To deploy a machine learning model with IBM Cloud Watson Studio

Create an IBM Cloud Account:

If you haven't already, sign up for an IBM Cloud account.

Set Up Watson Studio:

Access IBM Watson Studio from the IBM Cloud dashboard and create a new project.

Add Data Assets:

Add your dataset to the project. Watson Studio supports various data sources.

Create a Machine Learning Model:

Use tools like Jupyter Notebooks within Watson Studio to create and train your machine learning model. You can use popular libraries like scikit-learn or TensorFlow.

Save and Version Your Model:

Once your model is trained and optimized, save it and create different versions if needed. This helps in tracking changes and improvements over time.

Prepare Model for Deployment:

Ensure your model is serializable and can be deployed. Save necessary preprocessing steps and feature transformations.

Choose Deployment Type:

In Watson Studio, you can deploy your model as a web service or a batch job. Choose the appropriate deployment type based on your application requirements.

Configure Deployment:

Configure deployment settings such as resources, environment variables, and scaling options. Watson Studio provides a user-friendly interface for these configurations.

Deploy the Model:

Deploy your machine learning model. Watson Studio will provide you with an endpoint URL that you can use to interact with your deployed model.

```
wml_credentials={
    "apikey": "*****************************",
    "instance_id": "************************",
    "url": "*********************************"
}
client = WatsonMachineLearningAPIClient(wml_credentials)
```

Test the Deployed Model:

After deployment, test your model with sample data to ensure it's working as expected. This step is cr ucial for verifying the deployment's correctness.

Monitor and Update:

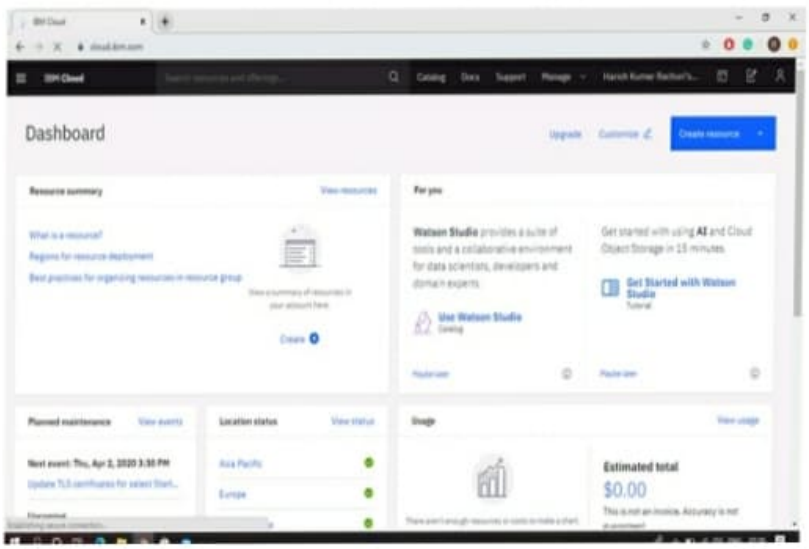
Monitor the deployed model's performance and user interactions. If needed, update the model by retraining it with new data and deploying the updated version.

Code to deploy watson

```
model UID
published_model_uid = client.repository.
get_model_uid(model_artifact)
#Deploy the model
created_deployment = client.deployments.
create(published_model_uid, name="
ChurnModelDeployment
######
Synchronous deployment creation for uid:
'f1cf615d-d9a9-436c-9771-
88df97c7e6ec' started
#######
INITIALIZING
DEPLOY_SUCCESS
------
Successfully finished deployment creation,
 deployment_uid='f9e80285-841e-4783-
bea0-0c76bf8a8ec4'
------
scoring_endpoint = client.deployments.get
_scoring_url(created_deployment)
scoring_payload = {"fields": list(X_test.
columns),
    "values":X_test.iloc[11:20].values.tolist()
}
predictions = client.deployments.score
(scoring_endpoint, scoring_payload)
print(predictions)
{'fields': ['prediction', 'probability'], 'values':
[[0, [0.9849121857890184, 0.
01508781421098155]], [0, [0.
766820123077614, 0.
23317987692223857]], [0, [0.
997714799880596 7, 0.
002285200119403259 7]], [0, [0.
975127668806959, 0.
024872331193040997]], [0, [0.
7327641504178833, 0.
26723584958211666]], [0, [0.
9916415671999173, 0.
008358432800082749]], [1, [0.
37651074677061636, 0.
6234892532293836]], [0, [0.
9986890733149208, 0.
001310926685079143 6]], [0, [0.
9828675236249786, 0.
017132476375021317]]]}
```

Task-1: Create the Watson Studio Instance

- After logging into the IBM Account, navigate to Dashboard



Remember, the exact steps and options might vary slightly based on the updates and changes IBM Watson Studio might have introduced. Always refer to the latest IBM Watson Studio documentation for the most accurate and detailed instructions.