1. What Airflow Is and Its Purpose

- **Airflow = Workflow Orchestrator**
  It's a tool that lets you define, schedule, and monitor workflows (pipelines). Instead of manually running scripts, Airflow automates them and ensures they run in the right order.
- **Function in Data Engineering**
  Airflow is the "traffic controller" of data pipelines. It doesn't do the heavy lifting itself (like transforming data), but it *organizes and triggers* the tasks that do.

2. Webserver and UI

- **Webserver**: When you run `airflow webserver`, you start the **UI** (User Interface).
- **UI Purpose**:
  - View DAGs (your workflows).
  - Trigger DAG runs manually.
  - Pause/unpause DAGs.
  - Monitor task status (success, failed, running, skipped).
- **Access**: Usually at `localhost:8080`. If you're on a VM, port forwarding lets you access it from your local browser.

3. Scheduler

- **Definition**: The scheduler is the Airflow component that *decides when tasks should run*.
- **Role**:
  - Reads DAG definitions from Python files.
  - Checks the metadata database for which tasks are ready.
  - Sends runnable tasks to the executor (the system that actually runs them).
- **UI Message**: "Scheduler doesn't appear to be running" means Airflow can show DAGs but won't execute them until you start the scheduler with `airflow scheduler`.

4. DAGs (Directed Acyclic Graphs)

- **Definition**: A DAG is the *blueprint* of your workflow.
- **Structure**:

- **Nodes = tasks** (Python functions, operators).
- **Edges = dependencies** (order of execution).

- **Properties**:
  - Directed = flows one way.
  - Acyclic = no loops (you can't go back).
- **Run Types**:
  - **Scheduled**: Runs automatically at defined intervals.
  - **Manual/External Trigger**: Runs when you manually start it or another system triggers it.

🔔 5. Trigger Rules

- **Trigger = Condition for running a task**.
- Examples:
  - **all_success**: Run only if all upstream tasks succeeded.
  - **one_failed**: Run if at least one upstream task failed.
  - **none_failed**: Run if no upstream task failed.
- **Status of Trigger**: Shows whether the condition was met (success, failed, skipped).

🗄️ 6. Metadata Database

- **Definition**: Central database where Airflow stores all state.
- **Contents**: DAG definitions, task instances, run history, user roles, logs.
- **Importance**: Without it, Airflow wouldn't know what has run, what failed, or what's scheduled next.

⚙️ 7. Cluster Activity & Metrics

- **Cluster**: Your Airflow environment (local machine or distributed setup).
- **Activity**: Shows if components (scheduler, webserver, workers) are healthy.
- **Metrics**:
  - **Live metrics**: Current status of tasks and jobs.

- **Historical metrics**: Past runs, failures, retries.

👥 8. User Roles and Permissions

- **Roles**: Admin, User, Viewer.
- **Actions**:
  - **can_read**: View DAGs and logs.
  - **can_edit**: Modify DAGs, connections, passwords.
- **Audit logs**: Track user activity (e.g., failed logins).

🛠️ 9. Configuration

- **Airflow.cfg file**: Main configuration file.
- **Not set in UI**: Most settings (database connections, executors, logging) are configured in the file or during installation.

🔌 10. Connections

- Airflow can connect to:
  - **Databases** (Postgres, MySQL, etc.).
  - **Cloud services** (AWS, GCP, Azure).
  - **Local systems** (files, APIs).
- These connections are used by operators/hooks to run tasks.

🐍 11. Running a DAG (Python Concepts)

- **DAG file = Python file**.
- **Operators**: Prebuilt task types (PythonOperator, BashOperator, etc.).
- **Hooks**: Interfaces to external systems (databases, APIs).
- **Retries**: You can set how many times a task retries if it fails.
- **Task IDs**: Each task has a unique ID (t1, t2, etc.).

- **Dependencies**: Defined with `t1 >> t2` (t1 runs before t2).

🚀 Putting It All Together

Airflow is like a **control tower**:

- **Webserver/UI** = dashboard to see flights (DAGs).
- **Scheduler** = decides when planes (tasks) take off.
- **Metadata DB** = records all flight history.
- **Triggers** = rules for when planes can depart.
- **Cluster activity** = health of the whole airport.
- **Operators/Hooks** = actual planes doing the work.