



Azure HPC CycleCloud

Managing High-
Performance Computing
(HPC) Workloads

Azure CycleCloud

- Cloud HPC orchestration tool for managing clusters
- Enables deployment, scaling, and management of HPC workloads

Key Features

- Automated cluster provisioning
- Policy-based autoscaling
- Cost and resource optimization
- Integration with popular schedulers
 - Slurm, PBS, LSF, etc.
- Secure access and compliance controls

Use Cases

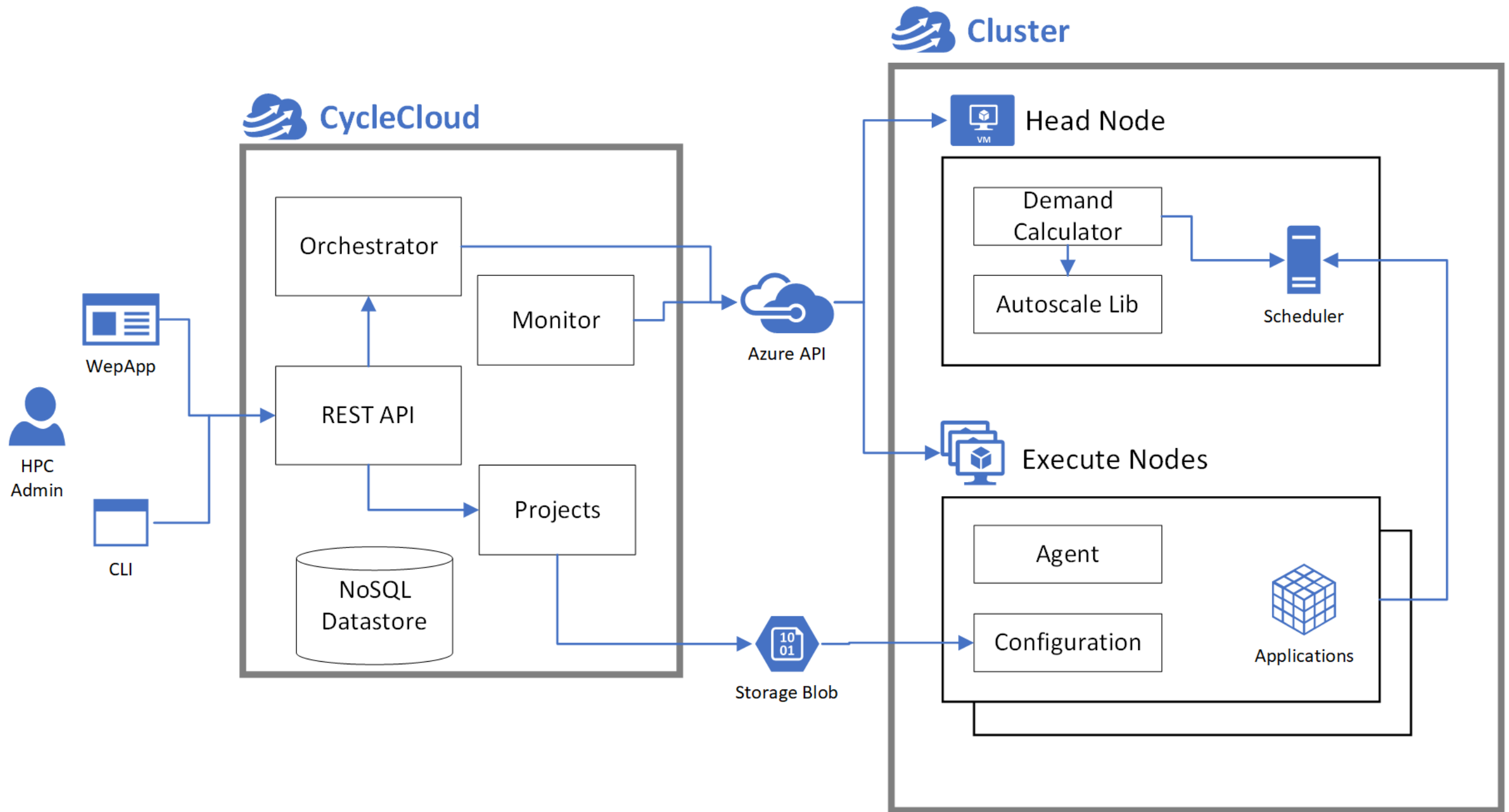
- Computational fluid dynamics (CFD)
- Machine learning and AI workloads
- Financial modeling and simulations
- Genomics and bioinformatics
- Oil and gas exploration

How it works

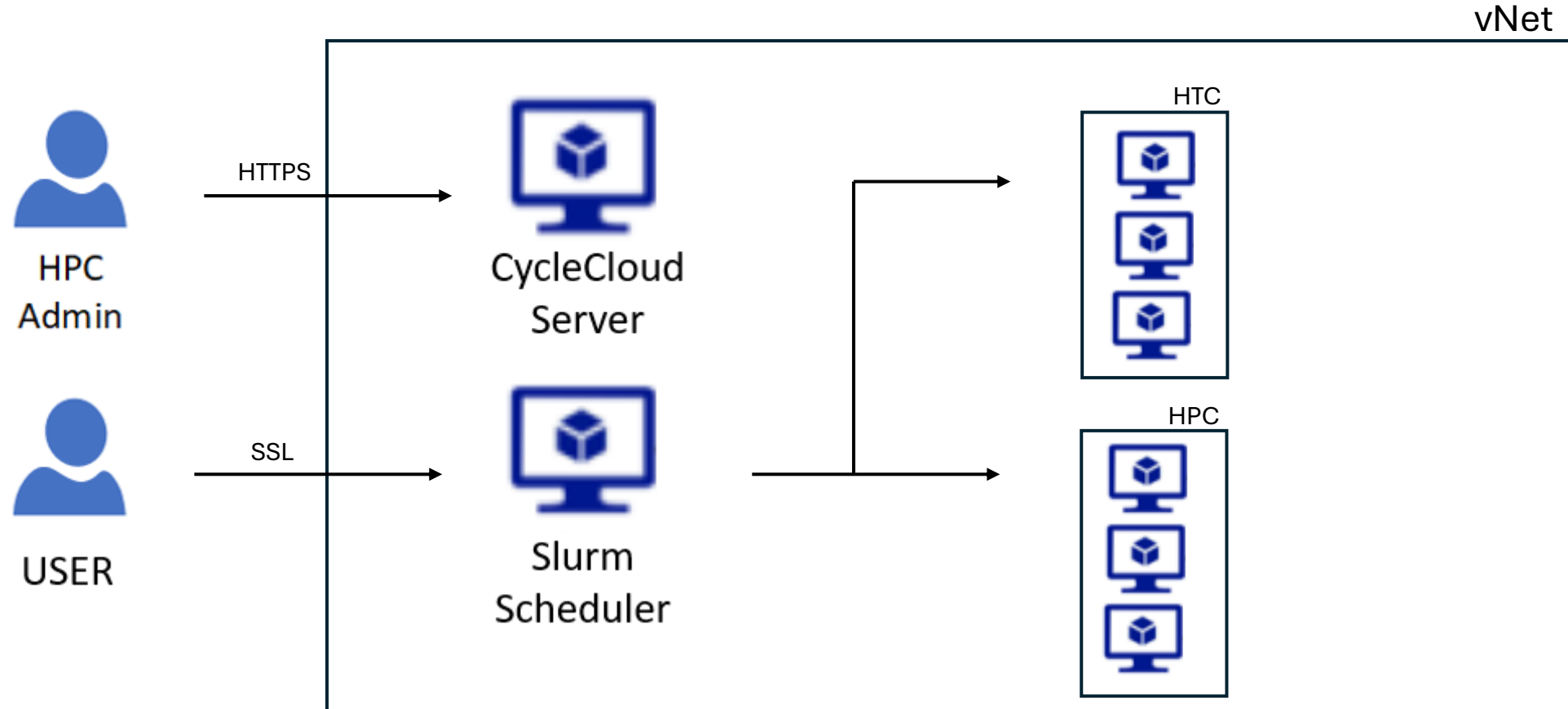
- Define and configure clusters
- Automate job scheduling and resource scaling
- Monitor and optimize performance
- Pay only for used resources

Architecture Overview

- Azure CycleCloud orchestrating HPC clusters
- Connection to Azure Compute (VMs, Spot instances)
- Integration with storage solutions (Blob, Azure Files, Lustre)
- Connectivity with on-prem and hybrid setups



CycleCloud Cluster



Deployment Process

1. Install Azure CycleCloud
2. Configure cluster templates
3. Define job scheduling and scaling policies
4. Deploy and monitor workloads

VM Types

VM Type	Characteristics	Description
HPC	High Performance Compute	Tightly coupled MPI jobs. Utilizes InfiniBand support
HTC	High Throughput Computing	Embarrassingly parallel jobs
GPU	Graphics Processing Unit	GPU-based workloads
DYN	Dynamic Partitions	Multiple VM sizes withing a single partition
Login	Login node	Place where users can login, edit files, view job results and submit jobs
Scheduler	Scheduler node	Job scheduler

VM Naming Convention

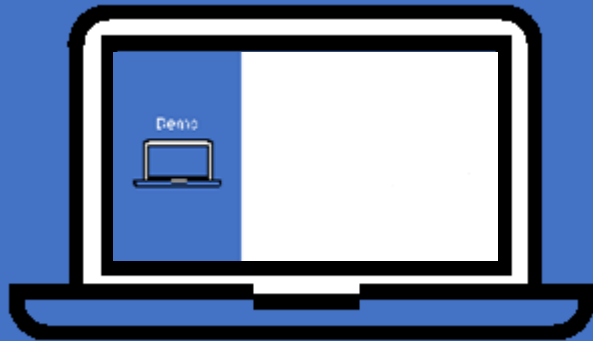
Value	Explanation
Family	Indicates the VM Family Series
*Subfamily	Used for specialized VM differentiations only
# of vCPUs	Denotes the number of vCPUs of the VM
*Constrained vCPUs	Used for certain VM sizes only. Denotes the number of vCPUs for the constrained vCPU capable size
Additive Features	<p>Lower case letters denote additive features, such as:</p> <ul style="list-style-type: none"> a = AMD-based processor b = Block Storage performance d = diskful (that is, a local temp disk is present); this feature is for newer Azure VMs, see Ddv4 and Ddsv4-series i = isolated size l = low memory; a lower amount of memory than the memory intensive size m = memory intensive; the most amount of memory in a particular size p = ARM Cpu t = tiny memory; the smallest amount of memory in a particular size s = Premium Storage capable, including possible use of Ultra SSD (Note: some newer sizes without the attribute of s can still support Premium Storage, such as M128, M64, etc.) C = Confidential NP = node packing
*Accelerator Type	Denotes the type of hardware accelerator in the specialized/GPU SKUs. Only the new specialized/GPU SKUs launched from Q3 2020 have the hardware accelerator in the name.
Version	Denotes the version of the VM Family Series

Examples

Standard_D4as_v4

Standard_F2s_v2

Demo



Azure CycleCloud

Creating and managing an HPC cluster using Azure CycleCloud

Slurm Basic Commands

Commands	Syntax	Description
squeue	squeue -u <username>	Show status of scheduling queue
sinfo	sinfo	Info on nodes
scancel	scancel <job-id>	Cancel running/queued job
sbatch	sbatch <job-id>	Submit script to queue
scontrol	scontrol show job <job-id>	View/modify Slurm configuration
sacct	sacct	Current & previous jobs
srun	srun <resource-parameters>	Run job interactively



Slurm Batch Directives

Directive	Description
#SBATCH --time 1:00:00	time of a job (or -t) in hour:minute:second
#SBATCH --partition=name	partition to use (or -p)
#SBATCH --nodes=1	number of nodes (or -N)
#SBATCH --ntasks=32	total number of tasks (cpu cores) (or -n)
#SBATCH --mem=128GB	memory per node
#SBATCH -o slurm-%j.out-%N	name for stdout; %j is job#, %N node
#SBATCH -e slurm-%j.err-%N	name for stderr; %j is job#, %N node

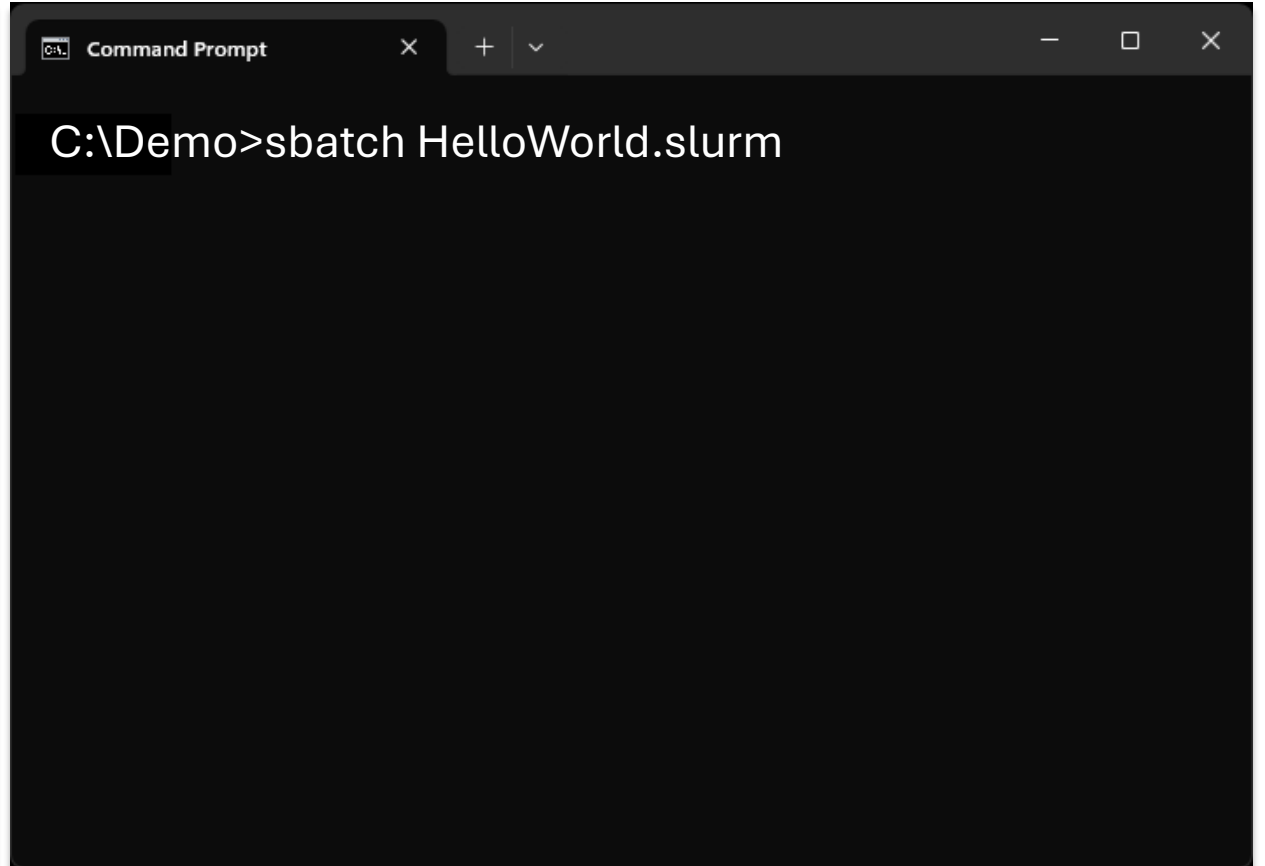
Slurm Example Script

HelloWorld.slurm

```
#!/bin/bash

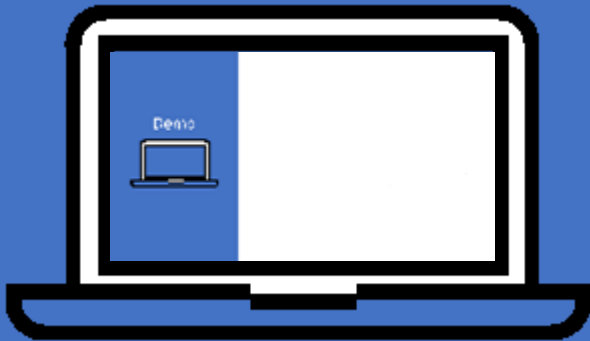
#SBATCH --job-name=HelloWorld
#SBATCH --partition=htc
#SBATCH --time=00:10:00
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=1G
#SBATCH --output=/shared/home/%u/slurm_%j.out
#SBATCH --error=/shared/home/%u/slurm_%j.err

# Your script goes here
sleep 30
echo "Hello World!"
```

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt" with standard window controls. The command prompt shows the directory "C:\Demo" and the command "sbatch HelloWorld.slurm" entered. The command has been executed, but no output is visible on the screen.

```
C:\Demo>sbatch HelloWorld.slurm
```

Demo



Azure CycleCloud

Slurm Example Script

Best Practices

- Optimize resource allocation to reduce costs
- Use spot instances for cost savings
- Monitor performance metrics with Azure Monitor
- Implement security best practices (RBAC, VNETs, etc.)

Summary

- .Azure CycleCloud simplifies HPC cluster management
- .Enables efficient, scalable, and cost-effective cloud computing
- .Ideal for enterprises and research institutions



Resources

Azure CycleCloud Documentation

<https://learn.microsoft.com/en-us/azure/cyclecloud/?view=cyclecloud-8>

Deploy Azure CycleCloud Workspace for Slurm

<https://learn.microsoft.com/en-us/azure/cyclecloud/qs-deploy-ccws?view=cyclecloud-8>

Link to Slides in GitHub

<https://github.com/danielecolon/Azure-CycleCloud>

