This last assignment is again an extension of the first. We will be modifying the last part of this assignment, and also adding a third piece, integrating what we’ve done so far into our private OpenStack cloud.

You should STILL be able to ask your Alexa Skill AT LEAST three questions, but this time we will be asking server related questions. You should be able to ask, for example:

1. How long has my server been powered on?
2. How much ram does my server have?
3. How much free ram does my server have?
4. What kind of network traffic is my server getting right now?

And your Alexa Skill should be able to give an appropriate response, including real-time information about your server.

**Modification of Homework 2**

The ONLY thing you’ll have to do for this part is to “retrain” Watson so that it responds to your new questions. The response doesn’t matter yet.

**Integration with OpenStack**

**High Level Overview:**

Deploy a VM in our OpenStack Cloud and use node-red “in a container” to periodically sent status updates to IBM Cloud.

In IBM Cloud, add the appropriate services to enable IoT device integration AND use node-red in IBM Cloud. Your server should be sending its updates to your IoT device portal, and you should configure node-red in IBM cloud to take those updates, extract the one it needs and use it to “update” the appropriate Watson conversation dialog so that when a user asks a question, they get an answer that includes details about your server.

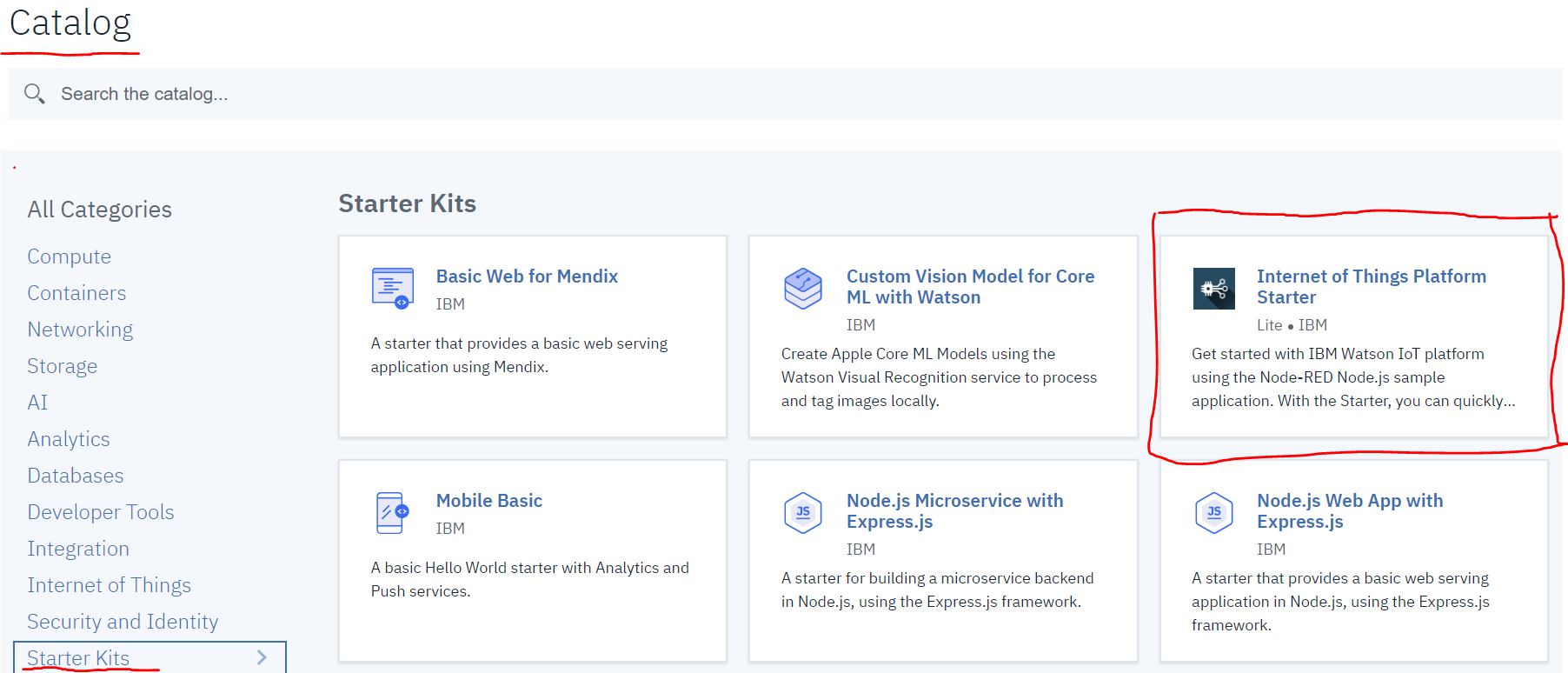
**Step-by-Step**

**OpenStack Side**

1. You will need to deploy a VM using one of the images available.
2. You will need to install a GUI on that VM.
   1. <https://www.techrepublic.com/article/how-to-install-a-gui-on-top-of-centos-7/>
   2. <https://www.cyberciti.biz/faq/ubuntu-linux-install-gnome-desktop-on-server/>
      1. If you follow guide B for ubuntu, I would NOT suggest using tasksel to install the gui
   3. You don’t “need” a gui, technically. This will allow you to connect to your node-red graphically. You could just use port forwarding (like what we do to connect to the cloud) and connect to the node-red editor from your local web browser.
3. You will need to install docker on that VM.
   1. <https://docs.docker.com/install/linux/docker-ce/centos/>
4. You will need to deploy a node-red container on that VM
   1. <https://nodered.org/docs/platforms/docker>
5. You will need to add the following “NODE” to your “palette”: node-red-contrib-device-stats
   1. <https://www.npmjs.com/package/node-red-contrib-device-stats>
6. You will need to add the IoT service in IBM Cloud and you will need to define one device type AND one device in your IoT service in ibm cloud. (*See step two in section: IBM Cloud Side*)
7. You should now be able to complete all of the steps needed in your VM

**IBM Cloud side**

1. I would recommend going to starter kits and selecting the Internet of Things Platform Starter as it will give you all of the pieces you need to make this part work.



1. You will need to select your IoT service (launch app), once it’s been created and define a device type and add a device. (Note: You’re ONLY adding descriptions here. Think of this part as you creating the shell. Your OpenStack VM IS the device. You must create the shell before you can connect to IBM cloud.)
2. Now, you should go to your Node Red (node.js) app. You will need to set this up so that it receives all information from your IoT device (Your VM). For EACH question you can ask your alexa app, you should have a separate function that takes the info you received from your vm and sets the msg.payload to a custom JSON formatted string that represents what you want your dialog node to say now (instead of what it said before). You should forward this to an “assistant workspace manager” node so that it can update the dialog node for you.
   1. <https://cloud.ibm.com/apidocs/assistant?code=node#update-dialog-node>
   2. The link above is the documentation of how you could programmatically update a dialog node (specifically, what fields you may need to include in a JSON string).

## Deliverables (What should you submit to Canvas)

**ONLY THE TEAM LEADER SHOULD SUBMIT ITEMS 1 - 6**

1. Screenshot(s) showing traffic coming from your VM into IBM Cloud IoT service
2. Screenshot of your “flow” in your VM in OpenStack ( I want to see your address bar, so that I can see you are logged into your vm in the web browser. If you used port forwarding, I also want to see your connection info (how you made the tunnel) ).
3. Screenshot of your flow in your IBM Cloud node-red
4. Screenshots of all of your functions (that transform the data received from the VM to the JSON format needed to update the appropriate dialog node).
5. A short video recording showing a group member testing your app on the echo that was assigned to your group
6. A short video recording showing a group member testing your app on your phone, via the alexa app

**EVERYONE MUST SUBMIT GROUP EVALUATION FORM**

1. Group evaluation form