Red Hat

IndiGo

Tech Talk

Streamline Your Architecture with Red Hat Camel and AMQ Streams

Red Hat Team

Date 12-Aug-2024

Table of Contents

1 Delete the previous workspace from the Devspace

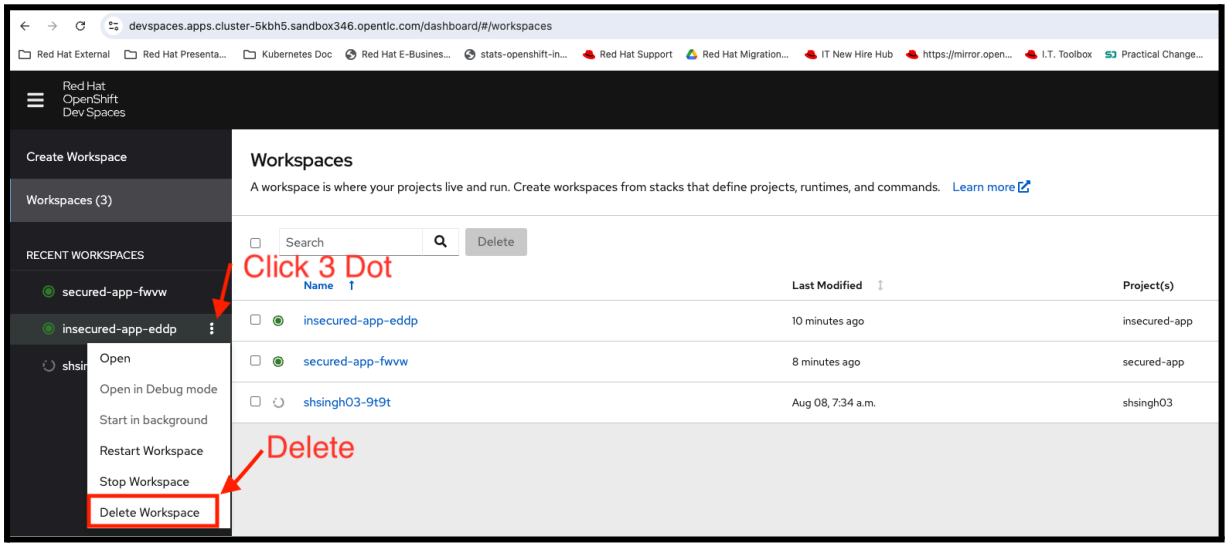
(Only when you have 2 Workspace)

Each of the following sections provides a brief description and a list of recommended actions. Additional explanations are provided where additional clarity is required.

URL:- https://devspaces.<CLUSTER-DOMAINNAME>

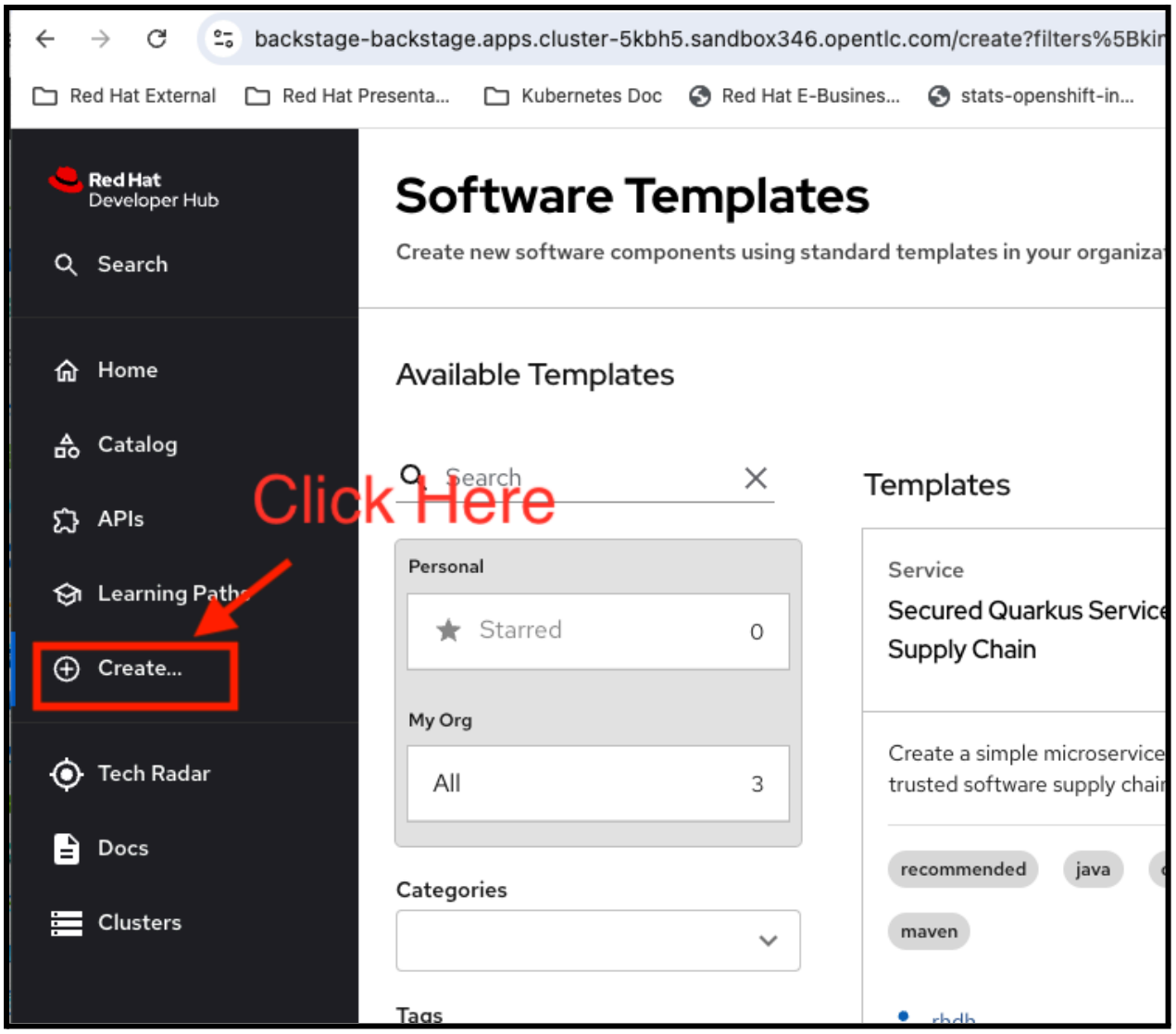
SAMPLE URL :-

https://devspaces.apps.cluster-5kbh5.sandbox346.opentlc.com/



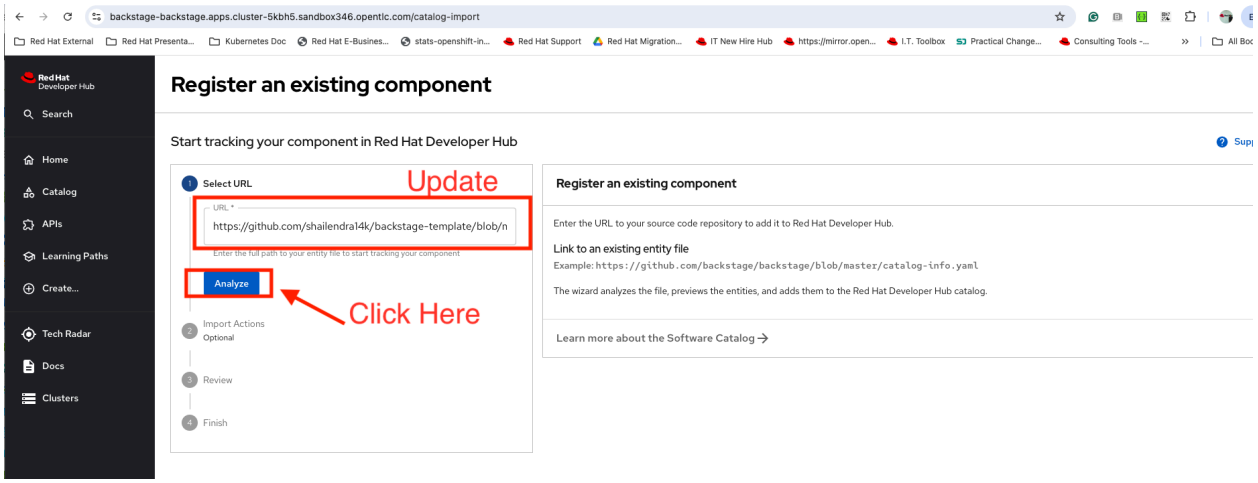
2 Import the DeveloperHub template for this demo .

2.1. Go to the Create Software Template section



2.2. Register the new template

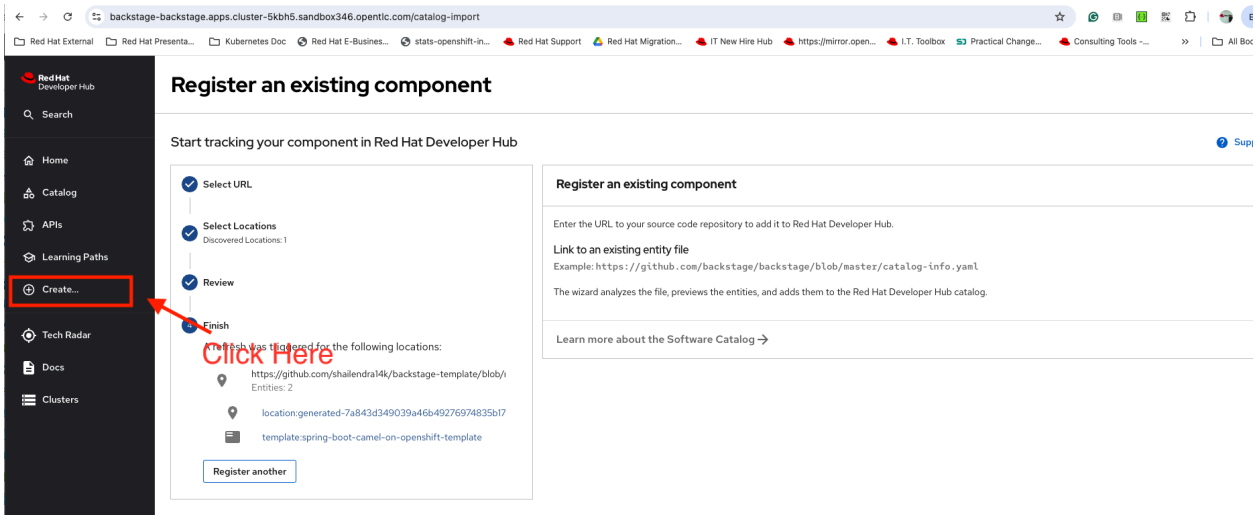
```
Template URL:-
https://github.com/shailendra14k/backstage-template/blob/main/backstage-camel/template.yaml
```



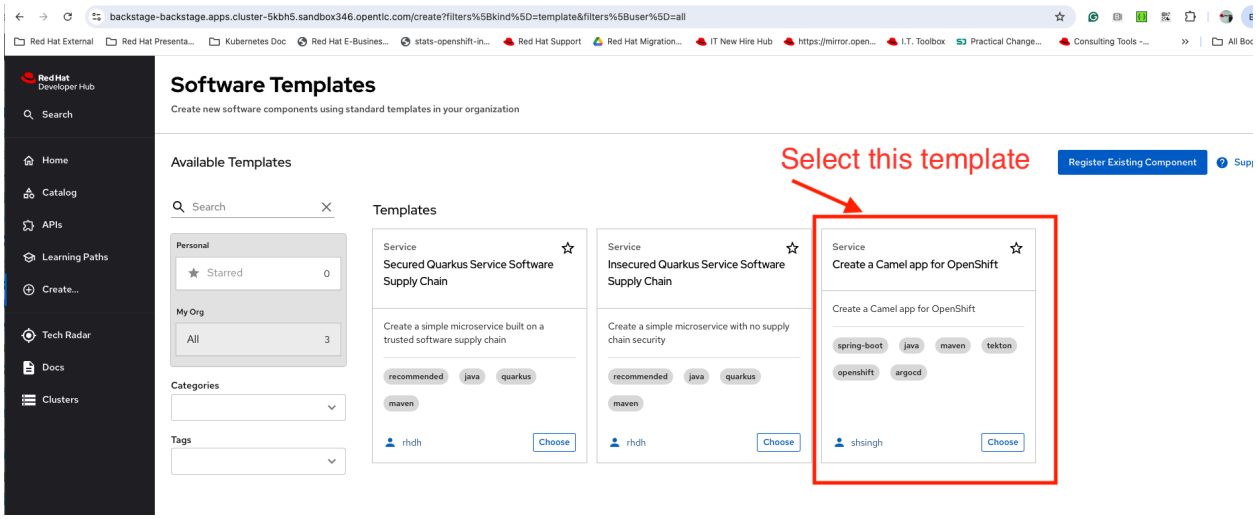
2.3. Finish the import

3 Create the Catalog from template

3.1 Go to the Template Create section

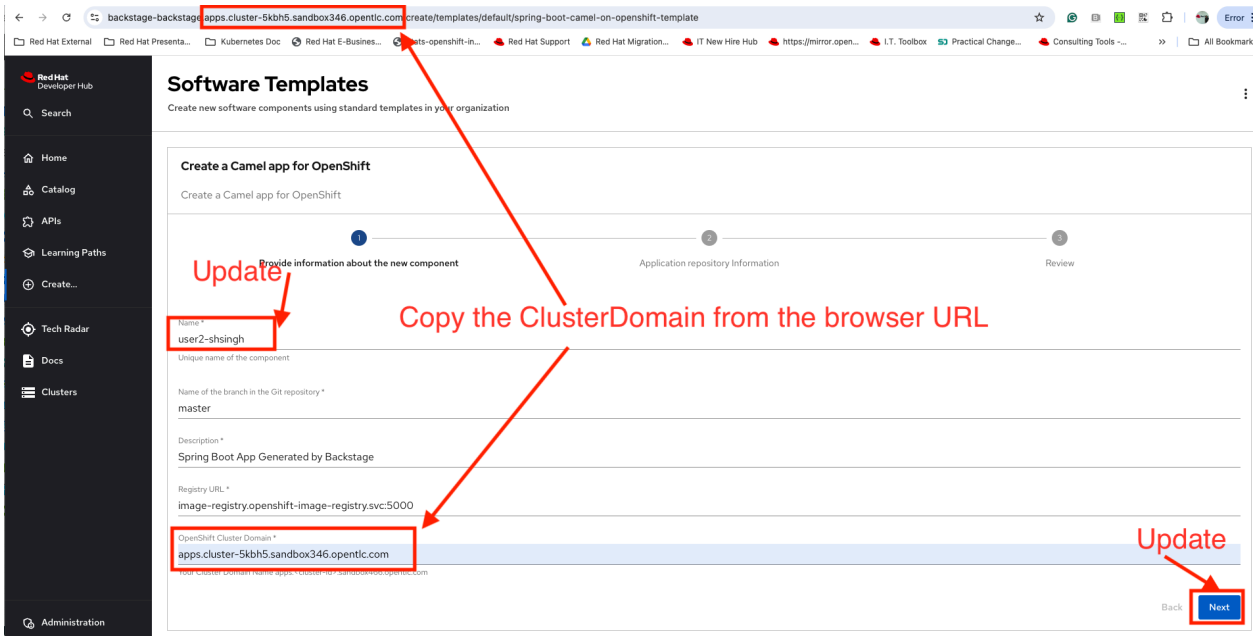


3.2. Choose the template “Create a Camel app for OpenShift”

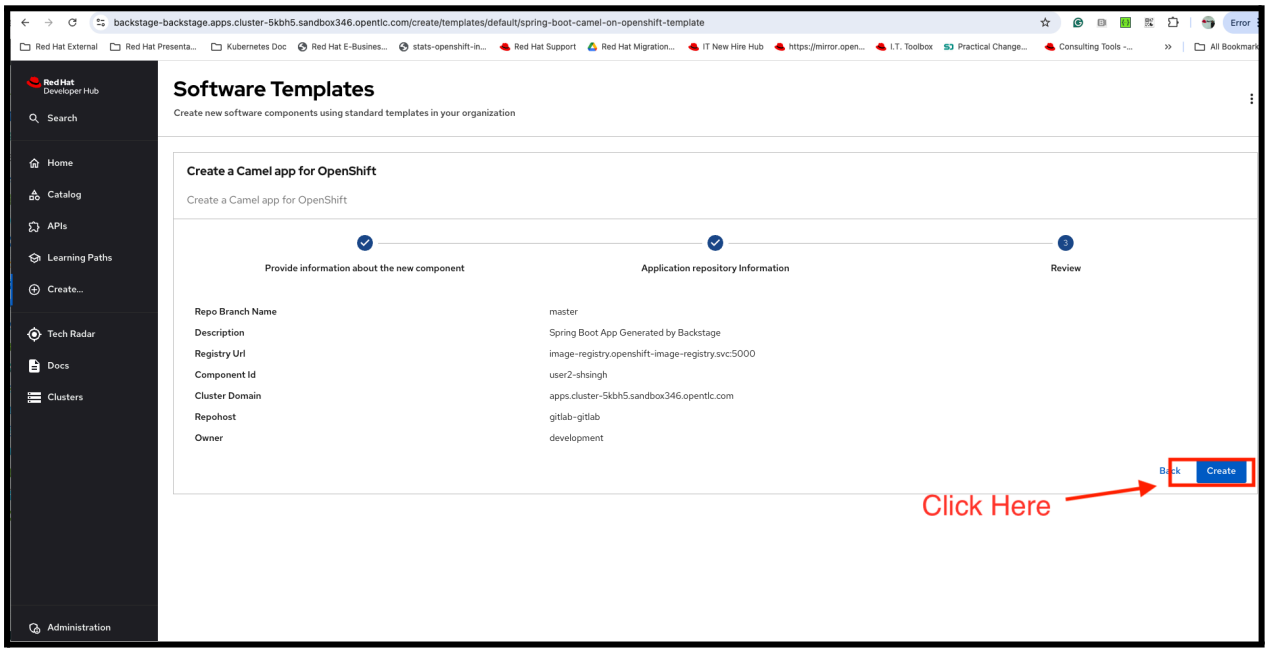


3.3 Add the details, Name and Cluster Domain, copy from the browser url as shown below

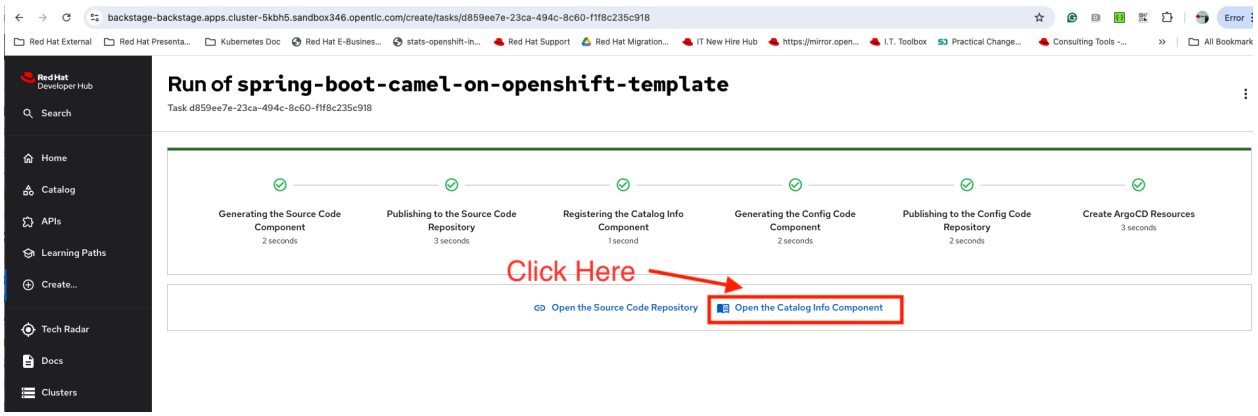
`Name :- user1-<yourname> (no cpas)`



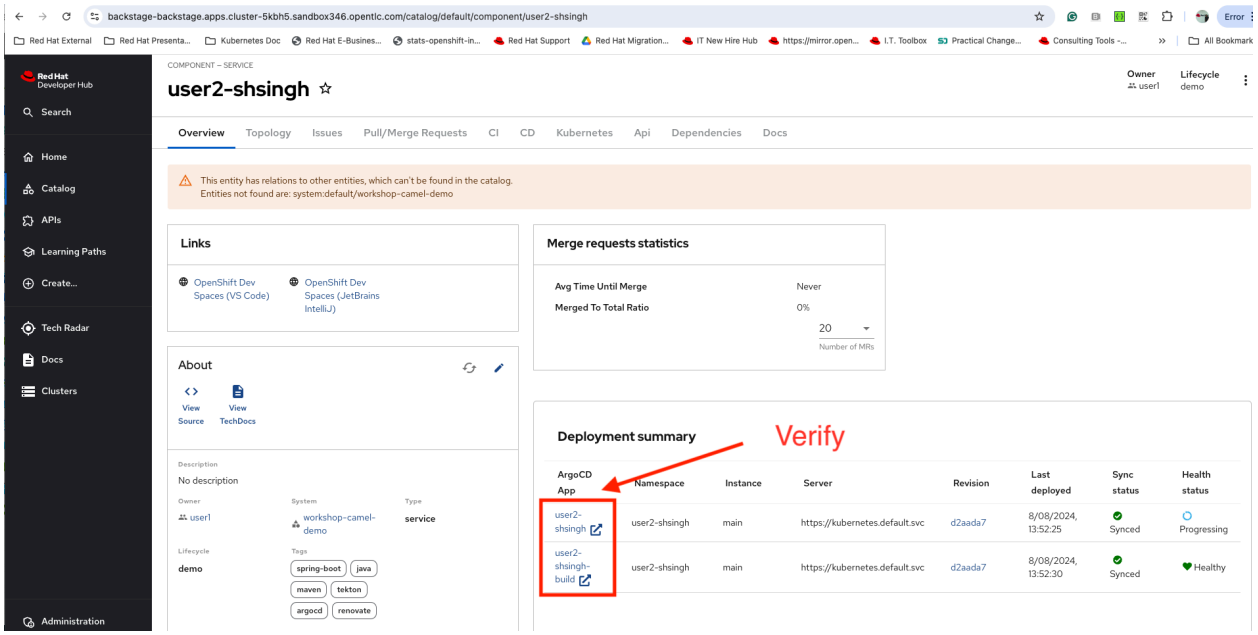
3.4 Click Next and verify the repo details and create



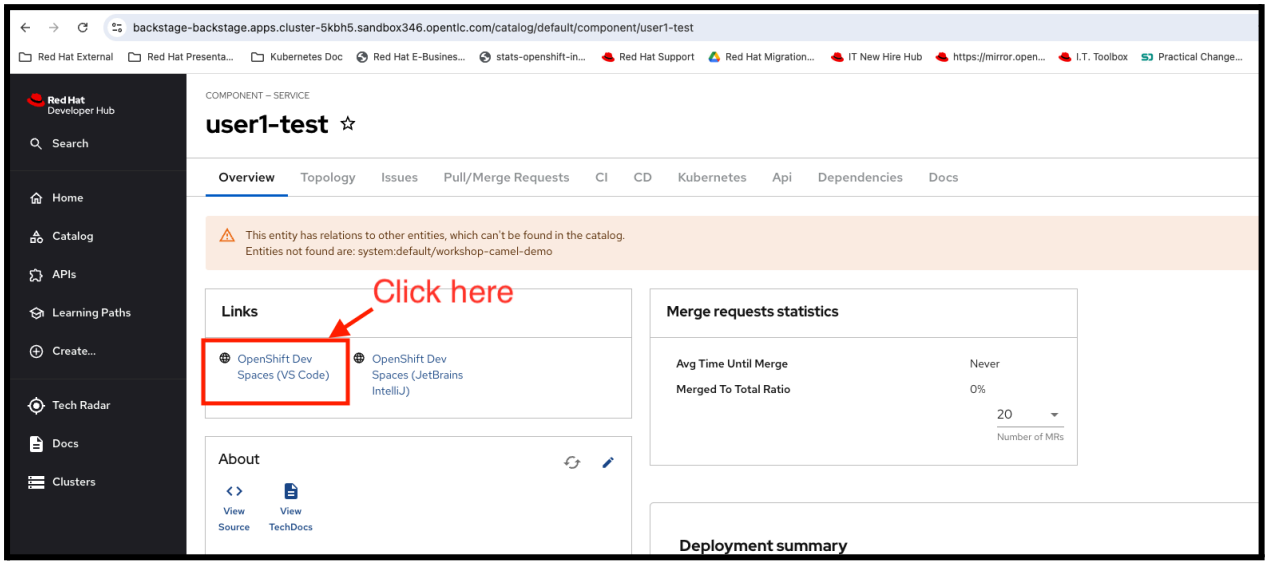
3.5. The template pipeline will be executed for the initial setup



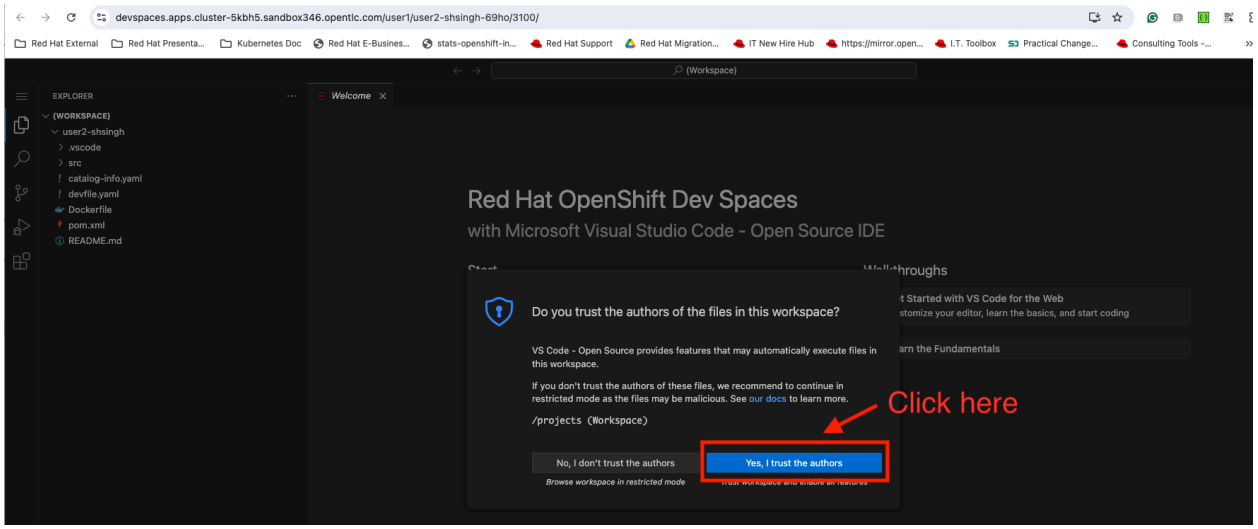
3.6 Verify the Catalog



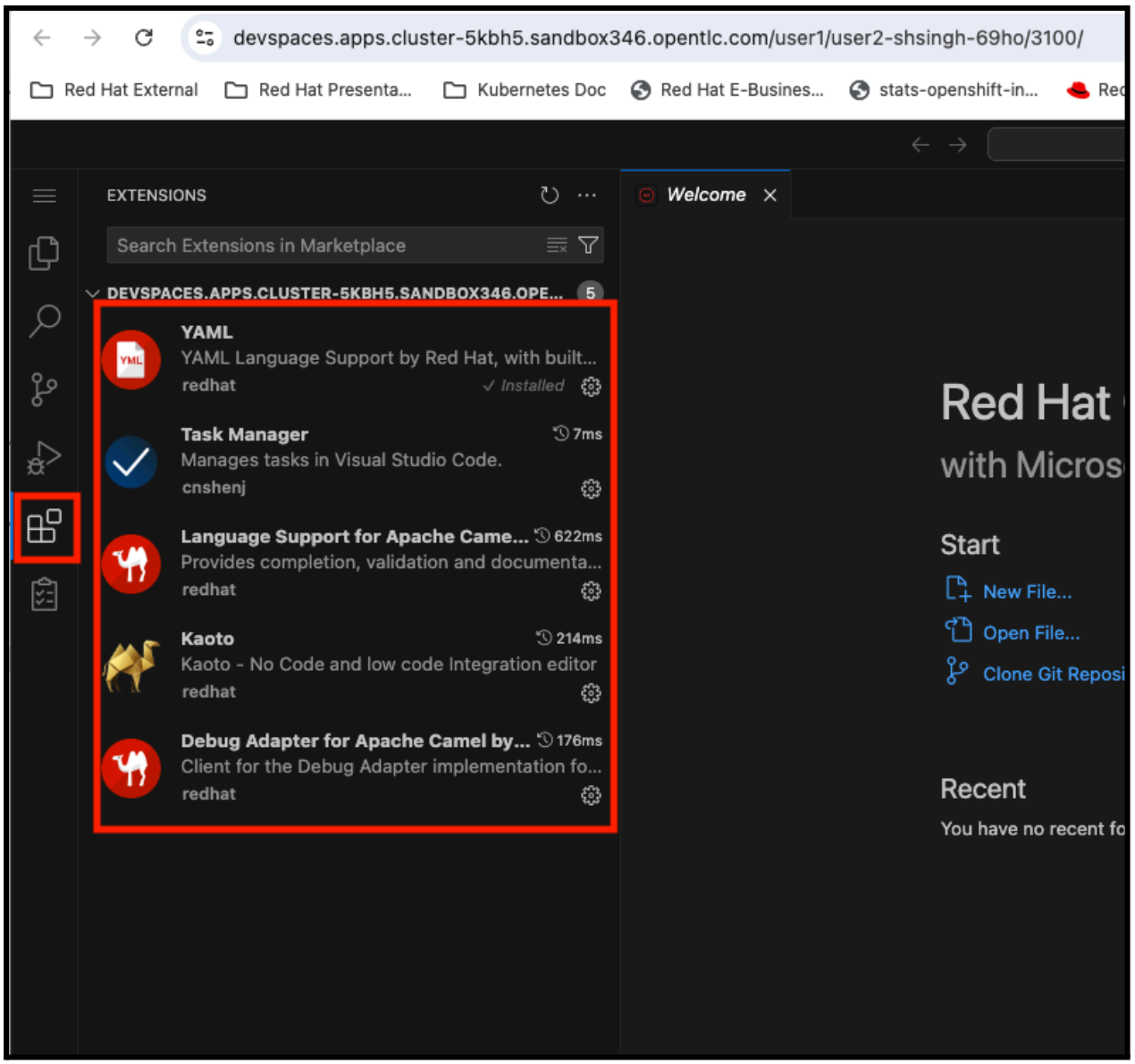
4 Open the new workspace from the developer Hub.



4.1 After waiting a few minutes for OpenShift Dev Spaces to finish setting up your workspace, You click the button Yes, I trust the authors.

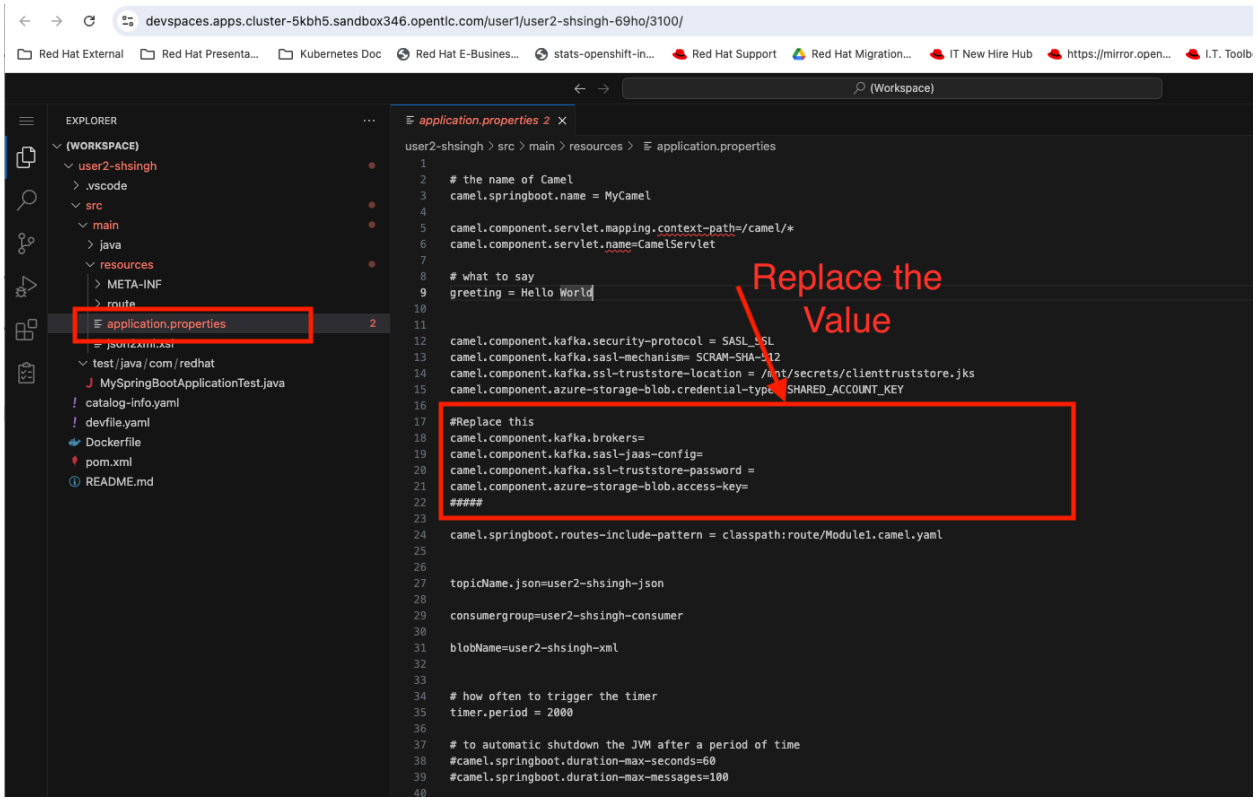

devspaces.apps.cluster-5kbh5.sandbox346.opentlc.com/user1/user2-shsingh-69ho/3100/
Red Hat OpenShift Dev Spaces
with Microsoft Visual Studio Code - Open Source IDE
Do you trust the authors of the files in this workspace?
VS Code - Open Source provides features that may automatically execute files in this workspace.
If you don't trust the authors of these files, we recommend to continue in restricted mode as the files may be malicious. See our docs to learn more.
/projects (Workspace)
No, I don't trust the authors
Yes, I trust the authors
Click here

4.2. Verify all the extensions have been installed correctly.



5 Module 1

5.1 Update the application.properties



~~~~

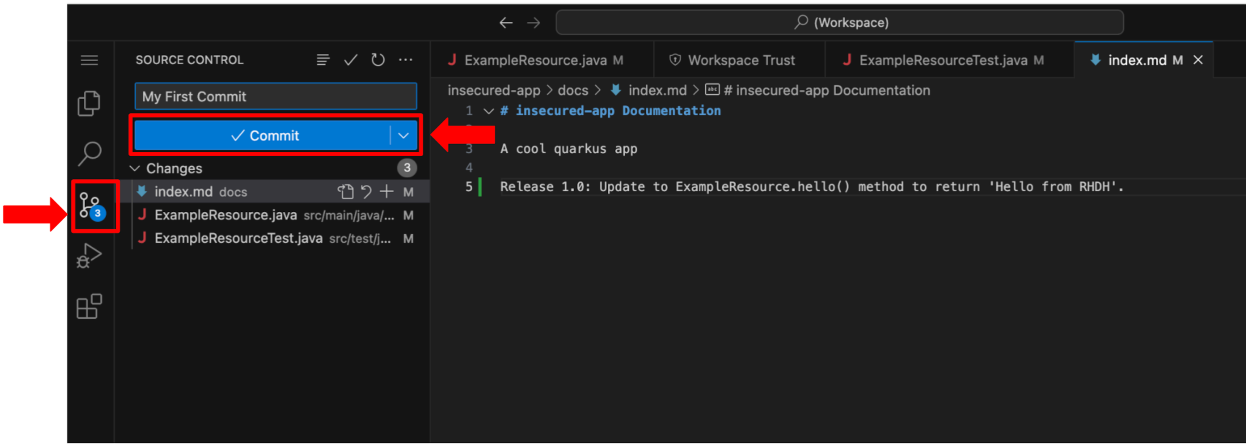
```
camel.component.kafka.brokers=my-cluster-kafka-route1-bootstrap-amq-stre
ams-kafka.apps.shailendra14k.in:443
camel.component.kafka.sasl-jaas-config=
org.apache.kafka.common.security.scram.ScramLoginModule required
username="console-cluster-user1"
password="JkP7hGT7rpJdlDdBQ3zblDTUZ6Gau0zH";
camel.component.kafka.ssl-truststore-password = indigo
camel.component.azure-storage-blob.access-key=
Y/fgFLVIPUGJ6MfQlWLxPWaILEE+BXV2FtzQwKwePp2XHKpF2djTaaNfc7utIHxD+m3OAxs8
Ih4i+AStY7QFJg==
```

~~~~~

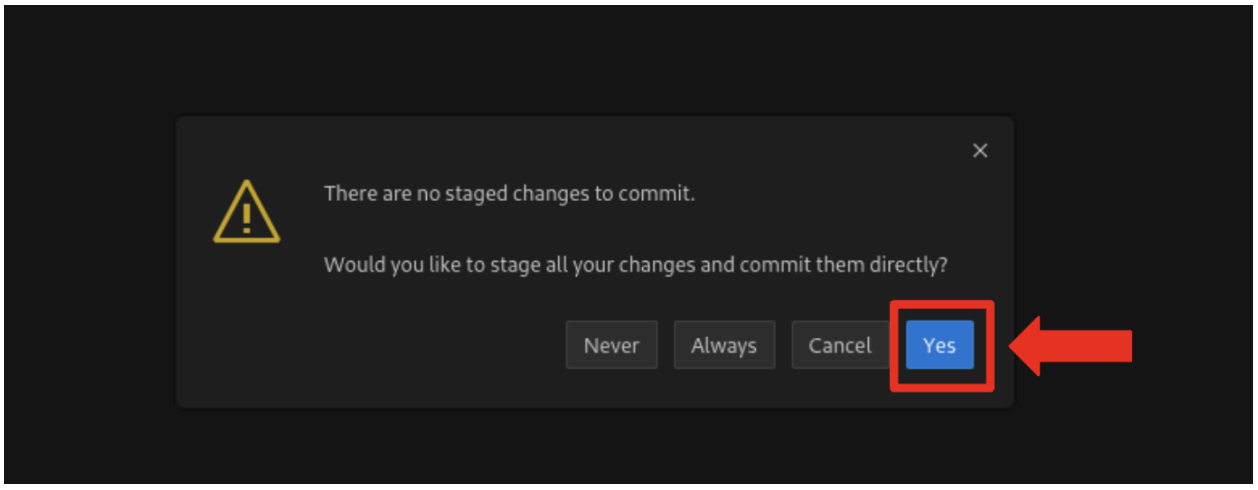
5.2 Commit the Changes

5.2.1. You click on the Source Control icon located in the left menu.

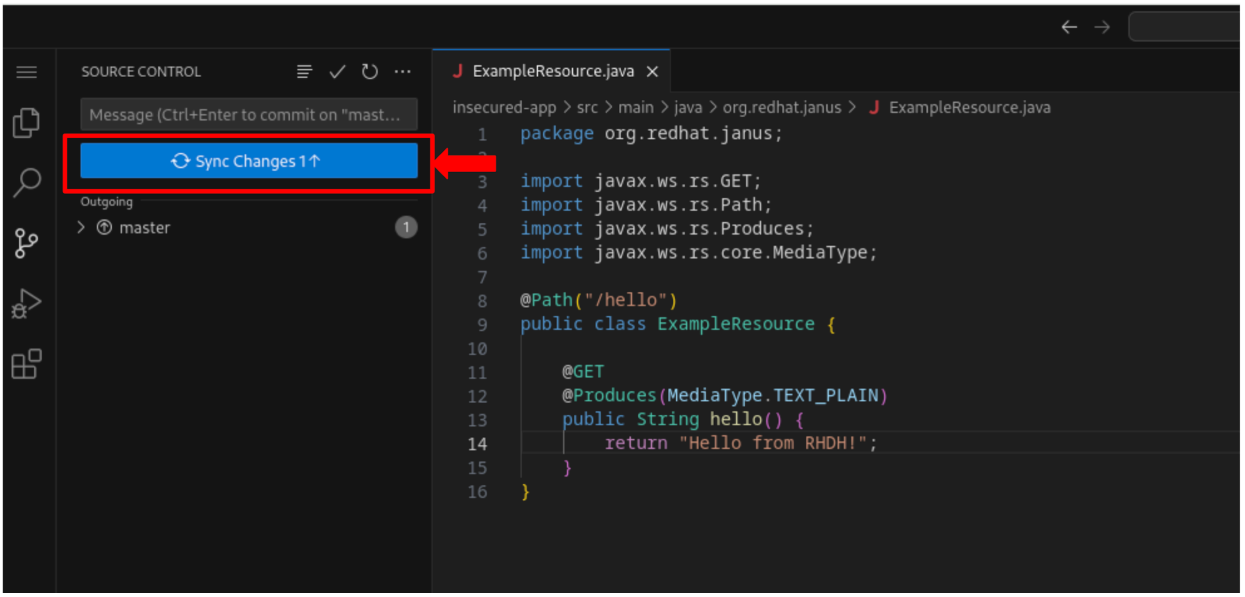
5.2.2. Then, you enter the commit message "updated application properties" and click on the Commit button to finalize your changes.



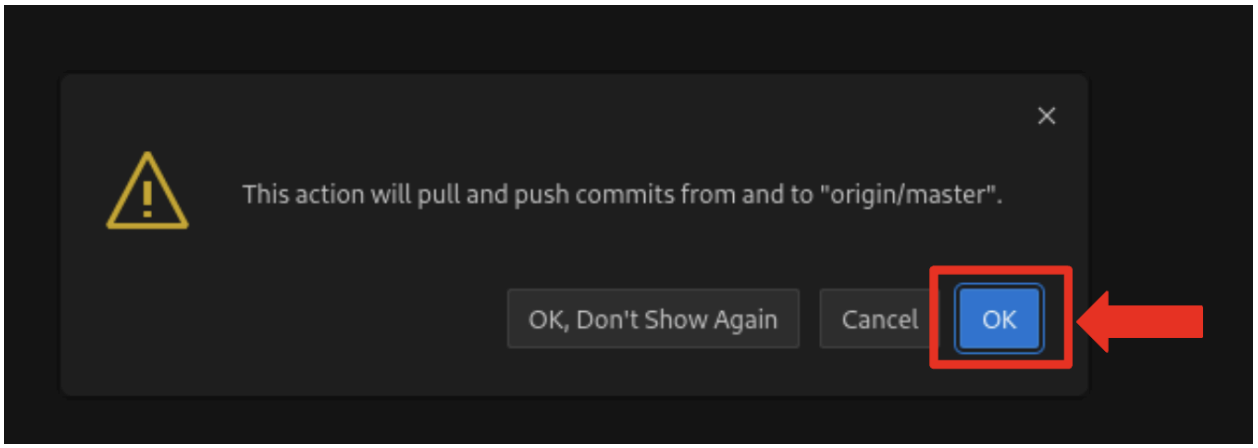
5.2.3. In the pop-up window that follows, you click Yes to stage your changes.



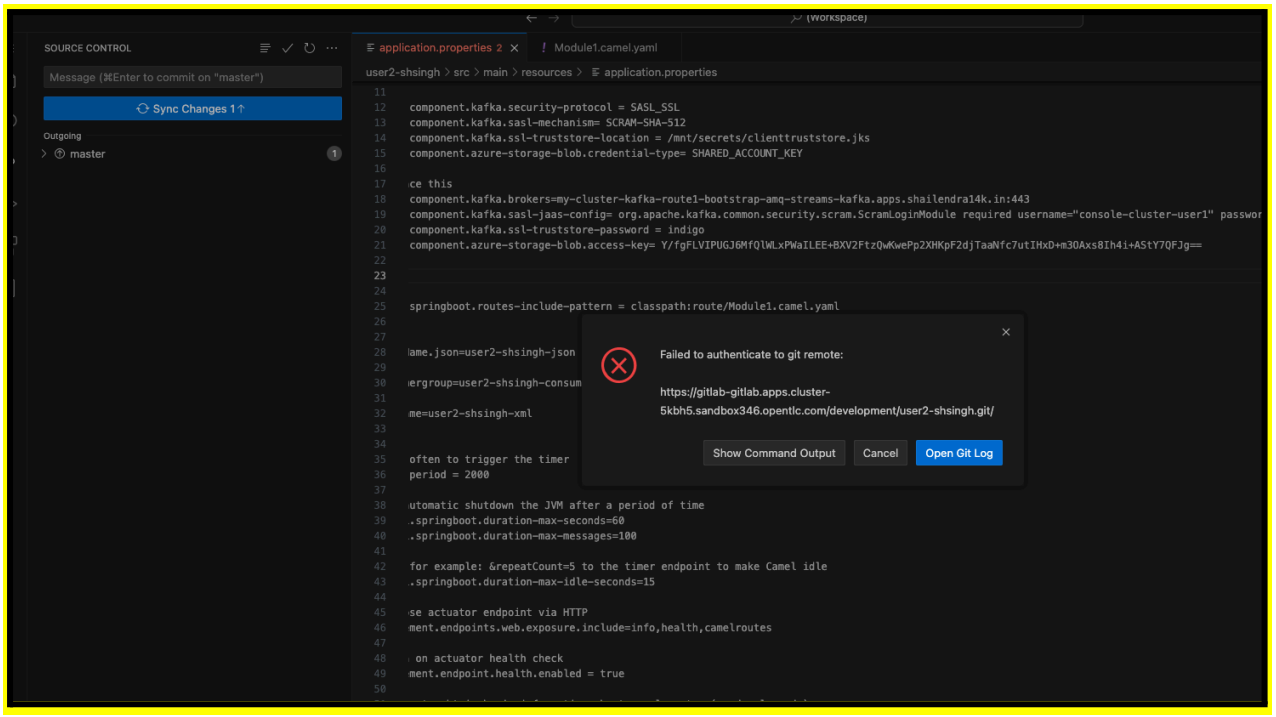
5.2.4. Finally, you click on the Sync Changes button.



6.2.5. In the pop-up that follows, you click OK to push your changes and complete the process.

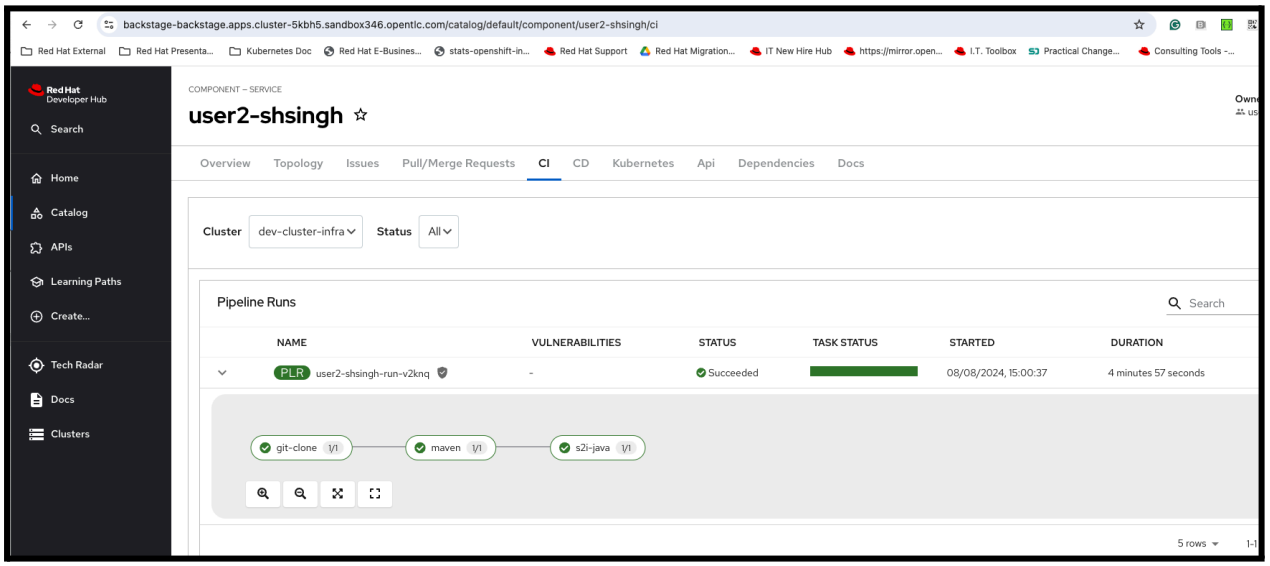


In Case the commit fails like the below



Delete all the workspaces from the Devspace URL and start the new workspace from the same software catalogue.

5.3 Verify the pipeline



5.4 Open the application

URL:-

https://<PROJECT-NAME>-<PROJECT-NAME>.<CLUSTER-DOMAINNAME>/camel/demo

SAMPLE URL :-

https://user2--shsingh03.apps.cluster-5kbh5.sandbox346.opentlc.com/camel/demo

6 Module 2

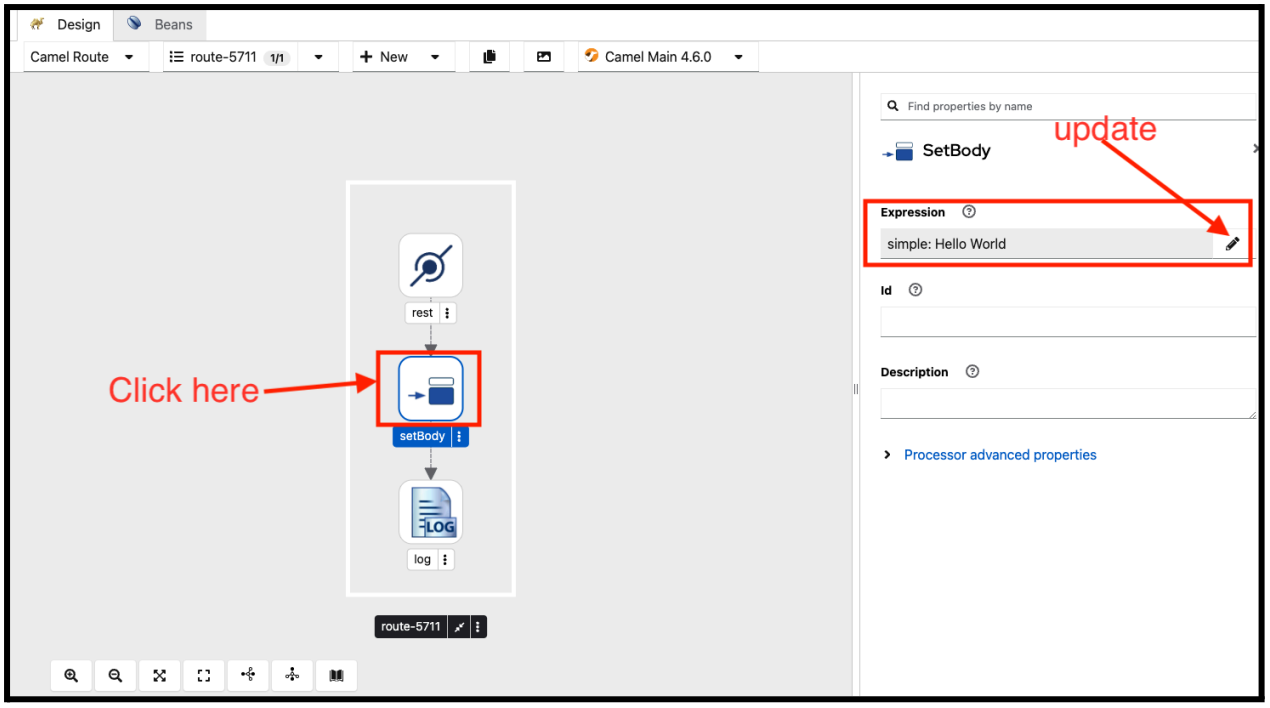
6.1 Update Expression of SetBody

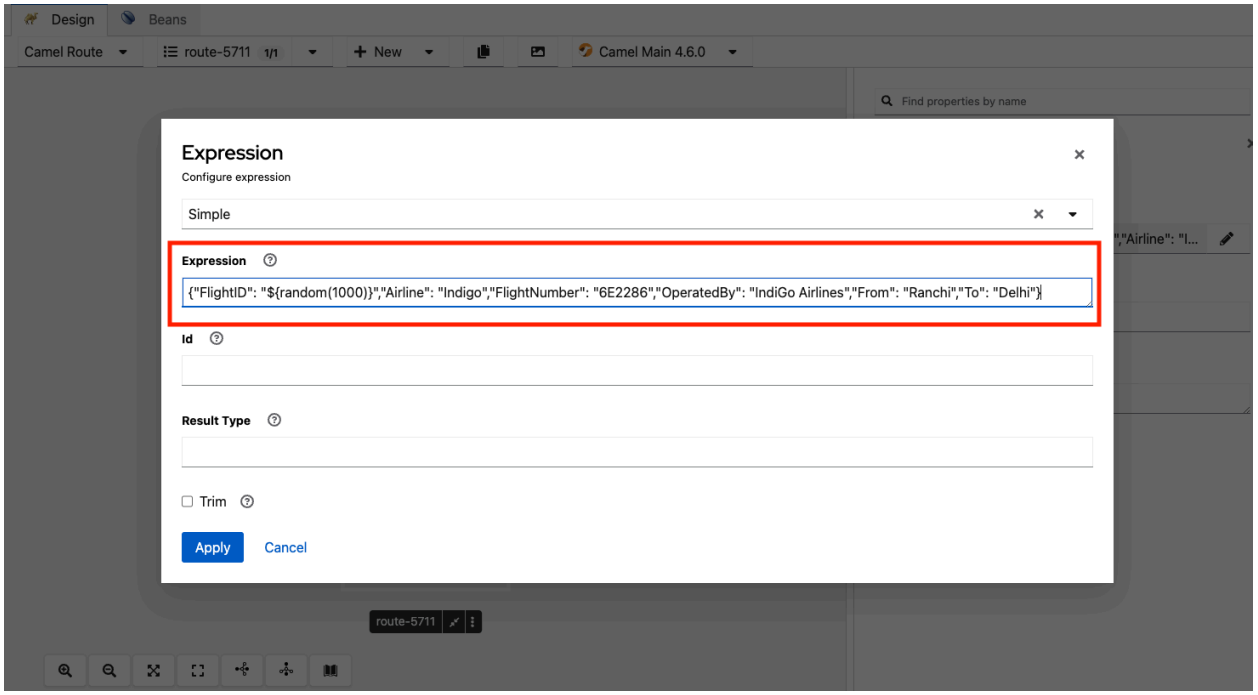
6.1.1. Go to src/main/java/resource/Module1.camel.yaml

6.2.2. Update the setBody component with below expression

Expression:-

```
{"FlightID": "${random(1000)}","Airline": "Indigo","FlightNumber":
"6E2286","OperatedBy": "IndiGo Airlines","From": "Ranchi","To": "Delhi"}
```





6.2 Commit the Changes

[Follow Module1 steps]

6.3 Verify the pipeline has been completed successfully.

Follow Module1 steps

6.4 Verify the Output

URL:-

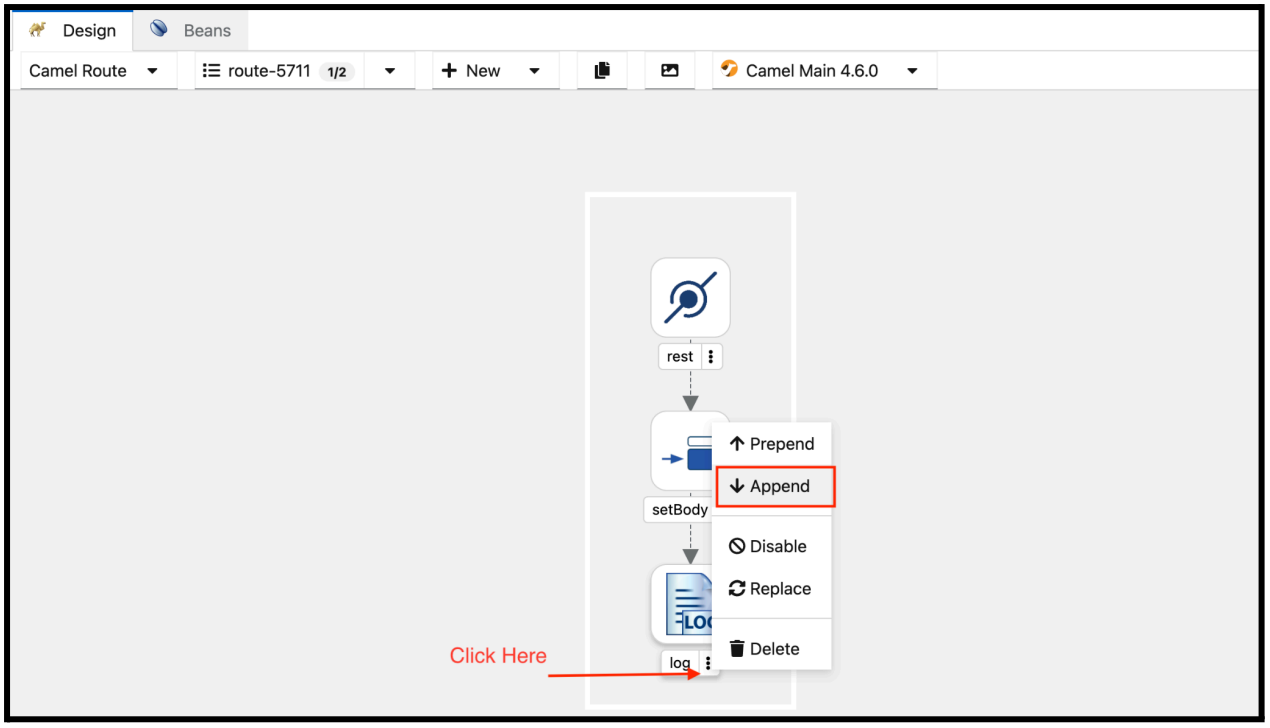
https://<PROJECT-NAME>-<PROJECT-NAME>.<CLUSTER-DOMAINNAME>/camel/demo

SAMPLE URL :-

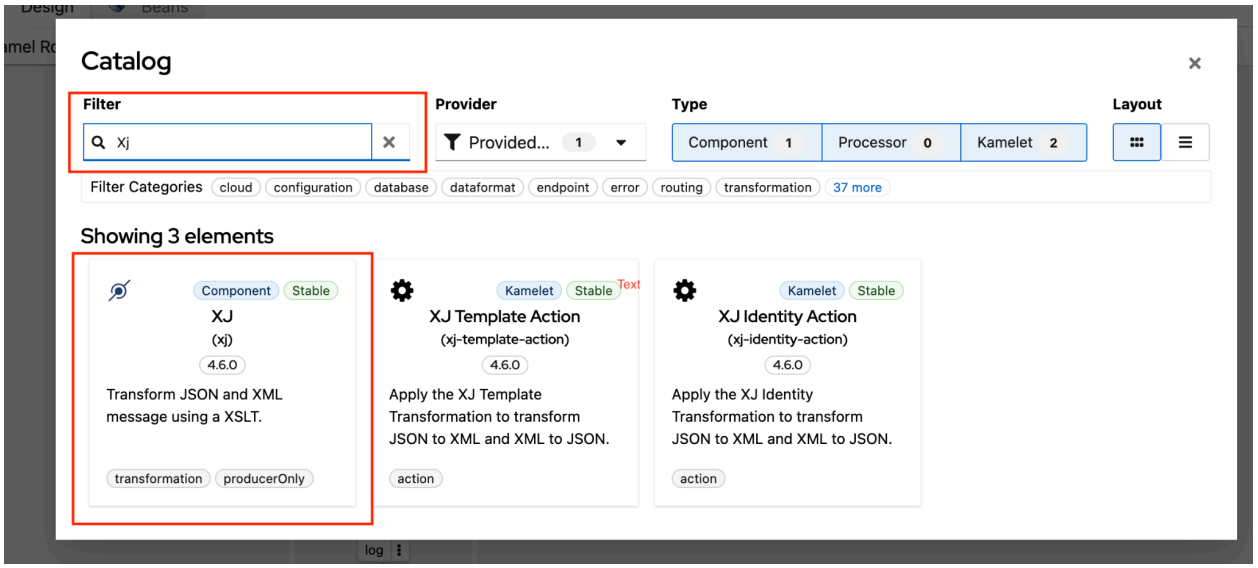
https://shsingh03-shsingh03.apps.cluster-5kbh5.sandbox346.opentlc.com/camel/demo

7. Module 3

7.1. Click on the 3 dot of the log component and select append

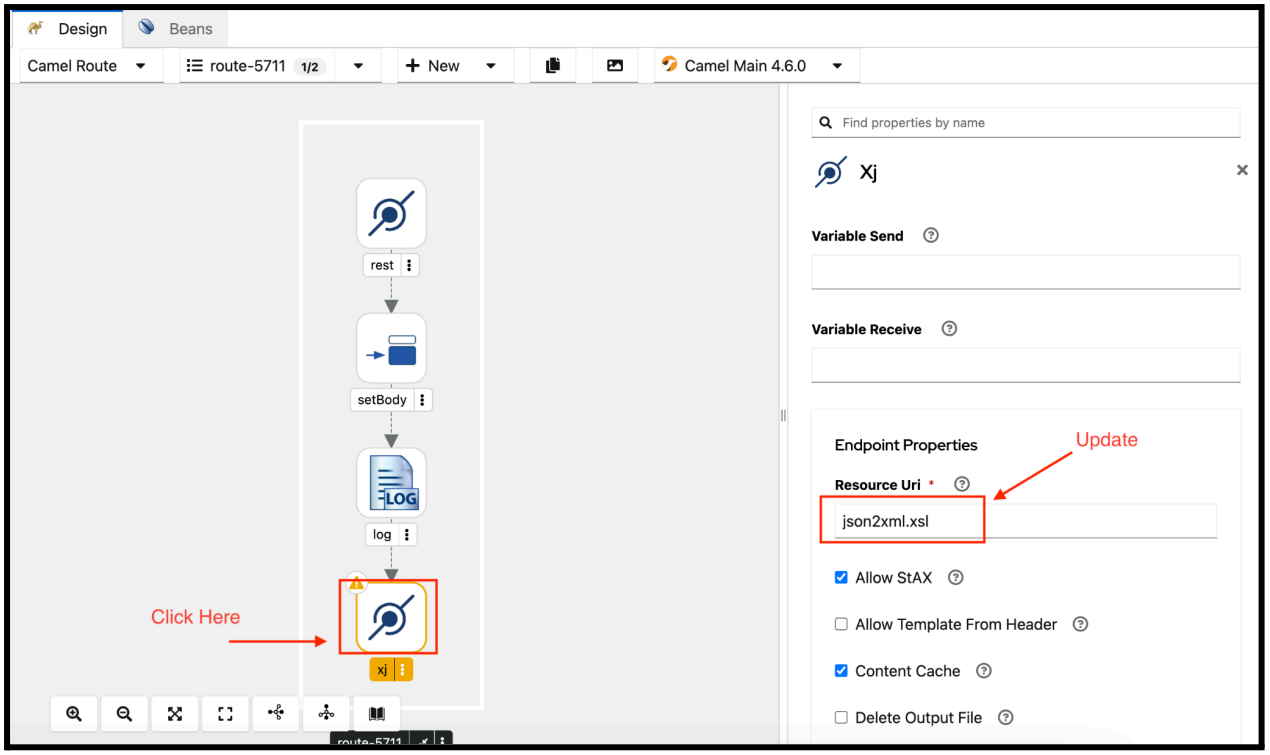


7.2. Select the xj component from the catalog

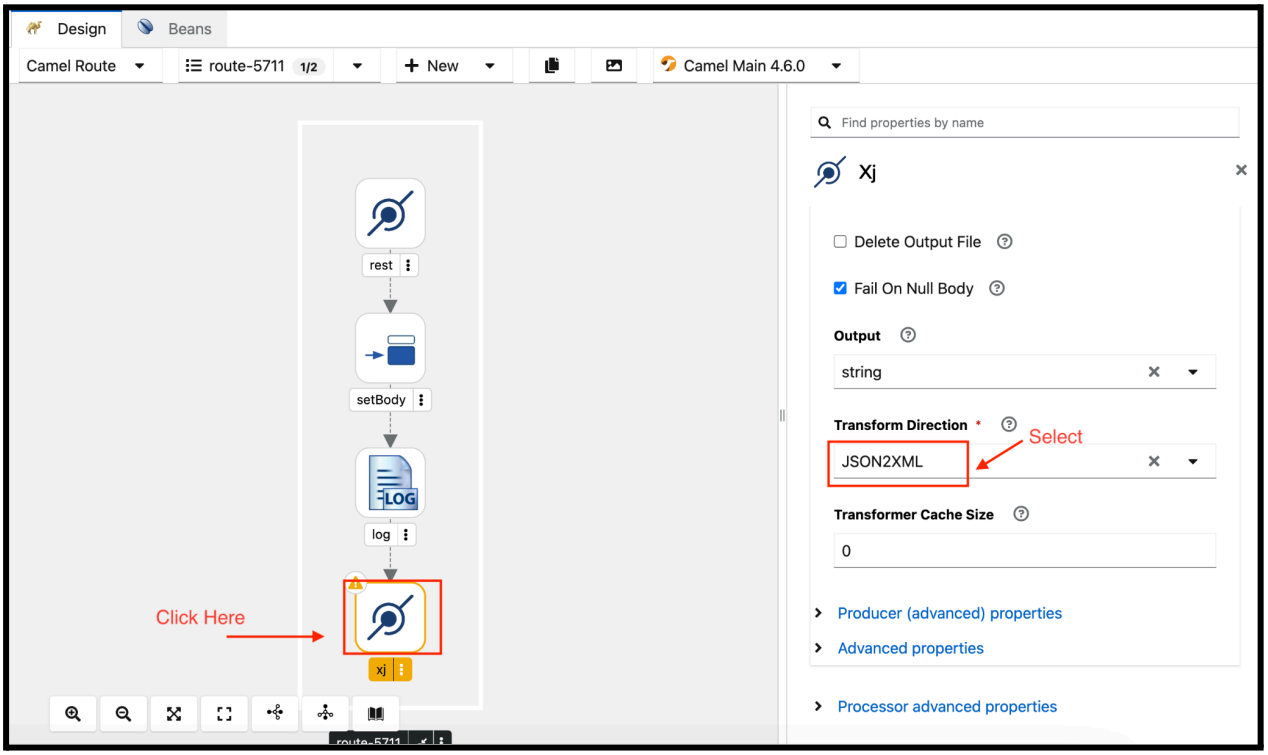


7.2.1 Update the below configuration for the xj component

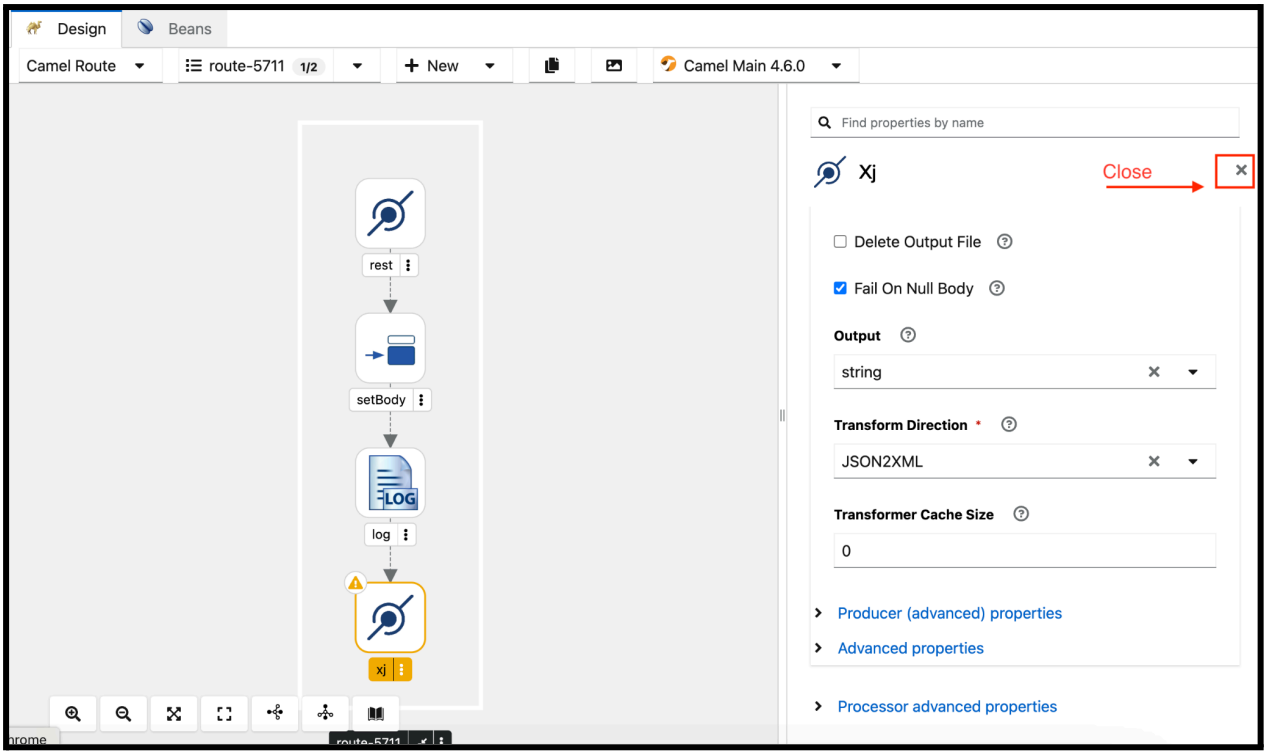
- Resource Uri --> json2xml.xsl



- Transform Direction ---> JSON2XML



- Close



7.3 Commit the Changes

Follow Module1 steps

7.4 Verify the pipeline has been completed successfully.

Follow Module1 steps

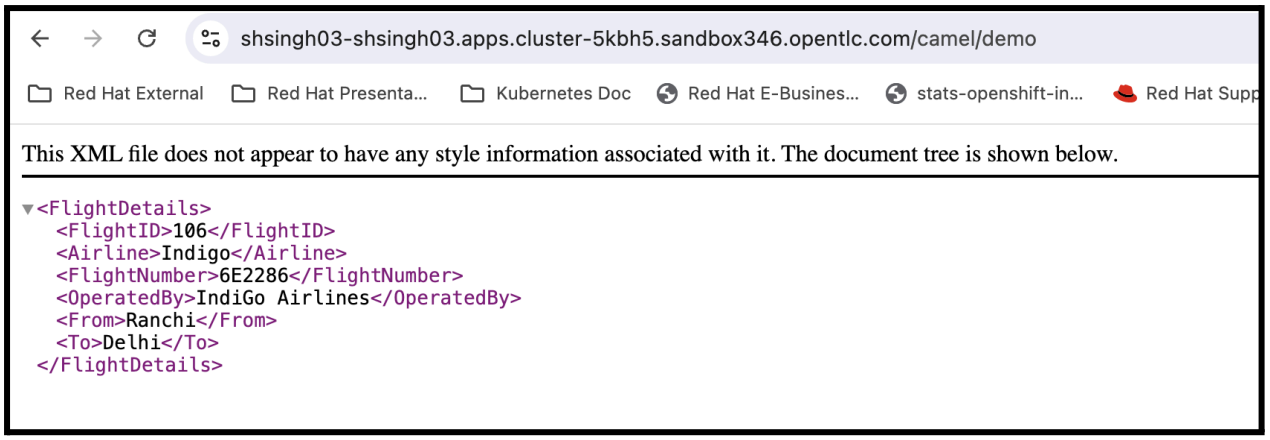
7.5 Verify the Output

URL:-

https://<PROJECT-NAME>-<PROJECT-NAME>.<CLUSTER-DOMAINNAME>/camel/demo

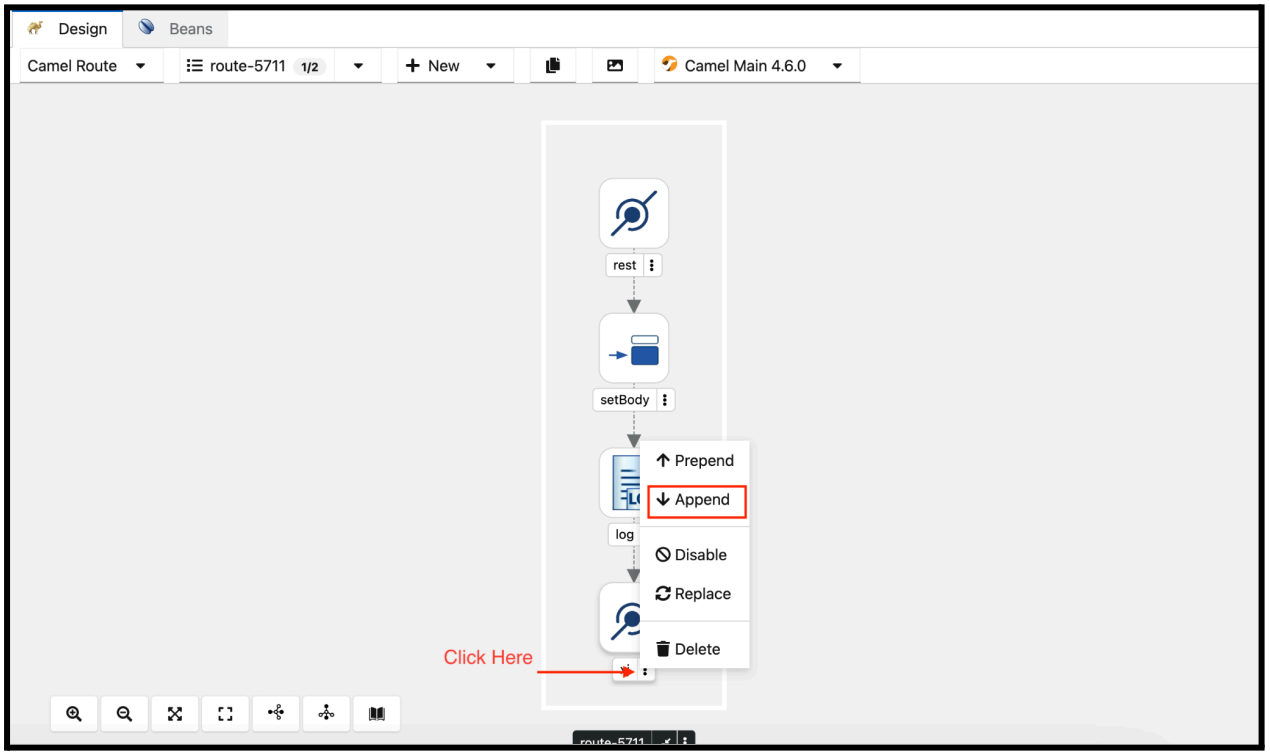
SAMPLE URL :-

https://shsingh03-shsingh03.apps.cluster-5kbh5.sandbox346.opentlc.com/camel/demo

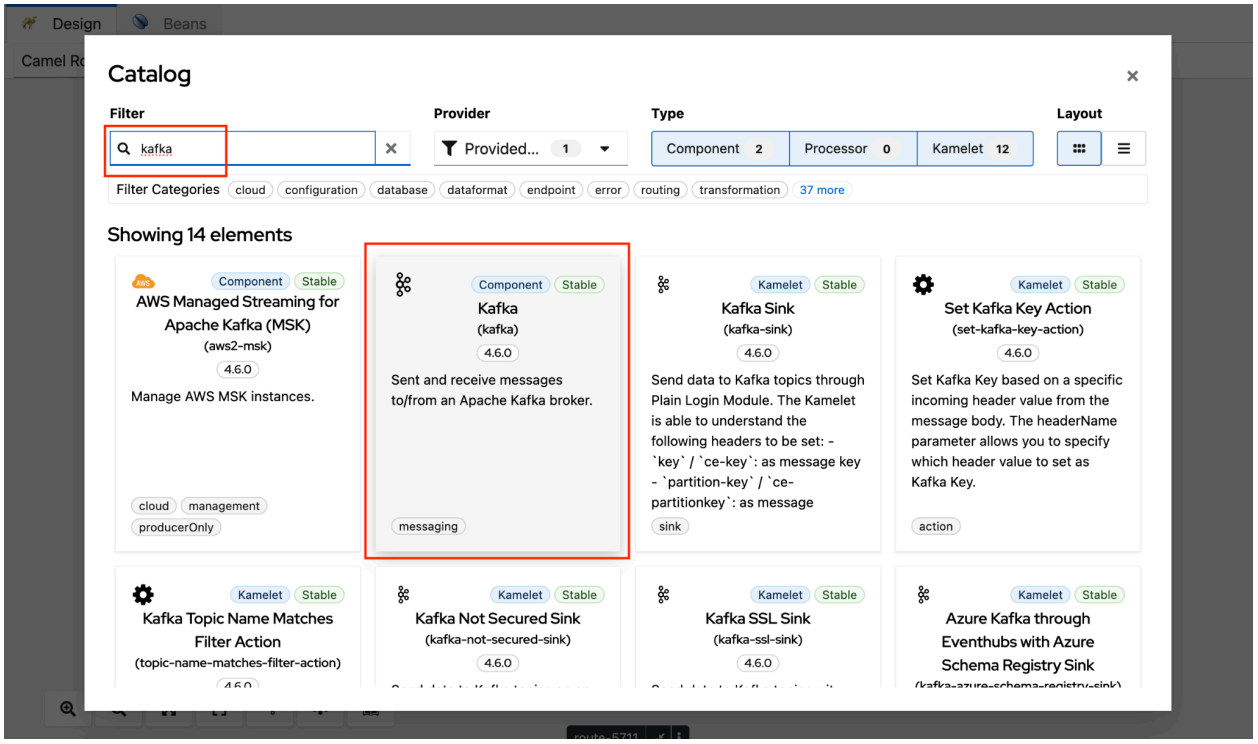


8 Module 4

8.1. Click on the 3 dot of the XJ component and select append

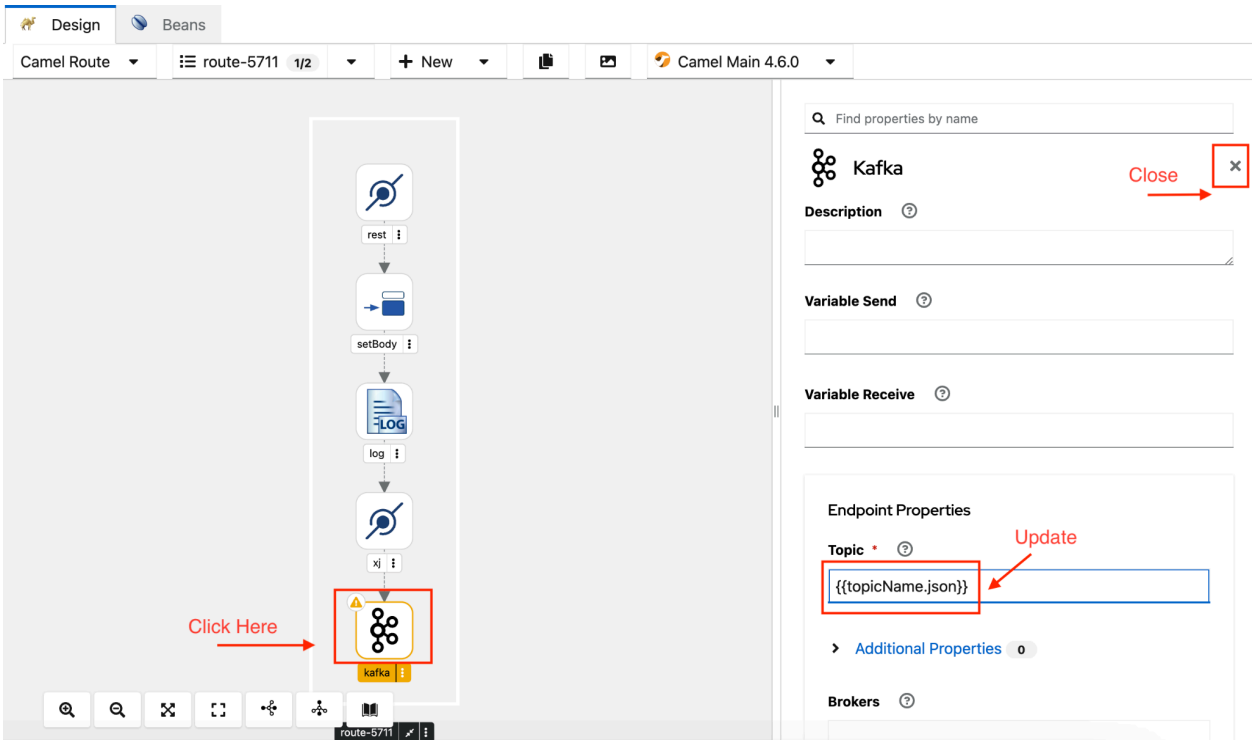


8.2.Select the kafka component from the catalog



8.2.1 Update properties for the Kafka component and close

- Topic --> {{topicName.json}}



8.3 Commit the Changes

Follow Module1 steps

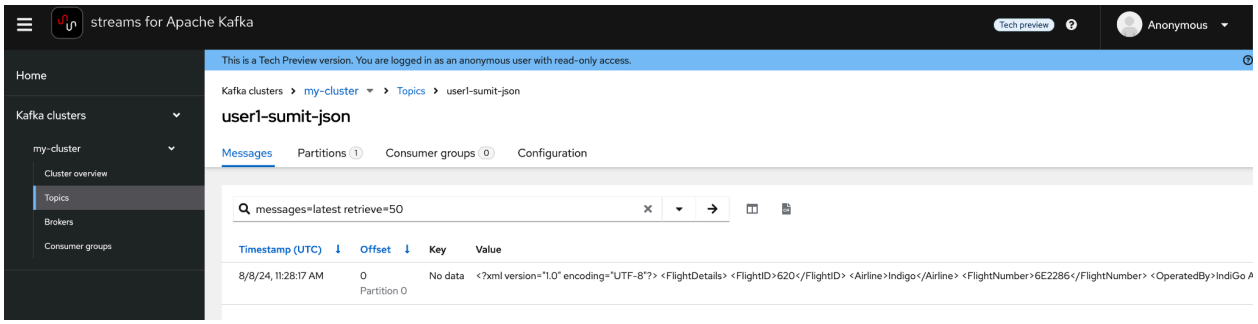
8.4 Verify the pipeline has been completed successfully.

Follow Module1 steps

8.5 Verify the Output

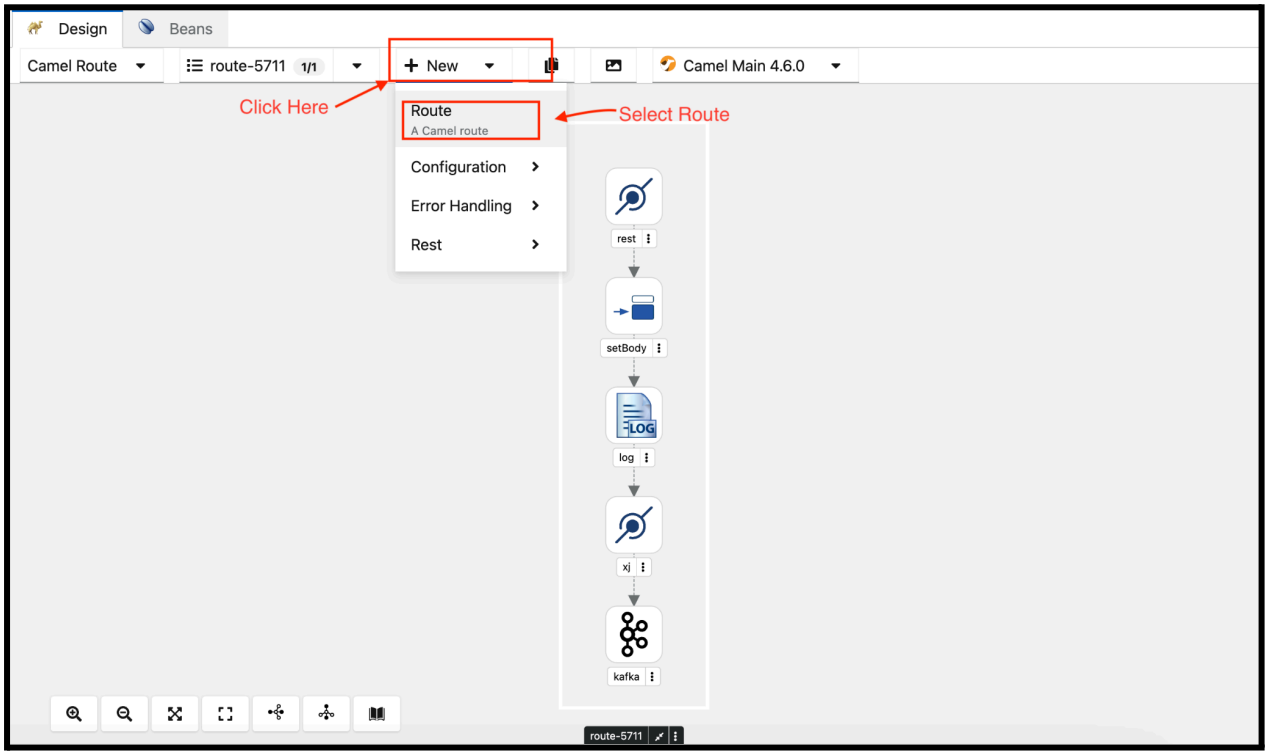
AMQ Stream(Kafka)access will be with the Instructor.

Topic with the <projectName>-json will be created and will have all the messages. The screenshot below is for user2.

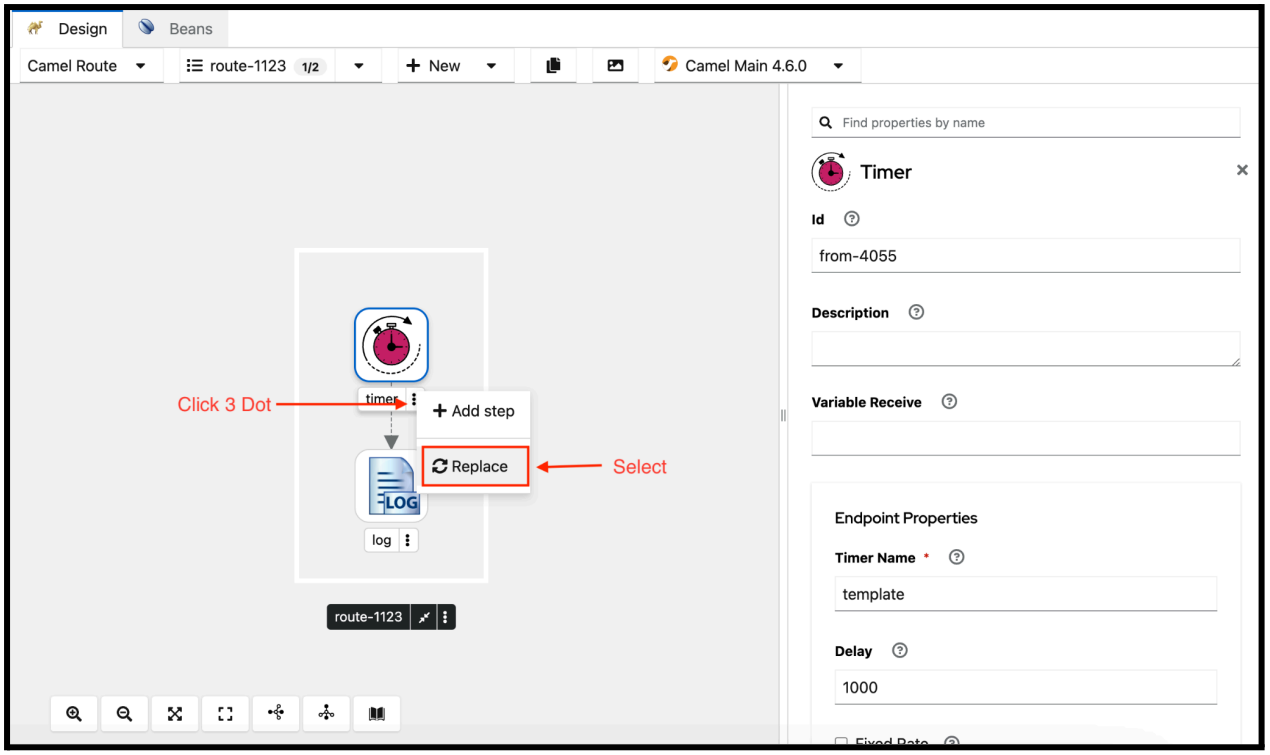


9 Module 5

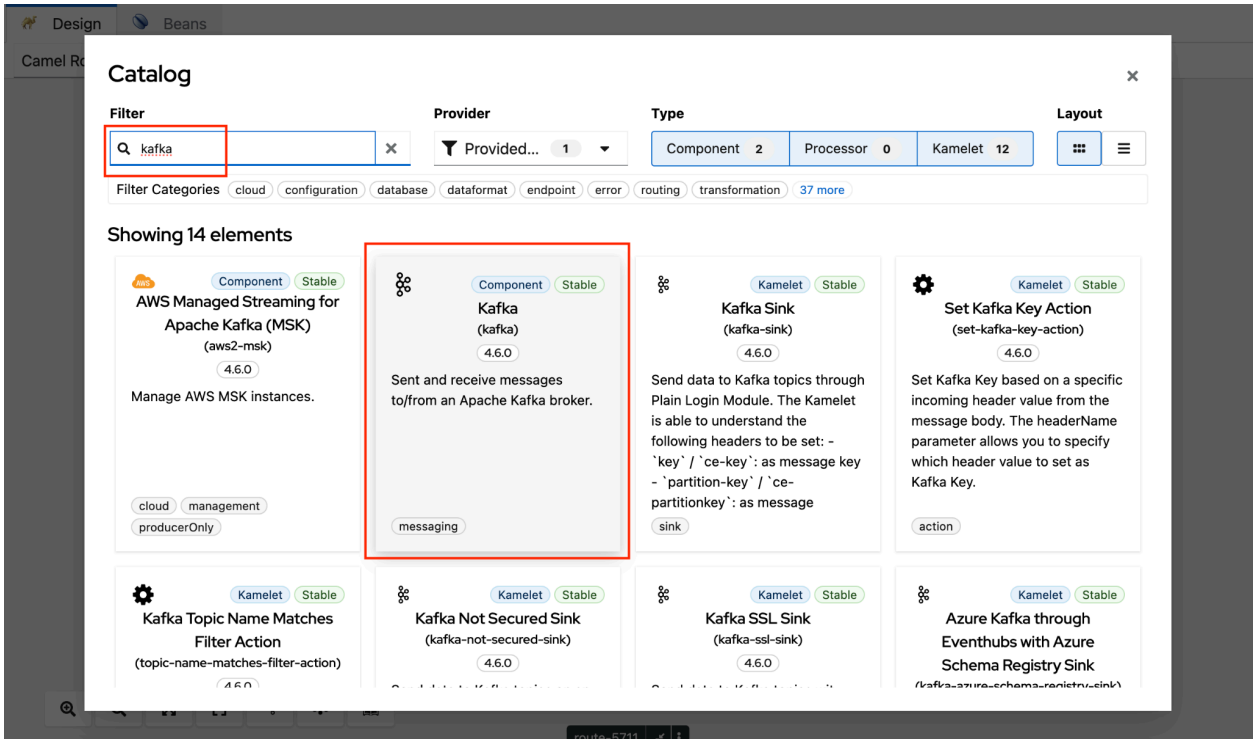
9.1. Create a new route



9.2. Replace Timer with Kafka component.

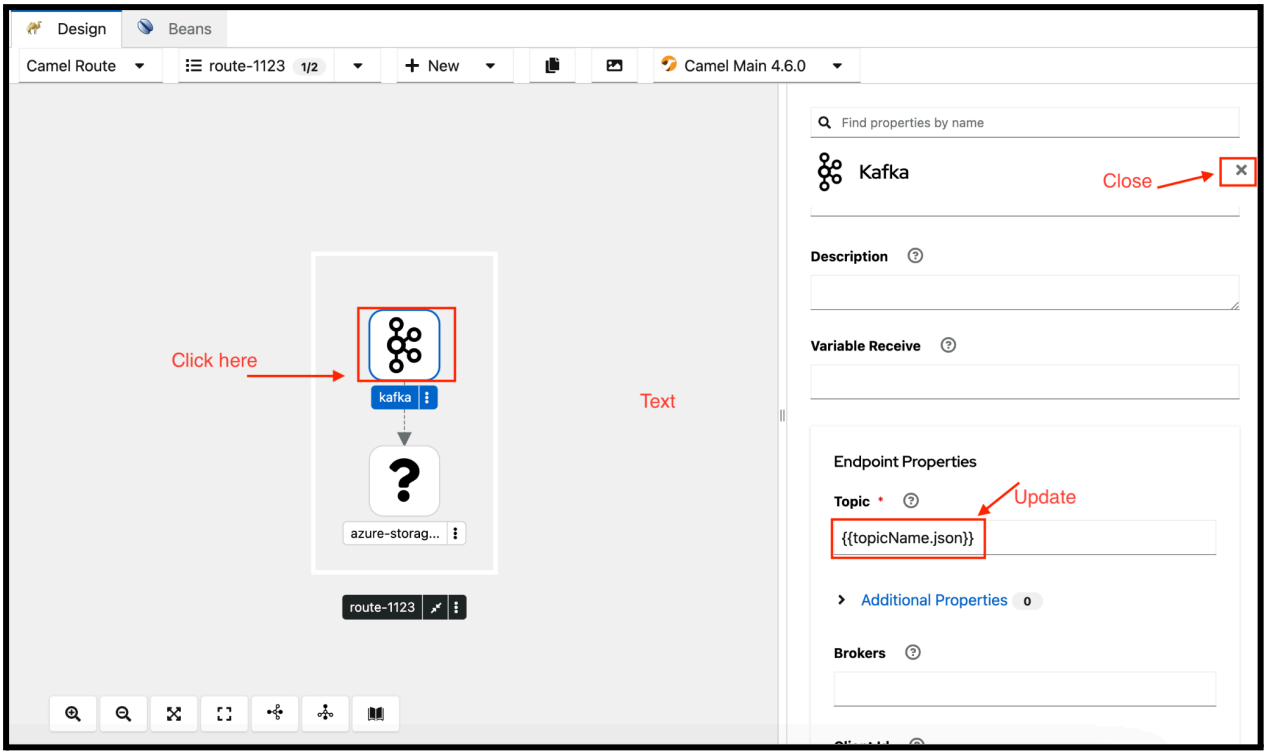


9.3.Select the kafka component from the catalog

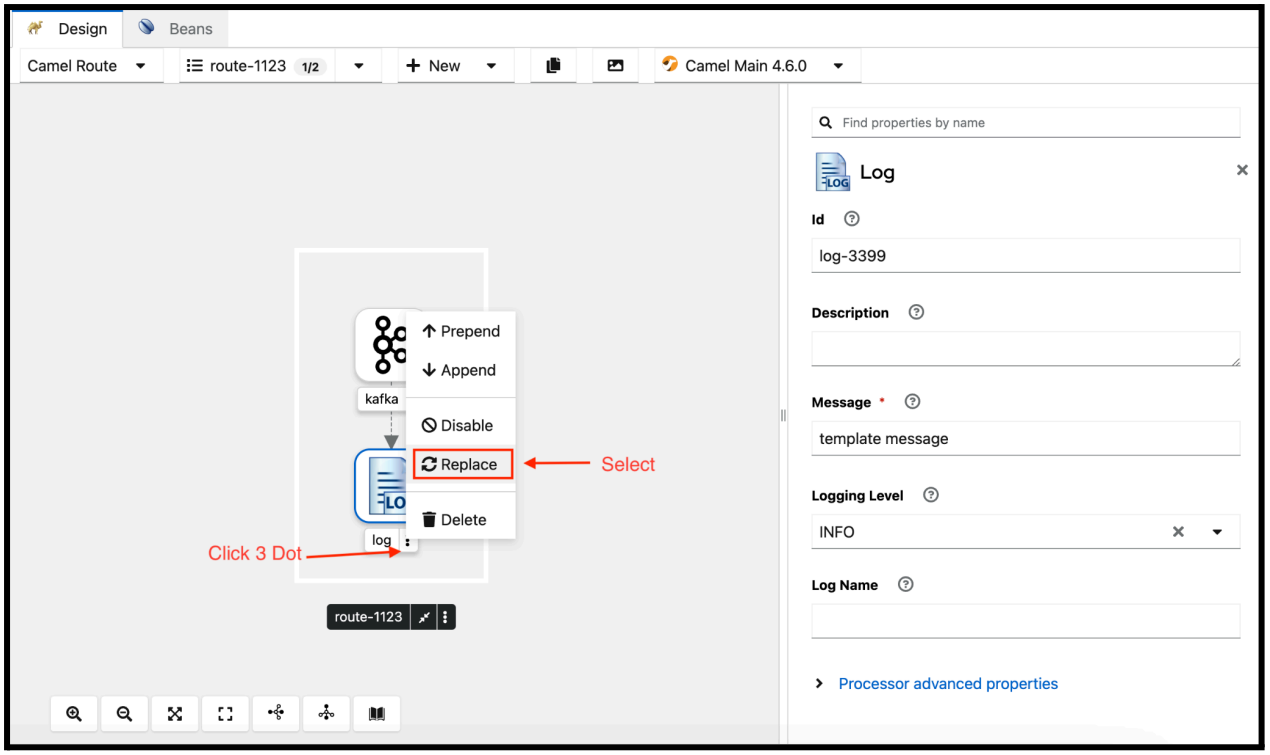


9.3.1 Update properties for the Kafka component and close

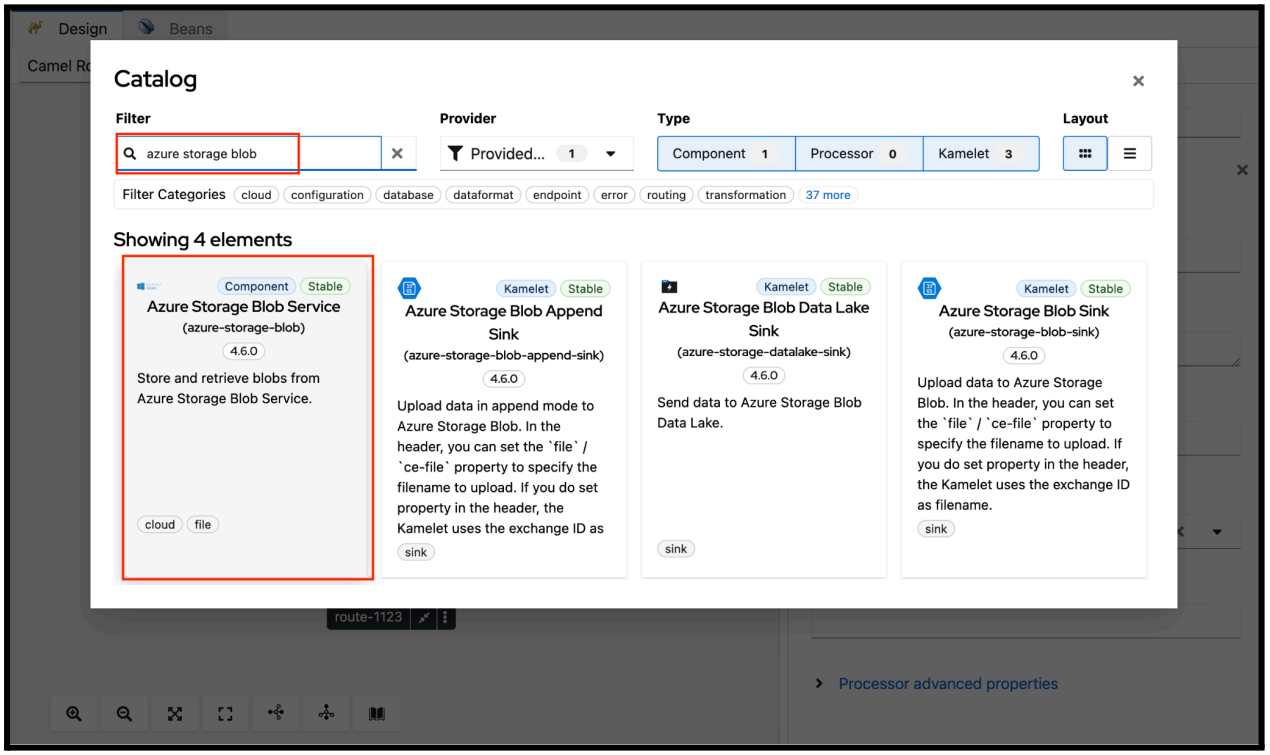
- Topic --> {{topicName.json}}



9.4. Replace Log with AzureBlob Storage component.

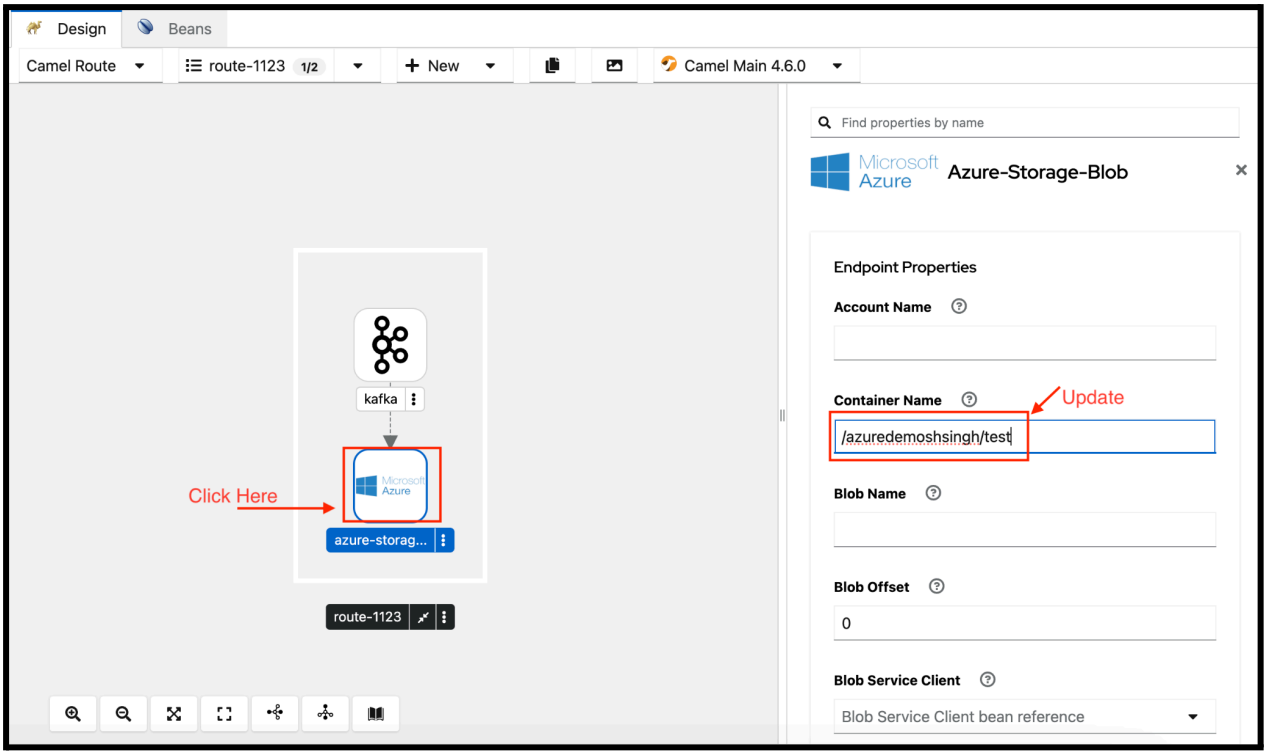


9.5. Select the AzureBlob Storage component from the catalog

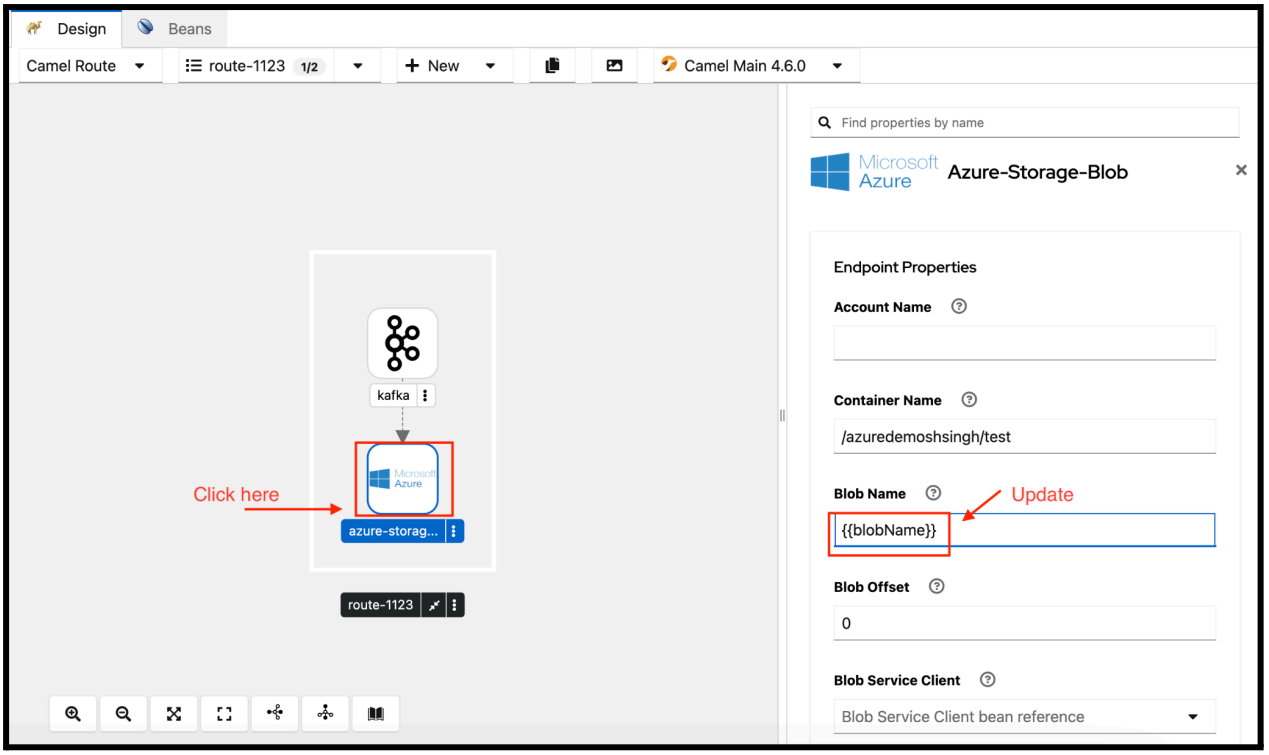


9.5.1 Update parameters for the azure-storage-blob component

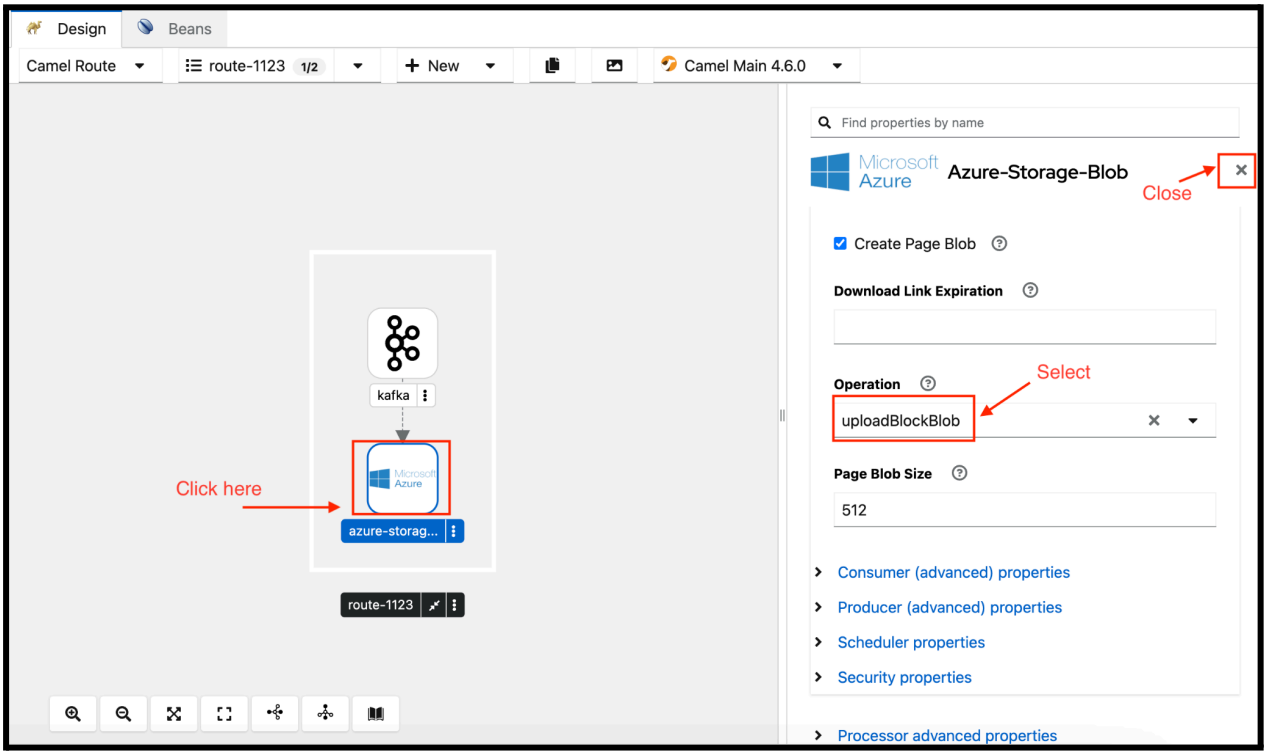
- Container Name ---> /azuredemoshsingh/test



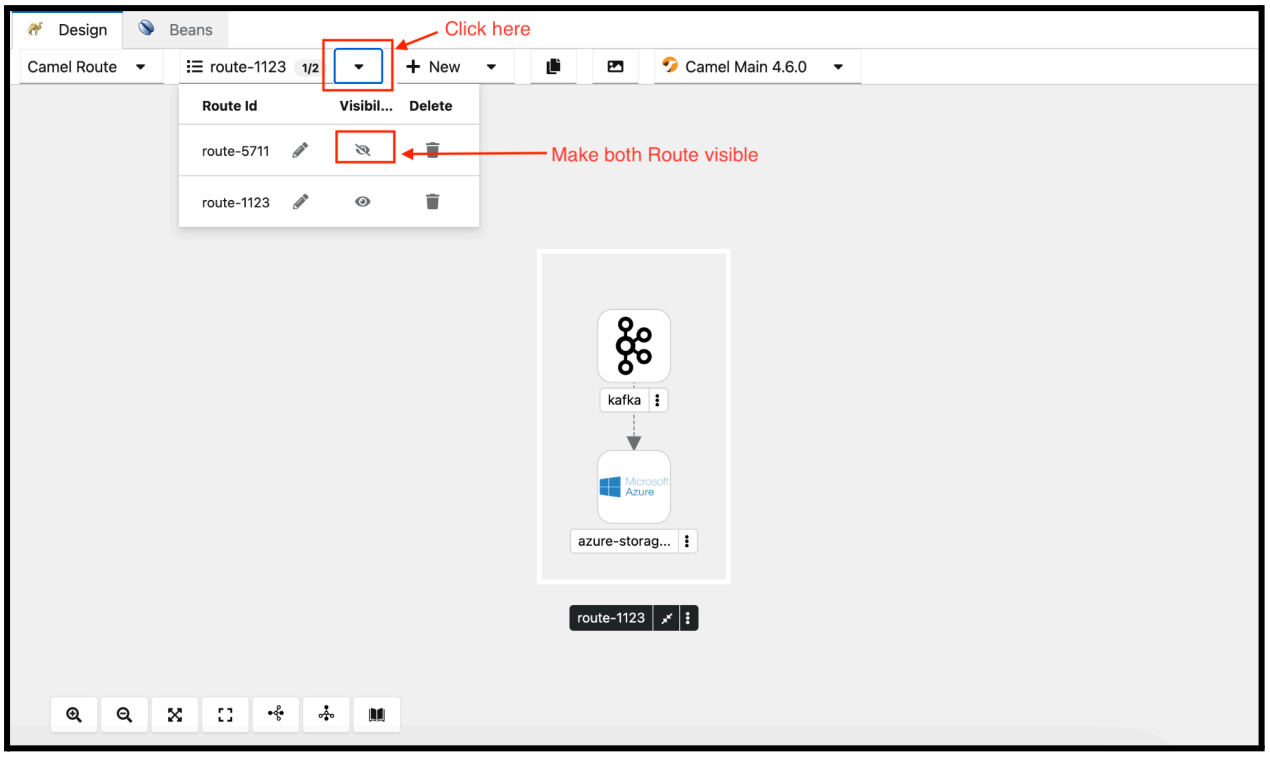
- Blob Name ---> {{blobName}}



- Operation ---> uploadBlockBlob



9.6. Select both the route



9.7 Commit the Changes

Follow Module1 steps

9.8 Verify the pipeline has been completed successfully.

Follow Module1 steps

9.9 Verify the Output

Azure Storage account access will be with the Instructor.

File with the name <projectName>-xml will be created on the azure storage. The screenshot below is for user2.

10. Update the DevSpace plugin

Each of the following sections provides a brief description and a list of recommended actions. Additional explanations are provided where additional clarity is required.

10.1. Login to the openshift console using admin credentials.

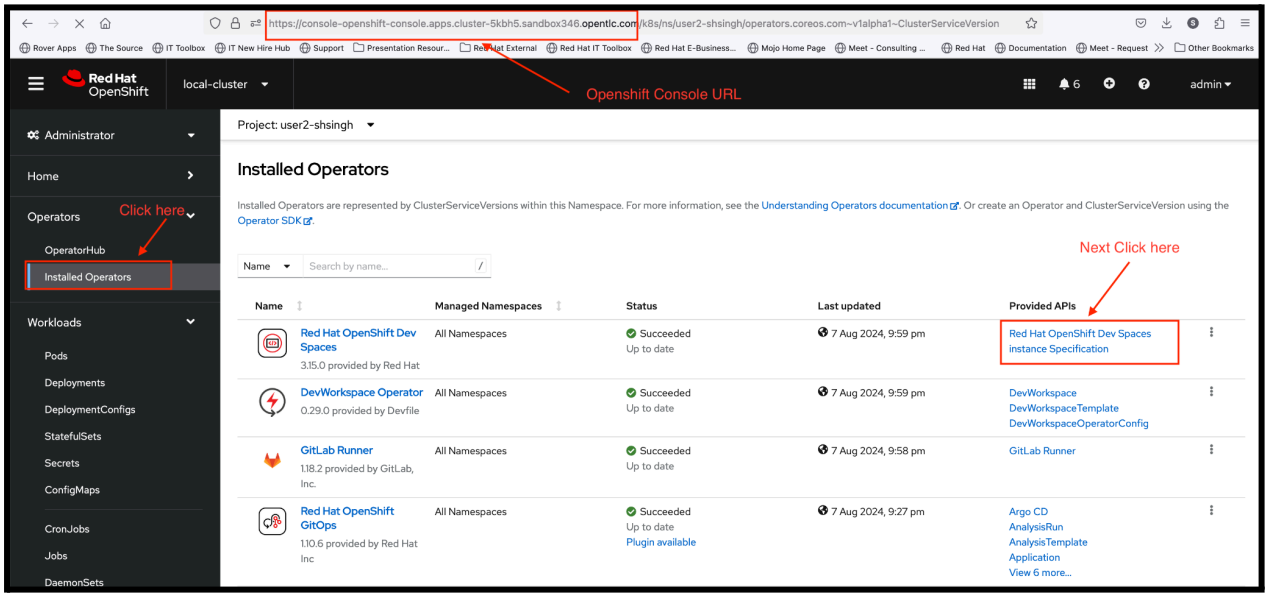
UserName:- admin

Password:- <SameAsUSer>

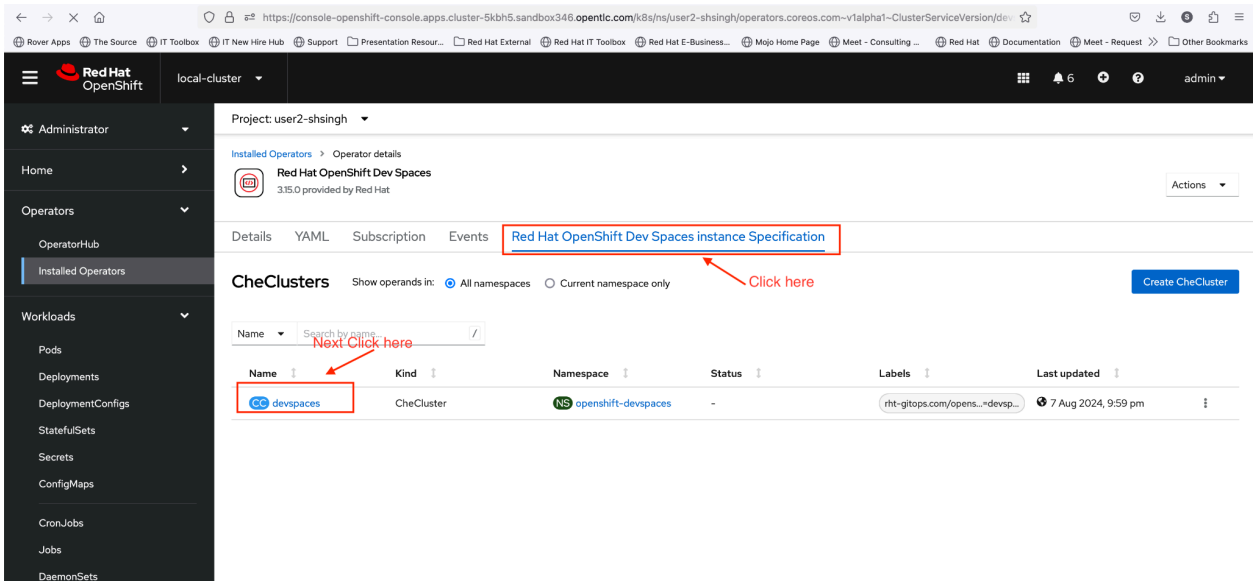
URL:- https://console-openshift-console.<CLUSTER-DOMAINNAME>

SAMPLE URL :- https://console-openshift-console.apps.cluster-5kbh5.sandbox346.opentlc.com

10.2. Go to the Installed OperatorPage and look for Red Hat OpenShift Dev Spaces



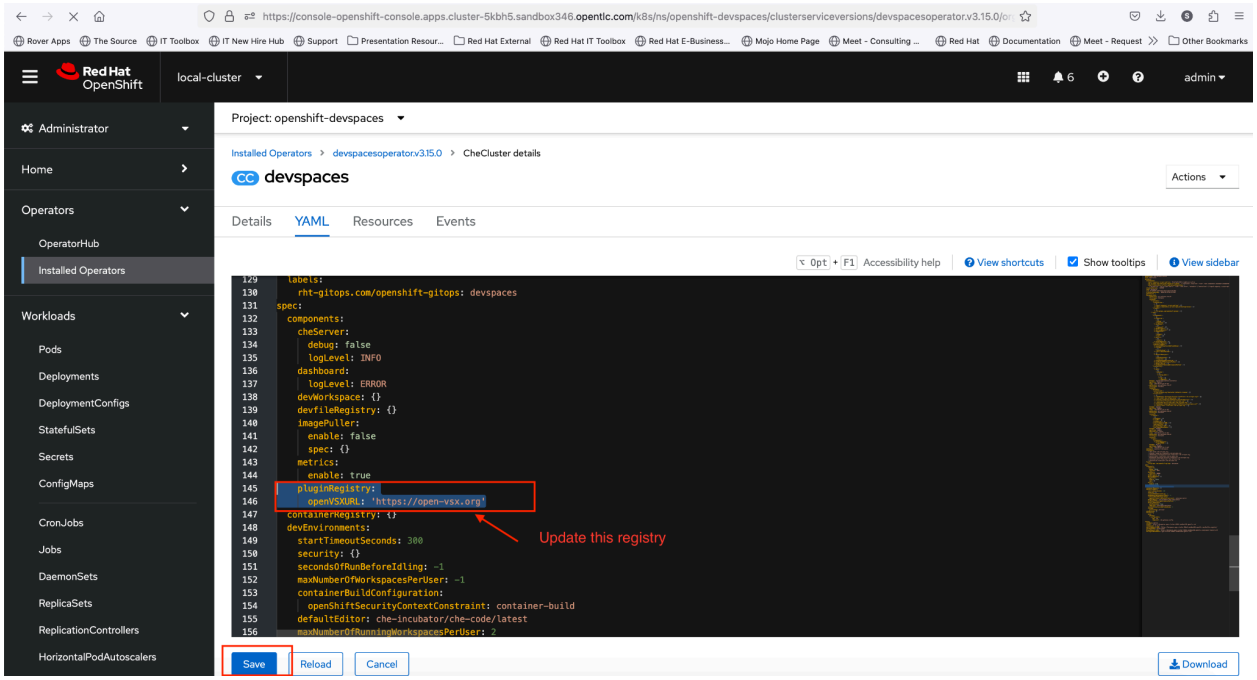
10.3 Update the devspace cheCluster Yaml configurations.



10.4. Update the pluginRegistry configurations.

```
oc patch checluster devspaces --type=merge -p '{"spec": {"components": {"pluginRegistry": {"openVSXURL": "https://open-vsx.org"}}}}' -n openshift-devspaces
```

```
pluginRegistry:
 openVSXURL: 'https://open-vsx.org'
```



10.5. Save the configuration and verify the changes after reload.