

# Deploy Agisoft PhotoScan on Azure

With Avere vFXT for Azure or BeeGFS

Paulo Marques da Costa

AzureCAT

December 2018

# Contents

Introduction to PhotoScan on Azure .....	4
Deployment templates .....	4
About Avere vFXT for Azure .....	5
About BeeGFS .....	5
Prerequisites .....	5
Avere vFXT prerequisites .....	6
BeeGFS prerequisites.....	6
Architecture with Avere vFXT storage.....	7
Architecture with BeeGFS storage.....	9
How the templates work.....	10
Special parameters.....	11
Linked templates.....	12
Active Directory configuration .....	12
Jumpbox configuration .....	12
Resource group configuration.....	13
Deploy the solution .....	13
Start Azure Cloud Shell and set up a storage account .....	13
Deploy the Avere vFXT storage option .....	14
Deploy the BeeGFS storage option .....	15
Clone and customize the PhotoScan project .....	16
Deploy PhotoScan.....	17
Optional: Deploy custom storage .....	17
Optional: Deploy Windows PhotoScan worker nodes .....	17
Benchmark results .....	18
Download a sample dataset.....	20
Learn more .....	20
Appendix.....	20
BeeGFS parameter files .....	20
PhotoScan parameter files .....	25

## List of figures

Figure 1. PhotoScan on Azure with Avere vFXT for storage.....	7
Figure 2. PhotoScan on Azure with BeeGFS for storage.....	9
Figure 3. The Ortho_and_DEM_without_GCPs dataset from Agisoft used to test this deployment. ....	18
Figure 4. This 3 GB image is the result of the benchmarking test. ....	18
Figure 5. Benchmarking results comparing BeeGFS (in blue) to Avere vFXT (in orange). .	19

Authored by Paulo Marques da Costa. Edited by Nanette Ray. Reviewed by AzureCAT.

© 2018 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

# Introduction to PhotoScan on Azure

PhotoScan photogrammetry software enables you to create 3D images from 2D photographs. The realistic-looking models, textures, and other digital assets are used to create visual effects for computer games and movies and to document archaeological digs and cultural heritage sites. They are also used in geographic information systems (GIS) and in industrial applications that require indirect measurements of objects of various scales.

This solution was created for an organization that processes aerial images. They wanted to run their PhotoScan workload on Azure and take advantage of virtual machines (VMs) with GPUs so that they could scale their environment up or down as needed. They also needed terabytes of storage, but performance mattered more. This solution is designed to use either Avere vFXT for Azure, a filesystem caching solution for data-intensive, high-performance computing (HPC) tasks, or BeeGFS for the back-end storage.

To take full advantage of Avere vFXT or BeeGFS, we selected Linux operating systems for the worker nodes. PhotoScan provides both Windows and Linux binaries. When we worked on their deployment and wrote this document, the BeeGFS Windows driver was available in beta version only and not recommended for production workloads.

## Deployment templates

Two templates deploy the end-to-end PhotoScan environment on Azure. First, use one of the following templates to create the storage back end:

- [Avere vFXT](#) template deploys a high-performance storage solution on Azure based on NFSv3 and comprised of at least four VMs. One VM is the Avere vFXT deployment controller and is used mainly to deploy the Avere vFXT cluster. Later it can be used to scale the vFXT cluster up or down by adding or removing nodes. For more information, see [What is Avere vFXT for Azure](#).

---

**NOTE:** Some manual configuration steps are required before deployment as described later in this guide and in the [repository](#). The vFXT cluster will export two namespaces: *home*, which is used as the HPC user's home folder, and *data*, which holds the PhotoScan projects.

---

- [BeeGFS template](#) deploys an alternative storage solution based on the BeeGFS parallel virtual file system on Azure. Similarly, this will also provide an export that the PhotoScan Linux worker nodes can mount using BeeGFS client package and a NFSv4-based export to be used as home folder for HPC user. The deployment includes Samba on separate VMs to allow the PhotoScan scheduler VM (Windows based) to access the project files and coordinate the HPC cluster. For details about BeeGFS performance on Azure, see [Parallel Virtual File Systems on Microsoft Azure – Part 1: Overview](#).

Next, use the [PhotoScan template](#) to deploy the Azure components needed by the workload. This template installs PhotoScan on related VMs—scheduler and nodes—to form the HPC cluster.

The [Avere vFXT](#), [BeeGFS](#), and [PhotoScan](#) templates are available on GitHub. They are designed to be deployed together in this order: storage, then PhotoScan. However, you can also use the PhotoScan template with a different storage solution. For example, you can customize the

templates or configure the environment manually after it is deployed without using these storage solutions. This document assumes you are installing the end-to-end solution using either the Avere vFXT or BeeGFS template.

## About Avere vFXT for Azure

Earlier this year, Microsoft signed an agreement to acquire Avere Systems. Avere vFXT for Azure supports high-performance cloud computing for file-based workloads. As a file caching solution, Avere vFXT optimizes storage performance and efficiencies by using object storage and traditional network-attached storage (NAS) to minimize the costs typically associated with storing large datasets close to Azure VMs.

Avere vFXT identifies the active data stored in Azure Blob and tiers it into a solid-state drive (SSD) cache in your Azure compute system while the job is running. When processing is complete, the data is written back into Blob. For more information, see the Avere vFXT [documentation](#).

## About BeeGFS

BeeGFS is a Linux-based, hardware-independent parallel file system that is designed for high-performance and high-throughput environments. BeeGFS is free to use and offers commercial support. It consists of four main components, each running as a multithreaded daemon process in the Linux operating system:

- Management service and attached management target disk
- OSS and attached storage target disk
- MDS and attached metadata target disk
- File system client

One advantage of the BeeGFS client service is that it provides a normal mount point that your applications can use to directly access the BeeGFS storage system. The BeeGFS User Guide illustrates this [system architecture](#).

# Prerequisites

Each template is a set of JSON files, each containing parameters that you must customize before deployment as instructed below. The templates also contain script files that automate deployment of all components within the VMs.

Before deploying the solution, review these assumptions and requirements:

- Make sure to have Contributor rights at the Azure subscription level.
- Confirm the download URLs before deploying and editing, if required, any templates. The templates install the required software based on the download URLs available at the time of this writing.
- Make sure to have enough Azure quota for GPU-based VMs. By default, the templates require the following minimum number of cores per VM family size depending on the storage solution you choose:

Avere vFXT		BeeGFS	
VM size	Cores	VM size	Cores
Standard_NC6_v2	120	Standard_NC6_v2	120
Standard_DS_v3	56	Standard_DS_v3	80
Standard_DS_v2	8	Standard_DS_v2	12
Standard_A_v2	1		

Use Azure Key Vault to store the following secrets and their values:

Parameter	Value
adminPassword	The password used by the account for your domain administrator and local Windows administrator.
activationCode	A valid PhotoScan activation code. You can request a trial version from the <a href="#">Agisoft website</a> .

For a quick step-by-step guide on how to create an Azure Key Vault and a secret, see [Quickstart: Set and retrieve a secret from Azure Key Vault using the Azure portal](#).

For more information about Azure Key Vault, see [What is Azure Key Vault?](#)

## Avere vFXT prerequisites

- Configure Avere vFXT so it can be integrated with PhotoScan template. See [Avere vFXT post-deployment configuration steps for PhotoScan template](#).
- Note that Avere vFXT nodes supports two VM sizes, Standard\_D16S\_v3 (default) or Standard\_E32S\_v3.
- Confirm the number of nodes you need using the Avere Node Count parameter. The default value is three.
- As above, confirm the cache size for the vFXT nodes.

## BeeGFS prerequisites

The templates assume that BeeGFS 7.1 is being used. If a new version becomes available, you can edit the setup scripts to refer to that version instead.

# Architecture with Avere vFXT storage

When you deploy the PhotoScan template using Avere vFXT storage, the following architecture is created on Azure.

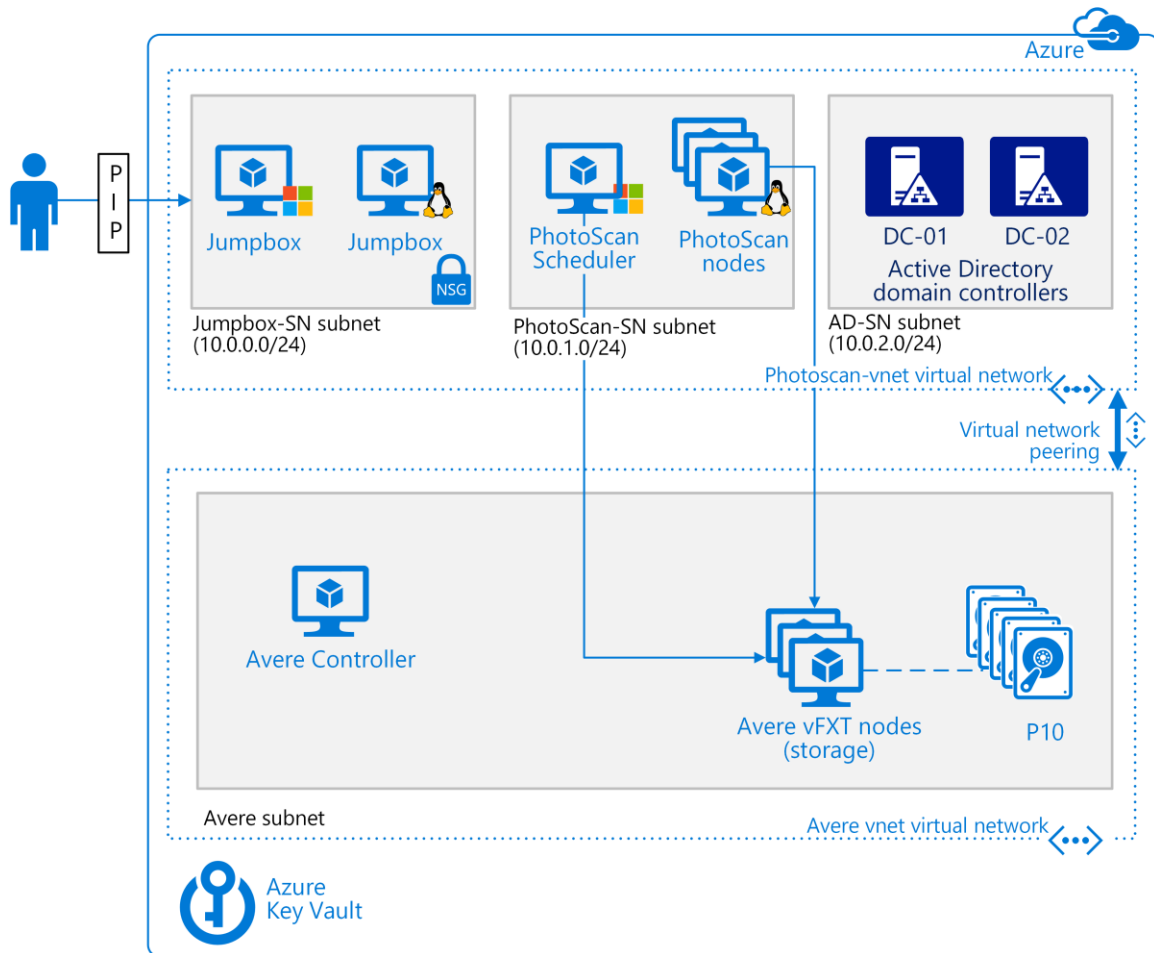


Figure 1. PhotoScan on Azure with Avere vFXT for storage.

This solution deploys the following components:

- A Standard\_D8S\_v3 VM running Windows to serve as the PhotoScan scheduler (head) node.
- Five Standard\_NC24s\_v2 VMs (GPUs) running CentOS\_7.5 to act as the PhotoScan processing nodes.
- Two Active Directory domain controllers. These VMs become domain controllers through the Azure PowerShell DSC (Desired State Configuration) VM extension. These domain controllers also provide DNS name resolution. For Avere vFXT, x number of A records are added—for example, *vfx*, where each A record points to the IP address of each Avere vFXT node. In this setup, all VMs use the round-robin pattern to access vFXT exports.
- Two Standard\_DS2\_v2 VMs used as administrator jumpboxes—one running Windows and one running Linux, which is optional.

- One network security group (NSG) associated with the jumpbox-SN subnet. This NSG allows ports 3389 and 22 access to those VMs.
- Peered virtual networks with the Avere vFXT virtual networks.
- A Standard\_A1\_v2 VM running Ubuntu 18.04 LTS as the Avere Controller. This VM is responsible for executing the vfx.py script that installs the cluster. This VM can be used later to add or remove cluster nodes and to destroy the cluster as well.
- At least three Standard\_D16s\_v3 VMs, one for each of the Avere vFXT nodes based on AvereOS\_V5.0.2.1 that form the vFXT cluster.

All Linux VMs are configured using the Azure Custom Script Extension for Linux. Some Windows VMs also use Custom Script Extensions to complete their configuration.

---

**NOTE:** Make note of the management IP address and also the IP address range of the cluster nodes that is an output of the Avere vFXT template. You will need this information later during configuration if you are using this storage option.

---



# Architecture with BeeGFS storage

When you deploy the PhotoScan template using BeeGFS storage, the following architecture is created on Azure.

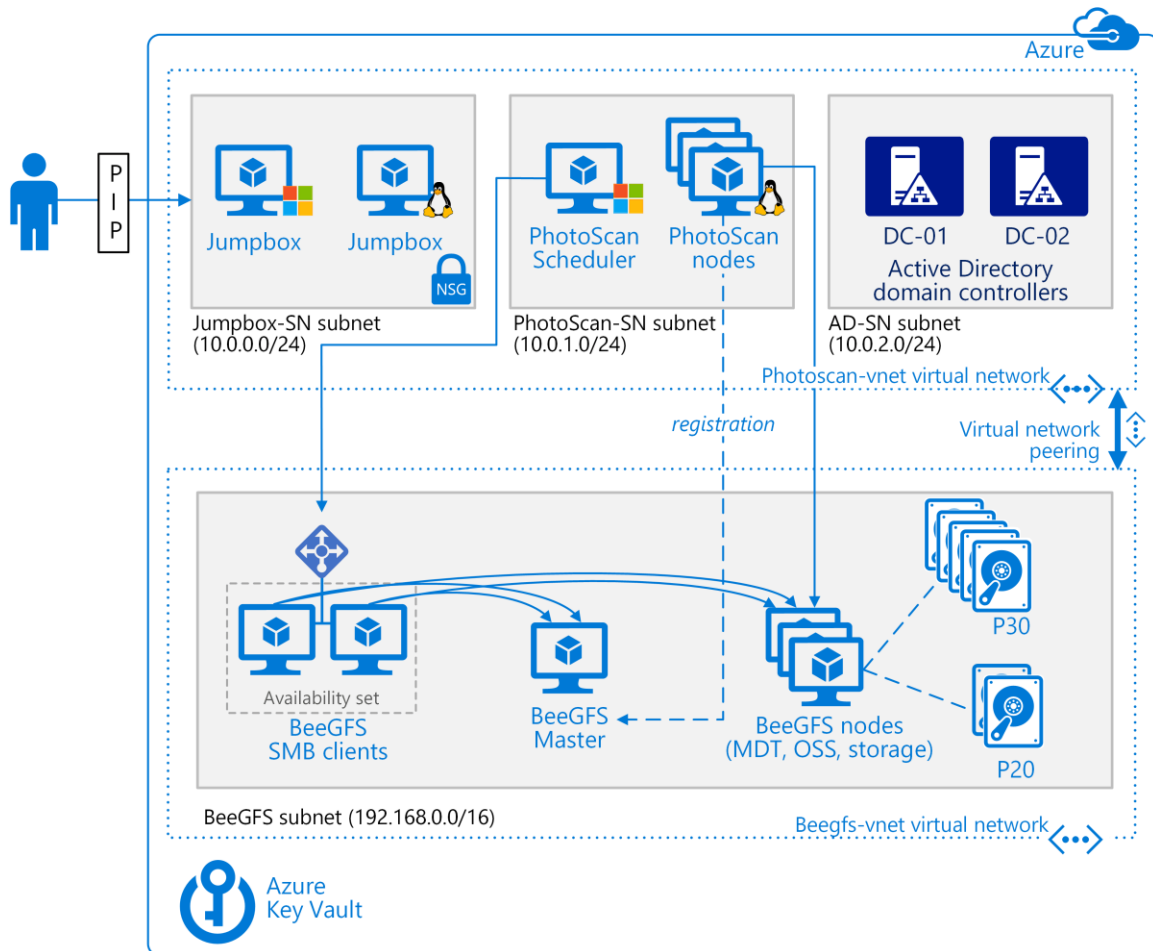


Figure 2. PhotoScan on Azure with BeeGFS for storage.

This solution deploys the following components:

- A Standard\_D8S\_v3 VM running Windows to serve as the PhotoScan scheduler (head) node.
- Five Standard\_NC24s\_v2 VMs (GPUs) running CentOS\_7.5 to act as the PhotoScan processing nodes.
- Two Active Directory domain controllers. These VMs become domain controllers through the Azure PowerShell DSC (Desired State Configuration) VM extension. The template creates a single A record for the virtual IP address of the load balancer that exposes the two BeeGFS SMB VMs and an A record for BeeGFS Master VM.
- Two Standard\_DS2\_v2 VMs used as administrator jumpboxes—one running Windows and one running Linux, which is optional.

- One NSG associated with the jumpbox-SN subnet. This NSG allows ports 3389 and 22 access to those VMs.
- Peered virtual networks with the BeeGFS virtual network.
- A Standard\_D4s\_v2 VM running CentOS 7.5 as the BeeGFS management server.
- Four Standard\_D16s\_v3 VMs running CentOS 7.5 used for the BeeGFS metadata server (MDS) and object storage server (OSS). Each VM includes both roles. The MDS role uses two P20 Premium SSD with 512 GiB per disk. The OSS role uses five P30 Premium SSDs with 1,024 GiB (1 TiB) per disk. Both disk groups use software RAID 0 and locally redundant storage (LRS).
- Two Standard\_D4s\_v3 VMs used for the BeeGFS file system clients, which expose an SMB storage exposing the BeeGFS mount point on those VMs to the scheduler VM, which is Windows OS.
- One load balancer for the BeeGFS clients that exposes the BeeGFS mount point via SMB. This configuration does not provide automatic reconnection to the SMB share as does Windows Storage Spaces Direct, which provides continuous availability in case of a node failure. Although this solution offers limited high availability, it was deemed a better alternative than a blind DNS round-robin configuration.

All Linux VMs are configured using the Azure Custom Script Extension for Linux. Some Windows VMs also use Custom Script Extensions to complete their configuration.

## How the templates work

If you use the Avere vFXT template, it sets up the backend storage based on NFSv3 by deploying a vFXT Controller VM and executing a custom script extension to execute the `installvfxt.sh` python script that creates the vFXT cluster attached to Azure Blob Storage.

Alternatively, the BeeGFS template sets up the backend storage based on a driver-dependent protocol. This template also installs Samba servers (called *clients* in our solution) so that the storage can be consumed by both Windows clients (through SMB) and Linux clients (through the BeeGFS driver). After this infrastructure is deployed, the PhotoScan template completes the solution.

In the PhotoScan template, the `azuredeploy.json` file contains all the linked templates used to execute these deployments in the following order:

1. A virtual network is deployed if one does not already exist (`/common/networking.json`).
2. The Active Directory domain controllers are deployed (`/common/activeDirectory.json`).
3. The following are deployed in parallel:
  - Jump boxes (`/common/jumpbox.json`).
  - Windows PhotoScan scheduler (`/photoscanHead.json`).
  - Linux nodes, if the Linux worker node type was selected (`/photoscanLinuxNodes.json`).
  - Windows nodes, if the Windows worker node type was selected (`/photoscanWindowsNodes.json`).

4. If **useNfsStorage** is *yes* and **useBeeGFSStorage** is *no* and **workerNodesType** is *linux*, the following occurs:
- The virtual network that is deployed is peered with the Avere vFXT virtual network.
  - NFS for Windows is installed and configured, and a drive is mapped at user sign in on Sscheduler VM.
  - The nfs-utils tool is installed on the clients.
  - Export is mounted and configured to mount during reboots via fstab.
  - HPC user is configured to use the common home export so it can share the SSH keys.

For more information about this type of network configuration, see [Virtual network peering](#) in the Azure documentation.

---

**NOTE:** To make use of all Windows NFS client configuration that the template provides, please use the mapped drive within PhotoScan application instead of UNC. UNC does not use all the configured items.

---

5. If **useBeeGFSStorage** is *yes* and **useNfsStorage** is *no* and **workerNodesType** is *linux*, the following occurs:
- The virtual network that is deployed is peered with the BeeGFS virtual network.
  - The BeeGFS client is installed on all Linux nodes (/useBeeGfsStorage.json).
  - HPC user is configured to use a NFSv4 export for its home directory so it can share SSH keys.

## Special parameters

All the templates include the **\_artifactsLocation** and **\_artifactsLocationSasToken** parameters. Their values are generated automatically and are used to store related scripts, PowerShell DSC files, and linked templates. These components must be accessible to Azure Resource Manager. The Deploy-AzureResourceGroup.sh script generates these components, tells Resource Manager where to find the files, and provides an authentication token that authorizes the download of these files during deployment. The scripts, PowerShell DSC configuration, and other files are downloaded directly from the VMs when specific VM extensions are installed.

The templates also use parameters that configure a special user and group on all Linux VMs:

- The **hpcUser** parameter establishes the user that owns and has access to all the files created on the storage solution you selected. This user also starts the PhotoScan node process in the Linux VMs, so that it does not have to be executed as root.
- The **hpcGroup** parameter establishes the group that owns and has access to all the files created on the storage solution you selected.

During the deployment process, the BeeGFS template attempts to configure high availability for BeeGFS, which requires a shutdown of all the MDS and OSS nodes. This operation is performed by the HPC User, which is granted shutdown rights on those VMs. The high availability configuration is controlled by the **deployHaConfiguration** parameter in the deploy-beegfs-nodes.json template file. The default value, *yes*, causes the install\_beegfs\_ha.sh script to execute

and complete this task. If *no*, the high availability configuration for BeeGFS is not enabled.

---

**NOTE:** This operation is not needed for Avere vFXT since it comes configured for high availability by default.

---

In both the Avere vFXT and BeeGFS templates, the home folder of the HPC user is created in the location indicated by the `sharedNfsStorageHpcUserHomeFolder` attribute. Initially the HPC user uses a local folder on each VM such as `/home/hpcuser`. Later, this user is configured to use a shared mount point. This way, the same home folder is provided for the HPC user on all VMs, allowing it to share one SSH key across all VMs.

## Linked templates

The PhotoScan template solution is comprised of several linked templates that are stored in the project's root folder and library (the `/library` folder). For example, these templates are used to deploy the VMs and network interfaces required by the other templates such as `jumpbox.json`.

---

**NOTE:** If you want to customize this solution, be aware that some deployments defined in the `azuredeploy.json` file are initiated using the `"condition": true` attribute. For example, resources such as the jump boxes are deployed in this way. To prevent any of these linked templates from executing, set the condition to `false`. Other linked templates use a more elaborate condition. To change the outcome of these expressions, take extra care in analyzing the condition.

---

To make the templates more flexible, arrays are used to describe resources such as VMs. For example, instead of copying the same VM resource over and over in the template, the **Variables** section in a template defines an array, and the `copy` property, in conjunction with `copyIndex()` function of the resource, is used to access an individual item in the array and deploy it with different values as needed without having to create another resource entirely.

## Active Directory configuration

The PhotoScan template deploys two `Standard_DS2_v2` VMs as domain controllers in an availability set. Each VM has an extra data disk (configured as drive F: ) where the Active Directory database, log files, and `SYSVOL` folder are stored. This infrastructure enables all the Windows VMs to be domain-joined and rely on Active Directory for central authentication and provides custom DNS services for all VMs.

## Jumpbox configuration

The PhotoScan template deploys one Windows and, optionally, one Linux jumpbox. The Windows jumpbox is mandatory, because PhotoScan scheduler runs on a Windows-based VM to support the PhotoScan GUI. The Windows jumpbox provides the connection to the scheduler using remote desktop protocol (RDP).

Deployment of the Linux jumpbox is controlled by the `deployLinuxJumpbox` parameter. The default value is *no*. If you do not deploy a Linux jumpbox, you can download PuTTY or any other SSH software on the Windows jumpbox to manage all the Linux VMs.

## Resource group configuration

For organizations with specific deployment requirements, the PhotoScan template supports a multiple resource-group deployment model. The advantage is that you can deploy individual components to different resource groups. For example, you can deploy the virtual networks to the resource group required by your networking team, such as *Networking-rg*.

To take advantage of multiple resource groups when deploying the PhotoScan template, edit the `azuredeploy.json` template as follows:

- Pass a *no* value for the `useSingleResourceGroup` parameter. (The default value is *yes*.) You can either create a parameter directly in the `azuredeploy.parameters.json` parameter file or, when deploying, use the sample `Deploy-AzureResourceGroup.sh` script and include the `-m` switch.
- In the **Variables** section, modify the object called `resourceGroups`. This object has attributes that represent the components to deploy and the resource group to deploy to. To use a multiple resource group deployment, change the default value of `photoscan-rg` to the resource group you want. You can customize the resource groups for virtual networks (`networkingRg`), the jumpboxes (`jumpboxRg`), Active Directory Domain controllers (`adRg`), and the PhotoScan scheduler and nodes (`photoscanRg`).

---

**NOTE:** As of this writing, Azure deployments support up to five different resource groups. Make sure one of the resource groups you define is the same as the one you identify in the `-g` parameter when executing the deployment script. Otherwise, the script creates an extra resource group just to perform the deployment.

---

## Deploy the solution

To deploy the end-to-end PhotoScan environment, you must execute four scripts in order. Exact steps follow, but in general, do the following:

1. Clone related repositories (Avere vFXT or BeeGFS, and PhotoScan) and edit the parameter files for your environment.
2. Deploy Avere vFXT or BeeGFS storage.
3. If using Avere vFXT, complete the post-deployment configuration steps described later in this guide.
4. Deploy PhotoScan.

### Start Azure Cloud Shell and set up a storage account

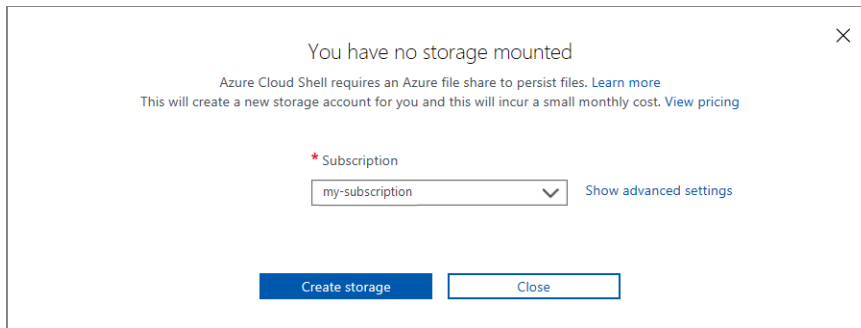
Cloud Shell machines are temporary and require a new or existing Azure Files share to be mounted as clouddrive to persist your files.

1. Go to <https://shell.azure.com> and log on using the account associated with your Azure subscription.
2. If you have access to more than one Azure Active Directory (Azure AD) tenant, on the **Select a directory** screen, select the Azure directory that is associated with your Azure subscription.

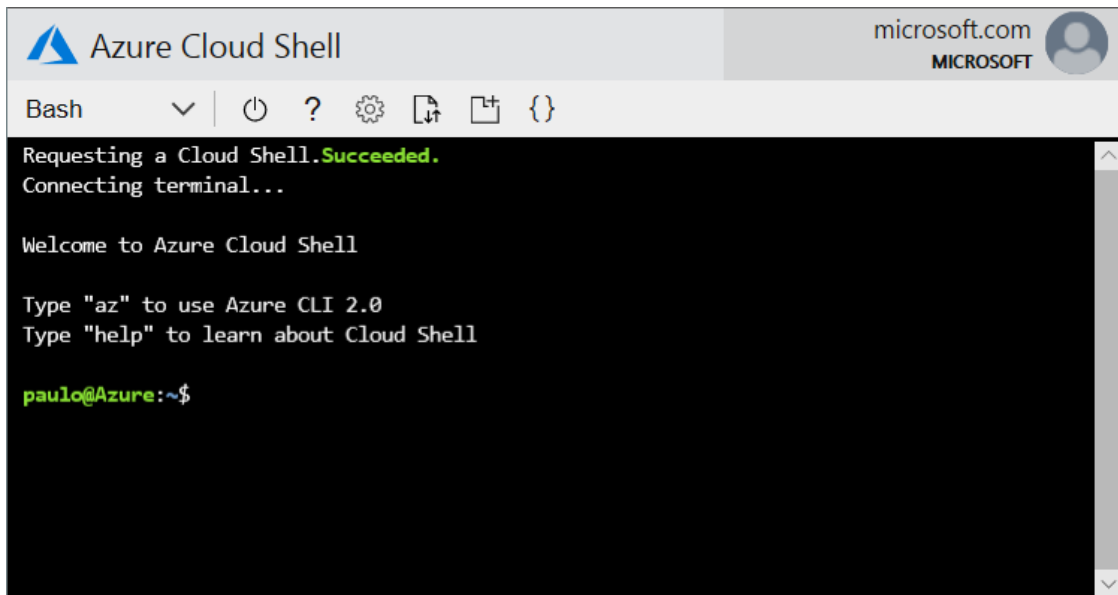
- If prompted, select **Bash (Linux)**. If PowerShell starts instead, in the environment drop-down box, select **Bash**.



- Select a subscription. The initialization process will create a storage account here.



- Click **Create storage**. The Azure Cloud Shell command prompt appears:



- If you have multiple subscriptions, make sure you select the correct one before executing next steps:

```
az account set --subscription=<your subscription id>
```

## Deploy the Avere vFXT storage option

The Avere product team at Microsoft maintains an automated deployment template that is the basis for the steps that follow. These steps assume that you have never deployed an Avere vFXT cluster in your Azure AD tenant and subscription.

Before proceeding, note the following:

- Use this procedure only if you are not installing BeeGFS.

- Make sure the Avere virtual network address space does not conflict with the one you are deploying on PhotoScan or the virtual network peering process will fail.
- If you have deployed an Avere vFXT cluster in your Azure subscription, do *not* repeat the setup steps as outlined in step 1 below. To reuse an existing service principal, follow the [step-by-step](#) process provided in the product team's instructions so that you can obtain its information and append a new password to be used in your deployment.

Follow these steps to deploy this solution:

1. Deploy the Avere vFXT cluster by following the steps in the [Experimental: Avere vFXT controller and vFXT - ARM template deployment](#).
2. Execute the mandatory post-deployment steps to ensure that the PhotoScan template can be integrated successfully. See [Avere vFXT post-deployment configuration steps for Photoscan template](#).
  - These steps require VPN or ExpressRoute connectivity to the virtual network where the Avere vFXT cluster is deployed.
  - Alternatively, you can [create a temporary SSH tunnel](#) with Avere Controller VM, and then continue with the post-deployment steps.

## Deploy the BeeGFS storage option

Use this procedure only if you are not installing Avere vFXT.

Clone and customize the BeeGFS project

1. At the Cloud Shell command prompt, change directories to the **clouddrive** directory to make sure changes are persisted in the storage account just created:

```
cd ~/clouddrive
```

2. Clone the repository:

```
git clone https://github.com/paulomarquesc/beegfs-template.git
```

3. Go to the beegfs-template folder:

```
cd beegfs-template
```

4. Review the three parameter files and do the following:

- Modify the values for **adminUsername** and **sshKeyData** (required).
- Make any other changes as necessary.
- Keep values consistent across templates.
- For a complete list of parameters, see Appendix: [BeeGFS parameter files](#).

---

**NOTE:** Instead of `vi`, you can use Visual Studio Code in Cloud Shell to quickly change the parameters. Type `code` in the Cloud Shell bash command prompt to launch it.

---

## Deploy BeeGFS

Execute the deployment scripts in the order shown, changing the command-line arguments as needed for your Azure environment. You must provide the storage account name (`storageaccountname`) and its resource group, or the script will create a new storage account each time it runs.

1. Deploy-beegfs-master.json:

```
./Deploy-AzureResourceGroup.sh -g beegfs-rg -l eastus -t deploy-beegfs-master.json -p deploy-beegfs-master-parameters.json -s storageaccountname -r storage-account-rg
```

2. Deploy-beegfs-nodes.json:

```
./Deploy-AzureResourceGroup.sh -g beegfs-rg -l eastus -t deploy-beegfs-nodes.json -p deploy-beegfs-nodes-parameters.json -s storageaccountname -r storage-account-rg
```

3. Deploy-clients.json:

```
./Deploy-AzureResourceGroup.sh -g beegfs-rg -l eastus -t deploy-clients.json -p deploy-clients-parameters.json -s storageaccountname -r storage-account-rg
```

---

**NOTE:** You can use a script to automate these three steps into a single script execution. The `beegfs-template/devtools` folder contains a sample script that ties all the deployments together in one command line. Copy the `Deploy-all.sh` script to the `beegfs-template` root folder, change its values, and then execute this script for the whole BeeGFS environment.

---

## Clone and customize the PhotoScan project

1. If your Cloud Shell session has expired, log on as described earlier. At the Cloud Shell command prompt, change directories to the **clouddrive** directory to make sure changes are persisted in the storage account just created:

```
cd ~/clouddrive
```

2. Clone the repository:

```
git clone https://github.com/paulomarquesc/photoscan-template.git
```

3. Go to the `photoscan-template` folder:

```
cd photoscan-template
```

4. Review `azuredeploy.parameters.json` and change the values as necessary. For a complete list of parameters, see Appendix: [PhotoScan parameter files](#).



## Deploy PhotoScan

After you have set up the storage infrastructure—whether Avere vFXT or BeeGFS—you can execute the PhotoScan deployment script, `azuredeploy.json`.

Run the following command, changing `storageaccountname` and other arguments to the appropriate values:

```
./Deploy-AzureResourceGroup.sh -g photoscan-rg -l eastus -s
storageaccountname -r storage-account-rg -v mykeyvault -p
azuredeploy.parameters.averes.json
```

Note that the `-p` parameter indicates which parameter file to use for your deployment. Two are provided—one for **Avere vFXT** (shown in the command line example), and the other (`azuredeploy.parameters.beegfs.json`) for **BeeGFS**.

Each parameter file contains both general parameters in addition to those specific to its solution. Parameters related to Avere vFXT start with `nfs`; those related to BeeGFS starts with `beegfs`. The `azuredeploy.json` template contains all parameters and their default values.

### Optional: Deploy custom storage

You can deploy PhotoScan with the storage solution of your choice, including GlusterFS, Lustre, or Windows Storage Spaces Direct. To do this, you must manually edit your cloned template to work with the chosen storage solution.

In `azuredeploy.parameters.json`, change the value of `useBeeGfsStorage` and `useNfsStorage` to `no` as shown:

```
"useBeeGfsStorage":{
  "value":"no"
},
"useNfsStorage":{
  "value":"no"
},
```

### Optional: Deploy Windows PhotoScan worker nodes

You can deploy the PhotoScan worker nodes with the Windows operating system instead of Linux, the default option. When choosing Windows nodes, storage integration options are not executed. You must manually integrate the environment with an existing storage solution or customize the PhotoScan template to provide such automation.

In `azuredeploy.parameters.json`, change the value of `workerNodesType` to `windows` as shown:

```
" workerNodesType":{
  "value":"windows"
},
```

For consistency, you can change the value of `useBeeGfsStorage` and `useNfsStorage` to `no`. With this option, the template makes the decision to deploy the storage integration template based on the `"linux" workNodesType` value.

# Benchmark results

To test our implementation, we used a dataset from Agisoft, shown in Figure 3. Both the Avere vFXT and BeeGFS benchmark tests used four Standard\_E32s\_v3 VMs for the storage. The benchmark results shown in the following table reveal that the final image (Figure 4) took a total of 1,341 seconds to create (at medium resolution) when using Avere vFXT storage based on the VM sizes and families described in this guide.



Figure 3. The Ortho\_and\_DEM\_without\_GCPs dataset from Agisoft used to test this deployment.



Figure 4. This 3 GB image is the result of the benchmarking test.

The following table compares the benchmarking results from the two storage solutions:

Test	Job time on Avere vFXT (in seconds)	Job time on BeeGFS (in seconds)
Matching	366	330
Alignment	99	84
Depth maps generation	177	119
Dense cloud generation	123	117
Processing DEM (digital elevation model)	47	59
Processing orthomosaic	345	426
Export raster image	184	317
<b>Total</b>	<b>1,341</b>	<b>1,452</b>

Figure 5 shows the results graphically to demonstrate how similar the storage solutions are in this scenario with PhotoScan.

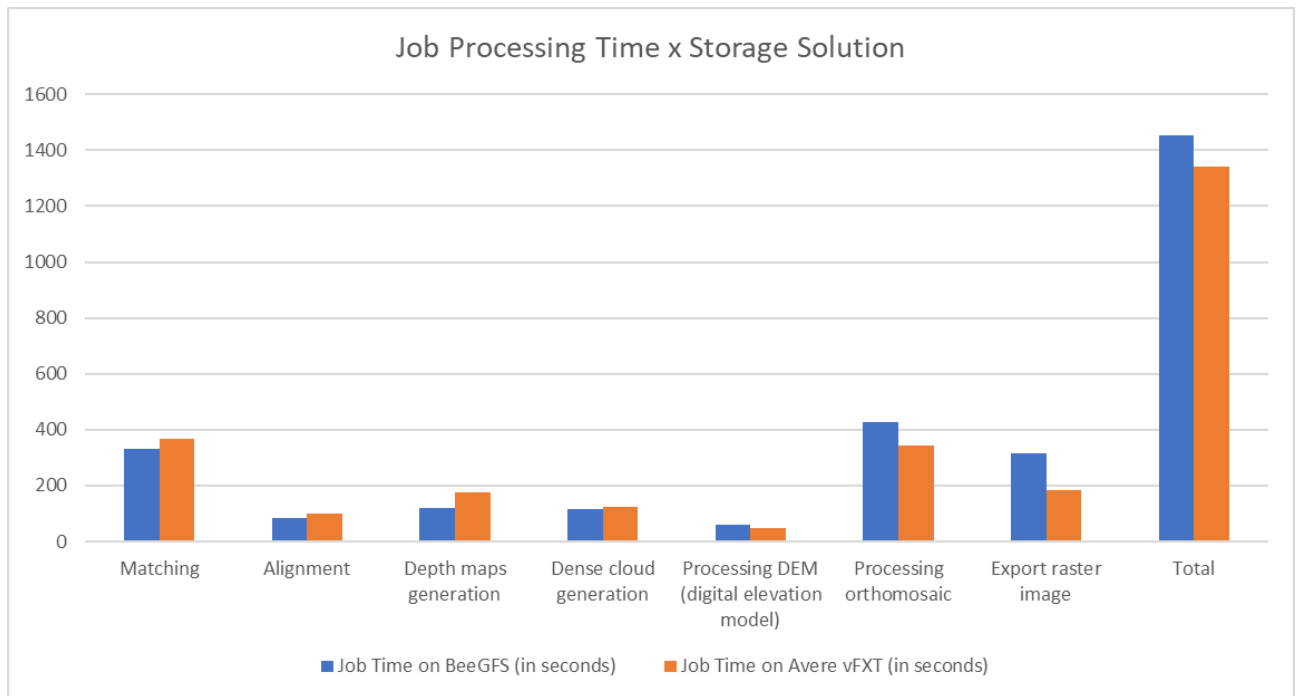


Figure 5. Benchmarking results comparing BeeGFS (in blue) to Avere vFXT (in orange).

# Download a sample dataset

The dataset used for our benchmark tests is Ortho\_and\_DEM\_without\_GCPs. To download this dataset for your own tests, go to <ftp://ftp.agisoft.com>

To get the username and password, please use the Agisoft support contact form at <http://www.agisoft.com/support/contact-form/> and request the FTP password.

Agisoft also provides smaller sample datasets on the [Agisoft Downloads](#) web page.

## Learn more

- [Avere website](#)
- [BeeGFS website](#)
- [Agisoft PhotoScan User Manual](#)
- [HPC, Batch, and Big Compute solutions using Azure VMs](#)

For more information about BeeGFS performance tuning on Azure, see the [Parallel Virtual File Systems on Microsoft Azure](#) ebook.

For alternative approaches to HPC workloads on Azure, see [HPC, Batch, and Big Compute solutions using Azure VMs](#).

## Appendix

The parameter files contain both general parameters in addition to those specific to the storage solution you select. Parameters related to Avere vFXT start with **nfs**; those related to BeeGFS starts with **beegfs**.

The azuredeploy.json template contains all parameters and their default values.

### BeeGFS parameter files

#### deploy-beegfs-masterparameters.json

Parameter	Description	Default value
_artifactsLocation	Auto-generated container in a staging storage account to receive post-build staging folder upload.	
_artifactsLocationSasToken	Auto-generated token to access _artifactsLocation.	
Location	Azure region where the resources used by this template are deployed.	eastus

dnsDomainName	DNS domain name used to build the host's FQDN. If using this parameter, make sure that there is a DNS server serving the virtual network before the BeeGFS servers are deployed.	
beeGfsMasterVmName	Name of the VM used for BeeGFS management (master).	beegfsmaster
VMSize	Size of VM to use for the storage nodes. Only premium disks are allowed.	Standard_DS4_v2
VMImage	The operating system used for all Linux VMs. (This template has been tested with CentOS 7.5 only.)	CentOS 7.5
vnetRG	Name of a resource group containing a virtual network. If none is specified, the deployment's resource group is used.	none
vnetName	Name of a virtual network to use for this deployment.	beegfs-vnet
subnetName	Name of the subnet where the BeeGFS components are deployed.	beegfs-subnet
addressPrefix	IP address for the beegfs-vnet virtual network.	192.168.0.0/16
subnetPrefix	IP address range for beegfs-subnet.	192.168.0.0/24
beeGfsMasterIpAddress	Static IP address used for the BeeGFS Management (master) node.	192.168.0.4
adminUsername	The username for the administrator user on all VMs in this deployment.	
sshKeyData	String containing the SSH RSA public key file.  <b>NOTE:</b> You must change this value.	
beegfsShareName	The BeeGFS mount point on the master and metadata/storage nodes.	/beegfs
beegfsHpcUserHomeFolder	The beegfs mount point on master and metadata/storage nodes for the hpcUser home folder, mounted on all nodes.	/mnt/beegfshome
hpcUser	The user that will be owner of all files in the HPC folder structure.	hpcuser

hpcUid	HPC user ID.	7007
hpcGroup	HPC group.	hpcgroup
hpcGid	HPC group ID.	7007

#### deploy-beegfs-nodes-parameters.json

Parameter	Description	Default Value
_artifactsLocation	Auto-generated container in a staging storage account to receive post-build staging folder upload.	
_artifactsLocationSasToken	Auto-generated token to access _artifactsLocation.	
Location	Azure region where the resources used by this template are deployed.	eastus
nodeNamePrefix	Name of the virtual machine scale set used for OSS and MDS (storage and metadata).	beegfsserver
dnsDomainName	DNS domain name used to build the host's FQDN.	
nodeType	Type of BeeGFS node to deploy (MDT and OSS in this template).	all
nodeCount	Number of BeeGFS nodes (100 or less).	4
VMSize	Size of the VMs used for the storage nodes. Only Premium Disk VMs are allowed.	Standard_D16s_v3
VMImage	Operating system for the VMs running the BeeGFS nodes.	CentOS_7.5
vnetName	Name of a virtual network to use for this deployment.	beegfs-vnet
subnetName	Name of the subnet where the BeeGFS components are deployed.	beegfs-subnet
adminUsername	Username for the administrator account used for all the VMs.  <b>NOTE:</b> You must change this value. Use security best practices and avoid common names.	
sshKeyData	String containing the SSH RSA	

public key file.

**NOTE:** You must change this value.

storageDiskSize	Premium storage disk size used for the storage services.	P30
StorageDisksCount	Number of storage disks.	5
metaDiskSize	Size of Premium storage disk used for the BeeGFS metadata services.	P20
MetaDisksCount	Number of metadata disks.	2
volumeType	Volume for data disks.	RAID0
vnetRg	Name of the resource group containing the virtual network used by the master server.	
masterName	Name of master VM.	beegfsmaster
beeGfsMountPoint	Shared BeeGFS data mount point. Smb Share (beeGfsSmbShareName) is a subfolder under this mount point.	/beegfs
beegfsHpcUserHomeFolder	The BeeGFS mount point on the master and nodes used for MDS/storage for the hpcUser home folder, mounted on all nodes.	/mnt/beegfshome
hpcUser	HPC user that owns all files in the hpc folder structure.	hpcuser
hpcUid	HPC user ID.	7007
hpcGroup	HPC group.	hpcgroup
hpcGid	HPC group ID.	7007
deployHaConfiguration	BeeGFS high availability configuration deployment.	yes

#### beegfs-clients-parameters.json

Parameter	Description	Default Value
_artifactsLocation	Auto-generated container in staging storage account to receive post-build staging folder upload.	
_artifactsLocationSasToken	Auto-generated token to access _artifactsLocation.	

Location	Azure region where the resources used by this template are deployed.	eastus
vnetRG	Name of the resource group containing the virtual network.	
vnetName	Name of the virtual network where the subnet will be added.	beegfs-vnet
subnetName	Existing subnet name.	beegfs-subnet
subnetIpAddressSuffix	Suffix for the static IP address used for the clients in the network part of a class C subnet.	192.168.0
startIpAddress	Start number of the host portion of the static IP addresses used for the clients.	50
nodeCount	Number of client nodes (100 or less).	2
vmNameSuffix	Suffix used in the client VM name.	beegfsclt
VMSize	Size of the VM used for the storage nodes. Only Premium disks VMs are allowed.	Standard_D4s_v3
VMImage	Operating system image used for the client VMs.	CentOS_7.5
dnsDomainName	DNS domain name used to build the host's FQDN.	
adminUsername	Username for the administrator account used for all the VMs. <b>NOTE:</b> You must change this value. Use security best practices and avoid common names.	
sshKeyData	String containing the SSH RSA public key file. <b>NOTE:</b> You must change this value.	
nodeType	Type of BeeGFS node to deploy.	client
masterName	Name of master VM.	beegfsmaster
sambaWorkgroupName	Name of Samba workgroup.	WORKGROUP
beeGfsMountPoint	Shared BeeGFS data mount point. Smb Share (beeGfsSmbShareName) will be a	/beegfs



	subfolder under this mount point.	
beeGfsSmbShareName	Samba share name that appears as a subfolder under beeGfsMountPoint.	beegfsshare
beegfsHpcUserHomeFolder	The BeeGFS mount point on master and MDS/storage nodes for the hpcUser home folder, mounted on all nodes.	/mnt/beegfshome
hpcUser	HPC user that owns all files in the hpc folder structure.	hpcuser
hpcUid	HPC user ID.	7007
hpcGroup	HPC group.	hpcgroup
hpcGid	HPC group ID.	7007
smbVip	Virtual IP address of the SMB clients.	192.168.0.55

## PhotoScan parameter files

The following table contains all the parameters for:

- azuredeploy.parameters.avery.json
- azuredeploy.parameters.beegfs.json

In practice, the files for Avere vFXT or BeeGFS specifically contain only the parameters for that solution. However, you can create a single file that contains all the parameters.

**NOTE:** Parameters that take effect only when deploying PhotoScan with Avere vFXT are prefixed with *nfs*. Parameters related to BeeGFS are prefixed with *beegfs*.

Parameter	Description	Default Value
_artifactsLocation	Auto-generated container in staging storage account to receive post-build staging folder upload.	
_artifactsLocationSasToken	Auto-generated token to access _artifactsLocation.	
location	Azure region where the resources of this template will be deployed.	eastus
adminPassword	Administrator password.	
useSingleResourceGroup	Whether to use multiple resource	yes

groups. If using multiple, change the resource groups manually in the variables section.

activationCode	PhotoScan activation code.	
adminUsername	Name of administrator account of the VMs.  <b>NOTE:</b> You must change this value. Use security best practices and avoid common names.	
sshKeyData	SSH rsa public key file as a string.  <b>NOTE:</b> You must change this value.	
vnetName	Virtual network name.	Photoscan-vnet
vnetAddressSpace	Virtual network address space.	10.0.0.0/16
jumpboxSubnetName	Jumpbox subnet name.	Jumpbox-SN
jumpboxSubnetAddressPrefix	Jumpbox subnet address prefix.	10.0.0.0/24
photoscanSubnetName	Name of the subnet containing the PhotoScan servers (scheduler and nodes).	Photoscan-SN
photoscanSubnetAddressPrefix	PhotoScan subnet address prefix.	10.0.1.0/24
adSubnetName	Subnet where the Active Directory domain controllers will be deployed.	AD-SN
adSubnetAddressPrefix	Active Directory subnet address prefix.	10.0.2.0/24
dc1Name	Name of domain controller 1.	DC-01
dc2Name	Name of domain controller 2.	DC-02
dc1IpAddress	IP address of domain controller 1.	10.0.2.4
dc2IpAddress	IP address of domain controller 2.	10.0.2.5
dcVmSize	Size of the VMs used for the domain controller.	Standard_DS2_v2
dnsDomainName	Active Directory FQDN.	testdomain.local
adDomainNetBIOSName	Active Directory NetBIOS domain name.	TESTDOMAIN
deployLinuxJumpbox	Indicates whether the template deploys a Linux jumpbox.	yes
windowsJumpboxVmName	Name of the VM used for the	wjb-01

Windows jumpbox.		
linuxJumpboxVmName	Name of the VM used for the Linux jumpbox.	ljb-01
windowsJumpboxVmSize	Size of the VM used for the Windows jumpbox.	Standard_DS2_v2
linuxJumpboxVmSize	Size of the VM used for the Linux jumpbox.	Standard_DS2_v2
windowsJumpboxIpAddress	IP address of the VM used for the Windows jumpbox.	10.0.0.4
linuxJumpboxIpAddress	IP address of the VM used for the Linux jumpbox.	10.0.0.5
headServerName	PhotoScan Server (head) name.	headnode
headVmSize	Size of the VM used for the PhotoScan head node.	Standard_D8S_v3
workerNodesType	Operating system type used for the worker nodes: Linux or Windows.	linux
nodeNamePrefix	Suffix used in the name of the GPU nodes.	workernode
nodeSubnetIpAddressSuffix	Network part of a class C subnet; nodes will have static IP addresses.	10.0.1
nodeStartIpAddress	Start number of the host part of the class C IP address; nodes will have static IP addresses.	20
nodeCount	Number of GPU VM nodes.	5
nodeVmSize	Size of the GPU VMs.	Standard_NC24s_v2
headRoot	Root path where the projects are located for the server.	\\beegfs\beegfsshare\Projects
nodeRoot	Root path where the projects are located for the nodes.	/beegfs/beegfsshare/Projects
dispatch	IP address of the PhotoScan server (head).	10.0.1.250
gpuMask	Decimal representation of how many GPUs will be enabled for processing. For example, 15 means 1111, that is equal to 4 GPUs.	15
windowsPhotoscanDownloadUrl	Windows binary PhotoScan download URL.	<a href="http://download.agisoft.com/photoscan-pro_1_4_4_x64.msi">http://download.agisoft.com/photoscan-pro_1_4_4_x64.msi</a>

linuxPhotoscanDownloadUrl	PhotoScan Linux binaries download URL.	http://download.agisoft.com/photoscan-pro_1_4_4_amd64.tar.gz
photoscanInstallPath	PhotoScan installation path.	/
photoscanAbsolutePaths	Use PhotoScan absolute paths. 0 = no, 1= yes.	0
useNfsStorage	Indicates whether the template uses NFS storage. If <i>yes</i> , make sure <i>useBeeGfsStorage</i> is set to <i>no</i> .	no
useBeeGfsStorage	Indicates whether the template uses BeeGfs storage. If <i>yes</i> , make sure <i>useNfsStorage</i> is set to <i>no</i> .	yes
storageVnetRG	Name of the resource group containing the storage virtual network.	beegfs-rg-eus
storageVnetName	Name of the virtual network containing the storage solution.	beegfs-vnet
sharedNfsStorageHpcUserHomeFolder	Indicates shared storage mount point on Linux VM nodes for the <i>hpcUser</i> home folder. It will be mounted on all Linux nodes.	/mnt/beegfshome
homeNfsExportPath	The export path configured in your NFS server for the home folder of the HPC User.  <b>NOTE:</b> If <i>useNfsStorage</i> is <i>yes</i> , this parameter with correct values is mandatory.	/home
sharedScratchMountPoint	Folder path where Shared Storage volume will be mounted on Linux VMs.	/beegfs
nfsScratchExportPath	The export path configured in your NFS server to be used by PhotoScan for project processing.  <b>NOTE:</b> If <i>useNfsStorage</i> is <i>yes</i> , this parameter with correct values is mandatory.	/data
nfsDnsEntry	A DNS record in this format:  <DNS A Record for NFS servers>,<IP1>,<IP2>,<IP3>,<IPx>	

For example:

vfx,10.0.0.11,10.0.0.12,10.0.0.13

**NOTE:** If useNfsStorage is *yes*, this parameter with correct values is mandatory.

nfsScratchFolderNfsVersion	NFS version used to mount scratch (data) folder. Used only when NFS storage is in use.	nfs
nfsHomeFolderNfsVersion	NFS version used to mount home folder for HPC User. When using BeeGFS, this value must be <i>nfs4</i> ; if using Avere vFXT, it must be <i>nfs</i> .	nfs
nfsScratchMountOptions	NFS scratch volume mount options, comma separated, no spaces. For example:  noatime,rsize=524288,wsize=524288	defaults
nfsExportPathUNC	The UNC NFS export path. Its value is used by the scheduler (head) running on Windows to map the NFS path to a drive letter when using NFS storage. For example, \\vfx!\msazure.	\\vfx!\msazure
nfsMountType	Type of NFS mounts, hard or soft.	hard
nfsMapDriveLetter	Drive letter used to map the path to nfsExportPathUNC.	z
nfsCaseSensitiveLookup	Indicates for NFS to use case-sensitive lookups.	False
nfsTimeout	NFS timeout.	60
nfsMountRetry	NFS mount retries.	3
nfsDefaultAccessMode	NFS default access mode.	777
nfsWindowsRsizeKb	Windows NFS Client read size in KB.	64
nfsWindowsWsizeKb	Windows NFS Client write size in KB.	64
beeGfsMasterName	Name of the VM used for the BeeGFS master node (single label).	beegfsmaster
beeGfsMasterIpAddress	IP address of the BeeGFS master. This address is added as an A record on DNS.	192.168.0.4

beeGfsSmbServersVip	IP address of the BeeGFS SMB client's load balancer. This address is added as an A record on DNS.	192.168.0.55
beeGfsSmbServerARecordName	BeeGFS A record to be used by Photoscan Server (head).	beegfs
hpcUser	HPC user that will be owner of all files in the HPC folder structure.	hpcuser
hpcUid	HPC user ID.	7007
hpcGroup	HPC group.	hpcgroup
hpcGid	HPC group ID.	7007