Machine learning project

Network Intrusion Detection system

Problem statement No.40 – Network Intrusion Detection

The Challenge:

Create a robust network intrusion detection system (NIDS) using machine learning. The system should be capable of analysing network traffic data to identify and classify various types of cyber-attacks (e.g., DoS, Probe, R2L, U2R) and distinguish them from normal network activity. The goal is to build a model that can effectively secure communication networks by providing an early warning of malicious activities.

About Dataset:

Background
The dataset to be audited was provided which consists of a wide variety of intrusions simulated in a military network environment. It created an environment to acquire raw TCP/IP dump data for a network by simulating a typical US Air Force LAN. The LAN was focused like a real environment and blasted with multiple attacks. A connection is a sequence of TCP packets starting and ending at some time duration between which data flows to and from a source IP address to a target IP address under some well-defined protocol. Also, each connection is labelled as either normal or as an attack with exactly one specific attack type. Each connection record consists of about 100 bytes.
For each TCP/IP connection, 41 quantitative and qualitative features are obtained from normal and attack data (3 qualitative and 38 quantitative features) .The class variable has two categories:
• Normal
• Anomalous

Kaggle dataset link - Used in this representation – this will generate a model that only give prediction of class(normal or anomaly)

Technology Used - IBM cloud lite services.

GitHub repo link

For upgraded model:

This model will classify network attacks like **DoS, Probe, R2L, U2R**

First, quick attack type breakdown:

Attack Type	**Description**
DoS (Denial of Service)	Overwhelm system/network to make services unavailable
Probe	Scanning/snooping to gather network info
R2L (Remote to Local)	Remote attacker tries to gain local access
U2R (User to Root)	Attacker with user access tries to escalate to root

1. DoS Attacks

These often involve **large volume** or **abnormal traffic**.

- src_bytes, dst_bytes: Sudden large/small byte transfer.
- count, srv_count: High numbers indicate a burst of connections.
- serror_rate, srv_serror_rate: High error rates (typical in SYN flood).
- dst_host_srv_count, dst_host_same_srv_rate: Repeated connection to same service on host.

Red flags for DoS:
High count, serror_rate, dst_host_srv_count with low diff_srv_rate.

2. Probe Attacks

These are scans/reconnaissance, often *low payload*, *high variety*.

- count, srv_count: High, but over varied services.
- same_srv_rate, diff_srv_rate, srv_diff_host_rate: Show diversity of target services/hosts.
- dst_host_diff_srv_rate: High for scan-like behavior.

Red flags for Probe:
High diff_srv_rate, high dst_host_diff_srv_rate, moderate count.

3. R2L Attacks

Low-volume but abnormal login patterns.

- logged_in: Often 0 (unsuccessful attempts).
- num_failed_logins: High.
- is_guest_login: 1 (suspicious).
- hot, num_access_files, num_compromised: Suspicious if increased.
- service: Often mail, ftp, telnet, or HTTP.

Red flags for R2L:
num_failed_logins > 0, logged_in = 0, maybe is_guest_login = 1.

4. U2R Attacks

Attempting to escalate privileges. Very subtle.

- num_compromised, root_shell, su_attempted, num_root: Big signs.
- hot, num_file_creations, num_shells: High if privilege escalation occurs.

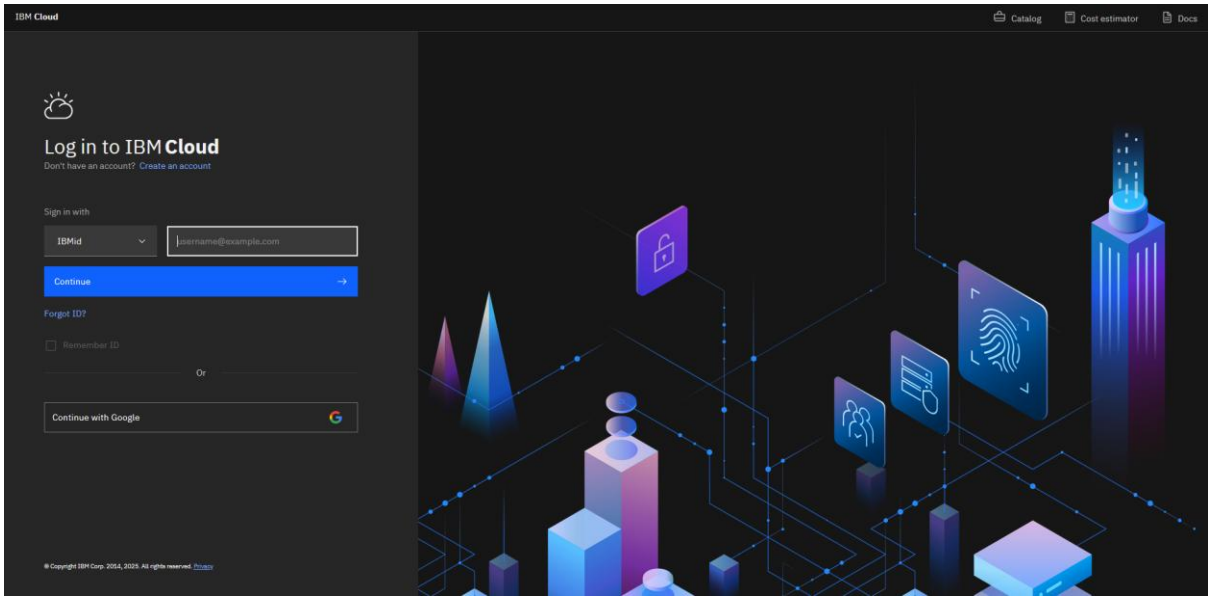
- Very **low** src_bytes/dst_bytes, often **single connection**.

Red flags for U2R:
root_shell = 1, num_file_creations > 0, su_attempted = 1, low traffic volume.

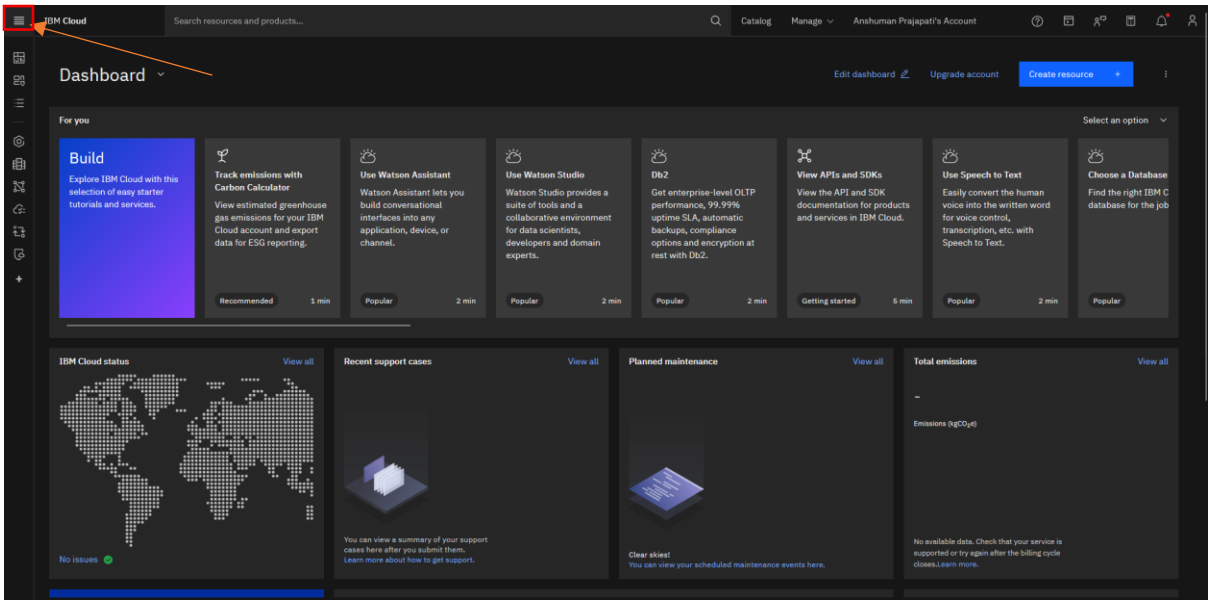
DATA sets used for this upgraded model:

-New data set for training of model

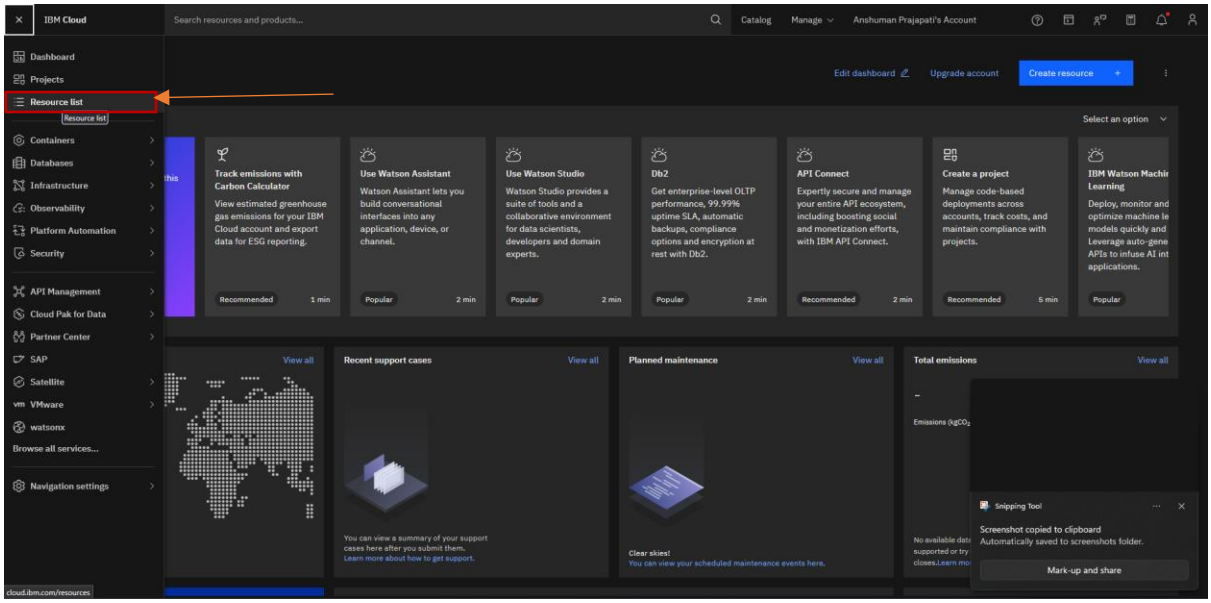
-New data set for testing of model



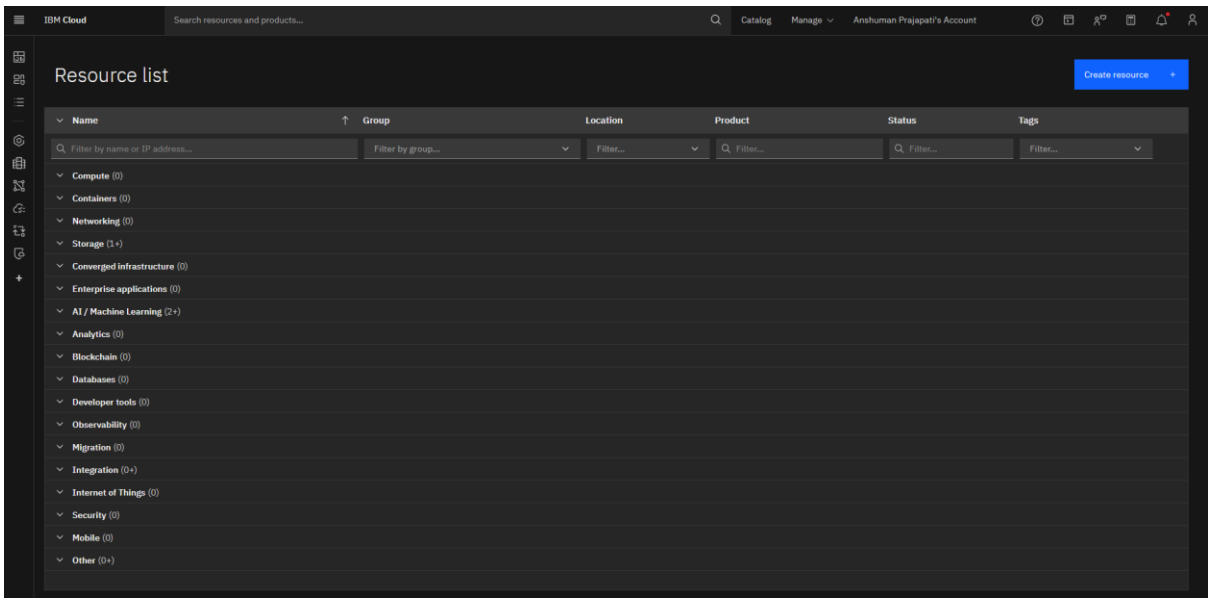
1. Login to IBM cloud with your credentials.



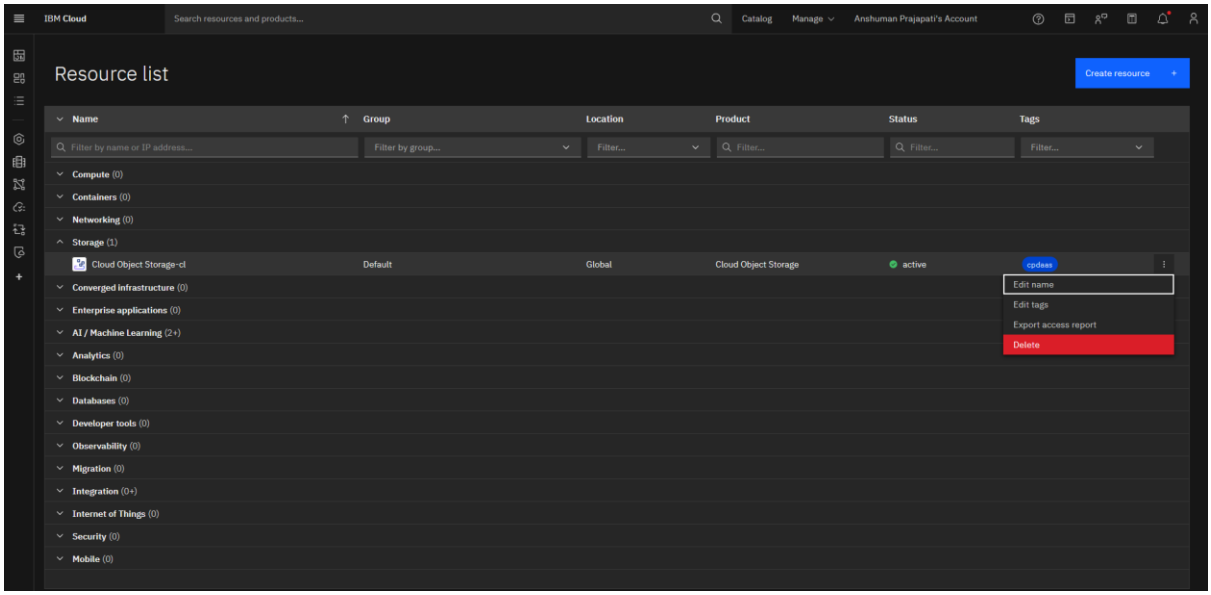
2. This is homepage of IBM cloud now press on top left corner of this page.

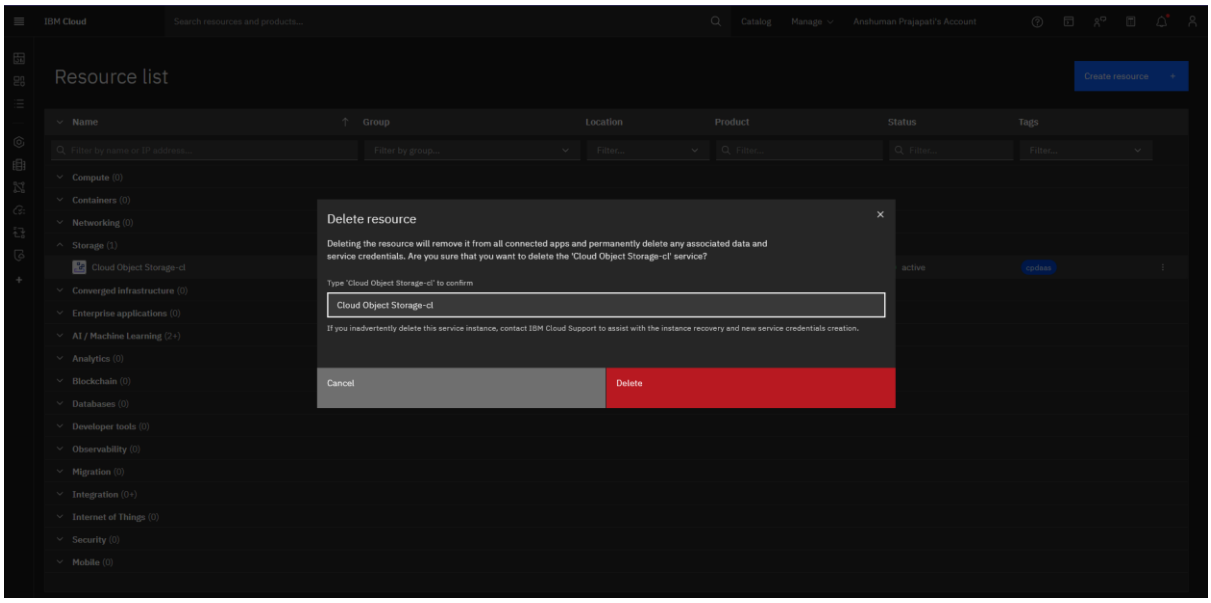


3. Press on Resource list

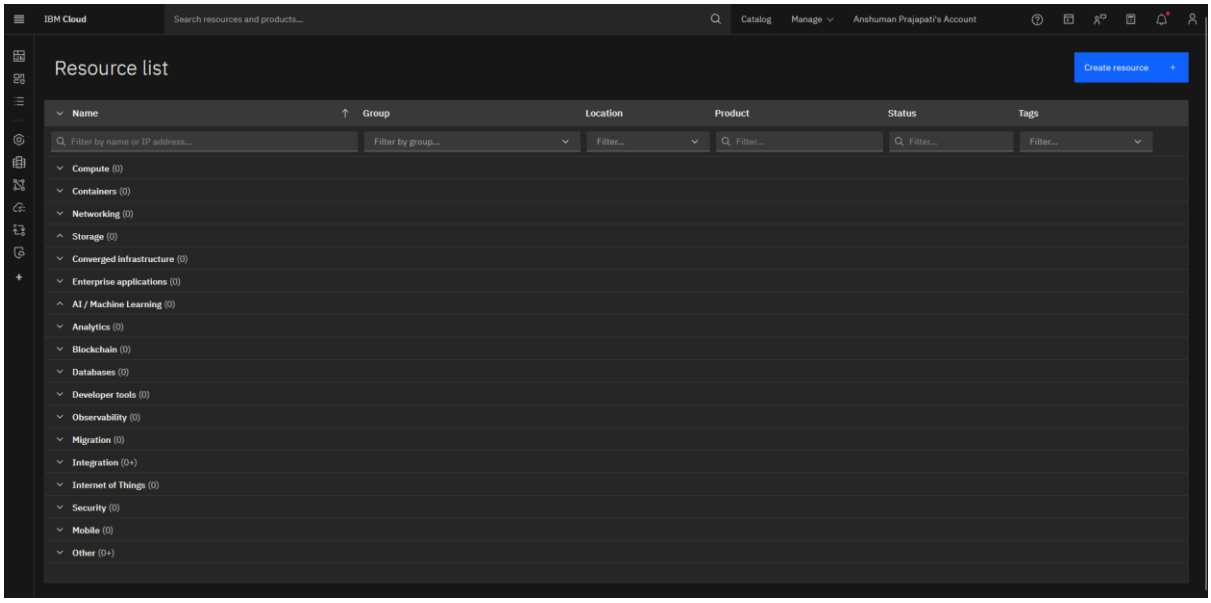


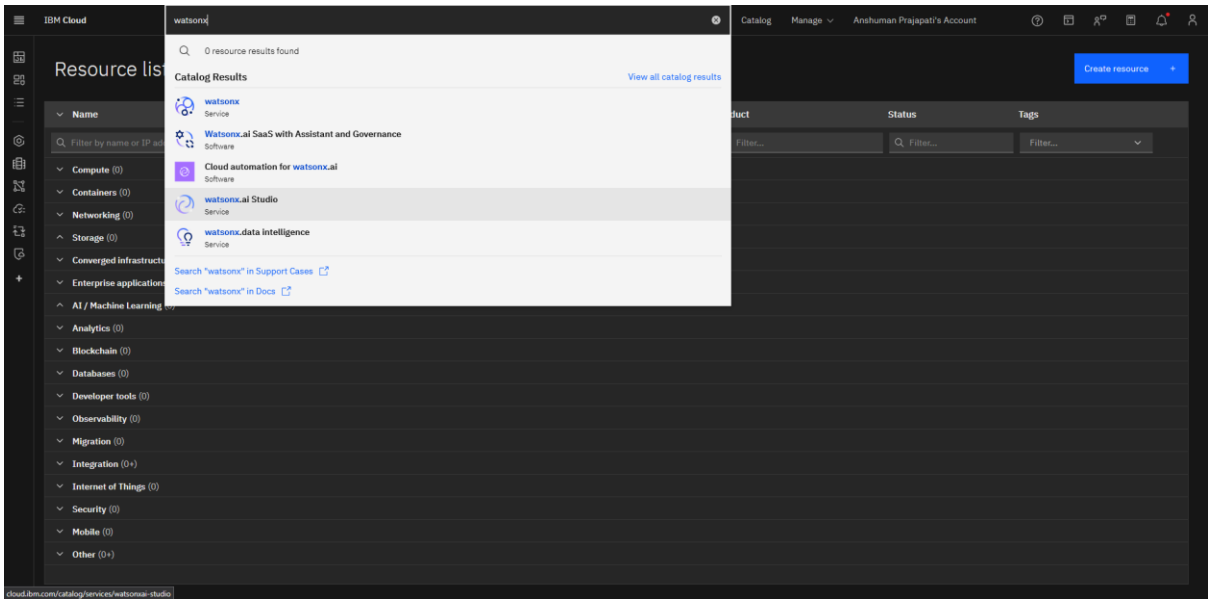
4. In this page if any resource is in use delete it and make it free.



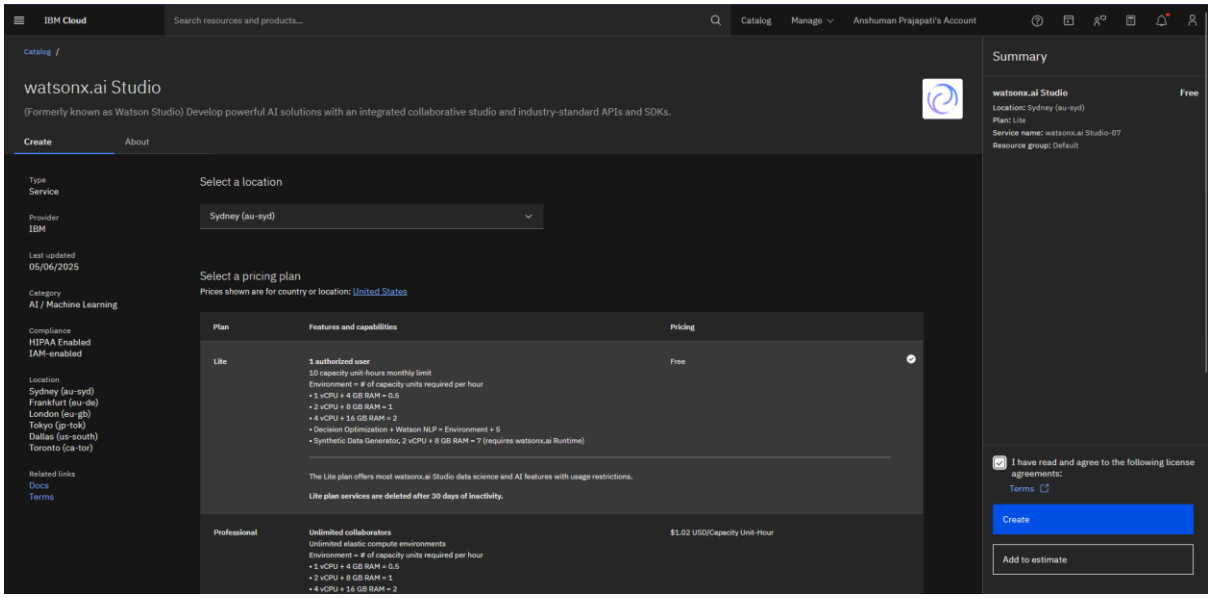


5. Repeat this for all the occupied resources and make sure no resources are in use.

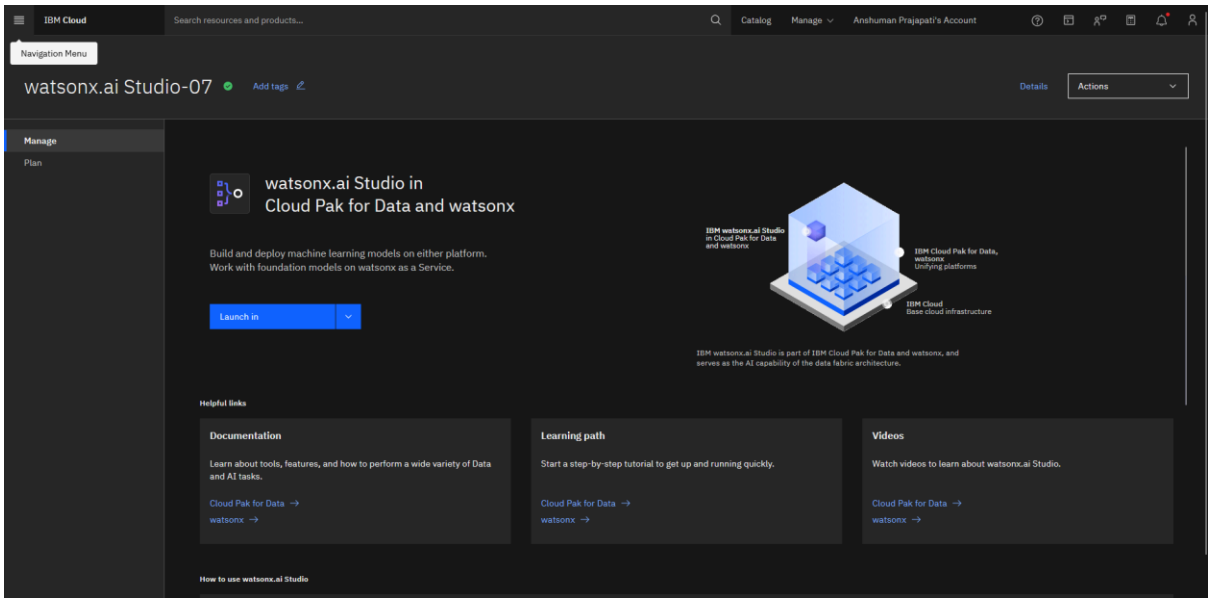




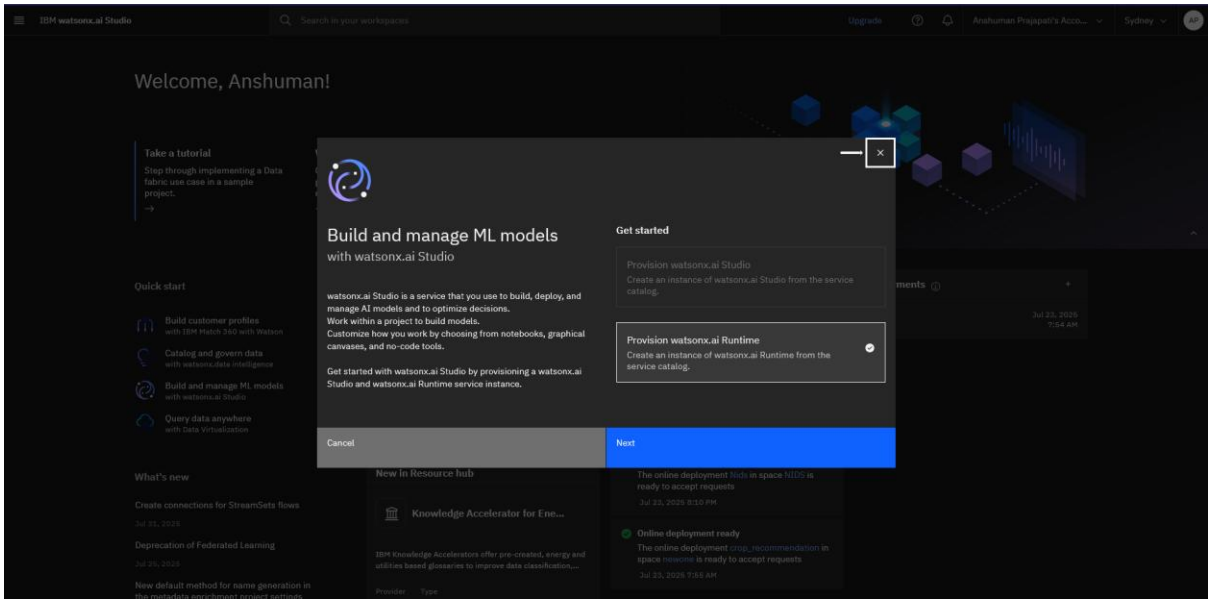
6. Now click search for watsonx.ai Studio in search bar and press enter.



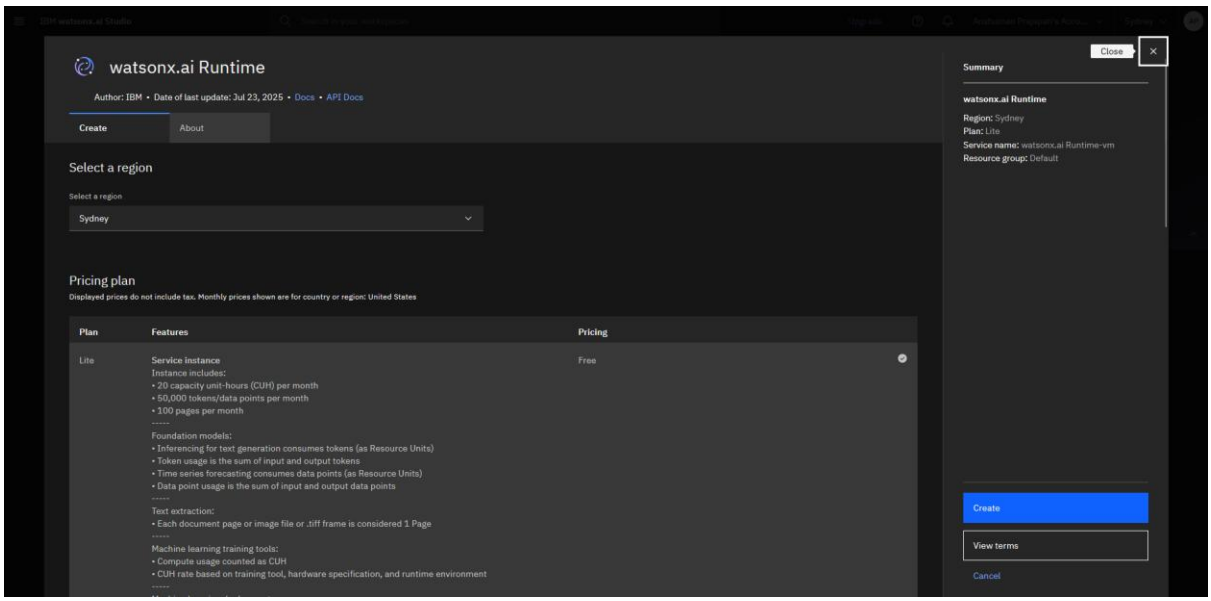
7. Now create with lite plan and and make sure to check on license agreements.



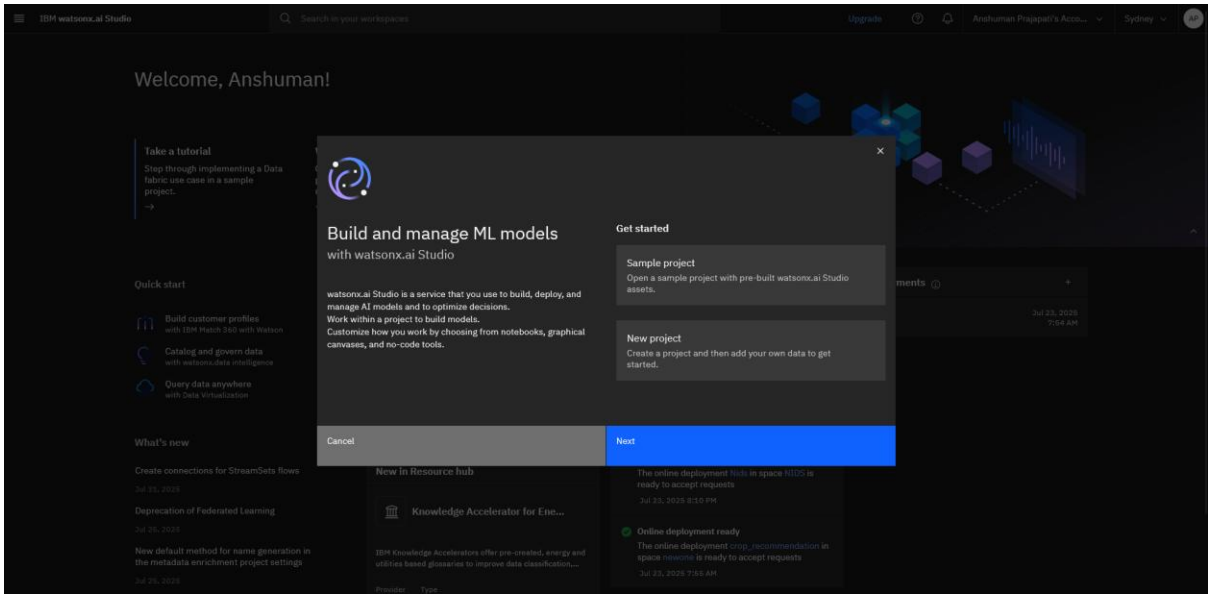
8. Now click on launch in.



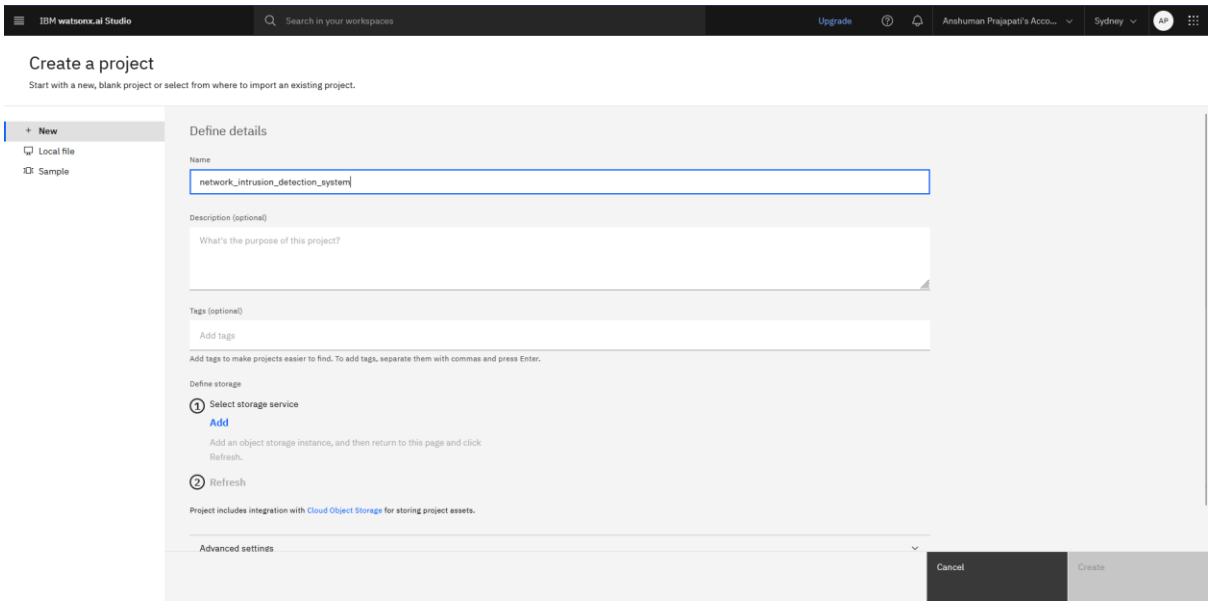
9. Now select provision watsonx.ai Runtime and press next.



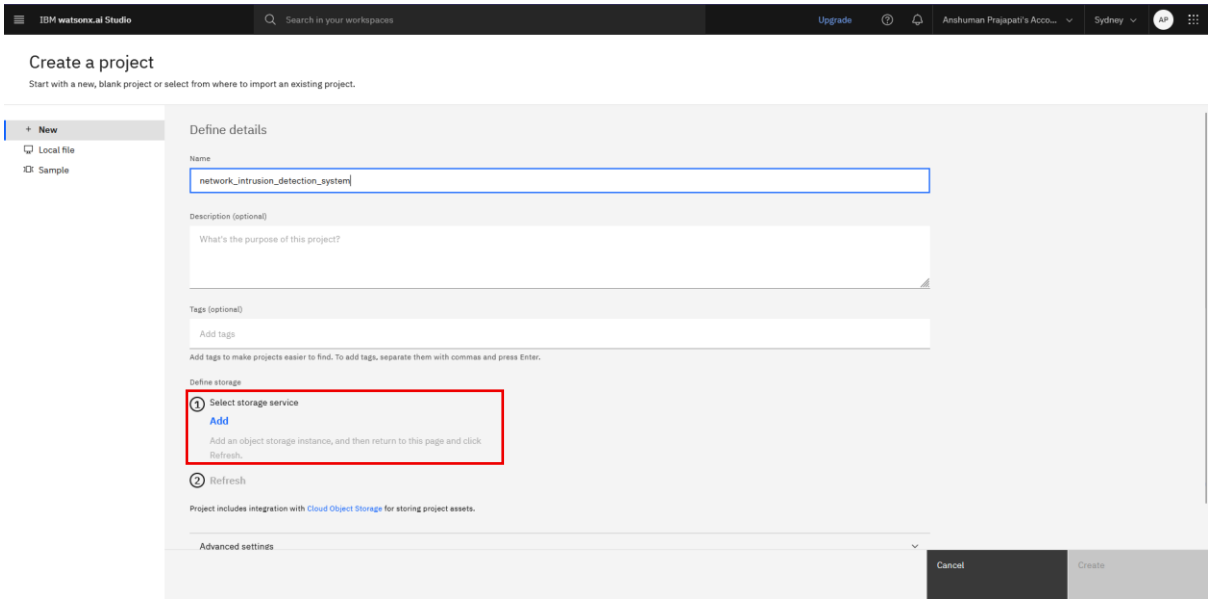
10. Press create.



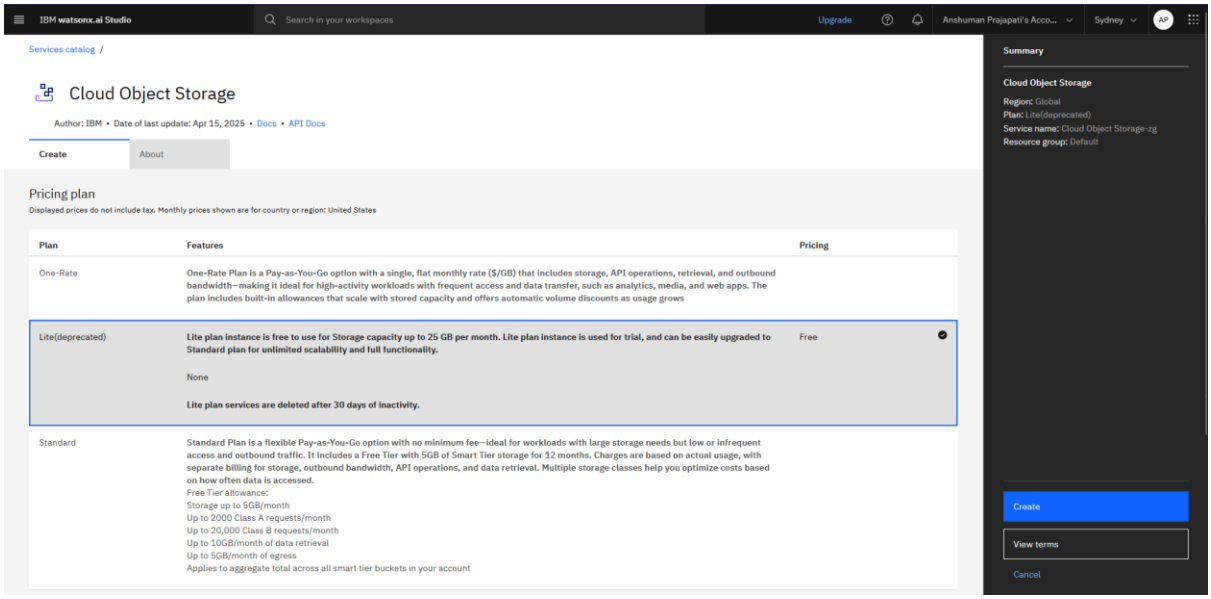
11. Select new project and press next.



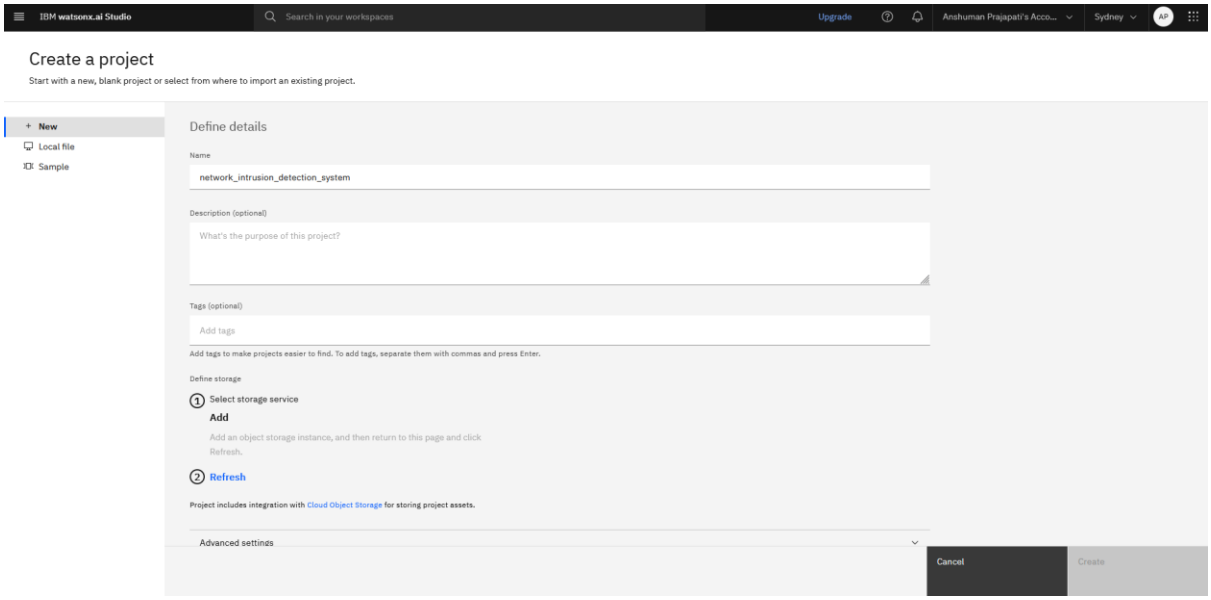
12. Give the project a name.



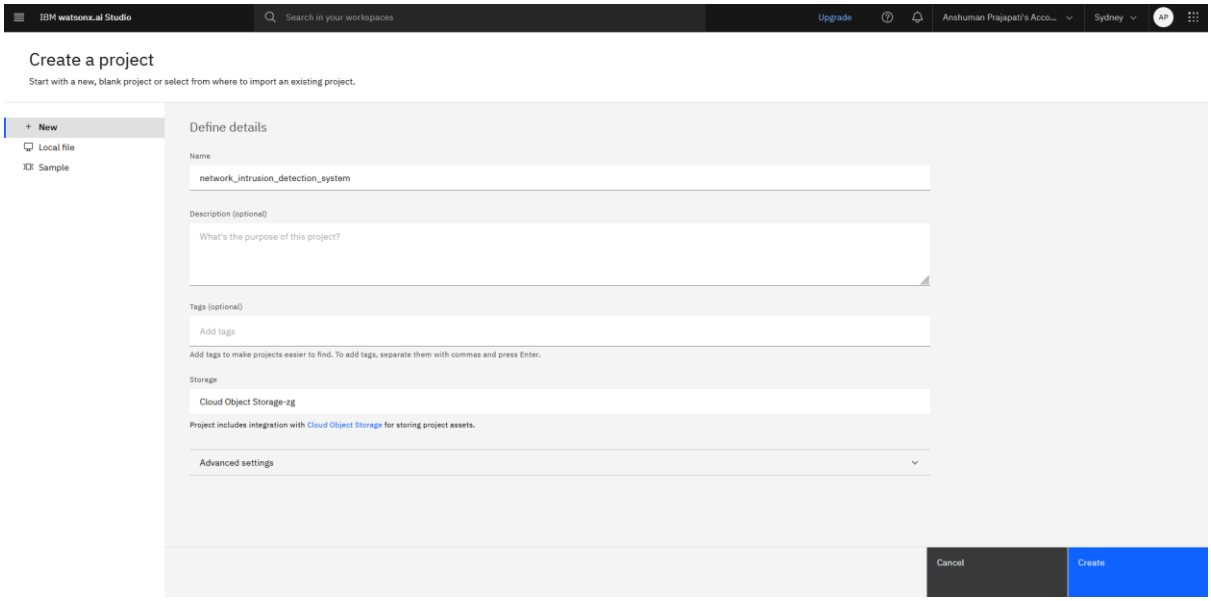
13. Now add storage to it.



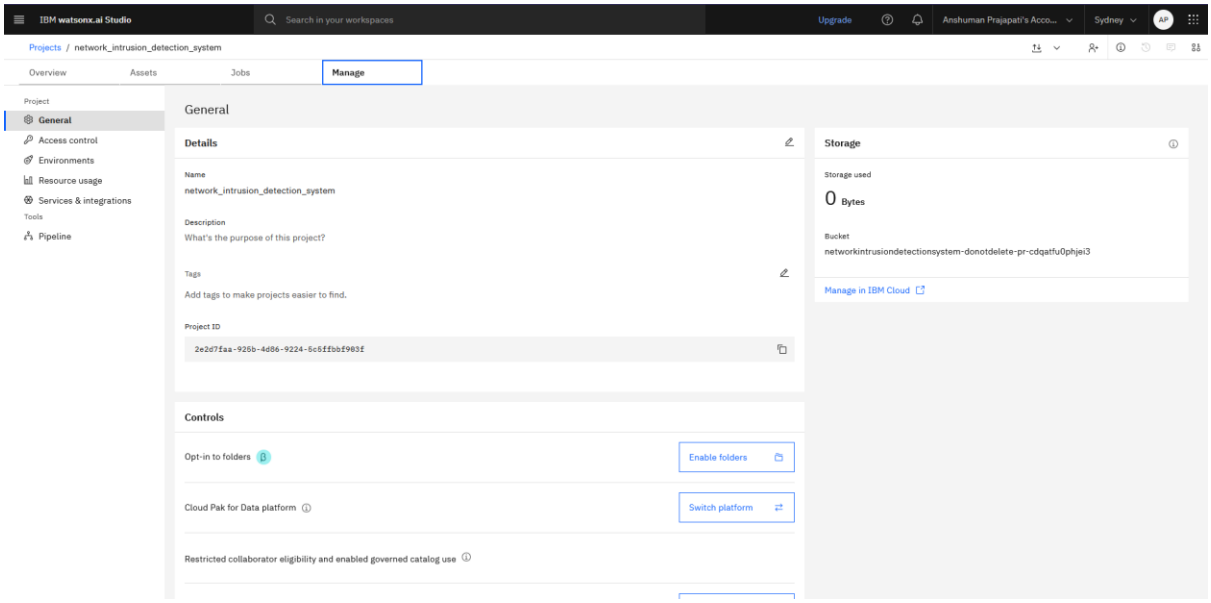
14. Create a object storge in lite plan.



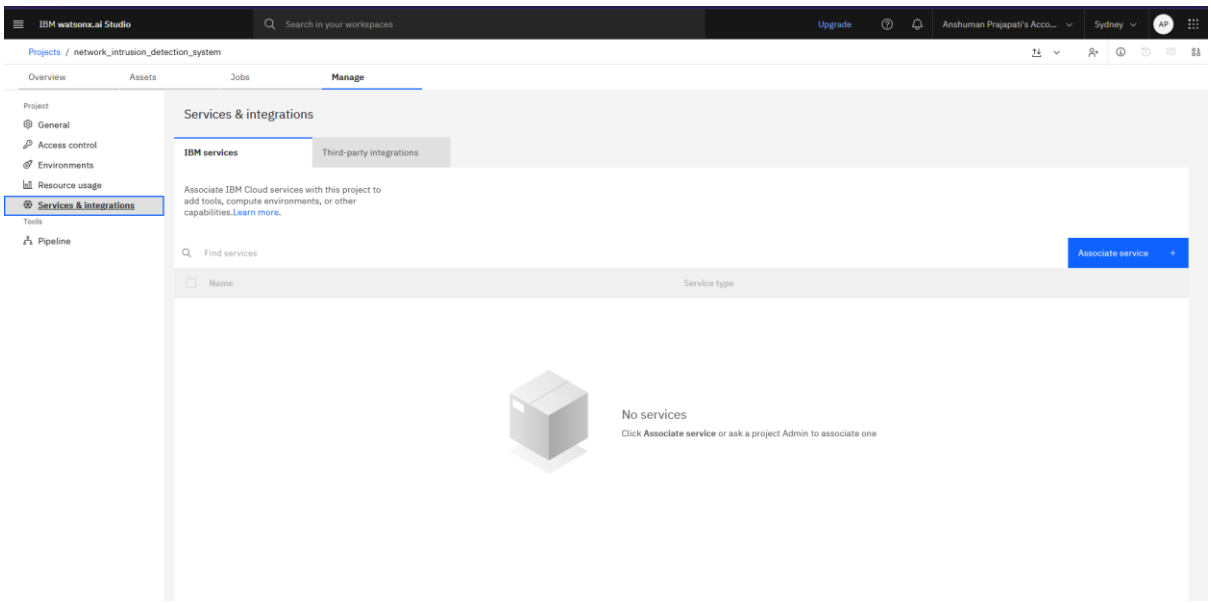
15. Press refresh to show up the storage.



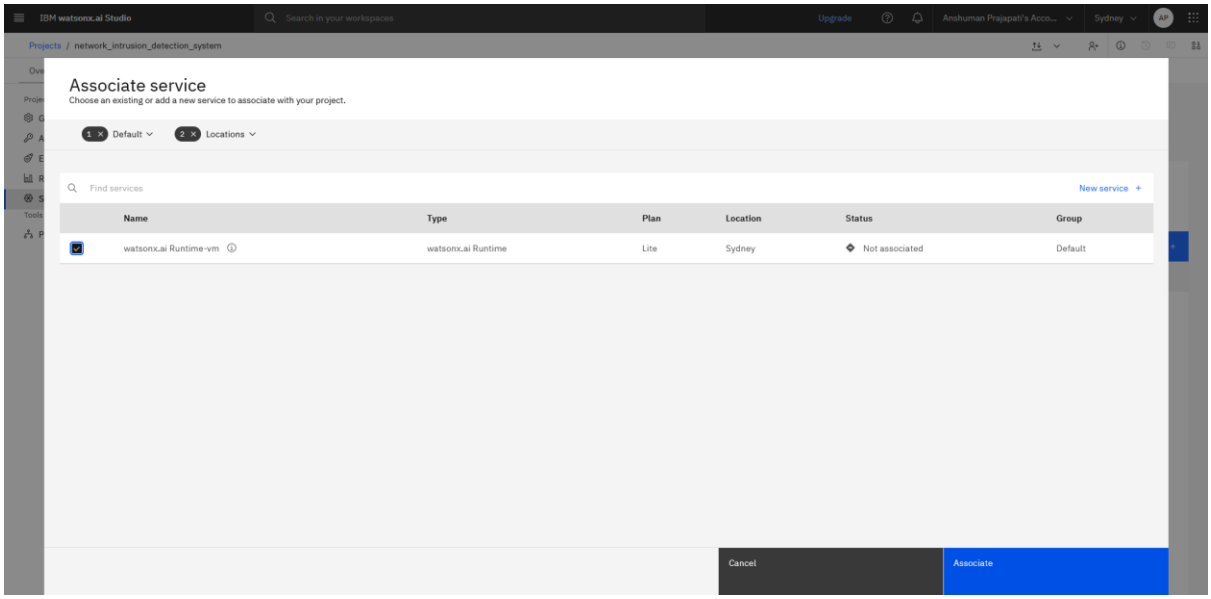
16. Press create.



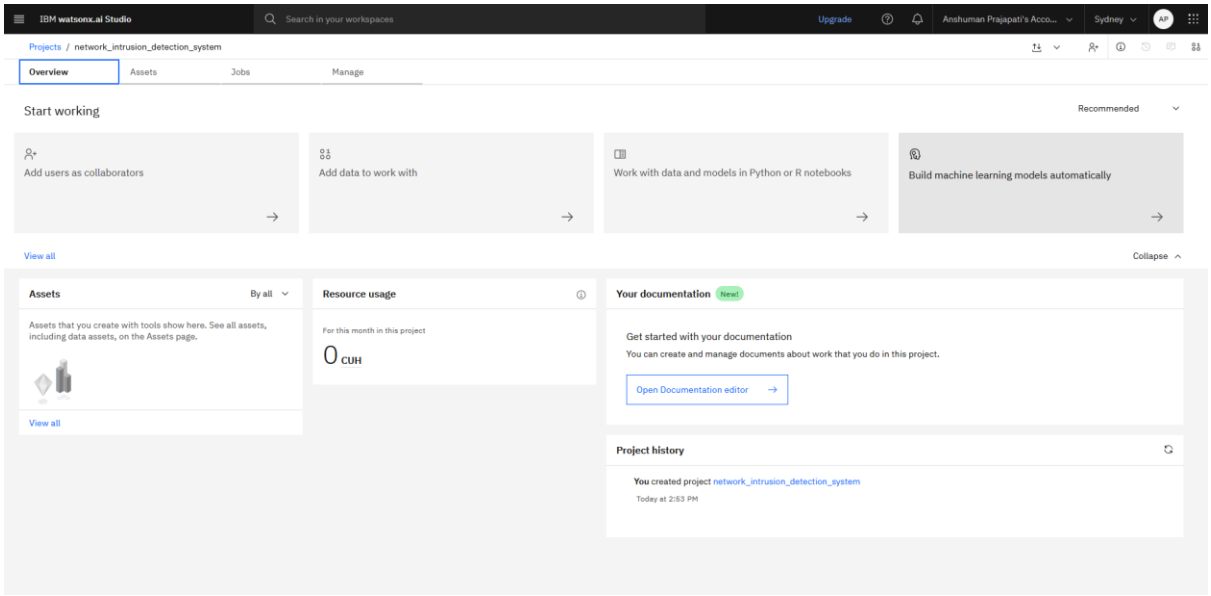
17. Now on manage tab go to services and integration.



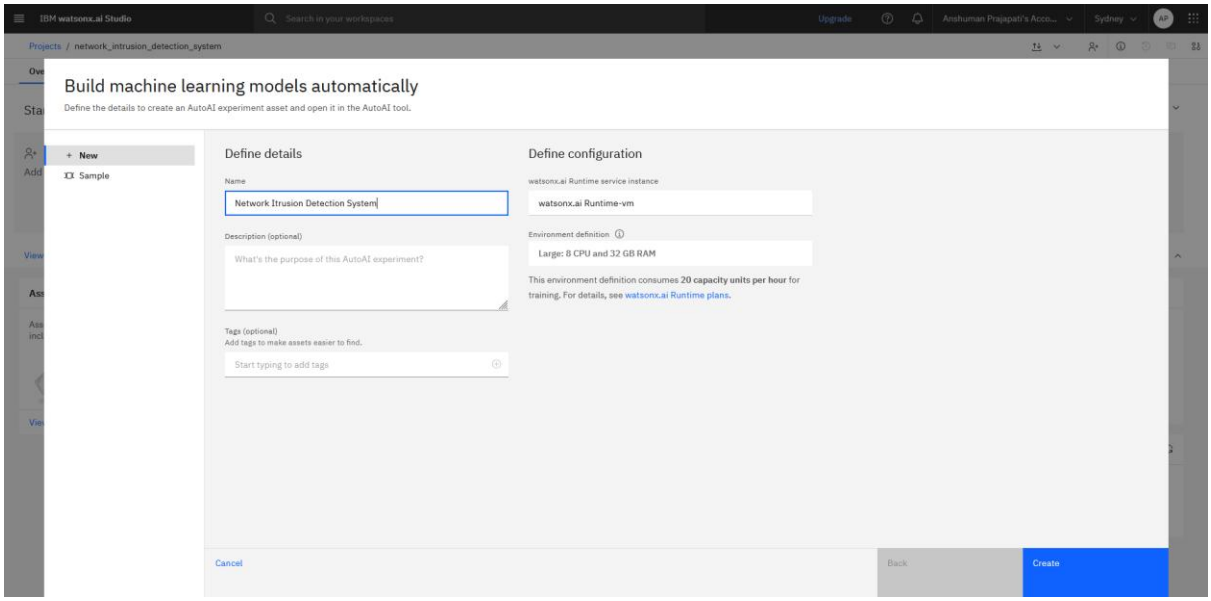
18. Now click on associate service.



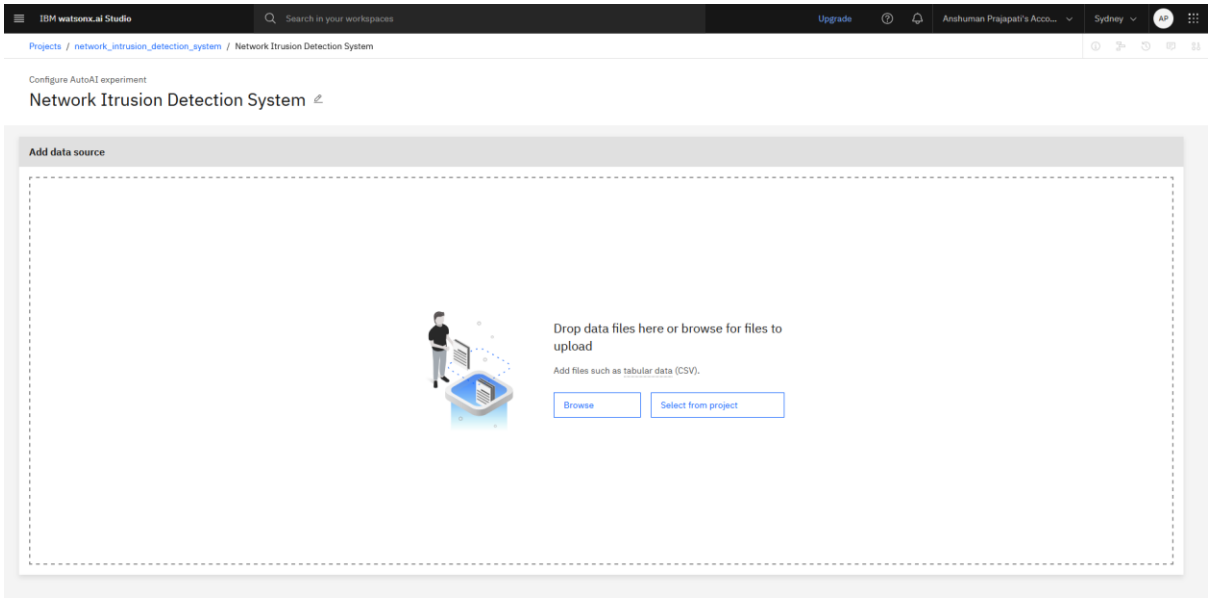
19. Check on watsonx.ai Runtime service and associate it.



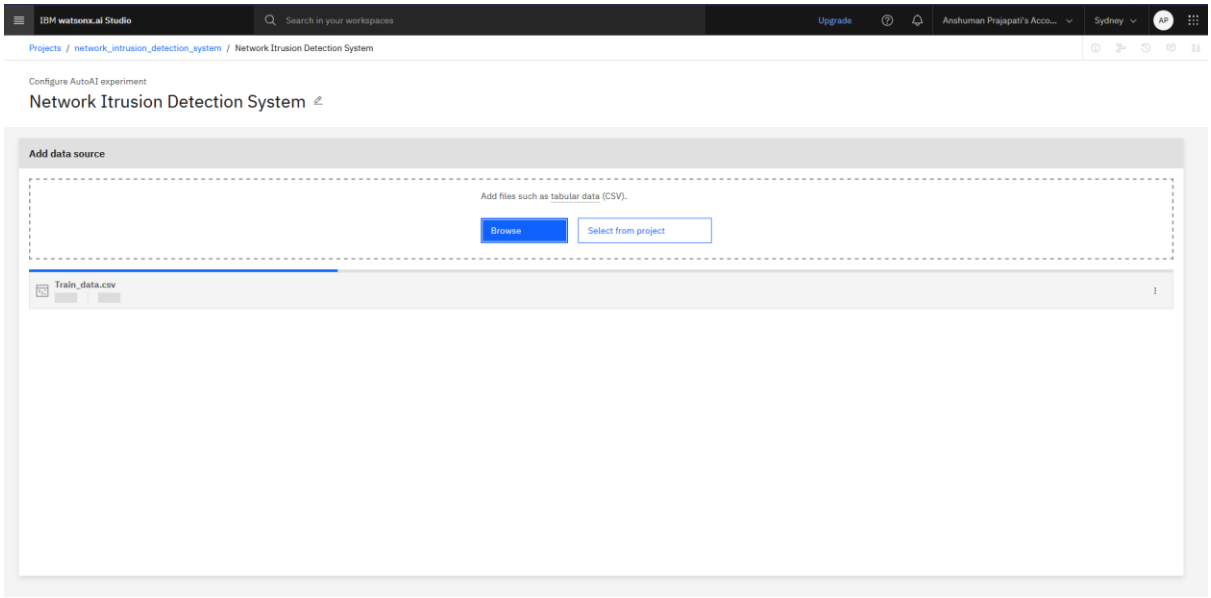
20. Now comeback to overview tab and select ***Build machine learning models automatically***



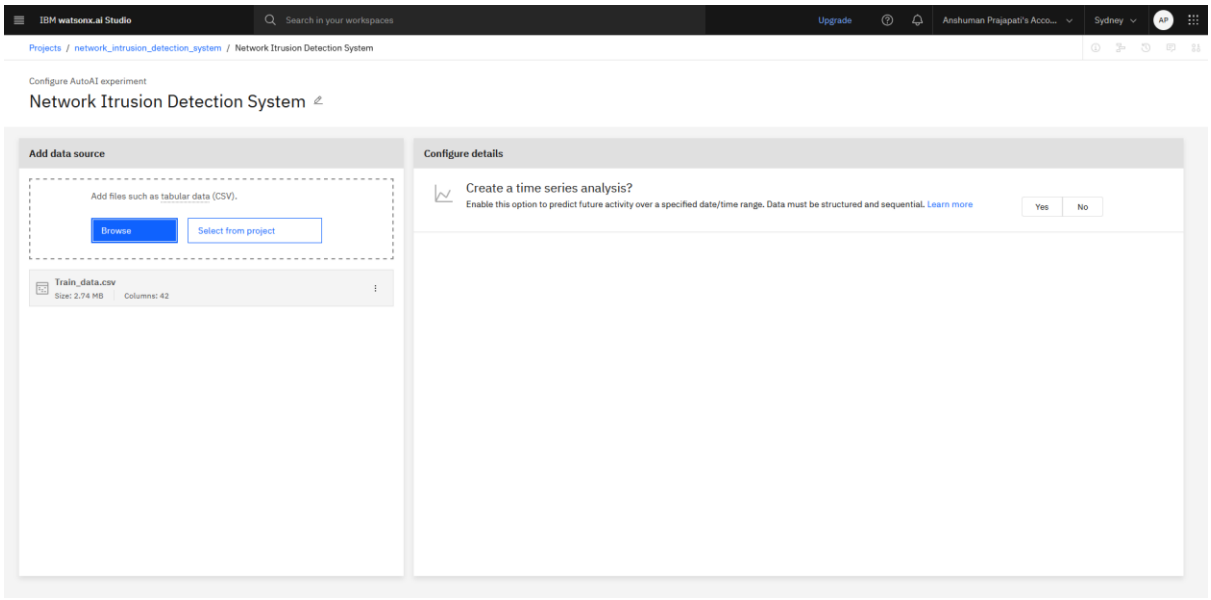
21. Give the experiment a name and click on create.



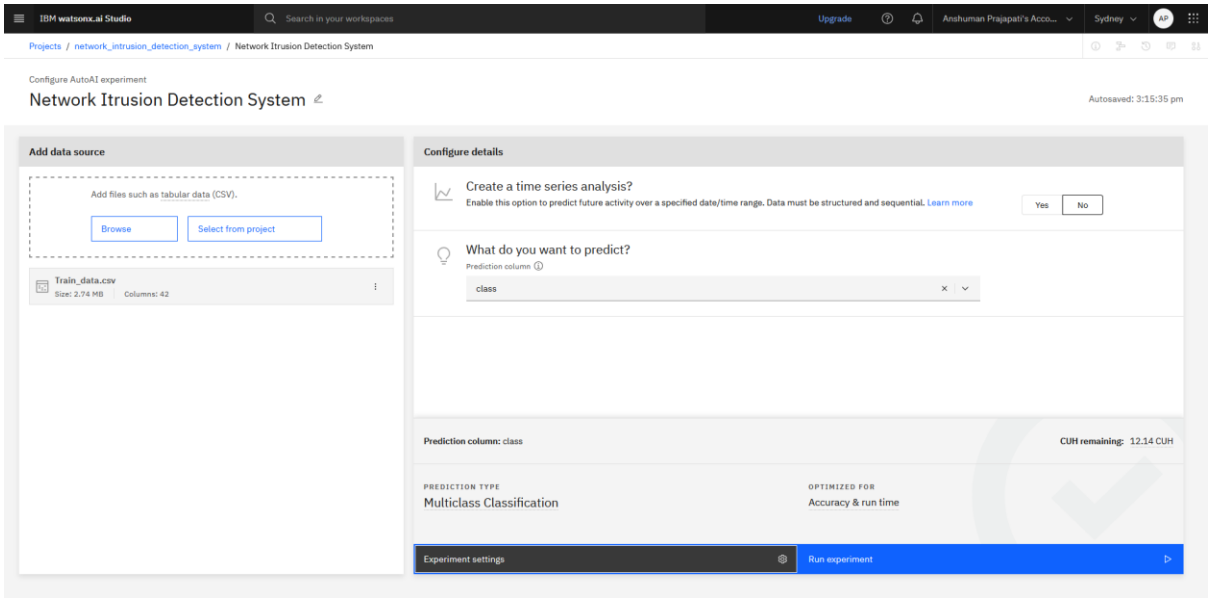
22. Now browse the file to upload (i.e. the .csv file to train the model).



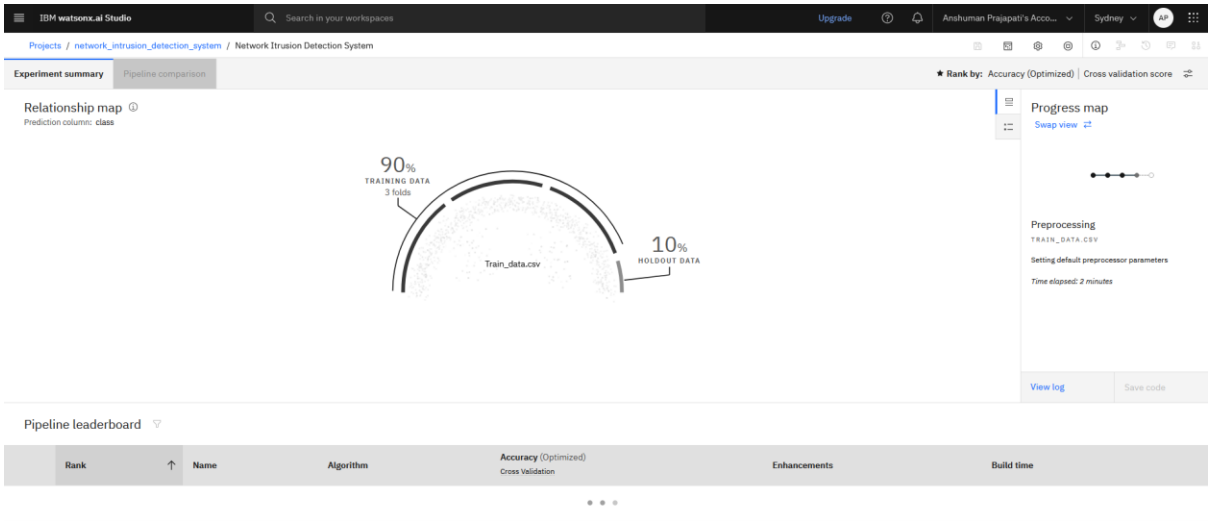
23. Wait for the file to upload.



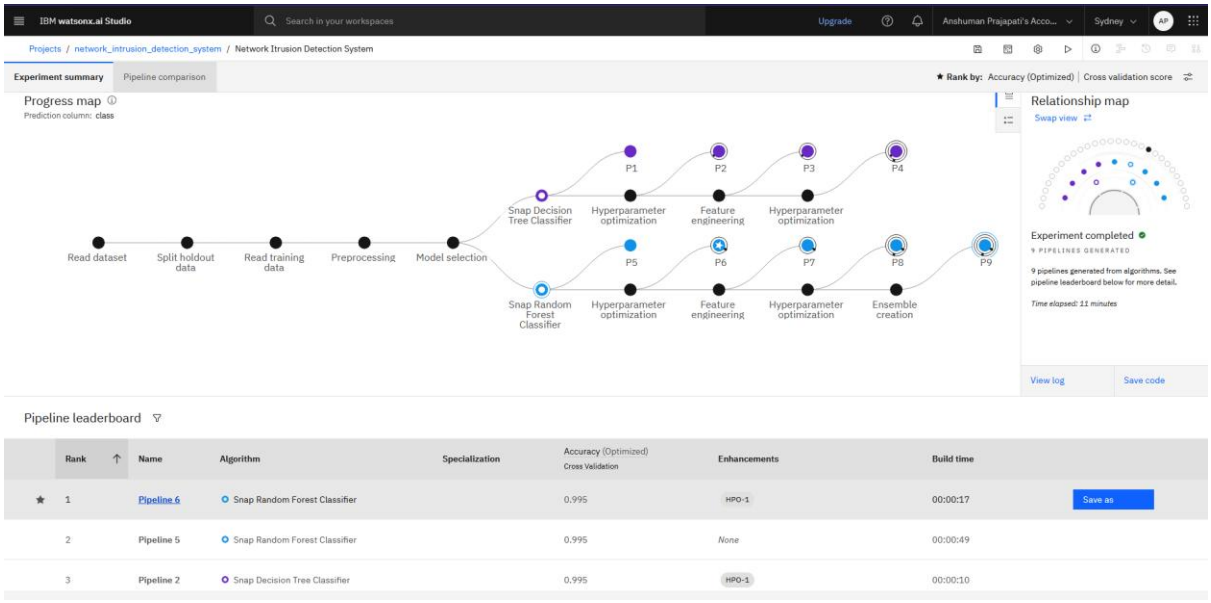
24. Now after the file gets completely uploaded it will ask for time series analysis select ***no***.



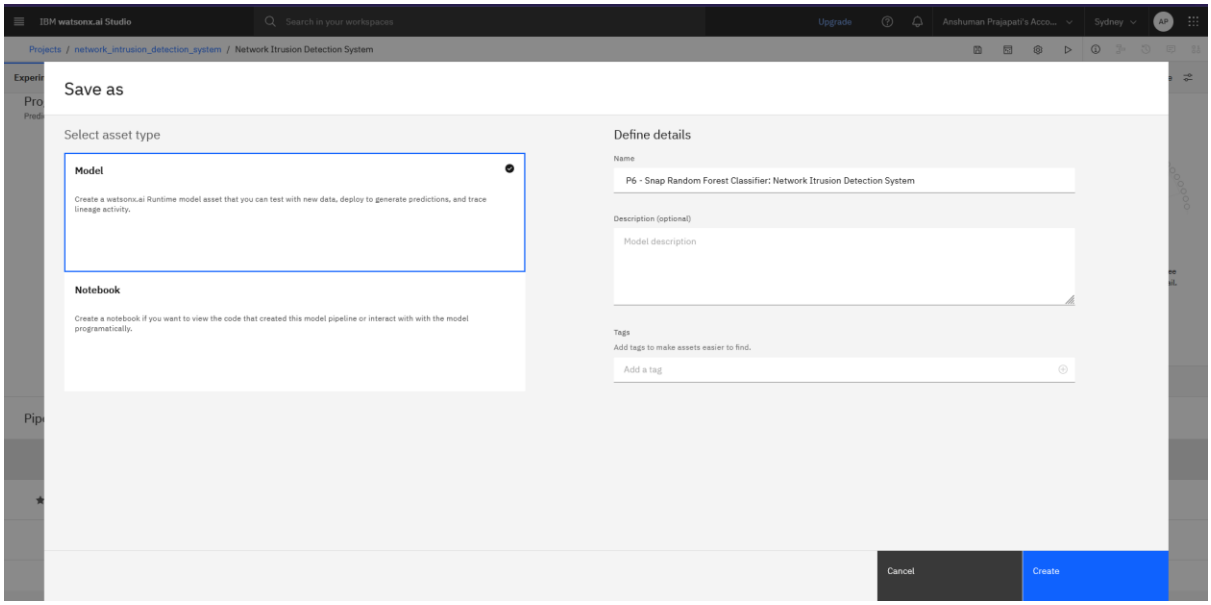
25. Select the prediction type (in this case its class) and make sure the prediction type is multiclass classification(as prediction column contains multiple distinct categories) and rum experiment.



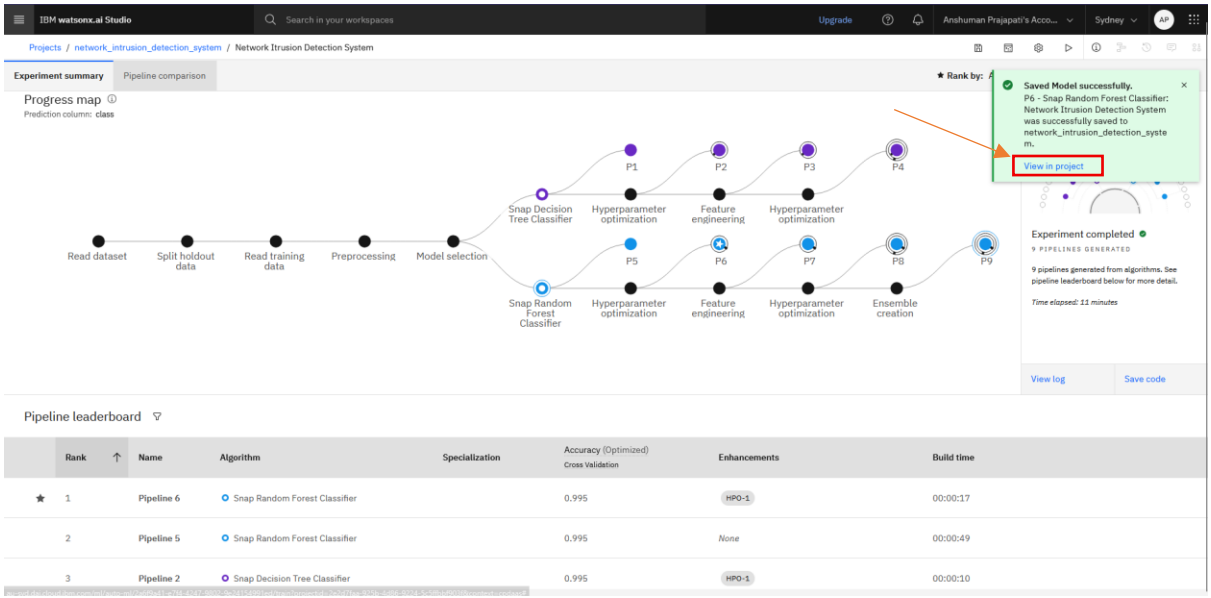
26. It will take some time to perform the experiment.



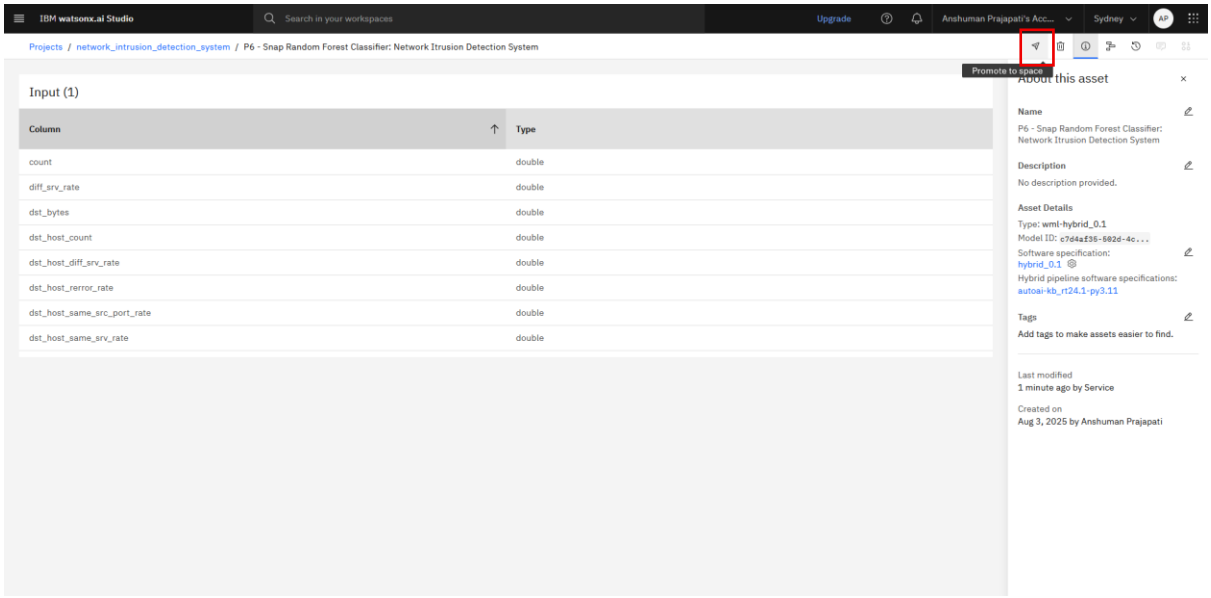
27. Now after the experiment is completed we can see the best performing algorithm is Pipeline 6 with enhancement of hyperparameter optimization now click on save as.



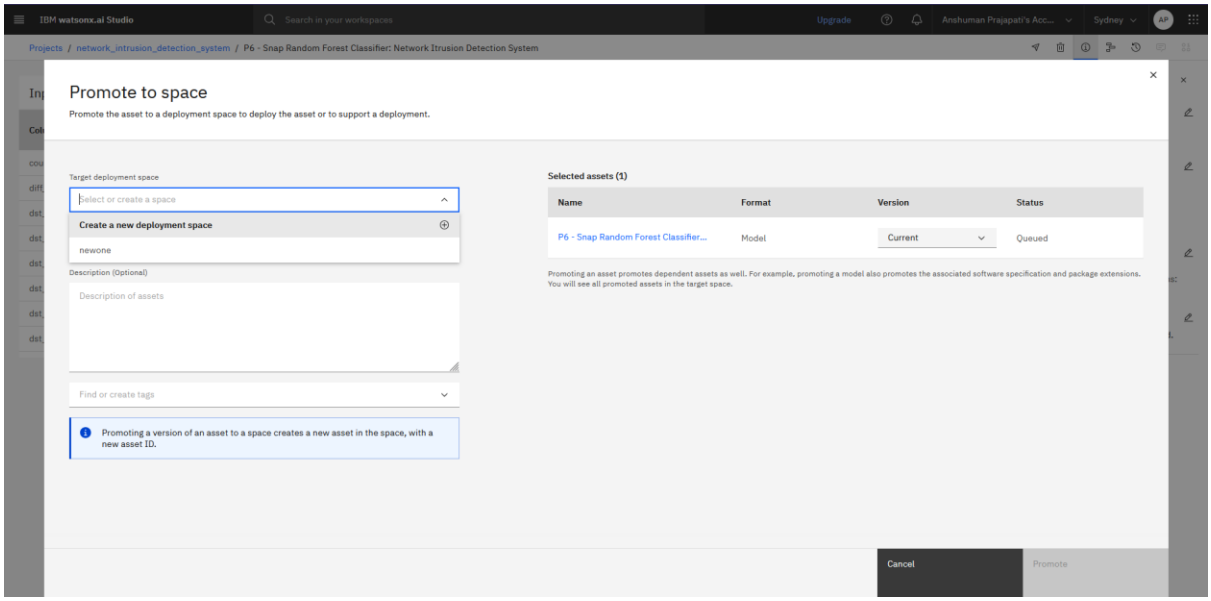
28. Press on create.



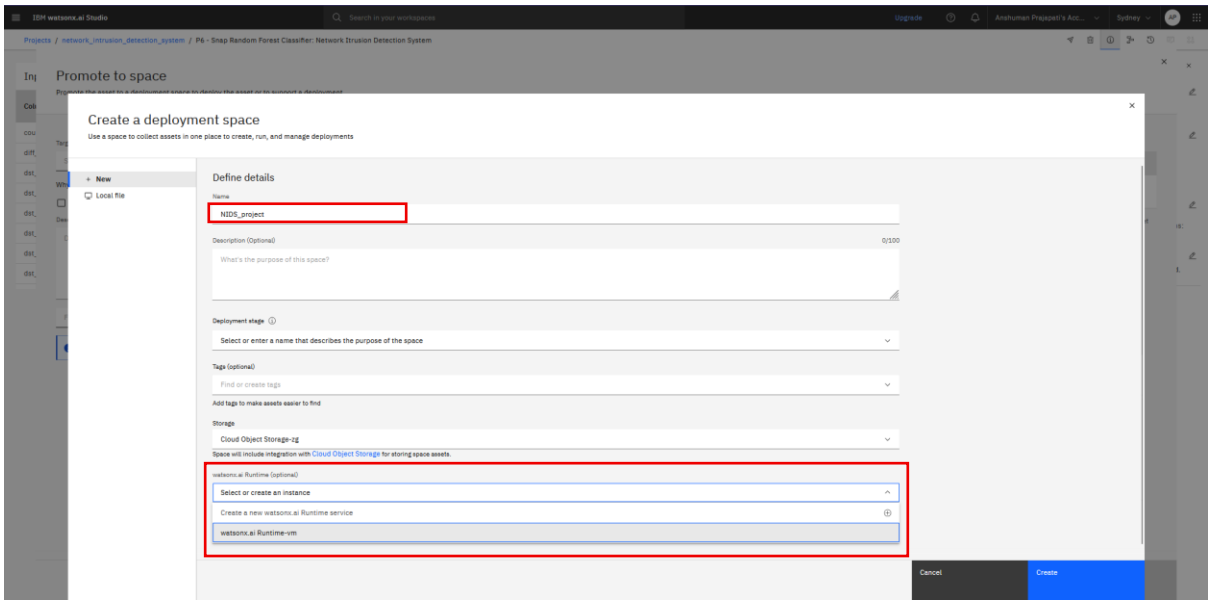
29. After the model is saved click on view in project.



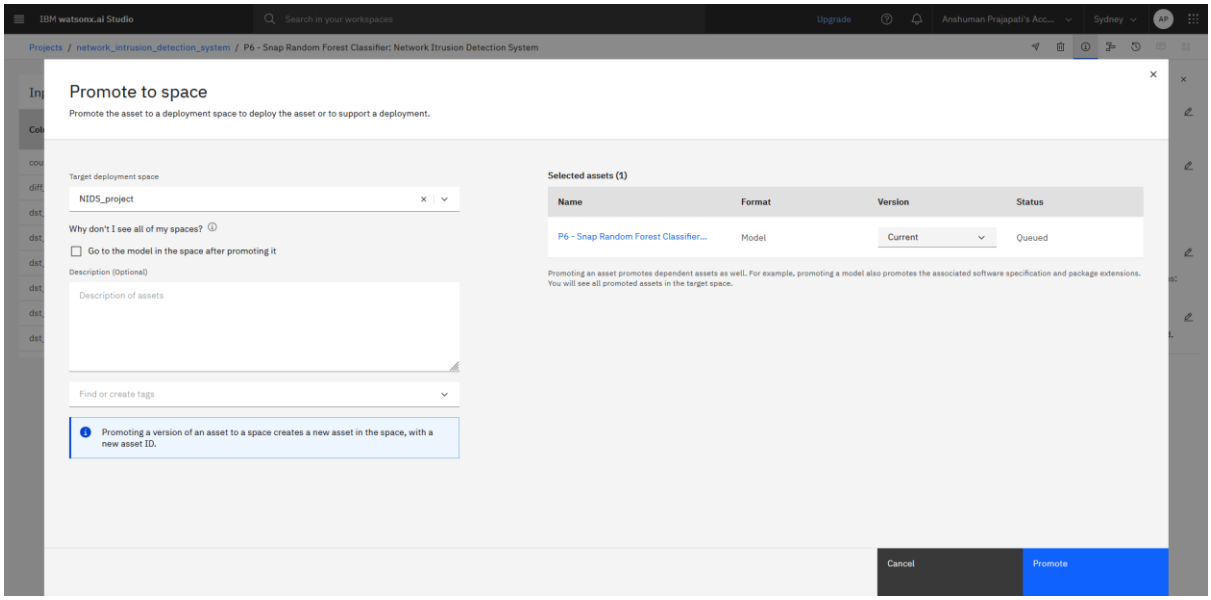
30. Now on the top towards right press on promote to space.



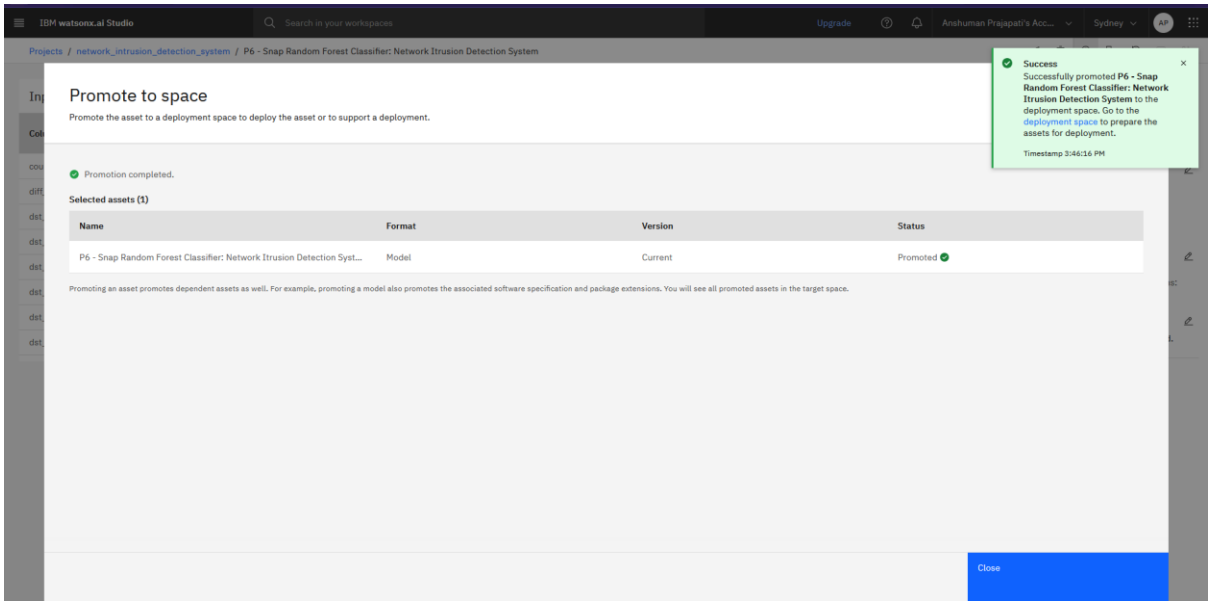
31. Select create new deployment .



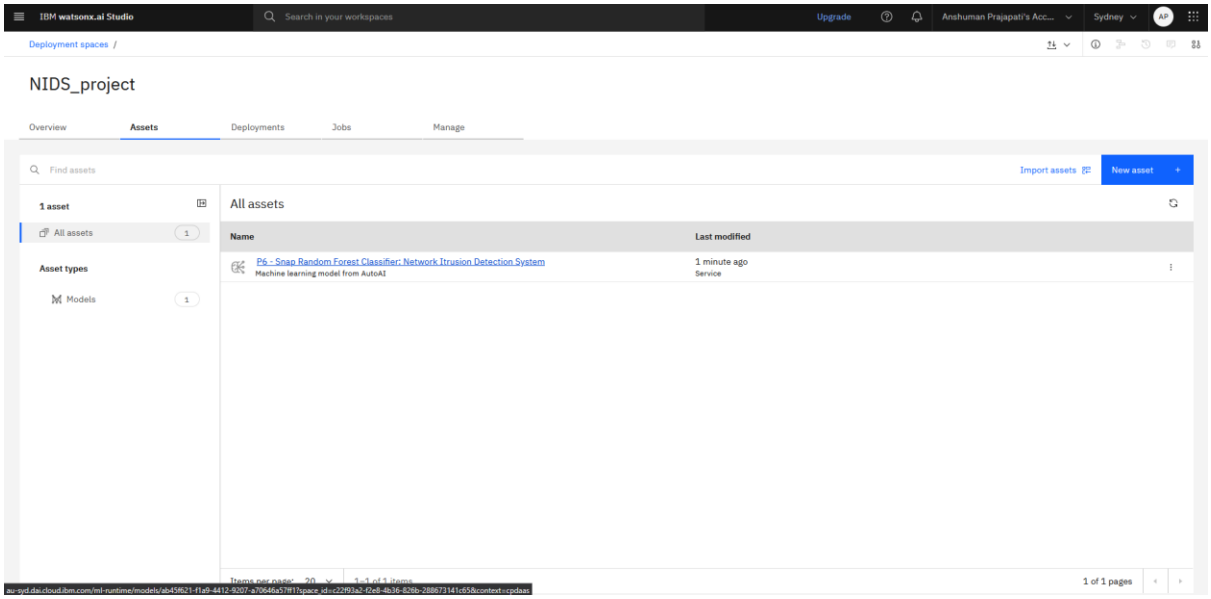
32. Name the deployment space and make sure to select instance of watsonx.ai Runtime and press create.



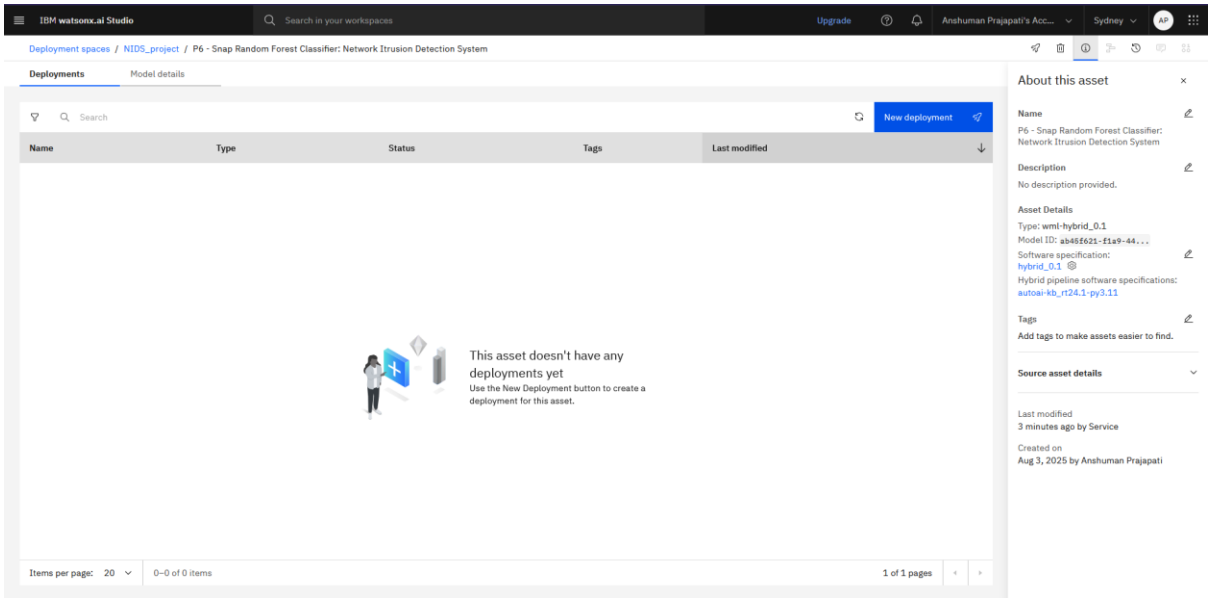
33. Now press on promote.



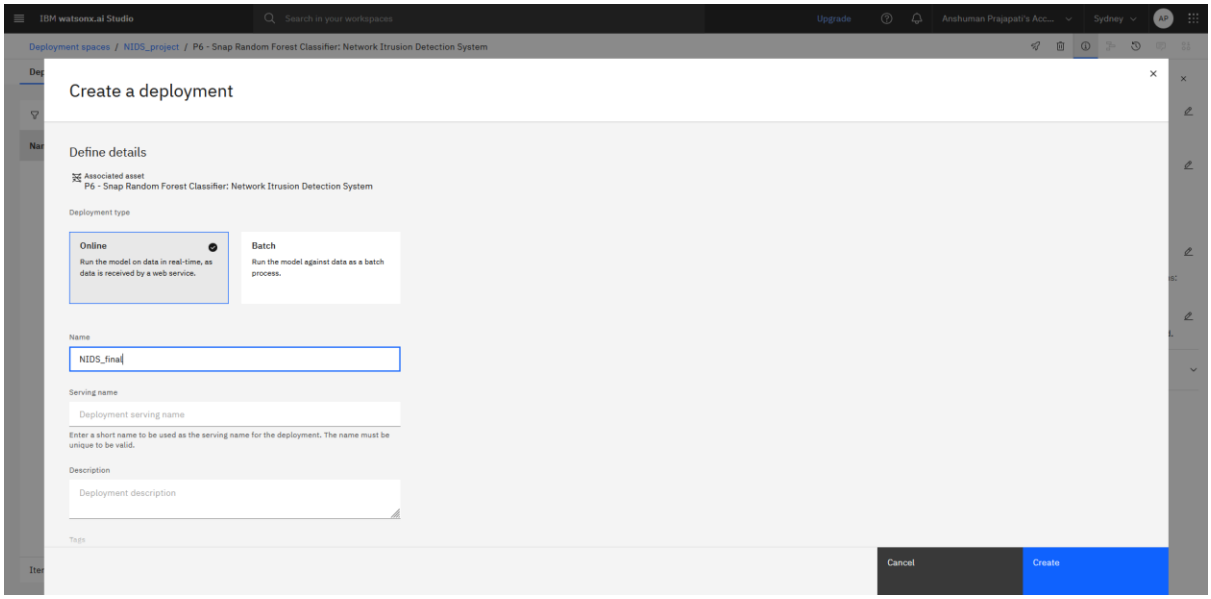
34. Now after it got promoted to space press on deployment space ,the pop up on top right corner.



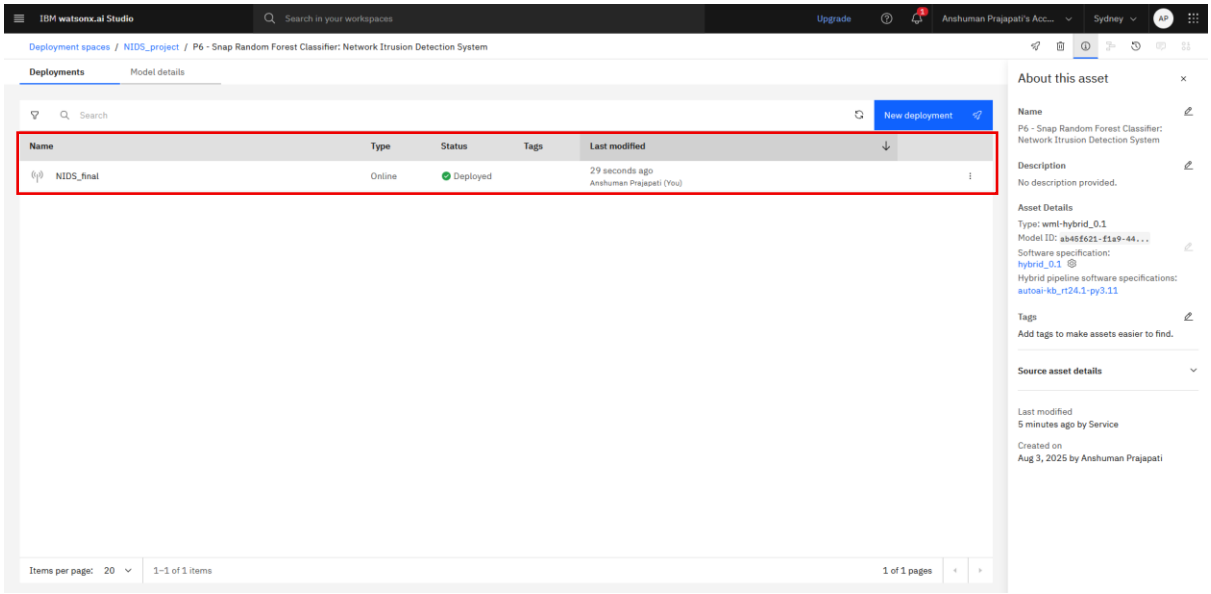
35. Now click on the asset.



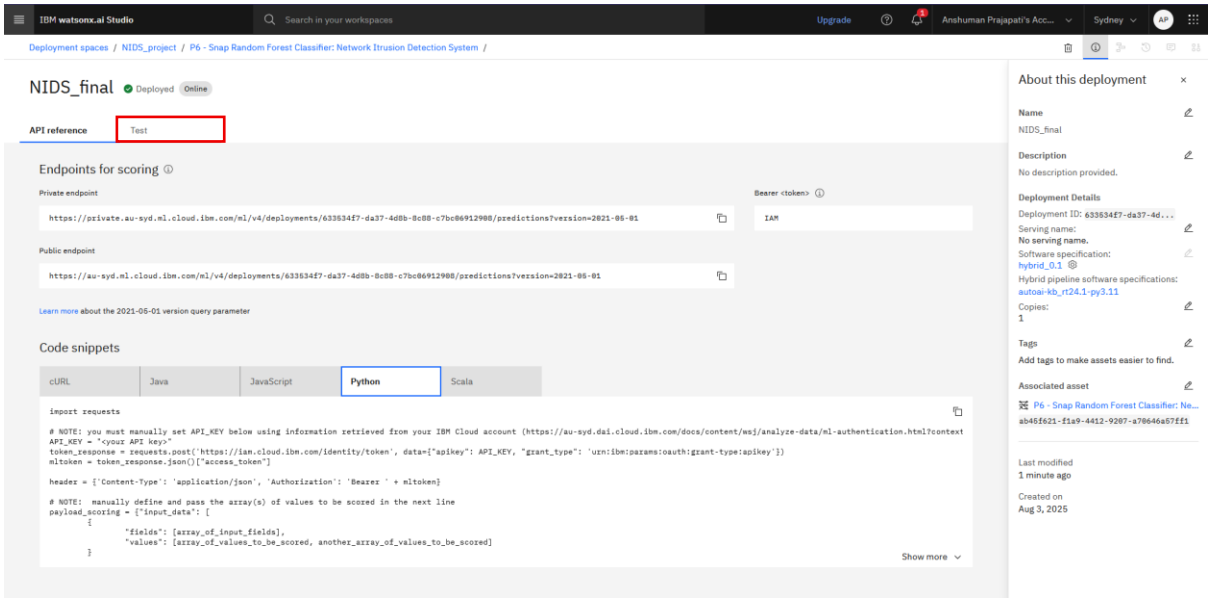
36. Now click on new ***New Deployment***



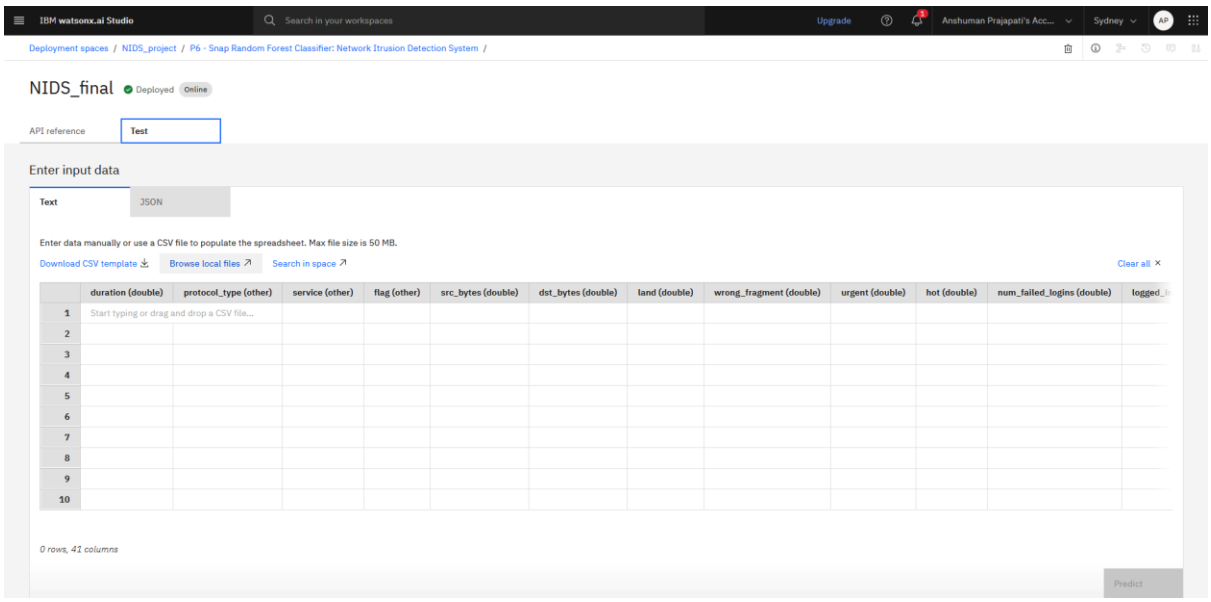
37. Now select online and give this final deployment a name and press create.



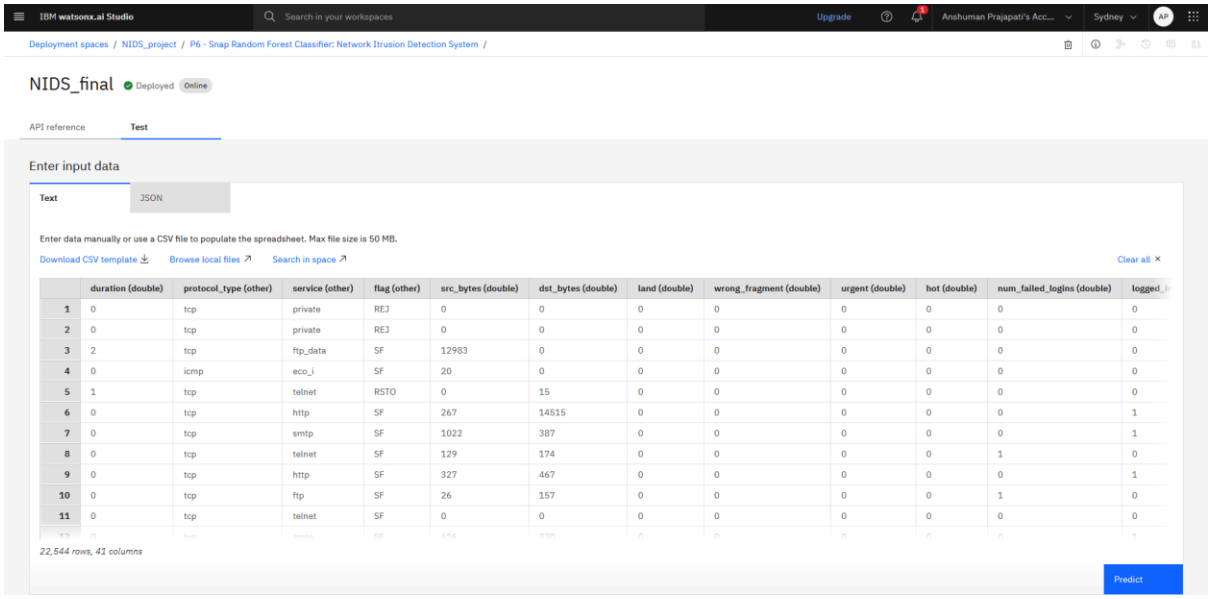
38. After some time the deployment will be completed, then press on the deployment name.



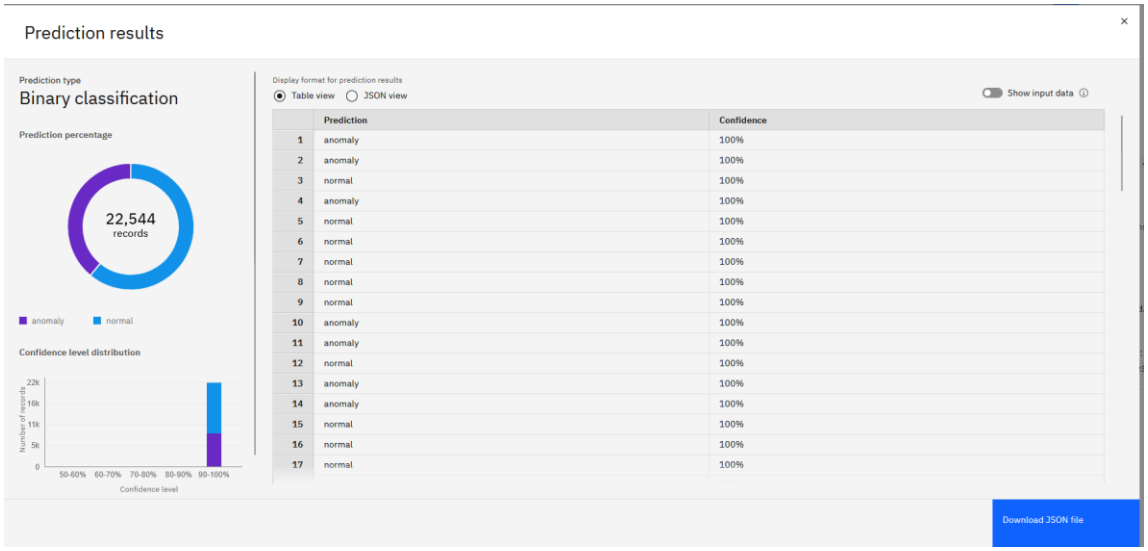
39. This will give the API reference page where API's are present to implement this model with other projects. Now press on test to test the model.



40. We have many options to give input for test ,I choose local file as I have one.



41. After the file is loaded press predict. It will take some time to generate the result.



42. This will generate a final result with up to 100% accuracy.

For upgrade of different types of attack detection just the data(.csv file) on which model is trained is to be changed for new data(enhanced.csv file) provided above and also you can refer this GitHub repo (the enhanced files).