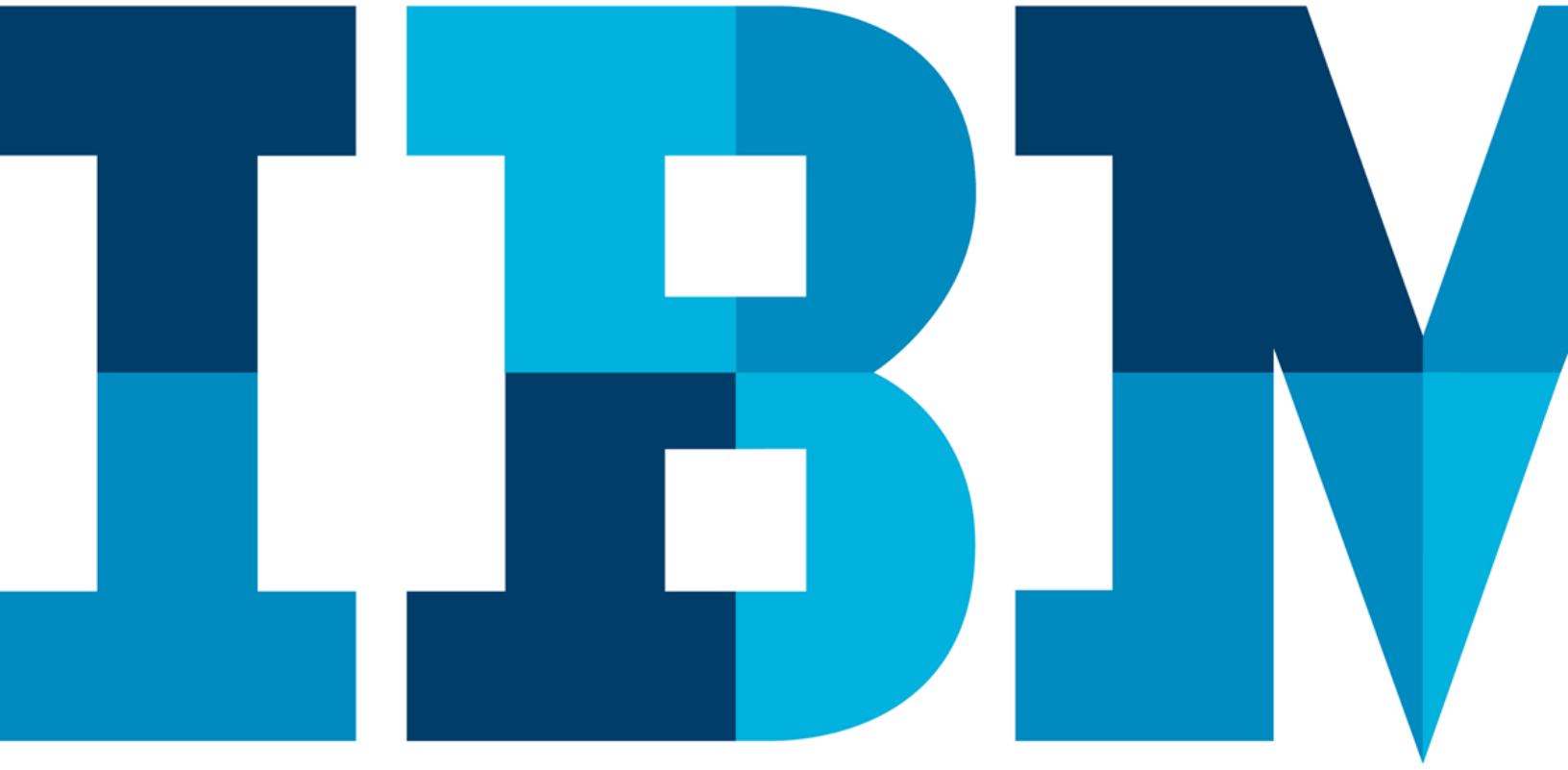
IBM Blockchain Platform Hands-On

Lab 5:

IBM Blockchain Platform for IBM Cloud Operations Lab



IBM

Table of Contents

Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it Each organization not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results like those stated here.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and

discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

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 about this publication 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. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

1 Overview of the lab environment and scenario

This lab is a guide to using the IBM Blockchain Platform operations console. In this lab, you will work with a partner to **Build** a new two organization network using the console and get your partner to **Join** it. You will then deploy an existing chaincode and issue a transaction between the two organizations and see the results.

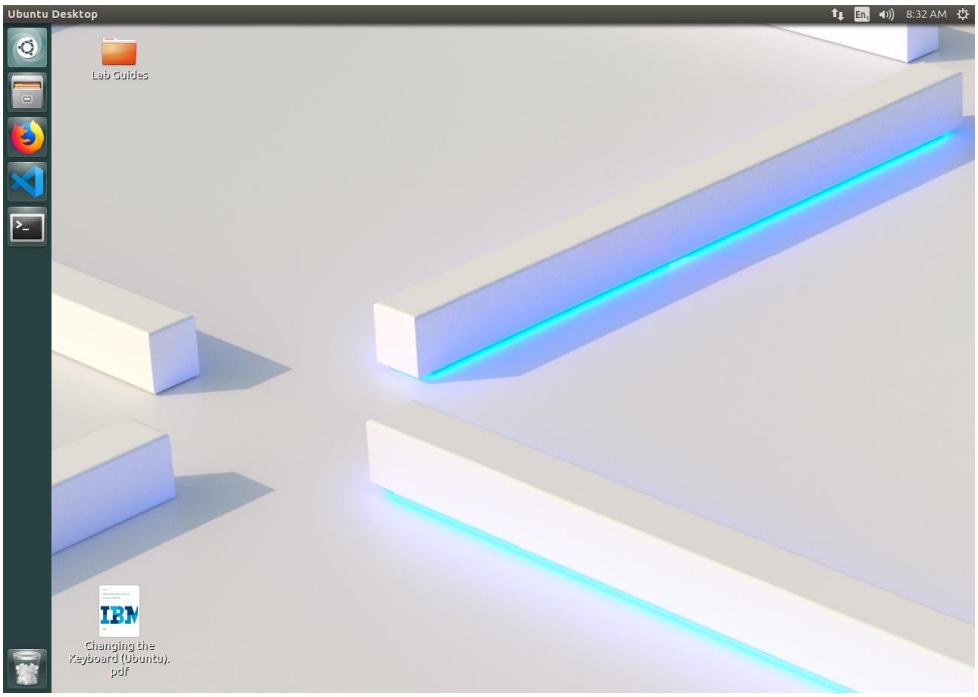
***Note**: The screenshots in this lab guide were taken using version **1.37.1** of **VS Code**, version **1.0.9** of the **IBM Blockchain Platform** plugin and version **0.3.50** of the **IBM Blockchain Platform** console. If you use different versions, you may see differences to those shown in this guide.*

Start here. Instructions are always shown on numbered lines like this one:

__ **1.** If it is not already running, start the virtual machine for the lab. The instructor will tell you how to do this if you are unsure.

__ **2.** Wait for the image to boot and for the associated services to start. This happens automatically but might take several minutes. The image is ready to use when the desktop is visible as per the screenshot below.

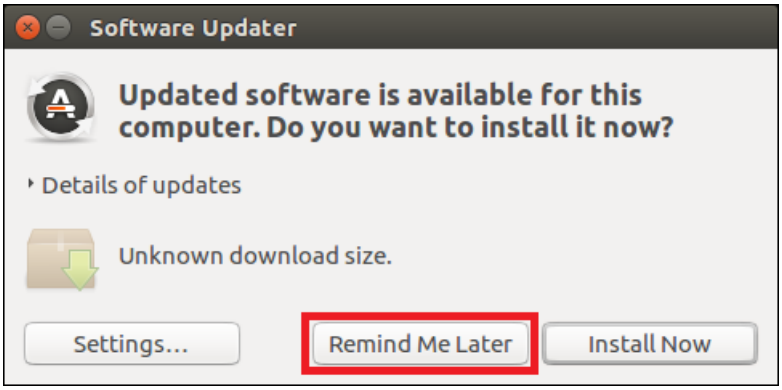
Note: If it asks you to login, the userid and password are both "**blockchain**".



1.1 Lab Scenario

In this lab, we will be creating a new network on the IBM Blockchain Platform running on IBM Cloud. This lab assumes that you already have an IBM Cloud account to use and have set-up a Kubernetes free tier cluster in it using the default name of "**mycluster**".

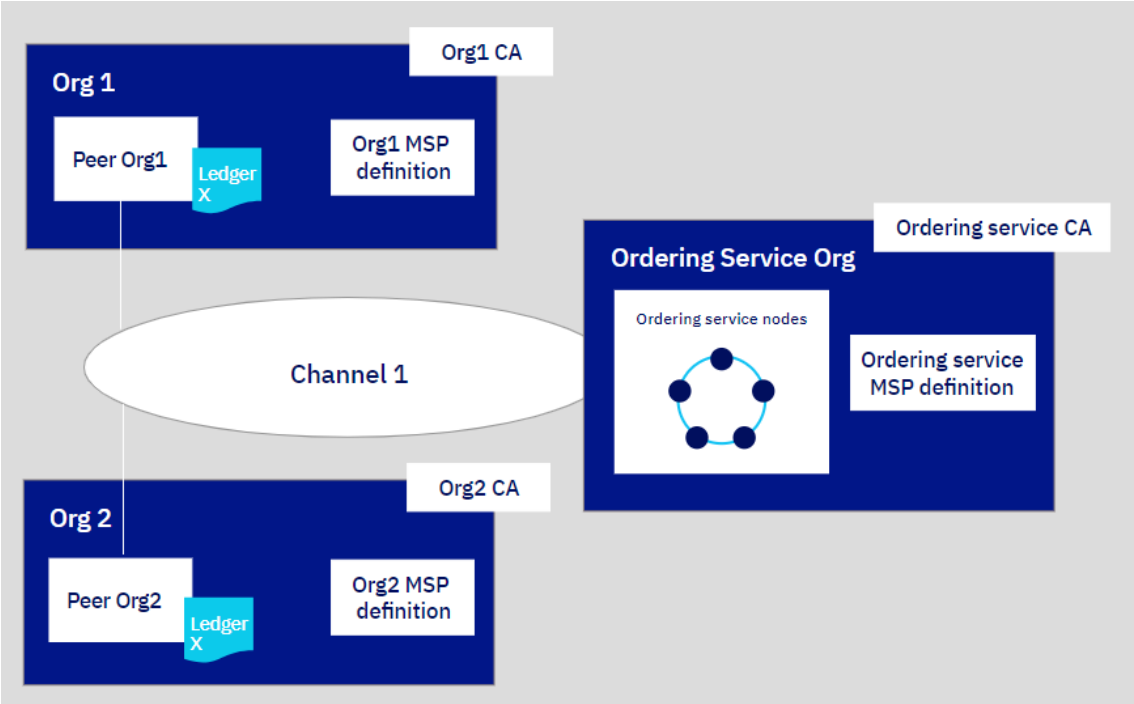
***Note** that if you get a "**Software Updater**" pop-up at any point during the lab, please click "**Remind Me Later**":*



2 IBM Blockchain Platform Operations

As mentioned above, in this lab we will be building a new network containing two new organizations using the IBM Blockchain platform **Build** and **Join** tutorials which we will look at shortly.

The network we will be building looks like this:



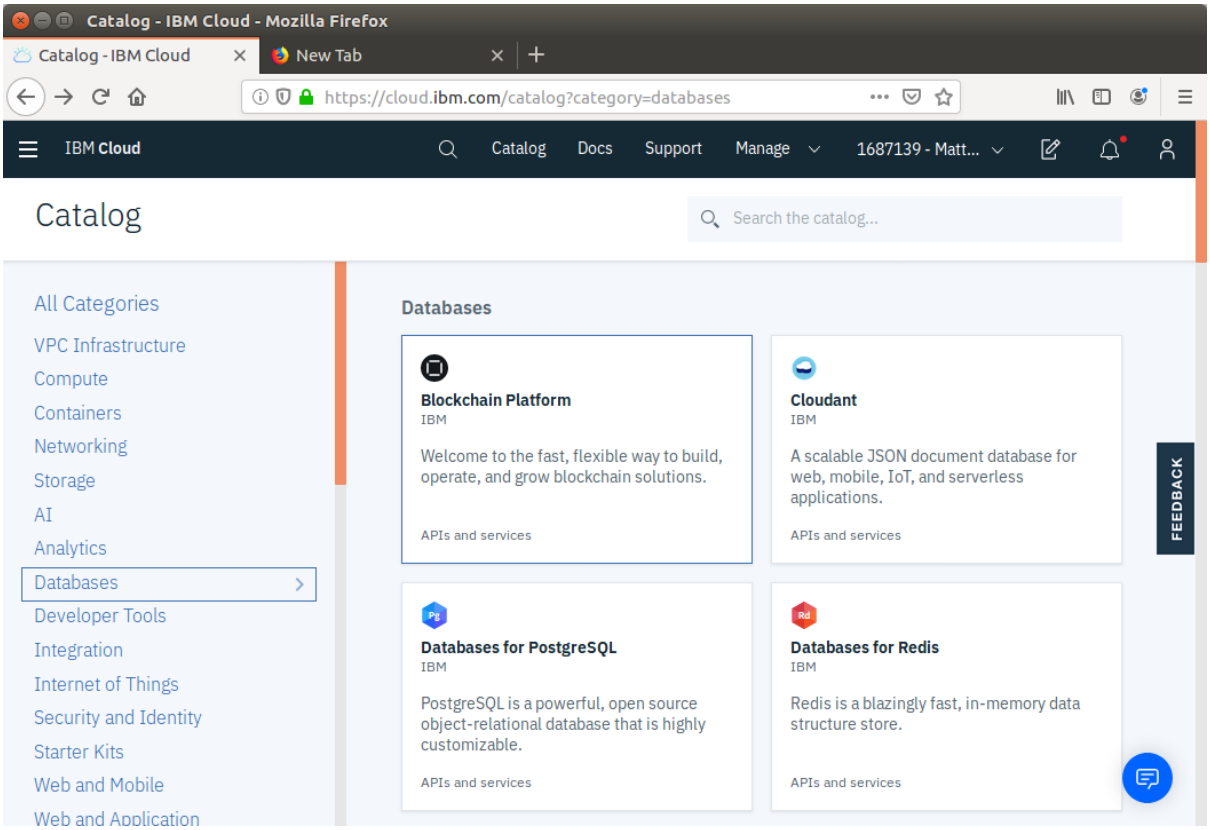
Although you can build this network on your own, we recommend you build it in pairs with one person taking on the role of Org1 and the Ordering Service Org and the second person taking on the role of Org2.

So in pairs, take a little time to decide who is going to represent **Org1** and who is going to represent **Org2**. Both orgs will have a lot to do, but we will start out with representatives from both orgs provisioning their IBM Blockchain Platform service.

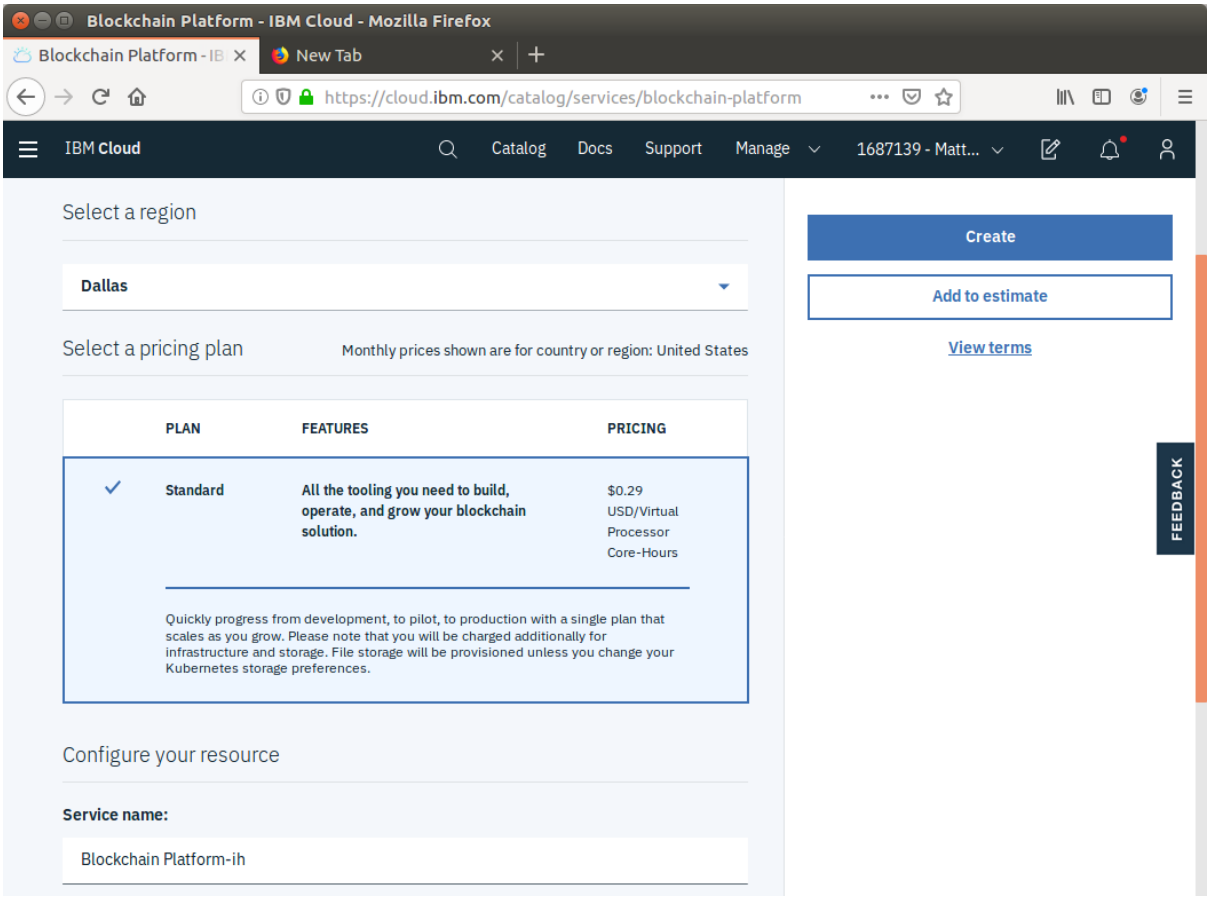
2.1 Setting up the IBM Blockchain Platform.

This section must be completed first by BOTH Org1 and Org2

__ **3.** Open the Firefox web browser inside the VM and go to https://cloud.ibm.com/catalog. Make sure you are logged in, then click on "**Databases**" to see the **Blockchain Platform** tile:

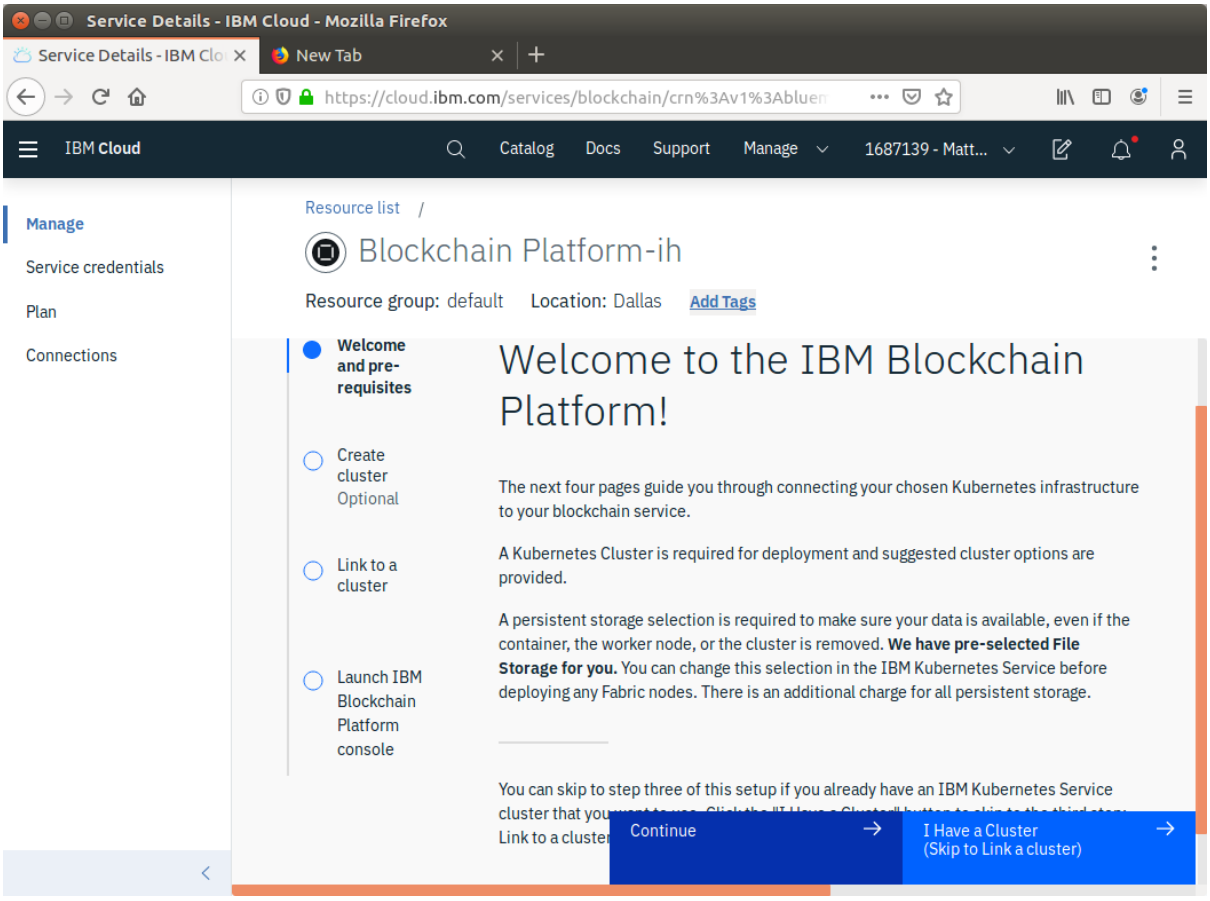


__ **4.** Click on the "**IBM Blockchain Platform**" title to see the provisioning page:

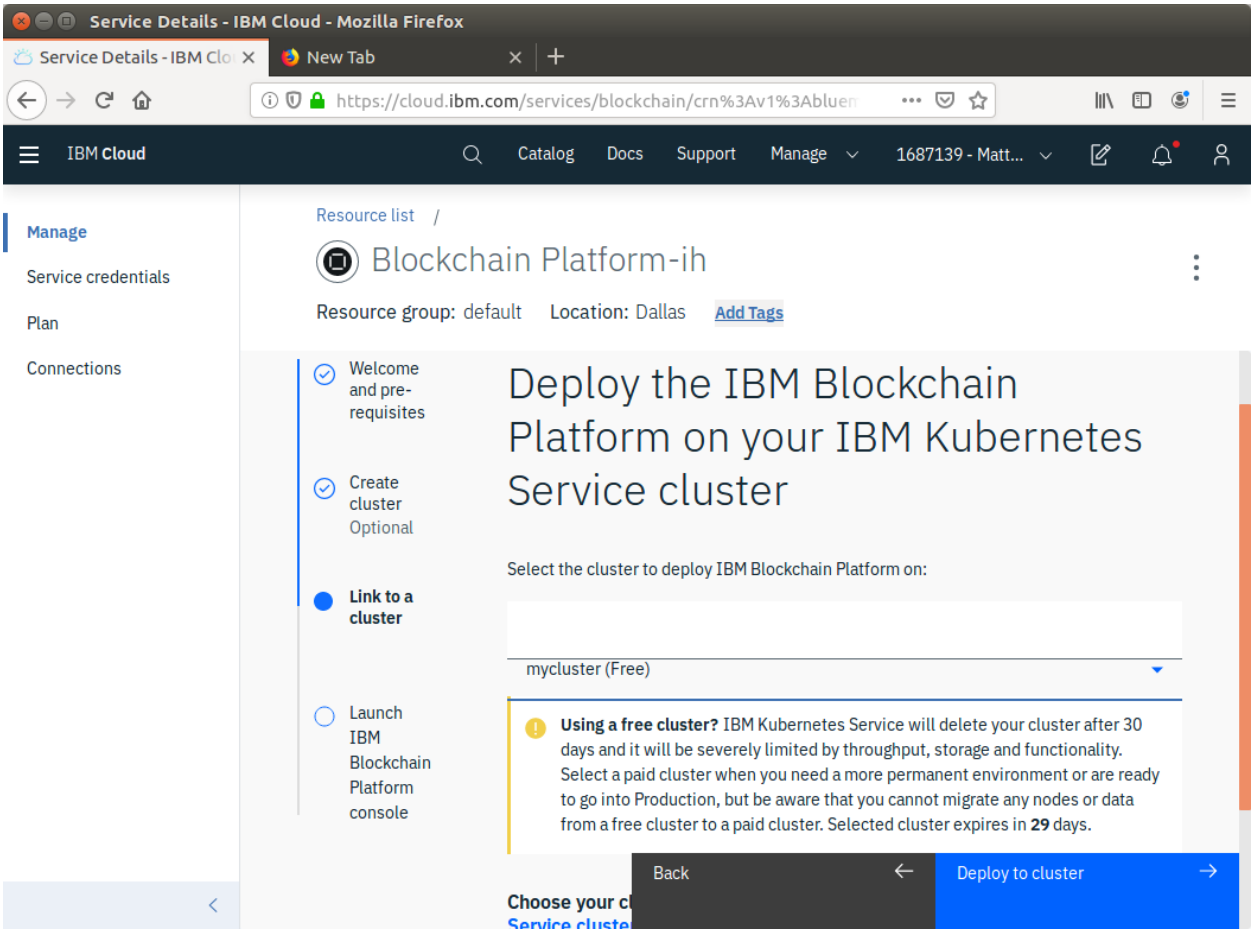


Each organization will have a different service name. In this screenshot, this is the generated name "**Blockchain Platform-ih**" but yours will be different. Make a note or remember your Service name for the following steps. You can leave the region settings and the resource group settings at the defaults.

__ **5.** Click on the "**Create**" button to move to the "**Welcome**" page shown below:

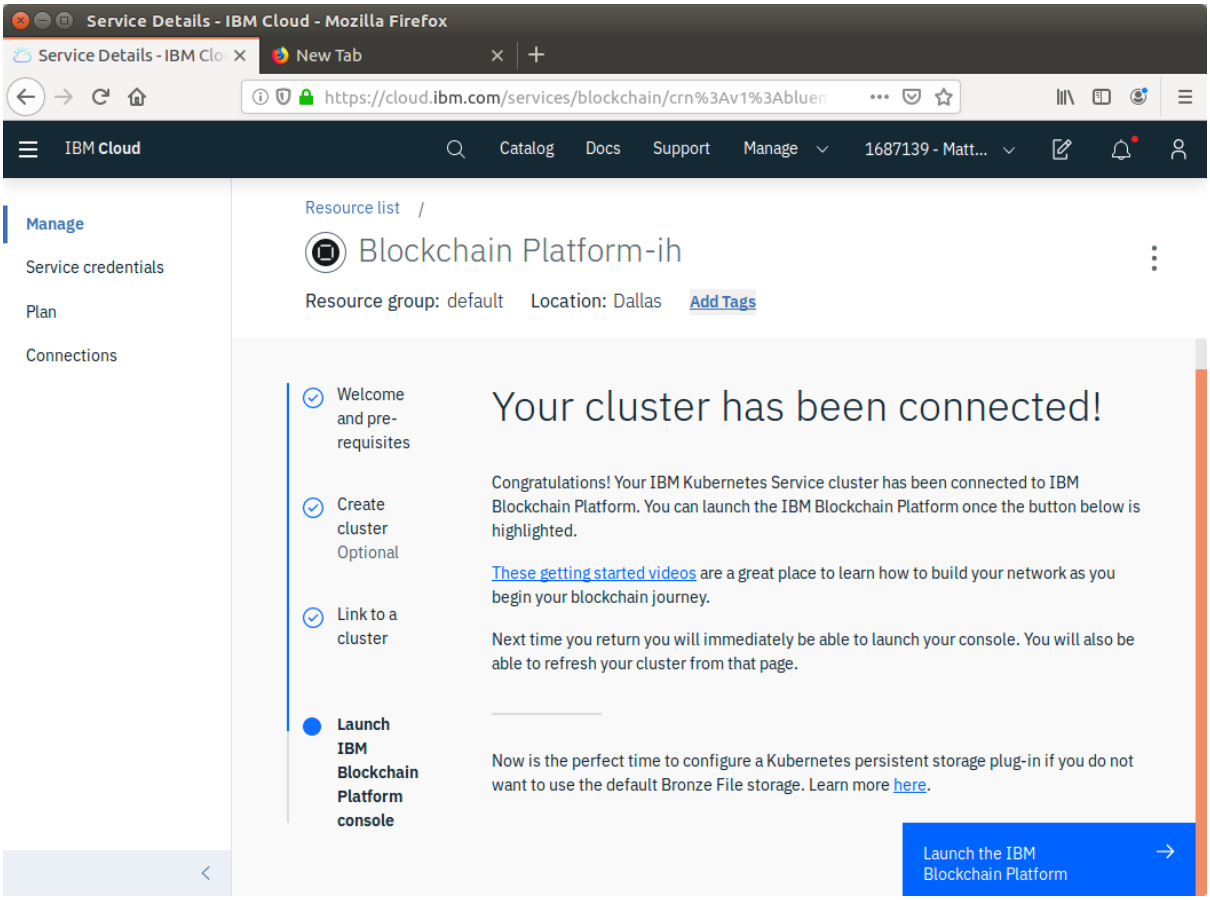


__ **6.** As you will already have a cluster defined, click on the "**I have a Cluster**" button to move to the "**Link to a Cluster**" page:

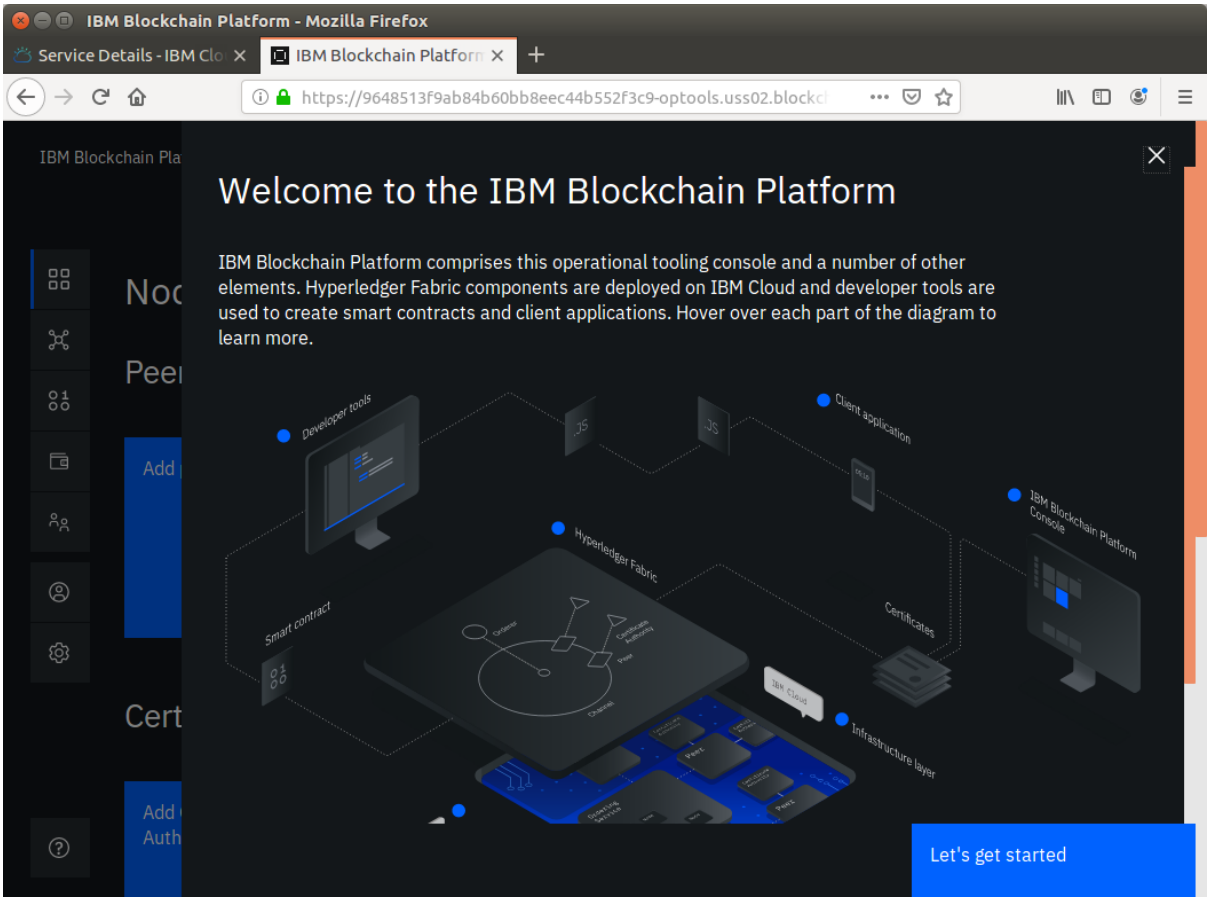


__ **7.** In the dropdown, choose your cluster which should be called "**myCluster (Free)**" and click on the "**Deploy to cluster**" button. This may take a few minutes to complete. When it is done, you should see the summary page shown in the next step.

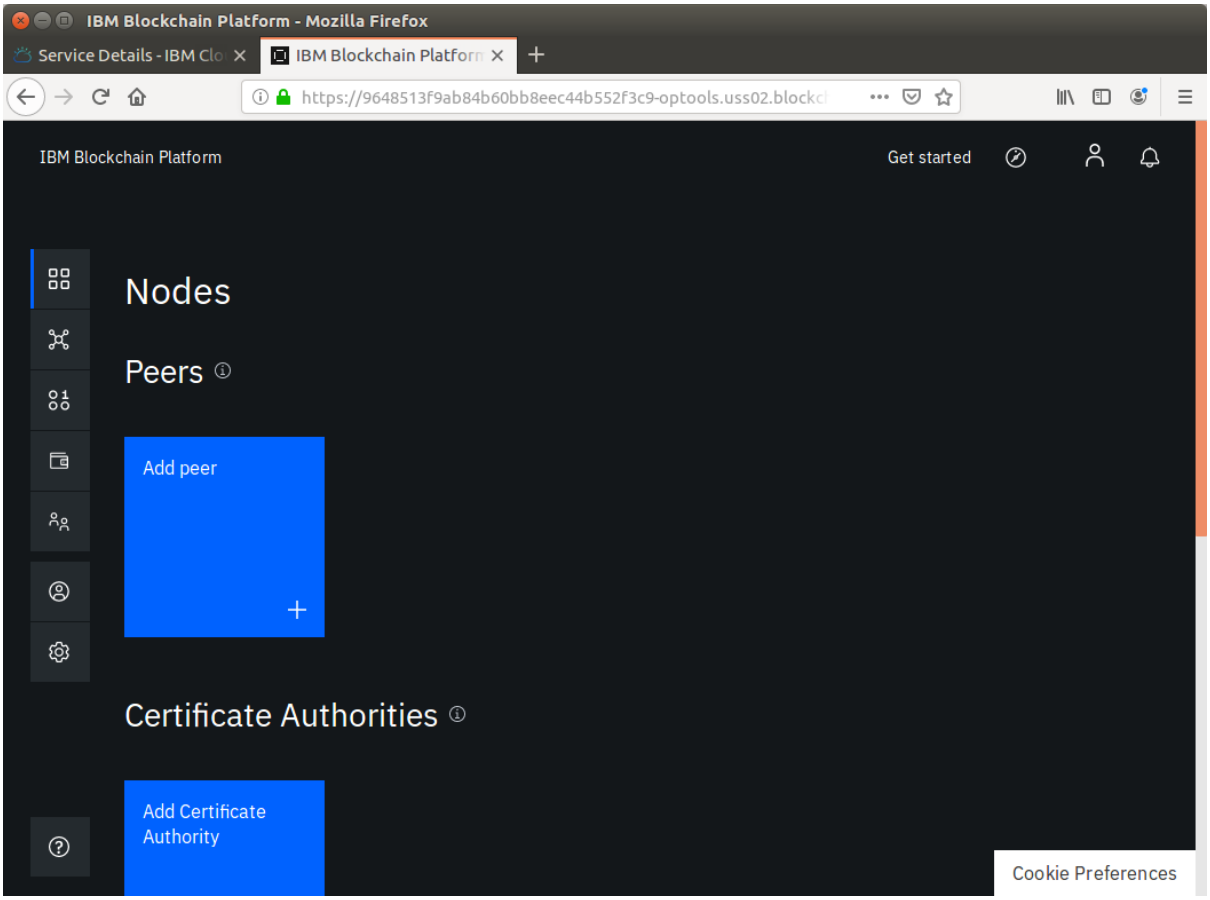
__ **8.** From the summary page, click the "**Launch the IBM Blockchain Platform**" button:



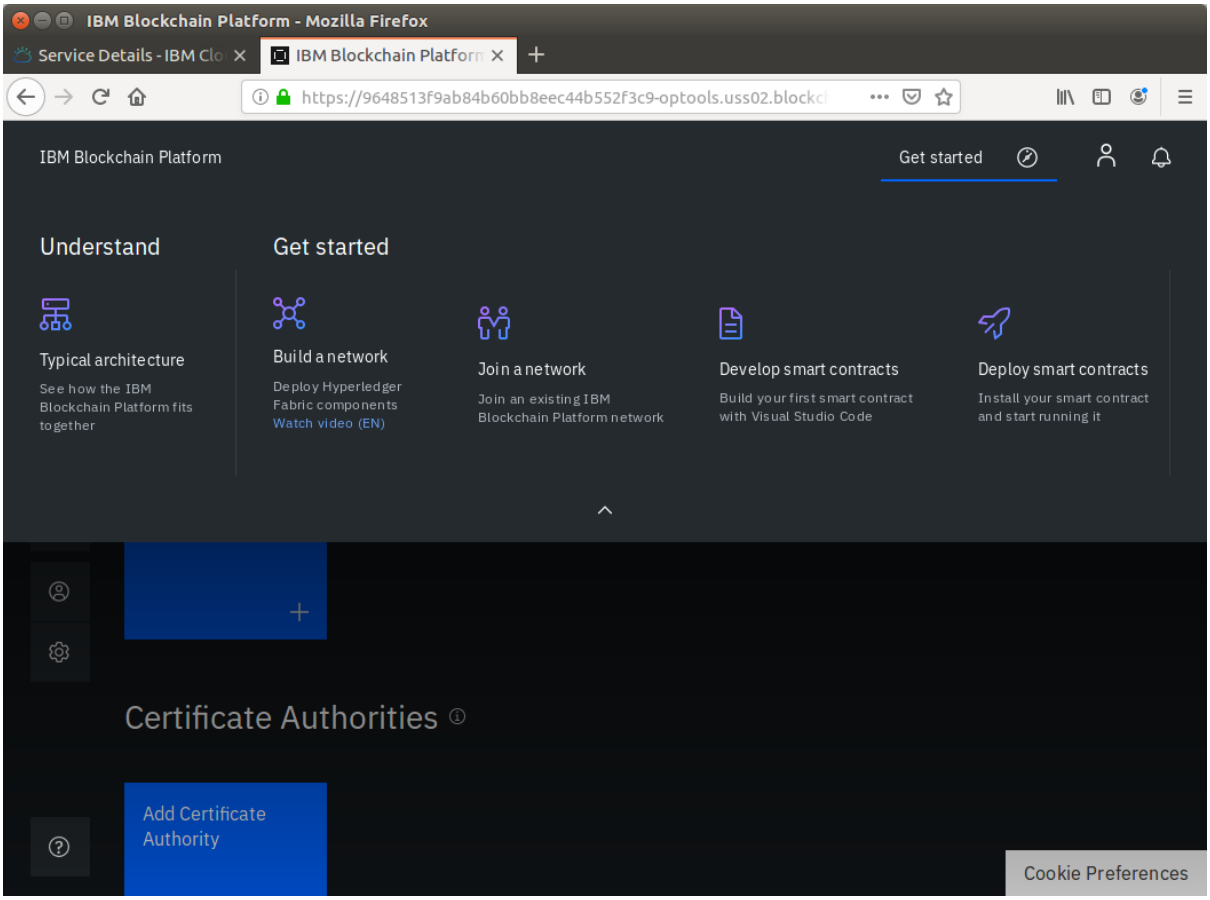
__ **9.** You will then see the **Welcome** page for the platform. Take a little time to move your mouse around the interactive diagram to see the different parts in a typical platform deployment:



__ **10.** When you are done, click the "**Let's get started**" button to close the diagram and move to the main console page:



__ **11.** From the main console page click the "**Get Started**" button at the top of the screen to bring up a list of links to the built-in tutorials:



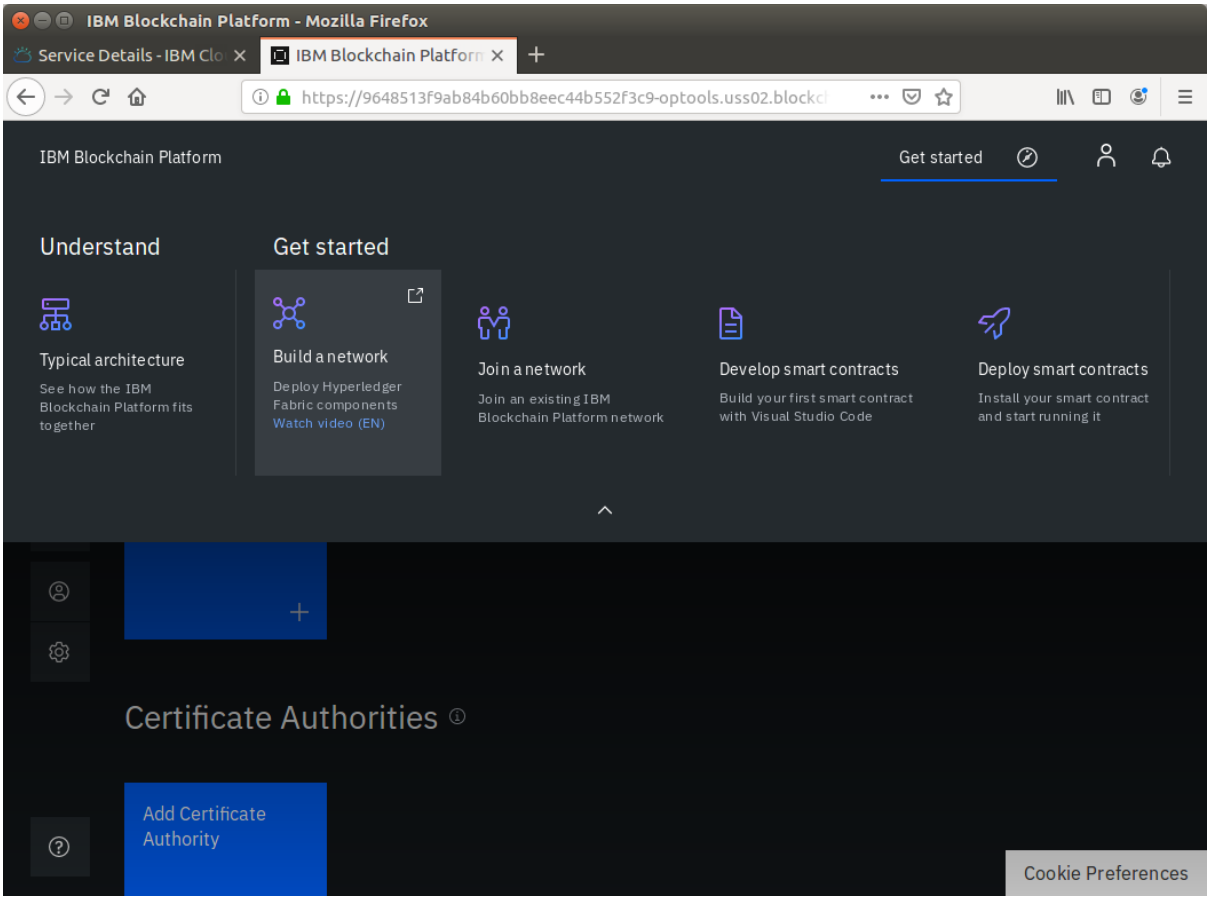
This concludes the setup part of this lab, where representatives from both organizations have deployed a new blockchain service into their respective Kubernetes clusters in the IBM Cloud accounts.

In the rest of this lab, we will work to create a network that spans both of these services and joins them together in a blockchain network.

2.2 Building the network

This section is for Org1 ONLY – Org2 should watch and help along.

__ **12.** As **Org1**, click on the "**Build a network**" link:



The **Build** tutorial will open in a separate tab. For reference the URL to this page is:

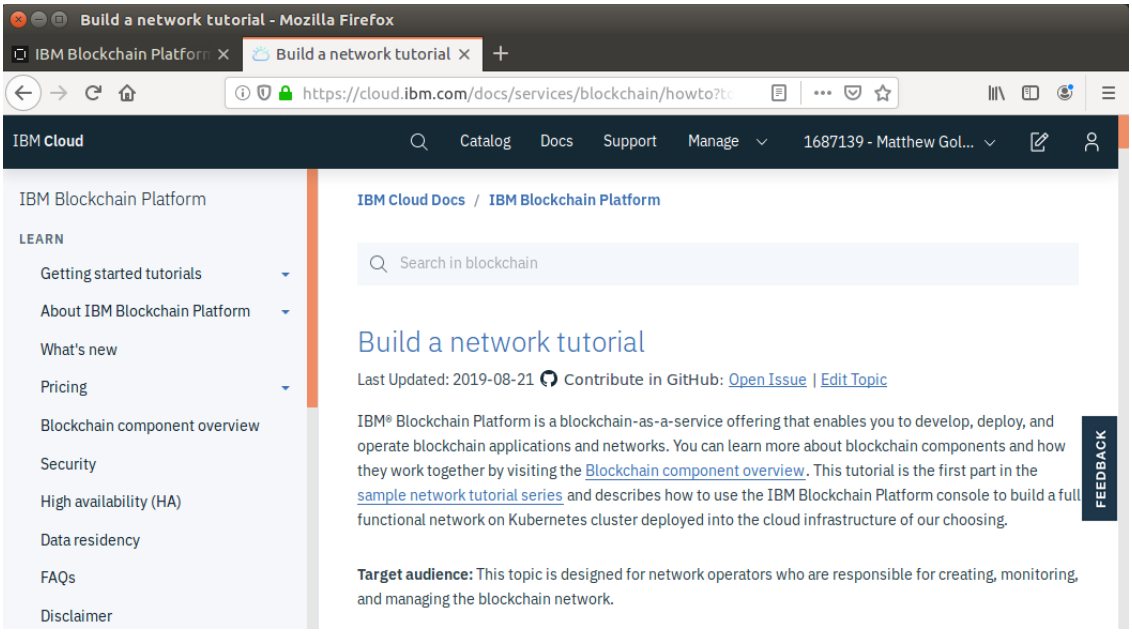
https://cloud.ibm.com/docs/services/blockchain/howto?topic=blockchain-ibp-console-build-network

The **Build** tutorial takes you through five steps:

Step one: Create a peer organization and a peer
Step two: Create the ordering service
Step three: Join the consortium hosted by the ordering service
Step four: Create a channel
Step five: Join your peer to the channel

__ **13.** Working as **Org1**, work through the **Build** tutorial, following the numbered steps in circles. Please use the recommended names shown in the tutorial.

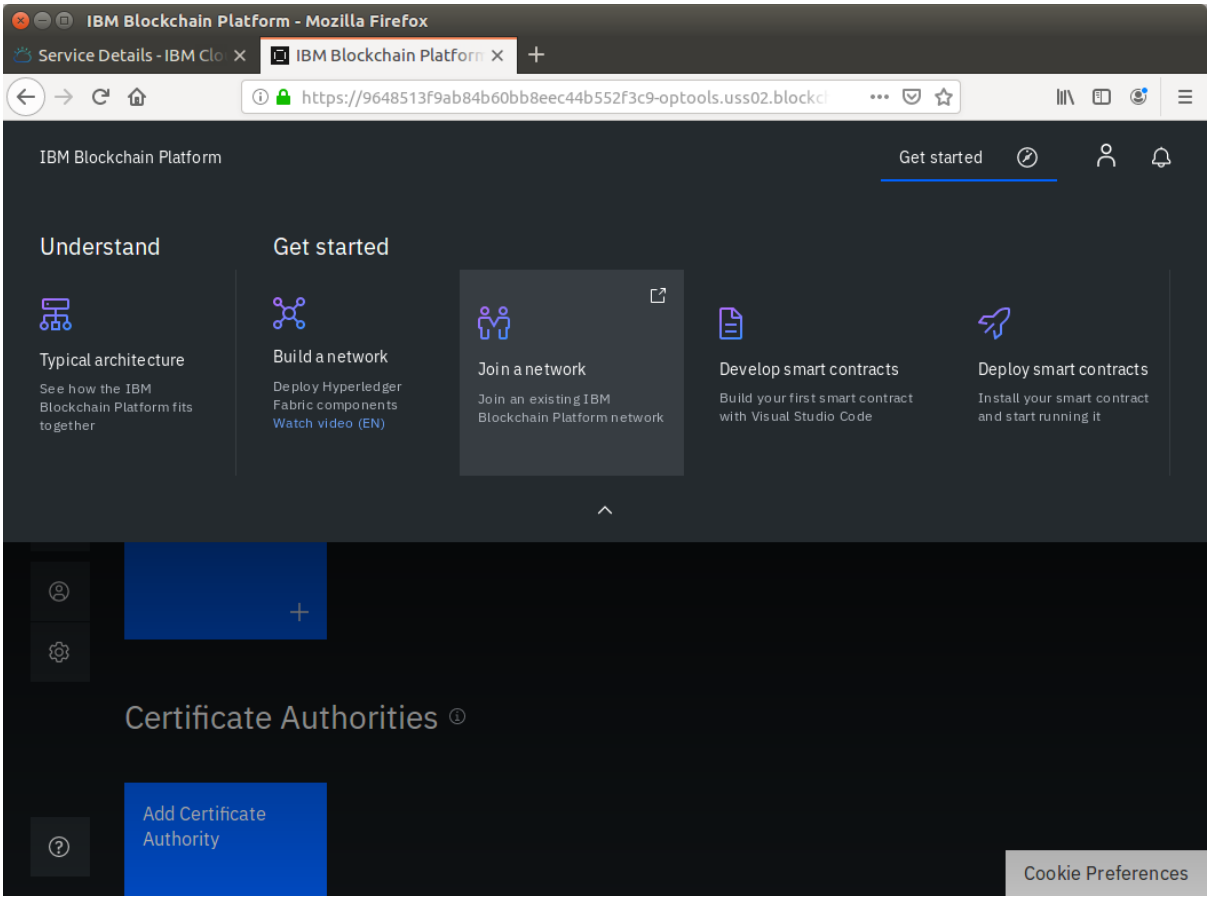
When you get to the **Next Steps** part, after **Step five**, stop following the tutorial and move on to the next section in this lab guide.



2.3 Joining the Network

This section is for Org2 ONLY except where indicated – Org1 should watch and help.

__ **14.** As **Org2**, click on the "**Join a network**" link:



The **Join** tutorial will open in a separate tab. For reference the URL to this page is:

`https://cloud.ibm.com/docs/services/blockchain/howto?topic=blockchain-ibp-console-join-network`

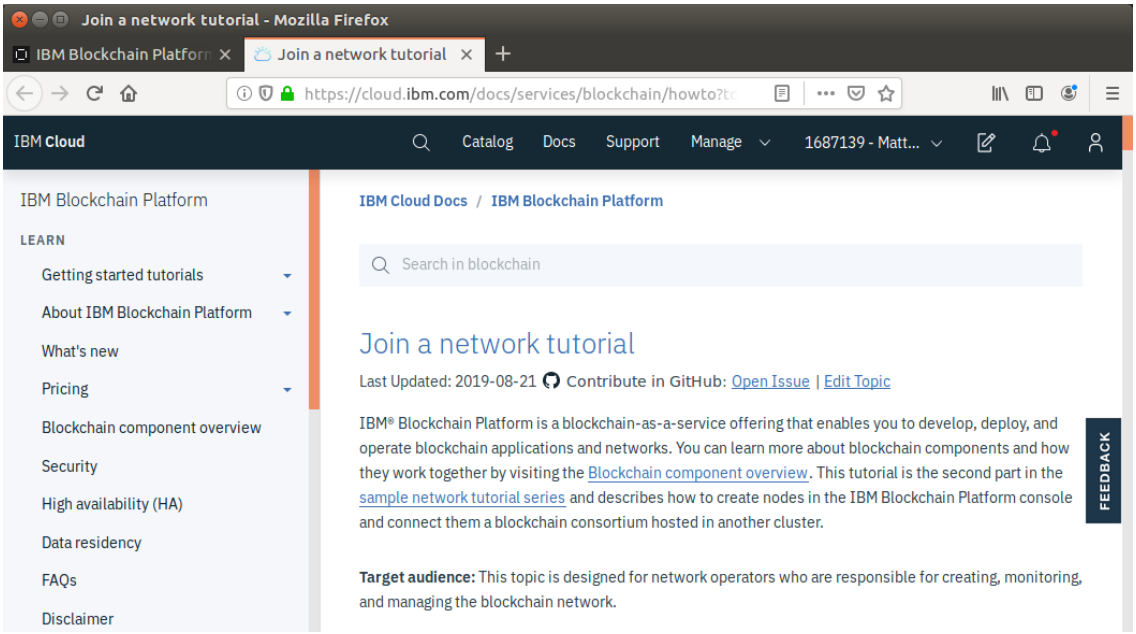
The **Join** tutorial takes you through four steps:

Step one: Create a peer organization and a peer
Step two: Join the consortium hosted by the ordering service
Step three: Add the peer's organization to an existing channel
Step Four: Join your peer to the channel

During a couple of the steps, you will need to exchange JSON information with **Org1** and vice versa. To do this you will need to copy the information into an email or use a similar mechanism to transfer this information to your partner in the other org.

__ **15.** Working as **Org2**, work through the **Join** tutorial, following the numbered steps in circles. However, there are some steps that need to be done by **Org1** so look out for these. Please use the recommended names shown in the tutorial.

When you get to the **Next Steps** part, after **Step four**, stop following the tutorial and move on to the next section in this lab guide. However, if you are running short of time you can stop and move to the next section when you get to the start of the optional "**Creating a Channel**" section.



2.4 Deploying into the network

This section must be completed by both organizations as indicated in each step

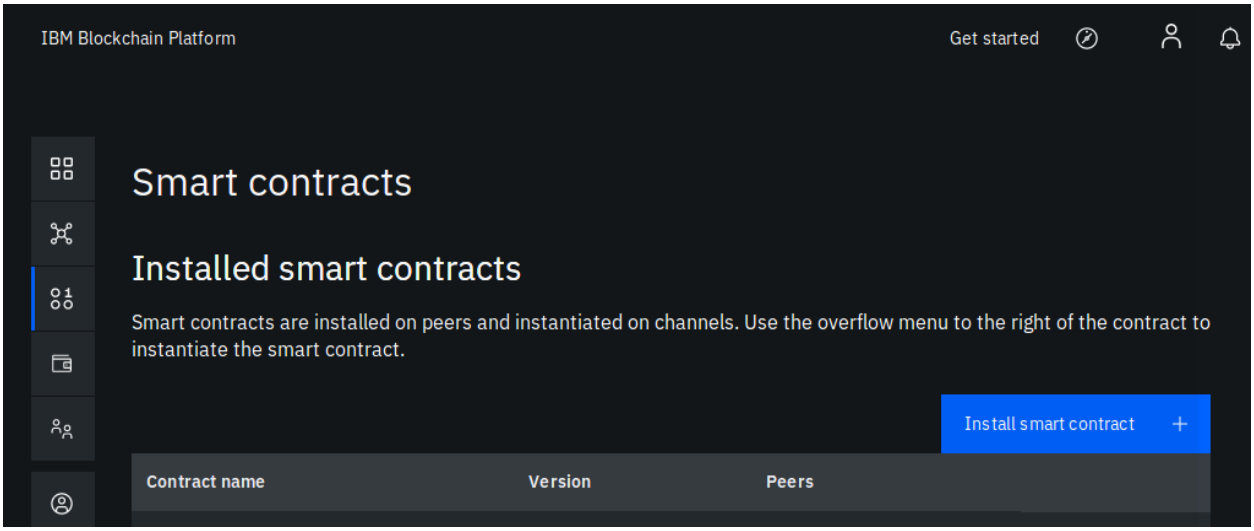
To test the network out we need to deploy a simple smart contract and issue some transactions against it. To do this we will use the **fabcar** smart contract again.

Because an identical contract needs to be deployed into both organizations, we are going to use an existing packaged version of the contract to save time.

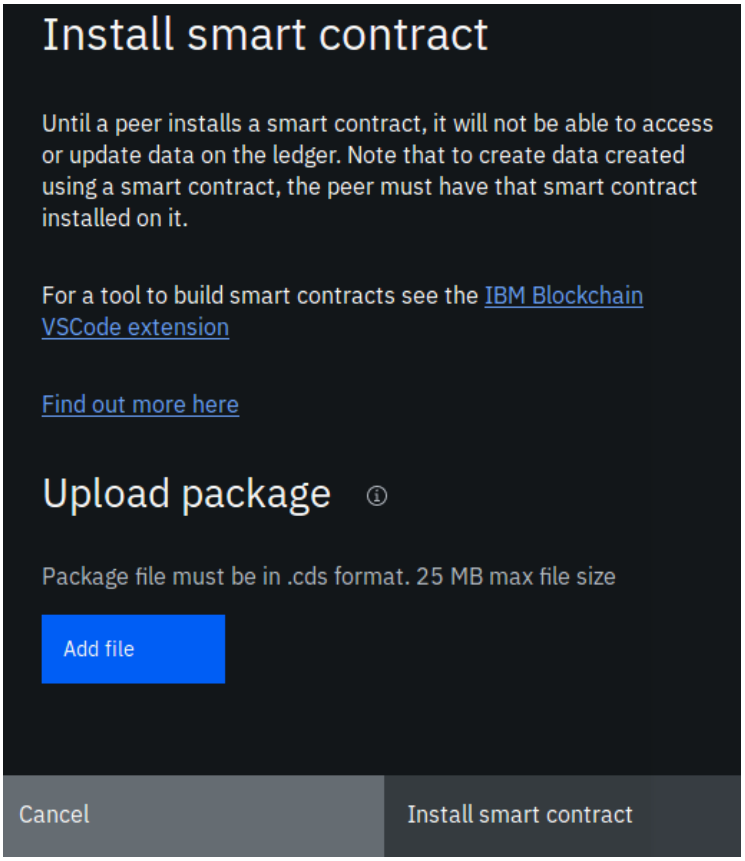
__ **16.** [**Org1 & Org2**] Copy and paste the URL below into Firefox to download the file "**fabcar@1.0.1.cds**" and save it in your **Downloads** folder:

```
https://github.com/m-g-k/ibp_workshop/raw/master/fabcar%401.0.1.cds
```

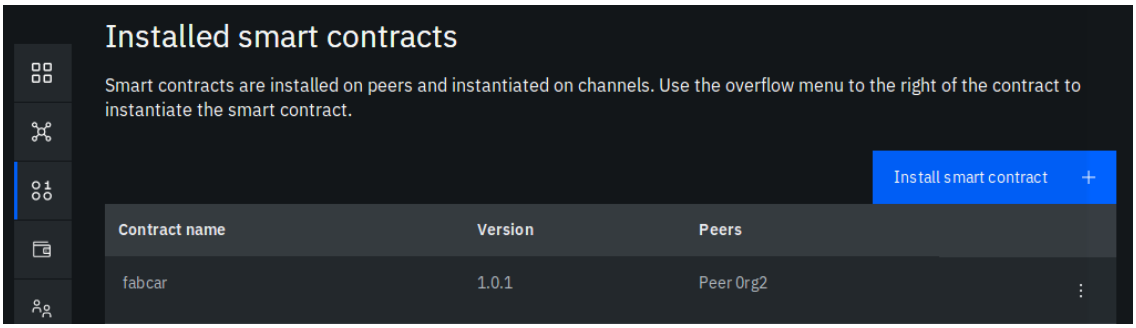
__ **17.** [**Org1 & Org2**] In the console UI, go to the **smart contracts** tab and click on the **Install smart contracts** button:



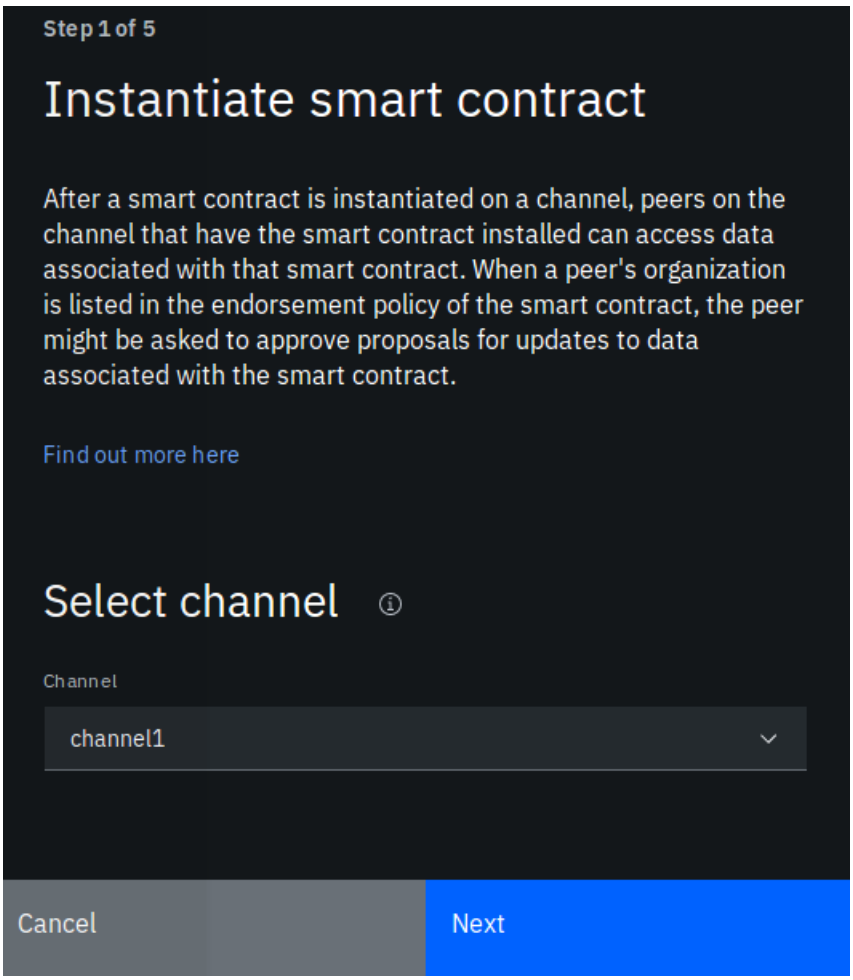
__ **18.** [**Org1 & Org2**] From the side bar, choose **add file** and navigate to the **fabcar@1.0.1.cds** contract you downloaded from GitHub above which will be in your **Downloads** folder. Once you have chosen the file, click "Open" and then click on the "**Install smart contract**" button:



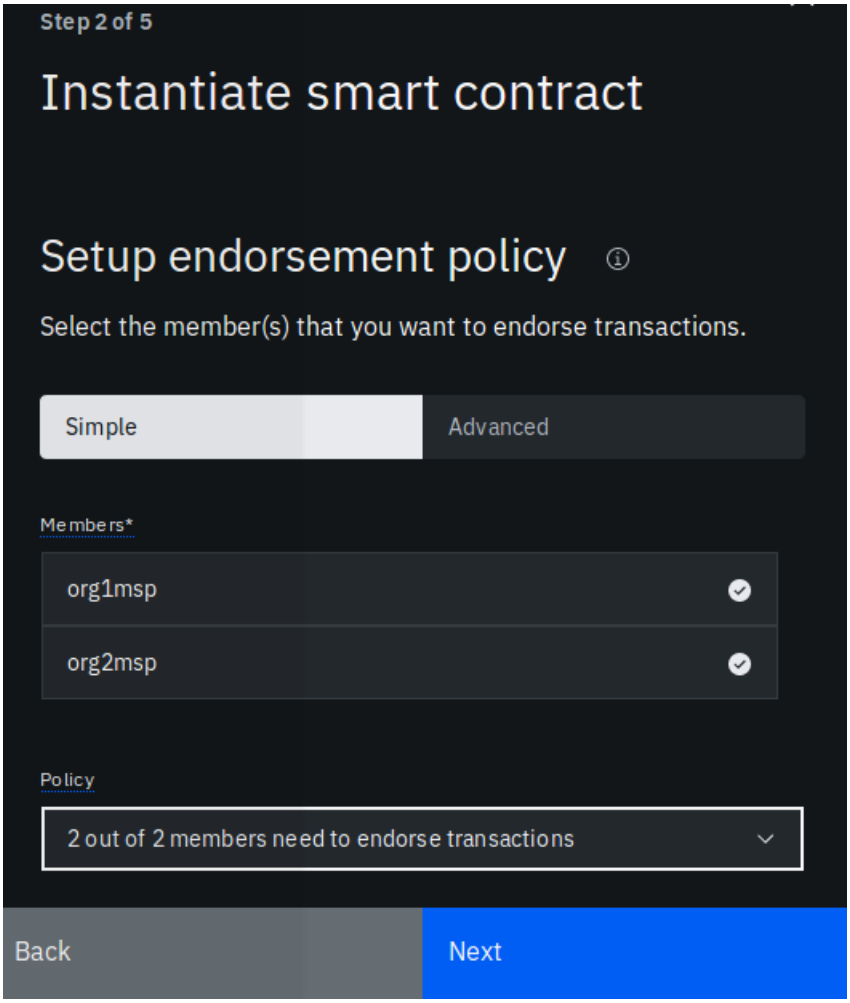
__ **19.** [**Org2 only**] On the same smart contracts page, find the **fabcar** contract we just installed and click on the vertical "**...**" button to the right and choose the "**Instantiate**" option:



__ **20.** [**Org2 only**] In the side panel, on **Step 1**, make sure you select the channel you joined earlier in the lab, **channel1**, and click **Next**:

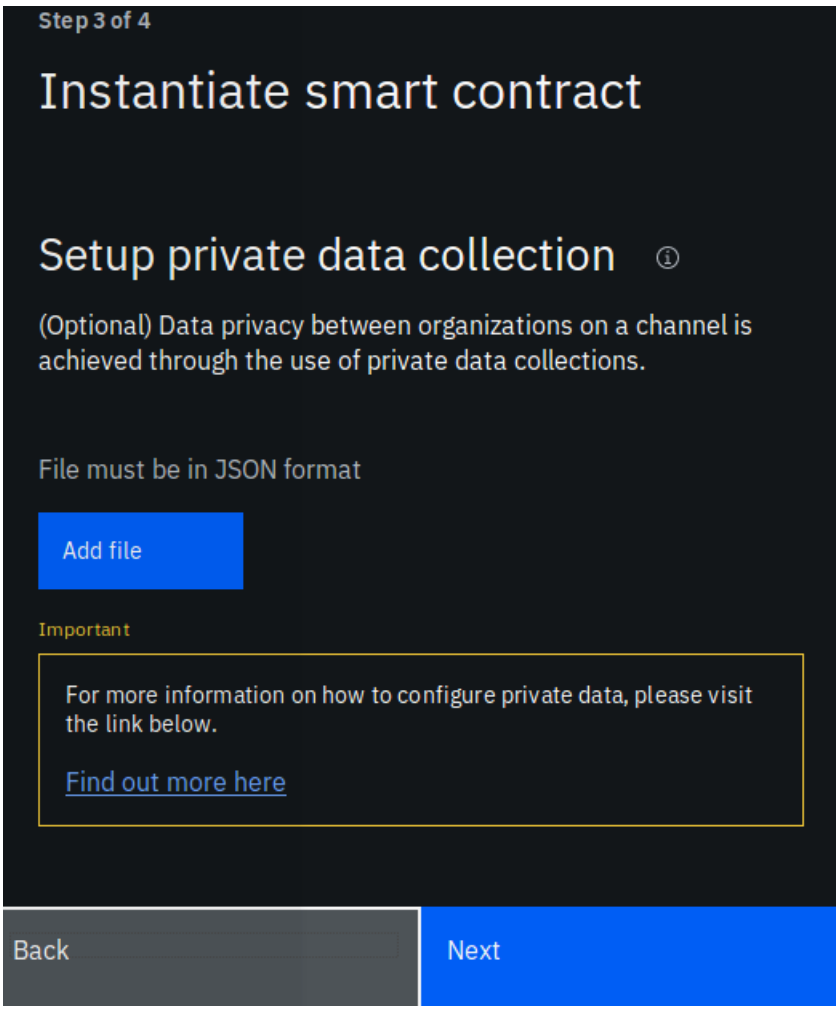


__ **21.** [**Org2 only**] In the side panel on **Step 2**, make sure both Members are selected (ticked) and change the policy to be "**2 out of 2 members need to endorse transactions**", the click **Next**:

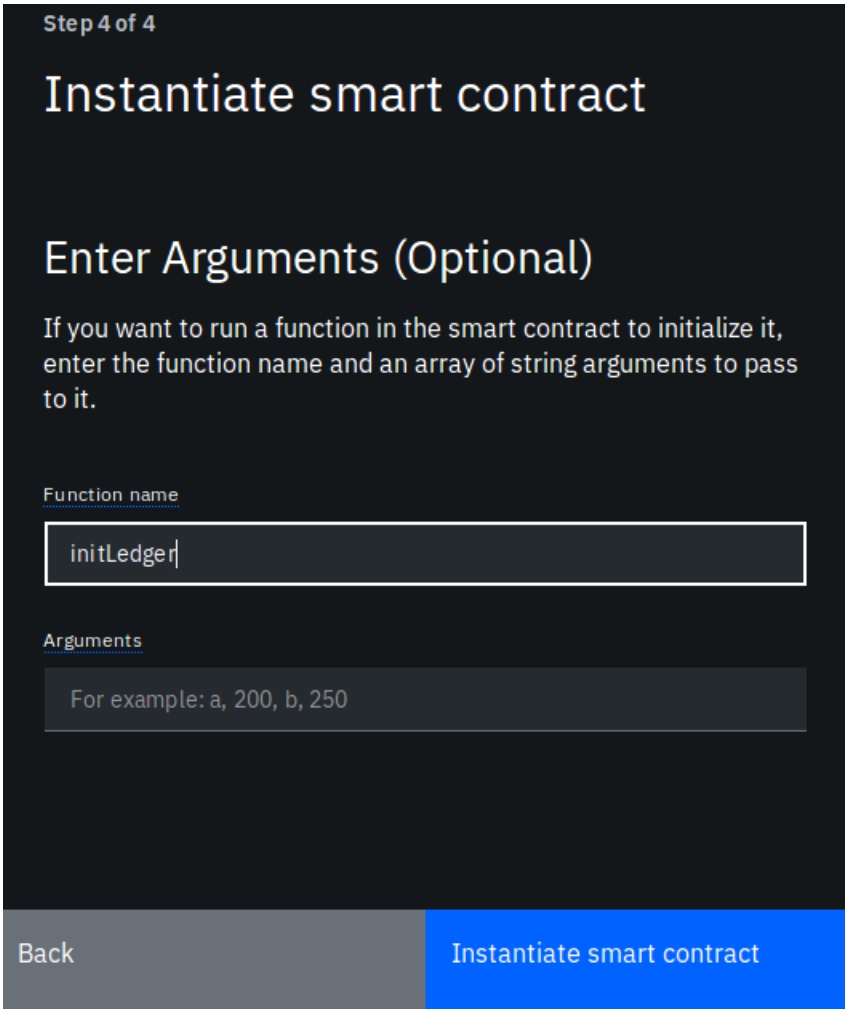


We set the endorsement policy to be "**2 of 2**" so we can ensure that both peers are required to execute and sign update transactions. However, this does mean we will need to configure the peer for **service discovery,** later in this lab.

__ **22.** [**Org2 only**] In the side panel on **Step 3**, as we are not using private data in this lab, you do not need to make any changes and can just click "**Next**":

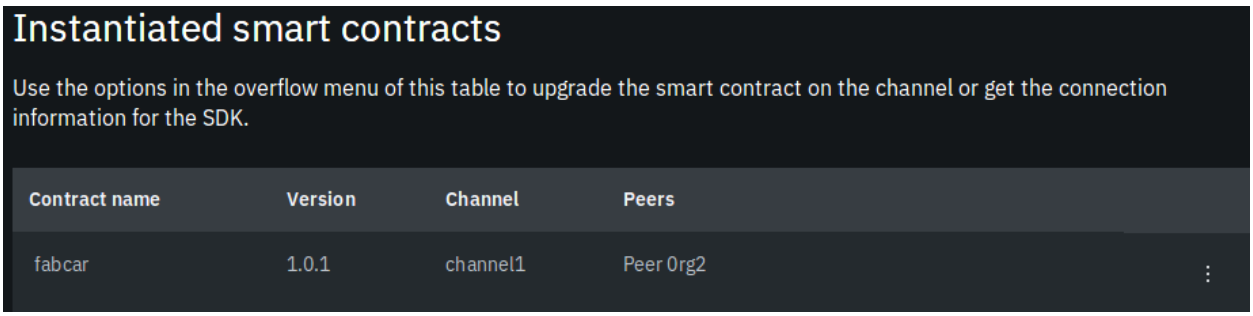


__ **23.** [**Org2 only**] In the side panel on **Step 4**, enter "**initLedger**" as the **Function name** to call which populates the ledger with a selection of sample cars. Leave the **Arguments** field empty and click "**Instantiate smart contract**":



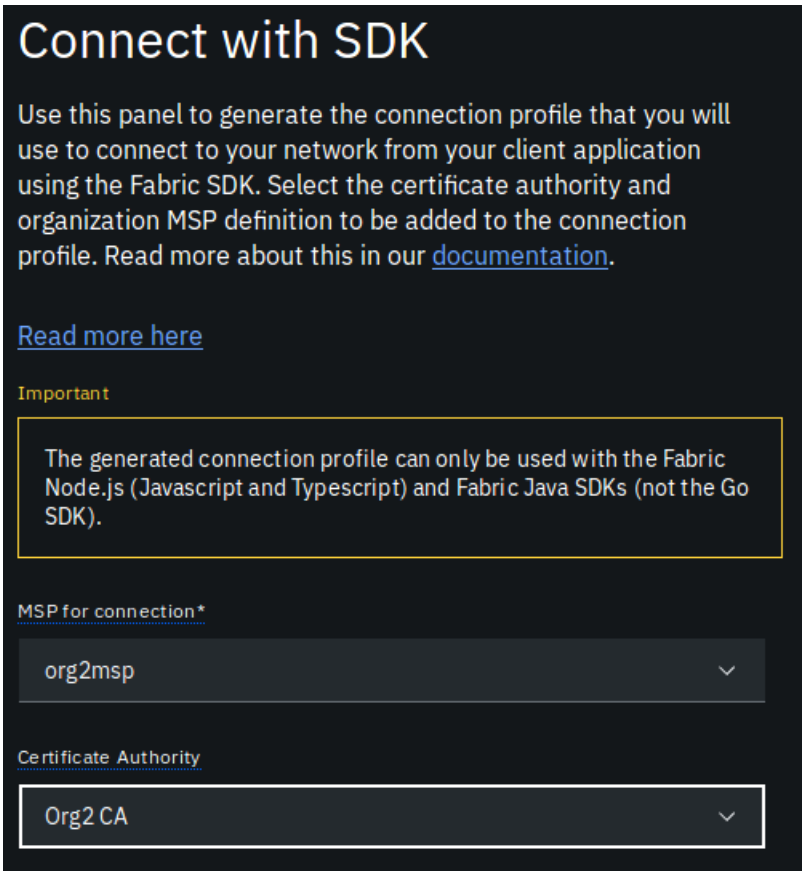
This step can take a few minutes to complete, please be patient.

__ **24.** [**Org1 & Org2**] On the same smart contracts page, scroll down to the "**Instantiated smart contracts**" section find the **fabcar** contract we just instantiated and click on the vertical "**...**" button to the right and choose the "**Connect with SDK**" option:

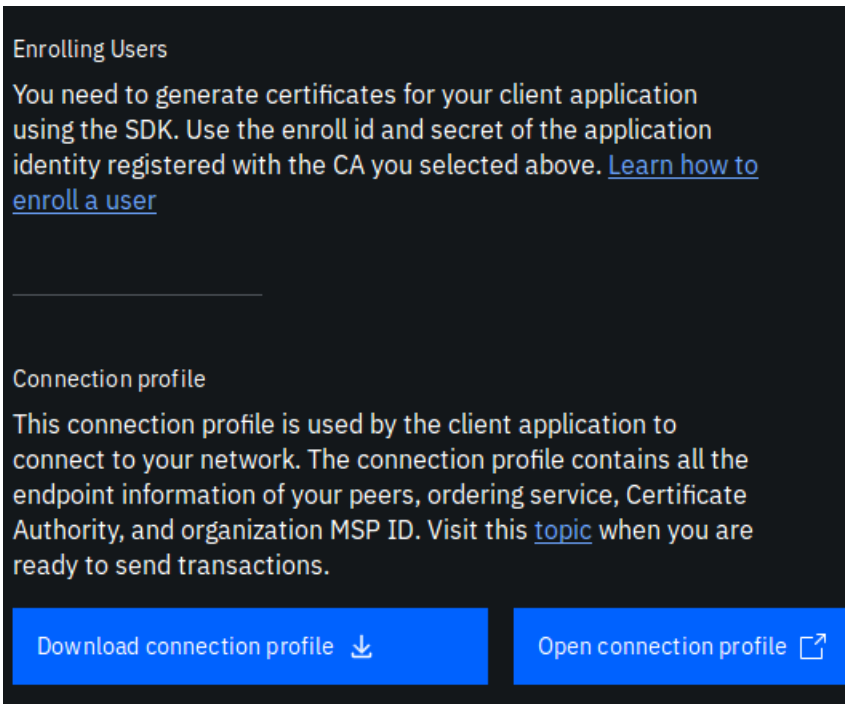


Note: **Org1** may need to refresh their page to see the instantiated contract.

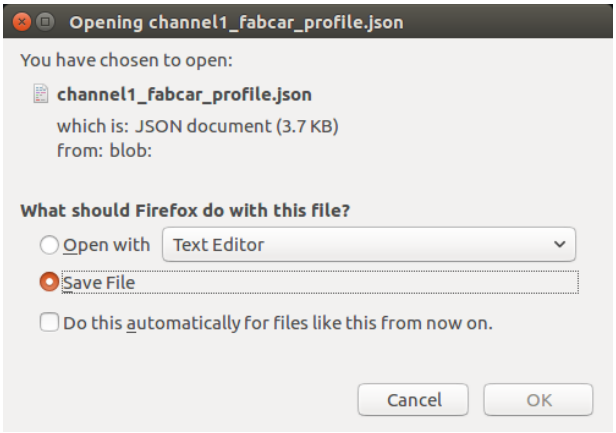
__ **25.** [**Org1 & Org2**] In the side panel select your MSP for connection and Certificate Authority. For **Org1** this will be "**org1msp**" and "**Org1 CA**", whilst for **Org2** this will be "**org2msp**" and "**Org2 CA**":



__ **26.** [**Org1 & Org2**] In the side panel scroll down and choose the "**Download connection profile**" button:



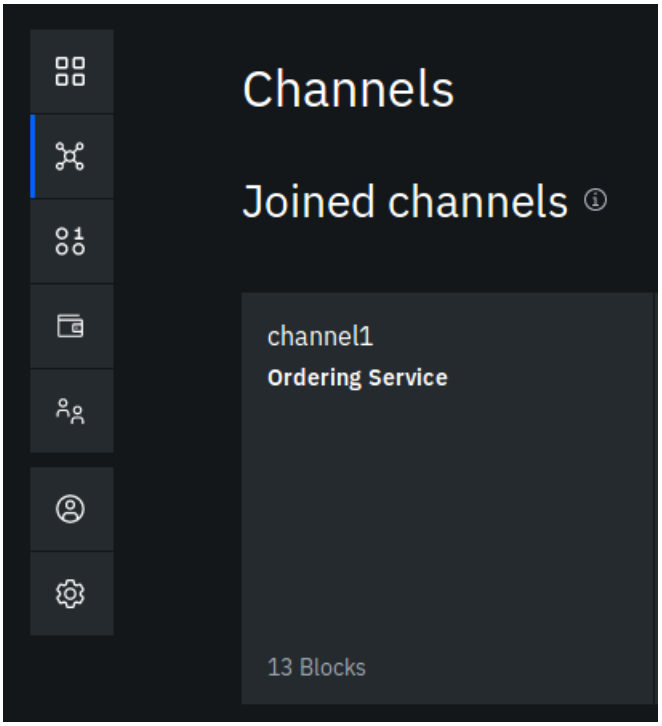
__ **27.** [**Org1 & Org2**] In the dialog, choose the "**Save File**" option and click "**OK**":



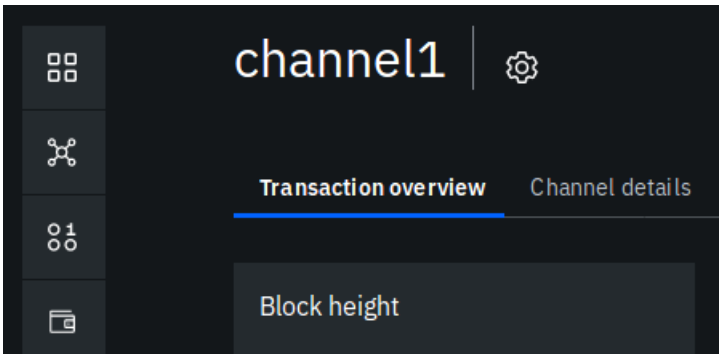
This will place the file "**channel1_fabcar_profile,json**" into the your **Downloads** folder. The path to this file is "**~/Downloads/channel1_fabcar_profile.json**".

__ **28.** [**Org1 & Org2**] Once you have downloaded the connection profile, click the "**Close**" button.

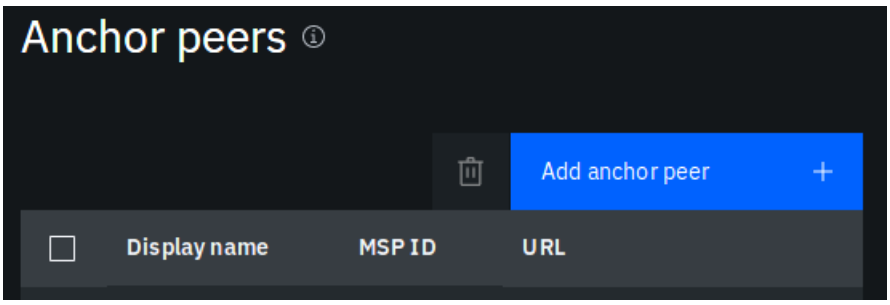
__ **29.** [**Org2 Only**] From the Channels icon, select **channel1**:



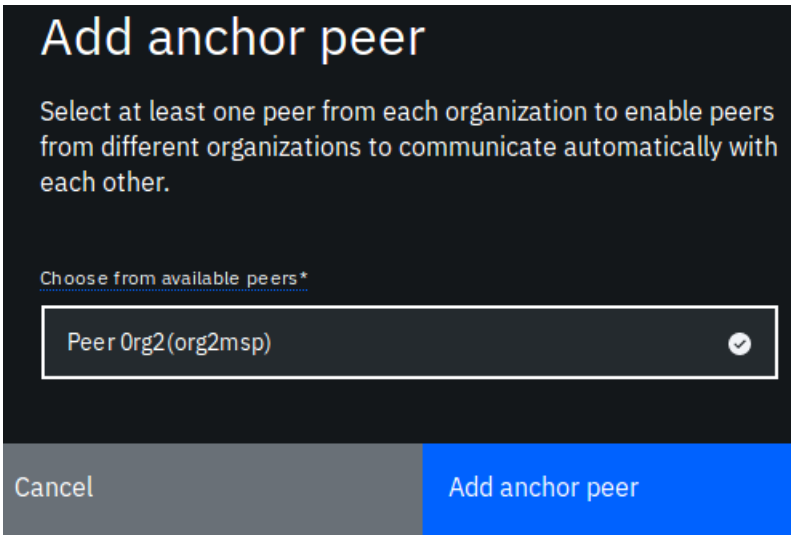
__ **30.** [**Org2 Only**] From the channels pane, choose the "**Channel details**" tab:



__ **31.** [**Org2 Only**] Scroll down the **channel1** details page to the **Anchor peers** section and click the "**Add anchor peer**" button:



__ **32.** [**Org2 Only**] From the side panel, choose your peer "**Peer Org2(org2msp)**" and click "**Add anchor peer**":

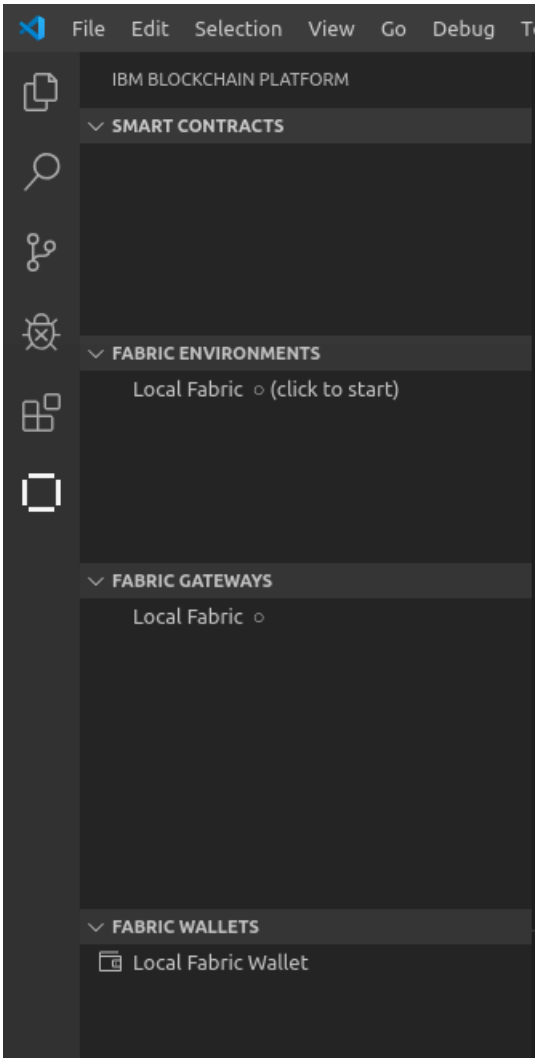


This will cause a configuration block to be added to the channel defining **Peer Org2** as an anchor peer for the channel which will allow Org1's **Peer Org1** to be able to discover Org2's peer and vice versa.

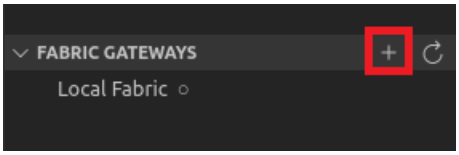
2.5 Connecting to the Network

This section must be completed by both organizations as indicated in each step

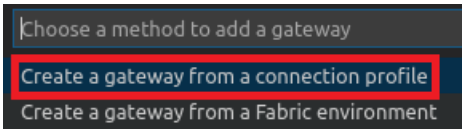
__ **33.** [**Org1 & Org2**] Open a new empty VS Code window and click on the **IBM Blockchain Platform** icon:



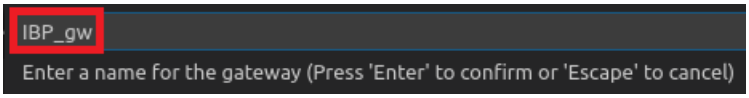
__ **34.** [**Org1 & Org2**] Move your mouse over the "**Fabric Gateways**" pane to make the "**+**" appear and click the "**+**" to start creating a new gateway:



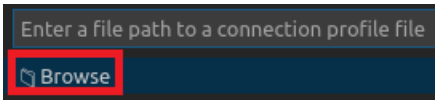
__ **35.** [**Org1 & Org2**] In the pop up, choose the "**Create a gateway from a connection profile**" option:



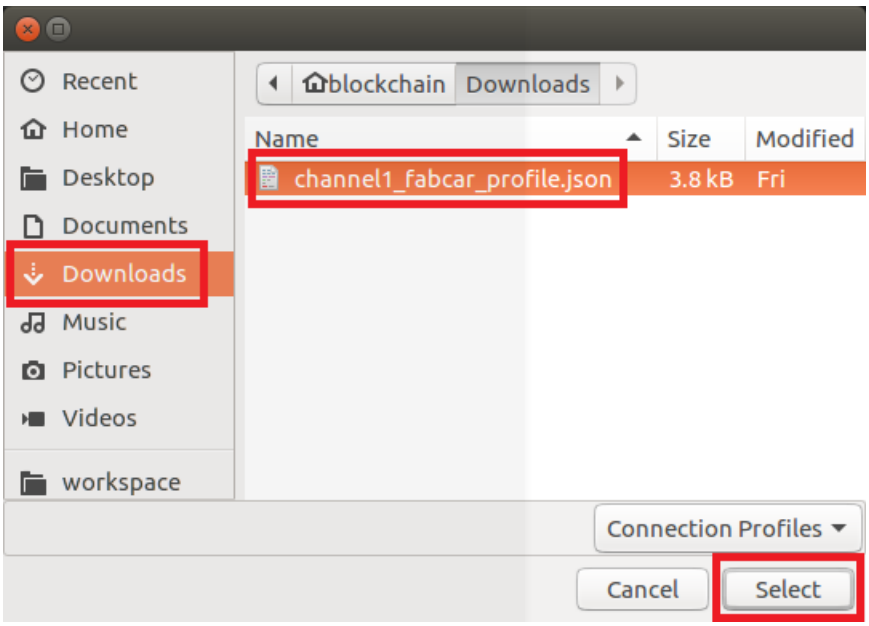
__ **36.** [**Org1 & Org2**] In the pop up enter "**IBP_gw**" as the name:



__ **37.** [**Org1 & Org2**] In the pop up click "**Browse**":

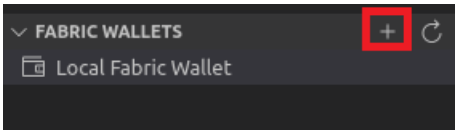


__ **38.** [**Org1 & Org2**] In the dialog click "**Downloads**" on the left, choose the "**channel1_fabcar_profile.json**" file then click "**Select**":

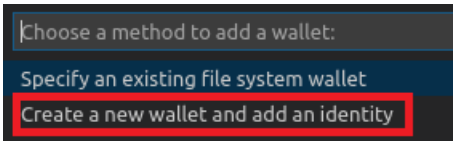


The Fabric Gateways will update with the new gateway, but before we can use it to connect we first need to create a new wallet with a new user id in it.

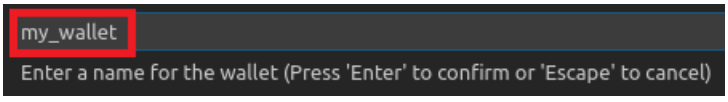
__ **39.** [**Org1 & Org2**] Move your mouse over the "**Fabric Wallets**" pane to make the "**+**" appear and click the "**+**" to start creating a new wallet:



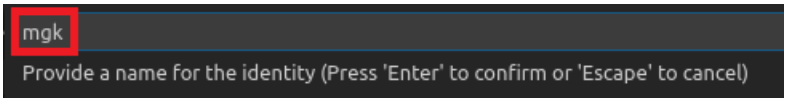
__ **40.** [**Org1 & Org2**] In the pop up choose "**Create a new wallet and add an identity**":



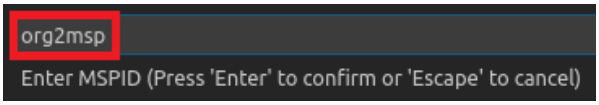
__ **41.** [**Org1 & Org2**] In the pop up enter the name "**my_wallet**" and press enter:



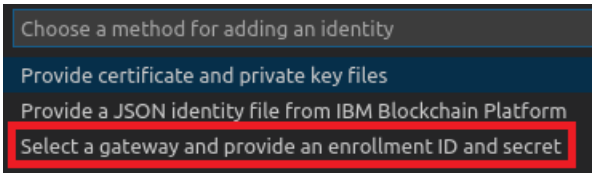
__ **42.** [**Org1 & Org2**] In the pop up enter your own name or initials (with no spaces or special characters) and press enter:



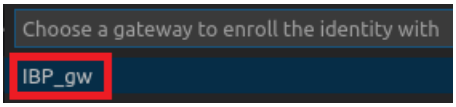
__ **43.** [**Org1 & Org2**] Enter your MSPID. For Org1 this is "**org1msp**" while for Org2 this is "**org2msp**", then press enter:



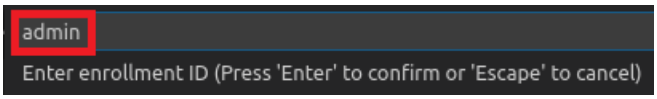
__ **44.** [**Org1 & Org2**] In the pop up choose "**Select a gateway and provide an enrollment ID and secret**":



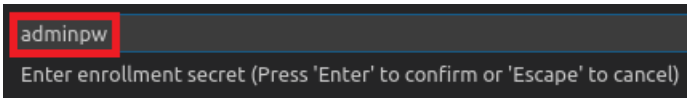
__ **45.** [**Org1 & Org2**] In the pop up choose the "**IBP_gw**" gateway we created earlier:



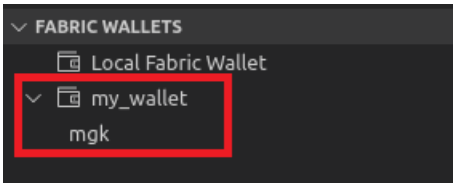
__ **46.** [**Org1 & Org2**] In the pop up enter the CA enrollment ID we created earlier "**admin**":



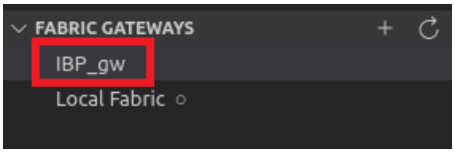
__ **47.** [**Org1 & Org2**] In the pop up enter the CA enrollment secret we created earlier "**adminpw**":



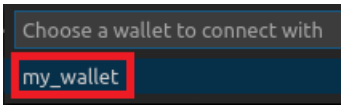
There will be a couple of Information Messages and then the wallets pane will update to show the new wallet with the new ID inside it:



__ **48.** [**Org1 & Org2**] From the "**Fabric Gateways**" view select the "**IBP_gw**" gateway we created earlier:

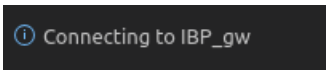


__ **49.** [**Org1 & Org2**] In the pop up choose the "**my_wallet**" wallet we just created:



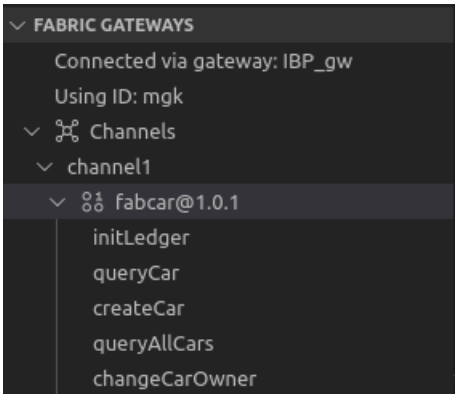
Note: It will not ask you to select the ID inside the wallet as we only have one ID. If we had more than one, there would be an extra step to choose the ID to use as well.

There will be an Information Message when the connection is complete:



Also the Fabric Gateway view will update to show the details of the network we are connected to.

__ **50.** [**Org1 & Org2**] If you expand the **channel1** channel and the **fabcar@1.0.1** smart contract you should be able to see the transactions available in the contract:

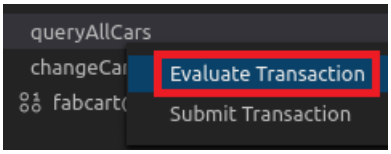


2.6 Issuing Transactions

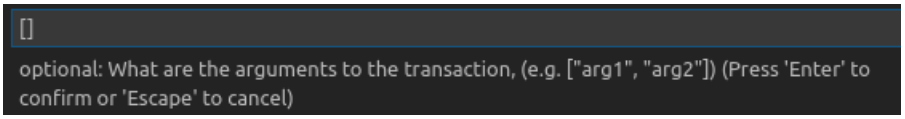
This section must be completed by both organizations as indicated in each step

We are now going to issue transactions to test that both networks are set up correctly.

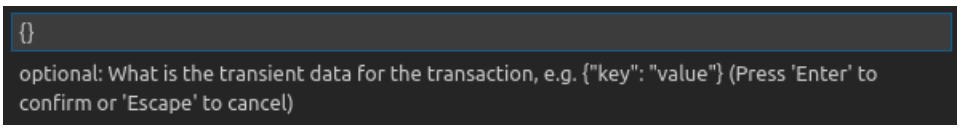
__ **51.** [**Org1 & Org2**] Right click on the "**queryAllCars**" transaction and choose "**Evaluate Transaction**":



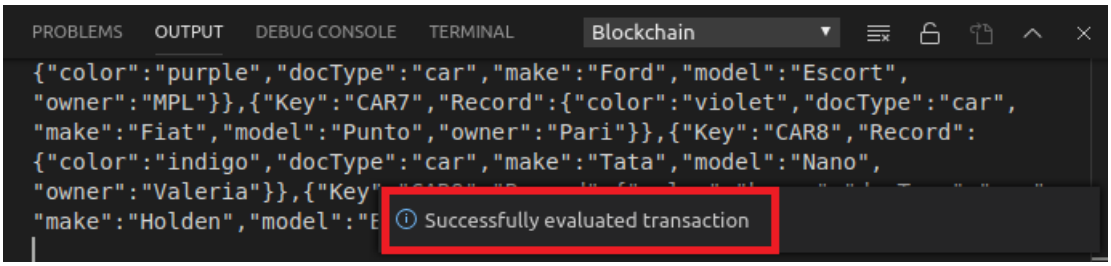
__ **52.** [**Org1 & Org2**] In the pop up just press enter as **queryAllCars** does not need any parameters:



__ **53.** [**Org1 & Org2**] In the next pop up just press enter as **queryAllCars** does not use any transient data:

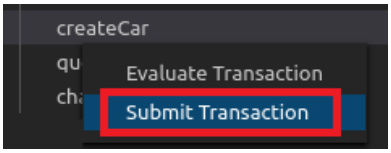


As well as an **Information Message**, you should see the output window update with the details of all the cars that the "initLedger" transaction created when we instantiated the contract:



Now we are going to create a new car. As there are now two peers in the network and both peers are required to endorse the transaction, it will automatically be sent to both peers by the VS Code extension.

__ **54.** [**Org1 & Org2**] Right click on the "**createCar**" transaction and choose "**Submit Transaction**":

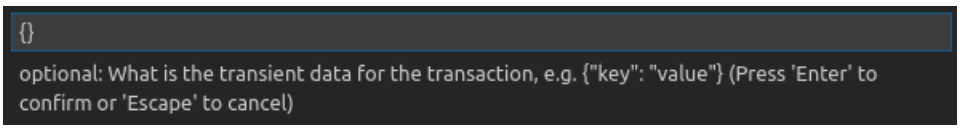


__ **55.** [**Org1 & Org2**] In the pop up enter the parameters for your new car, using a different ID for each org. For Org1 use **"CAR100"** and for Org2 use "**CAR200**" to differentiate between them. Enter some values of your choice like **"CAR200","Tesla","Model S","Red","MGK"** inside the square brackets and press "Enter". The order of the parameters for reference is:

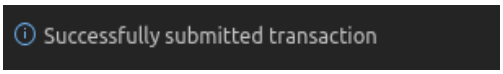
ID, Make, Model, Color, Owner

Note: Remember, you should not enter any quotes or extra spaces around this string as otherwise they may be taken as part of the string itself which will result in an error.

__ **56.** [**Org1 & Org2**] In the next pop up just press Enter as **createCar** does not use any transient data:



You should see a successful **Information Message**:



__ **57.** [**Org1 & Org2**] Use the **queryCar** transaction twice to query for your own car and your partner Organization's car. For example, each org should query for **CAR100** and **CAR200** and verify they can see the expected results.

__ **58.** [**Org1 & Org2**] **Congratulations**, you have now finished this Lab. Please use any remaining time to experiment with the environment you have created.

3 We Value Your Feedback!

- Please ask your instructor for an evaluation form. Your feedback is very important to us as we use it to continually improve the lab material.

- If no forms are available, or you want to give us extra information after the lab has finished, please send your comments and feedback to "**blockchain@uk.ibm.com**"