Databricks Certified Data Engineer Associate

Introduction to Databricks Cluster

Introduction to Databricks Cluster

- Databricks cluster is **a set of computation resources and configurations on which you run data engineering, data science, and data analytics workloads,** such as production ETL pipelines, streaming analytics, ad-hoc analytics, and machine learning
- You run these workloads as a set of commands in a notebook or as an automated job
- Azure Databricks has two types of clusters,

1. All-purpose clusters
2. Job clusters

All-Purpose vs Job Clusters

All-Purpose Clusters	Job Clusters
To analyse data collaboratively using interactive notebook	To run fast and robust automated jobs
Can be created using the UI, CLI, or REST API	Create by Databricks job scheduler
Can be terminated manually & restart	Can’t restart a job cluster, Terminated at the end of the job
Can be shared among multiple users	Isolated just for the job
Expensive to run	Cheaper to run

Databricks Cluster Configuration

Databricks Cluster Configuration

- Cluster Node Type
- Access Mode
- Databricks Runtime version
- Photon Acceleration
- Auto Scaling
- Auto Termination
- Cluster VM Type/Size
- Cluster Policy

Cluster Node Type

- A cluster consists of one driver node and zero or more worker nodes
- **Driver node:** Maintains state information of all notebooks attached to the cluster and , also maintains the SparkContext, interprets all the commands you run from a notebook or a library on the cluster, and runs the Apache Spark master that coordinates with the Spark executors
- **Worker node:** Run the Spark executors and other services required for proper functioning clusters. When you distribute your workload with Spark, all the distributed processing happens on worker nodes. Databricks runs one executor per worker node. Therefore, the terms executor and worker are used interchangeably in the context of the Databricks architecture

Access Mode

1. **Single user:** Only One User Access, Mode Supports Python, SQL, Scala, R
2. **Shared:** Multiple User Access, Only available in Premium. Supports Python, SQL
3. **No isolation shared:** Multiple User Access, Supports Python, SQL
4. **Custom:** Legacy Configuration

Databricks Runtime version

- Databricks Runtime is **the set of core components that run on your clusters**
- All Databricks Runtime versions include Apache Spark and add components and updates that improve usability, performance, and security
- You select the cluster's runtime and version using the Databricks Runtime Version dropdown when you create or edit a cluster

Photon Acceleration

- Photon is available for clusters running Databricks Runtime 9.1 LTS and above
- To enable Photon acceleration, select the Use Photon Acceleration checkbox
- If desired, you can specify the instance type in the Worker Type and Driver Type drop-down
- Databricks recommends the following instance types for optimal price and performance:

1. Standard_E4ds_v4
2. Standard_E8ds_v4
3. Standard_E16ds_v4

Auto Scaling

- You can specify min and max work nodes
- Auto scales between min and max based on the workload
- Not recommended for streaming workloads

Auto Termination

- Terminates the cluster after X minutes of inactivity
- Default value for Single Node and Standard clusters is 120 minutes
- Users can specify a value between 10 and 10000 mins as the duration

Cluster VM Type/Size

1. Memory Optimized
2. Compute Optimized
3. Storage Optimized
4. General Purpose
5. GPU Accelerated

Cluster Policy

- Cluster policies are **a set of rules used to limit the configuration options available to users when they create a cluster**

1. Unrestricted
2. Personal Compute
3. Power User Compute
4. Shared Compute

Azure Databricks Pricing

Azure Databricks Pricing

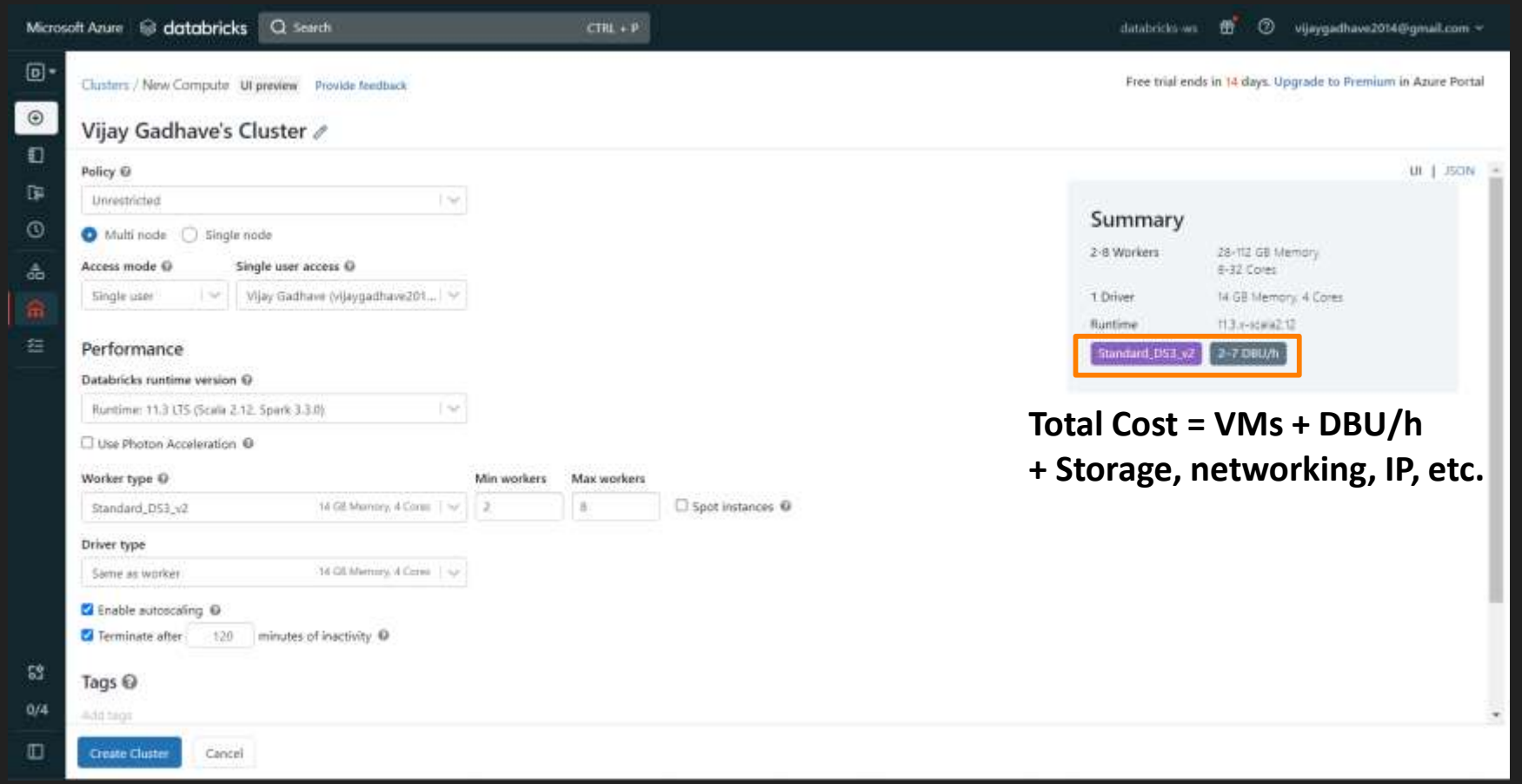
- Databricks pricing is based on your compute usage. Storage, networking & related costs will vary depending on the services you choose and your cloud service provider
- **Databricks Unit (DBU):** Normalized unit of processing power on the Databricks Lakehouse Platform used for measurement and pricing purposes
- The number of DBUs a workload consumes is driven by processing metrics, which may include the compute resources used and the amount of data processed
- Databricks prices and Cloud Infrastructure prices may vary based on geographic region and cloud service provider

Azure Databricks Pricing

- **Pricing factors:**

1. Type of workload (All Purpose/ Jobs/ SQL/ Photon)
2. Tier of Databrics Workspace (Premium/ Standard)
3. VM Type (General Purpose/ GPU/ Optimized)
4. Purchase Plan (Pay As You Go/ Pre-Purchase)

Actual price for running cluster



Total Cost = VMs + DBU/h
+ Storage, networking, IP, etc.

Azure Databricks Pricing

- Calculate the pricing using below 2 links, e.g. Databricks course completion

1. Azure Databricks pricing: https://azure.microsoft.com/en-gb/pricing/details/databricks/
2. Pricing calculator: https://azure.microsoft.com/en-gb/pricing/calculator/
3. Linux Virtual Machines Pricing: https://azure.microsoft.com/en-gb/pricing/details/virtual-machines/linux/#pricing

Azure Cost Control

Azure Cost Control

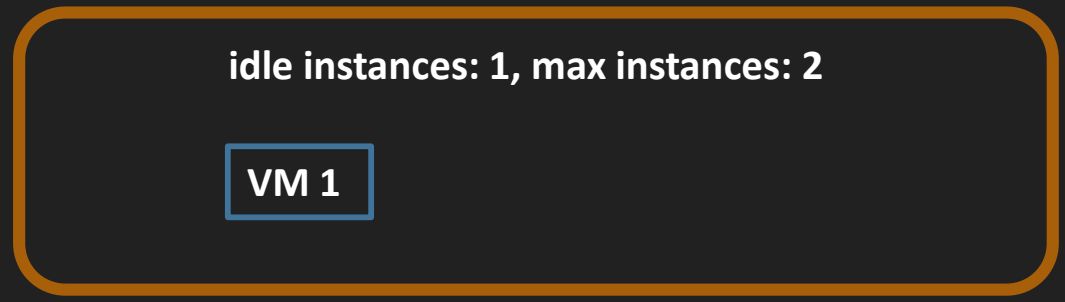
- Azure portal - Cost Management
- Explain: Billing, Cost analysis, Cost alerts, Budgets
- **Create a sample budget and show**

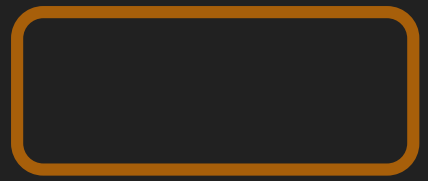
Databricks Cluster Pool

Databricks Cluster Pool

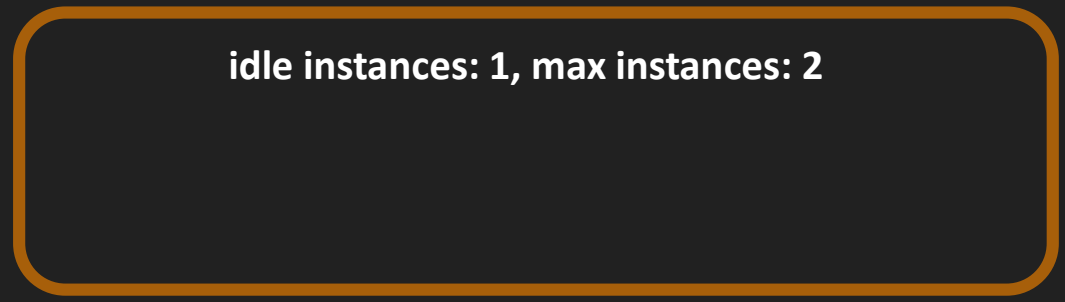
- Databricks pools reduce cluster start and auto-scaling times by maintaining a set of idle, ready-to-use instances
- When a cluster is attached to a pool, cluster nodes are created using the pool's idle instances
- If the pool has no idle instances, the pool expands by allocating a new instance from the instance provider in order to accommodate the cluster's request
- When a cluster releases an instance, it returns to the pool and is free for another cluster to use
- Only clusters attached to a pool can use that pool's idle instances

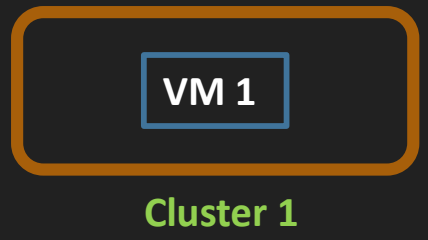
Databricks Cluster Pool



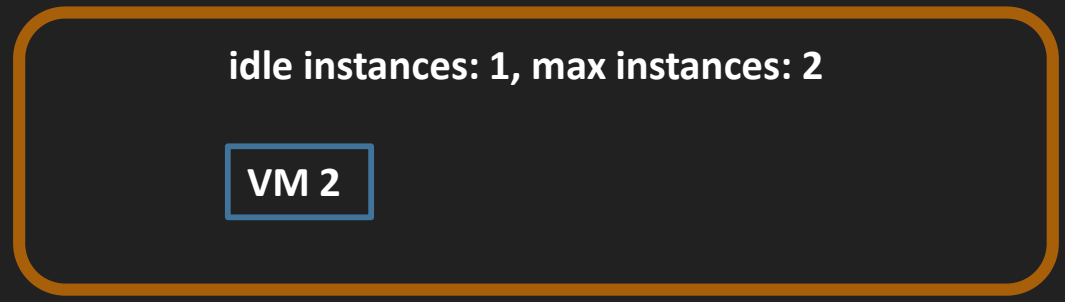


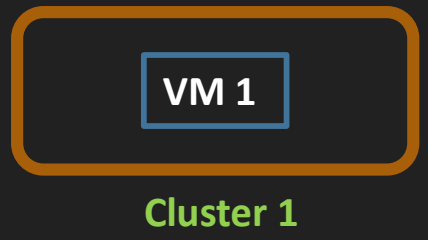
Databricks Cluster Pool



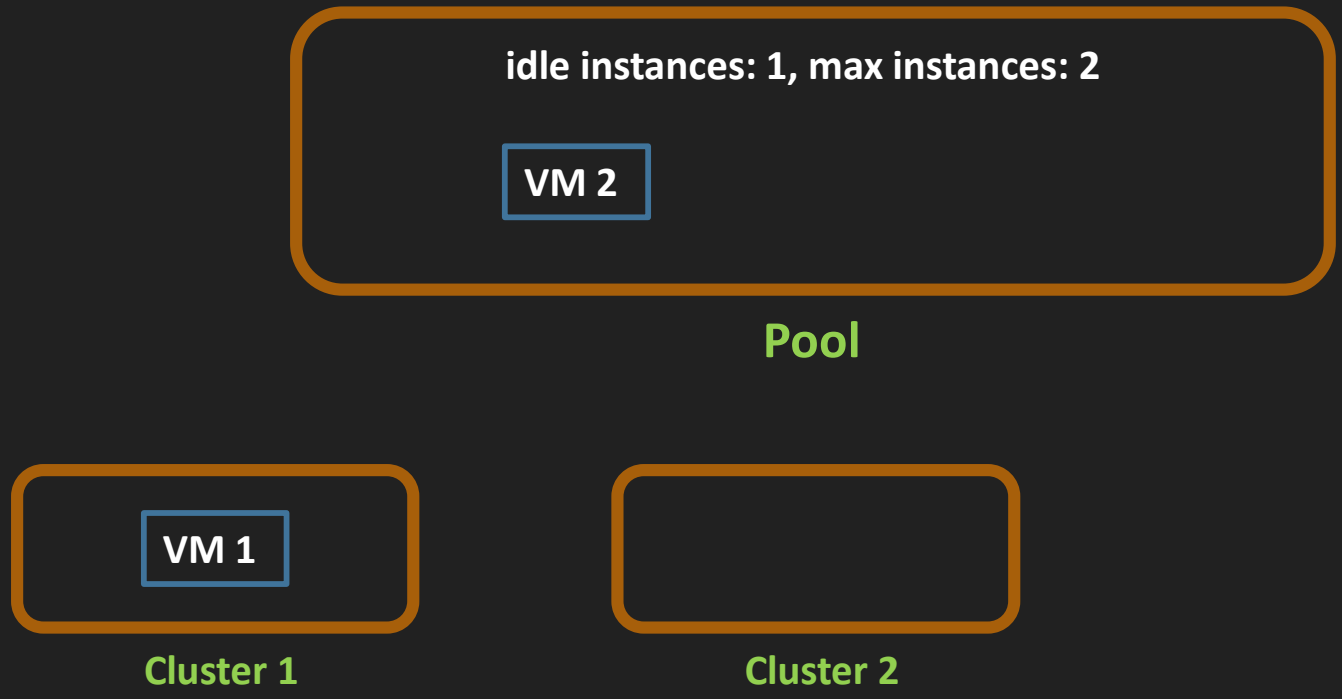


Databricks Cluster Pool

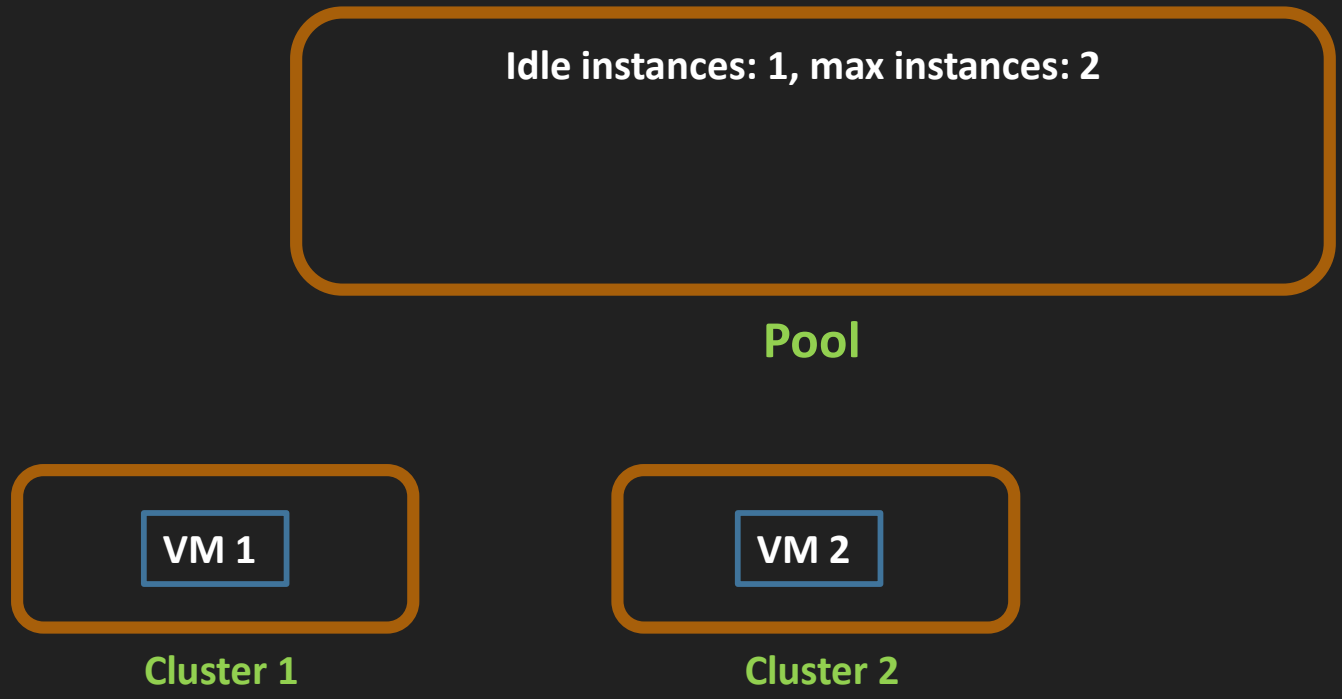




Databricks Cluster Pool



Databricks Cluster Pool



Databricks Cluster Pool

Cluster Policies

Cluster Policies

- A cluster policy is a tool used to limit a user or group's cluster creation permissions based on a set of policy rules
- **Cluster policies let you:**

1. Limit users to creating clusters with prescribed settings
2. Limit users to creating a certain number of clusters
3. Simplify the user interface and enable more users to create their own clusters (by fixing and hiding some values)
4. Control cost by limiting per cluster maximum cost (by setting limits on attributes whose values contribute to hourly price)

Cluster Policies

- Lab 1: Write a policy to default select spark latest runtime version (LTS)
- Lab 2: Add multiple conditions as discussed earlier and explain
- Lab 3: Use existing cluster policy templates or families (Personal Compute)
- Lab 4: Use existing cluster policy templates or families (Personal Compute - Edit)
- Delete the created policies and delete the workspace

THANK YOU