Deploying a Verified Model Downloaded from HuggingFace using AI Quick Actions

Referenced Documentation

https://docs.oracle.com/en-us/iaas/data-science/using/ai-quick-actions-models-byom.htm

https://docs.oracle.com/en-us/iaas/data-science/using/ai-quick-actions-model-deploy.htm

Description

If you have models you want to use instead of the cached models provided by Data Science, you can bring them into AI Quick Actions from Object Storage or from Hugging Face by registering the model.

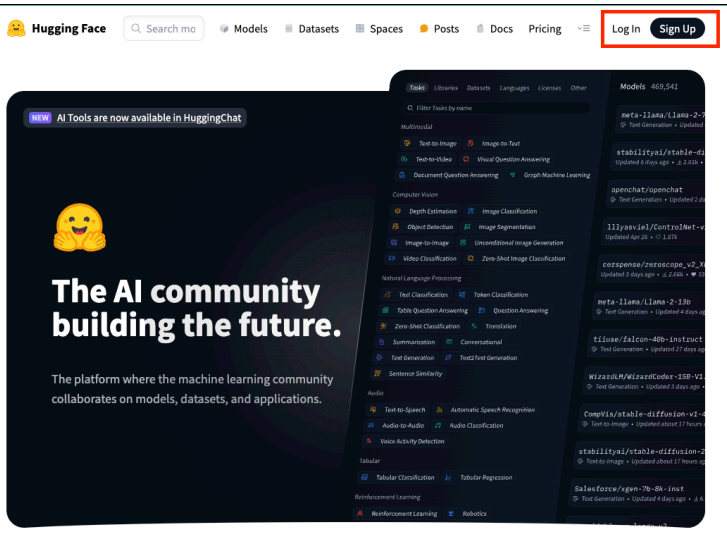
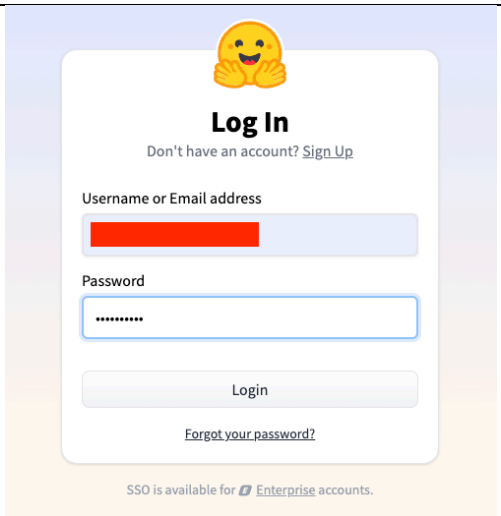
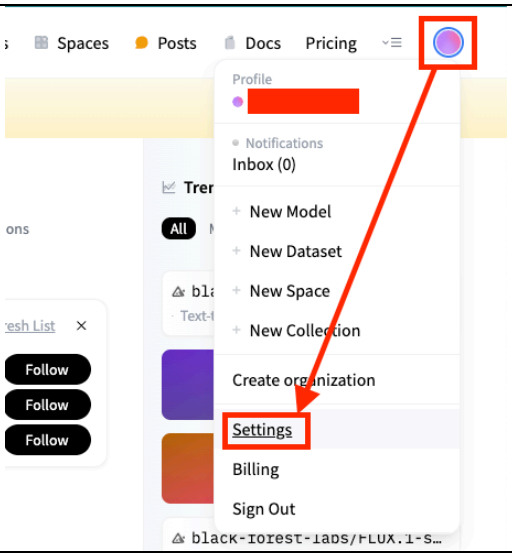
Hugging Face is an open-source model repository. You can bring in models from here to use in AI Quick Actions. Hugging Face offers certain gated models that require the acceptance of user agreement. To bring a gated model from Hugging Face into AI Quick Actions, sign in to Hugging Face using the Hugging Face CLI and your Hugging Face token from a terminal inside the Notebook.

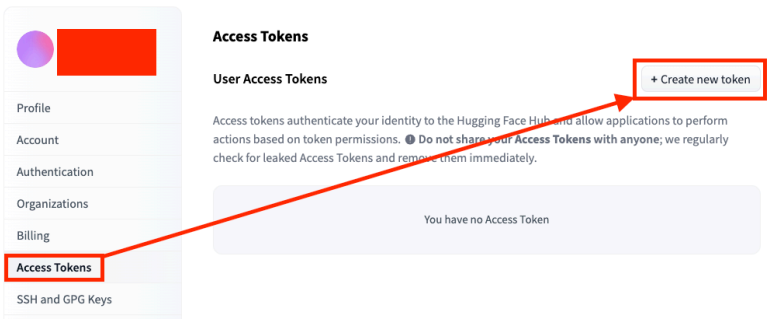
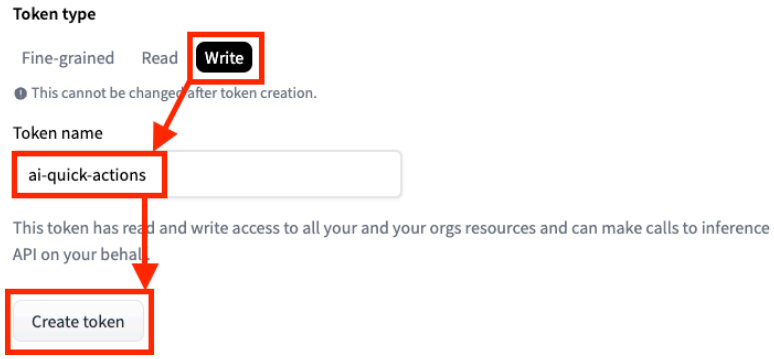
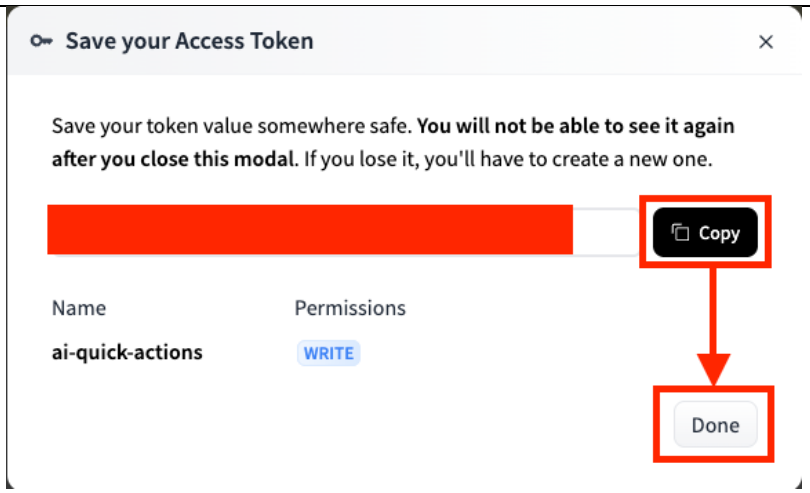
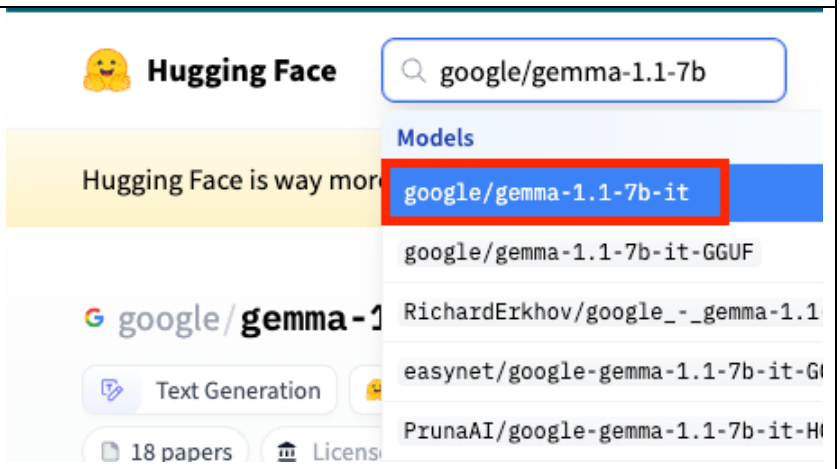
You can create a Model Deployment from the foundation models with the tag Ready to Deploy in the Model Explorer, or with fine-tuned models. When you create a Model Deployment in AI Quick Actions, you're creating an OCI Data Science Model Deployment, which is a managed resource in the OCI Data Science Service. You can deploy the model as HTTP endpoints in OCI.

Pre-Requisites

- Implement the required AI Quick Actions Policies - https://docs.oracle.com/en-us/iaas/data-science/using/ai-quick-actions-set-up.htm
- Ensure you have your OCI Data Science GPU service limits raised for the GPU Shapes you plan to use. This can be done from OCI Console.
- Provisioned OCI Data Science Project and Notebook Session (Must be deactivated and reactivated if created before the policies where implemented).
- OCI Log Group & Log Created (Optional)
- HuggingFace Login Details

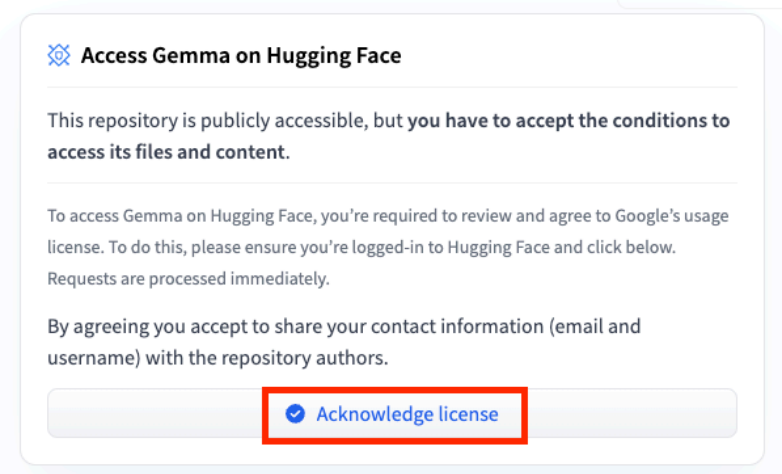
Guide

Step	Screenshot
First, we will login to HuggingFace to generate an Access Token and Accept the Licence agreement for the model we want to deploy. https://huggingface.co/ ***Click Login or Sign Up if required.***	
Login to HuggingFace.	
From your Profile Icon visit Settings.	

Navigate to Access Tokens ***Click Create new token.***	
Select token type as Write. ***Give the Token a Name.*** ***Click Create Token.***	
Copy your token and save it somewhere safe as it won't be displayed again. ***Click Copy (save somewhere).*** ***Click Done.***	
Within the HuggingFace Search Bar, search for the OCI Data Science Verified Model you plan to deploy. In our case we will go with ***google/gemma-1.1-7b-it*** ***Select the Model.***	

If required and the model is gated, you will need to Acknowledge the License before your account has the permissions to utilise within OCI Data Science AI Quick Actions.

Click Acknowledge License.

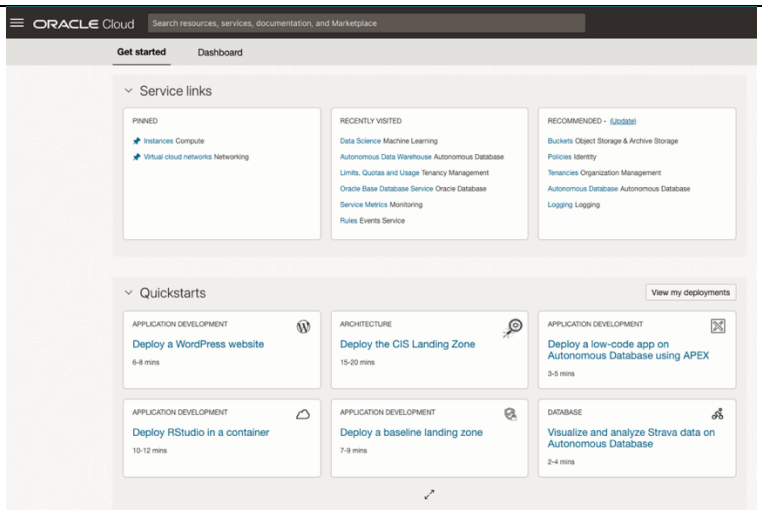


IT IS YOUR SOLE RESPONSIBILITY TO READ AND ACKNOWLEDGE THE LICENSE FOR THE MODEL YOU PLAN TO USE.

IT MAY TAKE 10-15 MINUTES FOR YOUR ACCOUNT TO REGISTER YOU HAVE ACCEPTED THE LICENSE AGREEMENT.

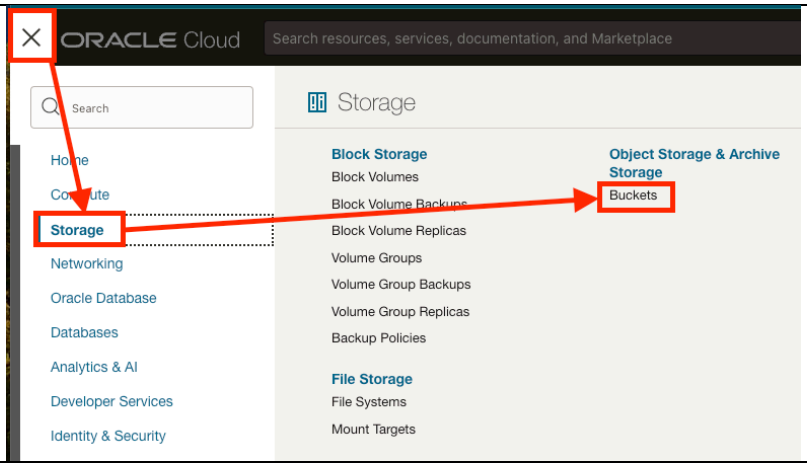
Login to the Cloud Console.

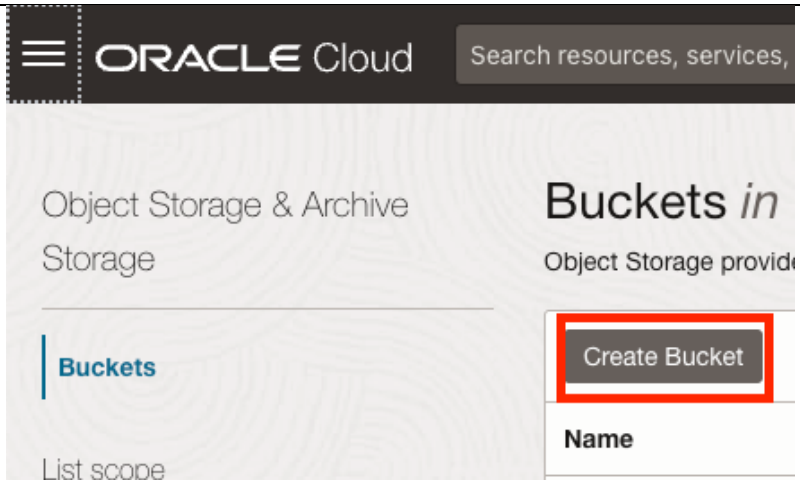
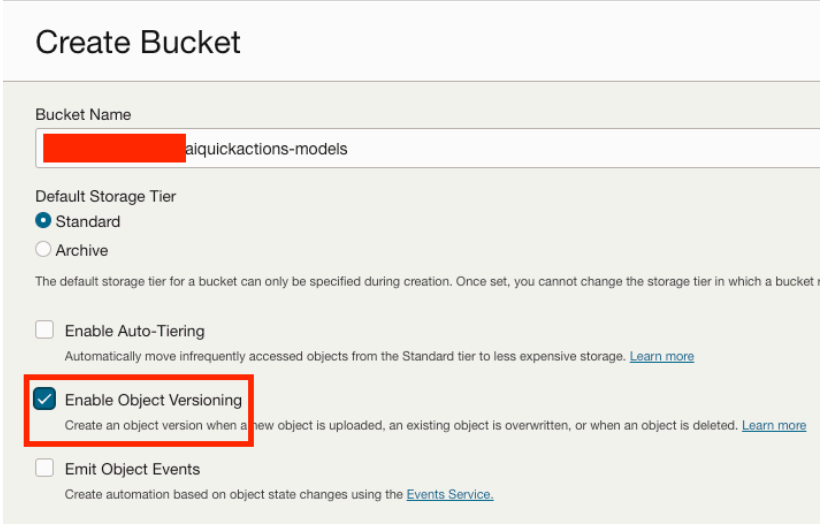
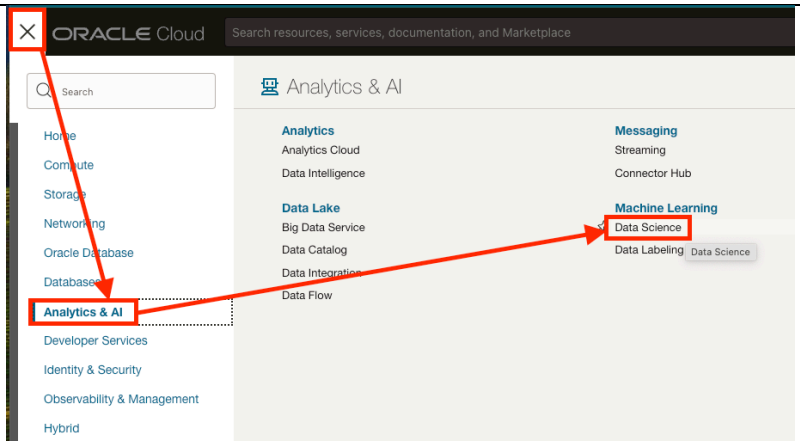
cloud.oracle.com

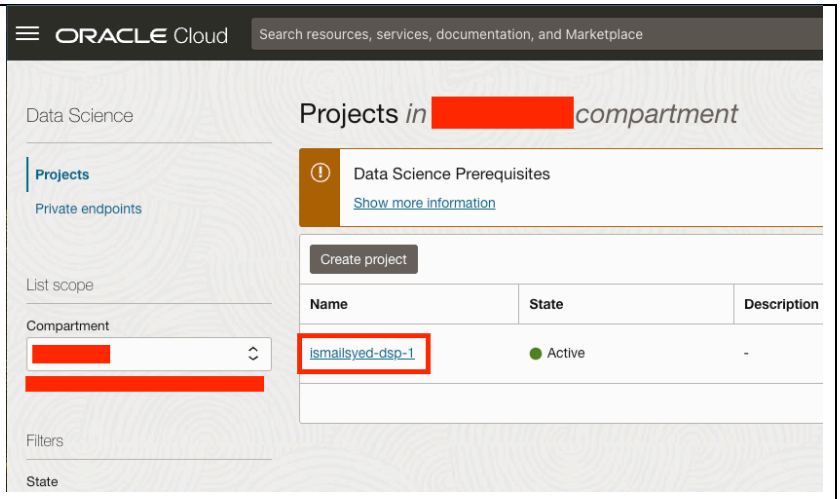
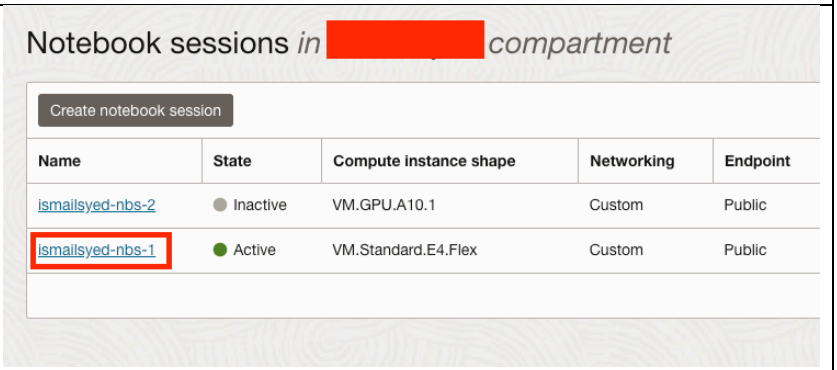
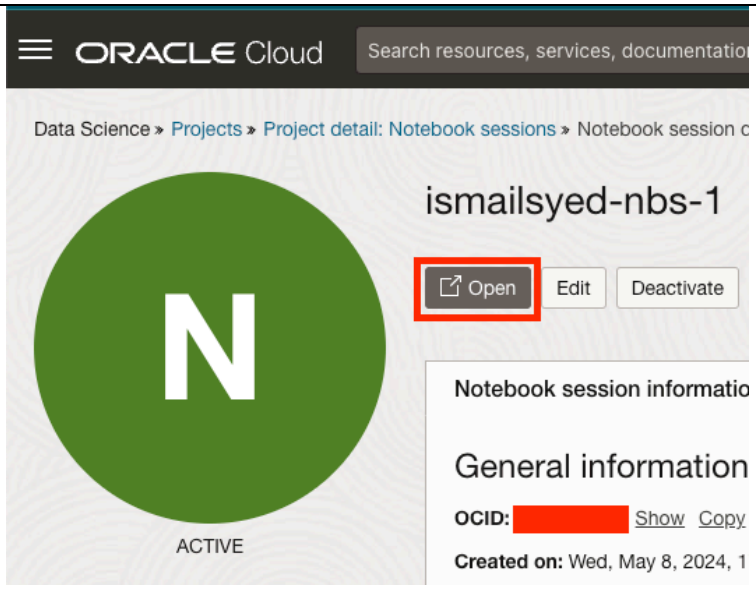


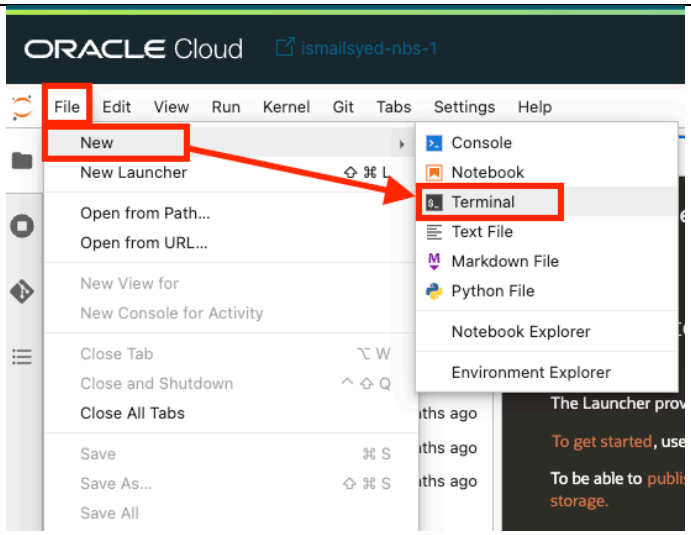
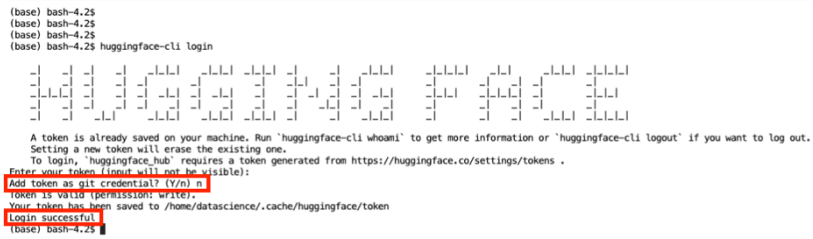
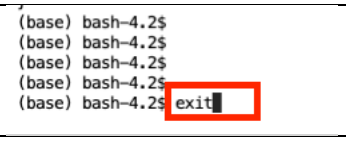
First, we will create a bucket to store our downloaded HuggingFace Model to.

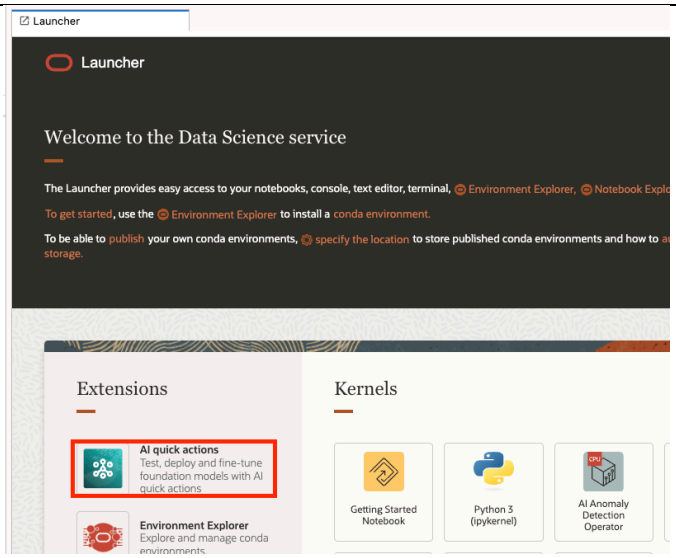
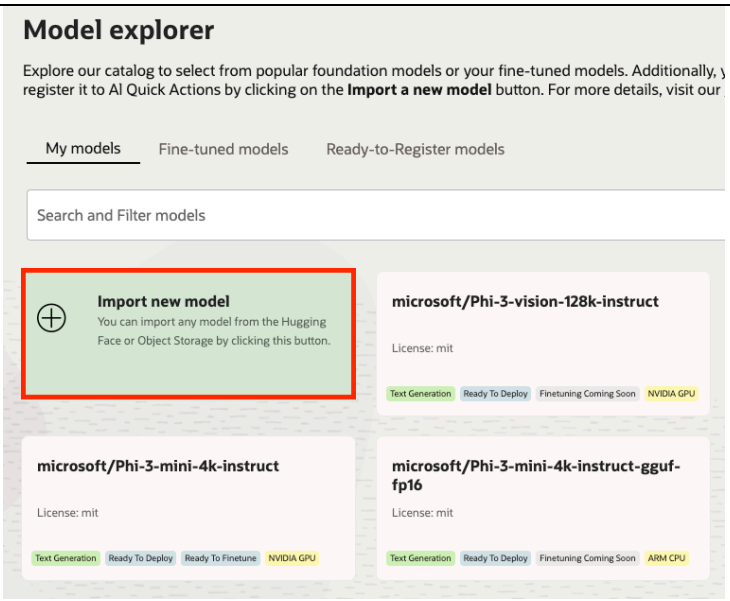
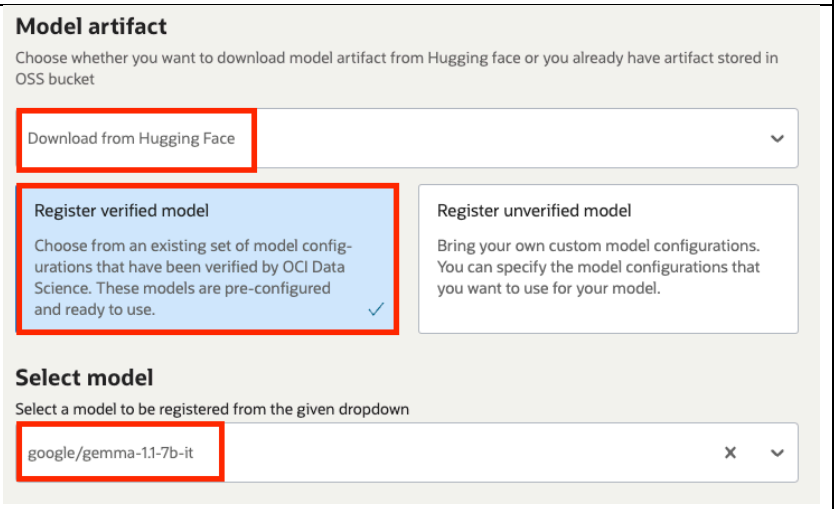
Navigate to OCI Menu > Storage > Buckets.



Click Create Bucket.	
The first bucket we create will be for our finetuned model. ***Enter a Name.*** ***Enable Object Versioning.*** ***Click Create.***	
Now navigate to your Data Science Projects. ***OCI Menu > Analytics & AI > Data Science.***	

Open up your existing Data Science Project.	
Click on your existing Data Science Notebook Session. Note – This does not have to be a GPU Shape.	
Click on Open. This will open up your Data Science Notebook Session. ***You will have to reauthenticate.***	

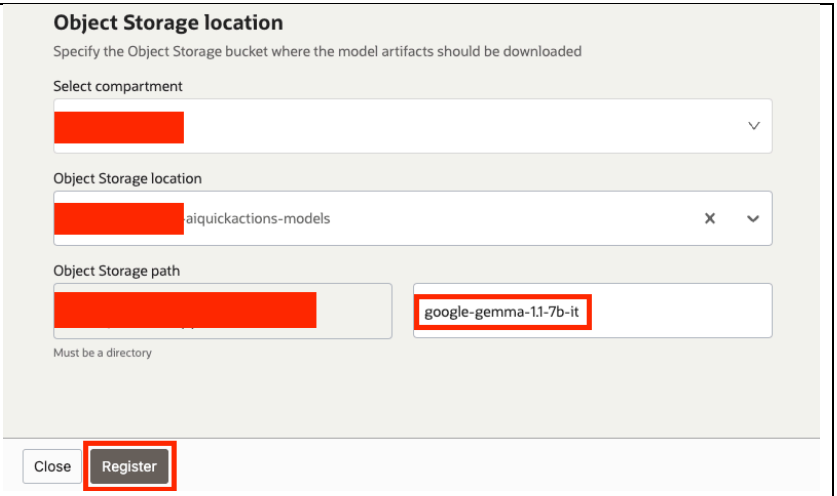
Once your Notebook Session is open, open a Terminal. ***File > New > Terminal.***	
We will then login to Hugging Face using the CLI Tool. Enter: ***huggingface-cli login***	
You will then be prompted to enter your Access Token we generated earlier within HuggingFace. ***Enter your Access Token.*** You will then be prompted to Add the token as a git credential. ***Enter: n*** You should get a message saying Login Successful.	
You can then exit the Terminal: ***exit***	

If the policies within the pre-requisites have been implemented correctly you should be able to open up the AI Quick Actions Extension within the Launcher. ***Click AI quick actions.***	
In this demo we will be deploying ***google/gemma-1.1-7b-it*** as it is not a cached model, we need to first download it into our environment. ***Click Import new model.***	
Select: Download from Hugging Face The Gemma Models is part of our verified model list. Select: ***Register verified model.*** Use the drop down to select ***google/gemma-1.1-7b-it***	

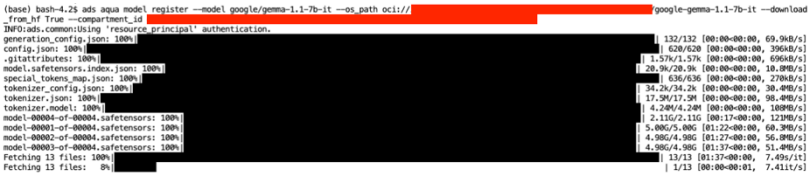
We then need to specify our Object Storage Bucket we created earlier to store our Downloaded Model.

Select:

- ***Compartment***
- ***Bucket Name***
- ***Add a file prefix if required.***



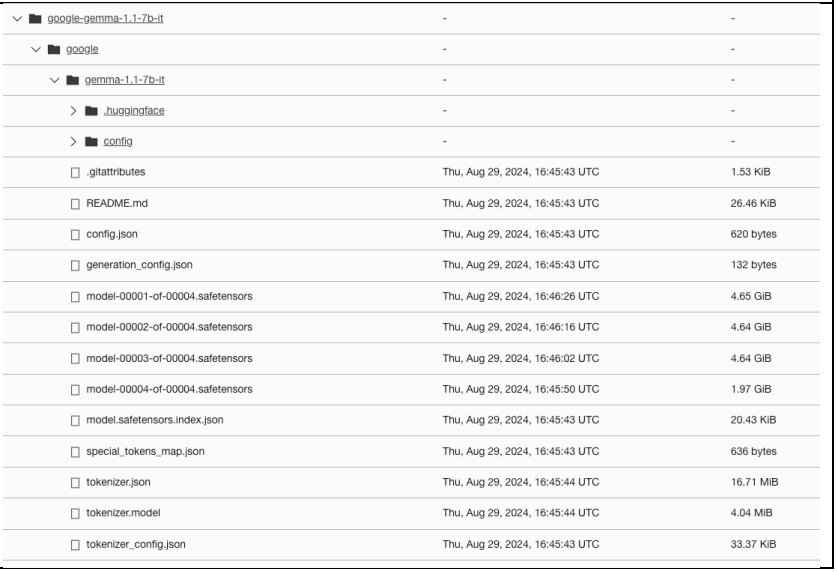
This will then open a Terminal and kick start the download of our model to Object Storage.



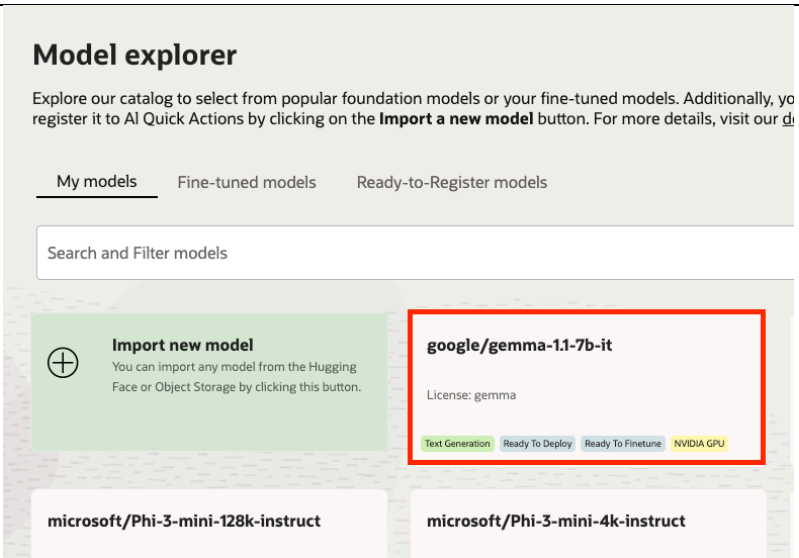
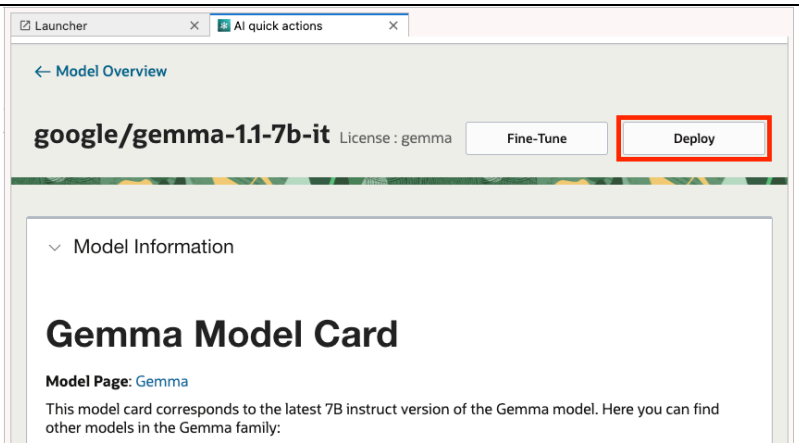
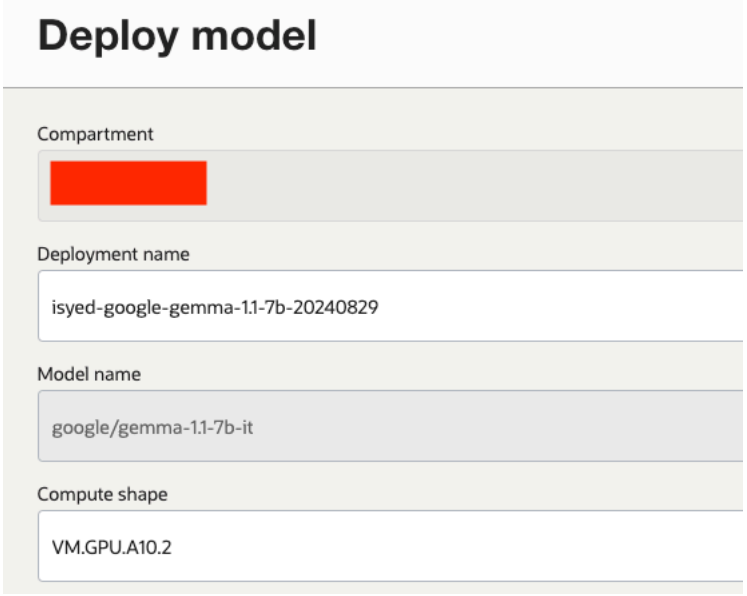
Once download is complete, we can exit the Terminal.

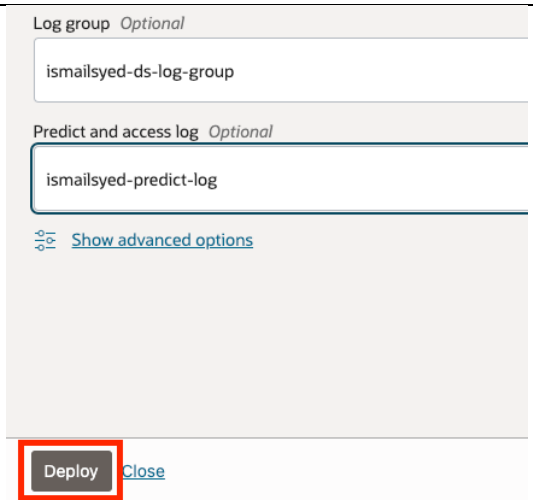
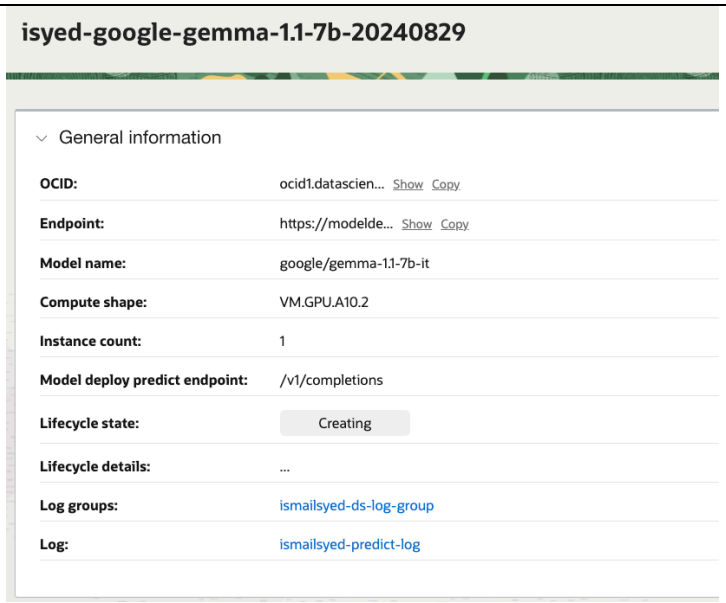
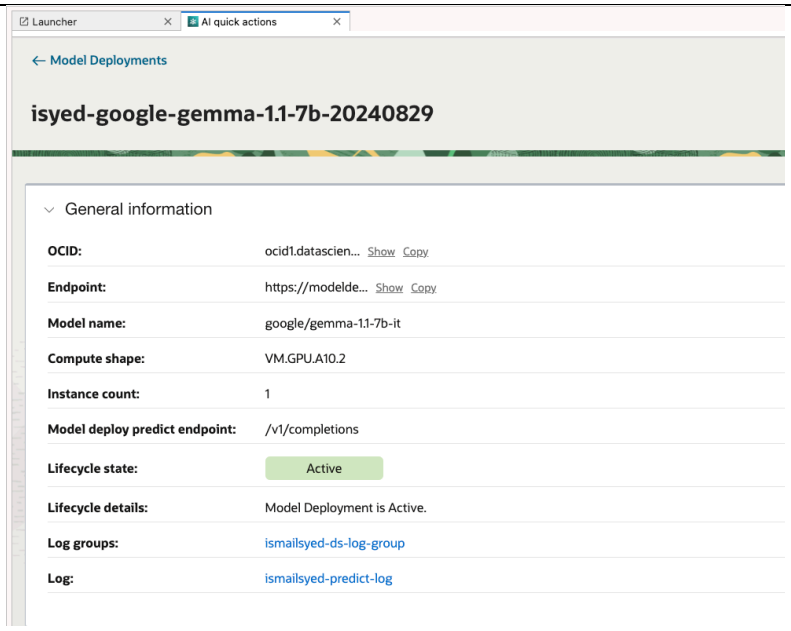
exit

We can then check within our Object Storage Bucket we created earlier, and we should be able to see all the downloaded model artifacts.



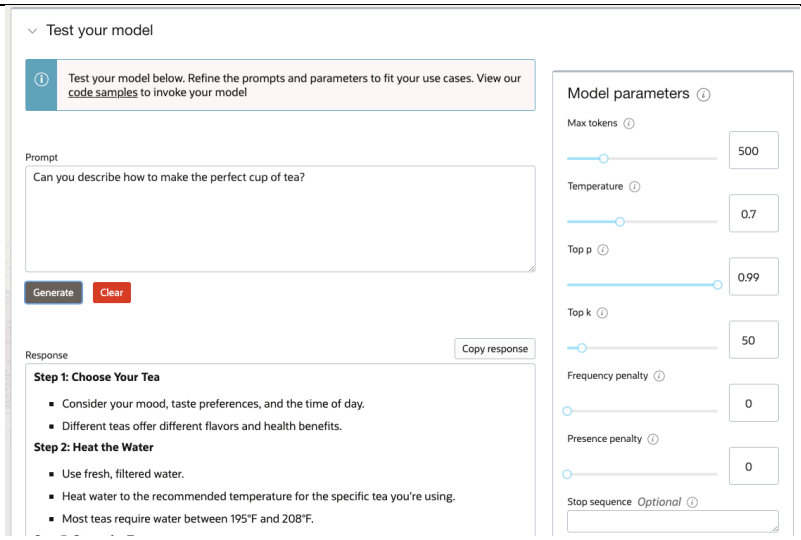
google-gemma-1.1-7b-it	-	-
google	-	-
gemma-1.1-7b-it	-	-
.huggingface	-	-
config	-	-
.gitattributes	Thu, Aug 29, 2024, 16:45:43 UTC	1.53 KiB
README.md	Thu, Aug 29, 2024, 16:45:43 UTC	26.46 KiB
config.json	Thu, Aug 29, 2024, 16:45:43 UTC	620 bytes
generation_config.json	Thu, Aug 29, 2024, 16:45:43 UTC	132 bytes
model-00001-of-00004.safetensors	Thu, Aug 29, 2024, 16:46:26 UTC	4.65 GiB
model-00002-of-00004.safetensors	Thu, Aug 29, 2024, 16:46:16 UTC	4.64 GiB
model-00003-of-00004.safetensors	Thu, Aug 29, 2024, 16:46:02 UTC	4.64 GiB
model-00004-of-00004.safetensors	Thu, Aug 29, 2024, 16:45:50 UTC	1.97 GiB
model.safetensors.index.json	Thu, Aug 29, 2024, 16:45:43 UTC	20.43 KiB
special_tokens_map.json	Thu, Aug 29, 2024, 16:45:43 UTC	636 bytes
tokenizer.json	Thu, Aug 29, 2024, 16:45:44 UTC	16.71 MiB
tokenizer.model	Thu, Aug 29, 2024, 16:45:44 UTC	4.04 MiB
tokenizer_config.json	Thu, Aug 29, 2024, 16:45:43 UTC	33.37 KiB

Within AI Quick Actions under the Model Explorer, we should now see a new Model Card for our ***google/gemma-1.1-7b-it*** Model. ***Click on it.***	
We can then Deploy the model. ***Click Deploy.***	
We can then give our Deployment a Name. ***Select the Compute Shape. I have left it as the default – VM.GPU.A10.2***	

As an optional task, I have selected a Log Group and Predict and Access Logs. ***Click on Deploy.***	
Your deployment will then enter the ***Creating Lifecycle State***. The deployment can take up to 10/15 minutes to deploy.	
Once the deployment is complete, the lifecycle state should be updated to ***Active***.	

If we scroll down, we then have a nice UI to test out our deployed model.

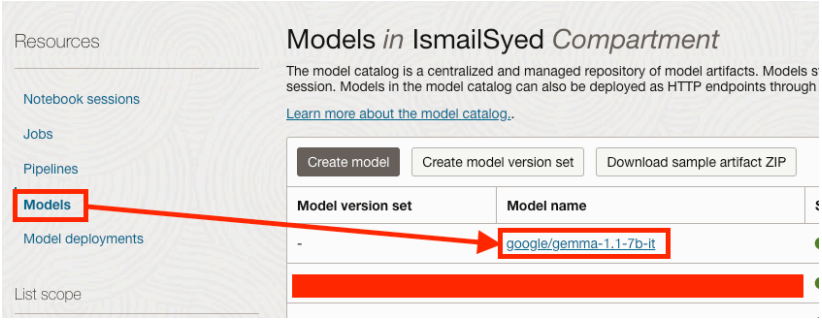
We can enter a prompt, tweak the parameters and then generate an answer.



If we head back to our OCI Data Science Project within the OCI Console.

Navigate to Models under Resources.

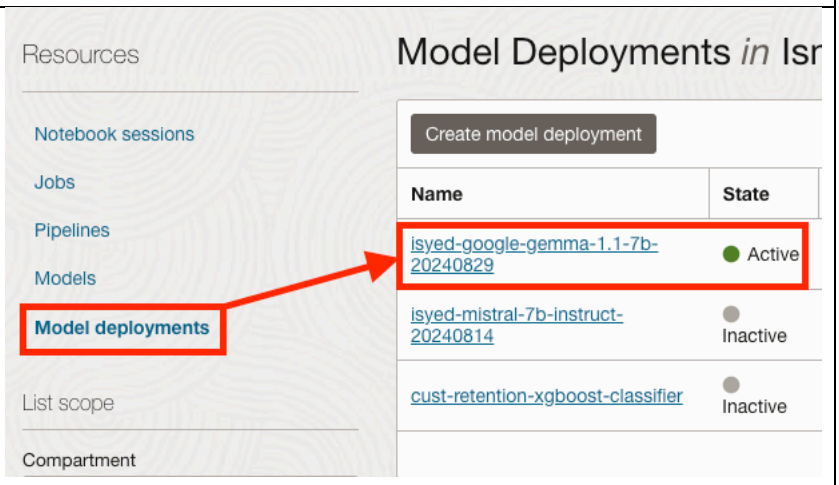
We can see a new entry in our Model Catalog for our Google Gemma Model.



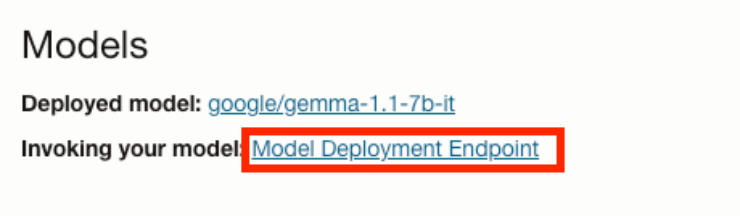
Next ***navigate to your Model Deployments under Resources.***

Here we can see our Google Gemma Deployment.

Select the Model Deployment.

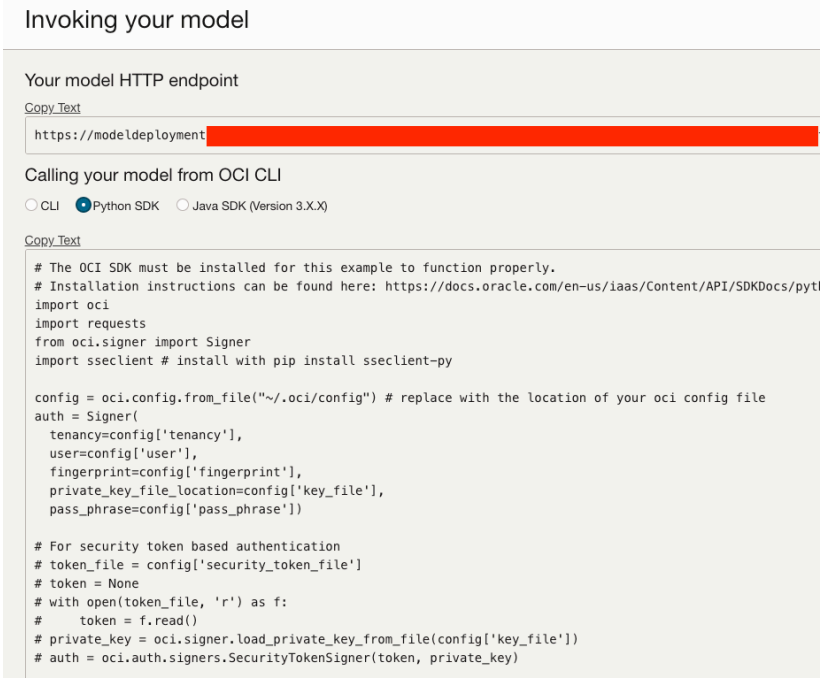


Click on Model Deployment Endpoint under Invoking your Model.



Here you will be displayed with your Model Endpoint and some sample code which can be used to invoke your model.

Note: Authentication to the model will be done via a configuration file, security token or resource principal. This is up to you to configure.



You can also run the ***01-invoke-deployed-gemma-model.ipynb*** Notebook that is provided as part of this guide to connect to your model endpoint and make a request.

This notebook assumes you have a resource principal configured.