Function Flow Overview

Function	Input	Processing	Output
`main()`	None	Orchestrates execution flow	None (prints RCA results)
`generate_synthetic_logs(num_logs)`	`num_logs` (int)	Generates synthetic alarm logs	`logs` (list of dicts)
`save_logs(logs)`	`logs` (list of dicts)	Saves logs to CSV	`file_path` (str)
`load_logs(file_path)`	`file_path` (str)	Reads logs from CSV and converts data types	`converted_logs` (list of dicts)
`preprocess_logs(logs)`	`logs` (list of dicts)	Converts logs into numerical format	`processed_logs` (numpy array)
`cluster_alarms(data)`	`data` (numpy array)	Performs DBSCAN clustering	`clusters` (list of ints)
`root_cause_analysis(logs, clusters)`	`logs` (list of dicts), `clusters` (list of ints)	Identifies most common issues per cluster	`rca_results` (dict)

Step-by-Step Function Flow

1️⃣ Main Execution (main())

- **Calls generate_synthetic_logs() → Creates synthetic logs.**
- **Calls save_logs() → Saves logs to a file.**
- **Calls load_logs() → Reads logs from the saved file.**
- **Calls preprocess_logs() → Converts logs into numerical format.**
- **Calls cluster_alarms() → Applies DBSCAN clustering.**
- **Calls root_cause_analysis() → Identifies root causes based on clusters.**
- **Prints results.**

2️⃣ Log Generation (generate_synthetic_logs())

- **Creates a list of dictionaries, each representing an alarm log with:**
  - **timestamp**
  - **component**
  - **type**
  - **severity**
- **Returns: logs (list of dicts)**

3️⃣ Saving Logs (save_logs())

- **Writes logs to a CSV file.**
- **Returns: file_path (string)**

4️⃣ Loading Logs (load_logs())

- **Reads the CSV file and converts**
  - **severity to integer.**
  - **timestamp to datetime object.**
- **Returns: converted_logs (list of dicts)**

5️⃣ Preprocessing Logs (preprocess_logs())

- **Converts logs into a numeric format for clustering:**
  - **Normalizes timestamps.**
  - **Encodes component and type.**
  - **Extracts severity.**
- **Returns: processed_logs (numpy array)**

6️⃣ Clustering (cluster_alarms())

- **Uses DBSCAN to find clusters based on:**
  - **eps (1000) → Sensitivity.**
  - **min_samples (3) → Minimum points in a cluster.**
- **Returns: clusters (list of ints, where -1 = noise)**

7️⃣ Root Cause Analysis (root_cause_analysis())

- **Groups logs by cluster.**
- **Identifies most common component and most common alarm type.**
- **Computes average severity.**
- **Returns: rca_results (dict)**