Create compute targets

5 minutes

The most common ways to create or attach a compute target are to use the **Compute** page in Azure Machine Learning studio, or to use the Azure Machine Learning SDK to provision compute targets in code.

Creating a managed compute target with the SDK

A *managed* compute target is one that is managed by Azure Machine Learning, such as an Azure Machine Learning compute cluster.

To create an Azure Machine Learning compute cluster, use the **azureml.core.compute.ComputeTarget** class and the **AmlCompute** class, like this:

Python Copy

```
from azureml.core import Workspace
from azureml.core.compute import ComputeTarget, AmlCompute

# Load the workspace from the saved config file
ws = Workspace.from_config()

# Specify a name for the compute (unique within the workspace)
compute_name = 'aml-cluster'

# Define compute configuration
compute_config = AmlCompute.provisioning_configuration(vm_size='STANDARD_DS11_V2',
                                                       min_nodes=0, max_nodes=4,
                                                       vm_priority='dedicated')

# Create the compute
aml_cluster = ComputeTarget.create(ws, compute_name, compute_config)
aml_cluster.wait_for_completion(show_output=True)
```

In this example, a cluster with up to four nodes that is based on the STANDARD_DS12_v2 virtual machine image will be created. The priority for the virtual machines (VMs) is set to dedicated, meaning they are reserved for use in this cluster (the alternative is to specify *lowpriority*, which has a lower cost but means that the VMs can be preempted if a higher-priority workload requires the compute).

> **Note**
>
> For a full list of **AmlCompute** configuration options, see the **AmlCompute class** SDK documentation.

Attaching an unmanaged compute target with the SDK

An *unmanaged* compute target is one that is defined and managed outside of the Azure Machine Learning workspace; for example, an Azure virtual machine or an Azure Databricks cluster.

The code to attach an existing unmanaged compute target is similar to the code used to create a managed compute target, except that you must use the **ComputeTarget.attach()** method to attach the existing compute based on its target-specific configuration settings.

For example, the following code can be used to attach an existing Azure Databricks cluster:

Python Copy

```
from azureml.core import Workspace
from azureml.core.compute import ComputeTarget, DatabricksCompute

# Load the workspace from the saved config file
ws = Workspace.from_config()

# Specify a name for the compute (unique within the workspace)
compute_name = 'db_cluster'

# Define configuration for existing Azure Databricks cluster
db_workspace_name = 'db_workspace'
db_resource_group = 'db_resource_group'
db_access_token = '1234-abc-5678-defg-90...'
db_config =
DatabricksCompute.attach_configuration(resource_group=db_resource_group,

workspace_name=db_workspace_name,
                                                    access_token=db_access_token)

# Create the compute
databricks_compute = ComputeTarget.attach(ws, compute_name, db_config)
databricks_compute.wait_for_completion(True)
```

Checking for an existing compute target

In many cases, you will want to check for the existence of a compute target, and only create a new one if there isn't already one with the specified name. To accomplish this, you can catch the **ComputeTargetException** exception, like this:

Python Copy

```
from azureml.core.compute import ComputeTarget, AmlCompute
from azureml.core.compute_target import ComputeTargetException

compute_name = "aml-cluster"

# Check if the compute target exists
try:
    aml_cluster = ComputeTarget(workspace=ws, name=compute_name)
    print('Found existing cluster.')
except ComputeTargetException:
    # If not, create it
    compute_config =
AmlCompute.provisioning_configuration(vm_size='STANDARD_DS11_V2',
                                                           max_nodes=4)
    aml_cluster = ComputeTarget.create(ws, compute_name, compute_config)

aml_cluster.wait_for_completion(show_output=True)
```

> **More Information**: For more information about creating compute targets, see Set up and use compute targets for model training in the Azure Machine Learning documentation.

Next unit: Use compute targets

Continue >