Databricks Workflows Overview

Exploring key features and effective management techniques

MC ID: 7124506 | Parveen KR | Parveen.R@hotmail.com

Presentation Outline

- Introduction to Databricks Workflows
- Understanding Databricks Workflows
- Clusters vs. Jobs Compute
- Creating a Basic Workflow
- Hands-On Lab Instructions
- Scheduling Tasks with the Jobs UI
- Demo – Creating a Scheduled Job
- Hands-On Lab 2 – Scheduled Tasks

Introduction to Databricks Workflows

Deploy Workloads with Databricks Workflows

Introduction to Workflows

This section covers the basics of Databricks Workflows, focusing on task automation and orchestration in data processing.

Jobs Compute

Jobs Compute allows users to manage and run jobs efficiently, optimizing resource utilization in Databricks environments.

Scheduling Tasks

Scheduling tasks through the Jobs UI enables users to automate job execution based on specific triggers and timings.

Understanding Databricks Workflows

Understanding the functionality of Databricks Workflows is crucial for managing complex data pipelines effectively.

Understanding Databricks Workflows

What are Databricks Workflows?

Unified Automation Tool

Databricks Workflows serve as a unified tool for automating data pipelines, analytics, and machine learning workflows.

Chaining Workflows

The platform allows you to easily chain notebooks, Python scripts, and SQL tasks into a cohesive workflow.

Task Management Features

Databricks Workflows support scheduling, retries, dependencies, and parameterization for better task management.

Scalability and Reliability

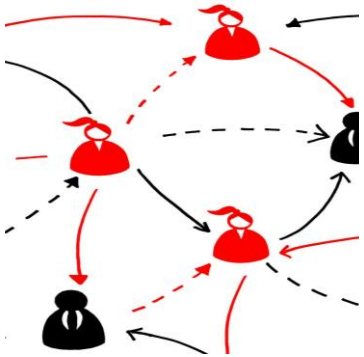
These workflows ensure repeatability, reliability, and scalability of your production data tasks.

Clusters vs. Jobs Compute

Comparison Table

FEATURE	ALL-PURPOSE CLUSTER	JOBS COMPUTE CLUSTER
Usage	Interactive development	Automated/job execution
Cost	Higher (always running)	Lower (auto-start/stop)
Permissions	User-managed	Job owner/role-managed
Termination	Manual	Auto-terminates

When to Use Each?

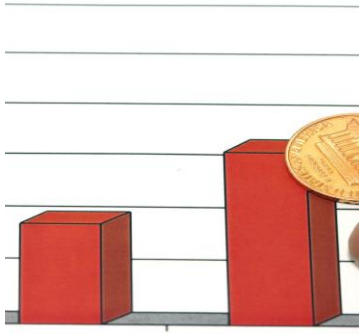


All-purpose Clusters Usage

All-purpose clusters are ideal for data exploration and collaborative development on shared notebooks.

Jobs Compute Usage

Jobs Compute is suitable for scheduled and production jobs, ensuring reliability and efficiency.



Cost and Security Benefits

Choosing the right compute type enhances operational security and reduces overall costs.

Creating a Basic Workflow

Steps to Create a Workflow

Access Workflows Tab

1. Start by navigating to the Databricks ‘Jobs and Pipeline' (Jobs) tab to begin creating your workflow.

Create a Job

2. Click on 'Create Job' to initiate the job setting process within the workflow.

Add a Task

3. Choose a notebook as the task type to add to your workflow for processing.

4. Execute the Job

Finally, click 'Run Now' to execute the job and see your workflow in action.

Hands-On Lab Instructions

Lab 1: Create and Run a Simple Job

Access Databricks Workspace

Begin by opening your Databricks workspace and navigating to the Workflows section to manage jobs.

Create a New Job

Create a new job named Simple-ETL-Job to start your data processing tasks efficiently.

Attach Notebook

Attach a notebook containing sample code or your custom logic to the job for execution.

Run Job and Observe Output

Run the created job and monitor the output in the job's run history for results and logs.

Scheduling Tasks with the Jobs UI

Scheduling Options

Time-Based Scheduling

Jobs UI supports time-based scheduling options like hourly, daily, and custom intervals for flexible task management.

Cron Expressions

Advanced scheduling can be achieved using Cron expressions, allowing for complex scheduling scenarios.

Notifications

Users receive notifications regarding job success or failure, ensuring effective task monitoring.

Parameterized Jobs

Parameterized jobs allow for reusability, enabling users to run similar tasks with different parameters easily.

Steps to Create a Scheduled Job

Open the Job

Start by opening the job that you want to schedule in the application interface.

Access the Schedule Tab

Next, click on the Schedule tab to set up your job's timing and frequency.

Add a Schedule

Specify a schedule, such as every 2 hours, or utilize a cron expression for advanced scheduling.

Save and Monitor

Finally, save the schedule and monitor future runs in the job’s run history to ensure it operates correctly.

Hands-On Lab – Scheduled Tasks

Lab: Schedule Your Job

Scheduling Your Job

Edit your Simple-ETL-Job to add a schedule for automated execution, enhancing efficiency.

Choose Custom Schedule

Select an option to run the job every 2 hours or configure a custom cron schedule suited to your needs.

Job Run History

Confirm and observe upcoming scheduled runs in the job's history to track performance and reliability.

Email Notifications

Optionally configure email notifications to stay updated on the job status and any issues.