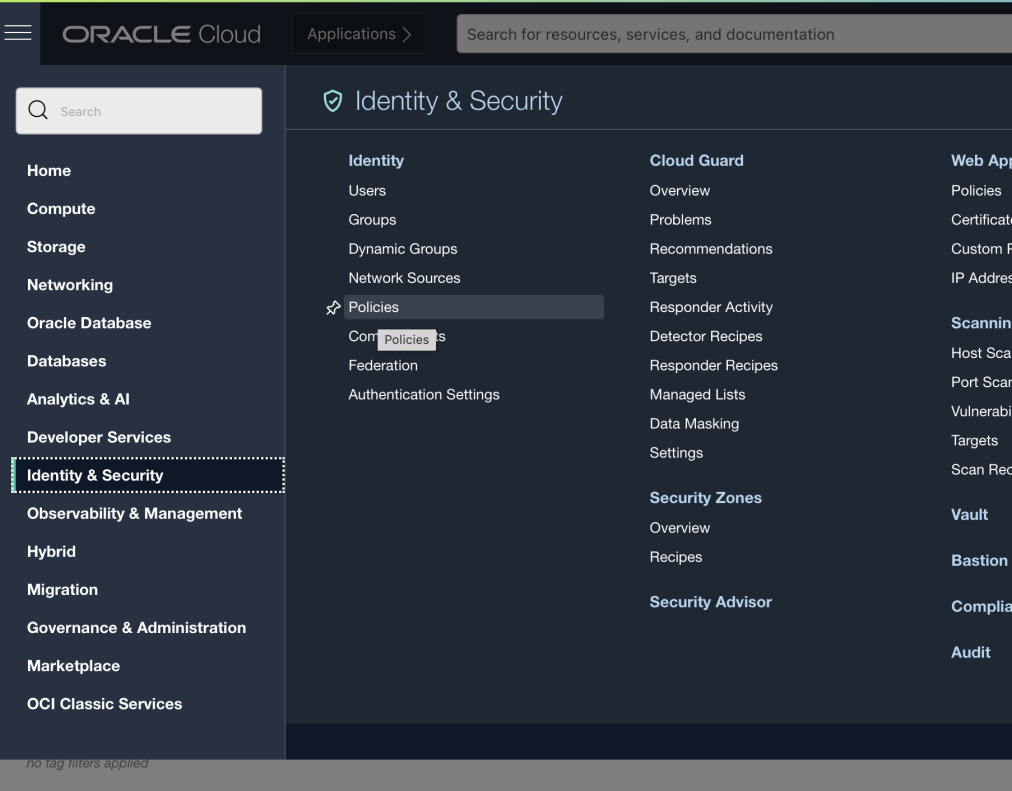
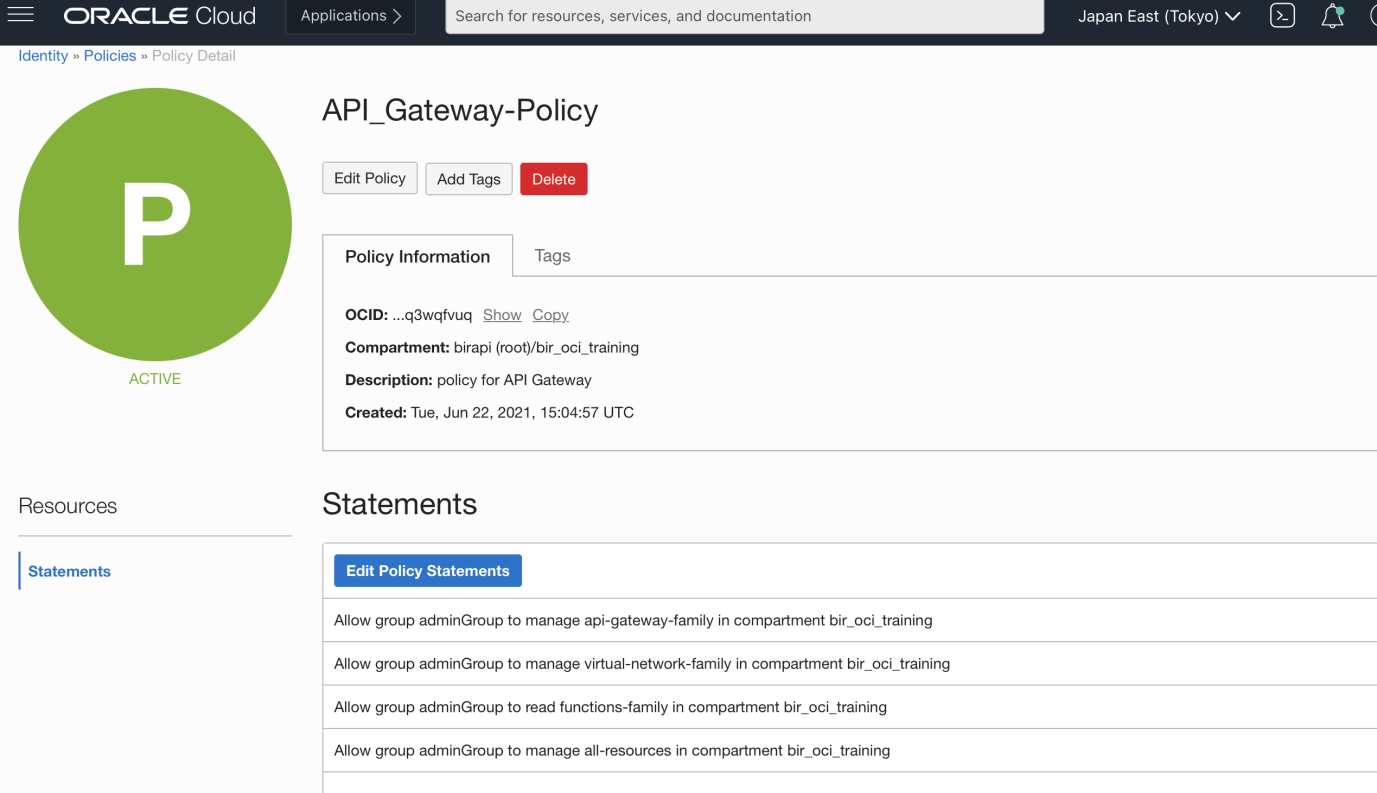
**Oracle API Gateway**

**Permissions required to enable API Gateway:**

1. Navigate to Policies in OCI Console.



1. Add new policies as mentioned below.



**Statements example**:

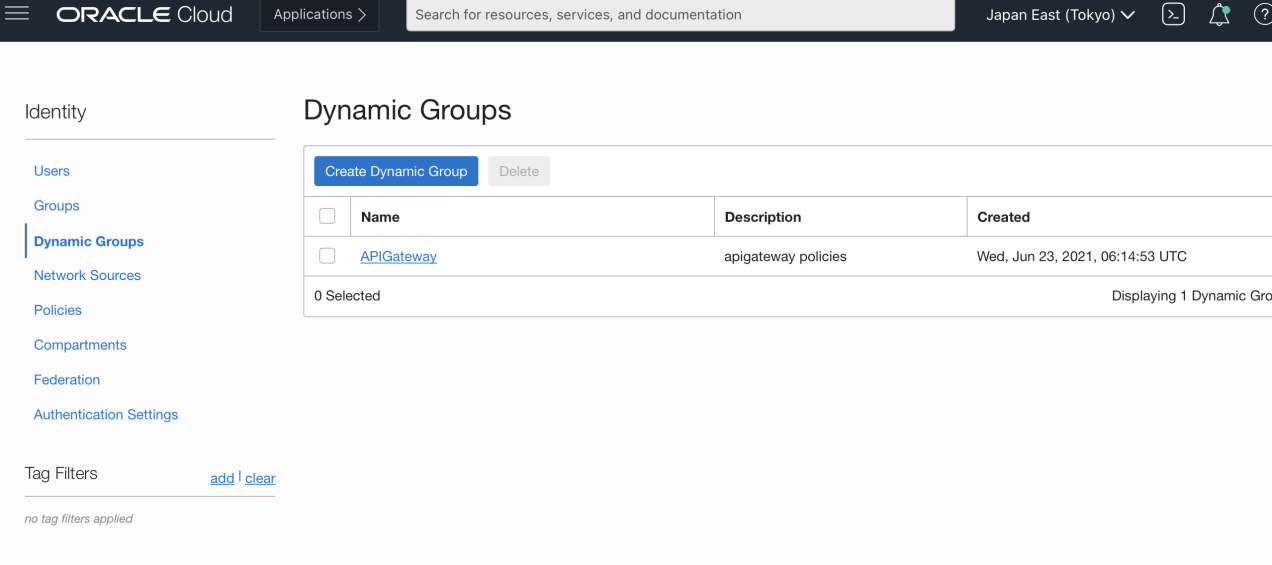
Allow group adminGroup to manage api-gateway-family in compartment bir\_oci\_training

Allow group adminGroup to manage virtual-network-family in compartment bir\_oci\_training (access to VCN)

Allow group adminGroup to read functions-family in compartment bir\_oci\_training (This is to enable functions)

Allow group adminGroup to manage all-resources in compartment bir\_oci\_training

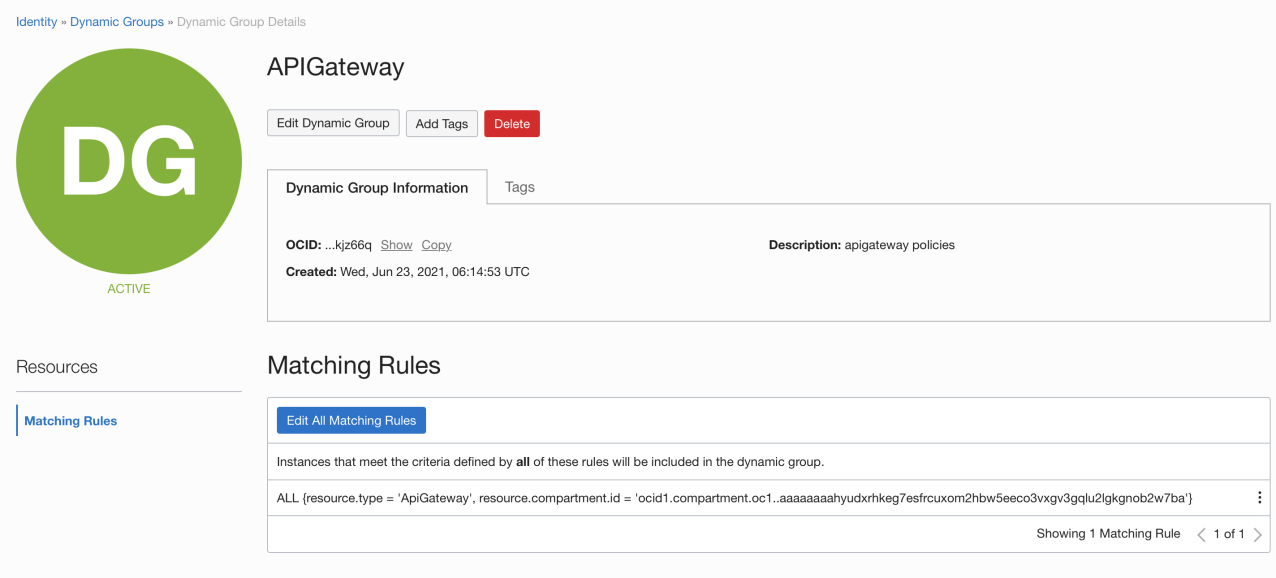
1. Next go to Dynamic groups and add below statement.



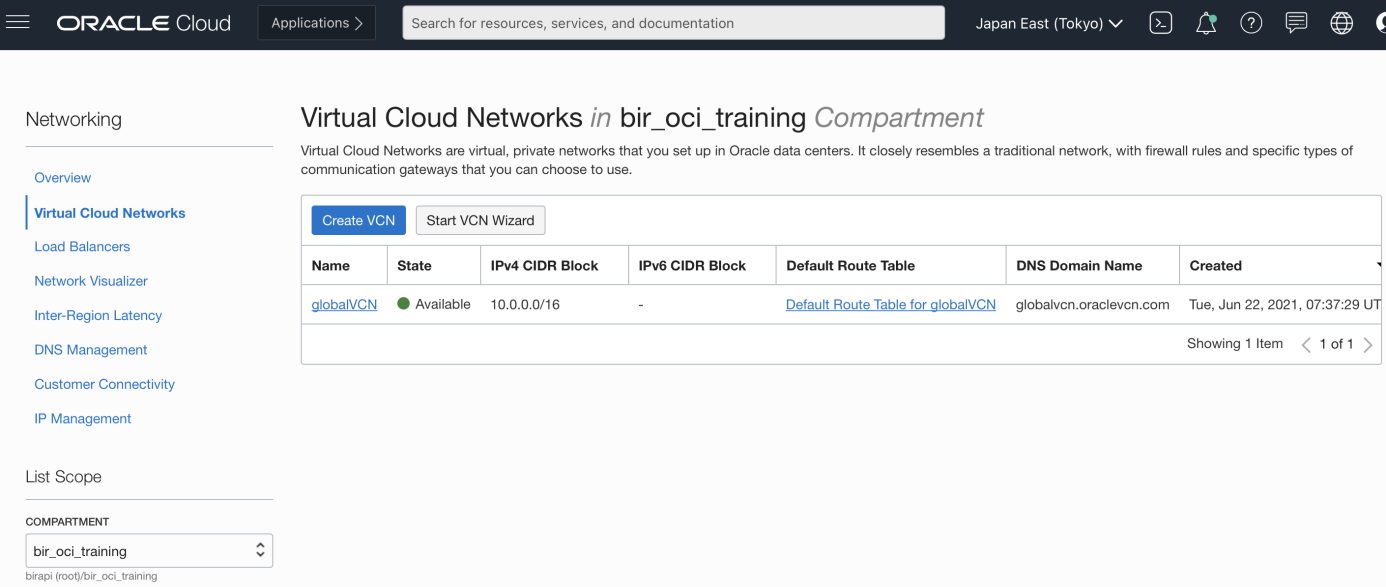
Under matching Rules add the statement.

**Like:**

ALL {resource.type = 'ApiGateway', resource.compartment.id = 'ocid1.compartment.oc1..aaaaaaaahyudxrhkeg7esfrcuxom2hbw5eeco3vxgv3gqlu2lgkgnob2w7ba'}



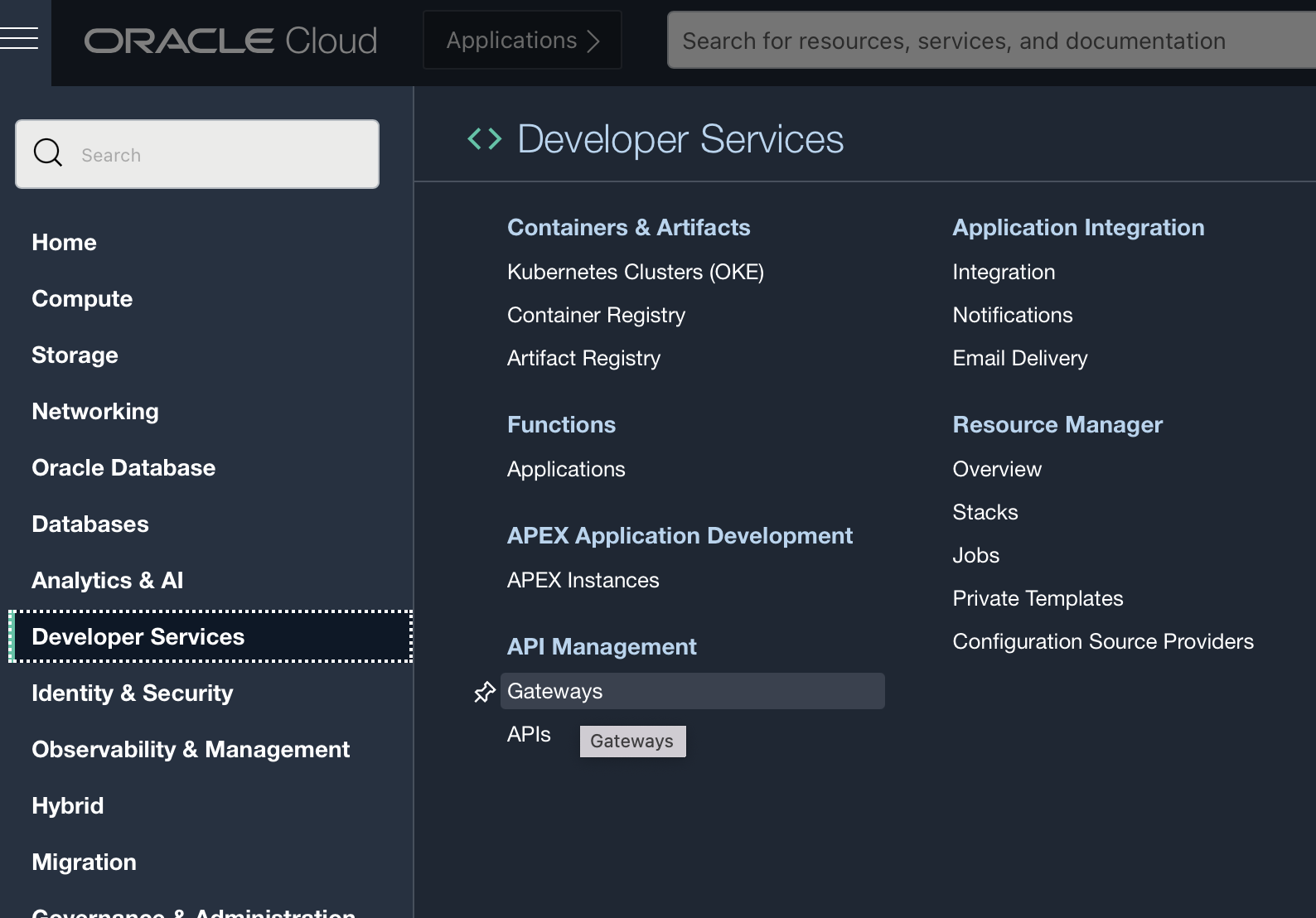
Make sure under Networking, you have a VCN created with Public and private subnets.



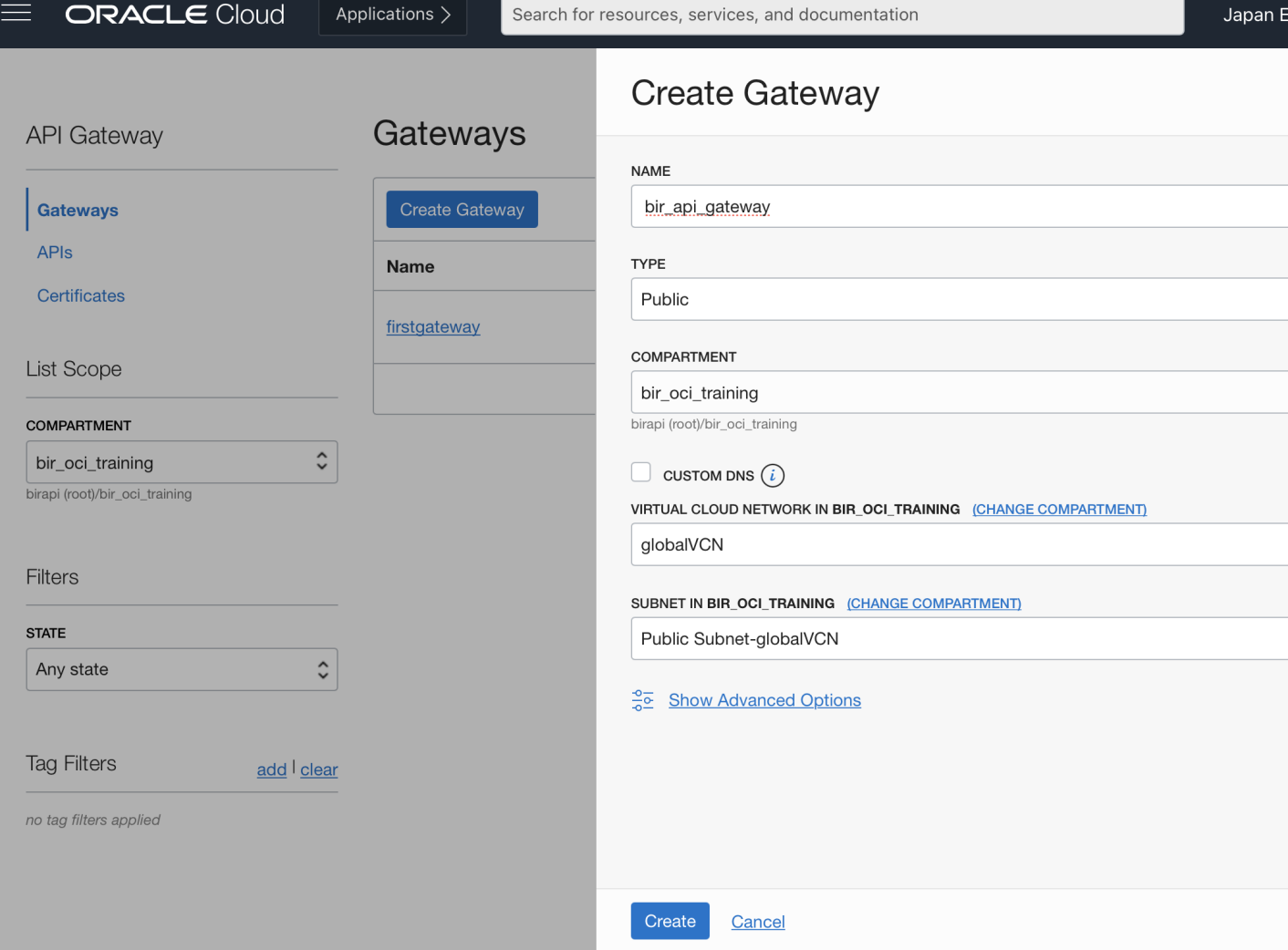
**LAB 1:**

**Go Create an API Gateway:**

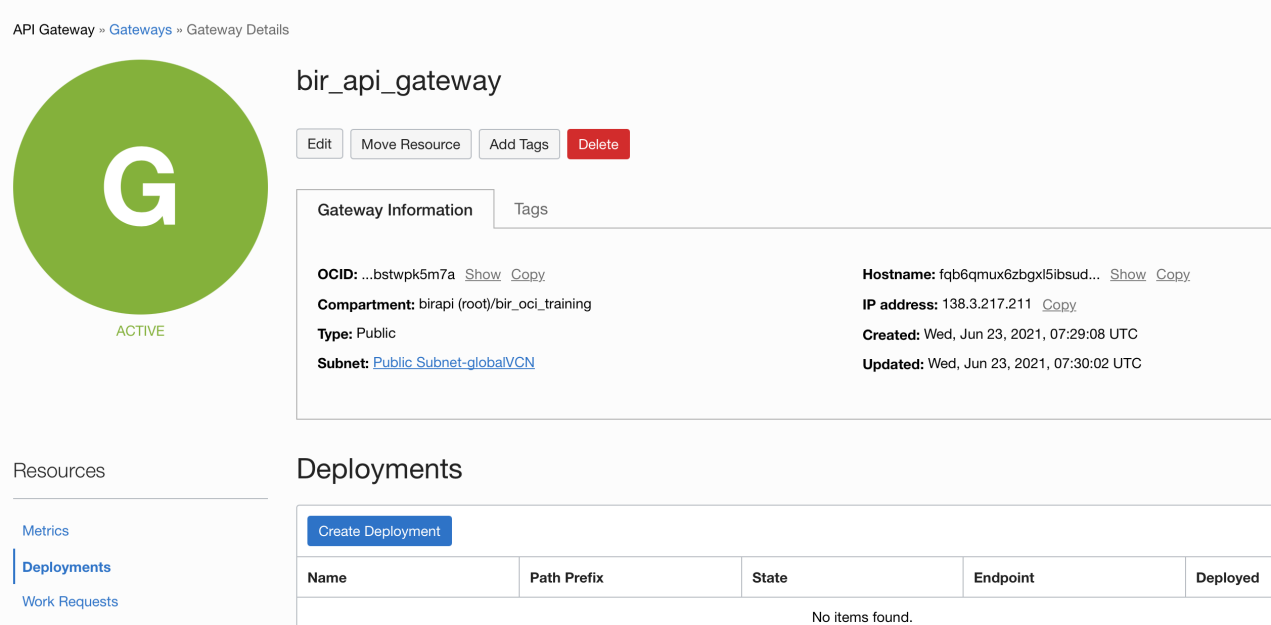
1. Now under Developer services navigate to gateways.



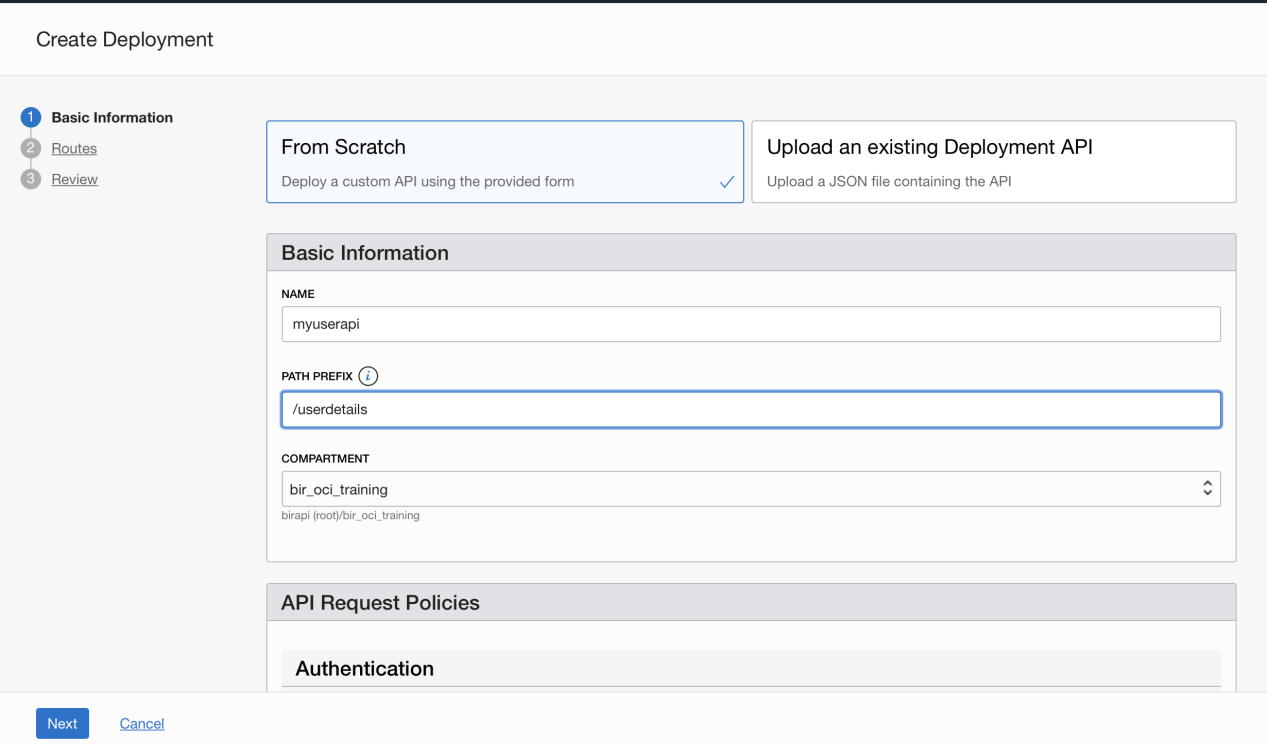
1. Click on Create Gateway and fill in the required details. Make sure to select a public subnet in this case.



1. After we click on Create a new API Gateway will be created. This may take a couple of minutes.



7. The image icon will turn Green in color. Now click on Create Deployment and fill in the details.



Please note there are two ways to create a gateway. We can configure it manually or use an “Existing Deployment API”.

In a typical scenario a API deployment file would look like this: Sample file.

{

"routes": [

{

"path": "/hello",

"methods": ["GET"],

"backend": {

"type": "ORACLE\_FUNCTIONS\_BACKEND",

"functionId": "ocid1.fnfunc.oc1.phx.aaaaaaaaab\_\_\_\_\_\_xmq"

}

},

{

"path": "/weather",

"methods": ["GET"],

"backend": {

"type": "HTTP\_BACKEND",

"url": "https://api.weather.gov"

}

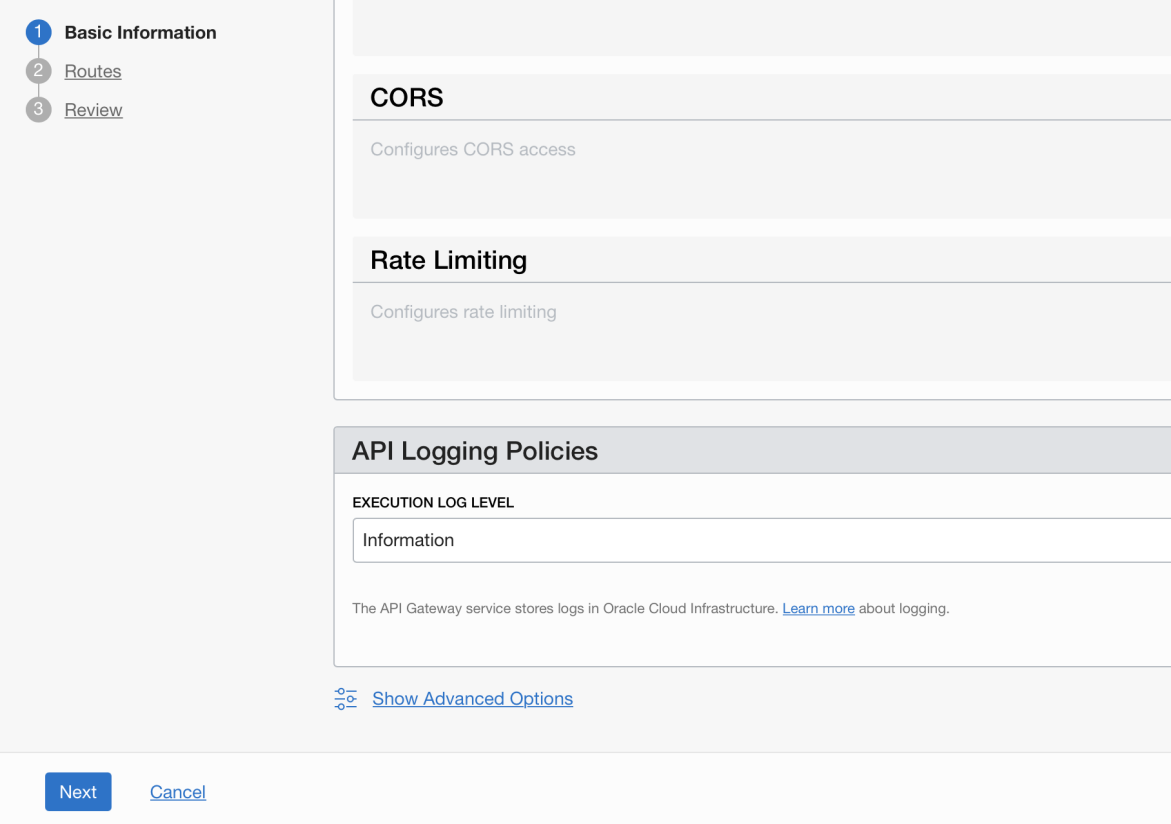
}

]

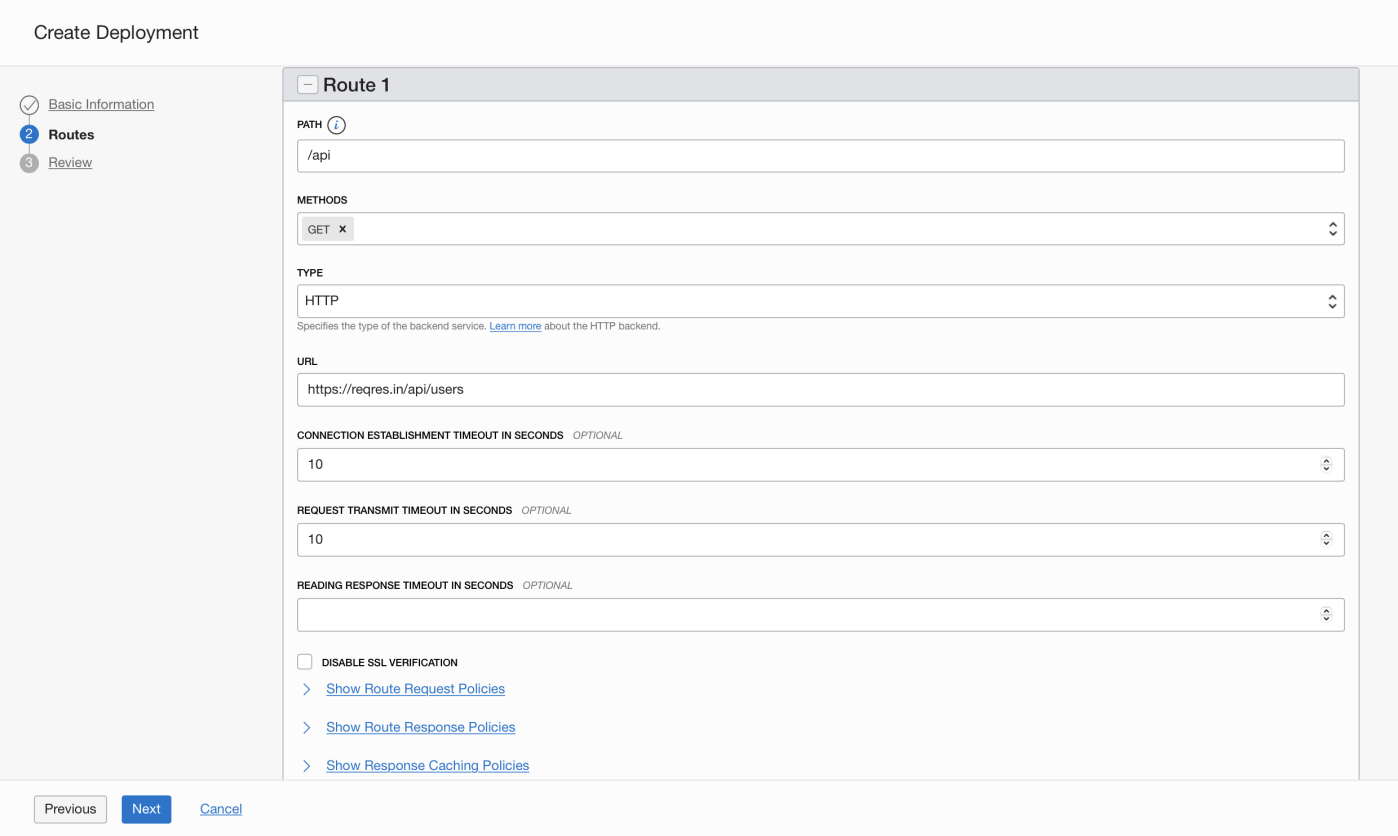
}

For our case we will create an API from scratch.

Don’t fill anything else and click on Next.



1. Now we are creating a Deployment. So fill in all the following details.



Here we can provide any path name we want.

Select method as GET and type as HTTP.

URL: <https://reqres.in/api/users>

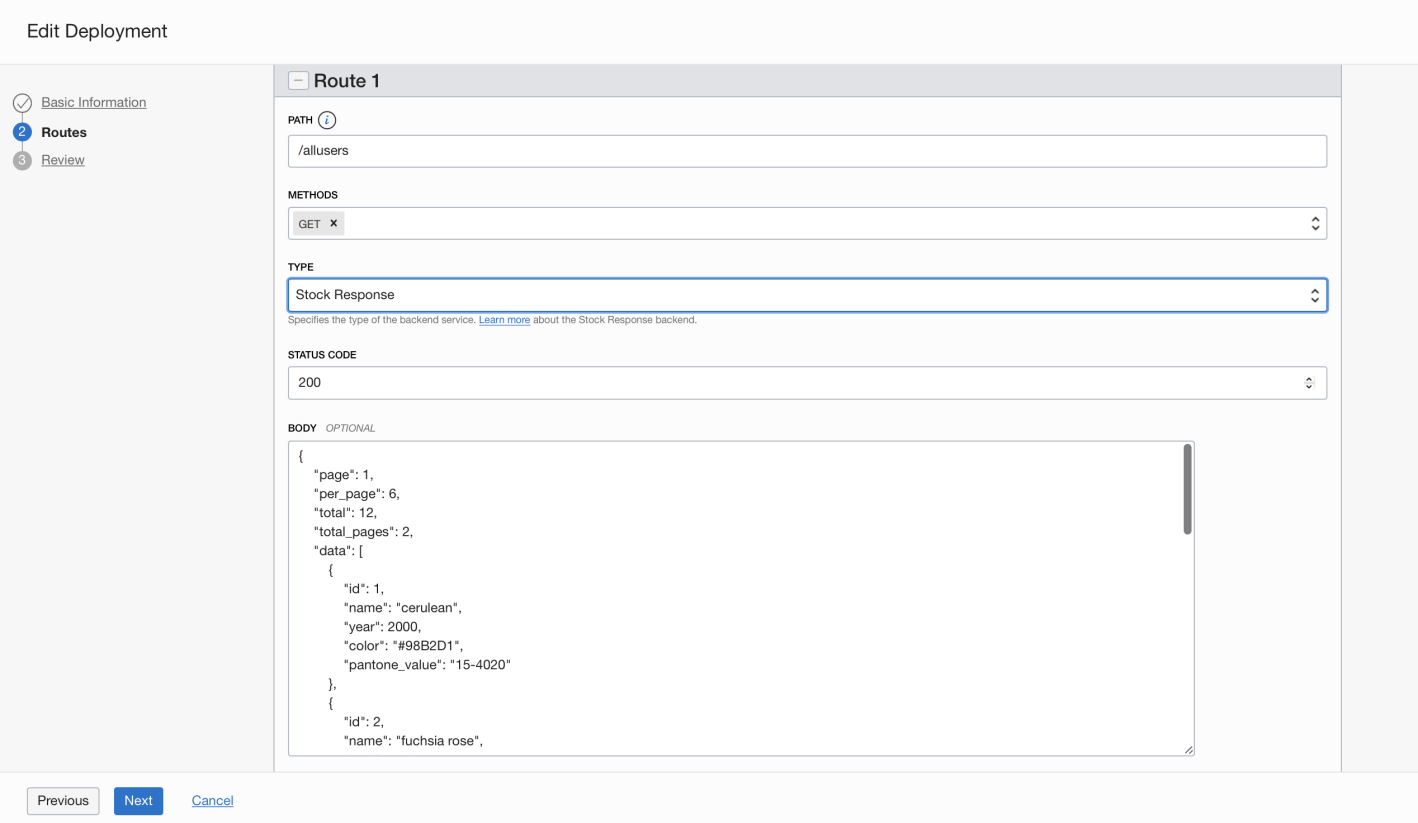
Note: This URL we are using is basically a sample URL that returns a list of users.

Set the connection timeout values and click on Next.

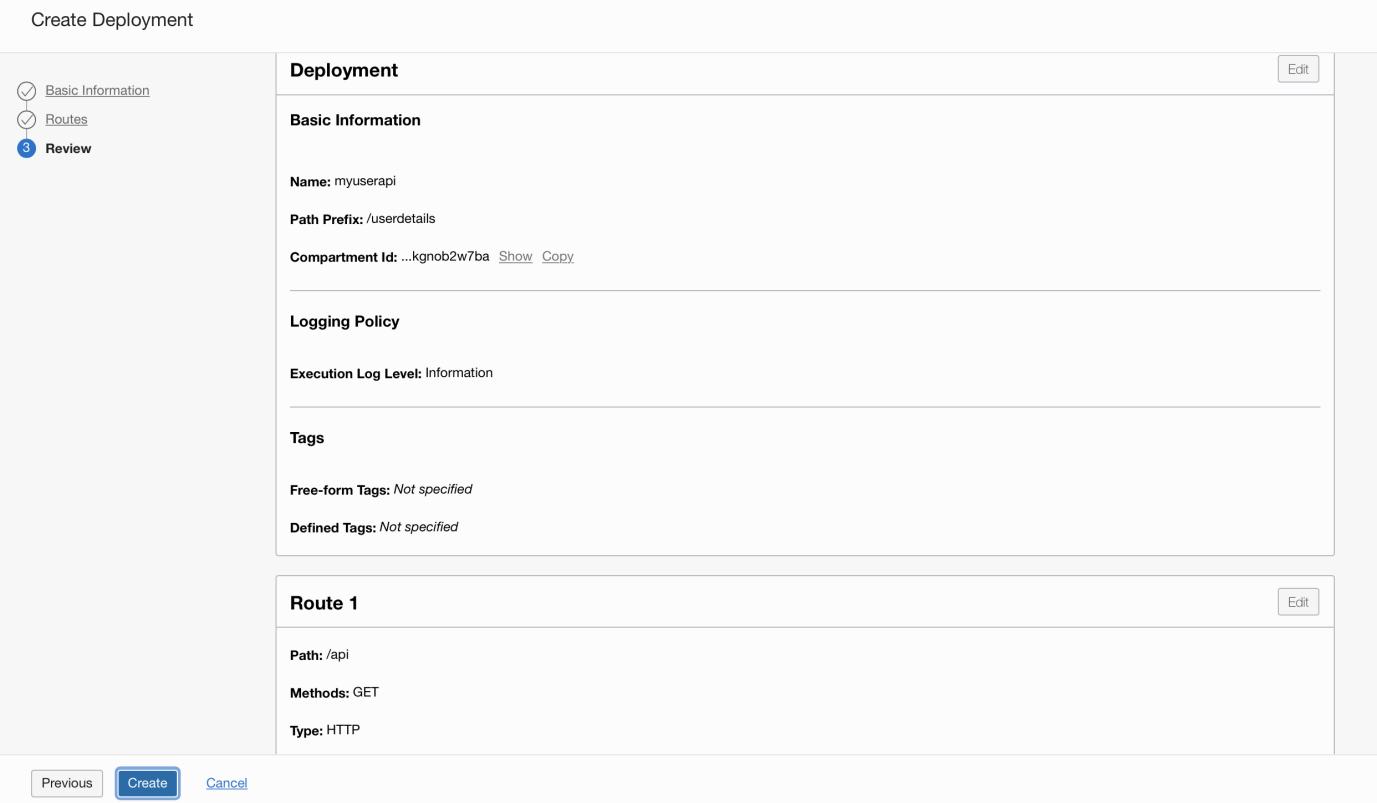
**Please note:** We can also select an option like “Stock Response” create a JSON response of our choice.

Here the JSON is used in the body.

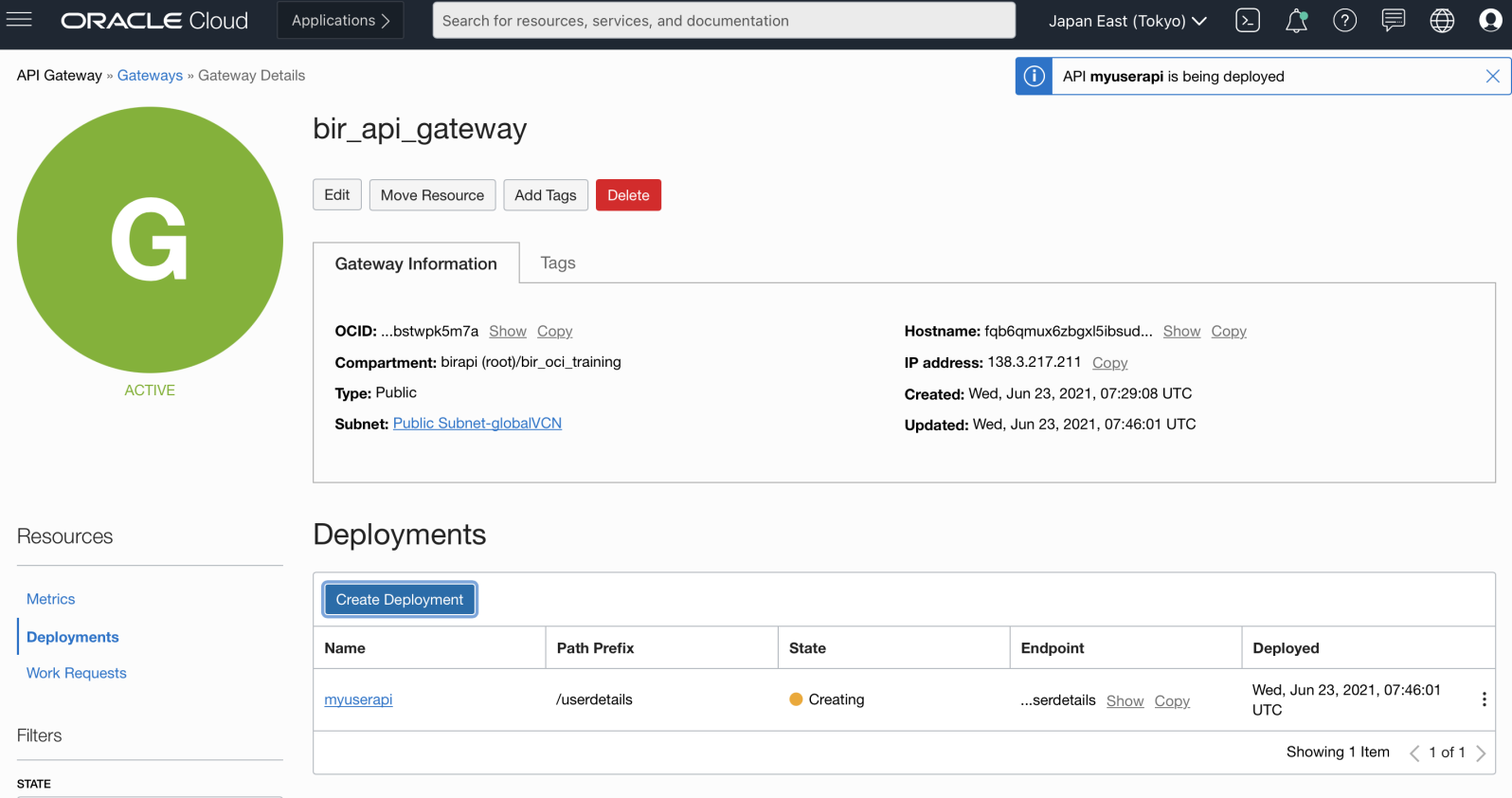
See below screenshot.



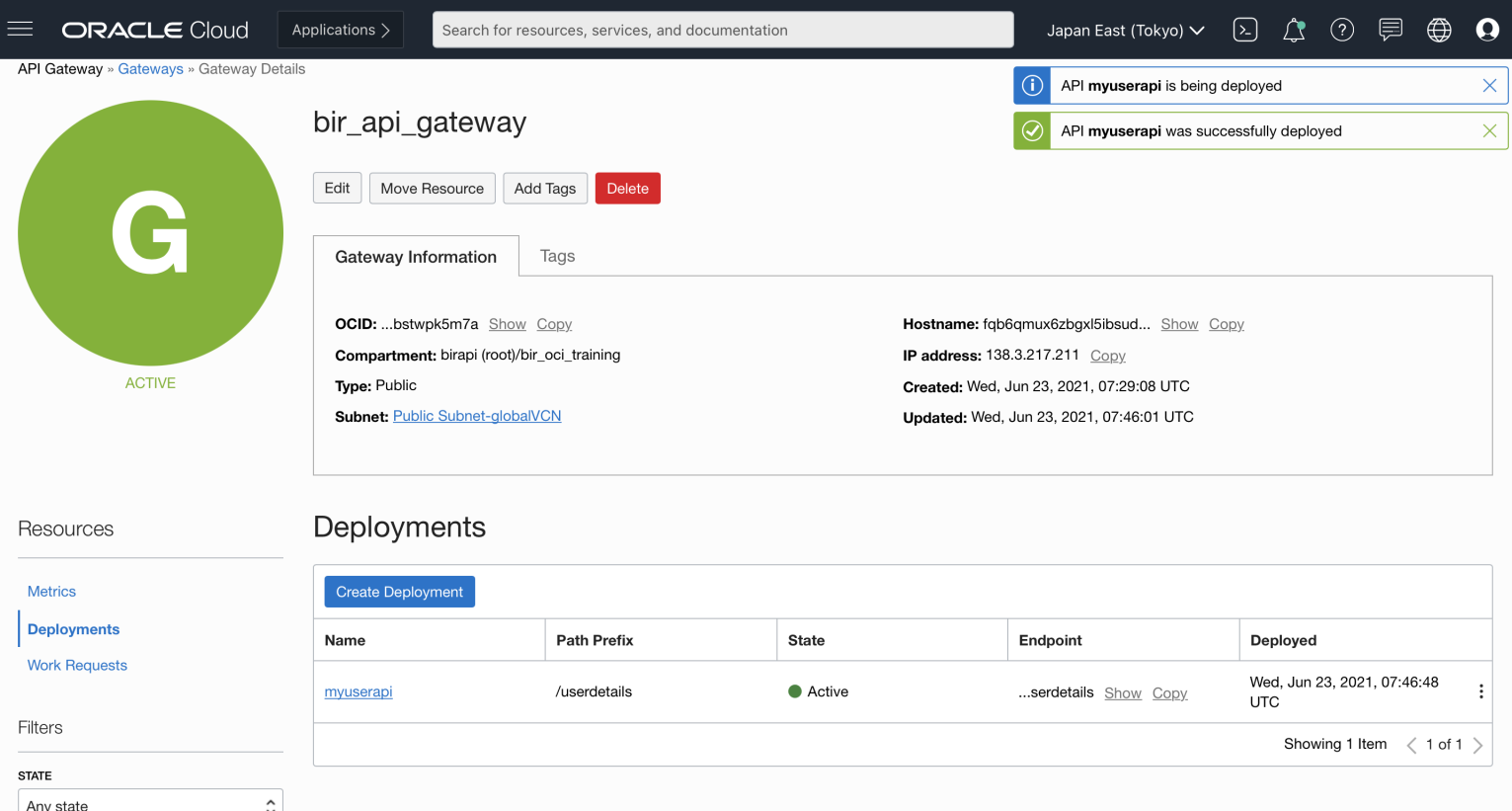
1. On the next screen verify all details and click on “Create”.



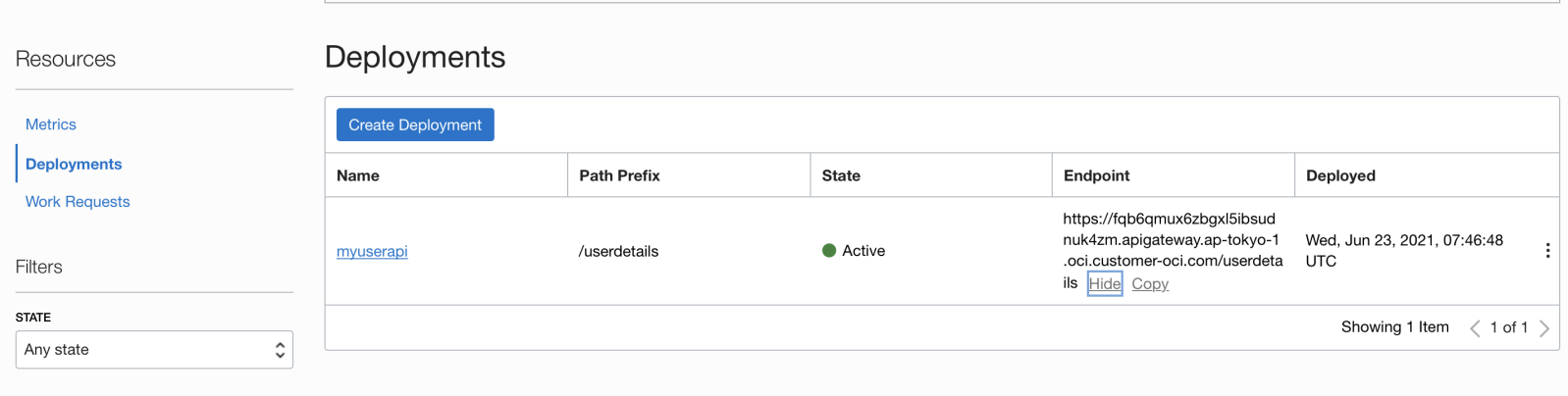
1. Now it will take a couple of minutes for the api to get created.



Once created we should get a confirmation like below.

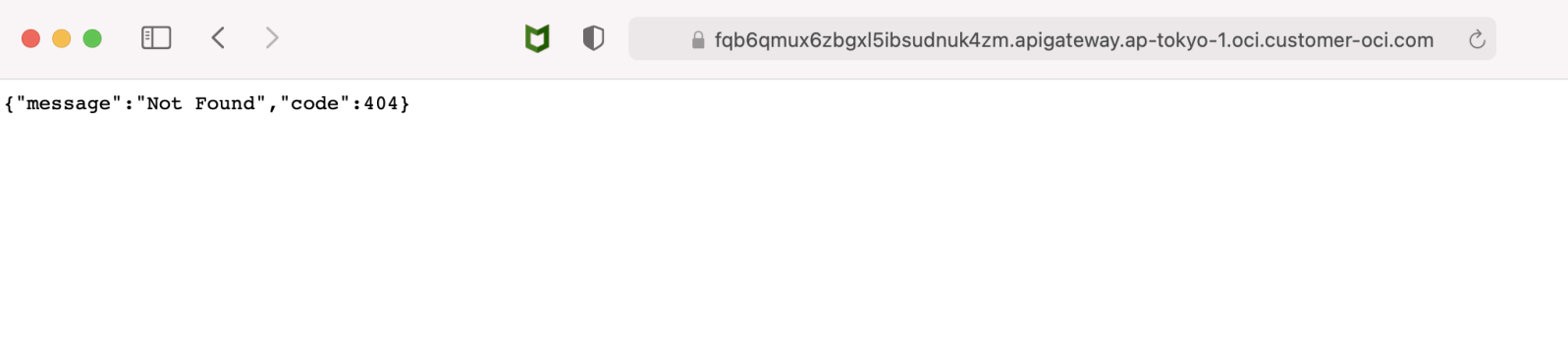


11. Click on the show button below “Endpoint” to show and then copy the endpoint url.



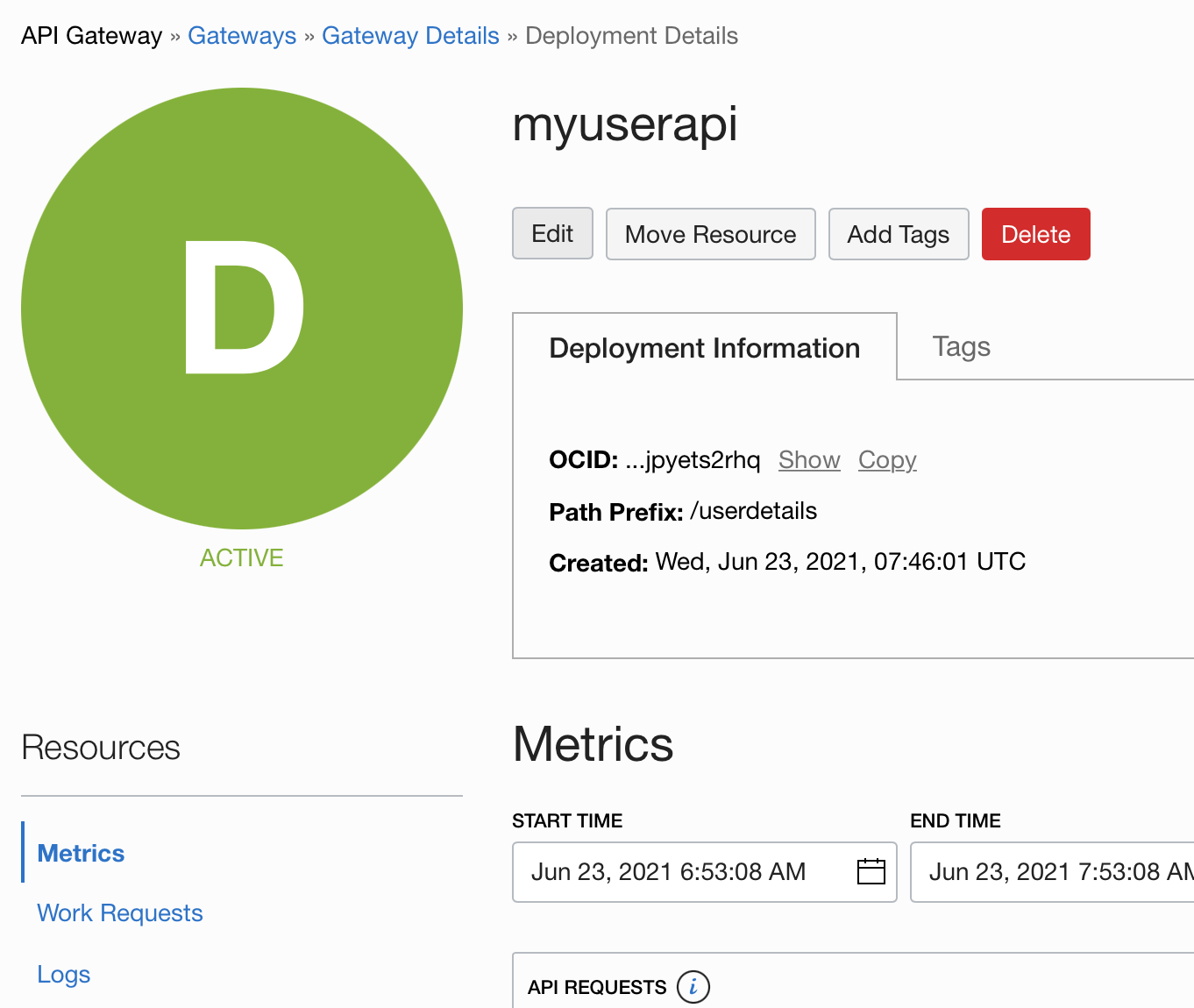
1. Now once the url is copied we can go to a browser and paste the url.

Eg: <https://fqb6qmux6zbgxl5ibsudnuk4zm.apigateway.ap-tokyo-1.oci.customer-oci.com/userdetails>



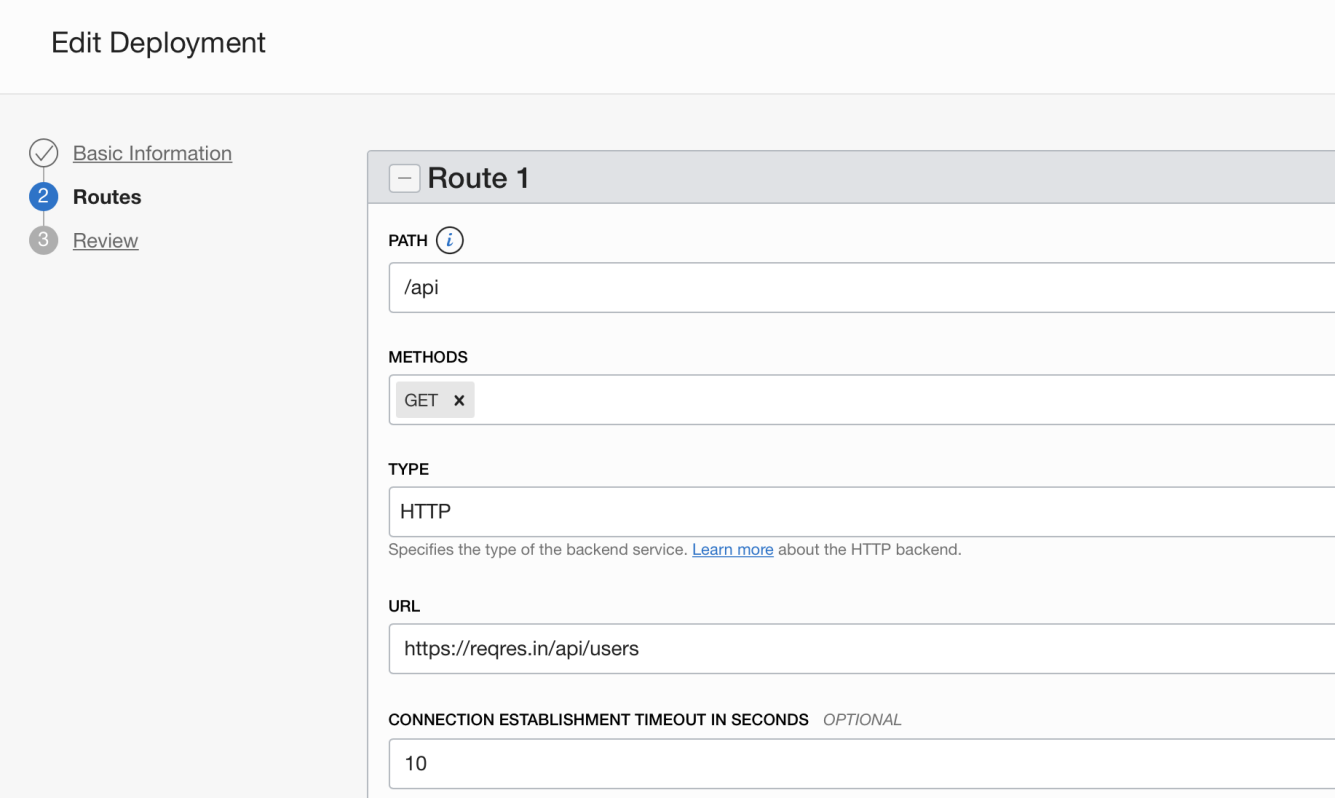
Once we paste the url we see that we get a “Not Found” error. We get this error because we have not defined the complete path for our url.

1. Navigate back to the api “myuserapi” and click on “Edit” option.



1. Look for the path name that we had entered earlier. Eg: /api

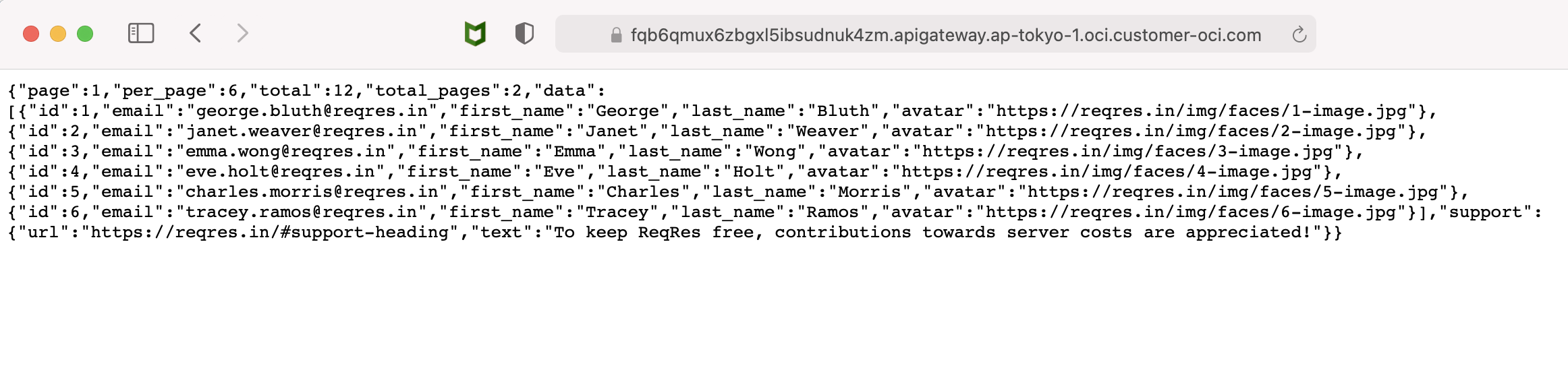
Basically this is the endpoint path that we need to enter after the url.



1. Now add this path at the end of the existing url and paste in a browser.

Eg: <https://fqb6qmux6zbgxl5ibsudnuk4zm.apigateway.ap-tokyo-1.oci.customer-oci.com/userdetails/api>

We should be able to see a response.

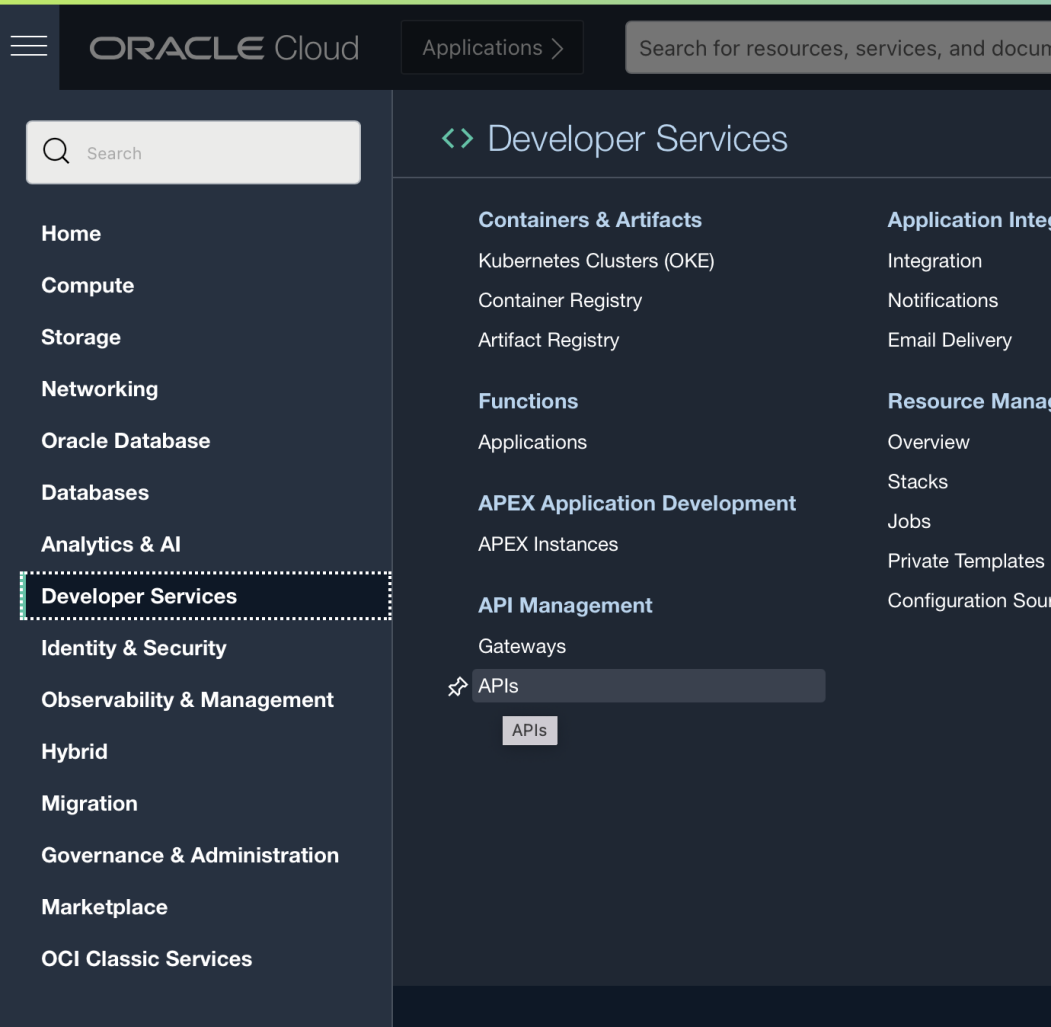


This completes our hands on lab 1 for Oracle API Gateway.

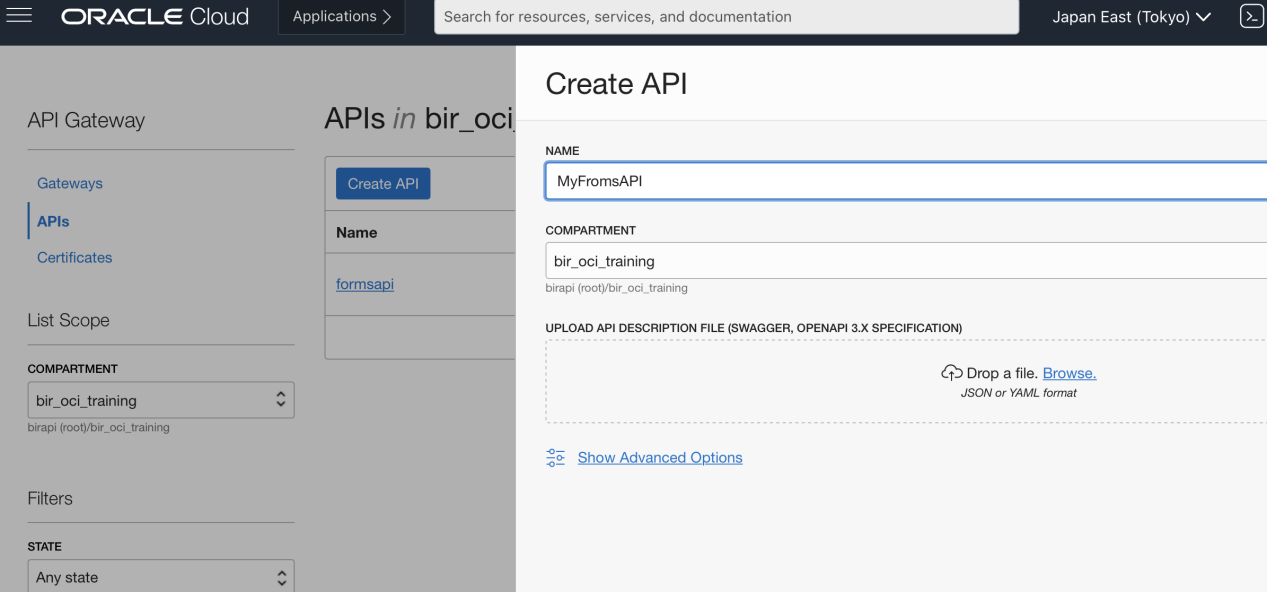
**LAB 2:**

**Create a API using a Swagger file/open API 2.0**

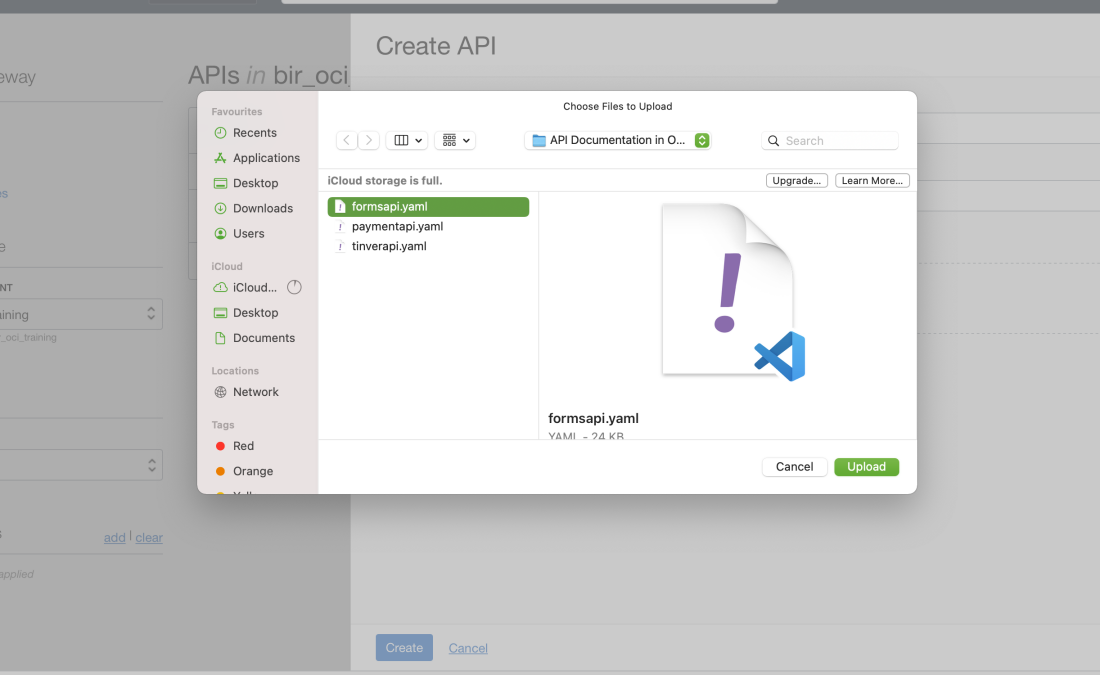
First we navigate to API in OCI console.

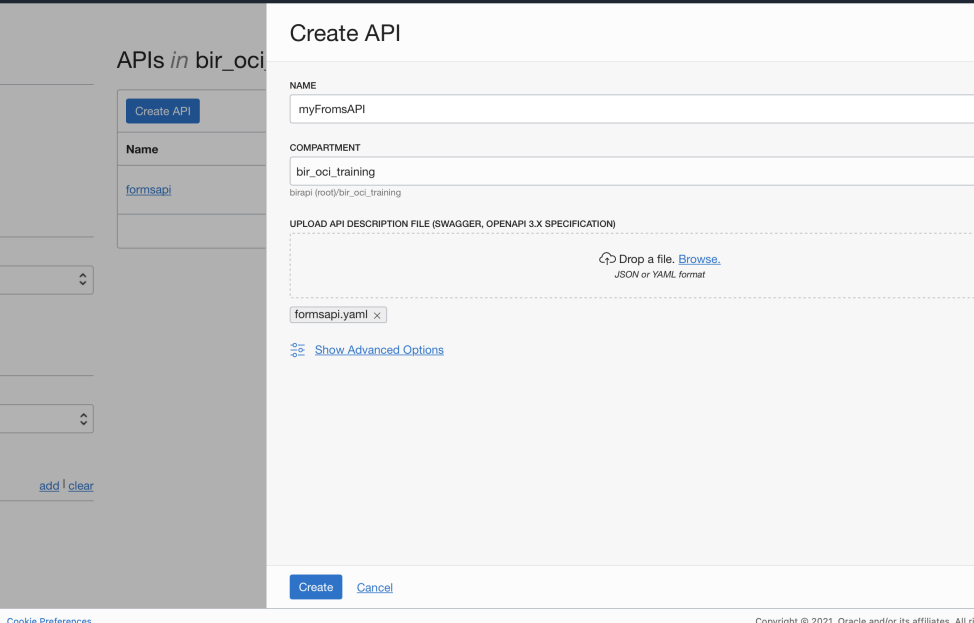


In the API page, click on “Create API”. This would launch a screen to enter name and upload your Swagger file.

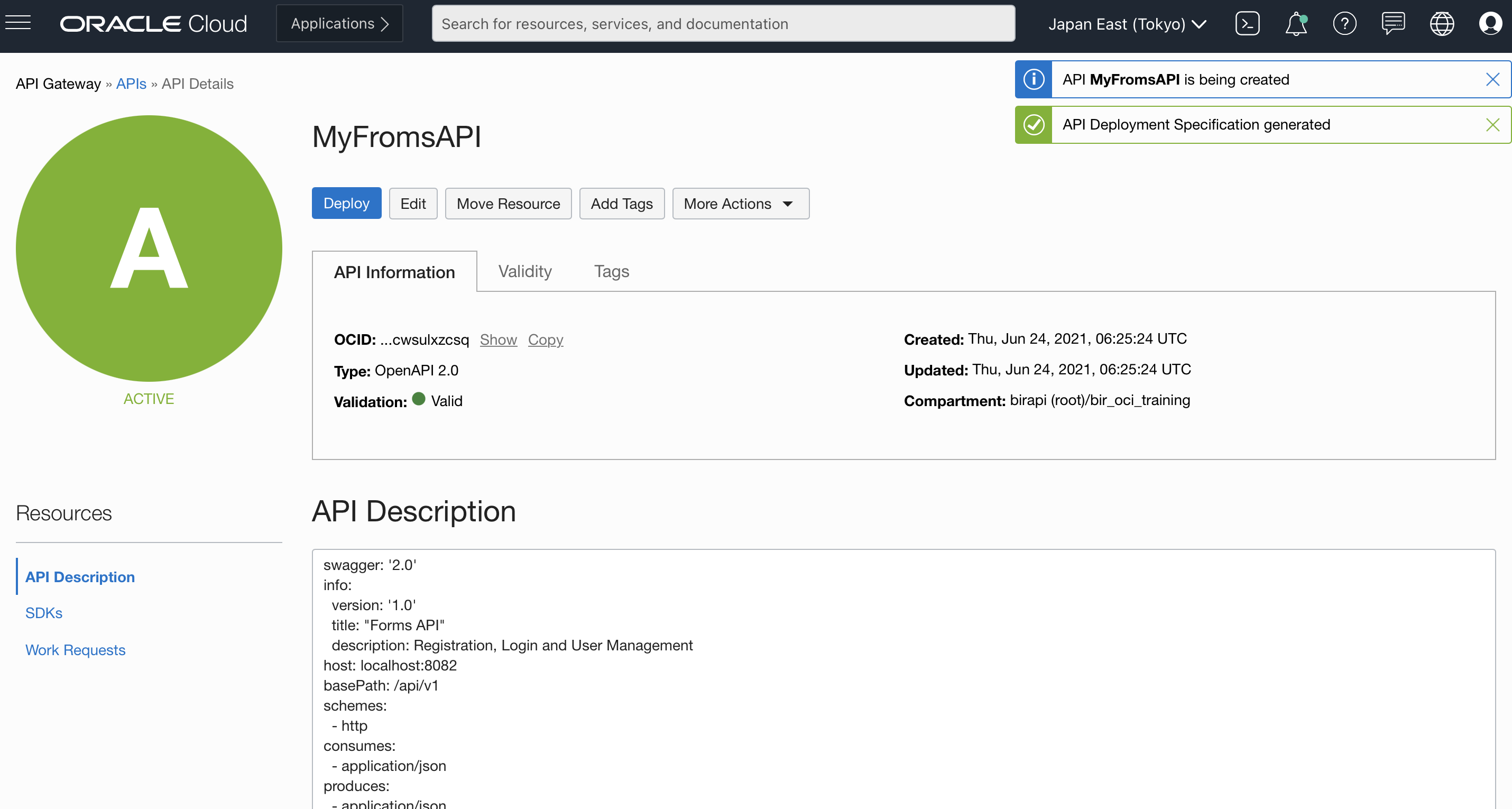


Select a yaml file and click on Upload.



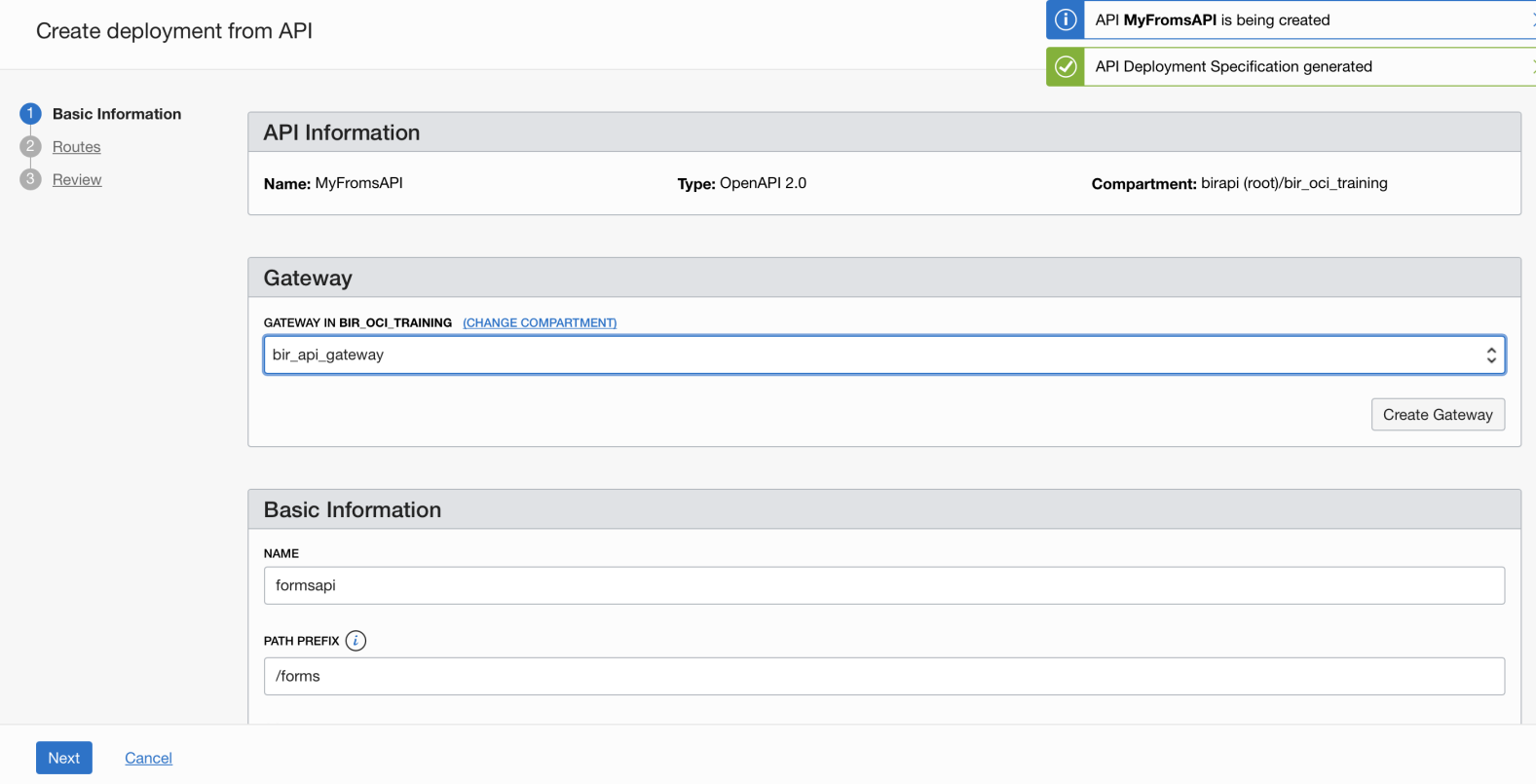


Once done. Click on Create button to create the API.

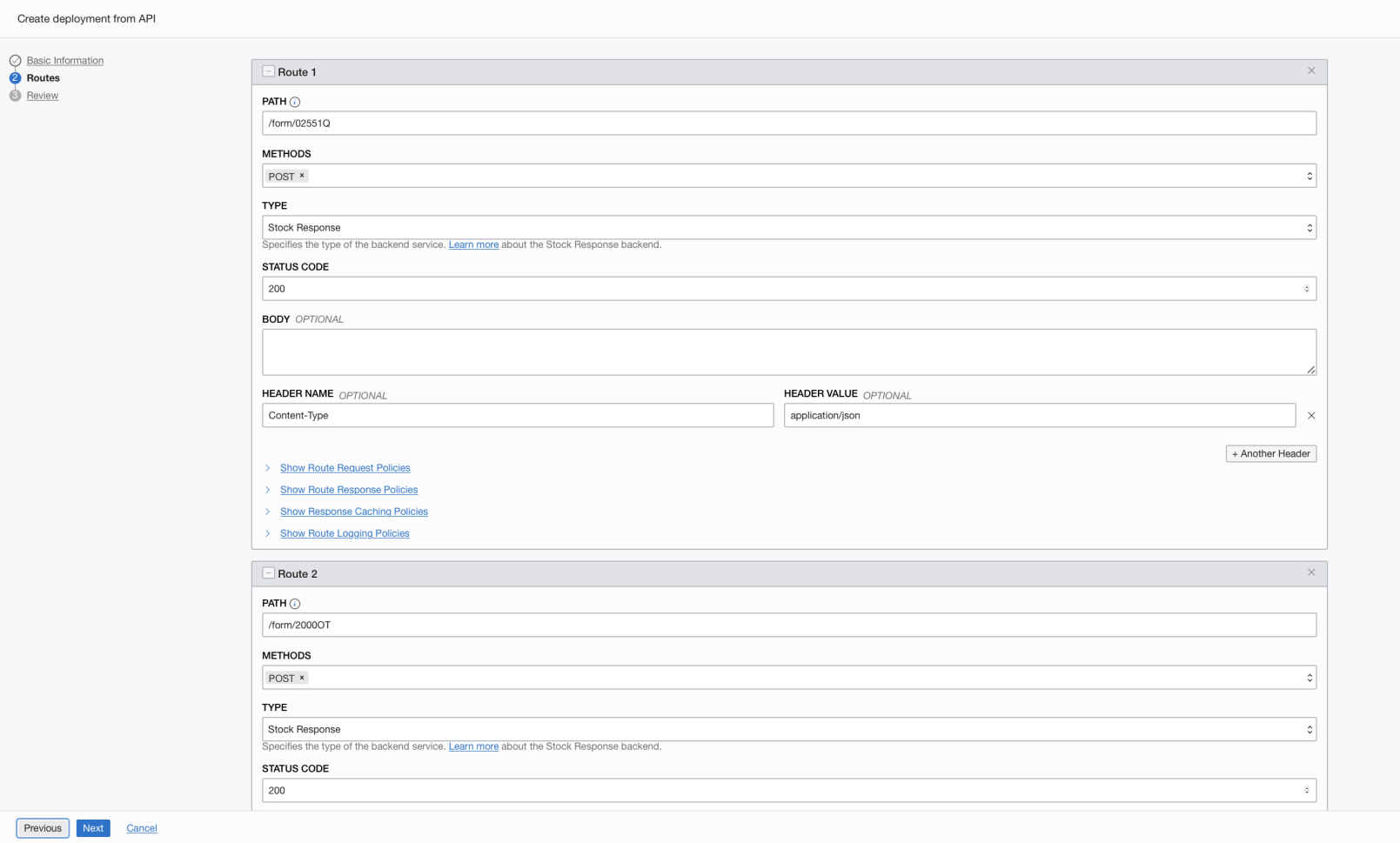


Next step would be to click on “Deploy” to create the api url.

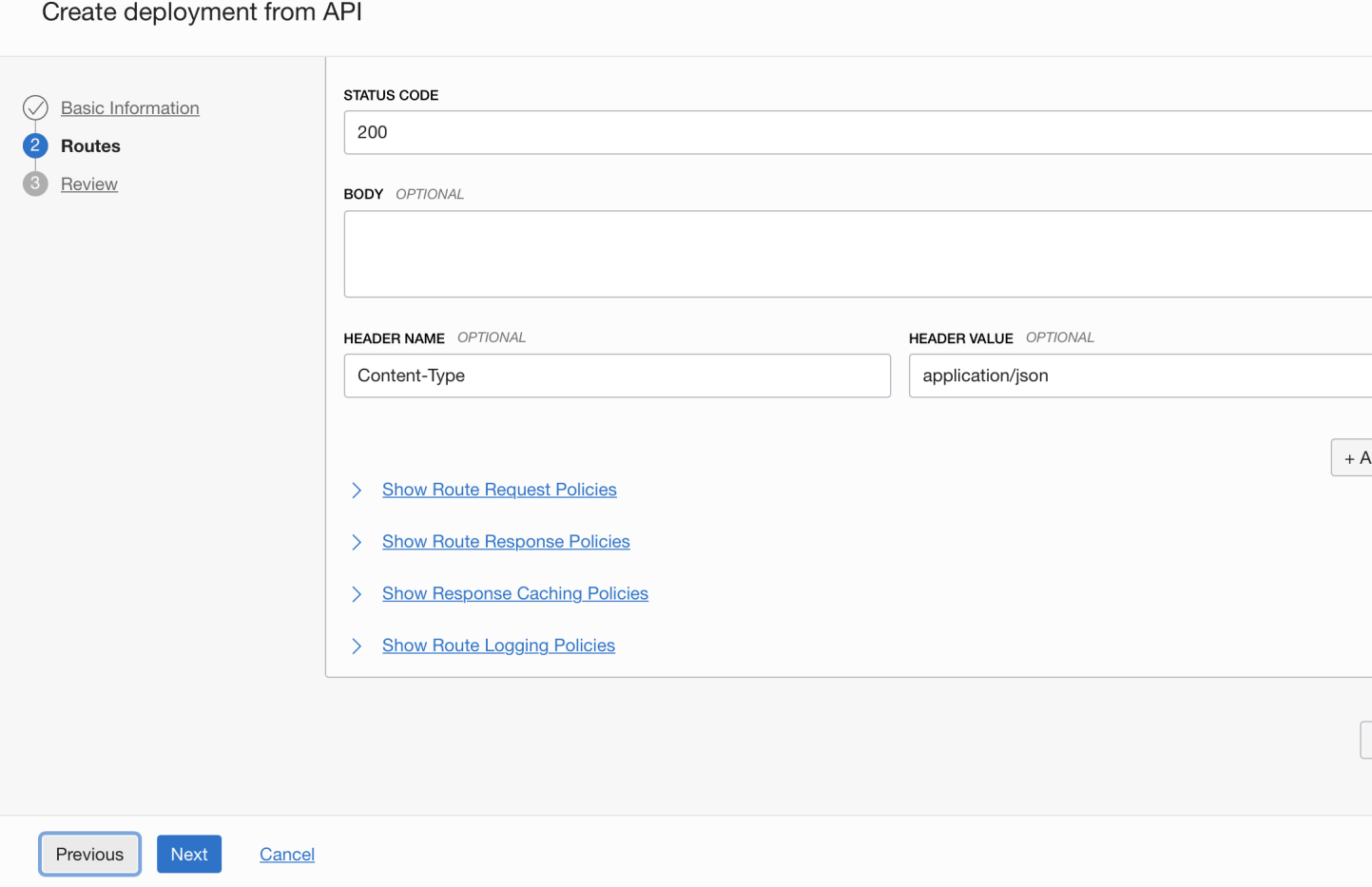
Fill in the rest of the details. Once done click on “Next” to go to the next screen.



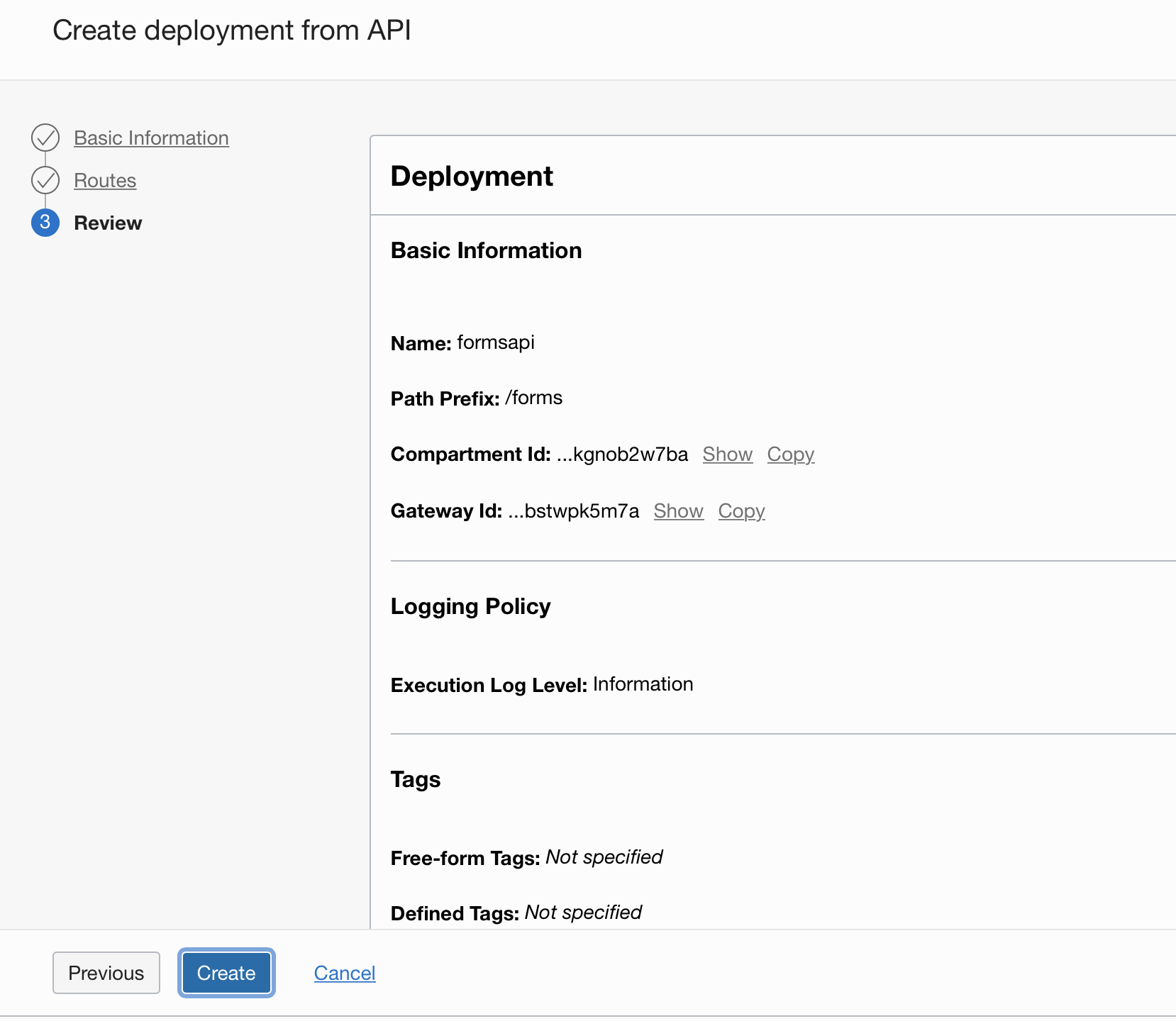
In this page we can see that the Route rules are already defined for us.



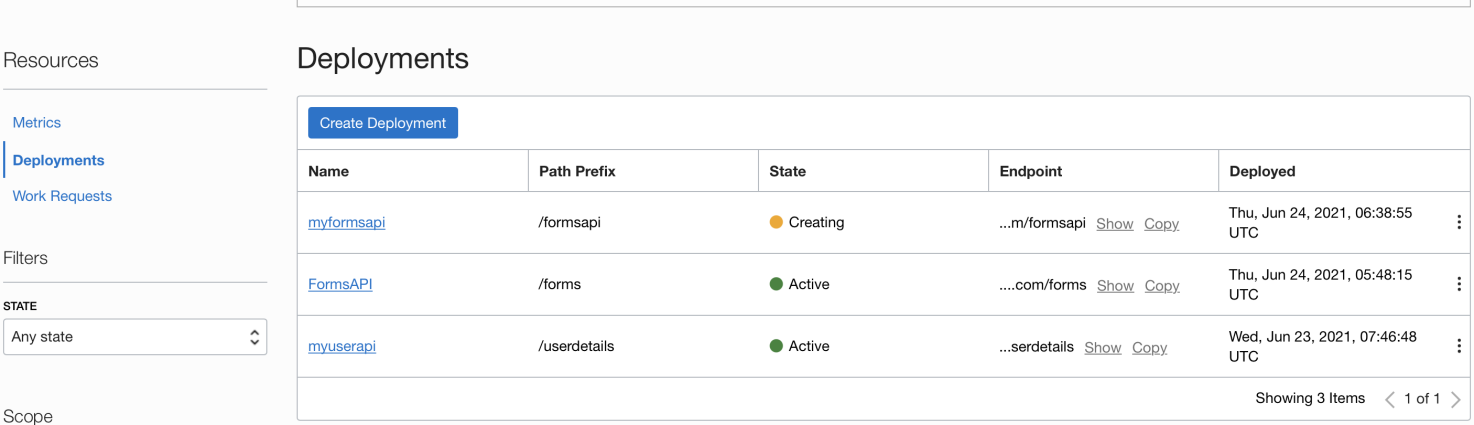
This swagger file that we used has multiple POST request defined in it hence we have multiple route rules here. Click Next to continue.



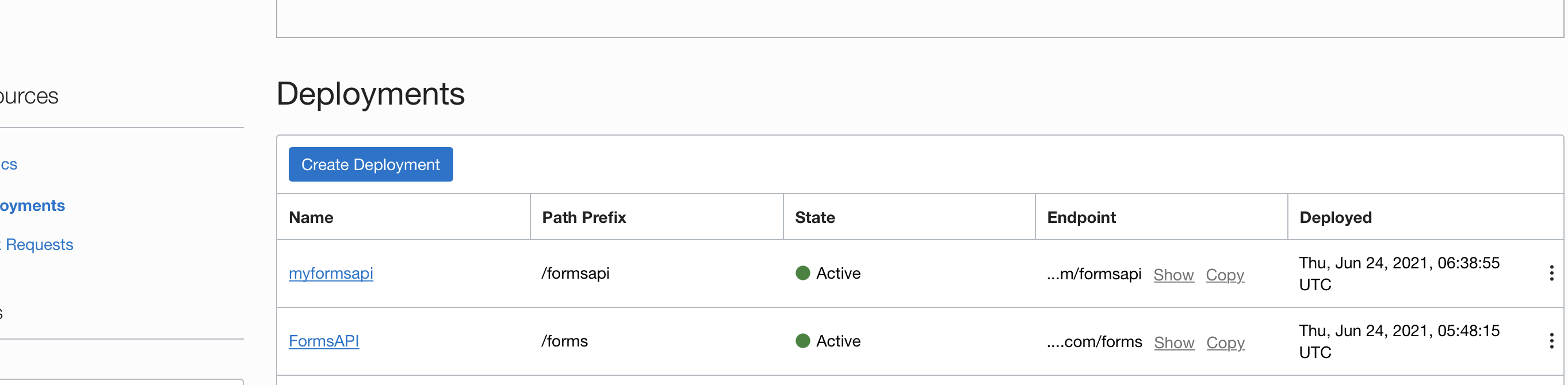
Verify the deployment information shown and then click on “Create”.

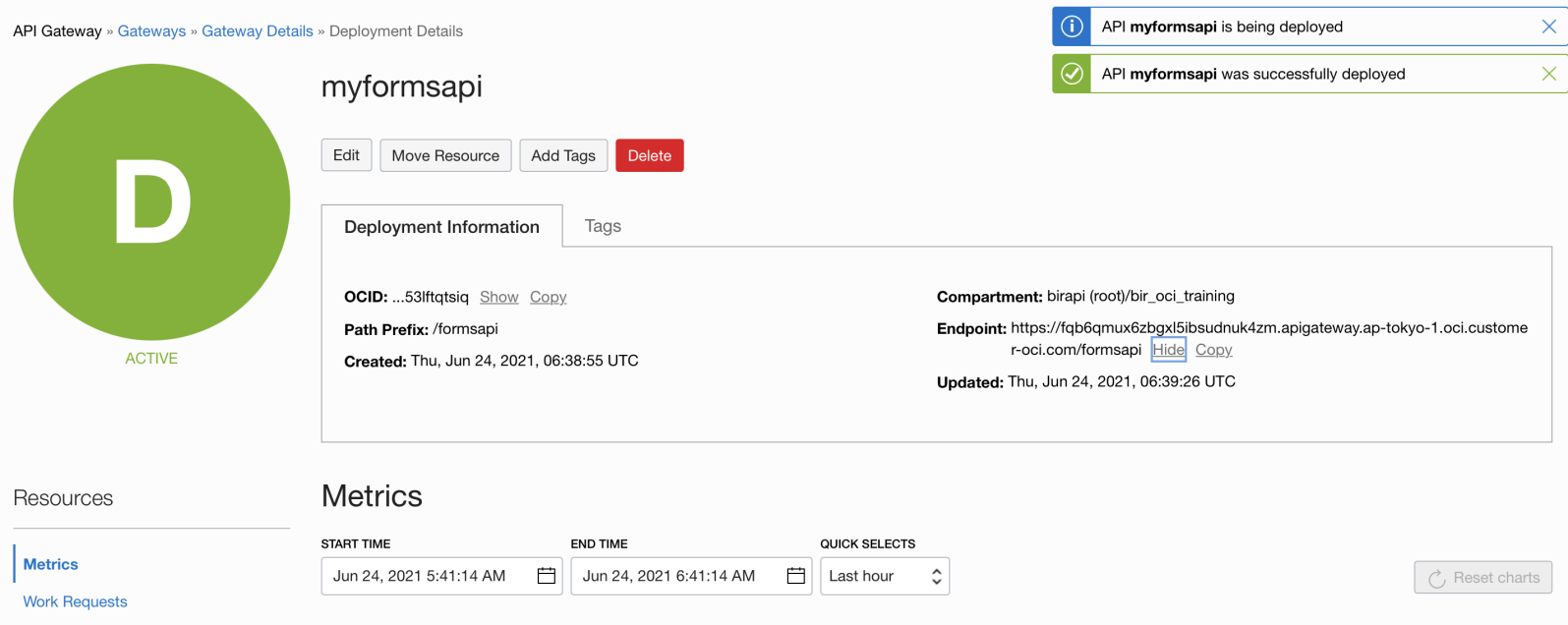


We should be able to see the status of the deployment getting created.



Once created, click on myformsapi to navigate to the next page.



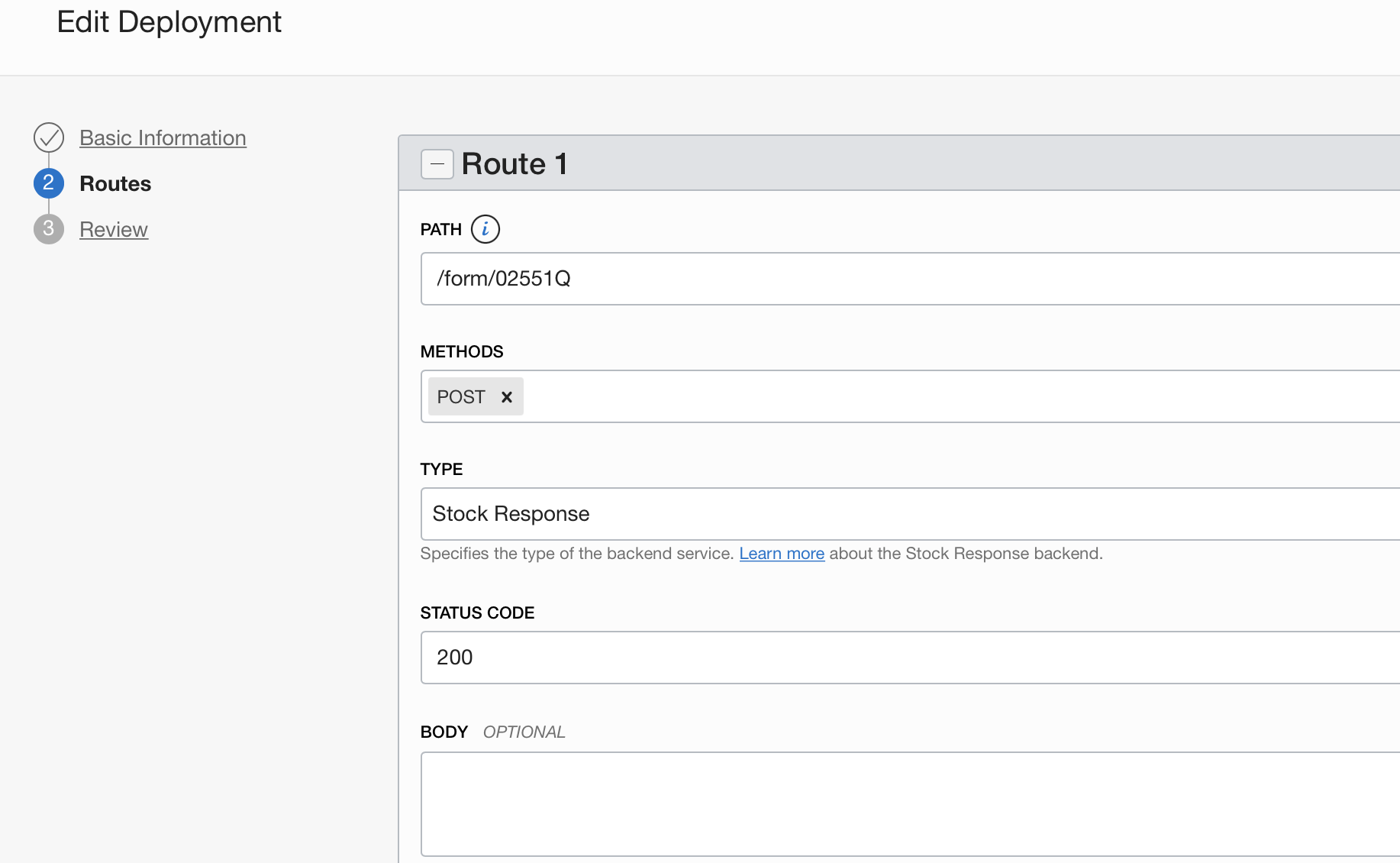


We should be able to see the Endpoint generated.

Eg: <https://fqb6qmux6zbgxl5ibsudnuk4zm.apigateway.ap-tokyo-1.oci.customer-oci.com/formsapi>

Now please note this is the common endpoint. To get the individual endpoints we will have to navigate to the route rules and identify the required endpoint.

We can click on the EDIT option and navigate to the route rules page.



Under PATH we can find our first endpoint.

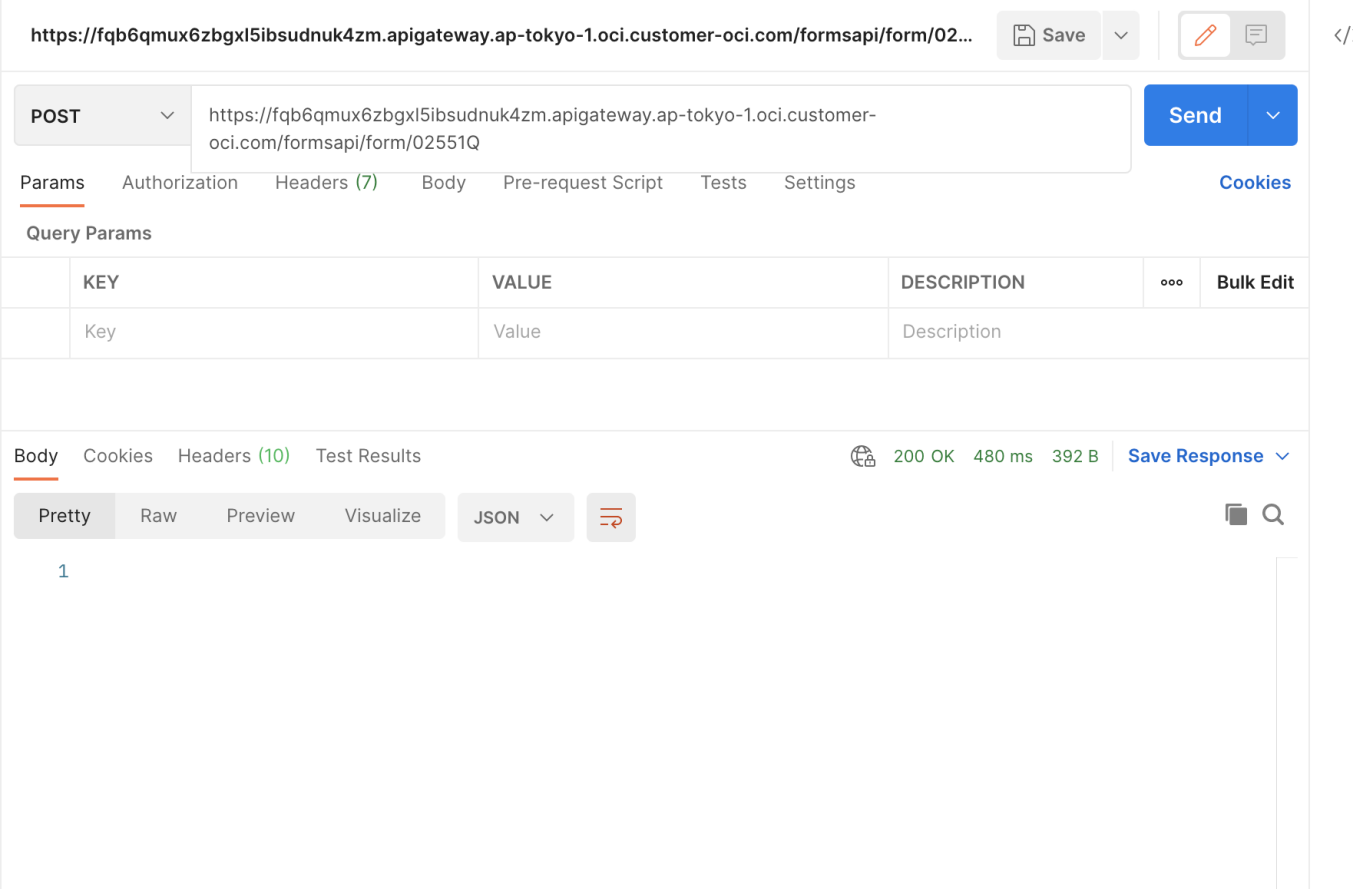
Please copy this endpoint and add it to the end of the actual endpoint url.

So the first POST endpoint is

Eg: <https://fqb6qmux6zbgxl5ibsudnuk4zm.apigateway.ap-tokyo-1.oci.customer-oci.com/formsapi>/form/02551Q

Now try this endpoint in a tool that from where we can make a POST call.

I am using Postman for testing this.



We can see we have a 200 OK response. This confirms our api endpoint is working.

We can repeat similar steps for the other endpoints to test them.

This completes our hands on tutorial for API Gateway.

------ eod ------

Created by Saikat Dey