

TUTORIAL

Learn step-by-step how to set up a basic blockchain network

A quick-start guide for IBM Blockchain Platform

Shikha Maheshwari | Published June 6, 2019

Blockchain Cloud

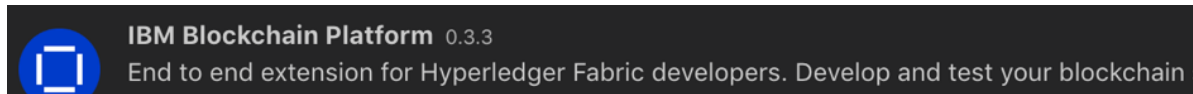
The IBM Blockchain Platform is a blockchain-as-a-service offering that provides users with deployment flexibility, scalability, and control over network components. The platform uses the Hyperledger Fabric 1.4.x code base and deploys components into a IBM Cloud Kubernetes Service that's managed and controlled by the user. The user interface simplifies and accelerates the process of deploying components of the blockchain network. This tutorial takes you through the steps for setting up a basic blockchain network with IBM Blockchain Platform.

Prerequisites

To complete this tutorial, you will need:

- Blockchain Basics: Hyperledger Fabric — You should have knowledge of blockchain networks and their components. [This article](#) can help you understand the basics of Hyperledger Fabric Network.
- An [IBM Cloud](#) account — Users with a “Lite” account will not be able to create this service.
- VS Code version 1.26 or greater.

- IBM Blockchain Platform extension for VS Code.



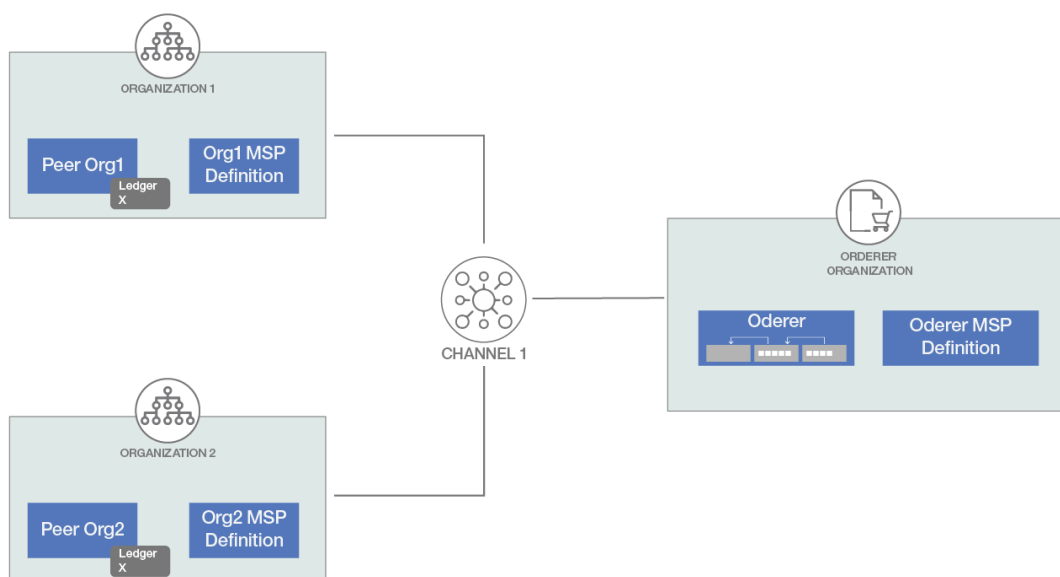
- Any sample Go/Node.js chaincode.
- [Go 1.11.x or greater](#) if using Go chaincode.
- [Node v8.x or greater](#) and npm v5.x or greater if using Node.js chaincode.

Estimated time

Deploying the Kubernetes cluster should take about 25 – 30 minutes, in addition to the time it takes to set up the Blockchain Platform. If a Kubernetes cluster already exists, then it should take about 60 minutes to complete this tutorial.

Steps

As a first step in designing any blockchain solution, you need to decide on the blockchain consortium — i.e. the blockchain network participants. Once you've chosen the components of your blockchain network (such as number of organizations, number of peers in each organization, orderer, and certificate authorities), you can start setting up the network on the cloud. If the structure of your blockchain network is similar to that shown in the following diagram...



...then, as this structure illustrates, you need to create the following:

- 3 certificate authorities (CAs)
- 3 membership service providers (MSPs)
- 2 peers
- 1 orderer
- 1 channel that includes 2 peer orgs and 1 orderer org

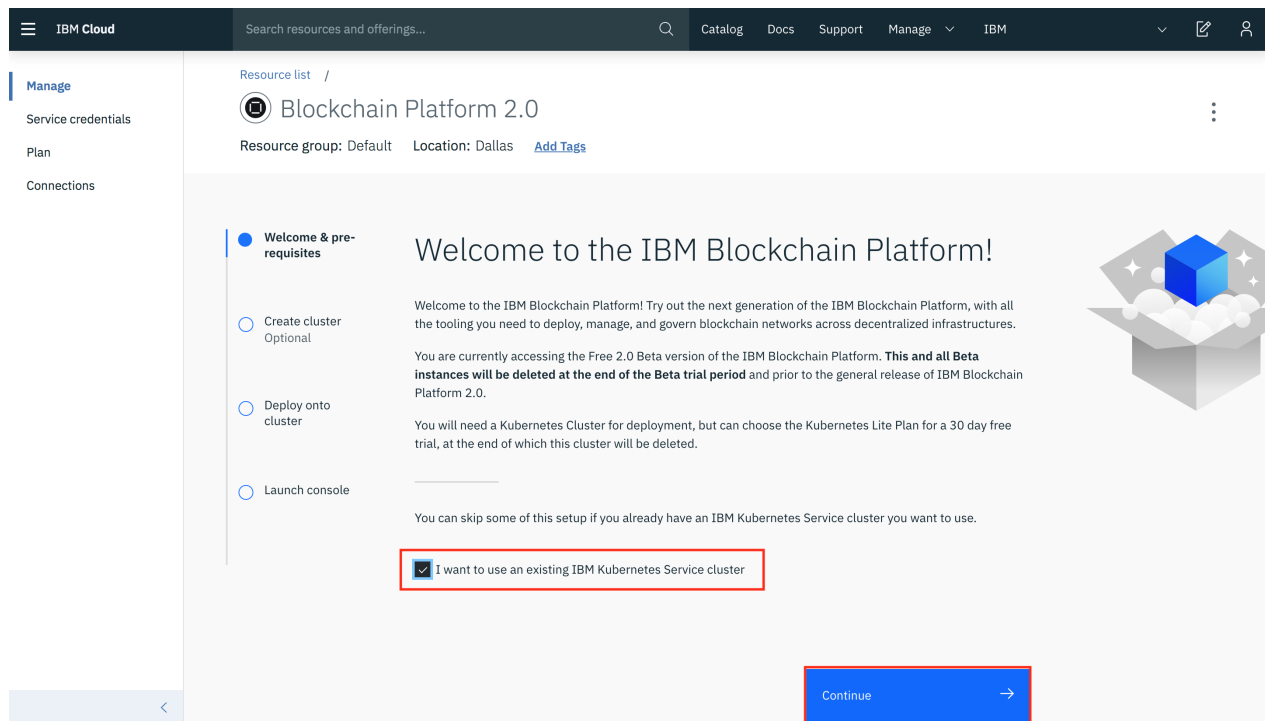
Note that this tutorial shows you how to create 1 CA, 1 MSP, 1 peer node, and so on. You should repeat the same steps in accordance with your network structure.

Follow the steps below to deploy your blockchain network using the IBM Blockchain Platform console.

1 Create the IBM Cloud Kubernetes Service

Create the [IBM Cloud Kubernetes Service \(IKS\)](#) on the IBM Cloud after you select the appropriate plan, Kubernetes version, flavor, etc. Make sure that the Kubernetes service cluster is completely deployed.

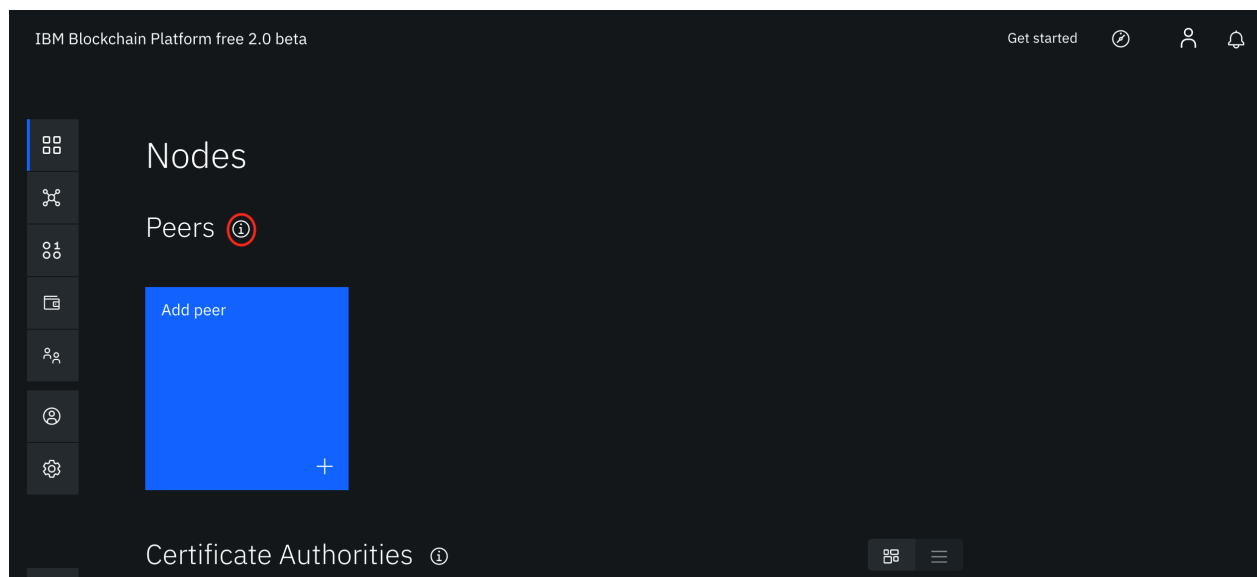
2 Create the IBM Blockchain Platform service instance on the IBM Cloud



1. Select the highlighted check-box shown in the above image if you have one existing cluster or already created one as mentioned in step 1, and click **Continue**. Provide a cluster name and click on **Deploy to Cluster**.
2. If you do not select the checkbox and click **Continue**, then it will create a new Kubernetes cluster and you need to wait until the cluster is completely deployed.

3 Launch the platform

Once it is successfully deployed to the cluster, it will give you an option to launch it. Click **Launch the IBM Blockchain Platform** and you should see the following screen:

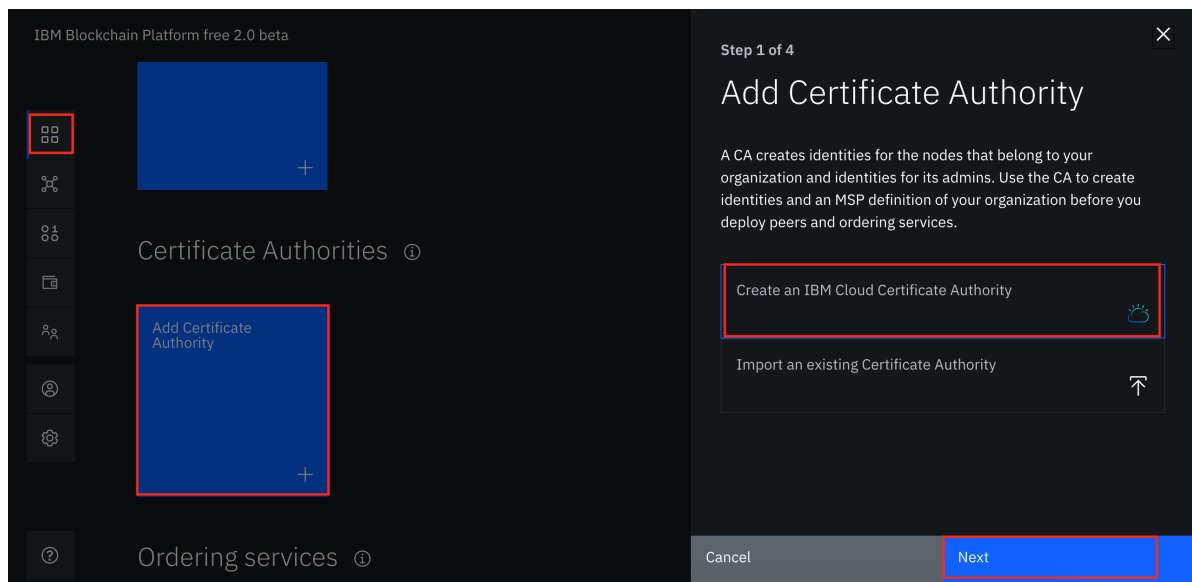


The information icon highlighted in the screenshot above gives you more details about that node/step. Such information is available to help you at every step while you're deploying the blockchain platform.

4 Add the certificate authority (CA)

As an entry point to your blockchain, you need to create a membership service provider (MSP) organization. To do this, you need an org admin user identity. The certificate authority (CA) will create all of the identities and certificates that belong to your organization in addition to defining the organization. Hence, as a first step to setting up the blockchain network, you need to create the CA.

1. Start setting your network with the CA. Click **Add Certificate Authority**, then choose **IBM Cloud** under the **Add Certificate Authority** section, and then click

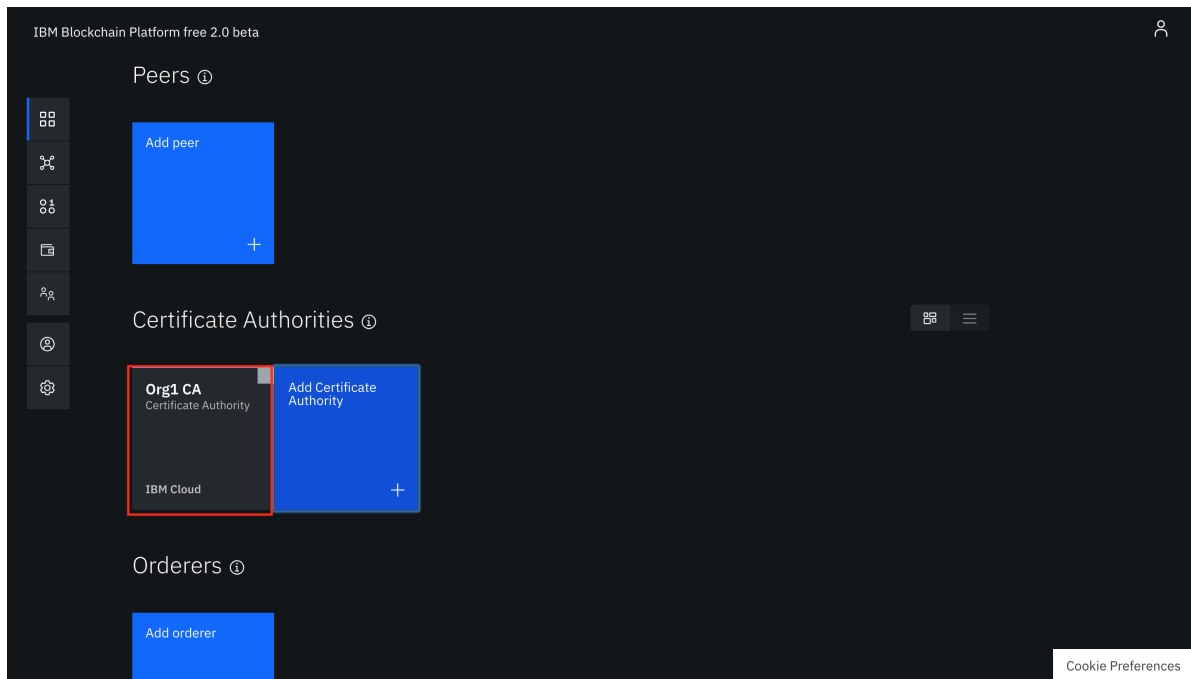
Next.

2. Provide a name for the CA (for example, “Org1CA”) and click **Next**.
3. Provide the CA administrator enroll ID and enroll secret (for example, ID “admin” and secret “adminpw”) and click **Next**.
4. When you see the summary, click **Add Certificate Authority**.

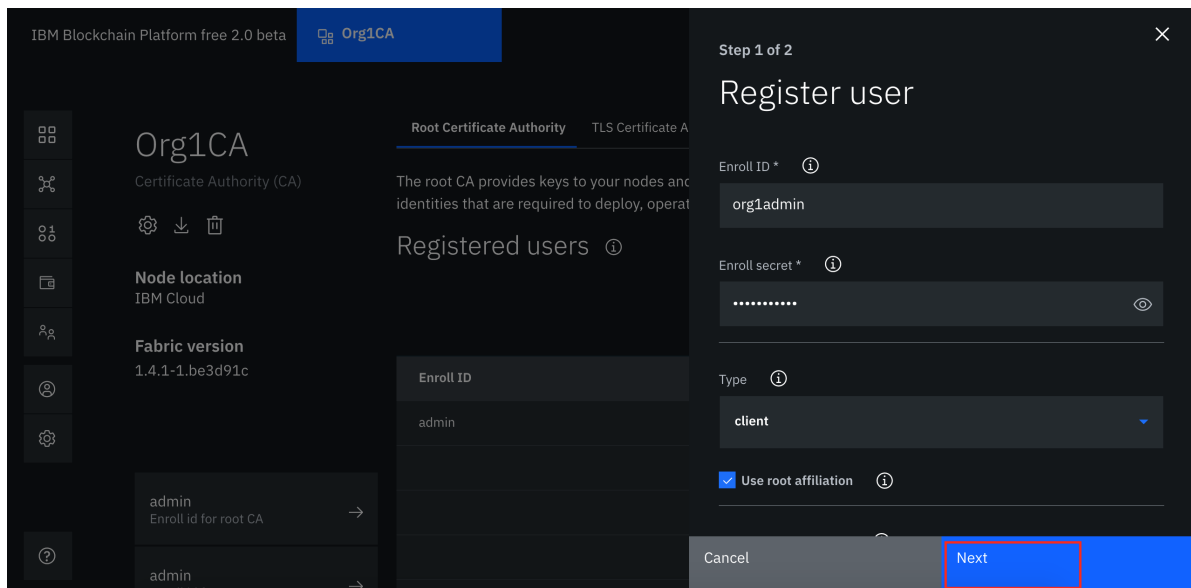
5 Register your users

You will use the CA created in the previous step to register your users:

1. Click on the CA that you created as shown here:



2. Click **Register User**, then register an admin for your organization. Provide an enroll ID (say, “org1admin”) and enroll secret (password). Set the **Type** for this identity as **client**. This identity works as an organization admin and allows you to operate nodes. The remaining attribute, **Maximum enrollments**, is optional. Leave it as blank and click **Next**.



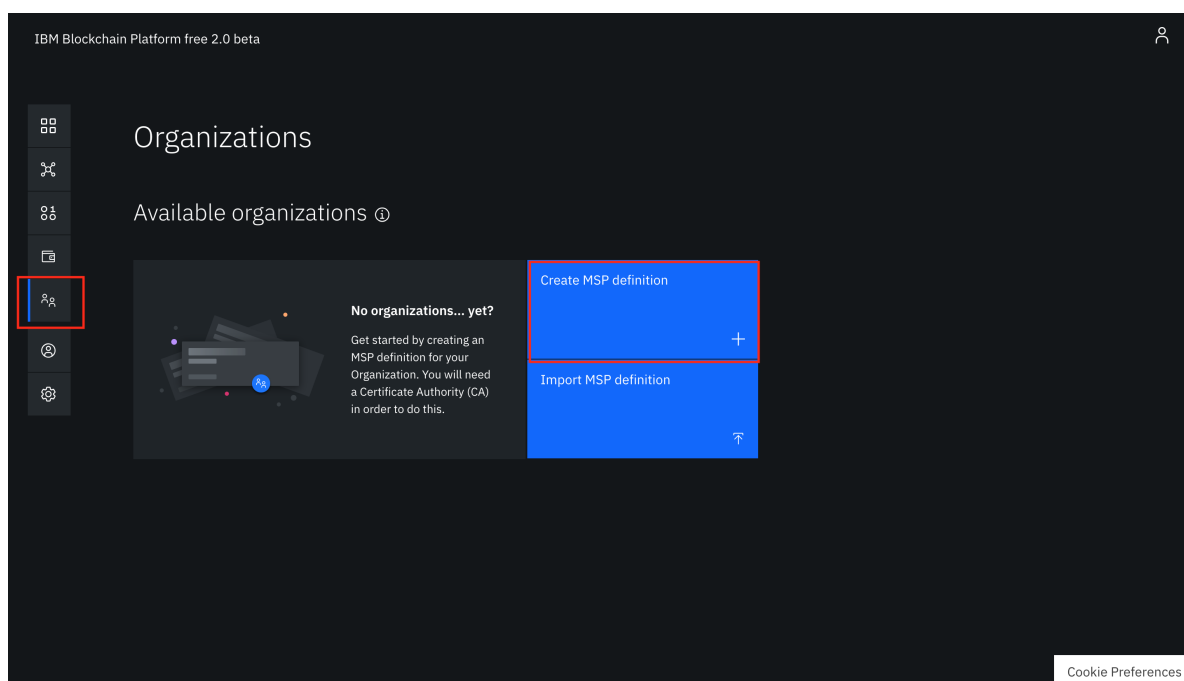
3. Next, it asks you to **Add attributes**. This will be used for role-based access control over resources. For the purposes of this tutorial, you do not need to use attributes. Click on **Register user**.

4. Repeat steps 1 – 4 above to register one more user with a **peer** type identity. This identity allows you to deploy a peer.

6 Create the organization's Membership Service Provider (MSP) definition

A formal definition of the peer's organization is known as the Membership Service Provider (MSP). You have already created the org CA and users, so you can now create the MSP definition.

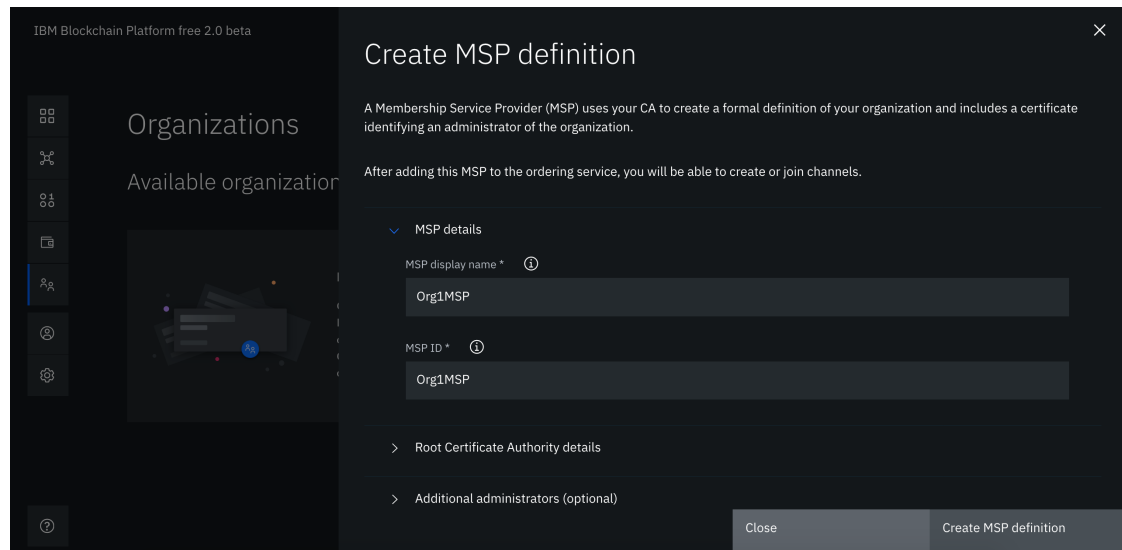
1. Navigate to the **Organizations** tab in the left side panel and click **Create MSP definition** as shown:



2. Provide all of the required information:

- **MSP details** — **MSP display name** and **MSP ID**
- **Root Certificate Authority details** — choose **Root Certificate Authority** (the root CA for the organization that you created in step 4) and provide an identity

name.

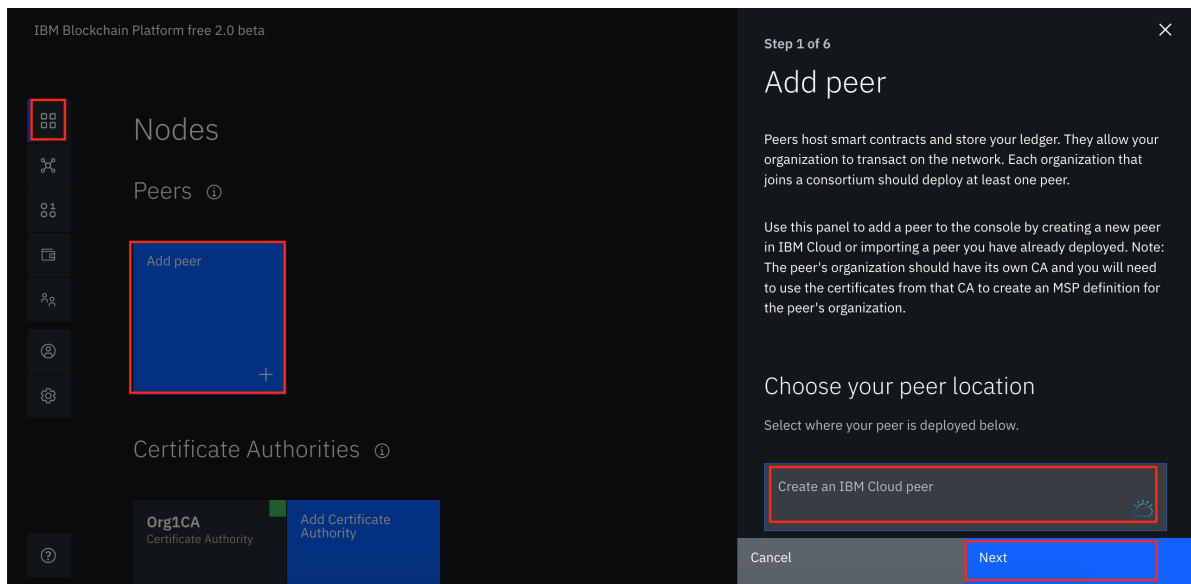


3. Click **Generate**. It will generate the identity as the admin for your organization and will be added to your Wallet.
4. To avoid losing these public and private keys, you must export them now and store them in a safe place. Finally, click **Create MSP definition**.

7 Create the peer node

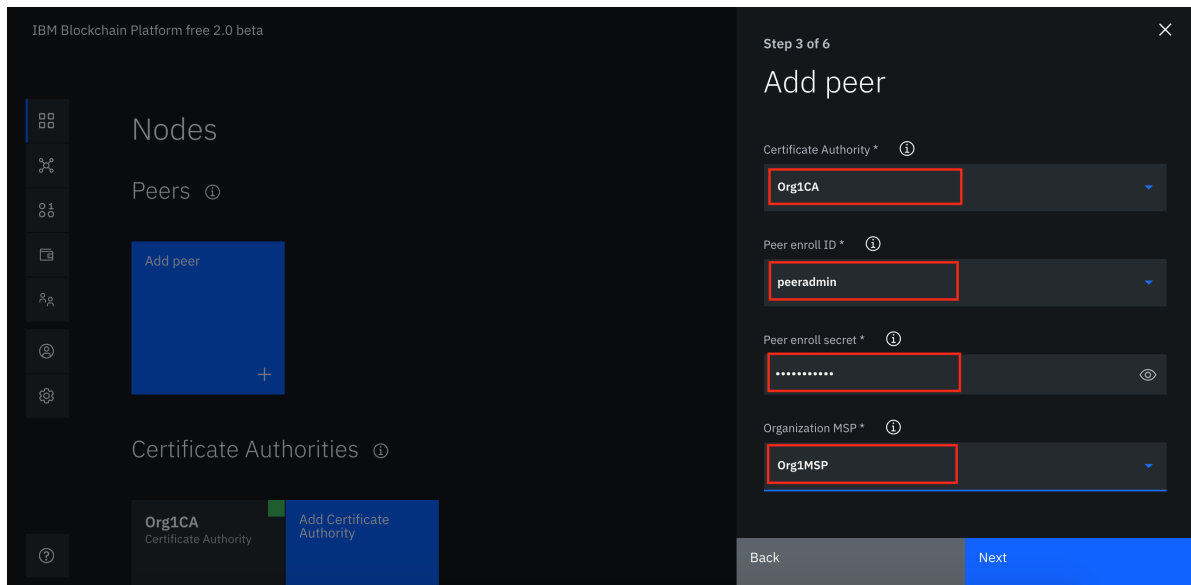
Peers are a fundamental element of the network that host ledgers and smart contracts.

1. Navigate to the **Nodes** tab, click **Add peer**, choose **IBM Cloud** as your peer location, and then click **Next**.



2. Provide a peer display name and click **Next**.

3. On the next screen, select the CA that you created in step 4. Select the enroll ID of the peer identity that you created for your peer at the end of step 5 and provide the associated secret. Then, select the **Organization MSP** from the drop-down list, and click **Next**.



4. Next it asks for Transport Layer Security (TLS) CA information. When you created the CA, a TLS CA was created alongside it. This CA is used to create certificates for the secure communication layer for nodes. Select the **TLS CA enroll ID** for the peer identity that you created for your peer (same as in the previous step) and provide its secret. The TLS Certificate Signing Request (CSR) hostname is an option that

enables you to specify a custom domain. You can leave the TLS CSR hostname blank. Click **Next**.

5. On the next screen, you will be asked to associate an identity. Choose the existing identity and select the display name of identity that you provided in step 6, then click **Next**.
6. Review the summary and click **Submit**. To add more peers in the same organization, *register a new user with peer identity as mentioned in step 5 and repeat this step*.

Note: Repeat steps 4 – 7 to create more organizations and peers as needed.

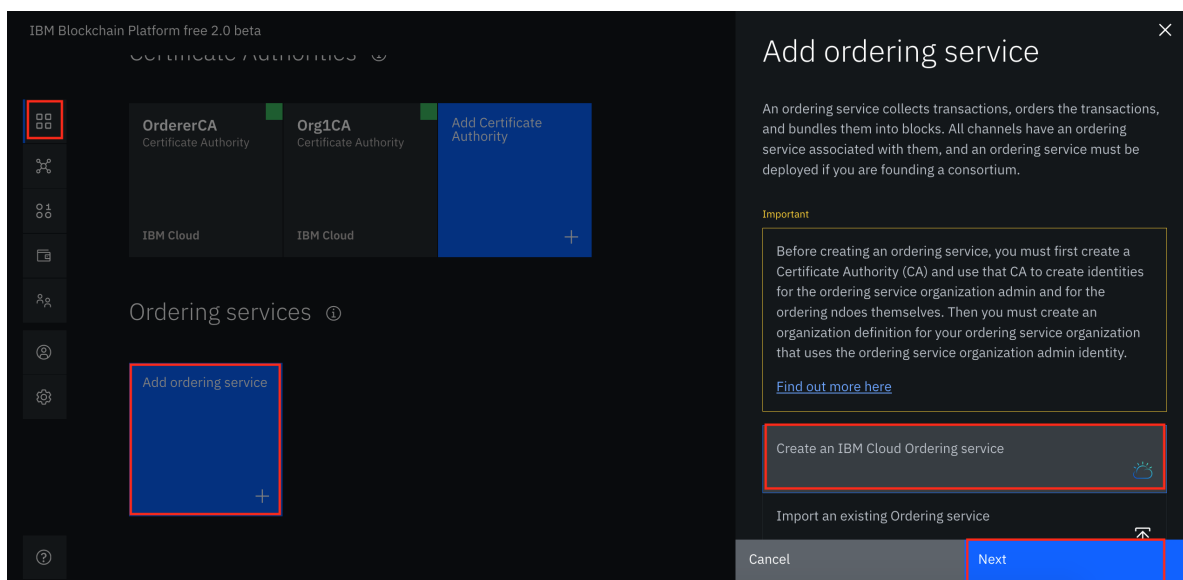
8 Create the orderer

The orderer node runs the communication service that guarantees transaction delivery in the network.

1. To create the orderer, you need to create the orderer CA first. Perform the same steps that you did in step 4 to create the orderer CA.
2. Next, register the orderer user identities using your orderer CA. Perform all of the steps in step 5 again to register users with “client” and “peer” identities.
3. Create the orderer MSP definition in the same way that you did in step 6. Make sure that you select the orderer CA for the Root Certificate Authority.
4. Now it's time to create the orderer node. Navigate to the **Nodes** tab and click **Add orderer**.

Provide the **Ordering service display name** and you can designate the **Number of ordering nodes** as one ordering node. (One ordering node is suitable for

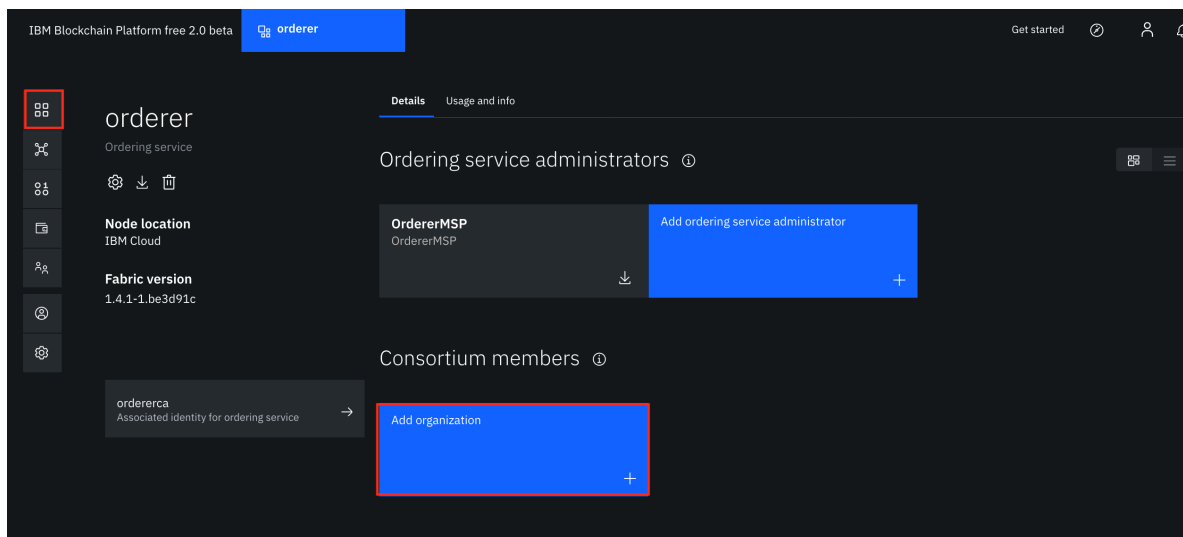
development and testing.) Then click **Next**.



5. Provide the required information as explained in step 7, but make sure you choose the correct orderer CA and orderer MSP.
6. Review the summary and then click **Add ordering service**.

9 Add an organization as a consortium member on the orderer

1. Navigate to the **Nodes** tab, and click on the orderer that you created in step 8.
2. Under **Consortium members**, click **Add organization**.

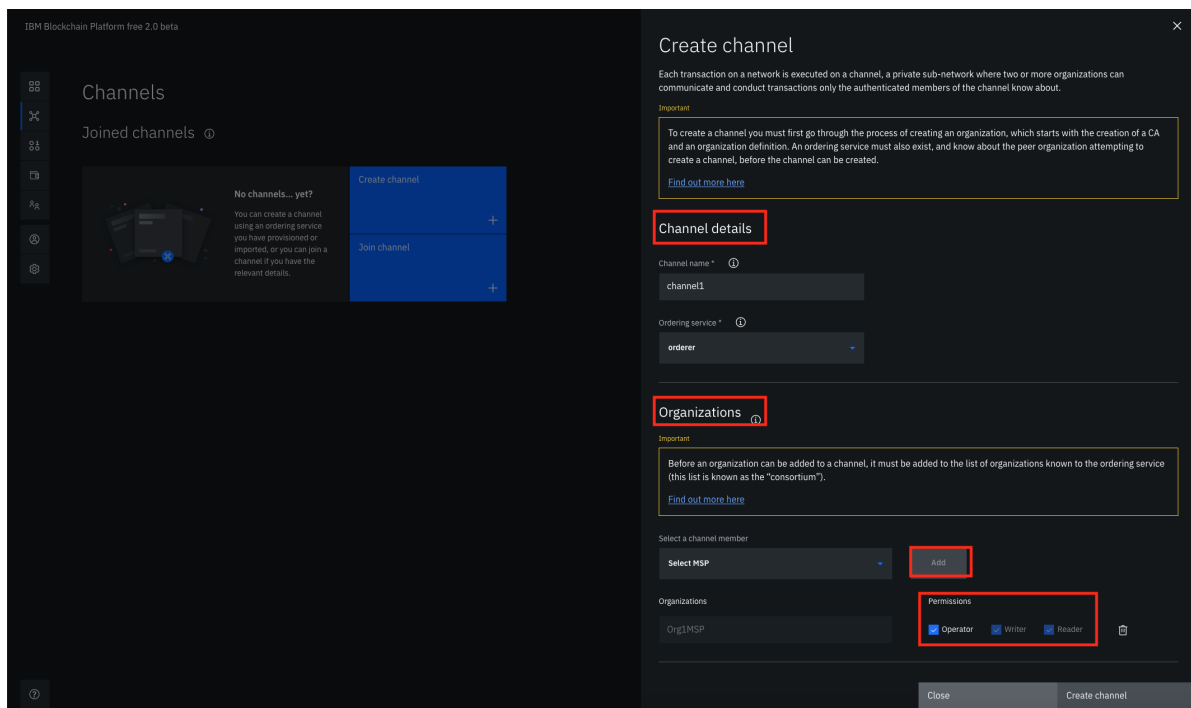


3. Choose **Existing MSP ID** and from the drop-down list, select the organization's MSP that you created in step 6 (the MSP represents the peer's organization), then click **Add organization**.
4. Repeat this step for all of the organizations that will be part of this consortium.

10 Create the channel

The channel is the mechanism through which a set of components within a blockchain network communicate and transact.

1. Navigate to the **Channels** tab in the left navigation and click **Create channel**.
2. In **Channel details**, provide a name for the channel. Select the orderer you created from the orderers drop-down list.
3. Now add organizations to the channel. Select the organizations (organization's MSP) as the channel member that will be part of the channel (one at a time) and click **Add**. Choose the appropriate permissions (operator/writer/reader) for the organization. Repeat this step to add all of the required organizations to your channel.



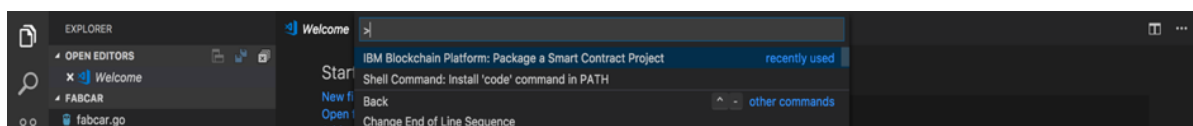
4. Choose the appropriate channel update policy from the available options.
5. The access control list is an advanced option. You can ignore it for now.
6. Next is the channel creator organization. Select the **Channel Creator MSP**, identifying an organization for channel creation from the drop-down list. Then associate an available admin identity for that organization.
7. When you've completed all of the above steps, click **Create channel**.

11 Join the channel

1. Navigate to the **Channels** tab in the left navigation and click **Join Channel**.
2. Select your orderer and click **Next**.
3. Provide the name of the channel you created in step 10 and click **Next**.
4. Select the peers you want to join the channel and click **Join Channel**.

12 Package the smart contract

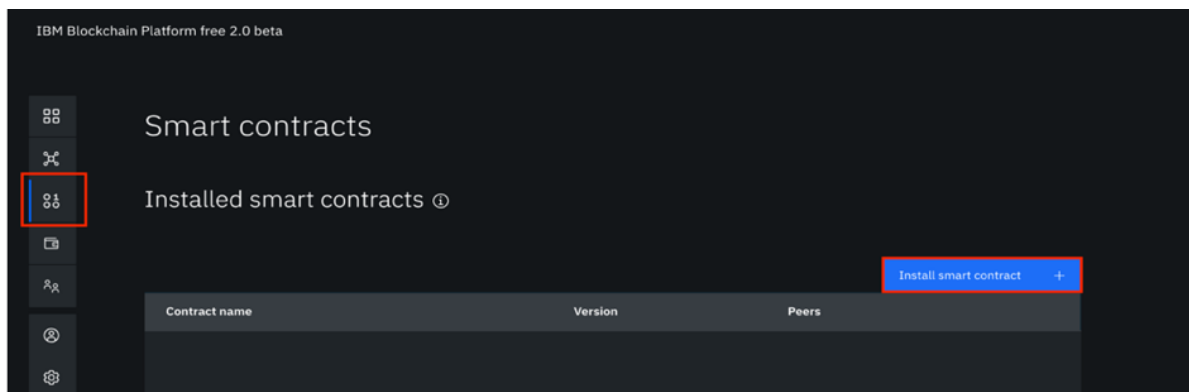
1. If the smart contract is written in Golang, then you should place the smart contract as `$HOME/go/src/<chaincode_folder>/<chaincode_file>.go`. Set `GOPATH` as `$HOME/go`, `GOROOT` as `<go install directory>`, and append `PATH` with `$GOROOT/bin`.
2. Open the folder in VS Code, which should have *only* the smart contract written in Go/Node.js. If you have any files in this folder other than smart contract files, then you may encounter issues while packaging the smart contract.
3. Press the **F1** key to see the various VS Code options. Choose **IBM Blockchain Platform: Package a Smart Contract Package**. Make sure Node, NPM, and Go are installed and the corresponding paths are set before you perform this step.



4. You will be prompted for the package name and version. It's important to provide the proper version as the smart contract can then be upgraded to deploy a newer version. After you've provided this information, the smart contract package will be created.
5. In the left navigation panel, click on **IBM Blockchain Platform** under smart contract packages. You should be able to see a .cds file named <package name>@<version>.
6. Right click on the package and click **Export Package**, then save the file.

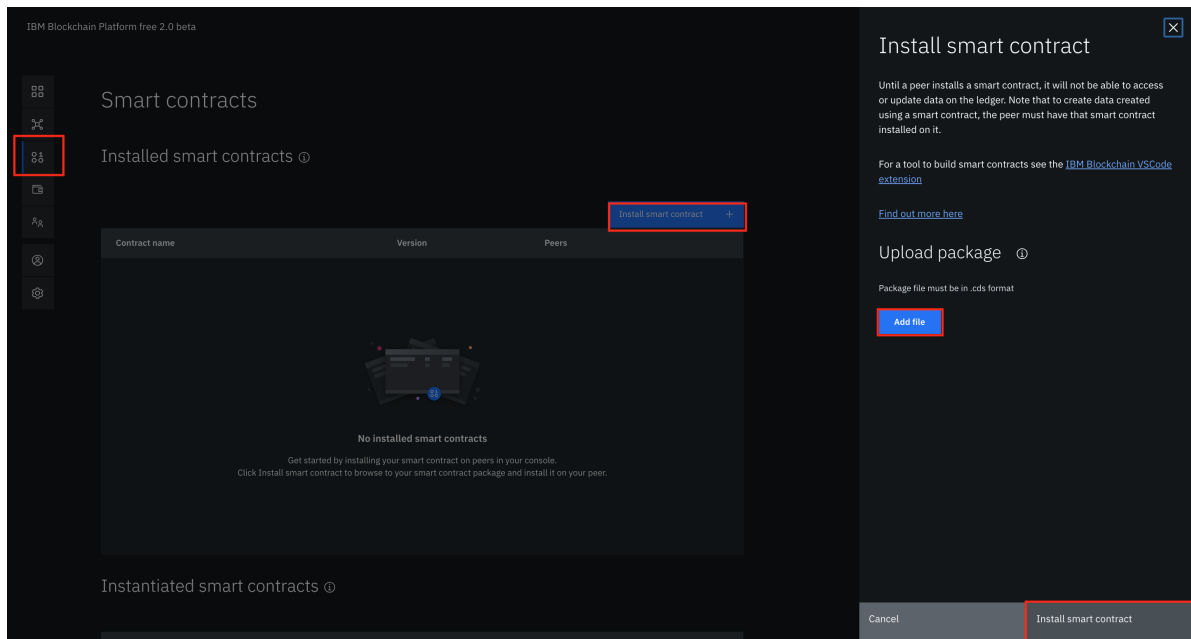
13 Install the smart contract

1. Navigate to the **Smart contracts** tab and click **Install smart contract**.



2. In the **Upload package** section, click **Add file** and provide your smart contract package file (.cds) to upload, which you packaged in step 12 using the VS Code

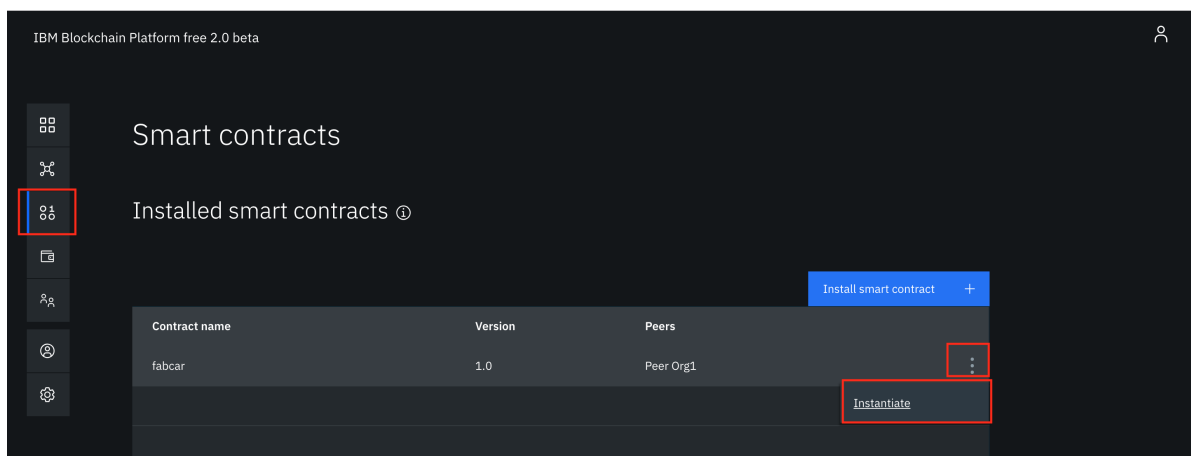
extension.



3. Once the smart contract is uploaded, click **Install smart contract**.

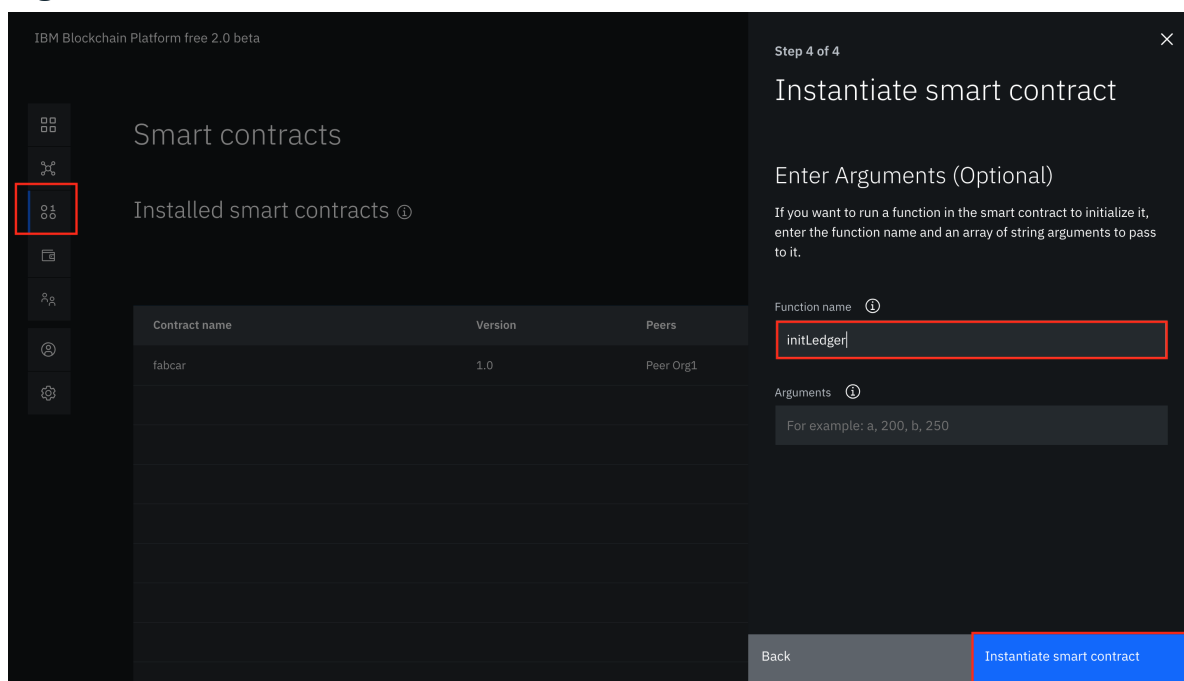
14 Instantiate the smart contract

1. Navigate to the **Smart contracts** tab. You should see a list of smart contracts that have already been installed. Find the smart contract from the list installed on your peers and select **Instantiate** from the overflow menu on the right side of the row.



2. On the side panel, select the channel to instantiate the smart contract on, then click **Next**.

3. Next, you need to set up the endorsement policy. Select the organization members to be included in the endorsement policy and click **Next**.
4. At this point, you can also set up the private data collection — but this step is optional, so we'll skip it this time. Click **Next**.
5. Provide the function name to initialize the smart contract with the required arguments and click **Instantiate smart contract**.



6. Once instantiated, it will be listed as one of the instantiated smart contracts under the **Smart contracts** tab.

Summary

This tutorial has showed you the steps to deploy your blockchain network using the IBM Blockchain Platform console. The IBM Blockchain Platform is highly customizable — you can even test, debug, and upgrade an instantiated smart contract in your network. Once the network is deployed successfully and a smart contract has been instantiated, you can then start writing your client applications to interact with the blockchain network using any Fabric SDK in the middle layer. To learn more about this, explore the IBM Developer Blockchain [code patterns](#).

One final note: The components in your network can be added depending on the availability of resources in your cluster. If you need to add more components (such as peers or orgs), you may need to scale the size of your Kubernetes cluster accordingly.

COMPONENTS HYPERLEDGER HYPERLEDGER FABRIC

SOCIAL



CONTENTS

Prerequisites

Estimated time

Steps

- 1 Create the IBM Cloud Kubernetes Service
- 2 Create the IBM Blockchain Platform service instance on the IBM Cloud
- 3 Launch the platform
- 4 Add the certificate authority (CA)
- 5 Register your users
- 6 Create the organization's Membership Service Provider (MSP) definition
- 7 Create the peer node
- 8 Create the orderer
- 9 Add an organization as a consortium member on the orderer
- 10 Create the channel
- 11 Join the channel
- 12 Package the smart contract
- 13 Install the smart contract

14 Instantiate the smart contract

Summary

RESOURCES

[IBM Blockchain Platform](#)[Hyperledger Fabric docs](#)[IBM Cloud docs: Getting started with IBM Blockchain Platform](#)[IBM Cloud docs: Build a network tutorial](#)

Related content

TUTORIAL | MAY 30, 2019

IBM Blockchain 101: Quick-start guide for developers

Build a kick-starter blockchain network and start coding with the next generation Blockchain Platform from IBM.

[Blockchain](#) [Cloud](#) +**SERIES** | MAY 31, 2019

IBM Blockchain Platform Console Video Series

Follow along this four part series to set up a business network on the IBM Blockchain Platform

[Blockchain](#)[Hyperledger Fabric](#)**SERIES** | APR 16, 2019

Build your first blockchain application

Get started with this secure, distributed, open ledger technology by completing a pattern series that shows you how to build...

[Blockchain](#)

IBM Developer

[About](#)[Site Feedback & FAQ](#)[Report abuse](#)[Third-party notice](#)

Follow us

[Code Patterns](#)[Articles](#)[Tutorials](#)

Select a language

[English](#)[中文](#)[日本語](#)[Русский](#)[Português](#)[Español](#)[한글](#)[Videos](#)[Newsletters](#)[Events](#)

[Recipes](#)

[Cities](#)

[Open Source Projects](#)

[Answers](#)

[Community](#)

[Privacy](#)

[Terms of use](#)

[Accessibility](#)

[Cookie Preferences](#)