

# Hyperparameter Tuning using HyperDrive

TODO: Import Dependencies. In the cell below, import all the dependencies that you will need to complete the project.

```
In [1]: import os
import time
import requests
import pandas as pd
from azureml.widgets import RunDetails
from azureml.core import Workspace, Experiment, Environment, ScriptRunConfig
from azureml.core.compute import ComputeTarget, AmlCompute
from azureml.train.sklearn import SKLearn
from azureml.train.hyperdrive.policy import BanditPolicy
from azureml.train.hyperdrive.sampling import RandomParameterSampling
from azureml.train.hyperdrive.parameter_expressions import loguniform, uniform
from azureml.train.hyperdrive.run import PrimaryMetricGoal
from azureml.train.hyperdrive.runconfig import HyperDriveConfig
from azureml.core.model import InferenceConfig
from azureml.core.webservice import AciWebservice
from azureml.core.model import Model
```

```
In [2]: time.strftime('%Y-%m-%d %H:%M:%S')
```

```
Out[2]: '2021-02-10 14:35:20'
```

```
In [3]: ws = Workspace.from_config()
print(ws.name, ws.resource_group, ws.location, ws.subscription_id, sep = '\n')
```

```
quick-starts-ws-138398
aml-quickstarts-138398
southcentralus
1b944a9b-fdae-4f97-aeb1-b7eea0beac53
```

```
In [4]: compute_name = os.environ.get("AML_COMPUTE_CLUSTER_NAME", "auto-ml")
vm_size = os.environ.get("AML_COMPUTE_CLUSTER_SKU", "STANDARD_D2_V2")
compute_min_nodes = os.environ.get("AML_COMPUTE_CLUSTER_MIN_NODES", 2)
compute_max_nodes = os.environ.get("AML_COMPUTE_CLUSTER_MAX_NODES", 4)

compute_names = [cn for cn in ws.compute_targets if cn in [compute_name, 'auto-ml', 'aml-compute']]

if compute_names:
    compute_target = ws.compute_targets[compute_names[0]]
    if compute_target and type(compute_target) is AmlCompute:
        print('Using existing compute target: ' + compute_names[0])
else:
    compute_config = AmlCompute.provisioning_configuration(
        vm_size=vm_size,
        min_nodes=compute_min_nodes,
        max_nodes=compute_max_nodes
    )
    compute_target = ComputeTarget.create(ws, compute_name, compute_config)
```

```
Using existing compute target: auto-ml
```

## Dataset

### Overview

We will be using the Wine Quality dataset made accessible [here \(https://archive.ics.uci.edu/ml/datasets/wine+quality\)](https://archive.ics.uci.edu/ml/datasets/wine+quality).

The dataset contains the physicochemical properties of 1599 red wine samples. The data includes, in column order: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and the final column is the target variable, quality (score between 0 and 10).

We will use HyperDrive to explore the parameter space in attempt to find a suitable parameter combination for an ElasticNet model to predict the quality of a particular sample of wine from measured properties.

## Hyperdrive Configuration

The ElasticNet model is a linear model that combines L1 and L2 regularization. The parameters `alpha` and `l1_ratio` control the regularization penalties. An early stopping policy that would terminate low performing runs sooner than later is needed.

The Bandit policy with a 20% slack ratio terminates runs if the primary metric does not match the best run so far within 20% (e.g. if the best run so far had a NRMSE of 1.0, any run with a NRMSE above 1.2 will be terminated.)

The configuration settings specify the `train.py` training script, the environment, the compute target, and the source directory where `train.py` is located.

```
In [5]: sklearn_env = Environment.get(workspace=ws, name='AzureML-Tutorial')

hyp_est = ScriptRunConfig(
    source_directory='./',
    script='train.py',
    environment=sklearn_env,
    compute_target=compute_target
)

'''
hyperparameters for sklearn.linear_model.ElasticNet:

Constant that multiplies the penalty terms.
alpha=1.0 # [.001, .01, .1, 1]

For l1_ratio = 0 the penalty is an L2 penalty.
For l1_ratio = 1 it is an L1 penalty.
For 0 < l1_ratio < 1, the penalty is a combination of L1 and L2.
l1_ratio=1.0 # [0..1]
'''

# Specify parameter sampler
ps = RandomParameterSampling({
    "alpha": loguniform(-4,-2),
    "l1_ratio": uniform(0,1)
})

# Specify a Policy for early stopping
policy = BanditPolicy(
    evaluation_interval = 10,
    slack_factor = 0.2
)
```

```
In [6]: # Create a HyperDriveConfig using the estimator, hyperparameter sampler, and policy.
hyperdrive_config = HyperDriveConfig(
    run_config=hyp_est,
    hyperparameter_sampling=ps,
    primary_metric_name='mean_squared_error',
    primary_metric_goal=PrimaryMetricGoal.MINIMIZE,
    policy=policy,
    max_total_runs=20,
    max_concurrent_runs=4
)
```

```
In [7]: experiment_name = 'winequality-hyperdrive'
hyp_exp = Experiment(ws, experiment_name)
```

```
In [8]: compute_target.wait_for_completion(show_output=True)
```

Succeeded.....  
AmlCompute wait for completion finished  
  
Minimum number of nodes requested have been provisioned

```
In [9]: hyp_run = hyp_exp.submit(config=hyperdrive_config)
hyp_run
```

Out[9]:

Experiment	Id	Type	Status	Details Page	Docs Page
winequality-hyperdrive	HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5	hyperdrive	Running	<a href="https://ml.azure.com/experiments/winequality-hyperdrive/runs/HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5?wsid=/subscriptions/1b944a9b-fdae-4f97-aeb1-b7eea0beac53/resourcegroups/aml-quickstarts-138398/workspaces/quickstarts-ws-138398">Link to Azure Machine Learning studio</a> <a href="https://ml.azure.com/experiments/winequality-hyperdrive/runs/HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5?wsid=/subscriptions/1b944a9b-fdae-4f97-aeb1-b7eea0beac53/resourcegroups/aml-quickstarts-138398/workspaces/quickstarts-ws-138398">https://ml.azure.com/experiments/winequality-hyperdrive/runs/HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5?wsid=/subscriptions/1b944a9b-fdae-4f97-aeb1-b7eea0beac53/resourcegroups/aml-quickstarts-138398/workspaces/quickstarts-ws-138398</a>	<a href="https://docs.microsoft.com/en-us/python/api/overview/azure/ml/rview=azure-ml-py">Link to Documentation</a> <a href="https://docs.microsoft.com/en-us/python/api/overview/azure/ml/rview=azure-ml-py">https://docs.microsoft.com/en-us/python/api/overview/azure/ml/rview=azure-ml-py</a>

Run Details

```
In [10]: RunDetails(hyp_run).show()
```

Non-numeric values for primary metric cannot be visualized in a 2D/3D/Parallel coordinates chart.

Run Properties

Status	Completed
Node run Start Time	2/10/2021 9:36:24 AM
Node run Duration	0:19:55
Run Id	HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5
Max concurrent runs	4
Max total runs	20

Output Logs

azureml-logs/hyperdrive.txt

Show Active log

azureml-logs/hyperdrive.txt

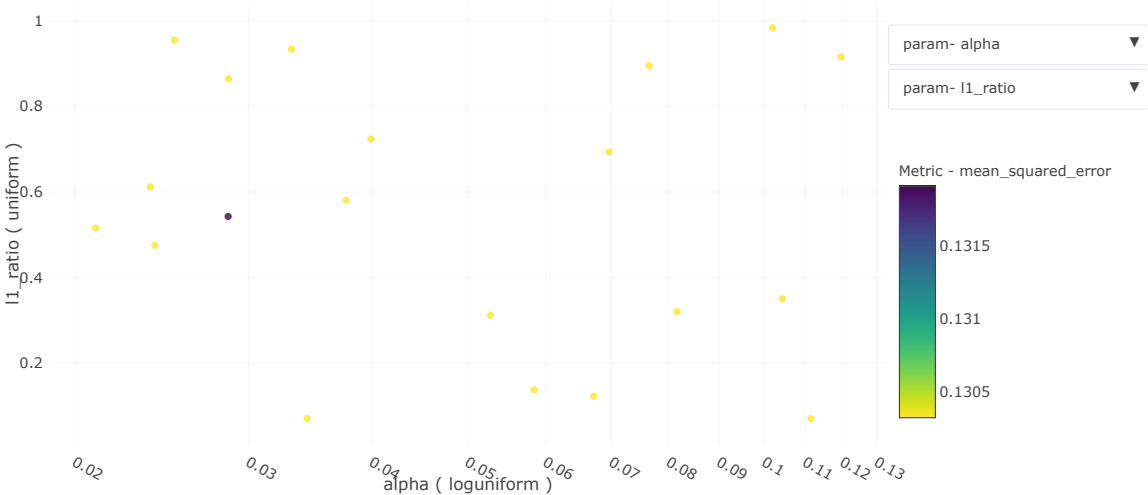
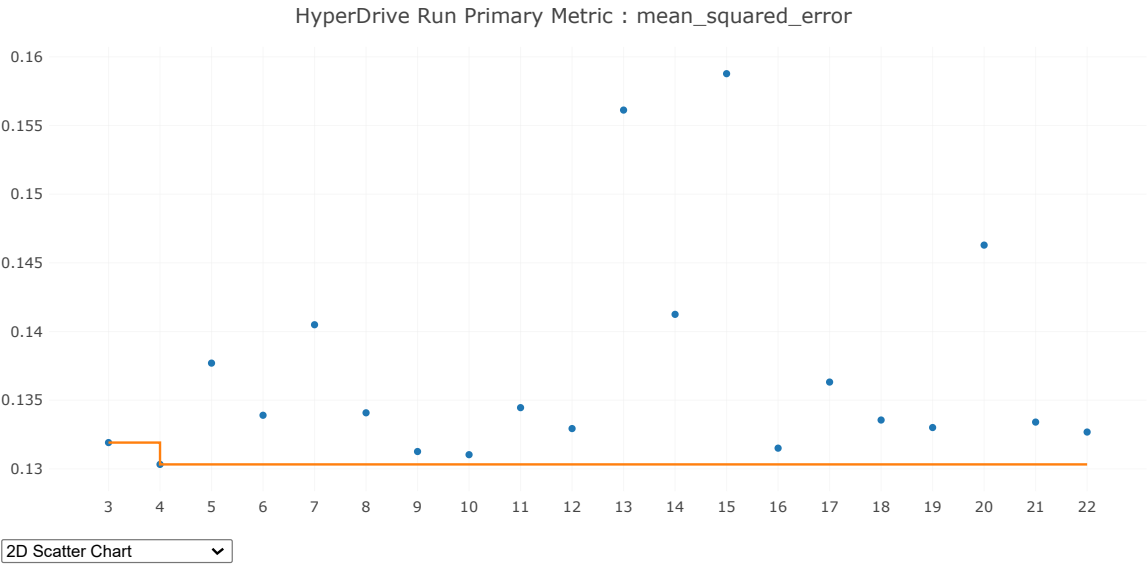
contain any metrics with the primary metric name at this moment, policy cannot be applied.  
[2021-02-10T14:55:04.291056][ENFORCER][INFO]Jobs  
[https://southcentralus.experiments.azureml.net/subscriptions/1b944a9b-fdae-4f97-aeb1-b7eea0beac53/resourceGroups/aml-quickstarts-138398/providers/Microsoft.MachineLearningServices/workspaces/quick-starts-ws-138398/experiments/\*\*SCRUBBED\*\*/runs/HD\_e9eae0f6-21a1-45ec-a8af-6de22828b4c5\_19] do not  
contain any metrics with the primary metric name at this moment, policy cannot be applied.  
[2021-02-10T14:56:19.794139][CONTROLLER][INFO]Experiment was 'ExperimentStatus.RUNNING', is  
'ExperimentStatus.FINISHED'.  
  
Run is completed.

Completed (20)

Run	Best Metric*	Status	Started	Duration	Run Id	alpha
4	0.13032434	Completed	Feb 10, 2021 9:37 AM	0:03:36	HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5_3	0.03440585
10	0.13103191	Completed	Feb 10, 2021 9:45 AM	0:02:26	HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5_6	0.02099064
9	0.13126246	Completed	Feb 10, 2021 9:46 AM	0:03:35	HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5_5	0.02409053
16	0.13150756	Completed	Feb 10, 2021 9:50 AM	0:02:30	HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5_13	0.0238488
3	0.13191365	Completed	Feb 10, 2021 9:39 AM	0:06:35	HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5_2	0.02860418

Pages: 1 2 3 4 Next Last

\* The best metric field is obtained from the min/max of primary metric achieved by a run



Click here to see the run in Azure Machine Learning studio ([https://ml.azure.com/experiments/winequality-hyperdrive/runs/HD\\_e9eae0f6-21a1-45ec-a8af-6de22828b4c5?wsid=subscriptions/1b944a9b-fdae-4f97-aeb1-b7eea0beac53/resourcegroups/aml-quickstarts-138398/workspaces/quick-starts-ws-138398](https://ml.azure.com/experiments/winequality-hyperdrive/runs/HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5?wsid=subscriptions/1b944a9b-fdae-4f97-aeb1-b7eea0beac53/resourcegroups/aml-quickstarts-138398/workspaces/quick-starts-ws-138398))

Models trained with a lower  $\alpha$  and  $l1\_ratio$  closer to 0 achieve a better fit to the training data. This indicates that the algorithm was able to fit a better model with less regularization on both L1 and L2 levels.

## Best Model

In [11]: `hyp_run.wait_for_completion()`

```
Out[11]: {'runId': 'HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5',
          'target': 'auto-ml',
          'status': 'Completed',
          'startTimeUtc': '2021-02-10T14:36:24.096473Z',
          'endTimeUtc': '2021-02-10T14:56:19.589718Z',
          'properties': {'primary_metric_config': '{"name": "mean_squared_error", "goal": "minimize"}',
                        'resume_from': 'null',
                        'runTemplate': 'HyperDrive',
                        'azureml.runsource': 'hyperdrive',
                        'platform': 'AML',
                        'ContentSnapshotId': 'e70b3e10-5272-4628-aed0-6dddb5df3283',
                        'score': '0.1303243415168726',
                        'best_child_run_id': 'HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5_3',
                        'best_metric_status': 'Succeeded'},
          'inputDatasets': [],
          'outputDatasets': [],
          'logFiles': {'azureml-logs/hyperdrive.txt': 'https://mlstrg138398.blob.core.windows.net/azureml/ExperimentRun/dcid.HD_e9eae0f6-21a1-45ec-a8af-6de22828b4c5/azureml-logs/hyperdrive.txt?sv=2019-02-02&sr=b&sig=ynVApZHdV98HB6%2FZVP669kRPKsHx9Yxo0HgIN40ArV8%3D&st=2021-02-10T14%3A46%3A28Z&se=2021-02-10T22%3A56%3A28Z&sp=r'},
          'submittedBy': 'ODL_User 138398'}
```

In [12]: `hyp_best_run = hyp_run.get_best_run_by_primary_metric()`  
`hyp_best_run_metrics = hyp_best_run.get_metrics()`  
`print('Best Run Id: ', hyp_best_run.id)`  
`[(k,round(v,4)) for k,v in hyp_best_run_metrics.items()]`

Best Run Id: HD\_e9eae0f6-21a1-45ec-a8af-6de22828b4c5\_3

```
Out[12]: [('Alpha', 0.0344),
          ('L1 Ratio', 0.0714),
          ('spearman_correlation', 0.5763),
          ('mean_squared_error', 0.1303),
          ('r2_score', 0.3303)]
```

In [13]: `# Get your best run and save the model from that run.`  
`# specify the whole path since we have a scaler object also`  
`hyp_model = hyp_best_run.register_model(`  
 `model_name='wine-quality-hyperdrive-best-model',`  
 `model_path='outputs'`  
`)`

## Model Deployment

The AutoML model performed better, however, for practice, we'll deploy and test this model anyway.

In [14]: `inference_config = InferenceConfig(`  
 `entry_script='score_hyperdrive.py',`  
 `environment=sklearn_env`  
`)`  
`aci_config = AciWebService.deploy_configuration(`  
 `cpu_cores=1,`  
 `memory_gb=1`  
`)`

In [15]: `service_name = 'wine-quality-predictor-hd'`  
`service = Model.deploy(`  
 `workspace=ws,`  
 `name=service_name,`  
 `models=[hyp_model],`  
 `inference_config=inference_config,`  
 `deployment_config=aci_config,`  
 `overwrite=True`  
`)`

Send a request to the deployed web service

```
In [16]: service.wait_for_deployment(show_output=True)
print("State: " + service.state)
print("Scoring URI: " + service.scoring_uri)
```

Tips: You can try `get_logs()`: <https://aka.ms/debugimage#dockerlog> (<https://aka.ms/debugimage#dockerlog>) or local deployment: <https://aka.ms/debugimage#debug-locally> (<https://aka.ms/debugimage#debug-locally>) to debug if deployment takes longer than 10 minutes.

Running.....  
Succeeded  
ACI service creation operation finished, operation "Succeeded"  
State: Healthy  
Scoring URI: <http://bdd10b94-7510-455b-80a8-d19845ee089c.southcentralus.azurecontainer.io/score> (<http://bdd10b94-7510-455b-80a8-d19845ee089c.southcentralus.azurecontainer.io/score>)

```
In [17]: data_file_source = 'https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv'
df = pd.read_csv(data_file_source, delimiter=';').dropna()
# select a few random rows from the test set to score
random_data = df.sample(5, random_state=42).values
random_data.shape
```

Out[17]: (5, 12)

```
In [18]: x_test = random_data[:, :-1].tolist()
y_test = random_data[:, -1].tolist()

input_data = "{\"data\": " + str(x_test) + "}"
headers = {'Content-Type': 'application/json'}

resp = requests.post(service.scoring_uri, input_data, headers=headers)

print("POST to url", service.scoring_uri)
print("input data:", input_data)
print("label:", y_test)
print("prediction:", resp.text)
```

POST to url <http://bdd10b94-7510-455b-80a8-d19845ee089c.southcentralus.azurecontainer.io/score> (<http://bdd10b94-7510-455b-80a8-d19845ee089c.southcentralus.azurecontainer.io/score>)  
input data: {"data": [[7.7, 0.56, 0.08, 2.5, 0.114, 14.0, 46.0, 0.9971, 3.24, 0.66, 9.6], [7.8, 0.5, 0.17, 1.6, 0.08199999999999999, 21.0, 102.0, 0.996, 3.39, 0.48, 9.5], [10.7, 0.67, 0.22, 2.7, 0.107, 17.0, 34.0, 1.0004, 3.28, 0.98, 9.9], [8.5, 0.46, 0.31, 2.25, 0.078, 32.0, 58.0, 0.998, 3.33, 0.54, 9.8], [6.7, 0.46, 0.24, 1.7, 0.077, 18.0, 34.0, 0.9948, 3.39, 0.6, 10.6]]}  
label: [6.0, 5.0, 6.0, 5.0, 6.0]  
prediction: [5.43, 5.15, 5.59, 5.37, 5.75]

```
In [19]: time.strftime('%Y-%m-%d %H:%M:%S')
```

Out[19]: '2021-02-10 15:00:41'

TODO: In the cell below, print the logs of the web service and delete the service

```
In [20]: logs = service.get_logs()
for line in logs.split('\n'):
    print(line)
```

```
/usr/sbin/nginx: /azureml-envs/azureml_df6ad66e80d4bc0030b6d046a4e46427/lib/libcrypto.so.1.0.0: no version information available (required by /usr/sbin/nginx)
/usr/sbin/nginx: /azureml-envs/azureml_df6ad66e80d4bc0030b6d046a4e46427/lib/libcrypto.so.1.0.0: no version information available (required by /usr/sbin/nginx)
/usr/sbin/nginx: /azureml-envs/azureml_df6ad66e80d4bc0030b6d046a4e46427/lib/libssl.so.1.0.0: no version information available (required by /usr/sbin/nginx)
/usr/sbin/nginx: /azureml-envs/azureml_df6ad66e80d4bc0030b6d046a4e46427/lib/libssl.so.1.0.0: no version information available (required by /usr/sbin/nginx)
/usr/sbin/nginx: /azureml-envs/azureml_df6ad66e80d4bc0030b6d046a4e46427/lib/libssl.so.1.0.0: no version information available (required by /usr/sbin/nginx)
2021-02-10T15:00:27,019711251+00:00 - iot-server/run
2021-02-10T15:00:27,020471966+00:00 - gunicorn/run
2021-02-10T15:00:27,022027495+00:00 - nginx/run
2021-02-10T15:00:27,036392763+00:00 - rsyslog/run
rsyslogd: /azureml-envs/azureml_df6ad66e80d4bc0030b6d046a4e46427/lib/libuuid.so.1: no version information available (required by rsyslogd)
EdgeHubConnectionString and IOTEDGE_IOTHUBHOSTNAME are not set. Exiting...
2021-02-10T15:00:27,189108021+00:00 - iot-server/finish 1 0
2021-02-10T15:00:27,190240642+00:00 - Exit code 1 is normal. Not restarting iot-server.
Starting gunicorn 19.9.0
Listening at: http://127.0.0.1:31311 (http://127.0.0.1:31311) (13)
Using worker: sync
worker timeout is set to 300
Booting worker with pid: 42
SPARK_HOME not set. Skipping PySpark Initialization.
Initializing logger
2021-02-10 15:00:28,281 | root | INFO | Starting up app insights client
2021-02-10 15:00:28,282 | root | INFO | Starting up request id generator
2021-02-10 15:00:28,282 | root | INFO | Starting up app insight hooks
2021-02-10 15:00:28,282 | root | INFO | Invoking user's init function
2021-02-10 15:00:28,283 | root | INFO | Users's init has completed successfully
/azureml-envs/azureml_df6ad66e80d4bc0030b6d046a4e46427/lib/python3.6/site-packages/sklearn/externals/joblib/__init__.py:15: FutureWarning: sklearn.externals.joblib is deprecated in 0.21 and will be removed in 0.23. Please import this functionality directly from joblib, which can be installed with: pip install joblib. If this warning is raised when loading pickled models, you may need to re-serialize those models with sklearn.externals.joblib 0.21+.
warnings.warn(msg, category=FutureWarning)
2021-02-10 15:00:28,285 | root | INFO | Skipping middleware: dbg_model_info as it's not enabled.
2021-02-10 15:00:28,285 | root | INFO | Skipping middleware: dbg_resource_usage as it's not enabled.
2021-02-10 15:00:28,286 | root | INFO | Scoring timeout is found from os.environ: 60000 ms
2021-02-10 15:00:35,885 | root | INFO | Swagger file not present
2021-02-10 15:00:35,885 | root | INFO | 404
127.0.0.1 - - [10/Feb/2021:15:00:35 +0000] "GET /swagger.json HTTP/1.0" 404 19 "-" "Go-http-client/1.1"
2021-02-10 15:00:40,560 | root | INFO | Swagger file not present
2021-02-10 15:00:40,560 | root | INFO | 404
127.0.0.1 - - [10/Feb/2021:15:00:40 +0000] "GET /swagger.json HTTP/1.0" 404 19 "-" "Go-http-client/1.1"
2021-02-10 15:00:41,127 | root | INFO | Validation Request Content-Type
2021-02-10 15:00:41,128 | root | INFO | Scoring Timer is set to 60.0 seconds
2021-02-10 15:00:41,130 | root | INFO | 200
127.0.0.1 - - [10/Feb/2021:15:00:41 +0000] "POST /score HTTP/1.0" 200 30 "-" "python-requests/2.25.1"
```

```
In [21]: try:
    service.delete()
    compute_target.delete()
except:
    print('Already deleted')
else:
    compute_target.wait_for_completion(show_output=False, is_delete_operation=True)
```

Current provisioning state of AmlCompute is "Deleting"

Current provisioning state of AmlCompute is "Deleting"

Current provisioning state of AmlCompute is "Deleting"

Current provisioning state of AmlCompute is "Deleting"

Current provisioning state of AmlCompute is "Deleting"

Current provisioning state of AmlCompute is "Deleting"

Provisioning operation finished, operation "Succeeded"