In **Azure Sentinel** (now part of **Microsoft Defender for SIEM and XDR**), two foundational components you need to understand are **Resource Group** and **Log Analytics Workspace**.

### 🗂️ ****1. Resource Group****

A **Resource Group** in Azure is a **container** that holds related resources for an application or service.

* You **deploy Sentinel inside a resource group**.
* It includes your **Log Analytics Workspace**, **connectors**, **playbooks**, etc.
* Enables **access control, billing, and lifecycle management** at a group level.

Think of it like a folder to organize and manage your Azure resources.

### 📊 ****2. Log Analytics Workspace****

A **Log Analytics Workspace** is the **data storage and query engine** used by Sentinel.

* **Sentinel sits on top of a Log Analytics Workspace**.
* It stores all your **security event data** (from connectors like firewalls, endpoints, Azure resources).
* Enables **KQL (Kusto Query Language)** for searching and analyzing logs.
* Used for **detections, hunting, dashboards, and incidents**.

Think of it like a specialized database where all your security logs go.

**🧠 Relationship:**

* **Azure Sentinel** is **enabled on a Log Analytics Workspace**.
* That workspace is **hosted inside a Resource Group**.

[Resource Group]

└── [Log Analytics Workspace]

└── [Azure Sentinel Instance]

In **Azure Sentinel (Microsoft Sentinel)**, the **Resource Group** is a container that includes and organizes all the related resources needed for your Sentinel deployment. Here’s a **complete list of key components** that are typically part of a Sentinel **Resource Group**, along with brief explanations:

**🔑 1. Log Analytics Workspace**

* **Core data storage** where all ingested security logs go.
* Powers queries, detections, workbooks, and incidents using **Kusto Query Language (KQL)**.

**🔌 2. Data Connectors**

* Prebuilt **integration modules** to ingest data from sources like Microsoft 365, Azure AD, firewalls, AWS, etc.
* Each connector defines:
  + Authentication
  + API endpoints
  + Data format handling

These are configured in the Sentinel UI but are logically tied to the workspace.

**📄 Analytic Rules**

* Custom or prebuilt **correlation rules** that detect threats based on log data patterns.
* Create **incidents** automatically when triggered.

Think of them like SIEM detection logic.

**🔥 Hunting Queries**

* Predefined **KQL queries** used by security analysts to **proactively search for threats**.
* Doesn't generate incidents, used for manual threat hunting.

**📊 Workbooks**

* **Interactive dashboards** for visualizing security data.
* Can be customized to show metrics like failed logins, firewall activity, etc.

**🤖 Playbooks (Logic Apps)**

* Automated workflows that respond to alerts/incidents (SOAR capability).
* Built using **Azure Logic Apps**.
* Examples:
  + Send an alert to Slack
  + Block an IP in a firewall
  + Open a ticket in ServiceNow

**🧠 Watchlists**

* **Custom data tables** (e.g., list of VIP users or suspicious IPs) that analytic rules and hunting queries can reference.

**🛠️ Notebooks**

* **Jupyter notebooks** integrated into Sentinel for advanced threat hunting and investigation using Python + KQL.

**🔔 Incident Management**

* Incidents are generated when analytic rules are triggered.
* Supports case management: assignment, status, tagging, comments, etc.

Incident metadata is tied to the workspace but managed via Sentinel UI.

**🔗 Threat Intelligence Indicators**

* You can upload or integrate **TI feeds** (e.g., malicious IPs/domains).
* Sentinel can use them in rules or hunting.

**⚙️ Optional Resources That May Be In the Same Resource Group:**

| **Resource** | **Description** |
| --- | --- |
| Azure Functions | Custom automation triggered by alerts |
| Azure Storage Accounts | Store raw logs, archive data |
| Key Vault | Securely store secrets for playbooks |
| Azure Automation Accounts | Legacy SOAR scripts (PowerShell, etc.) |
| Managed Identities | Allow automation to access other Azure resources securely |

**✅ Summary Table**

| **Component** | **Description** |
| --- | --- |
| Log Analytics Workspace | Stores and queries all logs |
| Data Connectors | Ingest data from sources |
| Analytic Rules | Detect threats from logs |
| Playbooks | Automate incident response |
| Workbooks | Dashboards/visuals |
| Hunting Queries | Manual threat search |
| Watchlists | Custom data sets |
| Notebooks | Advanced hunting with Python |
| Threat Indicators | Custom or imported TI feeds |
| Incidents | Grouped alerts for investigation |

## KQL Queries

* .show Tables
* count (returns count)
* take <number> (return first specified number)
* project <Column names separated by commas> (returns columns)
* project <new/aliased column name> = < column name> (return column with new name)
* extend <new/aliased column name> = <column name> (return all columns and specfieid column with new name)
  + extend NewCol = EndTime - StartTime
* distinct <col name> (returns unique values in particular column)
* project-away <col name> (removes specified column)
* where (extracts specific condition)
  + where State == ‘Texas’
  + where State == ‘Texas’ and/or EventType == ‘Sample’
  + where CreatedAt > ago(3d)
* sort by <col name> asc/desc (sort by the column name)
* top <number> by <col name> (extract top number values by column name)