.com:25885/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
kubernetes-dashboard is running at https://c2.us-south.containers.cloud.ibm.com:25885/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy
Metrics-server is running at https://c2.us-south.containers.cloud.ibm.com:25885/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

Get and build the application code

1. Change the working directory to c:\iks
2. Use below command to clone application from github

git clone https://github.com/lionelmace/mytodo

```
C:\IKS>git clone https://github.com/lionelmace/mytodo
Cloning into 'mytodo'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 812 (delta 15), reused 9 (delta 4), pack-reused 780Receiving objects:  99% (804/812), 564.01 KiB | 266.00
Receiving objects: 100% (812/812), 661.23 KiB | 284.00 KiB/s, done.

Resolving deltas: 100% (450/450), done.
```

3. Use below command to change directory and install npm

cd mytodo
npm install

```
C:\IKS\mytodo>npm install
npm WARN mytodo@2.0.0 No repository field.

added 151 packages from 172 contributors and audited 554 packages in 3.935s
found 3 high severity vulnerabilities
  run `npm audit fix` to fix them, or `npm audit` for details
```

4. Use below command to fix npm warnings

npm audit fix

```
C:\IKS\mytodo>npm audit fix
npm WARN mytodo@2.0.0 No repository field.

updated 2 packages in 2.626s
fixed 3 of 3 vulnerabilities in 554 scanned packages
```

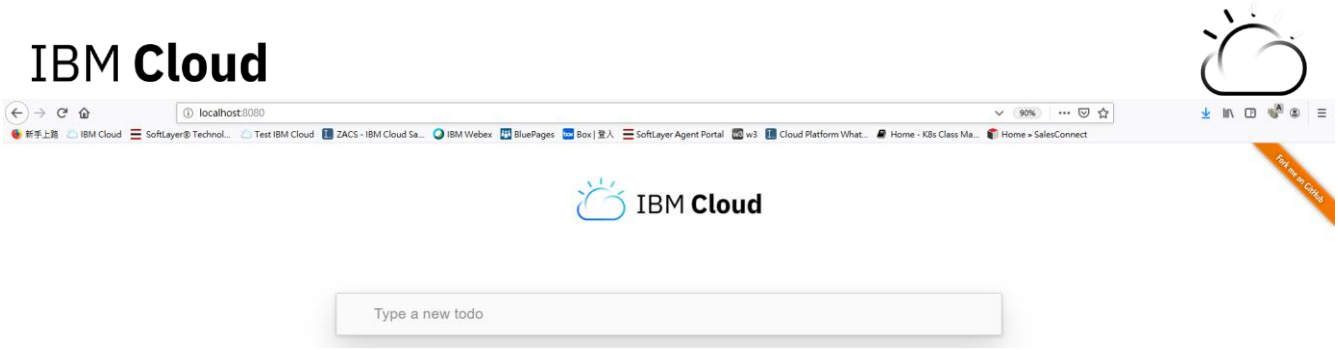
5. Use below command to start npm

npm start

```
C:\IKS\mytodo>npm start

> mytodo@2.0.0 start C:\IKS\mytodo
> node server.js

Cannot find module ./vcap-local.json
Cannot find credentials.env
Using In Memory
server starting on http://localhost:8080
```

x

6. Use browser to navigate http://localhost:8080
7. You can use Ctrl + C to stop npm process




```
server starting on http://localhost:8080
要終止批次工作嗎 (Y/N)? y

C:\IKS\mytodo>_
```

Build the docker image with your application code

1. Use below command to create IBM Cloud Container Registry namespace, ex: wk1201908000ns

ibmcloud cr namespace-add wk1201908000ns

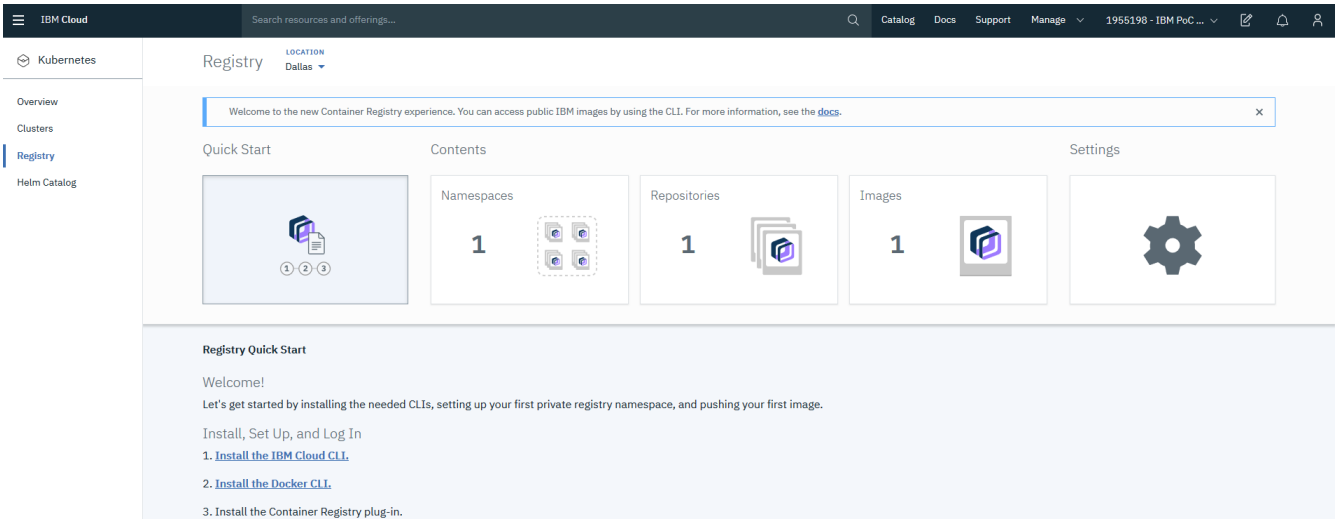
```
C:\IKS\mytodo>ibmcloud cr namespace-add wk1201908000ns

外掛程式 'container-registry 0.1.404' 現在可供使用（您有 0.1.395）。
使用 'ibmcloud plugin update container-registry' 以升級外掛程式。
使用 'ibmcloud config --check-version=false' 以停用更新檢查。

正在新增名稱空間 'wk1201908000ns'...

已順利新增名稱空間 'wk1201908000ns'
```

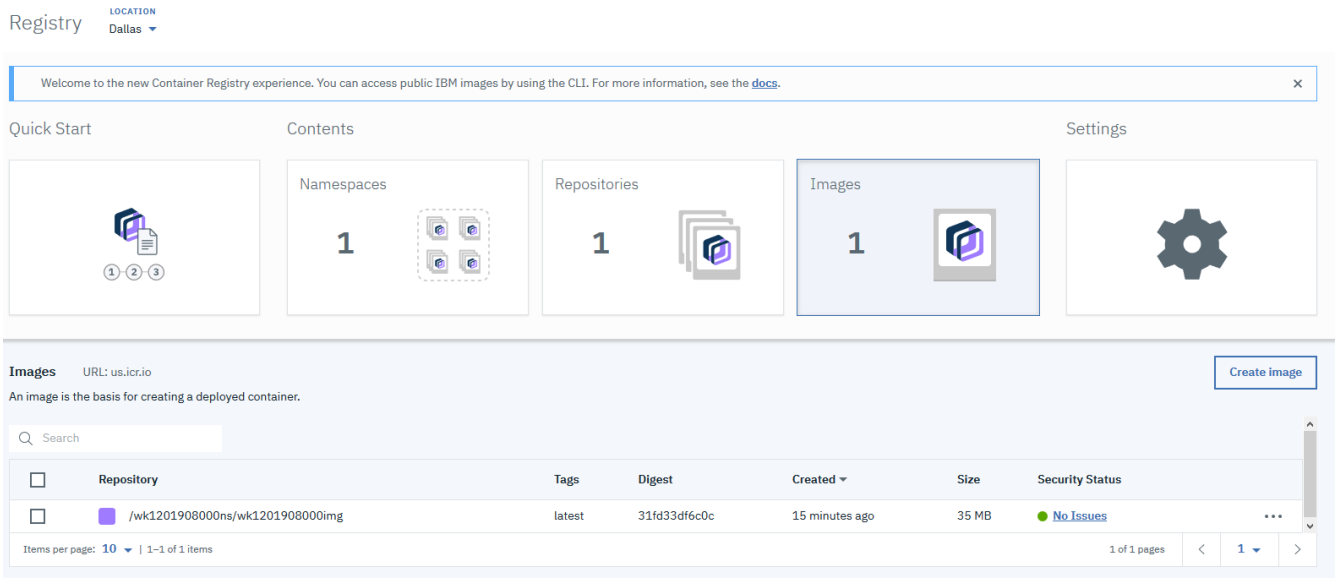
2. Use below command to create image, ex: wk1201908000img, with tag: latest in the IBM Cloud Container Registry namespace

ibmcloud cr build -t us.icr.io/wk1201908000ns/wk1201908000img:latest .

```
C:\IKS\mytodo>ibmcloud cr build -t us.icr.io/wk1201908000ns/wk1201908000img:latest .
Sending build context to Docker daemon  1.293MB
Step 1/8 : FROM node:11.13.0-alpine
11.13.0-alpine: Pulling from library/node

 [1B201b3a05: Pulling fs layer
 [1B04895ca9: Pulling fs layer
 [1BDigest: sha256:72aef1f7a5088ee3b162b1f9eafb48e8b76a9fc9df70169cb6cce75cff7fefce34MB/1.334MBB
Status: Downloaded newer image for node:11.13.0-alpine
 ---> f34bcc2815ed
Step 2/8 : RUN mkdir -p /usr/src/app
 ---> Running in 4ebd1d077bb8
Removing intermediate container 4ebd1d077bb8
 ---> 28bdd451e505
Step 3/8 : WORKDIR /usr/src/app
 ---> Running in a30ccd056d59
Removing intermediate container a30ccd056d59
 ---> 71bd185002d3
Step 4/8 : COPY package.json /usr/src/app/
 ---> 6d6dc8ce29ac
Step 5/8 : RUN npm install
 ---> Running in 2021cc7d949f
 [91mnpm notice created a lockfile as package-lock.json. You should commit this file.
 [0m [91mnpm WARN mytodo@2.0.0 No repository field.
 [0m [91m
 [0madded 161 packages from 171 contributors and audited 571 packages in 2.433s
found 0 vulnerabilities

Removing intermediate container 2021cc7d949f
 ---> 4d2a2624146d
Step 6/8 : COPY . /usr/src/app
 ---> e10f88f76913
Step 7/8 : EXPOSE 8080
 ---> Running in 77096d957946
Removing intermediate container 77096d957946
 ---> 5666f16d4d7f
Step 8/8 : CMD [ "npm", "start" ]
 ---> Running in 738cfda08684
Removing intermediate container 738cfda08684
 ---> 70b6d2847d61
Successfully built 70b6d2847d61
Successfully tagged us.icr.io/wk1201908000ns/wk1201908000img:latest
The push refers to repository [us.icr.io/wk1201908000ns/wk1201908000img]

 [1B822bd99c: Preparing
 [1Bbf26b704: Preparing
 [1B45abe542: Preparing
 [1Bb2b1604f: Preparing
 [1B064aa902: Preparing
 [1B49e341bb: Preparing
 [2Blatest: digest: sha256:31fd33df6c0c576ae374b232872f9b7f7c62b1805d75cf940da488a90fdacf30 size: 1786
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and direc
tories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permiss
ions for sensitive files and directories.

OK
```

3. You can see the IBM Cloud Container Registry namespace, Respository and Image on the IBM Cloud page



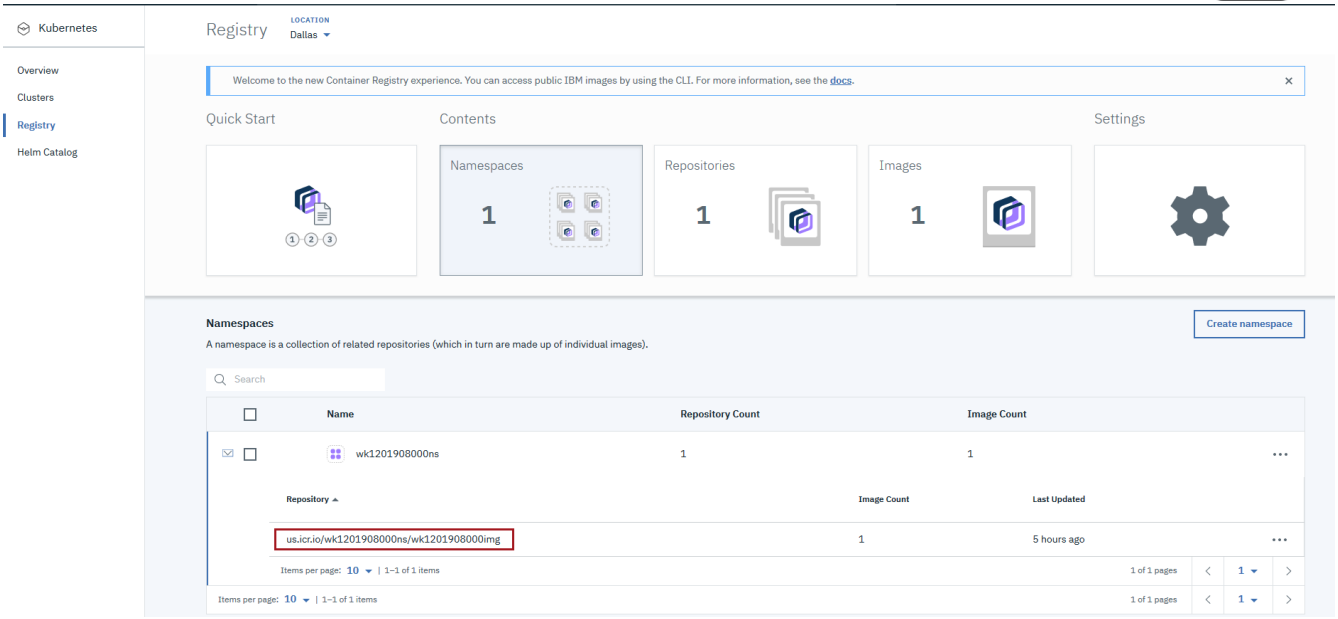


4. Use below command to list image in the IBM Cloud Container Registry namespace

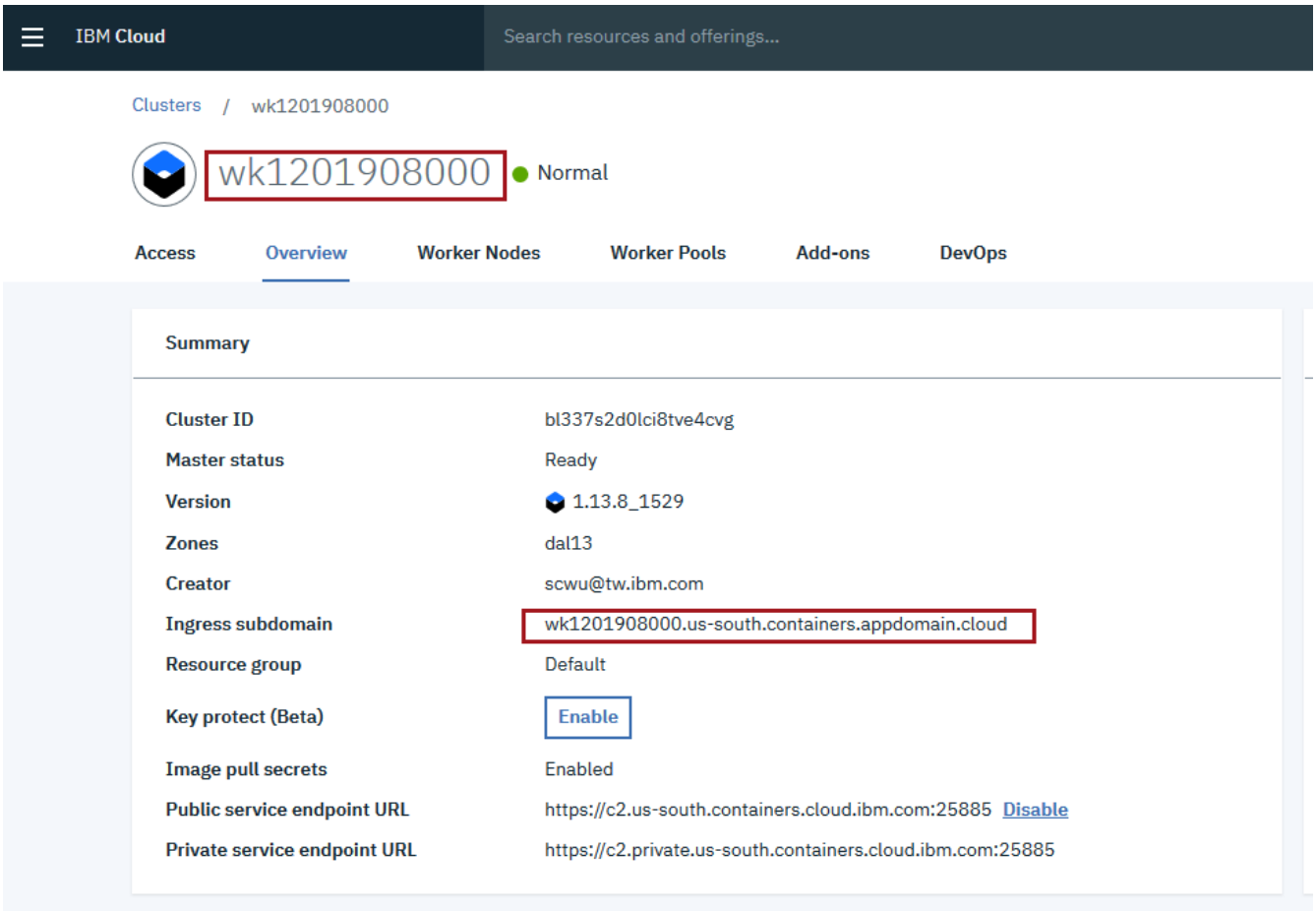
ibmcloud cr images

```
C:\IKS\mytodo>ibmcloud cr images
正在列出映像檔...

儲存庫                                        標籤      摘要          名稱空間          建立時間          大小    安全狀態

us.icr.io/wk1201908000ns/wk1201908000img      latest    31fd33df6c0c  wk1201908000ns    17 minutes ago    35 MB   沒有問題

OK
```

5. Use below command to change directory to kubernetes and use notepad to edit ingress-tls-deploy.yaml file
6. You should replace <registry-region>.icr.io to us.icr.io, <registry-namespace> to wk1201908000ns, todo-<lastname>:1.0 to wk1201908000img:latest
7. You can check the info. on the IBM Cloud page



8. You should replace <cluster-name>.<cloud-region>.containers.appdomain.cloud to wk1201908000.us-south.containers.appdomain.cloud, <cluster-name> to wk1201908000
9. You can check the info. on the IBM Cloud page
10. And you should remark envFrom section



cd kubernetes
notepad ingress-tls-deploy.yaml

ingress-tls-deploy.yaml - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
---
# Application to deploy
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mytodos
spec:
  replicas: 2 # tells deployment to run 2 pods
  selector:
    matchLabels:
      app: mytodos
  template:   # create pods using pod definition in this template
    metadata:
      labels:
        app: mytodos
        tier: frontend
    spec:
      containers:
      - name: mytodos
        image: us.icr.io/wk1201908000ns/wk1201908000img:latest
        imagePullPolicy: Always
        resources:
          requests:
            cpu: 250m     # 250 millicores = 1/4 core
            memory: 128Mi # 128 MB
          limits:
            cpu: 500m
            memory: 384Mi
        # envFrom:
        # - secretRef:
        #     name: database-credentials
        #     optional: true
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: mytodos-ingress
  annotations:
    # Force the use of https if the request is http
    ingress.bluemix.net/redirect-to-https: "True"
spec:
  tls:
  - hosts:
    - todo.wk1201908000.us-south.containers.appdomain.cloud
    secretName: wk1201908000
  rules:
  - host: todo.wk1201908000.us-south.containers.appdomain.cloud
    http:
      paths:
      - path: /
        backend:
          serviceName: mytodos
          servicePort: 8080
---
# Service to expose frontend
apiVersion: v1
kind: Service
```

Windows (CRLF) 第 45 列，第 29 行 100%

11. After modify the yaml file, you can use below command to deploy the app into your Kubernetes cluster
12. If you get the error msg., you should use curl command to download proper version kubectl.exe

kubectl apply -f ingress-tls-deploy.yaml

```
C:\IKS\mytodo\kubernetes>kubectl apply -f ingress-tls-deploy.yaml
error: SchemaError(io.k8s.apimachinery.pkg.apis.meta.v1.APIGroup): invalid object doesn't have additional properties
```

curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.13.8/bin/windows/amd64/kubectl.exe

```
C:\IKS\mytodo\kubernetes>curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.13.8/bin/windows/amd64
/kubectl.exe
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 37.8M  100 37.8M    0     0   945k      0  0:00:41  0:00:41 --:--:--  582k
```

13. Use below command to check kubectl.exe version. The client version and server version should be the same

kubectl version

```
C:\IKS>kubectl version
Client Version: version.Info{Major:"1", Minor:"13", GitVersion:"v1.13.8", GitCommit:"0c6d31a99f81476dfc9871ba3cf3f597be
c29b58", GitTreeState:"clean", BuildDate:"2019-07-08T08:46:01Z", GoVersion:"go1.11.5", Compiler:"gc", Platform:"windows
/amd64"}
Server Version: version.Info{Major:"1", Minor:"13", GitVersion:"v1.13.8+IKS", GitCommit:"fe3e332c2b0f47d4572433c3b0a168
7a27fb88c6", GitTreeState:"clean", BuildDate:"2019-07-11T13:45:03Z", GoVersion:"go1.11.5", Compiler:"gc", Platform:"lin
ux/amd64"}
```

14. After modify the yaml file, you can use below command to deploy the app into your Kubernetes cluster

kubectl apply -f ingress-tls-deploy.yaml

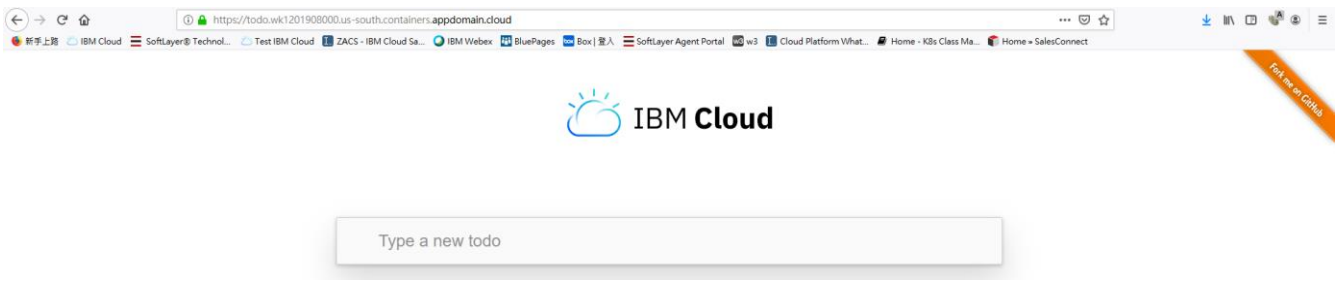
```
C:\IKS\mytodo\kubernetes>kubectl apply -f ingress-tls-deploy.yaml
deployment.apps/mytodos created
ingress.extensions/mytodos-ingress created
service/mytodos created
```

15. Use below command to check Kubernetes pods status

kubectl get pods

```
C:\IKS\mytodo\kubernetes>kubectl get pods
NAME                       READY   STATUS    RESTARTS   AGE
mytodos-5cf8d58f74-2f9pb   1/1     Running   0          41s
mytodos-5cf8d58f74-gctfh   1/1     Running   0          41s
```

16. Use browser to navigate https://todo.wk1201908000.us-south.containers.appdomain.cloud



Scale the application using the replica

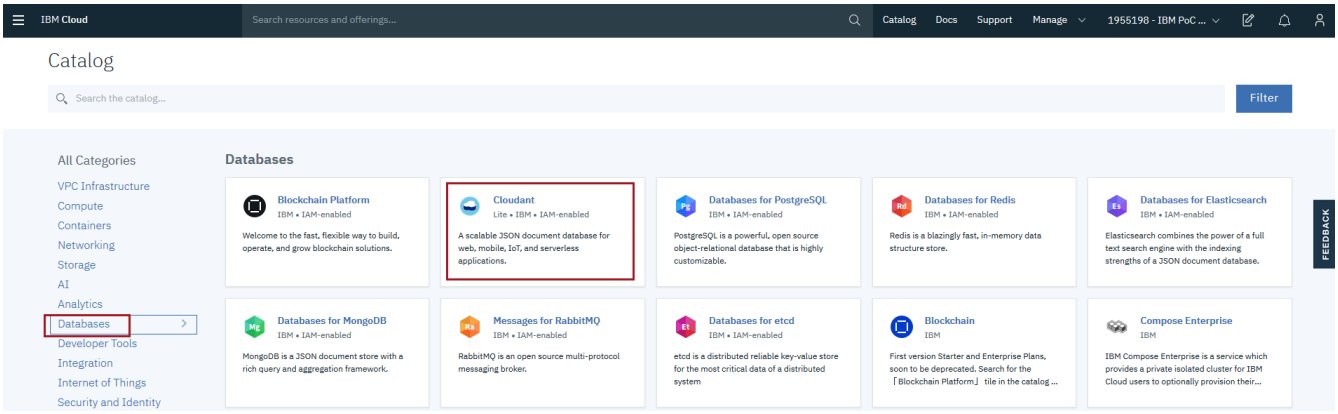
1. Use below command to scale up to 3 replicas and check Kubernetes pods status

kubectl scale --replicas=3 deployment/mytodos

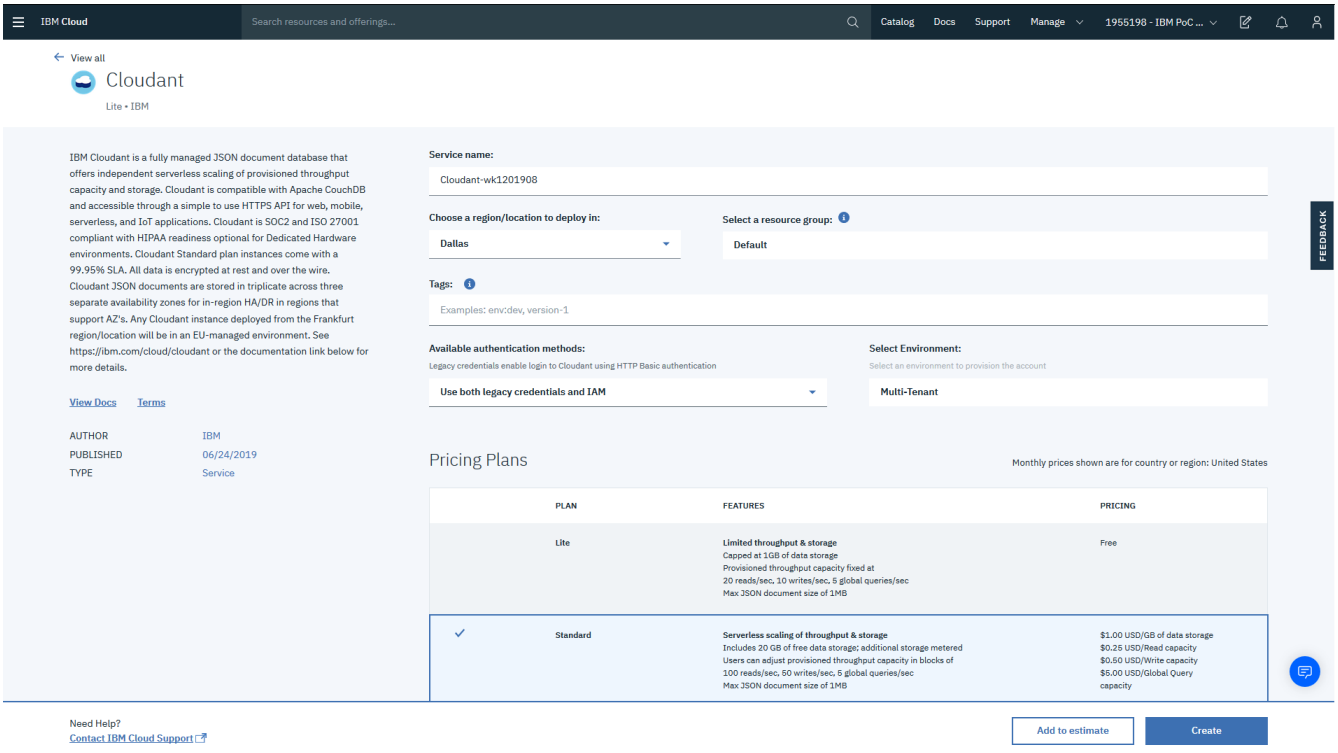
```
C:\IKS\mytodo\kubernetes> kubectl scale --replicas=3 deployment/mytodos
deployment.extensions/mytodos scaled
```

Use a Kubernetes Secret to connect to a managed Database service

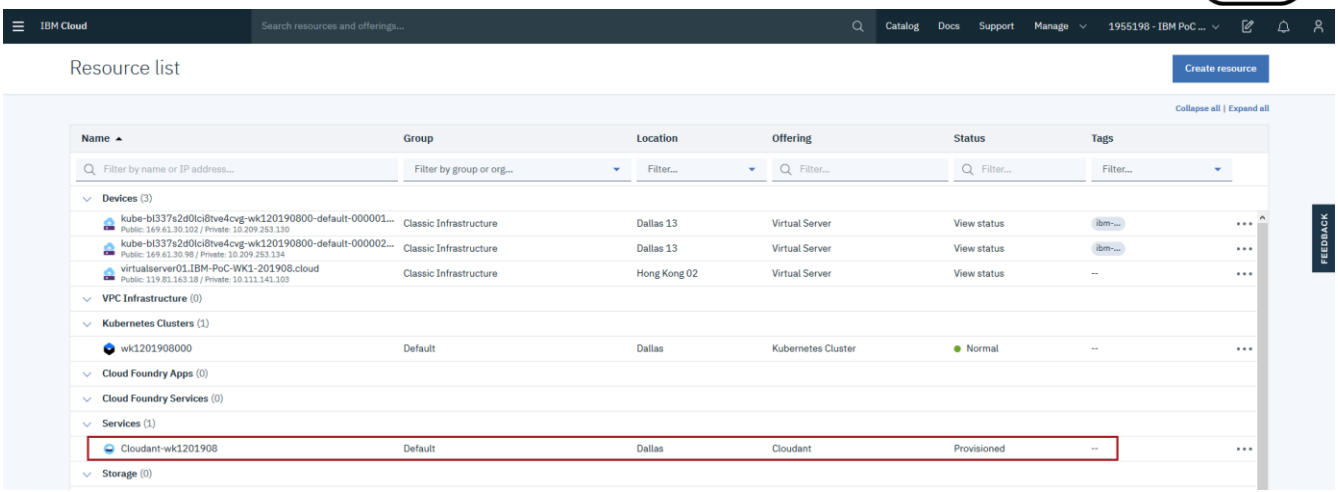
1. Click Cloudant with Database filter on the Catalog page

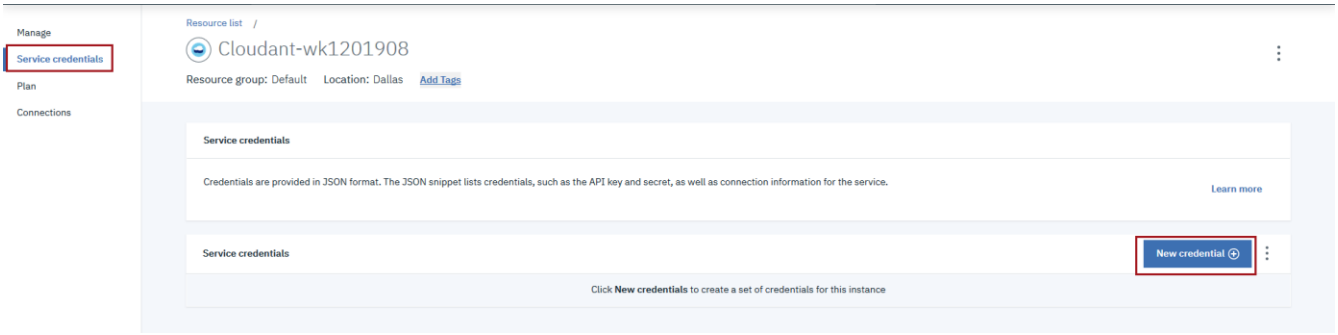


2. Create Cloudant service with name(ex: Cloudant-wk1201908000), Location: Dallas, Resource group: Default, Authentication methods: Use both legacy credentials and IAM, Pricing Plan: Standard

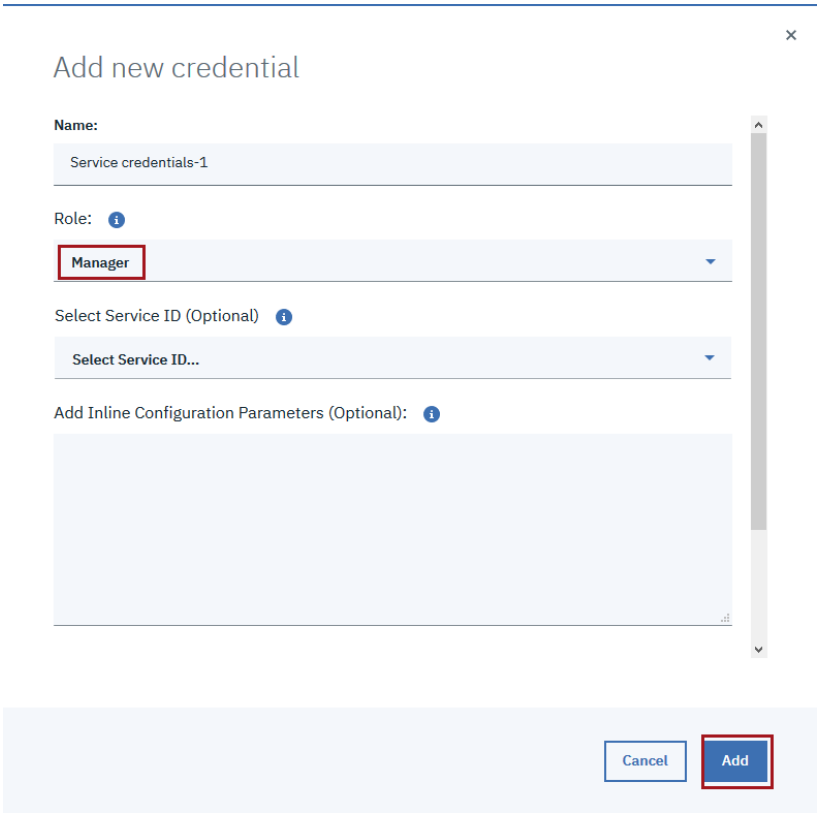


3. After the status of Cloudant service changed to Provisioned, click the item to see the detail
4. Switch to Service Credentials and add new credential

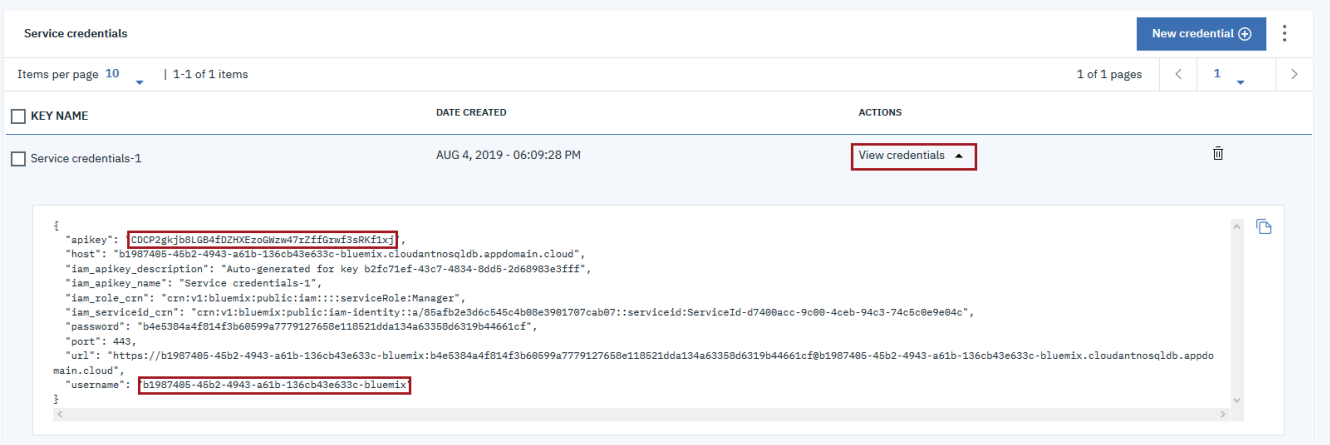




5. After The role should be Manager in the new credential window



6. After new credential created, you can view detail.
7. Get the apikey and username in the credential

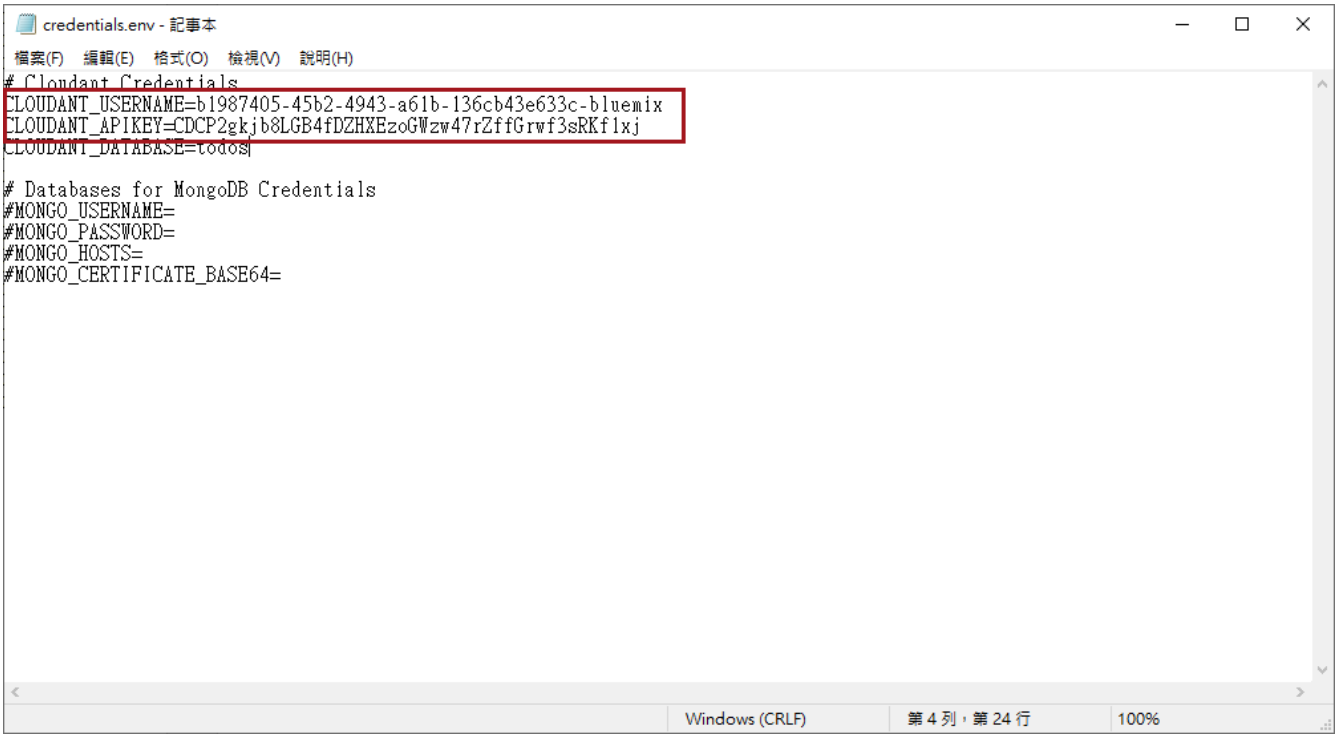


8. Change the directory to c:\iks\mytodo, then use below command to copy file and edit file
9. Input the apikey and username from credential in the credentials.env

cd c:\iks\mytodo

copy credentials.template.env credentials.env

notepad credentials.env



10. Use below command to get the Kubernetes namespaces list

kubectl get namespaces

```
C:\IKS\mytodo> kubectl get namespaces
NAME              STATUS    AGE
default           Active    8h
ibm-cert-store    Active    8h
ibm-system        Active    8h
kube-public       Active    8h
kube-system       Active    8h
```

11. Use below command to create secret(ex: wk1201908000db-secret) from credentials.env file

kubectl create secret generic wk1201908000db-secret --from-env-file=credentials.env

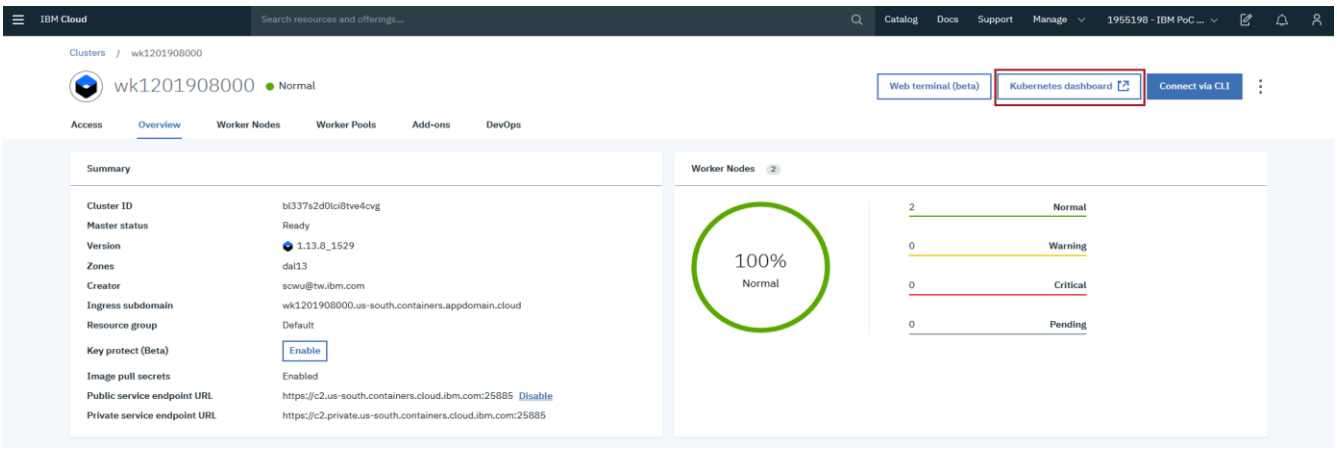
```
C:\IKS\mytodo>kubectl create secret generic wk1201908000db-secret --from-env-file=credentials.env
secret "wk1201908000db-secret" created
```

12. Use below command to check secret, you should see the secret you just created

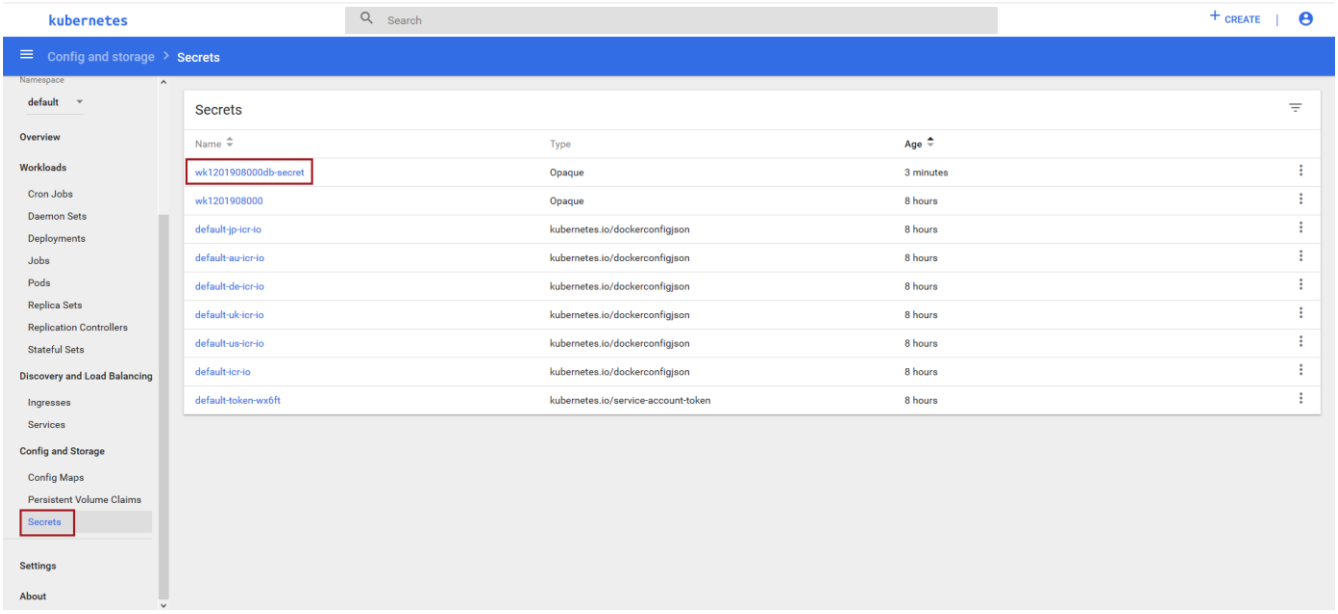
Kubectl get secrets

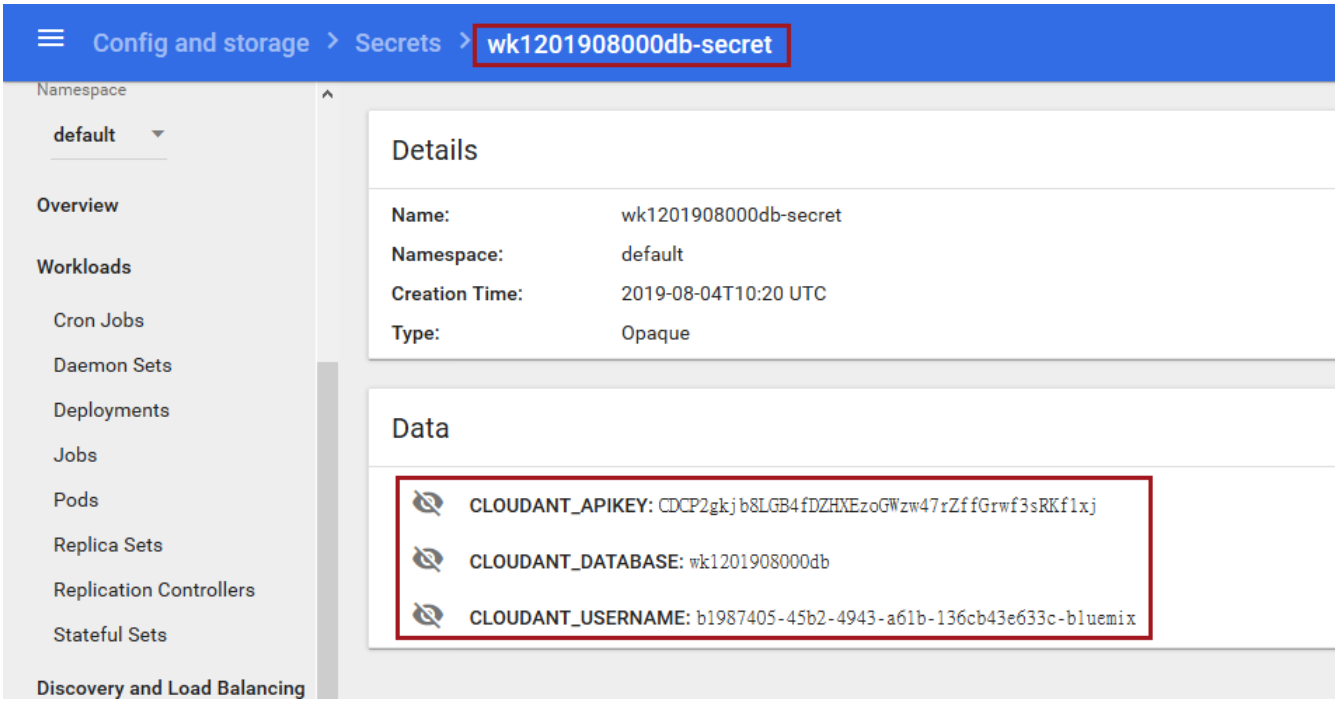
```
C:\IKS\mytodo>kubectl get secrets
NAME                  TYPE                                  DATA      AGE
default-au-icr-io     kubernetes.io/dockerconfigjson        1         8h
default-de-icr-io     kubernetes.io/dockerconfigjson        1         8h
default-icr-io        kubernetes.io/dockerconfigjson        1         8h
default-jp-icr-io     kubernetes.io/dockerconfigjson        1         8h
default-token-wx6ft   kubernetes.io/service-account-token   3         8h
default-uk-icr-io     kubernetes.io/dockerconfigjson        1         8h
default-us-icr-io     kubernetes.io/dockerconfigjson        1         8h
wk1201908000          Opaque                                2         8h
wk1201908000db-secret Opaque                                3         2m
```

13. You can check the secret detail from Kubernetes dashboard on the IBM Cloud page



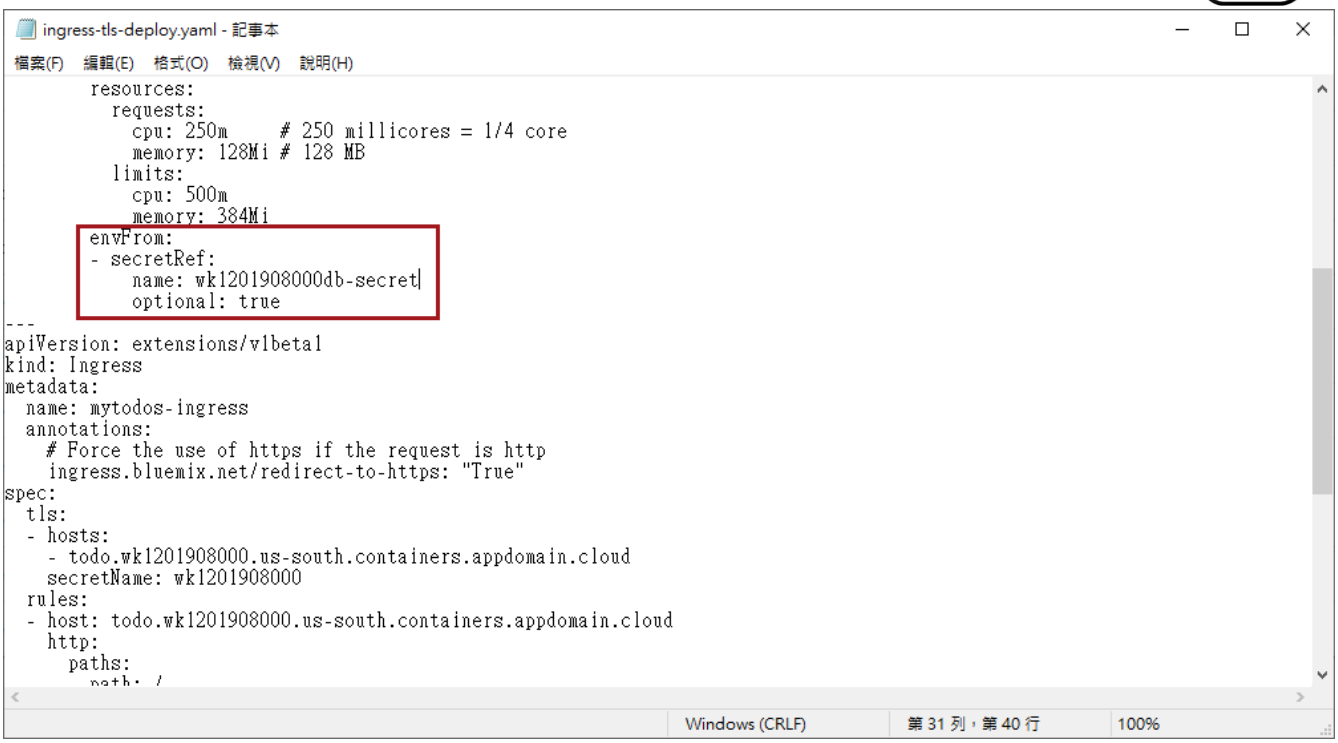
14. You switch to Secret page and click your secret name. You should see the content you input in the credential.env file





15. Change the directory to c:\iks\mytodo\kubernetes
16. Use notepad to edit ingress-tls-deploy.yaml file
17. Remove the remark from envFrom section

cd kubernetes



18. Use below command to apply new config to existed pods

kubectl apply -f ingress-tls-deploy.yaml

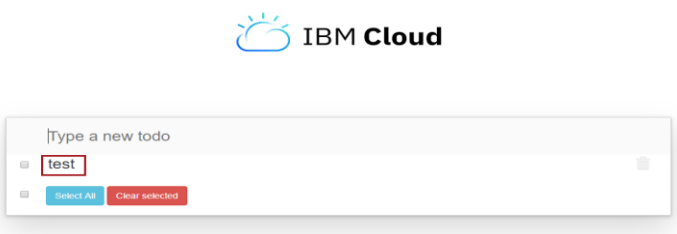
```
C:\IKS\mytodo\kubernetes> kubectl apply -f ingress-tls-deploy.yaml
deployment.apps/mytodos configured
ingress.extensions/mytodos-ingress unchanged
service/mytodos unchanged
```

19. Use below command to check Kubernetes pods status

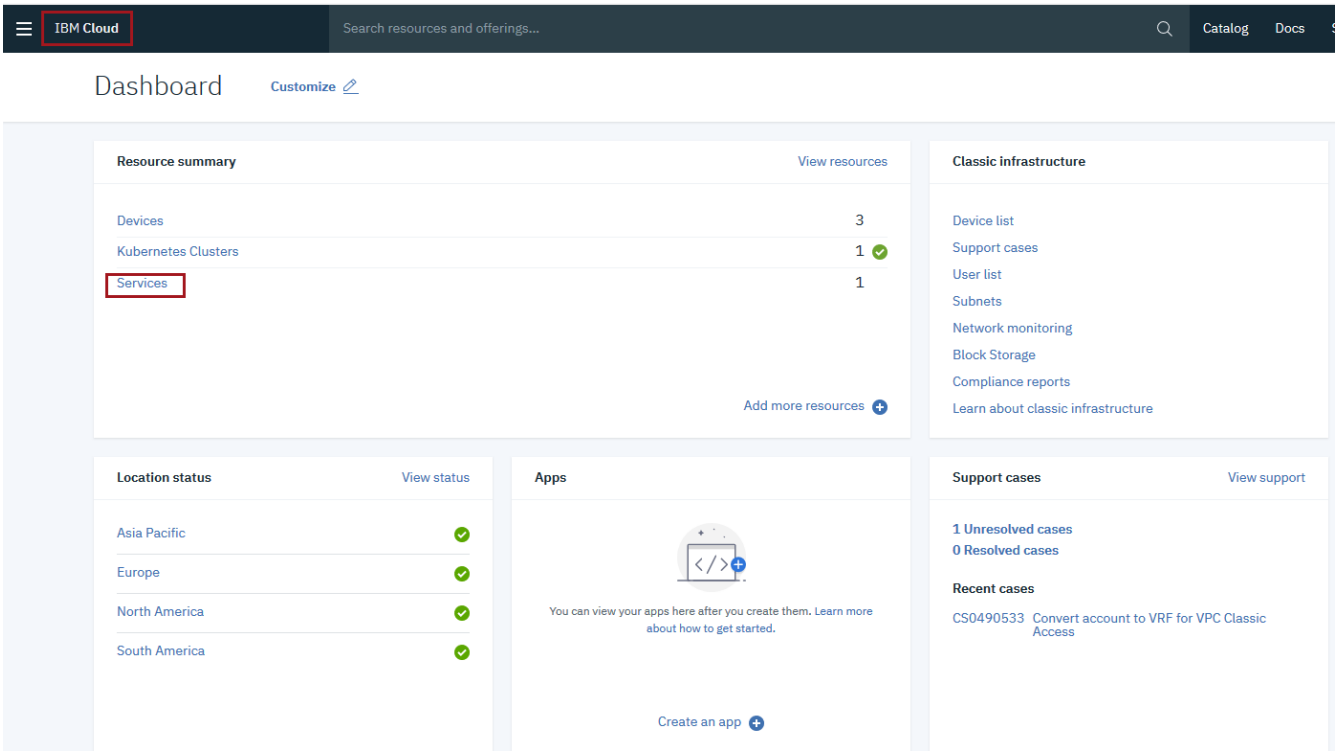
kubectl get pods

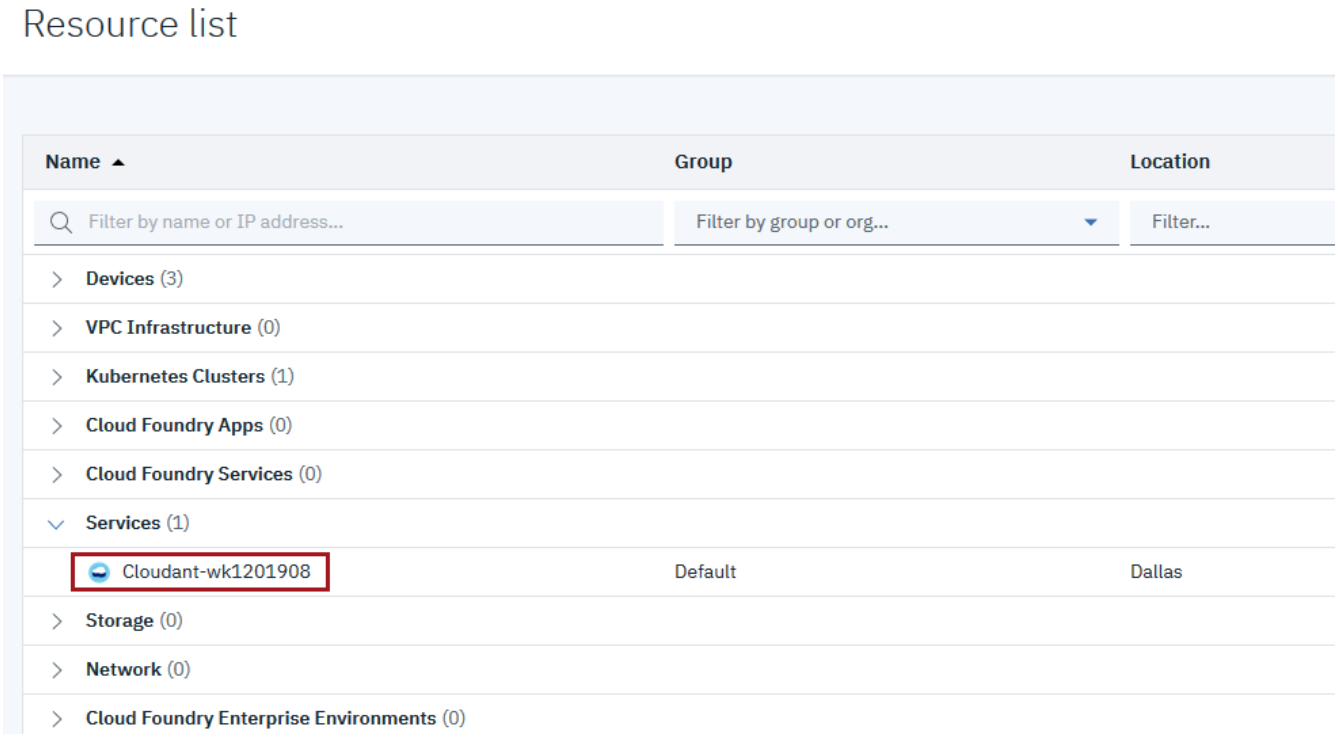
```
C:\IKS\mytodo\kubernetes>kubectl get pods
NAME                        READY   STATUS    RESTARTS   AGE
mytodos-6f8694b8bd-6hph5    1/1     Running   0          12s
mytodos-6f8694b8bd-b9556    1/1     Running   0          10s
```

20. Use browser to navigate https://todo.wk1201908000.us-south.containers.appdomain.cloud and input a test string.

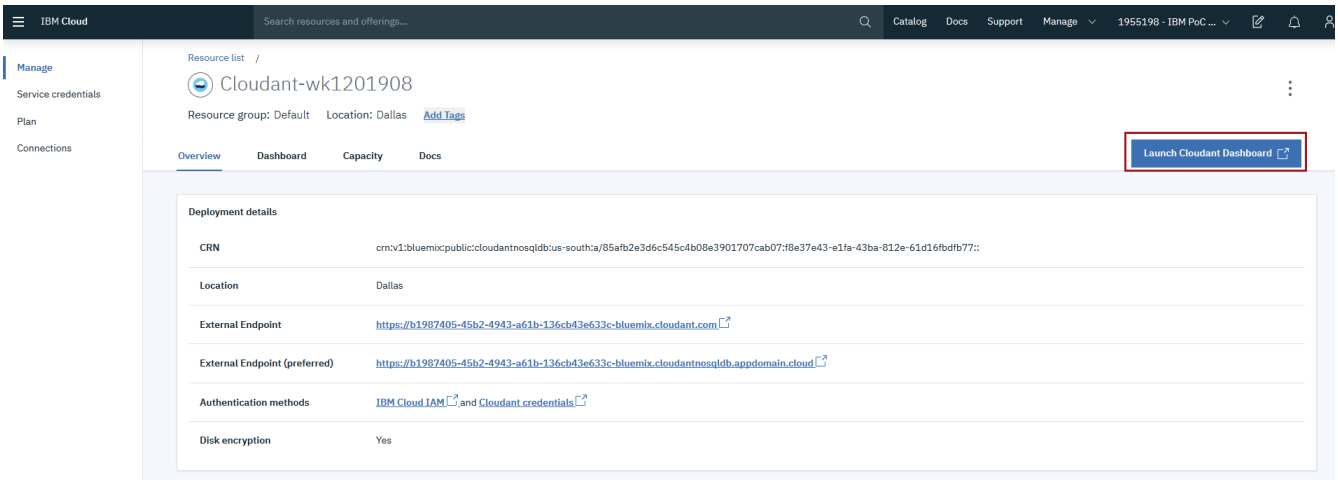


21. Go to IBM Cloud first page and click Cloudant service item to see detail

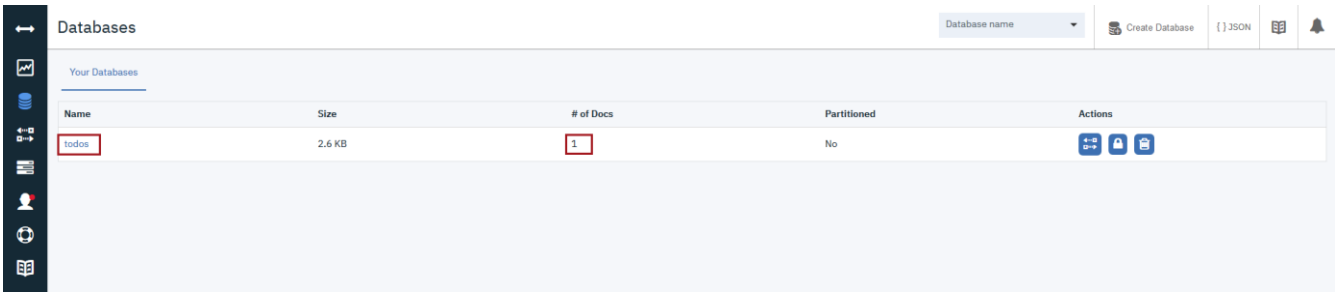




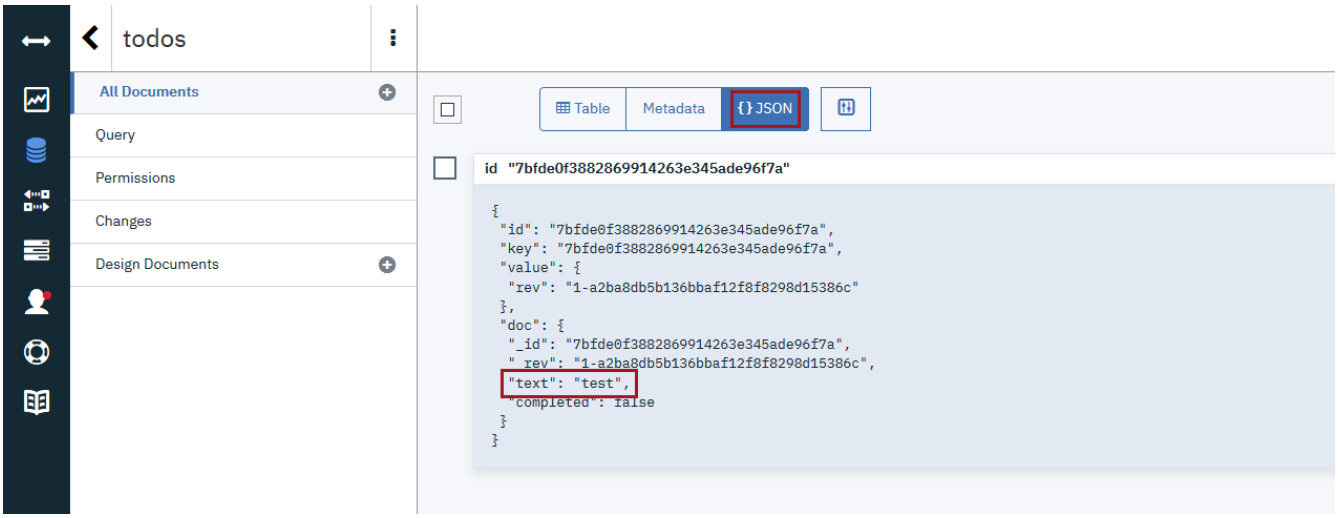
22. Launch Cloudant Dashboard



23. Switch to Databases page, you can see a new database: mytodo and one Docs



24. Click the database: mytodo and switch the format to {}JSON, you see the record you just created on the web page



Test Resiliency

1. Use below command to get Kubernetes nodes status

kubectl get nodes

```
C:\IKS\mytodo\kubernetes>kubectl get nodes
NAME             STATUS   ROLES    AGE   VERSION
10.209.253.130   Ready    <none>   10h   v1.13.8+IKS
10.209.253.134   Ready    <none>   10h   v1.13.8+IKS
```

2. Use below command to get Kubernetes pods detail status

kubectl get pods -o wide -w

```
C:\IKS\mytodo\kubernetes>kubectl get po -o wide -w
NAME                       READY   STATUS    RESTARTS   AGE     IP              NODE             NOMINATED NODE   READINESS GATES
mytodos-6f8694b8bd-bxlsk   1/1     Running   0          6m33s   172.30.218.15   10.209.253.134   <none>           <none>
mytodos-6f8694b8bd-ggcr2   1/1     Running   0          11m     172.30.218.14   10.209.253.134   <none>           <none>
```

3. Use below command to drain one node

kubectl drain 10.209.253.130 --ignore-daemonsets --delete-local-data

```
C:\IKS\mytodo\kubernetes>kubectl drain 10.209.253.130 --ignore-daemonsets --delete-local-data
node/10.209.253.130 cordoned
WARNING: Ignoring DaemonSet-managed pods: calico-node-ts688, ibm-keepalived-watcher-7v7mt, ibm-kube-fluentd-8rlxd
pod/ibm-cloud-provider-ip-169-61-1-54-67b5dcc796-n2jhg evicted
pod/public-crb1337s2d0lci8tve4cvg-alb1-5d88d685bc-dmg4g evicted
pod/mytodos-6f8694b8bd-hxc75 evicted
node/10.209.253.130 evicted
```

4. Use below command to get Kubernetes nodes status and see one node's status is changed

kubectl get nodes

```
C:\IKS\mytodo\kubernetes>kubectl get nodes
NAME             STATUS                     ROLES    AGE   VERSION
10.209.253.130   Ready,SchedulingDisabled   <none>   10h   v1.13.8+IKS
10.209.253.134   Ready                      <none>   10h   v1.13.8+IKS
```

5. Use below command to get Kubernetes pods detail status and see the pods still are running

kubectl get pods -o wide -w

```
C:\IKS\mytodo\kubernetes>kubectl get po -o wide -w
NAME                    READY   STATUS    RESTARTS   AGE     IP              NODE             NOMINATED NODE   READINESS GATES
mytodos-6f8694b8bd-bxlsk   1/1     Running   0          6m33s   172.30.218.15   10.209.253.134   <none>           <none>
mytodos-6f8694b8bd-ggcr2   1/1     Running   0          11m     172.30.218.14   10.209.253.134   <none>           <none>
```

6. Use below command to resume scheduling new pods onto the node

```
C:\IKS\mytodo\kubernetes>kubectl uncordon 10.209.253.130
node/10.209.253.130 uncordoned
```

7. Use below command to get Kubernetes nodes status

kubectl get nodes

```
C:\IKS\mytodo\kubernetes>kubectl get nodes
NAME             STATUS   ROLES    AGE   VERSION
10.209.253.130   Ready    <none>   11h   v1.13.8+IKS
10.209.253.134   Ready    <none>   11h   v1.13.8+IKS
```

Clean up the environment

1. Delete the deployment and the services

kubectl delete -f ingress-tls-deploy.yaml

Appendix

Installing the stand-alone IBM Cloud CLI

Installing with an installer

Reference URL: https://cloud.ibm.com/docs/cli?topic=cloud-cli-install-ibmcloud-cli

Use the following steps to install the latest stand-alone IBM Cloud CLI:

1. Use a browser to access the official ibm-cloud-cli-releases GitHub repository, and **select** the installer of your OS to begin the download. The following operating systems are supported: macOS X 64-bit, Windows™ 64-bit, Linux™ x86 64-bit, and Linux™ LE 64-bit (ppc64le).
2. Run the installer:
   - For Mac and Windows™, run the installer.
   - For Linux™, extract the package and run the install script.
3. Log in to IBM Cloud:

ibmcloud login

Now, you're ready to manage IBM Cloud resources. Enter ibmcloud help to view the command descriptions.

Installing from the shell

To install the latest CLI for your OS from the shell manually, use the following command for your OS:

- For **Mac**, copy and paste the following command to a terminal and run it:

curl -fsSL https://clis.cloud.ibm.com/install/osx | sh

- For **Linux™**, copy and paste the following command to a terminal and run it:

curl -fsSL https://clis.cloud.ibm.com/install/linux | sh

- For **Windows™**, copy and paste the following command to a Windows™ PowerShell terminal console and run it:

iex(New-Object Net.WebClient).DownloadString('https://clis.cloud.ibm.com/install/powershell')

Installing the IBM Cloud developer tools CLI plug-in manually

Reference URL: https://cloud.ibm.com/docs/cli?topic=cloud-cli-install-devtools-manually

You can manually install the IBM Cloud™ developer tools command line interface (CLI) plug-in if you prefer more granular control for installing the components. Otherwise, all prerequisites are automatically installed for most users by using the platform installers.

Before you begin

- Install the stand-alone IBM Cloud CLI to get support for installing command line plug-ins for IBM Cloud.
- Install the curl command for downloading packages through the command line.

Installing the IBM Cloud developer tools CLI plug-in

You can use the IBM Cloud developer tools CLI commands to create an application, manage, deploy, debug, and test it.

To install the IBM Cloud developer tools plug-in, run the following command:

ibmcloud plugin install dev

Installing Docker

For running and debugging apps locally, install Docker .

Installing the Kubernetes command line tool

To view a local version of the Kubernetes dashboard, and to deploy apps into your clusters,

install the Kubernetes command line tool for your platform:

- Mac:

curl --progress-bar -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/darwin/amd64/kubectl

- Linux™:

curl --progress-bar -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl

- Windows™:

curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.7.0/bin/windows/amd64/kubectl.exe

The prefix for running commands by using the Kubernetes command line tool is kubectl. For more information, see Setting up the CLI and API.

Installing IBM Cloud Object Storage CLI plug-in

The IBM Cloud Object Storage plug-in extends the IBM Cloud command line interface (CLI) with an API wrapper for working with Object Storage resources.

- To install the IBM Cloud Object Storage plug-in, run the following command:

ibmcloud plugin install cloud-object-storage

For more information, see the IBM Cloud Object Storage command reference.

Installing IBM Cloud Container Registry CLI plug-in

You can use the container-registry CLI plug-in to set up your own image namespace in an IBM-hosted, and managed, private registry. Where you can store and share Docker images with all users in your IBM Cloud account.

- To install the IBM® Cloud Container Registry plug-in, run the following command:

ibmcloud plugin install container-registry

For more information, see the IBM® Cloud Container Registry command reference.

Installing IBM Cloud Kubernetes Service CLI plug-in

To create and manage Kubernetes clusters in IBM® Cloud Kubernetes Service:

- To install the IBM Cloud Container Registry plug-in, run the following command:

ibmcloud plugin install container-service

For more information, see the IBM Cloud Container Registry command reference.

Installing Helm

Install Helm, which is a Kubernetes-based package manager.

- Mac and Linux™ users, run the following commands:

```
export DESIRED_VERSION=v2.7.2
curl -sL https://raw.githubusercontent.com/kubernetes/helm/master/scripts/get | bash
```

- Windows™ users can download and install the Helm binary .

Installing the Cloud Functions CLI plug-in

You can use the IBM® Cloud Functions CLI plug-in to manage your code snippets in actions, bundle actions into packages, and create triggers and rules to enable your actions to respond to events.

To install the Cloud Functions CLI plug-in, run the following command:

ibmcloud plugin install cloud-functions

For more information, see Installing the Cloud Functions CLI plug-in.

Uninstalling the stand-alone IBM Cloud CLI

Reference URL: https://cloud.ibm.com/docs/cli?topic=cloud-cli-uninstall-ibmcloud-cli

Use the following steps to uninstall the stand-alone IBM Cloud CLI on specific platforms.

Uninstalling on Windows

1. Click the **Start** button, and then select **Control Panel**.
2. In the pop-up window, click **Uninstall a program**.
3. In the pop-up application list, locate **IBM Cloud Command Line Interface**.
4. Right click **IBM Cloud Command Line Interface**, and select **Uninstall**.
5. The uninstaller is started. Follow the instructions to finish the uninstallation.

Uninstalling on Linux and macOS

The uninstallation steps are different depending on the version of the CLI that is installed.

To determine your IBM Cloud CLI version, run:

```
ibmcloud -v
```

To uninstall versions earlier than 0.9.0, run the following commands:

```
rm -rf /usr/local/ibmcloud
rm -f /usr/local/bin/ibmcloud
rm -f /usr/local/bin/bluemix
rm -f /usr/local/bin/bx
rm -f /usr/local/bin/ibmcloud-analytics
```

Clean up the autocompletion scripts, if you configured them. For more details, see Enabling shell autocompletion for IBM Cloud CLI (Linux and Mac only).

To uninstall versions 0.9.0 and later, run the following command:

```
/usr/local/ibmcloud/uninstall
```

Clean up any custom autocompletion scripts. For more details, see Enabling shell autocompletion for IBM Cloud CLI (Linux and Mac only).