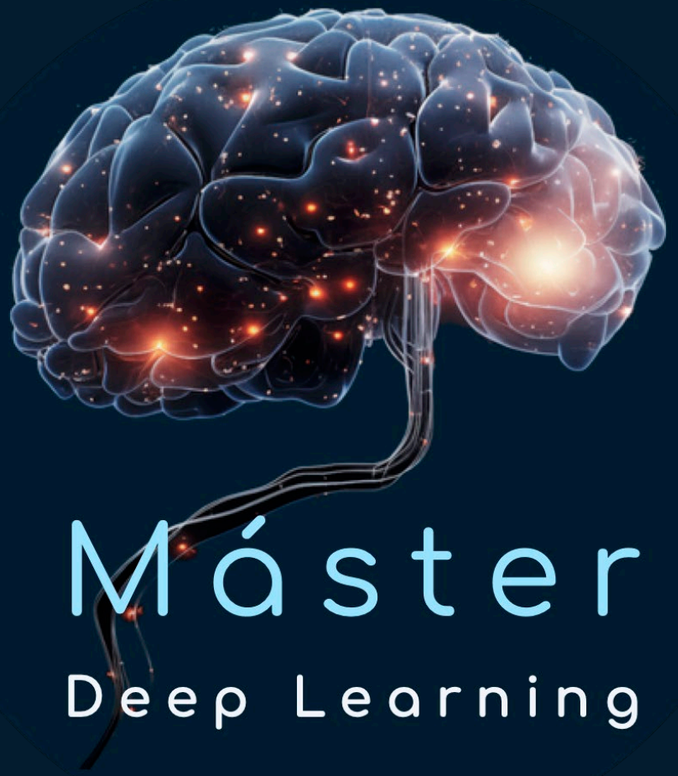
MLOps

Part 6.

W&B Artifacts | Track and version your ML pipeline

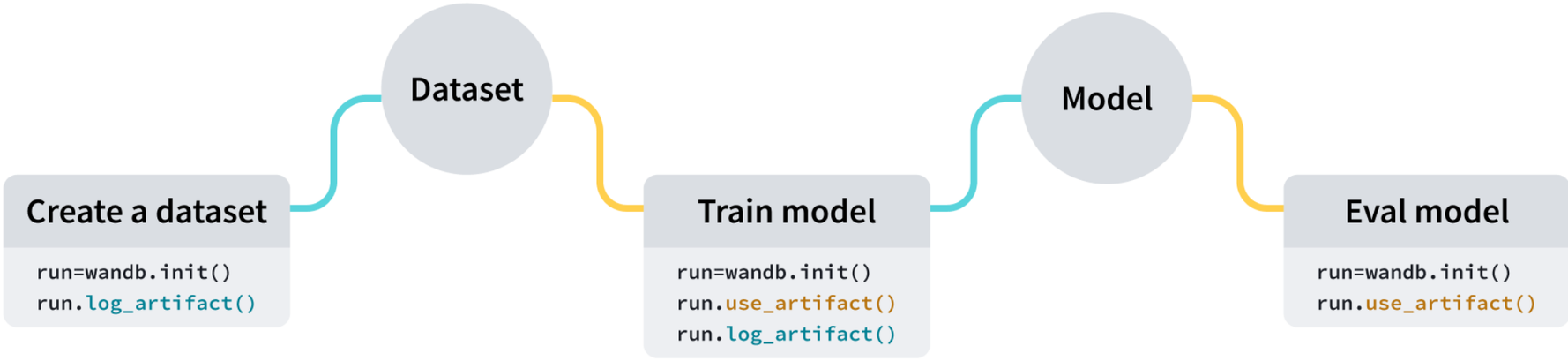


Máster
Deep Learning

Wandb artifacts

Máster
Deep Learning

- What: A system to track and version datasets, models, and other files as inputs and outputs of your W&B Runs.
- Why: Ensure reproducibility, understand lineage, manage datasets/models effectively, and collaborate better.
- Core Idea: Link data and code across your entire ML workflow.



https://docs.wandb.ai/guides/artifacts/

Key Use Cases

Máster
Deep Learning

1

Model Training

Input: Dataset Artifact -> Output: Trained Model Artifact

2

Data Preprocessing

Input: Raw Data Artifact -> Output: Processed Data Artifact

3

Model Evaluation

Input: Model + Test Data Artifacts -> Output: Evaluation Results (e.g., W&B Table)

4

Model Optimization

Input: Model Artifact -> Output: Optimized Model Artifact

- Track any file or directory relevant to your pipeline.

Creating an Artifact (The Basics)

Máster
Deep Learning

Initialize Run

run = wandb.init(project="...", job_type="...")

Create Artifact Object

artifact = wandb.Artifact(name="my-dataset", type="dataset")

- name: Unique identifier (e.g., "raw-images", "bert-base-uncased")
- type: Logical step in pipeline (e.g., "dataset", "model", "evaluation")

Add Files/Data

artifact.add_file("local/path/to/data.csv")

Log Artifact

run.log_artifact(artifact)

Adding content to artifacts

Máster
Deep Learning

Single File

artifact.add_file(local_path="model.h5", name="optional/path/in/artifact/model.h5")

Directory

artifact.add_dir(local_path="image_folder/", name="optional/prefix")
(Recursively adds contents)

External References

artifact.add_reference("s3://my-bucket/data/images")

- Track data without uploading it to W&B.
- Supports s3://, gs://, http(s)://, file://.

Using and downloading artifacts

Máster
Deep Learning

Declare Usage in Run

1

artifact = run.use_artifact("my-dataset:latest")

- Links artifact as input to the current run.
- Use aliases (latest, v0, best) or specific versions.
- Can reference artifacts from other projects/entities: "entity/project/artifact:alias"

Download Contents

2

data_dir = artifact.download()

- Downloads to ./artifacts/artifact-name:version/.
- Optional root parameter for custom path.
- Partial download: artifact.download(path_prefix="images/cats/")
- Get specific file path/ref: path_obj = artifact.get_path("file.txt") -> path_obj.download()

Versioning & Aliases

Máster
Deep Learning

Automatic Versioning

W&B automatically creates new versions (v0, v1, v2...) if content changes upon log_artifact.

Aliases

Human-readable pointers to specific versions.

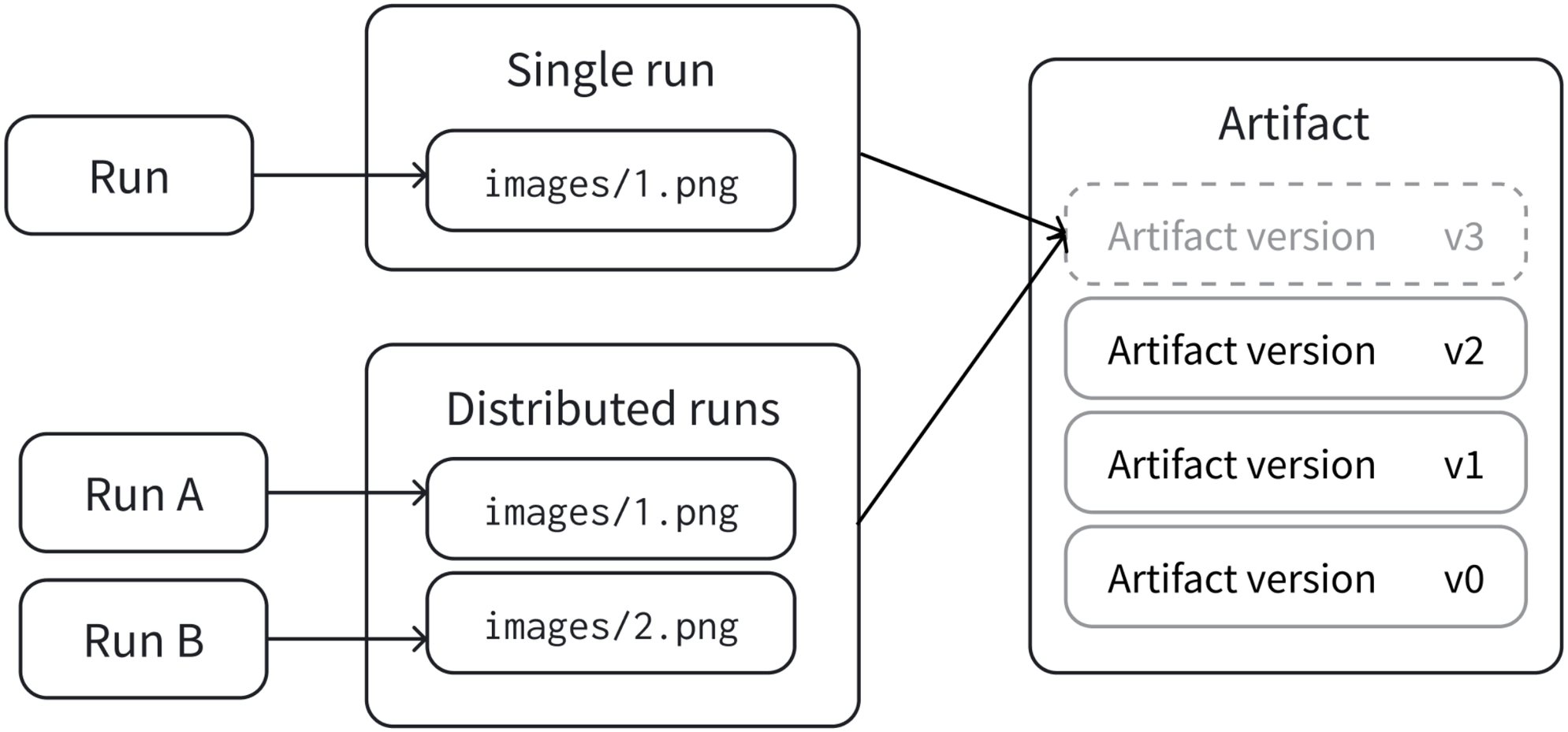
- latest: Automatically added/updated on log_artifact.
- vN: Permanent alias for each version.

Custom Aliases

Add meaningful tags like best-accuracy, production, staging.

run.log_artifact(artifact, aliases=["latest", "best-so-far"])

Update via API:
artifact.aliases.append("new-alias");
artifact.save()



https://docs.wandb.ai/guides/artifacts/create-a-new-artifact-version/

Advanced versioning: Incremental & Distributed

Máster
Deep Learning

Incremental Artifacts

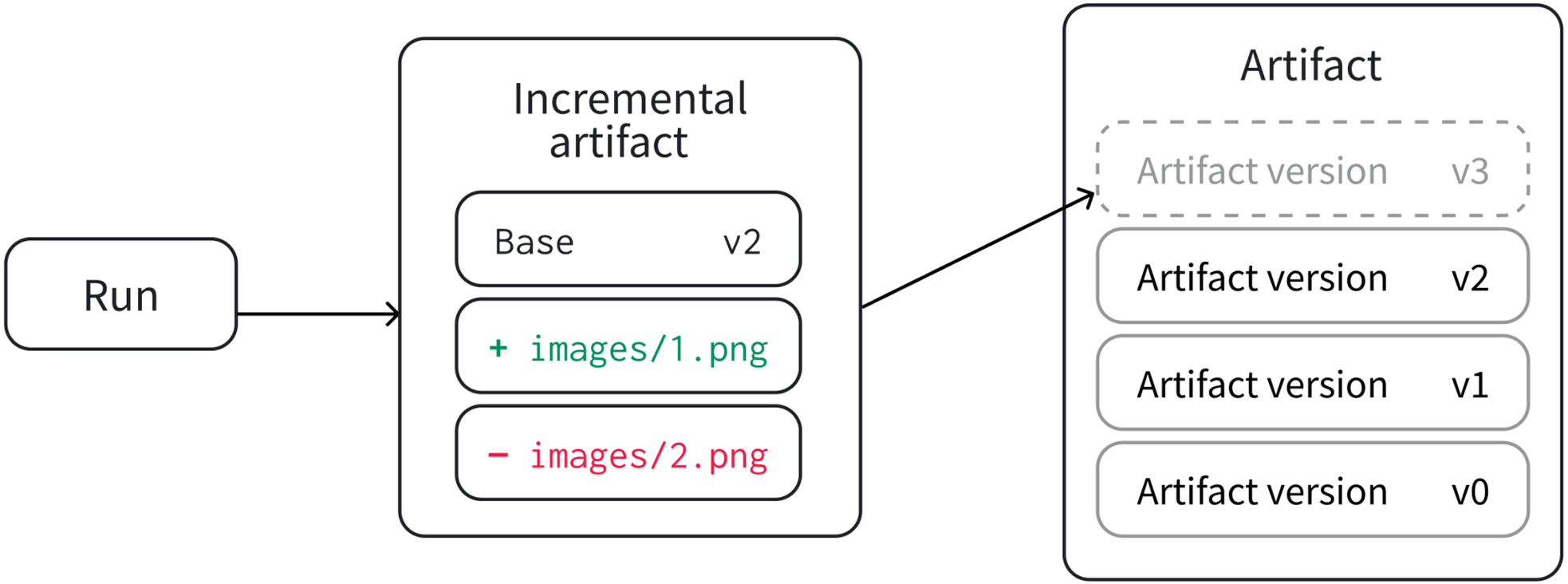
Modify existing versions without re-uploading everything.

- saved_artifact = run.use_artifact(...)
- draft = saved_artifact.new_draft()
- draft.add_file(...) / draft.remove(...)
- run.log_artifact(draft)

Distributed Artifacts

Multiple parallel runs contribute to one artifact version.

- Runs use run.upsert_artifact(artifact, distributed_id="...")
- A final run calls run.finish_artifact(artifact, distributed_id="...") to commit.



https://docs.wandb.ai/guides/artifacts/create-a-new-artifact-version/

Tracking external files (references)

Máster
Deep Learning

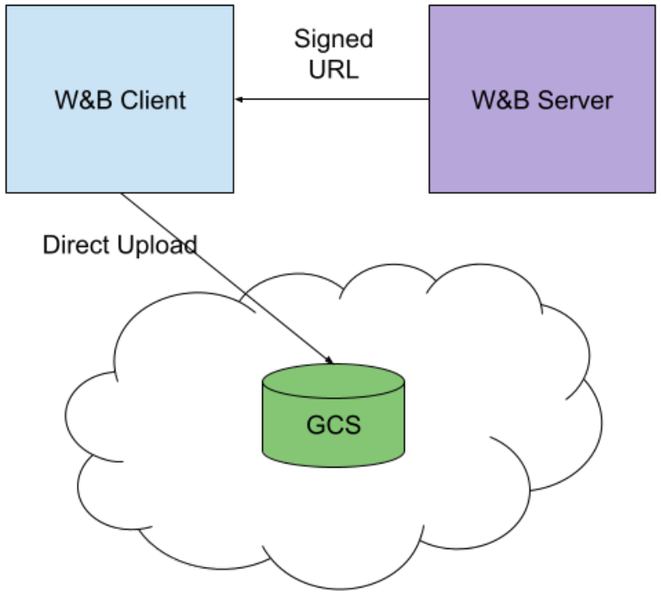
Problem

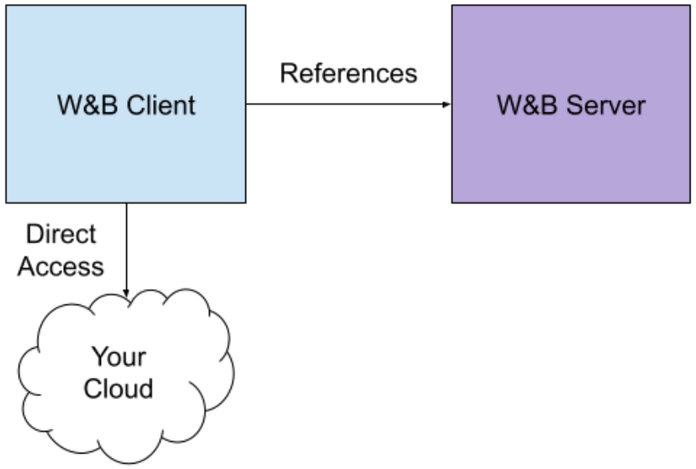
Large datasets or sensitive data shouldn't always be uploaded to W&B storage.

Solution

Reference Artifacts (artifact.add_reference(...))

- Store only metadata (URI, checksums, size, version ID) in W&B.
- Data remains in your S3, GCS, Azure Blob, HTTP server, or NFS mount.
- download() fetches directly from the source URI.
- Enables versioning and lineage tracking for external data.
- Requires appropriate credentials configured for W&B to access the source.


W&B Client
Signed
URL
W&B Server
Direct Upload
GCS


W&B Client
References
W&B Server
Direct
Access
Your
Cloud

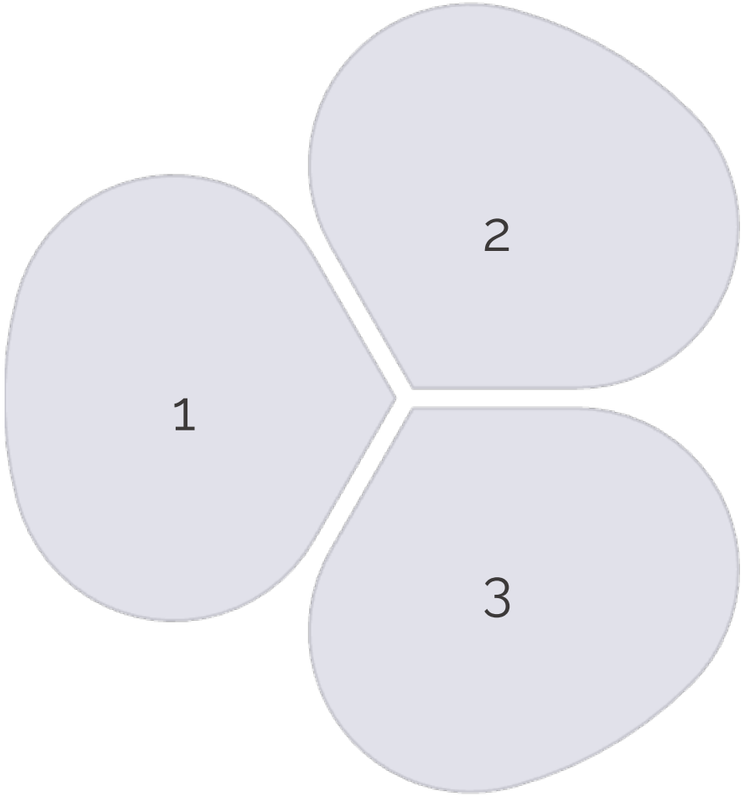
Lineage Graph: Visualizing your pipeline

Máster
Deep Learning

Automatic DAG

W&B automatically builds a Directed Acyclic Graph (DAG) connecting runs and artifacts.

View in UI: Project -> Artifacts -> Select Artifact -> Lineage Tab.



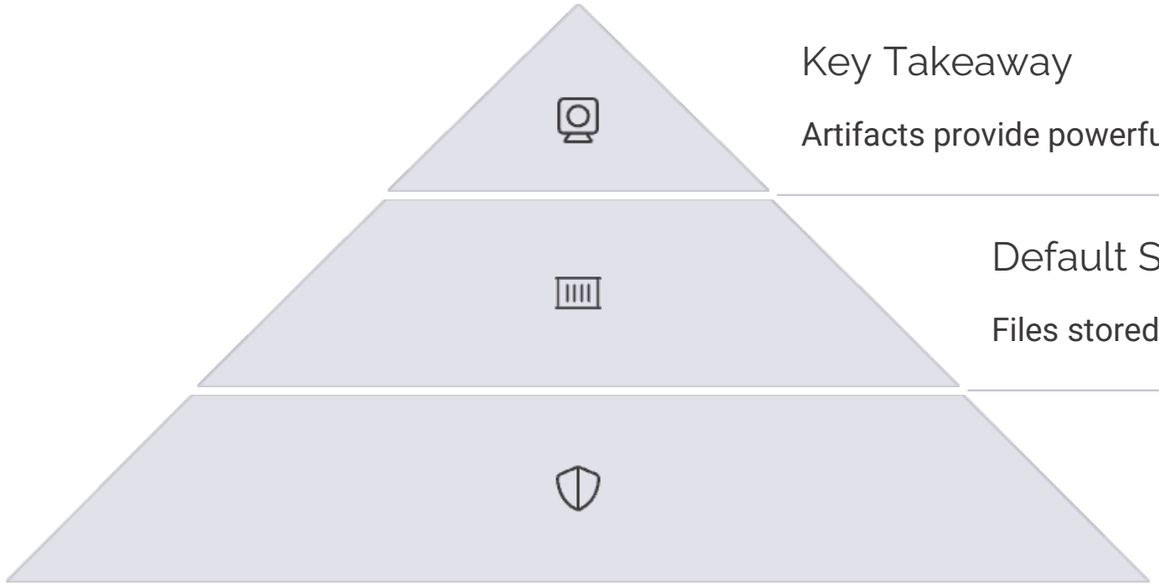
Benefits

- Understand data flow and dependencies.
- Trace model origins and dataset versions.
- Ensure reproducibility and auditability.
- Debug pipelines.

API Access

artifact.logged_by() (producer run), artifact.used_by() (consumer runs).

Data privacy &Summary



Key Takeaway

Artifacts provide powerful versioning, tracking, and lineage for ML workflows

Default Storage

Files stored securely in W&B's encrypted GCS buckets

Privacy Options

Reference Artifacts or Self-Hosted/Dedicated Cloud

- Default Storage: Artifact files uploaded via add_file/add_dir are stored securely in W&B's encrypted GCS buckets.
- Privacy Options:
    - Reference Artifacts: Keep data in your own storage (S3, GCS, Azure, etc.).
    - W&B Self-Hosted / Dedicated Cloud: Control the entire environment.
- Key Takeaway: Artifacts provide powerful versioning, tracking, and lineage for all components of your ML workflow, enhancing reproducibility and collaboration.