Deploying Machine Learning Models in IBM Watson Studio Cloud as APIs

Introduction

- Assuming that we've trained a good model, and want to deploy it, the immediate challenge is: how to represent the built model and deliver it for the deployment step. Here are a couple of options:

  Model coefficients transfer approach, such as rewriting the scoring process with model coefficients in a "production language" like Java. This often causes the model outcomes to change and it takes a significant amount of time.

- Model native serialization method, such as "pickle" of Python, and "save" function of Spark. It requires compatible environments on development and production like programming language, dependencies. It also leads to difficult configurations in production, and it might be slow and heavyweight.

-

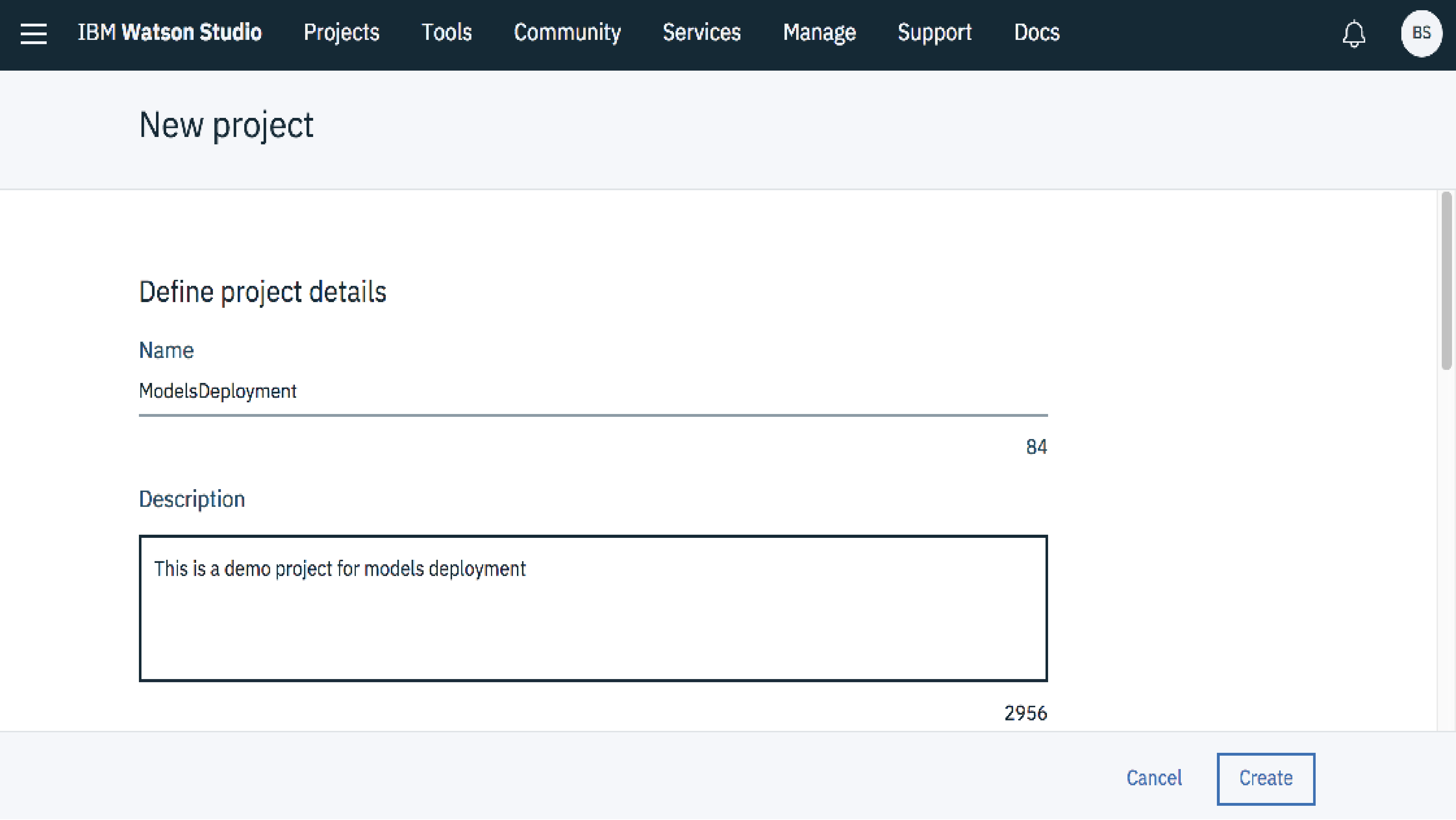
- Predictive Model Markup Language (PMML) is the leading standard for statistical and data mining models and supported by over 20 vendors and organizations. PMML allows different statistical and data mining tools to speak the same language. With PMML, it is easy to develop a model on one system that uses one application and deploy the model on another system that uses another application.
- The models in PMML format provide flexibility to users, and in this blog, we discuss the deployment of models in PMML format.

  IBM Watson Studio is an integrated environment that is designed for AI and machine learning. It supports the whole process of machine learning and data mining practice: from data capture, data cleaning, data preparation, to feature engineering. Watson studio also does the model building via various machine learning technologies, for example Spark, Python and SPSS, and the model deployment as well as the model monitoring and retraining. When we say model deployment, Watson Studio integrates the SPSS Scoring Engine, which is a PMML-compliant engine. It is a solid pure Java library, and picks up many optimizations of computation, serves SPSS family more than ten years.

- Association Rules
- Cluster Models
- General Regression
- K-Nearest Neighbors
- Mining Model
- Naïve Bayes
- Neural Network
- Regression
- Ruleset
- Scorecard
- Trees
- Vector Machine
- These models can have different versions and can be started via REST APIs. In this session, you will learn how to deploy PMML models in the IBM Watson Studio cloud and how to use them efficiently in applications.

IBM Watson Studio

- Click the IBM Watson link in the header to navigate to the Watson Studio home panel.
- Click **New project**.
- Choose a project type "Standard":
  - If you want to train complex neural networks using experiments, choose a "Deep Learning" project
  - For all other machine learning work, choose the "Modeler" project type
- If you don't already have any of the required services, such as Watson Machine Learning and IBM Cloud Object Storage, new service instances are created.



New project

Define project details

Name

ModelsDeployment

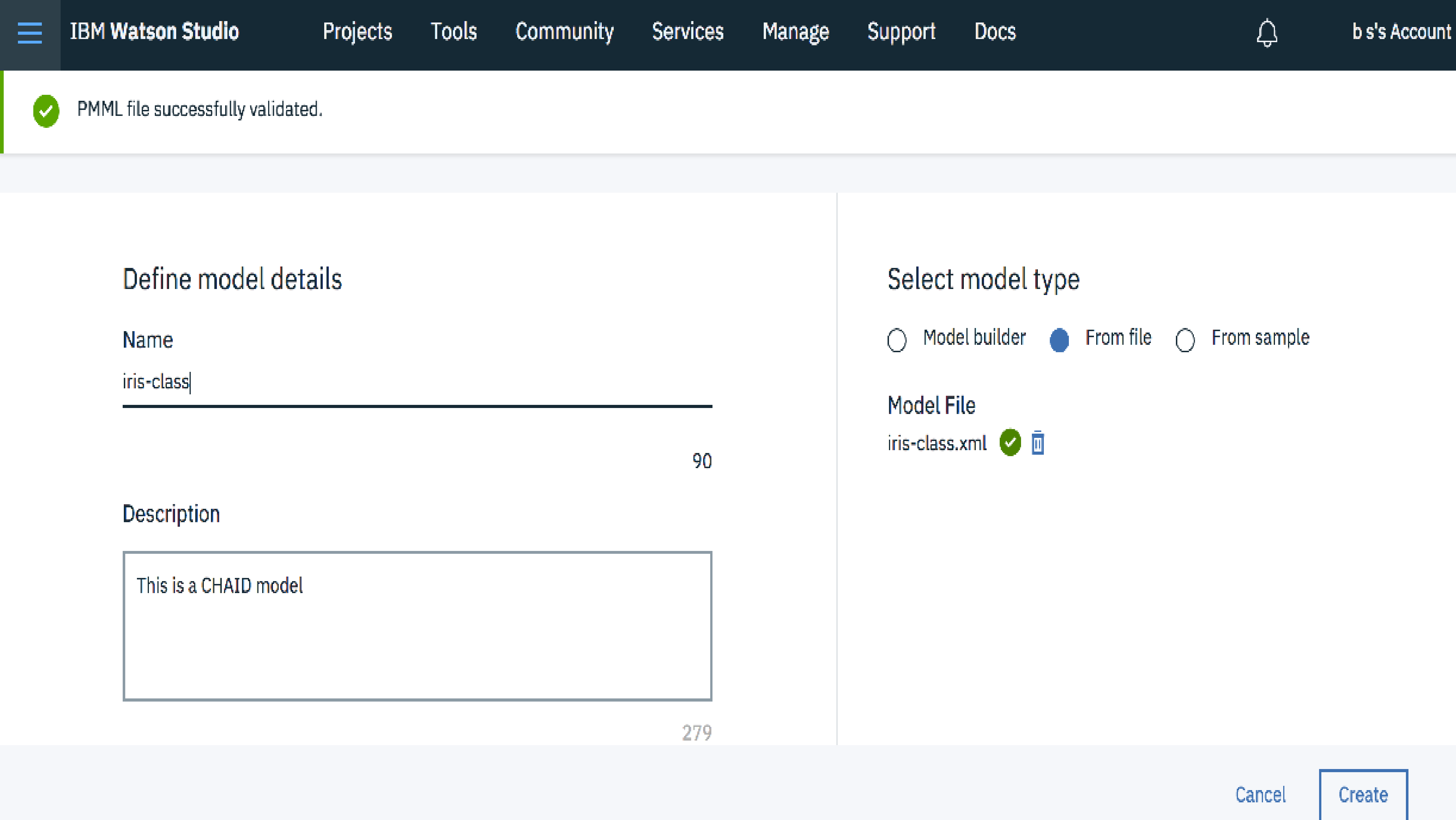
84

Description

This is a demo project for models deployment

2956

- **To upload a PMML model**
  1. From the Assets view of your project, click **New Watson Machine Learning model**.
  2. Select **From file** as the model type.
  3. Upload your PMML (.xml) file when you're prompted. In the example, my PMML model is exported from SPSS Modeler. It is a CHAID model based on the known Iris Data Set. The model will be validated.
  After PMML file is successfully validated, enter the required model name and optional descriptions.
  4. Click **Create**. The model page will be open once it is imported successfully.
  The default `Overview` page shows general information about the model, including model type, label column, and a list of features.



✓ PMML file successfully validated.

Define model details

Name

iris-class

90

Description

This is a CHAID model

279

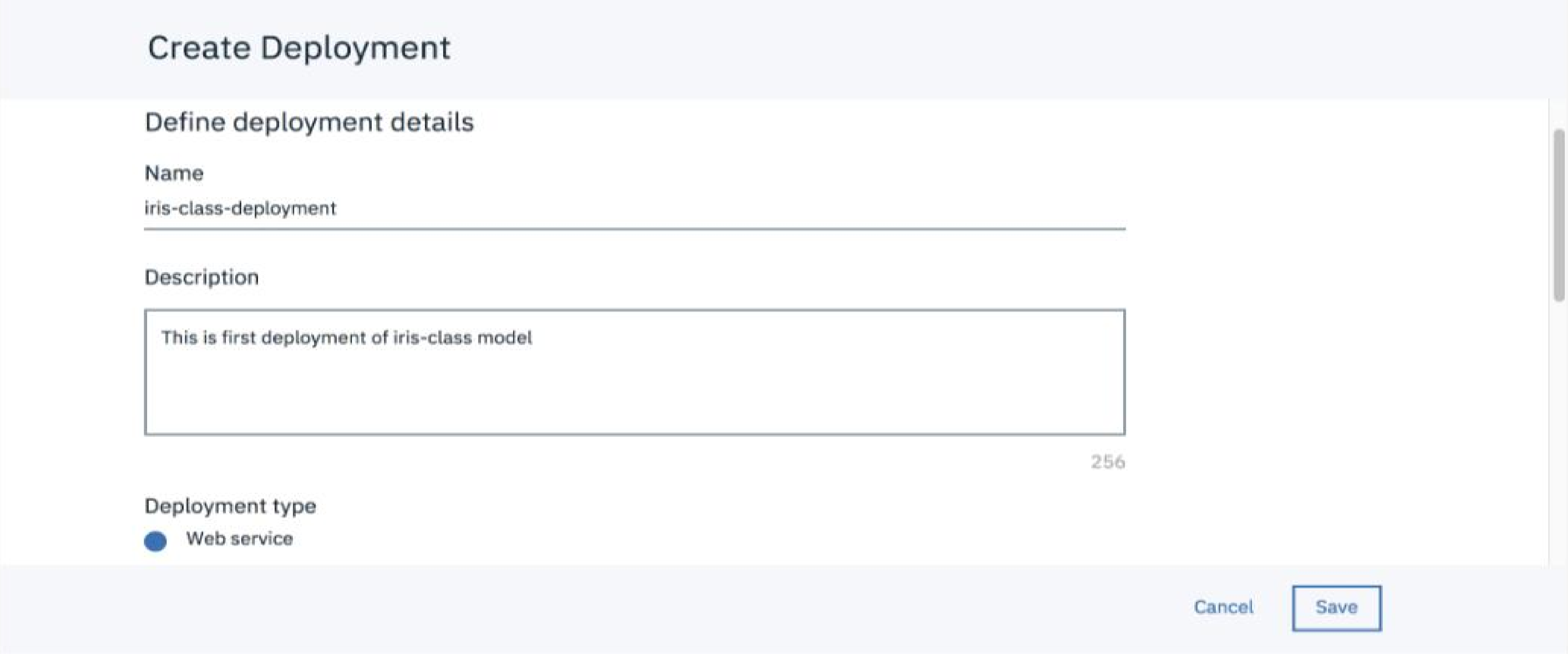
Select model type

○ Model builder ● From file ○ From sample

Model File

iris-class.xml ✓

- **To deploy the model**
- On the model page, click **Add Deployment**.
- Enter a deployment name, and optional descriptions.
- Select **Web service** deployment type and it exposes REST APIs for the deployed model. When the deployment phase is successful, we can open the deployment page to test the scoring function.



Create Deployment

Define deployment details

Name

iris-class-deployment

Description

This is first deployment of iris-class model

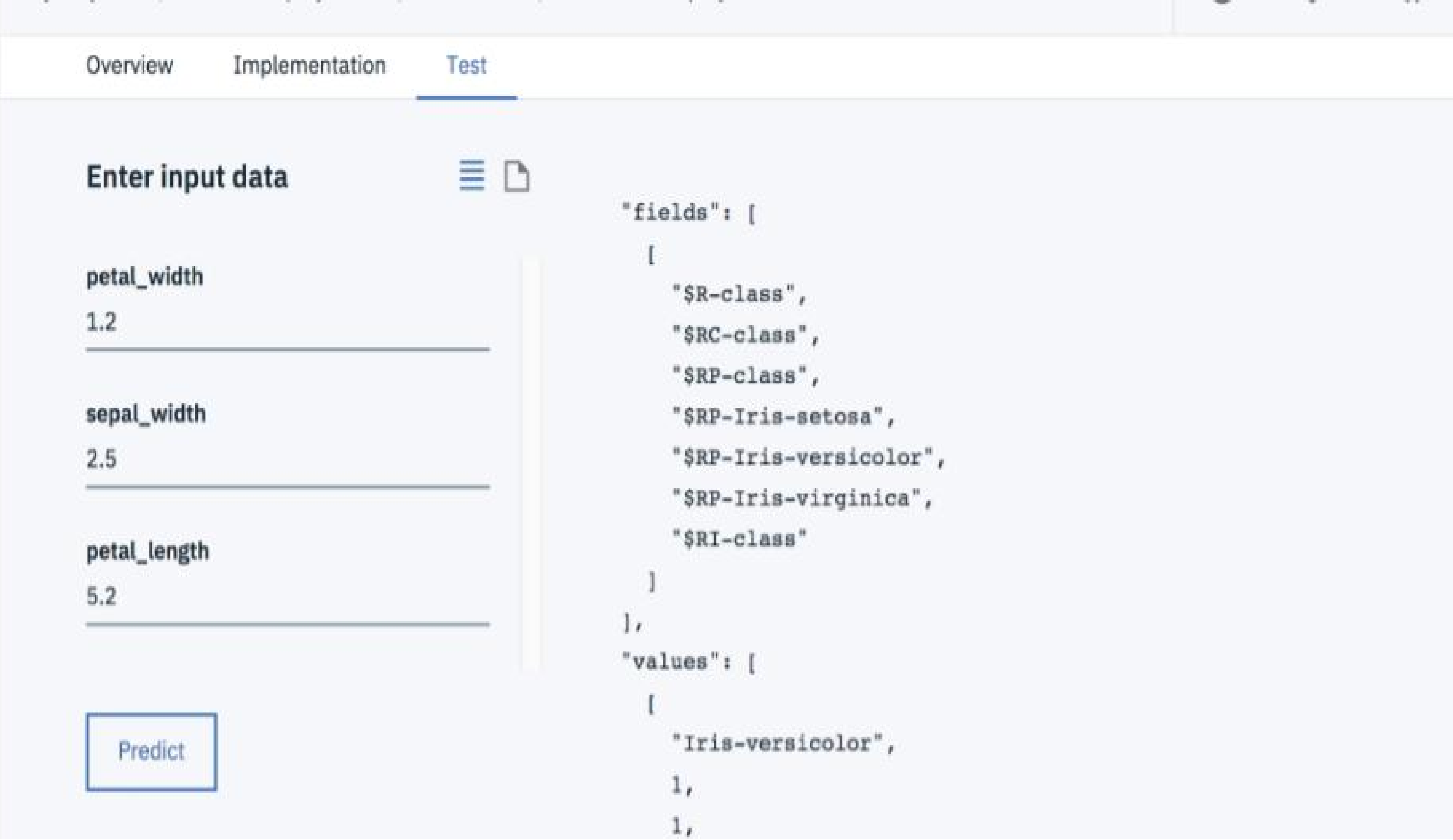
256

Deployment type

● Web service

4. Click **Test**
5. Enter data and click **Predict** to make a prediction.

- **To deploy the model**
- On the model page, click **Add Deployment**.
- Enter a deployment name, and optional descriptions.
- Select **Web service** deployment type and it exposes REST APIs for the deployed model. When the deployment phase is successful, we can open the deployment page to test the scoring function.



Enter input data

petal_width
1.2

sepal_width
2.5

petal_length
5.2

```
"fields": [
  [
    "$R-class",
    "$RC-class",
    "$RP-class",
    "$RP-Iris-setosa",
    "$RP-Iris-versicolor",
    "$RP-Iris-virginica",
    "$RI-class"
  ]
],
"values": [
  [
    "Iris-versicolor",
    1,
    1,
```

- Since the web service deployment exposes standard REST APIs, we can connect to any client.

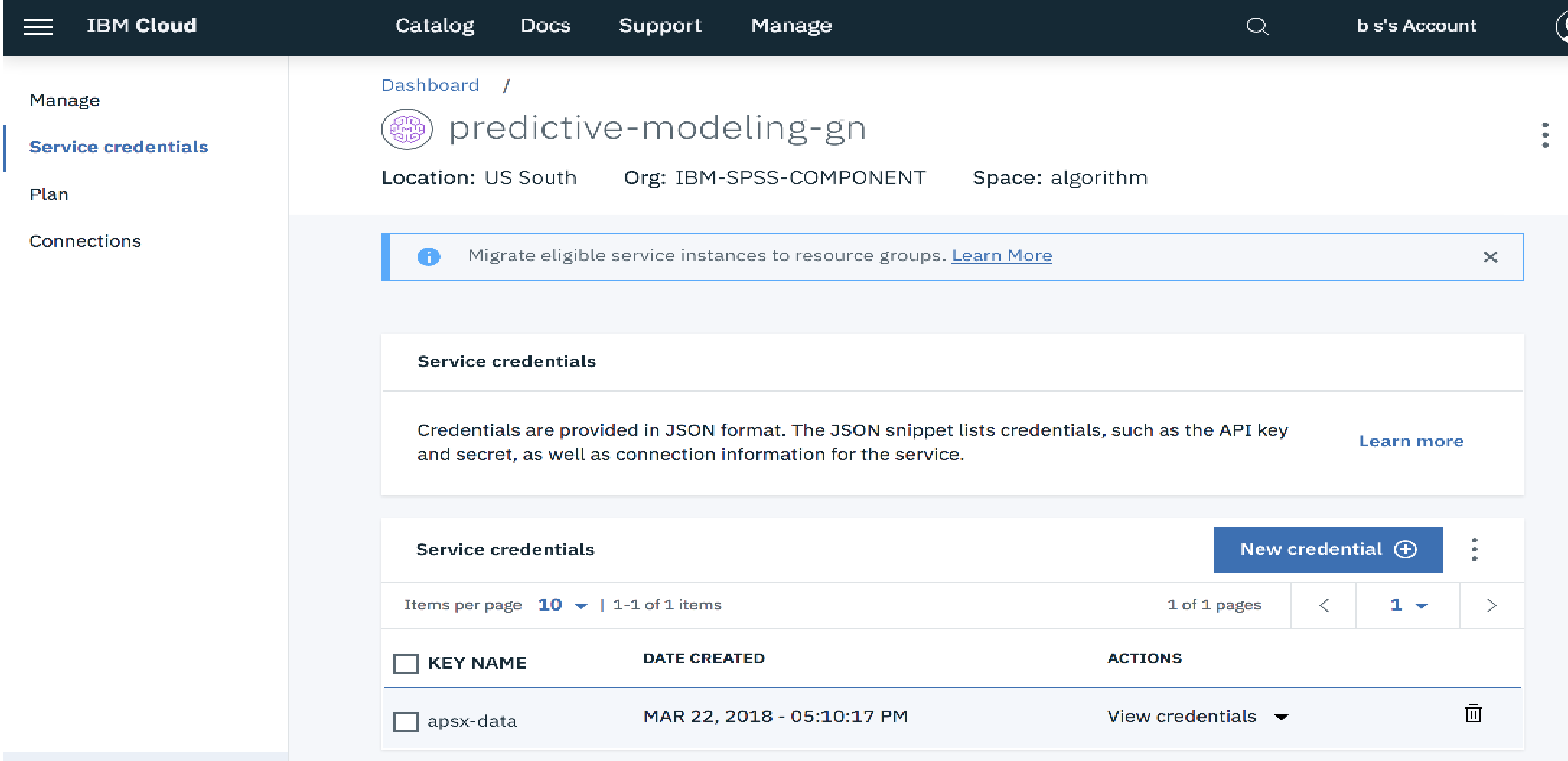
  6. Click **Implementation** to access information, such as endpoint and authorization of service. The Implementation page displays all of the information you can use and several code snippets of different languages for your reference.

  Now let’s look at an example of using curl. Based on the descriptions, you need to retrieve three variables from the service credentials associated with your IBM Cloud Watson Machine Learning Service instance.

  7. Click **Overview** of deployment to get the name of the associated machine learning service, and then go to its home page.

- Click **New credential** if there are no existing credentials, click **View credentials** to get the url, username, and password variables.

9. Start a shell to export the url, user name, and password variables.
10. Verify the variables to get a valid token and initiate a request with a record payload to make the prediction.



predictive-modeling-gn

Location: US South Org: IBM-SPSS-COMPONENT Space: algorithm

Migrate eligible service instances to resource groups. Learn More

Service credentials

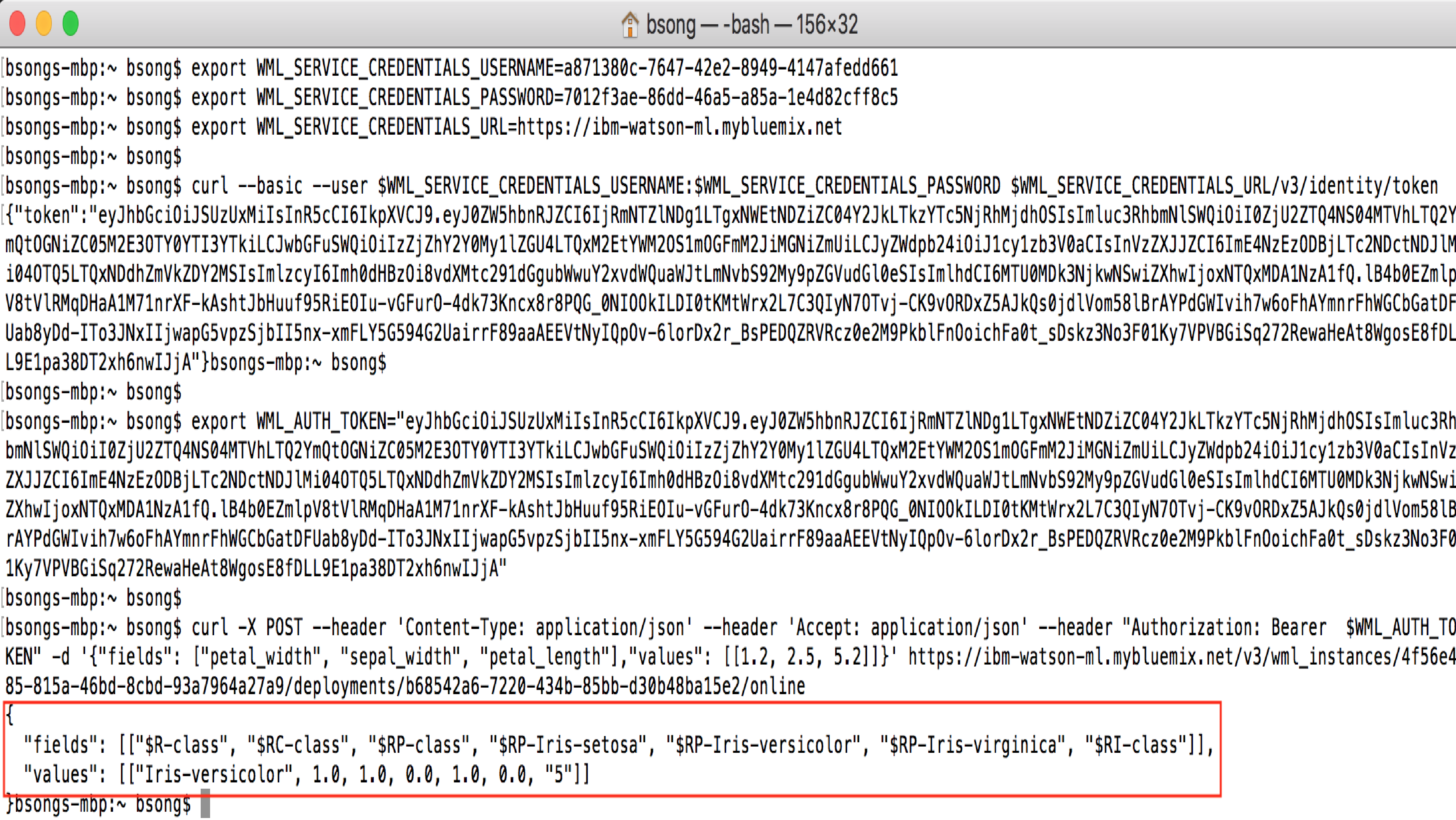
Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

Service credentials

Items per page 10 | 1-1 of 1 items 1 of 1 pages < 1 >

□ KEY NAME	DATE CREATED	ACTIONS
□ apsx-data	MAR 22, 2018 - 05:10:17 PM	View credentials

- . Start a shell to export the url, user name, and password variables.
  10. Verify the variables to get a valid token and initiate a request with a record payload to make the prediction.



```
bsongs-mbp:~ bsong$ export WML_SERVICE_CREDENTIALS_USERNAME=a871380c-7647-42e2-8949-4147afedd661
bsongs-mbp:~ bsong$ export WML_SERVICE_CREDENTIALS_PASSWORD=7012f3ae-86dd-46a5-a85a-1e4d82cff8c5
bsongs-mbp:~ bsong$ export WML_SERVICE_CREDENTIALS_URL=https://ibm-watson-ml.mybluemix.net
bsongs-mbp:~ bsong$
bsongs-mbp:~ bsong$ curl --basic --user $WML_SERVICE_CREDENTIALS_USERNAME:$WML_SERVICE_CREDENTIALS_PASSWORD $WML_SERVICE_CREDENTIALS_URL/v3/identity/token
{"token":"eyJhbGciOiJSUzUxMiIsInR5cCI6IkpXVCJ9.eyJ0ZW5hbnRJZCI6IjRmNTZlNDg1LTgxNWEtNDZiZC04Y2JkLTkzYTc5NjRhMjdhOSIsImluc3RhbmNlSWQiOiI0ZjU2ZTQ4NS04MTVhLTQ2Y
mQtOGNiZC05M2E3OTY0YTI3YTkiLCJwbGFuSWQiOiIzZjZhY2Y0My1lZGU4LTQxM2EtYWM2OS1mOGFmM2JiMGNiZmUiLCJyZWdpb24iOiJ1cy1zb3V0aCIsInVzZXJJZCI6ImE4NzEzODBjLTc2NDctNDJlM
i04OTQ5LTQxNDdhZmVkZDY2MSIsImlzcyI6Imh0dHBzOi8vdXMtc291dGgubWwuY2xvdWQuaWJtLmNvbS92My9pZGVudGl0eSIsImlhdCI6MTU0MDk3NjkwNSwiZXhwIjoxNTQxMDA1NzA1fQ.lB4b0EZmlp
V8tVlRMqDHaA1M71nrXF-kAshtJbHuuf95RiEOIu-vGFurO-4dk73Kncx8r8PQG_0NIOOkILDI0tKMtWrx2L7C3QIyN7OTvj-CK9vORDxZ5AJkQs0jdlVom58lBrAYPdGWIvih7w6oFhAYmnrFhWGCbGatDF
Uab8yDd-ITo3JNxIIjwapG5vpzSjbII5nx-xmFLY5G594G2UairrF89aaAEEVtNyIQpOv-6lorDx2r_BsPEDQZRVRcz0e2M9PkblFnOoichFa0t_sDskz3No3F01Ky7VPVBGiSq272RewaHeAt8WgosE8fDL
L9E1pa38DT2xh6nwIJjA"}bsongs-mbp:~ bsong$
bsongs-mbp:~ bsong$
bsongs-mbp:~ bsong$ export WML_AUTH_TOKEN="eyJhbGciOiJSUzUxMiIsInR5cCI6IkpXVCJ9.eyJ0ZW5hbnRJZCI6IjRmNTZlNDg1LTgxNWEtNDZiZC04Y2JkLTkzYTc5NjRhMjdhOSIsImluc3Rh
bmNlSWQiOiI0ZjU2ZTQ4NS04MTVhLTQ2YmQtOGNiZC05M2E3OTY0YTI3YTkiLCJwbGFuSWQiOiIzZjZhY2Y0My1lZGU4LTQxM2EtYWM2OS1mOGFmM2JiMGNiZmUiLCJyZWdpb24iOiJ1cy1zb3V0aCIsInVz
ZXJJZCI6ImE4NzEzODBjLTc2NDctNDJlMi04OTQ5LTQxNDdhZmVkZDY2MSIsImlzcyI6Imh0dHBzOi8vdXMtc291dGgubWwuY2xvdWQuaWJtLmNvbS92My9pZGVudGl0eSIsImlhdCI6MTU0MDk3NjkwNSwi
ZXhwIjoxNTQxMDA1NzA1fQ.lB4b0EZmlpV8tVlRMqDHaA1M71nrXF-kAshtJbHuuf95RiEOIu-vGFurO-4dk73Kncx8r8PQG_0NIOOkILDI0tKMtWrx2L7C3QIyN7OTvj-CK9vORDxZ5AJkQs0jdlVom58lB
rAYPdGWIvih7w6oFhAYmnrFhWGCbGatDFUab8yDd-ITo3JNxIIjwapG5vpzSjbII5nx-xmFLY5G594G2UairrF89aaAEEVtNyIQpOv-6lorDx2r_BsPEDQZRVRcz0e2M9PkblFnOoichFa0t_sDskz3No3F0
1Ky7VPVBGiSq272RewaHeAt8WgosE8fDLL9E1pa38DT2xh6nwIJjA"
bsongs-mbp:~ bsong$
bsongs-mbp:~ bsong$ curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization: Bearer  $WML_AUTH_TO
KEN" -d '{"fields": ["petal_width", "sepal_width", "petal_length"],"values": [[1.2, 2.5, 5.2]]}' https://ibm-watson-ml.mybluemix.net/v3/wml_instances/4f56e4
85-815a-46bd-8cbd-93a7964a27a9/deployments/b68542a6-7220-434b-85bb-d30b48ba15e2/online
{
  "fields": [["$R-class", "$RC-class", "$RP-class", "$RP-Iris-setosa", "$RP-Iris-versicolor", "$RP-Iris-virginica", "$RI-class"]],
  "values": [["Iris-versicolor", 1.0, 1.0, 0.0, 1.0, 0.0, "5"]]
}bsongs-mbp:~ bsong$
```

As you can see, the results are same as ones produced by the test function. At this point, we hope you see how easy it is to deploy PMML Model into a REST API with Watson Studio.

PHASE -4 (PROJECT)

Name: P HARIHARAN

Year:3rd year

Dept:CSE

College: University College Engineering Thirukkuvalai