IBM Cloud

IBM Cloud 用戶實作研習營

IBM Cloud Kubernetes Service (IKS) 使用教學(二)

2019/10/16

IBM Cloud

Creating a classic cluster in your Virtual Private Cloud (VPC)

With the **IBM Cloud™ Kubernetes Service clusters in VPC on Classic**, you can create your cluster on classic infrastructure in the next generation of the IBM Cloud platform, in your Virtual Private Cloud. VPC gives you the security of a private cloud environment with the dynamic scalability of a public cloud. VPC uses the next version of IBM Cloud Kubernetes Service infrastructure providers, with a select group of v2 API, CLI, and console functionality. You can create only standard clusters for VPC on Classic.

Objectives

In the tutorial lessons, you create an IBM Cloud Kubernetes Service cluster in a Virtual Private Cloud (VPC). Then, you deploy an app and expose the app publicly through a load balancer.

Prerequisites

Ensure that you have the following IBM Cloud IAM access policies.

- VPC on Classic clusters: **Administrator** platform role for VPC Infrastructure.
- **Administrator** platform role for IBM Cloud Kubernetes Service.
- **Writer** or **Manager** service role for IBM Cloud Kubernetes Service.
- **Administrator** platform role for Container Registry.

Install the command-line tools.

- Install the IBM Cloud CLI (ibmcloud), Kubernetes Serviceplug-in (ibmcloud ks), and IBM Cloud Container Registry plug-in (ibmcloud cr).
- Update your IBM Cloud Kubernetes Service plug-in to the latest version.
  Command:
  ibmcloud plugin update container-service

```
C:\workshop201910>ibmcloud plugin update container-service
外掛程式 'container-service/kubernetes-service 0.4.31' 已安裝。
正在從儲存庫 'IBM Cloud' 中檢查外掛程式 'container-service/kubernetes-service' 的升級...
沒有更新項目可用。
```

- To work with VPC, install the infrastructure-service plug-in. The prefix for running commands is ibmcloud is.
  Command:
  ibmcloud plugin install infrastructure-service

```
C:\workshop201910>ibmcloud plugin install infrastructure-service
正在從儲存庫 'IBM Cloud' 中查閱 'infrastructure-service'...
在儲存庫 'IBM Cloud' 中找到外掛程式 'vpc-infrastructure/infrastructure-service 0.5.3'
外掛程式 'vpc-infrastructure/infrastructure-service 0.5.3' 已安裝。您要將它重新安裝嗎？ [y/N] > n
外掛程式安裝已取消。
```

- Make sure that the kubectl version matches the Kubernetes version of your VPC cluster. This tutorial creates a cluster that runs version 1.15.

Lesson 1: Creating a cluster in VPC

Create an IBM Cloud Kubernetes Service cluster in your IBM Cloud Virtual Private Cloud (VPC) environment.

- Log in to the IBM Cloud region where you want to create your VPC environment. The VPC must be set up in the same multizone metro location where you want to create your cluster. In this tutorial you create a VPC in us-south. For other supported regions, see Multizone metros for VPC clusters. If you have a federated ID, include the --sso flag.
  Command:
  ibmcloud login -r <region> [--sso]
  <region>: jp-tok, us-south…

```
C:\workshop201910>ibmcloud login -r jp-tok
API 端點：https://cloud.ibm.com

Email> wujack0897@hotmail.com

Password>
正在鑑別...
確定

選取帳戶：
1. Jack Wu's Account (90b7dd2fc9a57526fe8eeafc580ea1dc) <-> 1613085
2. IBM PoC - SAP HANA 2018 (03deccc2e7de421fbdef7fe2b69375f3) <-> 1793315
3. IBM PoC - Parglory_SAP_POC_20181214 (1d727da8f23f468fb3541ab7104404da) <-> 1809847
4. Owen Chen's Account (fa23dcfce42e79b57526566fcac2cd55) <-> 1503677
5. IBM PoC - MyServer_POC_201901-201903 (00f3c65730f1409cbce0dea1a6a1a692) <-> 1821103
6. IBM PoC - ASTRO_POC_201902-201903 (6fd5533d53e44f91835fcdefc9bdacc5) <-> 1847747
7. Hsin Chong Machinery Works Co. Ltd. (ca2315e8d51a4e80ba9c5caecc97b74d) <-> 372964
8. IBM PoC - SYSAGE IaaS & PaaS Solution POC 2018 (82b67e5dbcee4defacb25b18faf674c9) <-> 1798303
請輸入數字> 8
已設定帳戶 IBM PoC - SYSAGE IaaS & PaaS Solution POC 2018 (82b67e5dbcee4defacb25b18faf674c9) <-> 1798303 的目標

已設定地區 jp-tok 的目標


API 端點：      https://cloud.ibm.com
地區：          jp-tok
使用者：        wujack0897@hotmail.com
帳戶：          IBM PoC - SYSAGE IaaS & PaaS Solution POC 2018 (82b67e5dbcee4defacb25b18faf674c9) <-> 1798303
資源群組：      未設定資源群組的目標，請使用 'ibmcloud target -g RESOURCE_GROUP'
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'。
```

  Command:
  ibmcloud target -g <resource group>
  <resource group>: Workshop

```
C:\workshop201910>ibmcloud target -g Default
已設定資源群組 Default 的目標


API 端點：      https://cloud.ibm.com
地區：          jp-tok
使用者：        wujack0897@hotmail.com
帳戶：          IBM PoC - SYSAGE IaaS & PaaS Solution POC 2018 (82b67e5dbcee4defacb25b18faf674c9) <-> 1798303
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'。
```

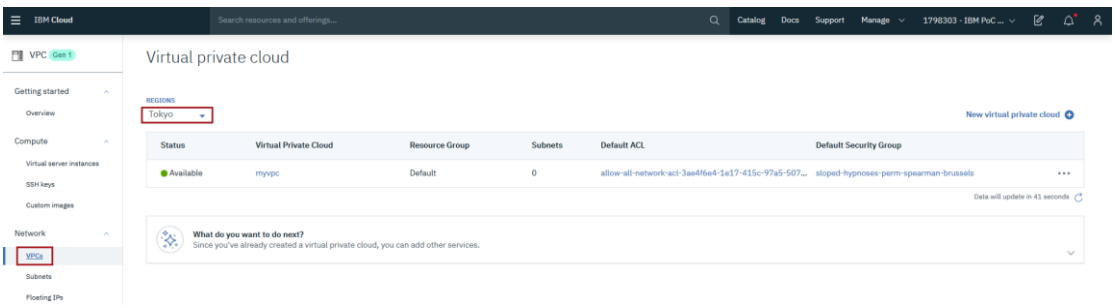
- Create a VPC for your cluster. For more information, see the docs for creating a VPC in the console or CLI.
  - Target the VPC on Classic infrastructure generation.
    Command:
    ibmcloud is target --gen 1

```
C:\workshop201910>ibmcloud is target --gen 1
OK
目標世代：1
```

  - Create a VPC that is called myvpc and note the **ID** in the output. VPCs provide an isolated environment for your workloads to run within the public cloud. You can use the same VPC for multiple clusters, such as if you plan to have different clusters host separate microservices that need to communicate with each other. If you want to separate your clusters, such as for different departments, you can create a VPC for each cluster.
    Command:
    ibmcloud is vpc-create <vpc name>
    <vpc name>: myvpc_”yourID”

```
C:\workshop201910>ibmcloud is vpc-create myvpc
正在以使用者 wujack0897@hotmail.com 身分，在帳戶 IBM PoC - SYSAGE IaaS & PaaS Solution POC 2018 下，在資源群組 Default 中建立 VPC myvpc...

ID                3ae4f6e4-1e17-415c-97a5-507ccbf022d1
名稱              myvpc
標準存取          no
預設網路 ACL      allow-all-network-acl-3ae4f6e4-1e17-415c-97a5-507ccbf022d1(dde68a37-3dae-4175-b763-fd93f6c3a38e)
預設安全群組      sloped-hypnoses-perm-spearman-brussels(2d364f0a-a870-42c3-a554-000002065182)
資源群組          7d50bf2967a74fa49f2943cf964a55d4
已建立            2019-10-12T17:43:39+08:00
狀態              available
```

You also can check on the IBM Cloud portal page.



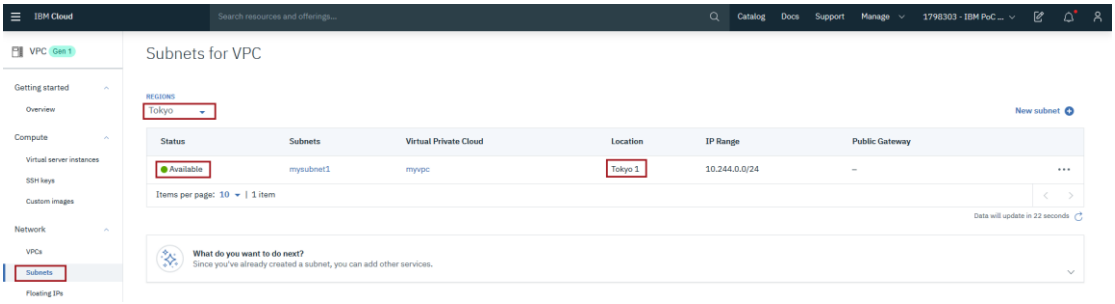
- Verify Kubernetes service can access VPCs.
  Command:
  ibmcloud ks vpcs

```
C:\workshop201910>ibmcloud ks vpcs
If you have clusters that run Kubernetes versions 1.10, 1.11 or 1.12, update them now to continue receiving important se
curity updates and support. Kubernetes version 1.12 is deprecated and will be unsupported 3 November 2019. Versions 1.11
 and earlier are already unsupported. For more information and update actions, see <https://ibm.biz/iks-versions>

Support is ending for the legacy IBM Cloud Log Analysis and IBM Cloud Monitoring services. Logging details: <https://ibm
.biz/BdzuVV>. Monitoring details: <https://ibm.biz/BdzuVJ>. To continue to forward cluster logs and metrics, set up IBM
Log Analysis with LogDNA and IBM Cloud Monitoring Service with Sysdig. Syslog forwarding configurations remain unaffecte
d. More info: <https://ibm.biz/Bdzbi3>.

正在以使用者 wujack0897@hotmail.com 身分取得帳戶 TW Marketing Events 下的 VPC...
OK
名稱    ID
myvpc   a06150ba-5748-43d6-8d69-57386046dcfe
```

- Create a subnet for your VPC, and note its ID.
  Command:
  ibmcloud is subnet-create <subnet name> <vpc_ID> --zone <zone> --ipv4-address-count 256
  <subnet name>: mysubnet_”yourID”
  <vpc_ID>: you can get the info. from previous screen or portal page
  <zone>: jp-tok-1, jp-tok-2, jp-tok-3, us-south-1…

```
C:\workshop201910>ibmcloud is subnet-create mysubnet1 3ae4f6e4-1e17-415c-97a5-507ccbf022d1 --zone jp-tok-1 --ipv4-address-count 256
正在以使用者 wujack0897@hotmail.com 身分，在帳戶 IBM PoC - SYSAGE IaaS & PaaS Solution POC 2018 下，建立子網路 mysubnet1...

ID                61d3d204-5ee4-4e4a-b49b-88365b720c89
名稱              mysubnet1
IPv4 CIDR         10.244.0.0/24
可用的位址        251
位址總計          256
ACL               allow-all-network-acl-3ae4f6e4-1e17-415c-97a5-507ccbf022d1(dde68a37-3dae-4175-b763-fd93f6c3a38e)
Public Gateway    -
已建立            2019-10-12T17:49:35+08:00
狀態              pending
區域              jp-tok-1
VPC               myvpc(3ae4f6e4-1e17-415c-97a5-507ccbf022d1)
```

You also can check on the IBM Cloud portal page.



- Create a cluster in your VPC in the same zone as the subnet. By default, your cluster is created with a public and a private service endpoint. You can use the public service endpoint to access the Kubernetes master, such as to run kubectl commands, from your local machine. Your worker nodes can communicate with the master on the private service endpoint. For more information about the command options, see the cluster create vpc-classic CLI reference docs.
  Command:

ibmcloud ks cluster create vpc-classic --name <cluster_name> --zone <zone> --flavor b2.4x16 --workers 1 --vpc-id <vpc_ID> --subnet-id <vpc_subnet_ID>

<cluster_name>: myvpc_cluster_"yourID"

<zone>: should be the same as your prior command used

<vpc_ID>: you can get the info. from previous screen or portal page

<vpc_subnet_ID>: you can get the info. from previous screen or portal page

```
C:\workshop201910>ibmcloud ks cluster create vpc-classic --name myvpc-cluster --zone jp-tok-1 --flavor b2.4x16 --workers 1 --vpc-id 6c25bda6-1827-48
7e-965d-8f3f382da28d --subnet-id c87939be-ee99-409f-85da-263a5177e92f
正在建立叢集...
OK
```

- Check the state of your cluster. The cluster might take a few minutes to provision.
  - Verify that the cluster State is normal.

    Command:

    ibmcloud ks cluster ls --provider vpc-classic

```
C:\workshop201910>ibmcloud ks cluster ls --provider vpc-classic
OK
名稱            ID                   狀態     建立時間      工作程式   位置    版本          資源群組名稱
myvpc-cluster   bmgst39t0u1k8pohqhog   normal   3 hours ago   1          Tokyo   1.15.4_1518   Workshop
```

  - Download the Kubernetes configuration files.

    Command:

    ibmcloud ks cluster config --cluster <cluster_name>

    <cluster_name>: should be the same as your prior command created

```
C:\workshop201910>ibmcloud ks cluster config --cluster myvpc-cluster
警告： This command is using deprecated behavior and will soon be unsupported. Set the '{{.BetaVar}}' environment variable to use the new behavior.
In {{.Shell}}, run '{{.Command}}'.
Note: Changing the beta version can include other breaking changes. 如需相關資訊，請參閱 http://ibm.biz/iks-cli-v1

OK
已順利下載 myvpc-cluster 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\myvpc-cluster\kube-config-tok02-myvpc-cluster.yml"

Command Prompt
SET KUBECONFIG=C:\Users\JACKWU\.bluemix\plugins\container-service\clusters\myvpc-cluster\kube-config-tok02-myvpc-cluster.yml
```

    When the download of the configuration files is finished, a command is displayed that you can use to set the path to the local Kubernetes configuration file as an environment variable.

  - Copy and paste the command that is displayed in your terminal to set the KUBECONFIG environment variable.

    Command:

    SET KUBECONFIG=C:\Users\<user_name>\.bluemix\plugins\kubernetes-service\clusters\<cluster_name>\kube-config-<datacenter>-<cluster_name>.yml

    You can directly copy and paste from screen, then enter.

```
C:\workshop201910>SET KUBECONFIG=C:\Users\JACKWU\.bluemix\plugins\container-service\clusters\myvpc-cluster\kube-config-tok02-myvpc-cluster.yml
```

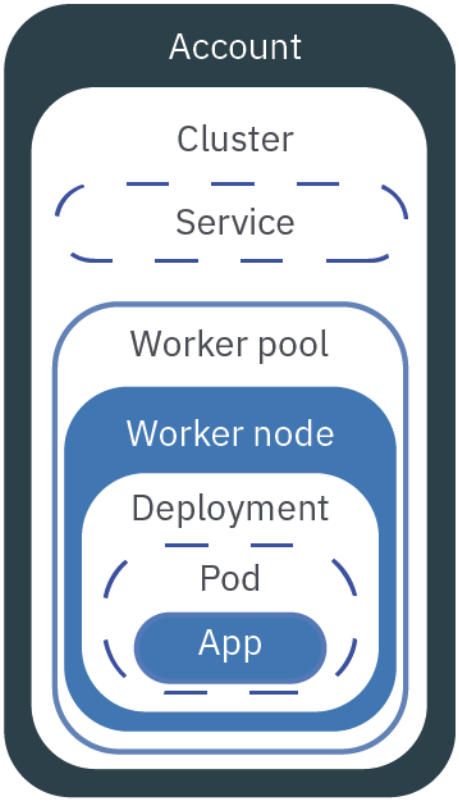
- Verify that the kubectl commands run properly with your cluster by checking the Kubernetes CLI server version.
  Command:
  kubectl version --short

  Example output:
  Client Version: v1.14.7
  Server Version: v1.14.7+IKS

```
C:\workshop201910>kubectl version  --short
Client Version: v1.14.6
Server Version: v1.15.4+IKS
```

Lesson 2: Deploying a privately available app

Create a Kubernetes deployment to deploy a single app instance as a pod to your worker node in your VPC cluster.
The components that you deploy by completing this lesson are shown in the following diagram.



To deploy the app:

- Clone the source code for the Hello world app to your user home directory. The repository contains different versions of a similar app in folders that each start with Lab. Each version contains the following files:
  - Dockerfile: The build definitions for the image.
  - app.js: The Hello world app.
  - package.json: Metadata about the app.

  Command:
  git clone https://github.com/IBM/container-service-getting-started-wt.git

```
C:\workshop201910>git clone https://github.com/IBM/container-service-getting-started-wt.git
Cloning into 'container-service-getting-started-wt'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 994 (delta 3), reused 8 (delta 3), pack-reused 976 eceiving objects:  99% (985/994), 2.96 MiB | 743.00 KiB/s
Receiving objects: 100% (994/994), 2.99 MiB | 749.00 KiB/s, done.
Resolving deltas: 100% (499/499), done.
```

- Navigate to the Lab 1 directory.
  Command:
  cd “container-service-getting-started-wt\Lab 1”

```
C:\workshop201910>cd "container-service-getting-started-wt\Lab 1"

C:\workshop201910\container-service-getting-started-wt\Lab 1>
```

- Log in to the IBM Cloud Container Registry CLI and note the registry region that you are in, such as jp.icr.io.
  Command:
  ibmcloud cr login

  Example output:
  Logged in to 'jp.icr.io'.

```
C:\workshop201910\container-service-getting-started-wt\Lab 1>ibmcloud cr login
正在登入 'jp.icr.io'...
已登入 'jp.icr.io'。

IBM Cloud Container Registry is adopting new icr.io domain names to align with the rebranding of IBM Cloud for a better user experience. The existin
g bluemix.net domain names are deprecated, but you can continue to use them for the time being, as an unsupported date will be announced later. For
more information about registry domain names, see https://cloud.ibm.com/docs/services/Registry?topic=registry-registry_overview#registry_regions_loc
al

OK
```

- Use an existing registry namespace or create one, such as vpcns.
  Command:
  ibmcloud cr namespace-list

```
C:\workshop201910\container-service-getting-started-wt\Lab 1>ibmcloud cr namespace-list
正在登錄 'jp.icr.io' 中列出帳戶 'IBM PoC - SYSAGE IaaS & PaaS Solution POC 2018' 的名稱空間...

這個帳戶沒有任何名稱空間，或者您未獲授權在此區域中檢視這個帳戶的任何名稱空間。
請建立一個名稱空間，或者請確定這個帳戶的 IAM 存取原則授與您對至少一個名稱空間的「讀者」或「管理員」存取權。 您必須具備至少一個名稱空間的存取權，才能
在 IBM Cloud Container Registry 中處理 Docker 映像檔。

OK
```

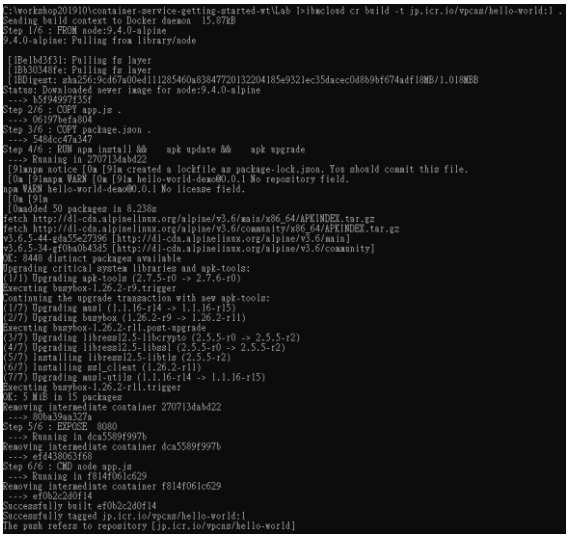
  Command:
  ibmcloud cr namespace-add <namespace>
  <namespace>: vpncs_”yourID”

```
C:\workshop201910\container-service-getting-started-wt\Lab 1>ibmcloud cr namespace-add vpc
失敗
所要求的名稱空間無效。
請選擇有效的名稱空間值。 名稱空間的長度必須介於 4 到 30 個字元之間，而且只能包含小寫字母、數字、連字號和底線。 名稱空間的開頭和結尾必須是字母或數字
。
```

```
C:\workshop201910\container-service-getting-started-wt\Lab 1>ibmcloud cr namespace-add vpcns
正在新增名稱空間 'vpcns'...

已順利新增名稱空間 'vpcns'

OK
```

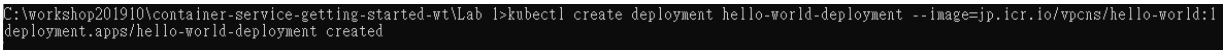
- Build a Docker image that includes the app files of the Lab 1 directory, and push the image to the IBM Cloud Container Registry namespace that you created in the previous tutorial. If you need to change the app in the future, repeat these steps to create another version of the image. Note: Learn more about securing your personal information when you work with container images.
  Use lowercase alphanumeric characters or underscores (_) only in the image name. Don't forget the period (.) at the end of the command. The period tells Docker to look inside the current directory for the Dockerfile and build artifacts to build the image.
  Command:
  ibmcloud cr build -t <registry region>/<namespace>/hello-world:1 .
  <registry region>: jp.icr.io, us.icr.io…
  <namespace>: should be the same as your prior command created

  When the build is complete, verify that you see the following success message:
  Example output:
  Successfully built <image_ID>
  Successfully tagged us.icr.io/<namespace>/hello-world:1
  The push refers to a repository [us.icr.io/vpc/hello-world]
  29042bc0b00c: Pushed
  f31d9ee9db57: Pushed
  33c64488a635: Pushed
  0804854a4553: Layer already exists
  6bd4a62f5178: Layer already exists
  9dfa40a0da3b: Layer already exists
  1: digest:
  sha256:f824e99435a29e55c25eea2ffcbb84be4b01345e0a3efbd7d9f238880d63d4a5
  size: 1576

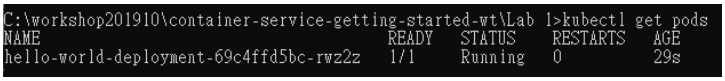


- Create a deployment for your app. Deployments are used to manage pods, which include containerized instances of an app. The following command deploys the app in a single pod. For the purposes of this tutorial, the deployment is named **hello-world-deployment**, but you can give the deployment any name that you want.
  Command:
  kubectl create deployment hello-world-deployment --image=<registry region>/<namespace>/hello-world:1
  <registry region>: jp.icr.io, us.icr.io…
  <namespace>: should be the same as your prior command created

  Example output:
  deployment "hello-world-deployment" created



  Check the pods that run your app
  Command:
  kubectl get pods



- Make the app accessible by exposing the deployment as a NodePort service. Because your VPC worker nodes are connected to a private subnet only, the NodePort is

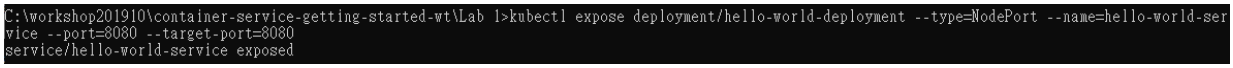
assigned only a private IP address and is not exposed on the public network. Other services that run on the private network can access your app by using the private IP address of the NodePort service.

Command:

kubectl expose deployment/hello-world-deployment --type=NodePort --name=hello-world-service --port=8080 --target-port=8080

Example output:

service "hello-world-service" exposed



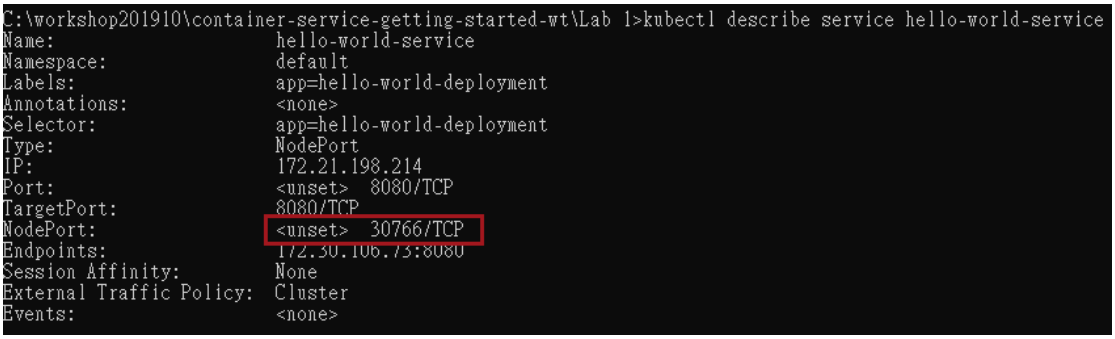
- Now that all the deployment work is done, you can test your app from within the cluster. Get the details to form the private IP address that you can use to access your app.
  - Get information about the service to see which NodePort was assigned. The NodePorts are randomly assigned when they are generated with the expose command, but within 30000-32767. In this example, the **NodePort** is 30872.

    Command:

    kubectl describe service hello-world-service

    Example output:

    Name: hello-world-service
    Namespace: default
    Labels: run=hello-world-deployment
    Selector: run=hello-world-deployment
    Type: NodePort
    IP: 10.xxx.xx.xxx
    Port: <unset> 8080/TCP
    NodePort: <unset> 30872/TCP
    Endpoints: 172.30.xxx.xxx:8080
    Session Affinity: None
    No events.



- List the pods that run your app, and note the pod name.
  Command:
  kubectl get pods

  Example output:
  NAME READY STATUS RESTARTS AGE
  hello-world-deployment-d99cddb45-lmj2v 1/1 Running 0 2d

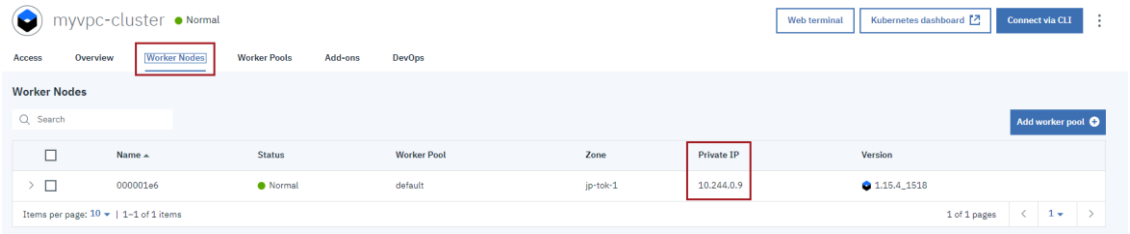
- Describe your pod to find out what worker node the pod is running on. In the example output, the worker node that the pod runs on is **172.30.xxx.xxx**.
  Command:
  kubectl describe pod <pod name>
  <pod name>: you can get the info. from previous screen

  Example output:
  Name: hello-world-deployment-d99cddb45-lmj2v
  Namespace: default
  Priority: 0
  PriorityClassName: <none>
  Node: 10.xxx.xx.xxx/10.xxx.xx.xxx
  Start Time: Mon, 22 Apr 2019 12:40:48 -0400
  Labels: pod-template-hash=d99cddb45
  run=hello-world-deployment
  Annotations: kubernetes.io/psp=ibm-privileged-psp
  Status: Running
  IP: 172.30.xxx.xxx

…

```
C:\workshop201910\container-service-getting-started-wt\Lab 1>kubectl describe pod hello-world-deployment-69c4ffd5bc-rwz2z
Name:               hello-world-deployment-69c4ffd5bc-rwz2z
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               10.244.0.12/10.244.0.12
Start Time:         Sun, 13 Oct 2019 01:14:16 +0800
Labels:             app=hello-world-deployment
                    pod-template-hash=69c4ffd5bc
Annotations:        kubernetes.io/psp: ibm-privileged-psp
Status:             Running
IP:                 172.30.106.73
Controlled By:      ReplicaSet/hello-world-deployment-69c4ffd5bc
Containers:
  hello-world:
    Container ID:   containerd://8bd2e77846297f072fb2ab1a6c156837f1a9b176024a13bb4cc446cdd08894f5
    Image:          jp.icr.io/vpcns/hello-world:1
    Image ID:       jp.icr.io/vpcns/hello-world@sha256:436a06b25c4096c5818682abfbb2a06a684c900167a6ddc6257a5f4f929727c2
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Sun, 13 Oct 2019 01:14:20 +0800
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-dwkz6 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-dwkz6:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-dwkz6
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 600s
                 node.kubernetes.io/unreachable:NoExecute for 600s
Events:
  Type    Reason     Age    From                Message
  ----    ------     ----   ----                -------
  Normal  Scheduled  9m24s  default-scheduler   Successfully assigned default/hello-world-deployment-69c4ffd5bc-rwz2z to 10.244.0.12
  Normal  Pulling    9m23s  kubelet, 10.244.0.12  Pulling image "jp.icr.io/vpcns/hello-world:1"
  Normal  Pulled     9m20s  kubelet, 10.244.0.12  Successfully pulled image "jp.icr.io/vpcns/hello-world:1"
  Normal  Created    9m20s  kubelet, 10.244.0.12  Created container hello-world
  Normal  Started    9m20s  kubelet, 10.244.0.12  Started container hello-world
```

You also can find the worker node private IP from portal page



- Log in to the pod so that you can make a request to your app from within the cluster.
  Command:
  kubectl exec -it <pod name> /bin/sh
  <pod name>: should be the same as your prior command used

```
C:\workshop201910\container-service-getting-started-wt\Lab 1>kubectl exec -it hello-world-deployment-69c4ffd5bc-rwz2z /bin/sh
/ #
```

- Make a request to the NodePort service by using the worker node private IP address and the node port that you previously retrieved.
  Command:
  wget -O - <worker node private IP>:<node port>

<worker node private IP>: you can get the info. from previous screen or portal page
<node port>: you can get the info. from previous screen

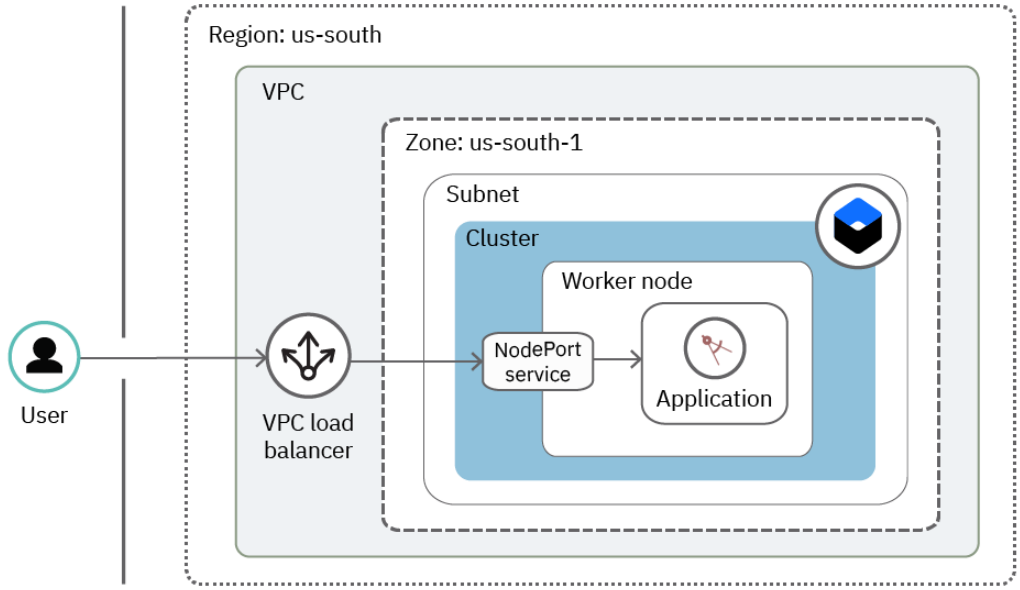
Example output:
Connecting to 10.xxx.xx.xxx:30872 (10.xxx.xx.xxx:30872)
Hello world from hello-world-deployment-d99cddb45-lmj2v! Your app is up and running in a cluster!
- 100%
|***|
88 0:00:00 ETA

- To close your pod session, enter exit.
  Command:
  exit

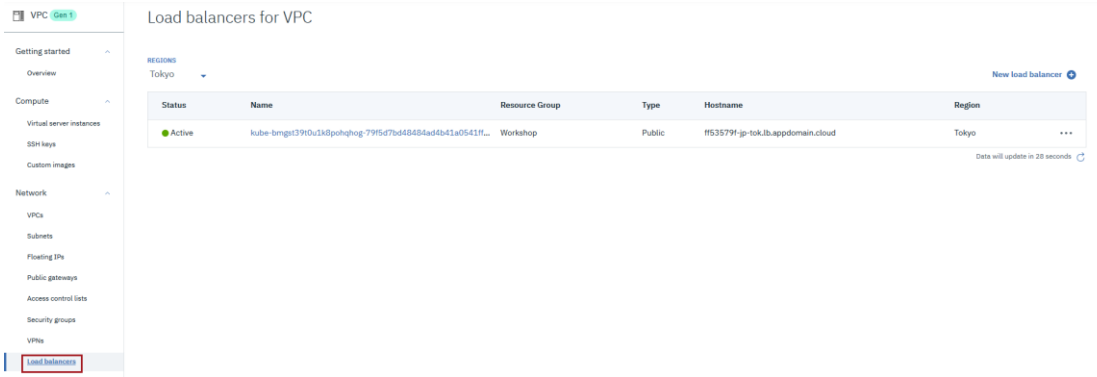
IBM Cloud

Lesson 3: Setting up a Load Balancer for VPC to expose your app publicly

Set up a VPC load balancer to expose your app on the public network.
When you create a Kubernetes LoadBalancer service in your cluster, a load balancer for VPC is automatically created in your VPC outside of your cluster. The load balancer is multizonal and routes requests for your app through the private NodePorts that are automatically opened on your worker nodes. The following diagram illustrates how a user accesses an app's services through the load balancer, even though your worker node is connected to only a private subnet.



Before create a Kubernetes LoadBalancer service, you can see a load balancer for VPC existed due to Kubernetes cluster creation.

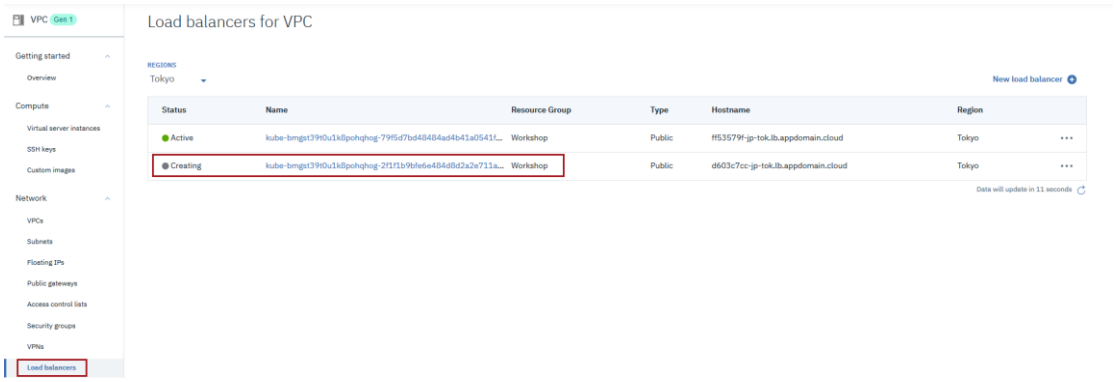


- Create a Kubernetes LoadBalancer service in your cluster to publicly expose the hello world app.
  Command:
  kubectl expose deployment/hello-world-deployment --type=LoadBalancer --name=<load balancer name> --port=8080 --target-port=8080
  <load balancer name>: hw-lb-svc_“yourID”

  Example output:
  service "hw-lb-svc" exposed

```
C:\workshop201910\container-service-getting-started-wt\Lab 1>kubectl expose deployment/hello-world-deployment --type=LoadBalancer --name=hw-lb-svc
--port=8080 --target-port=8080
service/hw-lb-svc exposed
```

  You can see a new load balancer for VPC is creating



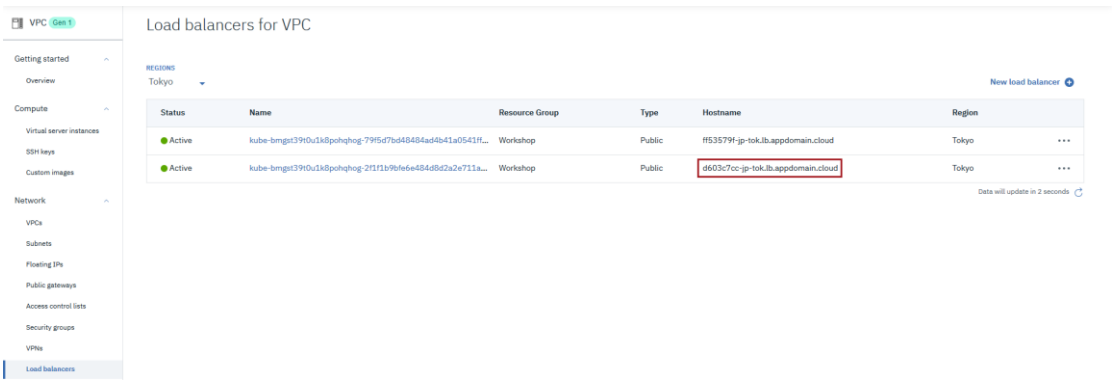
- Verify that the Kubernetes LoadBalancer service is created successfully in your cluster. When the Kubernetes LoadBalancer service is created, the LoadBalancer Ingress field is populated with a hostname that is assigned by the VPC load balancer that is automatically created.
  Command:
  kubectl describe service <load balancer name>
  <load balancer name>: should be the same as your prior command created

  Example CLI output:
  Name: hw-lb-svc
  Namespace: default
  Labels: app=hello-world-deployment
  Annotations: <none>
  Selector: app=hello-world-deployment
  Type: LoadBalancer
  IP: 172.21.xxx.xxx

LoadBalancer Ingress: 1234abcd-us-south.lb.appdomain.cloud
Port: <unset> 8080/TCP
TargetPort: 8080/TCP
NodePort: <unset> 32040/TCP
Endpoints:
Session Affinity: None
External Traffic Policy: Cluster
Events:

Type	Reason	Age	From	Message
Normal	EnsuringLoadBalancer	1m	service-controller	Ensuring load balancer
Normal	EnsuredLoadBalancer	1m	service-controller	Ensured load balancer

```
C:\workshop201910\container-service-getting-started-wt\Lab 1>kubectl describe service hw-lb-svc
Name:                     hw-lb-svc
Namespace:                default
Labels:                   app=hello-world-deployment
Annotations:              <none>
Selector:                 app=hello-world-deployment
Type:                     LoadBalancer
IP:                       172.21.49.137
LoadBalancer Ingress:     d603c7cc-jp-tok.lb.appdomain.cloud
Port:                     <unset>  8080/TCP
TargetPort:               8080/TCP
NodePort:                 <unset>  30456/TCP
Endpoints:                172.30.106.73:8080
Session Affinity:         None
External Traffic Policy:  Cluster
Events:
  Type    Reason                Age    From                Message
  ----    ------                ----   ----                -------
  Normal  EnsuringLoadBalancer  3m19s  service-controller  Ensuring load balancer
  Normal  EnsuredLoadBalancer   3m5s   service-controller  Ensured load balancer
```



- Verify that the VPC load balancer is created successfully in your VPC. In the output, verify that the VPC load balancer has an **Operating Status** of online and a **Provision Status** of active.

  Tip: The VPC load balancer is named in the format kube-<cluster_ID>-<kubernetes_lb_service_UID>. To see your cluster ID, run ibmcloud ks cluster get --

cluster <cluster_name>. To see the Kubernetes LoadBalancer service UID, run kubectl get svc hw-lb-svc -o yaml and look for the **metadata.uid** field in the output.
Command:
ibmcloud is load-balancers

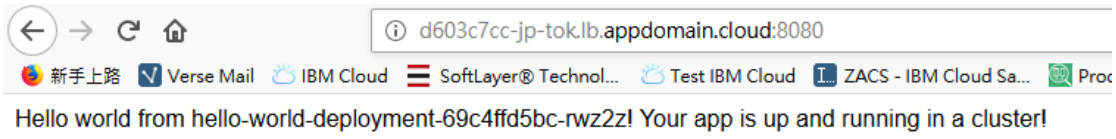
In the following example CLI output, the VPC load balancer that is named kube-bh077ne10vqpekt0domg-046e0f754d624dca8b287a033d55f96e is created for the hw-lb-svc Kubernetes LoadBalancer service:

Example output:
ID Name
Created Host Name Is Public Listeners
Operating Status Pools Private IPs
Provision Status Public IPs Subnets
Resource Group
06496f64-a689-4693-ba23-320959b7b677 kube-bh077ne10vqpekt0domg-046e0f754d624dca8b287a033d55f96e 8 minutes ago 1234abcd-us-south.lb.appdomain.cloud yes 95482dcf-6b9b-4c6a-be54-04d3c46cf017
online 717f2122-5431-403c-b21d-630a12fc3a5a 10.1.1.1,10.1.1.2
active 169.1.1.1,169.1.1.2 c6540331-1c1c-40f4-9c35-aa42a98fe0d9
00809211b934565df546a95f86160f62

- Curl the hostname and port of the Kubernetes LoadBalancer service that is assigned by the VPC load balancer. Example:
  Command:
  curl <LoadBalancer Ingress>:8080
  <LoadBalancer Ingress>: you can get the info. from previous screen

  Example output:
  Hello world from hello-world-deployment-5fd7787c79-sl9hn! Your app is up and running in a cluster!

You also can use browser to test.



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.

IBM Cloud

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).