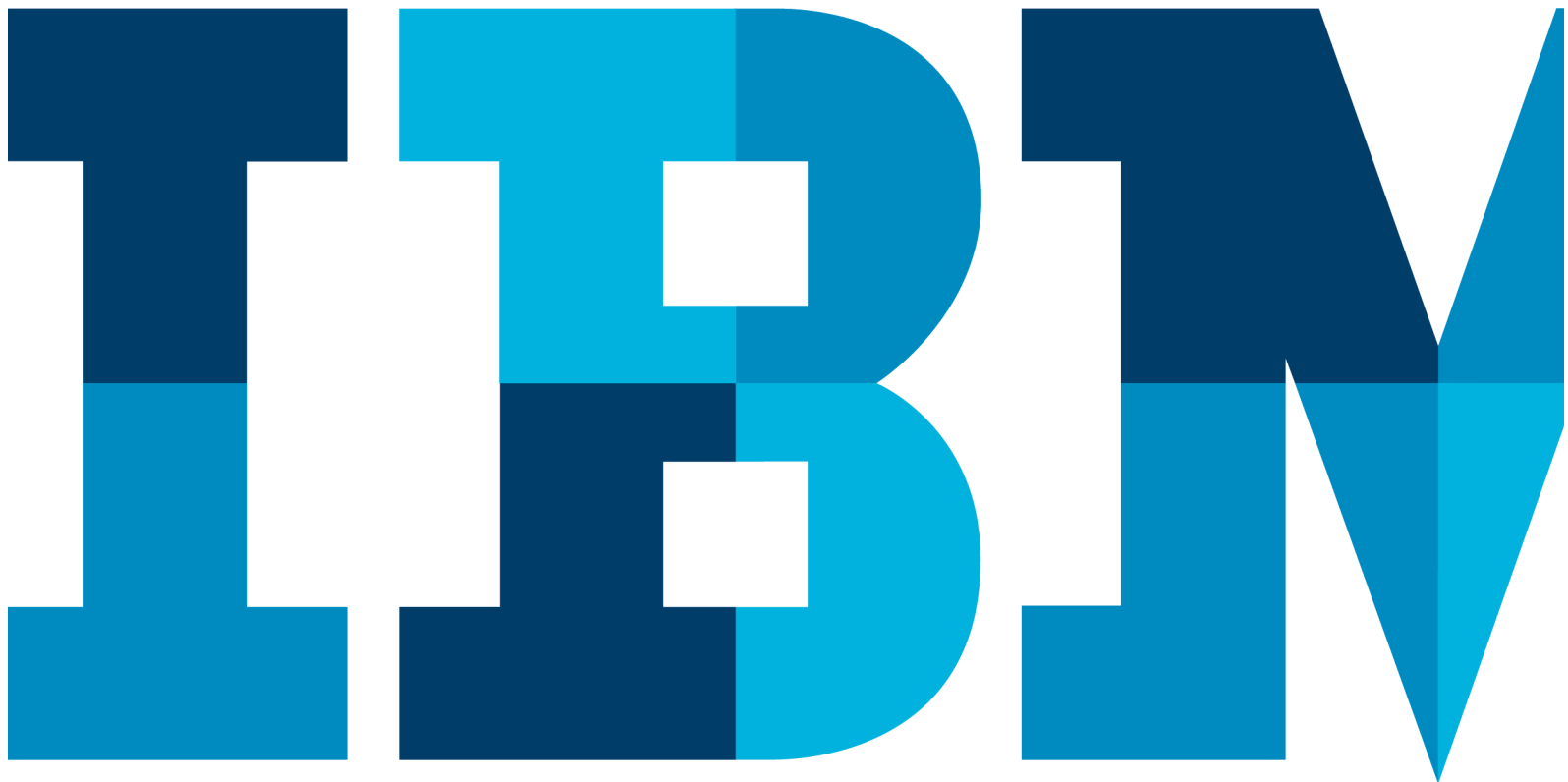


# IBM Blockchain Hands-On Blockchain Explored

*Lab Two – Bluemix – Exercises*



---

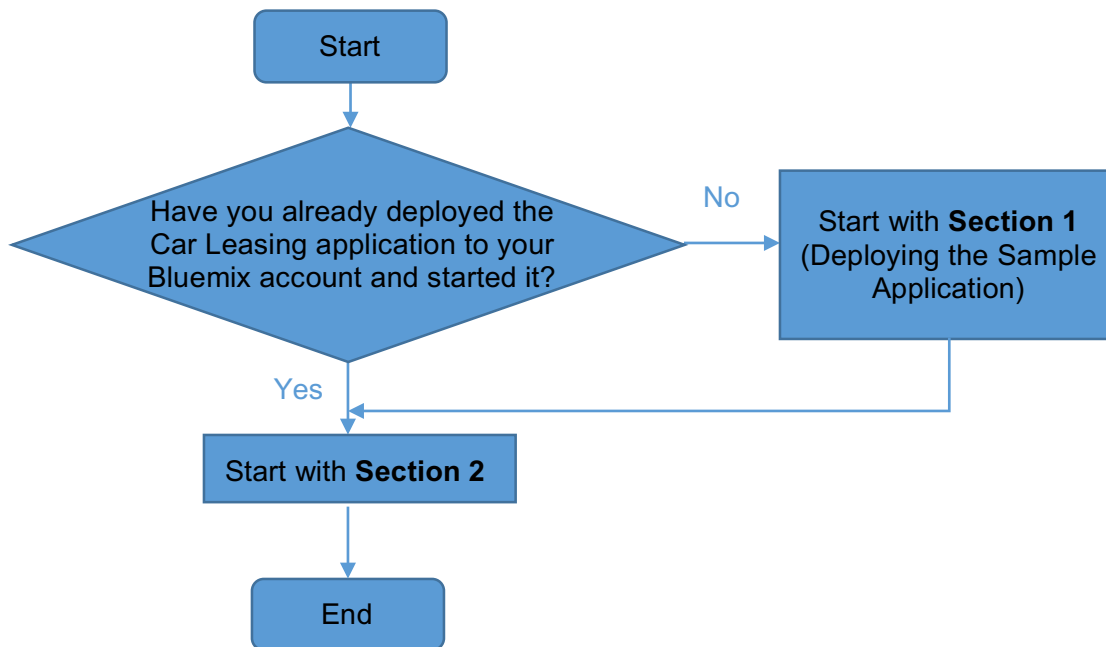
## Contents

<b>OVERVIEW</b>	<b>INTRODUCTION TO THE LAB.....</b>	<b>3</b>
<b>SECTION 1.</b>	<b>DEPLOYING THE SAMPLE APPLICATION .....</b>	<b>4</b>
	1.1. CREATING A BLOCKCHAIN SERVICE.....	4
	1.2. INITIALIZING THE CAR LEASING DEMO .....	6
<b>SECTION 2.</b>	<b>MANAGING THE SAMPLE APPLICATION .....</b>	<b>8</b>
	2.1. VIEWING THE COMPONENTS OF THE BLOCKCHAIN SERVICE .....	8
	2.2. VIEWING THE BLOCKCHAIN EXPLORER .....	10
	2.3. UNDERSTANDING THE BLOCKCHAIN PEERS.....	12
	2.4. INTERACTING WITH THE PEERS.....	13
	2.5. VIEWING THE SERVICE STATUS, SUPPORT CONTACTS AND SAMPLES .....	18
<b>SECTION 3.</b>	<b>REMOVING THE SAMPLE APPLICATION.....</b>	<b>19</b>
<b>APPENDIX A.</b>	<b>NOTICES.....</b>	<b>20</b>
<b>APPENDIX B.</b>	<b>TRADEMARKS AND COPYRIGHTS.....</b>	<b>22</b>

## Overview Introduction to the Lab

The purpose of this lab is to introduce you to the Blockchain service on Bluemix. We will build on the car leasing scenario that was introduced in the “Blockchain Explained” lab.

If you are using your own Bluemix account and have already completed the previous lab, you will have already deployed the car leasing application to your account. You can skip section 1 and re-use your existing application:



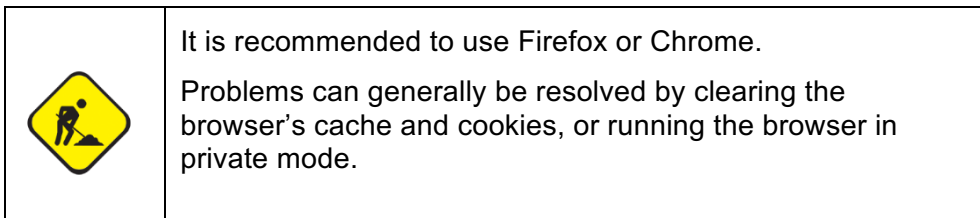
For V0.61 of Hyperledger in Bluemix

## Section 1. Deploying the sample application

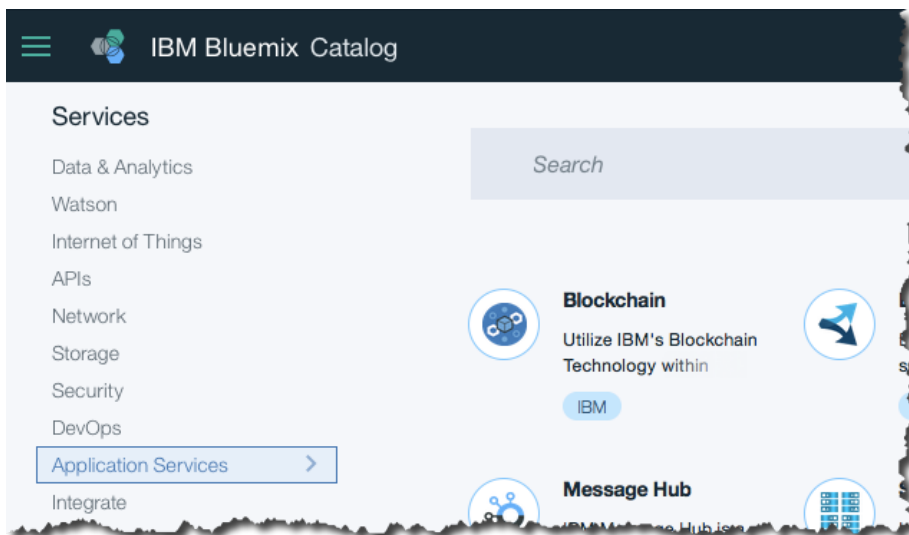
In this section we will use Bluemix to deploy a copy of the car leasing demo application.

### 1.1. Creating a Blockchain Service

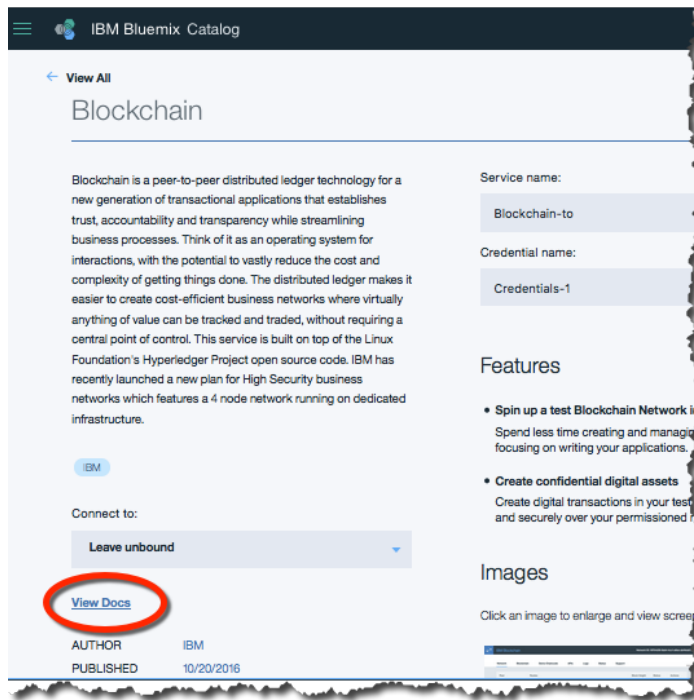
- \_\_1. Open a web browser and go to [www.bluemix.net](http://www.bluemix.net).




- \_\_2. Click '**Sign Up**' or '**Log In**' to create a new Bluemix account or log into your existing account.
- \_\_3. Once you have successfully signed up and logged into Bluemix, select **Catalog** from the top bar.
- \_\_4. In the 'Services' section of the sidebar, click '**Application Services**' and select **Blockchain**.



- \_\_5. Review the service description and information about the service.
- \_\_6. Click '**View Docs**' and learn about the process of creating a blockchain environment.



- \_\_7. Click **'Samples and Tutorials'** on the right of the page.
- \_\_8. Click  against the Car Lease demo. Log in to Bluemix again if necessary.
- \_\_9. If you are asked to pick an alias (usually the first time a given Bluemix ID deploys a sample), come up with a unique username and review and accept the terms of use. Click Create and then Continue on the subsequent page.


**IBM Bluemix™ DevOps Services**


### Pick an alias

To set up your Git repository, we need to associate your IBM id with an alias.

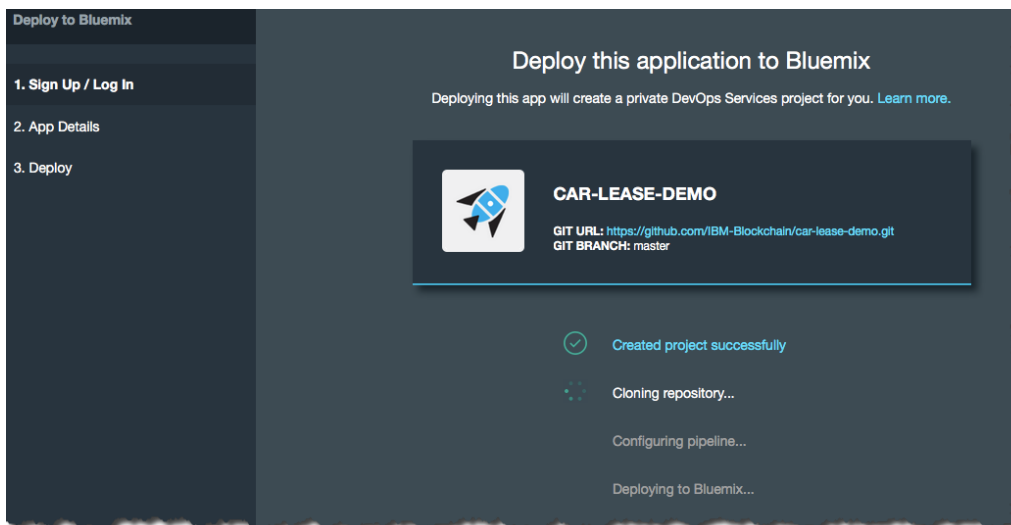
An alias is a unique, publicly visible short name used in Git repository paths, Track & Plan, and desktop and command line clients.



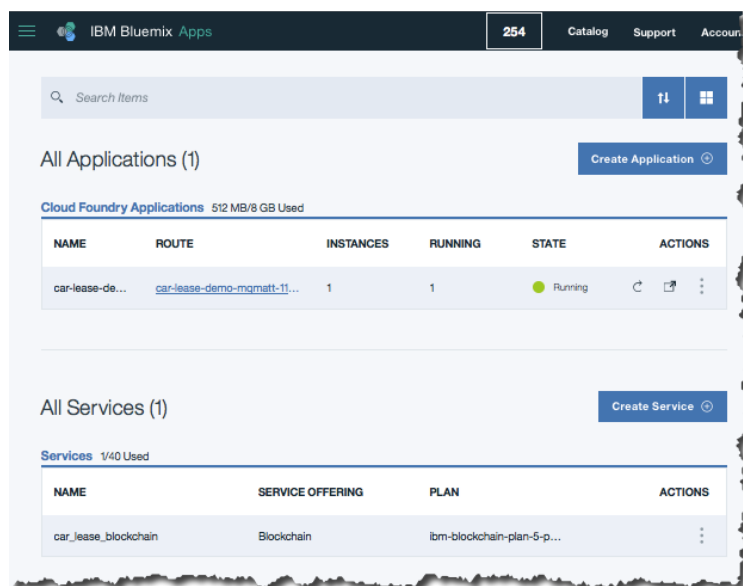
☐ I accept the DevOps Services [Terms of Use](#).

- \_\_10. Wait a few seconds for the default field values to be populated. Then leave the App Name, Region, Organization and Space default (unless told otherwise) and click .

This will cause the car leasing demo to be deployed into your Bluemix environment, and will probably take a few minutes to complete.



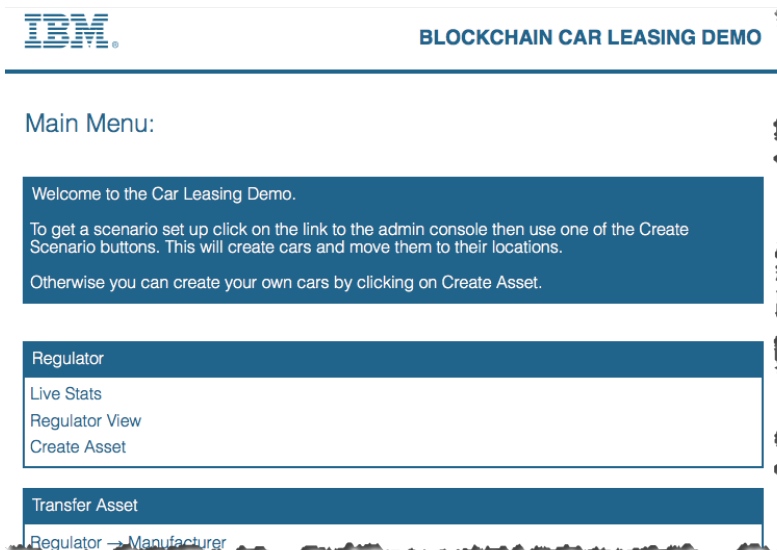
- \_\_\_11. Once you see the 'Success!' message click **DASHBOARD** to see a description of the new car leasing application (and associated Blockchain service) you created.



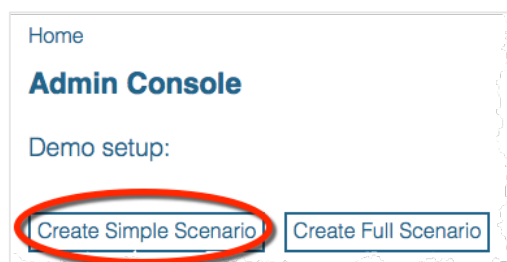
## 1.2. Initializing the Car Leasing Demo

- \_\_\_12. Click the blue hyperlink under the 'Route' column of your application, and this will load the Car Leasing demo webpage. (Do not click elsewhere on this line, as this will load the administration interface for the application, which we will look at later).

You will now see the front page of the car leasing demo.



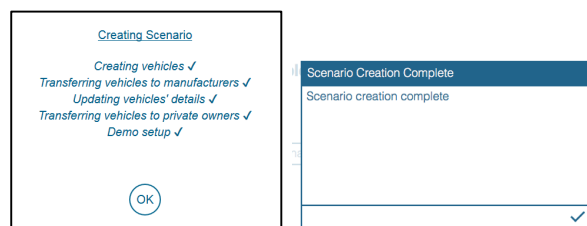
- \_\_\_13. From the Car Leasing demo front page, click '**Admin Console**' and '**Create Simple Scenario**'.



This will preload the blockchain with a set of transactions. (The Full Scenario works fine too; the difference between the Simple Scenario and the Full Scenario is that in the Full Scenario more assets are initially loaded onto the blockchain; this takes longer to initialize, however.)

Wait a few moments for the initialization to complete.


- \_\_\_14. Click '**OK**' to close the Creating Scenario log, and then dismiss the 'Scenario Creation complete' by clicking the check mark.

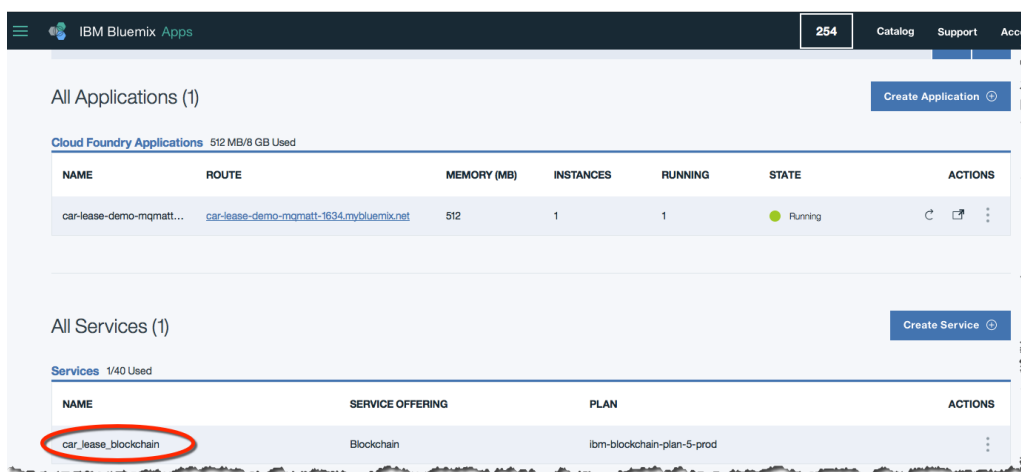


## Section 2. Managing the sample application

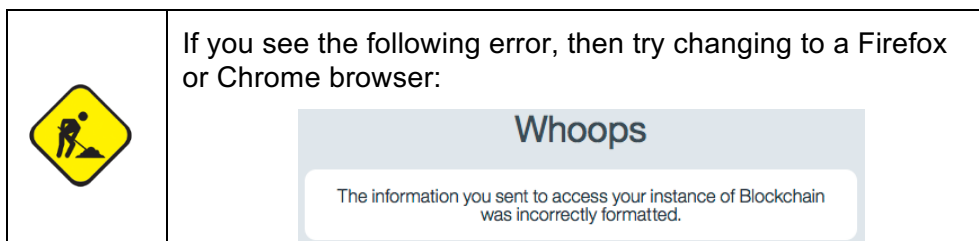
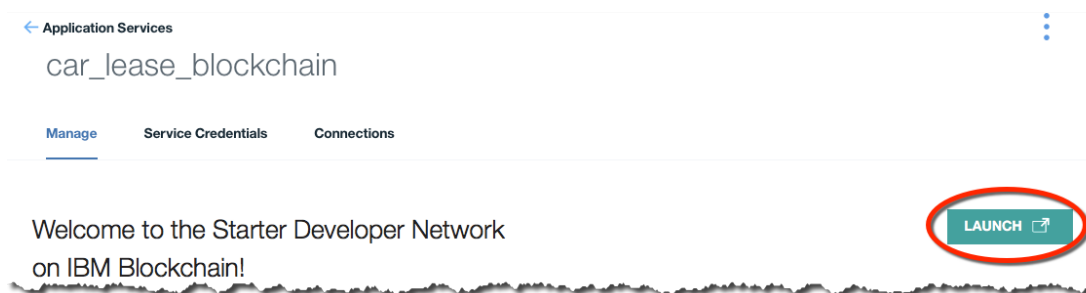
In this section we will use the monitoring tools available inside the Bluemix environment to view and manage the blockchain.

### 2.1. Viewing the components of the Blockchain service

- \_\_15. Return to the Bluemix dashboard, either by selecting the  icon in the top left of Bluemix and selecting 'Dashboard' or by going directly to <https://console.ng.bluemix.net/dashboard/applications>.
- \_\_16. Click on the **car\_lease\_blockchain** service in the Services section of the dashboard.

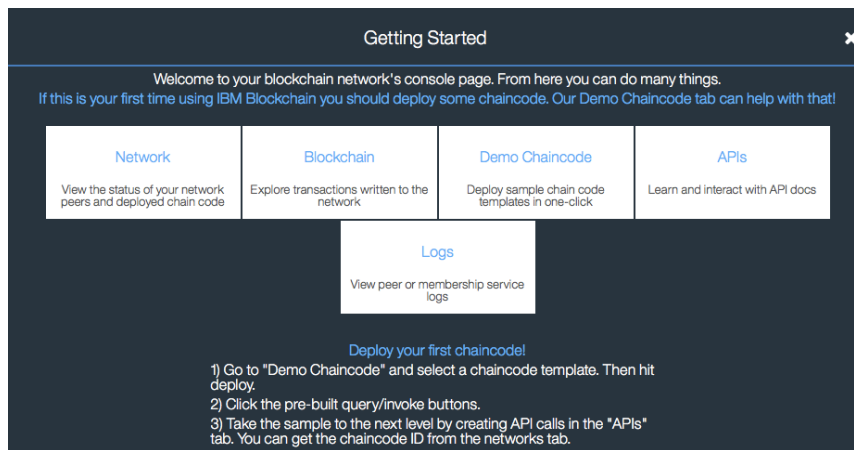


- \_\_17. Review the details and select  to launch the blockchain administration console.

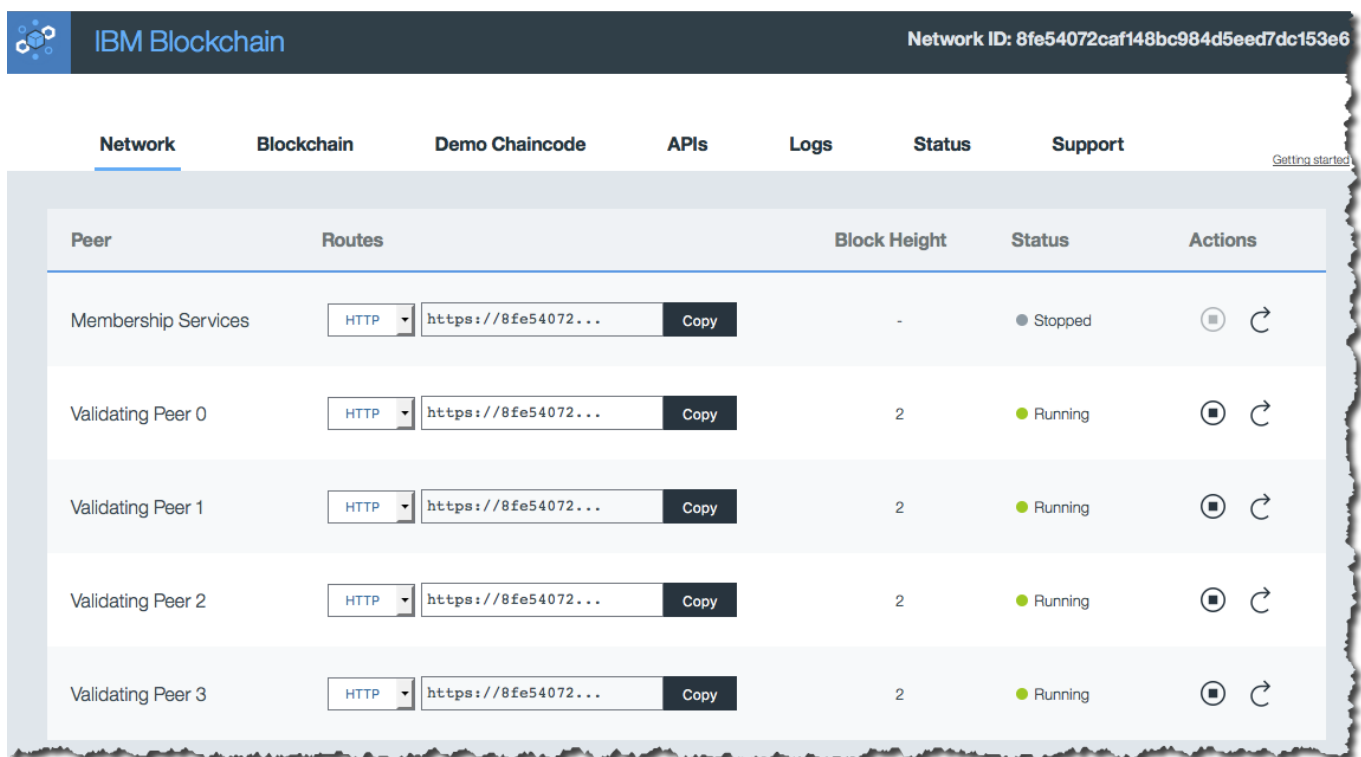




- \_\_\_18. Close the pop-up showing information about the sections. We'll look at these sections in more detail throughout this lab.



You should now see the administration page with seven tabs along the top. The **'Network'** tab will be selected.



This view confirms that four validating peers and a certificate authority (Membership Services) are running under your Blockchain service. The block height should be the same for each of the validating peers.

## 2.2. Viewing the Blockchain Explorer

The Blockchain tab shows a visual representation of the state of the Blockchain.

\_\_\_19. Click the '**Blockchain**' tab at the top of the page.



The icons show:

<b>8</b> Blocks all time	Total number of blocks in the chain
<b>0.1</b> blks/hour last 8 blocks	Average number of blocks per hour
<b>1.0</b> trans/blk last 8 blocks	Average number of transactions per block
<b>3</b> Deployments last 8 blocks	Number of deployment calls made to deploy chaincode
<b>4</b> Invocations last 8 blocks	Number of invoke requests made within this blockchain

Each block contains a set of transactions. In Hyperledger, a transaction is the record of the request to interact with chaincode (a smart contract). Two important transaction types are:

- **DEPLOY:** The request to deploy a piece of chaincode across all validating peers, so that it can be executed at a later date.
- **INVOKE:** The request to invoke a piece of chaincode (for example, invoke the chaincode to transfer the ownership of a car)

Other request types exist (e.g. query, update, terminate). Not all request types are recorded on the blockchain.



The blocks also include when that block was committed to the blockchain.

\_\_20. Click on a block that contains at least one invocation request.


\_\_21. Look through the list of transactions that are contained within the block.

DATE	TYPE	UUID	CHAINCODE ID	PAYLOAD
07/11 01:53pm UTC	INVOKE	9f973904-e178-45b3-8dbb-6cb7ae85e25c	f89c0bc12...	private_to_lease_companyLeaseCan SF1764514

Each line of information is a transaction stored within the block. A block may contain multiple transactions but in this demo there will often only be one transaction per block due to the low frequency of transactions being made. The information displayed is:

<b>Date</b>	The date the transaction was submitted.
<b>Type</b>	The type of transaction taking place (e.g. INVOKE or DEPLOY).
<b>UUID</b>	The unique identifier for each transaction.
<b>Chaincode ID</b>	Refers to the chaincode that is being invoked or deployed.
<b>Payload</b>	The input parameters to the chaincode.

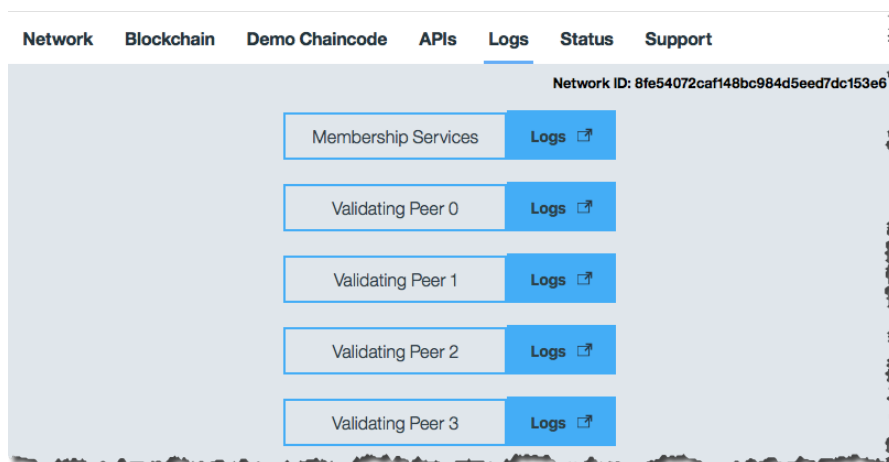
\_\_22. Repeat this for other blocks to understand how the transactions are stored.

	<p>When the Blockchain service is initialised for the car leasing application, the first two blocks in the chain usually contain 'Deploy' transactions, where the chaincode is deployed to the validating peers.</p> <p>View these blocks If you're willing to scroll down the Blockchain explorer that far!</p>
---	--

## 2.3. Understanding the Blockchain Peers

We are now going to review the logs associated with the peers. This is useful for understanding how the blockchain works, and for diagnosing problems.

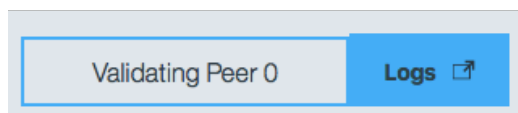
\_\_23. Click on the '**Logs**' tab at the top of the service page.



Here we can see that this Blockchain service contains four validating peers and a Certificate Authority.

By looking at the logs for each peer you can verify that every node has executed every transaction.

\_\_24. Click the Logs button against one of the validating peers.



This will show the logs for the selected peer in a new window.

```

ERR - 2016/05/26 13:46:28 Peer address: 79a7456b-58ed-4bfb-9037-5c05f34fdbba_vp1-discovery.blockchain.ibm.com
ERR - 2016/05/26 13:46:28 Yes, TLS is enabled
ERR - 2016/05/26 13:46:28 os.Args returns: [/go/bin/c6elf96377aac59bb0985b26ad78a8958dcdca7db5702ff4882fa]
ERR - 2016/05/26 13:46:28 Registering.. sending REGISTER
ERR - 2016/05/26 13:46:28 Chaincode Keepalive Time is
ERR - 2016/05/26 13:46:28 []Received message REGISTERED from shim
ERR - 2016/05/26 13:46:28 []Handling ChaincodeMessage of type: REGISTERED(state:created)
ERR - 2016/05/26 13:46:28 Received REGISTERED, ready for invocations
ERR - 2016/05/26 13:46:29 [c6elf963]Received message INIT from shim
ERR - 2016/05/26 13:46:29 [c6elf963]Handling ChaincodeMessage of type: INIT(state:established)
ERR - 2016/05/26 13:46:29 Entered state init
ERR - 2016/05/26 13:46:29 [c6elf963]Received INIT, initializing chaincode
ERR - 2016/05/26 13:46:29 [c6elf963]Inside putstate, isTransaction = true
ERR - 2016/05/26 13:46:29 [c6elf963]Sending PUT_STATE
ERR - 2016/05/26 13:46:29 [c6elf963]Received message RESPONSE from shim
ERR - 2016/05/26 13:46:29 [c6elf963]Handling ChaincodeMessage of type: RESPONSE(state:init)
ERR - 2016/05/26 13:46:29 [c6elf963]before send
ERR - 2016/05/26 13:46:29 [c6elf963]after send
ERR - 2016/05/26 13:46:29 [c6elf963]Received RESPONSE, communicated (state:init)
ERR - 2016/05/26 13:46:29 [c6elf963]Received RESPONSE. Successfully updated state
ERR - 2016/05/26 13:46:29 [c6elf963]Init succeeded. Sending COMPLETED
ERR - 2016/05/26 13:46:29 [c6elf963]Move state message COMPLETED
ERR - 2016/05/26 13:46:29 [c6elf963]Handling ChaincodeMessage of type: COMPLETED(state:init)
ERR - 2016/05/26 13:46:29 [c6elf963]send state message COMPLETED
ERR - 2016/05/26 13:46:29 [5852a903]Received message QUERY from shim
ERR - 2016/05/26 13:46:29 [5852a903]Handling ChaincodeMessage of type: QUERY(state:ready)
ERR - 2016/05/26 13:46:29 [5852a903]Sending GET_STATE
ERR - 2016/05/26 13:46:29 [5852a903]Received message RESPONSE from shim
ERR - 2016/05/26 13:46:29 [5852a903]Handling ChaincodeMessage of type: RESPONSE(state:ready)

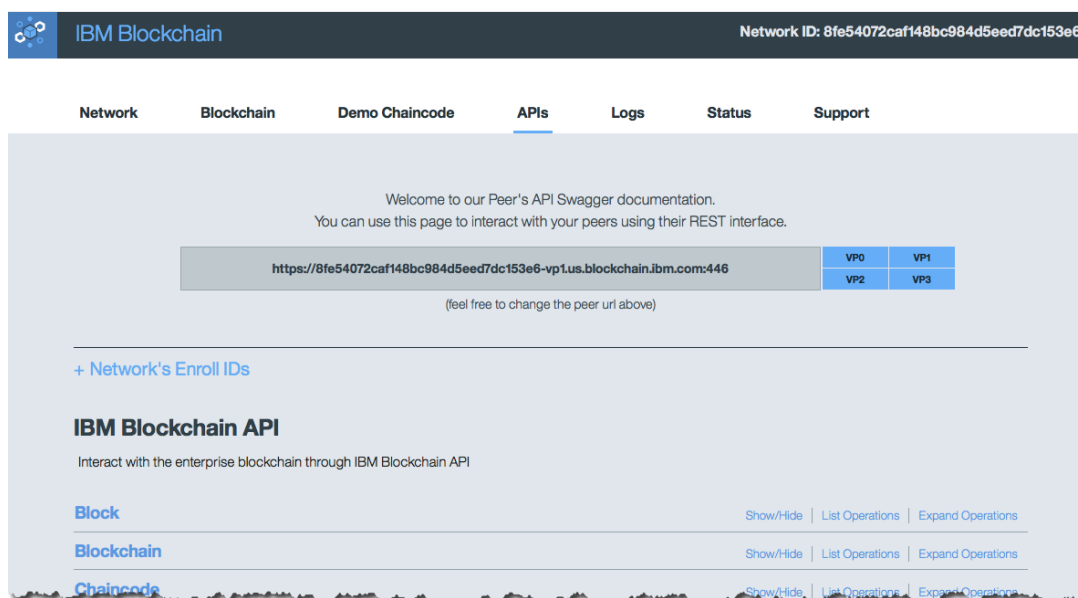
```

## 2.4. Interacting with the peers

It is possible to invoke the management APIs that interact directly with the peers. In this section we will be trying out these APIs directly from the Bluemix environment.

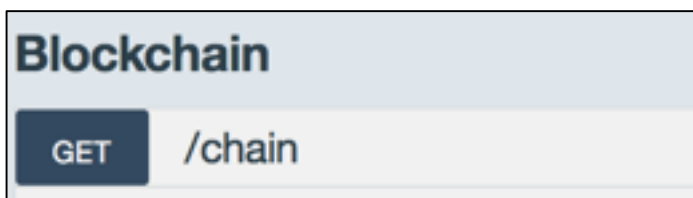
Note that the APIs concern *operationally managing* the Blockchain service – this is not the same as adding and invoking transactions through chaincode!

25. Click on the **'APIs'** tab on the service page.



This page allows you to invoke APIs that will directly interrogate and manage the blockchain. First we will use the API interface to query the height of the Blockchain (the number of blocks).

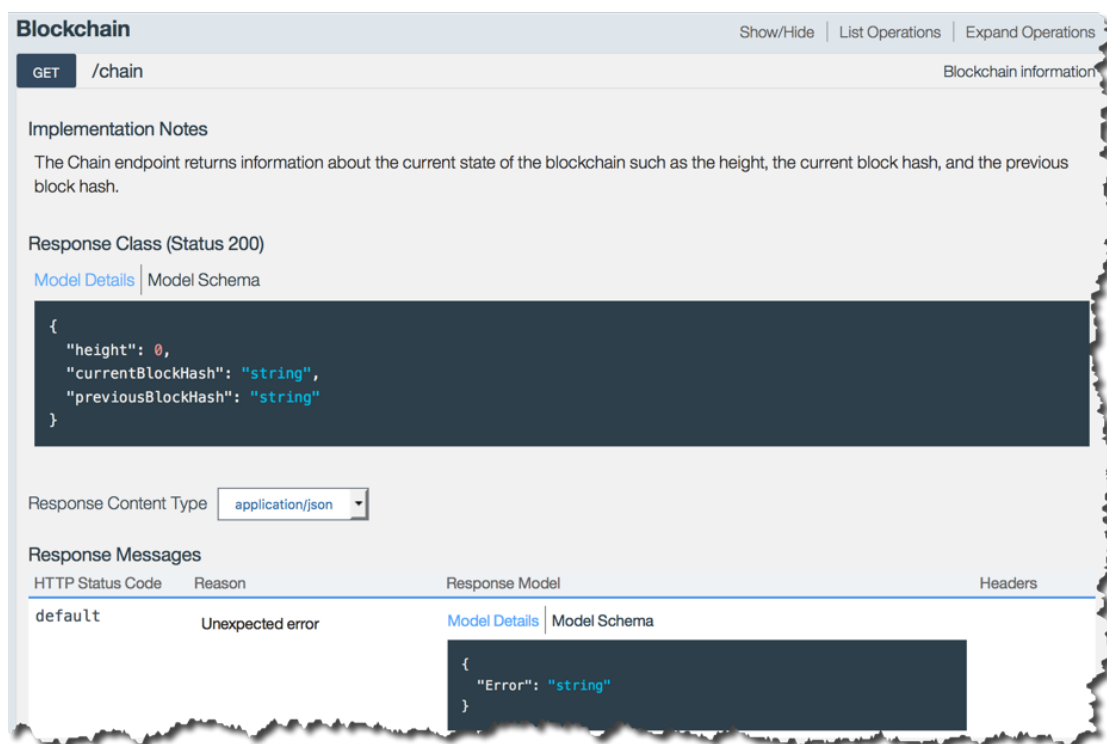
\_\_26. Click the '**Blockchain**' section.



This reveals the **GET /chain** operation which is a valid method to call on the peer.

\_\_27. Click '**Expand Operation**' to view information about this API.

This reveals the input and output data formats.



\_\_28. Click '**Try It Out**' to invoke the API.



Review the displayed fields:

- The *Request URL* shows the URL that was invoked, including the endpoint information of the peer (hostname:port) and the method call (/chain).
- The *Response Body* shows the information that was returned including, importantly, the height of the blockchain.
- The *Response Code* 200 shows that the request was successful.
- The *Response Headers* confirms that the response body has been returned in a JSON data structure.

\_\_\_29. Expand the '**Block**' section and review the information on how to interrogate an individual block in the Blockchain.

**Block** Show/Hide List Operations Expand Operations

**GET** `/chain/blocks/{Block}` Individual block information

**Implementation Notes**

The {Block} endpoint returns information about a specific block within the Blockchain. Note that the genesis block is block zero.

**Response Class (Status 200)**

[Model Details](#) | [Model Schema](#)

```
{
  "proposerID": "string",
  "timestamp": {
    "seconds": 0,
    "nanos": 0
  },
  "transactions": [
    ]
}
```

- \_\_\_30. Fill in the '**Block**' parameter to be a number less than the height of the chain and click '**Try it out!**'.

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
<b>Block</b>	<input type="text" value="40"/>	<b>Block number to retrieve</b>	path	integer

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
default	Unexpected error	<a href="#">Model Details</a>   <a href="#">Model Schema</a> <pre>{   "Error": "string" }</pre>	

[Try it out!](#)



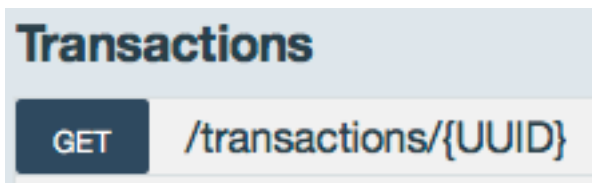
### \_\_31. Review the information returned in the Response Body.



Block content description:

<b>transactions</b>	An array of transactions stored in the block.
<b>type</b>	<p>This shows the possible values:</p> <ol style="list-style-type: none"> <li>0. Undefined</li> <li>1. Deployment</li> <li>2. Invoke</li> <li>3. Query</li> </ol>
<b>chaincodeID</b>	ID of the chaincode that was invoked or deployed.
<b>payload</b>	Input parameters to the chaincode.
<b>uuid</b>	Unique identifier of this transaction.
<b>timestamp</b>	Time at which the block or transaction order was proposed.
<b>cert</b>	Certificate of the participant submitting the transaction.
<b>signature</b>	Signature of the participant submitting the transaction.
<b>stateHash</b>	Hash of the world state changes.
<b>previousBlockHash</b>	Hash of the previous block in the chain.
<b>nonHashData</b>	Data stored with the block, but not included in the block's hash. This allows data to be different per peer or discarded without affecting the blockchain.
<b>localLedgerCommitTimestamp</b>	Time the block was added to the ledger on the local peer.

- \_\_\_32. Copy the UUID field of a transaction from a block; this will be of the form “a338564e-ceef-4df6-9efd-95b65fa43efc”.
- \_\_\_33. Click the **‘Transactions’** section.



This reveals the **GET /transactions/{UUID}** operation which is a valid method to call on the peer.

- \_\_\_34. Paste the transaction UUID and click **‘Try it out!’**.

The ‘payload’ field is base64 encoded (use a web tool such as <http://www.base64decode.org> for decoding this information); when decoded you’ll see that the payload includes the chaincode ID of the smart contract being called together with its input parameters. For example:



Note that this application does not encrypt the transactions, so the payloads are visible (albeit base64 encoded) to all.

- \_\_\_35. Now spend some time interacting with the other APIs available to you.

## 2.5. Viewing the Service Status, Support Contacts and Samples

- \_\_\_36. Click on the **‘Status’** tab at the top of the service page.

This page shows you the recent availability of the Blockchain service on Bluemix, and also the version of Hyperledger Fabric that is being used by your network.

- \_\_\_37. Click on the **‘Support’** tab at the top of the service page.

This page shows you how to get more help with IBM Bluemix and the Blockchain service.

- \_\_\_38. Click on the **‘Demo Chaincode’** tab at the top of the service page.

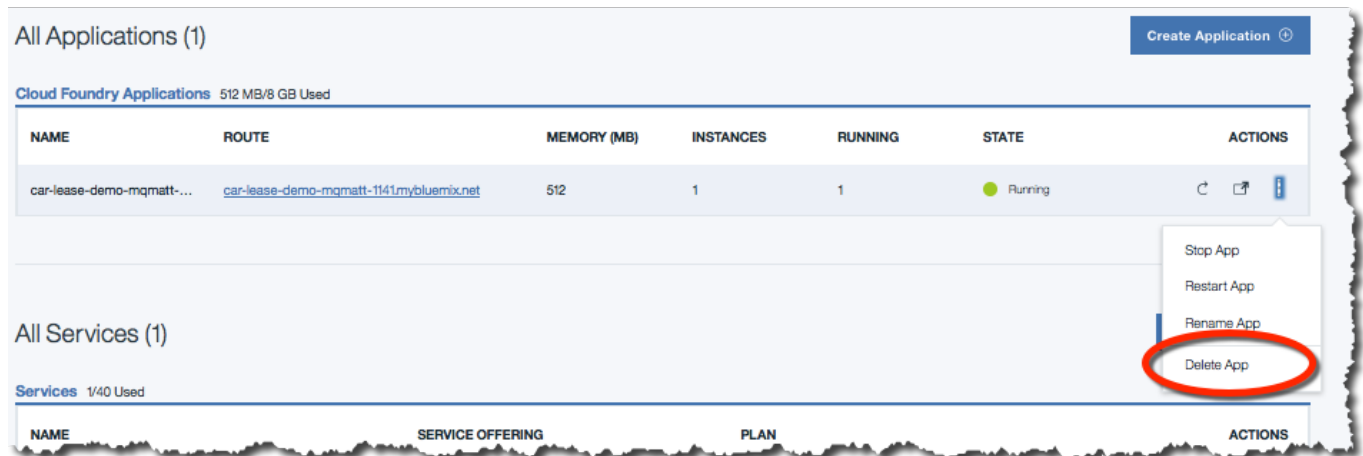
This page gives the opportunity to deploy more samples to the Blockchain service, and also some how to get started with writing your own blockchain applications and chaincode.

We will look at chaincode development in more detail in the follow-on lab “Blockchain Unchained”.

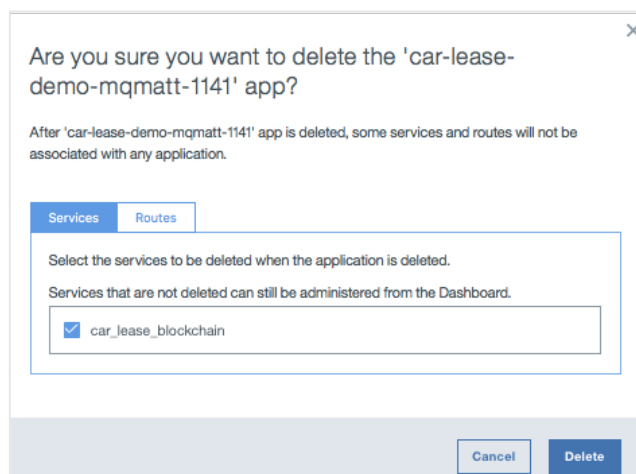
## Section 3. Removing the sample application

The final section of this lab aims to stop and remove the Blockchain service you created.

- \_\_39. Return to the Bluemix Dashboard (<https://console.ng.bluemix.net/dashboard/applications>).
- \_\_40. Click the three vertical dots at the right of the Car Leasing application Settings icon in the car lease demo application and select '**Delete App**' from the menu.



- \_\_41. Ensure that the 'car\_lease\_blockchain' service is selected for deletion and click '**Delete**'.



- \_\_42. Wait for the items to be stopped and deleted. Once this is done, both the application and the associated service will no longer be visible in the Bluemix dashboard.

**Congratulations on completing Lab two – “Blockchain Explored”!**

---

## Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries.

Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

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

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106-0032, Japan

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

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

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

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

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

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

products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

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

**COPYRIGHT LICENSE:**

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

---

## Appendix B. Trademarks and copyrights

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

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

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

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

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

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

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

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

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

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

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

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

## NOTES

[illegible]

## NOTES

[illegible]







---

© Copyright IBM Corporation 2016.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and [ibm.com](http://ibm.com) are trademarks of International Business Machines Corp., 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](http://www.ibm.com/legal/copytrade.shtml).



Please Recycle

---