

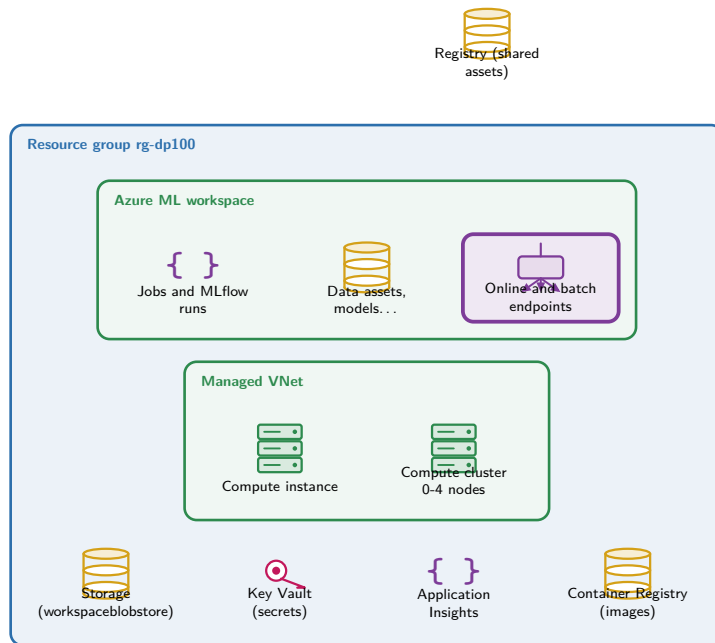


DP-100 Domain 1: Design and Prepare an ML Solution

DP-100 skills measured, domain 1 (20-25 %), as of 2026 | Workspace and its resources, compute targets, datastores and data assets, environments, registries, SDK v2 / CLI v2, security (RBAC, managed VNet, identities), Git and cost

DP-100 Domain 1 Azure ML workspace Compute Data assets

Workspace and its resources



A workspace depends on Storage, Key Vault and Application Insights; ACR is created on the first image build

```
Connect with SDK v2
from azure.ai.ml import MLClient
from azure.identity import DefaultAzureCredential
ml_client = MLClient(DefaultAzureCredential(),
    subscription_id="<sub>", resource_group_name="rg-dp100",
    workspace_name="mlw-dp100")
# or MLClient.from_config(DefaultAzureCredential())
# reads config.json written by the Studio / compute instance
for c in ml_client.compute.list():
    print(c.name, c.type)
```

Workspace kind	Purpose	Note
Default workspace	classic ML: jobs, AutoML, endpoints	the DP-100 core
Hub workspace	shared settings and connections for Foundry hub-based projects	projects inherit network and storage
Project (under a hub)	generative-AI work: prompt flow, evaluation, catalog deployments	see sheet 05
Registry	share assets across workspaces and regions	promote dev → test → prod

Choose the compute

Compute target	Use	Key settings
Compute instance	notebooks, VS Code, small experiments; one user	size, idle shutdown, schedules, assigned user
Compute cluster	training, sweeps, pipelines, batch endpoints	min/max nodes, idle_time_before_scale_down (default 120 s), tier dedicated or low priority
Serverless compute	jobs with no compute to manage	resources.instance_type, instance_count in the job
Serverless Spark / attached Synapse	big-data wrangling in notebooks or jobs	executor sizes, session timeout
Kubernetes compute (AKS or Arc)	training or inference on your cluster, on-premises	install the Azure ML extension, attach
Managed online endpoint	real-time inference, Microsoft-managed VMs	instance type and count (sheet 04)

```

0 $ az ml compute create -n cpu-cluster
1 $ az ml compute create -n cpu-cluster --type AmlCompute
  a cluster, not an instance
2 $ az ml compute create -n cpu-cluster --type AmlCompute --size Standard_DS3_v2
  VM size of each node
3 $ az ml compute create -n cpu-cluster --type AmlCompute --size Standard_DS3_v2
  --min-instances 0
  scale to zero when idle
4 $ az ml compute create -n cpu-cluster --type AmlCompute --size Standard_DS3_v2
  --min-instances 0 --max-instances 4
  cap parallel jobs and cost
5 $ az ml compute create -n cpu-cluster --type AmlCompute --size Standard_DS3_v2
  --min-instances 0 --max-instances 4 --idle-time-before-scale-down 300
  seconds before idle nodes are released
6 $ az ml compute create -n cpu-cluster --type AmlCompute --size Standard_DS3_v2
  --min-instances 0 --max-instances 4 --idle-time-before-scale-down 300 --tier low_priority
  cheaper, can be pre-empted: fine for retryable training
7 $ az ml compute create -n cpu-cluster --type AmlCompute --size Standard_DS3_v2
  --min-instances 0 --max-instances 4 --idle-time-before-scale-down 300 --tier low_priority -g rg-dp100 -w mlw-dp100
  resource group and workspace

```

CLUSTER THAT COSTS NOTHING WHEN IDLE

```

$ az ml compute create -n cpu-cluster --type AmlCompute --size Standard_DS3_v2 --min-instances
0 --max-instances 4 --idle-time-before-scale-down 300 --tier low_priority -g rg-dp100 -w
mlw-dp100

```

Worked example: cluster cost, scale-to-zero vs always-on

Illustrative 0.30 USD per node-hour. Team runs 3 h of 4-node training per working day, 22 days a month. Scale-to-zero (min 0): $4 \times 3 \times 22 = 264$ node-hours $\rightarrow$ 79 USD. Always-on min 1 node: 79 USD + 1 node idle the rest of the month ($730 - 66 = 664$ h) $\times$ 0.30 = 199 USD more $\rightarrow$ 278 USD. min 0 saves about 70 %; the price is a 2-5 minute node start at each job.

Data: datastores and data assets

Datastores (where)	Data assets (what, versioned)
workspaceblobstore (default) and workspacefilestore created with the workspace	uri_file: one file (CSV, parquet)
types: Azure Blob, ADLS Gen2, Azure Files (OneLake as of 2026)	uri_folder: a folder (images, many files)
auth: account key, SAS, or identity-based (no secret stored)	mltable: tabular schema via an MLTable file; AutoML needs it
URI: azureml://datastores/<name>/paths/<folder>/	reference: azureml:<name>:<version> or @latest

```

MLTable
1 paths:
2   - pattern: ./sales/*.csv
3 transformations:
4   - read_delimited:
5     delimiter: ','
6     header: all_files_same_headers
7     encoding: utf8
8   - convert_column_types:
9     - columns: order_date
10      column_type:
11        datetime:
12          formats: ['%Y-%m-%d']
13   - drop_columns: [internal_id]

```

1 glob over many files

2 how to parse the CSVs

3 typed columns at load time

4 never expose this column to training

Register and consume a data asset

```

from azure.ai.ml.entities import Data
from azure.ai.ml.constants import AssetTypes
from azure.ai.ml import Input
d = Data(name="sales", version="3", type=AssetTypes.MLTABLE,
        path="azureml://datastores/workspaceblobstore/paths/sales/")
ml_client.data.create_or_update(d)
inp = Input(type=AssetTypes.MLTABLE, path="azureml:sales:3")
# mode on a job input: ro_mount (default), download, direct
import mltable
df = mltable.load("azureml:/...").to_pandas_dataframe()

```

Mount or download?

ro_mount streams files on demand (large datasets, default for uri_folder); download copies everything first (many small files read repeatedly); direct passes the URI and your code reads it (Spark). Outputs use rw_mount or upload.

Environments, registries and code

```

Custom environment from a curated base + conda file
from azure.ai.ml.entities import Environment
env = Environment(name="sk-train", version="2",
                 image="mcr.microsoft.com/azureml/openmpi4.1.0-ubuntu22.04",
                 conda_file="env/conda.yml") # python, scikit-learn,
ml_client.environments.create_or_update(env) # mlflow
# curated: "AzureML-sklearn-1.5@latest"
# Dockerfile: build=BuildContext(path="docker/")

```

- **Image build** first job with a new environment builds an image in the workspace ACR (minutes); later jobs reuse the cached image
- **Pin versions** pin package versions in conda.yml; @latest is fine for labs, not production
- **Registries** ml_client_registry = MLCClient(cred, registry_name="contoso-ml"); share models, environments and components across workspaces
- **Git** clone repos on the compute instance; jobs record the commit hash of the code folder

- **IaC** Bicep `Microsoft.MachineLearningServices/workspaces`, Terraform `azurerm_machine_learning_workspace`; CLI v2 YAML for assets and jobs in CI/CD

🔒 Secure the workspace

Role	Can	Cannot
Owner	everything, including role assignments	-
Contributor	create workspace resources and compute	assign roles
AzureML Data Scientist	run jobs, register data and models, deploy endpoints	create compute, change workspace settings
AzureML Compute Operator	create and manage compute	run experiments
Reader	view assets and runs	change anything

- **Managed VNet** isolation modes: Allow internet outbound, Allow only approved outbound (outbound rules: private endpoints, service tags, FQDNs), or Disabled (bring your own VNet)
- **Private endpoint** workspace and its Storage, Key Vault, ACR behind private endpoints; public network access disabled
- **Identities** compute and endpoints get managed identities; grant them Storage Blob Data Reader for identity-based datastores
- **Encryption** customer-managed keys at workspace creation (cannot be added later)

Job fails reading the datastore: AuthorizationFailure
HIGH
COMMON

LIKELY CAUSES 1. Identity-based datastore, compute identity has no data role 2. Storage firewall blocks the managed VNet 3. SAS token expired	HOW TO CONFIRM Check role assignments on the storage account <pre>az role assignment list --scope <storage-id> -o table</pre>	FIX ✓ Grant Storage Blob Data Reader to the compute managed identity ✓ Add a private endpoint outbound rule ✓ Renew the SAS or switch to identity
--	---	--

Job stuck in Preparing for 10+ minutes
MEDIUM
COMMON

LIKELY CAUSES 1. First image build for a new environment 2. Cluster scaling from 0 nodes 3. Quota of the VM family reached	HOW TO CONFIRM Open the job's image build log and cluster node state <pre>az ml job show -n <job> -g rg-dp100 -w mlv-dp100</pre>	FIX ✓ Reuse a registered environment version ✓ Keep min 1 node during intense iteration ✓ Request more vCPU quota
---	--	--

🚩 Recap and self-test

YOU MAY HEAR
BUSTED

"A compute instance can be shared by the team"

✓ **ACTUALLY**
It has one assigned user
Use clusters or serverless compute for shared jobs.

YOU MAY HEAR
BUSTED

"Data assets copy the data into Azure ML"

✓ **ACTUALLY**
They reference data in a datastore, with a version
Delete the blob and the asset breaks.

YOU MAY HEAR
BUSTED

"AzureML Data Scientist can create clusters"

✓ **ACTUALLY**
It cannot create compute
Pair it with AzureML Compute Operator, or have admins pre-create compute.

Q1 Data asset type AutoML requires for tabular data?

A mtable.

Q2 Default datastore name?

A workspaceblobstore.

Q3 URI format of a datastore path?

A azureml://datastores/<datastore>/paths/<path>.

Q4 Compute for interactive notebooks of one person?

A Compute instance.

Q5 How do you make a cluster cheaper for retryable training?

A Low-priority tier and min 0 nodes.

Q6 Two ways to define a custom environment?

A Base Docker image + conda file, or a Dockerfile build context.

Q7 Share a model across three workspaces in two regions?

A Register it in an Azure ML registry.

Q8 Managed VNet mode that blocks all internet except listed destinations?

A Allow only approved outbound.

Q9 Job input mode that streams files on demand?

A ro_mount (read-only mount).

Tick a box only when you could explain the point to someone else



Workspace + Storage + Key Vault + App Insights + ACR



Compute by job

instance for notebooks, cluster or serverless for jobs



Datastore = where, data asset = what + version

MLTable for AutoML



Environments are versioned images

curated or custom



Least privilege

AzureML Data Scientist, identity-based data, managed VNet

