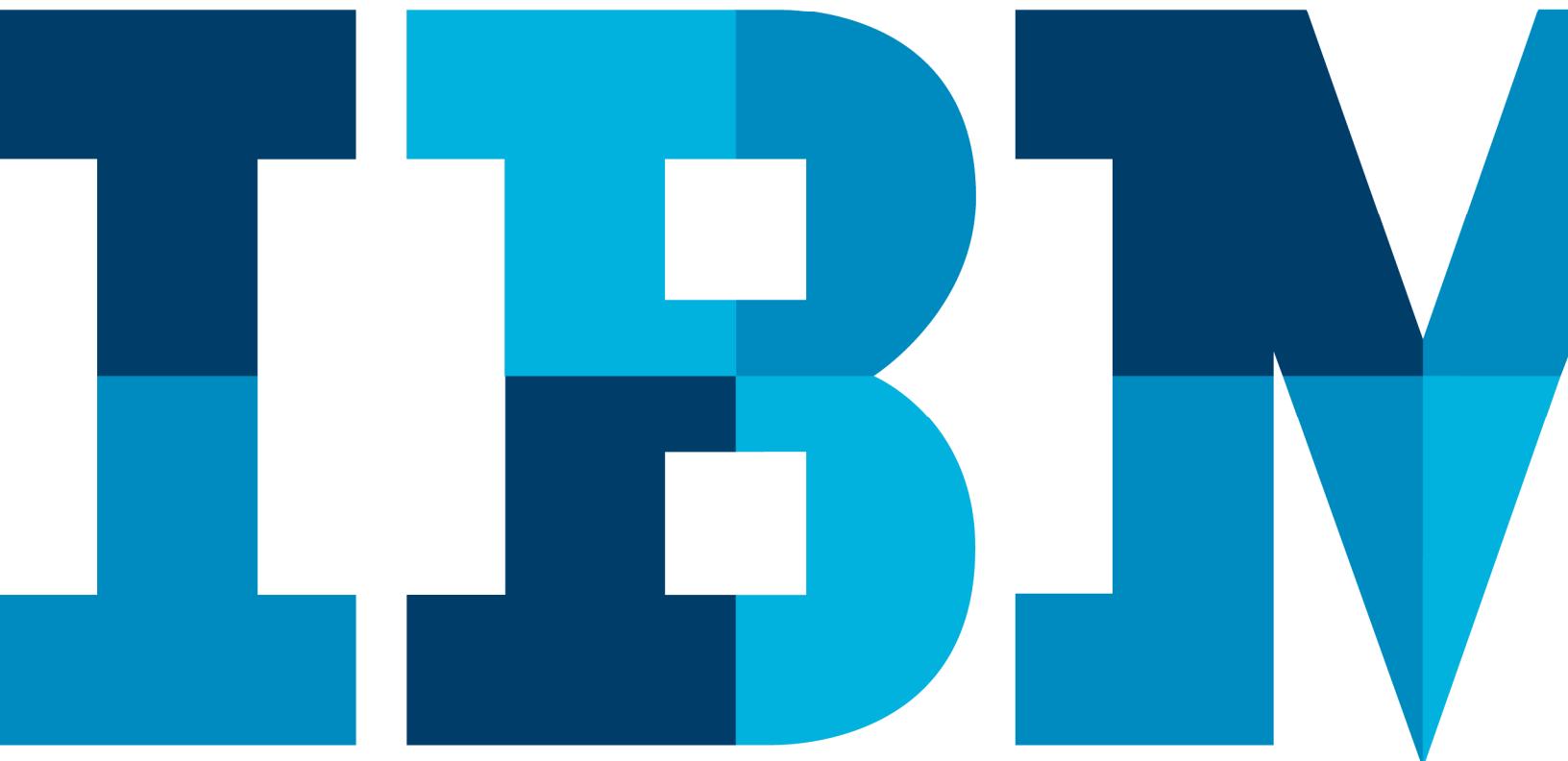


IBM Blockchain Proof of Technology Blockchain Explored

Lab Two - Exercises



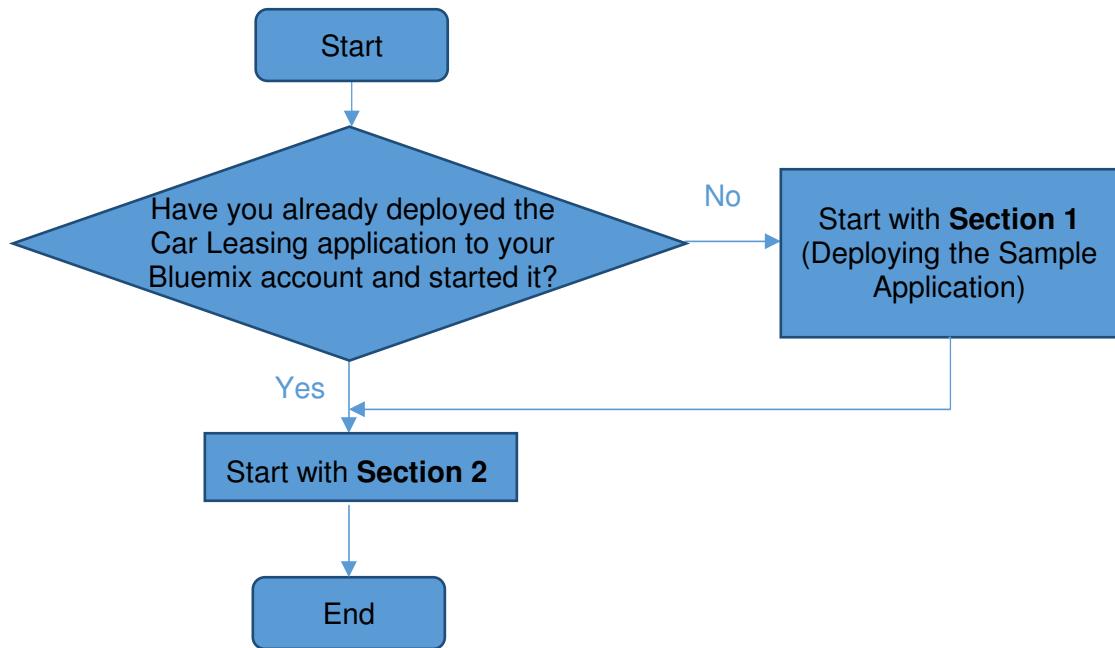
Contents

OVERVIEW	INTRODUCTION TO THE LAB	3
SECTION 1.	DEPLOYING THE SAMPLE APPLICATION.....	4
1.1.	CREATE A BLOCKCHAIN SERVICE	4
SECTION 2.	MANAGING THE SAMPLE APPLICATION.....	8
2.1.	VIEW THE COMPONENTS OF THE BLOCKCHAIN SERVICE	8
2.2.	VIEW THE BLOCKCHAIN EXPLORER	9
2.3.	UNDERSTANDING THE BLOCKCHAIN PEERS	12
2.4.	INTERACTING WITH THE PEERS	14
SECTION 3.	REMOVING THE SAMPLE APPLICATION.....	19
APPENDIX A.	NOTICES	21
APPENDIX B.	TRADEMARKS AND COPYRIGHTS	22

Overview Introduction to the Lab

The aim of this lab is to introduce you to the IBM 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 deployed the car leasing application to your account, which means you can skip section 1 and re-use your existing application:



Section 1. Deploying the sample application

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

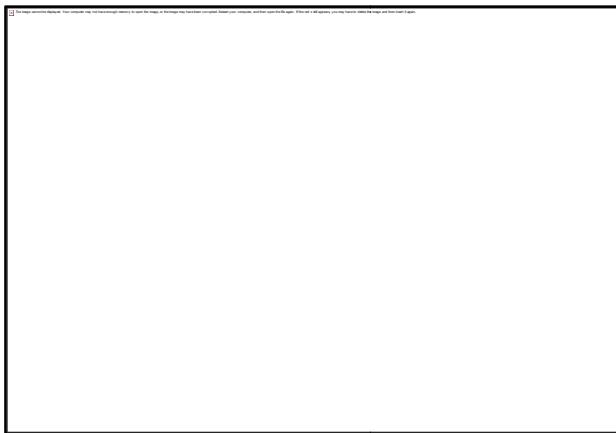
1.1. Create a Blockchain Service

1. Open a web browser and go to www.bluemix.net.

	<p>It is recommended to use Firefox or Chrome. Problems can generally be resolved by clearing the browser's cache and cookies, or running the browser in private mode.</p>
---	--

2. Select **CATALOG** from the top bar
3. Scroll down to the network section and click **Blockchain**
4. Review the service description and information about the service.
5. Click **VIEW DOCS** and learn about the process of creating a blockchain environment.
6. Click 'Sample Apps and Tutorials' on the left of the page to view the available apps.
7. Click **Deploy to Bluemix** against the Car Lease demo. Log in to Bluemix again if necessary.

	<p>The first time a bluemix ID creates a sample, a new DevOps alias is required. Pick a unique ID and click acceptance of the terms and then click create. For Proof-of-Technology workshops, use the first part of the email address, for example ibmpot000101 (without the '+' character). Then click continue on the following page.</p>
---	--



- 8. Leave the App Name, Region, Organization and Space default and click **DEPLOY**. (You might first need to wait a few seconds for the default field values to be populated.)

Clicking Deploy will cause the car leasing demo to be deployed into your Bluemix environment, and may take a couple of minutes to complete.

Deploy this application to Bluemix

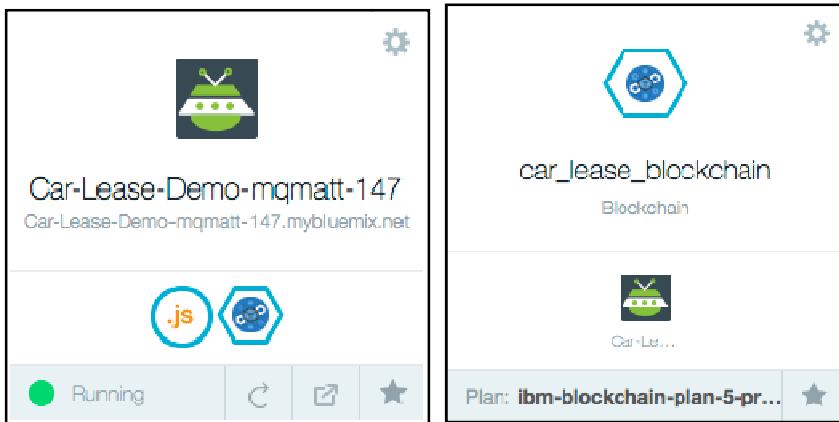
Deploying this app will create a private DevOps Services project for you. [Learn more.](#)

 **CAR-LEASE-DEMO**

GIT URL: <https://github.com/jpayne23/Car-Lease-Demo.git>
GIT BRANCH: master

- ✓ Created project successfully
- ✓ Cloned repository successfully
- ✓ Configured pipeline successfully
- ⠋⠋⠋ Deploying to Bluemix...

- __9. Once you see the ‘Success!’ message click **DASHBOARD** to see the new car leasing application (and associated blockchain service) you created.



- __10. Click the Node.js application’s icon in the dashboard (your icon may vary!)



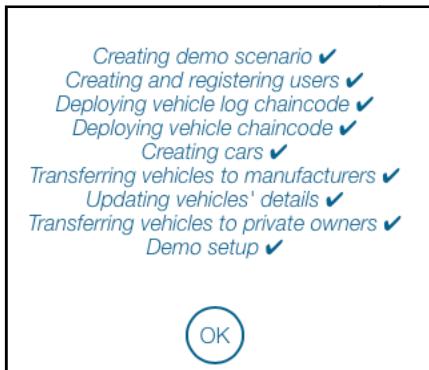
This will show you information about the application, including the memory that it is consuming and activity log.

- __11. Click the ‘Routes’ URL (something like **Routes: Car-Lease-Demo-mqmatt-147.mybluemix...**) in order to run the scenario. This will load a webpage which is served from the application.

- __12. Click ‘Admin Console’ and then ‘Create Simple Scenario’ to load the initial set of assets into the blockchain. This will take several minutes to complete.



13. The scenario setup is complete when 'Demo setup' is displayed.

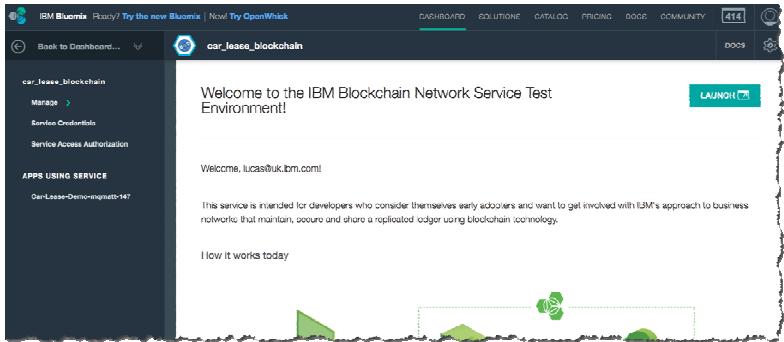


Section 2. Managing the sample application

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

2.1. View the components of the blockchain service

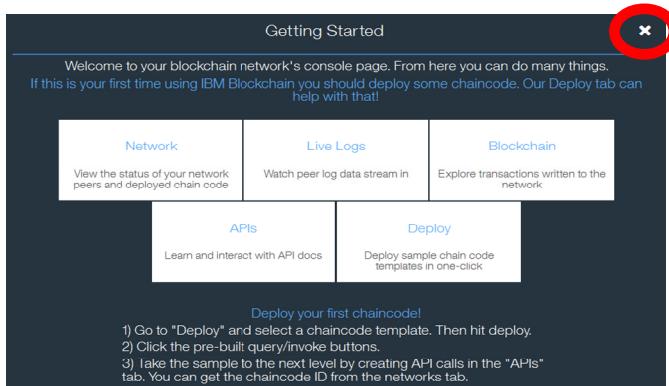
- __1. In Bluemix, select **DASHBOARD** to view the Car Leasing application.
- __2. Click on the service icon  for your new blockchain service in the Dashboard. This will take you to the service welcome screen.



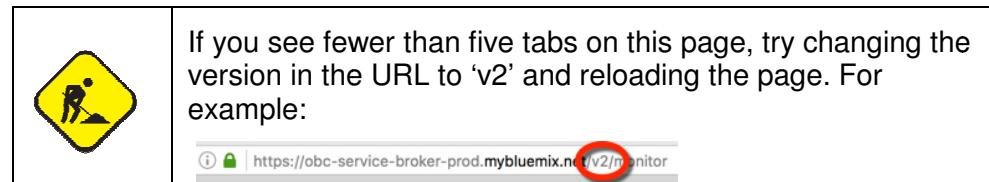
- __3. Review the details and select **LAUNCH** to launch the service console.

	If you see the following error, then try changing to a Firefox or Chrome browser:
	<p>Whoops</p> <p>The information you sent to access your instance of Blockchain was incorrectly formatted.</p>

- __4. Close the pop-up showing information about the sections. We shall look at these in more detail throughout this Lab.



- __5. This will take you to the monitor page, with the Network tab selected.



Peer	Routes	Status	Logs	Actions
Validating Peer 1	Discovery: 79a7456b-58ed-4fbf-9c37-5c05f34fdbba_vp1-discovery.blockchain.ibm.com:30303 API: 79a7456b-58ed-4fbf-9c37-5c05f34fdbba_vp1-api.blockchain.ibm.com:80	running		
Validating Peer 2	Discovery: 79a7456b-58ed-4fbf-9c37-5c05f34fdbba_vp2-discovery.blockchain.ibm.com:30303 API: 79a7456b-58ed-4fbf-9c37-5c05f34fdbba_vp2-api.blockchain.ibm.com:80	running		
Certificate Authority	Discovery: 79a7456b-58ed-4fbf-9c37-5c05f34fdbba_ca-discovery.blockchain.ibm.com:30303 API: 79a7456b-58ed-4fbf-9c37-5c05f34fdbba_ca-api.blockchain.ibm.com:80	running		

This view confirms that two validating peers and a certificate authority are running under the service you created.

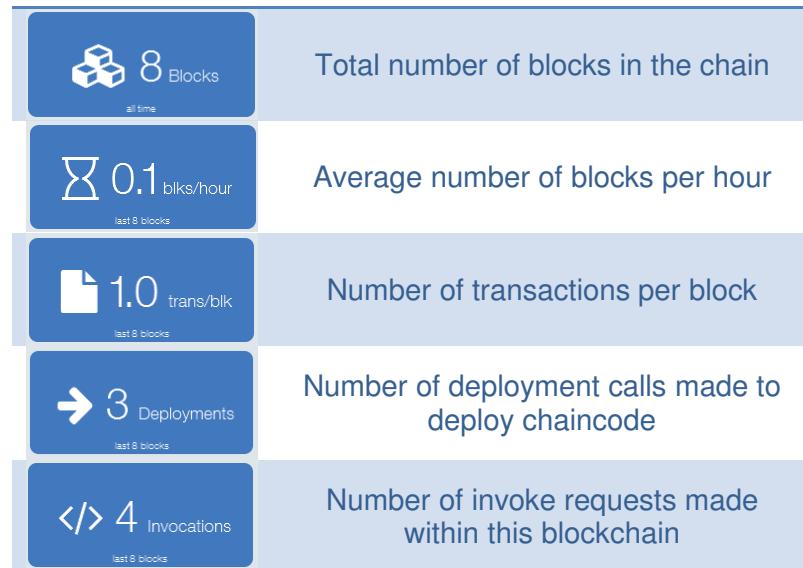
2.2. View the Blockchain Explorer

The blockchain explorer is a visual representation of the state of the blockchain.

- __1. Click the 'Blockchain' tab at the top of the page.



The icons show:



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:

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

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.

- 2. Click on a block that contains at least one invocation request.
- 3. Look through the list of transactions that are contained within the block.

DATE	TYPE	UUID	CHAINCODE ID	PAYOUT
05/26 01:49pm UTC	INVOKE	47897ad9-4356-4b3c-9405-38856383a5ee	9c48cd2b3...	create_vehicle_logTransferELeaseC ain & Joe Payne&72096598163005517 oyota Yaris Red, QD65 YKR DP88818 3826/5/2016 13:49:04LeaseCan Joe P ayne

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 1 transaction per block due to the low frequency of transactions being made. The information displayed is:

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

__4. Repeat this for other blocks to understand how the transactions are stored.

	<p>When the blockchain 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>
---	--

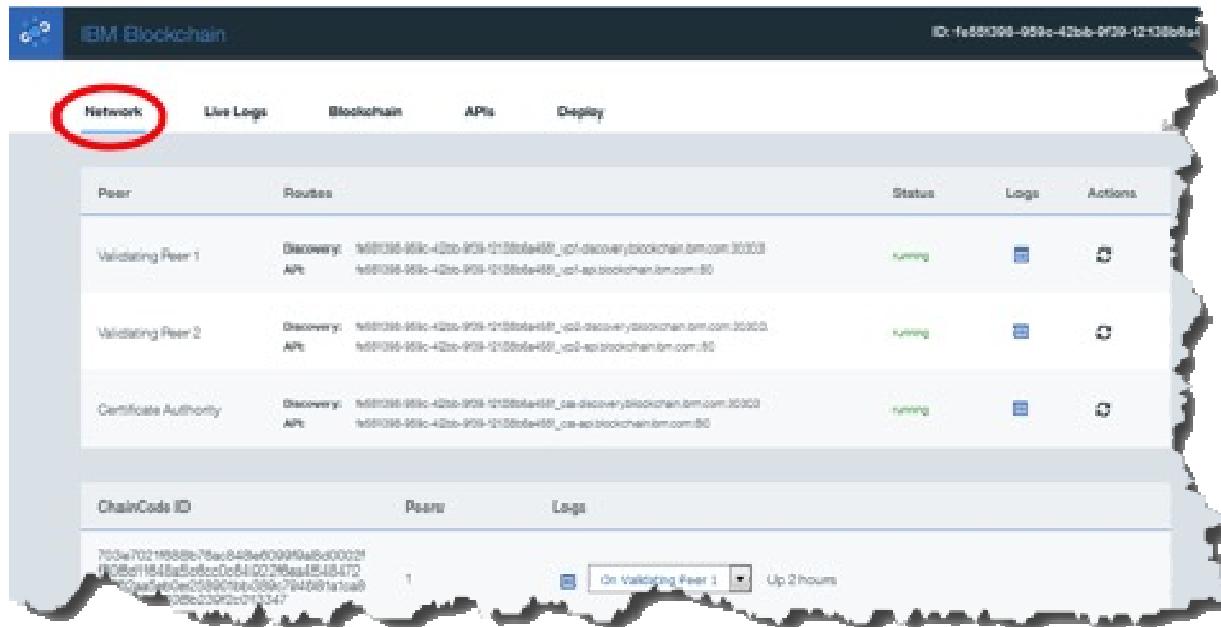
2.3. Understanding the Blockchain Peers

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.

There are two ways of accessing the logs of the peers:

- A “Logs” button on the Network tab. This is useful for downloading log files from the peers for offline analysis
- The “Live Logs” tab that shows you what the peers are doing now.

__1. Click on the ‘Network’ tab at the service page.



The screenshot shows the IBM Blockchain Service interface. The top navigation bar includes the IBM logo, a search bar, and a user ID. Below the navigation is a tabs menu with 'Network' (circled in red), 'Live Logs', 'Blockchain', 'APIs', and 'Deploy'. The main content area displays a table of peers:

Peer	Routes	Status	Logs	Actions
Validating Peer 1	Discovery: 45.103.64.103:4223>1234567890_>cd-discovery.blockchain.com:30303 API: 45.103.64.103:4223>903>1234567890_>cd-api.blockchain.com:80	running		
Validating Peer 2	Discovery: 45.103.64.103:4223>903>1234567890_>cd-discovery.blockchain.com:30303 API: 45.103.64.103:4223>903>1234567890_>cd-api.blockchain.com:80	running		
Certificate Authority	Discovery: 45.103.64.103:4223>903>1234567890_ca-discovery.blockchain.com:30303 API: 45.103.64.103:4223>903>1234567890_ca-api.blockchain.com:80	running		

Below the table is a summary section with 'ChainCode ID', 'Peers', and 'Logs'. It shows '703ae7021f0029a70ea843e039919a50d002f' for the first peer, '1' for the number of peers, and 'On Validating Peer 1' with a dropdown set to 'Logs' and a note 'Up 2 hours'.

Here we can see that this blockchain contains two validating peers and a Certificate Authority. The table underneath shows that there are two chaincode applications deployed to this network.

Requests to invoke chaincode (including the method name and any input parameters) are replicated onto every validating node, and when a block is created every validating node will execute the chaincode independently. (The validating peers then attempt to achieve consensus over any changes proposed to the world state as a result of running this chaincode, and as a consequence will persist or discard the changes.)

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

- __2. Select the peer for which you wish to see the logs and click on the  icon.



- __3. This will show the logs for the selected peer in a new tab.

```

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/c6elf96377aac59bb0985b26ad78a8958dcda7db5702ff4882fa]
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 invccations
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)
  
```

- __4. Click on the 'Live Logs' tab on the service page.

IBM Blockchain

ID: 79a7456b-58ed-4fbf-9037-5c05f34fd9ba

Network Live Logs Blockchain APIs Deploy

VP1 | VP2 | CA | Tail

```
["26/5/2016 13:47:45", {"text": "Model: undefined - X\u00e3.", "obj_id": "DU0361852", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:47:51"}, {"text": "Co-our: undefined - Black", "obj_id": "DU0361852", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:47:51"}, {"text": "VIN: undefined - 1M5E3ETX5F000051", "obj_id": "DU0361852", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Time: undefined - 59123558856123", "obj_id": "DU0361852", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Make: undefined - Jaguar", "obj_id": "G95568069", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Model: undefined - F-Type", "obj_id": "G95568069", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Color: undefined - White", "obj_id": "G95568069", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Registration: undefined - H101 KHZ", "obj_id": "G95568069", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "VIN: undefined - 1M5E3ETX5F000051", "obj_id": "G95568069", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Time: undefined - 523474131484831", "obj_id": "G95568069", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Make: undefined - Land Rover", "obj_id": "CA1384722", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Model: undefined - Defender", "obj_id": "CA1384722", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Color: undefined - Black", "obj_id": "CA1384722", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Registration: undefined - EK16 PRV", "obj_id": "CA1384722", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "VIN: undefined - 1D63C38097753", "obj_id": "CA1384722", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Time: undefined - 5463C38097753", "obj_id": "CA1384722", "users": [{"Jaguar Land Rover": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Make: undefined - Alfa Romeo", "obj_id": "SE619905", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Model: undefined - Giulietta", "obj_id": "SE619905", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Color: undefined - White", "obj_id": "SE619905", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Registration: undefined - JU05 KHM", "obj_id": "SE619905", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "VIN: undefined - 1N6A12A001261", "obj_id": "SE619905", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Time: undefined - 508944731", "obj_id": "SE619905", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:23"}, {"text": "Make: undefined - Mito", "obj_id": "K98944731", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "Model: undefined - Mito", "obj_id": "K98944731", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "Color: undefined - Black", "obj_id": "K98944731", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "Registration: undefined - YD5 FTBV", "obj_id": "K98944731", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "VIN: undefined - 1N6A12A001261", "obj_id": "K98944731", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "Time: undefined - 538494731", "obj_id": "K98944731", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "Make: undefined - Alfa Romeo", "obj_id": "V002271564", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "Model: undefined - Giulietta", "obj_id": "V002271564", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "Color: undefined - Black", "obj_id": "V002271564", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "Registration: undefined - V002271564", "obj_id": "V002271564", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "VIN: undefined - 1N6A12A001261", "obj_id": "V002271564", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}, {"text": "Time: undefined - 538494731", "obj_id": "V002271564", "users": [{"Alfa Romeo": 1}], "name": "Update", "time": "26/5/2016 13:48:32"}]
```

This page shows the same logs that were shown from the Network tab, however these are live updating (if you have ‘Tail’ selected), and you can also see a combined view of multiple validating peers and the Certificate Authority.

5. Click on the 'VP1', 'VP2' and 'CA' buttons to toggle on/off the viewing of live logs for each peer.



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 – this is not the same as adding and invoking transactions through chaincode!

1. 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).

- 2. Click the ‘Blockchain’ section.

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

- 3. Click ‘Expand Operations’ to view information about this API.

This reveals the input and output data formats.

The screenshot shows the IBM Blockchain API Explorer interface. At the top, it says "Blockchain" and has tabs for "Show/Hide", "List Operations", and "Expand Operations". Below that, it shows a "GET /chain" operation under the "Blockchain information" section. It includes "Implementation Notes" stating that the Chain endpoint returns information about the current state of the blockchain such as the height, the current block hash, and the previous block hash. A "Response Class (Status 200)" section shows a JSON schema:

```
{
  "height": 0,
  "currentBlockHash": "string",
  "previousBlockHash": "string"
}
```

Below this, there's a "Response Content Type" dropdown set to "application/json". Under "Response Messages", there's a "default" entry with a "Reason" of "Unexpected error" and a "Response Model" schema:

```
{
  "Error": "string"
}
```

4. Click 'Try It Out' to invoke the API.

The screenshot shows the results of invoking the /chain API. It includes sections for "Curl", "Request URL" (https://79a7456b-58ed-4bfb-9037-5c05f34fdbba_vp1-api.blockchain.ibm.com:443/chain), "Response Body", "Response Code" (200), and "Response Headers".

Curl

```
curl https://79a7456b-58ed-4bfb-9037-5c05f34fdbba_vp1-api.blockchain.ibm.com:443/chain
```

Request URL

https://79a7456b-58ed-4bfb-9037-5c05f34fdbba_vp1-api.blockchain.ibm.com:443/chain

Response Body

```
{
  "height": 80,
  "currentBlockHash": "mgbwJnFsZQEpCKk5t3YVF1z7Q6rUMdLlxLfelde6AX7AbidS5tc81dZsuuWC7sN2z0DBneawViCKklzOFc7Q==",
  "previousBlockHash": "VBr27Y6mkqHhRCLVWQbe+o28TmEQTYymSbJuk0CPAAvTUsl2vsnRpf1XBS47q95qViCJlewCCk3tIT+FZC5utQ=="
}
```

Response Code

200

Response Headers

```
{
  "content-type": "application/json"
}
```

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.

__5. Expand the ‘Block’ section and review the information on how to interrogate an individual block in the blockchain.

Block

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

Show/Hide | List Operations | Expand Operations

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": [
    ...
  ]
}
```

__6. 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
Block	40	Block number to retrieve	path	integer

Response Messages

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

[Try it out!](#)

__7. Review the information returned in the Response Body.

Request URL

```
https://79a7456b-58ed-4bf8-9037-5c05f34fdbba_v1-api.blockchain.ibm.com:443/chain/blocks/40
```

Response Body

```
{
  "transactions": [
    {
      "type": 2,
      "chaincodeID": "EoABYzZIMWY5NjM3N2FhYzU5YmlwOTgYVjI2WQ30GE1OTU4ZGNy2E3ZGInzAyZmY00DgyZmEzOTRzTAzMDU4YmQ3OTE5MmkyYV1ZmY1MmNkZGZjYThIMDA3MGE2NjA2OGQ0NDE",
      "payload": "CtBCAESgwEsqAFNmUxZjK2Mzc3WFNTllyA5ODVMJzhZDc4Ytg5NTkY2rjYtDkYjU3MDUmZjQ4ODJmYTM5NGJMDMwNTiZDc5MTkyYjF1NjVmZigY2RkZmNhOGlwMDowYTM2MDY4ZDQ0",
      "uuid": "a338564e-ceef-4df6-9efd-95b65fa43efc",
      "timestamp": {
        "seconds": 1464270464,
        "nanos": 266423527
      },
      "nonce": "s+rSjUX6x6UBLt4brf4YPp4sz56sjXo",
      "cert": "MIQCTCAjOgAwBgiTAoOr0tMhEMitZTWEAq+AUwCgYKoZjz0EAwMwKTELMAkQA1UEBhMCVVVmDDAKBgNVBAoTA0ICTTEMAoCA1UEAxMDdONhMB4XDTE2MDUyNjEzNDU1NjoxXDTE2MDQwggEAMQIBAjBrgMyGZY52Bwzp2i2WeShrGUYKhvPwMgoQc3zB24AlBDatRcqdyOMBHyUOeXQBAaDjNF+6ZBuDWqvrp1RYIrgw==",
      "signature": "MEQCIB4jBrgMyGZY52Bwzp2i2WeShrGUYKhvPwMgoQc3zB24AlBDatRcqdyOMBHyUOeXQBAaDjNF+6ZBuDWqvrp1RYIrgw=="
    }
  ],
  "stateHash": "0zUHgocVkoNafvxRADqMXNOET1ZRoy4aDP1gx2WBHENy+DPKZeBskSePwfYNHbzYINsExIZLXmcL9g29elg==",
  "prevHash": "xrCElkDT5Xm0Pj6R2lnRyo/B2EIL3NKs4x/Zj6TKPGqcqGoskVhpsMxUkZ/I7o2e0D4V4fF21+3_hBokPw=="
}
```

- __8. Copy the UUID field of a transaction from a block; this will be of the form “a338564e-ceef-4df6-9efd-95b65fa43efc”.
- __9. Click the ‘Transactions’ section.

Transactions

GET /transactions/{UUID}

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

10. 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:

c6e1f96377aac59bb0985b26ad78a8958dcda7db5702ff4882fa394be03058bd79192b1b65ff8
2cddfca8b0070a36068d44136ca178fc681cc54c4b6321f48c
update_vin 181255391772389 DU0061852

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

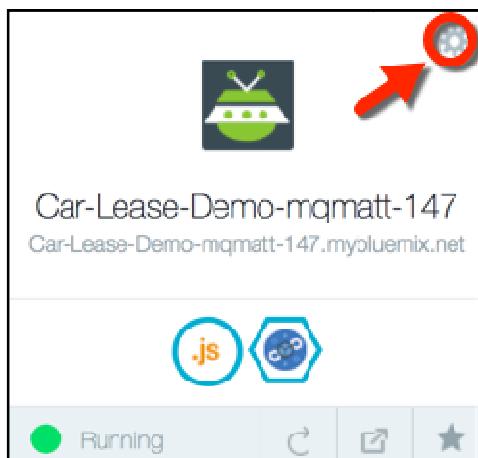
- _11. Now spend some time interacting with the other APIs available to you.

Section 3. Removing the sample application

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

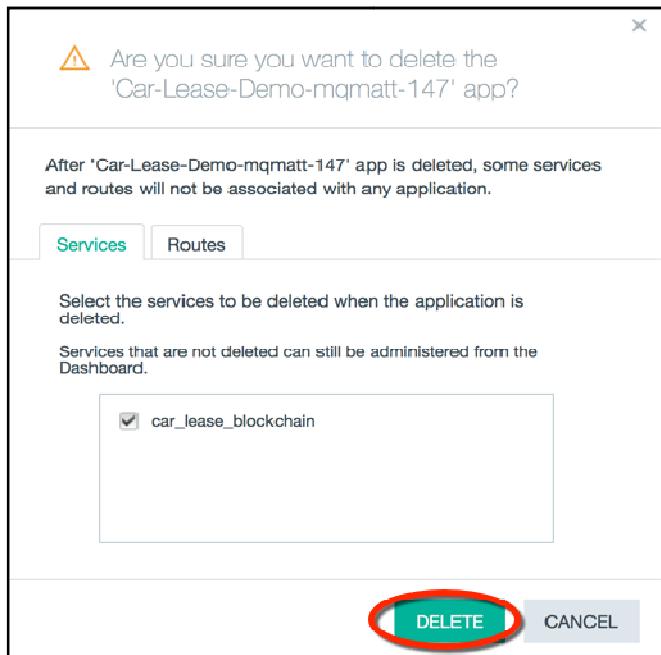
1. Return to the Bluemix Dashboard by clicking **DASHBOARD**

2. Click the Settings icon in the car lease demo application.



3. Select ‘Delete App’ from the menu.

- __4. Ensure that the ‘car_lease_blockchain’ service is also selected for deletion and click ‘Delete’.



- __5. 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

NOTES



© 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 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.



Please Recycle
