

# Project :



## what are projects ?

work spaces for teams of data scientists .

## create projects ?

there are tow ways to create a data science projects to organize your notebook session and model .

- from the console
- from the ADS SDK

### From the Console

- ✓ You must have necessary policies in place.
- ✓ You must select a compartment.
- ✓ (optional) Enter unique project

### From ADS SDK

- ✓ Use the ProjectCatalog object:
- ✓ Create a project by calling the `create_project()` method.
- ✓ Specify the compartment ID.

## Viewing , Editing and deleting projects ?

You can view all projects on the project list page ,

View displays project details and metadata:

- Display name
- Description
- OCID
- Created on date/time
- Created by (user OCID)
- Tags

## Editing

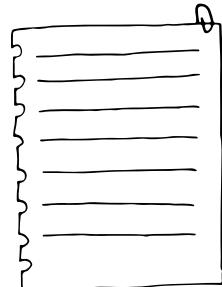
you can view and edit projects through any of the OCI interface

the only editable fields are :

- **Display name**
- **Tags**
- **description**

View	Edit	Delete
<p>You can view all projects on the Project List page.</p> <p>View displays project details and metadata:</p> <ul style="list-style-type: none"><li>➢ Display name</li><li>➢ Description</li><li>➢ OCID</li><li>➢ Created on date/time</li><li>➢ Created by (user OCID)</li><li>➢ Tags</li></ul>	<p>You can view and edit projects through any of the OCI interfaces.</p> <p>The only editable fields are:</p> <ul style="list-style-type: none"><li>➢ Display Name</li><li>➢ Description</li><li>➢ Tags</li></ul>	<p>You can delete a project if it's empty and all associated Data Science resources are deleted.</p> <p>Deleted projects remain in the list for 30 days.</p> <p>Use state filter to filter projects.</p>

## Notebook session :



### Q - what are notebook session ?

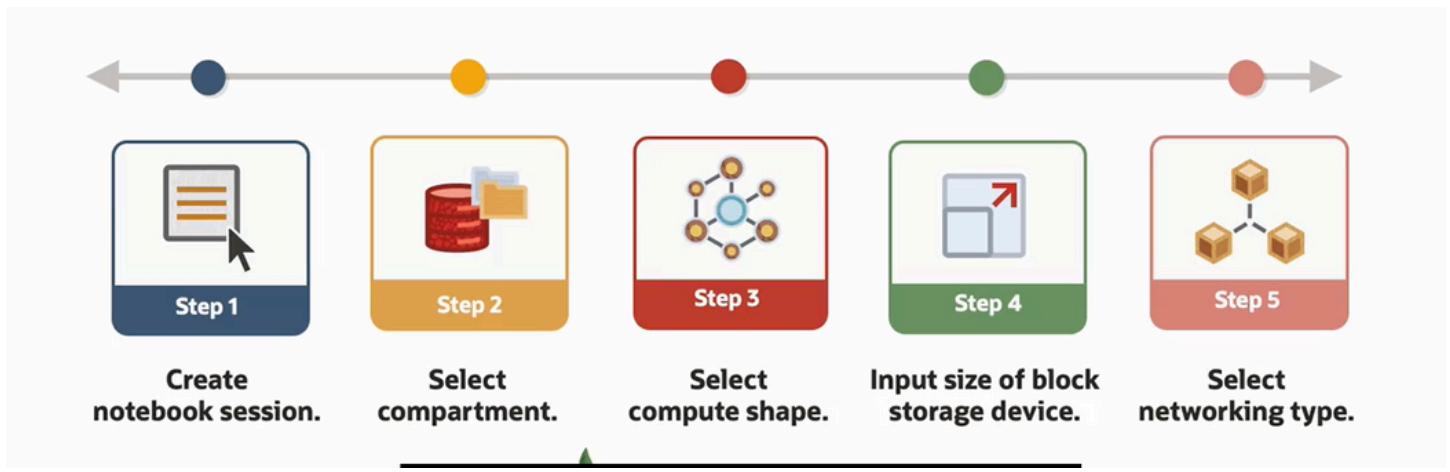
jupyterLab interface for building and training models .

- offer support for CPU and GPU shapes

provide persistent session storage for :

- Data
- Notebooks
- Environments

## create notebooks sessions from the console :



### Jupyterlab interface :

the open source and data science console interfaces differ in three main ways

- Launcher
- Environment Explorer
- Github Extension

### Features of jupyterlab :

- Menu bar
- Launcher
- Left sidebar

### what is a conda environment ?

conda is an open source package and environment management system

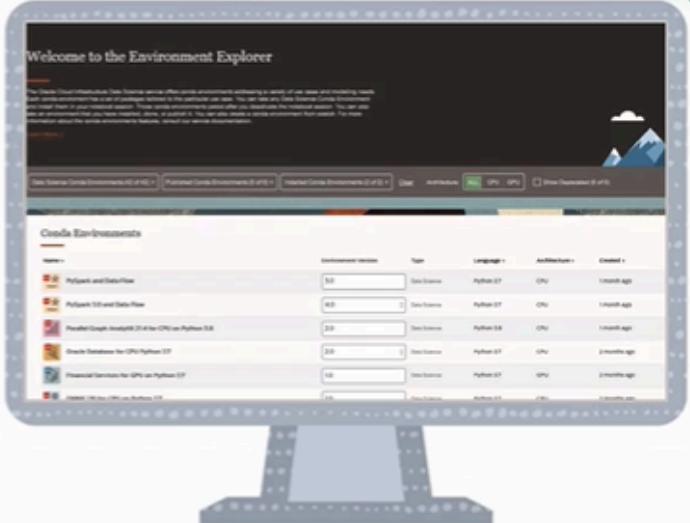
Benefits of conda environment .

1. install and update package and their dependencies
2. isolate different software configurations
3. switch between environment
4. share configurations with others
5. develop in a notebook and deploy a model or job
6. leverage a reproducible research environment

### environment Explorer :

## A GUI tool that allows users to:

- Explore and manage conda environments
  - Data Science Conda Environments
  - Published Conda Environments
  - Installed Conda Environments
- Discover information about the conda
- Card and list views
- Search for conda environments
- Apply filters CPU/GPU, deprecated



Are curated by the data science service team ?

- PyTorch
- Tensorflow
- Healthcare
- Exploratory Data analysis

## chapter 2

# Data science conda environment



oracle has created several data science conda environment they are :

- built on open source software
- downloadable and customizable
- Accessible from environment

Types of conda environment :

- ONNX

- PyTorch
- TensorFlow
- Rapids
- PyPGX
- PySpark
- intel Extension for Scikit - learn

## Application Based



### Use case based

Computer vision	General machine learning
Data exploration and manipulation	Natural language processing
Financial services	Neurophysiology
	Oracle Database

Conda environment Families :

Families are grouped based on

- python version
- Architecture

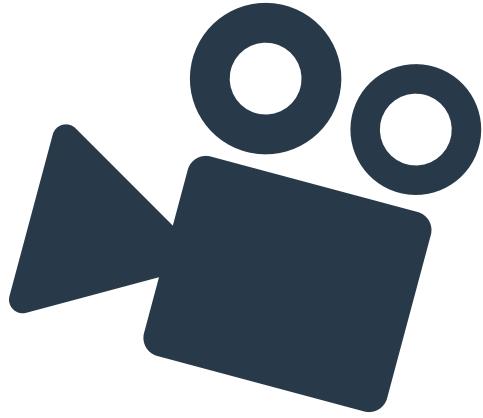
### Natural Language Processing

- + Natural Language Processing for CPU on Python 3.7 (version 2.0)
- + Natural Language Processing for GPU on Python 3.7 (version 2.0)
- + Natural Language Processing for CPU on Python 3.7 (version 1.0)
- + Natural Language Processing for GPU on Python 3.7 (version 1.0)

### General Machine Learning

- + General Machine Learning for CPU on Python 3.7 (version 1.0)
- + General Machine Learning for GPU on Python 3.7 (version 1.0)
- + General Machine Learning for CPU on Python 3.6 (version 1.0)
- + General Machine Learning for GPU on Python 3.6 (version 1.0)

## computer vision



### **Use cases :**

1. object identification and tracking
2. image stitching and compression
3. Facial recognition
4. eye movement tracking

### **Top libraries :**

- oracle ads
- open CV
- scikit -image
- Pillow
- Torchvision

---

### **data Exploration and manipulation :**

### **Use cases :**

- data set ingestion ,processing ,and visualization
- stream consumption from oracle cloud infrastructrue streaming

### **Top libraries :**

- oracle -ads
- Pandas
- seaborn
- plotly
- matplotlib
- kafka - python
- pandraallel

---

### **General machine learning :**

### **use cases :**

- data manipulation

- supervised machine learning
- Generic machine learning
- AutoML functionality
- Machine learning explainability (MLX)

### **Top libraries :**

- category -encoder
  - lightgbm
  - oracle AutoML
  - oracle MLX
  - oracle - ads
  - scikit - learn
  - Tensorflow
  - Xgboost
- 

## **Natural language processing**

### **Use cases :**

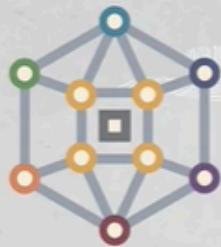
- Text extraction
- Key - phrase extraction
- Parts - of - speech tagging

### **Top libraries :**

- oracle - ads
  - pytorch - lightning
  - nltk
  - lime
  - eli5
  - transformers
  - keybert
  - simpletransformers
- 

## **ONNX**

# ONNX



## Use Cases

- Portability and interoperability between ML frameworks
- Transfer models between ML frameworks
- ONNX Runtime library allows you to run the models on different platforms



## Top Libraries

onnx  
onnxconverter-common  
onnxmltools  
onnxruntime  
oracle-ads

# oracle database :



## Use Cases

- Database queries using the ADS Connector, SQLAlchemy, and ipython-sql
- Perform analytics in the database without having to move the data to the notebook



## Top Libraries

ipython-sql  
mysql-connector-python  
oracle-ads  
SQLAlchemy

# PyTorch

## use cases :

- computer vision , natural language processing and general machine learning
- deep neural networks and algorithms for deep learning
- tensor computing with strong acceleration on GPUs

## Top libraries :

- category - encoder
  - daal4py
  - oracle - ads
  - pandas
  - scikit - learn
- 

## pyspark



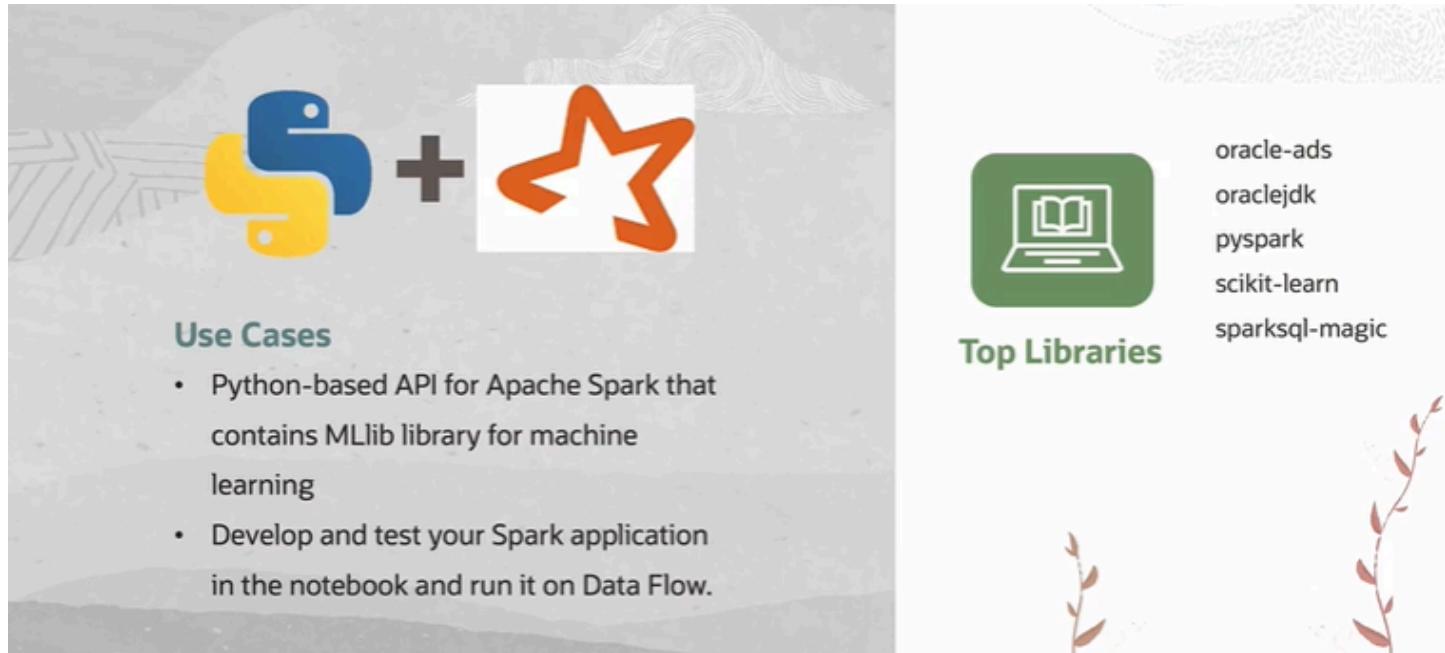
**Use Cases**

- Python-based API for Apache Spark that contains MLlib library for machine learning
- Develop and test your Spark application in the notebook and run it on Data Flow.

**Top Libraries**



- oracle-ads
- oraclejdk
- pyspark
- scikit-learn
- sparksql-magic



## Tensorflow

### use cases :

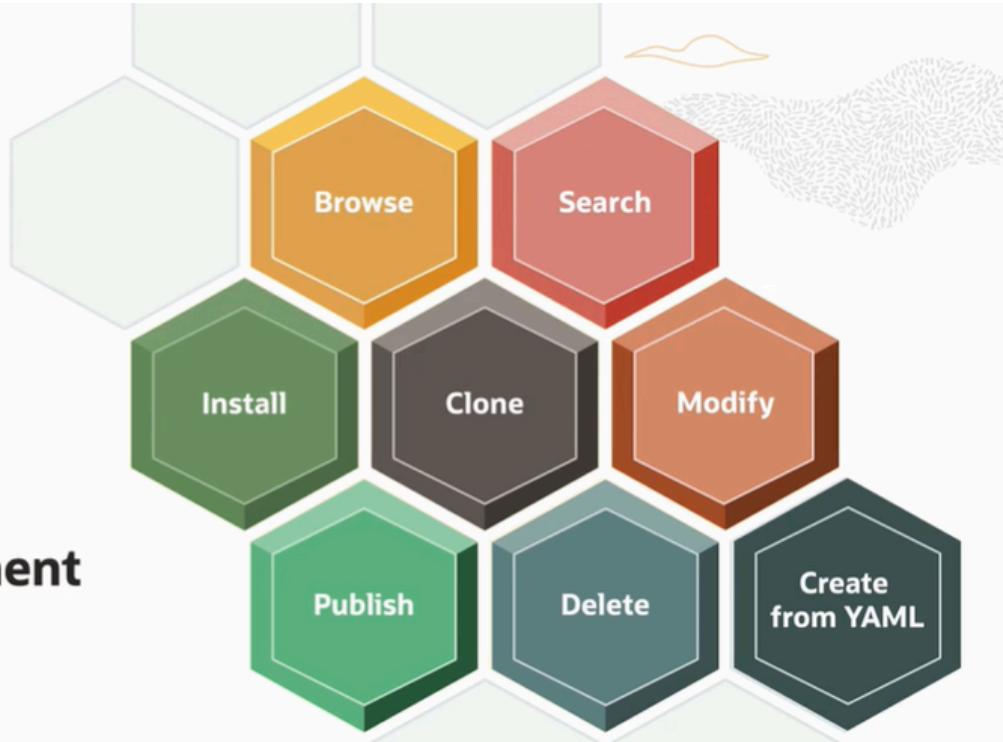
- Machine learning
- deep neural networks
- Flexible architecture runs on CPUs , GPUs and TPUs

### Top libraries :

- oracle - ads
  - pandas
  - category - encoders
  - scikit - learn
  - TensorFlow
  - Tensorboard
- 

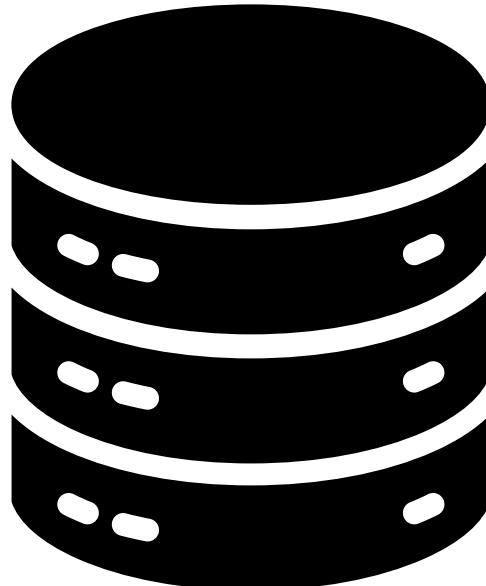
### conda environment Functionality :

## Conda Environment Functionality



clone : users can copy an installed conda environment .

OCI Vault



Q - why is the OCI vault important in data science ?

- interact with other services
- interacting requires the use of credentials
- store credentials in the OCI Vault
- do not store them insecurely in the code or configuration files

OCI vault supports AES,RSA and ECDSA algorithms

## Virtual private vault:

Is a dedicated isolated partition in a hardware security module (HSM)

Can store up to 1000 key versions by default

Provides better isolation of your keys/secrets

Can back up to object storage



### Master encryption Keys (MEK)

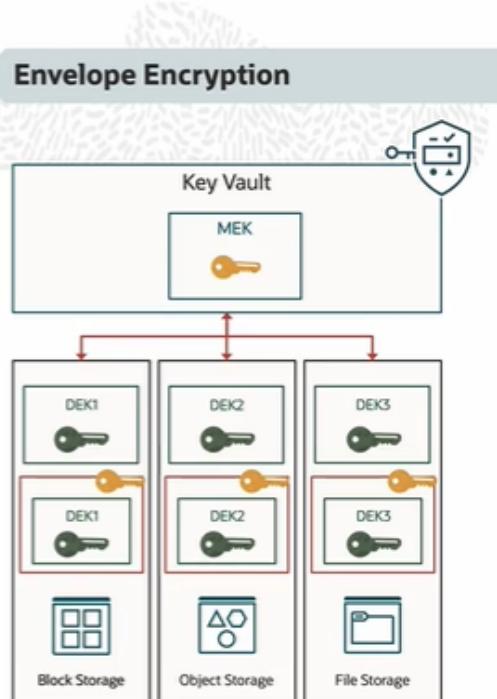
You create or import MEKs into vault

select the algorithm and bits

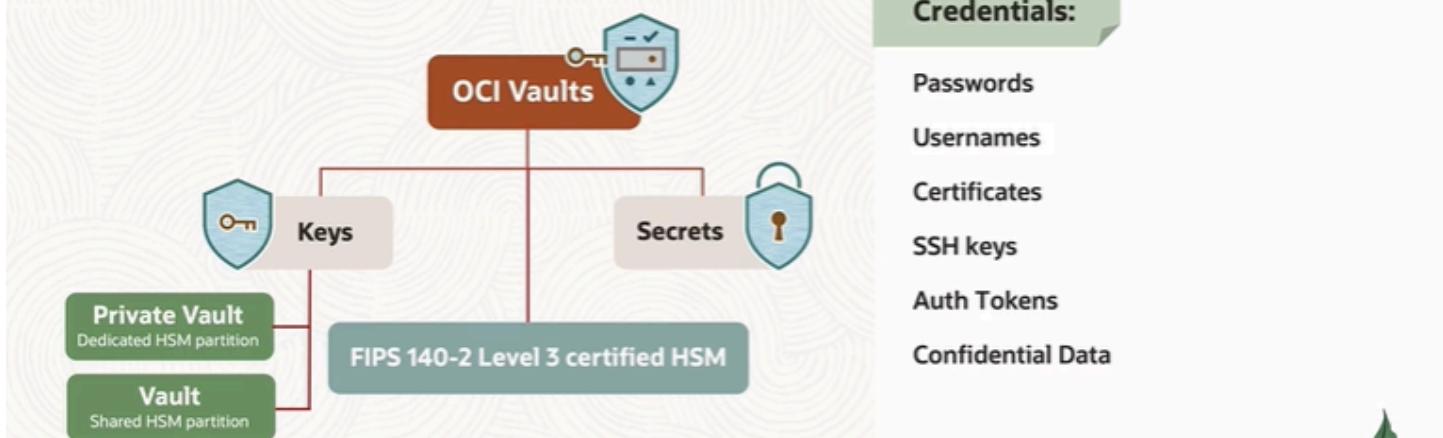
MEKs are used to generate data encryption keys

### Data encryption Keys (DEKs)

DEKs are generated by the MEKs used to encrypt data



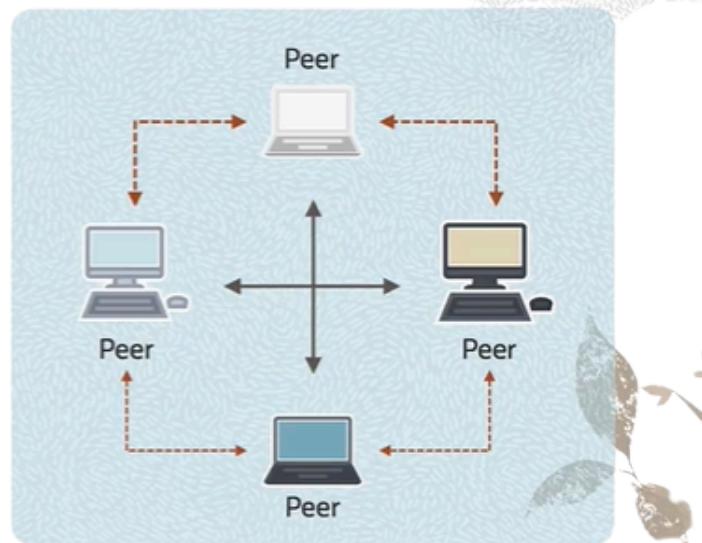
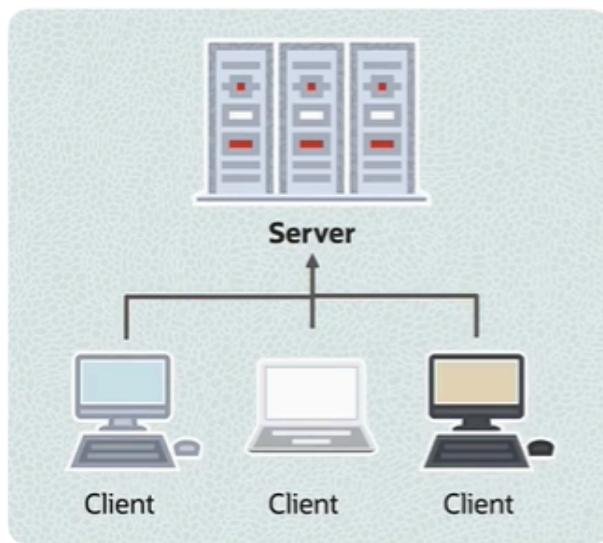
## What Are Secrets?



## Code Repositories (Git)

### what a version control system ?

also known as source code management a version control system manages changes to a computer program , documents or other ...

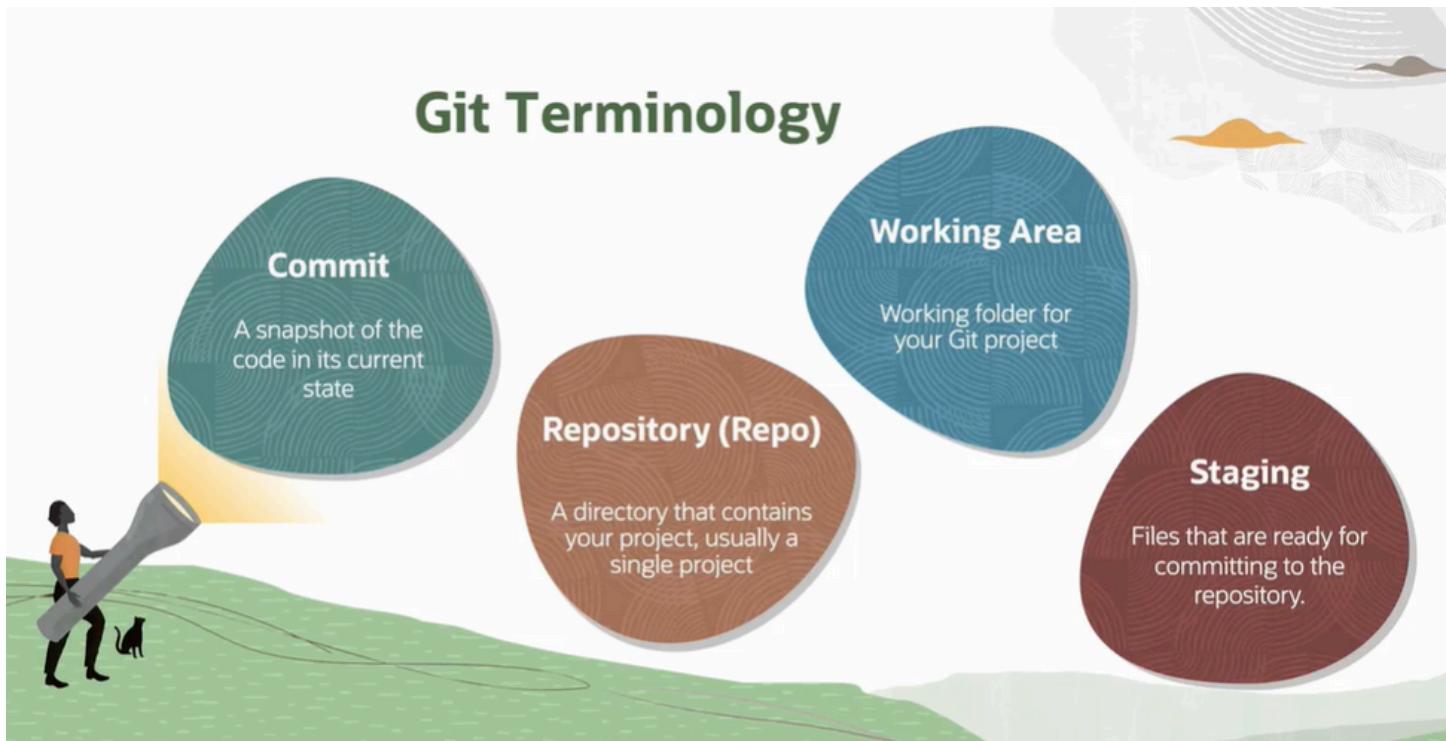


### using Git in data science workflows ?

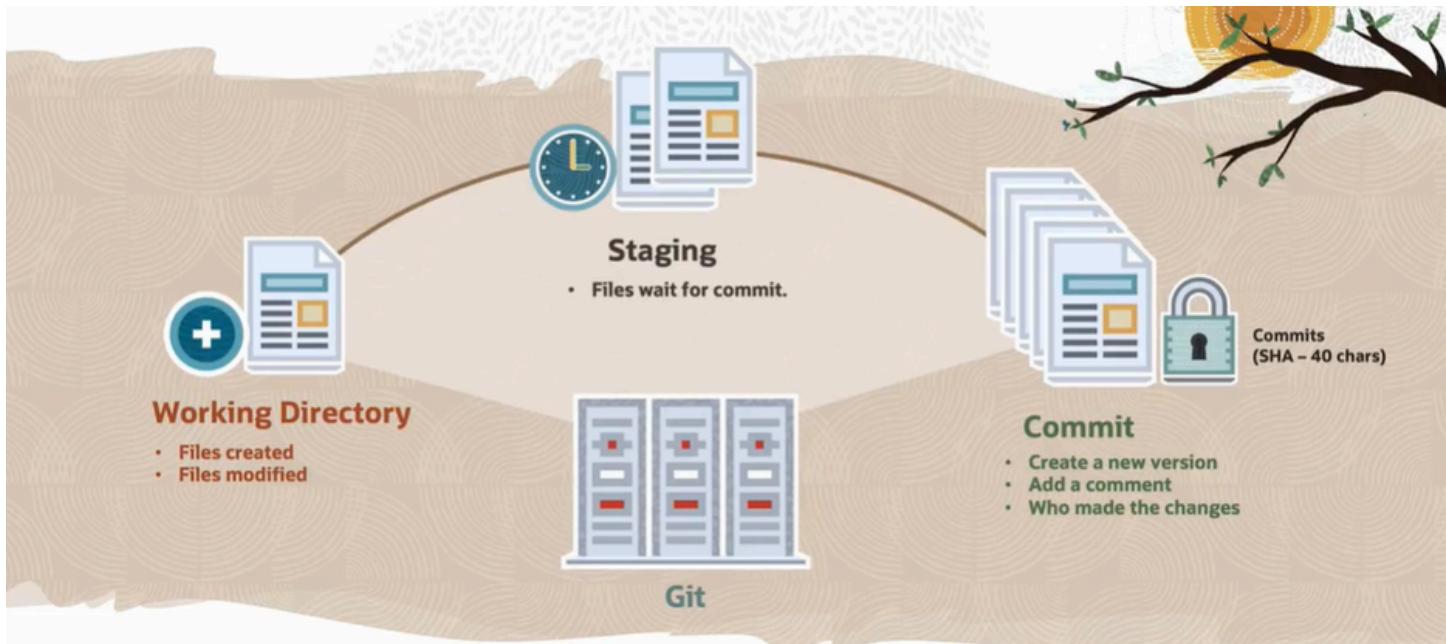
Git is version control system that allows you to :

- Track changes made to a set of files .

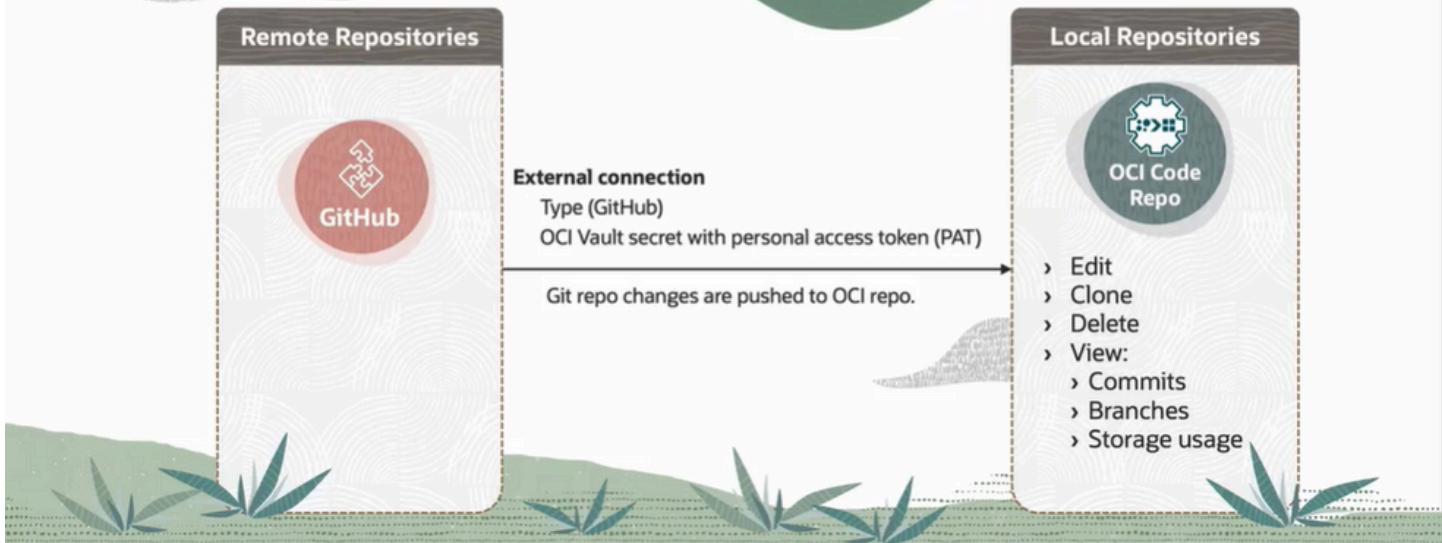
- Revert to previous version of the files as needed .
- 



**Git Flow Diagram**



# OCI Code Repository



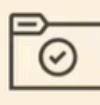
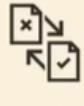
## Working with a Remote Repo: GitHub

Whereas Git is a version control system, GitHub is a cloud-based service that hosts Git projects.



# Commands in Git



<b>init</b> 	<b>clone</b> 	<b>add</b> 	<b>Commit</b> 
Create a new repo or convert an existing local project folder to git repo.	Create a local development clone of a remote repo.	Move the changes into the staging area for the next commit.	Take snapshots of the local repo for the next push to a remote repo.
<b>remote</b> 	<b>fetch</b> 	<b>push</b> 	<b>pull</b> 
Create, view, and delete connections to repositories.	Download commits, files, and data from a remote repo to a local repo.	Upload local content to a remote repo.	Update the local repo to match the content from a remote repo.

The end chapter 1