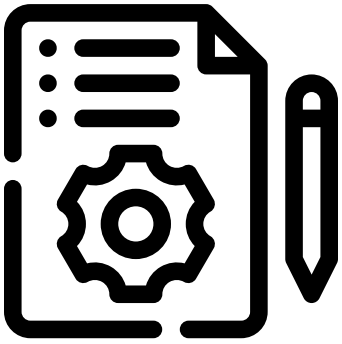
Project :



what are projects ?

work spaces for teams of data scientists .

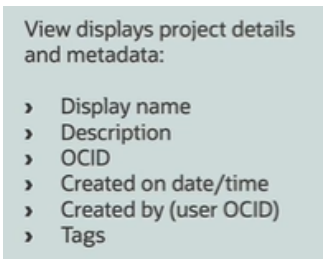
create projects ?

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

- from the console
- from the ADS SDK

Viewing , Editing and deleting projects ?

You can view all projects on the project list page ,

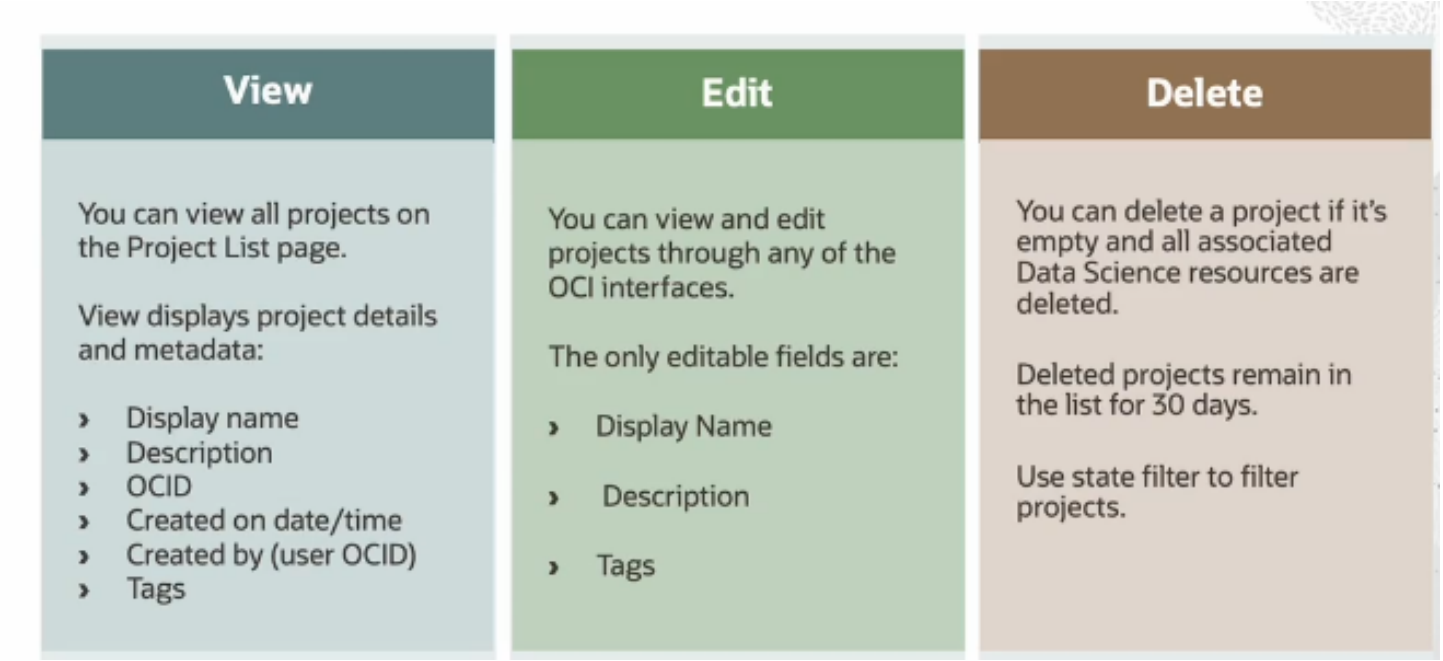


Editing

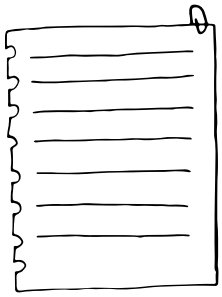
you can view and edit projects through any of the OCI interface

the only editable fields are :

- **Display name Tages description**



Notebook session :



Q - what are notebook session ?

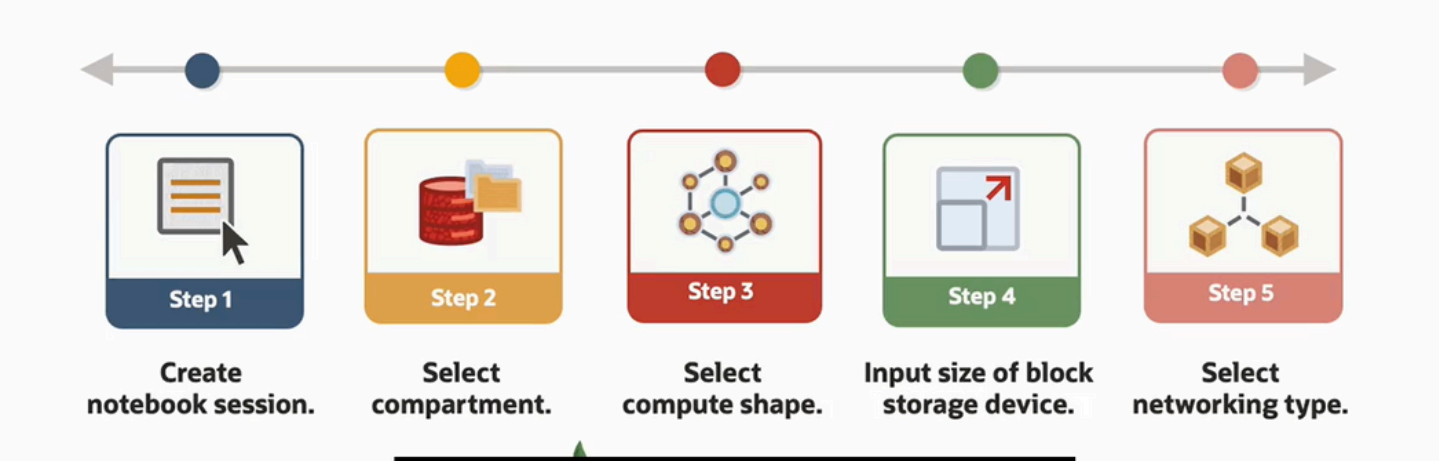
jupyterLab interface for building and training models .

- offer support for CPU and GPU shapes

provide presistent 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

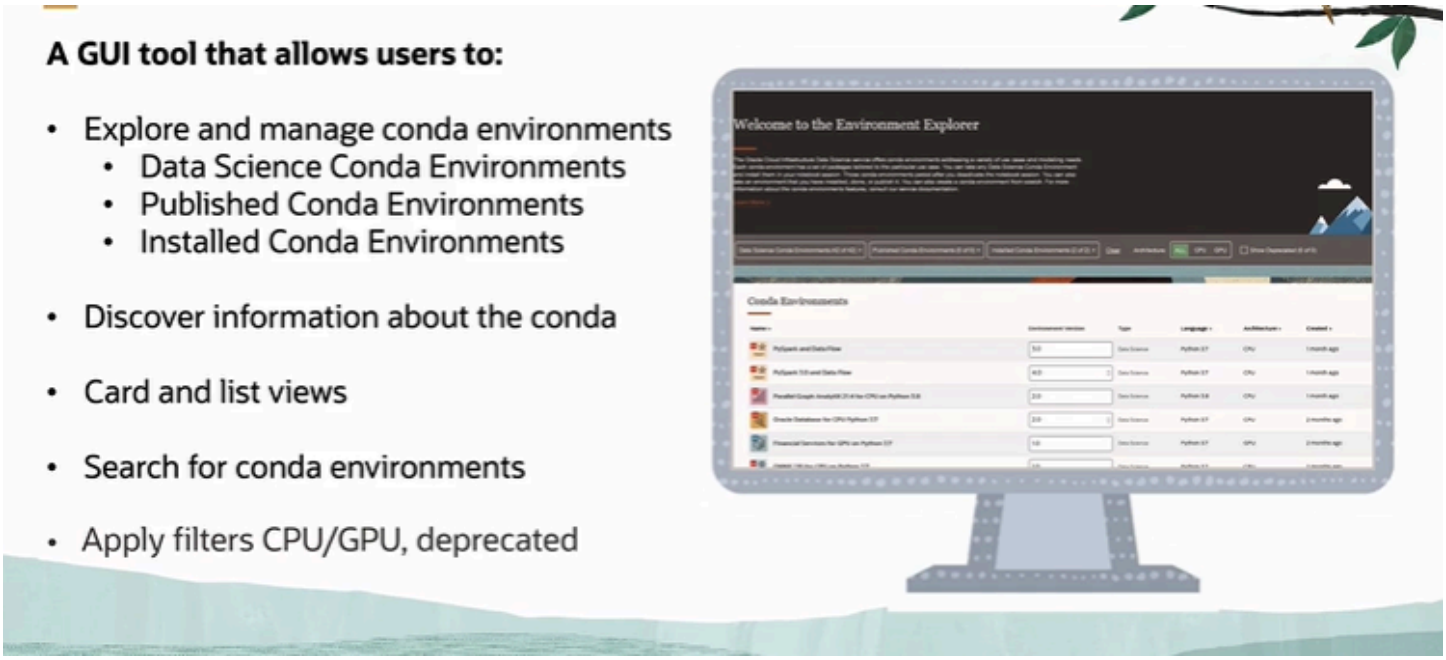
---

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 :



Are curated by the data science service team ?

- **PyTorch**
- **Tensorflow**
- **Healthcare**
- **Exploratory Data analysi**

---

chapter 2

Data science conda environment

oracle has created several data science conda environment they are :

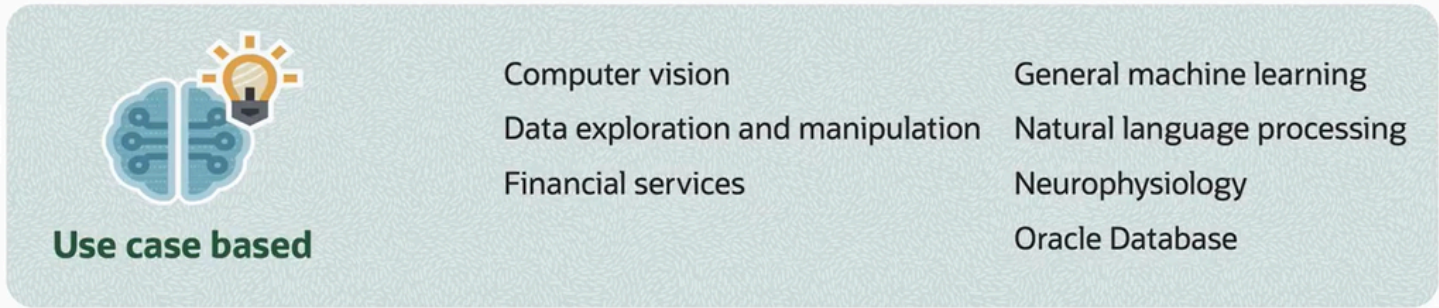
- built on open source software
- downloadable and customizable
- Accessible from environment

Types of conda environment :

- ONNX

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

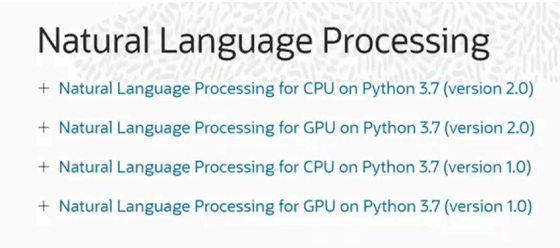
Application Based



Conda environment Families :

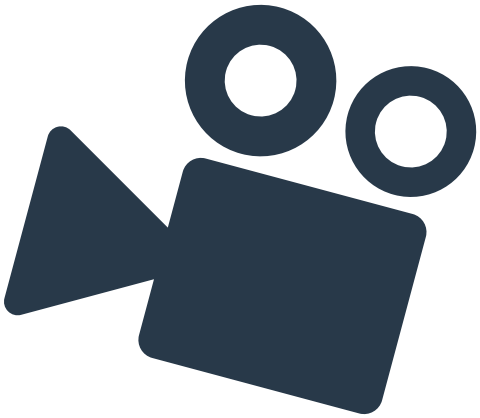
Families are grouped based on

- python version
- Architecture



---

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

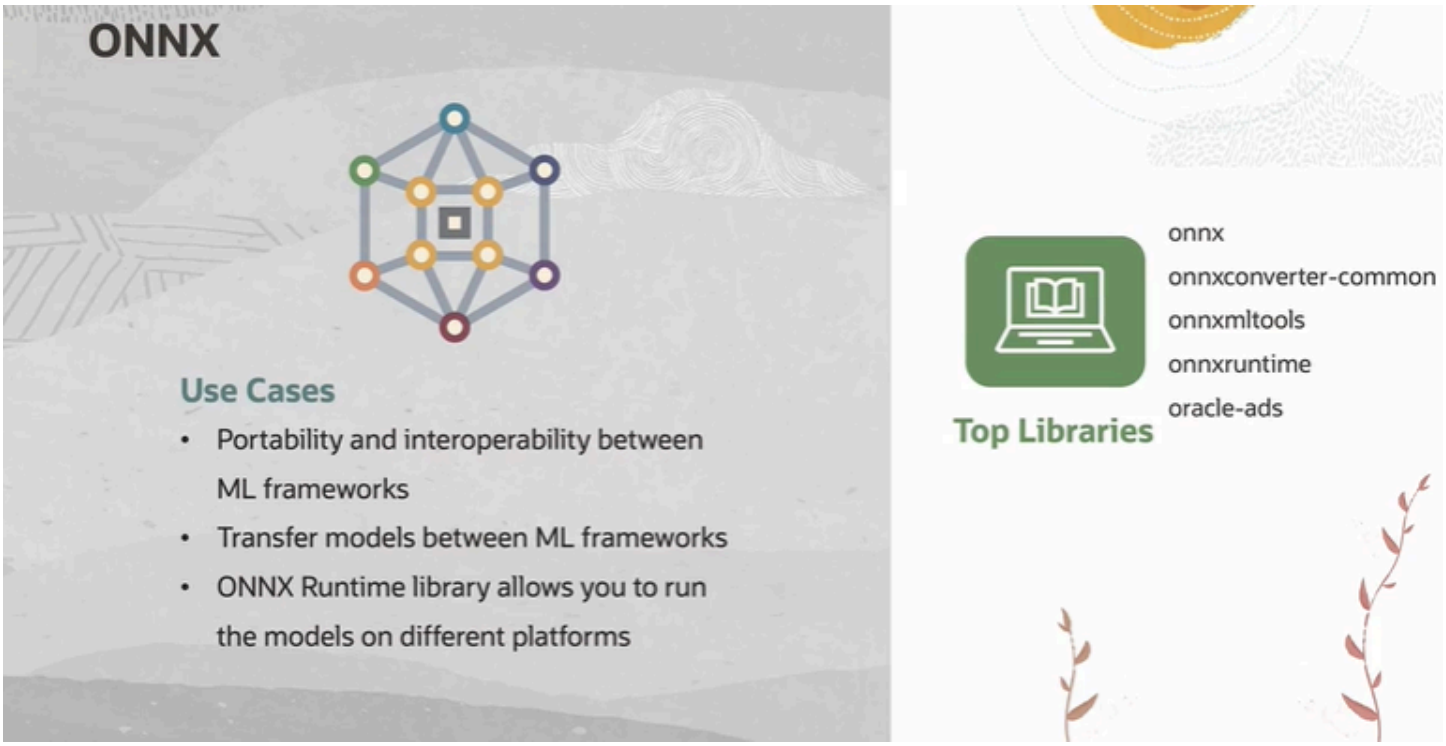
Use cases :

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

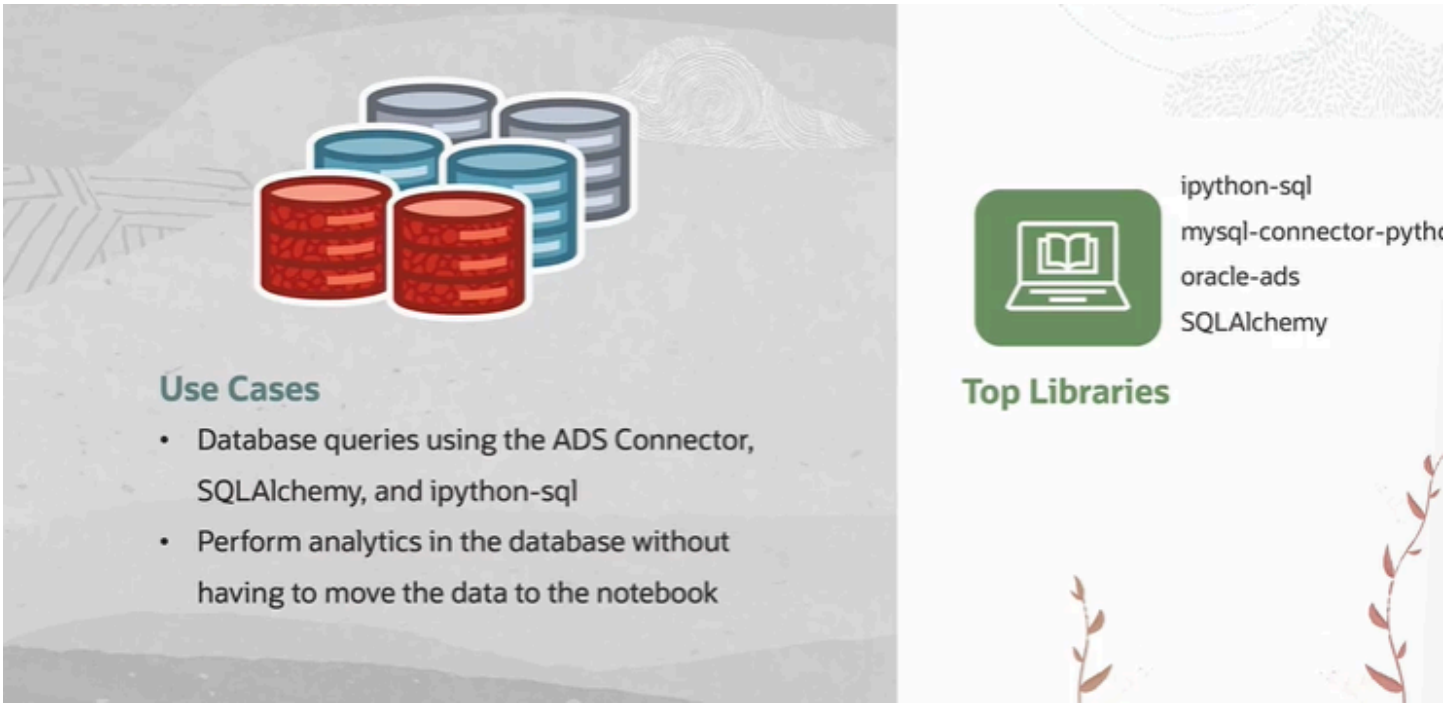
Top libraries :

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

ONNX



oracle database :



PyTorch

use cases :

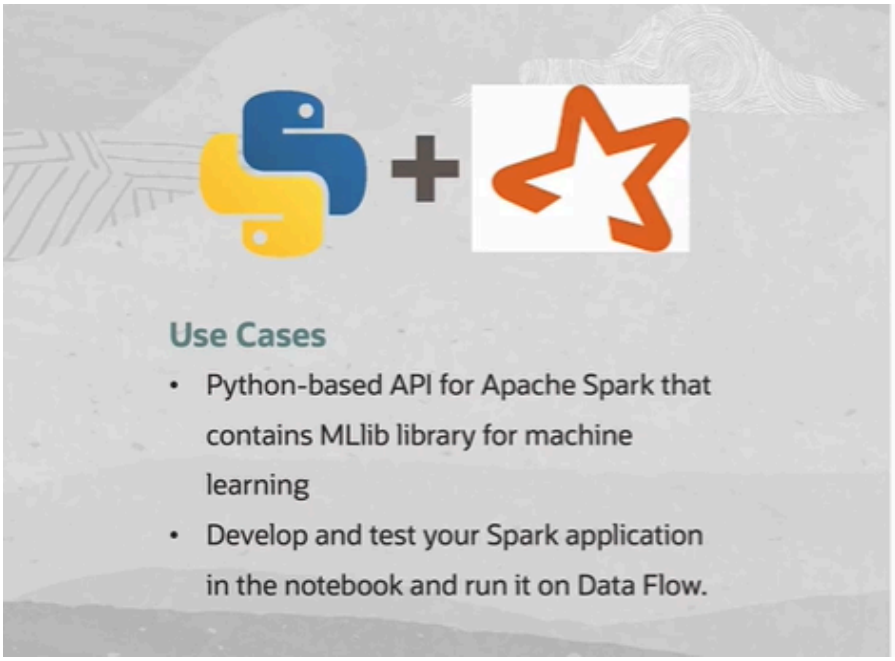
- computer vision , natural language processing and general machine learning
- deep neural networks and algorithms for deep learning
- tensor computing with strong acceleration on GPUs

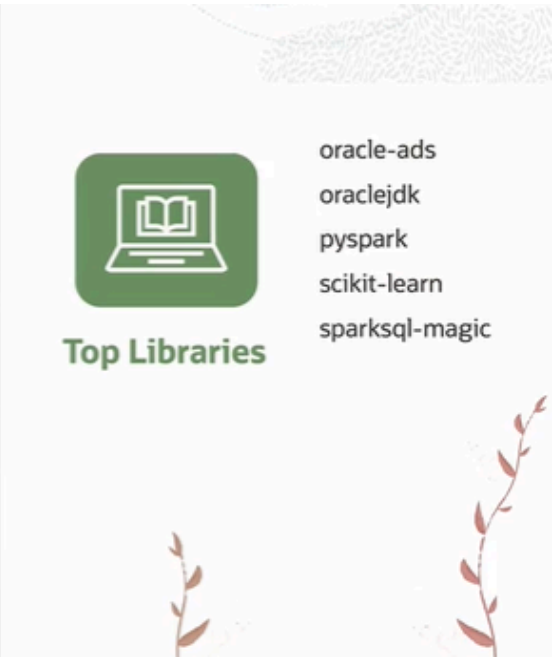
Top libraries :

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

---

pyspark





Tensorflow

use cases :

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

Top libraries :

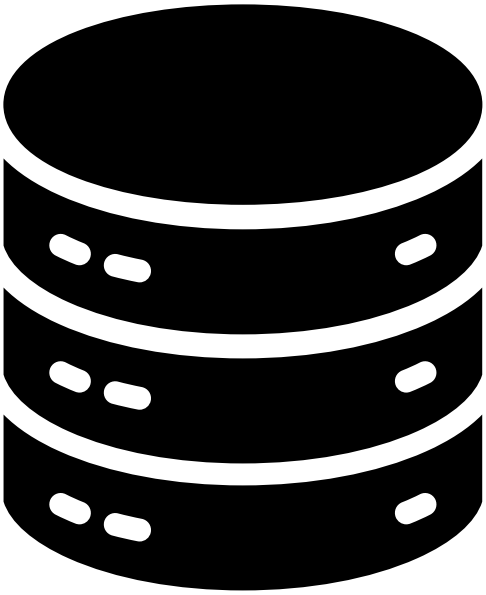
- oracle - ads
- pandas
- category - encoders
- scikit - learn
- TensorFlow
- Tensorboard

---

conda environment Functionality :

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)

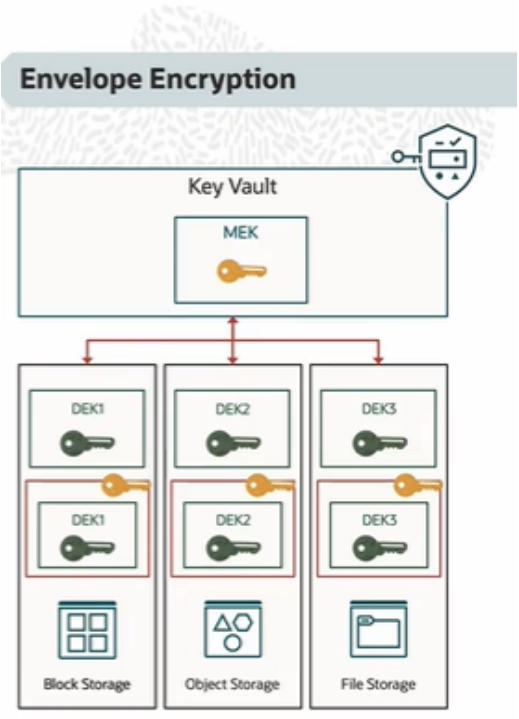
You create or import MEKs into vault

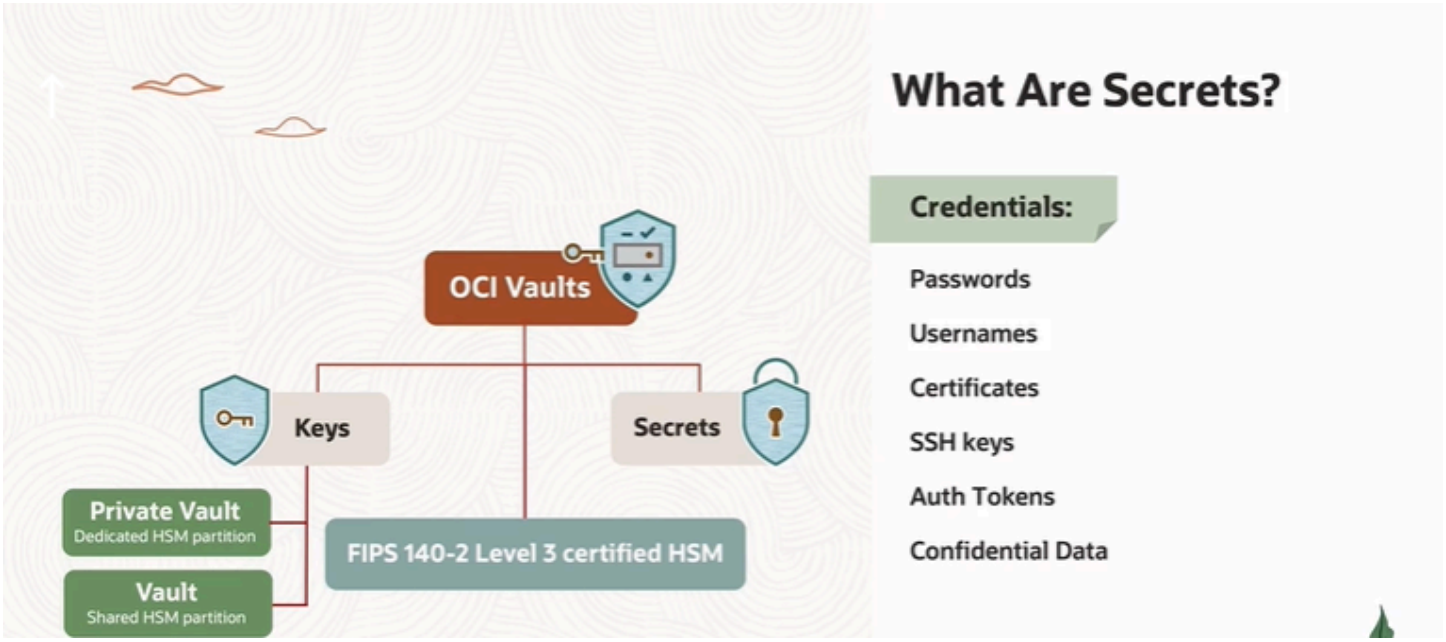
select the algorithm and bits

MEKs are used to generate data encryption keys

Data encryption Keys (DEKs)

DEKs are generated by the MEks used to encrypt data

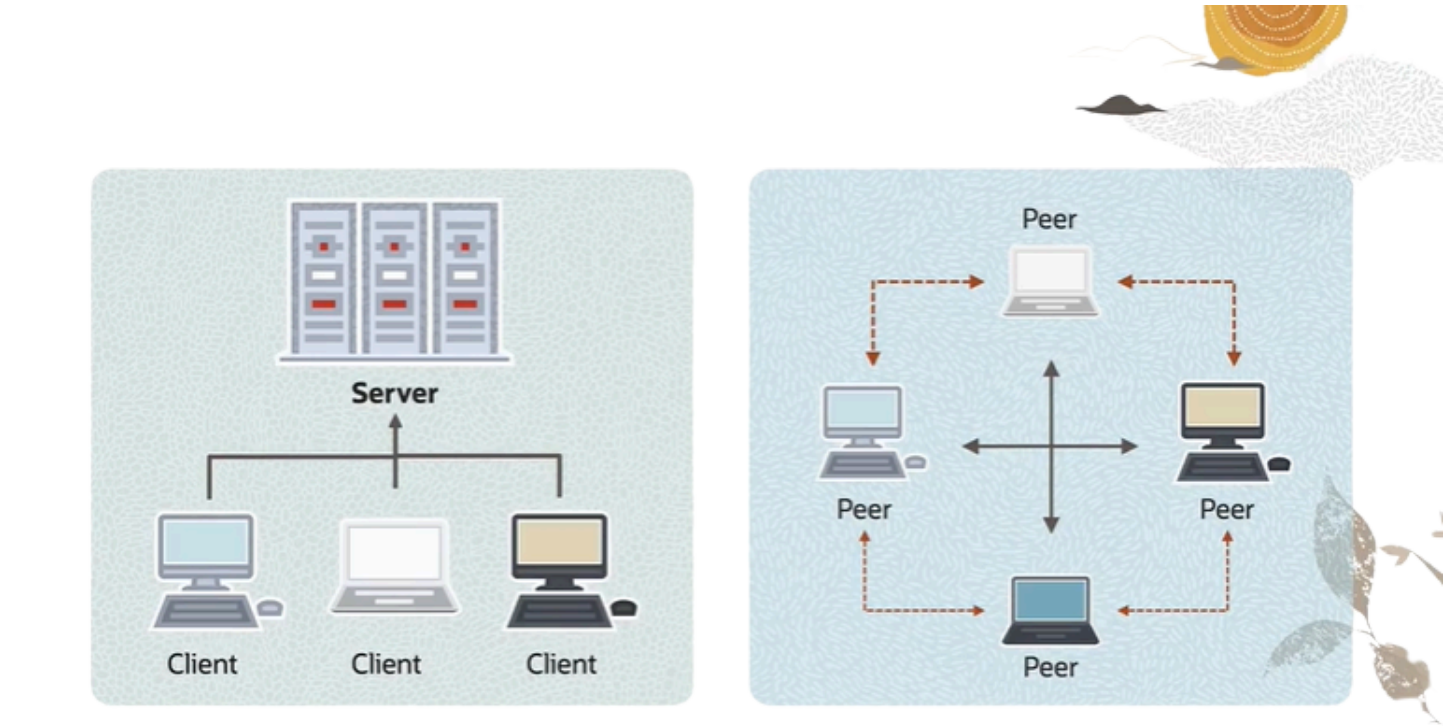




Code Repositories (Git)

what a version control system ?

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



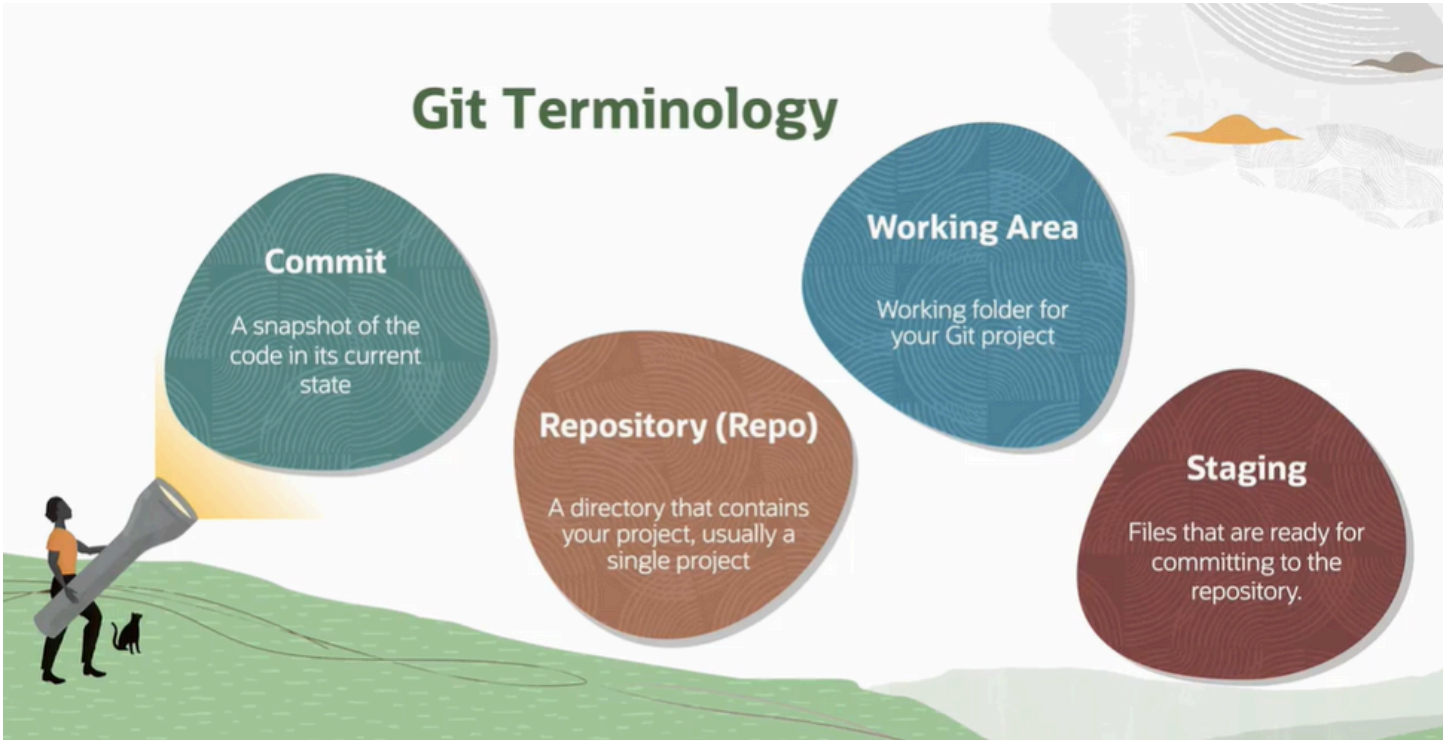
---

useing Git in data science workflows ?

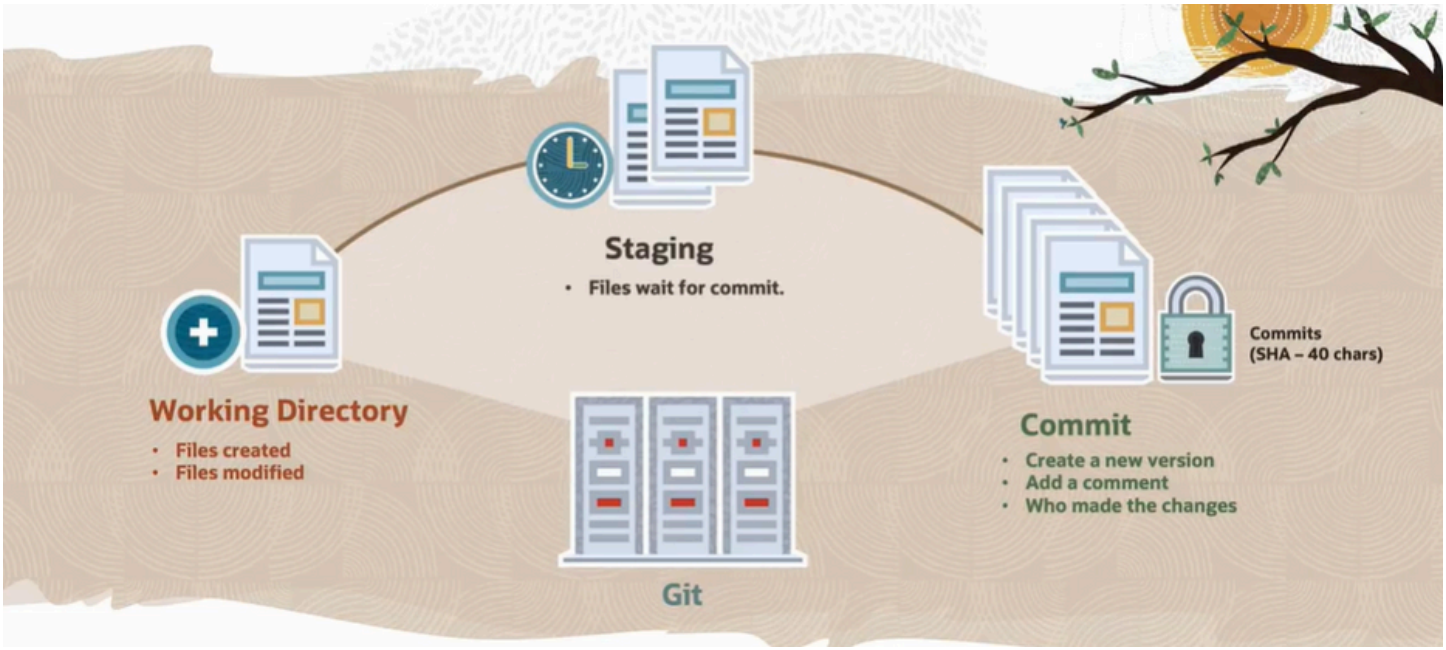
Git is version control system that allows you to :

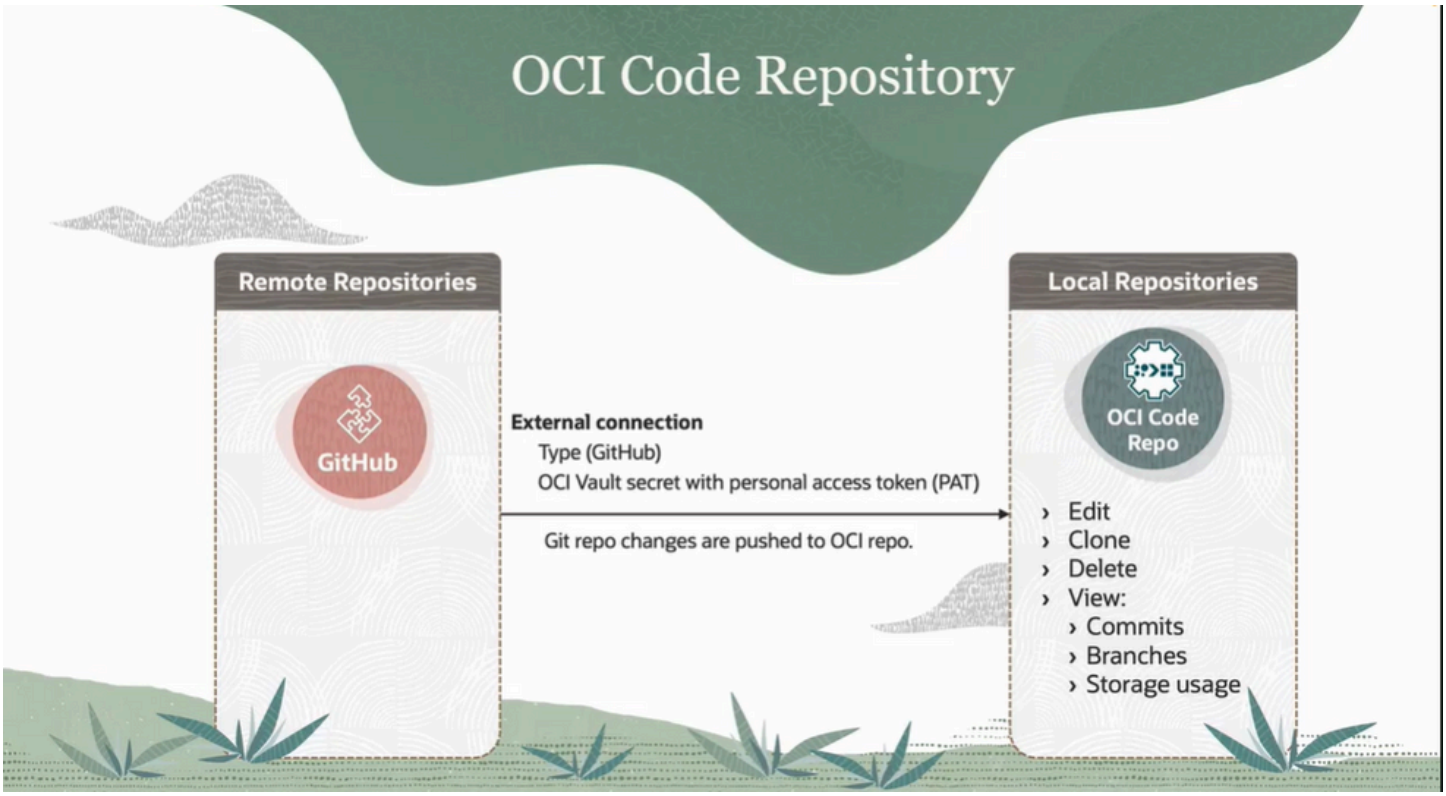
- Track changes made to a set of files .

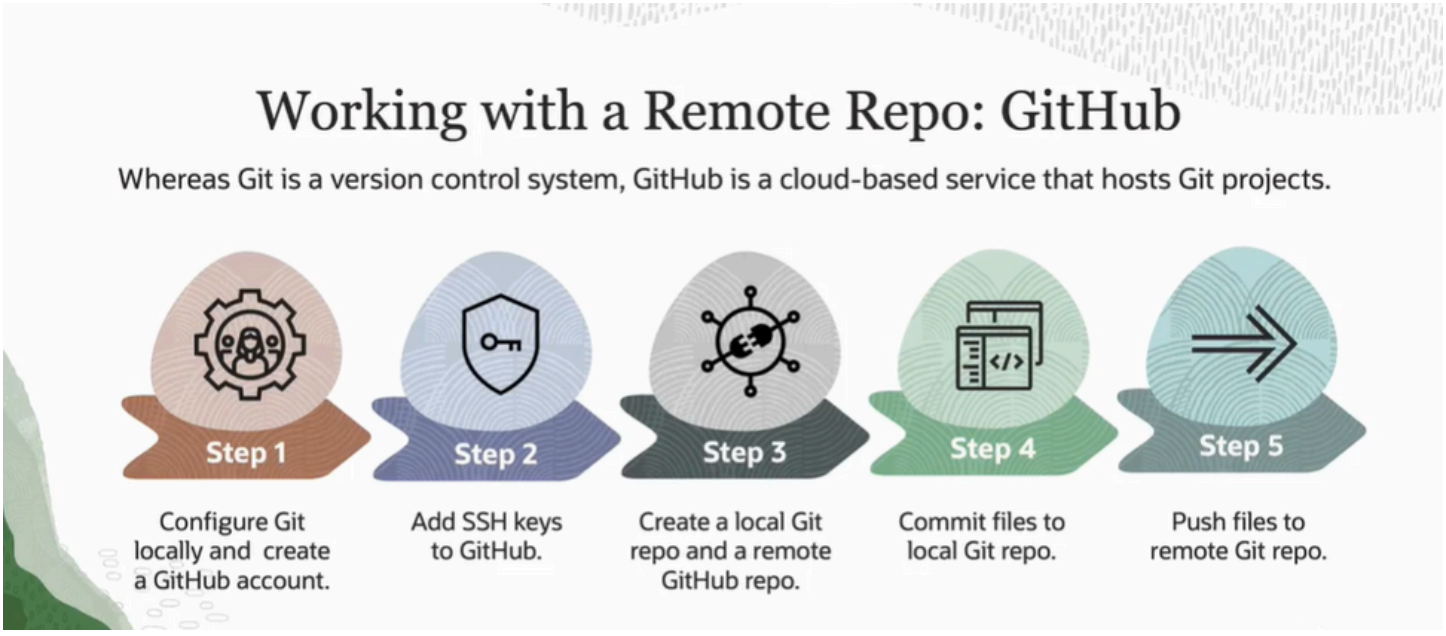
- Revert to previous version of the files as needed .



Git Flow Diagram




OCI Code Repository
Remote Repositories
GitHub
External connection
Type (GitHub)
OCI Vault secret with personal access token (PAT)
Git repo changes are pushed to OCI repo.
Local Repositories
OCI Code Repo
› Edit
› Clone
› Delete
› View:
› Commits
› Branches
› Storage usage


Working with a Remote Repo: GitHub
Whereas Git is a version control system, GitHub is a cloud-based service that hosts Git projects.
Step 1
Configure Git locally and create a GitHub account.
Step 2
Add SSH keys to GitHub.
Step 3
Create a local Git repo and a remote GitHub repo.
Step 4
Commit files to local Git repo.
Step 5
Push files to remote Git repo.

Commands in Git

init

Create a new repo or convert an existing local project folder to git repo.

clone

Create a local development clone of a remote repo.

add

Move the changes into the staging area for the next commit.

Commit

Take snapshots of the local repo for the next push to a remote repo.

remote

Create, view, and delete connections to repositories.

fetch

Download commits, files, and data from a remote repo to a local repo.

push

Upload local content to a remote repo.

pull

Update the local repo to match the content from a remote repo.

~~The end chapter 1~~