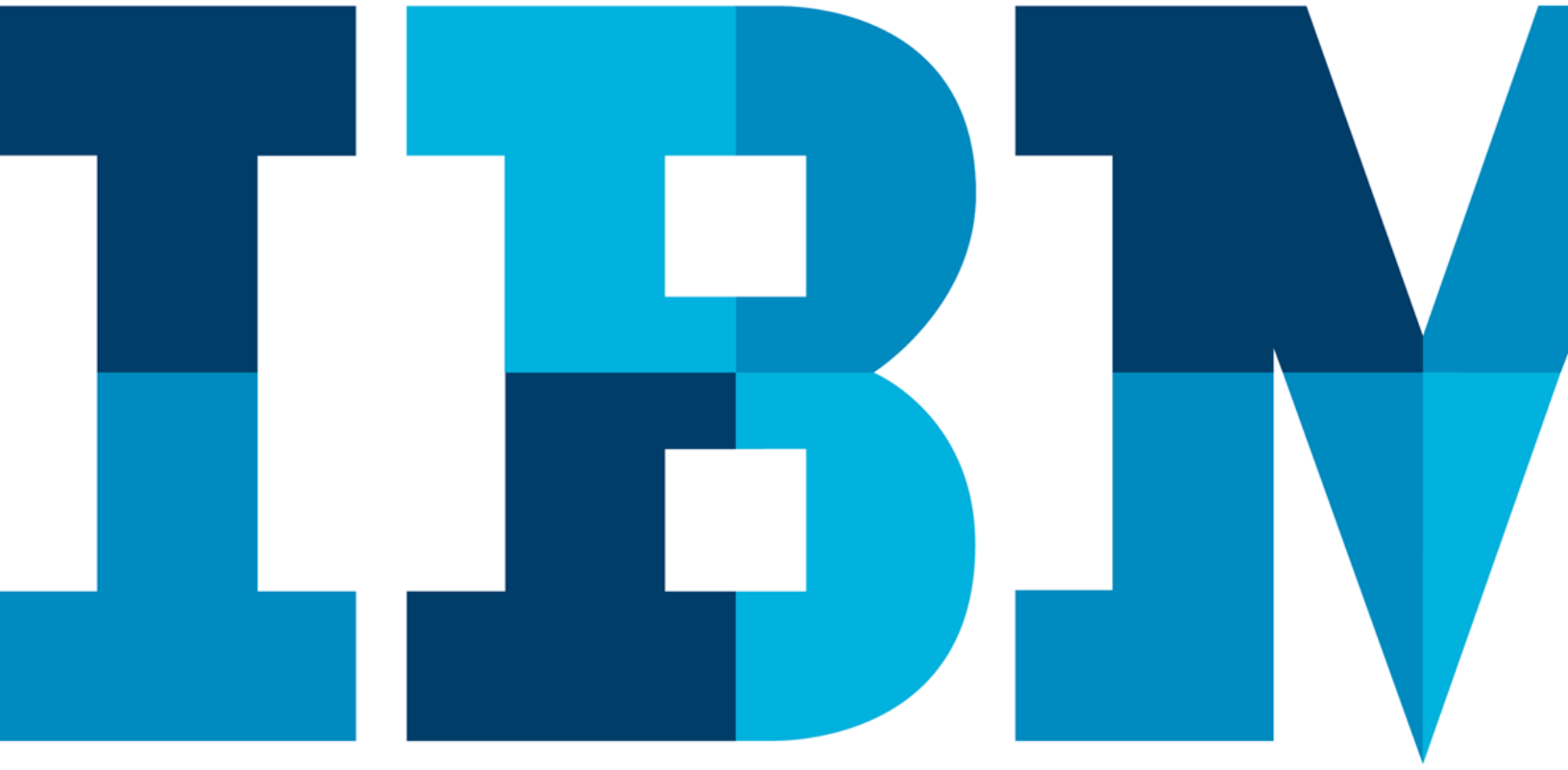
IBM Blockchain Platform Hands-On

Lab 5:

IBM Blockchain Platform
Building a Network 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 should 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 to **Build** a new blockchain network using the console. You will then deploy an existing smart contract, issue transactions and see the results.

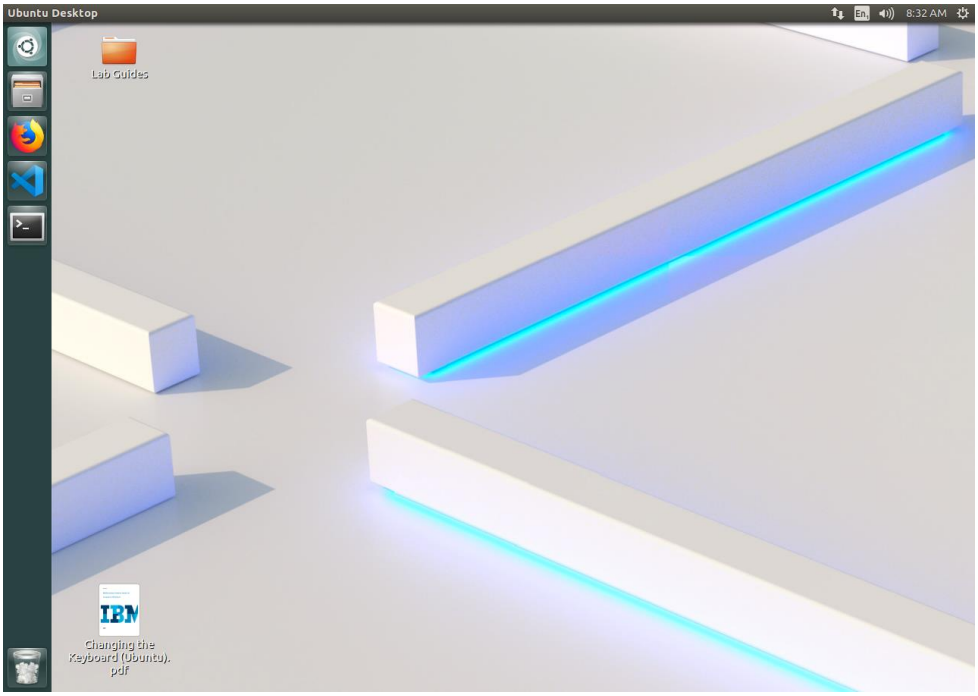
***Note**: The screenshots in this lab guide were taken using version **1.41.1** of **VS Code**, version **1.0.18** of the **IBM Blockchain Platform** plugin and version **0.4.30** of the **IBM Blockchain Platform** console. If you use different versions, you may see differences to those shown in this 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.

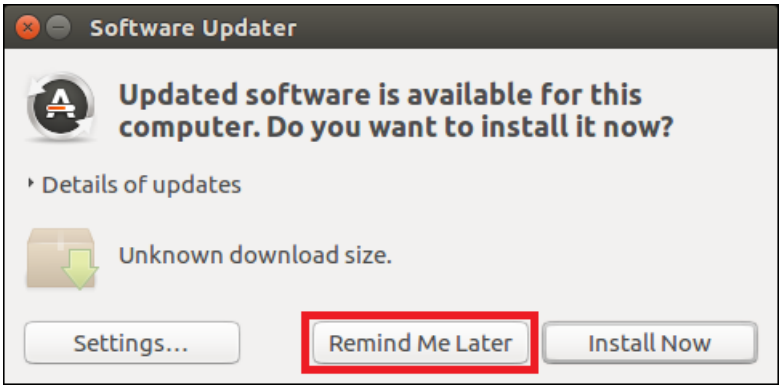
Note: If it asks you to login, the userid and password are both "**blockchain**".



1.1 Lab Scenario

In this lab, we will be creating a new network on the **IBM Blockchain Platform** running on **OpenShift Origin**, also known as **OKD**, which is the open source version of **Red Hat OpenShift**.

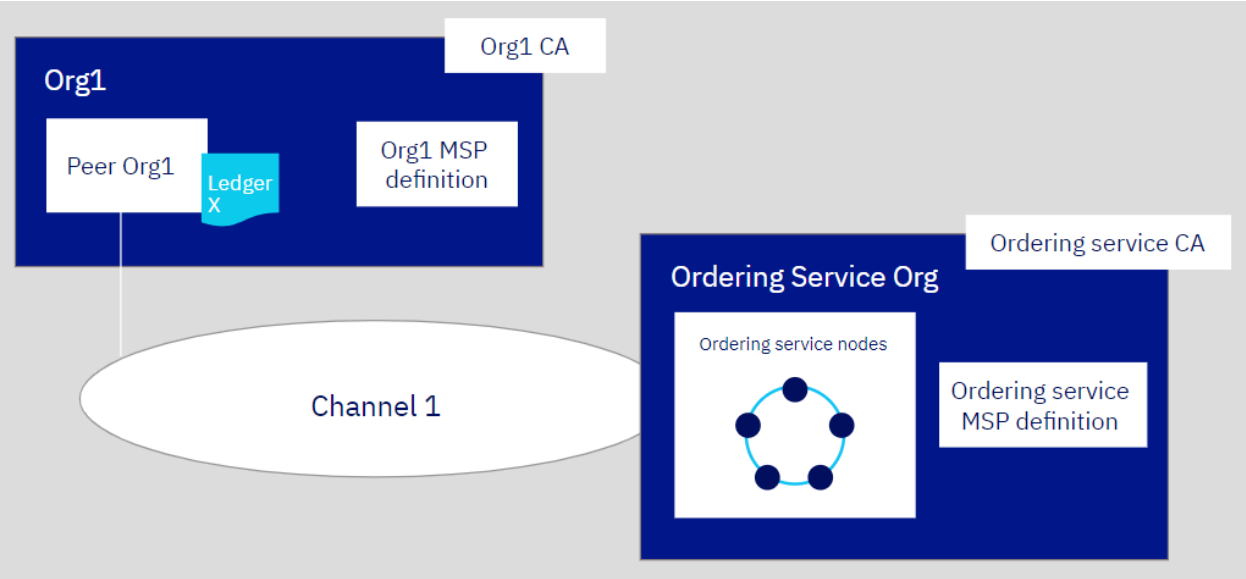
***Note** that if you get a "**Software Updater**" pop-up at any point during the lab, please click "**Remind Me Later**":*



2 IBM Blockchain Platform Operations

As mentioned above, in this lab we will be building a new blockchain network containing one transacting organization that manages a peer and one management organization that runs the ordering service. In addition, we will be using the IBM Blockchain platform **Build a network** tutorial which we will look at shortly.

The simple network we will be building looks like this:



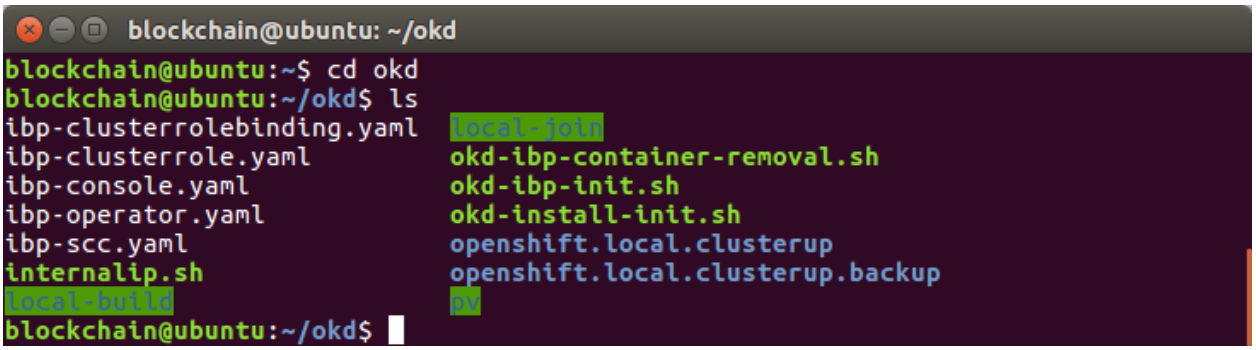
2.1 Setting up the IBM Blockchain Platform.

__ **3.** Open a new terminal prompt and change to the **okd** folder in your home directory:

```
cd okd
```

__ **4.** Run the ls command to look at the files available

```
ls
```



Here you can see several files that make up the configuration of IBP and OKD along with a couple of scripts to set things up.

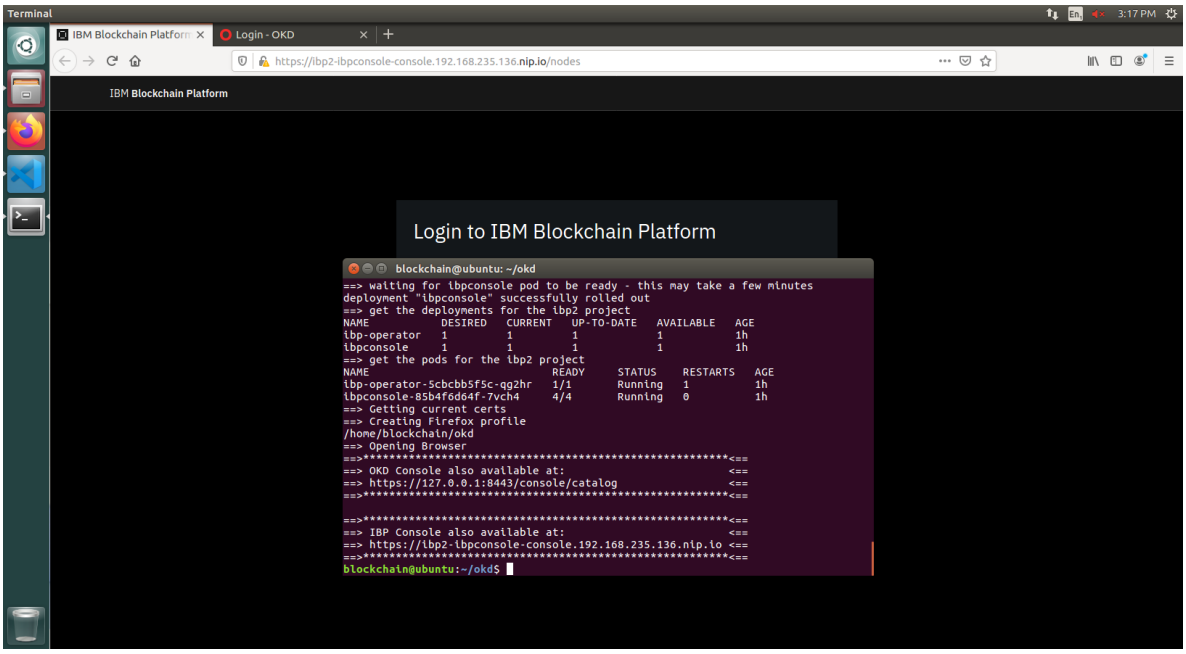
OKD provides management for groups of containers by grouping them into pods and clusters. In this lab we will use a single node cluster to manage IBM Blockchain Platform. The cluster we will use is pre-configured in the lab with the containers that make up IBM Blockchain Platform, so now we need to bring the cluster up and start the IBP Console.

__ **5.** First, we need to start the OKD cluster and deploy a new instance of the IBP Console into it by running a script. In the terminal window enter:

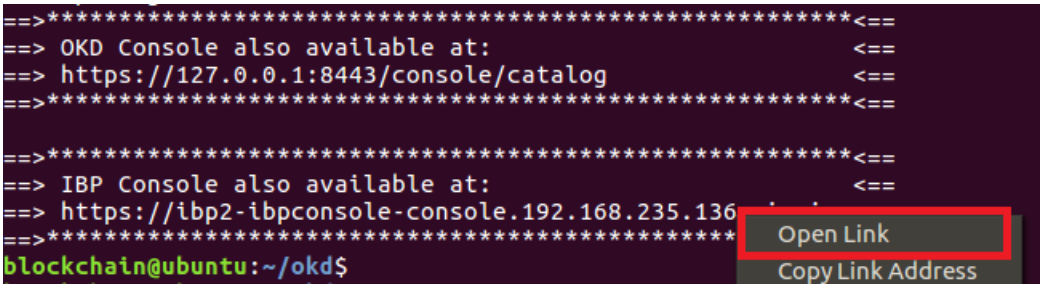
```
./local-build/start-ibp-console.sh
```

Note: Make sure you have the right script name (which must have **ibp** in its name) and that you type the leading period and forward slash "**./** "

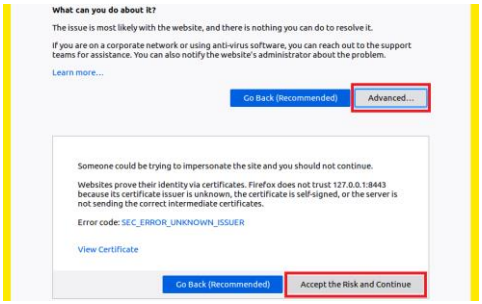
This command will take a few minutes to run and will produce several screens of output. When finished, Firefox should open automatically, and your screen should be similar to this:



__ **6.** At the end of the terminal output, in case you close your Firefox window by accident, there are links to the IBP Console and OKD admin console. If you need to reopen them, right-click on the link and choose "**Open Link**":

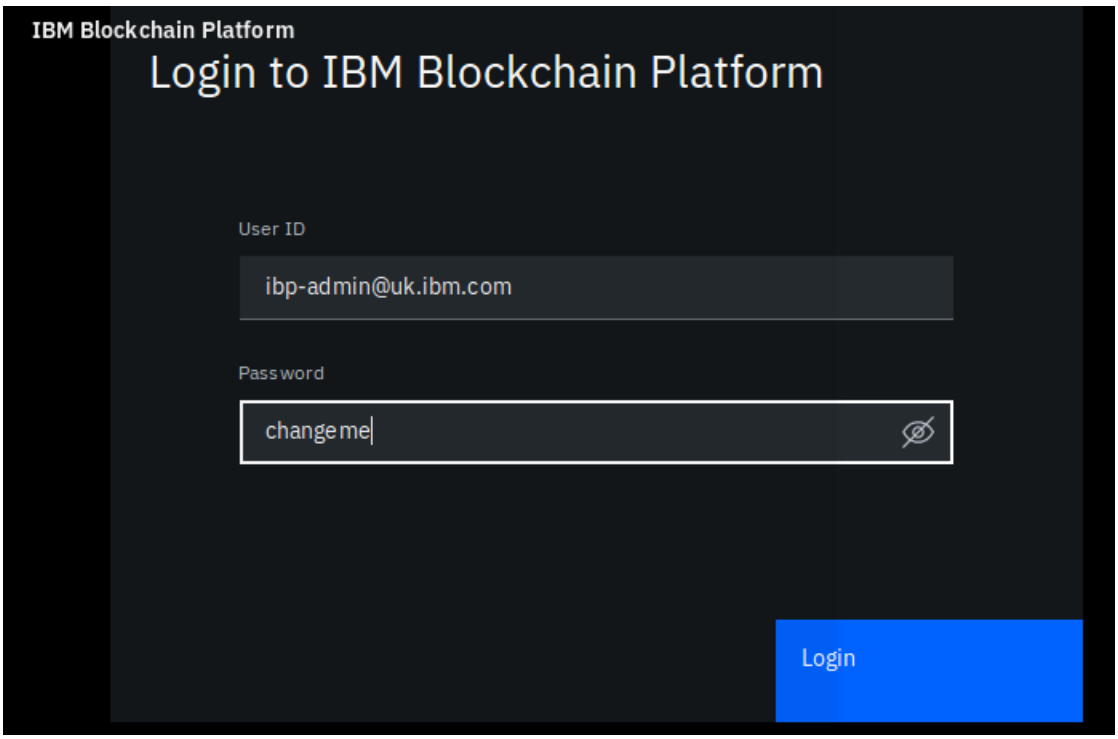


Note: At this point Firefox should be open with the console homepage. However, if you do see a security warning in Firefox you will need to accept the certificate to continue, so click the "**Advanced...**" button to expand the information then click the "**Accept the Risk and Continue**" button.



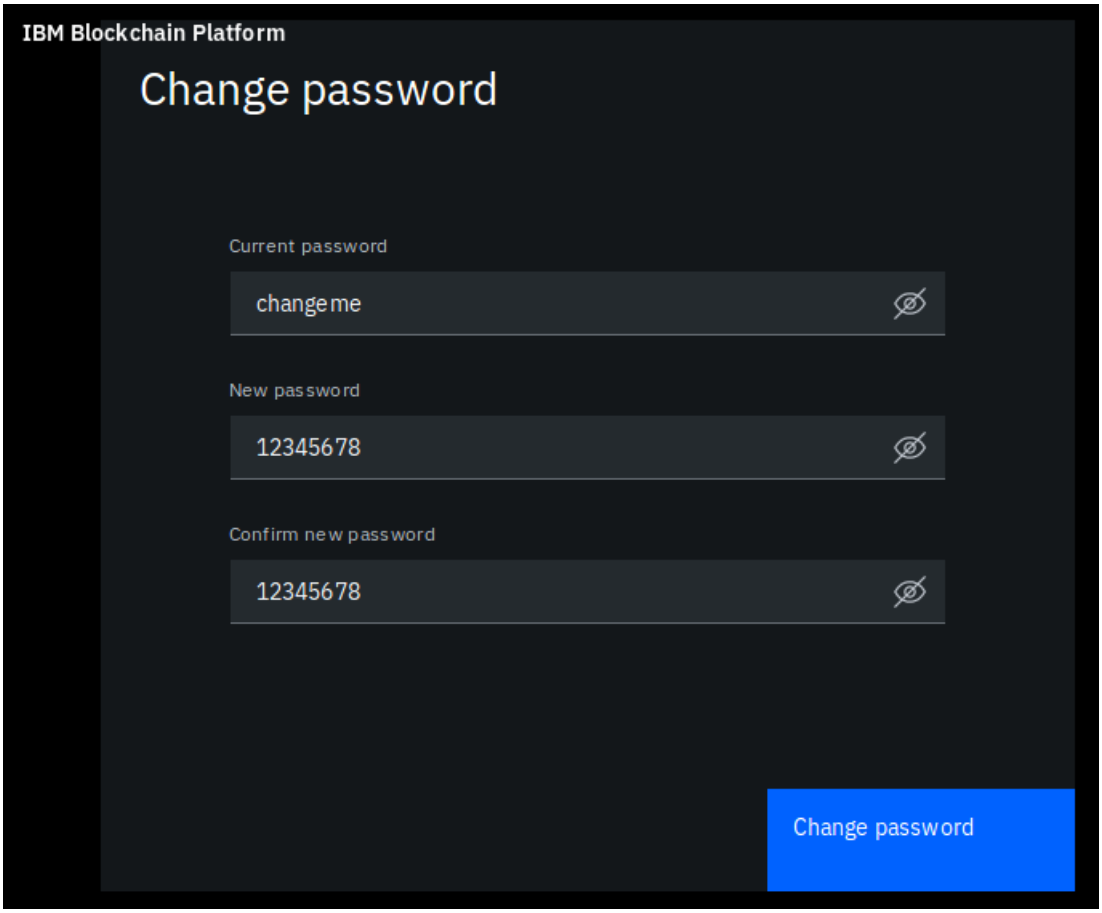
__ **7.** Make sure the **IBM Blockchain Platform** tab is selected in Firefox. You will then be presented with the "**Login to IBM Blockchain Platform**" page:

__ **8.** On the Login page, enter the User ID **ibp-admin@uk.ibm.com** and the Password **changeme** then click the "**Login**" button:



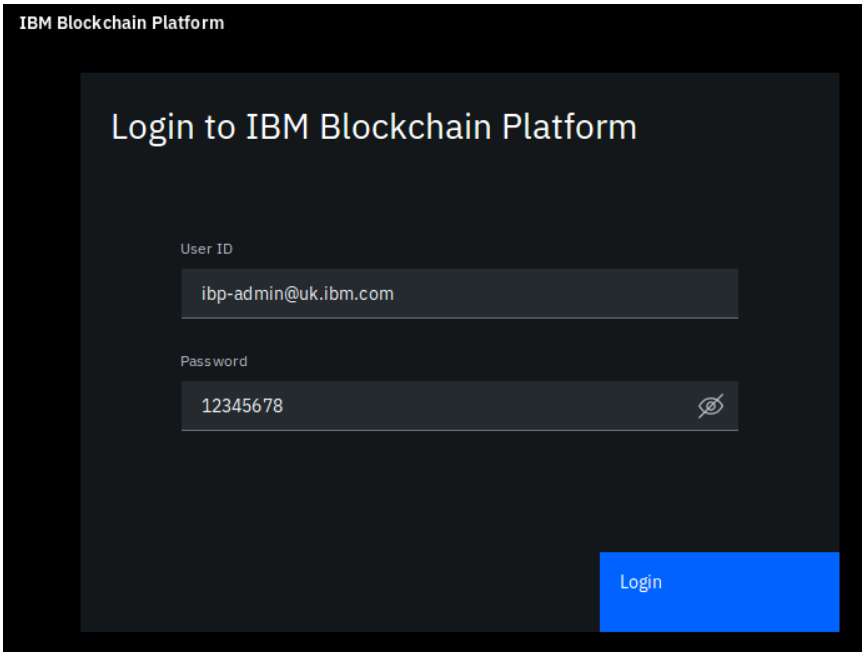
__ **9.** You will then be asked to "**Change password**" because it is the first usage of the console. In the "**Current password**" field enter "**changeme**" again. In the "**New password**" and the "**Confirm new password**" fields enter "**12345678**" as a simple password for the lab.

Make sure you click on the "**eye**" symbol after entering the password to make sure it has been entered correctly as shown below, then click the "**Change password**" button:

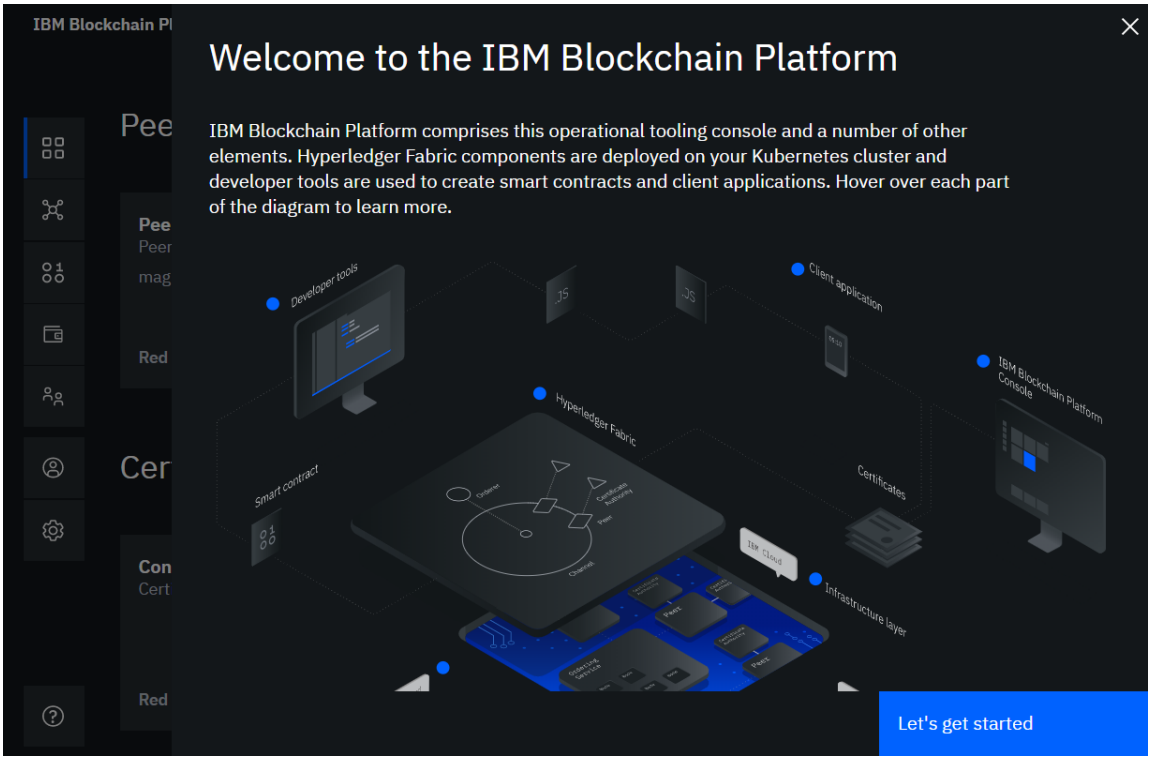


Note: Make sure you remember the password as you will need it again in the next step.

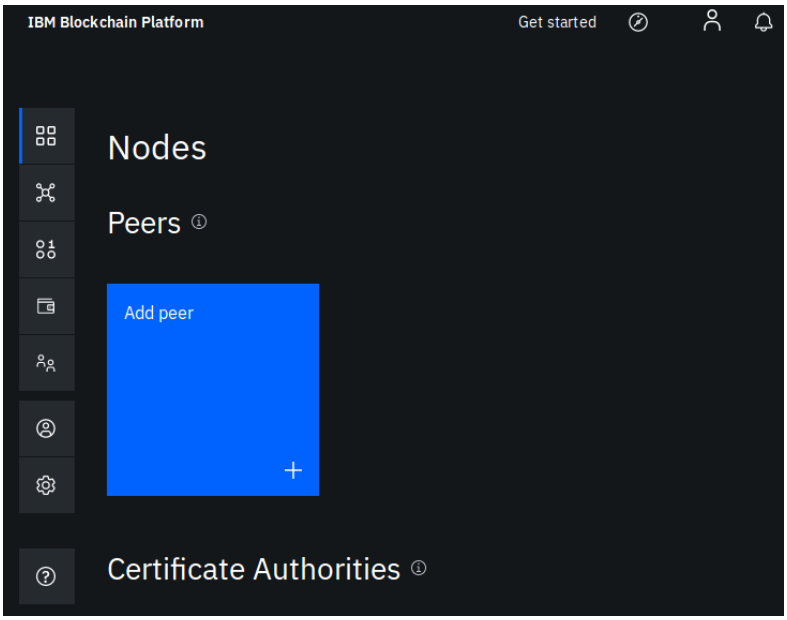
__ **10.** You will now be presented with the "**Login to IBM Blockchain Platform**" page again, but this time you must login with the newly changed password details. In the "User ID" field enter **ibp-admin@uk.ibm.com** and in the "**Password**" field enter **12345678** as you did in the previous step:



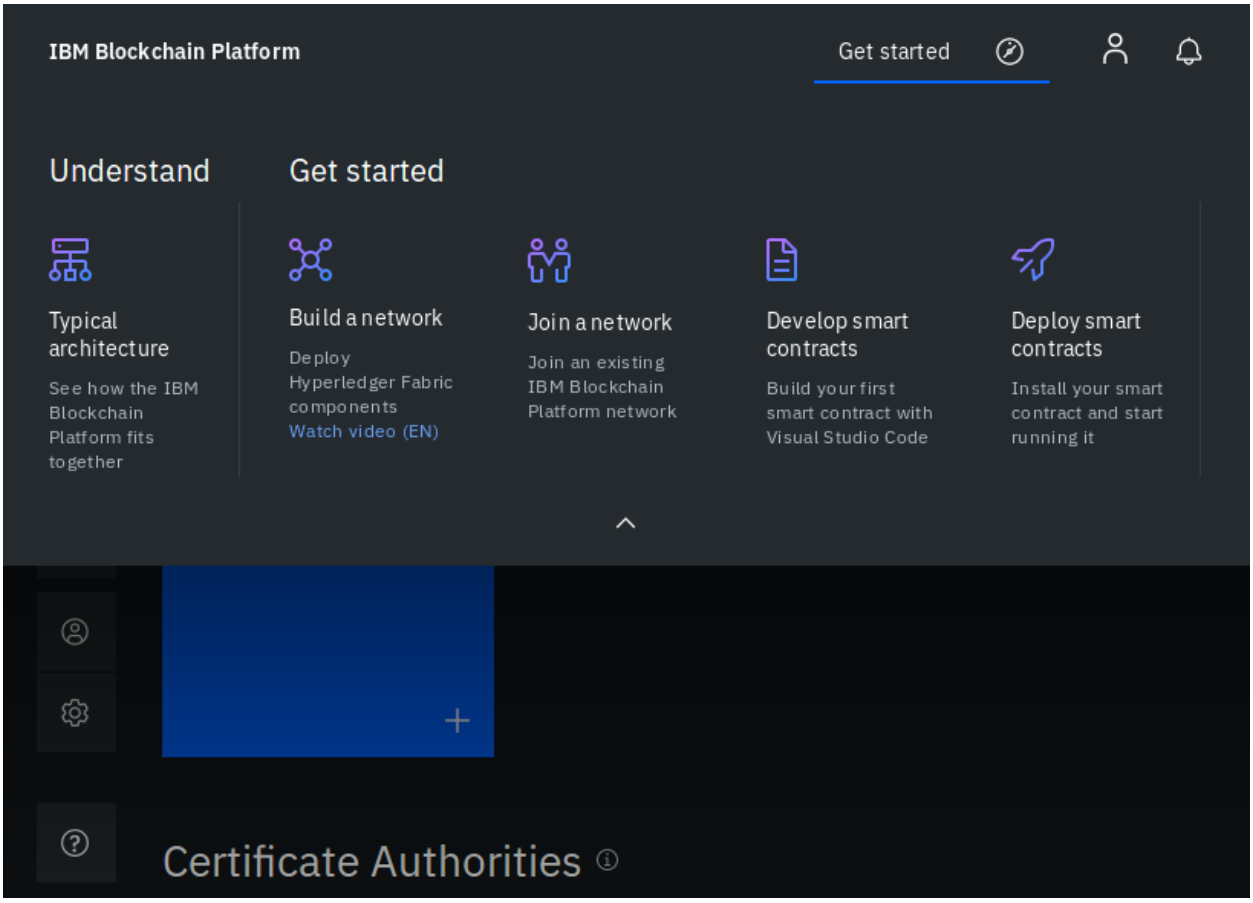
__ **11.** You will then see the **Welcome** page for the platform. Take a little time to move your mouse around the interactive diagram to see the different parts in a simple platform deployment:



__ **12.** When you are done, click the "**Let's get started**" button to close the diagram and move to the main console page:



__ **13.** 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:



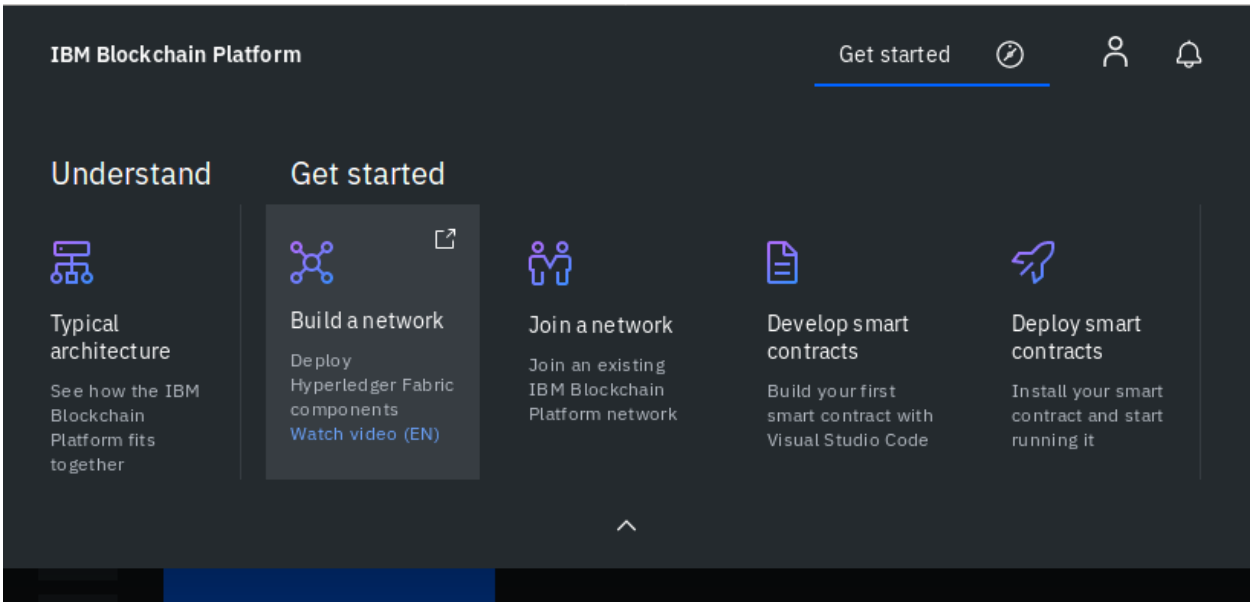
This concludes the setup part of this lab, you have deployed a new blockchain service into an OKD cluster.

In the rest of this lab, we will work to build a blockchain network.

2.2 Building the network

We are now going to follow the built in "**Build a network**" tutorial. The tutorial covers a more complex scenario than we need as it adds in a third organization, so for simplicity we are going to concentrate on building the smaller network we discussed earlier.

__ **14.** Click on the "**Build a network**" link:



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

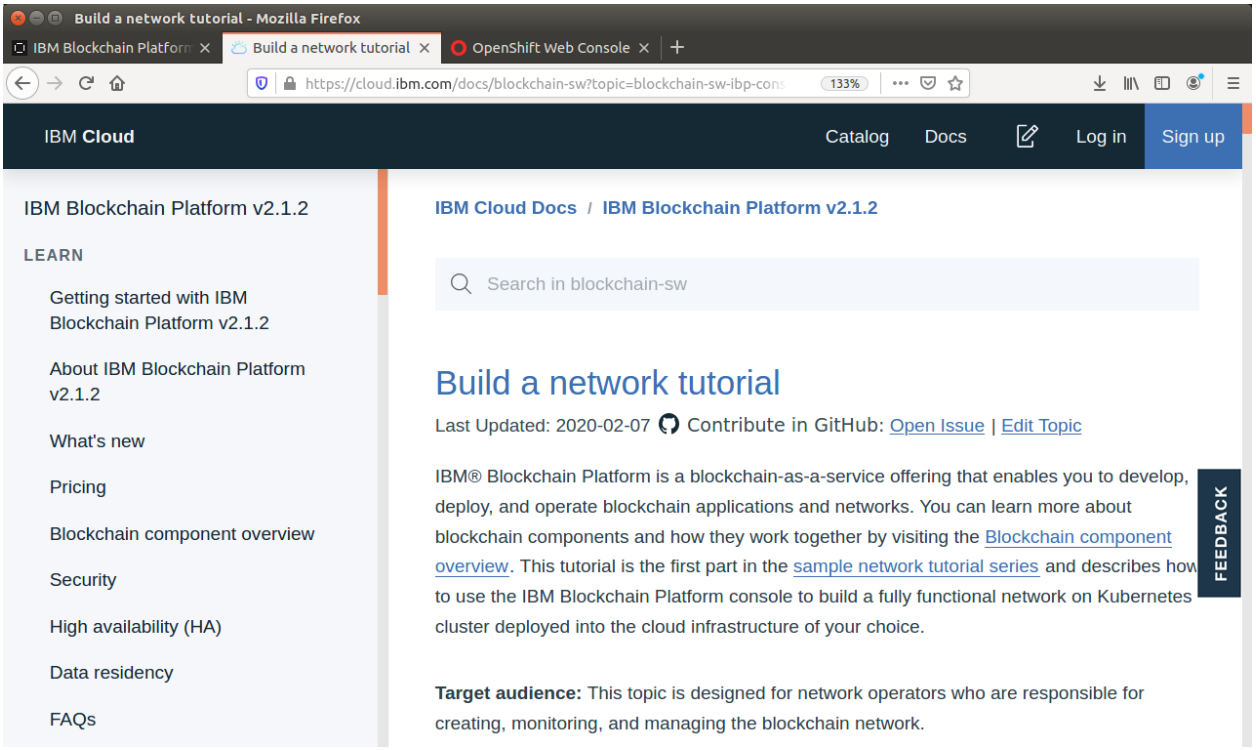
https://cloud.ibm.com/docs/services/blockchain-rhos?topic=blockchain-rhos-ibp-console-build-network

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

__ **15.** 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.



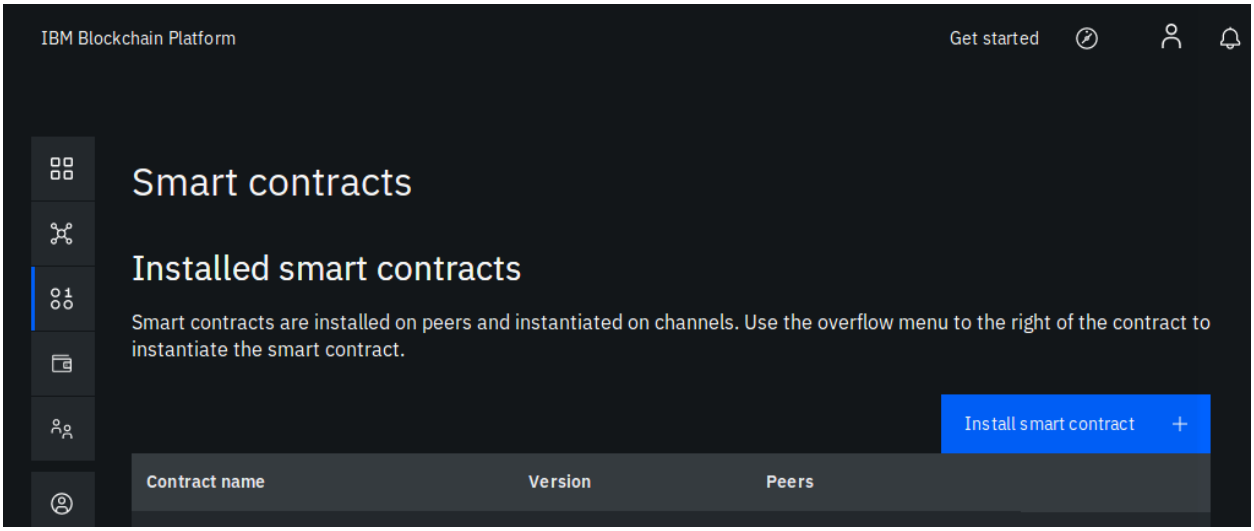
When working through this section, whilst you are waiting for components such as CAs, Peers an Ordering nodes to deploy and start you can check on their progress with the **OKD Web Console** using the **Monitoring** tab you if you wish. Instructions to do this are shown in **Appendix 1** at the end of this guide. You can also use OKD commands like "`oc get pods`" in the terminal window to check on progress from the command line as well if you wish.

2.3 Deploying into the network

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.

To save time, we are going to use an existing packaged version of the contract.

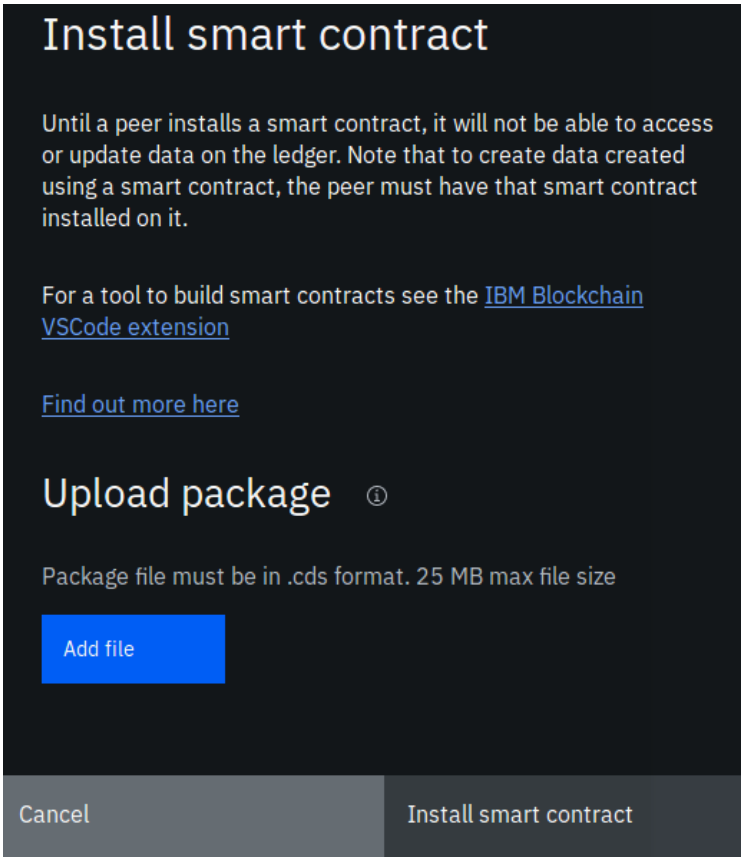
__ **16.** In the console UI, go to the **smart contracts** tab and click on the **Install smart contracts** button:



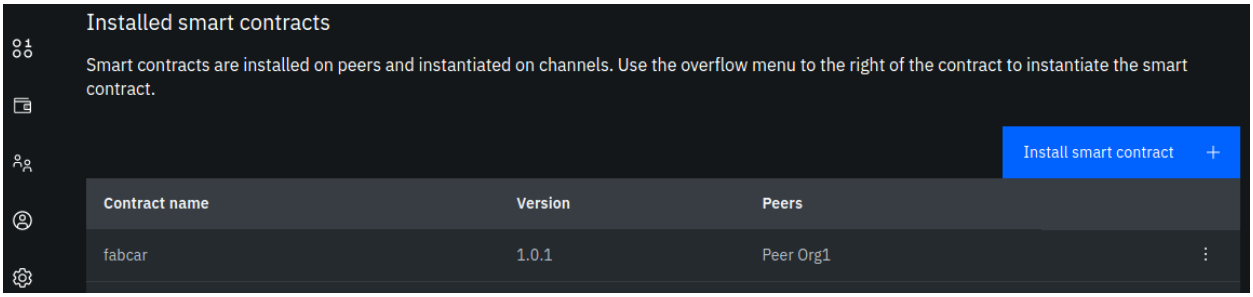
__ **17.** From the side bar, choose **add file** and navigate to the pre-prepared **fabcar@1.0.1.cds** contract in the following folder:

```
~/workspace/fabric-getting-started/fabric-samples/fabcar-complete/packages/
```

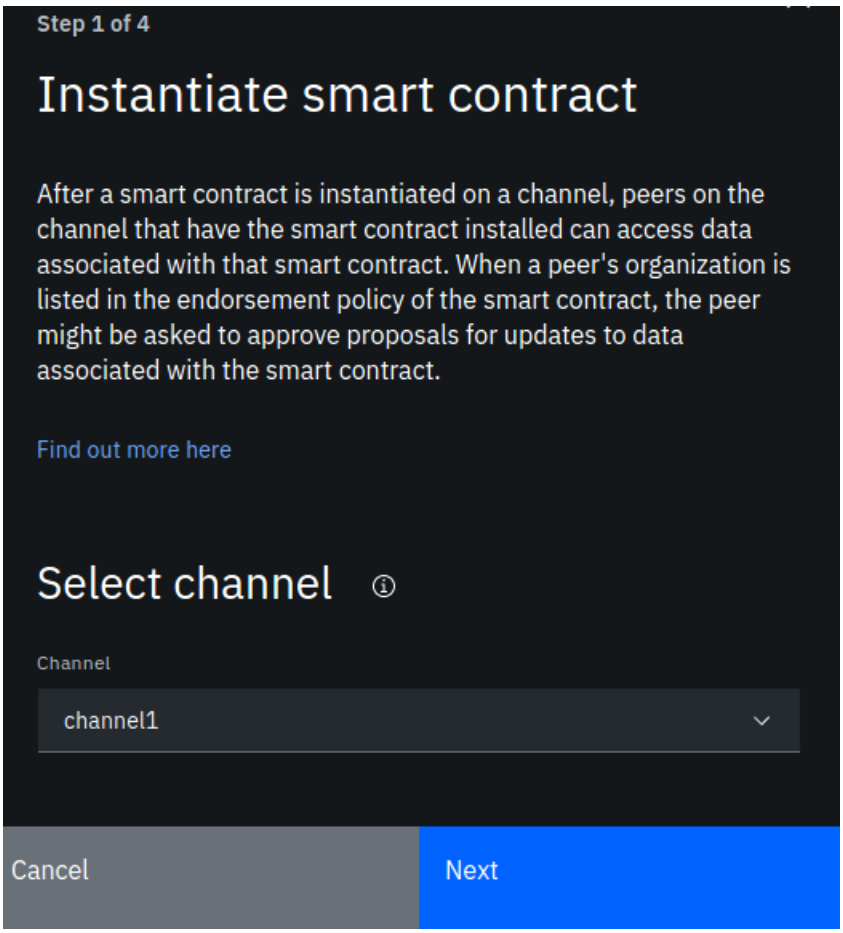
Once you have chosen the file, click "**Open**" and then click on the "**Install smart contract**" button:



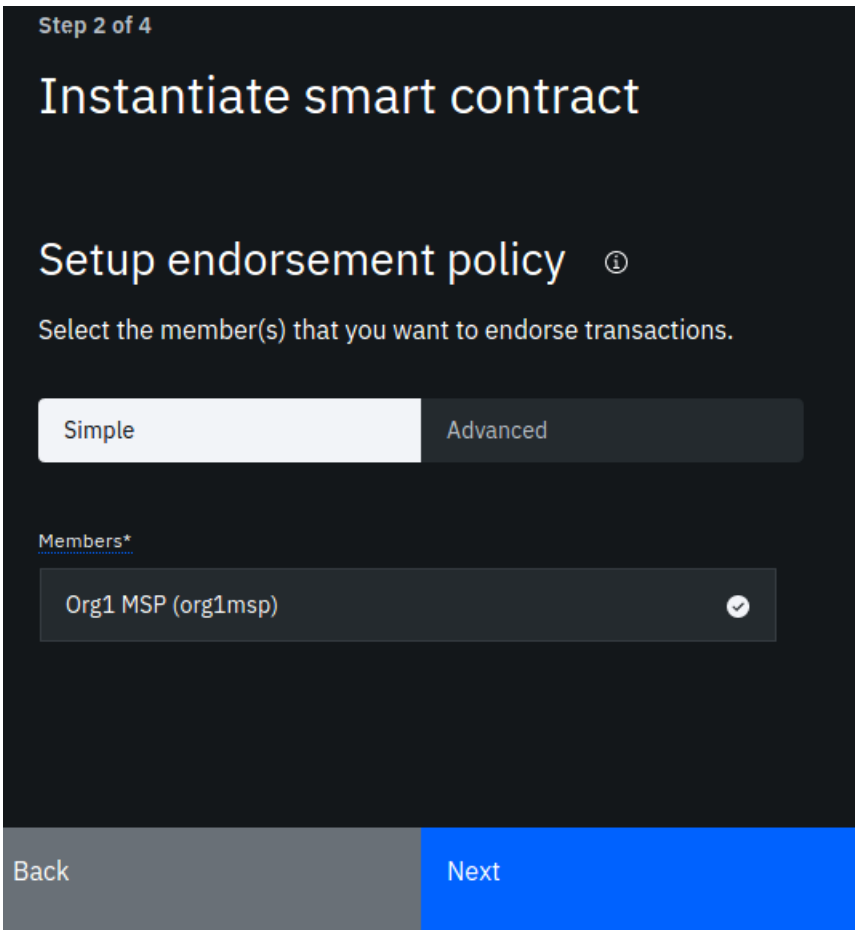
__ **18.** On the same smart contracts page, find the **fabcar** contract we just installed and click on the vertical "**...**" button to the right and choose the "**Instantiate**" option:



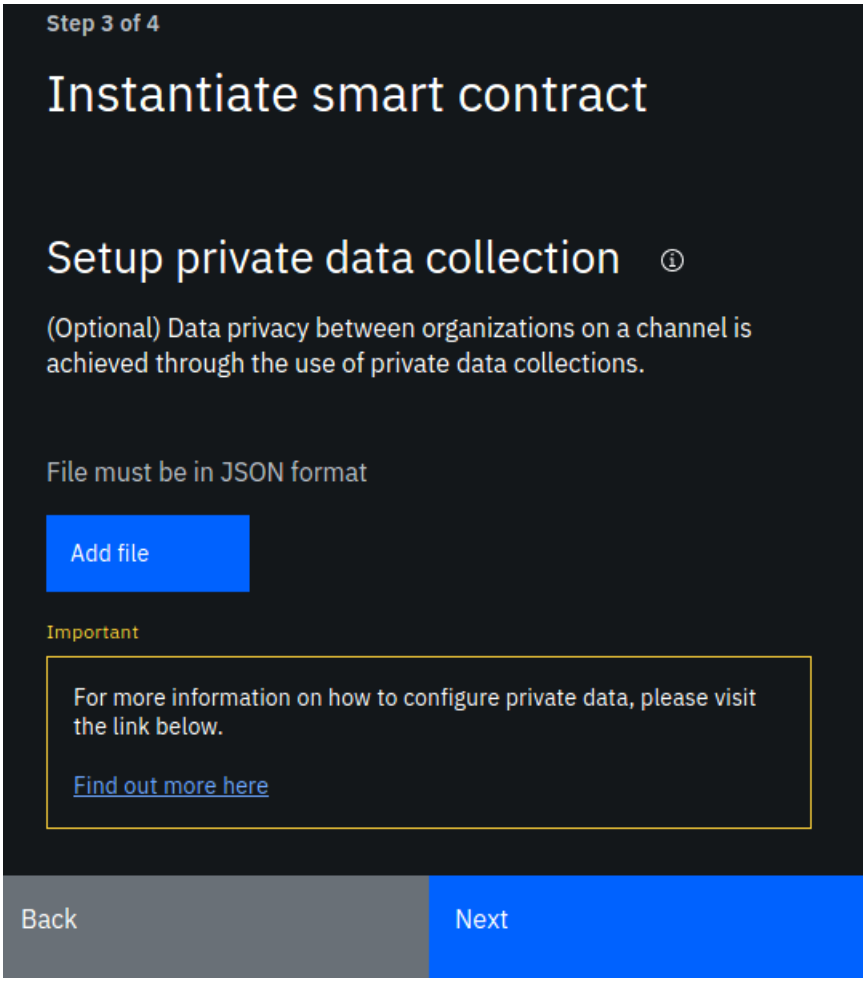
__ **19.** In the side panel, on **Step 1**, make sure you select the channel you joined earlier in the lab, **channel1**, and click **Next**:



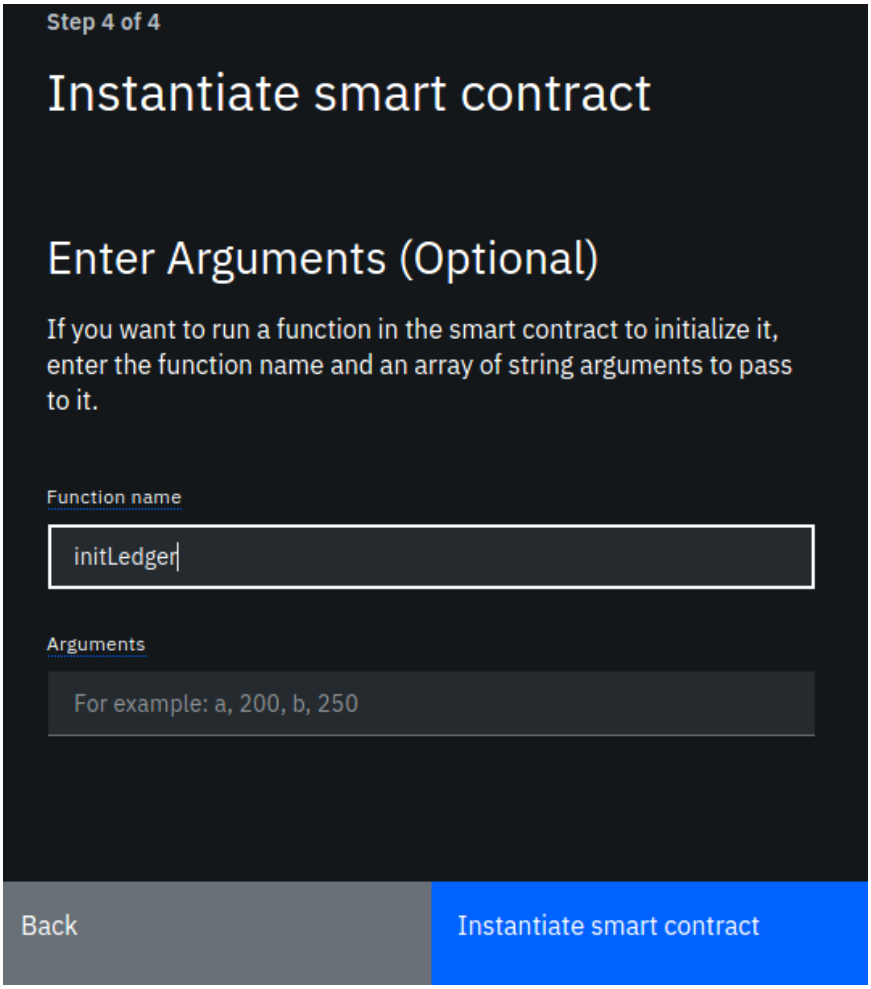
__ **20.** In the side panel on **Step 2**, make sure **org1msp** is selected (ticked), then click **Next**:



__ **21.** 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**":

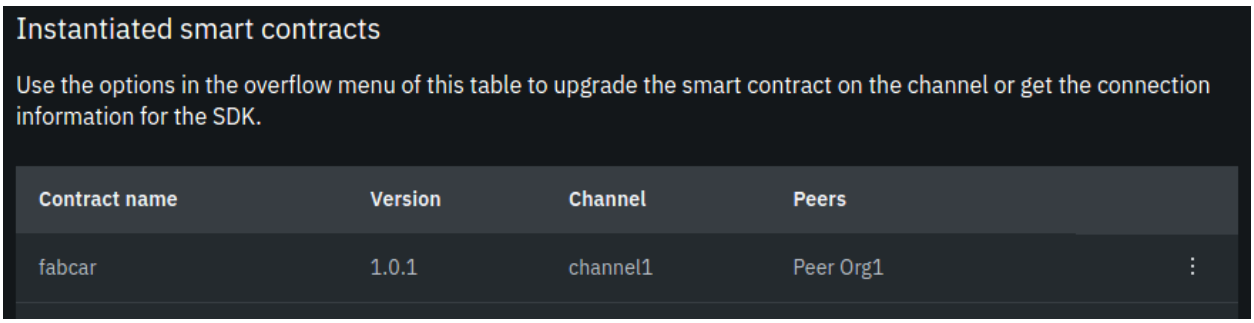


__ **22.** In the side panel on **Step 4**, enter "**initLedger**" as the **Function name** to call which populates the ledger with a selection of sample cars. Leave the **Arguments** field empty and click "**Instantiate smart contract**":

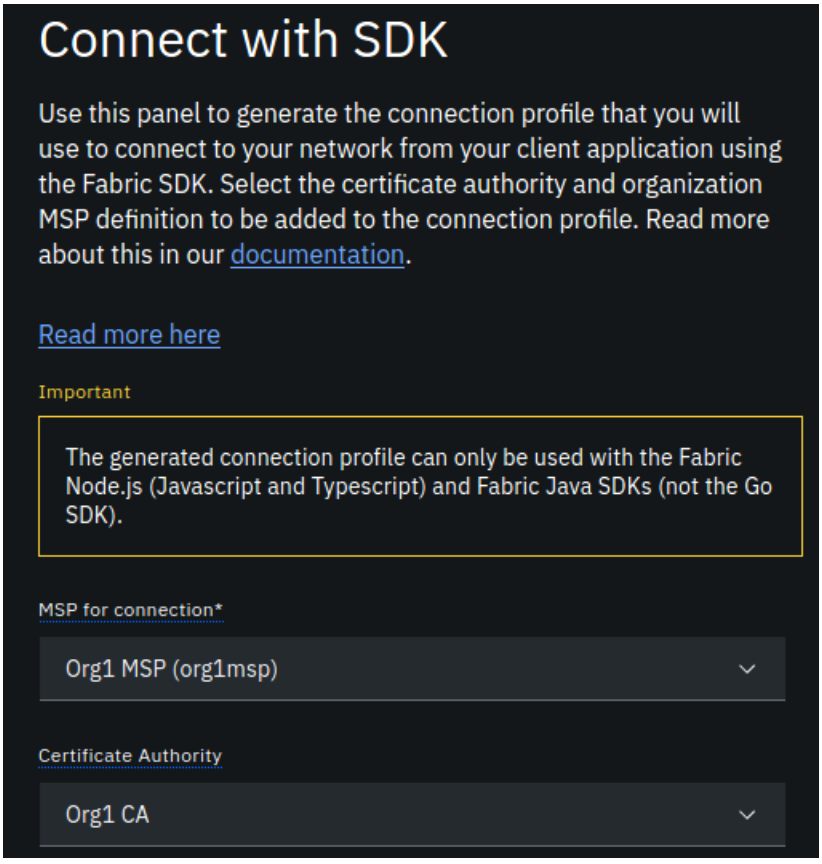


This step can take a few minutes to complete, please be patient.

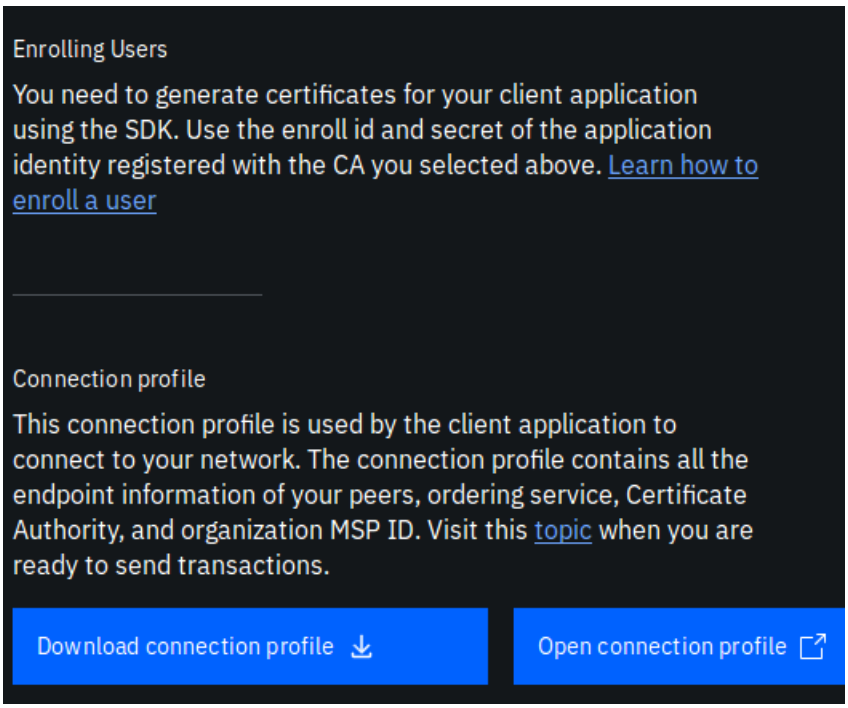
__ **23.** On the same smart contracts page, scroll down to the "**Instantiated smart contracts**" section find the **fabcar** contract we just instantiated and click on the vertical "**...**" button to the right and choose the "**Connect with SDK**" option:



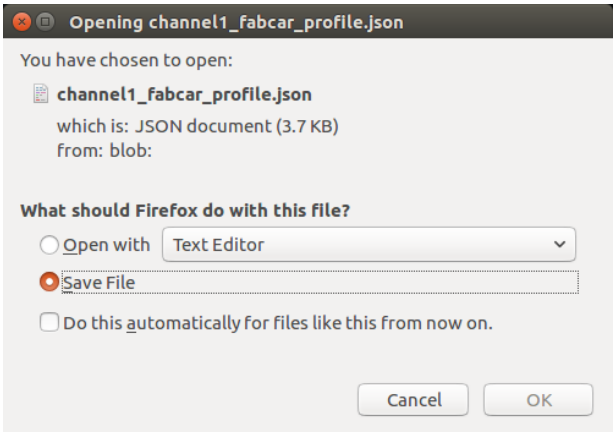
__ **24.** In the side panel select your MSP for connection and Certificate Authority. For **Org1** this will be "**org1msp**" and "**Org1 CA**".



__ **25.** In the side panel scroll down and choose the "**Download connection profile**" button:



__ **26.** In the dialog, choose the "**Save File**" option and click "**OK**":

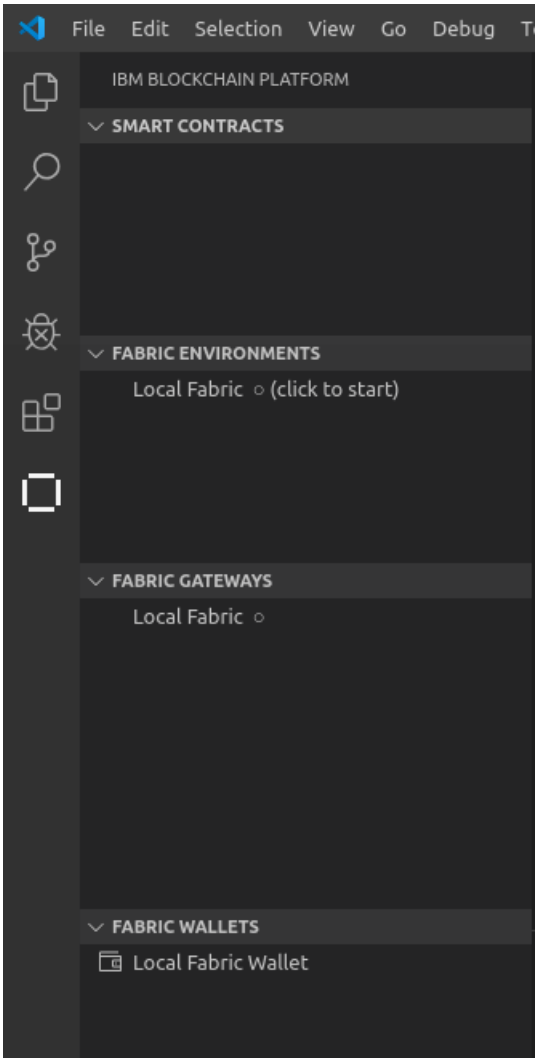


This will place the file "**channel1_fabcar_profile,json**" into the your **Downloads** folder. The path to this file is "**~/Downloads/channel1_fabcar_profile.json**".

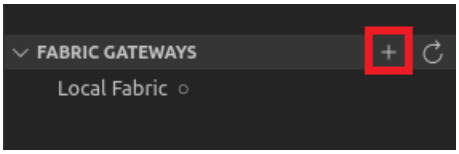
__ **27.** Once you have downloaded the connection profile, click the "**Close**" button.

2.4 Connecting to the Network

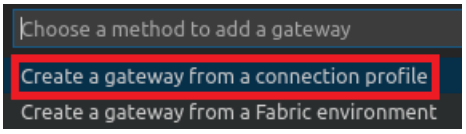
__ **28.** Open a new empty VS Code window and click on the **IBM Blockchain Platform** icon:



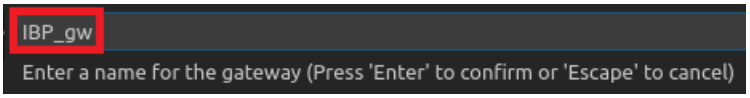
__ **29.** Move your mouse over the "**Fabric Gateways**" pane to make the "**+**" appear and click the "**+**" to start creating a new gateway:



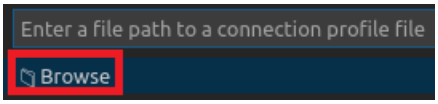
__ **30.** In the pop up, choose the "**Create a gateway from a connection profile**" option:



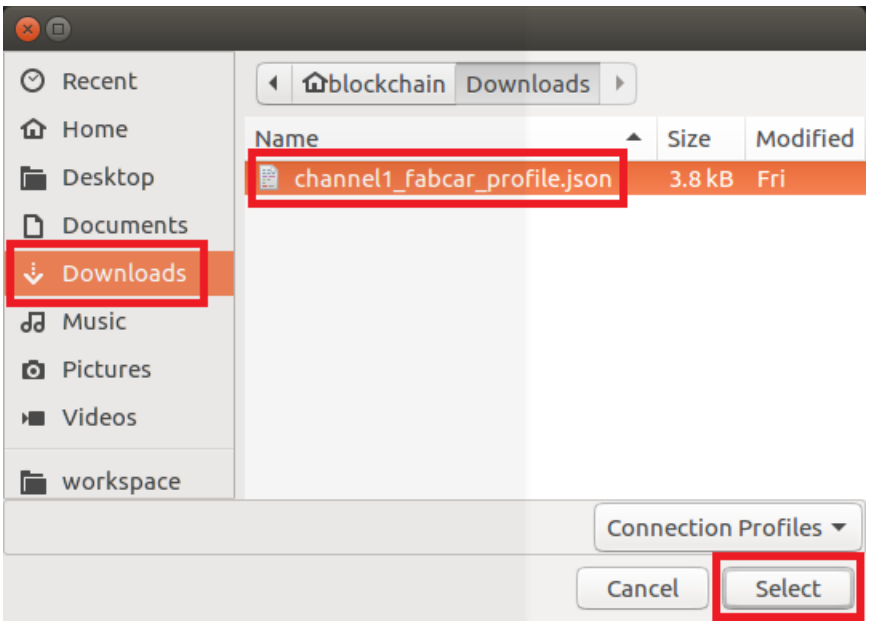
__ **31.** In the pop up enter "**IBP_gw**" as the name:



__ **32.** In the pop up click "**Browse**":

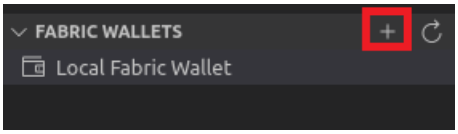


__ **33.** In the dialog click "**Downloads**" on the left, choose the "**channel1_fabcar_profile.json**" file then click "**Select**":

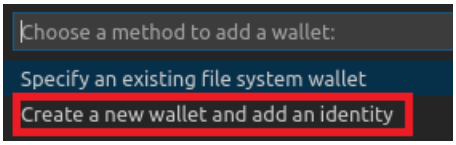


The Fabric Gateways will update with the new gateway, but before we can use it to connect we first need to create a new wallet with a new user id in it.

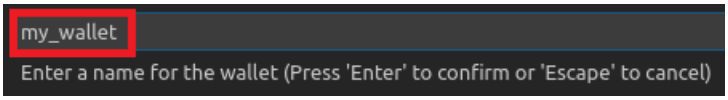
__ **34.** Move your mouse over the "**Fabric Wallets**" pane to make the "**+**" appear and click the "**+**" to start creating a new wallet:



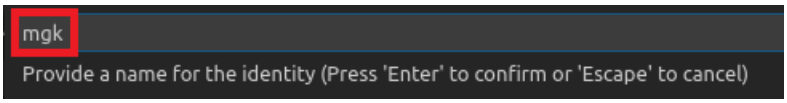
__ **35.** In the pop up choose "**Create a new wallet and add an identity**":



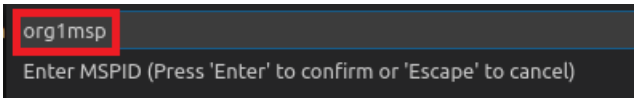
__ **36.** In the pop up enter the name "**my_wallet**" and press enter:



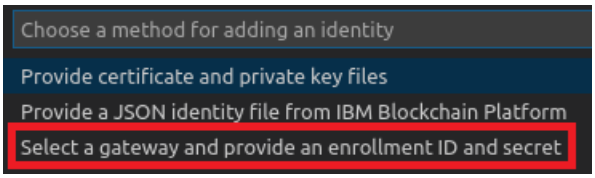
__ **37.** In the pop up enter your own name or initials (with no spaces or special characters) and press enter:



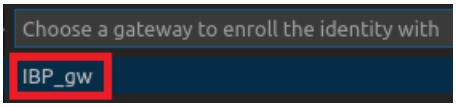
__ **38.** Enter your MSPID. For Org1 this is "**org1msp**", then press enter:



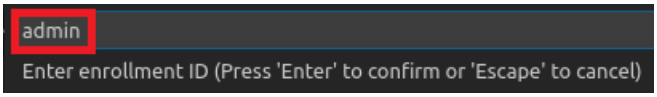
__ **39.** In the pop up choose "**Select a gateway and provide an enrollment ID and secret**":



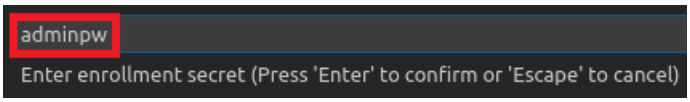
__ **40.** In the pop up choose the "**IBP_gw**" gateway we created earlier:



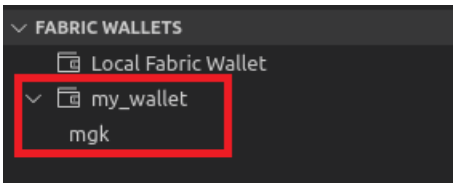
__ **41.** In the pop up enter the CA enrollment ID we created earlier "**admin**":



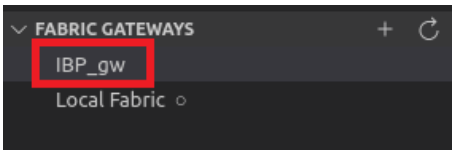
__ **42.** In the pop up enter the CA enrollment secret we created earlier "**adminpw**":



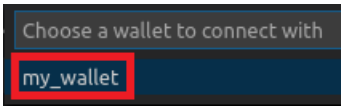
There will be a couple of Information Messages and then the wallets pane will update to show the new wallet with the new ID inside it:



__ **43.** From the "**Fabric Gateways**" view select the "**IBP_gw**" gateway we created earlier:

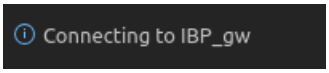


__ **44.** In the pop up choose the "**my_wallet**" wallet we just created:



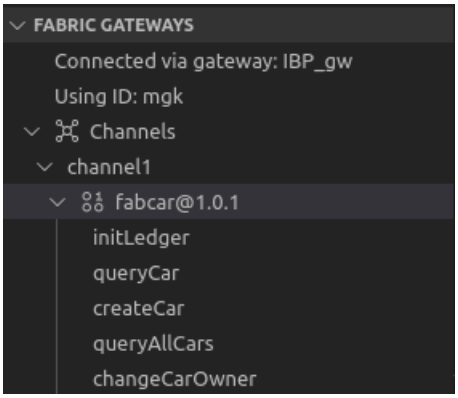
Note: It will not ask you to select the ID inside the wallet as we only have one ID. If we had more than one, there would be an extra step to choose the ID to use as well.

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.

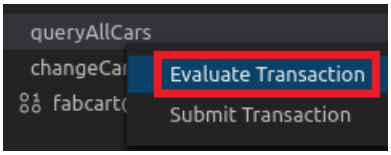
__ **45.** If you expand the **channel1** channel and the **fabcar@1.0.1** smart contract you should be able to see the transactions available in the contract:



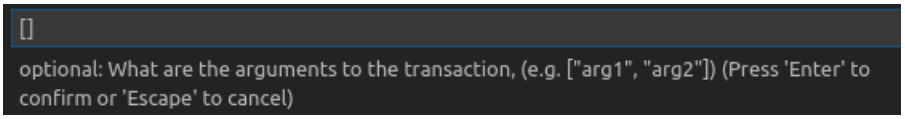
2.5 Issuing Transactions

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

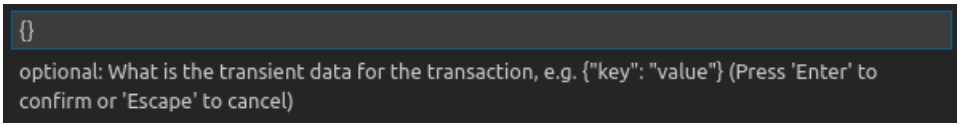
__ **46.** Right click on the "**queryAllCars**" transaction and choose "**Evaluate Transaction**":



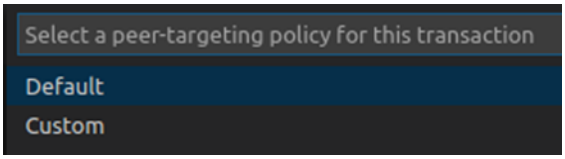
__ **47.** In the pop up just press enter as **queryAllCars** does not need any parameters:



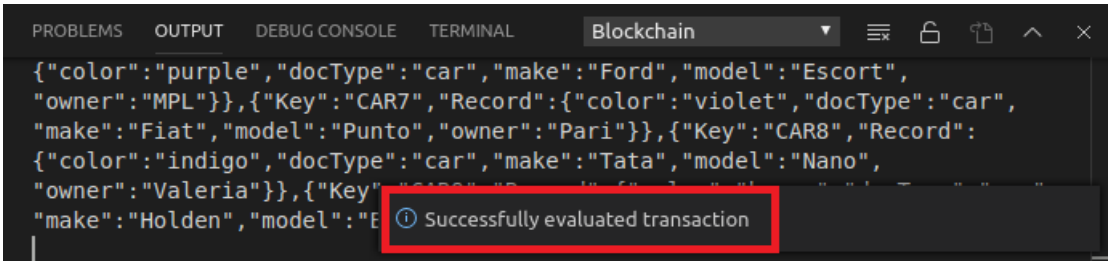
__ **48.** In the next pop up just press enter as **queryAllCars** does not use any transient data:



__ **49.** If asked to "**Select a peer targeting policy for this transaction**" just press enter to accept the Default policy:

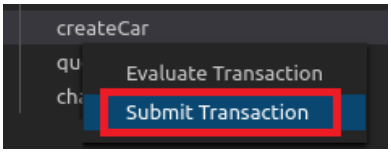


As well as an **Information Message**, you should see the output window update with the details of all the cars that the "initLedger" transaction created when we instantiated the contract:



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.

__ **50.** Right click on the "**createCar**" transaction and choose "**Submit Transaction**":

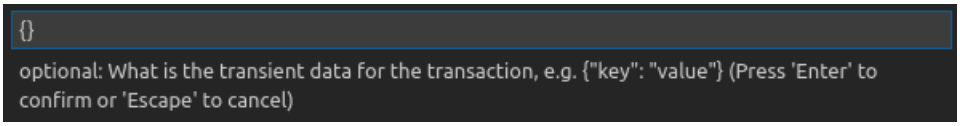


__ **51.** In the pop up enter the parameters for your new car, using a new ID. For example, use **"CAR100"**. Enter some values of your choice like **"CAR100","Tesla","Model S","Red","MGK"** inside the square brackets and press "Enter". The order of the parameters for reference is:

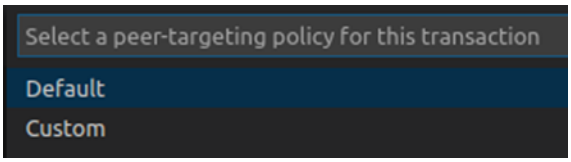
ID, Make, Model, Color, Owner

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.

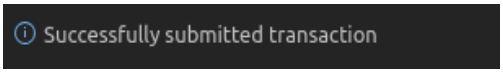
__ **52.** In the next pop up just press Enter as **createCar** does not use any transient data:



__ **53.** If asked to "**Select a peer targeting policy for this transaction**" just press enter to accept the Default policy:



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



__ **54.** Use the **queryCar** transaction to query for your car. For example, you should query for **CAR100** and verify you can see the expected results. You can run the queryCar with **Evaluate Transaction** for a regular query or **Submit Transaction** for an audited query that is recorded on the ledger, and the parameter will be supplied as follows:

```
["CAR100"]
```

__ **55. Congratulations**, you have now finished this Lab. Please use any remaining time to experiment with the environment you have created or have a look at the underlying OKD infrastructure layer which is shown in **Appendix 1** below.

3 Appendix 1: Using OKD and Simple Troubleshooting

The IBP network in this lab is running on OKD. If you want to see how OKD lets you manage pods or if you have any problems when deploying IBP components such as peers, CA's and Orderers, you can use the OKD monitoring console to see any issues.

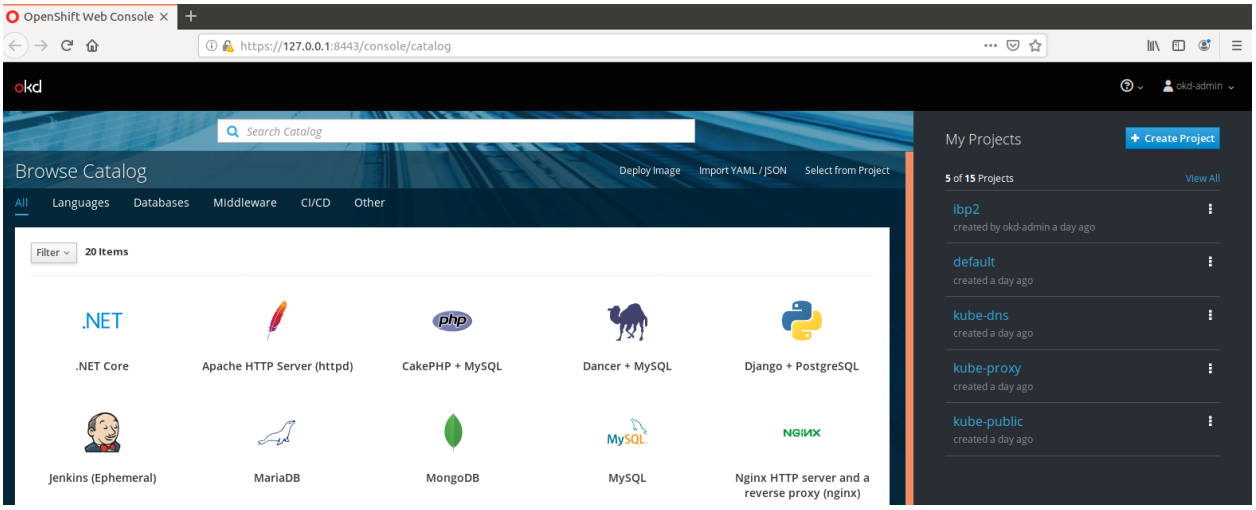
__ **56.** To see the OKD console, you need to switch to the **OpenShift Web Console** tab in Firefox if it is open. If it is closed, you can go to this address to see it:

https://127.0.0.1:8443/console/catalog

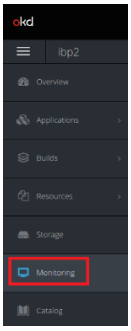
Firefox will show you the OKD console login screen. Enter the Username **okd-admin** and a Password of **12345678** and click "**Log In**":

__ **57.** Firefox will ask you if you wish to save the password, click **Save** to continue.

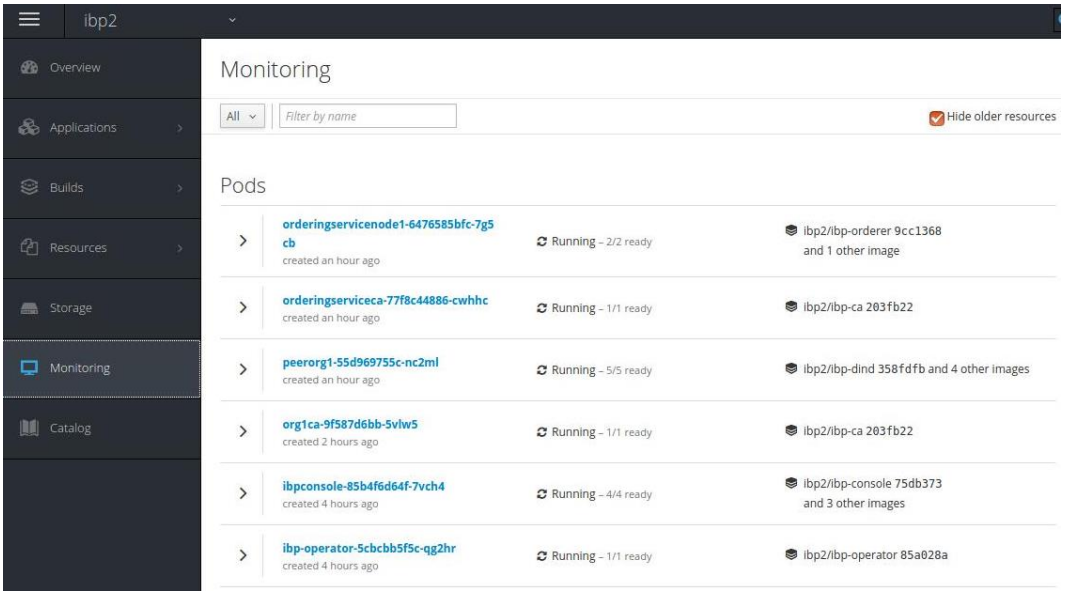
__ **58.** When you are logged in, the main console will appear. On the right-hand side of the console you will see a list of existing projects. Click on the existing "**ibp2**" project:



__ **59.** You will see the "**Overview**" page where currently deployed components are displayed. From this page click on the "**Monitoring**" tab towards the bottom on the left-hand side:



The monitoring page will show you all deployed components and it will also let you look at logs for the components and troubleshoot issues. What you see will depend on what you have deployed, but a completed lab should look similar to this:



IBM Blockchain Platform has two main components, an "**operator**" and a "**console**". The operator's job is to monitor the console and help it perform its duties and you can see examples of these deployed above.

__ **60.** OKD also has comprehensive command line tools available. Some of the useful commands that you could try out in the **terminal** include:

```
oc get deployment -n ibp2
oc get pods -n ibp2
```

4 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**"