

# IBM Watson Knowledge Studio

## Demo Guide:

## Building a Machine-Learning Annotator

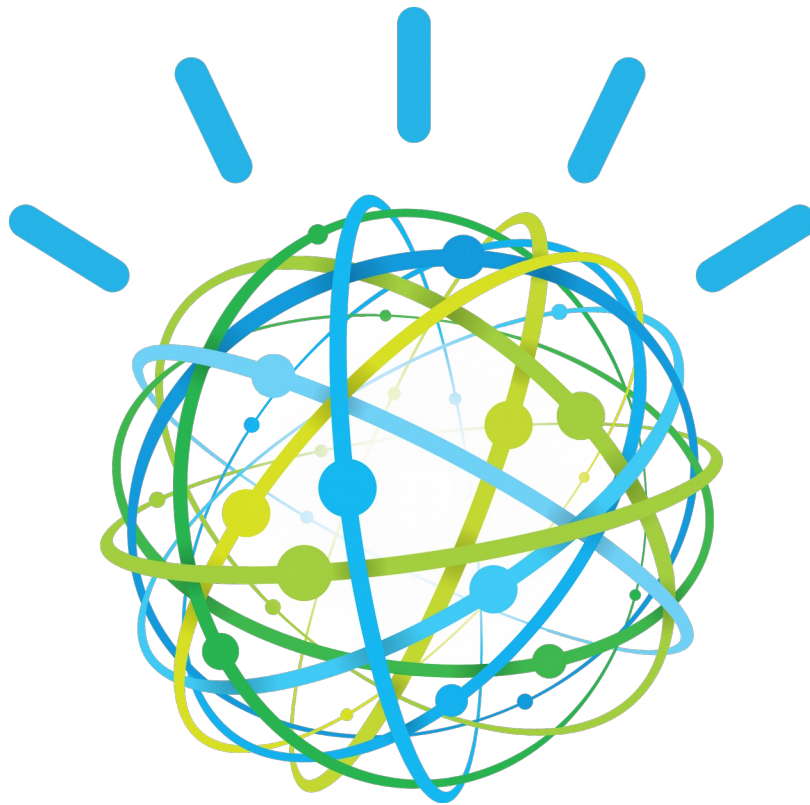
IBM Global Business Partners

Duration: 30 minutes

Updated: Sep 16, 2019

Klaus-Peter Schlotter

[kps@de.ibm.com](mailto:kps@de.ibm.com)



# IBM

Version 3.0

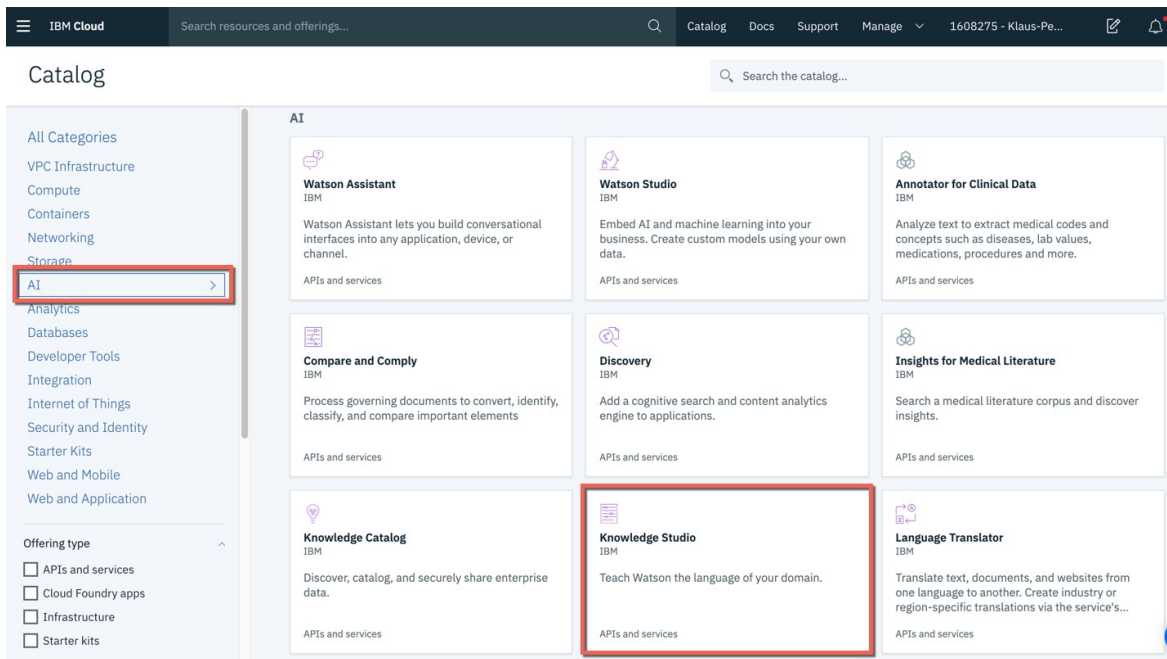
# IBM Watson Services Workshop

The document describes the creation of a Machine Learning-annotator with IBM Watson Knowledge Studio (WKS) available as a service on the IBM Cloud Platform.

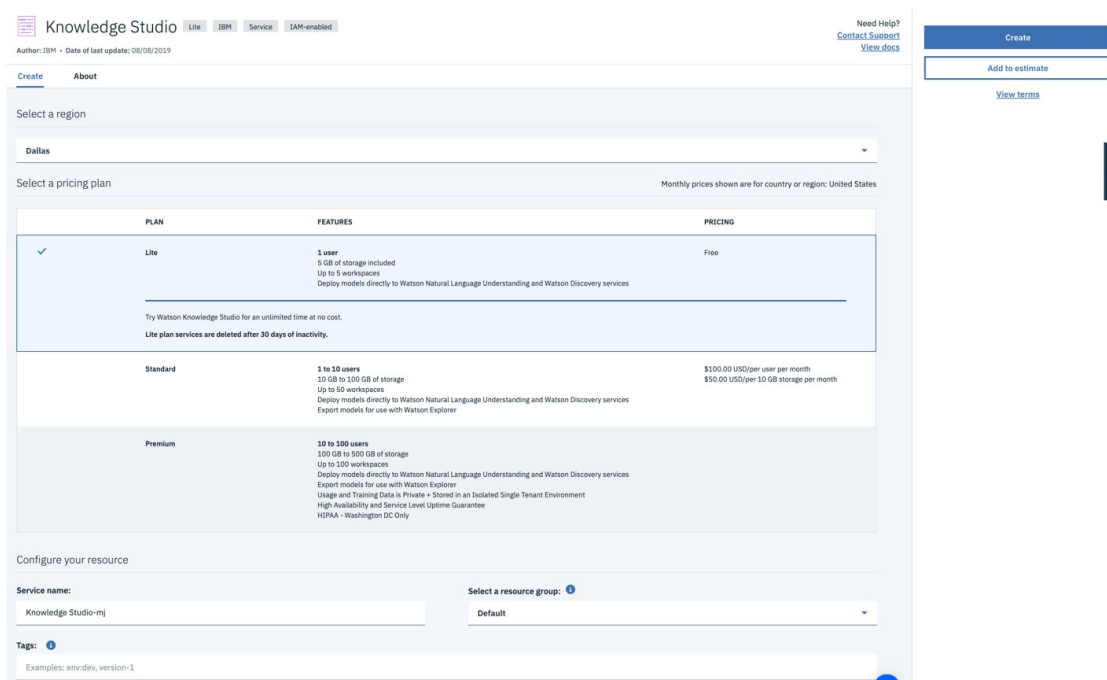
Version 1.x of this document describes this process with WKS available in the IBM Marketplace.

In the demo a new project will be created and configured until Step 28, then the model from an already existing project will be used due to the time needed for training and evaluating the new model.

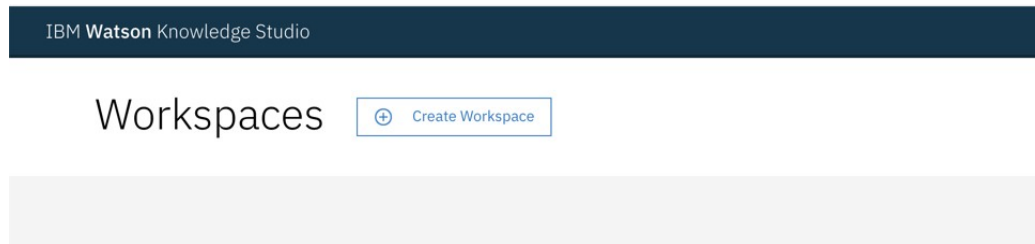
**Step 1** In your IBM Cloud Platform console in the AI category, **click** on the Knowledge Studio service



**Step 2** Enter a unique *name* (you can accept the default) and **select** the *Lite* plan. **Click Create**.



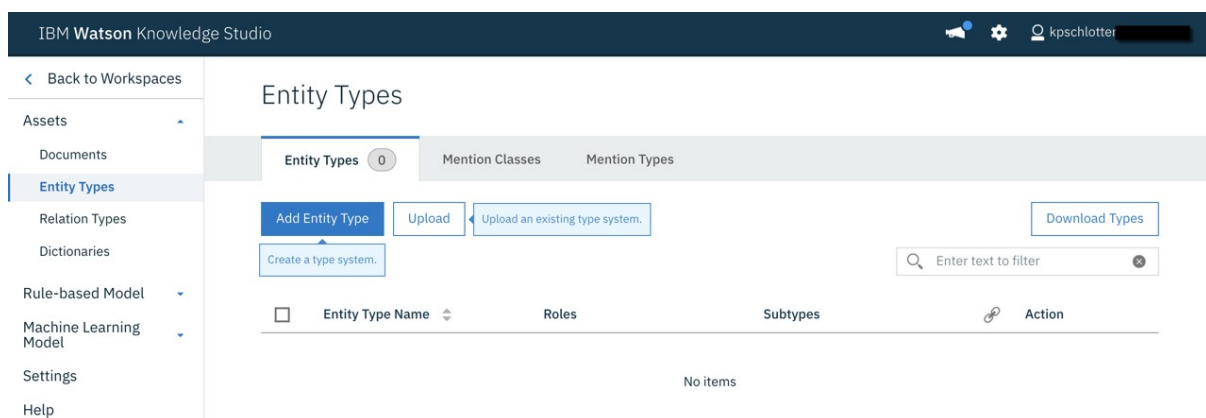
**Step 3** In the service instance click the **Launch Watson Knowledge Studio** button to open Knowledge Studio.



**Step 4** Click the **Create Workspace** button. **Name** the workspace and **select English** as the language. **Click Create**.

The 'Create Workspace' form is displayed. It includes a 'Workspace name' field with the text 'My\_Pizzeria', a 'Language of documents' dropdown menu set to 'English', and a 'Workspace description (optional)' text area containing 'Comments on Pizzas'. At the bottom, there are 'Cancel' and 'Create' buttons.

**Step 5** The workspace opens with *Entity Types* page displayed.



**Step 6** On the *Assets* → *Entity Types* click [Add Entity Type](#). As *Entry Type Name* enter PIZZA\_TYPE and click [Save](#).

**Step 7** Also add an Entity Type NEGATIVE\_FEEDBACK


**Step 8** On *Assets* → *Relation Types* click [Add Relation Type](#).  
name it complaintWith  
**First Entity Type** PIZZA\_TYPE  
**Second Entity Type** NEGATIVE\_FEEDBACK  
Click [Save](#).

Relation Types 1			
<a href="#">Add Relation Type</a>			
<input type="text" value="Enter text to filter"/>			
Relation Type	First Entity Type / Role	Second Entity Type / Role	Action
complaintWith	PIZZA_TYPE	NEGATIVE_FEEDBACK	<a href="#">Edit</a> <a href="#">Delete</a>

**Step 9** On *Assets* → *Dictionaries* click [Create Dictionary](#),  
**Name:** pizza\_types\_dict  
click [Save](#)

**Step 10** Upload the dictionary entries for the *Pizza Types*  
(Dictionary-items-PizzaTypes.csv)

### Upload Dictionary Entries



Drag a file here, or click to browse for the file that you want to upload.  
You can upload any CSV file that contains dictionary terms.  
Maximum file size: 1MB  
[Dictionary-items-PizzaTypes.csv](#)

[Close](#) [Upload](#)

**Step 11** Map this dictionary to the PIZZA\_TYPE.

[Back to Workspaces](#)

- Assets
  - Documents
  - Entity Types
  - Relation Types
  - Dictionaries**
  - Rule-based Model
  - Machine Learning Model
  - Settings
  - Help

### Dictionaries

[Create Dictionary](#)

pizza\_types\_dict 5

**pizza\_types\_dict**  
Language: English | 5 entries

Entity type:

PIZZA\_TYPE

Rule class:

None

[Add Entry](#) [Upload](#) [Download](#)

<input type="checkbox"/>	Lemma	Surface Forms	Part of Speech	Action
<input type="checkbox"/>	margherita	margherita, Margarita	Noun	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	bacon pizzaburger	bacon pizzaburger, Bacon PizzaBurger	Noun	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	mediterranean	mediterranean	Noun	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	white pizza	white pizza	Noun	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	las pizzas del fuego	las pizzas del fuego	Noun	<a href="#">Edit</a> <a href="#">Delete</a>

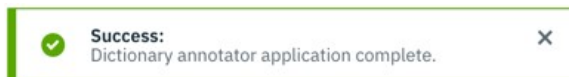
**Step 12** Click **Add Entry**, **enter** `four seasons as Surface Form` and **select** `Noun` as `Part of Speech`. Click **Save**.

**Note:** For Brazilian Portuguese, English, French, German, Italian, and Spanish, Knowledge Studio does not currently provide an option to specify case-insensitive dictionary-matching, but dictionary entries match text that has a higher case. For example, `vehicle` in the dictionary matches `vehicle`, `Vehicle` or `VEHICLE` in text, while `Sat` in the dictionary matches `Sat` or `SAT` in text, but not `sat`.

**Step 13** On `Assets → Documents` click **Upload Document Sets** to import the training documents (`pizza_reviews.txt`).

**Step 14** On `Machine Learning Model → Pre-annotation → Dictionaries` you see the dictionary mapping done in a step above. Click **Apply This Pre-annotator**, **select** the document set imported previously and click **Run**.

The following message should be displayed and the pizza types are annotated in our annotation set.



**Step 15** On `Machine Learning Model → Annotations → Annotations` click **Add Task**.

**Note:** You could do this directly on the Ground Truth tab when there is only one person to work on this annotation. But for multiple annotators an Annotation Task is needed to split the work.

**Step 16** On `Create Annotation Tasks` click **Create Annotation Sets** to create a copy of our training documents for annotation.

**Base set:** `pizza_reviews_text_set`

**Overlap:** `100%`

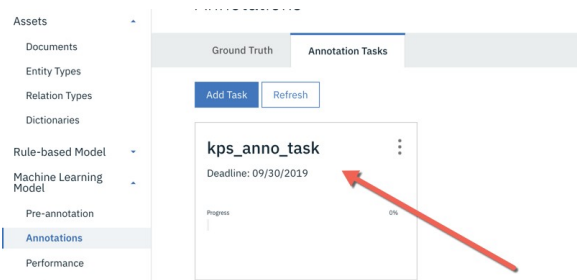
**Annotator:** `<your id>`

**Set name:** `..._pizza_anno_set`

**Click:** **Generate**

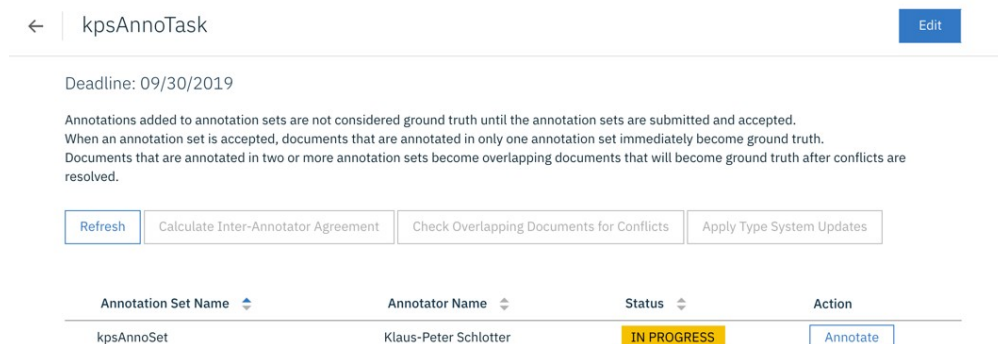
**Step 17** Back on the `Create Annotation Task` page **enter** a `Task name` `..._anno_tasks` an optional `Deadline` and click **Save**.

**Step 18** On *Machine Learning Models* → *Annotations* → *Annotation Tasks*, click on the task created above.

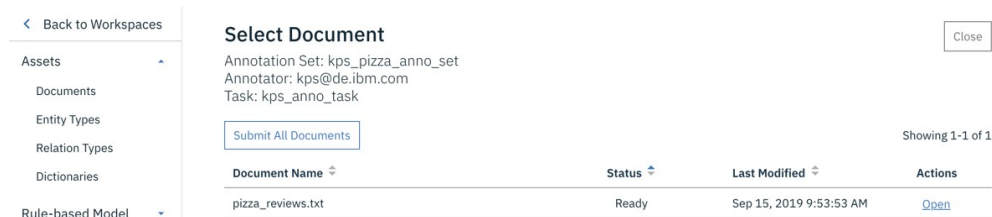


**Step 19** The annotation set is in status In Progress because of the pre-annotation we did before. Now we want to manually annotate the other items.

Click [Annotate](#).



**Step 20** [Open](#) the *pizza\_reviews.txt* document.




Here you see that the pre-annotation with the *pizza\_types\_dic* dictionary already marked as **PIZZA\_TYPES**.

## pizza\_reviews.txt

- 1 I ordered a large **mediterranean** pizza from your Mopac location and the pizza was cold.
- 2 Loved the Far West but the **White Pizza** was less desirable.
- 3 I was in your Pflugerville store and had the Brazos with thin crust, which was awesome.
- 4 I thought the Farmer's Market was supposed to have mushrooms on it, but mine did not.
- 5 The Meaty Madness I had on Sunday was burnt, so I want my money back.
- 6 The **Mediterranean** I had the other day was charred on the edges.
- 7 The Brazos really needs thicker crust.


**Step 21** Mark the remaining PIZZA\_TYPES and the NEGATIVE\_FEEDBACK entries.

The screenshot shows the 'Mention' tab in the IBM Watson Knowledge Studio interface. The document 'pizza\_reviews.txt' is open, displaying seven sentences. Various terms are highlighted in red, indicating they are marked as entities. The right sidebar shows the 'Entity' list with 'NEGATIVE\_FEEDBACK' and 'PIZZA\_TYPE'.

**Step 22** Click the  tab on the document panel. Click on the PIZZA\_TYPE and on the associated NEGATIVE\_FEEDBACK, and from the *Relations Type* side bar select *complaintWith*.

The screenshot shows the 'Relation' tab in the IBM Watson Knowledge Studio interface. The document 'pizza\_reviews.txt' is open, displaying the same seven sentences. The entities 'PIZZA\_TYPE' and 'NEGATIVE\_FEEDBACK' are now connected by 'complaintWith' relations, which are visualized as lines between the entities. The right sidebar shows the 'Relation Type' list with 'complaintWith' selected.

Click  .

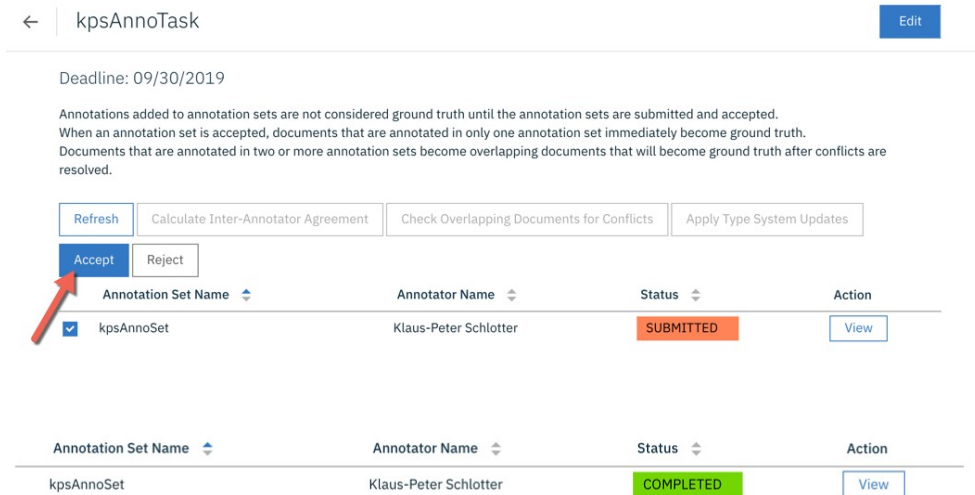
**Step 23** Click  [Open document list](#) to go back to the document list. You see the above document with status in progress.




**Step 24** On the *Select Document* panel click . Now all documents are in the **Completed** state.

**Step 25** Click  to close the *Select Document* panel. Then click  ← *Back to Task*



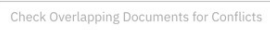

**Step 26** On the Annotation Task Accept the submitted annotation set.






← kpsAnnoTask 

Deadline: 09/30/2019


Annotations added to annotation sets are not considered ground truth until the annotation sets are submitted and accepted. When an annotation set is accepted, documents that are annotated in only one annotation set immediately become ground truth. Documents that are annotated in two or more annotation sets become overlapping documents that will become ground truth after conflicts are resolved.


   


Annotation Set Name	Annotator Name	Status	Action
<input checked="" type="checkbox"/> kpsAnnoSet	Klaus-Peter Schlotter	SUBMITTED	

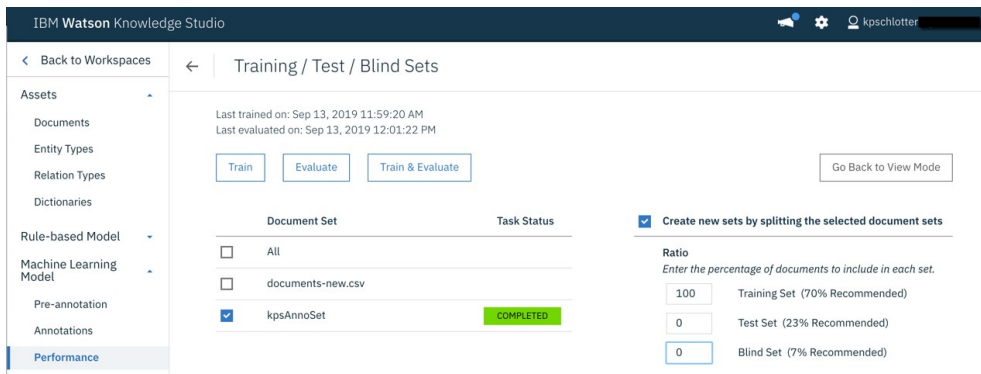
  


Annotation Set Name	Annotator Name	Status	Action
kpsAnnoSet	Klaus-Peter Schlotter	COMPLETED	

**Step 27** On Machine Learning Model → Performance click  to create the model from our annotations.

**Step 28** On the *Training/Test/Blind Sets* page select the document set **COMPLETED** 100% *Training Set*, 0% *Test Set* and 0% *Blind Set*. (we only have 1 document)





Then click .



IBM Watson Knowledge Studio  kpschlotter

← Back to Workspaces ← Training / Test / Blind Sets

Last trained on: Sep 13, 2019 11:59:20 AM  
Last evaluated on: Sep 13, 2019 12:01:22 PM

Document Set	Task Status
<input type="checkbox"/> All	
<input type="checkbox"/> documents-new.csv	
<input checked="" type="checkbox"/> kpsAnnoSet	COMPLETED

☒ Create new sets by splitting the selected document sets

Ratio  
Enter the percentage of documents to include in each set.

Ratio	Set
100	Training Set (70% Recommended)
0	Test Set (23% Recommended)
0	Blind Set (7% Recommended)

This process takes about 15 minutes.



Success: Sep 13, 2019 12:01:22 PM  
Machine learning model evaluation complete.



**Step 29** On *Machine Learning Model* → *Versions* click **Create Version** to create a version of your model for deployment. Enter a Description and click **OK**.

Version	Base	Creation Date	Entity Scores	Relation Scores	Description	Action
1.1	Current Version		N/A	N/A		<a href="#">Create Version</a>
1.0		09/14/2019	N/A	N/A	WKS_190913 demo model	<a href="#">Promote</a> <a href="#">Delete</a> <a href="#">Deploy</a>

**Step 30** Click [Deploy](#) to deploy your model. Select *Natural Language Understanding* service created earlier and click *Next*.

**Step 31** Select your service and click *Deploy*.

The deployment process starts. Click **OK**.

## Deployment Started.

### Deploying to Natural Language Understanding

It might take a few minutes for publishing and deployment to complete, and for this model to be available to your applications. You can view your deployed models, withdraw a model from deployment, or deploy a newer version.

**Model ID:** b0875e3d-391d-4f4d-be1b-72dc4d7f248a

You can [view documentation](#) to learn how to implement the deployed model into your application.

OK

You then have a deployment history.

### Version History and Deployment

Version	Base	Creation Date	Entity Scores	Relation Scores	Description	Action
1.1		Current Version	N/A	N/A		<a href="#">Create Version</a>
1.0		09/14/2019	N/A	N/A	WKS_190913 demo model	<a href="#">Promote</a> <a href="#">Delete</a> <a href="#">Deploy</a>
▼ Deployed Models (1)						
			Model ID: b0875e3d-391d-4f4d-be1b-72dc4d7f248a		Service ID: d373e60d-dcd1-4efd-b305-2eac19e1c79c	<a href="#">Undeploy</a> <a href="#">Status</a>

## Step 32 Testing the model with Postman

- List your deployed models. You need Basic Auth (username and password from your Natural Language Understanding Service from your Bluemix account and the following URL:

<https://gateway.watsonplatform.net/natural-language-understanding/api/v1/models?version=2019-07-12>

The screenshot shows the Postman interface. The request is a GET to `https://gateway.watsonplatform.net/natural-language-understanding/api/v1/models?version=2019-07-12`. The Authorization tab is selected, showing Basic Auth with Username `apikey` and Password `*****`. The response is a 200 OK status with a JSON body:

```

1 {
2   "models": [
3     {
4       "workspace_id": "c9f17760-d600-11e9-815d-b173999649ae",
5       "version_description": "WKS_190913 demo model",
6       "version": "1.0",
7       "status": "available",
8       "name": null,
9       "model_id": "b0875e3d-391d-4f4d-be1b-72dc4d7f248a",
10      "language": "en",
11      "description": null,
12      "created": "2019-09-14T07:03:45Z"
13    }
14  ]
15 }

```

- b) Test the model with the Basic Auth information from previous step and in the *Headers* section add

key: Content-type                      value: application/json

In the *URL* **change** models **to** analyze

In the *Body* section enter the text you want to analyze and specify the service feature(s) you want to apply.

```
{ "text": "I ordered a Mediterranean from your mopac location on July 25
and the cheese was stuck to the top of the box.",
  "features": { "relations": { "model": "<your model ID goes here>" } } }
```

- c) With the following result:

The screenshot displays the IBM Watson NLU Test Model interface. The top section shows the API endpoint: `https://gateway.watsonplatform.net/natural-language-understanding/api/v1/analyze?version=2019-07-12`. The **Body** tab is selected, showing the request JSON:

```
{ "text": "NCR, which counts IBM founder Thomas Watson as one of its early employees, said its products and services account for more than $400 billion in annual commerce and 23 billion consumer serf-service transactions.",
  "features": {
    "relations": {
      "model": "825a30e9-c3ac-45ec-8bf5-311c981db164"
    },
    "entities": {
      "model": "825a30e9-c3ac-45ec-8bf5-311c981db164"
    }
  }
}
```

The bottom section shows the response JSON in the **Test Results** tab:

```
{
  "usage": {
    "text_units": 1,
    "text_characters": 210,
    "features": 2
  },
  "relations": [
    {
      "type": "founderOf",
      "sentence": "NCR, which counts IBM founder Thomas Watson as one of its early employees, said its products and services account for more than $400 billion in annual commerce and 23 billion consumer serf-service transactions.",
      "score": 0.956072,
      "arguments": [
        {
          "text": "Thomas Watson",
          "location": [
            30,
            43
          ],
          "entities": [
            {
              "type": "PERSON",
              "text": "Thomas Watson",
              "disambiguation": {
                "subtype": [
                  "NONE"
                ]
              }
            }
          ]
        }
      ]
    }
  ],
  {
    "text": "IBM",
    "location": [
      18,
      21
    ]
  }
}
```