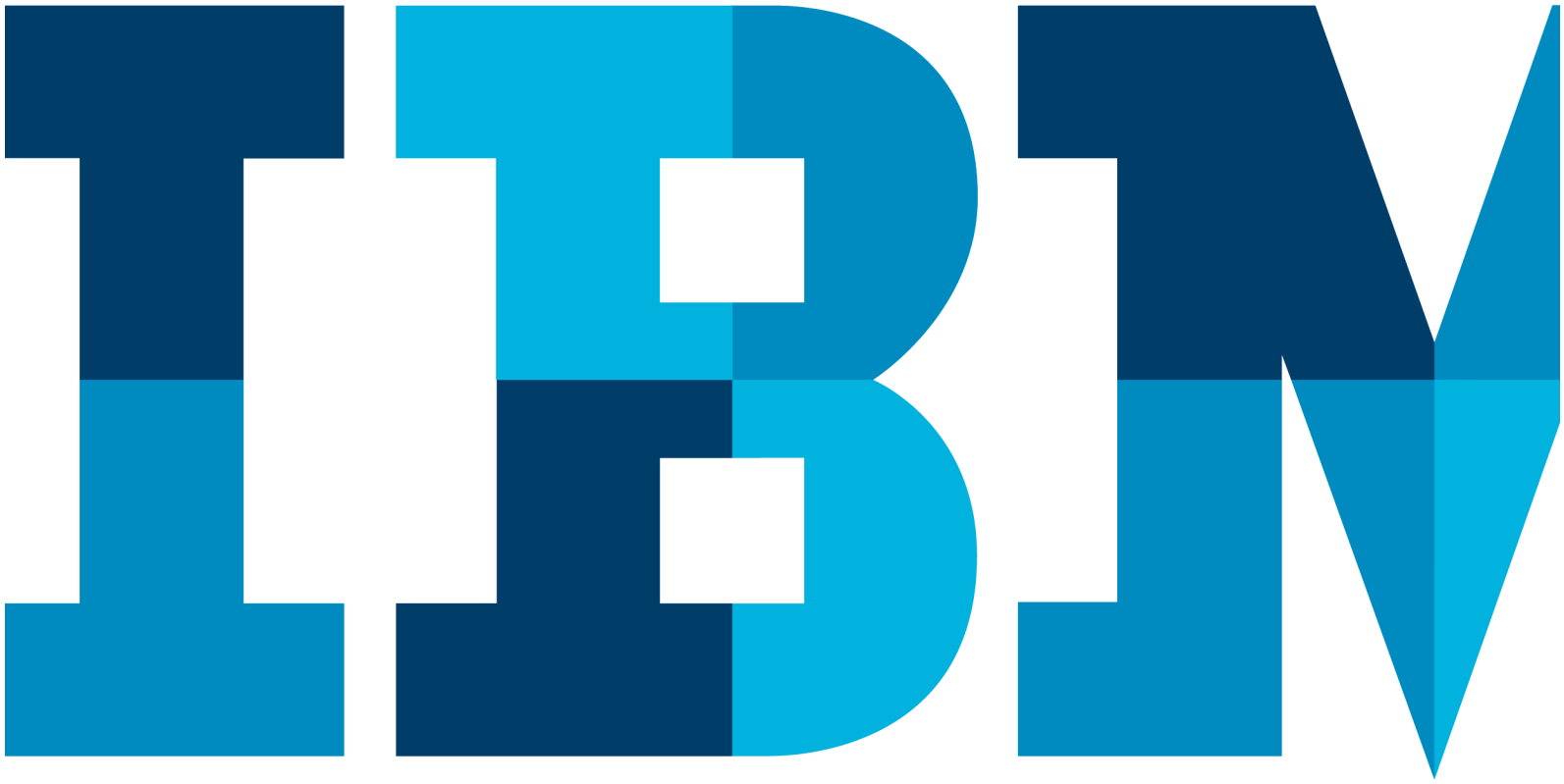
IBM Blockchain Hands-On
Blockchain Explored

Lab Two – VM – Exercises

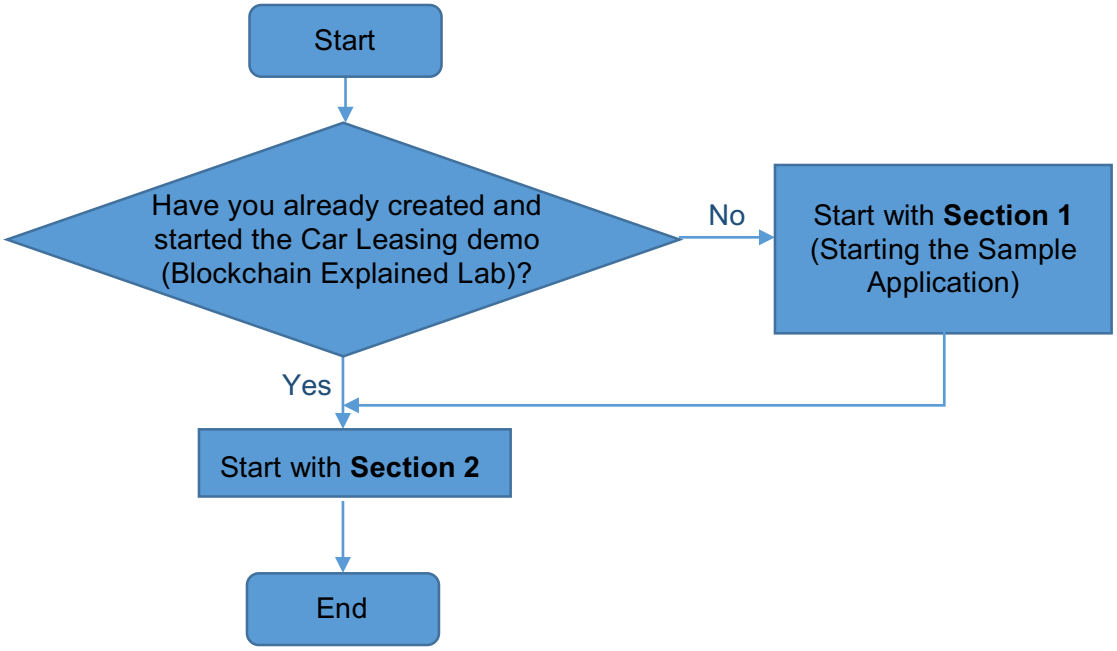


IBM

Contents

Overview

This lab will start to show the operational aspects of the Car Leasing demo using the command line interfaces of Hyperledger Fabric.



Introduction

Pre-requisites:

- 4 cores
- 4GB RAM
- VMWare V10+
- The lab virtual machine (IBM_Hyperledger_Car_Leasing_Demo_v0.8)

The virtual machine is based on Linux Ubuntu 14.04 and contains Hyperledger Fabric V0.6, Golang, Git, Vagrant, Visual Studio Code with the “Encode Decode” extension and Firefox.

A network needs to be visible to the virtual machine (even if the network is just to the host environment). If you do not see the up/down arrows in the status bar at the top of the screen, or if you receive errors about no network being available, please tell the lab leader. The virtual machine might need to be reconfigured in NAT mode.

There are no additional files or software that is proprietary to the lab in the virtual machine. This means that the lab may be run on a machine without the without a lab virtual machine if Hyperledger Fabric and the other pre-requisites have been installed.

It is recommended that students have previously completed the Blockchain Explained lab.

Section 1. Starting the Sample Application

If you have already run the Blockchain Explained lab on the same machine as you are going to use for this lab, **please skip to section 2.**

The IBM Blockchain Asset Transfer Demo environment exists in a VMWare virtual machine. The operating system is Linux Ubuntu. The following section will guide you through what you need to do in order to access the Main demo page.

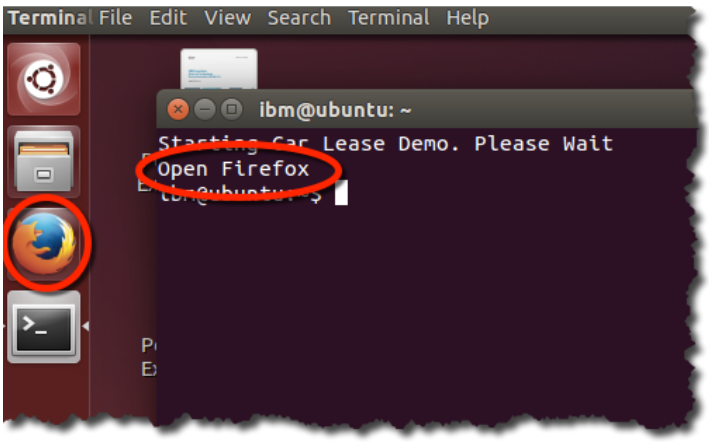
The VM should log you in automatically. If it doesn't or if the system locks later you can sign on to the Ubuntu system with the following credentials:

User: **`IBM`**
Password: **`passw0rd`**

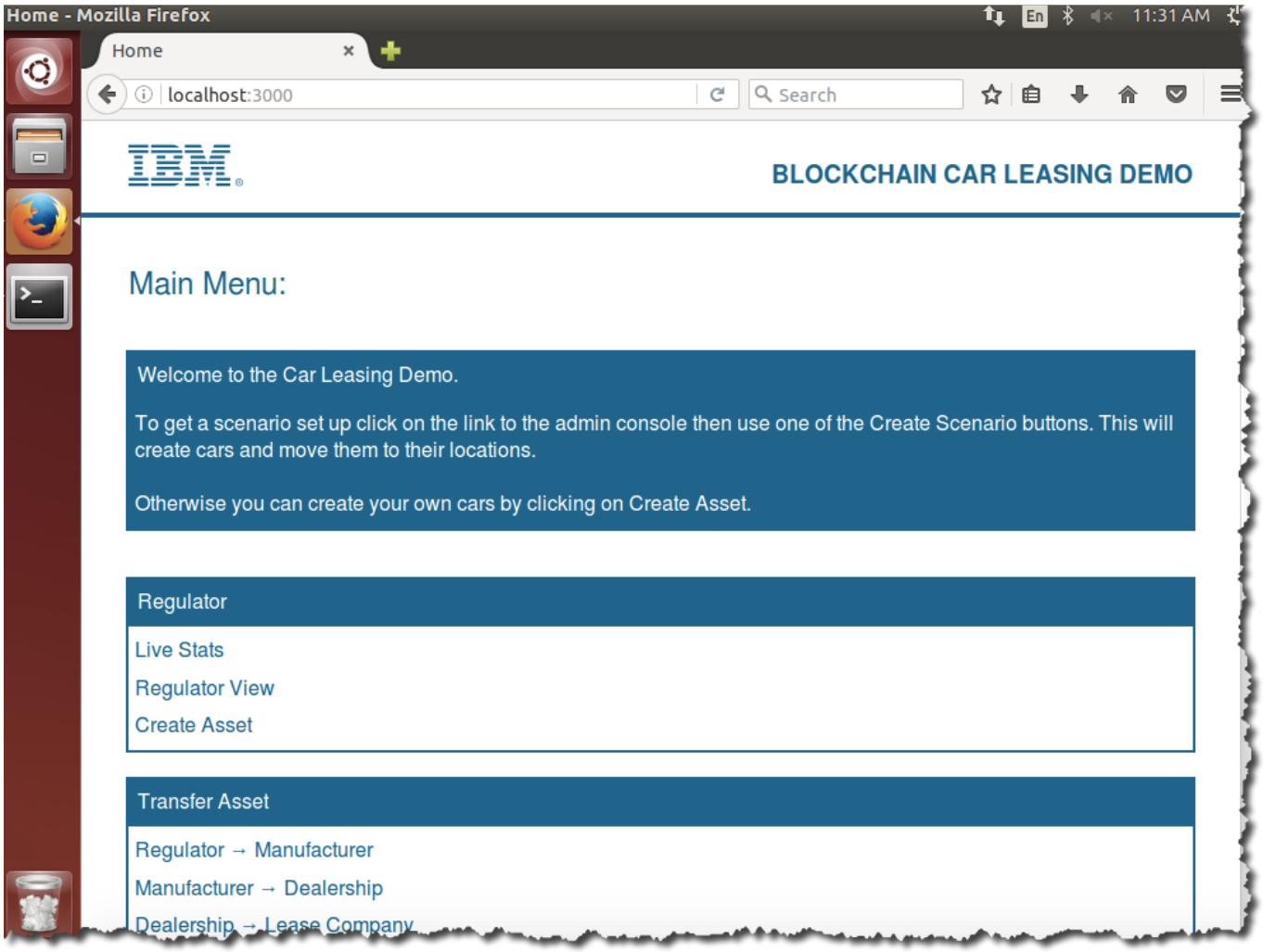
(*to sign on just press enter after entering the password*)

When the machine starts and user is logged, in you might see a window with "Starting Car Lease Demo" displayed. Before continuing, wait for the words "Open Firefox" to appear. Alternatively, the Firefox web browser might already be open at the Car Leasing demo main menu.

__1. If the web browser is not already open, wait for the initial setup to complete. Setup has completed when the words "Open Firefox" appear. At this point, click the Firefox icon on the left hand side of the screen.



You should now see the car leasing demo main menu.

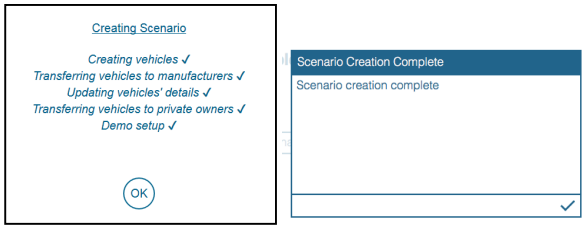


__2. From the Car Leasing demo front page, click '**Admin Console**' and '**Create Simple Scenario**'.

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 a couple of minutes longer to initialize, however.)

Wait for the initialization to complete.

__3. Click '**OK**' to close the Creating Scenario log, and then dismiss the 'Scenario Creation complete' by clicking the check mark. Click 'Home' to return to the main menu.

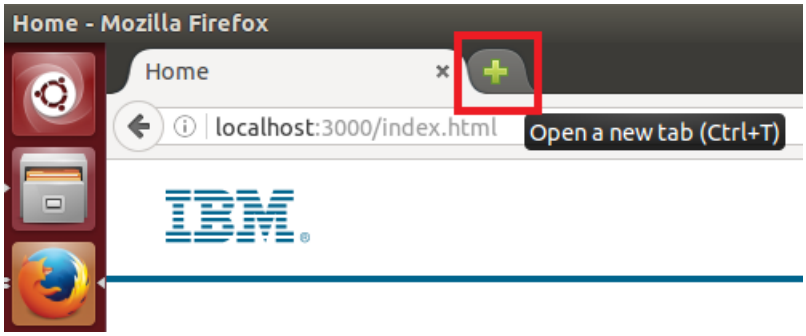


Section 2. Administering the blockchain using REST

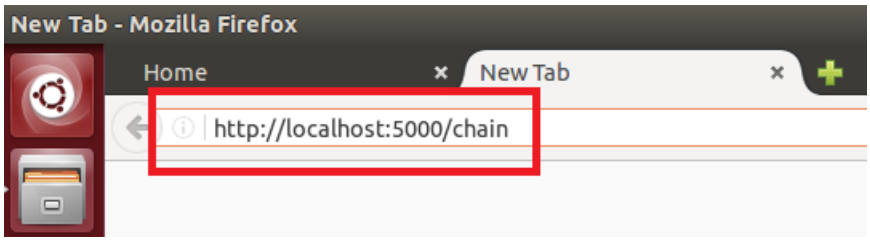
This section will show you how to view the number of blocks that make up the blockchain, which is called the “block height” and show you how to inspect the contents of a block on the chain.

2.1. Viewing the length of the chain

__4. To see the length of the chain, open up a new “tab” in the Firefox browser by clicking on the green ‘**+**’ symbol:

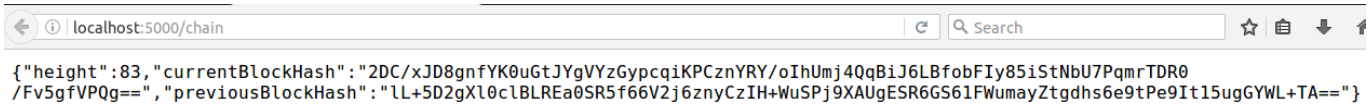


__5. Enter the address into the address box **`http://localhost:5000/chain`** and press enter:

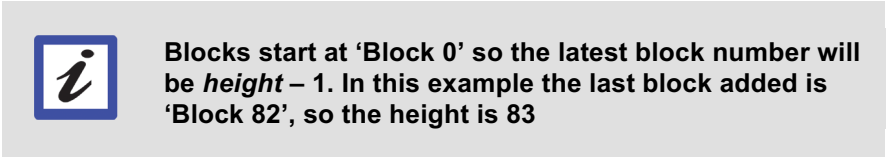


__6. Review the returned data structure.

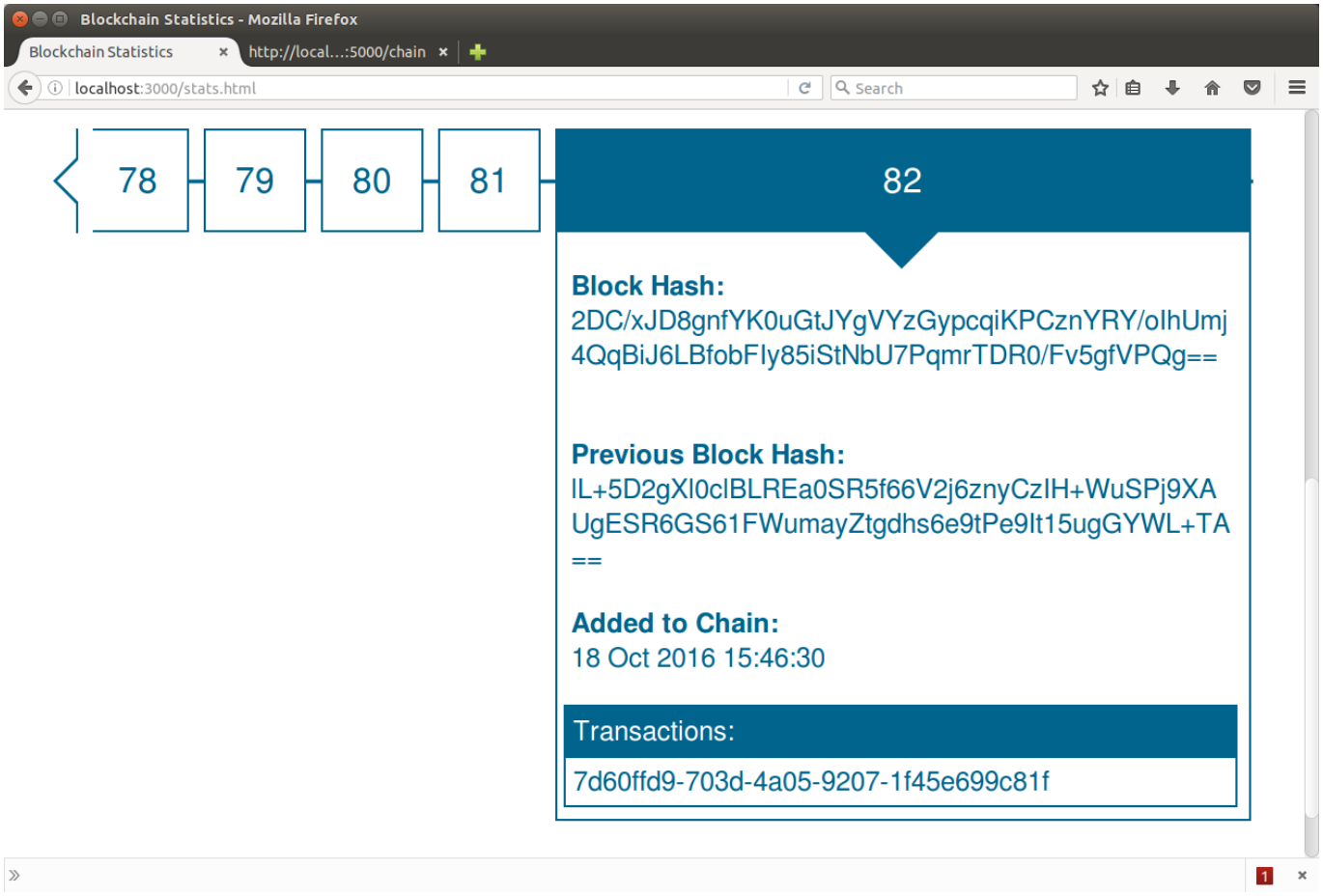
You will see a screen similar to the one below. On this screen, you will see several pieces of information: “height”, “currentBlockHash” and “previousBlockhash” in a JSON formatted message. The “height” contains “block height” or the number of blocks on the chain. The height you see will depend on the length of your chain and may differ from the number shown below (83). Factors such as number of cars transferred and type of scenario run (Simple or Full) will determine how many blocks are on your chain.



The “currentBlockHash” and the “previousBlockHash” fields contain the hashes for the most recently added blocks. You can compare these values to the ones shown for the most recent (largest numbered) block in the live stats view as shown below. Note that because blocks are numbered from zero (0) the latest block is one less than the block height. If you look at the screenshot of the Car Leasing demo’s blockchain explored view (below), you can see that for block 82, the hashes match the current and previous ones shown above.

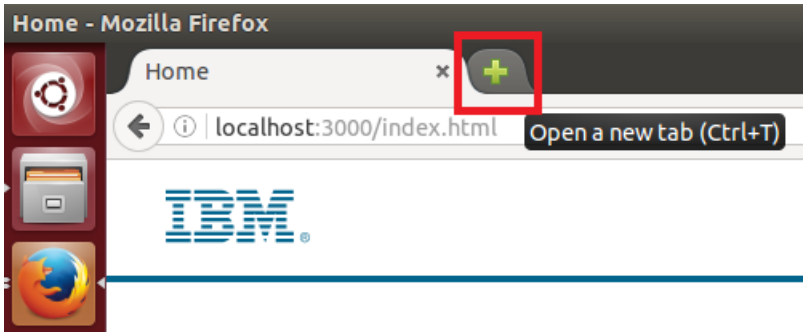


Blocks start at ‘Block 0’ so the latest block number will be *height* – 1. In this example the last block added is ‘Block 82’, so the height is 83

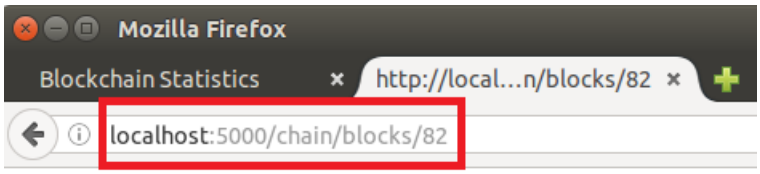


2.2. Viewing the contents of a block

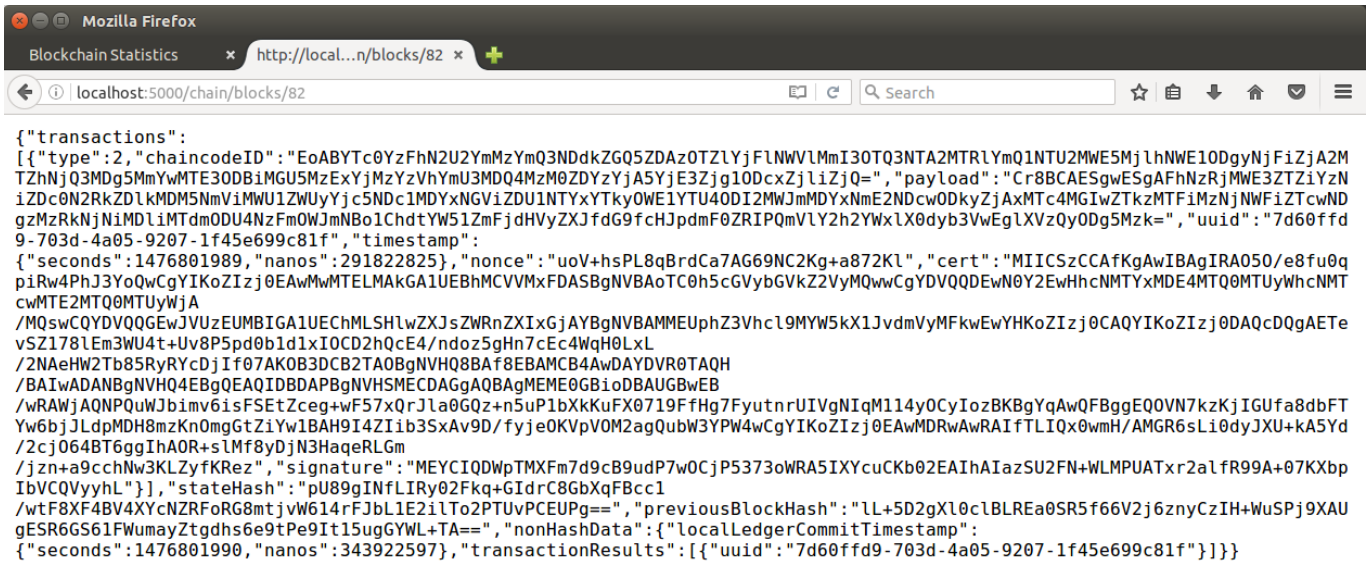
__7. To view the contents of a block open up a new “tab” in the Firefox browser by clicking on the green ‘**+**’ symbol:



__8. Enter the address **`http://localhost:5000/chain/blocks/82`** and press enter. Note that '82' is the number of the block and can be changed to any other block whose contents you want to see. However, if you have followed lab one first, using block 82 will allow your screen to match the ones below more clearly.

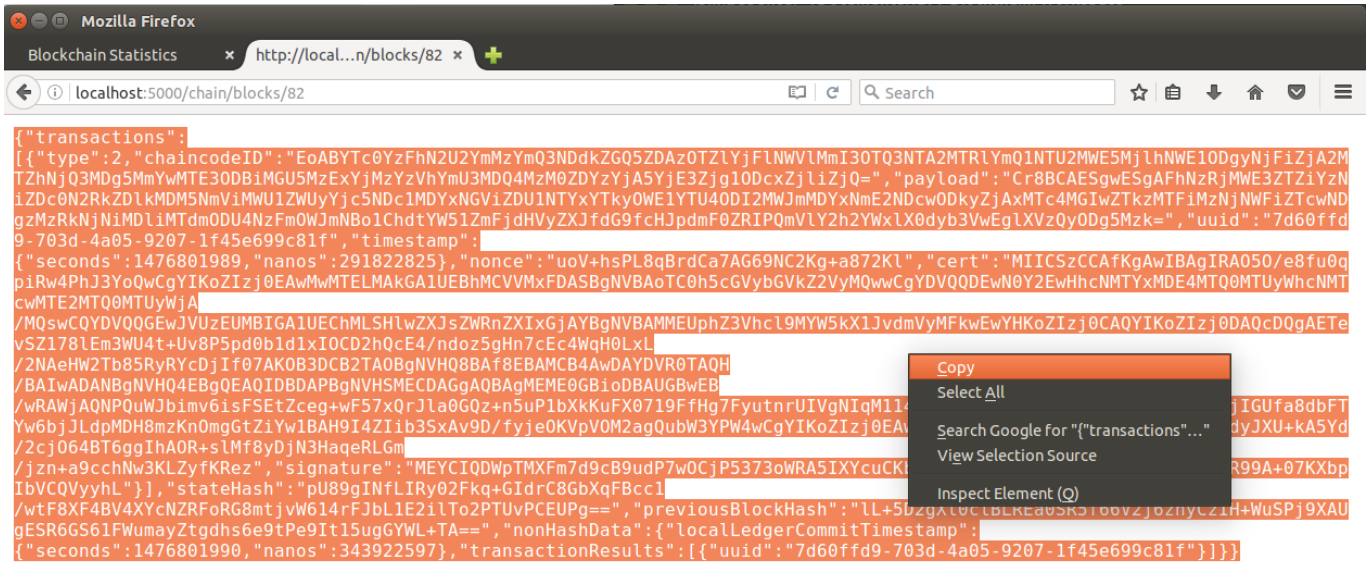


__9. You will see a screen similar to the one below. As there is a lot of detail, we will format the information in the next step to make it easier to read.



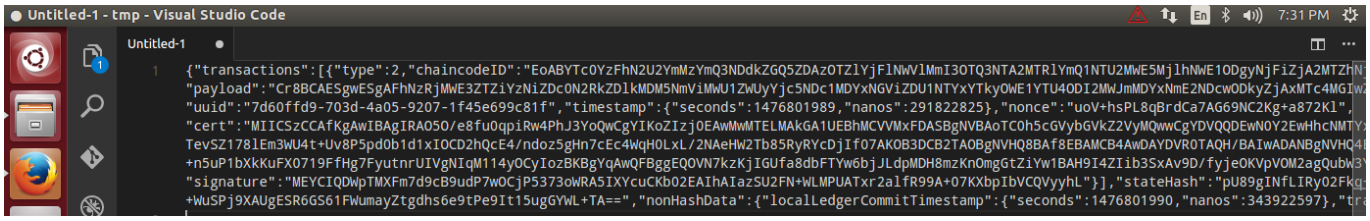

```
{"transactions":
[{"type":2,"chaincodeID":"EoABYTc0YzFhN2U2YmMzYmQ3NDdkZGQ5ZDAzOTZlYjFlNWVlMmI3OTQ3NTA2MTRlYmQ1NTU2MWE5MjlhNWE1ODgyNjFiZjA2M
TZhNjQ3MDg5MmYwMTE3ODBiMGU5MzExYjMzYzVhYmU3MDQ4MzM0ZDYzYjA5YjE3Zjg1ODcxZjliZjQ=","payload":"Cr8BCAESgwESgAFhNzRjMWE3ZTZiYzN
iZDc0N2RkZDlkMDM5NmViMWU1ZWUyYjc5NDc1MDYxNGViZDU1NTYxYTkyOWE1YTU4ODI2MWJmMDYxNmE2NDcwODkyZjAxMTc4MGIwZTkzMTFiMzNjNWFiZTcwND
gzMzRkNjNiMDliMTdmODU4NzFmOWJmNBo1ChdtYW51ZmFjdHVyZXJfdG9fcHJpdmF0ZRIPQmVlY2h2YWxlX0dyb3VwEglXVzQyODg5Mzk=","uuid":"7d60ffd
9-703d-4a05-9207-1f45e699c81f","timestamp":
{"seconds":1476801989,"nanos":291822825},"nonce":"uoV+hsPL8qBrdCa7AG69NC2Kg+a872Kl","cert":"MIICSzCCAfKgAwIBAgIRAO5O/e8fu0q
piRw4PhJ3YoQwCgYIKoZIzj0EAwMwMTELMAkGA1UEBhMCVVMxFDASBgNVBAoTC0h5cGVybGVkZ2VyMQwwCgYDVQQDEwN0Y2EwHhcNMTYxMDE4MTQ0MTUyWhcNMT
cwMTE2MTQ0MTUyWjA
/MQswCQYDVQQGEwJVUzEUMBIGA1UEChMLSHlwZXJsZWRnZXIxGjAYBgNVBAMMEUphZ3Vhcl9MYW5kX1JvdmVyMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAETe
vSZ178lEm3WU4t+Uv8P5pd0b1d1xIOCD2hQcE4/ndoz5gHn7cEc4WqH0LxL
/2NAeHW2Tb85RyRYcDjIf07AKOB3DCB2TAOBgNVHQ8BAf8EBAMCB4AwDAYDVR0TAQH
/BAIwADANBgNVHQ4EBgQEAQIDBDAPBgNVHSMECDAGgAQBAgMEME0GBioDBAUGBwEB
/wRAWjAQNPQuWJbimv6isFSEtZceg+wF57xQrJla0GQz+n5uP1bXkKuFX0719FfHg7FyutnrUIVgNIqM114yOCyIozBKBgYqAwQFBggEQOVN7kzKjIGUfa8dbFT
Yw6bjJLdpMDH8mzKnOmgGtZiYw1BAH9I4ZIib3SxAv9D/fyjeOKVpVOM2agQubW3YPW4wCgYIKoZIzj0EAwMDRwAwRAIfTLIQx0wmH/AMGR6sLi0dyJXU+kA5Yd
/2cjO64BT6ggIhAOR+slMf8yDjN3HaqeRLGm
/jzn+a9cchNw3KLZyfKRez","signature":"MEYCIQDWpTMXFm7d9cB9udP7wOCjP5373oWRA5IXYcuCKb02EAIhAIazSU2FN+WLMPUATxr2alfR99A+07KXbp
IbVCQVyyhL"}],"stateHash":"pU89gINfLIRy02Fkq+GIdrC8GbXqFBcc1
/wtF8XF4BV4XYcNZRFoRG8mtjvW614rFJbL1E2ilTo2PTUvPCEUPg==","previousBlockHash":"lL+5D2gXl0clBLREa0SR5f66V2j6znyCzIH+WuSPj9XAU
gESR6GS61FWumayZtgdhs6e9tPe9It15ugGYWL+TA==","nonHashData":{"localLedgerCommitTimestamp":
{"seconds":1476801990,"nanos":343922597},"transactionResults":[{"uuid":"7d60ffd9-703d-4a05-9207-1f45e699c81f"}]}}
```


__10. Move the mouse over the text, right-click and choose "Select All". Then right-click again and choose "Copy":



__11. Open “Visual Studio Code” by clicking on its blue icon on the toolbar:

__12. Choose **File** > **New File** from the menu or press “**Ctrl+N**” to create a new file. Once the empty file opens press “**Ctrl-V**” to paste the text copied above into the new file window.



__13. In the bottom right of the editor window, click the text that says “Plain Text” and from the list that appears choose (or type) “JSON” to set the editor format to show JSON data.

You will see the text is now colour coded, but is not properly formatted yet.

__14. Right-click and choose "**Format Document**".

The text will now look similar to the screen shot below. You can also choose **View > Toggle Word Wrap** to wrap the text onto the screen.

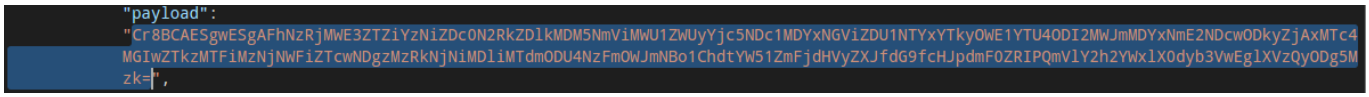
__15. Now you can see the data formatted properly; the most interesting parts are described below:

transactions	An array of transactions stored in the block.
type	This shows the possible values: 0. Undefined 1. Deployment 2. Invoke 3. Query

chaincodeID	ID of the chaincode that was invoked or deployed.
payload	Input parameters to the chaincode.
uuid	Unique identifier of this transaction.
timestamp	Time at which the block or transaction order was proposed.
cert	Certificate of the participant submitting the transaction.
signature	Signature of the participant submitting the transaction.
stateHash	Hash of the world state changes.
previousBlockHash	Hash of the previous block in the chain.
nonHashData	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.
localLedgerCommitTimestamp	Time the block was added to the ledger on the local peer.

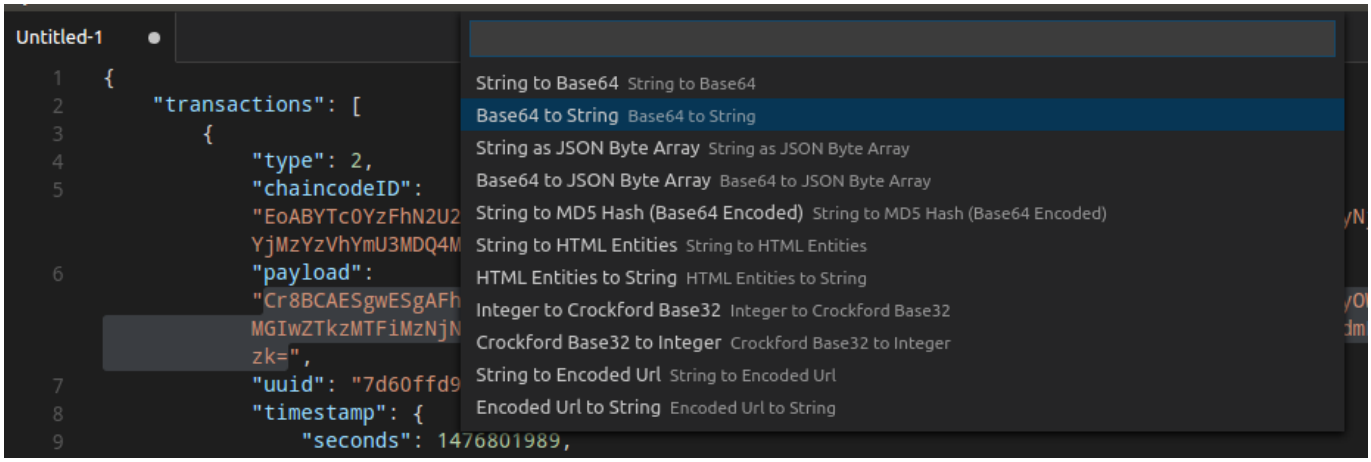
Looking at the data you can see that much of it is not easily understood, because the data is also encoded using "base64" encoding. To see a little more of the contents we will decode the value of the "payload" field.

__16. First select all the data in the payload field within the quotes, but do not select the quotes themselves.



The lab's virtual machine has the "Encode Decode" plugin pre-installed into Visual Studio Code, which the next few steps require. If you are running this lab in a different environment, you need to install a similar means of decoding base64-encoded content.

__17. Now press "**Ctrl+Alt+C"** and the conversion menu will appear.



__18. From the conversion menu select the second option '**Base64 to String**' and the selected data will be decoded:

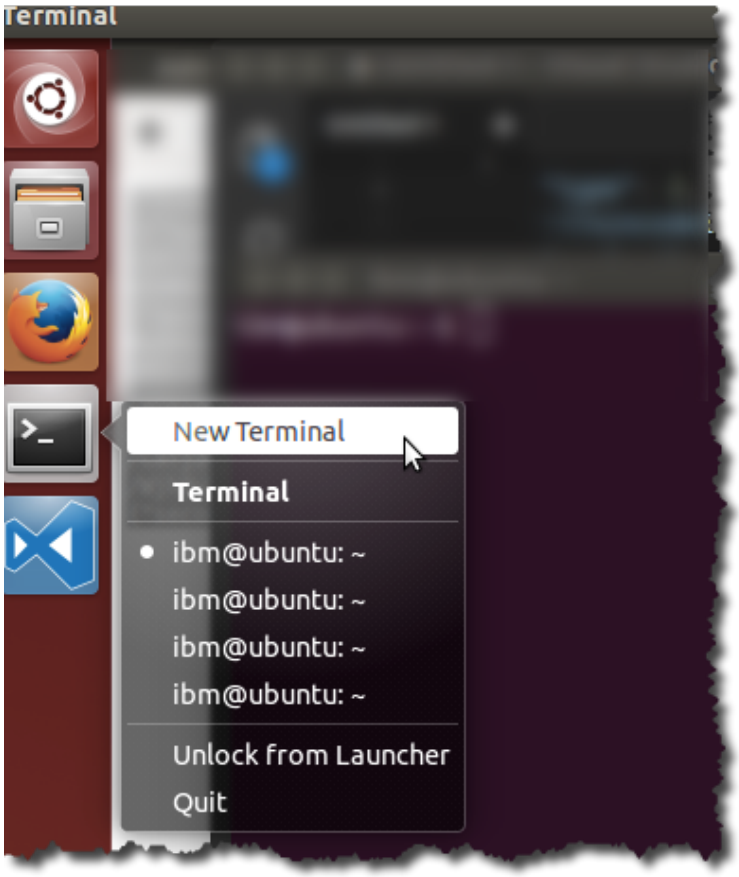
Although not all the data is fully readable (because it is actually encoded in another format), enough is readable to be able to understand the content. The first long number is the (base64 decoded) chaincodeID, and you will also be able to read the name of the chaincode function that was invoked along with any string parameters passed to it. In this example for block 82, the function was "manufacturer_to_private" and the parameters were "Beechvale_Group" and "WW4288939". This is a transaction from a manufacturer to transfer vehicle ID WW4288939 to the "Beechvale Group" as shown in the regulator view:

__19. If you want to delve a little deeper you can even base64 decode the "cert" part of the block and in the data you will be able to see the certificate holder is "Jaguar_Land_Rover" who initiated the transaction as shown above.

Section 3. Administering the blockchain using the Terminal

In this section we will see how to start and stop the peer.

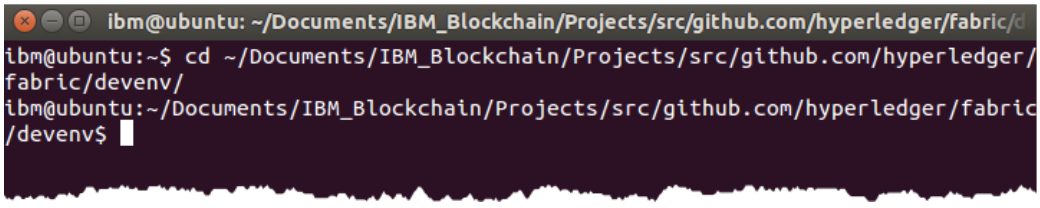
__20. Open a new Terminal window by right clicking the Terminal application on the left hand side of the window and selecting '**New Terminal**'. (**ctrl+alt+t**)



__21. Change the directory by typing

```
cd ~/Documents/IBM_Blockchain/Projects/src/github.com/hyperledger/fabric/devenv
```

Tip: Use the tab button to autocomplete folder names.



__22. Type

```
vagrant ssh
```

__23. Wait for vagrant to load.

Vagrant is a virtual OS *inside* the lab virtual machine that is used to compile and run the Hyperledger Fabric. Having a nested virtualisation environment helps ensure consistency in the runtime environment, allowing you to see similar compilation and runtime results whether you are developing on MacOS, Windows, Linux or whatever. The Vagrant VM is also based on Ubuntu Linux.

```
ibm@ubuntu:~/Documents/IBM_Blockchain/Projects/src/github.com/hyperledger/fabric
/devenv$ vagrant ssh
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-86-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Fri Nov  4 11:21:30 UTC 2016

  System load:  0.79               Processes:              93
  Usage of /:   11.6% of 38.75GB   Users logged in:        0
  Memory usage: 2%                 IP address for eth0:    10.0.2.15
  Swap usage:   0%                 IP address for docker0: 172.17.0.1

  Graph this data and manage this system at:
    https://landscape.canonical.com/

New release '16.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Oct 14 14:16:21 2016 from 10.0.2.2
vagrant@hyperledger-devenv:v0.0.10-3e0e80a:~$
```

__24. We will now look at the 'peer' command which allows you to administer a validating peer (the location of which is defined in a configuration file: core.yaml). The command is run inside the vagrant virtual machine.

__25. In the Vagrant environment, type

```
peer
```

This will show you the command syntax for the peer command.

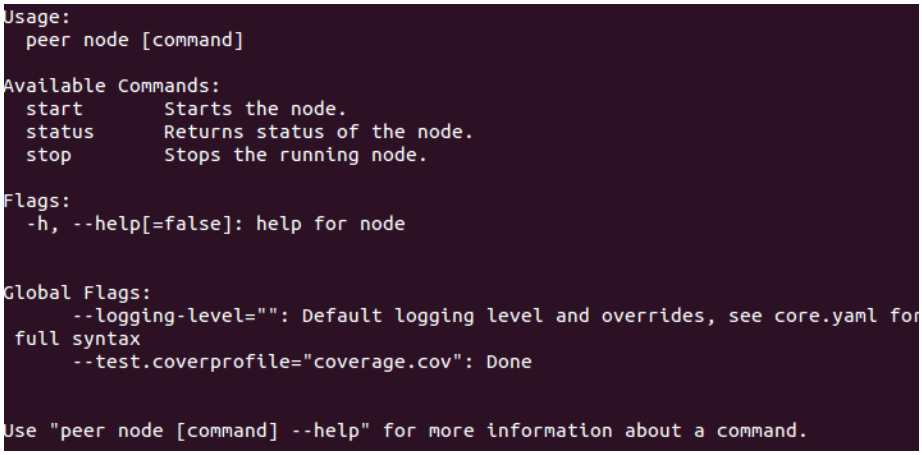
```
Usage:
  peer [command]

Available Commands:
  node        node specific commands.
  network     network specific commands.
  chaincode   chaincode specific commands.
  help        Help about any command

Flags:
  -h, --help[=false]: help for peer
      --logging-level="": Default logging level and overrides, see core.yaml for
 full syntax
      --test.coverprofile="coverage.cov": Done


Use "peer [command] --help" for more information about a command.

12:57:14.251 [main] main -> INFO 003 Exiting.....
vagrant@hyperledger-devenv:v0.0.10-3e0e80a:~$
```

This syntax shows that there are several variants of the peer command depending on which areas of the validating peers you wish to control.

__26. Type

```
peer node
```

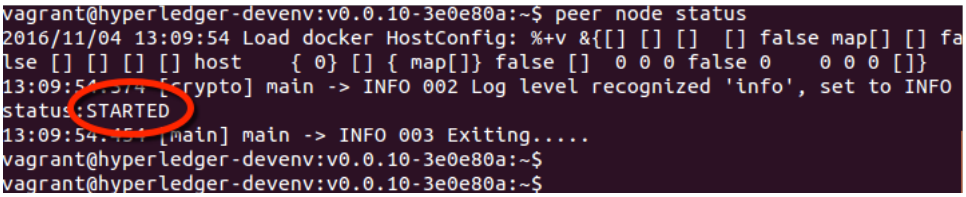
to see the commands that can be issued against the validating peer.



__27. Type

```
peer node status
```

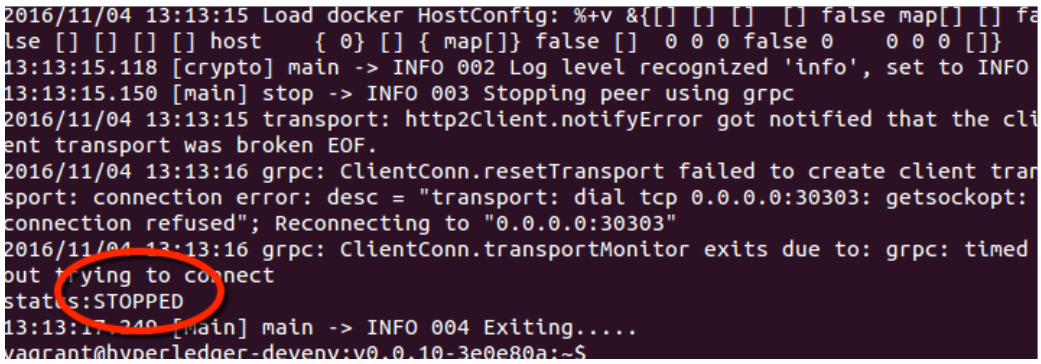
to see the status of the node. It should return "**STARTED**".



__28. Type

```
peer node stop
```

to stop the node. Look for the word "**STOPPED**" to confirm that the peer has indeed stopped.



__29. Type

```
peer node start
```

This will start the node again. Note that the command does not return control to the terminal window while the node is running.

In the next lab “Blockchain Unchained” we will look in more depth at the peer command for deploying, invoking and querying chaincode.

Congratulations on completing the lab “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.

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	
System z	*Tivoli*	*WebSphere*	*Workplace*	*System p*	

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

Please Recycle

IBM Software