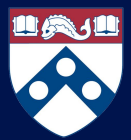
ESE516 CLOUD STARTER

Eduardo Garcia
Spring 2022



Penn Engineering

MQTT Tools

Penn Engineering

Setting up a mqtt broker

- Go to http://www.mqtt-dashboard.com/index.html

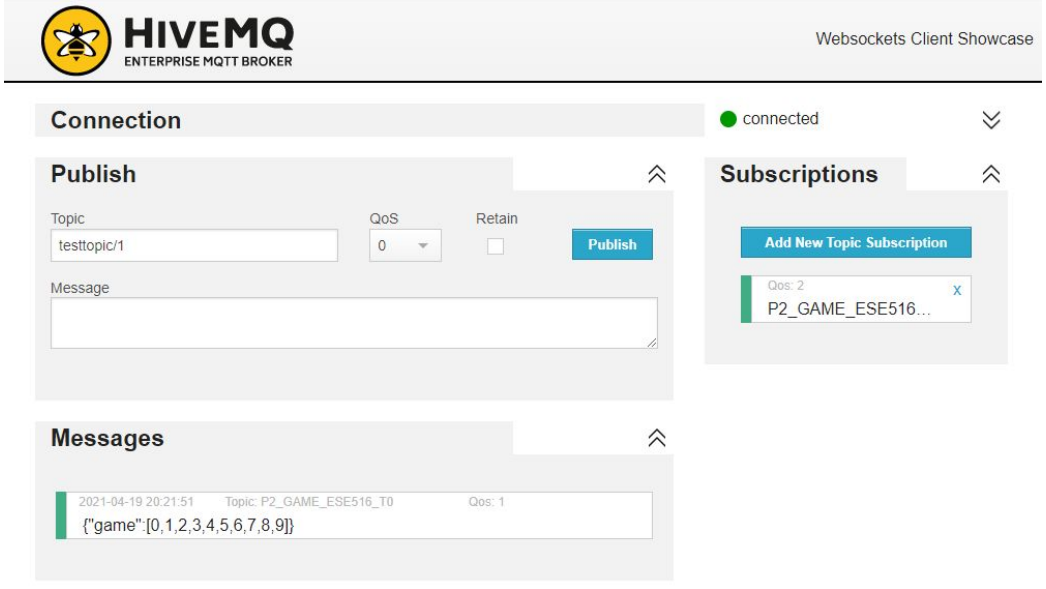
- This is a public MQTT broker. You can find the parameters to join on the right. You do not need to do any action at this point – it is just informative!

MQTT connection settings
Host: broker.hivemq.com
TCP Port: 1883
Websocket Port: 8000

MQTT Client

http://www.hivemq.com/demos/websocket-client/

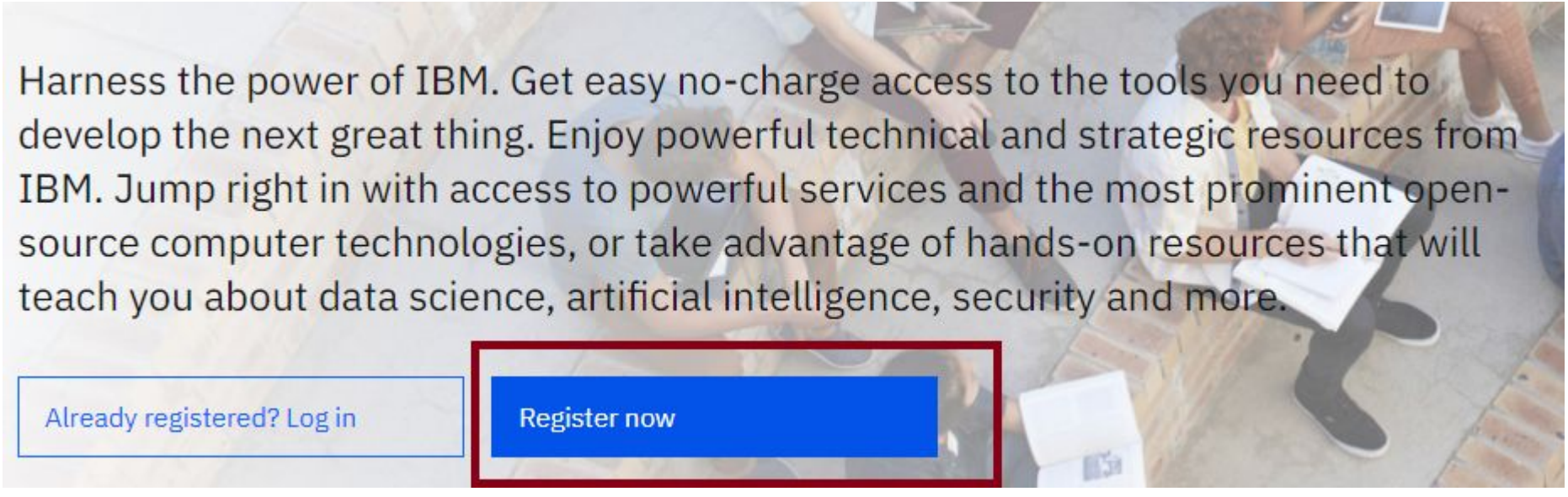
You can use this website client to connect to the MQTT server and publish/subscribe to topics. This can be really useful to determine that data is getting to the system.



SETUP – IBM NODE RED

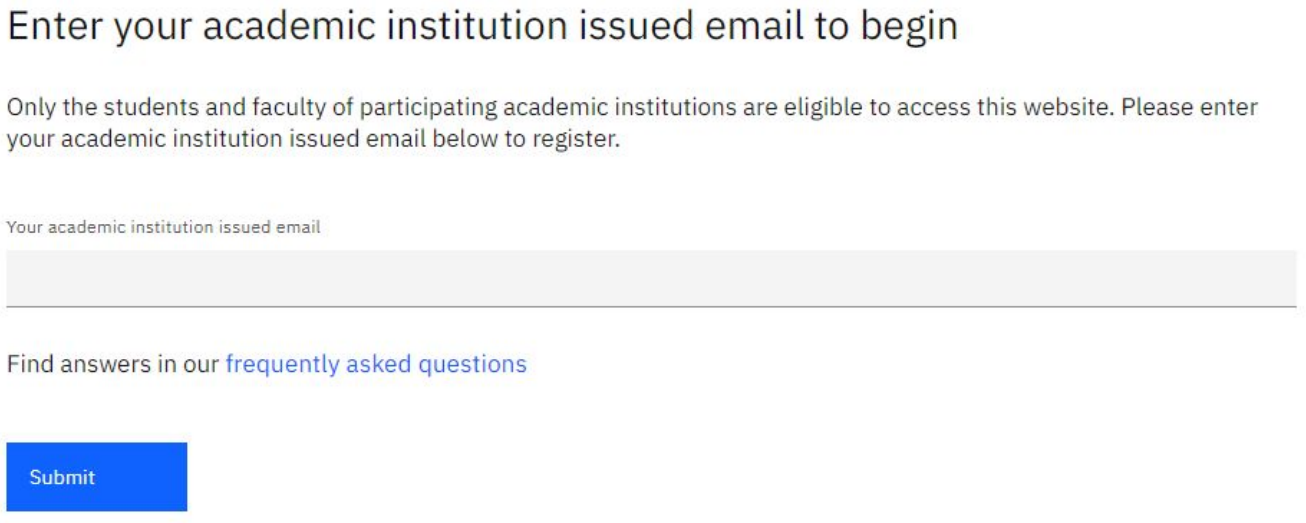
IBM CLOUD ACCOUNT

- Go to ibm.com/academic and hit **Register Now** button



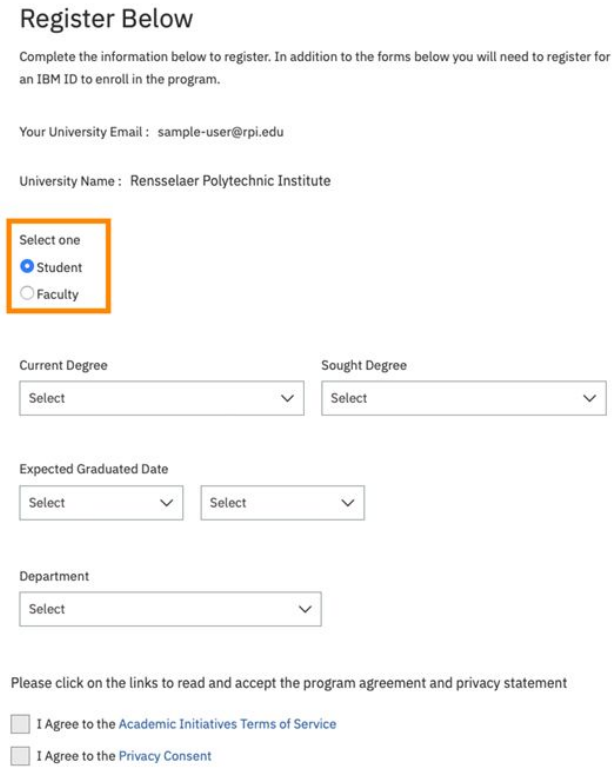
IBM CLOUD ACCOUNT

- Enter your email and fill the information. You should register for an account



IBM CLOUD ACCOUNT

- Enter your email and fill the information. You should register for an account.

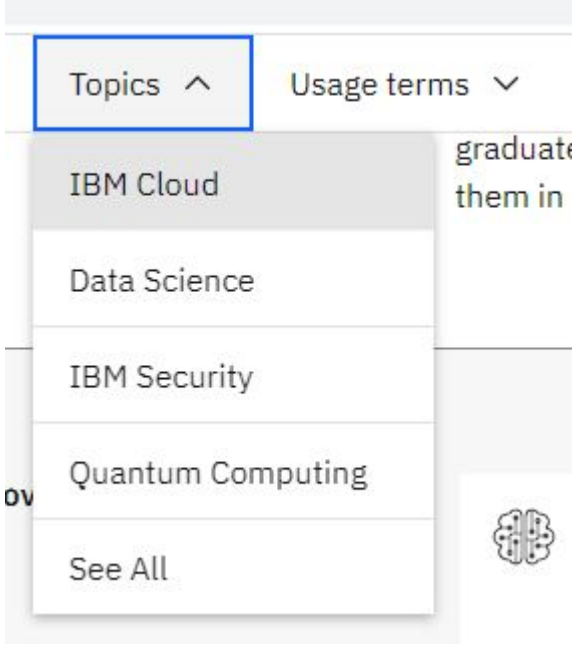


IBM CLOUD ACCOUNT

- Once you have an account, go back to www.ibm.com/academic

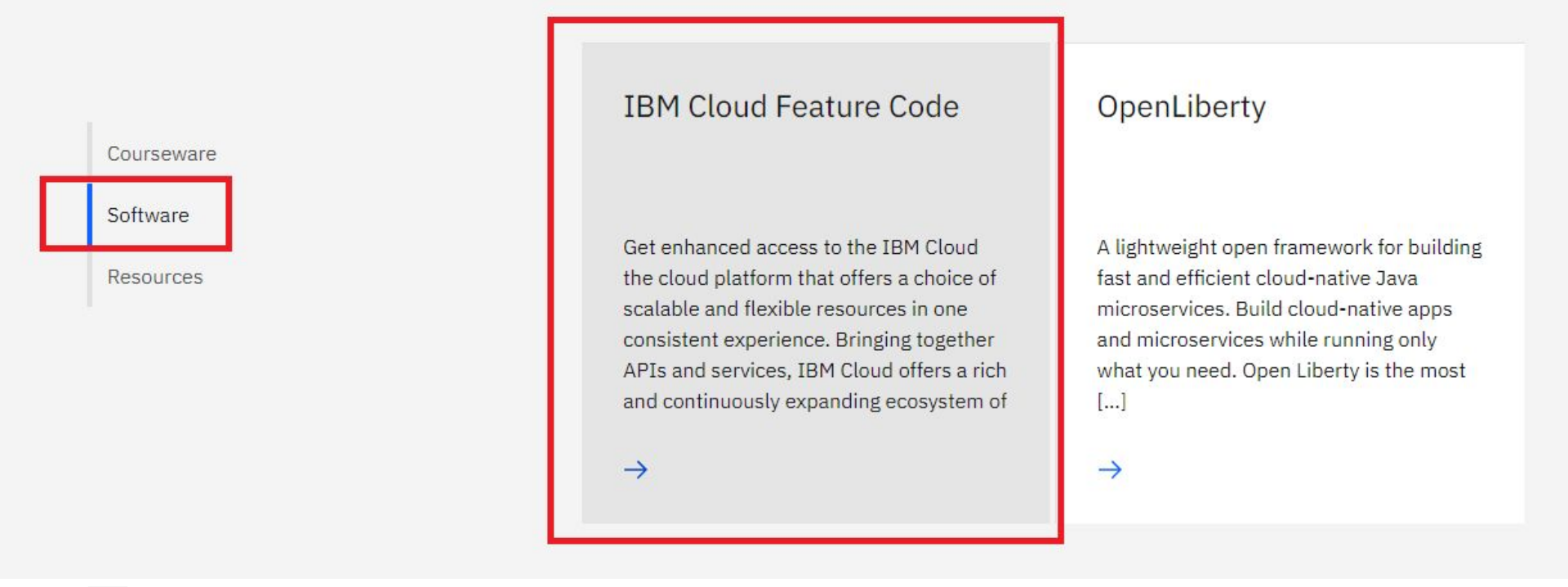
IBM CLOUD ACCOUNT

- Go to **Topics-> Cloud** on the top toolbar of the webpage



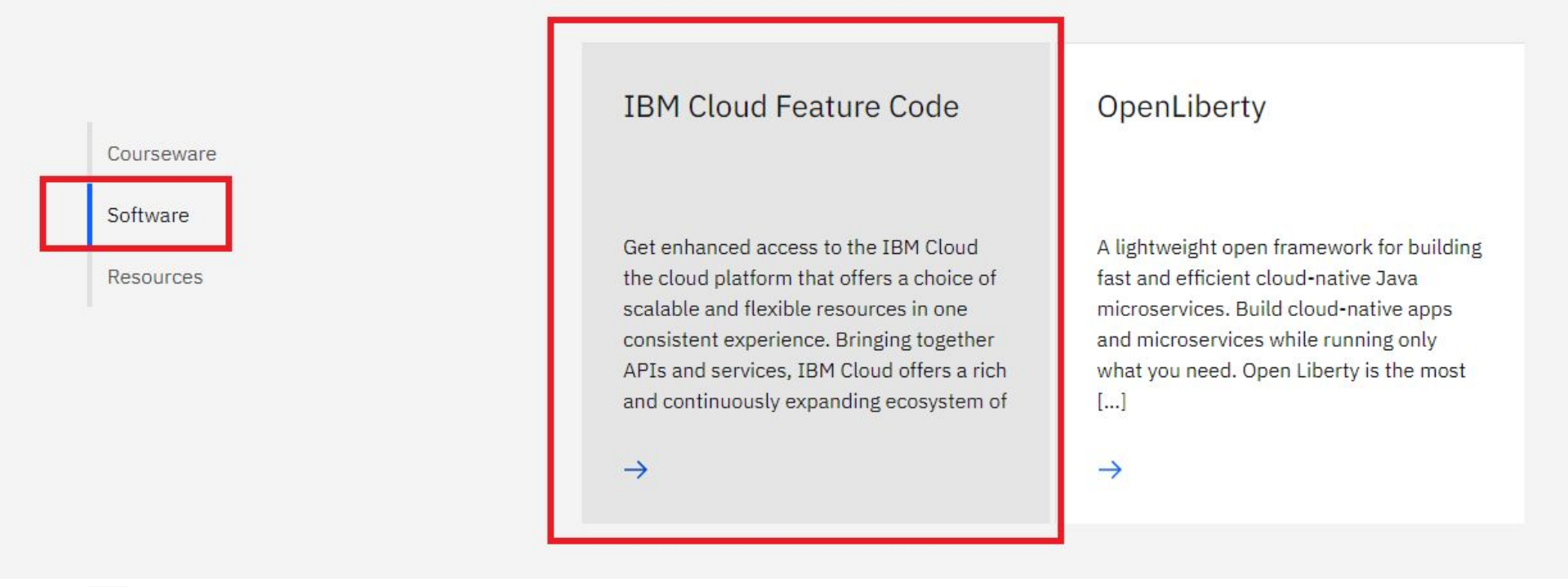
IBM CLOUD ACCOUNT

- Scroll down until you get to the following. Select **Software – IBM Cloud Feature Code**



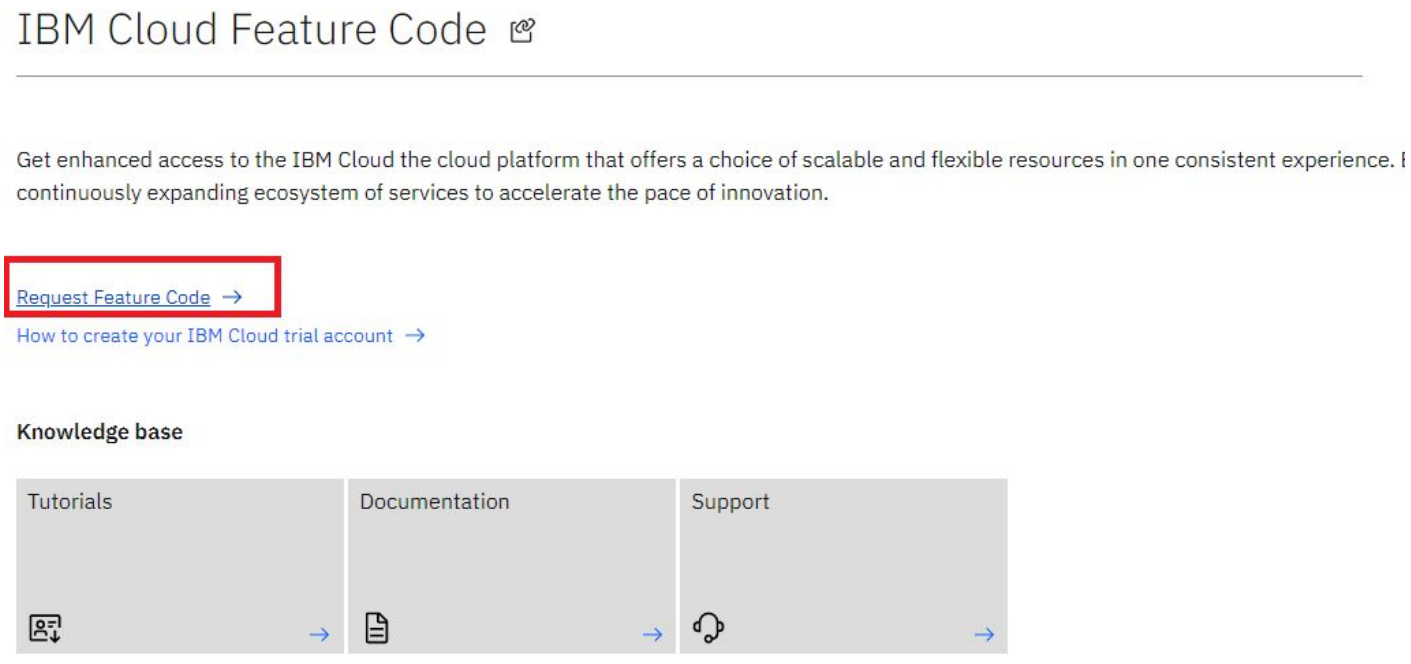
IBM CLOUD ACCOUNT

- Scroll down until you get to the following. Select **Software – IBM Cloud Feature Code**



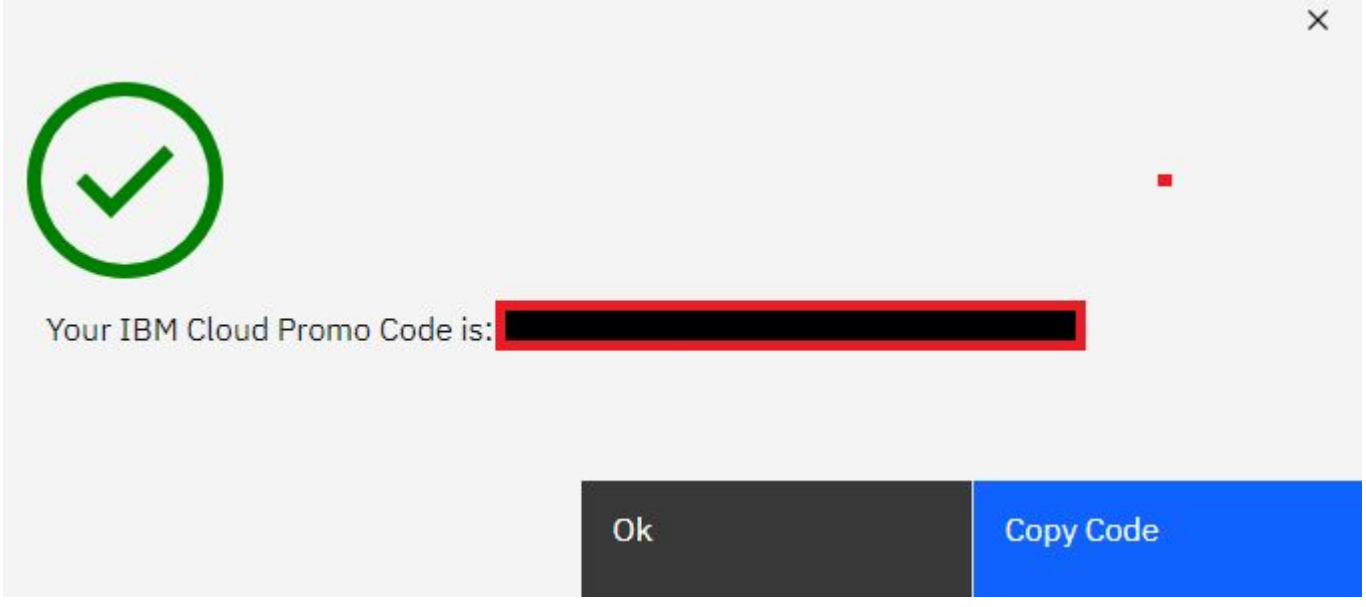
IBM CLOUD ACCOUNT

- Click on **Request Feature Code**



IBM CLOUD ACCOUNT

- Copy the Promo Code.

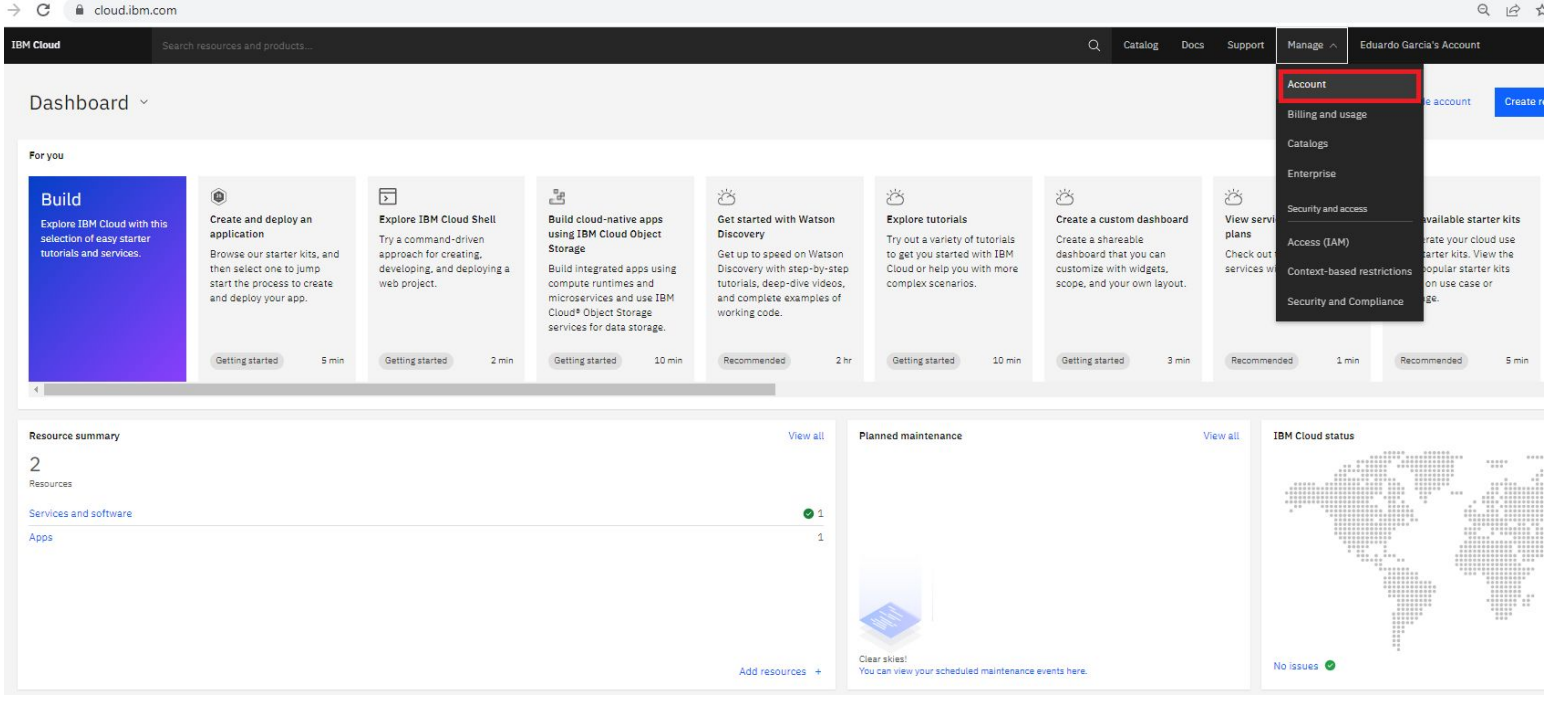


IBM CLOUD ACCOUNT

- Go to https://cloud.ibm.com/ and log in

IBM CLOUD ACCOUNT

- On the top right, select **Manage -> Account**

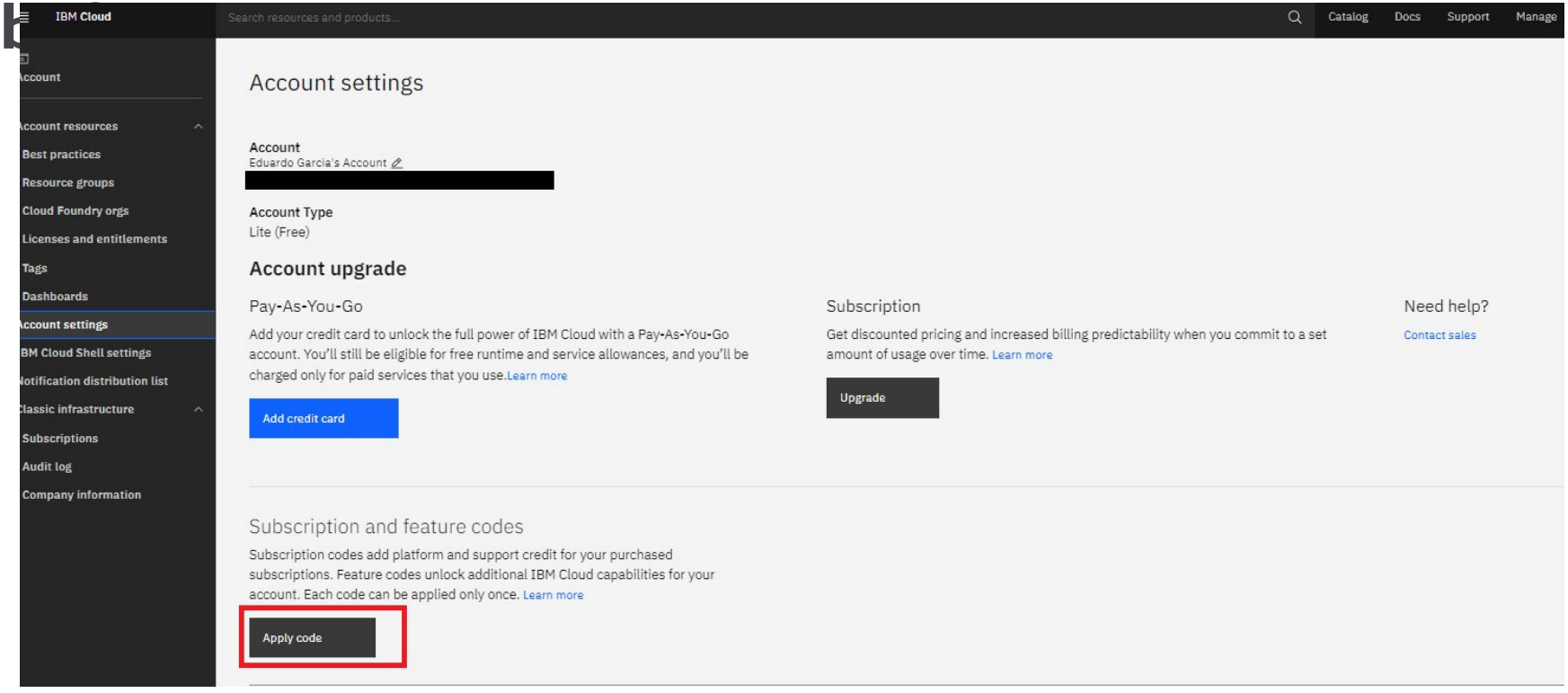


IBM CLOUD ACCOUNT

- Select **Account Settings** on the left toolba

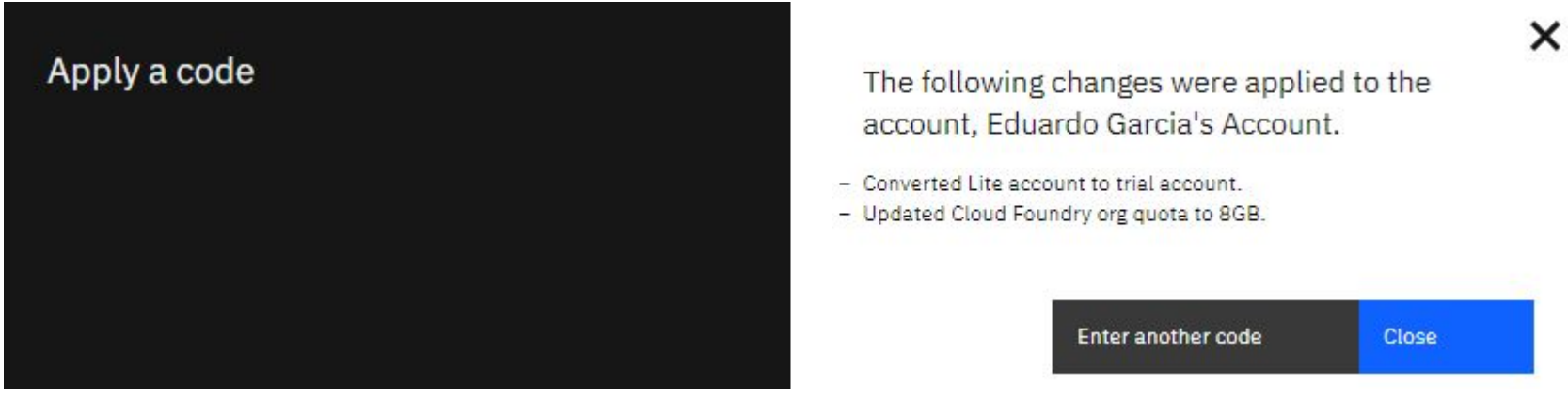
IBM CLOUD ACCOUNT

- Select **Apply code and type the code you copied**



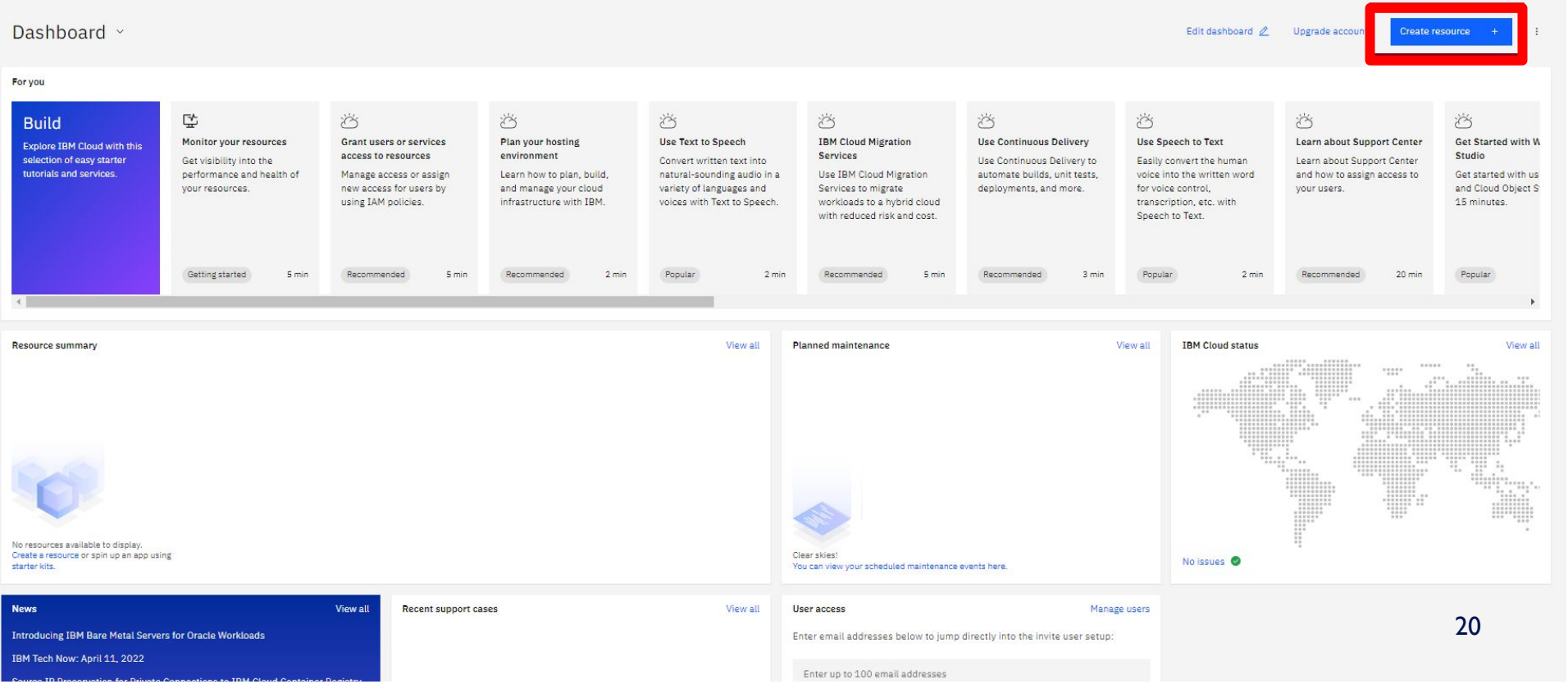
IBM CLOUD ACCOUNT

- You should see a message like the following. NOTE: it might not be exactly the same as my account was made years ago (Lite Accounts do not exist anymore)



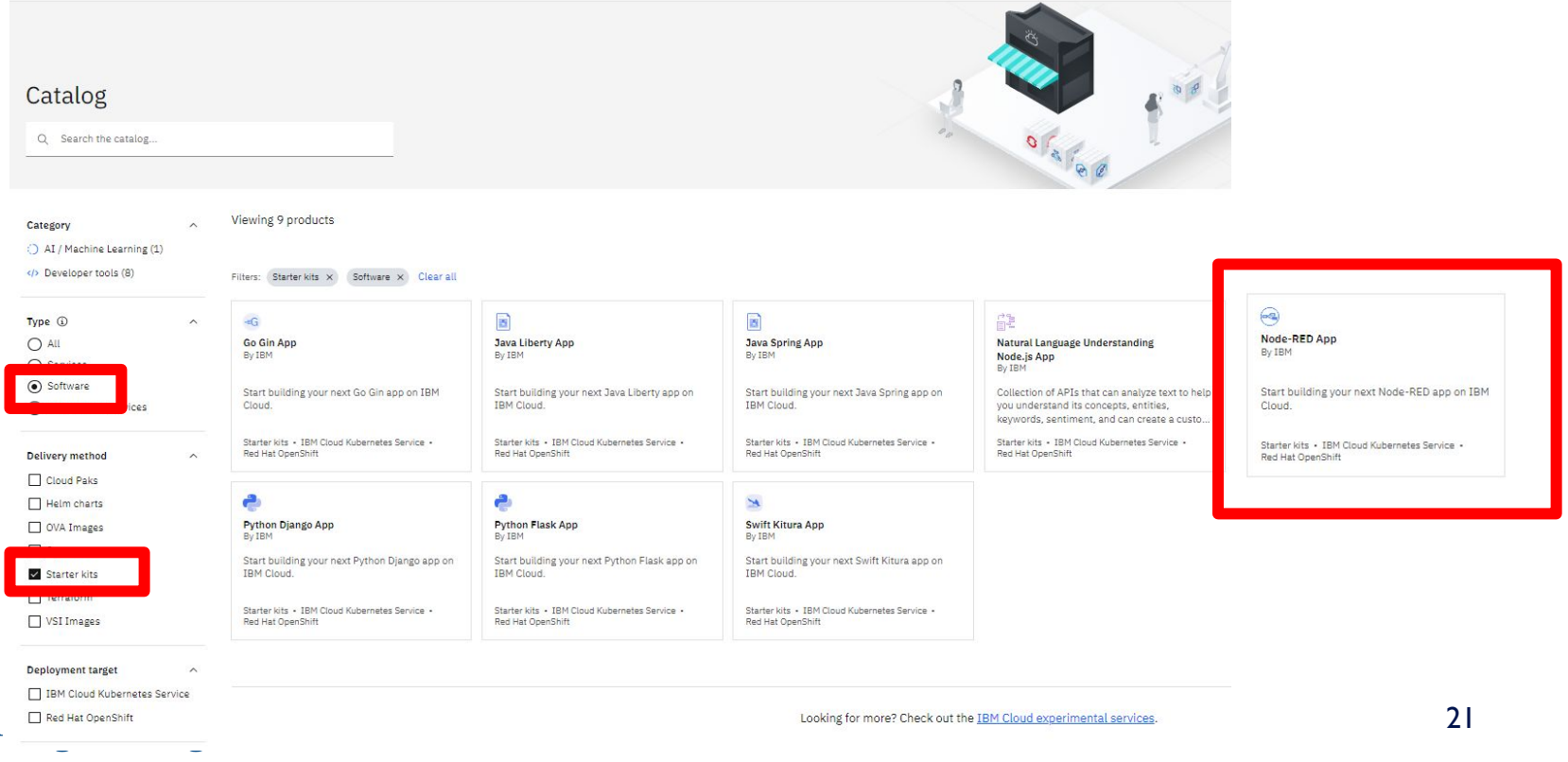
Add resource

- Once you have your account ready, go back to the dashboard. You will be greeted with the IBM Cloud Dashboard. Click on the "**Create**" button on the top right

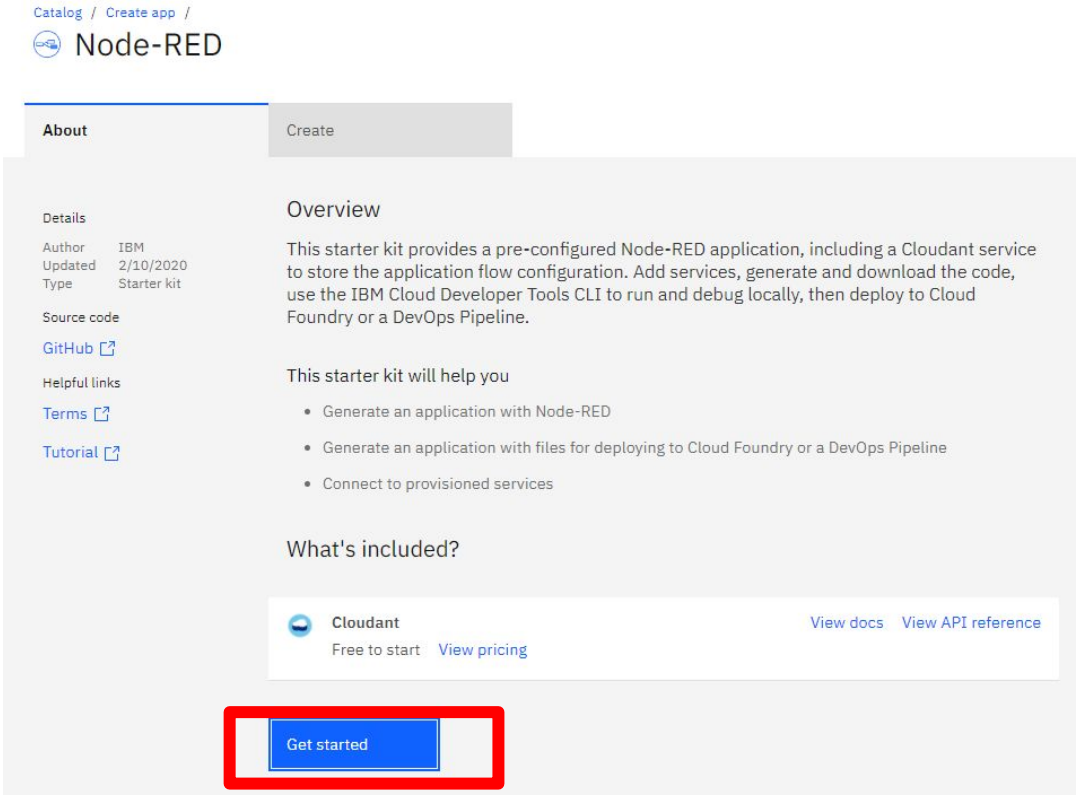


Add resource

- On the Catalog page, select **Type – Software**, and **Delivery Method – Starter Kit**.
- Then, select the **NODE-RED APP**

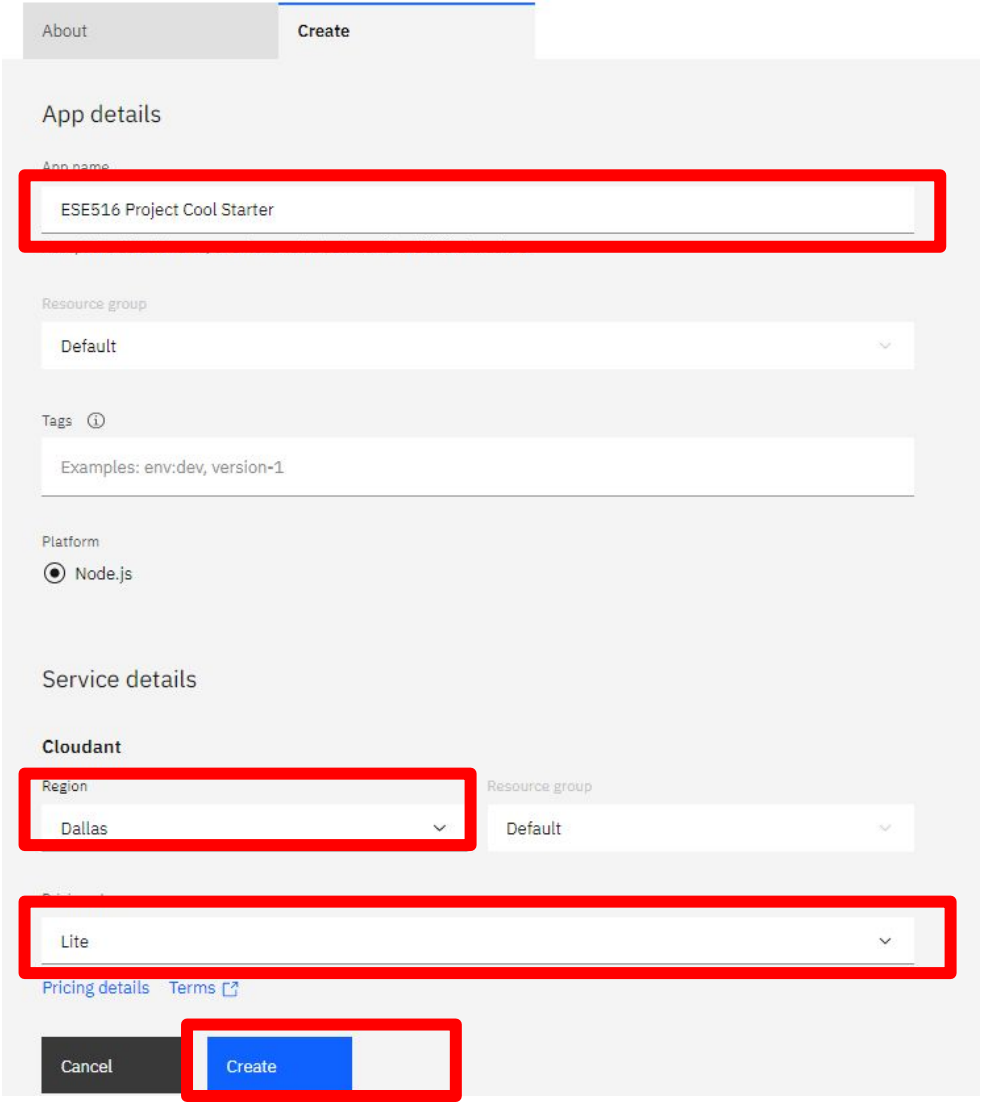


Get Started

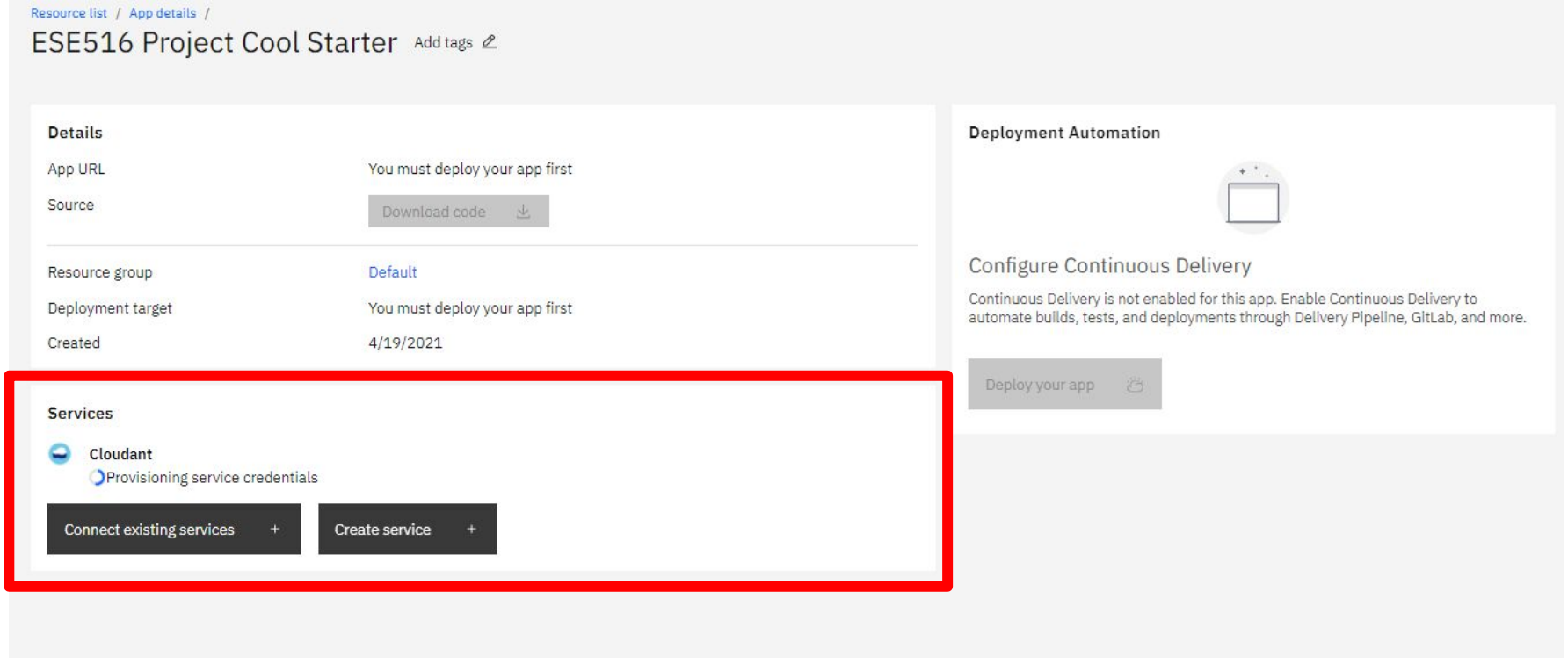


Hit "Get Started"

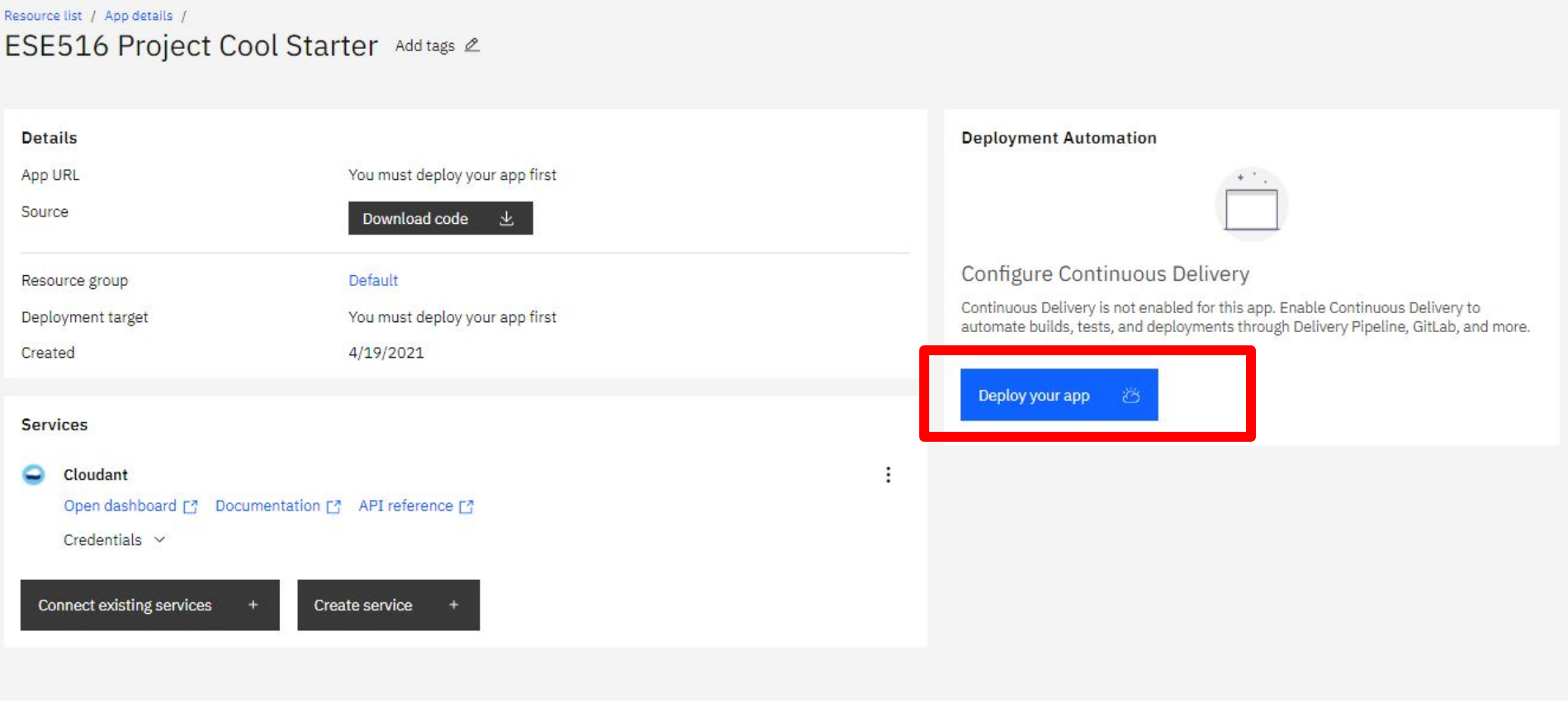
Set a name – any name!



Wait a minute for system to load!

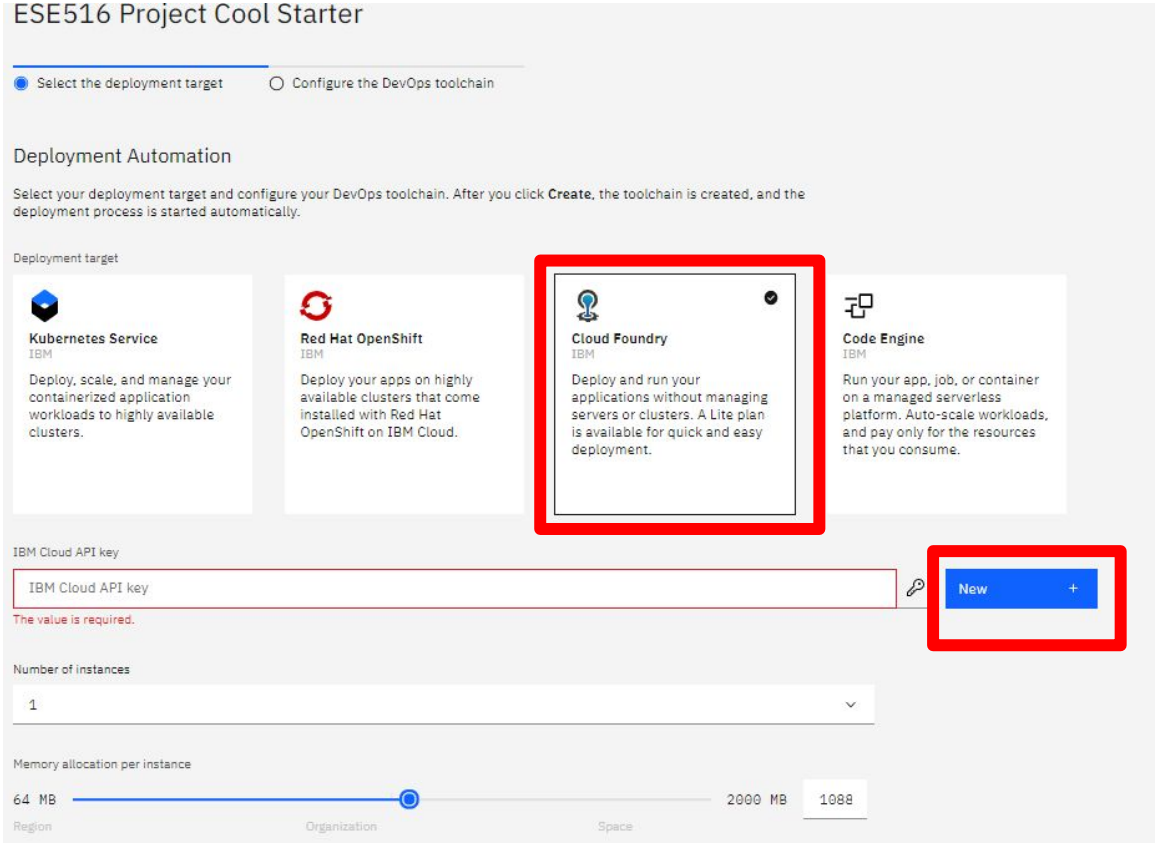


Click **Deploy your app**



Select the following.

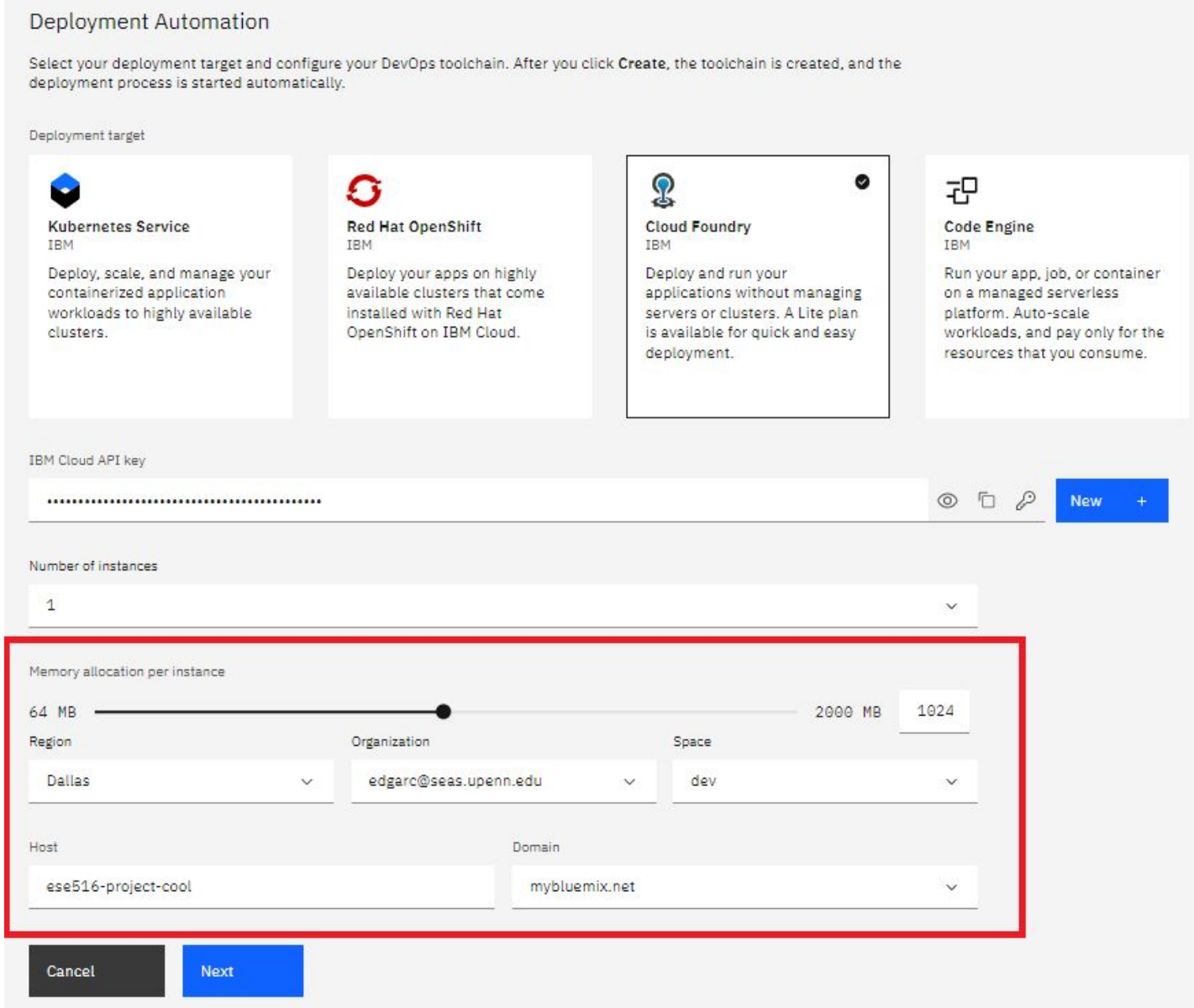
Press “New” to generate a key. You can copy it somewhere if you like.



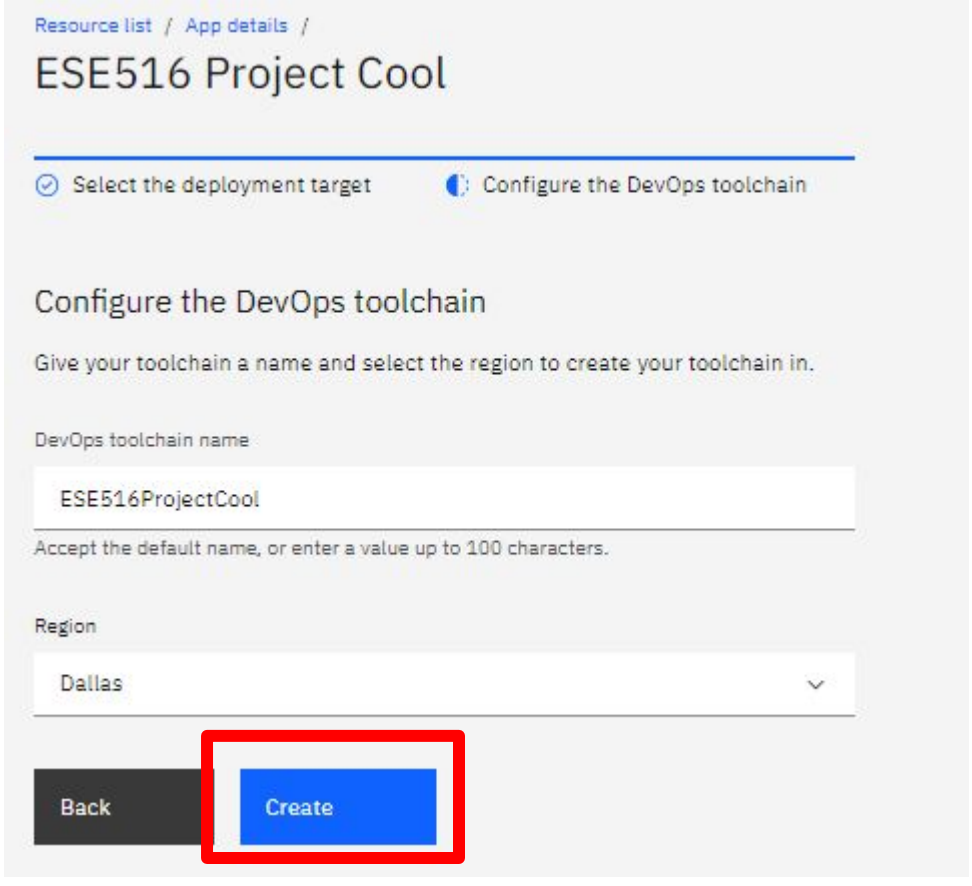
Give it 1024 MB of memory and hit next.

Do not provide more than 1 (one) GB of memory

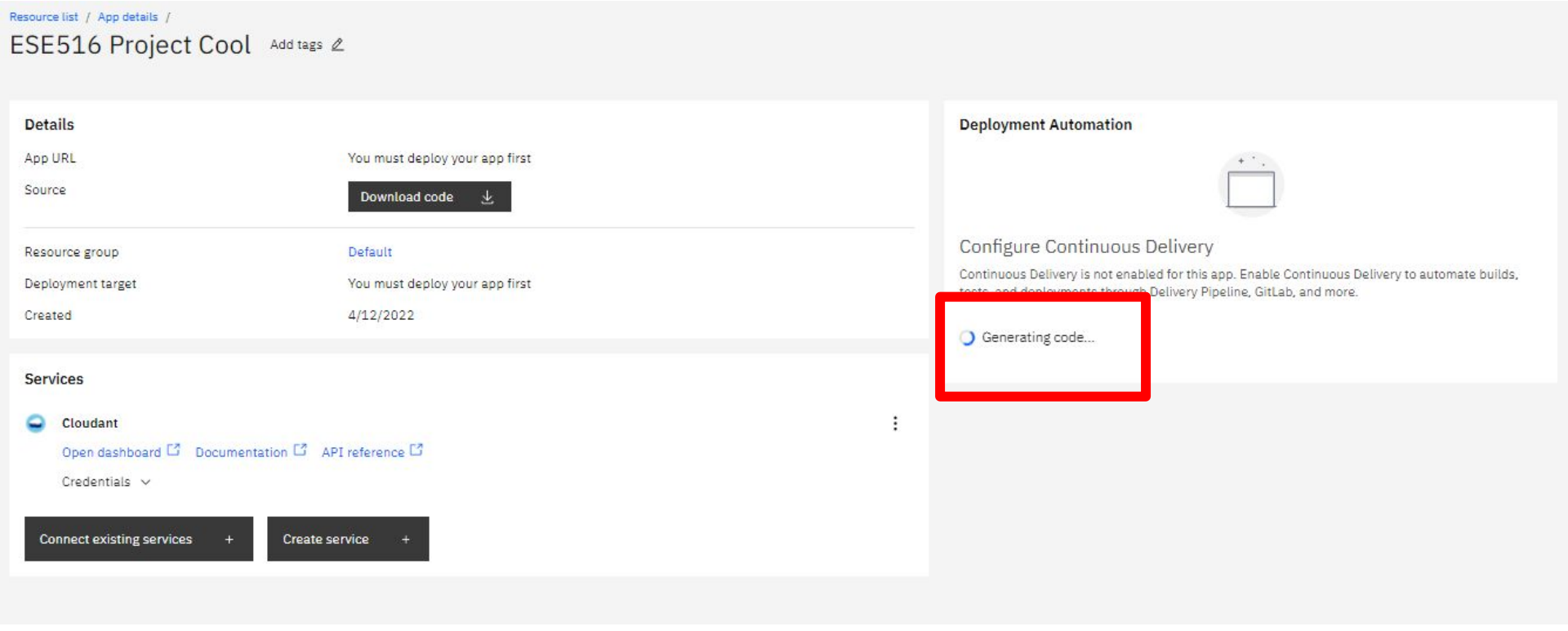
Hit Next



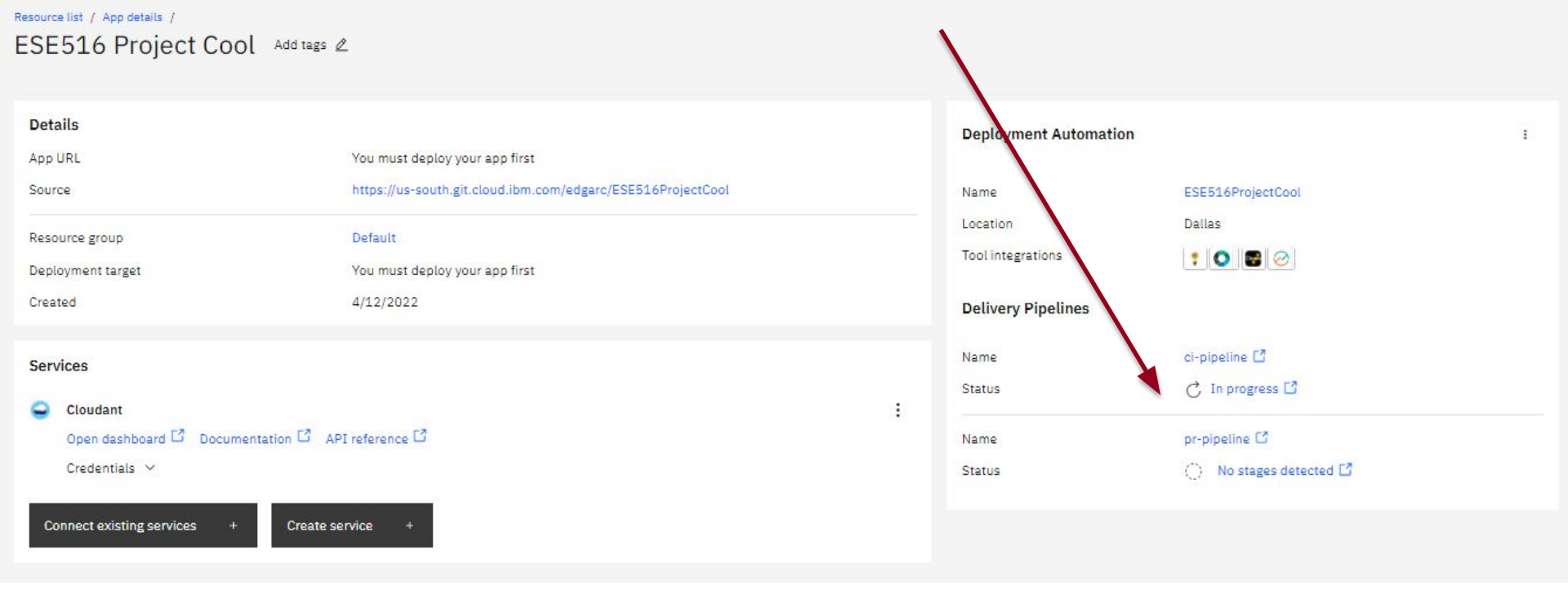
Penn Engineering


Resource list / App details /
ESE516 Project Cool
Select the deployment target
Configure the DevOps toolchain
Configure the DevOps toolchain
Give your toolchain a name and select the region to create your toolchain in.
DevOps toolchain name
ESE516ProjectCool
Accept the default name, or enter a value up to 100 characters.
Region
Dallas
Back
Create

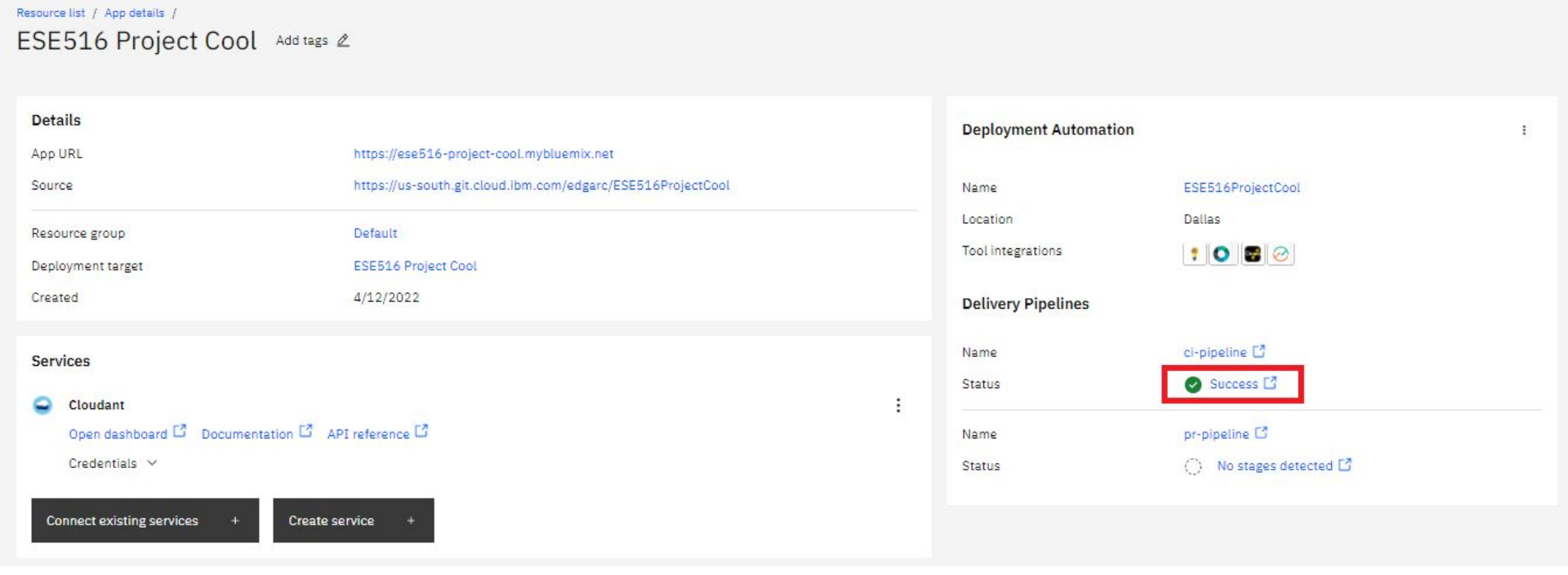
Wait a minute

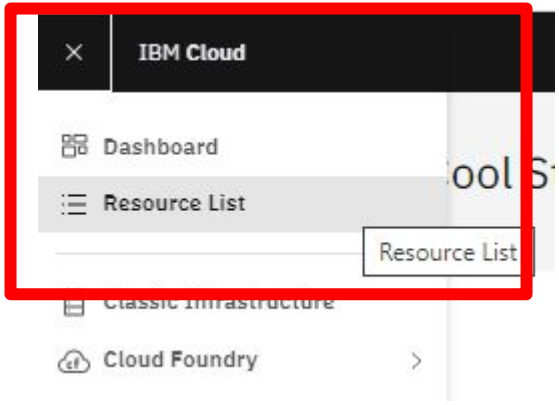


App will be deploying! Take a coffee 9-15 minute break – this will take a while, Mine took 12.5 minutes

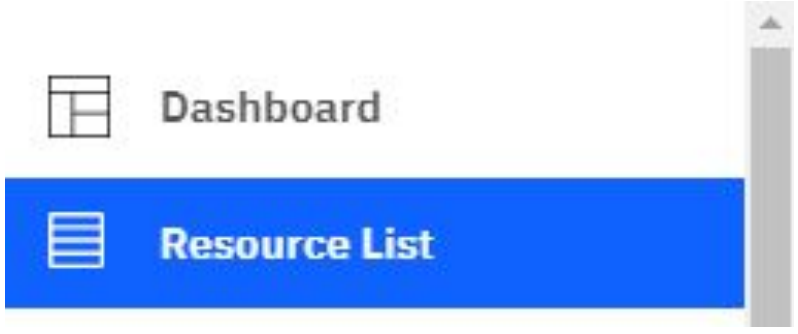


Done!


Resource list / App details /
ESE516 Project Cool
Add tags
Details
App URL
https://ese516-project-cool.mybluemix.net
Source
https://us-south.git.cloud.ibm.com/edgarc/ESE516ProjectCool
Resource group
Default
Deployment target
ESE516 Project Cool
Created
4/12/2022
Services
Cloudant
Open dashboard
Documentation
API reference
Credentials
Connect existing services
Create service
Deployment Automation
Name
ESE516ProjectCool
Location
Dallas
Tool integrations
Delivery Pipelines
Name
ci-pipeline
Status
Success
Name
pr-pipeline
Status
No stages detected

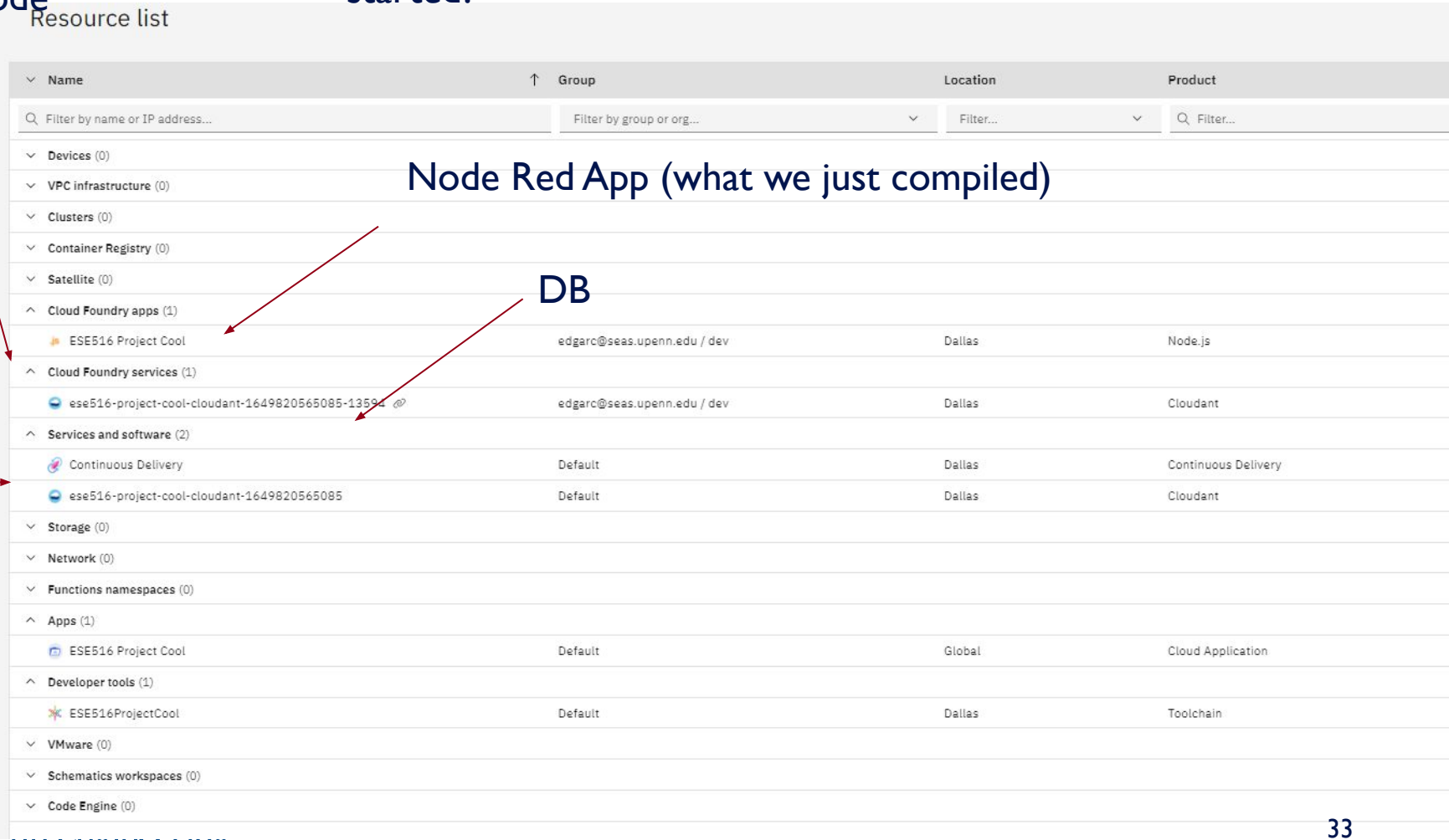


You can now access the resources made on the top left bar – under "Resource List"



Node JS (For Node Red)

Here you can see the services started!

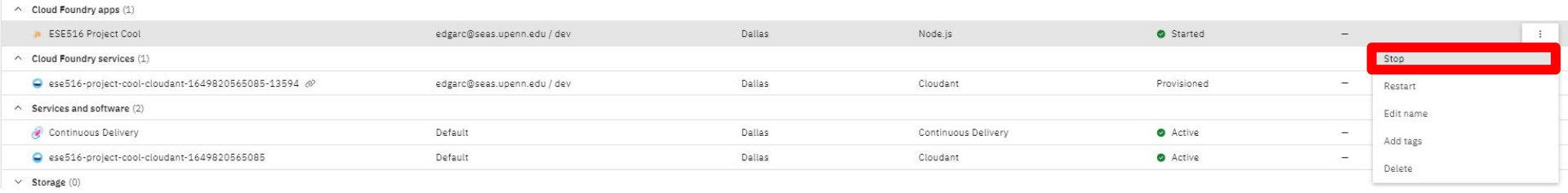


Node Red App (what we just compiled)

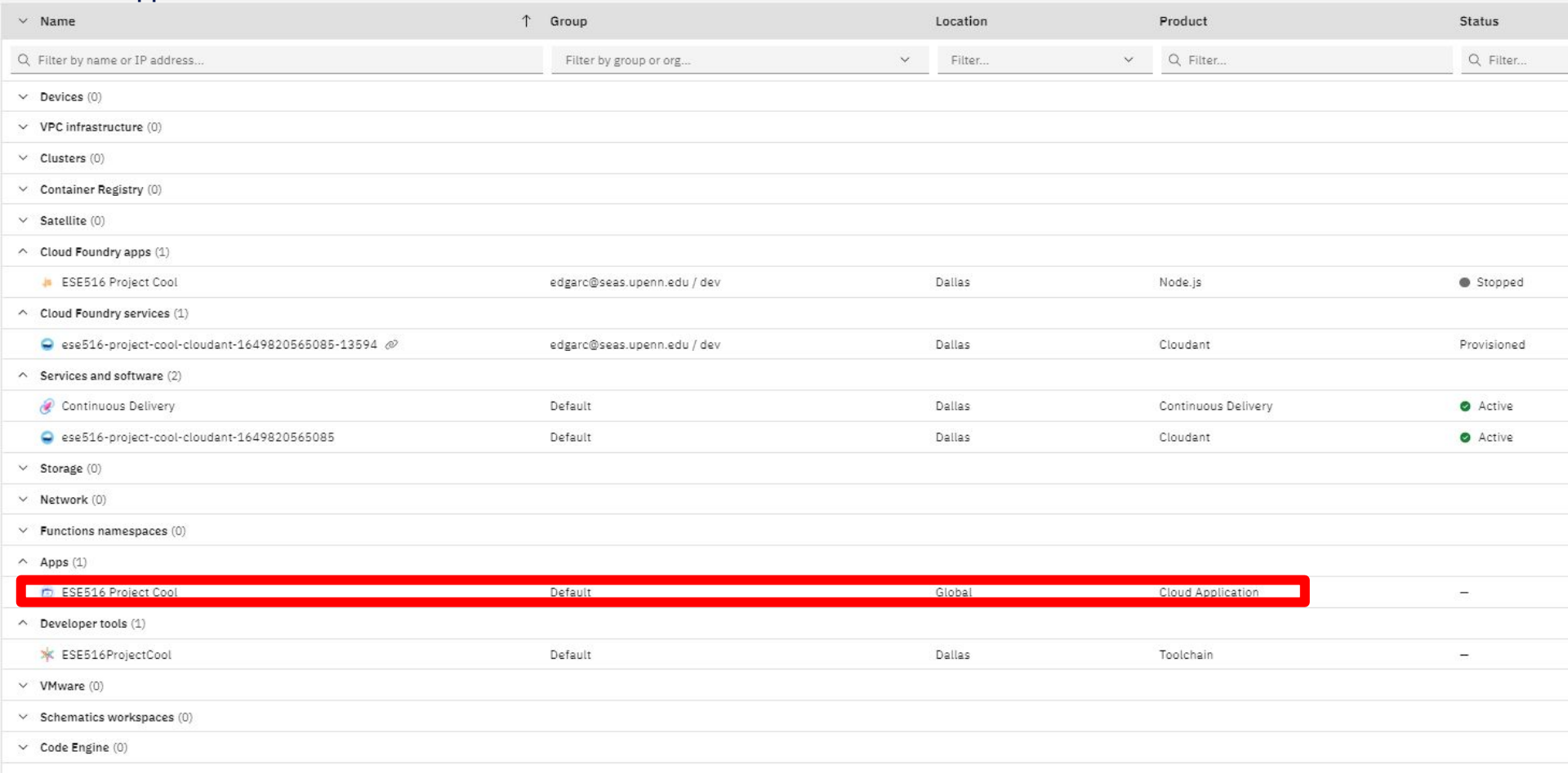
DB

Continuous Delivery

Before our next steps, we need to stop the current app. On the Could Foundry Apps, click on "Stop" to stop that process. You will encounter errors if you do not do this step.

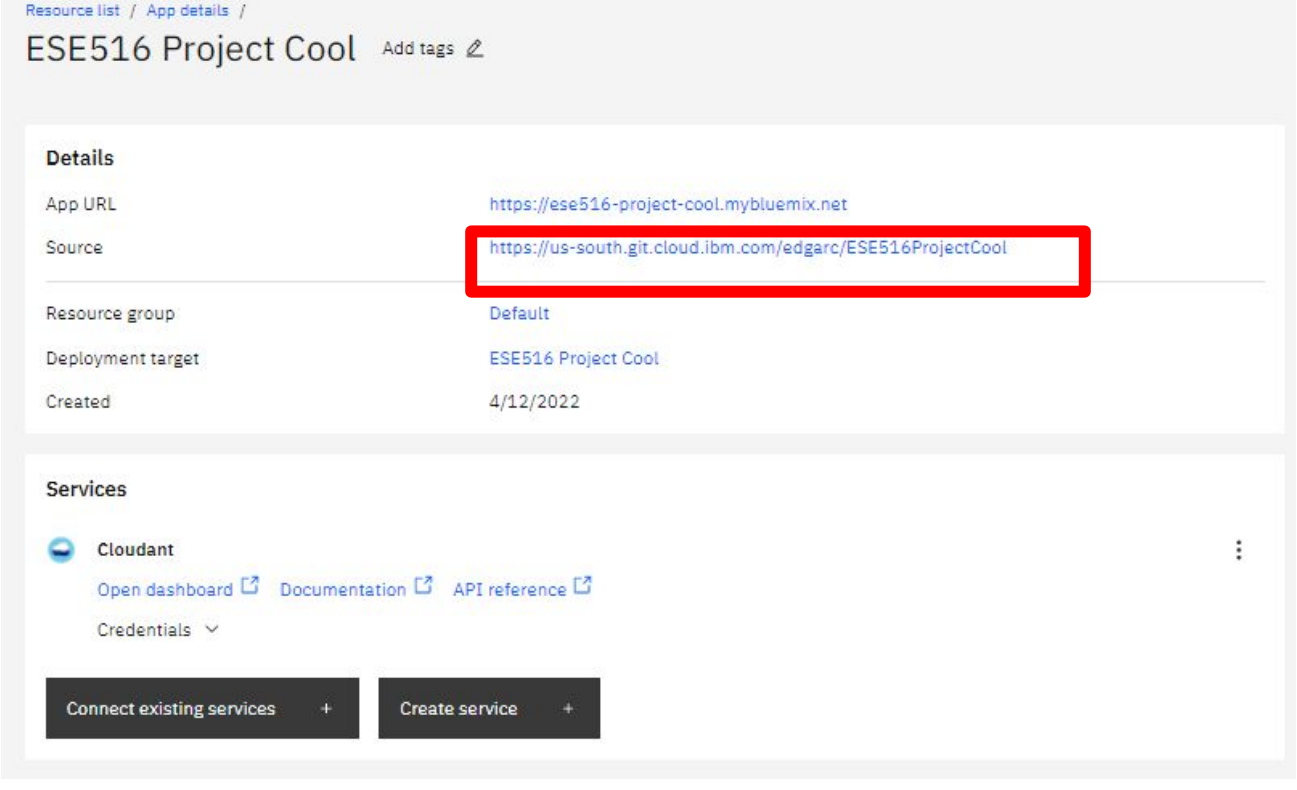


Before we begin, we have to install Node-Red modules for the system! Click on the Cloud Application under APPS



Name	Group	Location	Product	Status
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...
Devices (0)				
VPC infrastructure (0)				
Clusters (0)				
Container Registry (0)				
Satellite (0)				
Cloud Foundry apps (1)				
ESE516 Project Cool	edgarc@seas.upenn.edu / dev	Dallas	Node.js	Stopped
Cloud Foundry services (1)				
ese516-project-cool-cloudant-1649820565085-13594	edgarc@seas.upenn.edu / dev	Dallas	Cloudant	Provisioned
Services and software (2)				
Continuous Delivery	Default	Dallas	Continuous Delivery	Active
ese516-project-cool-cloudant-1649820565085	Default	Dallas	Cloudant	Active
Storage (0)				
Network (0)				
Functions namespaces (0)				
Apps (1)				
ESE516 Project Cool	Default	Global	Cloud Application	–
Developer tools (1)				
ESE516ProjectCool	Default	Dallas	Toolchain	–
VMware (0)				
Schematics workspaces (0)				
Code Engine (0)				

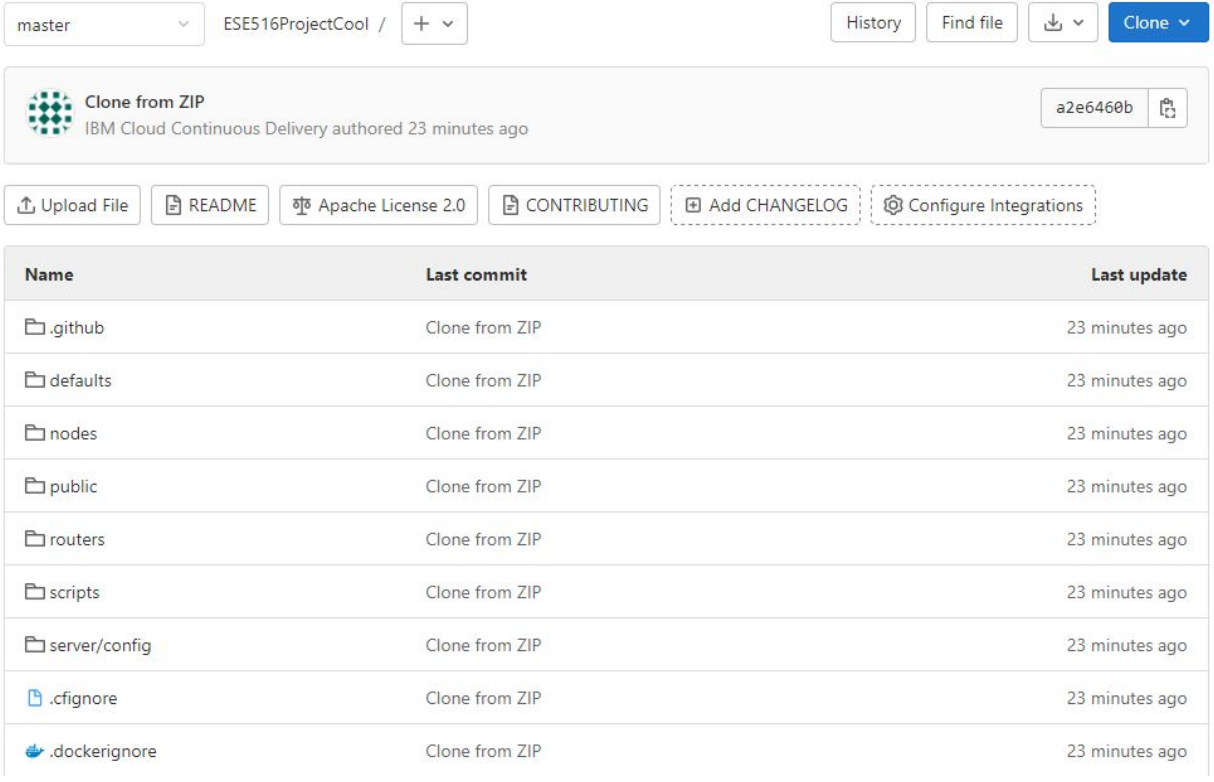
Here we will be able to access the Github project for our deployment. Click the link on the right as shown.



This will take you to a Github page with code for the Node-Red deployment.

We will modify one file:

- Package.json: To install some useful NODE-RED nodes!



Open package.json

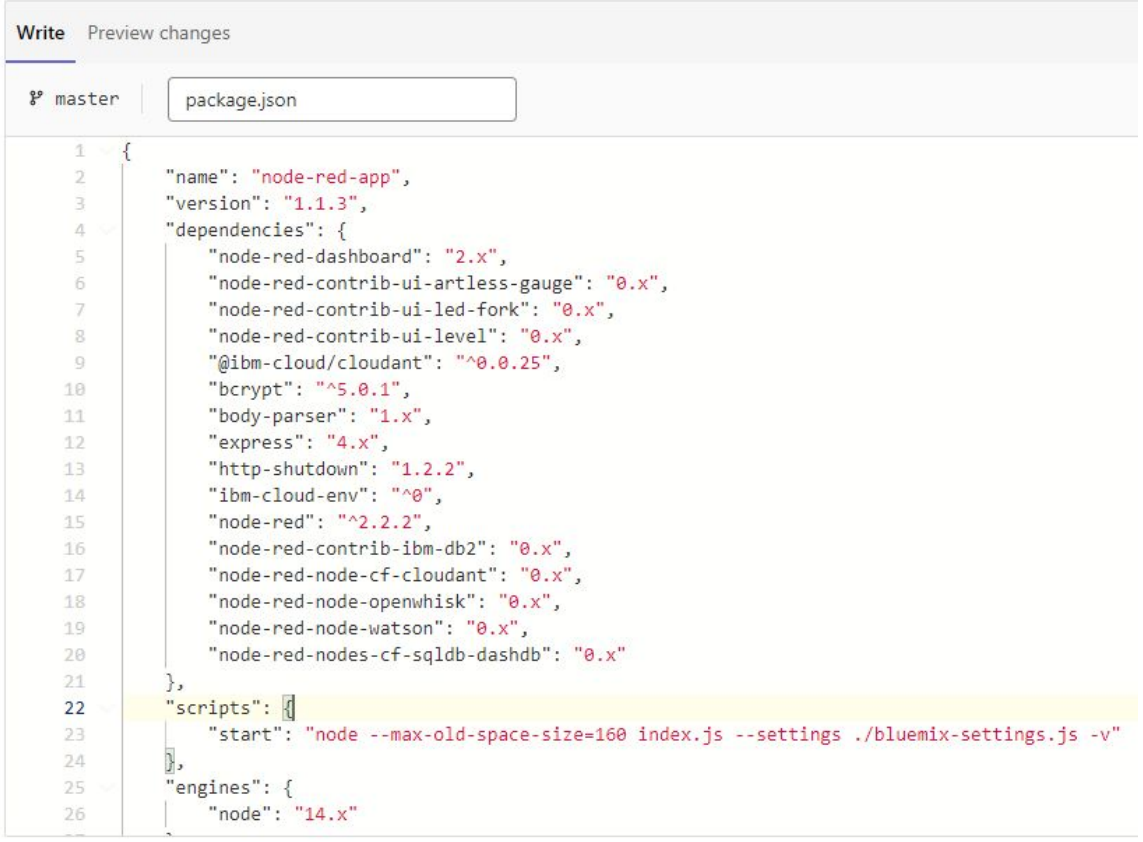
Hit “Edit”

Add the following at the very top of the dependencies:

"node-red-dashboard": "2.x",

"node-red-contrib-ui-artless-gauge" : "0.x",

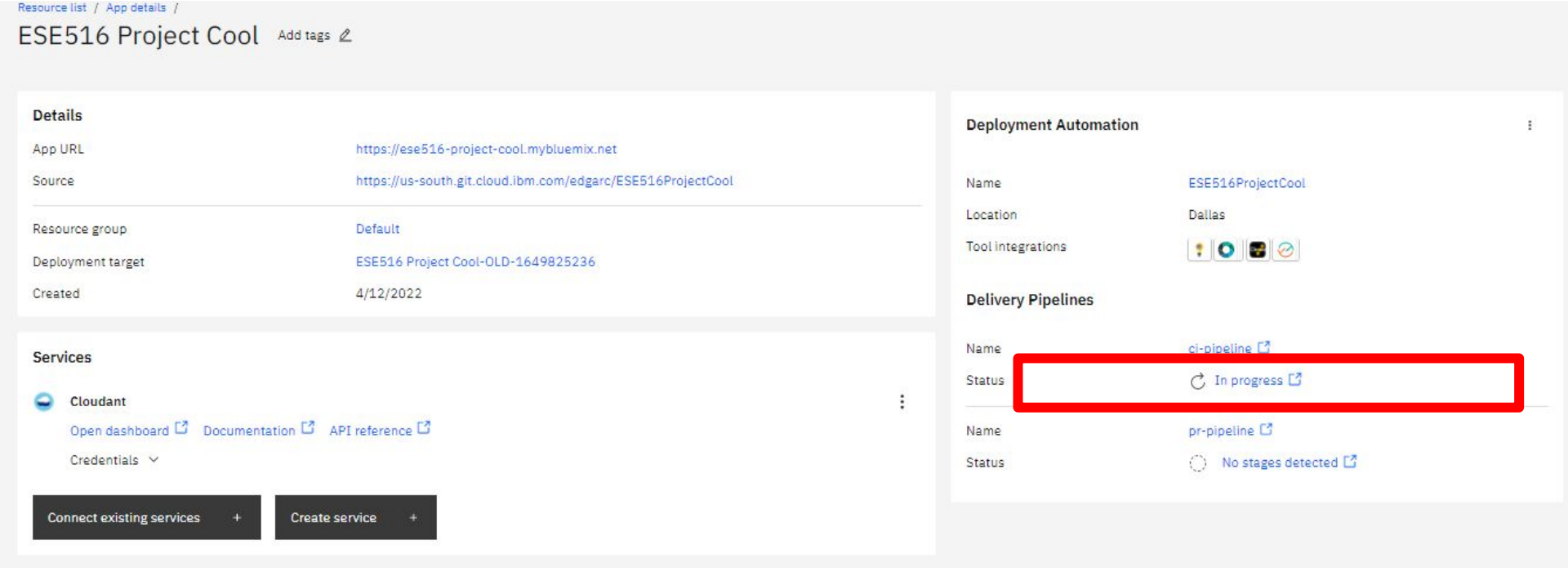
"node-red-contrib-ui-led-fork": "0.x",
"node-red-contrib-ui-level": "0.x",



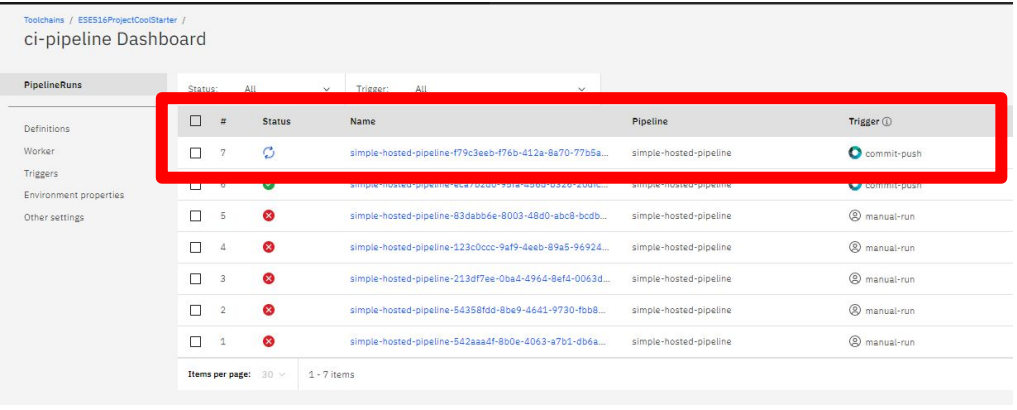
Penn Engineering

Hit “Commit Changes”

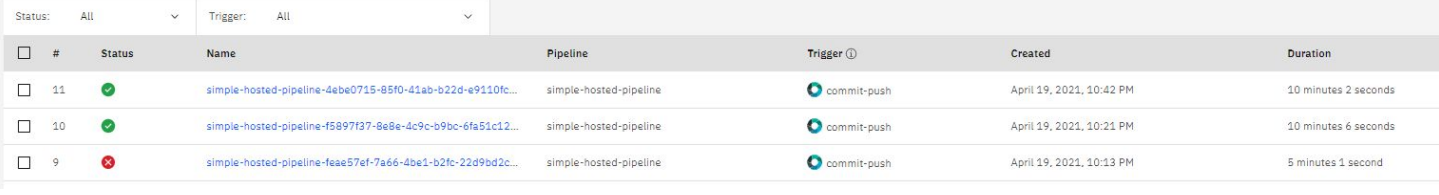
When you hit "Commit Changes", the system will compile the project again. We need to wait until it compiles again. Click to see progress!



When you changed the code, it triggered an automatic rebuild. The system will tell you the state of the new build and deploy on the cloud. Wait until everything passed! If you had errors… Time for Piazza! It will take a while! ~ 5-10 min.

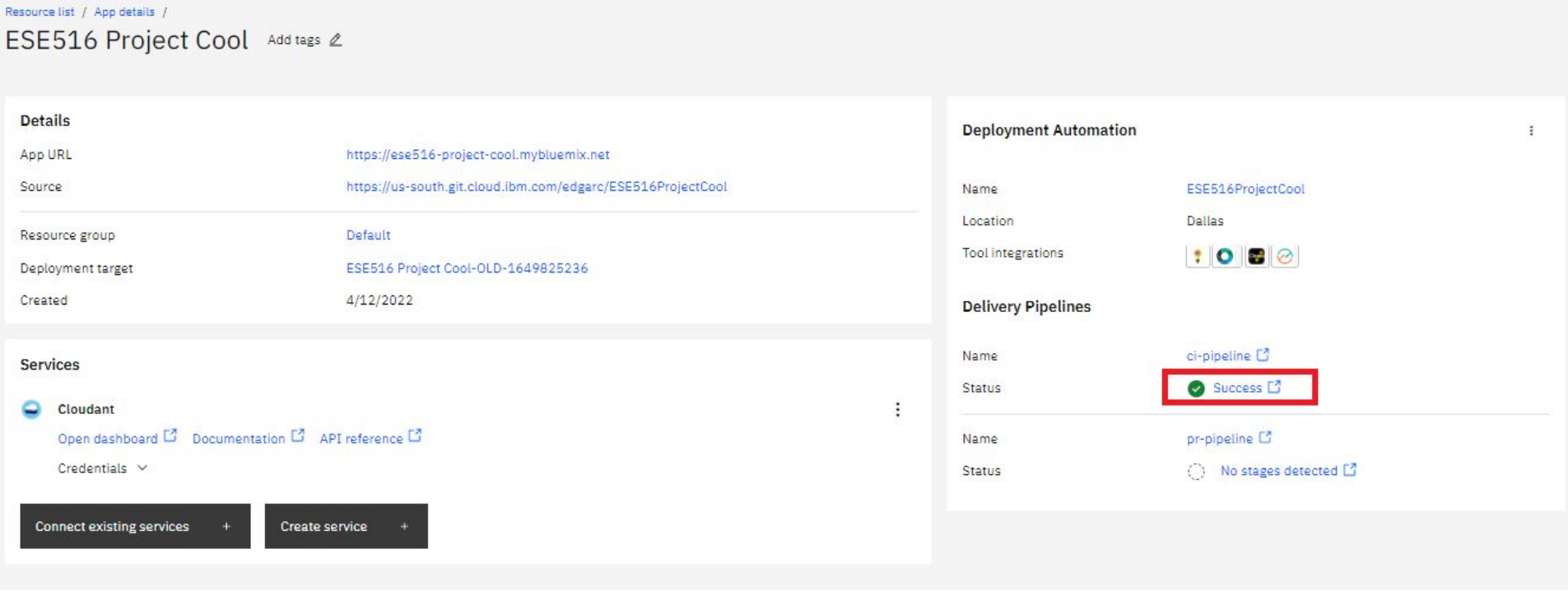


Yay! It passed!



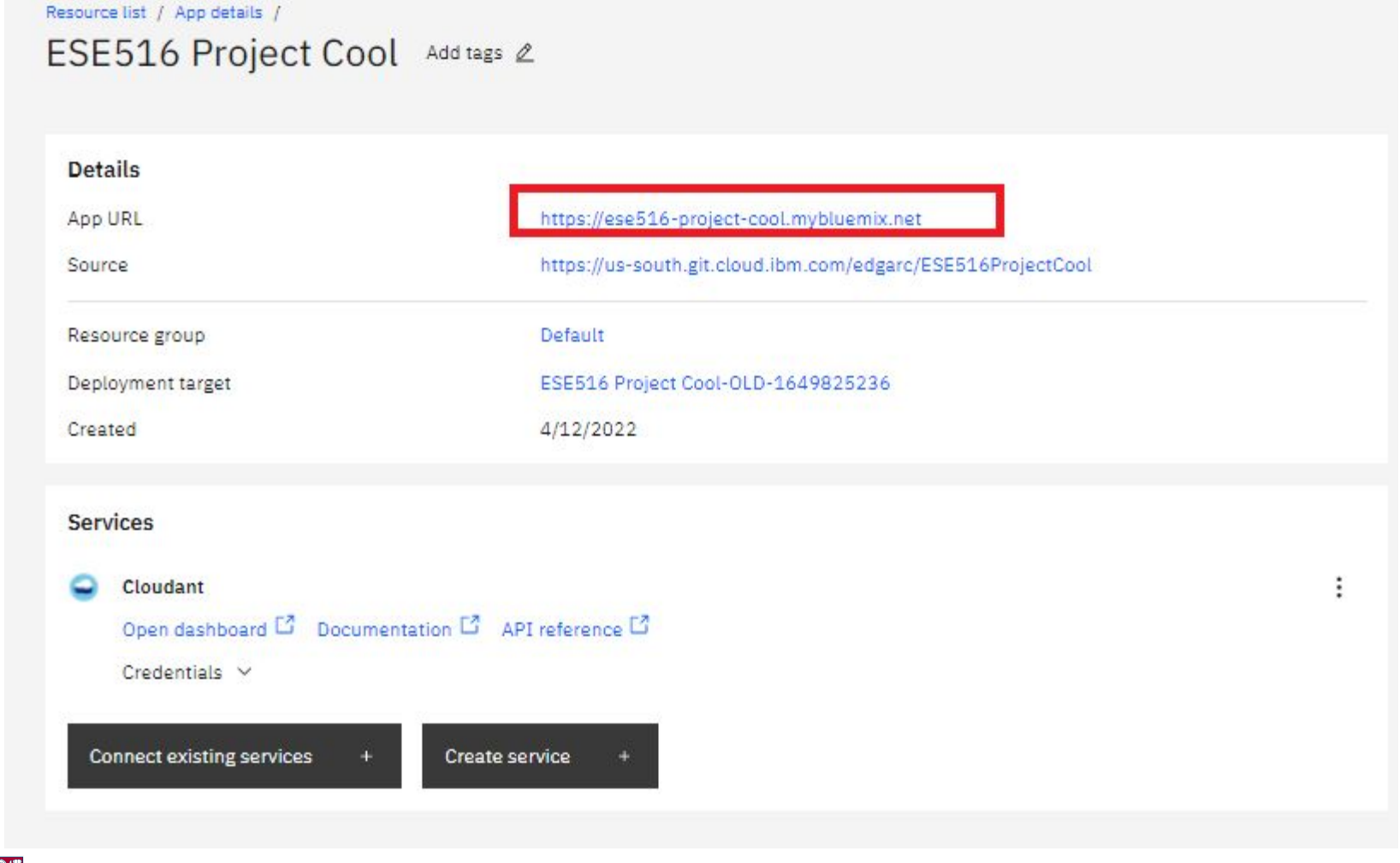
Penn Engineering

Success!



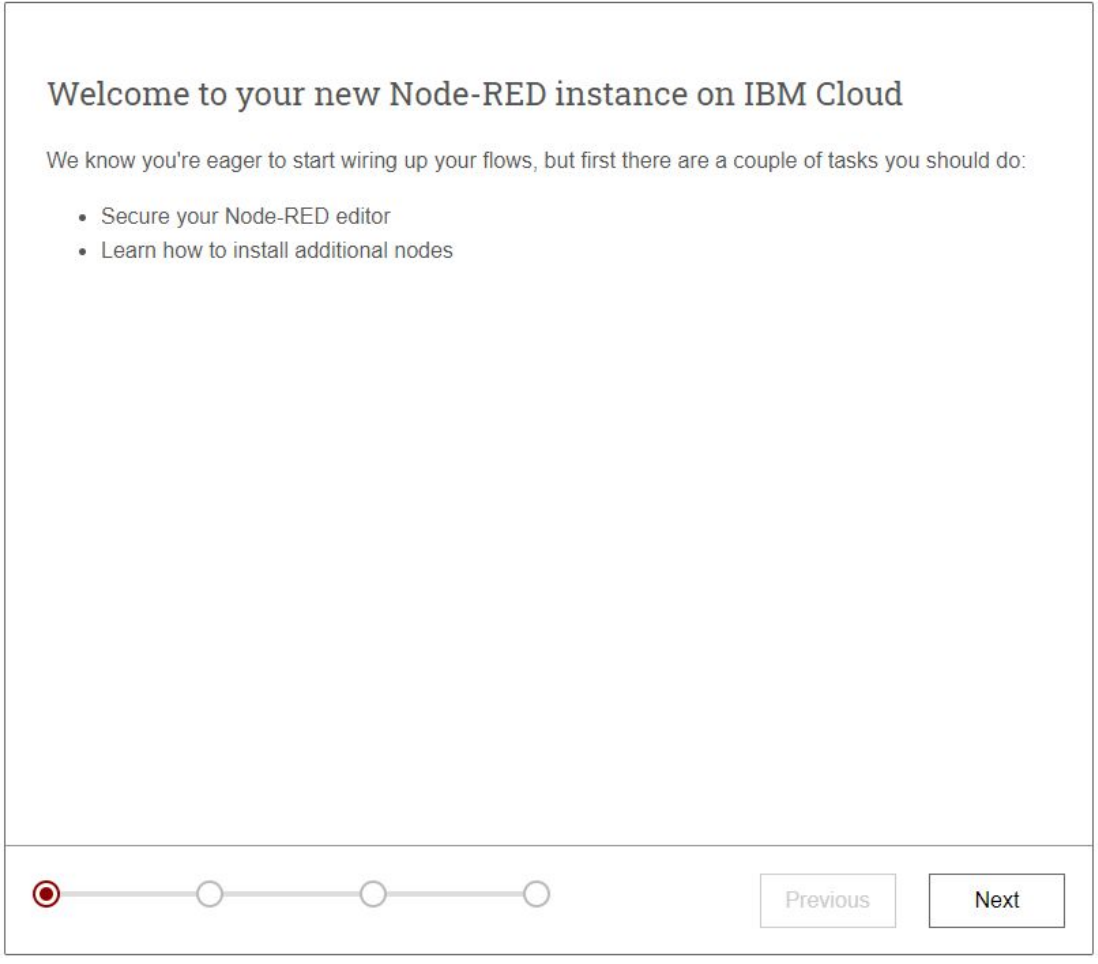
Penn Engineering

Go back and click this link. It will take you to the Node Red instance



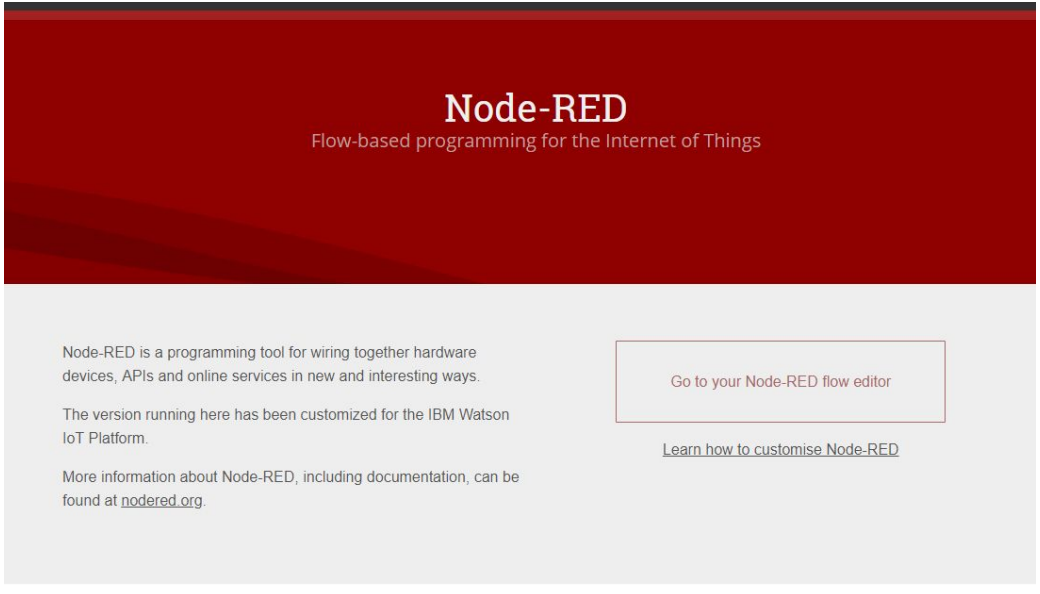
Go thru the next steps

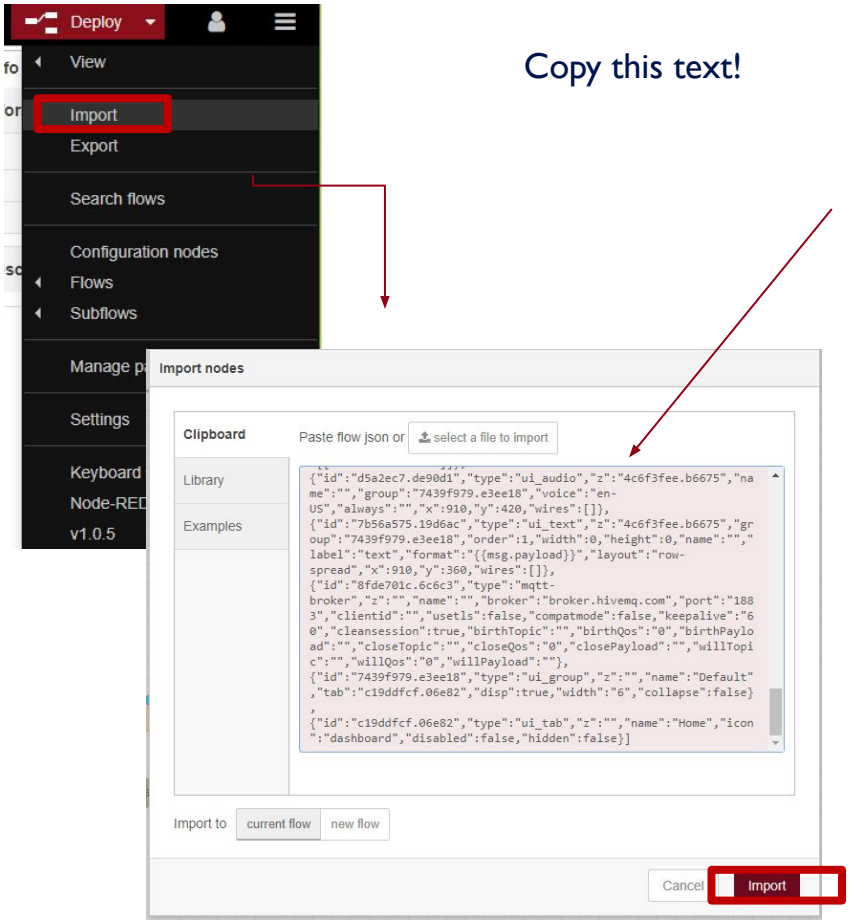
Choose a password you can share with your class partner, and is easy to remember!



Node-Red

- Your Node-Red is now ready! Click on the "Go To your Node-RED flow editor" to open the editor. Otherwise, click on "Learn how to customize Node RED" for a quick tutorial to start up.





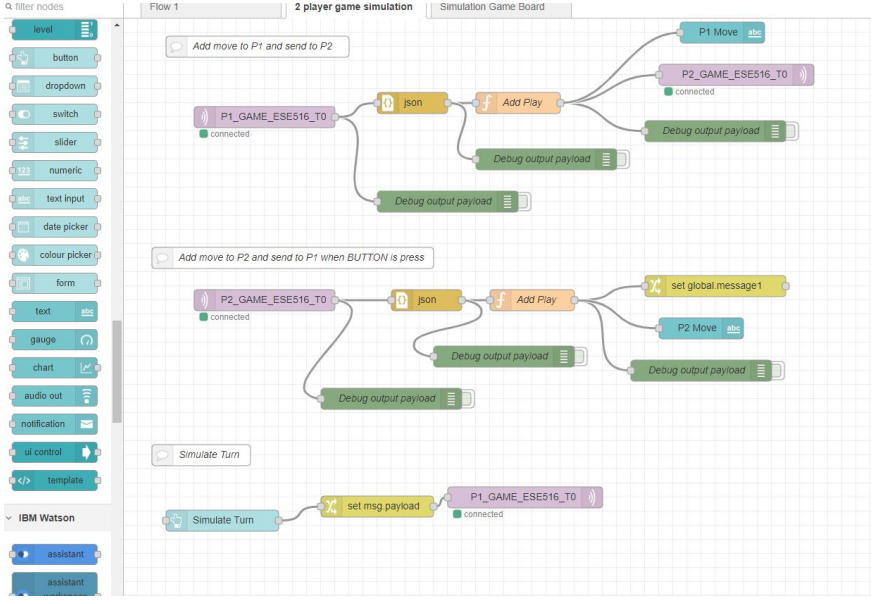
Copy this text!

Penn Engineering

[{"id":"4c6f3fee.b6675","type":"tab","label":"Flow 1","disabled":false,"info":""},{"id":"5ac91ff3.86fef","type":"tab","label":"2 player game simulation","disabled":false,"info":""},{"id":"116f1ff.480b0e","type":"tab","label":"Simulation Game Board","disabled":false,"info":""},{"id":"8fde701c.6c6c3","type":"mqtt-broker","z":"","name":"","broker":"broker.hivemq.com","port":"1883","clientid":"","usetls":false,"compatmode":false,"keepalive":"60","cleansession":true,"birthTopic":"","birthQos":"0","birthPayload":"","closeTopic":"","closeQos":"0","closePayload":"","willTopic":"","willQos":"0","willPayload":""},{"id":"c19ddfcf.06e82","type":"ui_tab","z":"","name":"Home","icon":"dashboard","disabled":false,"hidden":false},{"id":"7439f979.e3ee18","type":"ui_group","z":"","name":"Default","tab":"c19ddfcf.06e82","disp":true,"width":"6","collapse":false},{"id":"3dd7f020.68187","type":"ui_base","theme":{"name":"theme-dark","lightTheme":{"default":"#0094CE","baseColor":"#0094CE","baseFont":"-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif","edited":true,"reset":false},"darkTheme":{"default":"#097479","baseColor":"#097479","baseFont":"-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif","edited":true,"reset":false},"customTheme":{"name":"Untitled Theme 1","default":"#4B7930","baseColor":"#4B7930","baseFont":"-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"},"themeState":{"base-color":{"default":"#097479","value":"#097479","edited":false},"page-titlebar-backgroundColor":{"value":"#097479","edited":false},"page-backgroundColor":{"value":"#111111","edited":false},"page-sidebar-backgroundColor":{"value":"#000000","edited":false},"group-textColor":{"value":"#0eb8c0","edited":false},"group-borderColor":{"value":"#555555","edited":false},"group-backgroundColor":{"value":"#333333","edited":false},"widget-textColor":{"value":"#eeeeee","edited":false},"widget-backgroundColor":{"value":"#097479","edited":false},"widget-borderColor":{"value":"#333333","edited":false},"base-font":{"value":"-apple-system,BlinkMacSystemFont,Segoe UI,Roboto,Oxygen-Sans,Ubuntu,Cantarell,Helvetica Neue,sans-serif"}},"angularTheme":{"primary":"indigo","accents":"blue","warn":"red","background":"grey"}},"site":{"name":"Node-RED Dashboard","hideToolbar":"false","allowSwipe":"false","lockMenu":"false","allowTempTheme":"true","dateFormat":"DD/MM/YYYY","sizes":{"sx":48,"sy":48,"gx":4,"gy":4,"cx":6,"cy":6,"px":0,"py":0}}},{"id":"622b3681.b637e8","type":"ui_group","z":"","name":"2 Player Game Simulation","tab":"3c01459a.82910a","order":1,"disp":true,"width":"6","collapse":false},{"id":"3c01459a.82910a","type":"ui_tab","z":"","name":"2 Player Simulation","icon":"dashboard","order":2,"disabled":false,"hidden":false},{"id":"43e5bb99.cfe0e4","type":"ui_tab","z":"","name":"One Player Game","icon":"dashboard","order":3,"disabled":false,"hidden":false},{"id":"eff19175.c358e","type":"ui_group","z":"","name":"Color","tab":"43e5bb99.cfe0e4","order":1,"disp":true,"width":"6","collapse":false},{"id":"d7d463f4.cae2c","type":"ui_group","z":"","name":"P1 LED Board","tab":"3c01459a.82910a","order":2,"disp":true,"width":"4","collapse":false},{"id":"4c314b97.169624","type":"ui_spacer","name":"spacer","group":"d7d463f4.cae2c","order":2,"width":1,"height":1},{"id":"1980e43f.b4d11c","type":"function","z":"4c6f3fee.b6675","name":"temp","func":"return
{payload:msg.payload.d.temp};","outputs":1,"noerr":0,"x":390,"y":480,"wires":[["48c3b8d0.9c3558"]]},{"id":"48c3b8d0.9c3558","type":"switch","z":"4c6f3fee.b6675","name":"temp
thresh","property":"payload","propertyType":"msg","rules":[{"t":"lte","v":"40","vt":"str"},{"t":"gt","v":"40","vt":"str"}],"checkall":"true","repair":false,"outputs":2,"x":550,"y":480,"wires":[["5af543bd.70630c"],["c3eb20e6.6fb63"]]},{"id":"6e974680.39cd28","type":"debug","z":"4c6f3fee.b6675","name":"cpu
status","active":false,"complete":"false","x":890,"y":480,"wires":[]},{"id":"ea07eb98.bc0448","type":"debug","z":"4c6f3fee.b6675","name":"device data","active":false,"console":"false","complete":"true","x":410,"y":560,"wires":[]},{"id":"5af543bd.70630c","type":"template","z":"4c6f3fee.b6675","name":"safe","field":"payload","fieldType":"msg","syntax":"mustache","template":"Temperature ({{payload}}) within safe limits","x":710,"y":460,"wires":[["6e974680.39cd28","7b56a575.19d6ac"]]},{"id":"c3eb20e6.6fb63","type":"template","z":"4c6f3fee.b6675","name":"danger","template":"Temperature ({{payload}})
critical","x":720,"y":500,"wires":[["6e974680.39cd28"]]},{"id":"3db4e4f7.4a9dfc","type":"inject","z":"4c6f3fee.b6675","name":"Send Data","topic":"","payload":"true","payloadType":"bool","repeat":"","crontab":"","once":false,"x":120,"y":140,"wires":[["20164732.524cf8"]]},{"id":"20164732.524cf8","type":"function","z":"4c6f3fee.b6675","name":"Device payload","func":"// Thermostat's location:\nvar longitude1 = -98.49;\nvar latitude1 = 29.42;\n\n// Array of pseudo random temperatures\nvar temp1 = [15,17,18.5,20,21.5,23,24,22.2,19,18];\n\n// Array of pseudo random relative humidities\nvar humidity1 = [50,55,61,68,65,60,53,49,45,47];\n\n// Counter to select from array.\nvar counter1 = context.get('counter1')||0;\ncounter1 = counter1+1;\nif(counter1 > 9) counter1 = 0;\ncontext.set('counter1',counter1);\n\n// Create MQTT message in JSON\nmsg = {\n payload: JSON.stringify(\n {\n d:{\n \"temp\" : temp1[counter1],\n \"humidity\" : humidity1[counter1],\n \"location\" :\n {\n \"longitude\" : longitude1,\n \"latitude\" : latitude1\n },\n }\n }\n)\n};\nreturn
msg;\n","outputs":1,"noerr":0,"x":340,"y":140,"wires":[["5f9f3a85.1b8ff4","d63227c.8819e8"]]},{"id":"5f9f3a85.1b8ff4","type":"debug","z":"4c6f3fee.b6675","name":"Debug output
payload","active":false,"console":"false","complete":"payload","x":580,"y":200,"wires":[]},{"id":"33e5da99.a54c16","type":"comment","z":"4c6f3fee.b6675","name":"Device Simulator","info":"Sends simulated device sensor data to IBM Watson IoT Plaform.\n\nCan be configured to send on click or on an automatic interval.\n\n\n#Prerequisite\nOutput node device type and device ID need to match a device that it registered in a running IBM Watson IoT Platform service.\n\n# Watson IoT Platform docs\n[Connecting

Node-Red

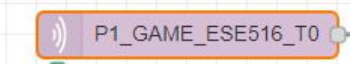
You should now have the ESE516 Simulator on your flow. Take some time to explore what is going on! The next slider will explain some of the most important nodes.

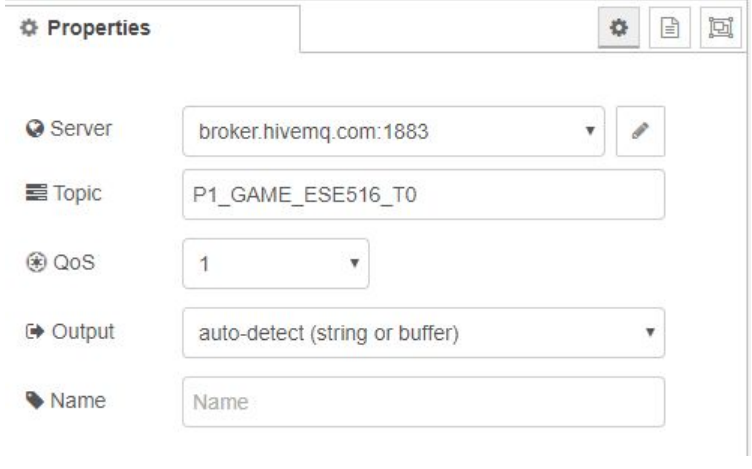


Penn Engineering

Node-Red – MQTT IN

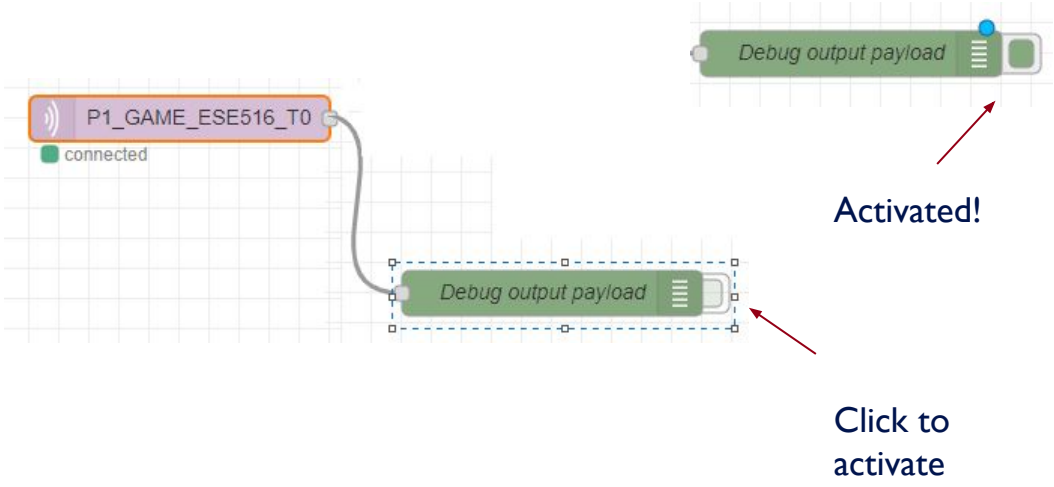
MQTT Node: The MQTT IN NODE listens to a command received from a broker. If you double click it you can edit what it listens to! Notice that it is connected to the MQTT broker we mentioned before, and is listening to the P1_GAME_ESE516_T0. You will need to change this T0 to a unique number so it does not conflict with your colleagues!



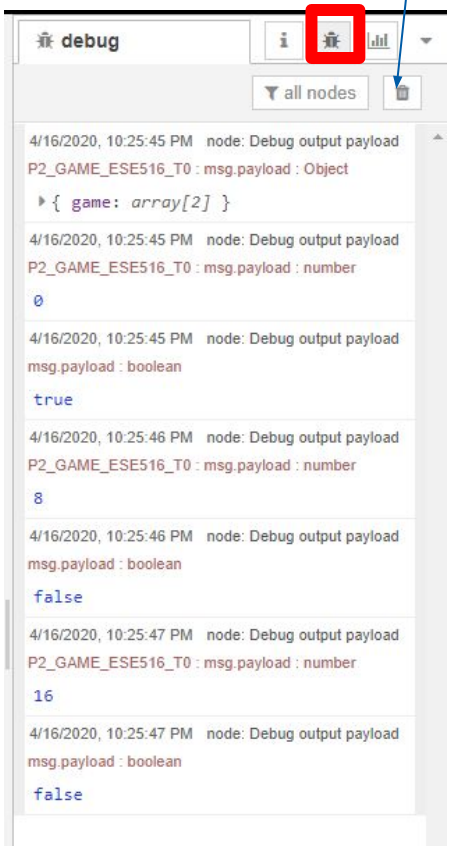


Node-Red – debug node

Debug NODE: Connected to this node is a DEBUG NODE. A debug node will print whatever it is connected to on the debug window. You can get to this window on the right screen of the backend view of Node Red (where you see the nodes). You can click on the green rectangle to activate/deactivate debug output.

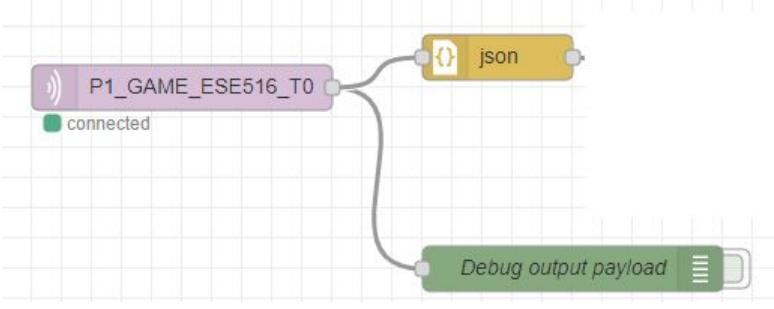


Click to clear output!



Node-Red JSON Block

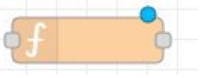
- The JSON block converts a JSON string into a Javascript Object. This will allow us to use Javascript on the data later on!

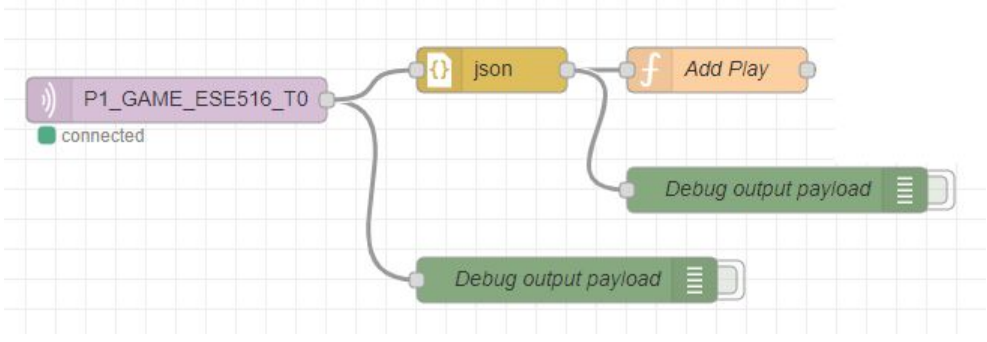


Node-Red Javacript Block (Function)

- The function block can be added to run Javascript code! Double click on "Add Play" to see its code. In our example, this block is used to add a game play (random number between 0 to 15) to the end of the received play!

Function block



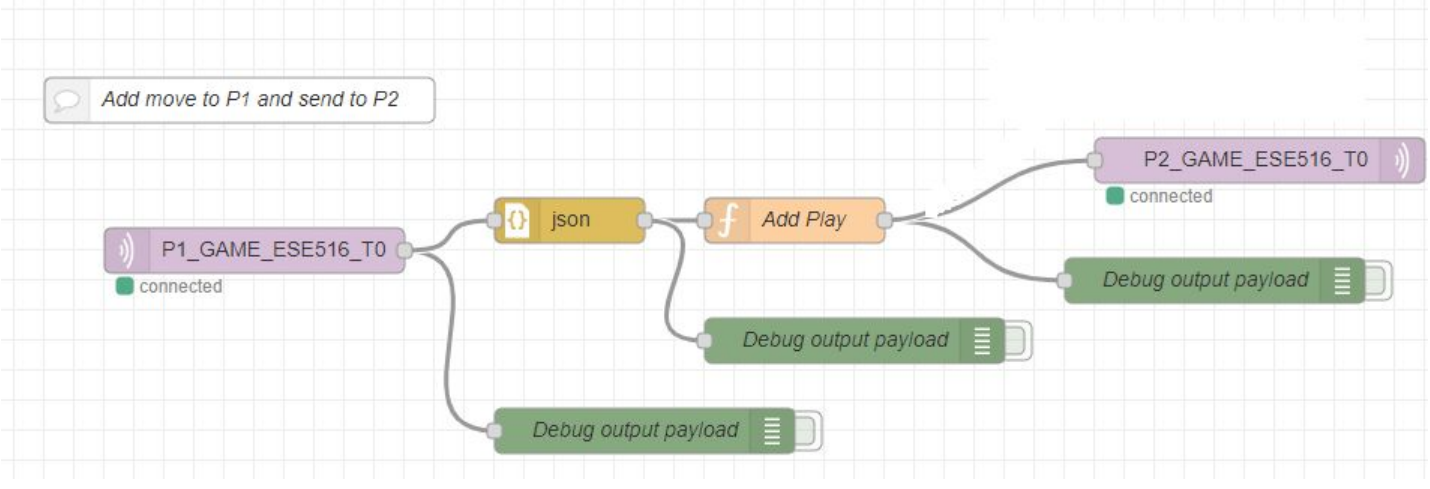


Function

```
1
2  if(msg.payload.game.length <= 20){
3
4   //Concatenate a play
5   msg.payload.game.push(Math.round(Math.random()*15))
6
7  }
8  else{
9   //Pop out first game in list and add a new one at th
10  msg.payload.game.shift(); // Remove an item from the
11  msg.payload.game.push(Math.round(Math.random()*15))
12 }
13
14
15 return msg
```

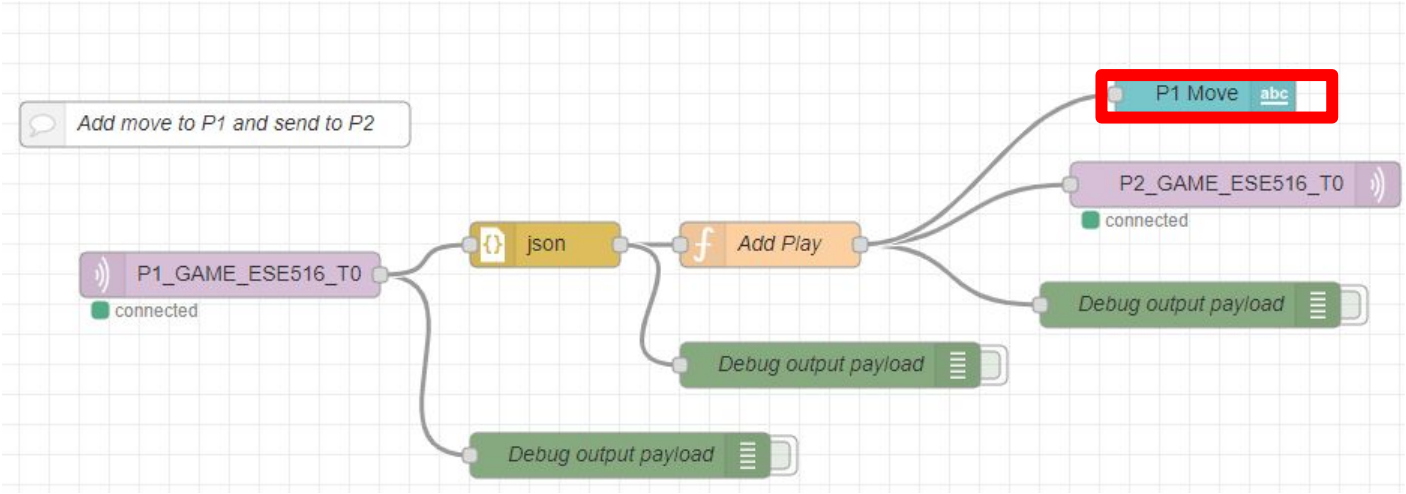
MQTT Out Block

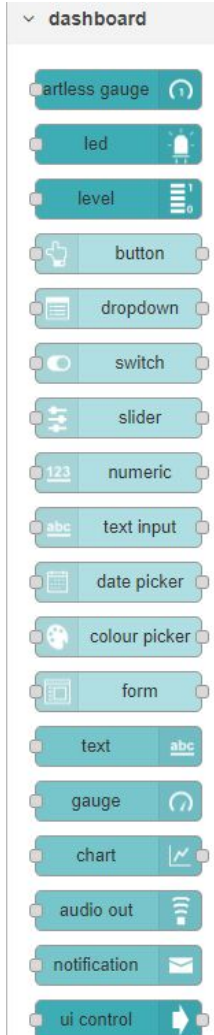
- MQTT Out button: Similar to the MQQT in, the MQTT Out will publish data to an MQTT Topic. In this case, we publist to the player 2 (P2_GAME_ESE516_T0).



Front end (ui) blocks

- There are going to be special nodes (listed under DASHBOARD) that will allow us to develop a front end GUI. In this example, we use a text label that prints the output to P2.

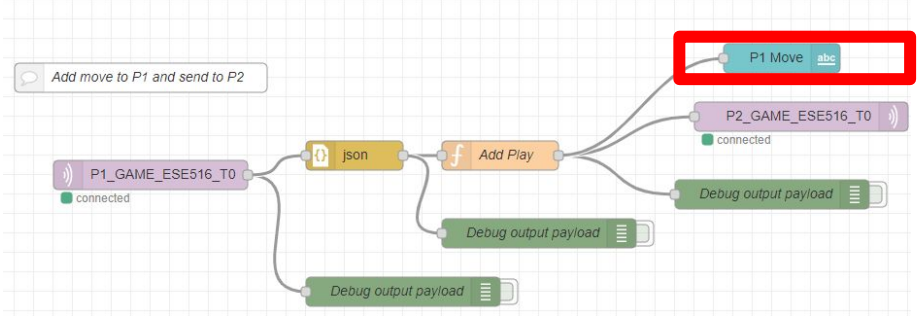


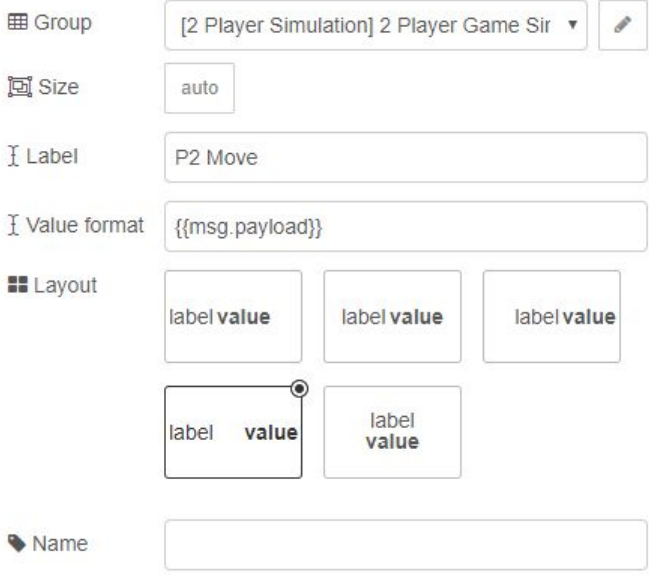


Penn Engineering

Node-Red UI

- When you double click a dashboard component you can modify how it will look! You can also assign to what group it will be added to (where it will be shown on the UI).

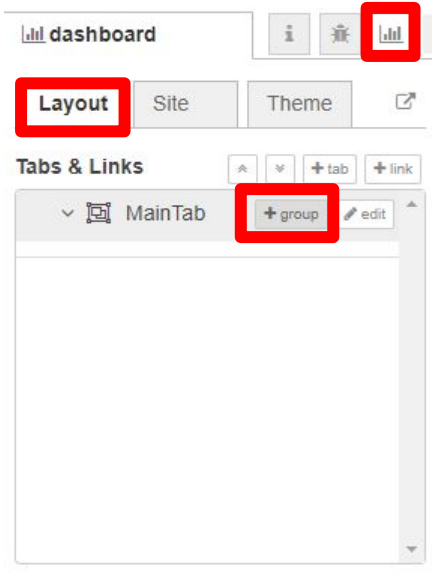




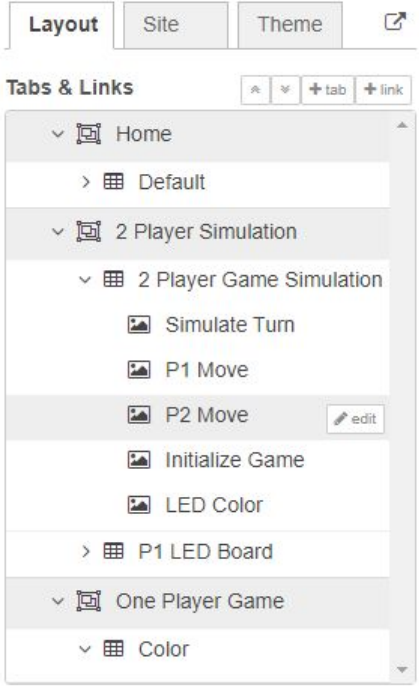
Penn Engineering

Node-Red UI

- To configure the Layout of Dashboard block, you will need to configure the layout. To do so you can click on the topright layout button. On the layout tab you can add tabs and modify the position of existing DASHBOARD blocks into new groups.



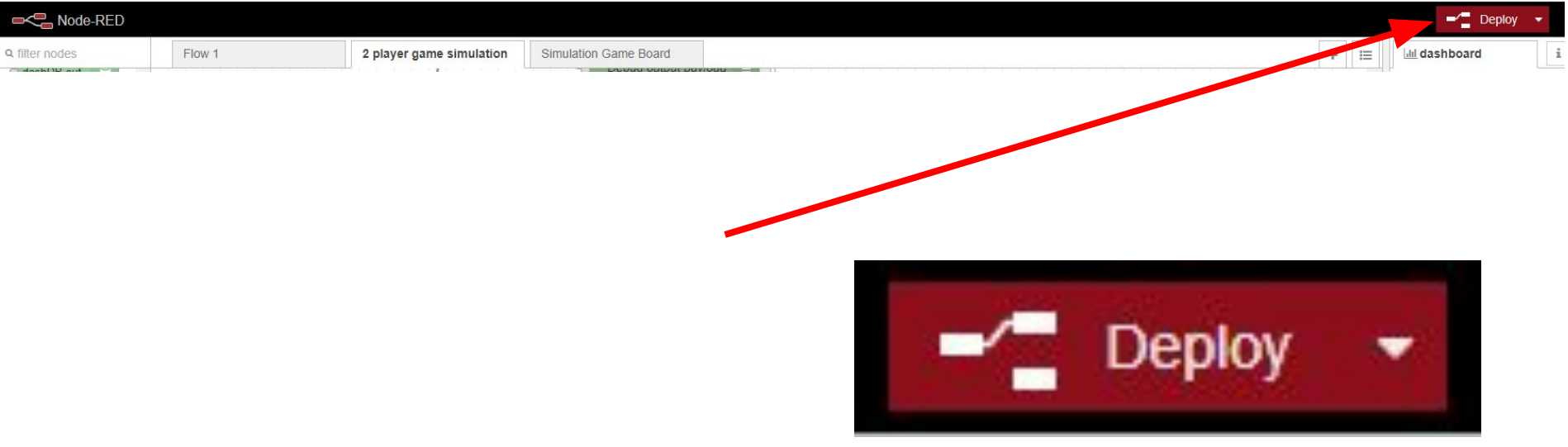
Example of layout of our flow. For example the string we saw, P2 Move, is on the 2 PLAYER GAME SIMULATION Tab.



Penn Engineering

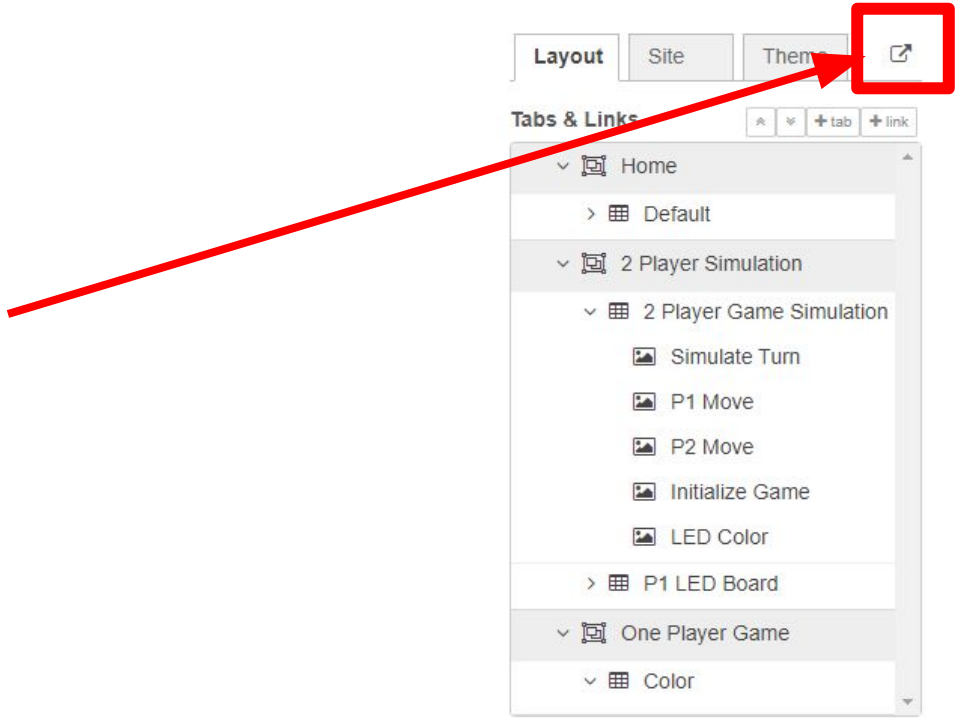
Node-Red- Deploy

- Once you made changes and want to deploy your new code, hit the "Deploy" button on the top right. This will deploy your flow! It will mention any issues that might arise if they are present.



GOING TO THE UI

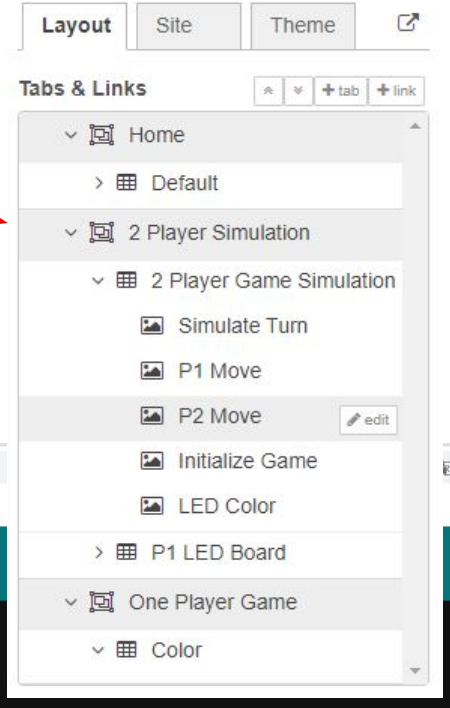
- You can open the UI Website by clicking on the link in the Layout table.

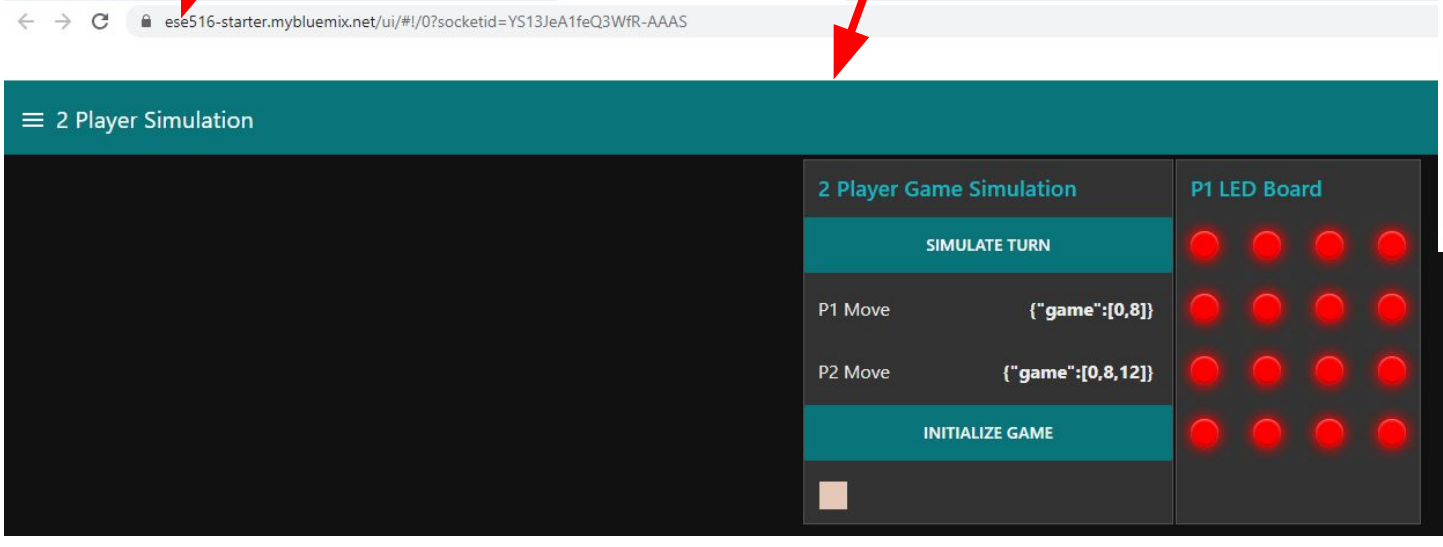


NODE RED DASHBOARD

- Website you can access from anywhere to see your dashboard

UI Organized by how you did it on the UI options





COOL! I want to learn more!

- Node Red Cookbook: https://cookbook.nodered.org/
- Node Red Tutorial: http://noderedguide.com/