

# 3D video rendering on Azure

07/13/2018 • 5 minutes to read • Contributors      all

## In this article

[Relevant use cases](#)

[Architecture](#)

[Considerations](#)

[Deploy the scenario](#)

[Pricing](#)

[Related resources](#)

3D video rendering is a time consuming process that requires a significant amount of CPU time to complete. On a single machine, the process of generating a video file from static assets can take hours or even days depending on the length and complexity of the video you are producing. Many companies will purchase either expensive high end desktop computers to perform these tasks, or invest in large render farms that they can submit jobs to. However, by taking advantage of Azure Batch, that power is available to you when you need it and shuts itself down when you don't, all without any capital investment.

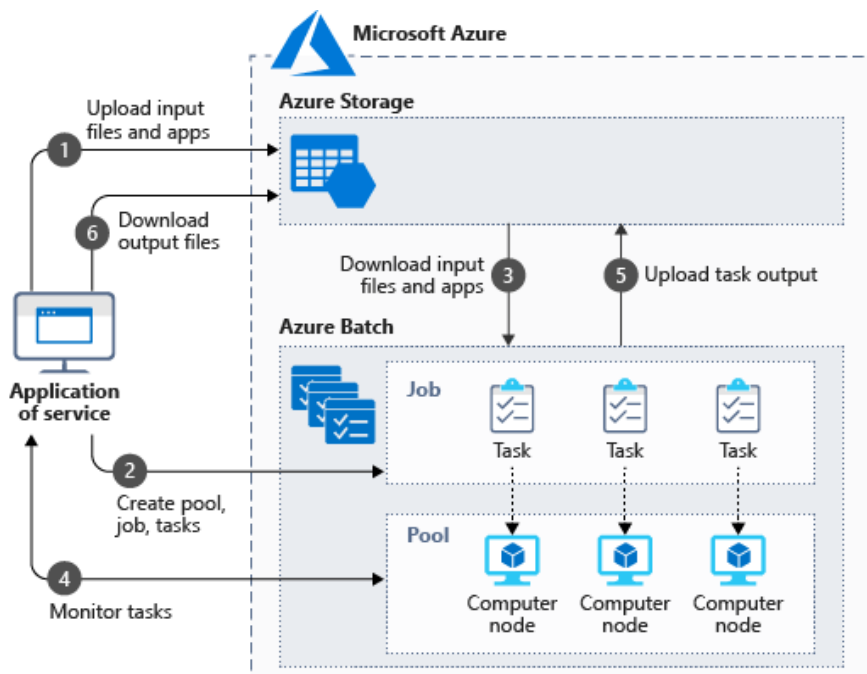
Batch gives you a consistent management experience and job scheduling, whether you select Windows Server or Linux compute nodes. With Batch, you can use your existing Windows or Linux applications, including AutoDesk Maya and Blender, to run large-scale render jobs in Azure.

## Relevant use cases

Other relevant use cases include:

- 3D modeling
- Visual FX (VFX) rendering
- Video transcoding
- Image processing, color correction, and resizing

## Architecture



This scenario shows a workflow that uses Azure Batch. The data flows as follows:

1. Upload input files and the applications to process those files to your Azure Storage account.
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.
3. Download input files and the applications to Batch.
4. Monitor task execution.
5. Upload task output.
6. Download output files.

To simplify this process, you could also use the [Batch Plugins for Maya and 3ds Max](#)

## Components

Azure Batch builds on the following Azure technologies:

- [Virtual Networks](#) are used for both the head node and the compute resources.
- [Azure Storage accounts](#) are used for synchronization and data retention.
- [Virtual machine scale sets](#) are used by CycleCloud for compute resources.

## Considerations

### Machine Sizes available for Azure Batch

While most rendering customers will choose resources with high CPU power, other workloads using virtual machine scale sets may choose VMs differently and will depend on a number of factors:

- Is the application being run memory bound?
- Does the application need to use GPUs?
- Are the job types embarrassingly parallel or require infiniband connectivity for tightly coupled jobs?
- Require fast I/O to access storage on the compute Nodes.

Azure has a wide range of VM sizes that can address each and every one of the above application requirements, some are specific to HPC, but even the smallest sizes can be used to provide an effective grid implementation:

- [HPC VM sizes](#) Due to the CPU bound nature of rendering, Microsoft typically suggests the Azure H-Series VMs. This type of VM is built specifically for high end computational needs, they have 8 and 16 core vCPU sizes available, and features DDR4 memory, SSD temporary storage, and Haswell E5 Intel technology.
- [GPU VM sizes](#) GPU optimized VM sizes are specialized virtual machines available with single or multiple NVIDIA GPUs. These sizes are designed for compute-intensive, graphics-intensive, and visualization workloads.
- NC, NCv2, NCv3, and ND sizes are optimized for compute-intensive and network-intensive applications and algorithms, including CUDA and OpenCL-based applications and simulations, AI, and Deep Learning. NV sizes are optimized and designed for remote visualization, streaming, gaming, encoding, and VDI scenarios using frameworks such as OpenGL and DirectX.
- [Memory optimized VM sizes](#) When more memory is required, the memory optimized VM sizes offer a higher memory-to-CPU ratio.
- [General purposes VM sizes](#) General-purpose VM sizes are also available and provide balanced CPU-to-memory ratio.

## Alternatives

If you require more control over your rendering environment in Azure or need a hybrid implementation, then CycleCloud computing can help orchestrate an IaaS grid in the cloud. Using the same underlying Azure technologies as

Azure Batch, it makes building and maintaining an IaaS grid an efficient process. To find out more and learn about the design principles use the following link:

For a complete overview of all the HPC solutions that are available to you in Azure, see the article [HPC, Batch, and Big Compute solutions using Azure VMs](#)

## Availability

Monitoring of the Azure Batch components is available through a range of services, tools, and APIs. Monitoring is discussed further in the [Monitor Batch solutions](#) article.

## Scalability

Pools within an Azure Batch account can either scale through manual intervention or, by using a formula based on Azure Batch metrics, be scaled automatically. For more information on scalability, see the article [Create an automatic scaling formula for scaling nodes in a Batch pool](#).

## Security

For general guidance on designing secure solutions, see the [Azure Security Documentation](#).

## Resiliency

While there is currently no failover capability in Azure Batch, we recommend using the following steps to ensure availability if there is an unplanned outage:

- Create an Azure Batch account in an alternate Azure location with an alternate Storage Account
- Create the same node pools with the same name, with zero nodes allocated
- Ensure Applications are created and updated to the alternate Storage Account
- Upload input files and submit jobs to the alternate Azure Batch account

# Deploy the scenario

## Create an Azure Batch account and pools manually

This scenario demonstrates how Azure Batch works while showcasing Azure Batch Labs as an example SaaS solution that can be developed for your own customers:

[Azure Batch Masterclass](#)

## Deploy the components

The template will deploy:

- A new Azure Batch account
- A storage account
- A node pool associated with the batch account
- The node pool will be configured to use A2 v2 VMs with Canonical Ubuntu images
- The node pool will contain zero VMs initially and will require you to manually scale to add VMs

Click the link below to deploy the solution.



## Pricing

The cost of using Azure Batch will depend on the VM sizes that are used for the pools and how long these VMs are allocated and running, there is no cost associated with an Azure Batch account creation. Storage and data egress should be taken into account as these will apply additional costs.

The following are examples of costs that could be incurred for a job that completes in 8 hours using a different number of servers:

- 100 High-Performance CPU VMs: [Cost Estimate](#)

100 x H16m (16 cores, 225 GB RAM, Premium Storage 512 GB), 2 TB Blob Storage, 1-TB egress

- 50 High-Performance CPU VMs: [Cost Estimate](#)

50 x H16m (16 cores, 225 GB RAM, Premium Storage 512 GB), 2 TB Blob Storage, 1-TB egress

- 10 High-Performance CPU VMs: [Cost Estimate](#)

10 x H16m (16 cores, 225 GB RAM, Premium Storage 512 GB), 2 TB Blob Storage, 1-TB egress

### Pricing for low-priority VMs

Azure Batch also supports the use of low-priority VMs in the node pools, which can potentially provide a substantial cost saving. For more information, including a price comparison between standard VMs and low-priority VMs, see [Azure Batch Pricing](#).

#### Note

Low-priority VMs are only suitable for certain applications and workloads.

## Related resources

[Azure Batch Overview](#)

[Azure Batch Documentation](#)

[Using containers on Azure Batch](#)