

Watson IoT Integration with Blockchain – Part 2

Rajesh K Jeyapaul
(jrkumar@in.ibm.com)
Cloud solution Arch / Startup Mentor

In Part 1, we have seen how to generate IoT data using simulator and publishing it to Watson IoT platform

- <https://github.com/ECODIndia/iot-blockchain/> Watson-IoT-device-simulator.pdf

In this section, we will see how to deploy Blockchain fabric and create chaincode using Bluemix

Lab Architecture

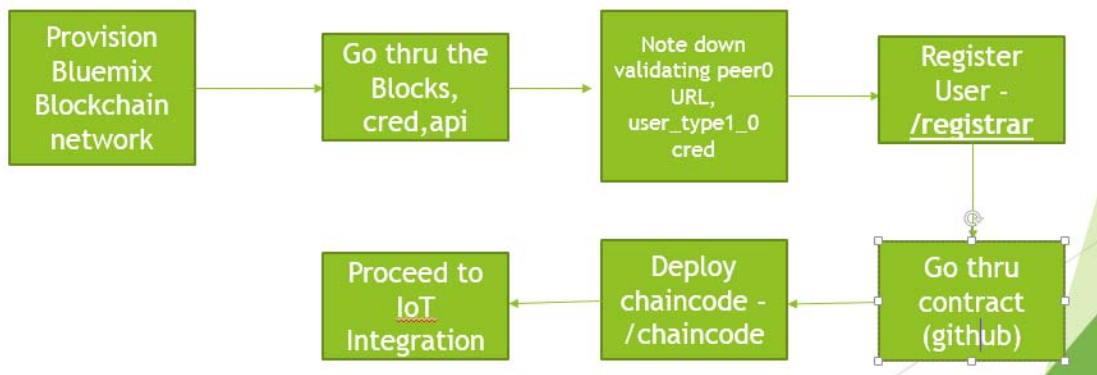
Part 1:

IoT Data to Watson IoT platform using Simulator

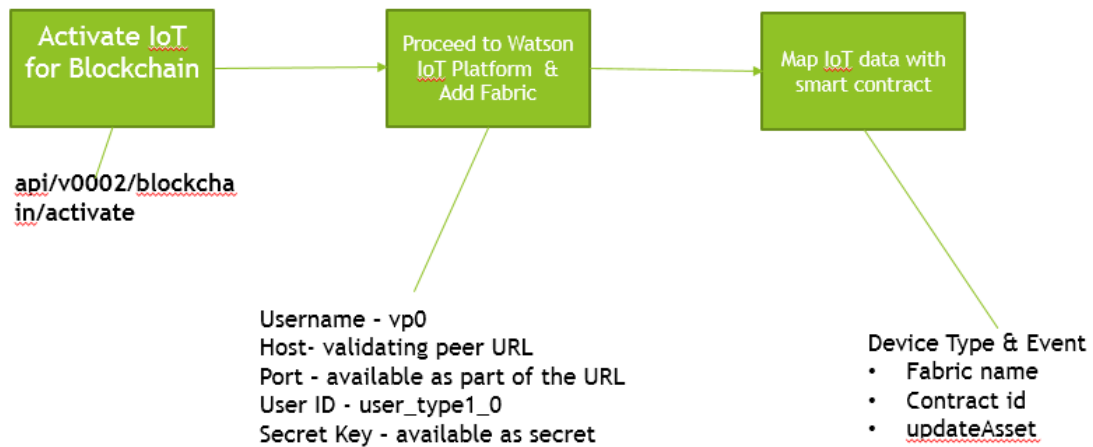


Part 2:

Deploying Bluemix Blockchain network

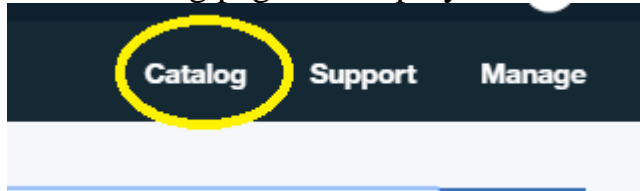


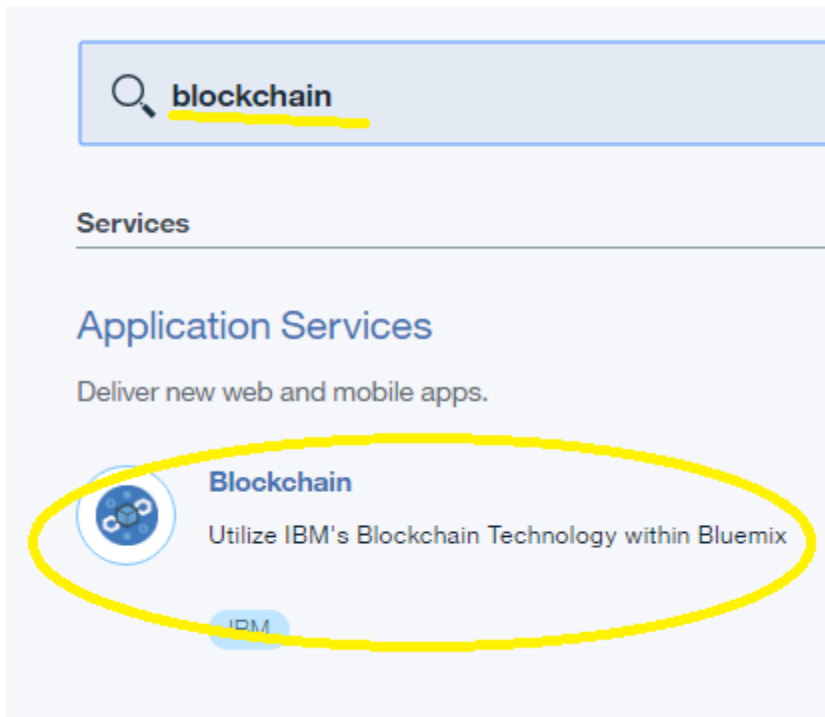
Configuring and Integrating Watson IoT platform with Blockchain



Step 1 : Deploy Blockchain network (Fabric)

- Login to Bluemix
 - Bluemix.net
- Go to catalog page and deploy Blockchain network



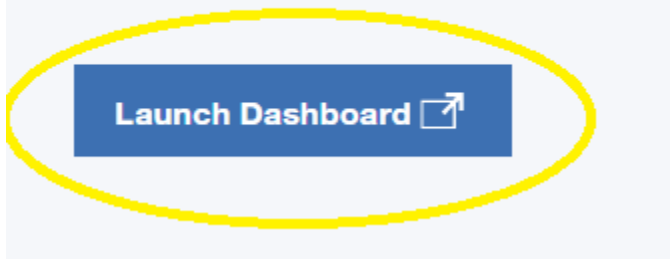


- Take the default service name and proceed to create:

A screenshot of the service creation form. It features a 'Service name:' label followed by a text input field containing 'Blockchain-6s'. Below this is a 'Credential name:' label. A yellow oval highlights the 'Service name:' label and the input field. At the bottom of the form is a blue button labeled 'Create'.

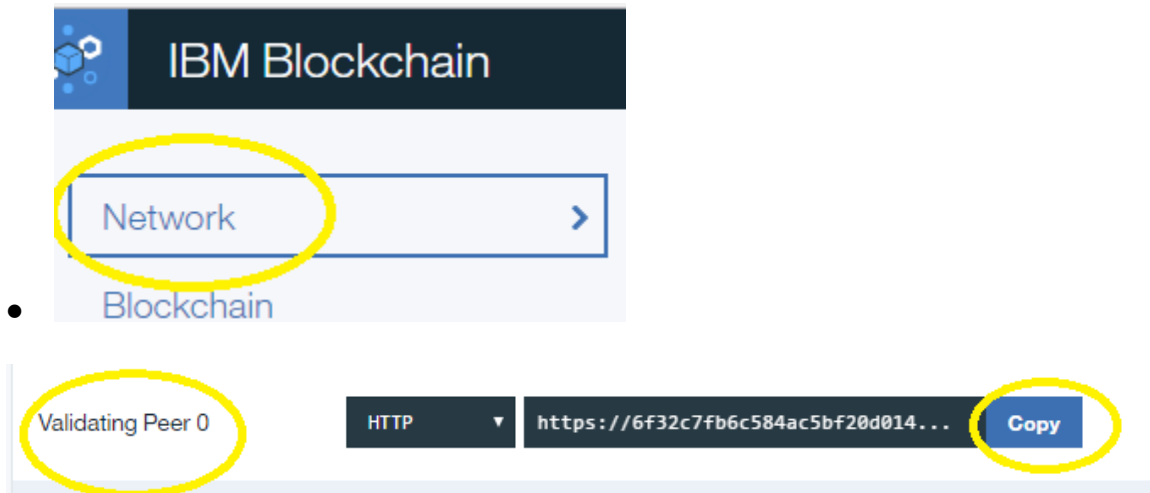
- Wait for couple of minutes and once the service is deployed, go ahead and launch the dashboard
- Go to dashboard and double click on the Blockchain service then Launch the dashboard

to the IBM Blockchain S



Step 2 :As a next step , note down the following credentials:

- URL & Port of the validating peer ,
- user name and
- secret ID



URL & PORT - <https://xxxxxxxxx-vp0.us.blockchain.ibm.com:500x>

User name and secret ID can be retrieved from Blockchain service dashboard as shown:

All items

Blockchain-webse

lanage

Service credentials

Co

Service credentials

New credential +

<input type="checkbox"/>	KEY NAME	DATE CREATED	ACTIONS
<input type="checkbox"/>	Credentials-1	Mar 15, 2017 - 11:20:01	View credentials ▲

```
{
  "peers": [
```

```
},
{
  "enrollId": "user_type1_0",
  "enrollSecret": "xxxxxxx3b3b",
  "affiliation": "group1",
  "username": "user_type1_0",
  "secret": "xxxxxxx3b"
},
{
  "enrollId": "user_type1_1",
```

User Name – user_type1_0

Secret – xxxxxxxx

Step 3: Register user with Blockchain network

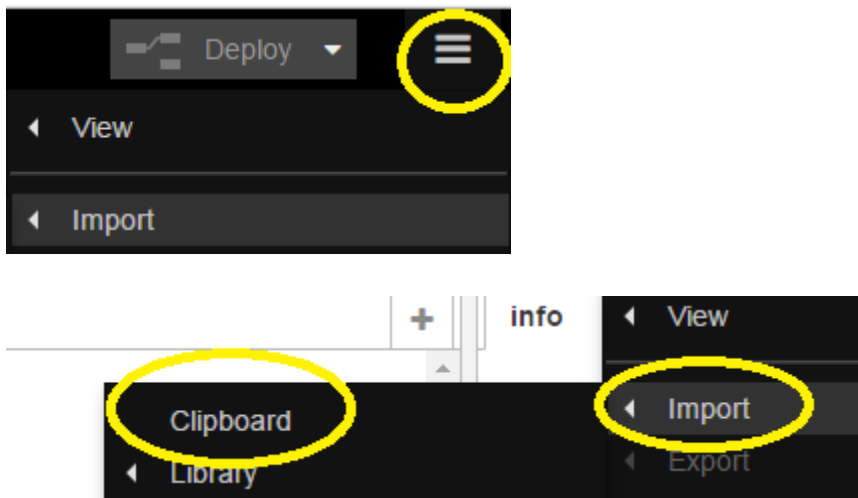
Input : validating peer URL and port

REST API : /registrar (POST)

Output: Successful registration with Blockchain network

Note:

- This functionality is done using nodejs scripts. Hence we need to use the NodeRed environment provisioned earlier (as part of part 1 - <https://github.com/ECODIndia/iot-blockchain/Watson-IoT-device-simulator.pdf>)
- Complete part1 to get the NodeRed editor OR deploy the Node RED Starter boilerplate from Bluemix to get the node red environment
- Download the node script from github
 - https://github.com/ECODIndia/iot-blockchain/blockchain_chaincode_activate.json
- Go to node red editor and import the json code
 - In my case - <https://iot-blockchain-integration.mybluemix.net/> , deployed as part of Part 1



Import nodes

```

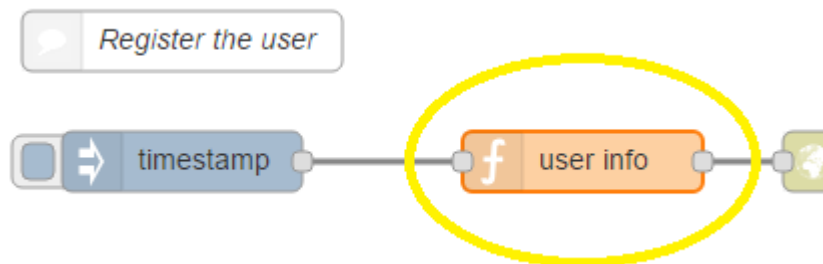
{
  "active": true,
  "console": "false",
  "complete": "true",
  "x": 670,
  "y": 440,
  "wires": []
},
{
  "id": "dec47d9b.375d4",
  "type": "comment",
  "z": "51a96ef5.4a80b",
  "name": "enable block chain in Watson IoT Platform",
  "info": "",
  "x": 280,
  "y": 400,
  "wires": []
}

```

Import to current flow new flow

Cancel Import

- Now the node code is installed. Proceed to register the user with blockchain network
- **Register the User**
 - Double click the “_user info” node and modify the “copiedURL with the URL assigned to your blockchain which was captured in the previous step (URL)



- _____
- var copiedUrl = <https://...vp0.us.blockchain.ibm.com:500x>
- provide the Id and Secret as well, the details which was noted in the previous step (User Name and Secret)
- msg.payload = {
 "enrollId": "user_type1_0",
 "enrollSecret": "xxxxxxx"
 }

Step 4: Deploying chaincode

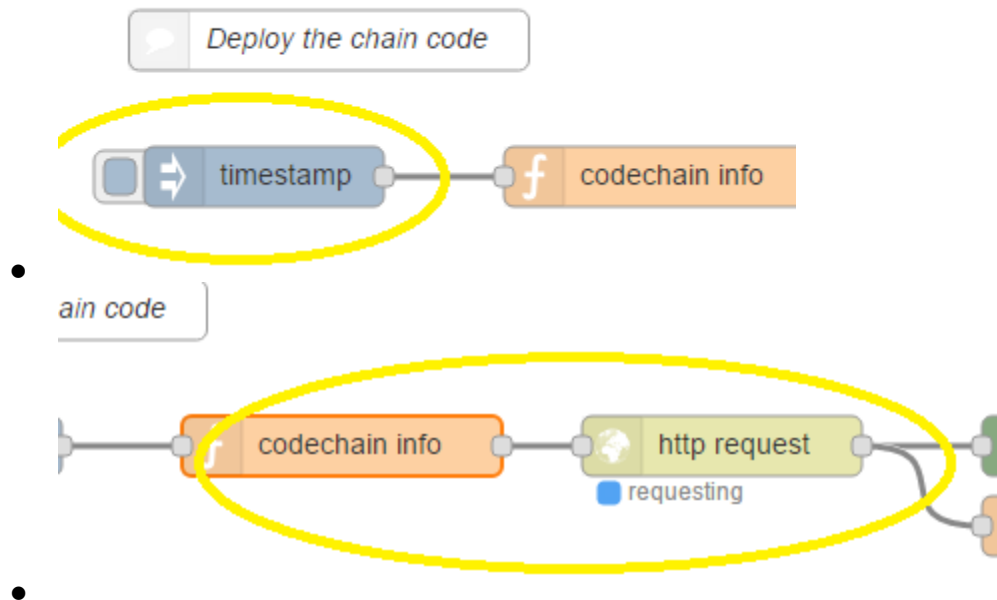
Input: Smart contract (Example - https://github.com/ibm-watson-iot/blockchain-samples/contracts/basic/simple_contract.0.6)

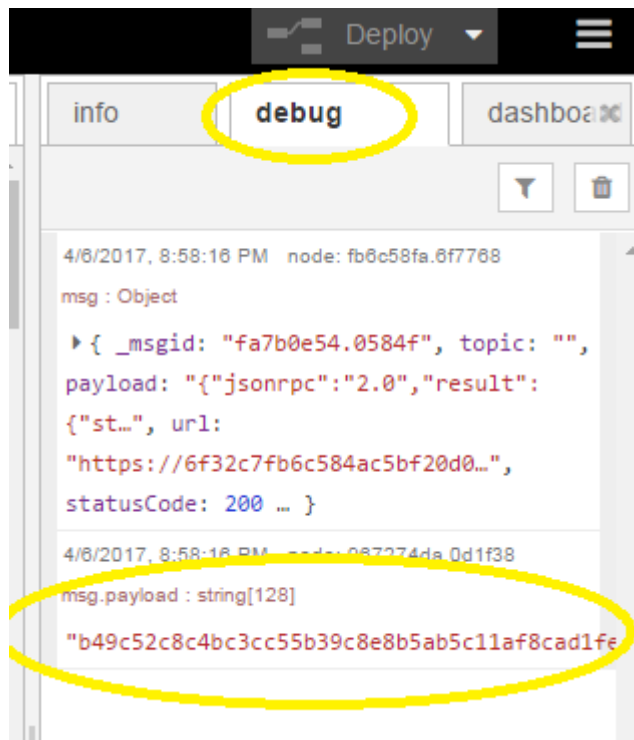
HTTP API - /chaincode

Method – Deploy

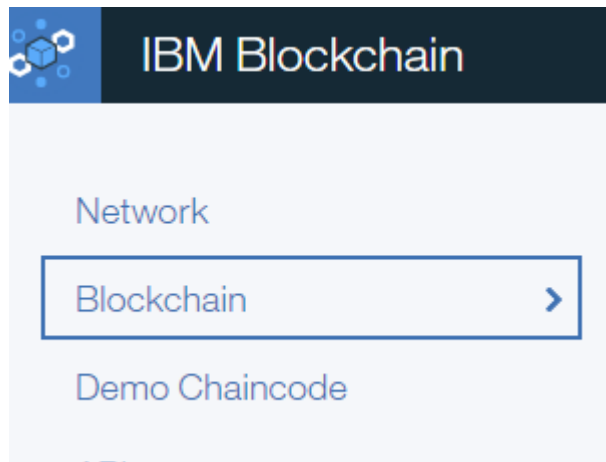
Output – chaincode ID (handle to refer the smart contract)

- In the same node red Window where the user registration is completed, proceed with injecting the next flow “Deploy the chaincode”





- As shown above, this return the chaincode , which indicates the smart contract is deployed.
- This can be verified in the Blockchain console as well as shown below:



Block Activity									
	Time	Block #	Deployments	Invocations	Date	Type	UUID	Chaincode ID	Payload
	24min 30sec ago	20	1	0	04/06 04:05pm UTC	DEPLOY	b49c52c8c4bc3cc55b39c8e8b5ab5c11af8cad1fe5088088ae32059a4d21d94b7608288ab44bb5cd55fa64134000c474a0d242572f36508138a91724c2b2dcb7	na	{ "init": { "version": "1.0" } }
	1hr 3min ago	19	0	0					
	2hr 1min ago	18	0	1					

Note down the contact ID for further usage @ step 7.

Step 5: Integrating IoT with Blockchain

This is very important step , connecting both IoT and Blockchain

- **Activate Blockchain @ Watson IoT Platform:** Use the node red flow to activate the Blockchain at Watson IoT Platform


Input : ORG , username and password

Ouput: Blockchain activation @ Watson IoT platform

- Launch the Watson IoT platform and note down the following
 - organization ,
 - username (apiKey) and
 - password (apiToken)
- The Watson Iot platform was provisioned as part of our earlier activity (Part1)
-


○

Connections




IoT-Blockchain-integration

● Running IoT-Blockchain-integration.mybluemix.net



IoT-Blockchain-integrati...
Cloudant NoSQL DB
Lite

View credentials [Docs](#)



IoT-Blockchain-integrati...
Internet of Things Platform
iotf-service-free

View credentials [Docs](#)

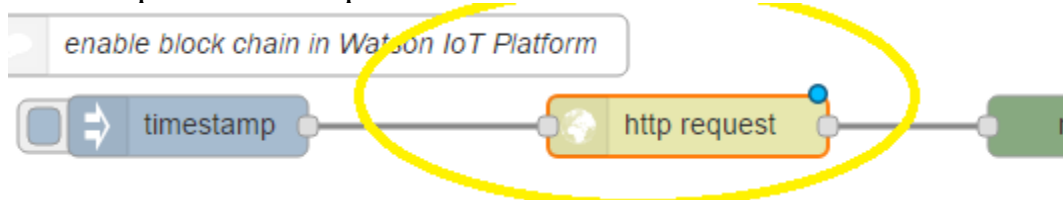
- Note down “org” , “api key” and “apiToken”

IoT-Blockchain-integration-iotf-service

Service credentials

```
{
  "iotf-service": [
    {
      "credentials": {
        "iotCredentialsIdentifier": "a2g6k39sl6r5",
        "mqtt_host": "siyzim.messaging.internetofthings.ibmcloud.com",
        "mqtt_u_port": 1883,
        "mqtt_s_port": 8883,
        "http_host": "siyzim.internetofthings.ibmcloud.com",
        "org": "siyzim",
        "apiKey": "siyzim-h2w4djqpj",
        "apiToken": "15hV6 1d1qV/V 3ts"
      },
      "syslog_drain_url": null,
      "label": "iotf-service",
    }
  ]
}
```

-
- Go back to the node Editor and update the ORG, username and password details
 - <ORG> in the URL
 - “apiKey” as “Username”
 - “apiToken” as “password”



Method: GET

URL: https://<ORG>.internetofthings.ibmcloud.com/api

☐ Enable secure (SSL/TLS) connection


☒ Use basic authentication

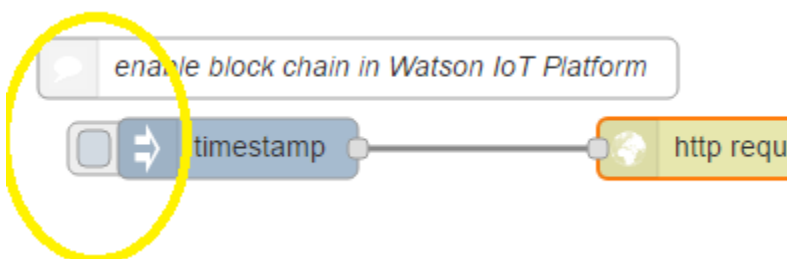
Username: tsiyam-h3rr1d3r, i

Password:

Return: a UTF-8 string

Name: Name

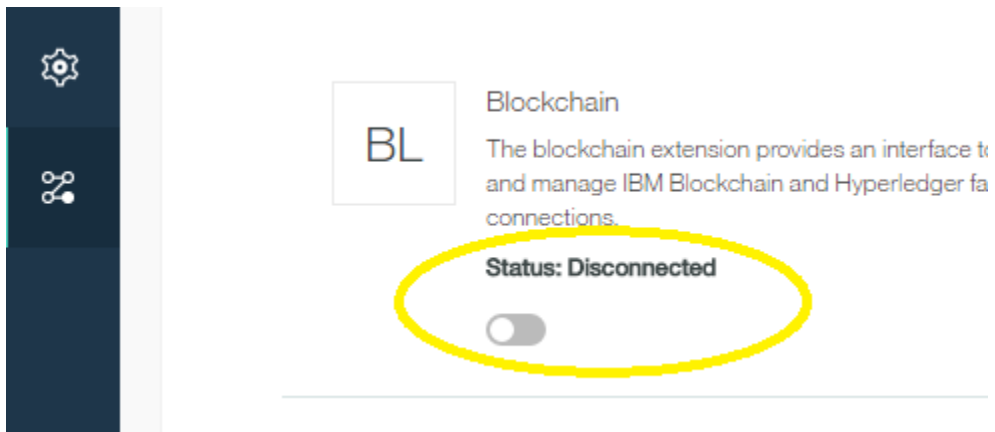
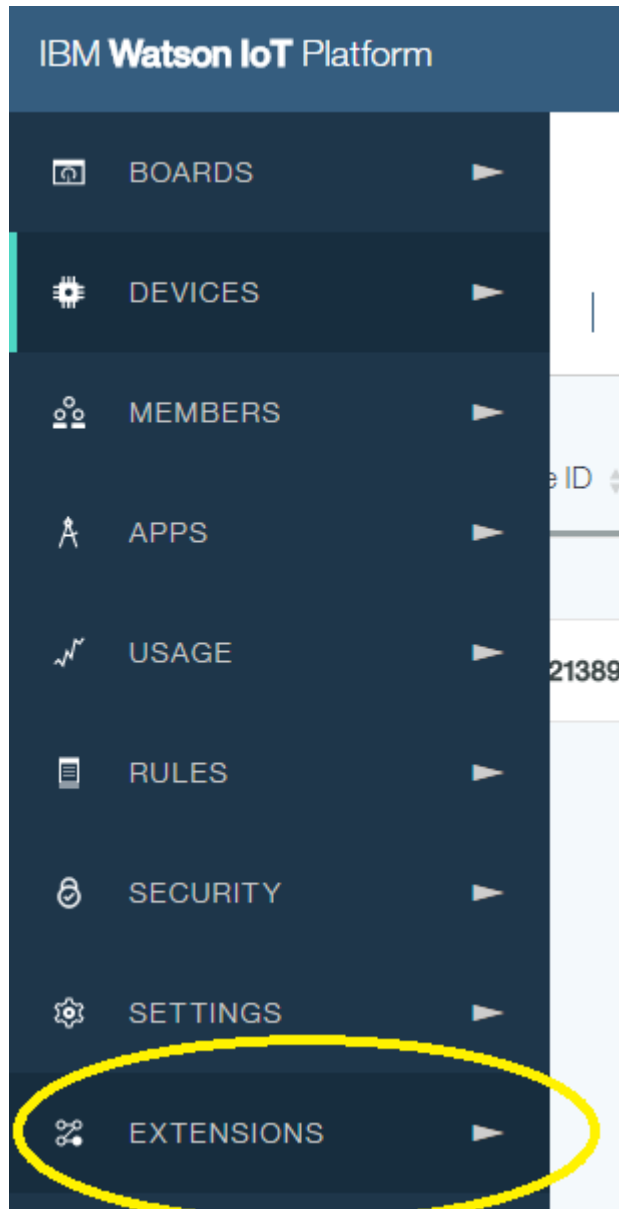
- Deploy the node
- 
- Inject it to invoke the GET URL



- Debug console gives the successful message:



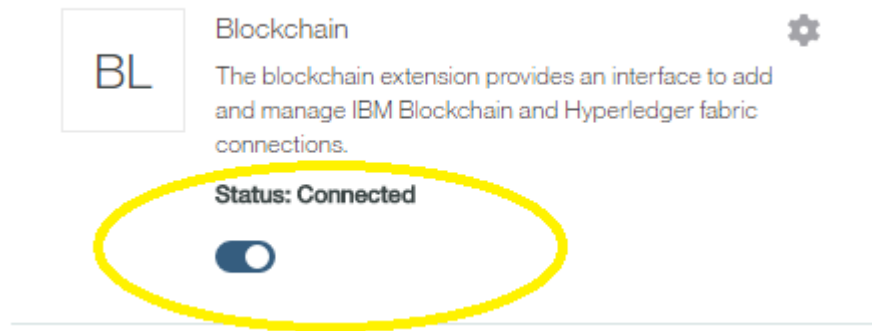
-
- To validate it, Launch the Watson IoT platform and click on Extensions..



- **Step 6: Provide Blockchain Fabric information to Watson IoT platform**

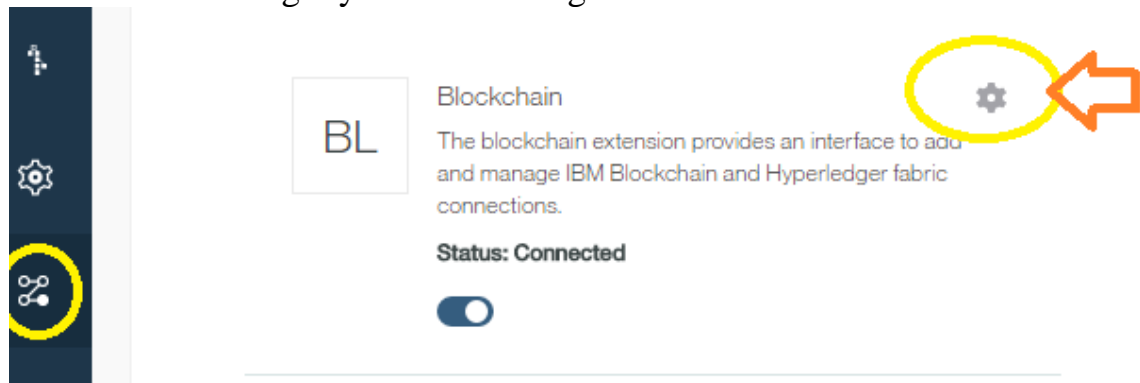
- Turn on the knob to make it connected (below image)

-
-



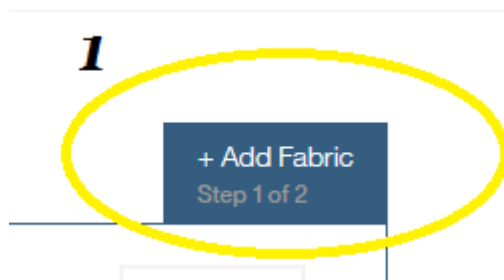
-

- Click on the settings symbol on the right



-

- Provide the Fabric name



-

Fabric

To get started, enter a name to ic

2

myfirstfabric

- Provide username, hostname, port, userID and secret Key all part of the blockchain network which was provisioned. Take this information from the Blockchain credentials (Step 3 above)
- - Username – vp0
 - Host- validating peer URL
 - Port – available as part of the URL
 - User ID - user_type1_0
 - Secret Key – available as secret

Configure Peer

Enter the connection informa

Name



vp0

Host



6f32c7fb6c584ac5b20d0c44381

Port



5001

User ID



user_type1_0

Secret Key



757b7313b

Use TLS?



Finish

Fabric Name

Status

Peer Host

myfirstfabric

✓ online

6f3?c. b?c584ac?bf20d014

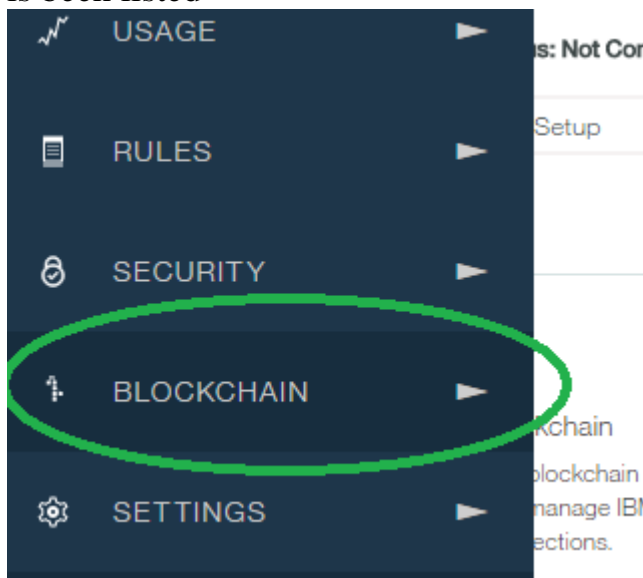
Step 7 : Mapping IoT Data with Blockchain contract

Input: Device details and Fabric,contract ID

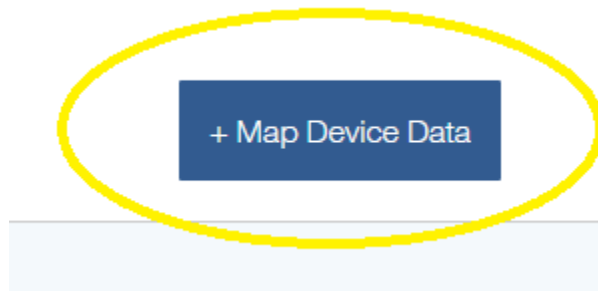
Output: Successful establishment of IoT & Blockchain handshake

In this step , we need to map the IoT data with the smart contract deployed as part of Blockchain

- Refresh the Watson IoT page again to see that new icon “ Blockchain” is been listed



- Select the Blockchain tab in the Watson IoT Platform and click on Map Device Data.



- This helps in defining the Device event and Fabric details for the IoT & Blockchain integration
- For this lab, the device type is “Thermostat” and Event is “getAll”

(This is defined through the IoT simulator as part of previous exercise-part1)

Add Route

Device Type and Event

Select the device type and specify an event that contains the p

Device type*	Thermostat
Event*	getAll

- Provide the Fabric name which we defined in step 6. In this case, since the binding is already done, it will list the available fabric
-

Select Fabric

Select the blockchain fabric that contains the contract to that you want to add the route.

Fabric name*

myfirstfabric

- Now, provide contact name and ID

Add Route

Link Contract

Enter a contract ID and provide a contract name to use with Watson IoT Cloud.

The contract that you map must, at a minimum, support the following

- readAssetSchemas

Contract name*

myfirstcontract

Contract ID*

b43_52c8c4bc3cc55b39c8
79a4d2fd94c7608288ab44'
38f08138a91724c2b2dcb7

- Note: . (contract ID from step 4) or get it from Blockchain dashboard as shown

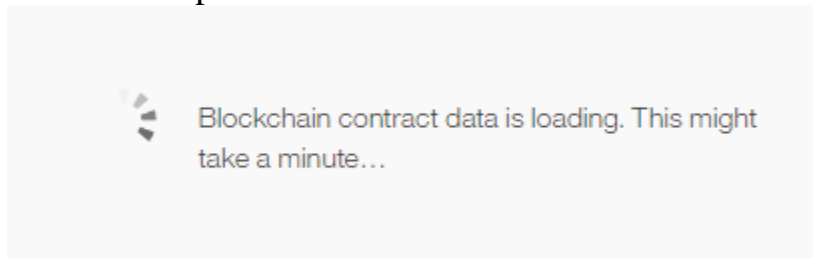
Chaincode ID

contract ID

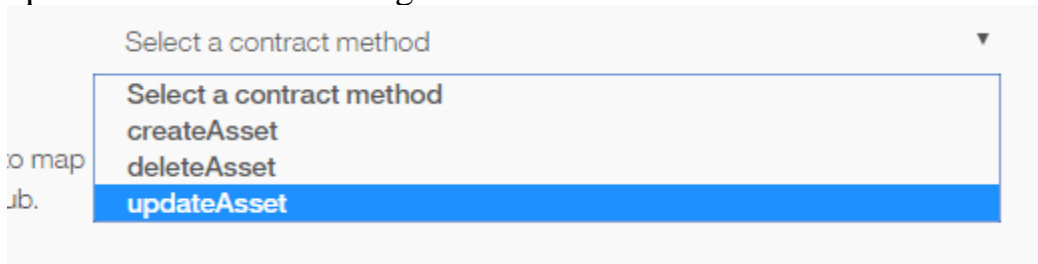
9c52c8c4bc3cc55b39c8e8ab5c1af8ad1fe5088088ae2059a4d2fd94b7608288ab44bb5cd55fa4134000...

Copy

- Wait for couple of minutes



-
- Select the contract method as “updateAsset” since in this lab, we will update the Blockchain ledger with the IoT data



-
- Now, we are ready to define the data template that is coming from IoT device.
- In our lab, we are simulating the following data: provide this details as shown
 - d.assetID
 - d.ActualTemperature

assetID *	d.assetID	
carrier	Enter a device property to map to carrier	
temperature	d.ActualTemperature	

- All set to see the IoT data being updated at Blockchain fabric..Submit the configuration

Block Activity

Time	Block #	Deployments	Invocations	Date	Type	UUID	Chaincode ID	Payload
7sec ago	40	0	1	04/07 07:31am UTC	INVOKE	41cfbb9b-dbf6-4390-8dc5-fb27701fad7a	↑fb49c52c8c4b...	λupdateAsset % { "assetID": "1234", "temperatu re": 23.0 }
9sec ago	39	0	1					
11sec ago	38	0	1					

- Hurrah !! we could see the IoT Data being updated at Blockchain

You have successfully completed the Lab involving integrating IoT data with Blockchain.

For queries reach out to jrkumar@in.ibm.com

Node Red code contribution by – **Hari P Vishwanathan , IoT Developer , IBM** (harihvis@in.ibm.com)

Smart contract supported by – **Srirama Sharma - Blockchain Developer , IBM** , sriramsh@in.ibm.com