IBM Blockchain

IBM Blockchain
Hyperledger Fabric Hands-On
Create your blockchain on IBM Cloud

Lab One

Contents

1 Overview

In this lab, you will deploy in the IBM Cloud a preconfigured blockchain consisting of:

- Two organisations Org1 and Org2
- One peer per organisation called peer1 and peer2
- One orderer called orderer
- One certificate authority for each organisation running on one ca server.

You will install the following piece of software:

- Hyperledger Fabric 1.1.0
- Hyperledger Composer v0.19.5
- Hyperledger Composer REST Server 0.19.5

1.1 Introduction

There are several prerequisites to be fulfilled for this lab:

- An individual IBM Cloud account is required for each attendee in the session.
- A Kubernetes cluster deployed in the IBM Cloud, using the “Containers Services” free offering.
- A workstation running Linux, or Windows 10 with the Windows Subsystem for Linux.

The first section of the lab will guide you through the fulfilment of those prerequisites.

2 Preparing your lab workstation for the lab.

This section focuses on preparing your workstation for the lab.

The lab requires to have a Linux “like” environment. MacOS and Linux workstations have already a system compatible with Linux.

For the Windows 10 workstations, you have to install the Windows 10 Linux subsystem as described in this section.

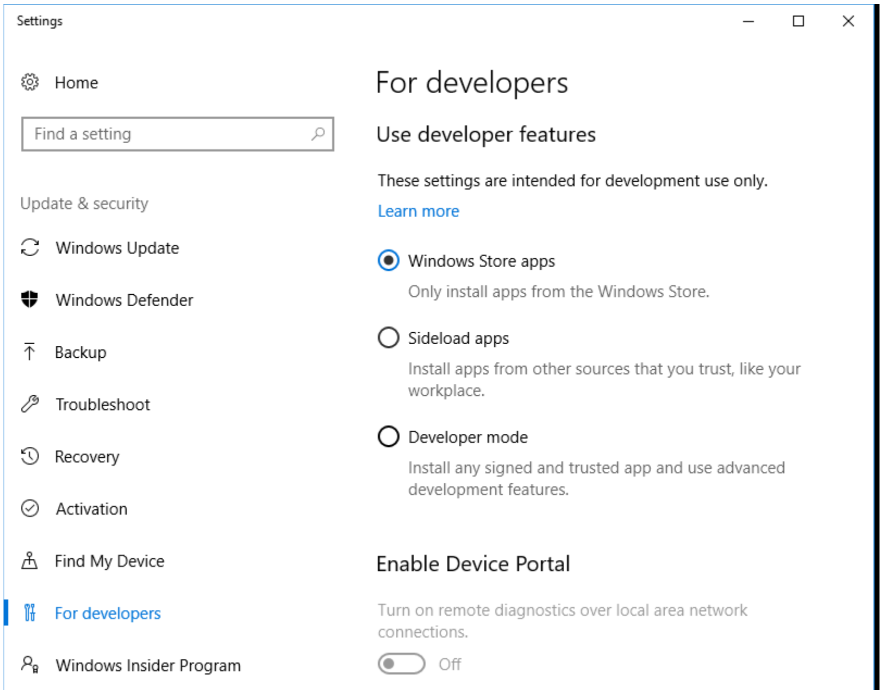
You must own an IBM Cloud account to be able to do this hands-on. Contact your instructor if this is not the case.

2.1 Enable the Linux subsystem only for Windows 10 user

All commands in these hands-on labs will be issued in the Windows Linux subsystem. The following steps will allow you to install it on your PC.

2.1.1 Open the “Settings” panel and then search for the “For Developers” section,

2.1.2 Enable Developer Mode, as shown below:



For developers

Use developer features

These settings are intended for development use only.

Learn more

● Windows Store apps
Only install apps from the Windows Store.

○ Sideload apps
Install apps from other sources that you trust, like your workplace.

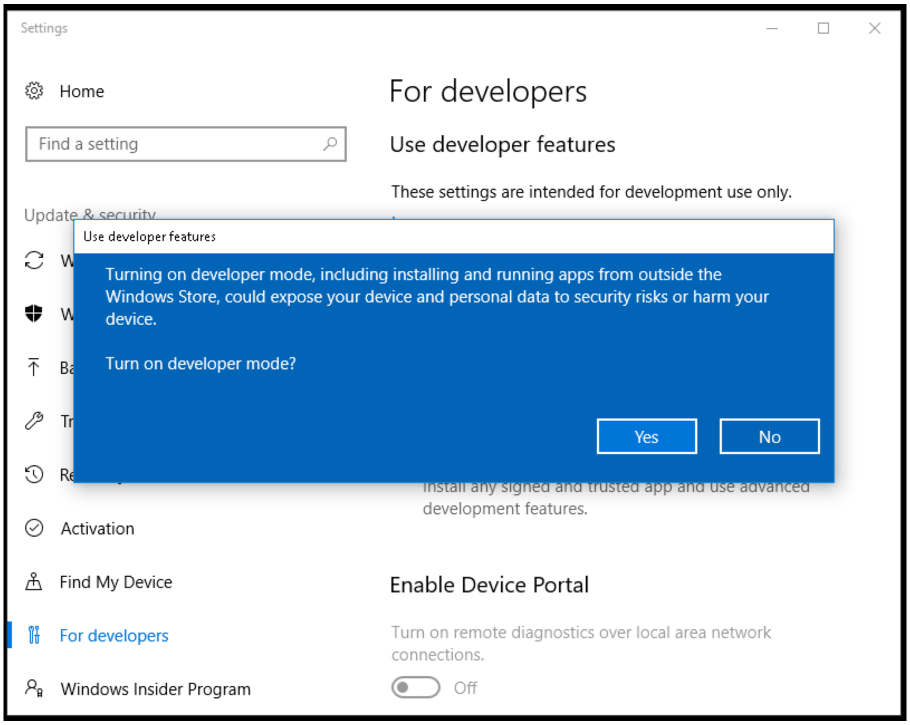
○ Developer mode
Install any signed and trusted app and use advanced development features.

Enable Device Portal

Turn on remote diagnostics over local area network connections.

Off

2.1.3 Once the selection is made, just click Yes on the popup window to proceed with the installation of the required Windows components, as shown below:



For developers

Use developer features

These settings are intended for development use only.

Use developer features

Turning on developer mode, including installing and running apps from outside the Windows Store, could expose your device and personal data to security risks or harm your device.

Turn on developer mode?

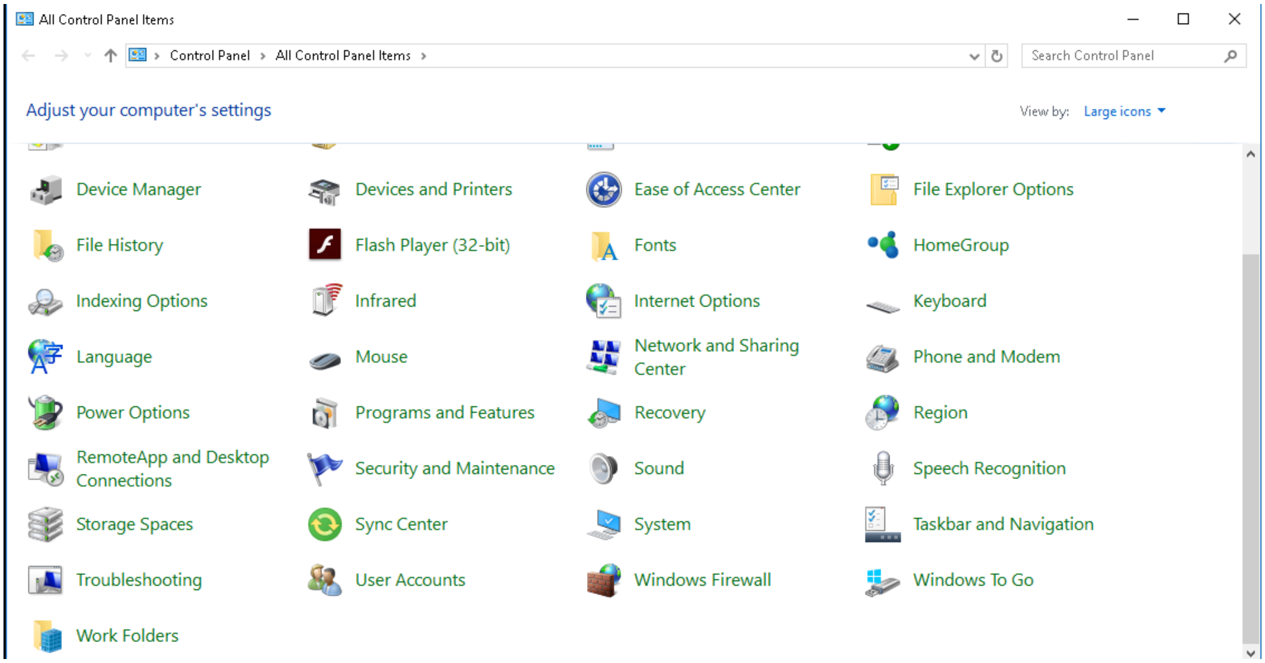
Install any signed and trusted app and use advanced development features.

Enable Device Portal

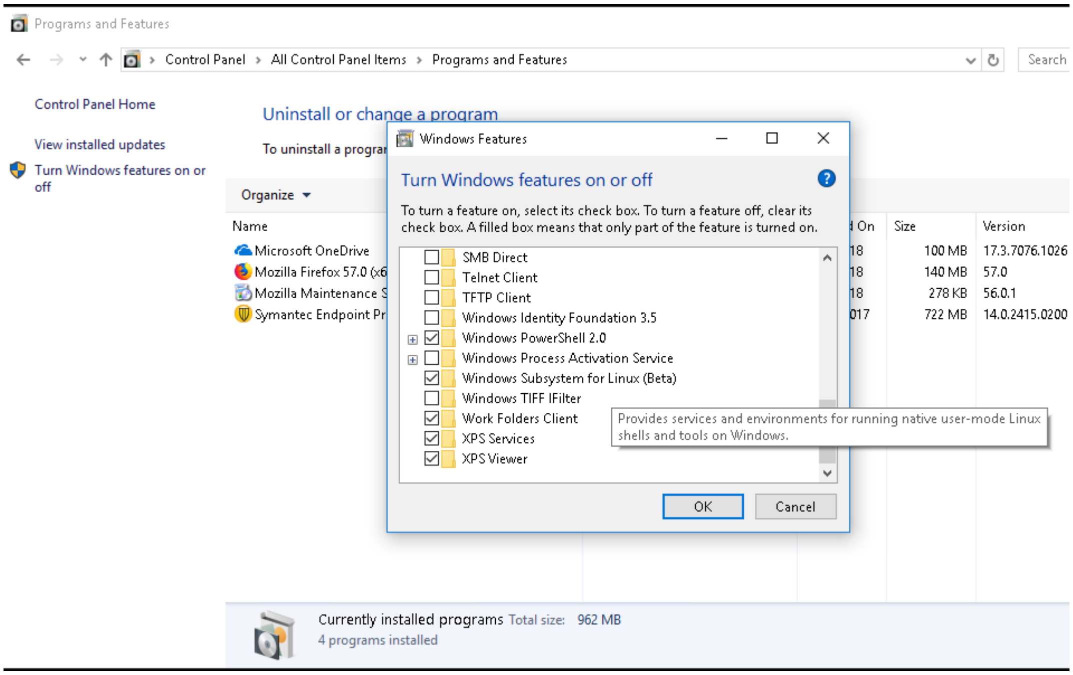
Turn on remote diagnostics over local area network connections.

2.1.4 Activate the Windows Subsystem for Linux

Once Developer Mode is enabled, go to the Control Panel, then open the “**Programs and Features**” section:



2.1.5 Then select “**Turn Windows features on or off**” in the left hand-side menu, to finally navigate down to the “**Windows Substem for Linux (beta)**” and select it.


Programs and Features
Control Panel > All Control Panel Items > Programs and Features
Search
Control Panel Home
View installed updates
Turn Windows features on or off
Uninstall or change a program
To uninstall a prograr
Organize ▾

Name	d On	Size	Version
Microsoft OneDrive	18	100 MB	17.3.7076.1026
Mozilla Firefox 57.0 (x6	18	140 MB	57.0
Mozilla Maintenance S	18	278 KB	56.0.1
Symantec Endpoint Pr	017	722 MB	14.0.2415.0200

Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- □ SMB Direct
- □ Telnet Client
- □ TFTP Client
- □ Windows Identity Foundation 3.5
- ⊞ ☑ Windows PowerShell 2.0
- ⊞ □ Windows Process Activation Service
- ☑ Windows Subsystem for Linux (Beta)
- □ Windows TIFF IFilter
- ☑ Work Folders Client
- ☑ XPS Services
- ☑ XPS Viewer

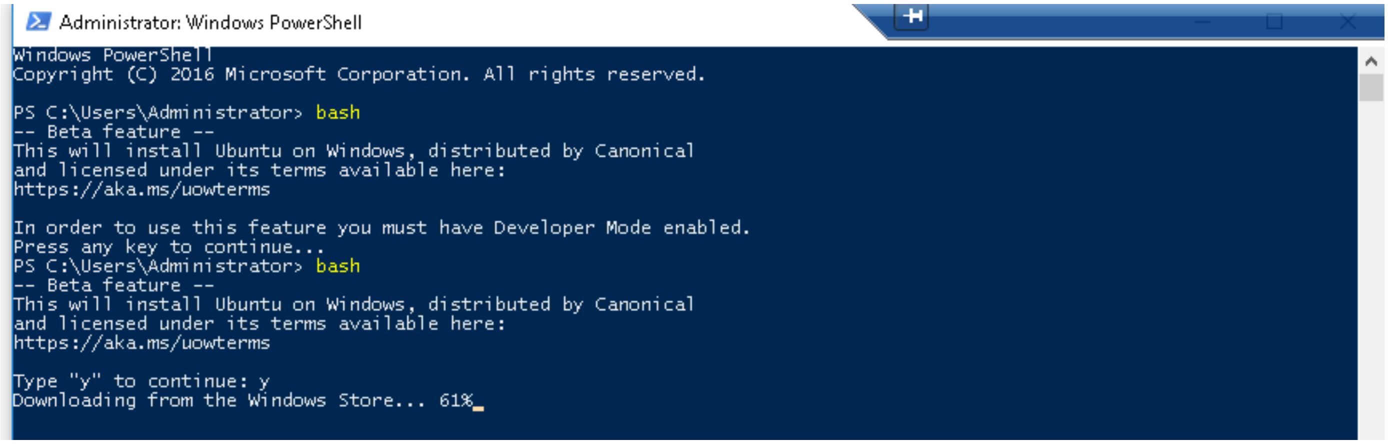
Provides services and environments for running native user-mode Linux shells and tools on Windows.

Currently installed programs Total size: 962 MB
4 programs installed

Once installation is completed, agree to restart the machine.

2.2 Install Ubuntu Linux Subsystem on Windows10

2.2.1 Open a command prompt, and issue the “bash” command, as shown below:



```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> bash
-- Beta feature --
This will install Ubuntu on Windows, distributed by Canonical
and licensed under its terms available here:
https://aka.ms/uowterms

In order to use this feature you must have Developer Mode enabled.
Press any key to continue...
PS C:\Users\Administrator> bash
-- Beta feature --
This will install Ubuntu on Windows, distributed by Canonical
and licensed under its terms available here:
https://aka.ms/uowterms

Type "y" to continue: y
Downloading from the Windows Store... 61%_
```

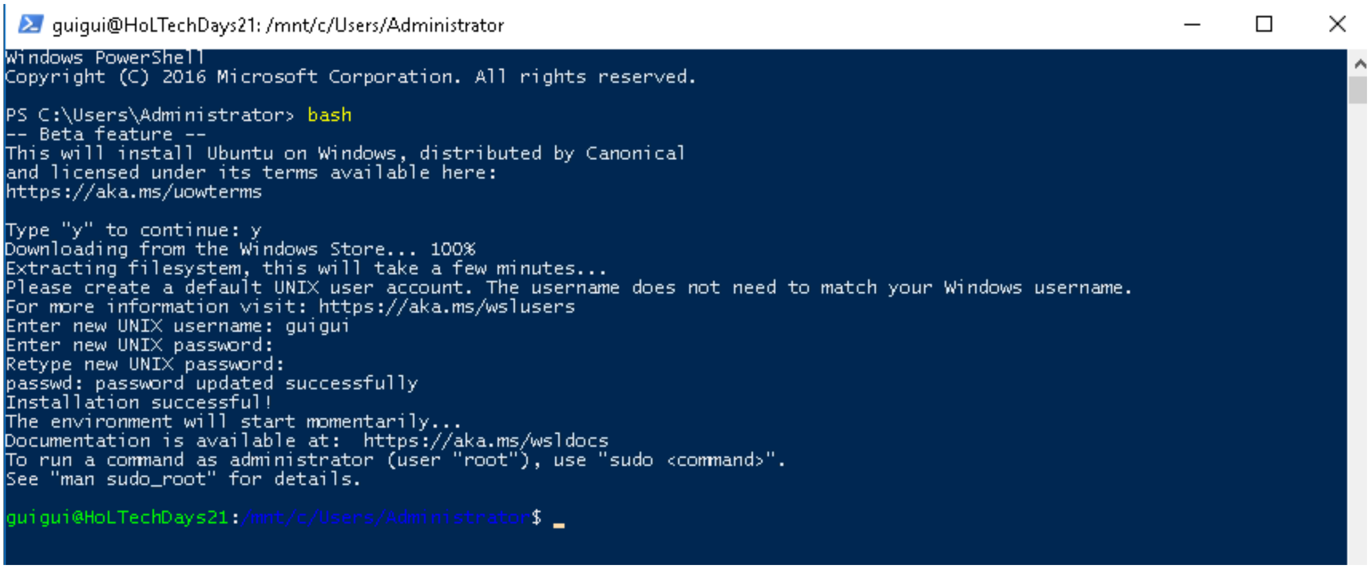
Then run the command :

lxrun /install

You’ll be prompted to install Ubuntu on Windows (the code will then be downloaded and installed).

2.2.2 When done, you will be prompted to create a user. Use the following credentials:

Username: **blockchain**
Password: **block4ever**



```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> bash
-- Beta feature --
This will install Ubuntu on Windows, distributed by Canonical
and licensed under its terms available here:
https://aka.ms/uowterms

Type "y" to continue: y
Downloading from the Windows Store... 100%
Extracting filesystem, this will take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: guigui
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Installation successful!
The environment will start momentarily...
Documentation is available at:  https://aka.ms/wsldocs
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

guigui@HoLTechDays21:/mnt/c/Users/Administrator$ _
```

3 Linux Subsystem software prerequisites

In this section, we are installing the tools required for the lab:

- Bluemix/IBM Cloud CLI tools
- Kubernetes command

3.1 Install the Bluemix command line environment

3.1.1 For Windows 10 user:

1. download the Bluemix CLI installer by issuing in the Linux terminal the following command:

```
cd
wget https://clis.ng.bluemix.net/download/bluemix-cli/latest/linux64
```

2. Untar the downloaded file, by issuing the following command:

```
tar -zxvf  linux64
```

3. Run the installation program:

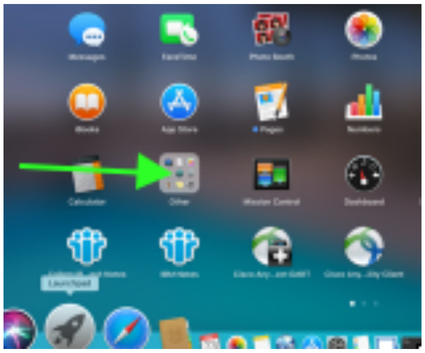
```
Cd Bluemix_CLI
./install_bluemix_cli
```

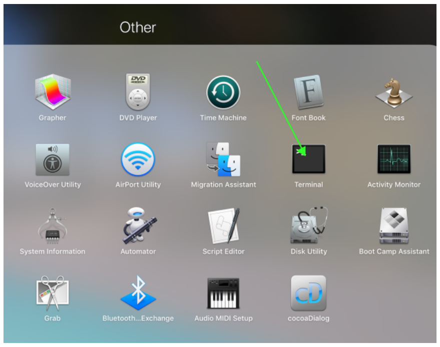
4. Install the Bluemix Container Service by issuing the following command in the Windows Linux subsystem

```
bx plugin install container-service -r bluemix
```

3.1.2 For MacOS user:

1. Open a terminal window (via Launchpad, select Other icon, then click on Terminal icon)





2. download the Bluemix CLI installer and install it by issuing in the terminal the following command:

```
curl -fsSL https://clis.ng.bluemix.net/install/osx | sh
```

It will download then execute the install.
Starting the execution, it will ask for the password of your system. Fill in the password and press Enter.
The result should be :

```
        installer: Package name is IBM Cloud Command Line Interface
        installer: Installing at base path /
        installer: The install was successful.
```

The path of the Bluemix CLI is /usr/local/Bluemix

3. Install the Bluemix Container Service by issuing the following command in the Windows Linux subsystem

```
        bx plugin install container-service -r bluemix
```

The result should be:

```
Plug-in 'container-service x.x.xxx' was successfully installed into
/Users/<user>/.bluemix/plugins/container-service. Use 'bx plugin show
container-service' to show its details.
```

3.2 Deploy the IBM Cloud Container Service

3.2.1 Select an IBM Cloud datacenter (regions)

Available production API endpoints can be retrieved by issuing the following command in a Linux terminal

```
        bx regions
```

Select your IBM Cloud datacenter by issuing the following:

```
        bx api https://api.eu-de.bluemix.net
```

3.2.2 Login the IBM Cloud by issuing the following command:

```
        bx login
```

3.2.3 Create the cluster container service

Use “blockchain” as cluster container name and issue the following command in the Linux subsystem to create your cluster:

```
        bx cs cluster-create --name blockchain
```

You should see:

```
        Creating cluster...
        The machine-type flag was not specified. So a lite cluster with default
        parameters will be created. To customize the parameters, create a standard
        cluster and include all required flags.
        OK
```

You can monitor progress using the **bx cs clusters** command

```
      bx cs clusters
OK
Name         ID           State       Created          Workers    Datacenter   Version
blockchain   4a40[…]37d   deploying   8 minutes ago    1          mil01
1.7.4_1506
```

The state of the cluster goes from requesting to pending to deploying to normal.
The provisioning of the cluster requires about 10 to 45 minutes.

3.2.4 Retrieve the network configuration of the cluster

Retrieve the public IP address of your cluster by issuing the following commands

```
      bx cs workers blockchain
```

What is your public IP address?

3.3 Install Kubernetes

3.3.1 Download Lab setup scripts.

Issue the following commands:

```
      cd
      mkdir git
      cd git

      git clone https://github.com/jyg007/lab.git
      cd lab
      ./LabInstallKubectl
```

3.3.2 Close your current Linux terminal and open a new one

4 Deploy your first blockchain on Hyperledger fabric

In this section, you will discover how to deploy the various components from the Hyperledger project constituting the Developer Sandbox environment.

You will deploy the following Hyperledger Fabric infrastructure:

- One fabric orderer
- Two organizations running one peer each one and managed by one certificate server
- A composer playground server

4.1 Issue the following command to retrieve installation script

```
       cd
       cd git
       git clone https://github.com/ovallod/hlfv11.git
```

4.2 Create persistent storage for the blockchain cluster

Issue the following commands to create persistent storage, then retrieve the config files for the Hyperledger Fabric installation:

```
       cd hlfv11/cs-offerings/scripts
       create/create_storage.sh
       create/create_sampleconfig.sh
```

4.3 Create your blockchain

Issue the following command to create the blockchain topology

```
       create/create_blockchain.sh --with-couchdb
```

You should see the following log

```
Configured to setup network with couchdb
Creating Services for blockchain network
Running: kubectl create -f /home/blockchain/ibm-container-service/cs-
offerings/scripts/../kube-configs/blockchain-couchdb-services-free.yaml
service "blockchain-ca" created
service "blockchain-orderer" created
service "blockchain-org1peer1" created
service "blockchain-org2peer1" created
service "blockchain-couchdb1" created
service "blockchain-couchdb2" created
Creating new Deployment
Running: kubectl create -f /home/blockchain/ibm-container-service/cs-
offerings/scripts/../kube-configs/blockchain-couchdb.yaml
pod "utils" created
deployment "blockchain-orderer" created
deployment "blockchain-ca" created
deployment "blockchain-org1peer1" created
deployment "blockchain-org2peer1" created
deployment "blockchain-couchdb1" created
deployment "blockchain-couchdb2" created
```

```
Checking if all deployments are ready
Waiting on pending deployments. Deployments pending = 6
Waiting on pending deployments. Deployments pending = 6
[…]
Waiting on pending deployments. Deployments pending = 5
Waiting on pending deployments. Deployments pending = 4
Waiting on pending deployments. Deployments pending = 3
[…]
Waiting on pending deployments. Deployments pending = 3
Waiting on pending deployments. Deployments pending = 1
Waiting for all containers in Utils pod to complete. Left = 2/3
Waiting for 15 seconds for peers and orderer to settle
```

4.4 Create a fabric channel.

The scripts provide an easy way to create a Blockchain channel. The name of the channel is defined as `labchannel`.

Issue the following commands to easily create the channel and have the two peers joined the channel:

```
      CHANNEL_NAME="labchannel" create/create_channel.sh

      CHANNEL_NAME="labchannel" PEER_MSPID="Org1MSP" PEER_ADDRESS="blockchain-
org1peer1:30110" MSP_CONFIGPATH="/shared/crypto-
config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp"
create/join_channel.sh

      CHANNEL_NAME="labchannel" PEER_MSPID="Org2MSP" PEER_ADDRESS="blockchain-
org2peer1:30210" MSP_CONFIGPATH="/shared/crypto-
config/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp"
create/join_channel.sh
```

4.5 Verify the channel was successfully created

We will look at the orderer logs to check if the channel has been properly created.

3.5.1 Retrieve the name of the pod

Retrieve the name of the orderer pod using the **`kubectl get pods`** command.

3.5.2 Retrieve the pod logs

Retrieve the pod logs, using the **`kubectl logs`** command. The command takes the pod name as argument.

Can you see that the channel has been successfully created?

4.6 Check the peer has been added to the channel

Hyperledger Fabric peer command provides the *peer channel list* option to check connectivity for peers.

Use *kubectl exec* to execute this command on one of the peers.

Identify the pod name of the peer1 for you and issue the following commands:

```
        kubectl exec -it <peer1 pod name> peer channel list
```

Check in the log that you have joined the labchannel channel.

```
2018-01-18 16:55:13.271 UTC [msp] GetLocalMSP -> DEBU 001 Returning existing local MSP
2018-01-18 16:55:13.272 UTC [msp] GetDefaultSigningIdentity -> DEBU 002 Obtaining default
signing identity
2018-01-18 16:55:13.277 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer
connections initialized
2018-01-18 16:55:13.278 UTC [msp/identity] Sign -> DEBU 004 Sign: plaintext:
0A8A070A5C08031A0C08F1A283D30510...631A0D0A0B4765744368616E6E656C73
2018-01-18 16:55:13.278 UTC [msp/identity] Sign -> DEBU 005 Sign: digest:
E16F2BB8F262F4221910AA37E14542B3DAA96D4592C02AD369653384FE229C8B
2018-01-18 16:55:13.283 UTC [channelCmd] list -> INFO 006 Channels peers has joined to:
2018-01-18 16:55:13.283 UTC [channelCmd] list -> INFO 007 labchannel
2018-01-18 16:55:13.283 UTC [main] main -> INFO 008 Exiting.....
```

4.7 Install Composer

Next step is to deploy Hyperledger Composer, which you will be use in the next lab to develop and deploy a smart contract. Use the script *create_composer-playground.sh* for this.

Issue the following command:

```
        CHANNEL_NAME="labchannel" create/create_composer-playground.sh
```

Check the log

```
Creating composer-card-import pod
Running: kubectl create -f /home/blockchain/ibm-container-service/cs-offerings/scripts/../kube-
configs/composer-card-import.yaml
pod "composer-card-import" created
Waiting for composer-card-import container to be Completed
[…]
Waiting for composer-card-import container to be Completed
Composer Card Import Completed Successfully
Deleting composer-card-import pod
Running: kubectl delete -f /home/blockchain/ibm-container-service/cs-offerings/scripts/../kube-
configs/composer-card-import.yaml
pod "composer-card-import" deleted
Creating composer-playground deployment
Running: kubectl create -f /home/blockchain/ibm-container-service/cs-offerings/scripts/../kube-
configs/composer-playground.yaml
deployment "composer-playground" created
```

```
Creating composer-playground service
Running: kubectl create -f /home/blockchain/ibm-container-service/cs-offerings/scripts/../kube-
configs/composer-playground-free.yaml
service "composer-playground" created
Checking if all deployments are ready
Waiting on pending deployments. Deployments pending = 1
[…]
Waiting on pending deployments. Deployments pending = 1
Composer playground created successfully
```

At this point, you have deployed the main item of the developer sandbox environment: Hyperledger Fabric and Hyperledger Composer, both running in a single-node Kubernetes cluster in the IBM Cloud.

4.8 Kubernetes Dashboard

Kubernetes Dashboard is a general purpose, web-based UI for Kubernetes clusters. It allows users to manage applications running in the cluster and troubleshoot them, as well as manage the cluster itself.

4.8.1 Get the Kubernetes token id

You have to retrieve the token in order to sign in: use the following command and copy the id-token value :

```
kubectl config view.
```

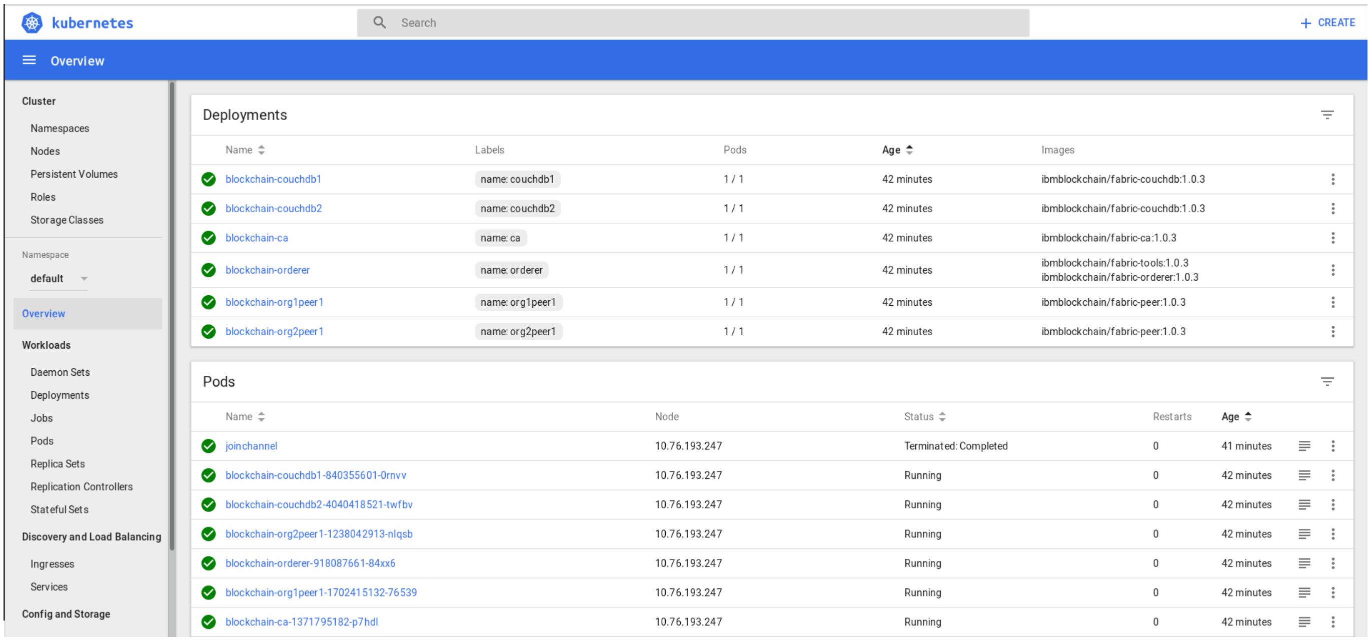
4.8.2 Launch the proxy

Start the Kubernetes proxy by issuing the following command:

```
kubectl proxy
```

4.8.3 Connect to the web interface and explore

Open your favourite web browser and connect to the following URL: http://127.0.0.1:8001/ui to start exploring the Kubernetes web interface.



Deployments

	Name	Labels	Pods	Age	Images
✓	blockchain-couchdb1	name: couchdb1	1 / 1	42 minutes	ibmblockchain/fabric-couchdb:1.0.3
✓	blockchain-couchdb2	name: couchdb2	1 / 1	42 minutes	ibmblockchain/fabric-couchdb:1.0.3
✓	blockchain-ca	name: ca	1 / 1	42 minutes	ibmblockchain/fabric-ca:1.0.3
✓	blockchain-orderer	name: orderer	1 / 1	42 minutes	ibmblockchain/fabric-tools:1.0.3 ibmblockchain/fabric-orderer:1.0.3
✓	blockchain-org1peer1	name: org1peer1	1 / 1	42 minutes	ibmblockchain/fabric-peer:1.0.3
✓	blockchain-org2peer1	name: org2peer1	1 / 1	42 minutes	ibmblockchain/fabric-peer:1.0.3

Pods

	Name	Node	Status	Restarts	Age
✓	joinchannel	10.76.193.247	Terminated: Completed	0	41 minutes
✓	blockchain-couchdb1-840355601-0rnvv	10.76.193.247	Running	0	42 minutes
✓	blockchain-couchdb2-4040418521-twfbv	10.76.193.247	Running	0	42 minutes
✓	blockchain-org2peer1-1238042913-nlqsb	10.76.193.247	Running	0	42 minutes
✓	blockchain-orderer-918087661-84xx6	10.76.193.247	Running	0	42 minutes
✓	blockchain-org1peer1-1702415132-76539	10.76.193.247	Running	0	42 minutes
✓	blockchain-ca-1371795182-p7hdl	10.76.193.247	Running	0	42 minutes

Notices

This information was developed for products and services offered in the U.S.A.
IBM may not offer the products, services, or features discussed in this document in other countries.

Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.
IBM may have patents or pending patent applications covering subject MGKer described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

Appendix A. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
System z	*Tivoli*	*WebSphere*	*Workplace*	*System p*	

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

NOTES

NOTES