NAME : NAVIYA DHARSHINI A S
ROLL NO : 23AD083
DAY -08 - 11/07/2025

DVC (Data Version Control) - Full Notes from Basics to Advanced

---

🔰 1. What is DVC?

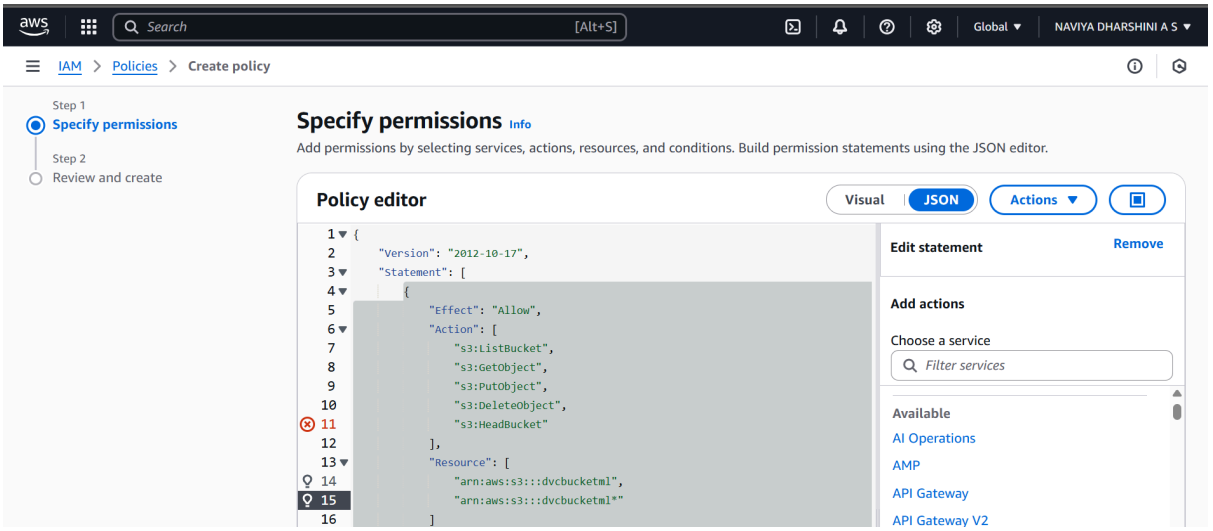
- **DVC = Git for Data + ML Pipelines**
- Manages **data**, **models**, and **experiments** in ML projects.
- Works with Git but tracks **large files**, **directories**, and **ML stages**.
- Makes projects **reproducible**, **shareable**, and **collaborative**.

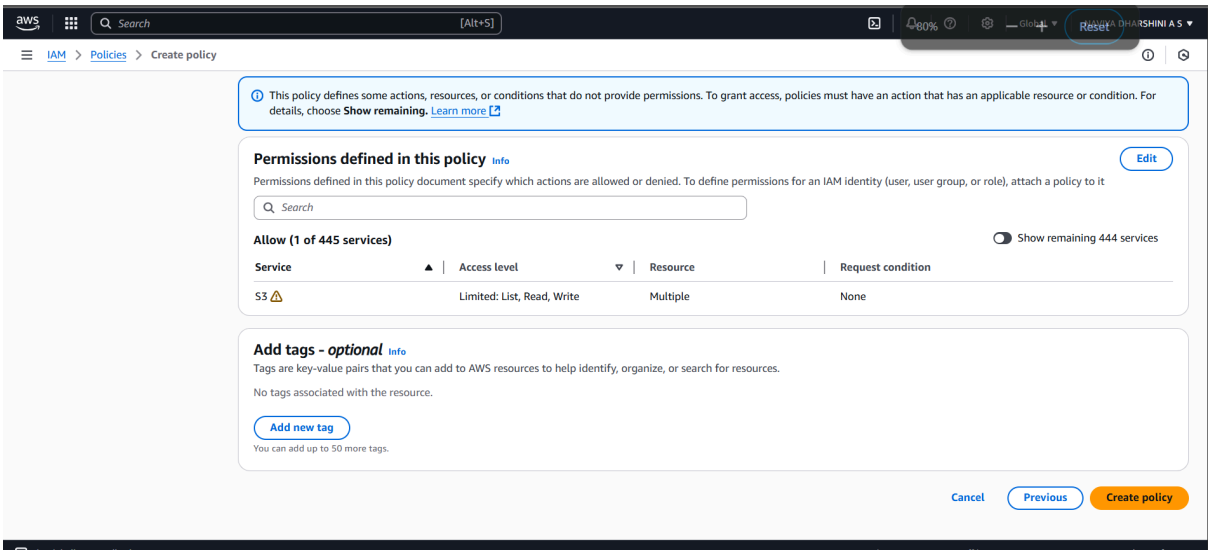
⚙️ 2. Why DVC?

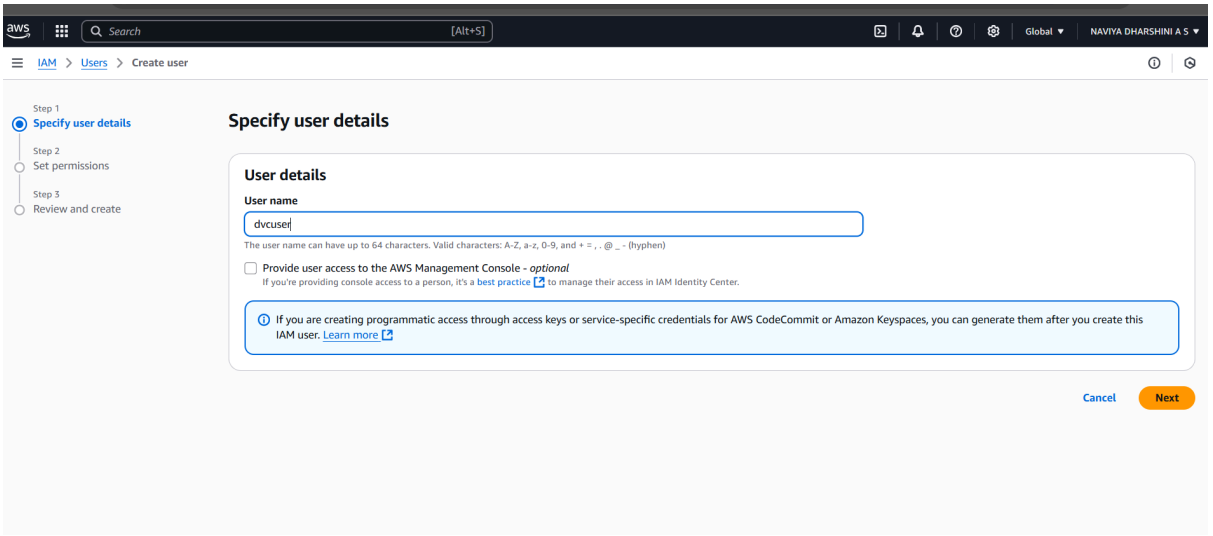
🔥 Git Problem	✅ DVC Solution
Can’t track large datasets	DVC tracks large files via `.dvc` metadata
Git lacks pipeline tracking	DVC handles pipelines + stages like `make`
No reproducibility for ML	DVC ensures same input = same output
Can't share data easily	DVC supports remote storage (S3, GDrive etc)

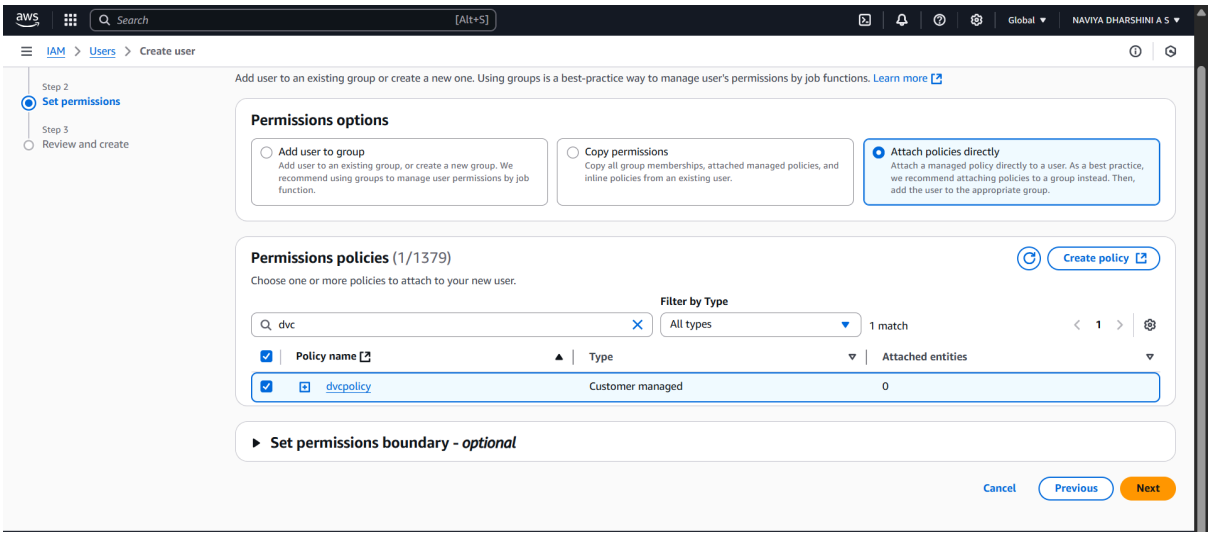
📁 4. DVC Project Structure

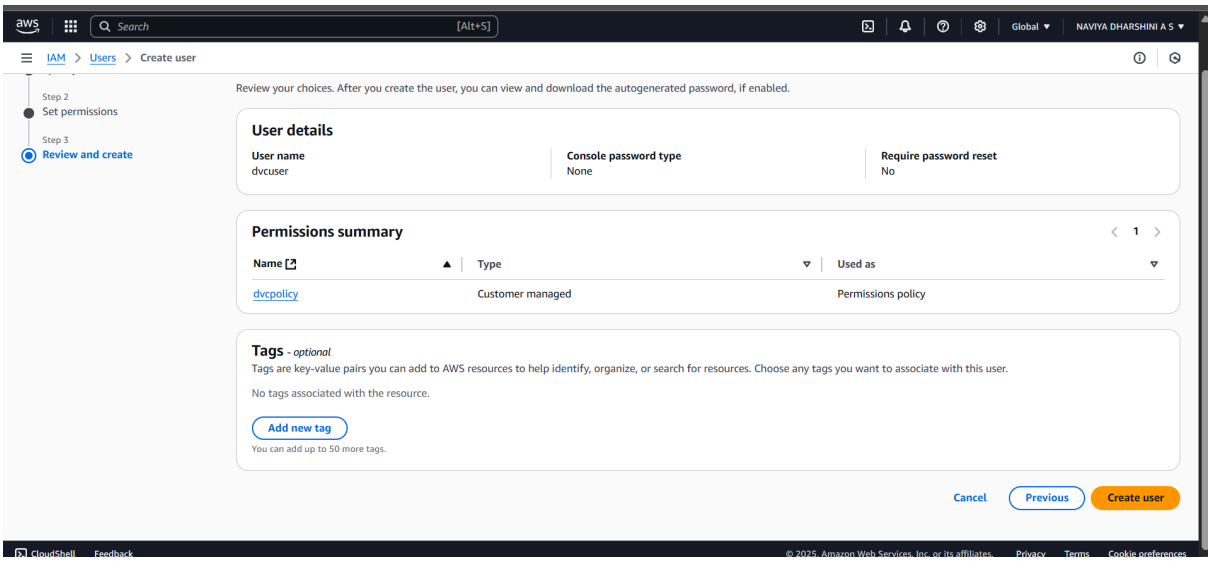
```
my-ml-project/
├── data/               # Raw datasets (not tracked by Git)
├── data.dvc            # DVC metadata for dataset
├── model.pkl           # Trained model (tracked by DVC)
├── model.pkl.dvc
├── dvc.yaml            # ML pipeline stages
├── dvc.lock            # Exact command runs (hashes etc.)
├── .dvc/               # DVC internal files
└── .git/               # Git repo
```

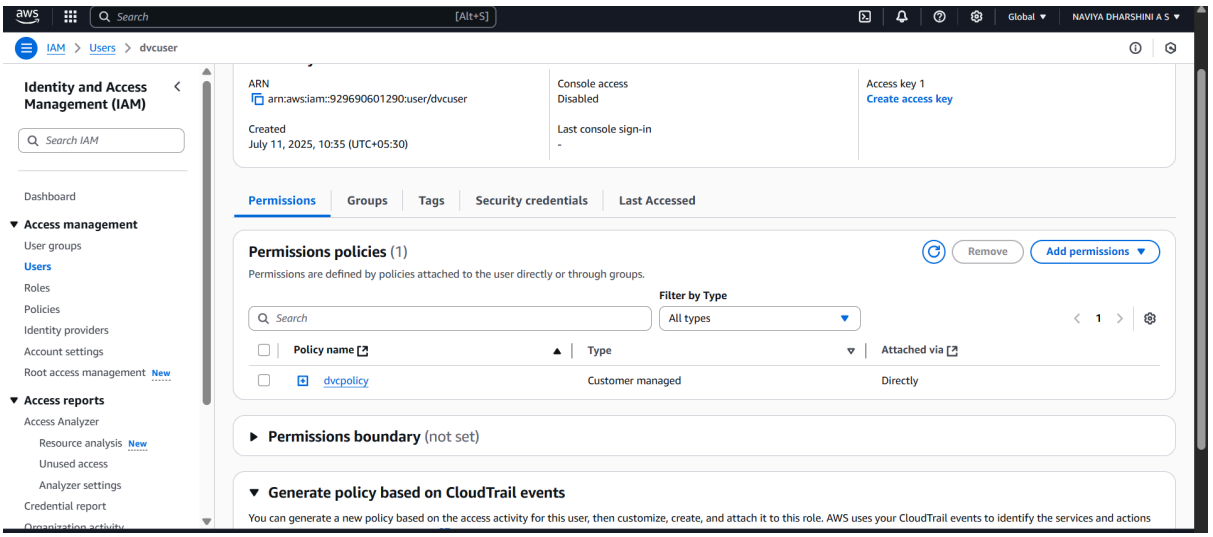


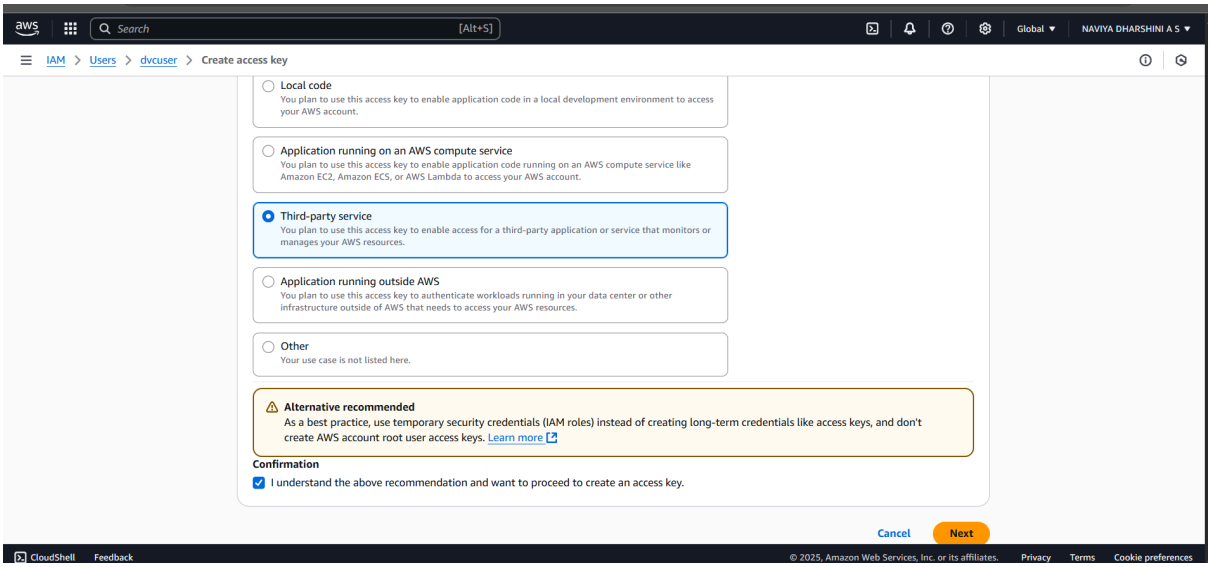


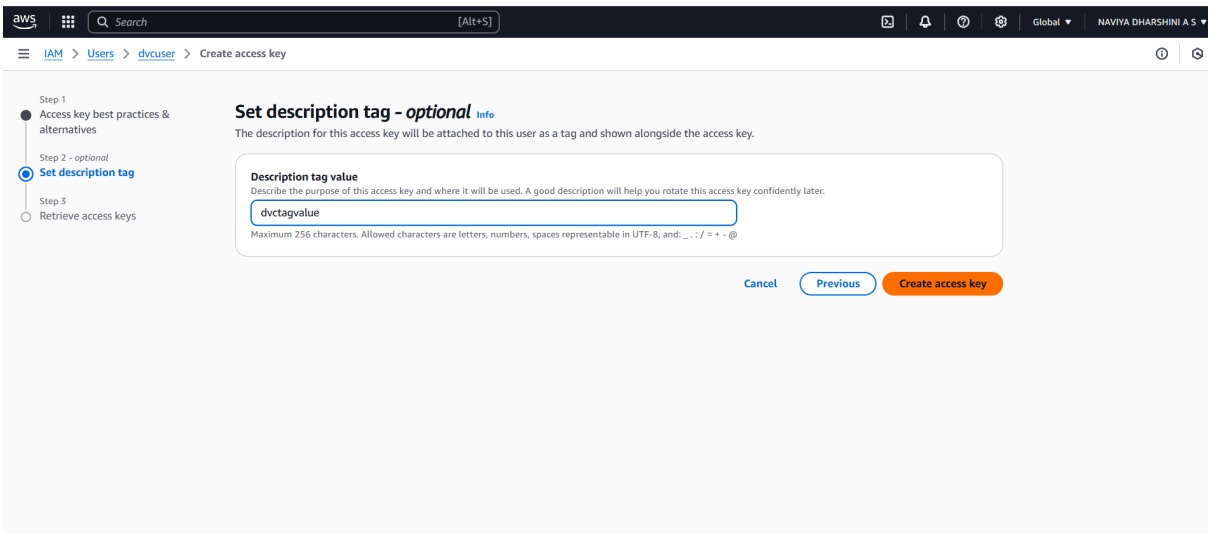

aws
Search
[Alt+S]
Global
NAVIYA DHARSHINI A S
IAM > Users > Create user
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Specify user details
User details
User name
dvcuser
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)
Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more
Cancel
Next

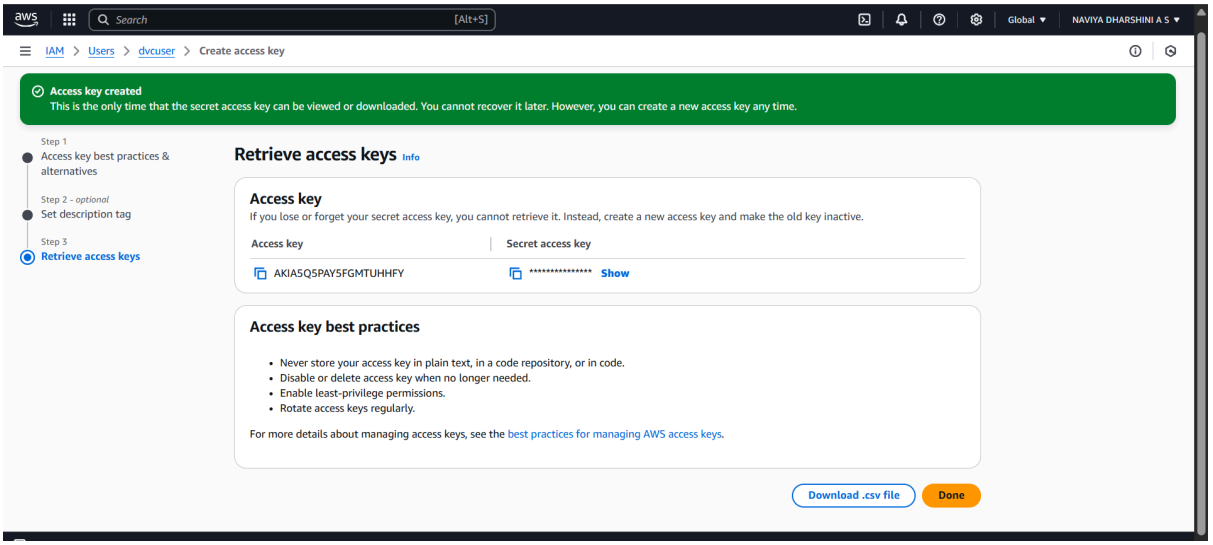

aws
Search
[Alt+S]
Global
NAVIYA DHARSHINI A S
IAM > Users > Create user
Step 2
Set permissions
Step 3
Review and create
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
Permissions policies (1/1379)
Choose one or more policies to attach to your new user.
Create policy
Filter by Type
dvc
All types
1 match
Policy name
Type
Attached entities
dvcpolicy
Customer managed
0
Set permissions boundary - optional
Cancel
Previous
Next


aws
Search
[Alt+S]
Global
NAVIYA DHARSHINI A S
IAM > Users > Create user
Step 2
Set permissions
Step 3
Review and create
Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.
User details
User name
dvcuser
Console password type
None
Require password reset
No
Permissions summary
Name
Type
Used as
dvcpolicy
Customer managed
Permissions policy
Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Previous
Create user
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
Global
NAVIYA DHARSHINI A S
IAM > Users > dvcuser
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Access reports
Access Analyzer
Resource analysis New
Unused access
Analyzer settings
Credential report
ARN
arn:aws:iam::929690601290:user/dvcuser
Created
July 11, 2025, 10:35 (UTC+05:30)
Console access
Disabled
Last console sign-in
-
Access key 1
Create access key
Permissions
Groups
Tags
Security credentials
Last Accessed
Permissions policies (1)
Permissions are defined by policies attached to the user directly or through groups.
Remove
Add permissions
Filter by Type
Search
All types
Policy name
Type
Attached via
dvcpolicy
Customer managed
Directly
Permissions boundary (not set)
Generate policy based on CloudTrail events
You can generate a new policy based on the access activity for this user, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions


Search [Alt+S]
Global
NAVIYA DHARSHINI A S
IAM > Users > dvcuser > Create access key
Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.
Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
Other
Your use case is not listed here.
Alternative recommended
As a best practice, use temporary security credentials (IAM roles) instead of creating long-term credentials like access keys, and don't create AWS account root user access keys. Learn more
Confirmation
I understand the above recommendation and want to proceed to create an access key.
Cancel
Next
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
Global
NAVIYA DHARSHINI A S
IAM > Users > dvcuser > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Set description tag - optional Info
The description for this access key will be attached to this user as a tag and shown alongside the access key.
Description tag value
Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.
dvctagvalue
Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @
Cancel
Previous
Create access key



🌐 . DVC Remotes (Cloud Backends)

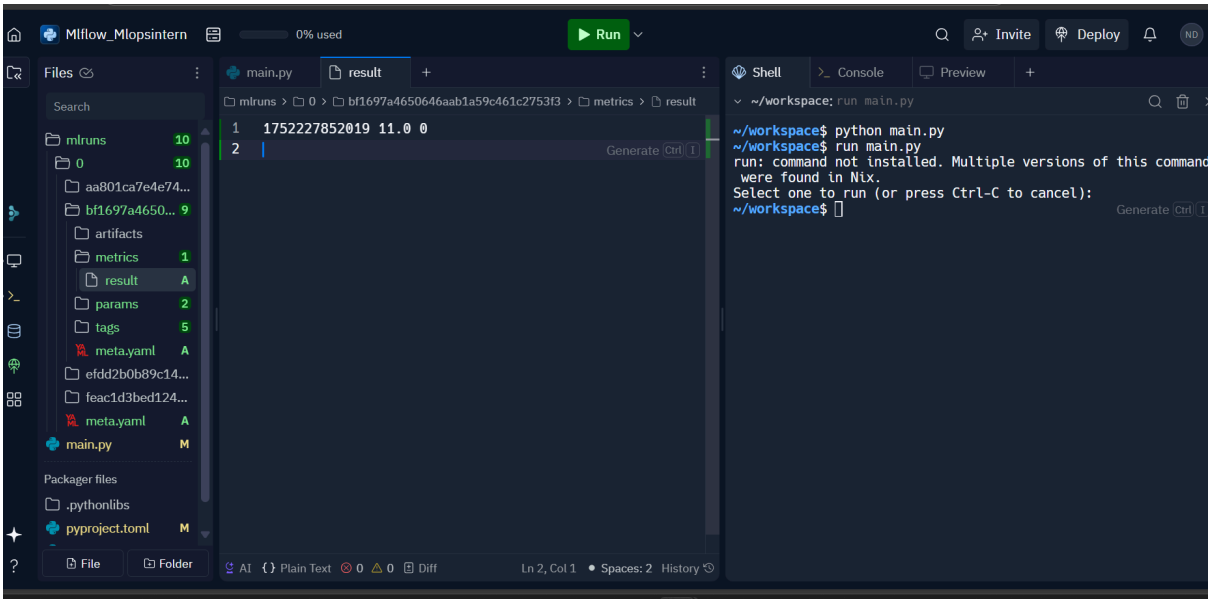
Supports:
✅ AWS S3
✅ Google Drive
✅ Azure Blob
✅ GCP Bucket
✅ SSH, HDFS, WebDAV

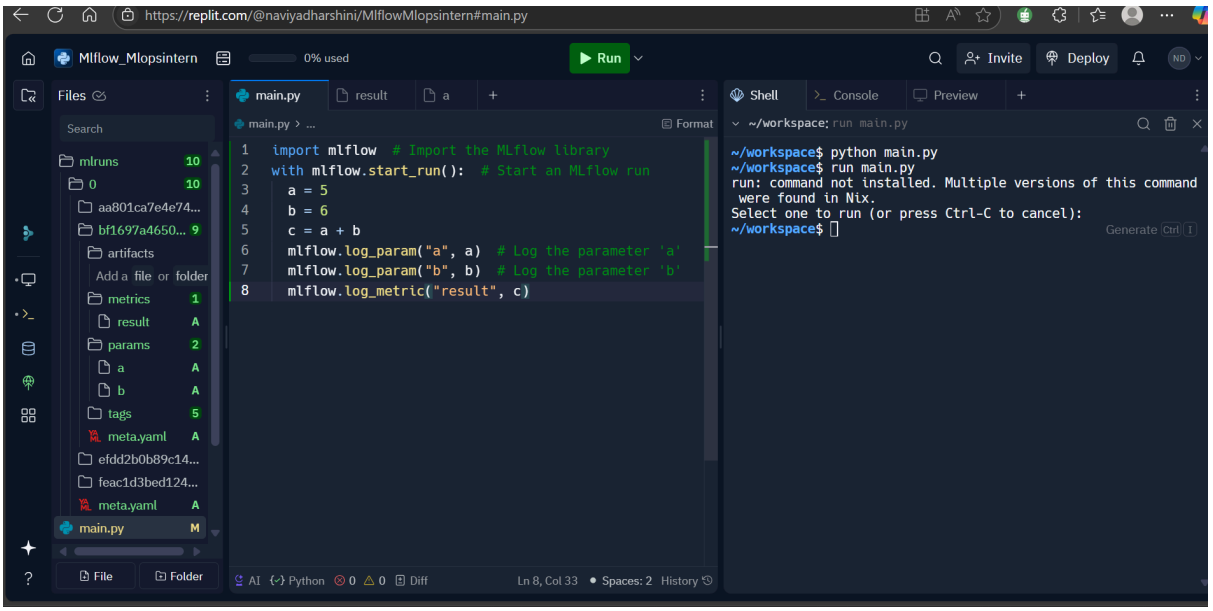
bash
CopyEdit

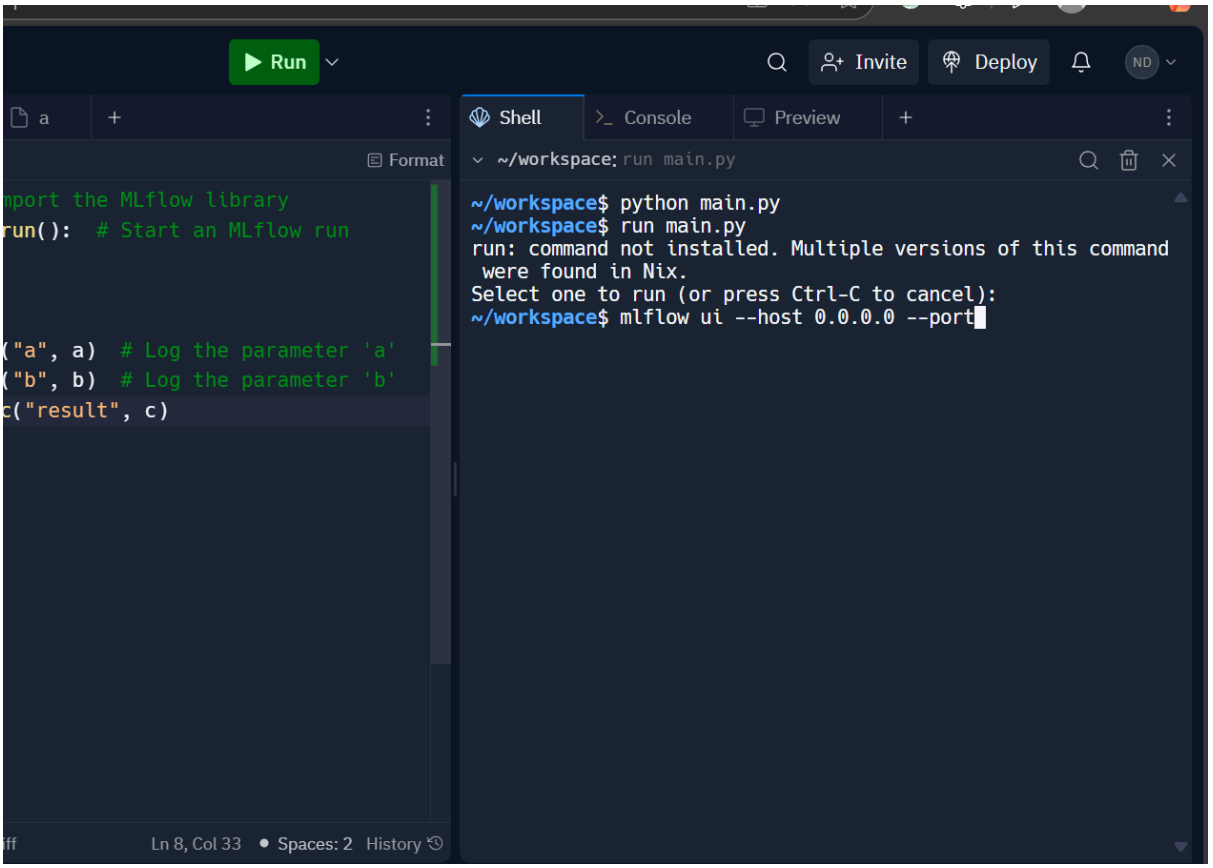
```
dvc remote add -d storage s3://mybucket
dvc remote modify storage access_key_id ...
dvc push
```

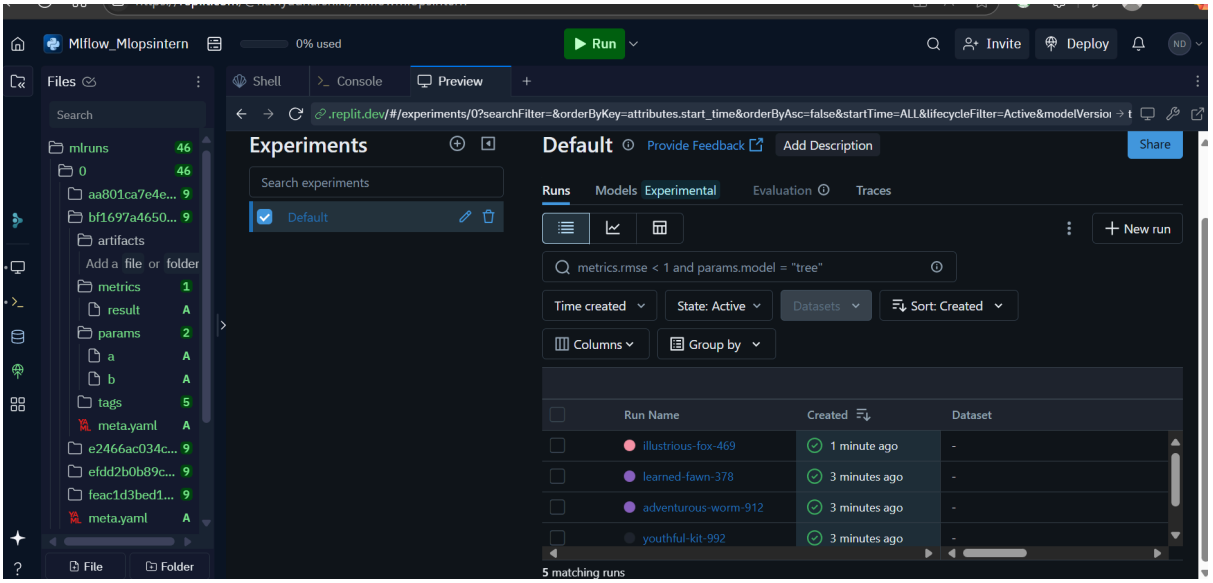
MLflow is a platform to manage the **machine learning lifecycle**. It helps **track, compare, and reproduce** ML experiments efficiently. Here's how the **end-to-end flow** works, from start to finish:

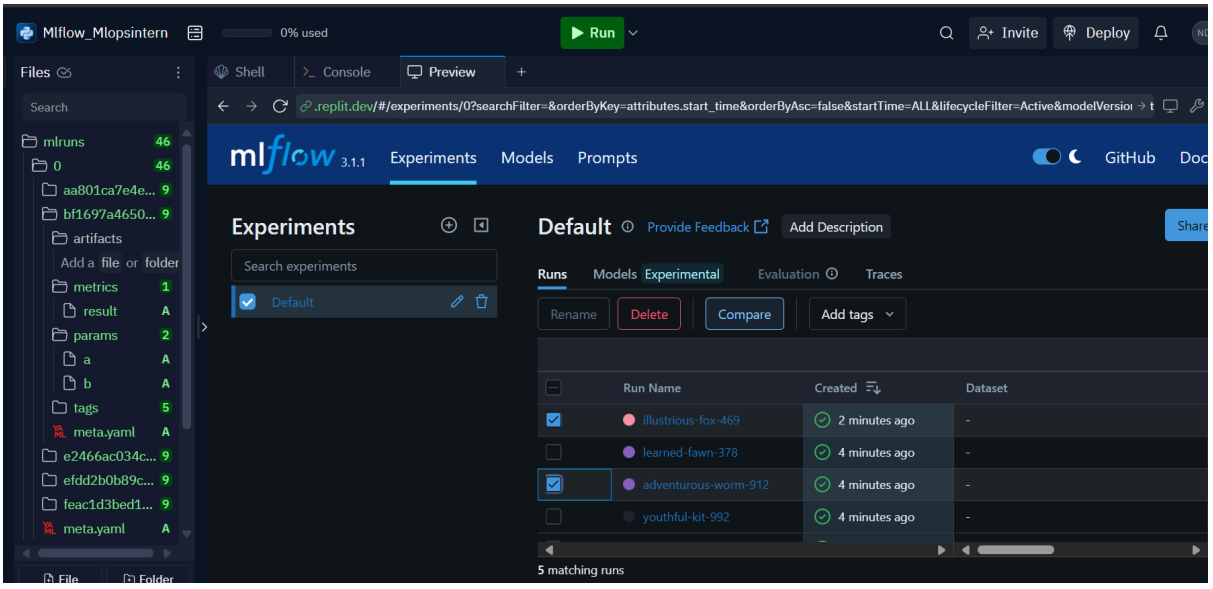
USING REPLIT :

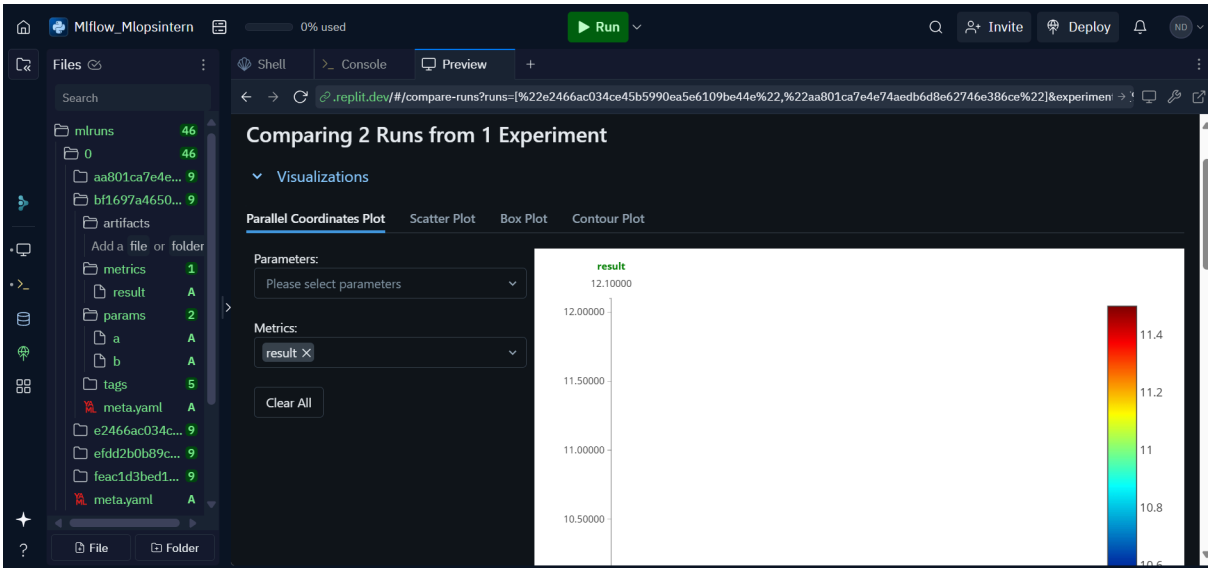

Mlflow_Mlopsintern
0% used
Run
Invite
Deploy
Files
Search
mlruns 10
0 10
aa801ca7e4e74...
bf1697a4650... 9
artifacts
metrics 1
result A
params 2
tags 5
meta.yaml A
efdd2b0b89c14...
feac1d3bed124...
meta.yaml A
main.py M
Packager files
.pythonlibs
pyproject.toml M
File
Folder
main.py
result
mlruns > 0 > bf1697a4650646aab1a59c461c2753f3 > metrics > result
1 1752227852019 11.0 0
2
Generate Ctrl I
AI { } Plain Text 0 0 Diff
Ln 2, Col 1 Spaces: 2 History
Shell
Console
Preview
~/workspace: run main.py
~/workspace$ python main.py
~/workspace$ run main.py
run: command not installed. Multiple versions of this command
were found in Nix.
Select one to run (or press Ctrl-C to cancel):
~/workspace$
Generate Ctrl I

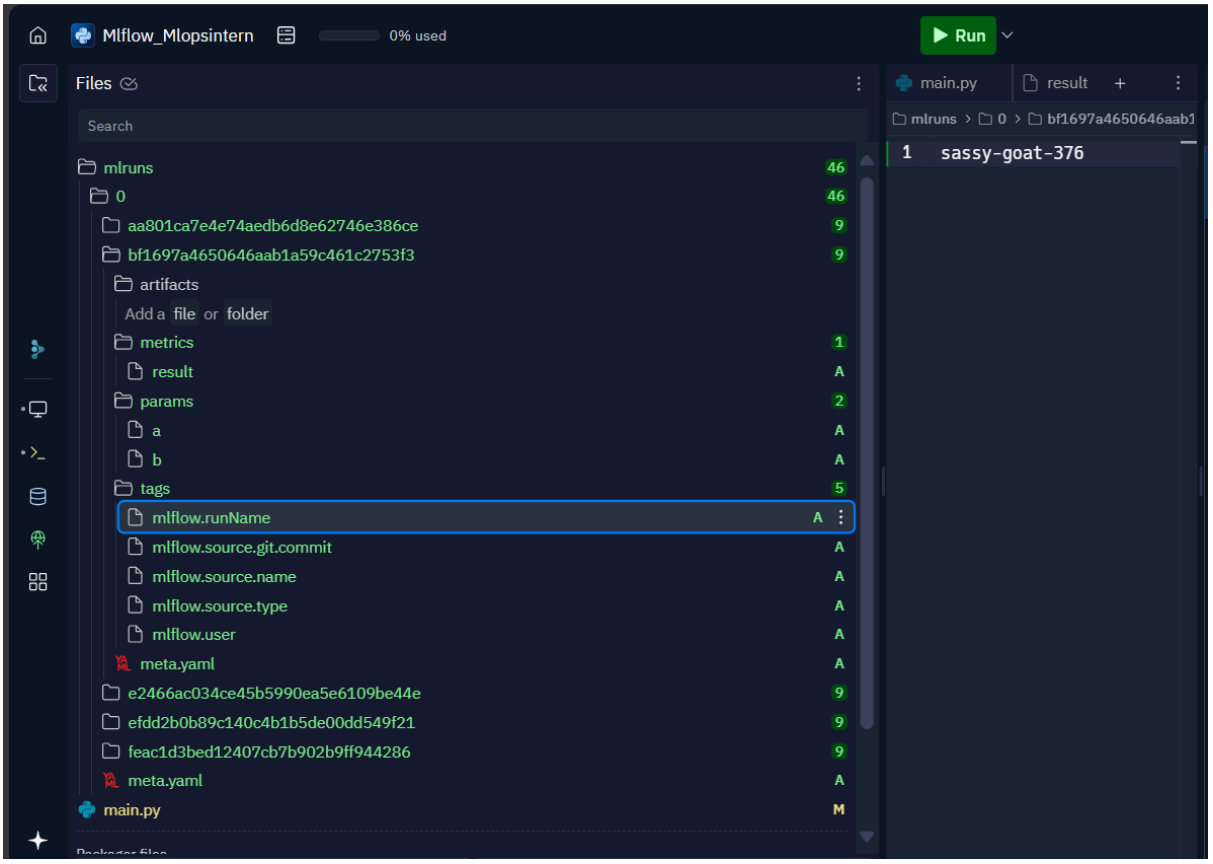

https://replit.com/@naviyadharshini/MlflowMlopsintern#main.py
Mlflow_Mlopsintern
0% used
Run
Invite
Deploy
Files
Search
mlruns 10
0 10
aa801ca7e4e74...
bf1697a4650... 9
artifacts
Add a file or folder
metrics 1
result A
params 2
a A
b A
tags 5
meta.yaml A
efdd2b0b89c14...
feac1d3bed124...
meta.yaml A
main.py M
File
Folder
main.py
result
a
main.py > ...
Format
1 import mlflow # Import the MLflow library
2 with mlflow.start_run(): # Start an MLflow run
3 a = 5
4 b = 6
5 c = a + b
6 mlflow.log_param("a", a) # Log the parameter 'a'
7 mlflow.log_param("b", b) # Log the parameter 'b'
8 mlflow.log_metric("result", c)
AI
Python
0
0
Diff
Ln 8, Col 33
Spaces: 2
History
Shell
Console
Preview
~/workspace: run main.py
~/workspace$ python main.py
~/workspace$ run main.py
run: command not installed. Multiple versions of this command
were found in Nix.
Select one to run (or press Ctrl-C to cancel):
~/workspace$
Generate Ctrl I


Run
Invite
Deploy
ND
a
Format
mport the MLflow library
run(): # Start an MLflow run
("a", a) # Log the parameter 'a'
("b", b) # Log the parameter 'b'
c("result", c)
Shell
Console
Preview
~/workspace: run main.py
~/workspace$ python main.py
~/workspace$ run main.py
run: command not installed. Multiple versions of this command
were found in Nix.
Select one to run (or press Ctrl-C to cancel):
~/workspace$ mlflow ui --host 0.0.0.0 --port
iff
Ln 8, Col 33
Spaces: 2
History


Mlflow_Mlopsintern
0% used
Run
Invite
Deploy
Files
Shell
Console
Preview
Experiments
Search experiments
Default
Default
Provide Feedback
Add Description
Share
Runs
Models
Experimental
Evaluation
Traces
New run
metrics.rmse < 1 and params.model = "tree"
Time created
State: Active
Datasets
Sort: Created
Columns
Group by
Run Name
Created
Dataset
illustrious-fox-469
1 minute ago
learned-fawn-378
3 minutes ago
adventurous-worm-912
3 minutes ago
youthful-kit-992
3 minutes ago
5 matching runs


Mlflow_Mlopsintern
0% used
Run
Invite
Deploy
Files
Shell
Console
Preview
mlflow 3.1.1
Experiments
Models
Prompts
GitHub
Experiments
Search experiments
Default
Default
Provide Feedback
Add Description
Runs
Models
Experimental
Evaluation
Traces
Rename
Delete
Compare
Add tags
Run Name
Created
Dataset
illustrious-fox-469
2 minutes ago
learned-fawn-378
4 minutes ago
adventurous-worm-912
4 minutes ago
youthful-kit-992
4 minutes ago
5 matching runs


Mlflow_Mlopsintern
0% used
Run
Invite
Deploy
Files
Shell
Console
Preview
Comparing 2 Runs from 1 Experiment
Visualizations
Parallel Coordinates Plot
Scatter Plot
Box Plot
Contour Plot
Parameters:
Please select parameters
Metrics:
result
Clear All
result
12.10000
12.00000
11.50000
11.00000
10.50000
11.4
11.2
11
10.8



1️⃣ Experiment

🔹 Think of an **Experiment** as a **folder** or **container** for all ML model tests.

🔸 It groups multiple **runs** under one goal — like testing different algorithms or hyperparameters to solve a problem (e.g., predicting student results or sales forecast).

📝 Example:

> “Student Grade Prediction” can be the experiment name.

2️⃣ Run

🔹 A **Run** is a **single trial or training session** within an experiment.

🔸 Every time you train a model (with different inputs or settings), it’s recorded as one run.

📝 Purpose: Track the details of that attempt — like which model was used, how it performed, etc.

> Example: Trying a Random Forest with 100 trees = 1 run
> Changing it to 200 trees = another run

3️⃣ Parameters

🔹 These are the **inputs** or **settings** used in each run. Think of them like the ingredients in a recipe.

🔸 They include things like learning rate, number of estimators, batch size, etc.

📝 Benefit: You can compare which parameters gave the best results.

> Example: `n_estimators=100`, `max_depth=10`

4️⃣ Metrics

🔹 These are the **outcomes or performance scores** measured during training/testing.

🔸 Includes metrics like accuracy, F1 score, precision, loss, etc.

📝 Goal: Helps decide which run/model performs best.

> Example: `accuracy = 92%`, `F1_score = 0.89`

5️⃣ Artifacts

🔹 Artifacts are **output files** or **saved results** generated during the run.

🔸 Includes:

- Trained models (pickle files, .pkl)
- Plots or graphs
- Logs
- Preprocessed data

📝 These can be reused for deployment or further testing.

Example: Trained model file → `student_model.pkl`

🎯 Summary Flow:

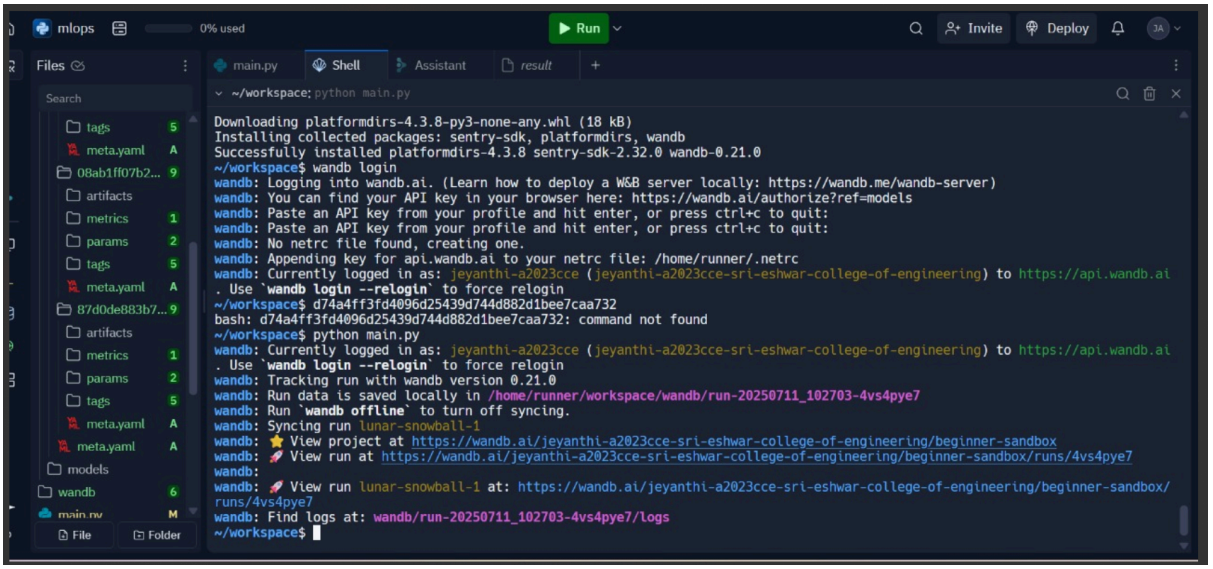
text
CopyEdit

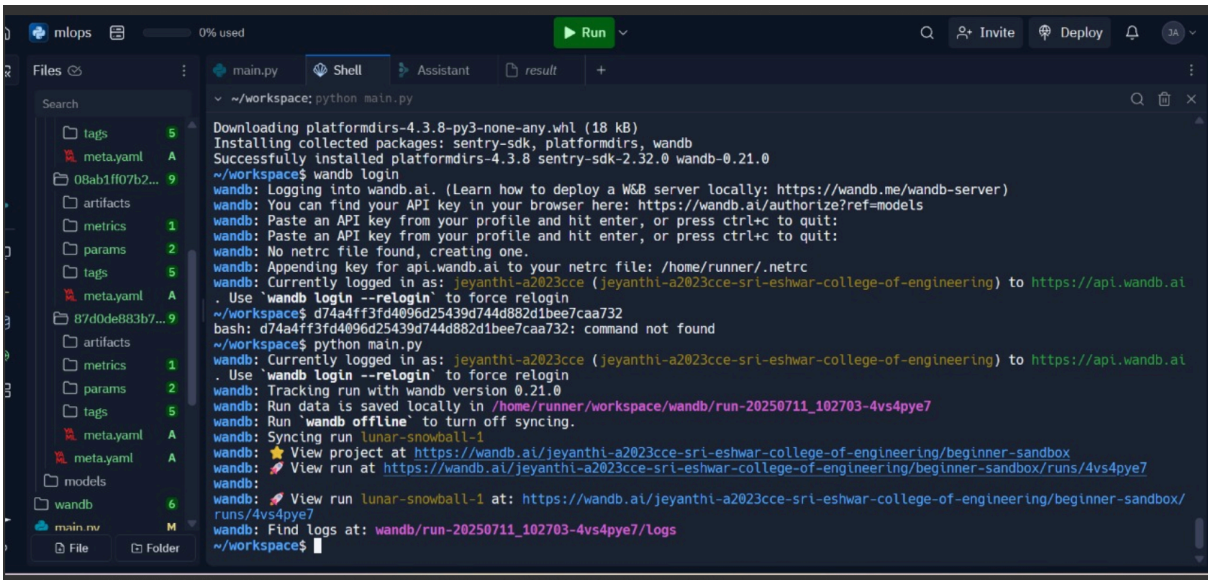
```
Experiment
   └── Run #1
   |     ├── Parameters → (e.g. learning_rate=0.01)
   |     ├── Metrics    → (e.g. accuracy=92%)
   |     └── Artifacts  → (e.g. saved_model.pkl)
   └── Run #2
         ├── Parameters → ...
         ├── Metrics    → ...
         └── Artifacts  → ...
```

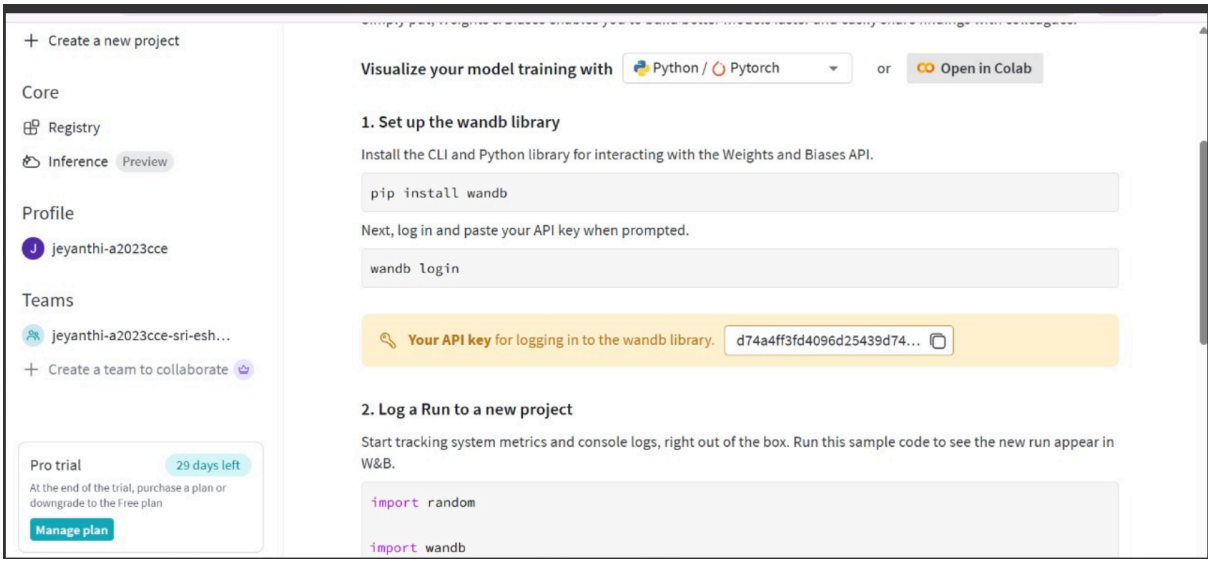
✅ Benefits for / Business Teams:

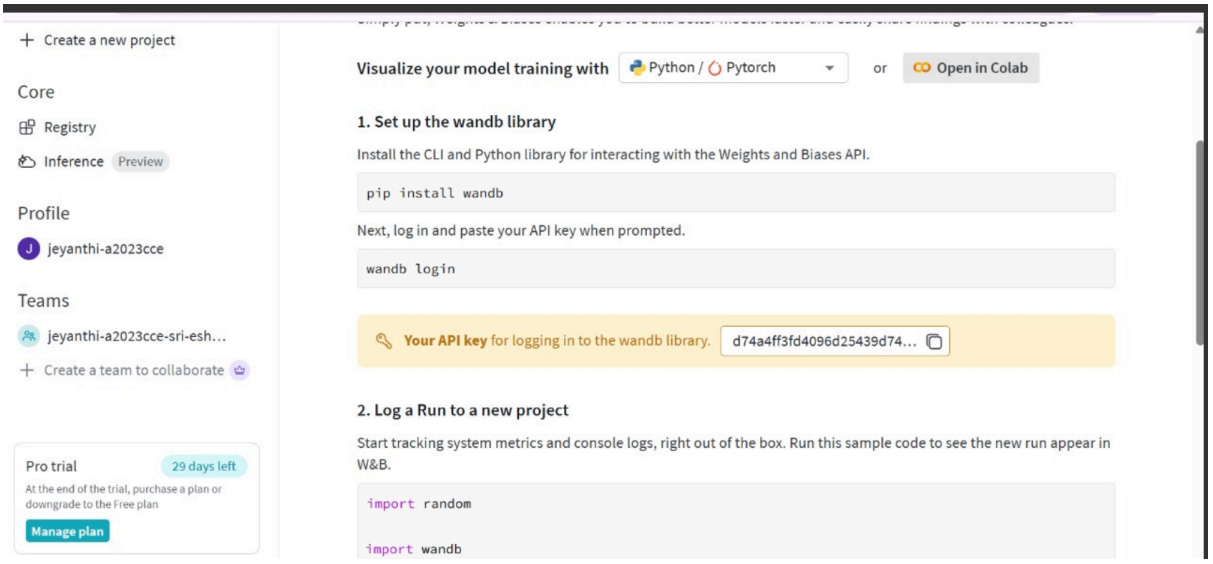
Feature	Business Value
Experiment tracking	Clear visibility of model improvements
Run comparison	Helps choose the best-performing model
Parameter & metric logging	Scientific, reproducible ML approach
Artifact storage	Easy deployment and audit-ready models

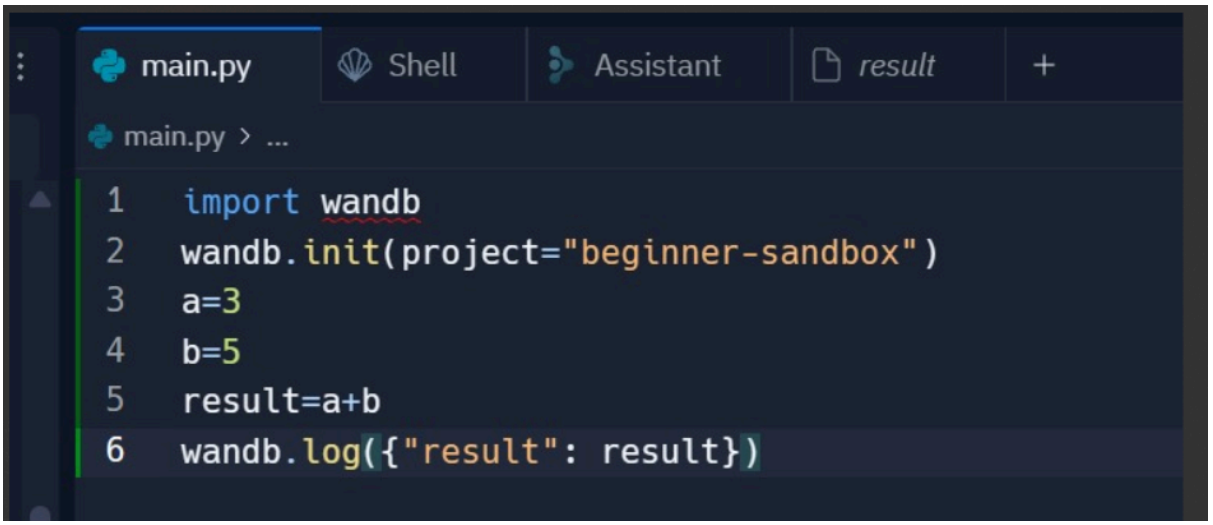
WEIGHTS AND BIASES

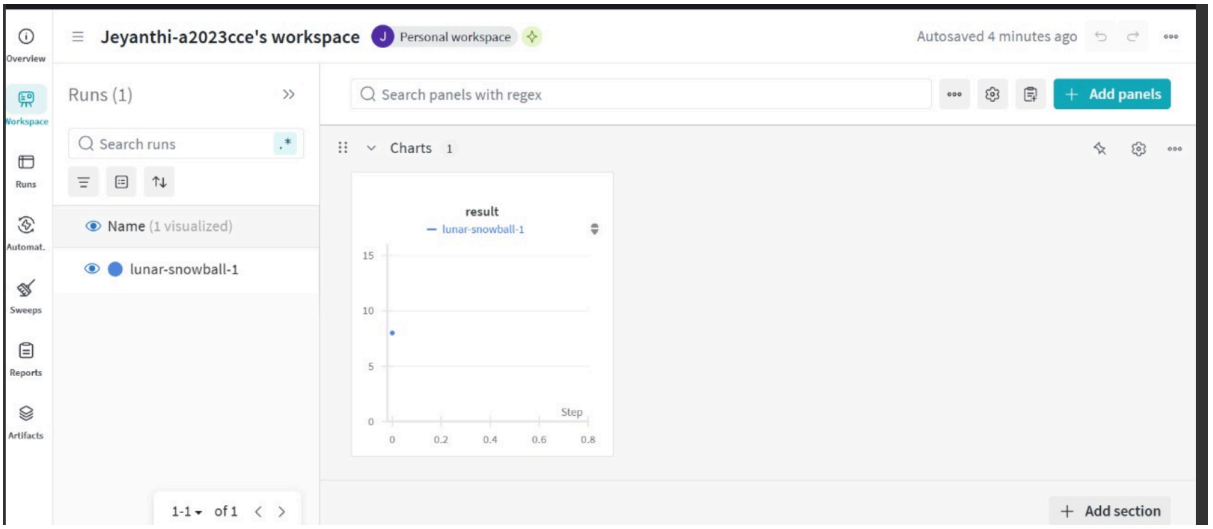





+ Create a new project
Core
Registry
Inference Preview
Profile
jeyanthi-a2023cce
Teams
jeyanthi-a2023cce-sri-esh...
+ Create a team to collaborate
Pro trial 29 days left
At the end of the trial, purchase a plan or downgrade to the Free plan
Manage plan
Visualize your model training with Python / Pytorch or Open in Colab
1. Set up the wandb library
Install the CLI and Python library for interacting with the Weights and Biases API.
pip install wandb
Next, log in and paste your API key when prompted.
wandb login
Your API key for logging in to the wandb library. d74a4ff3fd4096d25439d74...
2. Log a Run to a new project
Start tracking system metrics and console logs, right out of the box. Run this sample code to see the new run appear in W&B.
import random
import wandb


+ Create a new project
Core
Registry
Inference Preview
Profile
jeyanthi-a2023cce
Teams
jeyanthi-a2023cce-sri-esh...
+ Create a team to collaborate
Pro trial 29 days left
At the end of the trial, purchase a plan or downgrade to the Free plan
Manage plan
Visualize your model training with Python / Pytorch or Open in Colab
1. Set up the wandb library
Install the CLI and Python library for interacting with the Weights and Biases API.
pip install wandb
Next, log in and paste your API key when prompted.
wandb login
Your API key for logging in to the wandb library. d74a4ff3fd4096d25439d74...
2. Log a Run to a new project
Start tracking system metrics and console logs, right out of the box. Run this sample code to see the new run appear in W&B.
import random
import wandb


main.py
Shell
Assistant
result
+
main.py > ...
1 import wandb
2 wandb.init(project="beginner-sandbox")
3 a=3
4 b=5
5 result=a+b
6 wandb.log({"result": result})


Jeyanthi-a2023cce's workspace
Personal workspace
Autosaved 4 minutes ago
Overview
Workspace
Runs
Automat.
Sweeps
Reports
Artifacts
Runs (1)
Search runs
Name (1 visualized)
lunar-snowball-1
1-1 of 1
Search panels with regex
Add panels
Charts 1
result
lunar-snowball-1
15
10
5
0
Step
0
0.2
0.4
0.6
0.8
Add section