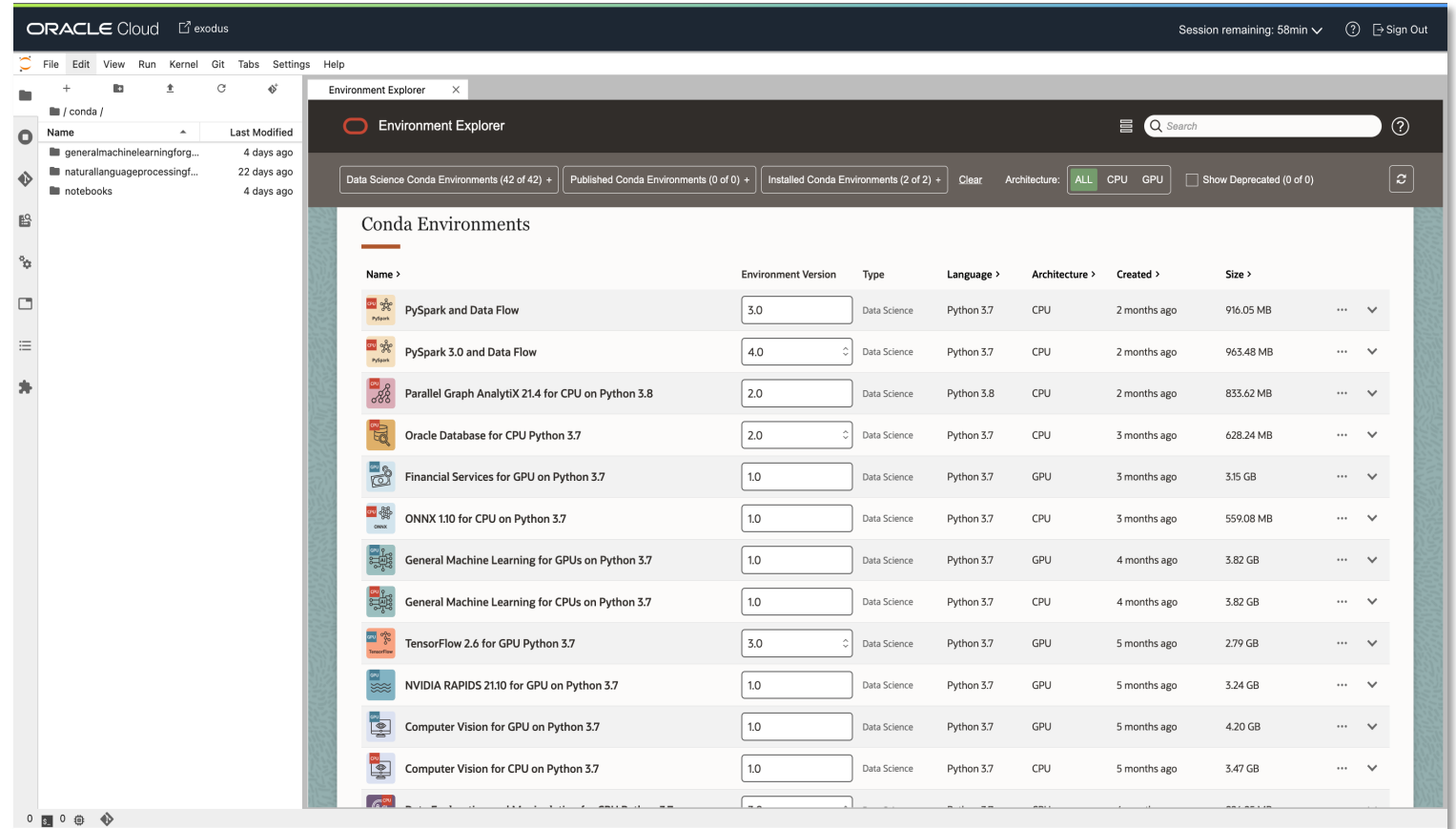


Using Conda in Oracle Data Science

By Nicholas Toscano

What is Conda Environments?

- Conda is like a virtual environment
- Let you run Python processes in different environments with different versions of the same library
- Manages different versions of Python that aren't installed system-wide Lets you upgrade libraries
- Supports the installation of packages for R, Python, Node.js, Java, etc.



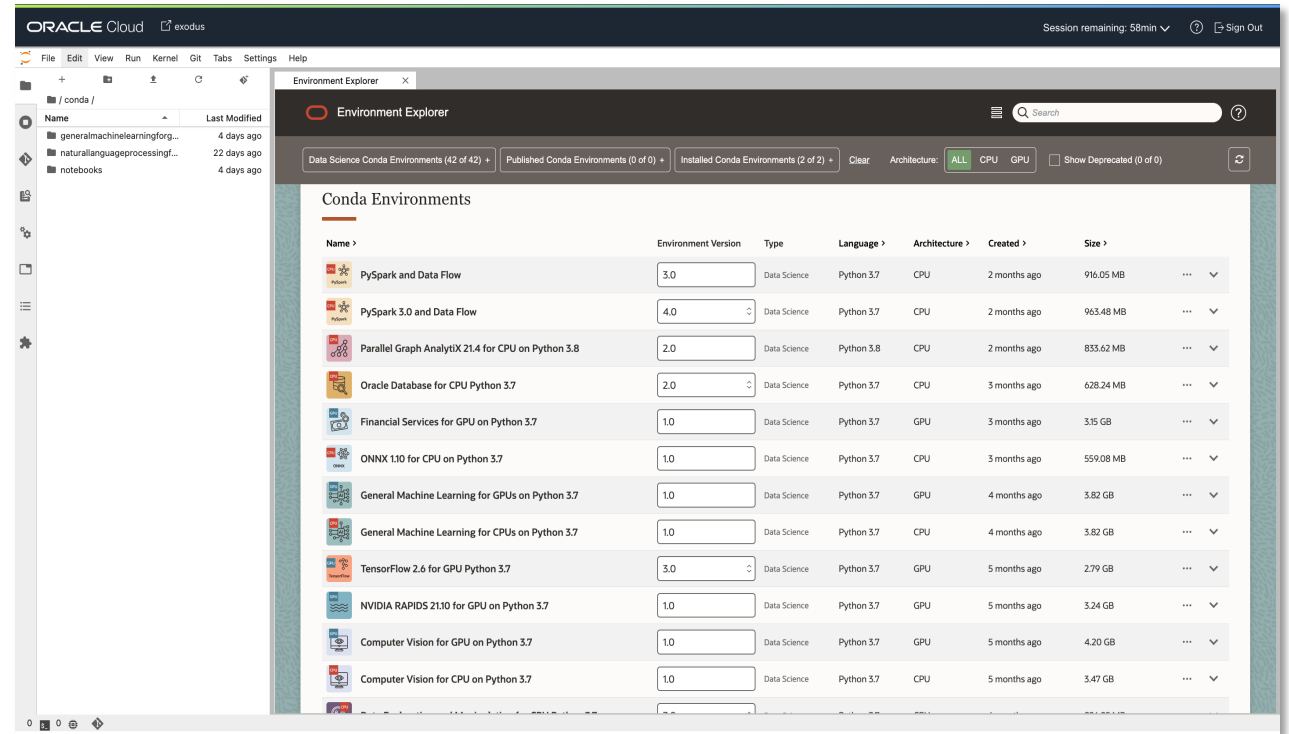
There are now over 42 pre-built conda environments to choose from, including ones dedicated to [Oracle PyPGX](#), [PySpark](#), [NVIDIA RAPIDS](#), and more.

Benefits of Conda Environments

- Install Python libraries from the different Conda channels:
 - [conda-forge](#)
 - pypi service
 - Third-party version control provider, such as github.com
- Environments portable through the [conda-pack](#) tool
 - Archive them in an Object Storage bucket
 - Or shipped across platforms and operating systems
- Access different Conda Environments as different notebook kernels in JupyterLab
 - Simultaneously execute different notebooks in different kernels with potentially conflicting sets of dependencies

Install Curated Conda Environments

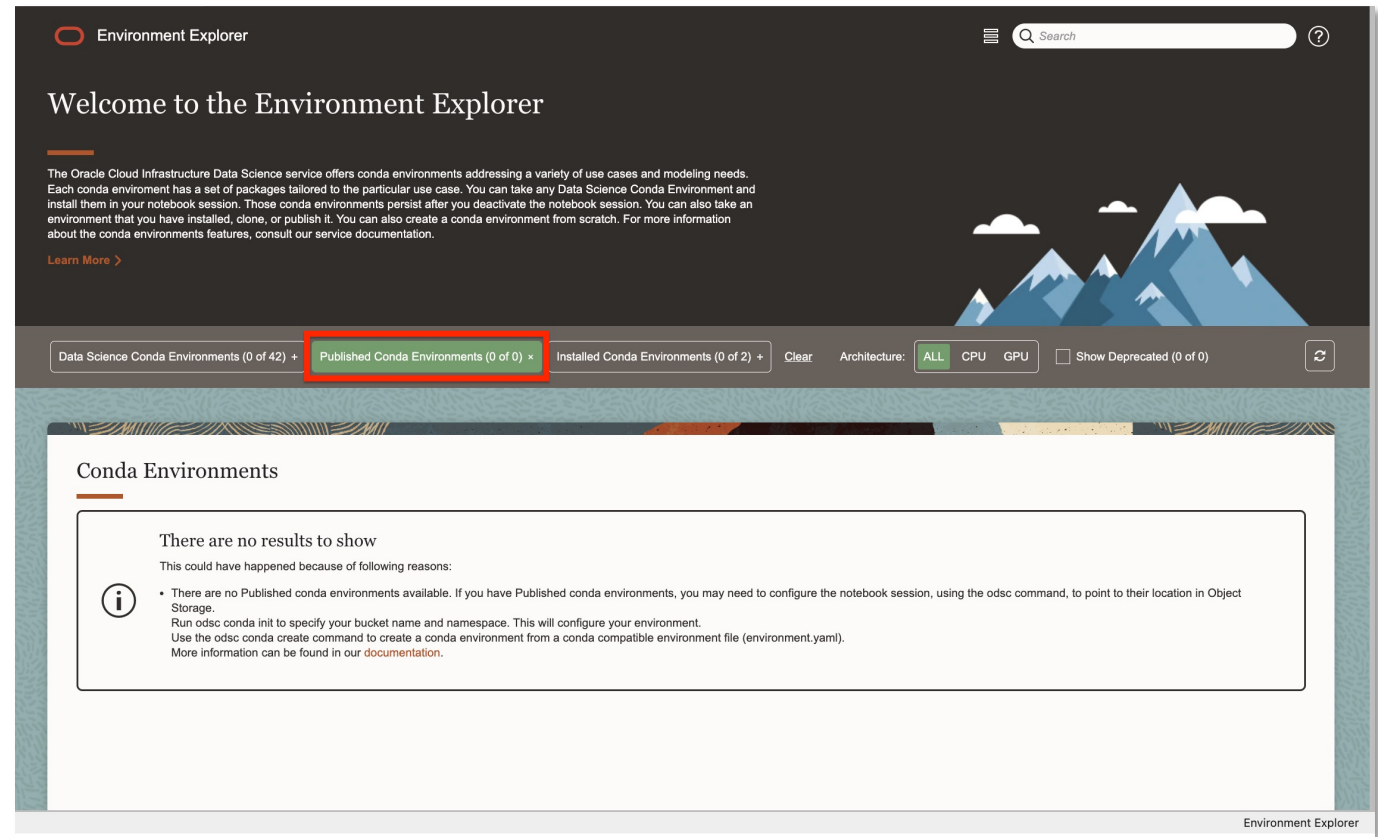
- From the odsc conda CLI or the Explorer extension, you can install one or more of the Data Science Conda Environments
- Env are built and curated by the OCI Data Science service team
- More Data Science Conda Environments are added over time:



Create Your Own Environment

Publish an environment and share it with colleagues across notebook sessions

- Create your own Conda Environment using **odsc conda create** command
- List what libraries you want to install in a Conda **environment.yaml** file
- Conda supports the installation of libraries from Conda channels and pip
- Publish your env to object storage bucket:
 - Use the **odsc conda publish** command
 - Share Conda Environments with colleagues
 - Install a published Conda in a different notebook session



Example Environments

PySpark

Provides a local development environment for a PySpark job. Ideal environment to test your Oracle Cloud Infrastructure Data Flow jobs before submitting them with ADS (also included in this environment).

General machine learning for CPUs

Includes the new versions of ADS, AutoML, and MLX, along with the usual machine learning suspects, including sklearn, xgboost, lightGBM, and others

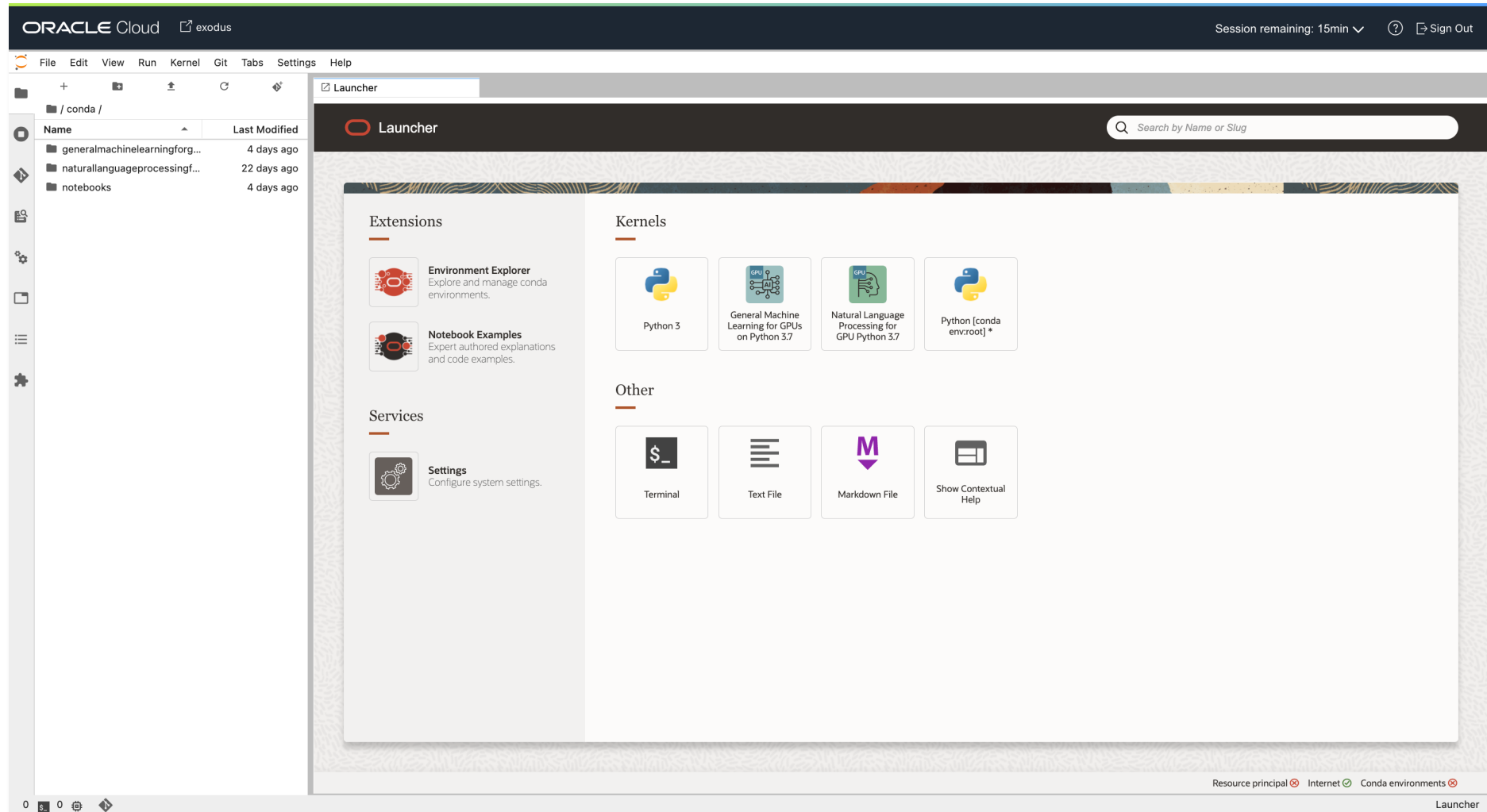
General machine learning for GPUs

Includes the new versions of ADS, AutoML, and MLX. This environment also includes TensorFlow 2.3.1 optimized for GPUs.



* See Oracle documentation for up-to-date information.

Step 1: Open or launch a notebook session



Step 2: Write a conda-compatible environment.yml File

```
channels:  
  - conda-forge  
dependencies:  
  - numpy  
  - pandas
```

- This file contains the channels and the dependencies that you want to install in your conda environment
- You can also select packages from [pypi](#)

Adding pip packages to the list of dependencies

```
channels:  
  - conda-forge  
dependencies:  
  - numpy  
  - pandas  
  - pip:  
    - scikit-learn==0.24.2
```

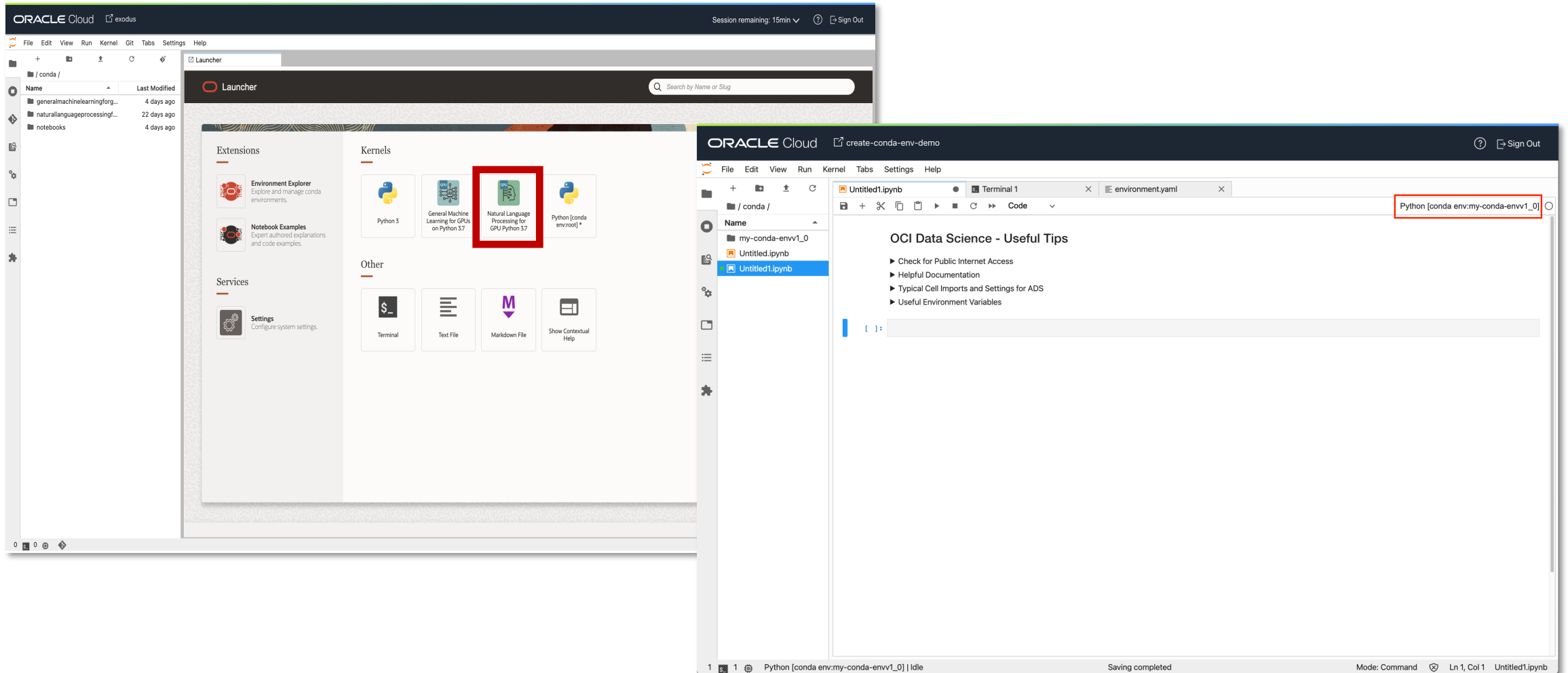
- You can install packages directly from [pypi](https://pypi.org/)

Step 3: Create the conda environment with odsc conda create Command

Open a terminal window in your notebook session and run:

- This command will create a brand new kernel in your notebook session called my-conda-env
- A version v1.0 will be assigned to the conda environment by default and appended to the name of conda slugname
- You can change that by assigning a value to the create command optional parameter -v

Step 4: Validate the new conda environment



Step 4: Validate the new conda environment

In your notebook, import numpy and pandas and confirm that these libraries are available in your environment. Do the same thing for scikit-learn if you installed it from pypi:

```
import numpy
import pandas
import sklearn

print(numpy.__version__)
print(pandas.__version__)
print(sklearn.__version__)
```

Step 5: Publish the new environment

- Publishing a conda environment consists of creating a [pack](#) and uploading it to an Object Storage bucket that you specify.
- We recommend that you publish conda environments to ensure that a model training environment can be reproduced or [re-used for model deployment](#)
- You can use the odsc CLI to publish an environment.
- First, you need to specify the target object storage bucket where the published environment will be stored. This can be done through the [odsc conda init](#) command:

```
odsc conda init -b -n -a {api_key | resource_principal}
```

Step 5: Publish the new environment

- Use the `odsc conda publish` command. Specify the slug name of the conda environment you just created
- The slug name is the name of the conda environment and its version. It corresponds to the notebook kernel name minus the "conda-env:" part

```
odsc conda publish -s my-conda-envv1_0
```

- Go to your object storage bucket in the OCI console and confirm that the new conda pack is stored in the bucket.

—END—