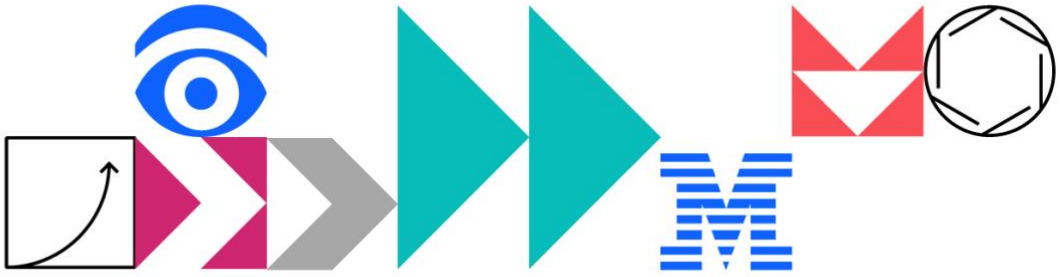
IBM TechXchange



Horizontally scaling IBM MQ and your applications

Session 3277

Lab Exercise Guide v1

Joel Gomez
WW Executive Cloud Integration Architect
joel.gomez@us.ibm.com

Martin Evans
IBM MQ Technical Product Manager
martin.evans1@ibm.com

Notices and disclaimers

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.
Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled,
isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply

with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers (Continued)

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**
The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, 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.

Table of Contents

1 INTRODUCTION 5

1.1 High Level Architecture 5
1.2 Lab Overview 7

2 GETTING STARTED 8

2.1 Login to the Lab Environment 8
2.2 Launch the Lab Environment 9
2.3 Launch RedHat OpenShift Console 11
2.4 Login into the RedHat OpenShift Console 12

3 CREATE ARGOCD APP TO DEPLOY MQ UNIFORM CLUSTER 14

3.1 Review and Update GitOps App Definition 17
3.2 Create and Review GitOps App 19
3.3 Synchronize GitOps App 21
3.4 Review the Queue Managers Deployment 23

4 DEPLOY MQ APPLICATION 26

4.1 Get jar file 27
4.2 Deploy jar file 29
4.3 Review deployment 31

5 VALIDATE UNIFORM CLUSTER CONNECTIVITY 33

5.1 Go to Queue Manager 2 33
5.2 Review connections to QM02 34
5.3 Go to Queue Manager 1 38
5.4 Review Connections to QM01 39

6 SCALE MQ APPLICATION 41

6.1 Increase the number of instances (aka pods) 41
6.2 Review connectivity 42

7 REBALANCE CONNECTIONS 44

7.1 Delete Queue Manager 44
7.2 Review connectivity 46

8 REVIEW ARGOCD DECLARATIVE MODEL 48

8.1 Review ArgoCd 48
8.2 Synchronize GitOps App 48
8.3 Review connectivity 50

9 CONGRATULATIONS 52

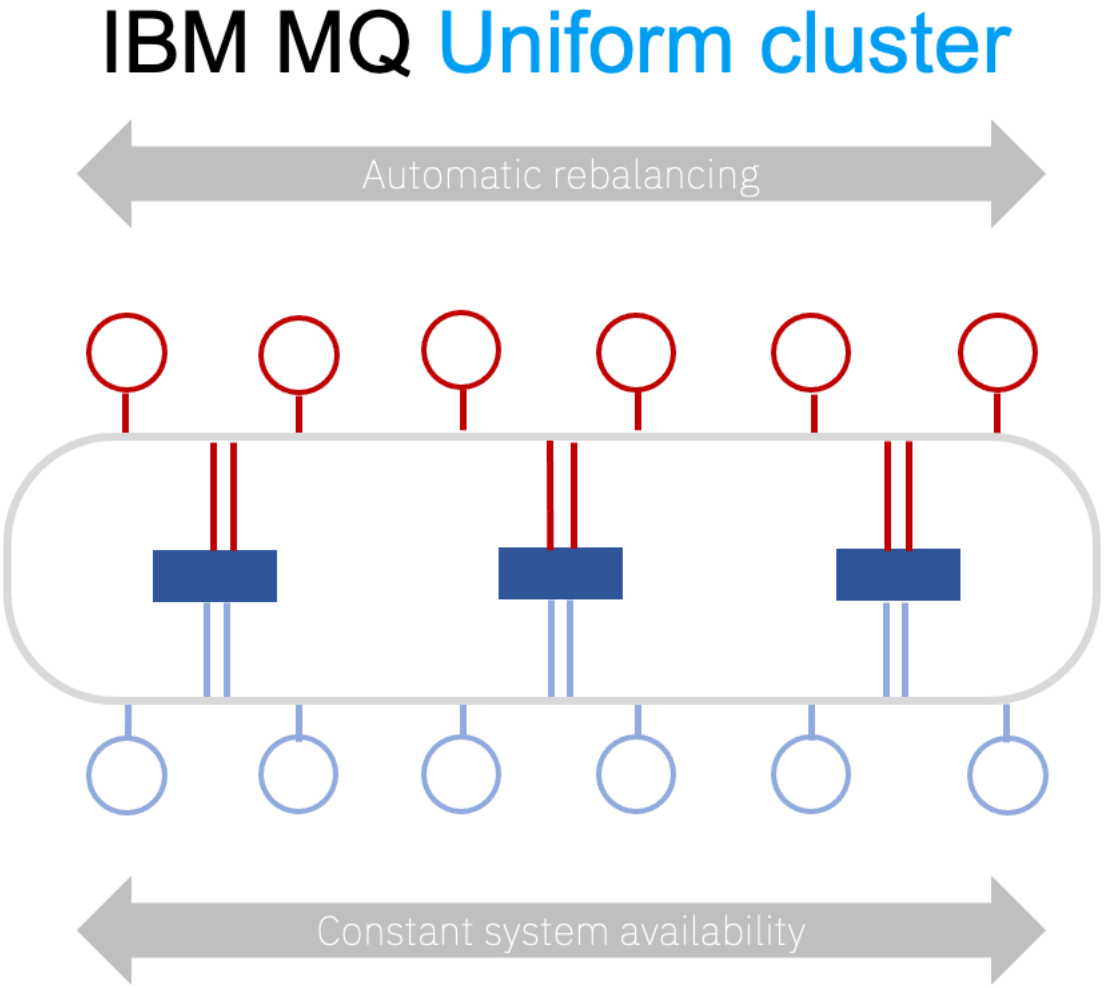
1 Introduction

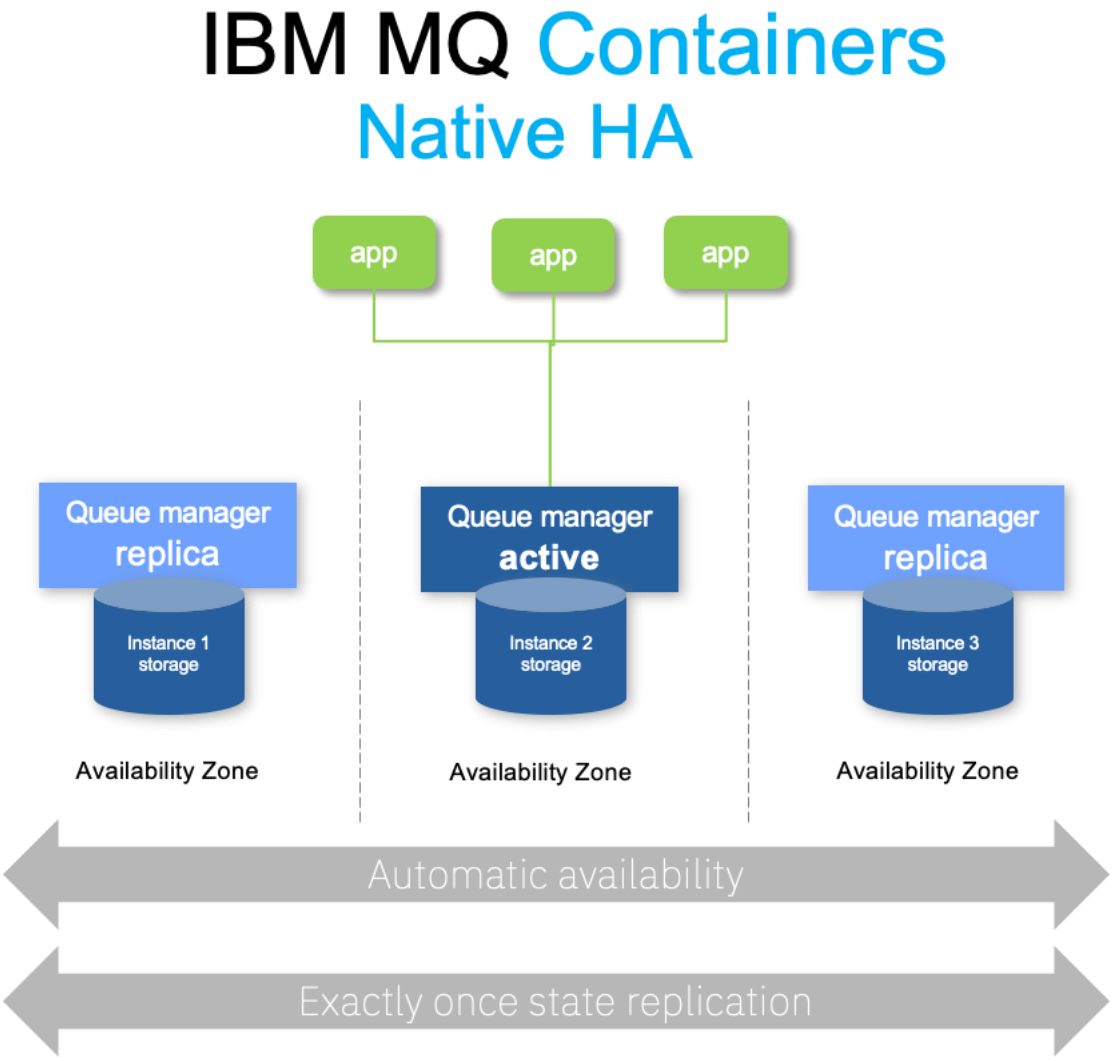
This cookbook will walk you through the deployment process of a MQ Uniform Cluster using GitOps principles where each queue manager uses the Native HA functionality to provide high availability.

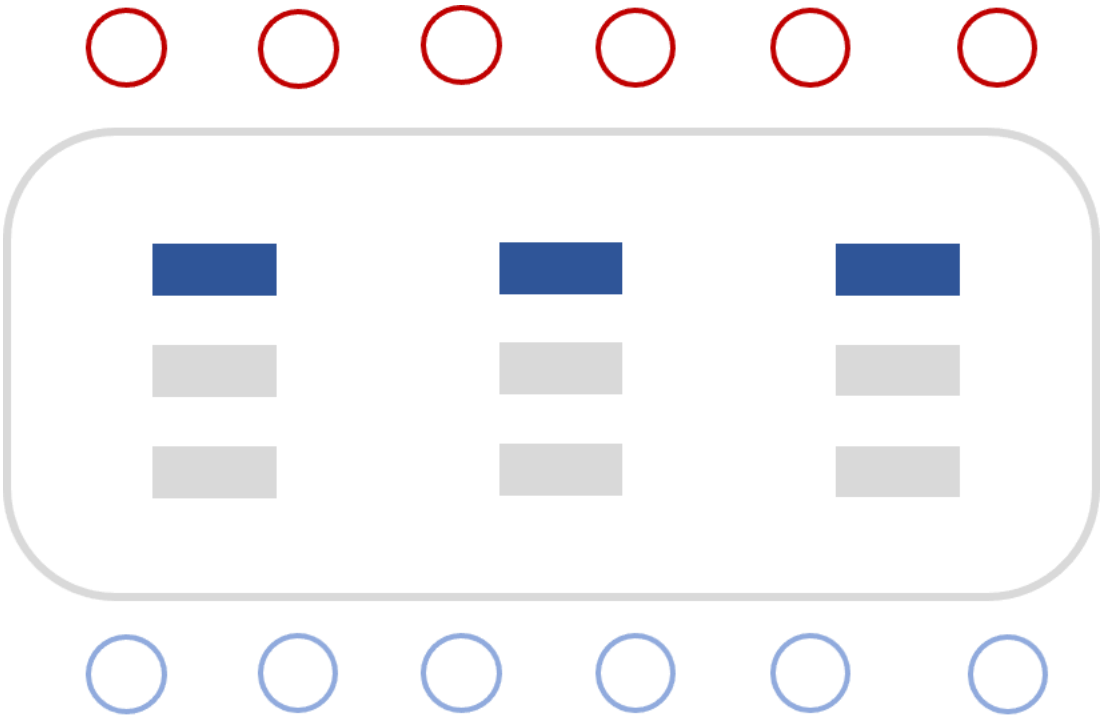
Then we will deploy an MQ application to demonstrate how it can take advantage of the Uniform Cluster when it scales.

1.1 High Level Architecture

The high-level architecture for the environment described in this document may be summarized as follows:




IBM MQ Containers
Native HA
app
app
app
Queue manager replica
Queue manager active
Queue manager replica
Instance 1 storage
Instance 2 storage
Instance 3 storage
Availability Zone
Availability Zone
Availability Zone
Automatic availability
Exactly once state replication



1.2 Lab Overview

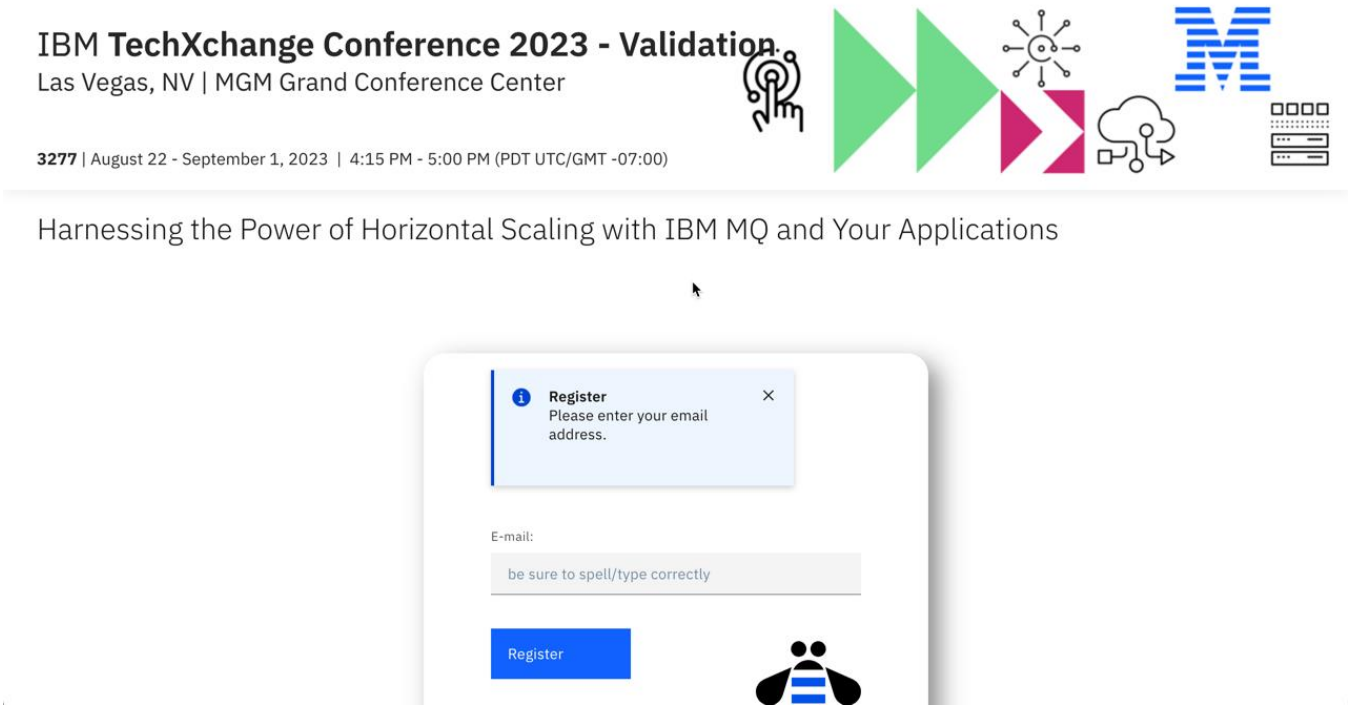
In this lab, your persona will be an Integration Specialist who will be deploying an MQ Cluster using GitOps principles, as well as an MQ Application with the OCP facilities. Your lab's computer will use a web browser to connect to the RedHat Openshift Cluster as well as the ArgoCD web UI to conduct the different activities.

The MQ Operator as well as an instance of RedHat OpenShift GitOps has been already installed, alongside with an instance of nginx to serve the CCDT that will be used by the MQ Application.

Happy Messaging!

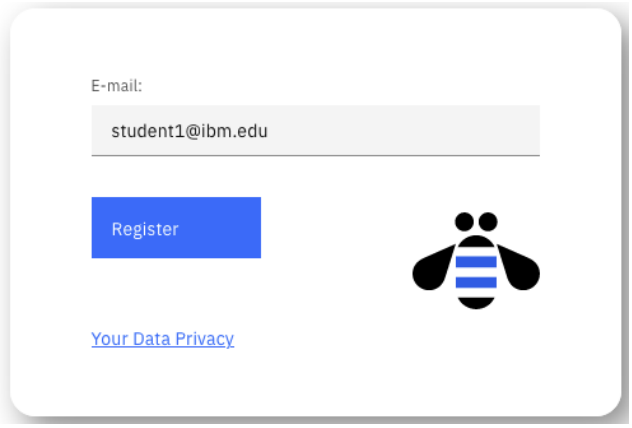
2 Getting Started

The laptop you have been assigned you should have a screen.

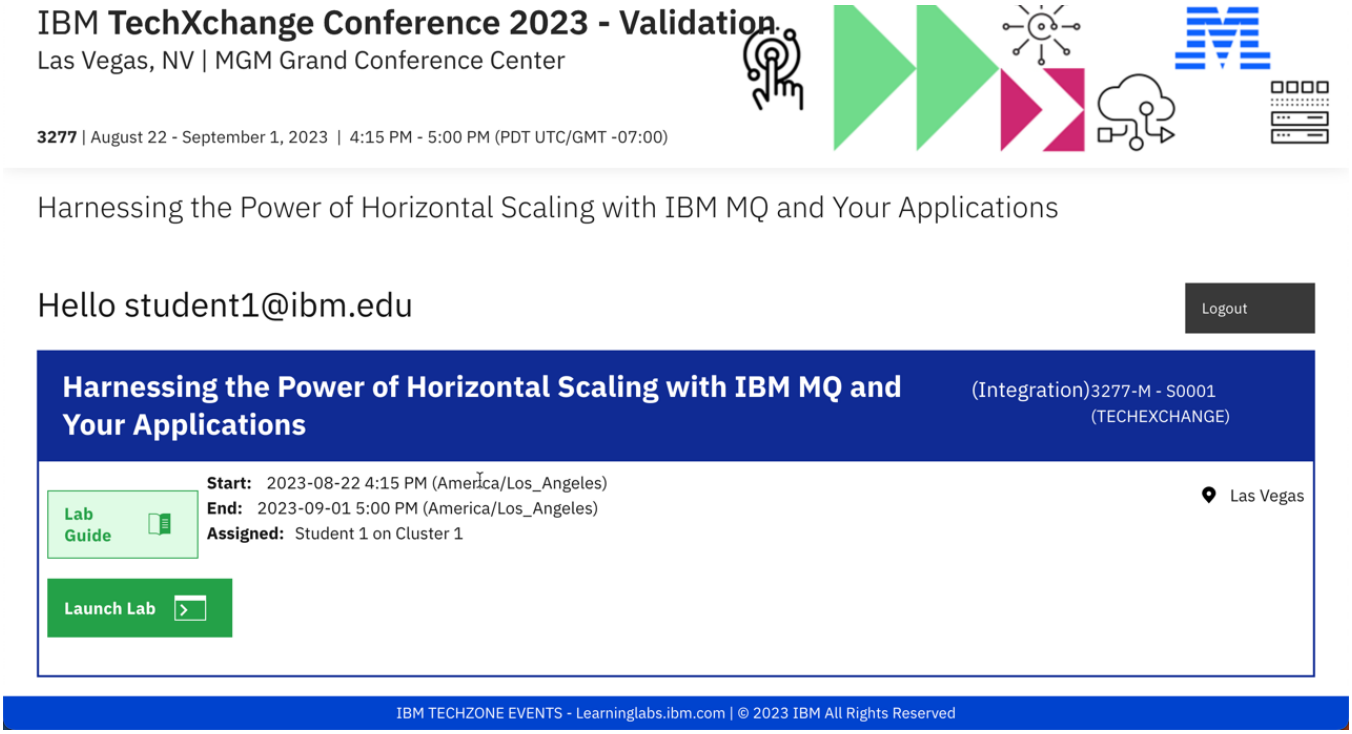


2.1 Login to the Lab Environment

In the Login form below, enter the e-mail that you used to register for TechXchange 2023 in the E-mail form and click on the Register button.



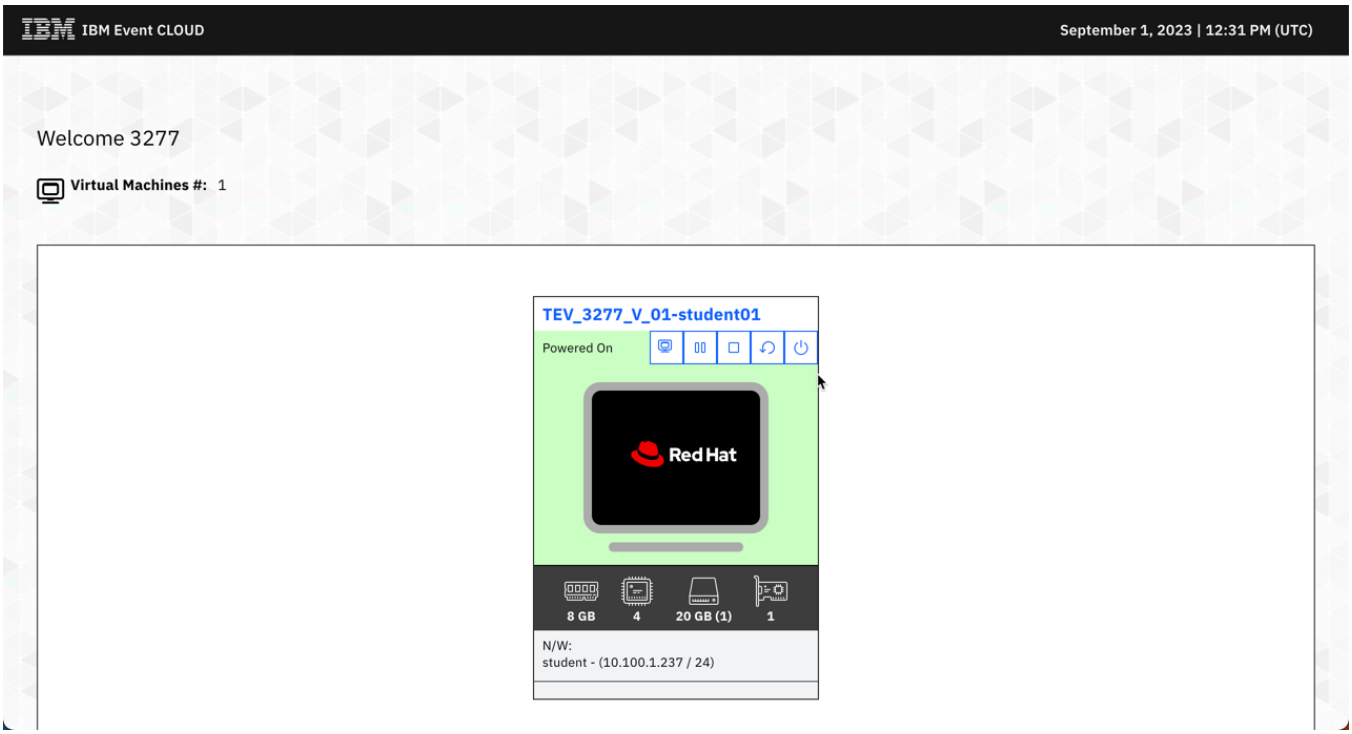
The Lab Home web page is displayed.



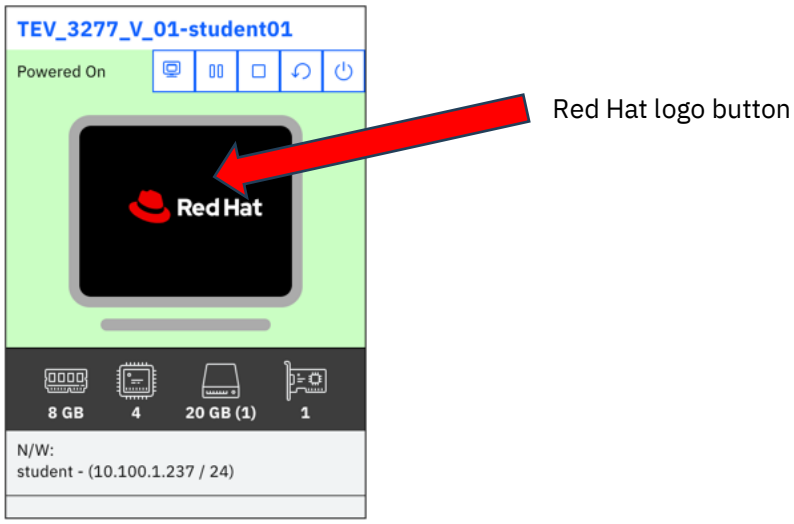
2.2 Launch the Lab Environment

Click on the 

button to launch the Lab Environment.

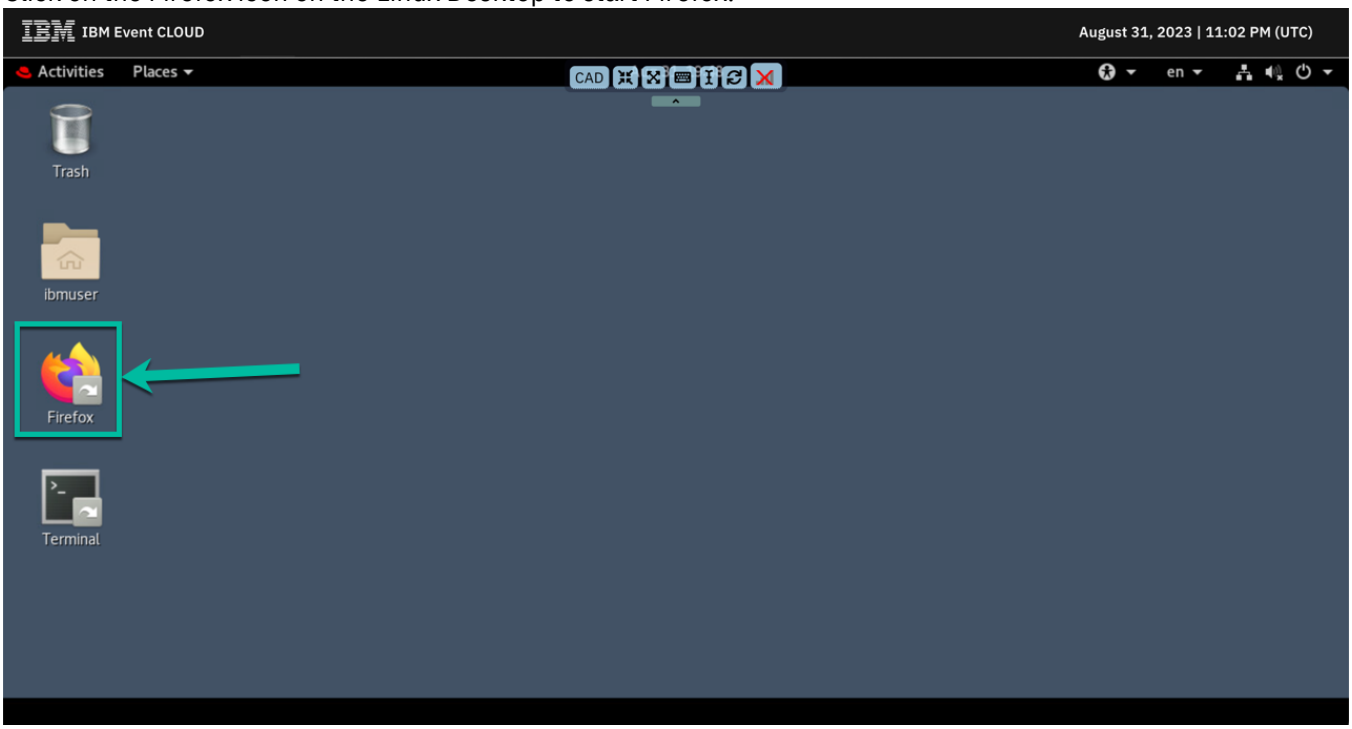


The VM should be named TEV_3277_V_01-student<<Number>> where <<number>> is your student number.



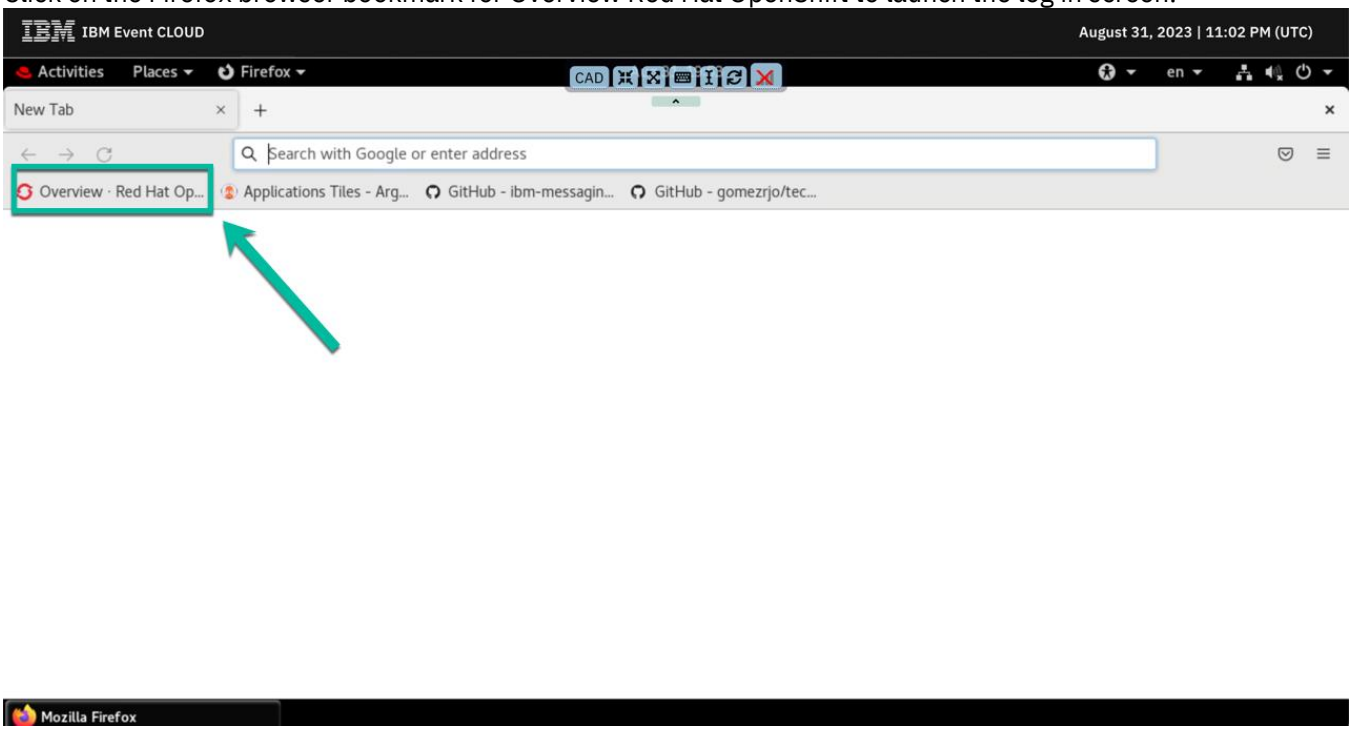
Click on the Red Hat logo button. This will launch the Lab Environment.

Click on the Firefox icon on the Linux Desktop to start Firefox.



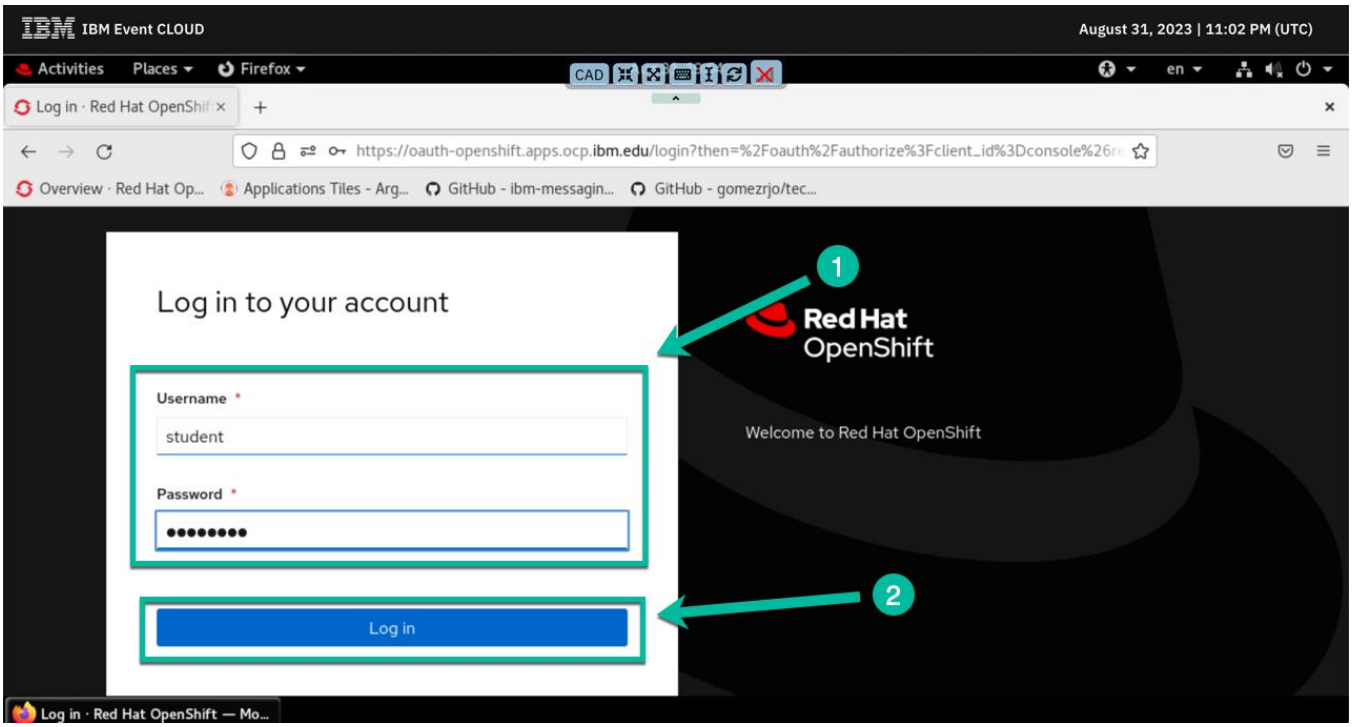
2.3 Launch RedHat OpenShift Console

Click on the Firefox browser bookmark for Overview Red Hat OpenShift to launch the log in screen.

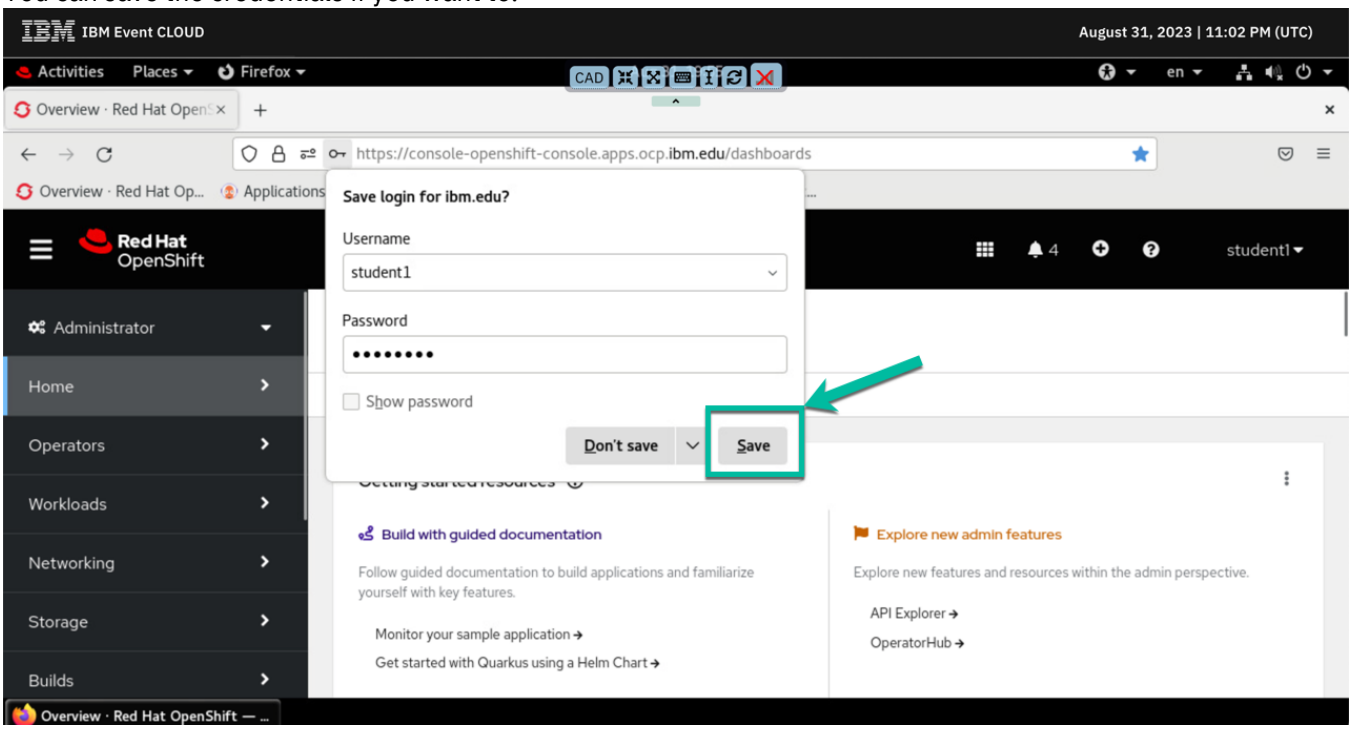


2.4 Login into the RedHat OpenShift Console

- For the Username field, put your ID like “student#” with no quotes where # is your number
- For the Password field, put in “Passw0rd” with no quotes.

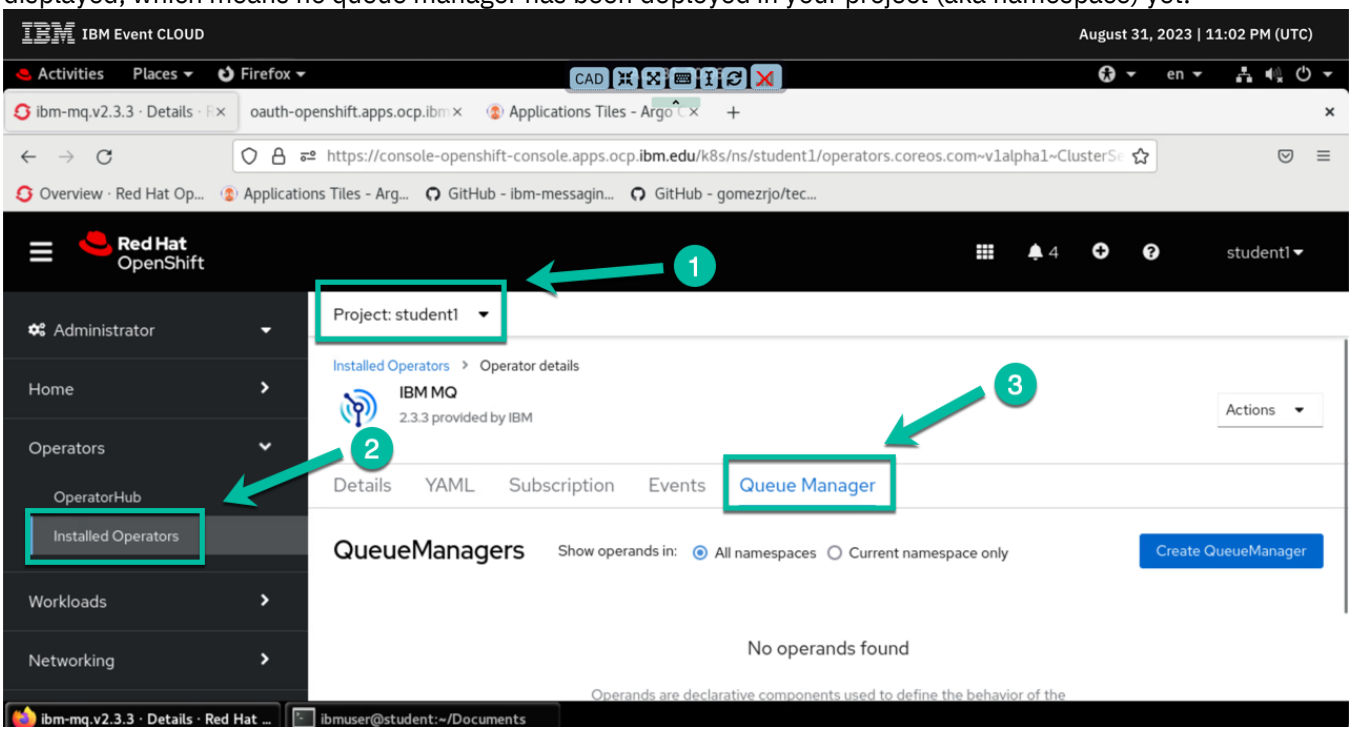


You can save the credentials if you want to.



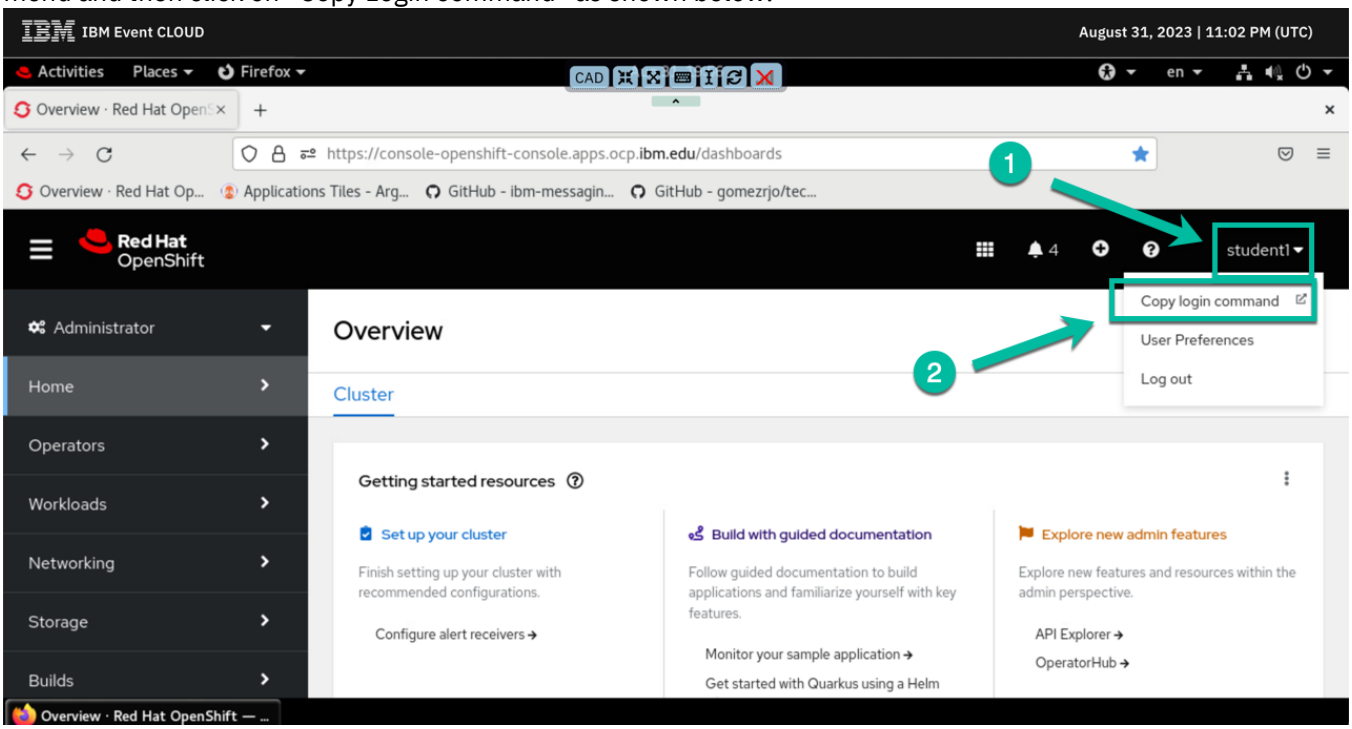
The OCP console is displayed. We will check that the MQ Operator is installed but no Queue Managers have been deployed yet.

First make sure you are in project “student#” (where # is your assigned student number), then navigate to the “Installed Operators” section and select “Queue Manager” as shown below and there are no queue managers displayed, which means no queue manager has been deployed in your project (aka namespace) yet.

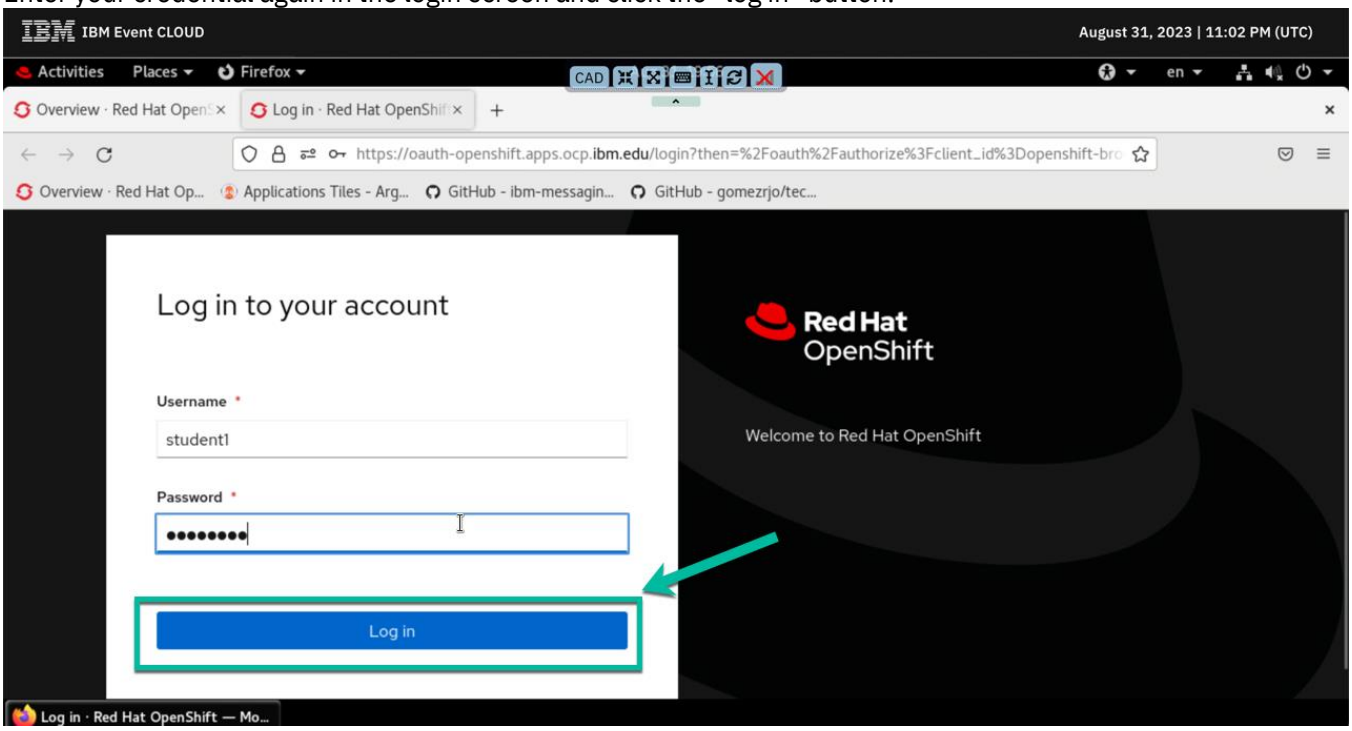


3 Create ArgoCD App to deploy MQ Uniform Cluster

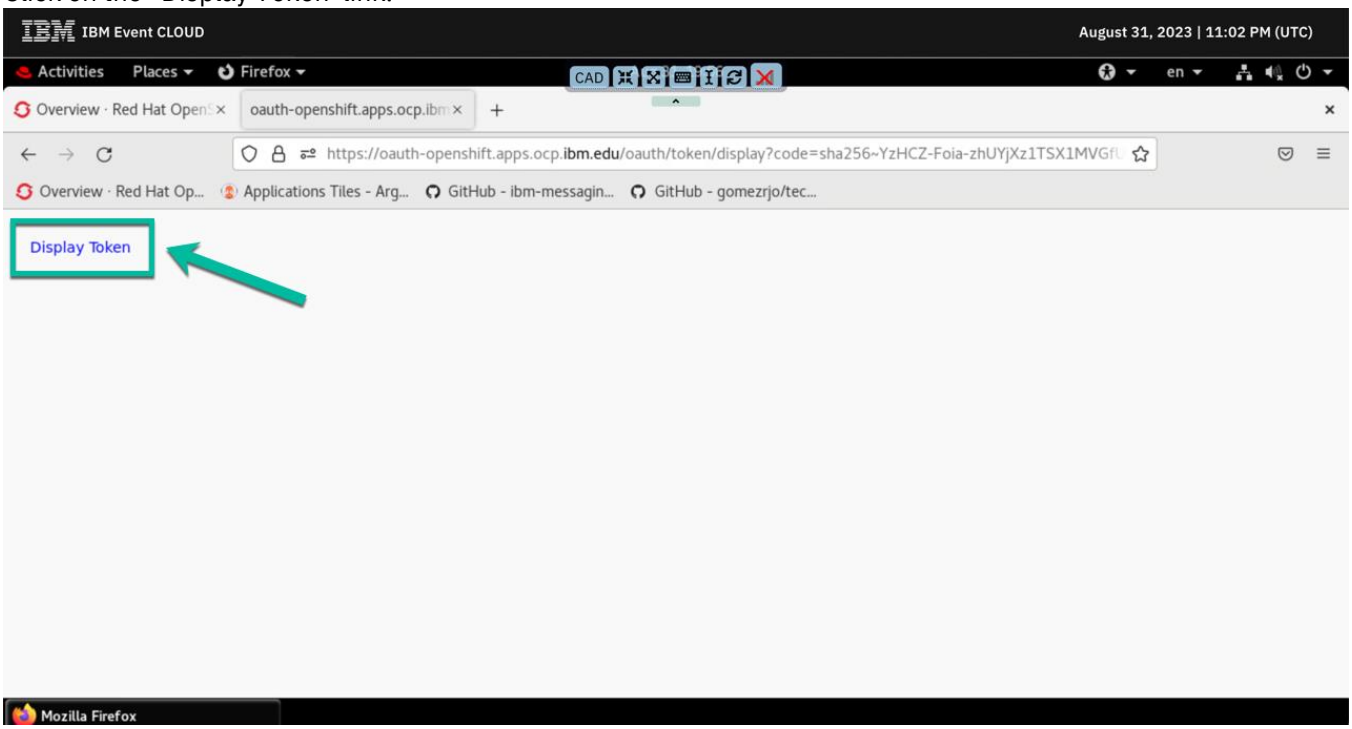
We will deploy the application using the oc cli. For that we need to get an OCP token via the “login command”. Navigate to the username in the upper right corner of the browser and click on the drop down button to get the menu and then click on “Copy Login command” as shown below.



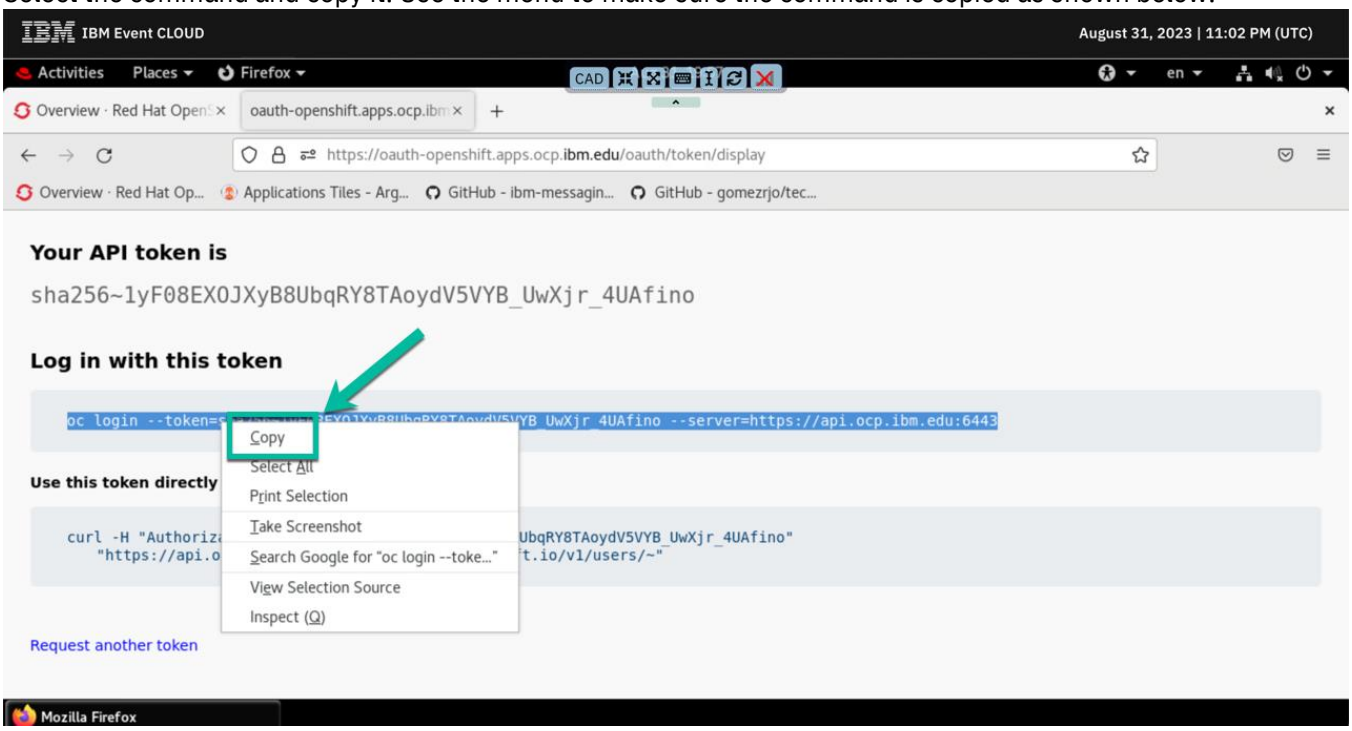
Enter your credential again in the login screen and click the “log in” button.



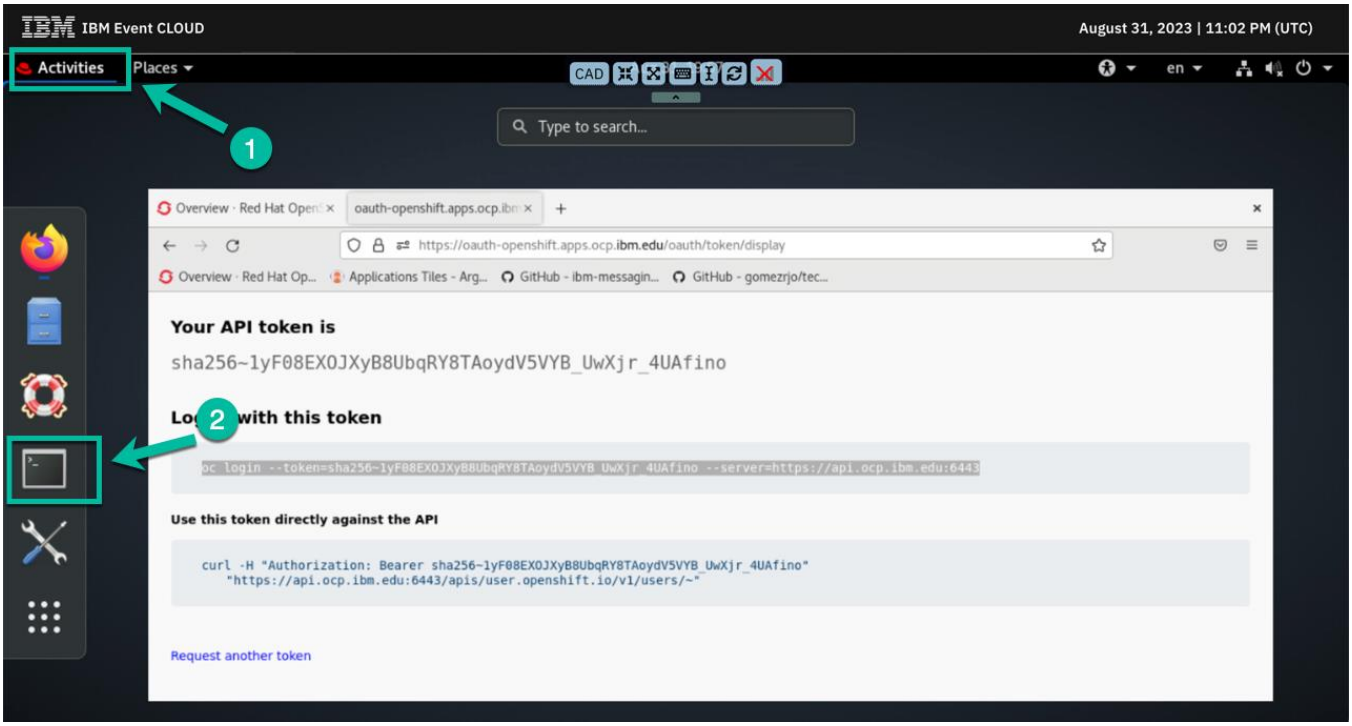
Click on the “Display Token”link.



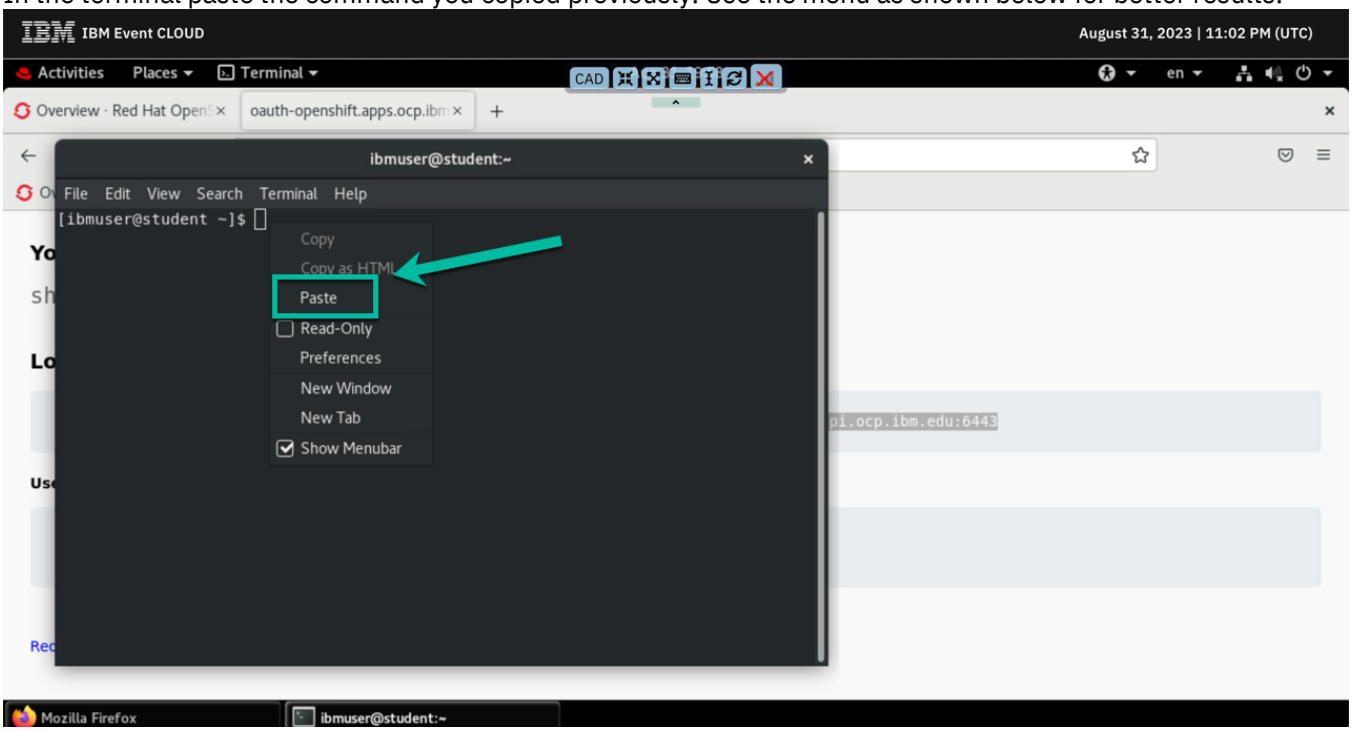
Select the command and copy it. Use the menu to make sure the command is copied as shown below.



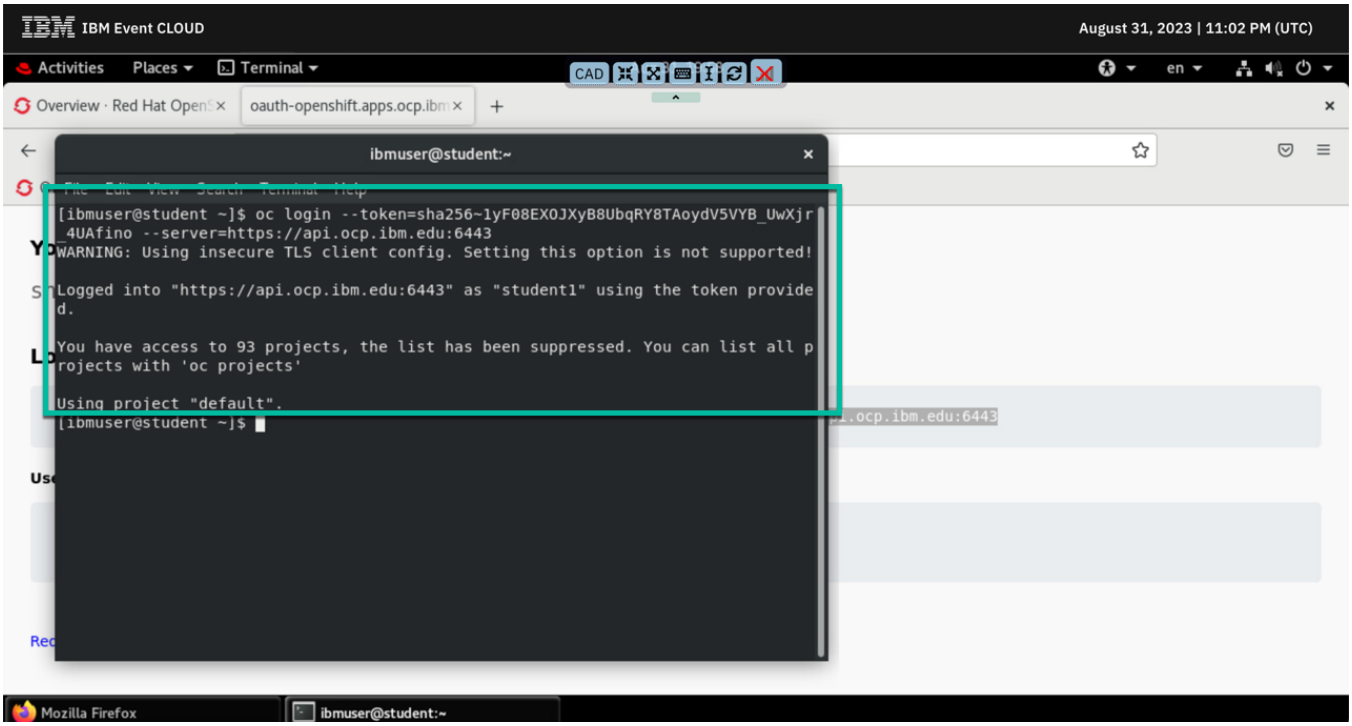
Now we will open a terminal to run the command. If you are not familiar with RHEL, select the “Activities” button in the upper left corner and then the terminal icon as shown below.



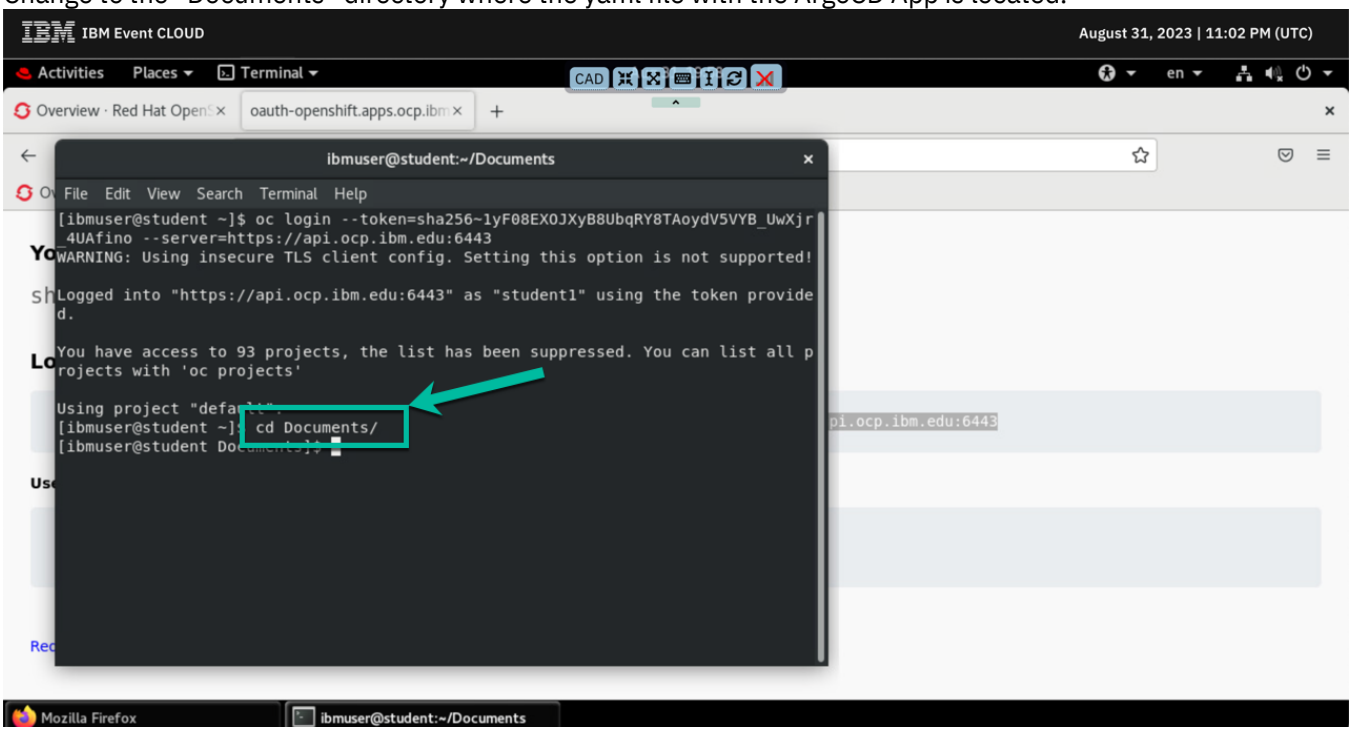
In the terminal paste the command you copied previously. Use the menu as shown below for better results.



You will get a message as shown below to confirm login was successful.

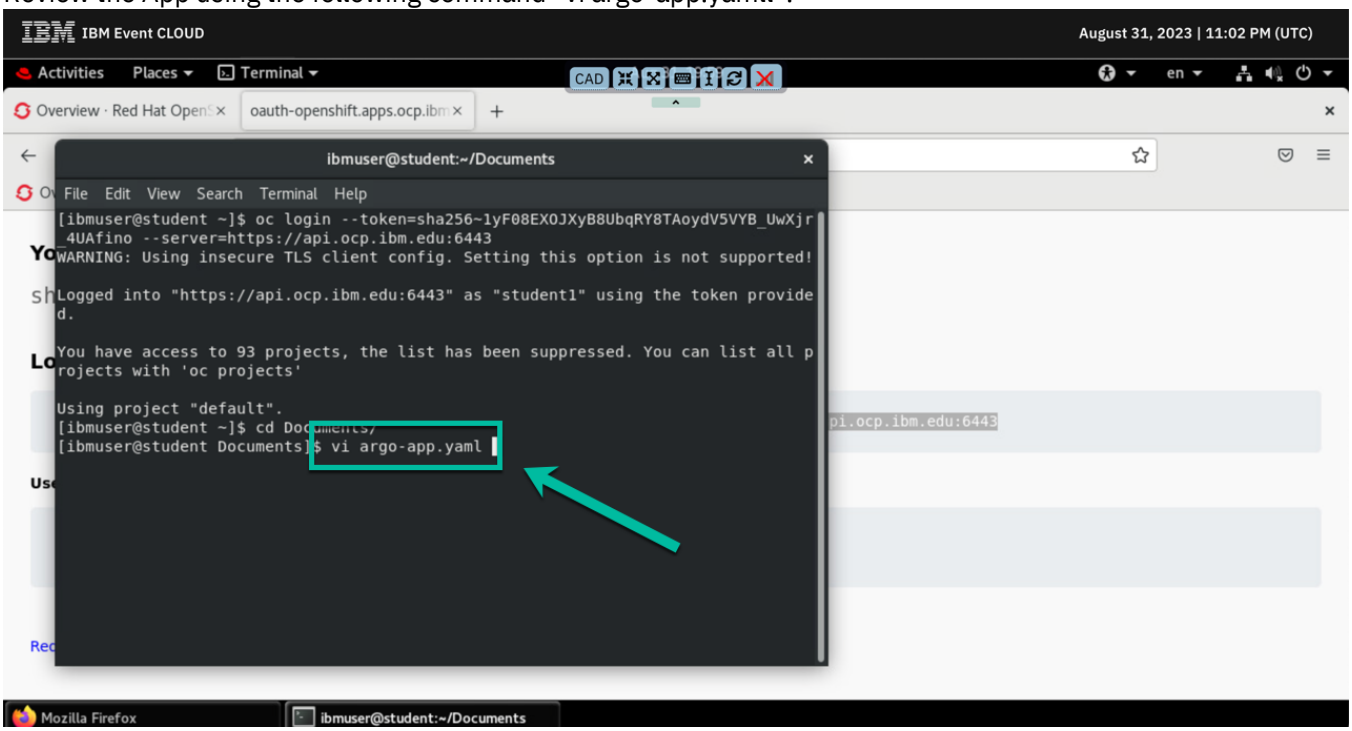


Change to the “Documents” directory where the yaml file with the ArgoCD App is located.

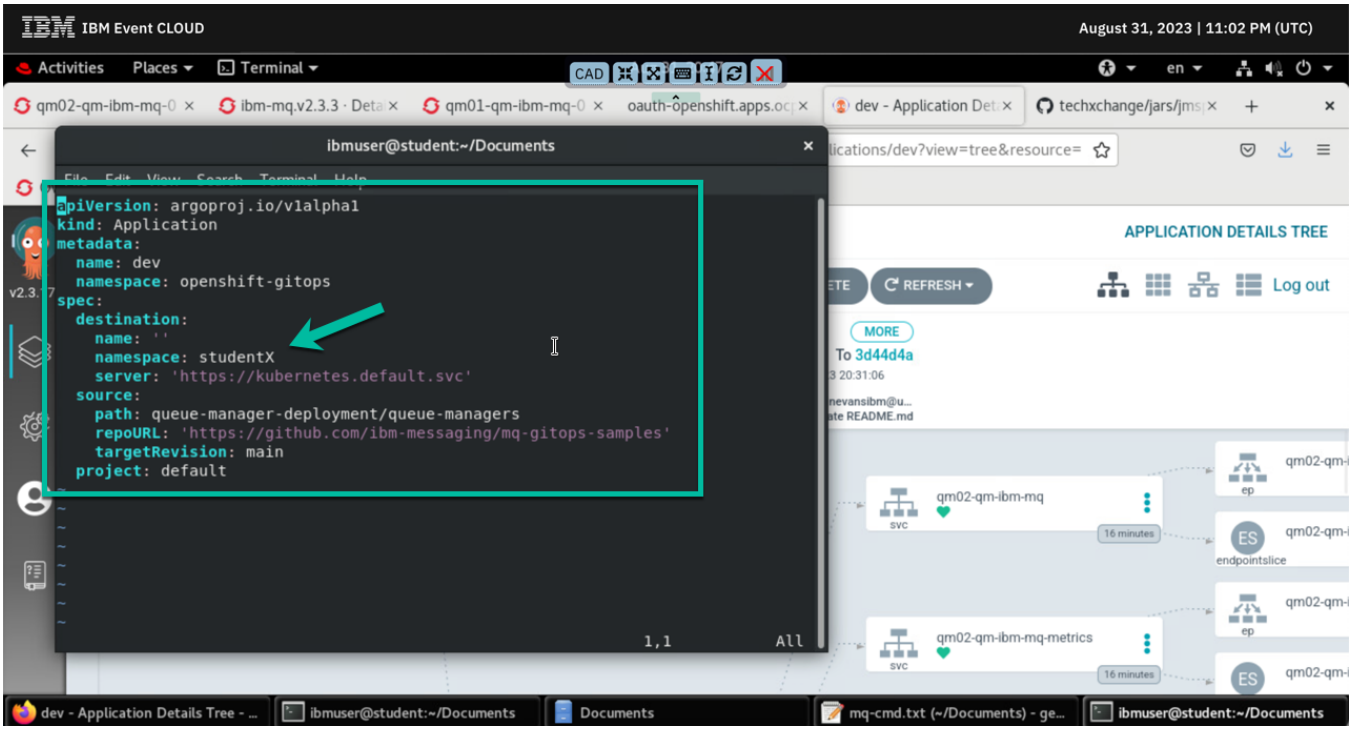


3.1 Review and Update GitOps App Definition

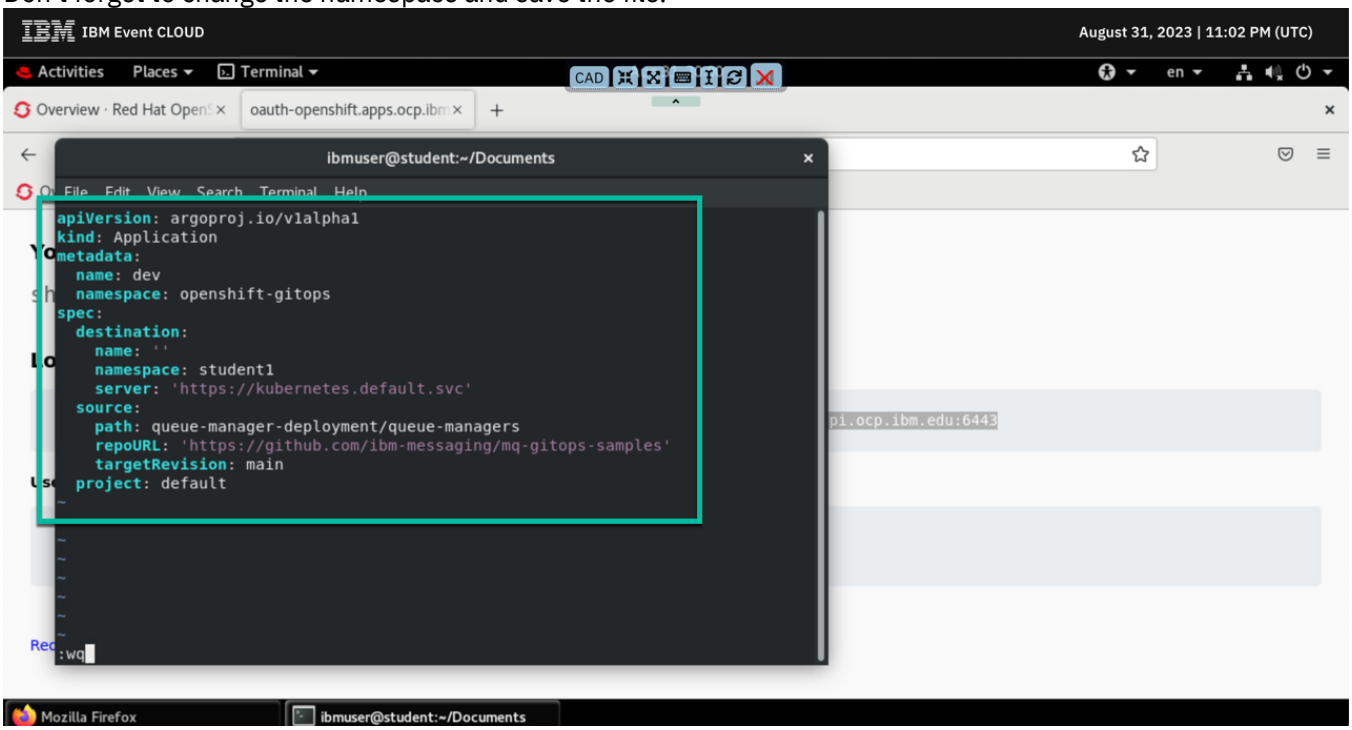
Review the App using the following command “vi argo-app.yamll”.



Before we review the content note that “namespace” has a value of “studentX”, and you will need to replace the “X” for your student number before you deploy the app.
This is not an ArgoCD class, but the two main sections are “source” and “destination”. The “repoURL” is pointing to the github repo that contains all the artifacts used by ArgoCD to deploy the queue managers. There is a bookmark in the browser if you want to explore.
The “destination” section is the location where the app and all the associated resources will be deployed. This is why you need to make sure it points to your assigned namespace.

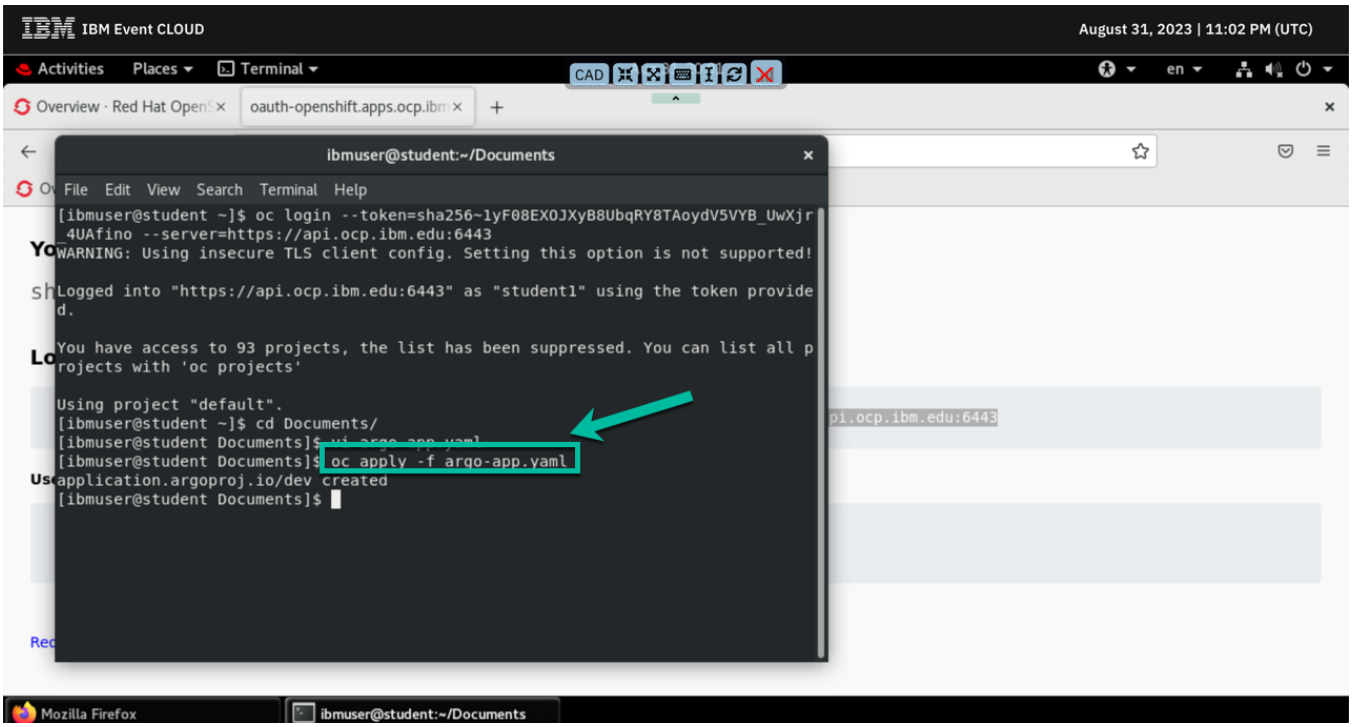


Don’t forget to change the namespace and save the file.

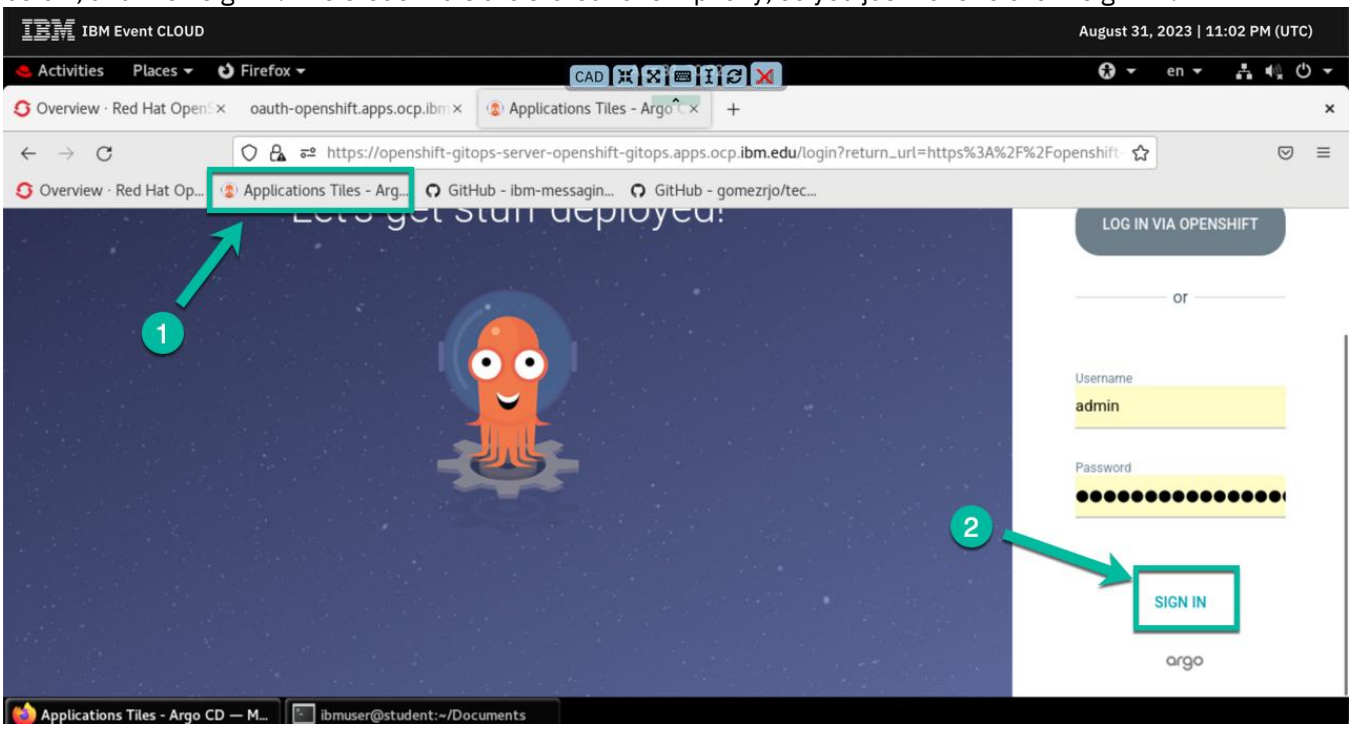


3.2 Create and Review GitOps App

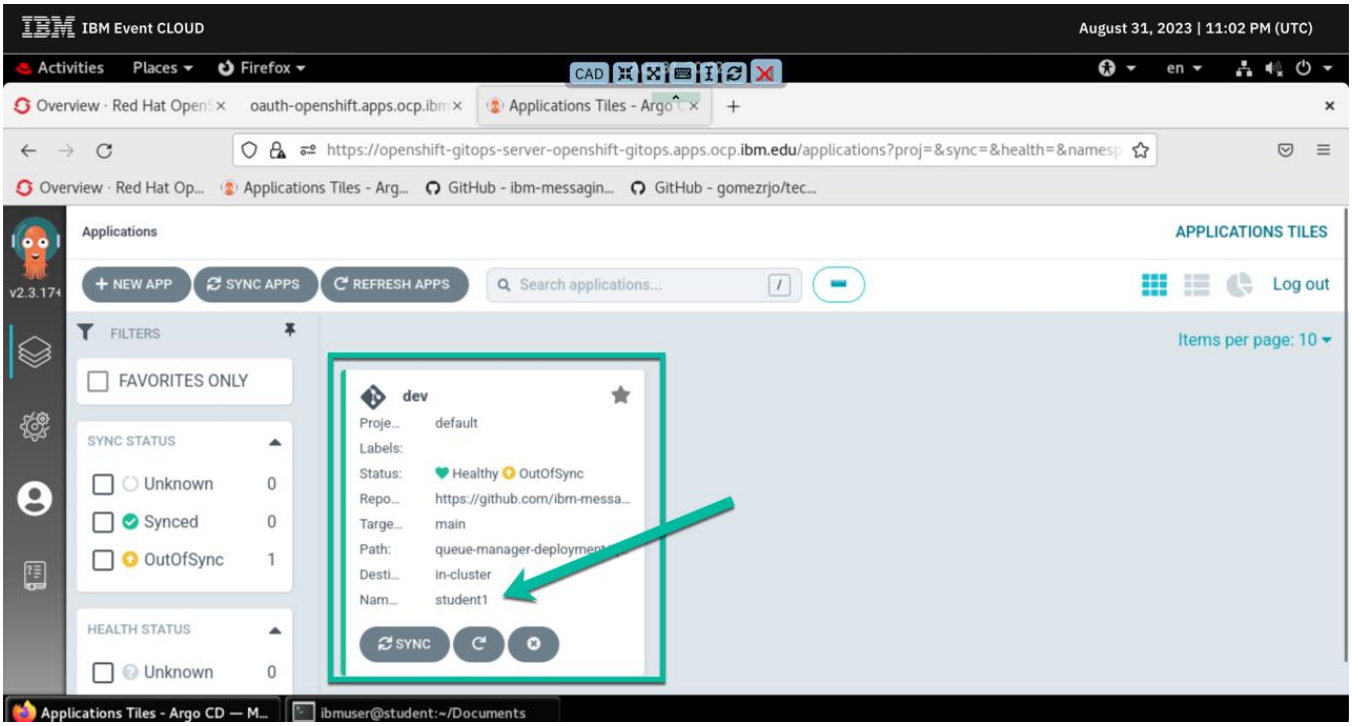
Now you can create the GitOps App using the following command “oc apply -f argo-app.yaml”, as shown below.



Navigate to the ArgoCD web UI clicking in the “Application Tiles – ArgoCD” bookmark in the browser and shown below, and then sign in. The credentials are stored for simplicity, so you just have to click “sign in”.

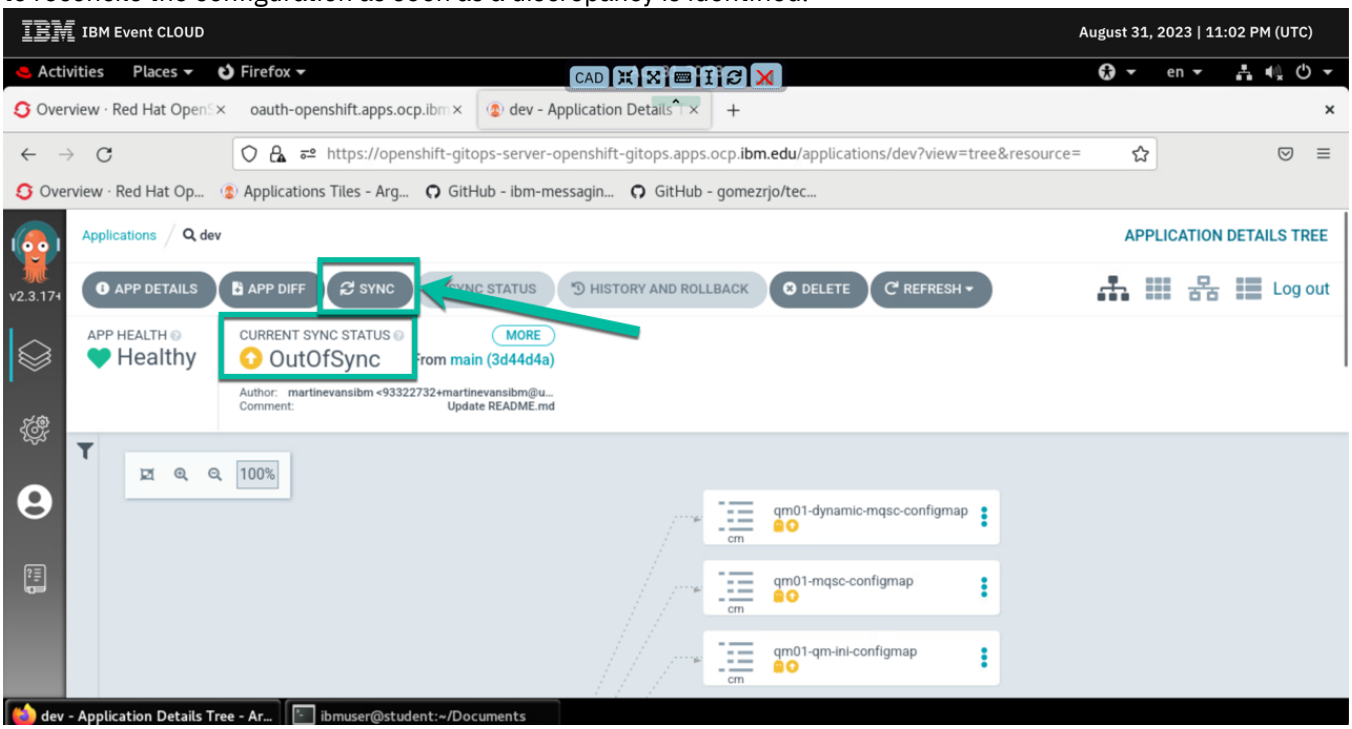


Note we are sharing a single account with the whole group so make sure you only work with your app since you may see other applications. The best way to identify yours is by the namespace where it is running. Click on the tile that represents you app to review the details.

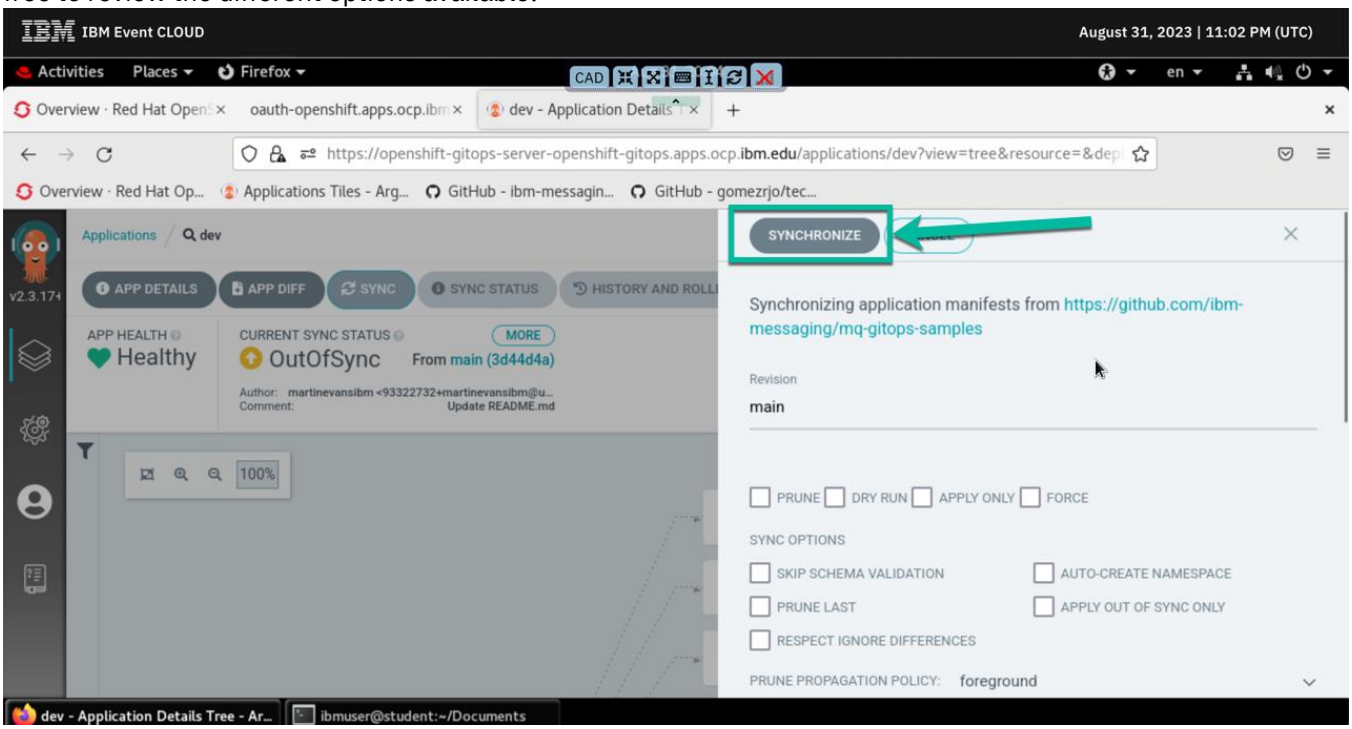


3.3 Synchronize GitOps App

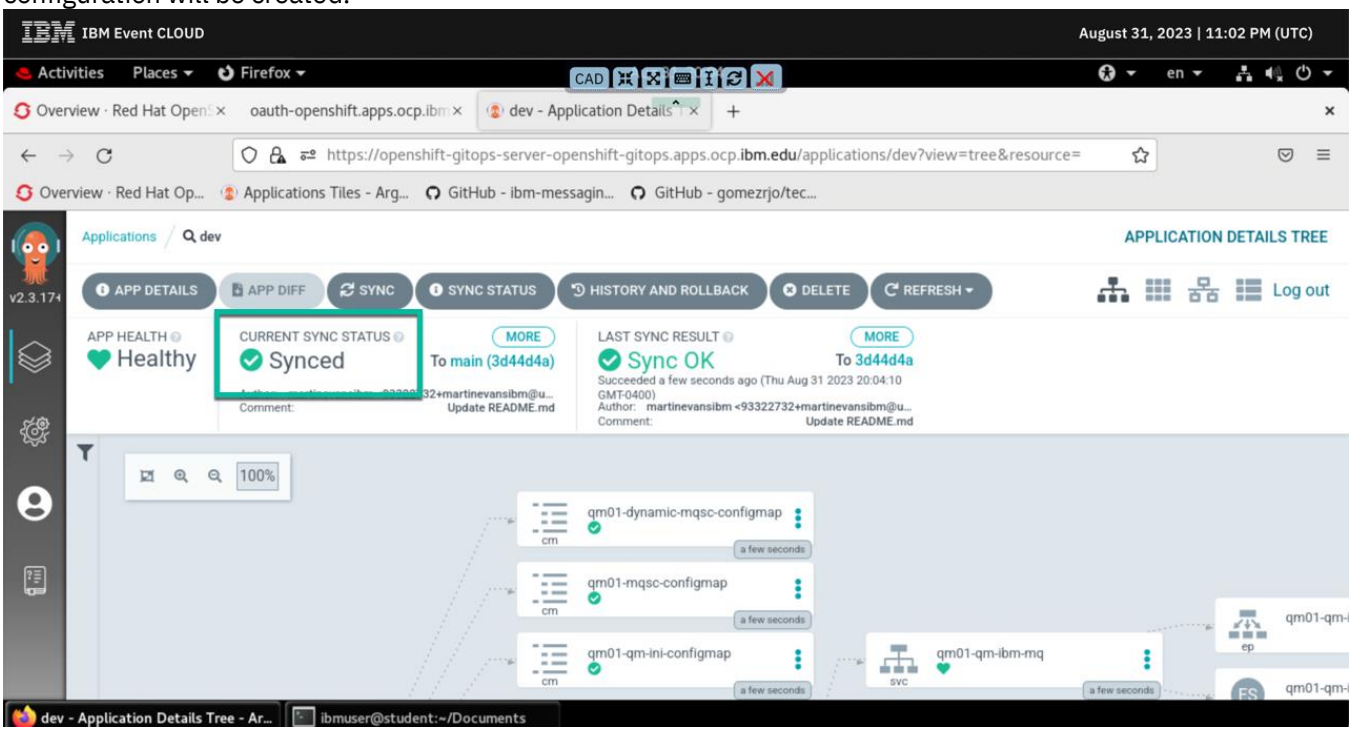
Note the status of the app is “OutOfSync”. This is on purpose because we have configured manual synchronization to have full control, but in a production environment you can configure automatic syncronziation to reconcile the configuration as soon as a discrepancy is identified.



Click the “Synchronize” button in the new window to start the process. We will use the default values but feel free to review the different options available.

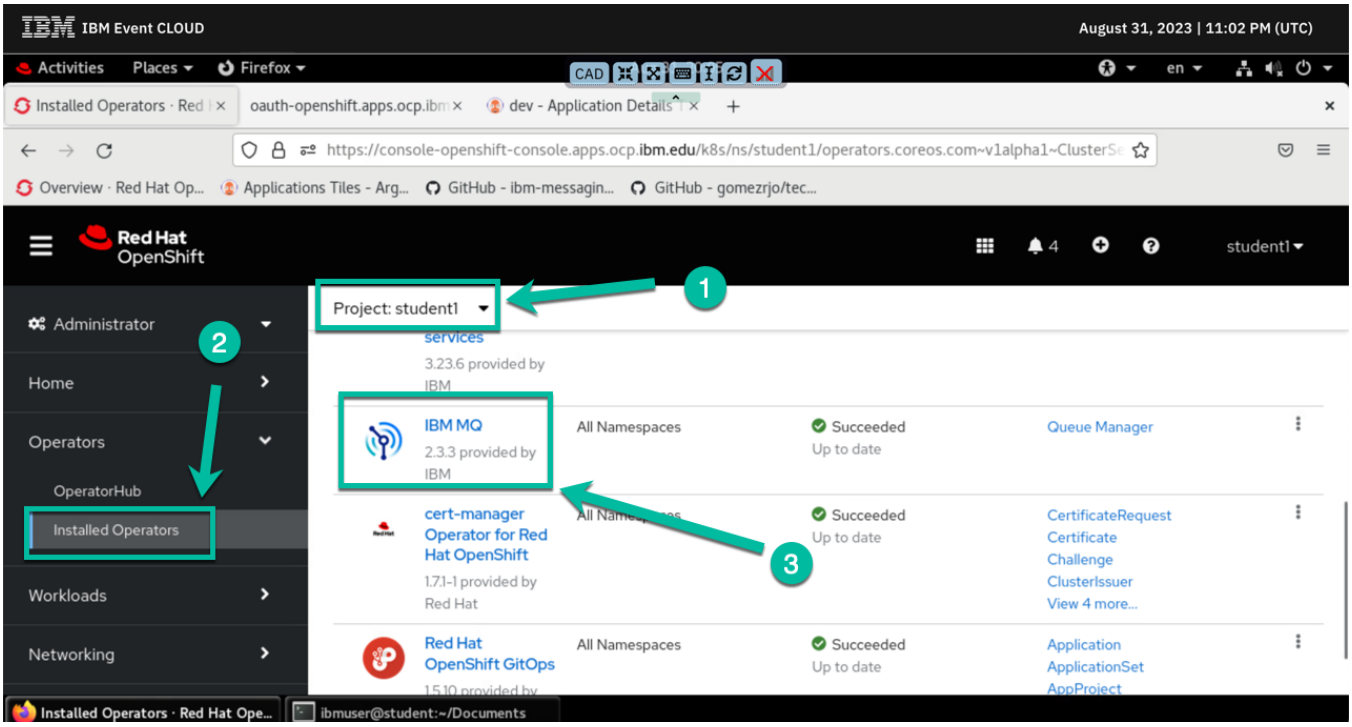


After a few seconds the status will change to “Synced” and the queue managers with the corresponding configuration will be created.

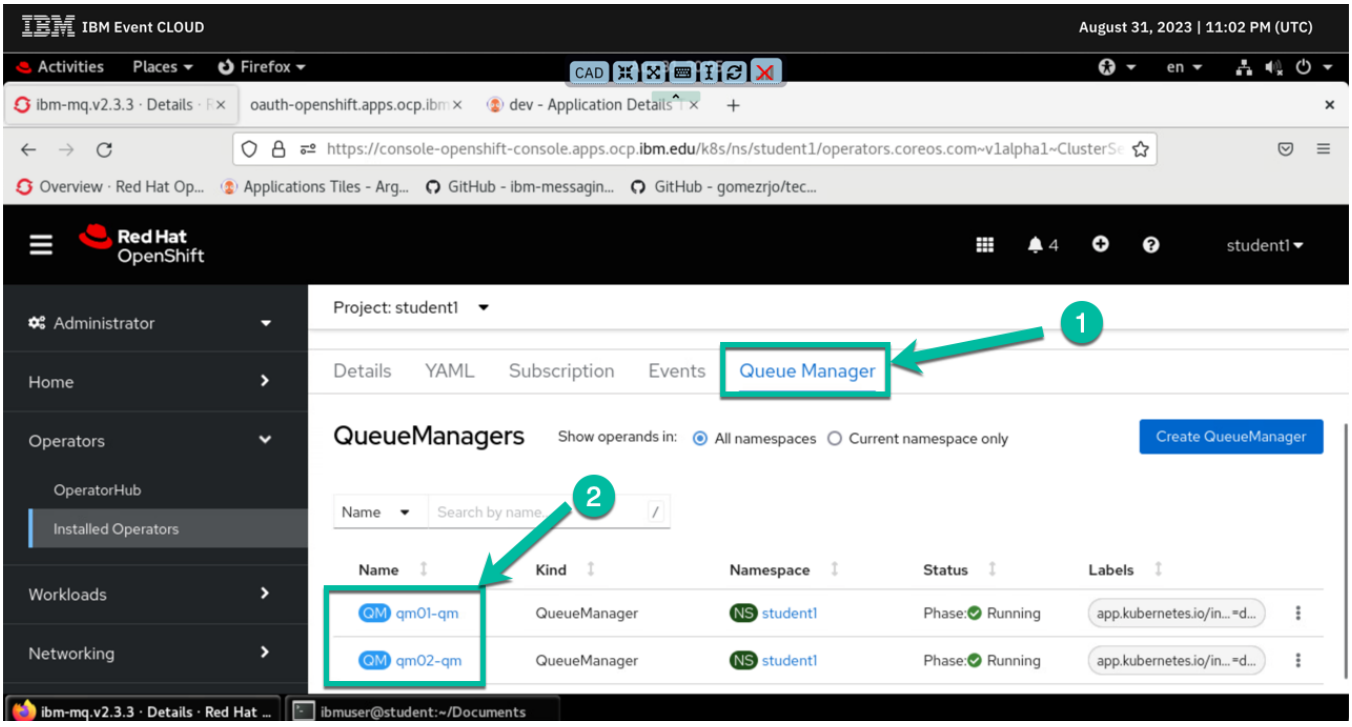


3.4 Review the Queue Managers Deployment

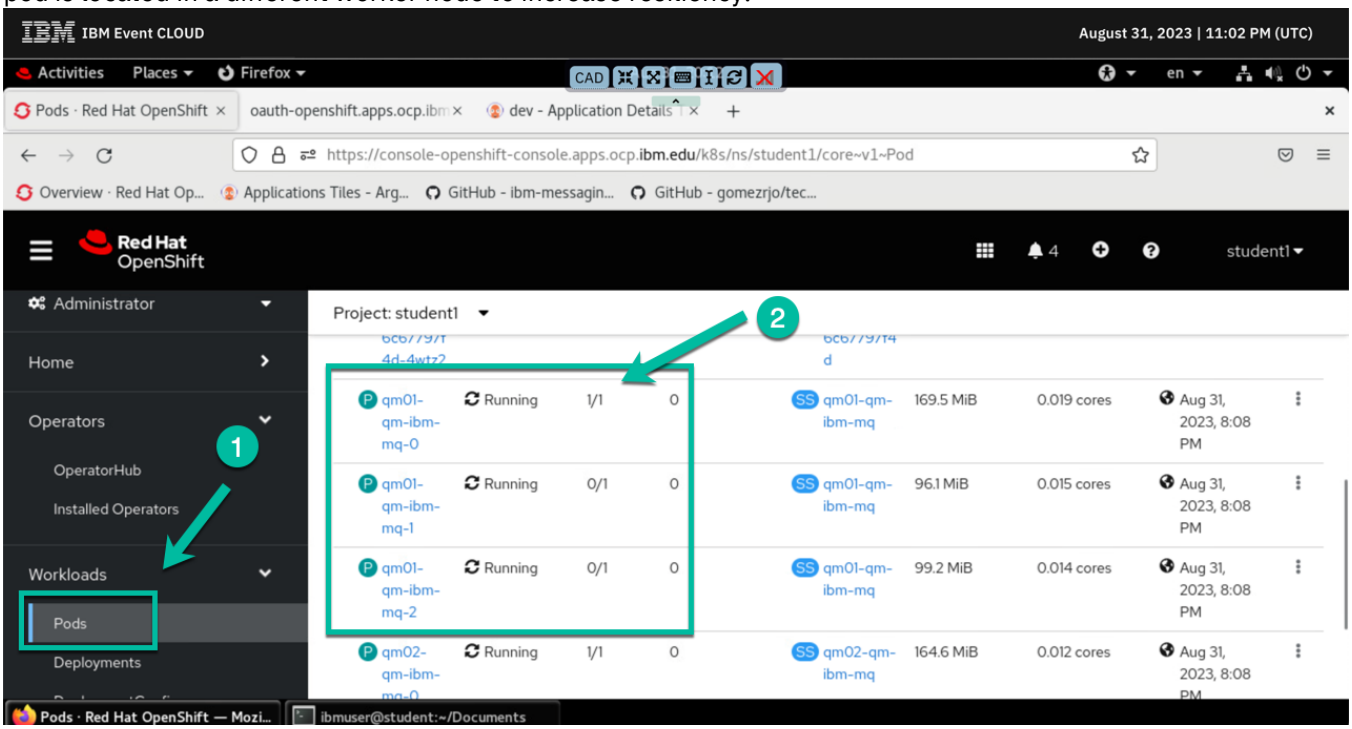
Navigate to the OCP Console in the other tab and go to the MQ Operator. Making sure first you are in the right project then selecting the “Installed Operators” section and finally clicking in the “IBM MQ Operator” row as shown below.



Then select “Queue Manager” and you will see two queue managers deployed. Both queue managers conform a Uniform Cluster.



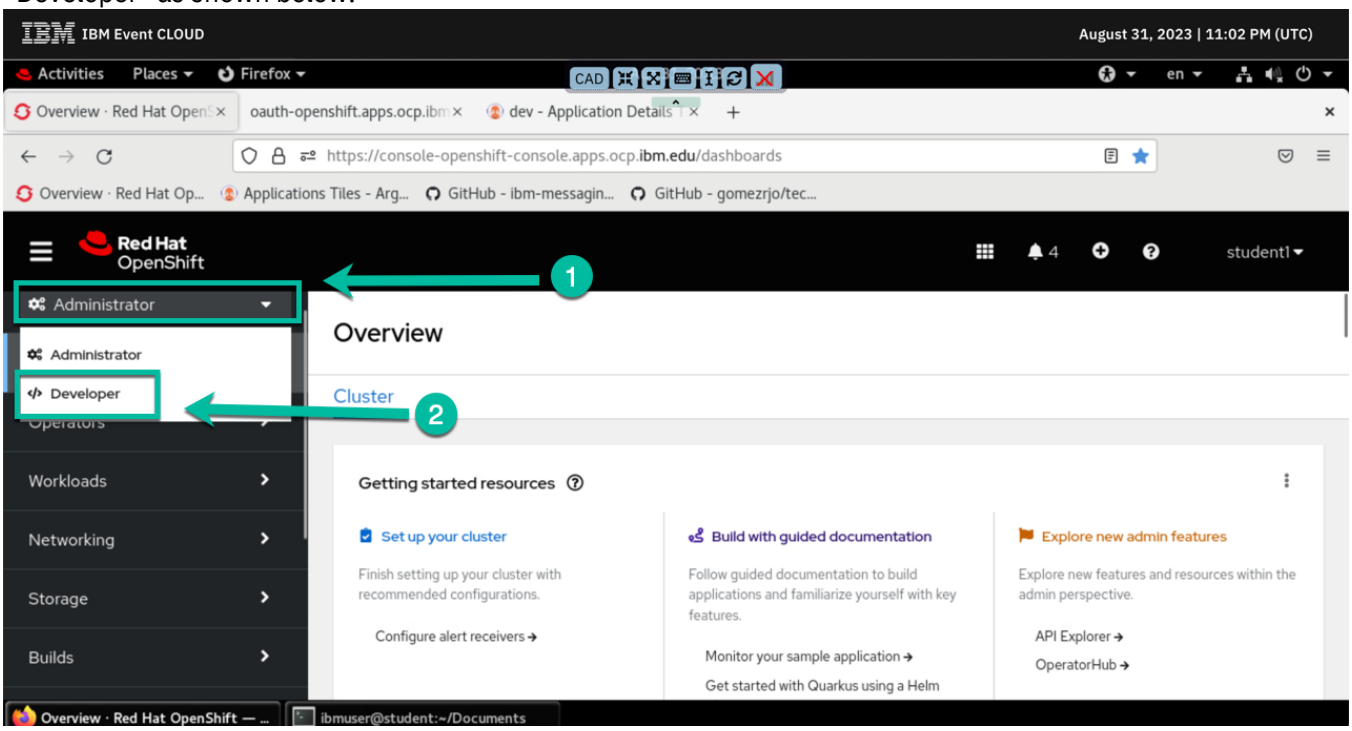
Additionally, each queue manager was configured with Native HA availability, so if you navigate to the “Pods” section in your own namespace you will see each queue manager is composed of 3 pods, but only one of them is active. You can identify it with the “1/1” value. The other two are “0/1” which means they are in standby mode waiting to take over in case there is an issue with the primary pod. If you look deeper, you will notice that each pod is located in a different worker node to increase resiliency.



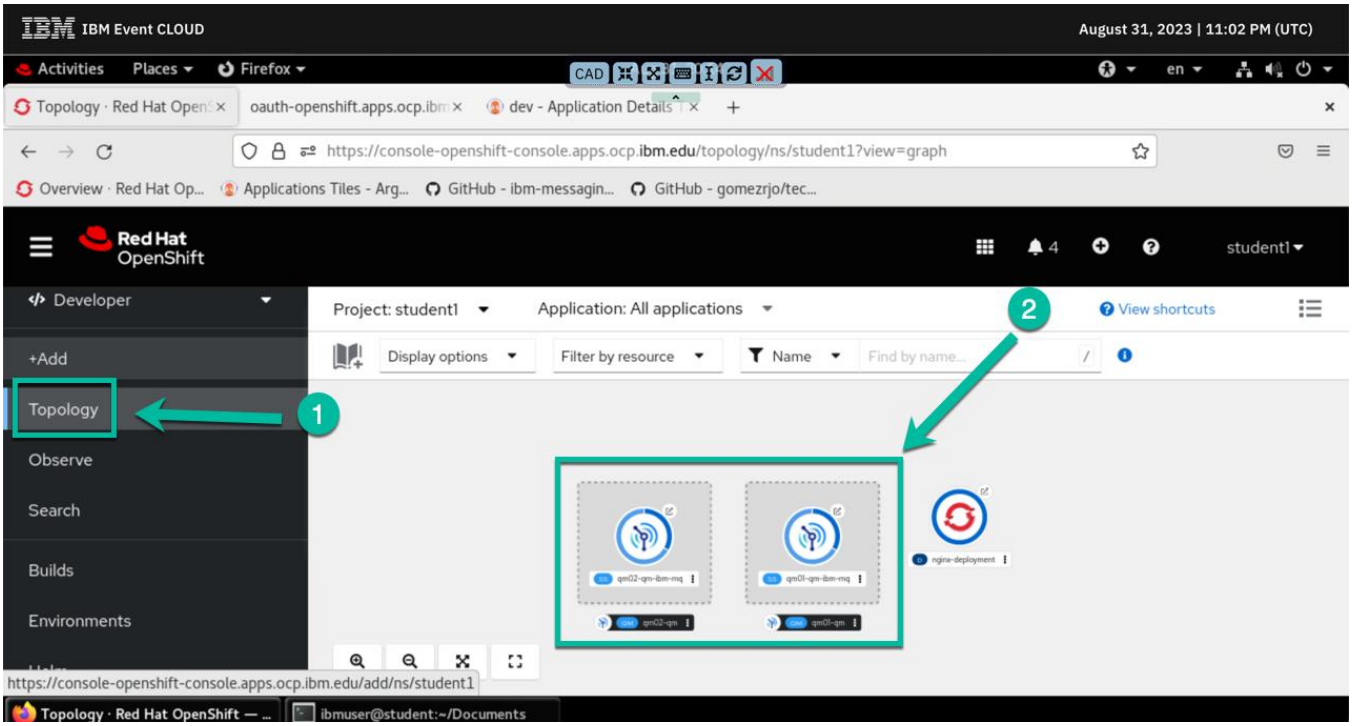
4 Deploy MQ Application

Now that the uniform cluster is running, we can proceed to deploy the application that will be interaction with the queue managers.

First, we will switch to the “Developer” perspective. To do so, click on the “Administrator” option and select “Developer” as shown below.

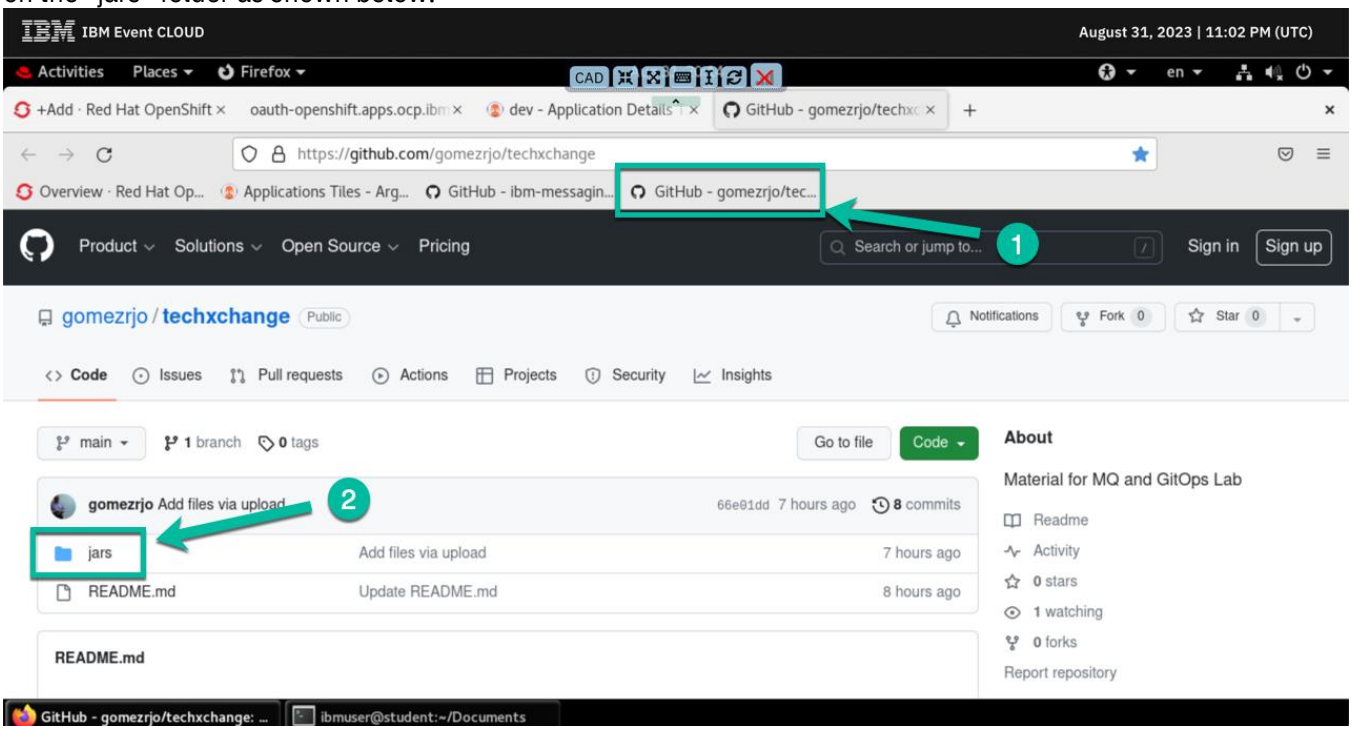


In this perspective you can view the queue managers. Select the topology section and you will see the tiles representing each queue manager.

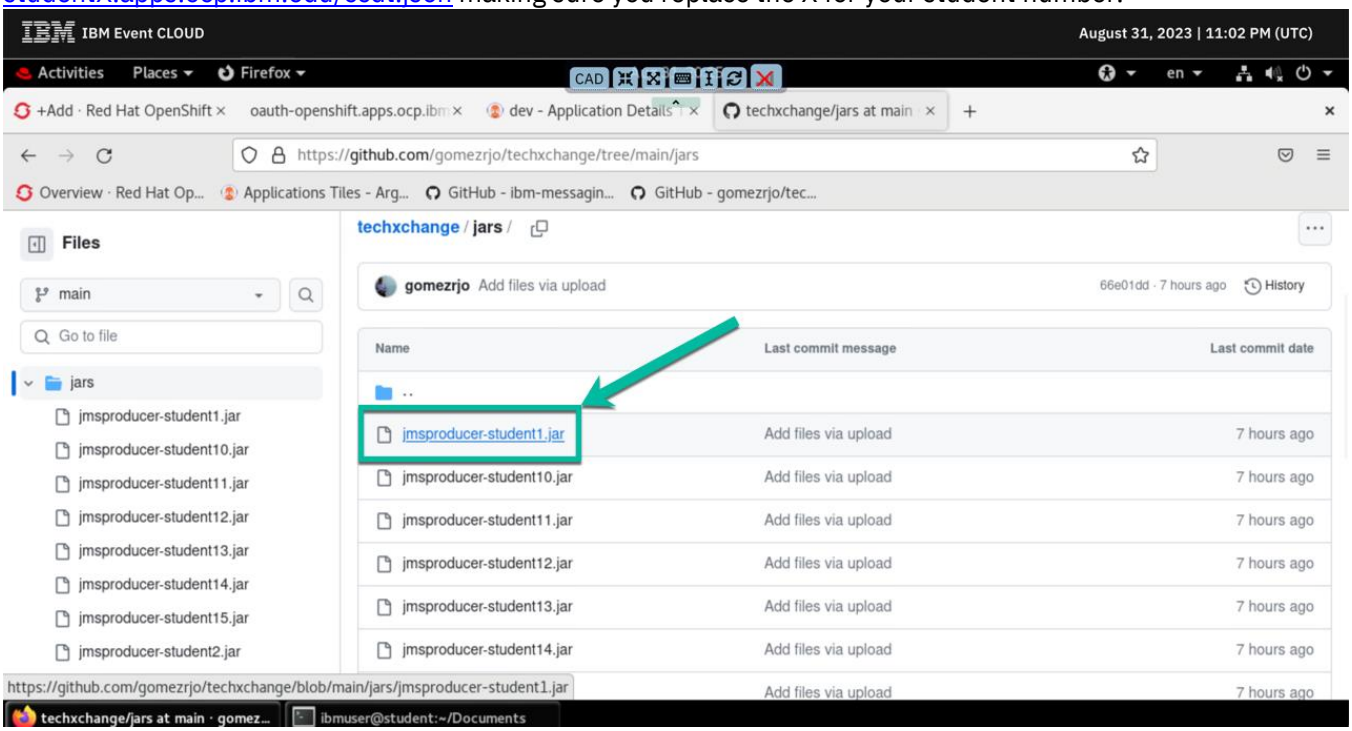


4.1 Get jar file

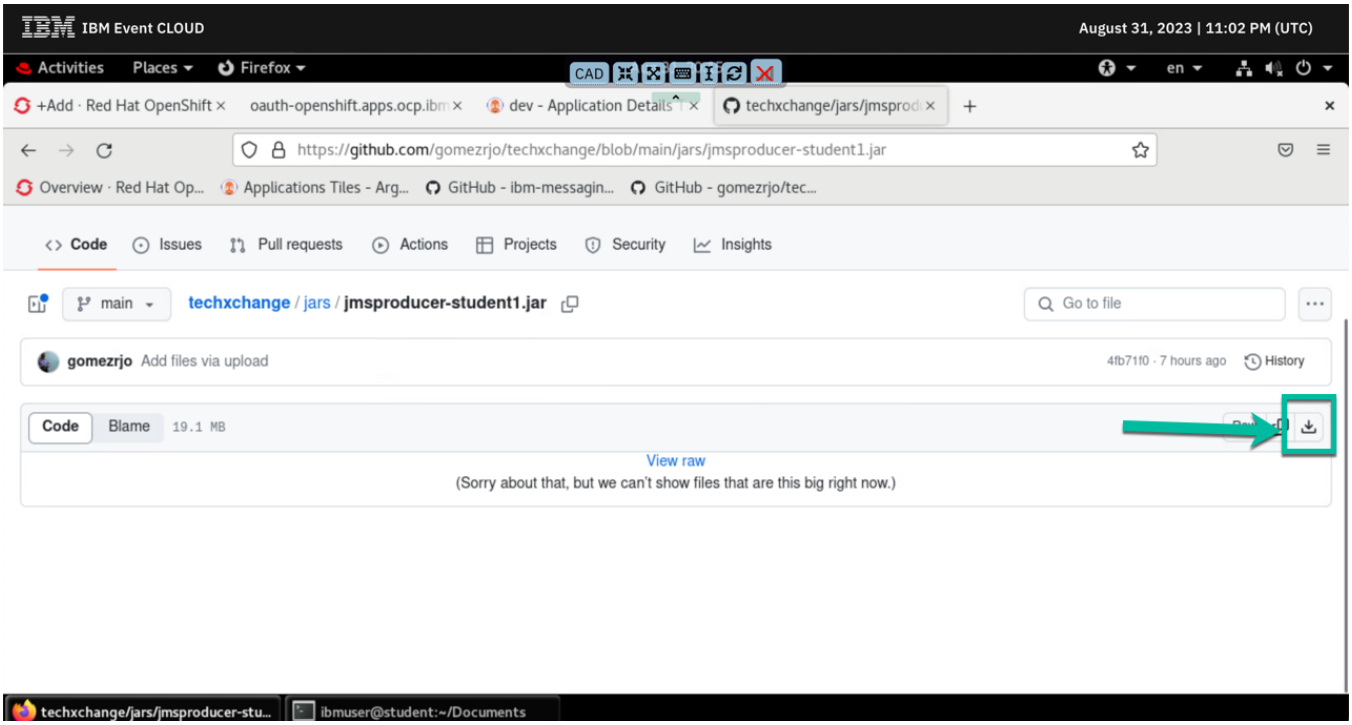
For demo purposes we have created the application for you, and it is available in a public repo. Open a new tab in the browser and use the bookmark “GitHub – gomezrjo/TechXchange” to navigate to the repo and then click on the “jars” folder as shown below.



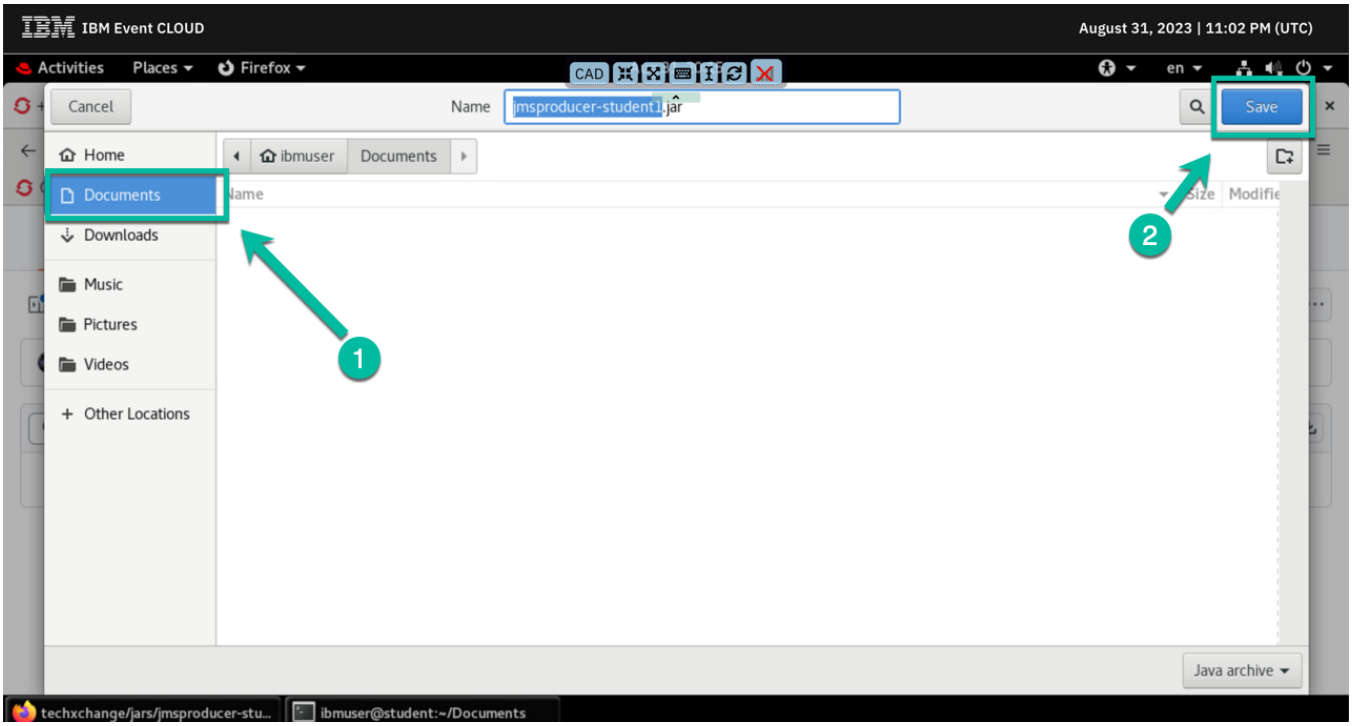
Click on the jar file associated with your student number. Each jar file is unique because is using the local nginx located in the same namespace to get the CCDT that includes the information to connect to the Uniform Cluster. If you want to see the content of the CCDT open a new tab and use the following URL: http://example-studentX.apps.ocp.ibm.edu/ccdt.json making sure you replace the X for your student number.



Then click on the “Download raw file” icon as shown below.

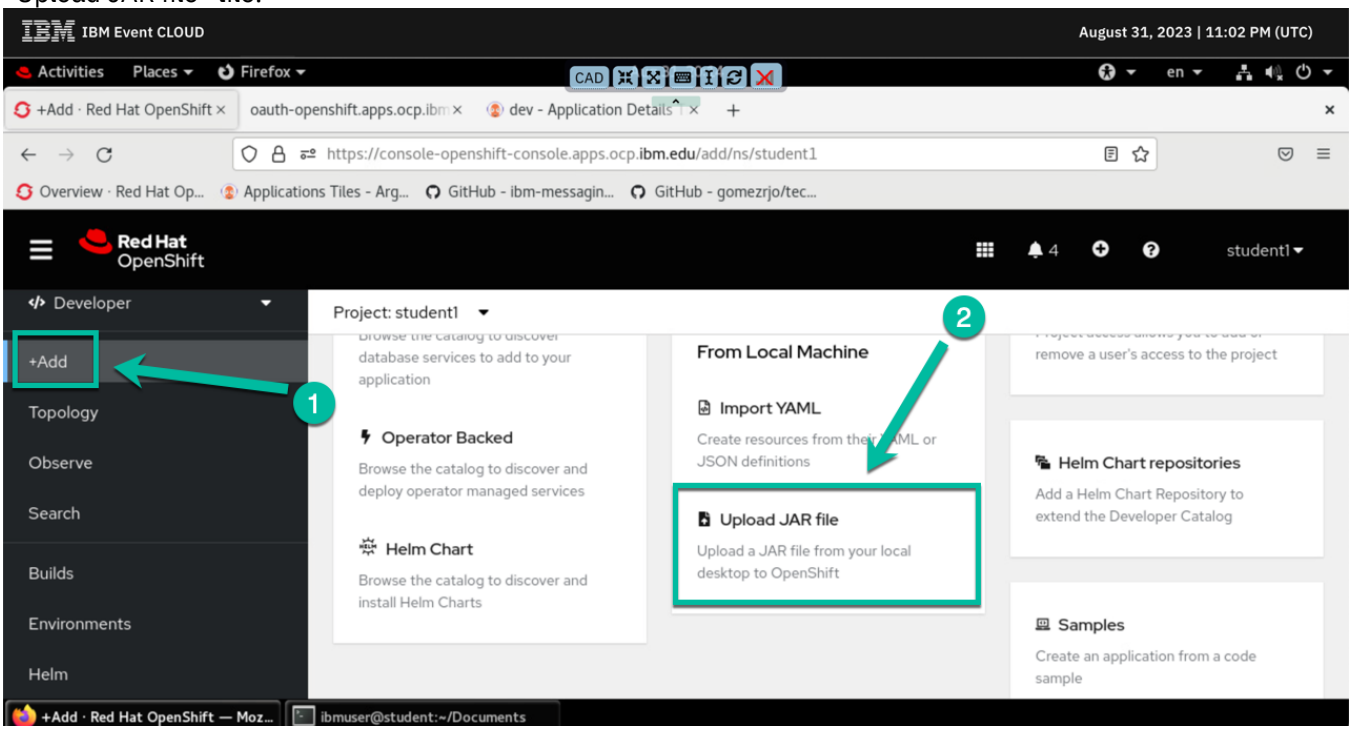


In the new dialog select “Documents” as the destination folder and then click “Save”.

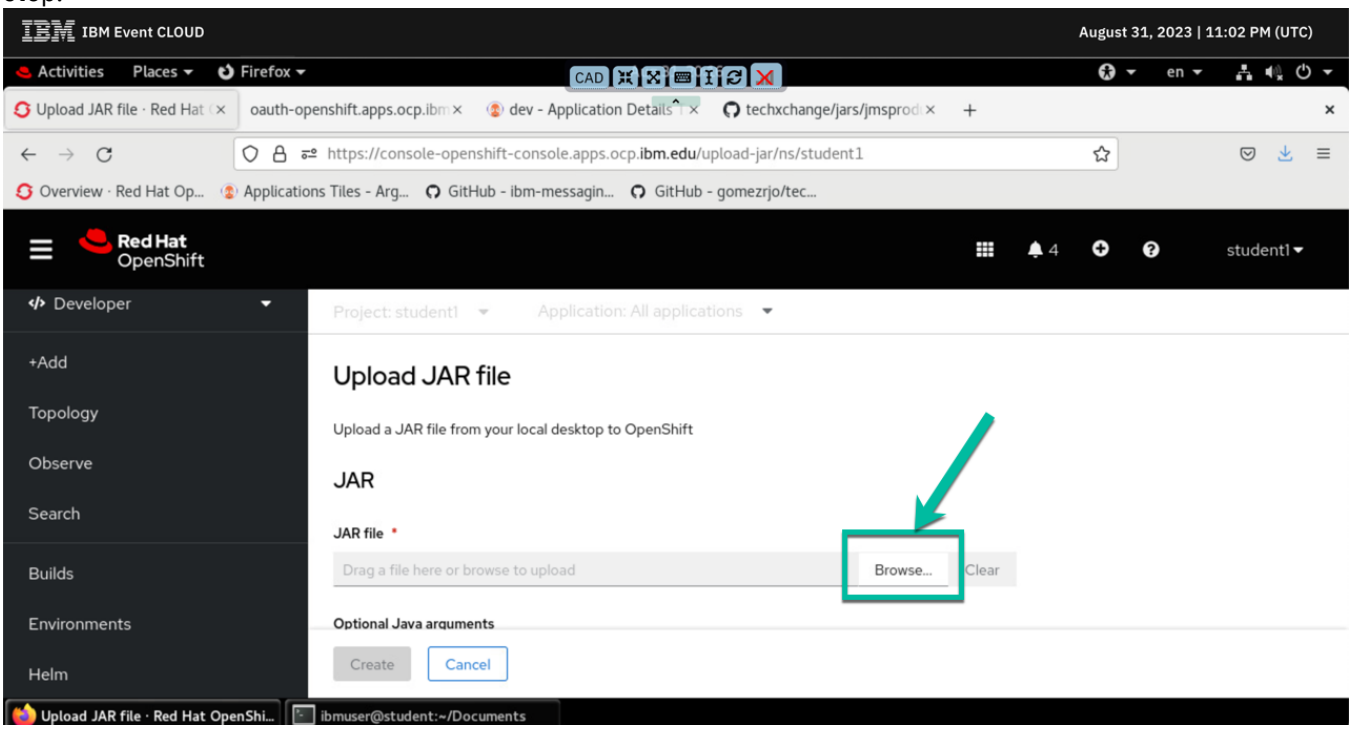


4.2 Deploy jar file

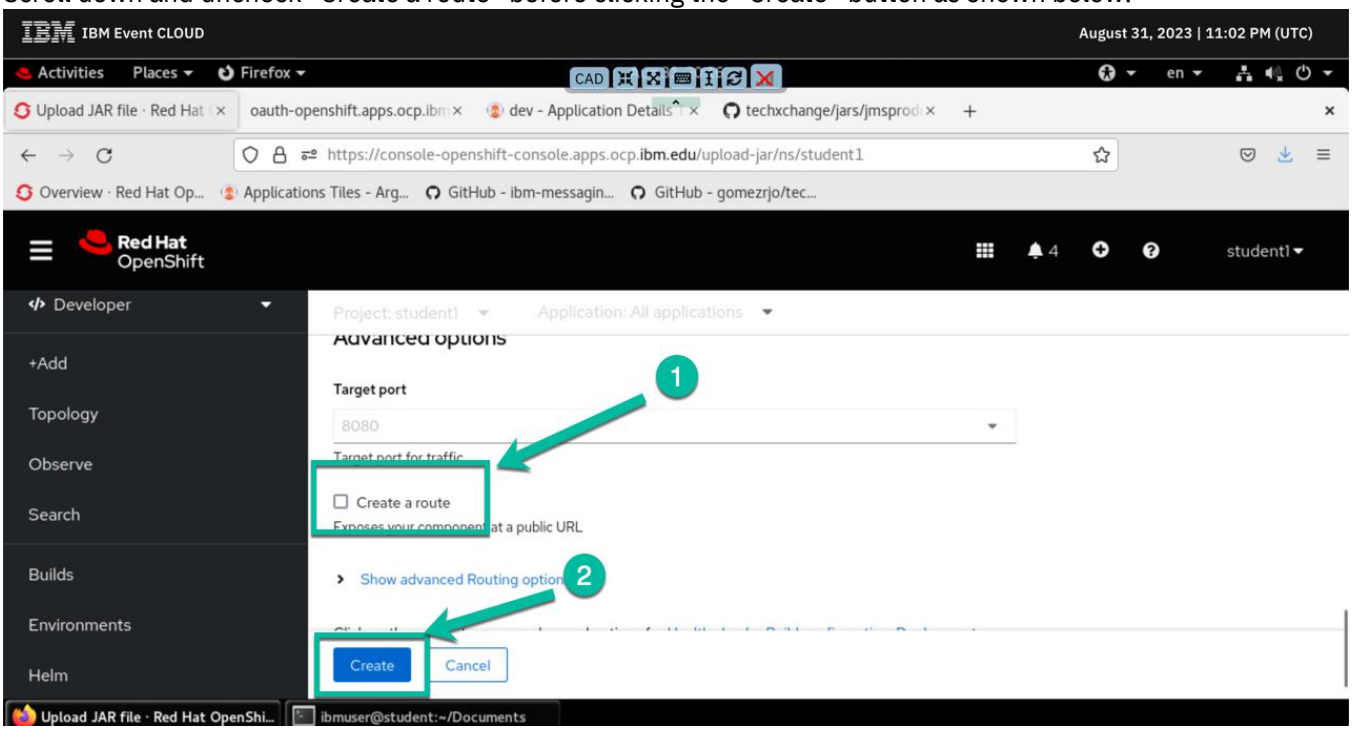
Once the jar file is in your local file system navigate to the “+Add” section in the OCP Console and select the “Upload JAR file” tile.



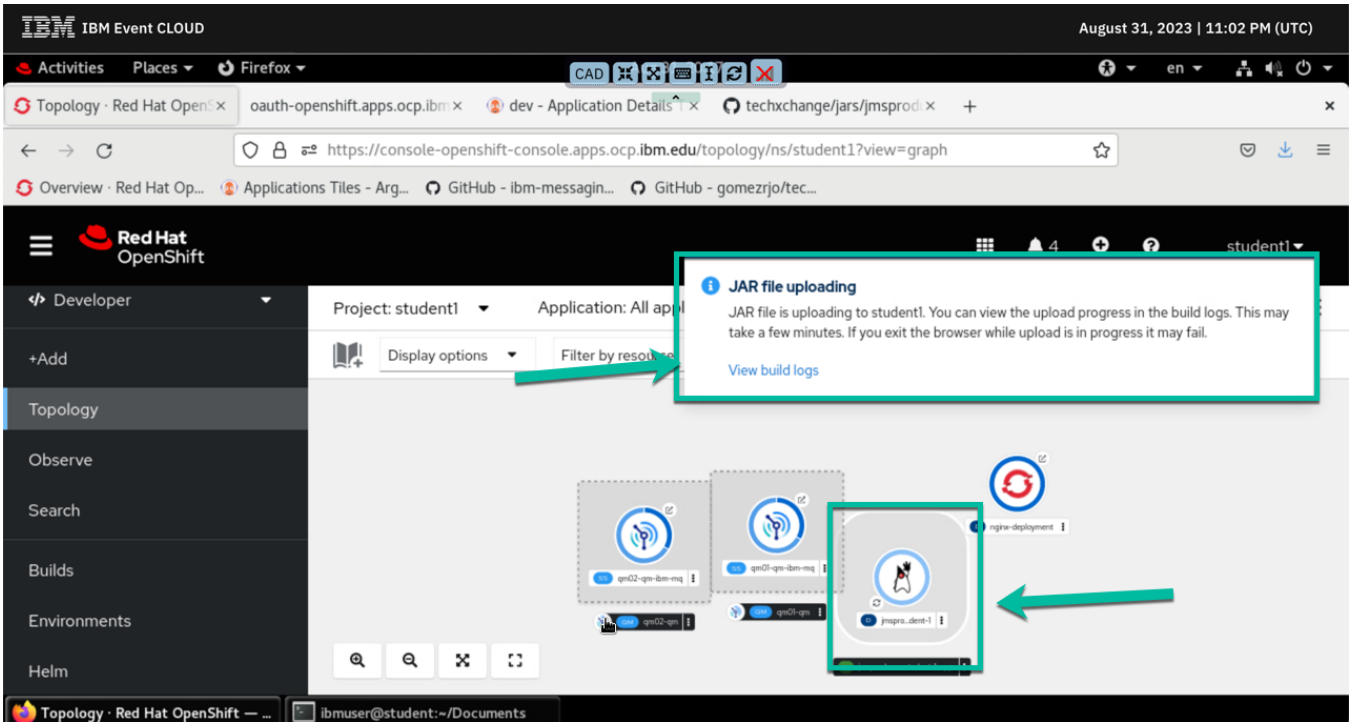
In the new window select “Browse” and follow the dialogs to select the jar file we downloaded in the previous step.



Scroll down and uncheck “Create a route” before clicking the “Create” button as shown below.

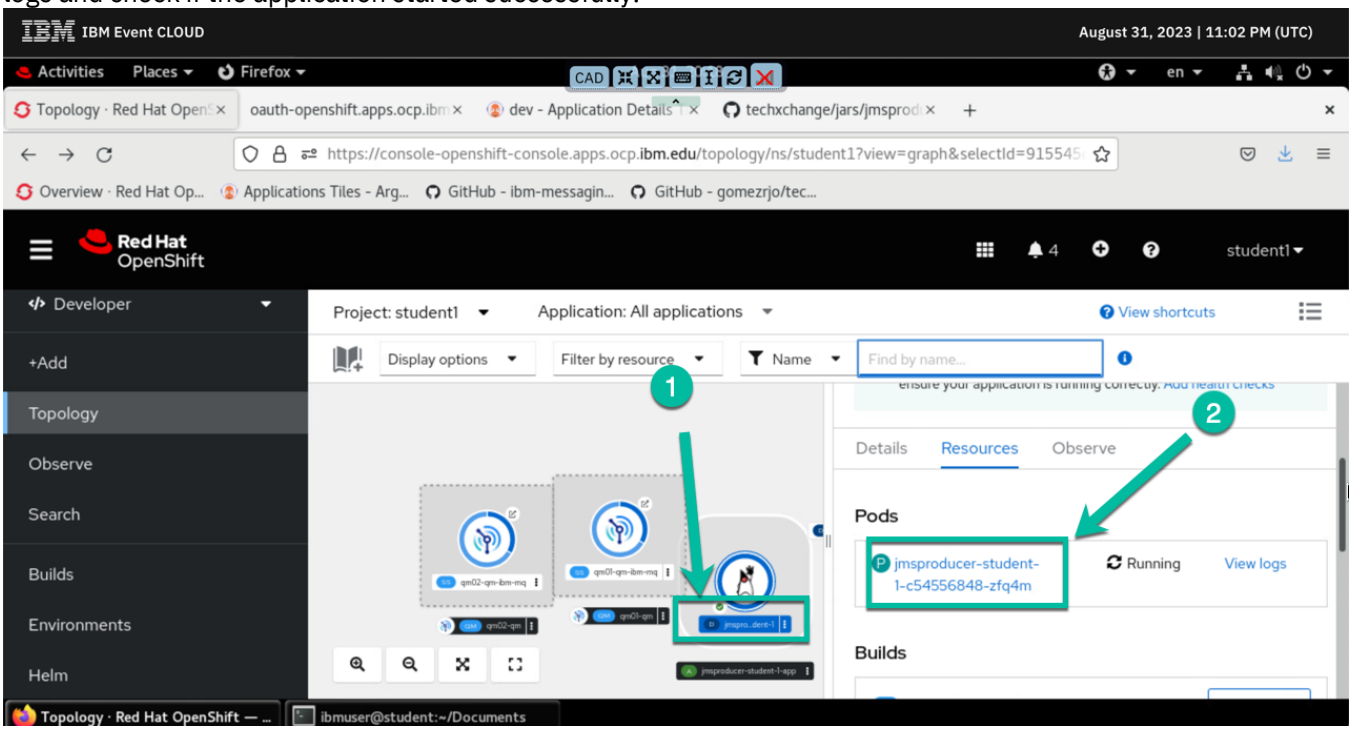


An informative message appears stating the jar file is being uploaded, and a tile representing the deployment is created as well.

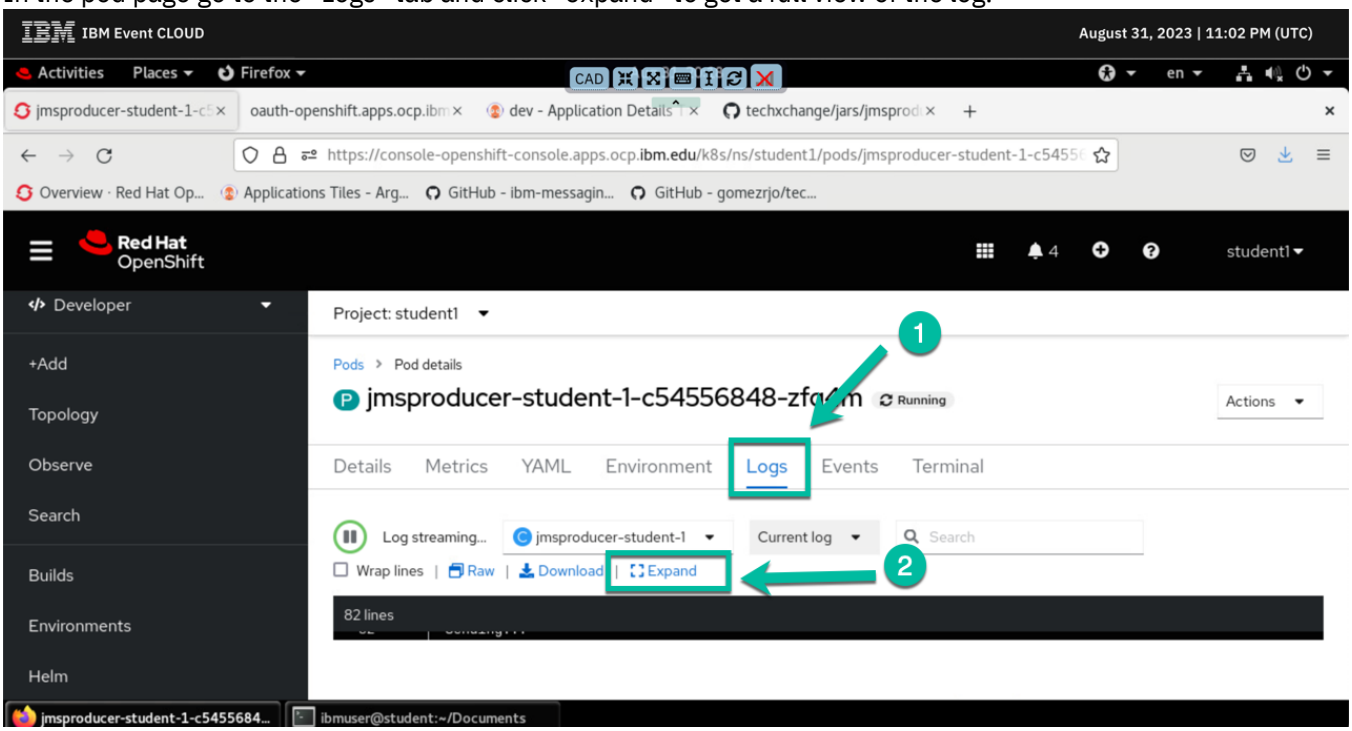


4.3 Review deployment

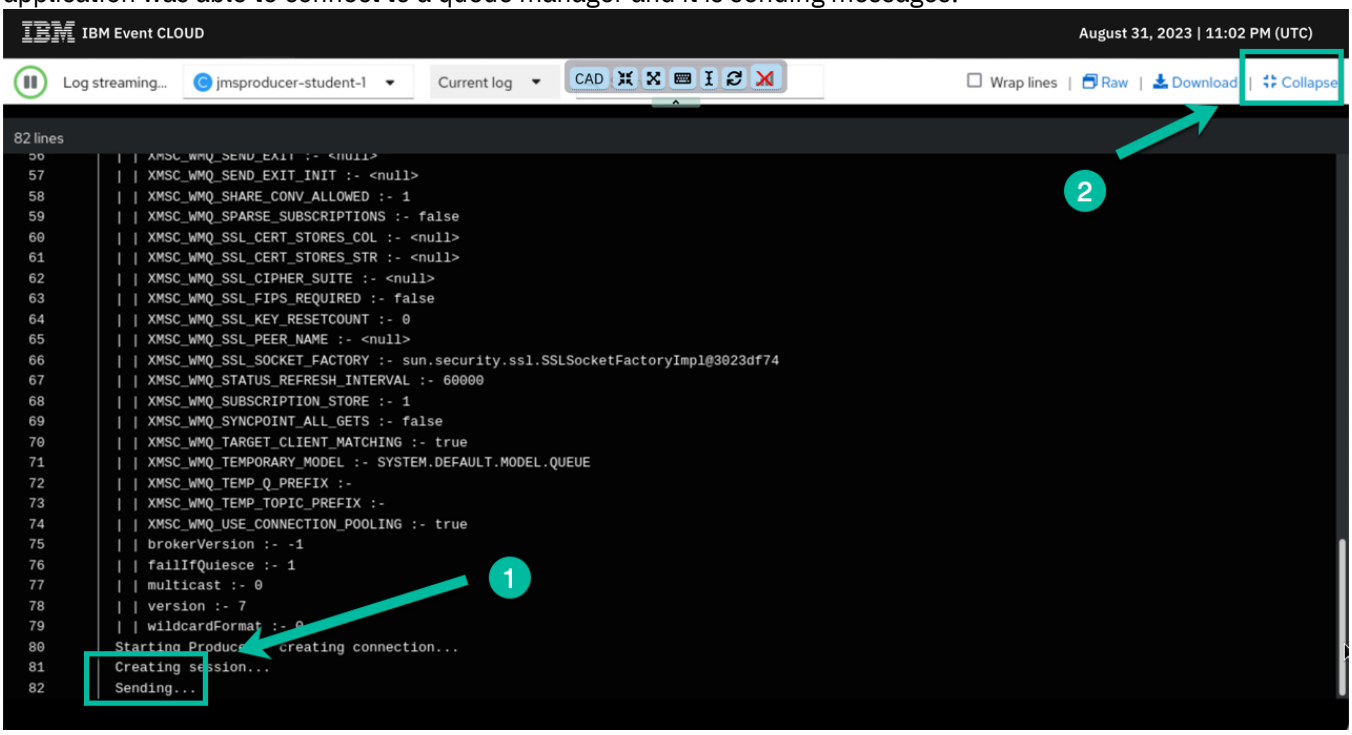
After a few second click on the tile representing the deployment to display the pod and click on it to review the logs and check if the application started successfully.



In the pod page go to the “Logs” tab and click “expand” to get a full view of the log.



Take your time reviewing the log, and at the bottom you will find a message saying “Sending...” confirming the application was able to connect to a queue manager and it is sending messages.

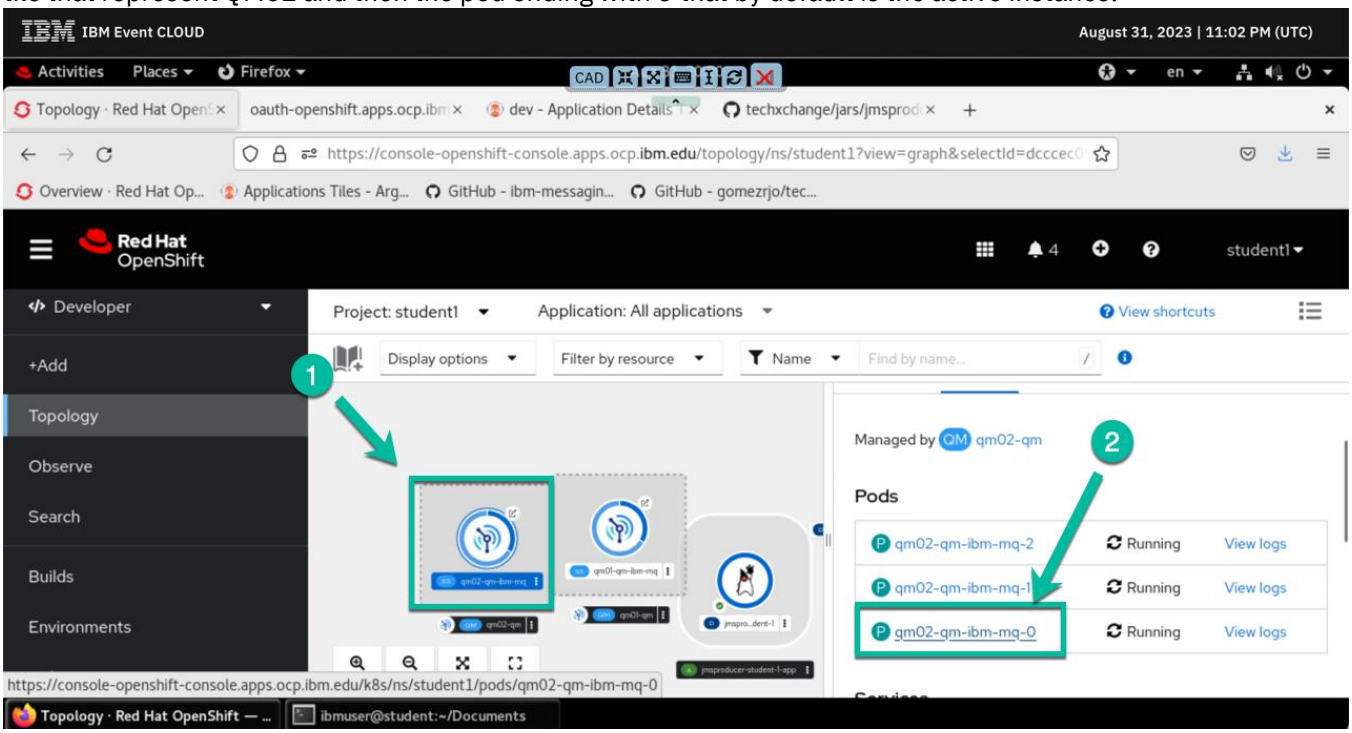


5 Validate Uniform Cluster connectivity

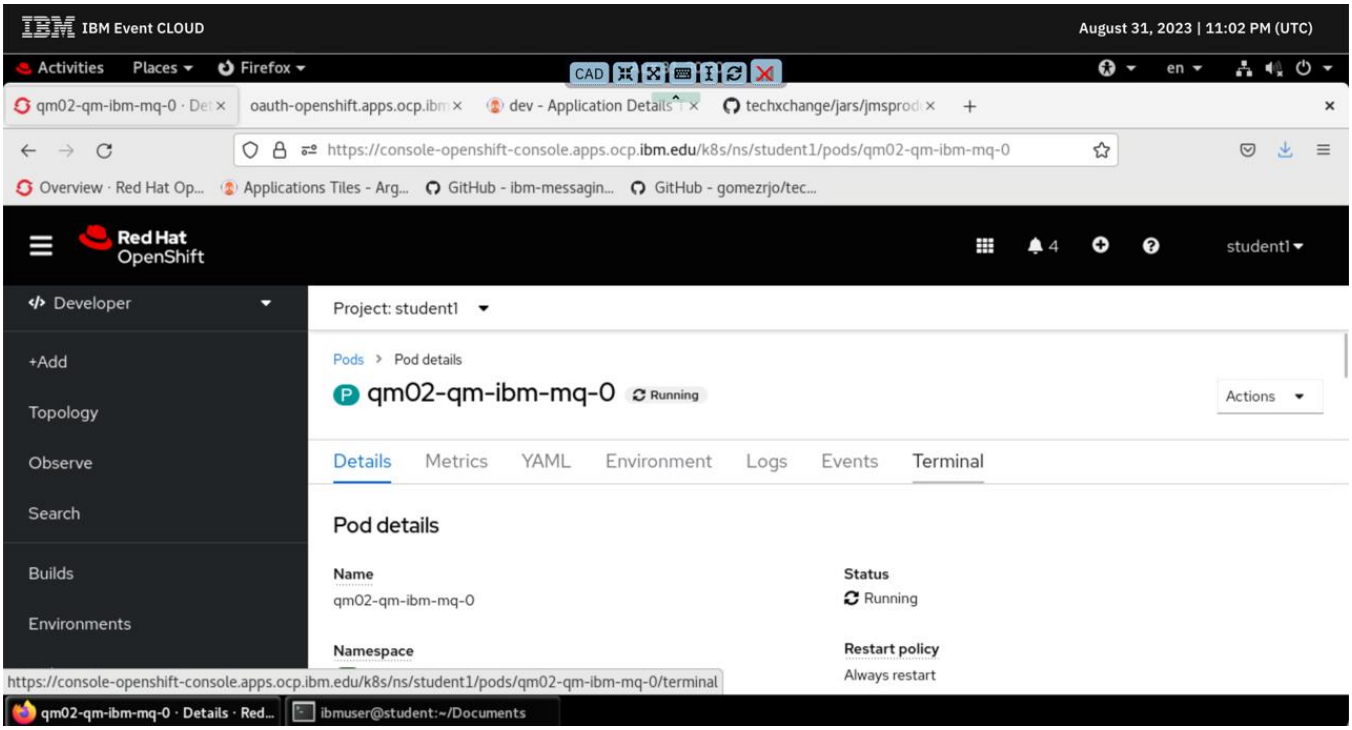
Now that the mq application is deployed let’s check the behavior with the Uniform Cluster.

5.1 Go to Queue Manager 2

Navigate back to the browser window with the Developer Perspective and in the “Topology” section select the tile that represent QM02 and then the pod ending with 0 that by default is the active instance.

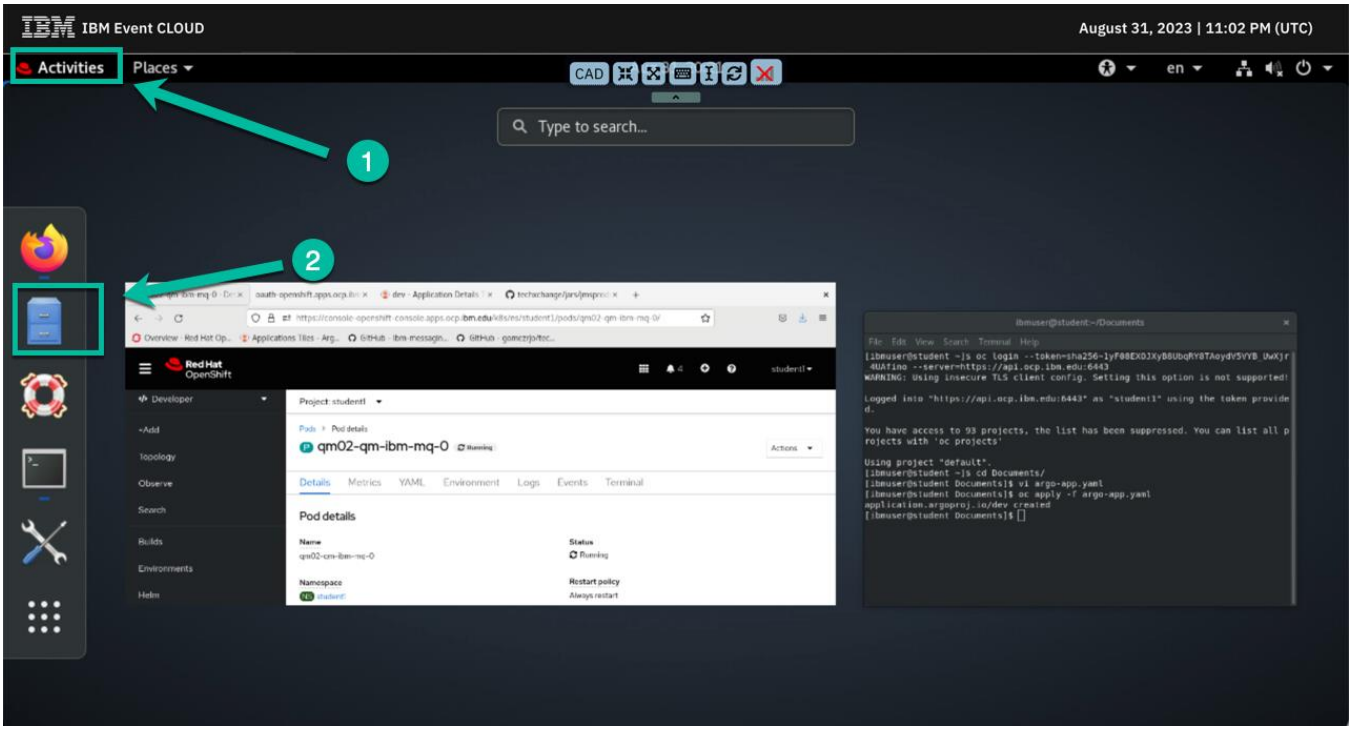


This will take you to the pod view.

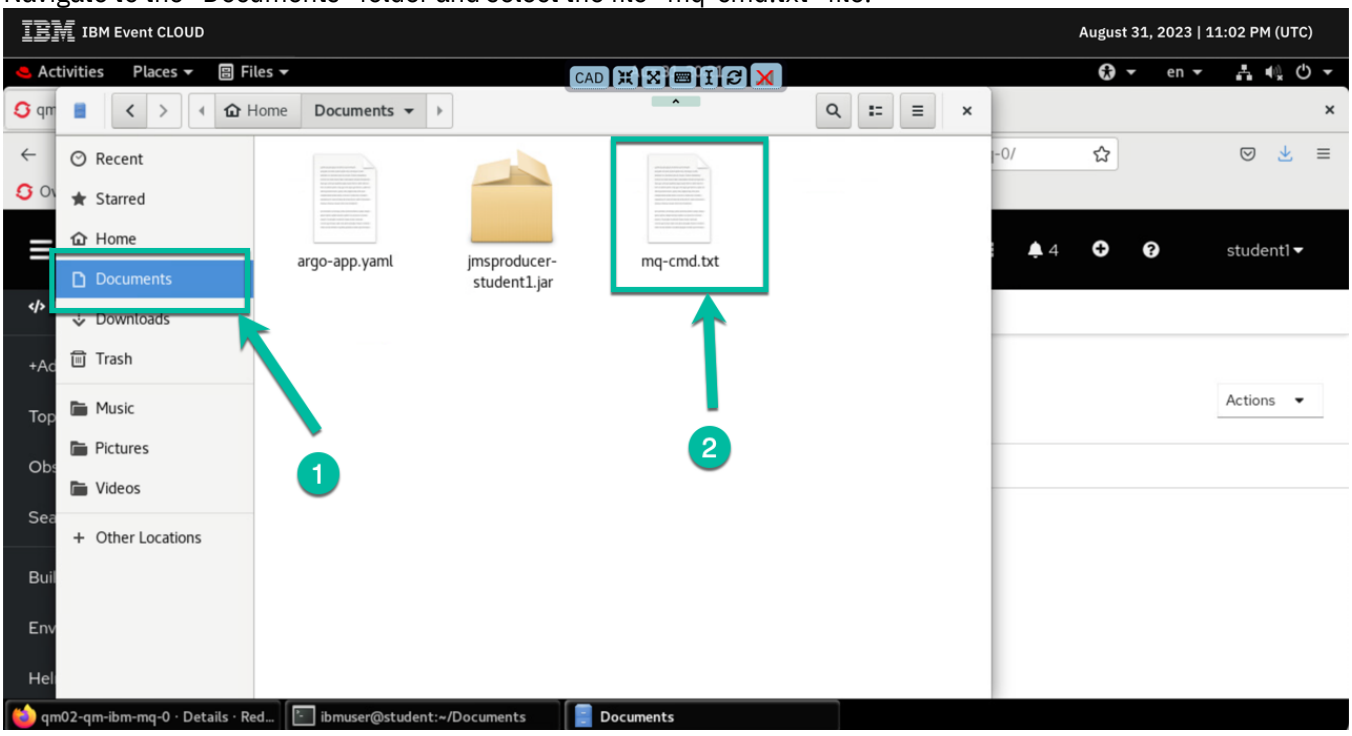


5.2 Review connections to QM02

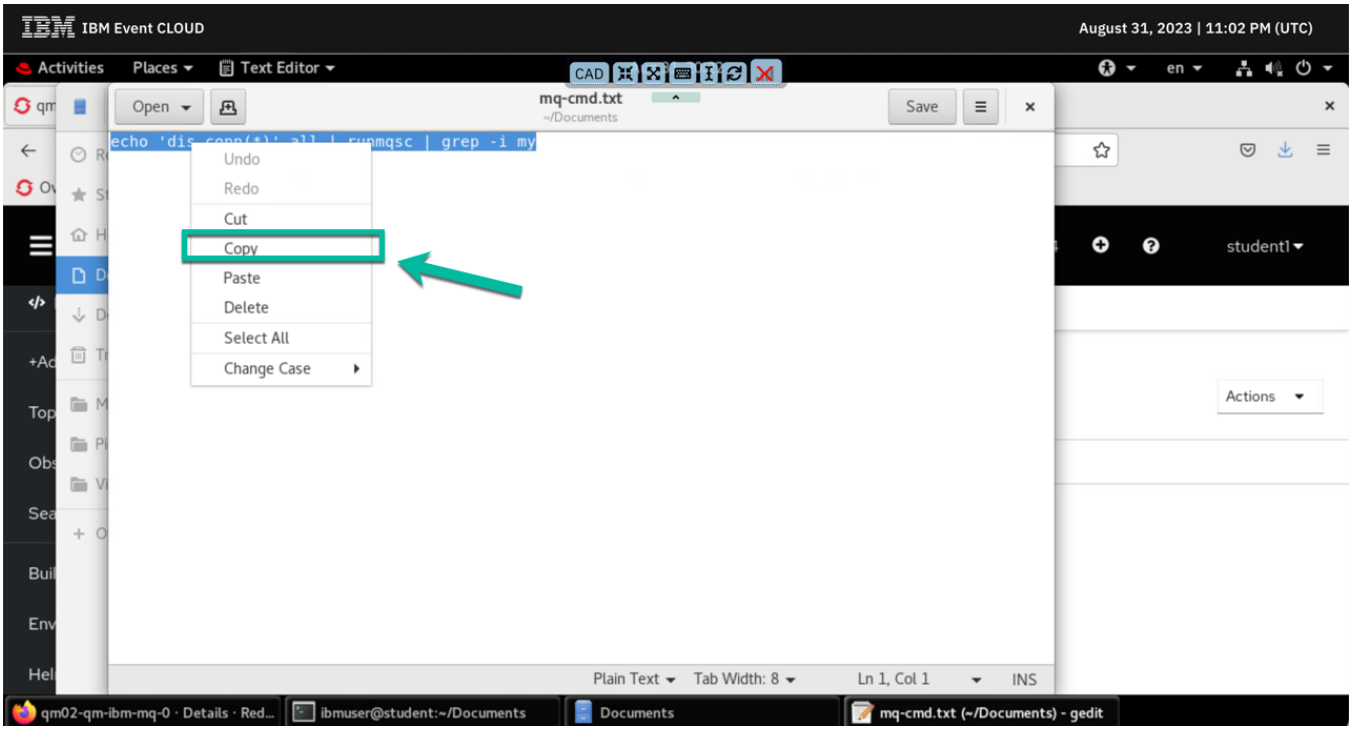
In order to check the connection status we will use the command “display connections” and we will filter by the MQ App named that is “MY-PRODUCER”. We will execute the command directly from the terminal in each MQ pod. The exact command we will use is the following: “echo 'dis conn(*)' all | runmqsc | grep -i my”. The command is not complex, but to minimize typo issues we have provided a text file with the command inside the workstation in case you want to copy and paste the command. To do so, click on the ”Activities” menu and then the “Files” icon as shown below.



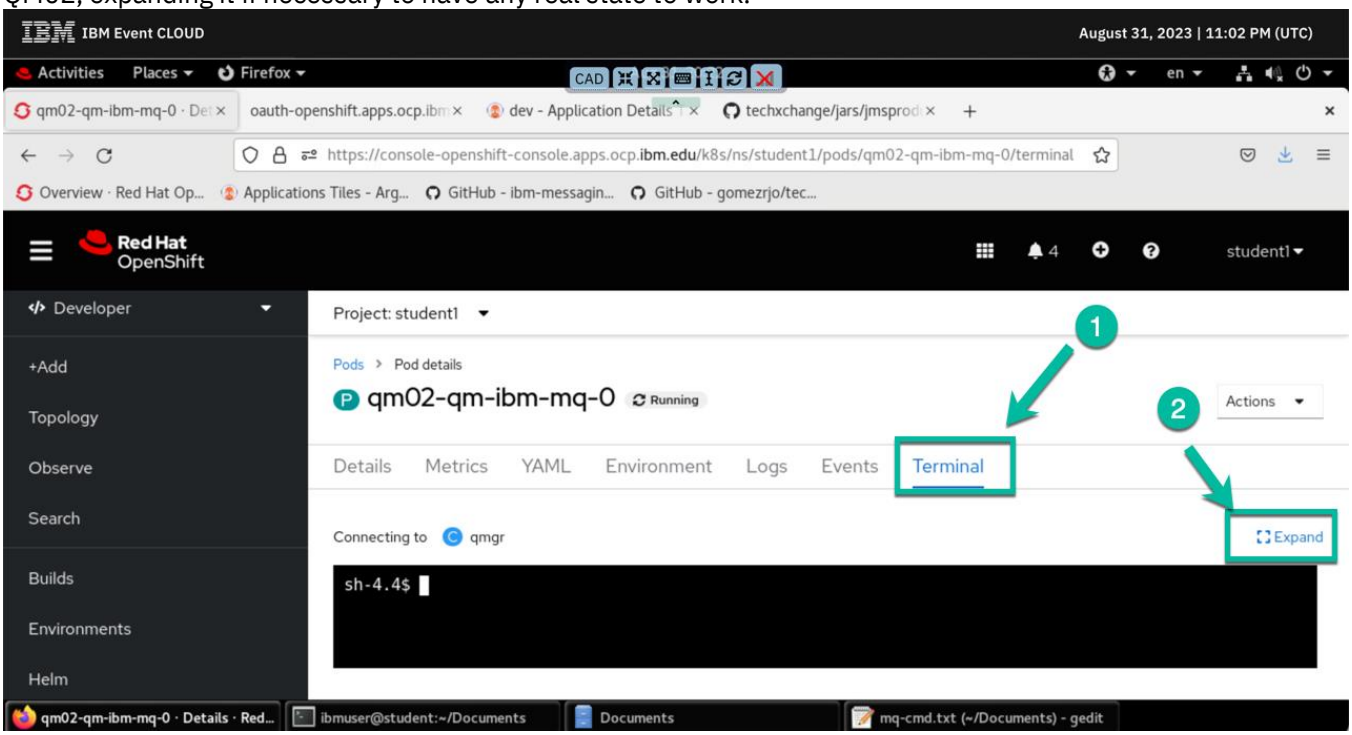
Navigate to the “Documents” folder and select the file “mq-cmd.txt” file.



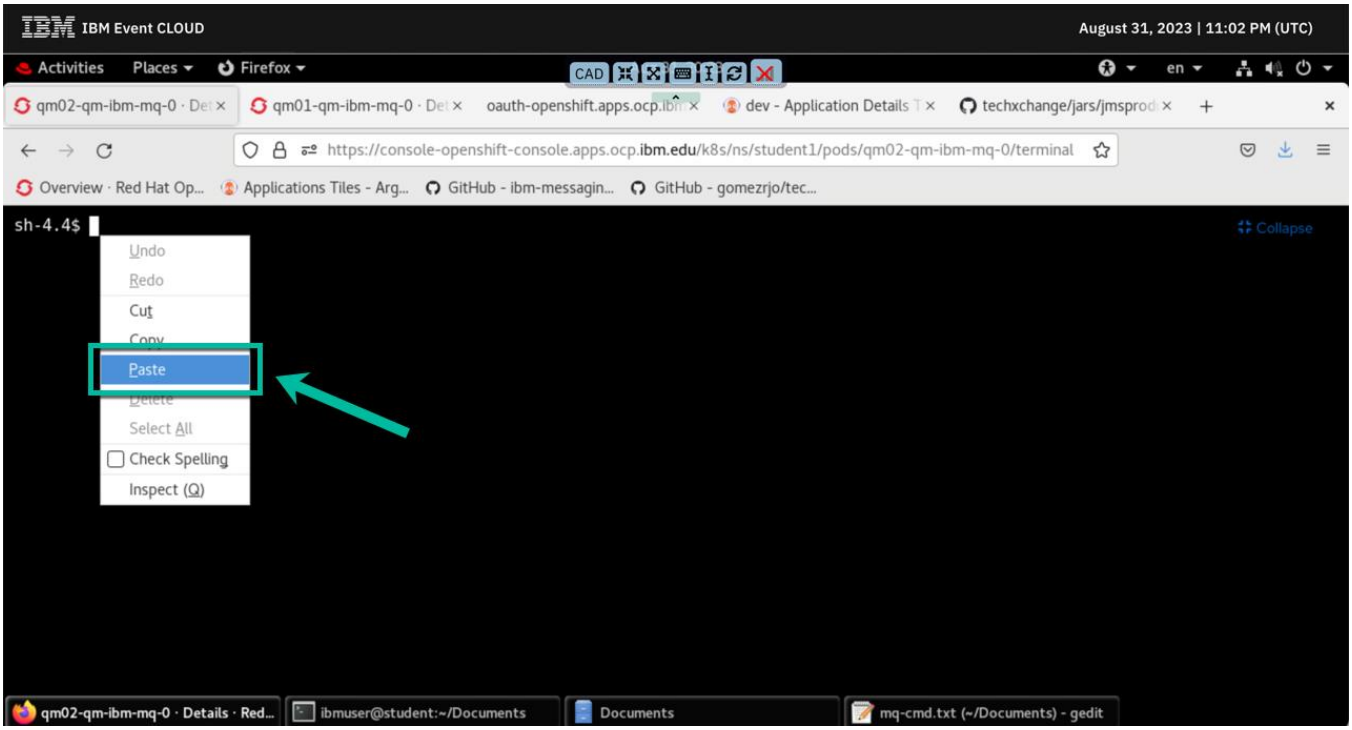
Select the content and copy it to the clipboard. Use the context menu for better results as shown below.



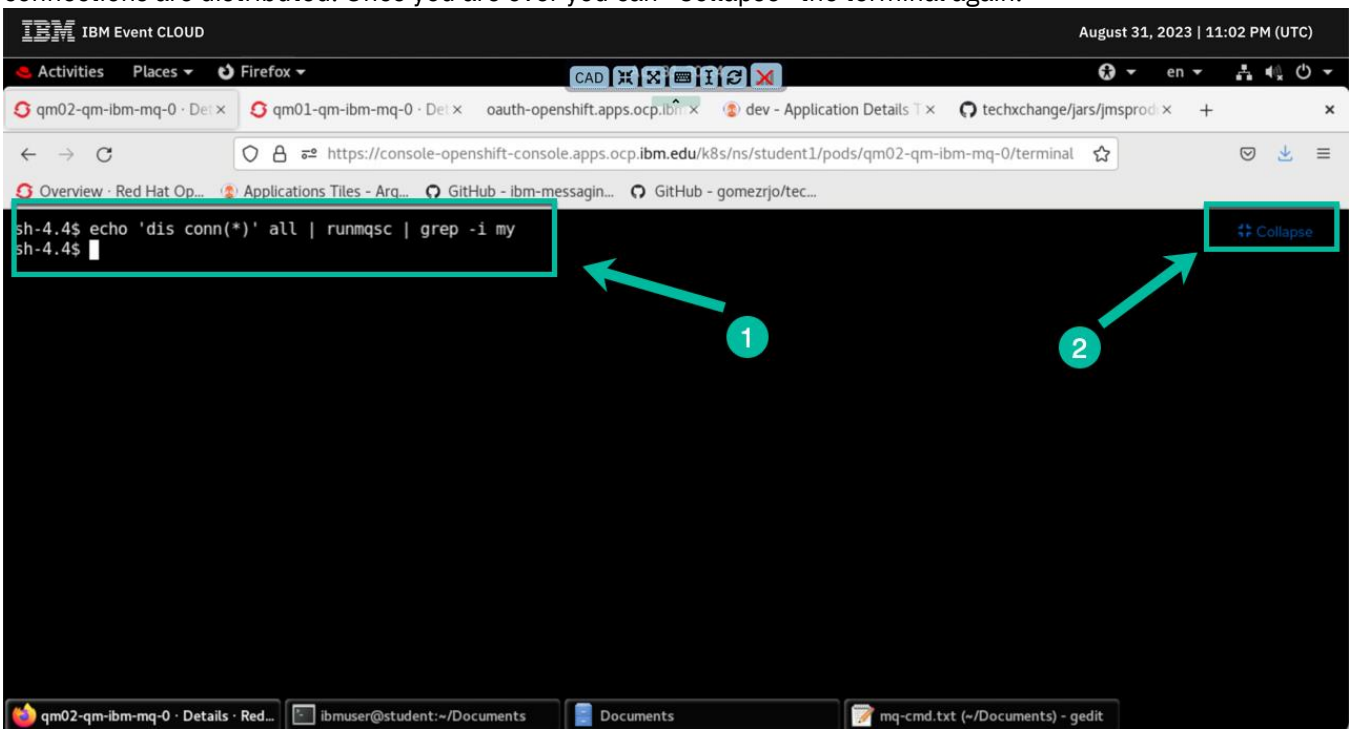
Now that we have the command, go back to the OCP console and navigate to the terminal of the active pod in QM02, expanding it if necessary to have any real state to work.



Paste the command into the terminal using the context menu.

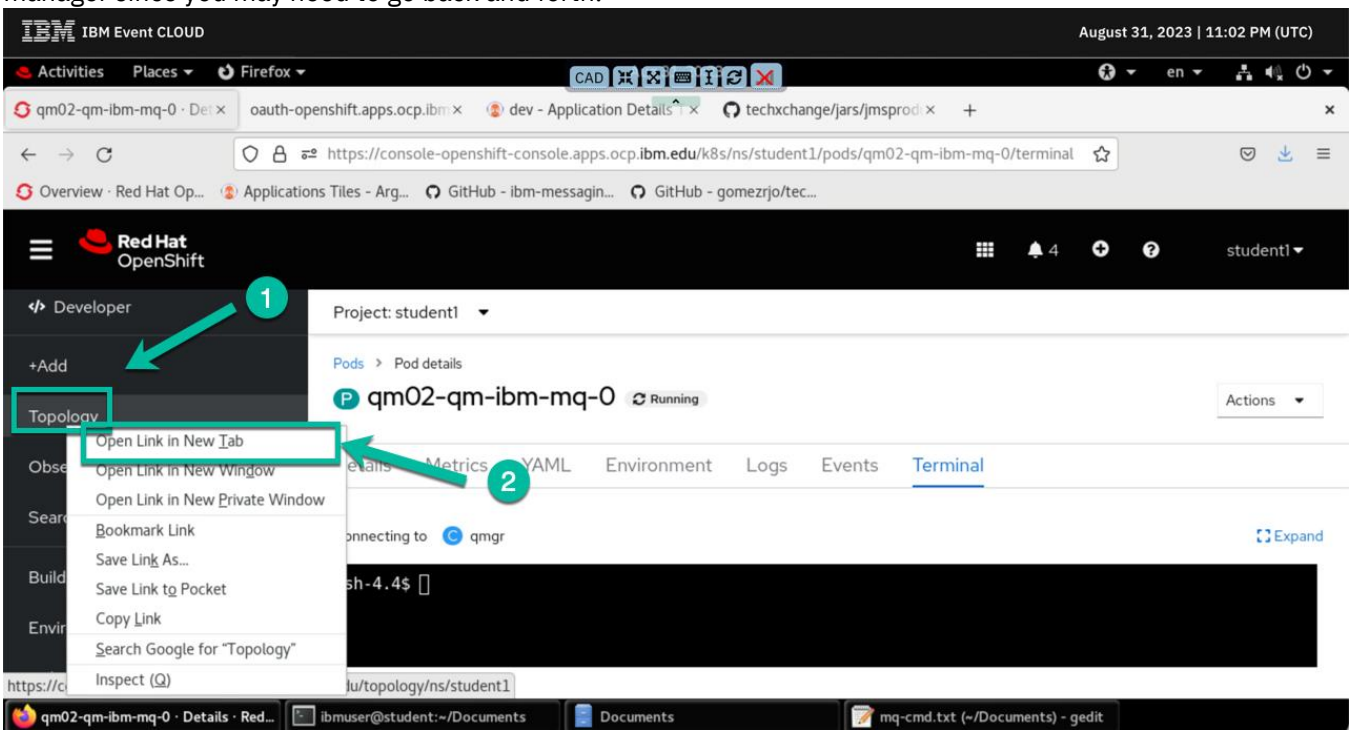


Hit <enter> to check how many active connections associated with our application are in this queue manager. Most likely you will have none as shown below, but in the next step we will get a better picture on how the connections are distributed. Once you are over you can “Collapse” the terminal again.

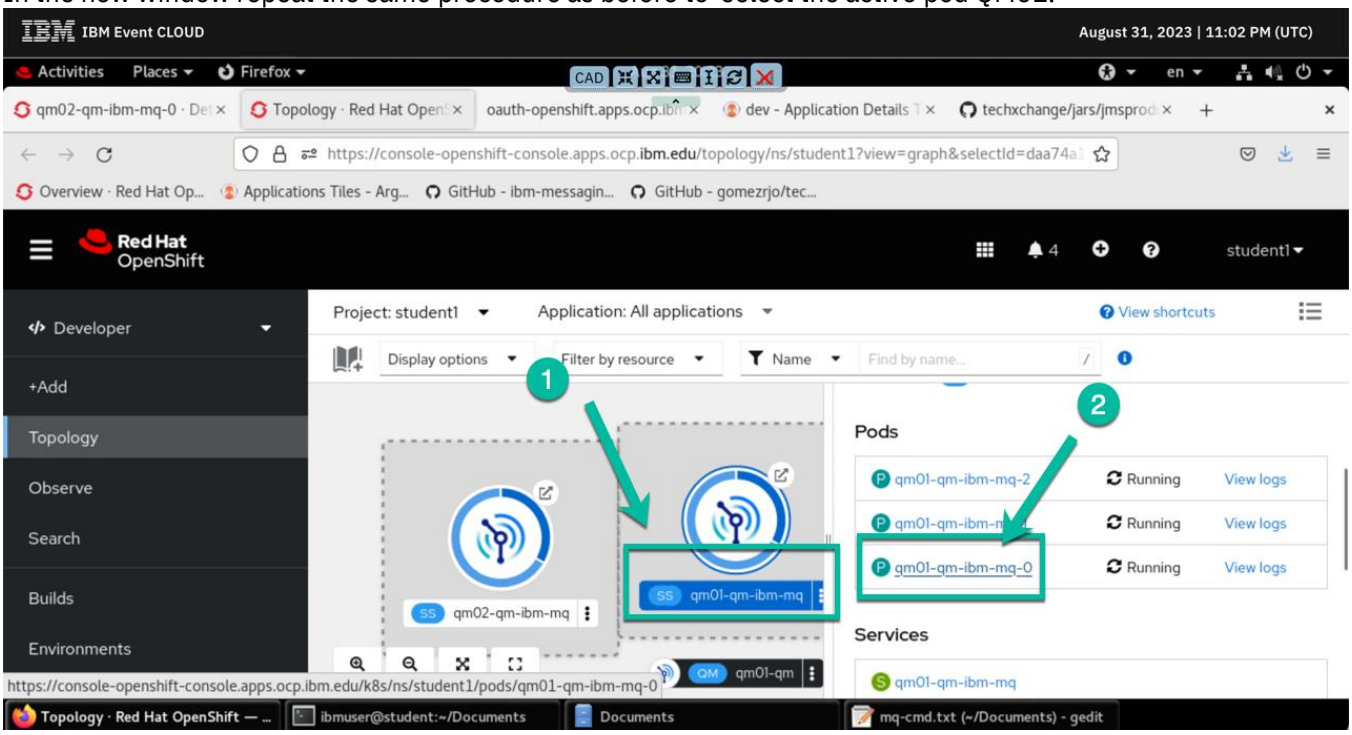


5.3 Go to Queue Manager 1

Right click the “Topology” menu and select “Open Link in New Tab” to keep a window open for each queue manager since you may need to go back and forth.

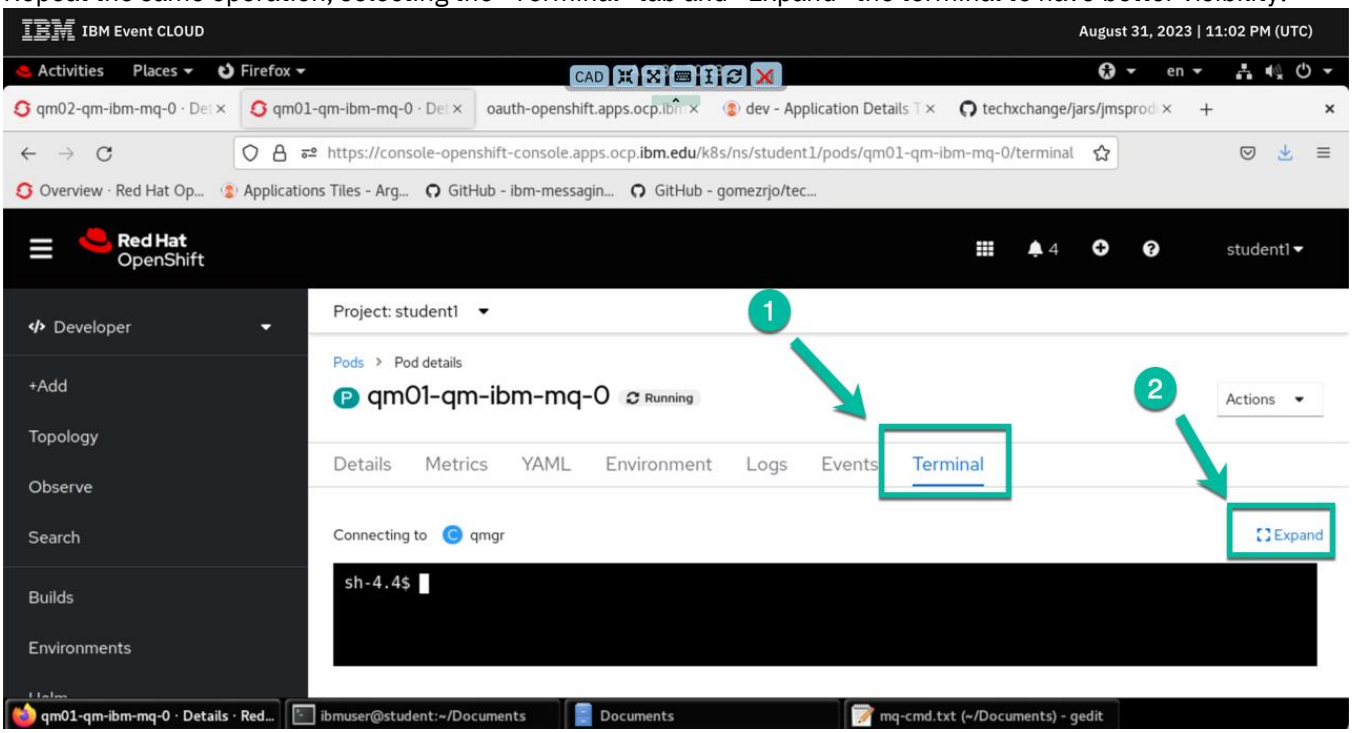


In the new window repeat the same procedure as before to select the active pod QM01.



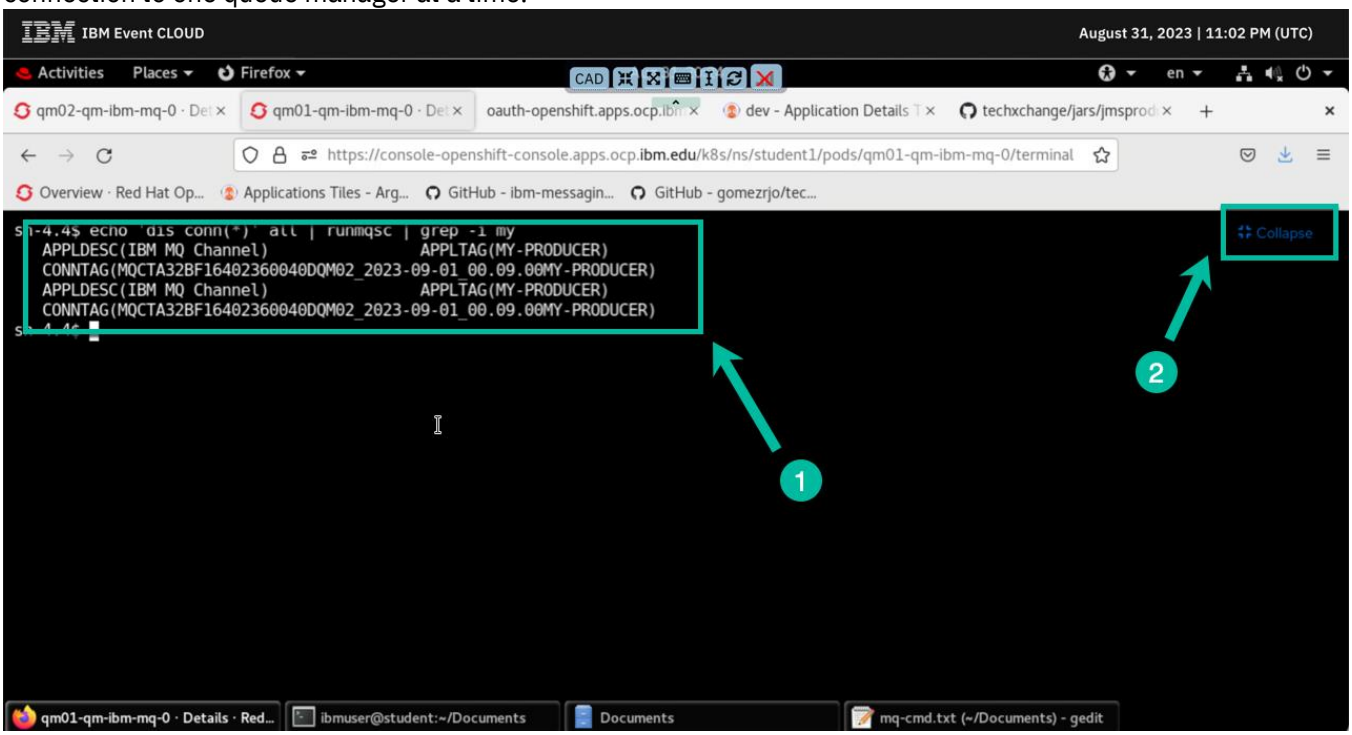
5.4 Review Connections to QM01

Repeat the same operation, selecting the “Terminal” tab and “Expand” the terminal to have better visibility.



Since we still have the command in the clipboard, paste the content again.

This time you should see a couple of active connections, probing that the application we deployed is connected to the cluster. In your case the result could potentially be the opposite since we have no affinity defined and the application will connect to any queue manager, but since we only have one instance there will only be a connection to one queue manager at a time.

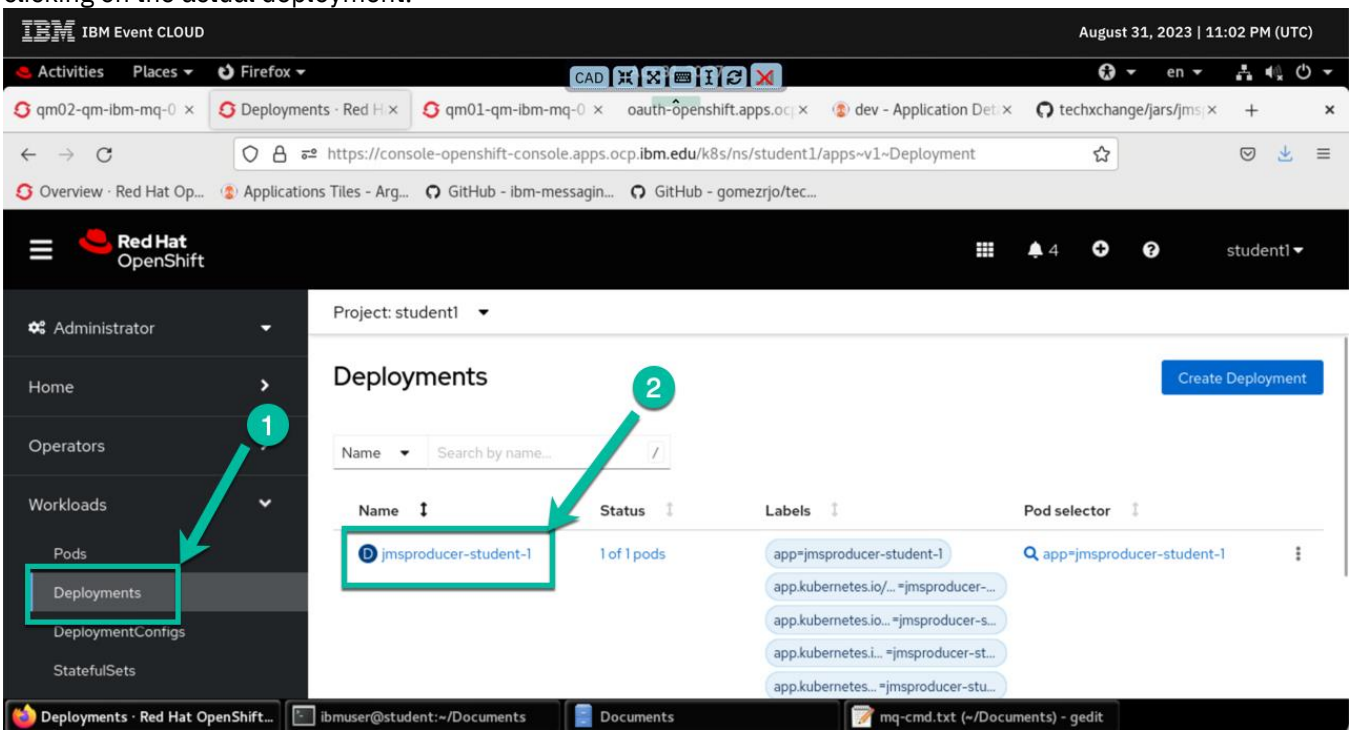


6 Scale MQ Application

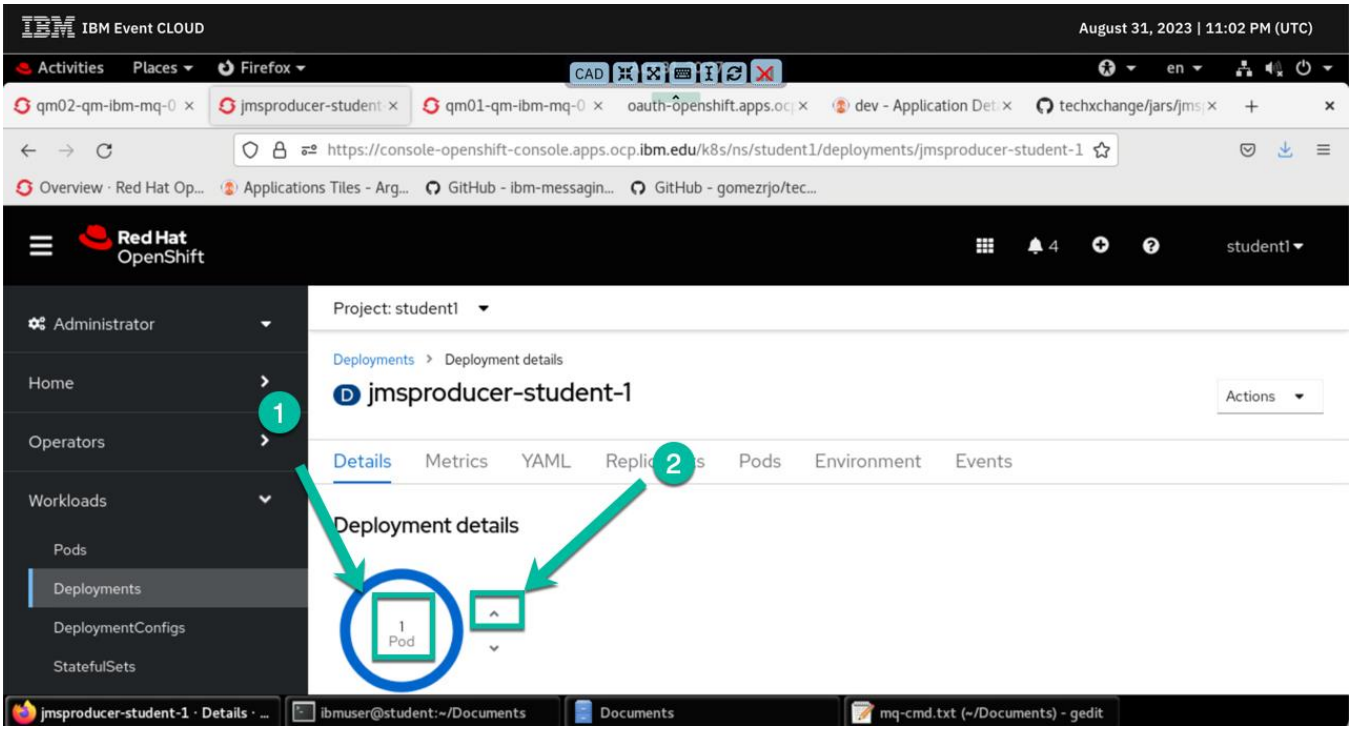
At the moment our application is running in a single pod and therefore it is only connect to one of the queue managers, but what if the workload increases and I need to scale my app. Let's simulate the scenario and see how the connections are distributed.

6.1 Increase the number of instances (aka pods)

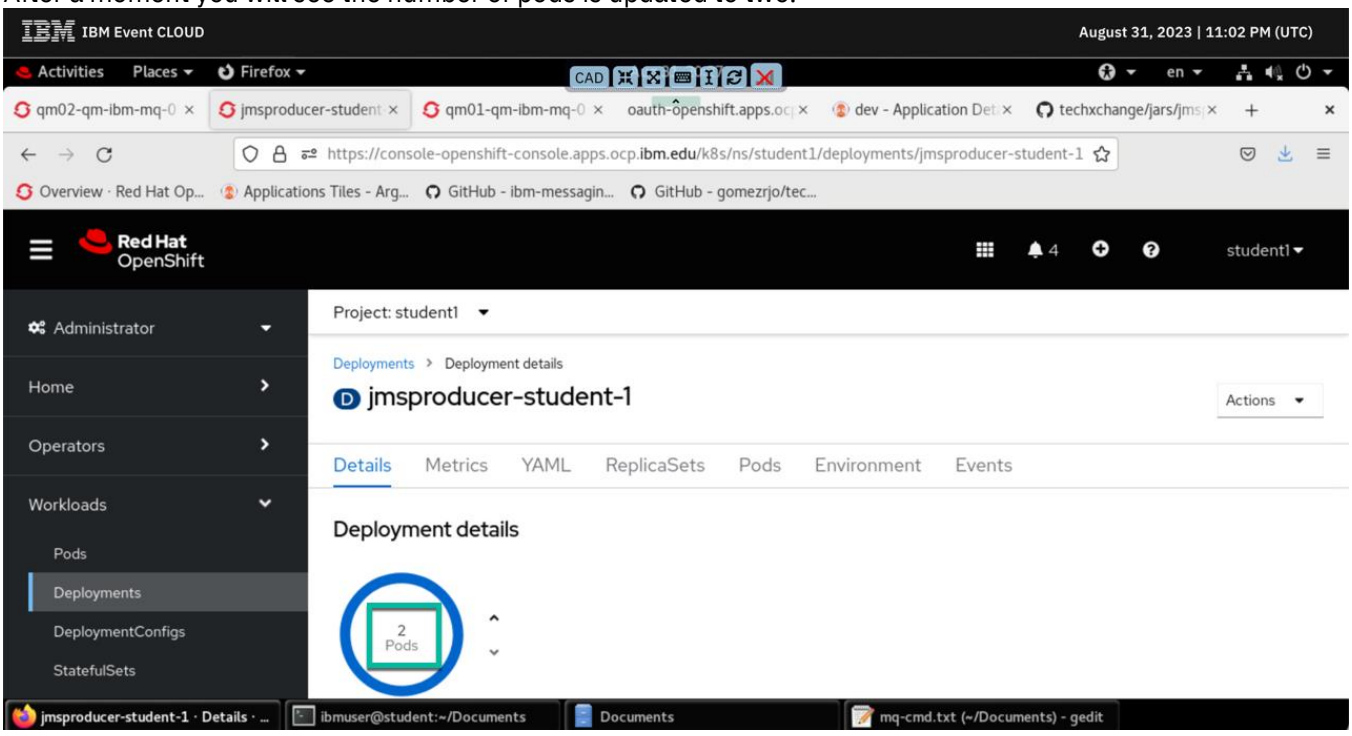
Navigate to the application deployment selecting the "Deployments" menu in the "Workloads" section and then clicking on the actual deployment.



In the deployment details you will see there is only one pod. Click on the arrow up icon to increase the number of instances to two.



After a moment you will see the number of pods is updated to two.



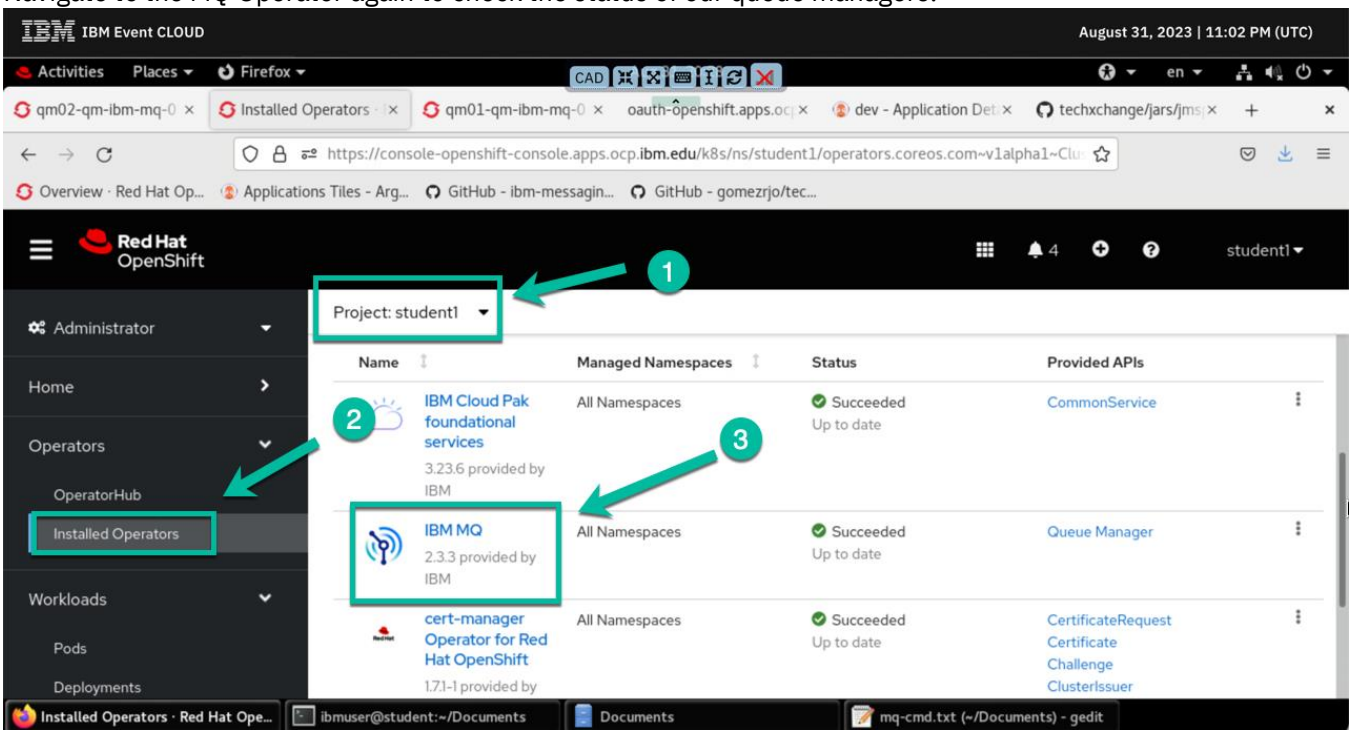
6.2 Review connectivity

Repeat the steps outlined in the previous section to check how many connections you have per queue manager. This time you should see that each queue manager has a couple of connections.

7 Rebalance connections

We observed how each instance will connect to a different queue manager trying to keep a homogenous distribution, but what would happen if one of the queue managers goes down? Let's find out.

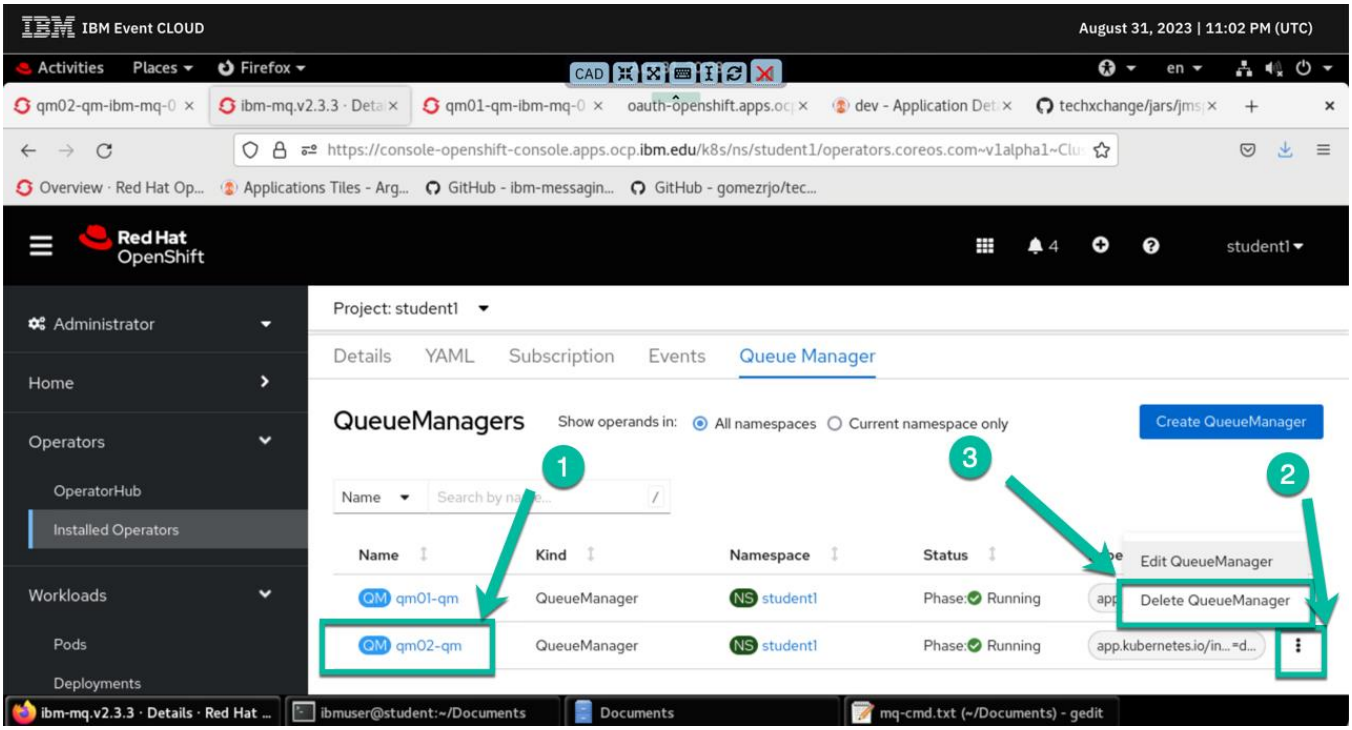
Navigate to the MQ Operator again to check the status of our queue managers.



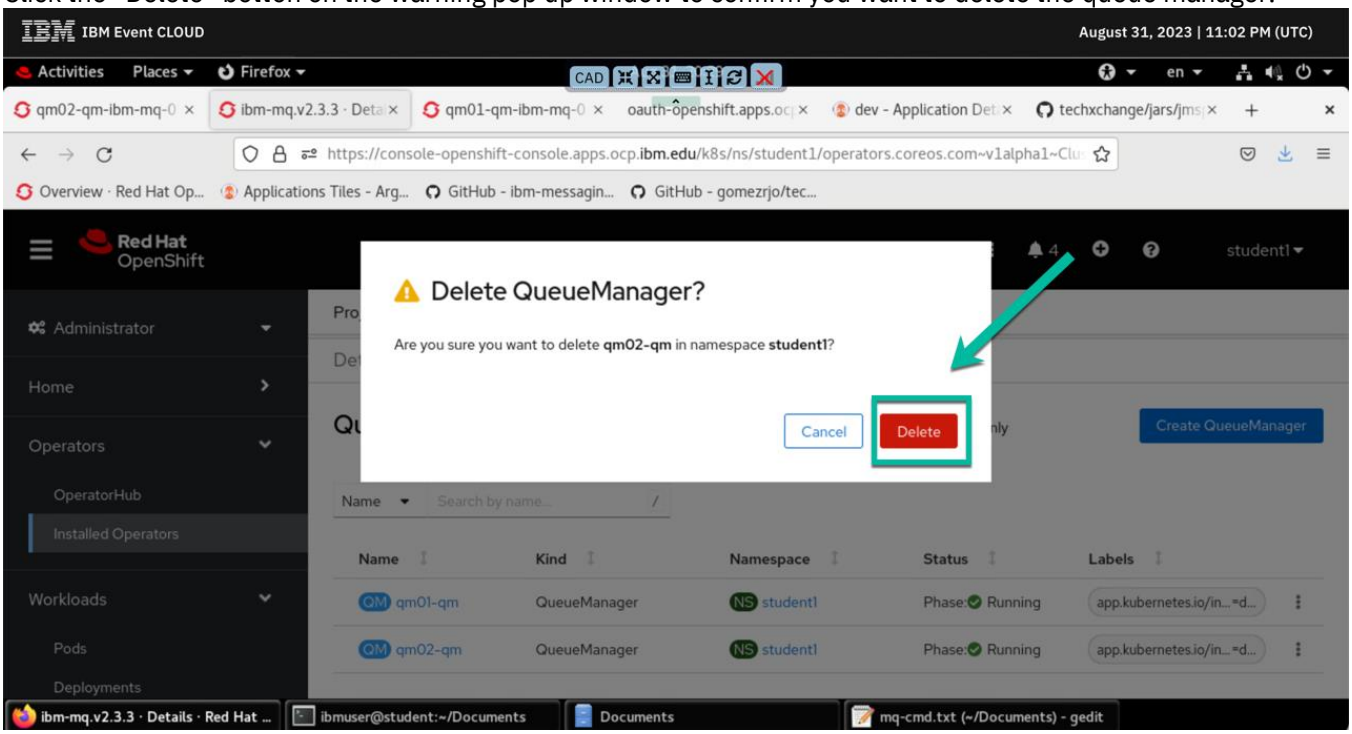
7.1 Delete Queue Manager

We could kill one of the active pods for any of the queue managers, but since we have configured Native HA, one of the standby instances will take over and at the end each queue manager will keep a couple of connections, so in this case we will go ahead and fully delete the queue manager. This will also help to close the loop on GitOps.

Click on the hamburger menu for QM02 and select "Delete Queue Manager".

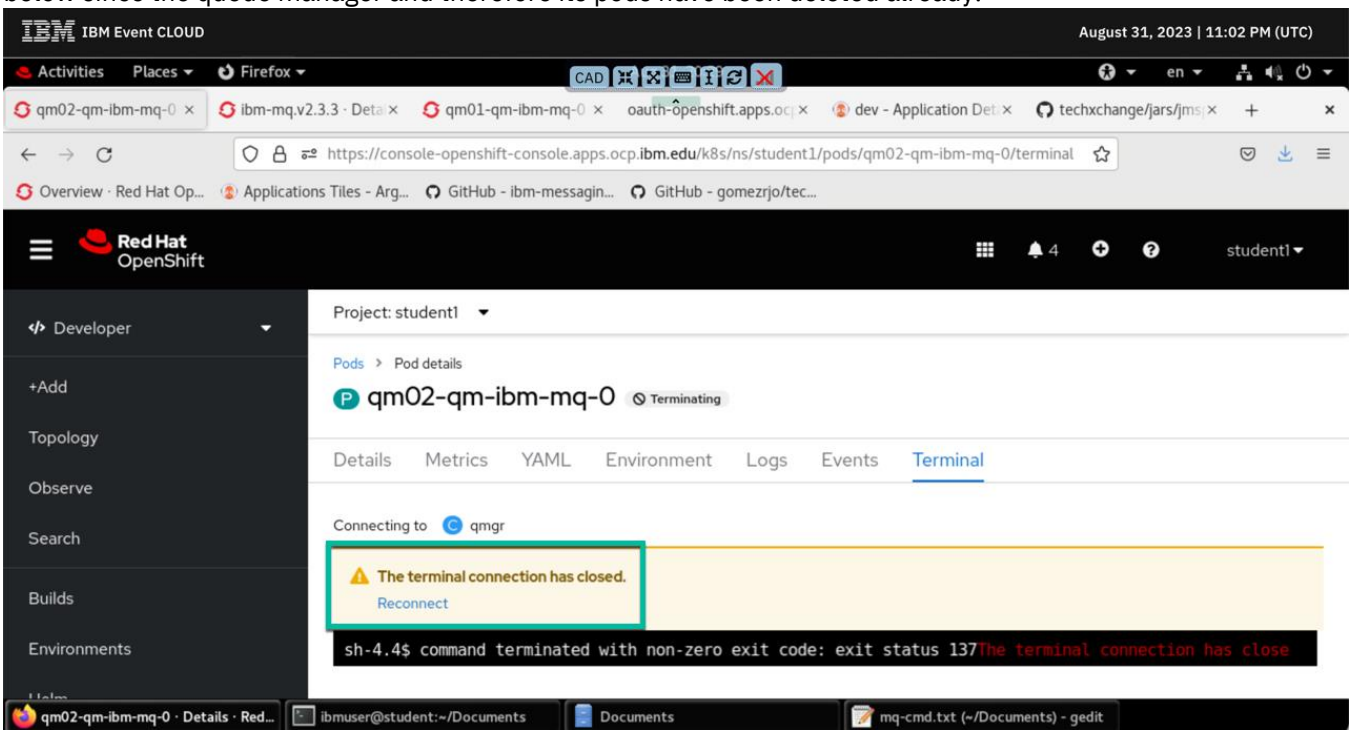


Click the “Delete” botton on the warning pop up window to confirm you want to delete the queue manager.

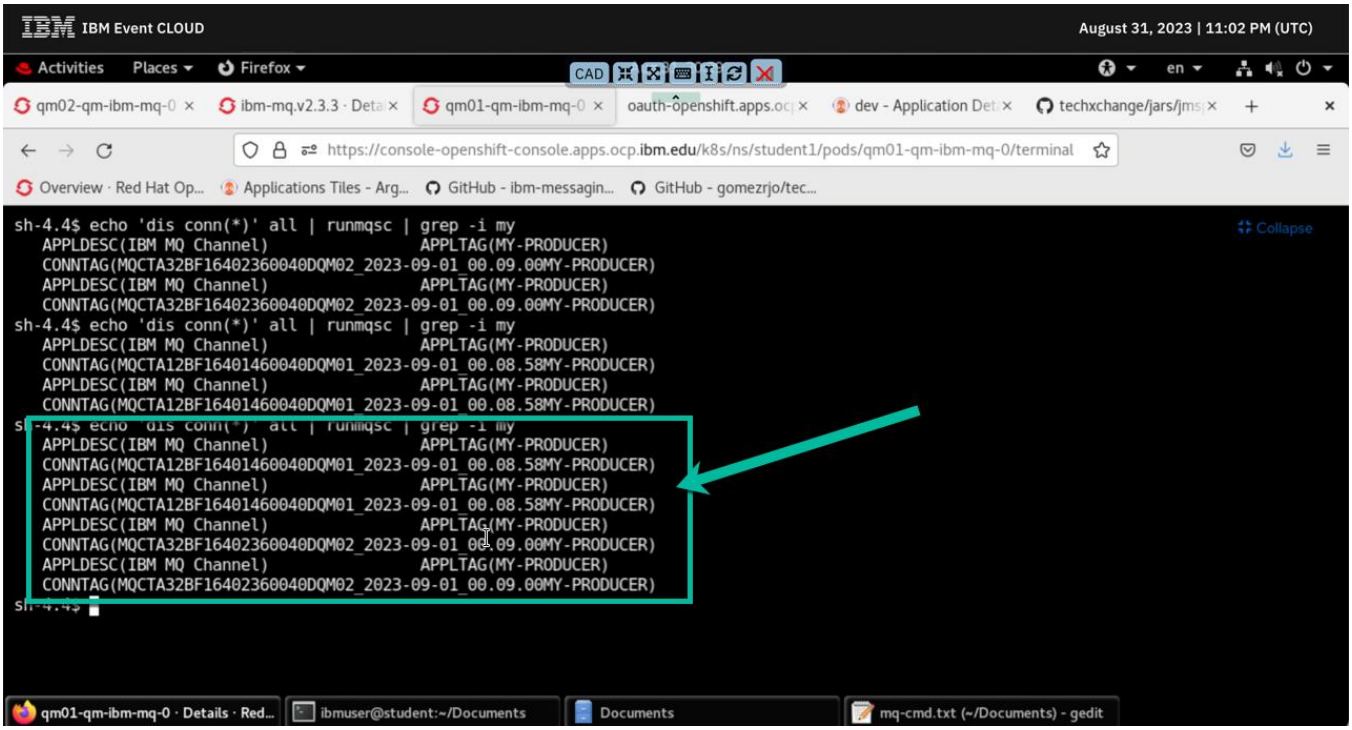


7.2 Review connectivity

If you try to navigate back to the active pod for queue manager 2 you will get a similar error as the one shown below since the queue manager and therefore its pods have been deleted already.



However, if you navigate to the active pod for queue manager one and submit the command to check the number of active connection you will see all the connections are directed to the active queue manager assuring the client application can continue sending messages.

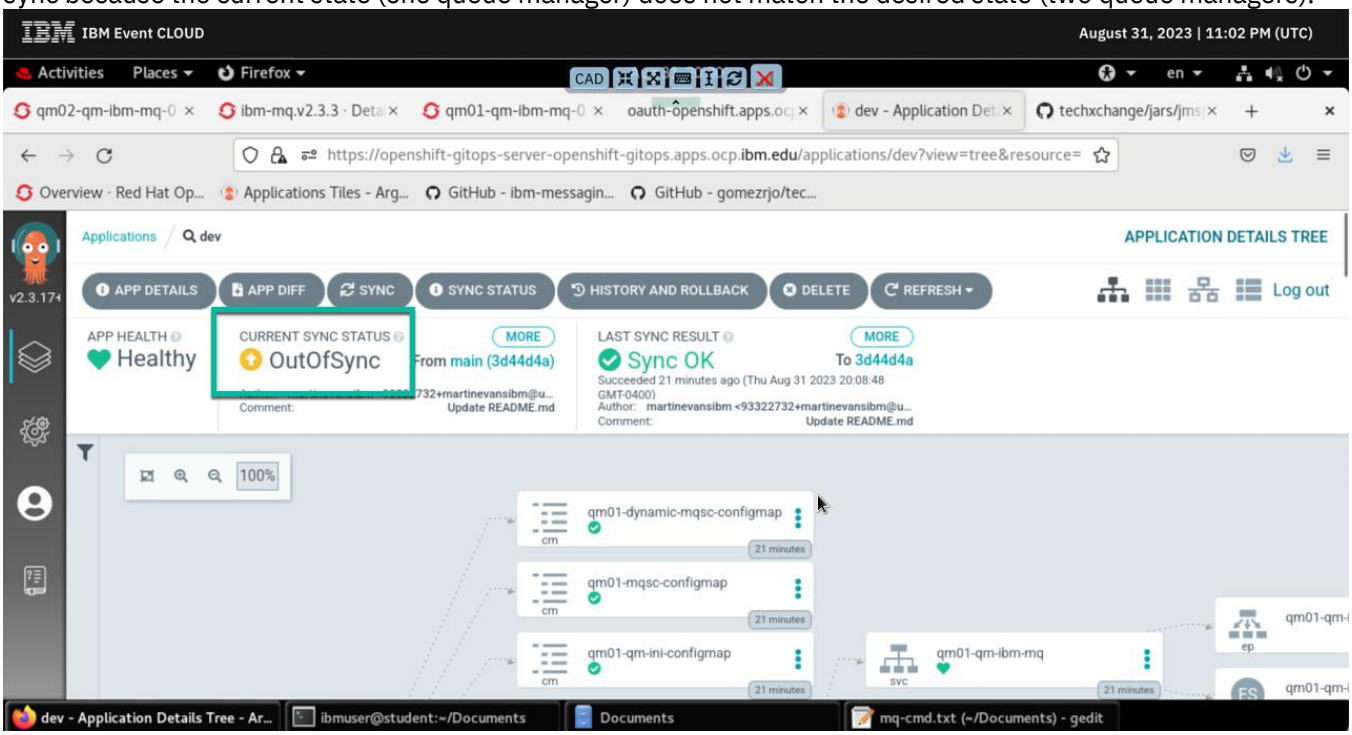

IBM Event CLOUD
August 31, 2023 | 11:02 PM (UTC)
Activities
Places
Firefox
https://console-openshift-console.apps.ocp.ibm.edu/k8s/ns/student1/pods/qm01-qm-ibm-mq-0/terminal
Collapse
sh-4.4$ echo 'dis conn(*)' all | runmqsc | grep -i my
APPLDESC(IBM MQ Channel) APPLTAG(MY-PRODUCER)
CONNTAG(MQCTA32BF16402360040DQM02_2023-09-01_00.09.00MY-PRODUCER)
APPLDESC(IBM MQ Channel) APPLTAG(MY-PRODUCER)
CONNTAG(MQCTA32BF16402360040DQM02_2023-09-01_00.09.00MY-PRODUCER)
sh-4.4$ echo 'dis conn(*)' all | runmqsc | grep -i my
APPLDESC(IBM MQ Channel) APPLTAG(MY-PRODUCER)
CONNTAG(MQCTA12BF16401460040DQM01_2023-09-01_00.08.58MY-PRODUCER)
APPLDESC(IBM MQ Channel) APPLTAG(MY-PRODUCER)
CONNTAG(MQCTA12BF16401460040DQM01_2023-09-01_00.08.58MY-PRODUCER)
sh-4.4$ echo 'dis conn(*)' all | runmqsc | grep -i my
APPLDESC(IBM MQ Channel) APPLTAG(MY-PRODUCER)
CONNTAG(MQCTA12BF16401460040DQM01_2023-09-01_00.08.58MY-PRODUCER)
APPLDESC(IBM MQ Channel) APPLTAG(MY-PRODUCER)
CONNTAG(MQCTA12BF16401460040DQM01_2023-09-01_00.08.58MY-PRODUCER)
APPLDESC(IBM MQ Channel) APPLTAG(MY-PRODUCER)
CONNTAG(MQCTA32BF16402360040DQM02_2023-09-01_00.09.00MY-PRODUCER)
APPLDESC(IBM MQ Channel) APPLTAG(MY-PRODUCER)
CONNTAG(MQCTA32BF16402360040DQM02_2023-09-01_00.09.00MY-PRODUCER)
sh-4.4$
qm01-qm-ibm-mq-0 · Details · Red...
ibmuser@student:~/Documents
Documents
mq-cmd.txt (~/Documents) - gedit

8 Review ArgoCD declarative model

One of the advantages of ArgoCD is its declarative model making sure the environment matches the desired state. When we deployed our ArgoCD application we established that the desired estate was to have two queue managers as part of the uniform cluster, but in the previous step we deleted one of the queue managers, so let's review what happens in this situation.

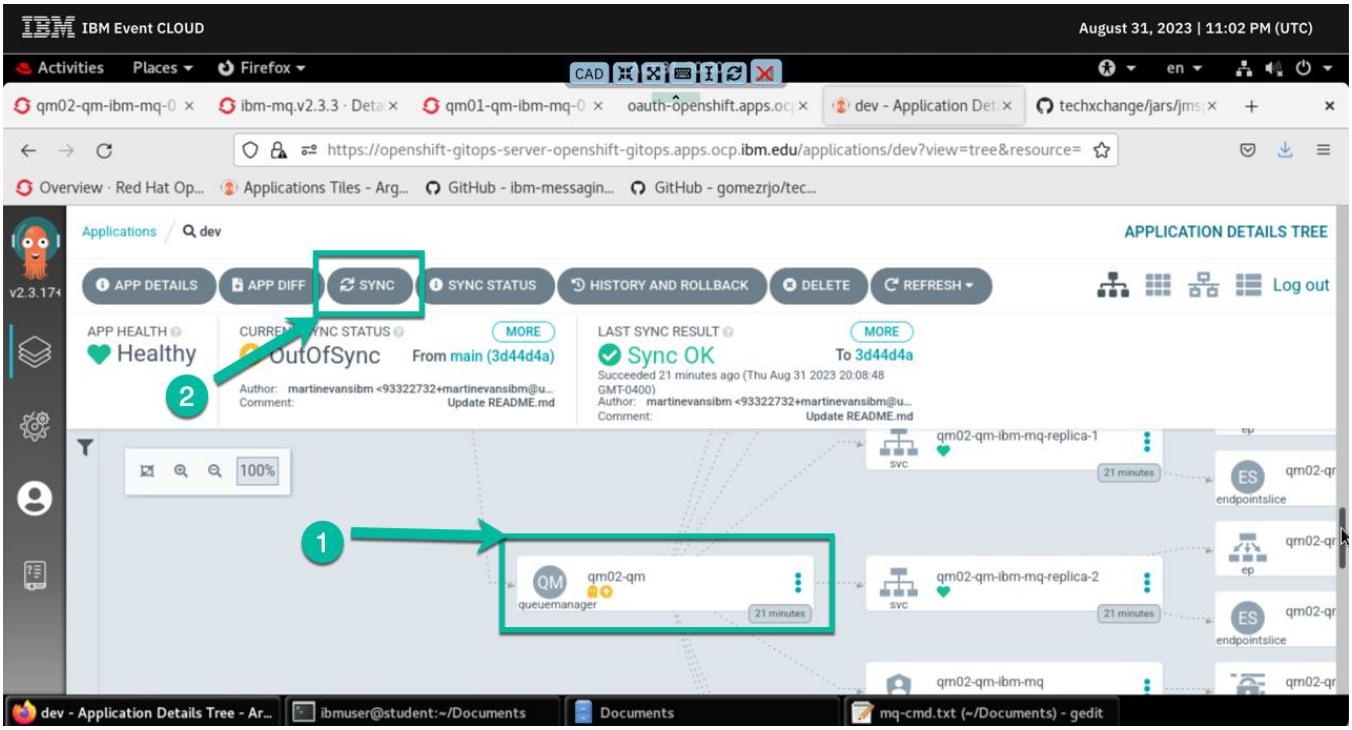
8.1 Review ArgoCd

Navigate to the ArgoCD web ui and check that status of the application. You will see that it is reported as out of sync because the current state (one queue manager) does not match the desired state (two queue managers).

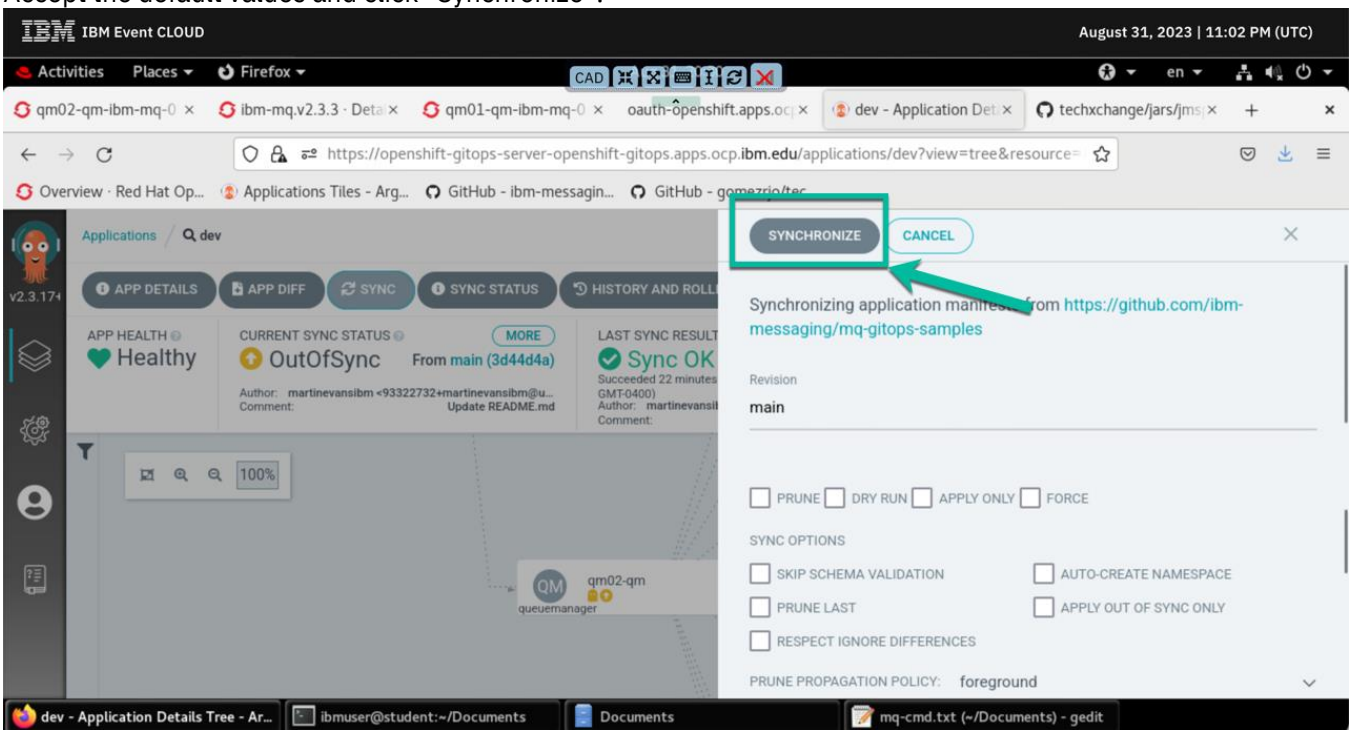


8.2 Synchronize GitOps App

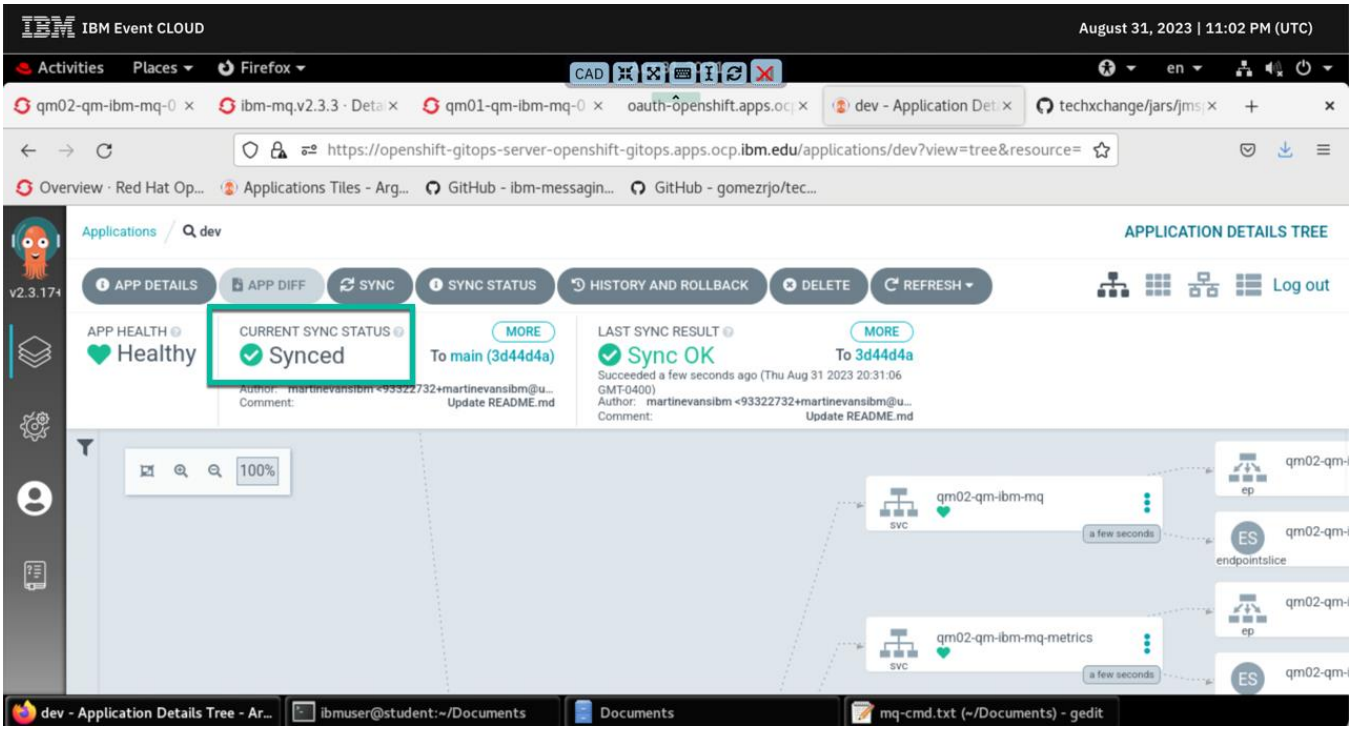
If you look at the application, you will see that QM2 is the one showing out of sync. As mentioned before we have selected a manual approach for learning purposes so you can have time to review the status of the different components, otherwise the queue manager and all its components would be recreated automatically. To take the deployment to the desired state click the "Sync" button.



Accept the default values and click “Synchronize”.



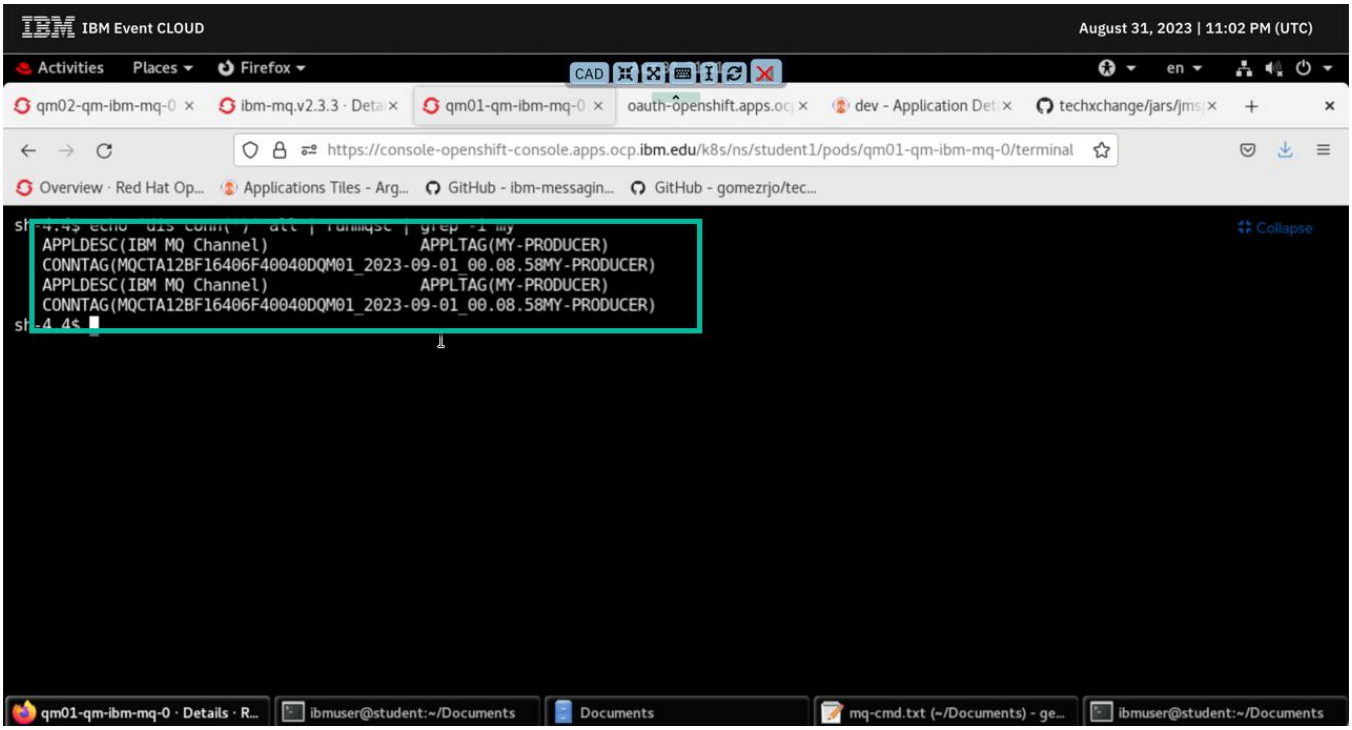
After a few second the status will change back to “Synced”

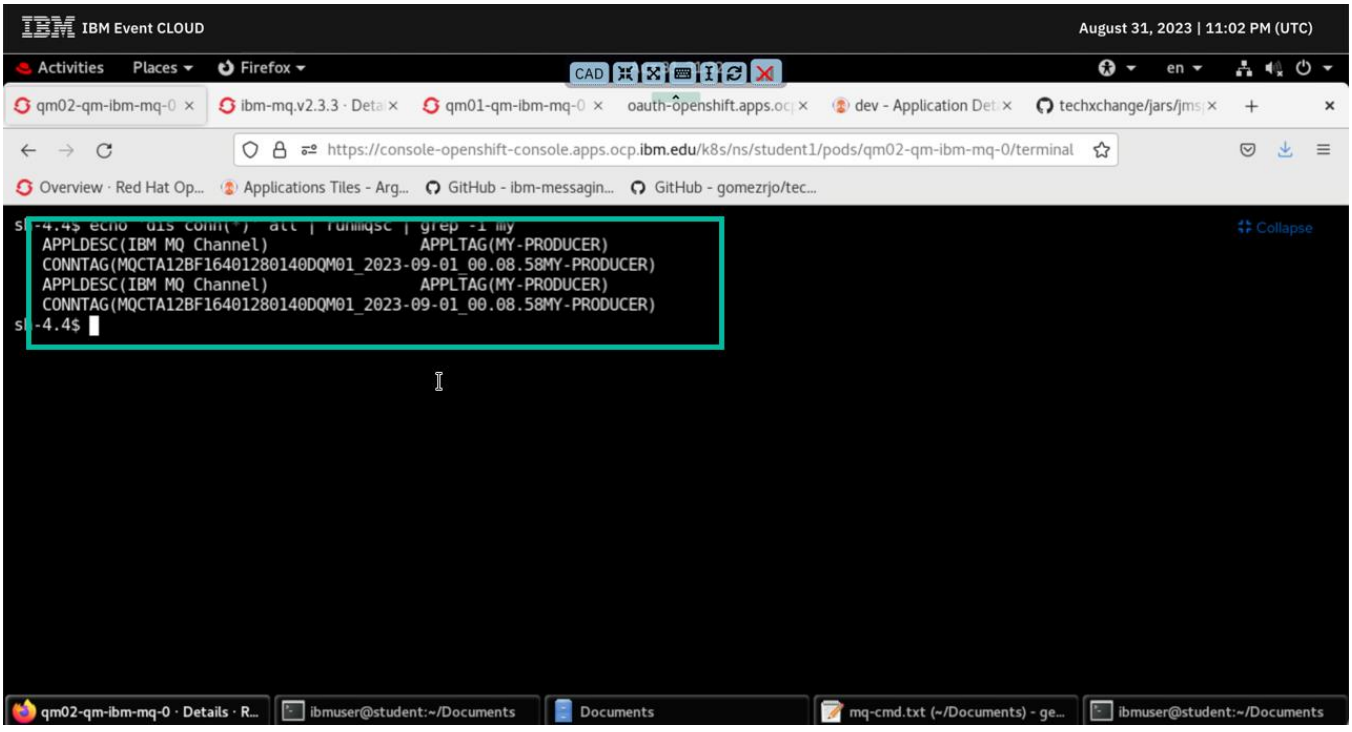


You can go back to the OCP Console and the MQ Operator to confirm queue manager two is back.

8.3 Review connectivity

Once you confirm both queue managers are up and running you can go back to the terminal of the active pod for each queue manager to check the number of active connections and you should see an even distribution as shown below.





If you want, you can add more instances to the MQ application to see how the connections are rebalances when you add or remove instances. A similar behavior would happen if additional queue manager are added to the uniform cluster and the connection would be rebalanced providing a way to scale horizontally.

9 Congratulations

Great job. You have completed the lab.