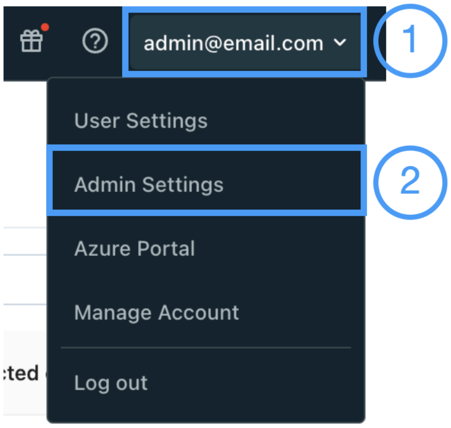
Workspace Estimator

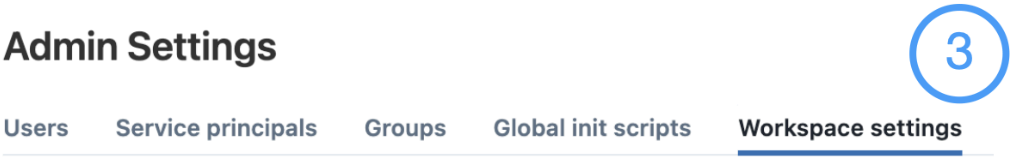
Step By Step Guide

1. Verify permissions

1. Log in to your databricks workspace.
2. Verify that you have the required permissions by navigating to the upper right corner of the screen. And click in the drop down. Select **Admin Settings**.



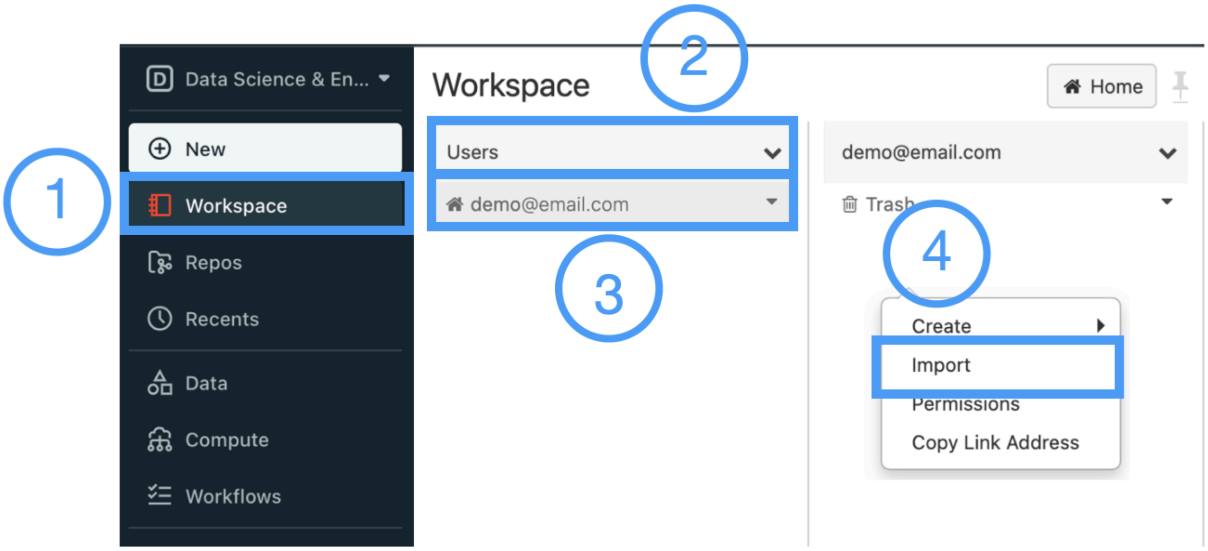
3. Click in the **Workspace settings** tab.



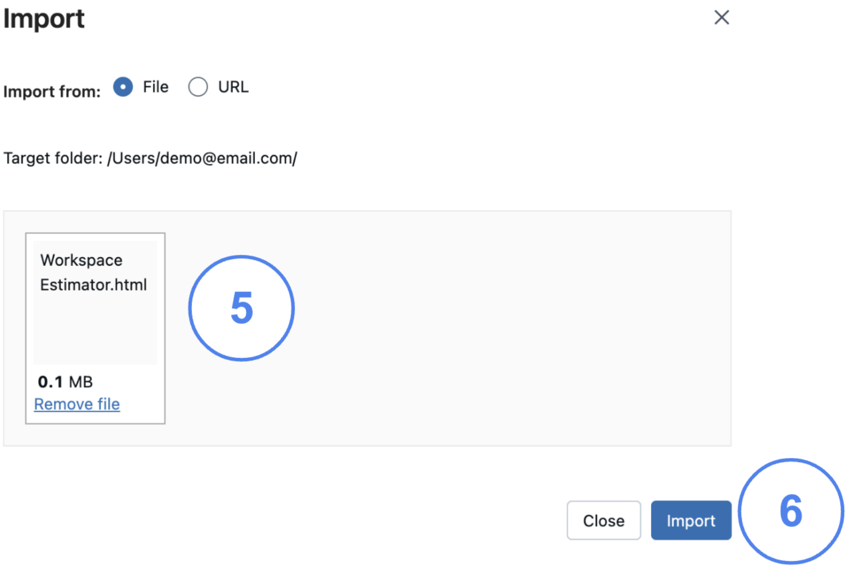
4. Verify permissions in following sections
   a. **Access Control**
      i. **Workspace Access Control:** Enabled
      ii. **Cluster, Pool, and Jobs Access Control:** Enabled
      iii. **Personal Access Tokens:** Enabled
      iv. **Workspace Visibility Control:** Enabled
      v. **Cluster Visibility Control:** Enabled
      vi. **Job Visibility Control:** Enabled
   b. **Advanced**
      i. **DBFS File Browser:** Enabled

2. Importing Workspace Estimator

1. First logging in your databricks workspace.
2. Import the Notebook by Clicking in the **Workspace** menu in the left panel, then Click in **Users** Submenu, and Click in your user name home. After that Right Click in your user folder to open the context menu to import the notebook.



3. Drag and Drop or select the **Workspace Estimator.html** file, and press import.


Import

Import from: ● File ○ URL

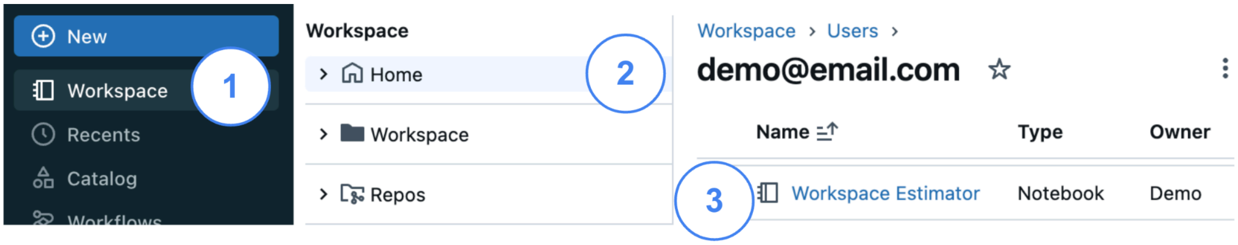
Target folder: /Users/demo@email.com/

Workspace Estimator.html

0.1 MB

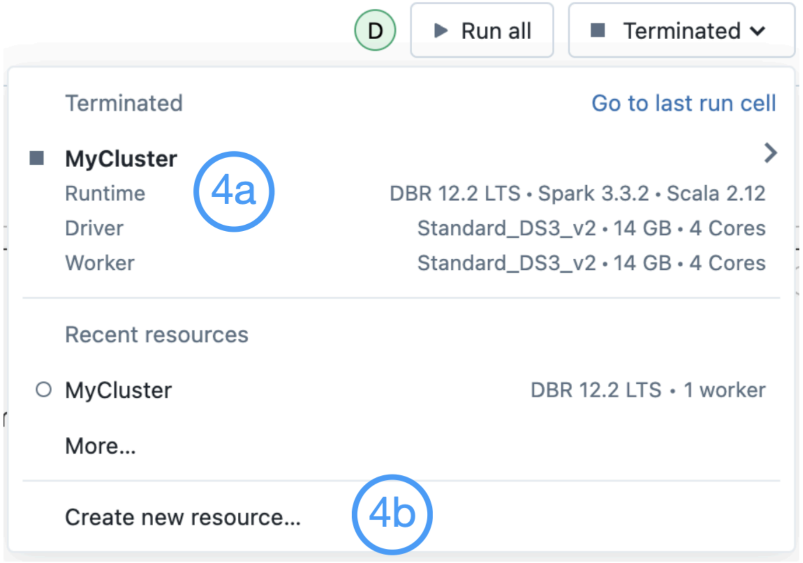
5. Executing Notebook

1. First logging in your databricks workspace.
2. Open your workspace folder, and click on **workspace estimator** Notebook.



Name	Type	Owner
Workspace Estimator	Notebook	Demo

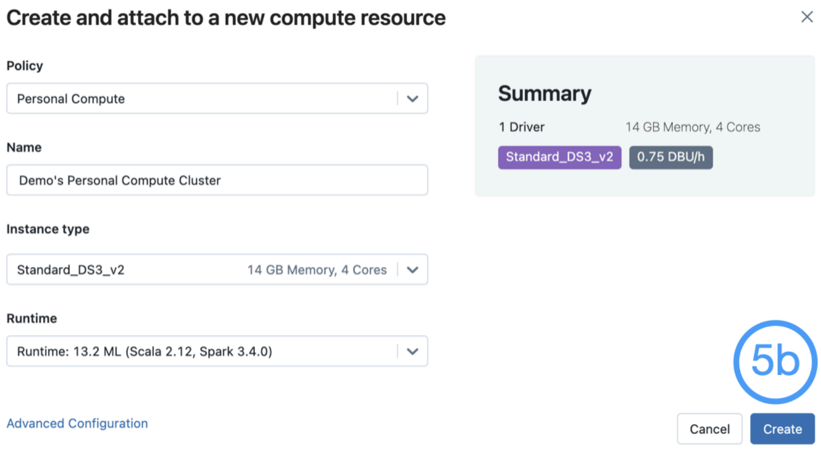
3. Attach an existing Cluster or Create a new Resource.



■ MyCluster

Runtime	DBR 12.2 LTS • Spark 3.3.2 • Scala 2.12
Driver	Standard_DS3_v2 • 14 GB • 4 Cores
Worker	Standard_DS3_v2 • 14 GB • 4 Cores

○ MyCluster DBR 12.2 LTS • 1 worker



Create and attach to a new compute resource

Policy: Personal Compute

Name: Demo's Personal Compute Cluster

Instance type: Standard_DS3_v2 14 GB Memory, 4 Cores

Runtime: Runtime: 13.2 ML (Scala 2.12, Spark 3.4.0)

Summary

1 Driver 14 GB Memory, 4 Cores

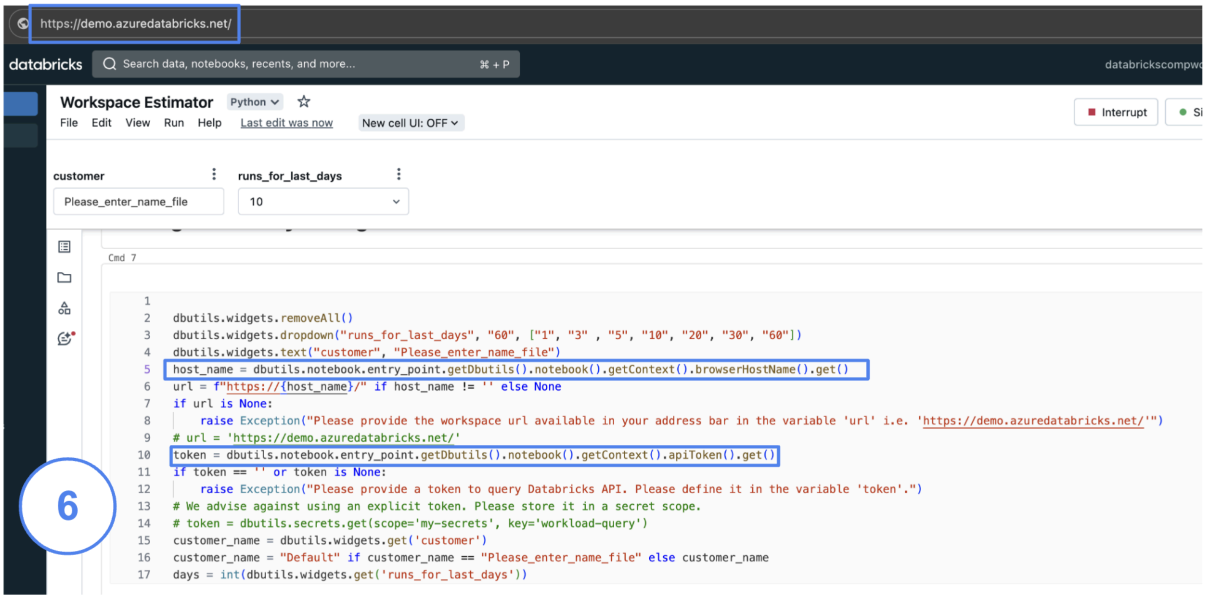
Standard_DS3_v2 0.75 DBU/h

4. After the cluster starts up, you can select **Run All.**

4.b However if you want to set the **URL for the api manuall**y you can follow this steps:

Execute cell by cell and follow instructions. At cell **Cmd 7,** we tried to get the hostname and token, however, if that fails you can adjust settings with your workspace settings.

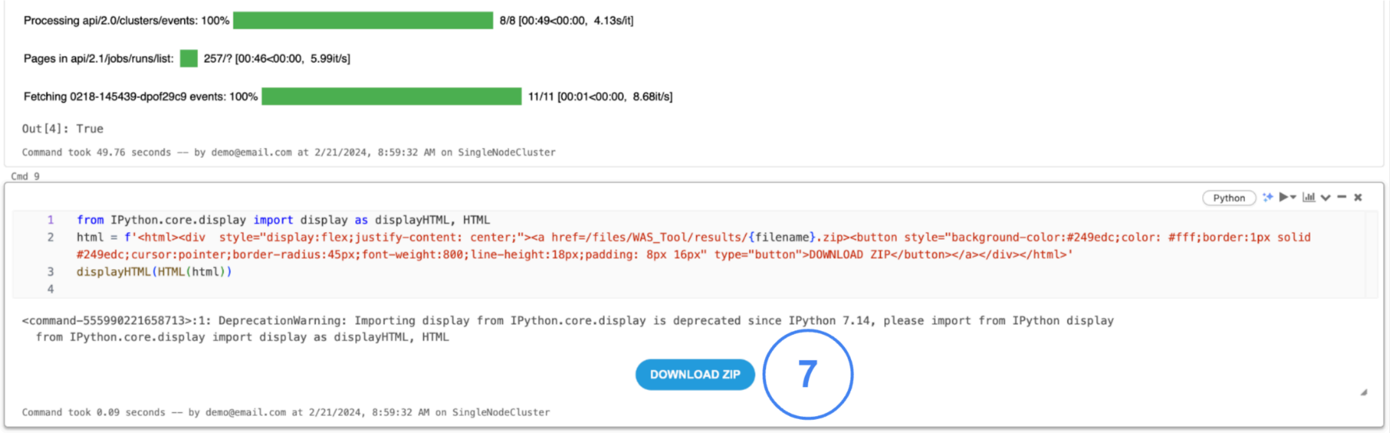
a. **url**: is the workspace address it's shown in your address bar.
b. **token**: Should match the scope and key where you earlier stored the token. See **Appendix I** for step by step instructions



Cmd 7

```
dbutils.widgets.removeAll()
dbutils.widgets.dropdown("runs_for_last_days", "60", ["1", "3" , "5", "10", "20", "30", "60"])
dbutils.widgets.text("customer", "Please_enter_name_file")
host_name = dbutils.notebook.entry_point.getDbutils().notebook().getContext().browserHostName().get()
url = f"https://{host_name}/" if host_name != '' else None
if url is None:
    raise Exception("Please provide the workspace url available in your address bar in the variable 'url' i.e. 'https://demo.azuredatabricks.net/'")
# url = 'https://demo.azuredatabricks.net/'
token = dbutils.notebook.entry_point.getDbutils().notebook().getContext().apiToken().get()
if token == '' or token is None:
    raise Exception("Please provide a token to query Databricks API. Please define it in the variable 'token'.")
# We advise against using an explicit token. Please store it in a secret scope.
# token = dbutils.secrets.get(scope='my-secrets', key='workload-query')
customer_name = dbutils.widgets.get('customer')
customer_name = "Default" if customer_name == "Please_enter_name_file" else customer_name
days = int(dbutils.widgets.get('runs_for_last_days'))
```

5. At the end of all the notebook, a button should be displayed which you can click to download the Zip file with the results.



```
Processing api/2.0/clusters/events: 100%  8/8 [00:49<00:00,  4.13s/it]

Pages in api/2.1/jobs/runs/list:  257/? [00:46<00:00,  5.99it/s]

Fetching 0218-145439-dpof29c9 events: 100%  11/11 [00:01<00:00,  8.68it/s]

Out[4]: True
Command took 49.76 seconds -- by demo@email.com at 2/21/2024, 8:59:32 AM on SingleNodeCluster
```

Cmd 9

```
from IPython.core.display import display as displayHTML, HTML
html = f'<html><div  style="display:flex;justify-content: center;"><a href=/files/WAS_Tool/results/{filename}.zip><button style="background-color:#249edc;color: #fff;border:1px solid #249edc;cursor:pointer;border-radius:45px;font-weight:800;line-height:18px;padding: 8px 16px" type="button">DOWNLOAD ZIP</button></a></div></html>'
displayHTML(HTML(html))
```

```
<command-555990221658713>:1: DeprecationWarning: Importing display from IPython.core.display is deprecated since IPython 7.14, please import from IPython display
  from IPython.core.display import display as displayHTML, HTML
```

```
Command took 0.09 seconds -- by demo@email.com at 2/21/2024, 8:59:32 AM on SingleNodeCluster
```

APPENDIX I

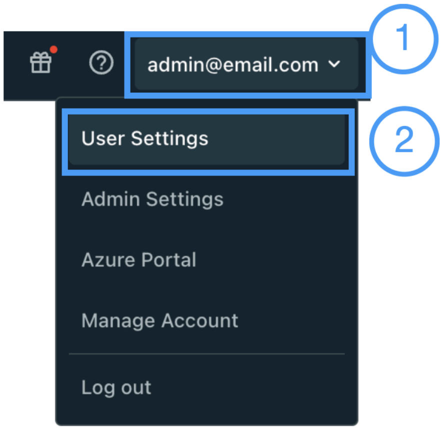
How to setup your Personal Access Token

1. Install Databricks CLI

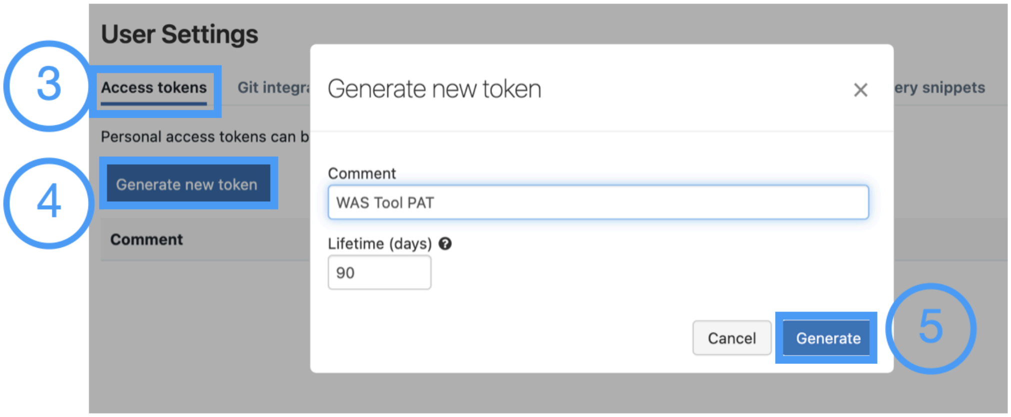
Please follow this instructions

2. Setting up a Personal Access Token (PAT)

1. Create a PAT by navigating to the upper right corner of the screen. And click in the drop down and select **User Settings.**



2. Generate a new Token.In the **Access token** Tab, click on **Generate new token**, add comment for token and desired lifetime, and finally press **Generate**.



Comment
WAS Tool PAT

Lifetime (days)
90

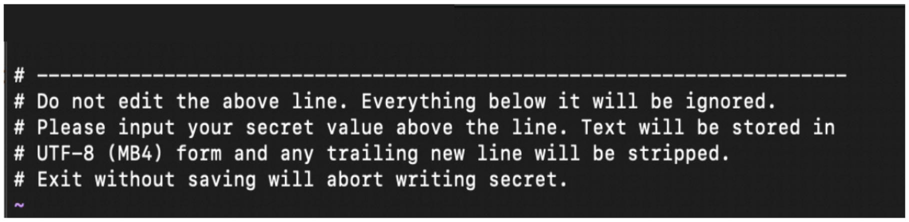
3. Open terminal which already has databricks CLI configure and create a scope with your desired name (i.e my-secrets, *if you have conflicts you can use another name or skip this step*) by executing the following command.

   *databricks secrets create-scope --scope **my-secrets***

4. Create a new key called **workload-query** in your scope to store the token by executing the following command.

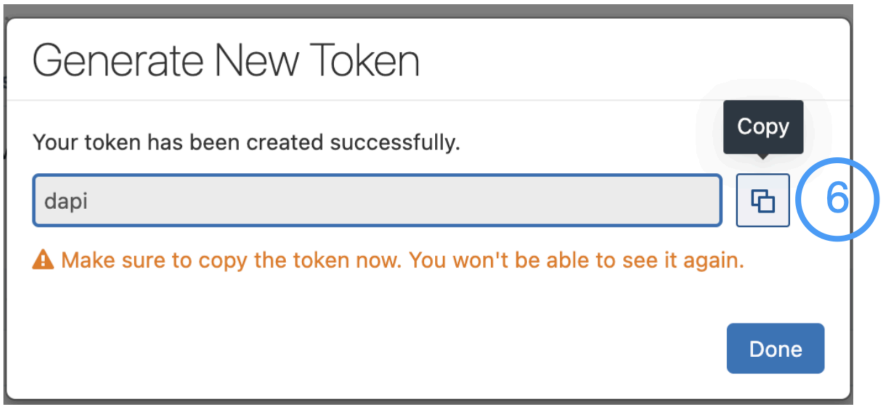
   *databricks secrets write --scope **my-secrets** --key **workload-query***

5. Previous command will Open a file like the following



```
# ----------------------------------------------------------------------
# Do not edit the above line. Everything below it will be ignored.
# Please input your secret value above the line. Text will be stored in
# UTF-8 (MB4) form and any trailing new line will be stripped.
# Exit without saving will abort writing secret.
~
```

6. Copy the Token from Step 2 in that file, and save it, by pressing **ESC**, then type **:wq** and then press **ENTER** this will close the editor**.**



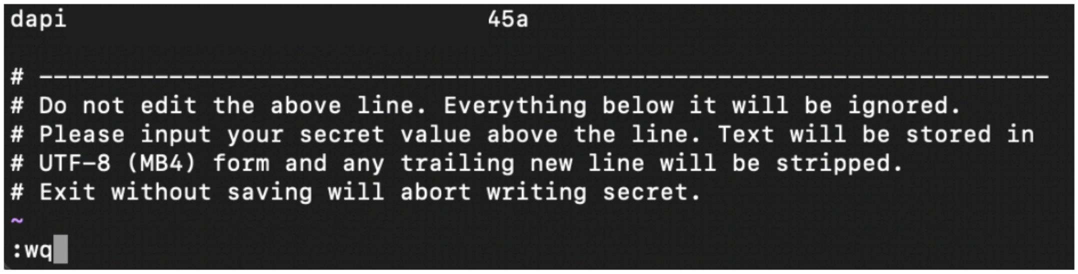
Generate New Token

Your token has been created successfully.

dapi

⚠ Make sure to copy the token now. You won't be able to see it again.

7



```
dapi                             45a

# ----------------------------------------------------------------------
# Do not edit the above line. Everything below it will be ignored.
# Please input your secret value above the line. Text will be stored in
# UTF-8 (MB4) form and any trailing new line will be stripped.
# Exit without saving will abort writing secret.
~
:wq
```