AZ – 203 Developer Training

Azure Batch Services

Agenda

- ★ Intro to Azure Batch
- ★ Manage batch jobs by using Batch Service API
- ★ Run a batch job using Azure CLI and other tools
- ★ Write code to run an Azure Batch Services batch job

Intro to Azure Batch

Azure Batch

Use Azure Batch to run large-scale parallel and high-performance computing (HPC) batch jobs efficiently in Azure. Azure Batch creates and manages a pool of compute nodes (virtual machines), installs the applications you want to run, and schedules jobs to run on the nodes. There is no cluster or job scheduler software to install, manage, or scale. Instead, you use Batch APIs and tools, command-line scripts, or the Azure portal to configure, manage, and monitor your jobs.

Use of Azure Batch

- Developers can use Batch as a platform service to build SaaS applications or client apps where large-scale execution is required.
- For example, build a service with Batch to run a Monte Carlo risk simulation for a financial services company, or a service to process many images.
- There is no additional charge for using Batch.
- You only pay for the underlying resources consumed, such as the virtual machines, storage, and networking.
- Run Parallel Workloads with Azure Batch, for example:
  - Batch works well with intrinsically parallel (called as "embarrassingly parallel") workloads.
  - Intrinsically parallel workloads are those where the applications can run independently, and each instance completes part of the work.
  - When the applications are executing, they might access some common data, but they do not communicate with other instances of the application. Intrinsically parallel workloads can therefore run at a large scale, determined by the amount of compute resources available to run applications simultaneously.

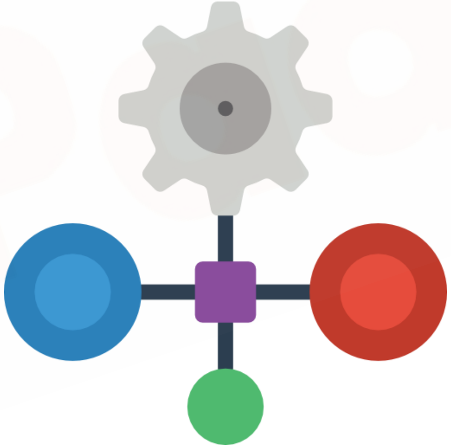
Use of Azure Batch....

- Some examples of intrinsically parallel workloads you can bring to Batch:
  - Financial risk modelling using Monte Carlo simulations
  - VFX and 3D image rendering
  - Image analysis and processing
  - Media transcoding
  - Genetic sequence analysis
  - Optical character recognition (OCR)
  - Data ingestion, processing, and ETL operations
  - Software test execution

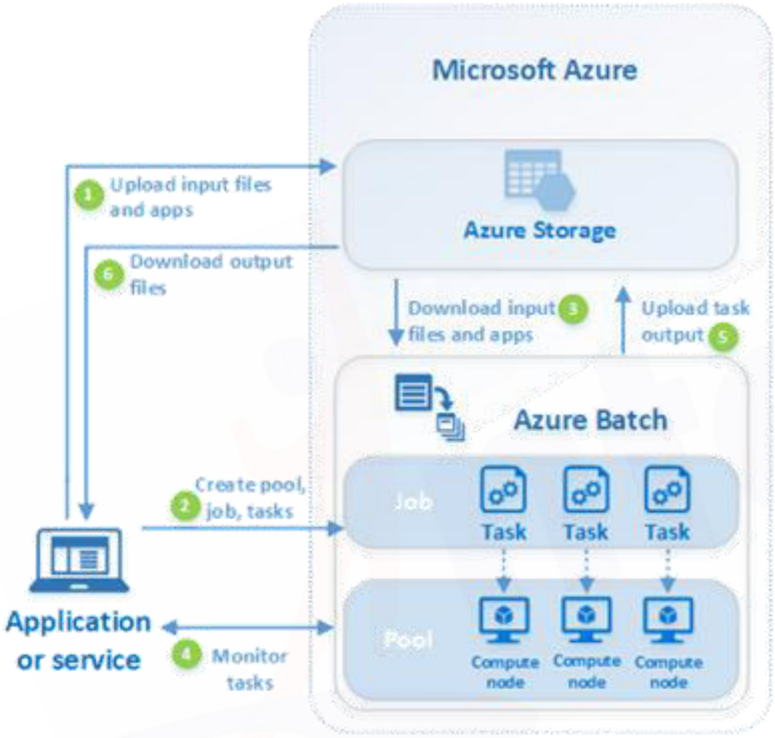
Azure Batch

How it Works?

- A common scenario for Batch involves scaling out intrinsically parallel work like:
  - Rendering of images for 3D scenes, on a pool of compute nodes.
  - This pool of compute nodes can be your "**render farm**" that provides tens, hundreds, or even thousands of cores to your rendering job.



Workflow Diagram



Step 1: Upload input files and the applications to process those files to your Azure Storage account.

Step 2: Create a Batch pool of compute nodes in your Batch account, a job to run the workload on the pool, and tasks in the job.

Step 3: Download input files and the applications to Batch.

Step 4: Monitor task execution.

Step 5: Upload task output.

Step 6: Download output files.

Batch Management REST API

Batch Management

Azure Batch enables you to run large-scale parallel and high-performance computing (HPC) applications efficiently in the cloud. It's a platform service that schedules compute-intensive work to run on a managed collection of virtual machines, and can automatically scale compute resources to meet the needs of your jobs.

- ★ The Batch Management REST API provides operations for working with the Batch service through the *Microsoft.Batch* provider.

REST Operation Groups

Operation group	Description
Application	Provides operations for working with the applications in your Batch account. An application has one or more application packages.
Application Package	Provides operations for working with application packages.
Batch Account	Provides operations for working with Batch accounts.
Certificate	Provides operations for working with certificates in your Batch account.
Location	Gets information about Batch account name availabilty and quotas in an Azure region.
Operations	Lists available operations for the Microsoft.Batch provider.
Pool	Provides operations for working with pools in your Batch account.

Running Batch Job in Azure CLI & Azure Portal

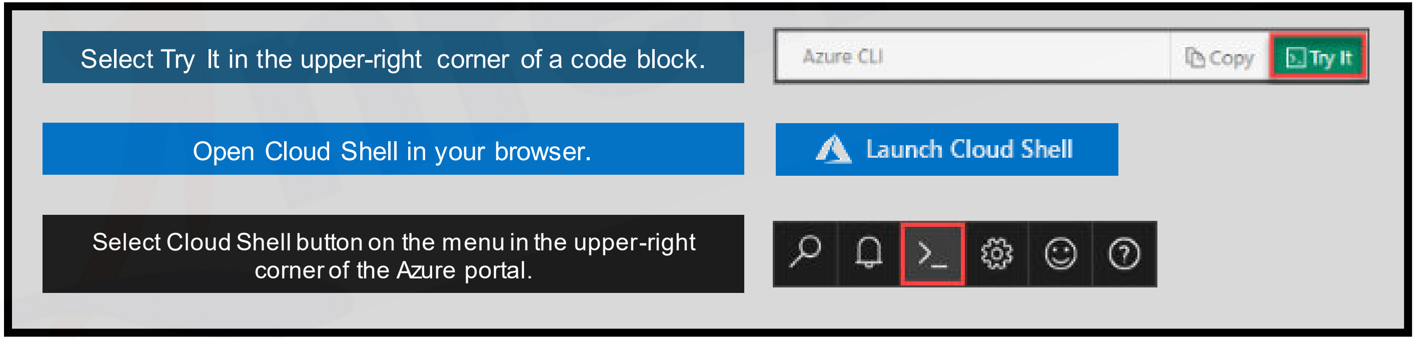
Run your first Batch job with the Azure CLI

The Azure CLI is used to create and manage Azure resources from the command line or in scripts.

Now, let's run our first batch job using the Azure CLI:

Open Azure Cloud Shell

- Common Azure tools are preinstalled and configured in Cloud Shell for you to use with your account. Select Copy to copy the code, paste it in Cloud Shell, and then press Enter to run it.
  - There are a few ways to open Cloud Shell:



Select Try It in the upper-right corner of a code block.

Open Cloud Shell in your browser.

Select Cloud Shell button on the menu in the upper-right corner of the Azure portal.

Continued…

Steps Continued.....

Create a resource group

- Create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed.
- The following example creates a resource group named *myResourceGroup* in the eastus2 location.

Azure CLI

```
az group create \
   --name myResourceGroup \
   --location eastus2
```

Continued…

Steps Continued…..

Create a storage account

- You can link an Azure Storage account with your Batch account.
- Although not required, the storage account is useful to deploy applications and store input and output data for most real-world workloads.
- Create a storage account in your resource group with the az storage account create command.

Azure CLI

```
az storage account create \

    --resource-group myResourceGroup \

    --name mystorageaccount \

    --location eastus2 \

    --sku Standard_LRS
```

Continued…

Steps Continued…..

Create a Batch account

- Create a Batch account with the az batch account create command. You need an account to create compute resources (pools of compute nodes) and Batch jobs.
  - The following example creates a Batch account named *mybatchaccount* in *myResourceGroup*, and links the storage account you created:

Azure CLI

```
az batch account create \

   --name mybatchaccount \

   --storage-account mystorageaccount \

   --resource-group myResourceGroup \

   --location eastus2
```

Continued…

Steps Continued…..

- To create and manage compute pools and jobs, you need to authenticate with Batch. Log in to the account with the az batch account login command. After you log in, your az batch commands use this account context.

Azure CLI

```
az batch account login \
    --name mybatchaccount \
    --resource-group myResourceGroup \
    --shared-key-auth
```

Continued…

Steps Continued…..

Create a Batch account

- Create a Batch account with the az batch account create command. You need an account to create compute resources (pools of compute nodes) and Batch jobs.
  - The following example creates a Batch account named *mybatchaccount* in *myResourceGroup*, and links the storage account you created:

Azure CLI

```
az batch account create \

   --name mybatchaccount  \

   --storage-account  mystorageaccount  \

   --resource-group  myResourceGroup  \

   --location eastus2
```

Continued…

Steps Continued…..

- To create and manage compute pools and jobs, you need to authenticate with Batch. Log in to the account with the az batch account login command.
- After you log in, your *az batch* commands use this account context.

Azure CLI

```
az batch account login \
    --name mybatchaccount \
    --resource-group myResourceGroup \
    --shared-key-auth
```

Continued…

Steps Continued…..

Create a pool of compute nodes

- Now that you have a Batch account, create a sample pool of Linux compute nodes using the az batch pool create command.
- The following example creates a pool named *mypool* of 2 size Standard_A1_v2 nodes running Ubuntu 16.04 LTS. The suggested node size offers a good balance of performance versus cost for this quick example.

Azure CLI

```
az batch pool create \

    --id mypool --vm-size Standard_A1_v2 \

    --target-dedicated-nodes 2 \

    --image canonical:ubuntuserver:16.04-LTS \

    --node-agent-sku-id "batch.node.ubuntu 16.04"
```

Continued…

Steps Continued…..

Batch creates the pool immediately, but it takes a few minutes to allocate and start the compute nodes. During this time, the pool is in the ***resizing*** state. To see the status of the pool, run the az batch pool show command. This command shows all the properties of the pool, and you can query for specific properties. The following command gets the allocation state of the pool:

Azure CLI

```
az batch pool show --pool-id mypool \

    --query "allocationState"
```

Command for az batch pool show:

```
az batch pool show --pool-id
                    [--account-endpoint]
                    [--account-key]
                    [--account-name]
                    [--expand]
                    [--if-match]
                    [--if-modified-since]
                    [--if-none-match]
                    [--if-unmodified-since]
                    [--select]
                    [--subscription]
```

Continue the following steps to create a job and tasks while the pool state is changing. The pool is ready to run tasks when the allocation state is *steady* and all the nodes are running.

Continued…

Steps Continued…..

Create a job

- Now that you have a pool, create a job to run on it. A Batch job is a logical group for one or more tasks.
- A job includes settings common to the tasks, such as priority and the pool to run tasks on.
- Create a Batch job by using the az batch job create command.
- The following example creates a job *myjob* on the pool *mypool*. Initially the job has no tasks.

Azure CLI

```
az batch job create \

    --id myjob \

    --pool-id mypool
```

Command for az batch job create:

```
az batch job create [--account-endpoint]
                    [--account-key]
                    [--account-name]
                    [--id]
                    [--job-manager-task-command-line]
                    [--job-manager-task-environment-settings]
                    [--job-manager-task-id]
                    [--job-manager-task-resource-files]
                    [--job-max-task-retry-count]
                    [--job-max-wall-clock-time]
                    [--json-file]
                    [--metadata]
                    [--pool-id]
                    [--priority]
                    [--subscription]
                    [--uses-task-dependencies]
```

Continued…

Steps Continued…..

Create tasks

- Now use the az batch task create command to create some tasks to run in the job.
- Each task runs a *command-line* to display the Batch environment variables on a compute node, and then waits 90 seconds.
- When you use Batch, this command line is where you specify your app or script.
- Batch provides a number of ways to deploy apps and scripts to compute nodes.

The following Bash script creates 4 parallel tasks (*mytask1* to *mytask4*).

Azure CLI

```
for i in {1..4}
do
  az batch task create \
   --task-id mytask$i \
   --job-id myjob \
   --command-line "/bin/bash -c 'printenv | grep AZ_BATCH; sleep 90s'"
done
```

Command for az batch task create:

```
az batch task create --job-id
                    [--account-endpoint]
                    [--account-key]
                    [--account-name]
                    [--affinity-id]
                    [--application-package-references]
                    [--command-line]
                    [--environment-settings]
                    [--json-file]
                    [--max-task-retry-count]
                    [--max-wall-clock-time]
                    [--resource-files]
                    [--retention-time]
                    [--subscription]
                    [--task-id]
```

The command output shows settings for each of the tasks. Batch distributes the tasks to the compute nodes.

Continued…

Steps Continued…..

View task status

- After you create a task, Batch queues it to run on the pool.
- Once a node is available to run it, the task runs.
- Use the az batch task show command to view the status of the Batch tasks. The following example shows details about *mytask1* running on one of the pool nodes.

Azure CLI

```
az batch task show \

   --job-id myjob \

   --task-id mytask1
```

The command output includes many details, but take note of the ***exitCode*** of the task command line and the ***nodeId***. An ***exitCode*** of 0 indicates that the task command line completed successfully. The ***nodeId*** indicates the ID of the pool node on which the task ran.

Continued…

Steps Continued…..

View task output

- To list the files created by a task on a compute node, use the az batch task file list command.
- The following command lists the files created by *mytask1*:

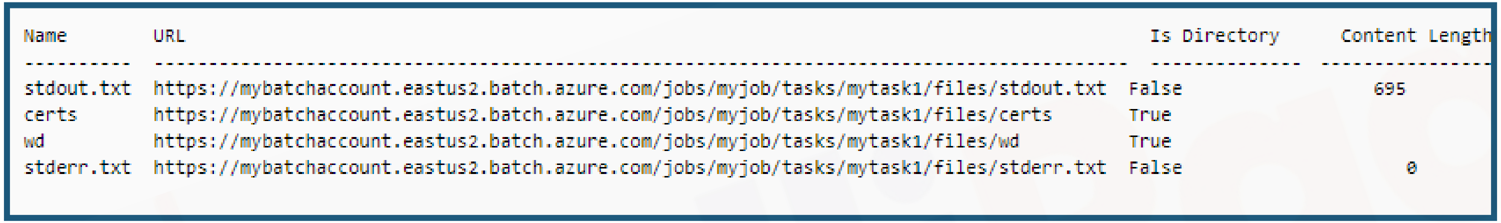
Azure CLI

```
az batch task file list \
    --job-id myjob \
    --task-id mytask1 \
    --output table
```

Continued…

Steps Continued…..

Output is similar to the following:



```
Name        URL                                                                                         Is Directory      Content Length
----------  ------------------------------------------------------------------------------------------  --------------  ----------------
stdout.txt  https://mybatchaccount.eastus2.batch.azure.com/jobs/myjob/tasks/mytask1/files/stdout.txt  False                      695
certs       https://mybatchaccount.eastus2.batch.azure.com/jobs/myjob/tasks/mytask1/files/certs       True
wd          https://mybatchaccount.eastus2.batch.azure.com/jobs/myjob/tasks/mytask1/files/wd          True
stderr.txt  https://mybatchaccount.eastus2.batch.azure.com/jobs/myjob/tasks/mytask1/files/stderr.txt  False                         0
```

To download one of the output files to a local directory, use the *az batch task file* download command. In this example, task output is in *stdout.txt*.

Azure CLI

```
az batch task file download  \
    --job-id  myjob \
    --task-id mytask1 \
    --file-path  stdout.txt \
    --destination  ./stdout.txt
```

Continued…

Steps Continued…..

- You can view the contents of stdout.txt in a text editor.
- The contents show the Azure Batch environment variables that are set on the node.
- When you create your own Batch jobs, you can reference these environment variables in task command lines, and in the apps and scripts run by the command lines.
    - For example:

```
AZ_BATCH_TASK_DIR=/mnt/batch/tasks/workitems/myjob/job-1/mytask1
AZ_BATCH_NODE_STARTUP_DIR=/mnt/batch/tasks/startup
AZ_BATCH_CERTIFICATES_DIR=/mnt/batch/tasks/workitems/myjob/job-
1/mytask1/certs
AZ_BATCH_ACCOUNT_URL=https://mybatchaccount.eastus2.batch.azure.com/
AZ_BATCH_TASK_WORKING_DIR=/mnt/batch/tasks/workitems/myjob/job-
1/mytask1/wd
AZ_BATCH_NODE_SHARED_DIR=/mnt/batch/tasks/shared
AZ_BATCH_TASK_USER=_azbatch
AZ_BATCH_NODE_ROOT_DIR=/mnt/batch/tasks
AZ_BATCH_JOB_ID=myjobl
AZ_BATCH_NODE_IS_DEDICATED=true
AZ_BATCH_NODE_ID=tvm-257509324_2-20180703t215033z
AZ_BATCH_POOL_ID=mypool
AZ_BATCH_TASK_ID=mytask1
AZ_BATCH_ACCOUNT_NAME=mybatchaccount
AZ_BATCH_TASK_USER_IDENTITY=PoolNonAdmin
```

Continued…

Steps Continued…..

Clean up resources:

You are charged for pools while the nodes are running, even if no jobs are scheduled. When you no longer need a pool, delete it with the *az batch pool delete* command. When you delete the pool, all task output on the nodes is deleted.

Azure CLI

```
az batch pool delete --pool-id mypool
```

When no longer needed, you can use the *az group delete* command to remove the resource group, Batch account, pools, and all related resources. Delete the resources as follows:

Azure CLI

```
az group delete --name myResourceGroup
```

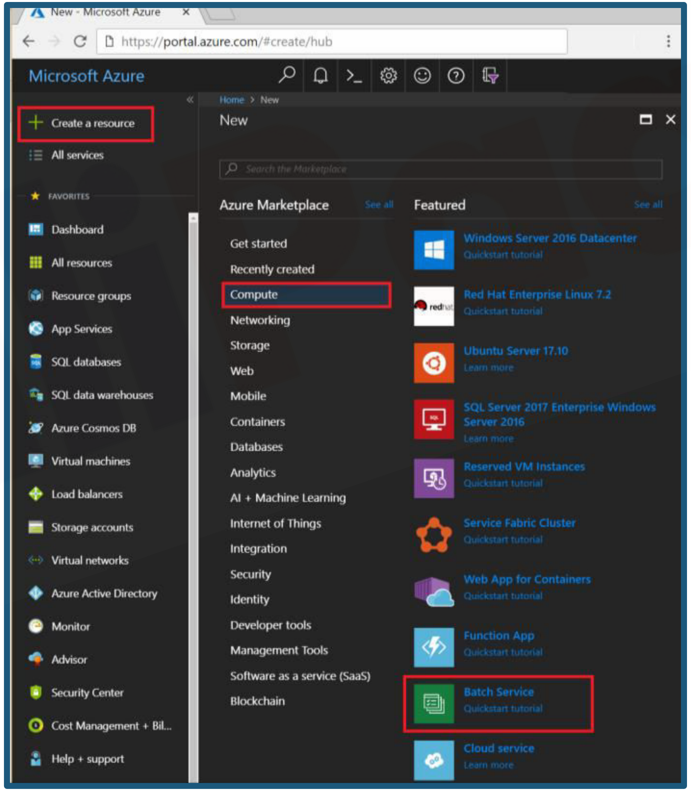
Next is running Batch job in Azure portal….

Azure Portal

Run your Batch job with the Azure Portal

You have to follow the steps as shown below:

- ★ Sign in to Azure
- ★ Create a Batch account
- ★ Create a pool of compute nodes
- ★ Create a job
- ★ Create tasks
- ★ View task output
- ★ Clean up resources

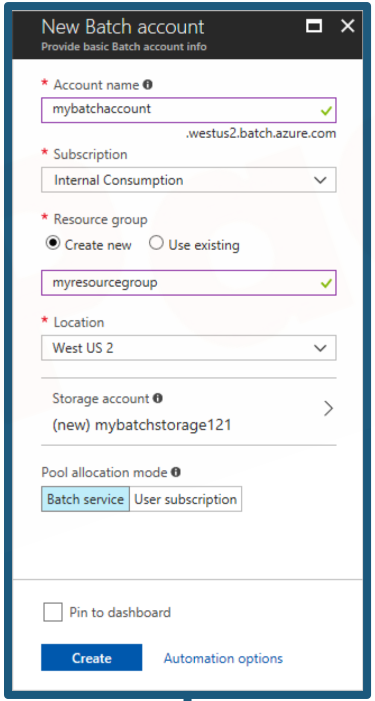


Continued…

Run your Batch job with the Azure Portal

Continued……
…

- Select *Create a resource* > *Compute* > *Batch Service.*
- Enter values for *Account name* and *Resource group.*
- The account name must be unique within the Azure Location selected, use only lowercase characters or numbers, and contain 3-24 characters.
- In *Storage account*, select an existing *storage account* or *create a new one.*
- Keep the defaults for remaining settings, and select Create to create the account.



New Batch account
Provide basic Batch account info

* Account name: mybatchaccount ✓
.westus2.batch.azure.com

* Subscription: Internal Consumption

* Resource group
● Create new ○ Use existing
myresourcegroup ✓

* Location: West US 2

Storage account: (new) mybatchstorage121

Pool allocation mode: [✓ Batch service] User subscription

□ Pin to dashboard

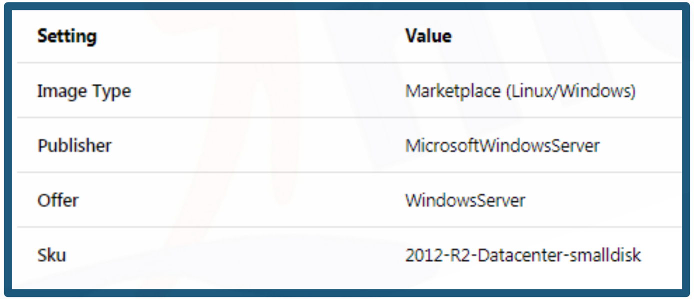
When the **Deployment succeeded** message appears, go to the **Batch account** in the portal.

Continued…

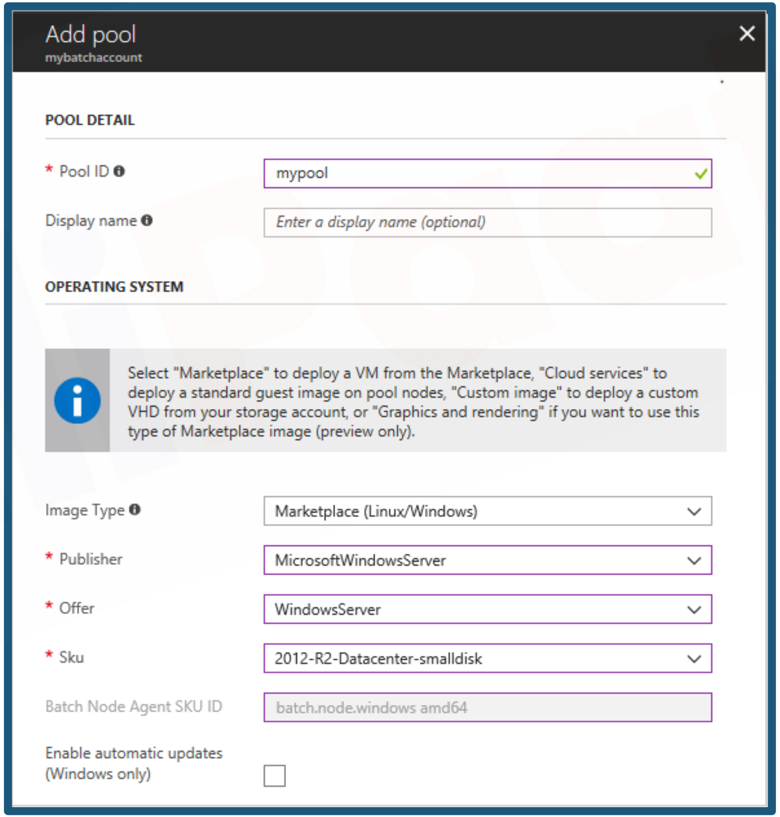
Run your Batch job with the Azure Portal

Create a pool of compute nodes

- In the Batch account, select Pools > Add.
- Enter a Pool ID called mypool.
- In Operating System, select the following settings (you can explore other options).



Setting	Value
Image Type	Marketplace (Linux/Windows)
Publisher	MicrosoftWindowsServer
Offer	WindowsServer
Sku	2012-R2-Datacenter-smalldisk



Add pool
mybatchaccount

POOL DETAIL

* Pool ID: mypool

Display name: *Enter a display name (optional)*

OPERATING SYSTEM

Select "Marketplace" to deploy a VM from the Marketplace, "Cloud services" to deploy a standard guest image on pool nodes, "Custom image" to deploy a custom VHD from your storage account, or "Graphics and rendering" if you want to use this type of Marketplace image (preview only).

Image Type: Marketplace (Linux/Windows)

* Publisher: MicrosoftWindowsServer

* Offer: WindowsServer

* Sku: 2012-R2-Datacenter-smalldisk

Batch Node Agent SKU ID: batch.node.windows amd64

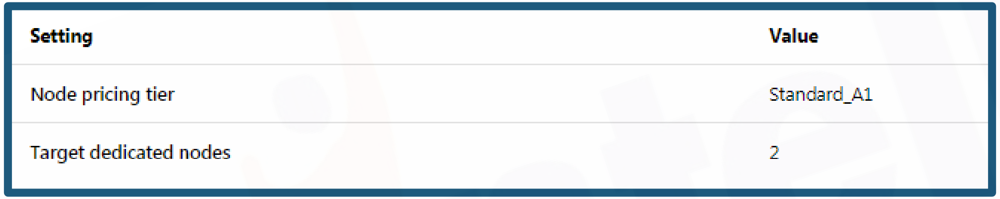
Enable automatic updates (Windows only) □

Continued…

Run your Batch job with the Azure Portal

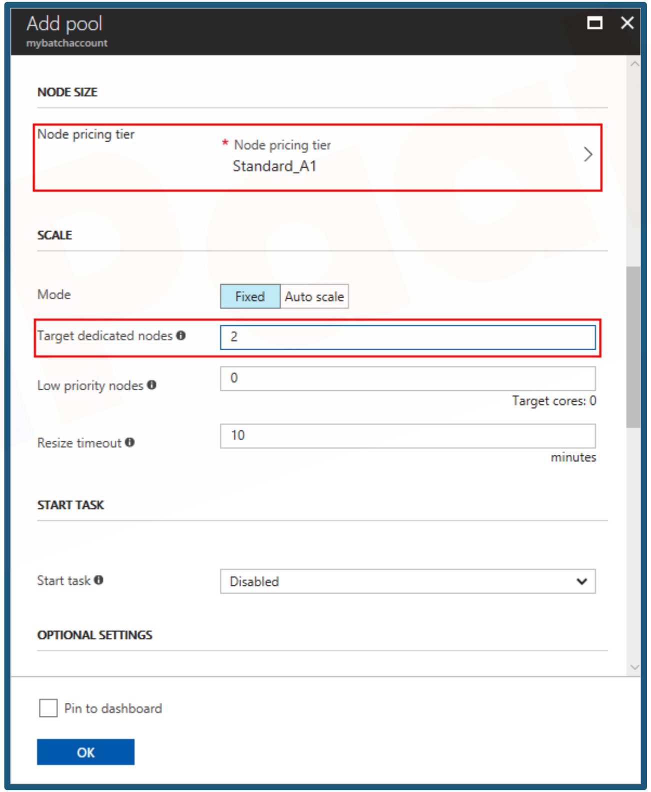
Continued......
...

★ Enter Node Size and Scale settings



Setting	Value
Node pricing tier	Standard_A1
Target dedicated nodes	2

★ Keep the defaults for remaining settings, and select **OK** to create the pool.



Add pool
mybatchaccount

NODE SIZE

Node pricing tier
* Node pricing tier
Standard_A1

SCALE

Mode: Fixed | Auto scale

Target dedicated nodes: 2

Low priority nodes: 0
Target cores: 0

Resize timeout: 10 minutes

START TASK

Start task: Disabled

OPTIONAL SETTINGS

□ Pin to dashboard

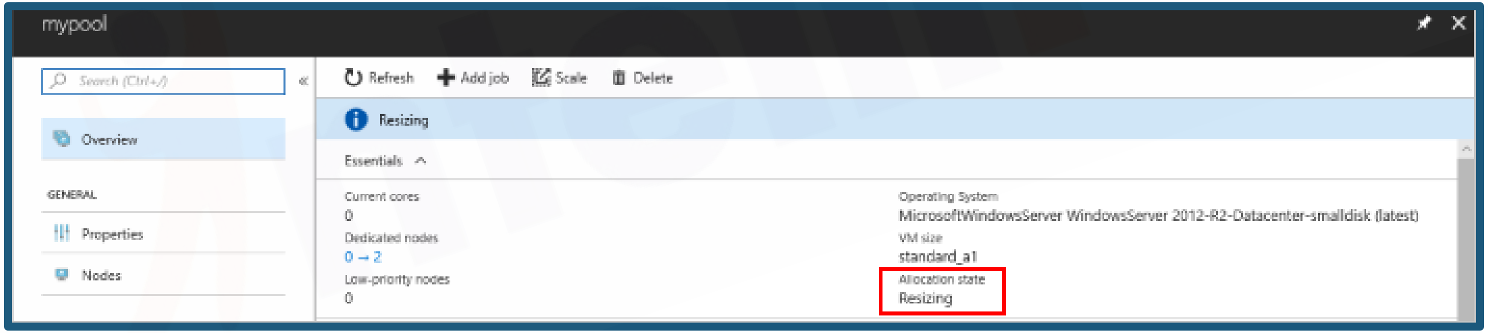
OK

Continued...

Run your Batch job with the Azure Portal

Continued………

- ★ Batch creates the pool immediately, but it takes a few minutes to allocate and start the compute nodes. During this time, the pool's **Allocation state** is Resizing.
- ★ You can go ahead and create a job and tasks while the pool is resizing.



Resizing

Current cores
0

Dedicated nodes
0 → 2

Low-priority nodes
0

Operating System
MicrosoftWindowsServer WindowsServer 2012-R2-Datacenter-smalldisk (latest)

VM size
standard_a1

Allocation state
Resizing

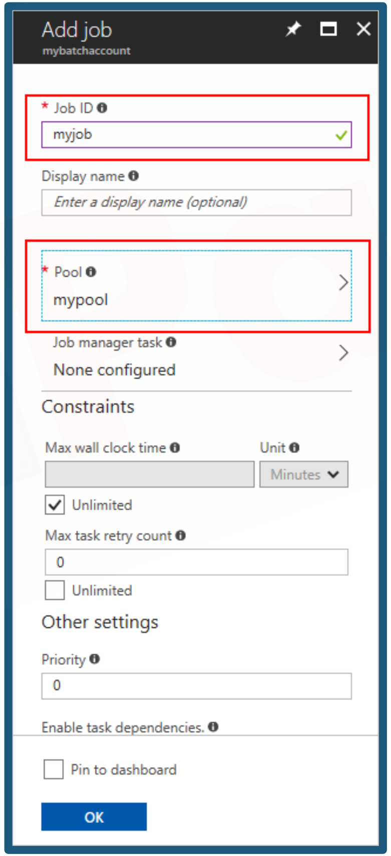
★ After a few minutes, the state of the pool is **Steady**, and the nodes start. Select **Nodes** to check the state of the nodes. When a node's state is **Idle**, it is ready to run tasks.

Continued…

Run your Batch job with the Azure Portal

Create a job

- Now that you have a pool, create a job to run on it. A Batch job is a logical group for one or more tasks. A job includes settings common to the tasks, such as priority and the pool to run tasks on. Initially the job has no tasks.
  - In the Batch account view, select **Jobs > Add**.
  - Enter a **Job ID** called *myjob*. In **Pool**, select *mypool*. Keep the defaults for the remaining settings, and select **OK**.



Add job
mybatchaccount

* Job ID: myjob

Display name: Enter a display name (optional)

* Pool: mypool

Job manager task: None configured

Constraints

Max wall clock time: Unit: Minutes

☑ Unlimited

Max task retry count: 0

□ Unlimited

Other settings

Priority: 0

Enable task dependencies.

□ Pin to dashboard

OK

After the job is created, the **Tasks** page opens.

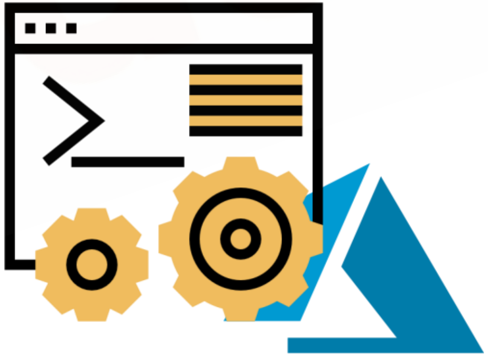
Continued…

Run your Batch job with the Azure Portal

Create tasks

- Now create sample tasks to run in the job. Typically you create multiple tasks that Batch queues and distributes to run on the compute nodes.
  - Create two identical tasks.
  - Each task runs a command line to display the Batch environment variables on a compute node, and then waits 90 seconds.

- When you use Batch, the command line is where you specify your app or script.
- Batch provides a number of ways to deploy apps and scripts to compute nodes.



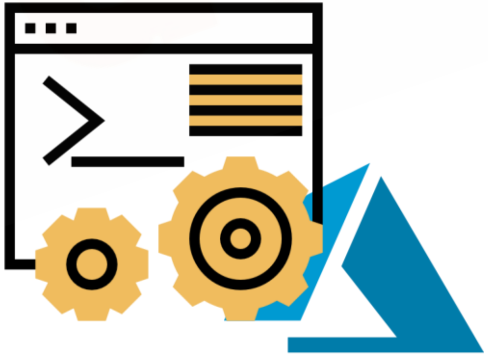
Continued…

Run your Batch job with the Azure Portal

To create the first task:

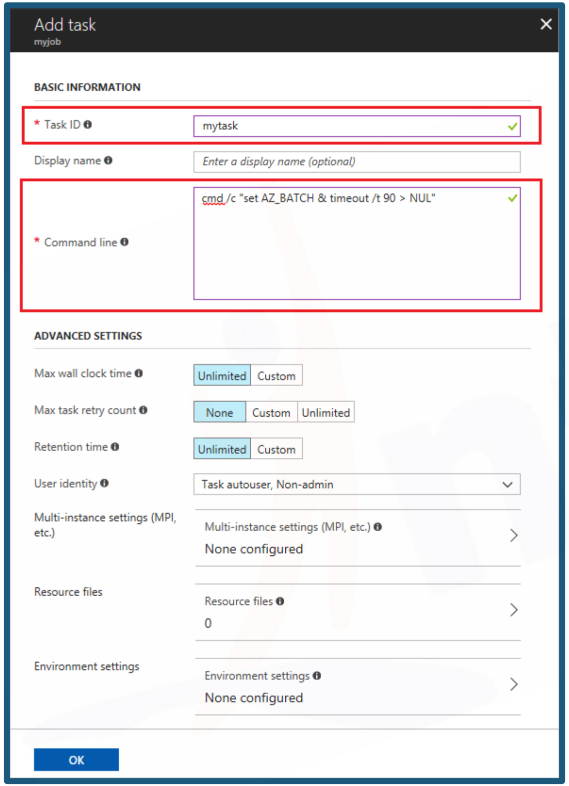
- Select **Add**.
- Enter a **Task ID** called *mytask*.
- In **Command line**, enter *cmd /c "set AZ_BATCH & timeout /t 90 > NUL"*.
- Keep the defaults for the remaining settings, and select **OK**.

- When you use Batch, the command line is where you specify your app or script.
- Batch provides a number of ways to deploy apps and scripts to compute nodes.



Continued…

Run your Batch job with the Azure Portal



Add task
myjob

BASIC INFORMATION

* Task ID	mytask
Display name	Enter a display name (optional)
* Command line	cmd /c "set AZ_BATCH & timeout /t 90 > NUL"

ADVANCED SETTINGS

Max wall clock time	[✓ Unlimited] Custom
Max task retry count	[✓ None] Custom Unlimited
Retention time	[✓ Unlimited] Custom
User identity	Task autouser, Non-admin
Multi-instance settings (MPI, etc.)	Multi-instance settings (MPI, etc.) None configured
Resource files	Resource files 0
Environment settings	Environment settings None configured

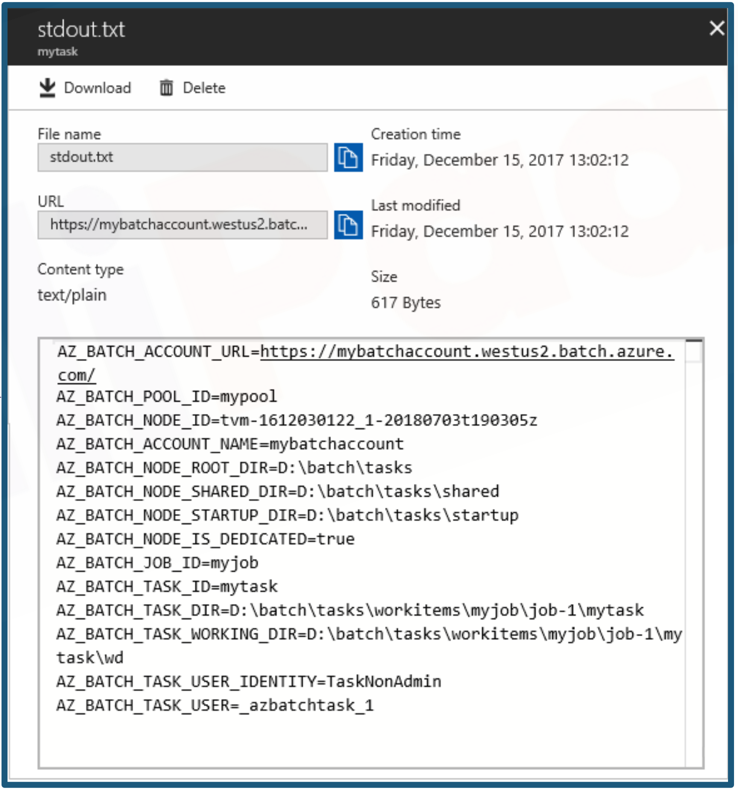
- ★ After you create a task, Batch queues it to run on the pool.
- ★ When a node is available to run it, the task runs.
- ★ To create a second task, go back to step 1.
- ★ Enter a different Task ID, but specify an identical command line. If the first task is still running, Batch starts the second task on the other node in the pool.

Continued…

Run your Batch job with the Azure Portal

View task output:

- The preceding task examples complete in a couple of minutes. To view the output of a completed task, select Files on node, and then select the file *stdout.txt*.
- This file shows the standard output of the task.
- The contents are similar to the following:



stdout.txt
mytask

File name: stdout.txt

URL: https://mybatchaccount.westus2.batc...

Content type: text/plain

Creation time: Friday, December 15, 2017 13:02:12

Last modified: Friday, December 15, 2017 13:02:12

Size: 617 Bytes

```
AZ_BATCH_ACCOUNT_URL=https://mybatchaccount.westus2.batch.azure.
com/
AZ_BATCH_POOL_ID=mypool
AZ_BATCH_NODE_ID=tvm-1612030122_1-20180703t190305z
AZ_BATCH_ACCOUNT_NAME=mybatchaccount
AZ_BATCH_NODE_ROOT_DIR=D:\batch\tasks
AZ_BATCH_NODE_SHARED_DIR=D:\batch\tasks\shared
AZ_BATCH_NODE_STARTUP_DIR=D:\batch\tasks\startup
AZ_BATCH_NODE_IS_DEDICATED=true
AZ_BATCH_JOB_ID=myjob
AZ_BATCH_TASK_ID=mytask
AZ_BATCH_TASK_DIR=D:\batch\tasks\workitems\myjob\job-1\mytask
AZ_BATCH_TASK_WORKING_DIR=D:\batch\tasks\workitems\myjob\job-1\my
task\wd
AZ_BATCH_TASK_USER_IDENTITY=TaskNonAdmin
AZ_BATCH_TASK_USER=_azbatchtask_1
```

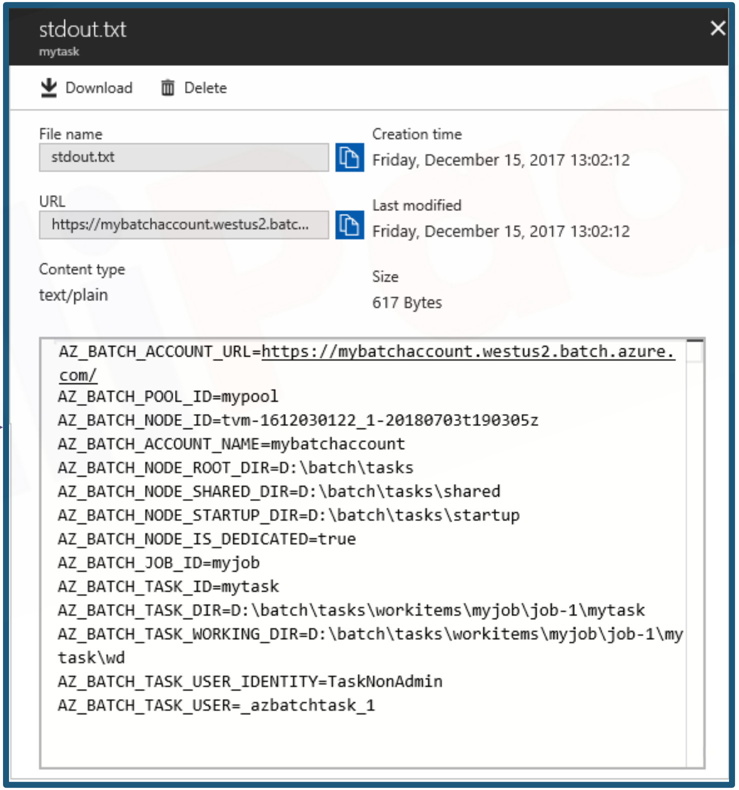
After this, Clean-Up the Resources using the Delete Command. Select the resource group for the Batch account and select Delete resource group.

Continued…

Run your Batch job with the Azure Portal

Now Clean-Up the Resources:

- The preceding task examples complete in a couple of minutes. To view the output of a completed task, select Files on node, and then select the file *stdout.txt*.
- This file shows the standard output of the task.
- The contents are similar to the following:



File name
stdout.txt

Creation time
Friday, December 15, 2017 13:02:12

URL
https://mybatchaccount.westus2.batc...

Last modified
Friday, December 15, 2017 13:02:12

Content type
text/plain

Size
617 Bytes

```
AZ_BATCH_ACCOUNT_URL=https://mybatchaccount.westus2.batch.azure.
com/
AZ_BATCH_POOL_ID=mypool
AZ_BATCH_NODE_ID=tvm-1612030122_1-20180703t190305z
AZ_BATCH_ACCOUNT_NAME=mybatchaccount
AZ_BATCH_NODE_ROOT_DIR=D:\batch\tasks
AZ_BATCH_NODE_SHARED_DIR=D:\batch\tasks\shared
AZ_BATCH_NODE_STARTUP_DIR=D:\batch\tasks\startup
AZ_BATCH_NODE_IS_DEDICATED=true
AZ_BATCH_JOB_ID=myjob
AZ_BATCH_TASK_ID=mytask
AZ_BATCH_TASK_DIR=D:\batch\tasks\workitems\myjob\job-1\mytask
AZ_BATCH_TASK_WORKING_DIR=D:\batch\tasks\workitems\myjob\job-1\my
task\wd
AZ_BATCH_TASK_USER_IDENTITY=TaskNonAdmin
AZ_BATCH_TASK_USER=_azbatchtask_1
```

Continued…

Hands-On

Hands-On

- Run your first Azure Batch job with the .NET API
  - Sign in to Azure
  - Get account credentials
  - Download the sample from Github, clone it using command (git clone https://github.com/Azure-Samples/batch-dotnet-quickstart.git)
  - Build and run the app
  - Review the code
  - Clean up resources

QUIZ

IntelliPaat

India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

sales@intellipaat.com

24X7 Chat with our Course Advisor