PHOTOMESH V.7.6.2 – GETTING STARTED ON AMAZON WEB SERVICES (AWS)

Skyline PhotoMesh is designed and built to fully exploit computer clusters and cloud computing. Cloud computing provides the flexibility to quickly scale up and down based on resource needs - even within a single project. Projects (or even steps within a single project) with demanding processing requirements can be run simultaneously on hundreds of virtual fuser machines, vastly accelerating mesh model creation, while for less demanding projects (or project steps), users can quickly scale down. Users only pay-per-use, thus avoiding any significant upfront investment or wasted resources.

Amazon Web Services (AWS) is a secure cloud services platform that provides a perfect fit for PhotoMesh users interested in leveraging cloud products for scalable processing power with only minimal investment in infrastructure.

This Quick Guide outlines the basic workflow for setting up PhotoMesh on your AWS account. The general workflow involves the following steps:

- Step 1: Setting up an Amazon Virtual Private Cloud (VPC).
- Step 2: Creating the Master PhotoMesh machine, by launching and customizing a Windows instance from an existing Amazon Machine Image (AMI) and setting up storage on Amazon EBS Volumes.
- Step 3 / Step 4: Creating a Linux or Windows AMI for a fuser computer by launching and customizing an instance from an existing AMI, and then creating a new AMI from the instance.
- Step 5: Building a PhotoMesh project by using the configured environment to create and then build a project.

The workflow outlined in this document is intended only as a recommendation for a typical workflow, and it can be modified based on other AWS and IT knowledge and preferences.

Note: This document is based on Amazon's AWS documentation, adapted for PhotoMesh's specific requirements.

Step 1 Create a Virtual Private Cloud (VPC)

To create a Virtual Private Cloud for PhotoMesh production, configure a VPC and then create a security group.

See: Create the VPC in Amazon's AWS documentation for more information.

1.1. Configuring a VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. On the dashboard choose **Launch VPC Wizard**.
3. Choose the first option, **VPC with a Single Public Subnet**, and then choose **Select**.
4. On the configuration page, enter the following information:

    a. **VPC name**: *PM VPC*.

b. **Public subnet's IPv4 CIDR**: Change X.X.X.X/24 to X.X.X.X/22 (E.g. The default 10.0.0.0/22).
   This allows up to 1019 instances on your network.

c. **Subnet name**: *PM Subnet*.

5. Click **Create VPC**.
6. In the navigation pane, choose **Subnets**.
7. Select **PM subnet**, choose **Actions**, and then **Modify Auto-Assign IP Settings**.
8. Select the **Enable Auto-assign Public IPv4 address** check box.
9. Click **Save**.

1.2. Creating a Security Group

See: Create a Security Group in Amazon's AWS documentation for more information.

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Security Groups**.
3. Choose **Create Security Group**.
4. In the **Security group name** field, enter *PM_SG* as the name of the security group, and type a description.
5. Select the ID of your PM VPC from the **VPC** menu, and then choose **Create**.
6. Select the **PM_SG** security group that you just created (you can view its name in the Group Name column).
7. On the **Inbound Rules** tab, choose **Edit** and add rules for inbound traffic as follows, and then choose **Save Rules** when you're done:

   a. Select **RDP** (Remote Desktop Protocol) from the **Type** list, and enter your network's public IP address range in the **Source** field. If you don't know this address range, you can use 0.0.0.0/0.

   b. Choose **Add Rule**, and select **SSH** from the **Type list.** Then enter your network's public IP address range in the **Source** field. If you don't know this address range, you can use 0.0.0.0/0.

   c. Choose **Add Rule**, select **Custom TCP Rule** from the **Type** list, and enter *5900-5901* (default VNC ports) in the **Port Range** field. Then enter your network's public IP address range in the **Source** field. If you don't know this address range, you can use 0.0.0.0/0.

   Note: For the rules above**,** when you use 0.0.0.0/0 for the Source field, you enable all IP addresses to access your instance using RDP, SSH, or VNC. This is suitable for a short exercise, but it is unsafe for production environments. In production, you will want to authorize only a specific IP address or range of addresses to access your instance.

   d. Choose **Add Rule**, select **Custom TCP Rule**, from the **Type** list, and enter *445* in the **Port Range** field. In the **Source** field, start typing *sg*, and select the group ID of your security group.

   Note: You can also add another identical rule for your network's IP address range, to allow file sharing directly from your computer. If you don't know this address range, you can use 0.0.0.0/0, but keep in mind that this will enable all IP addresses to access your instance using file sharing. This is suitable for a short exercise, but it is unsafe for production environments. In production, you will want to authorize only a specific IP address or range of addresses to access your instance.

Step 2 Create a Master Instance

Creating an Amazon Machine Image (AMI) for the master computer is a multi-part step that entails: launching an instance from an existing AMI and customization of the instance.

To create a master Instance, follow the steps below. For more information, **see**: Getting Started with Amazon EC2 Windows Instances in Amazon's AWS documentation.

2.1. Creating an IAM Policy and Role

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane on the left, choose **Policies**.
   If this is your first time choosing **Policies**, the **Welcome to Managed Policies** page appears. Choose **Get Started**.
3. Choose **Create policy**.
4. Choose the **JSON** tab.

   Note: You can switch between the **JSON** and **Visual editor** tabs any time, and add each of the permissions below individually.
5. Paste the following JSON policy document:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSpotPriceHistory",
        "ec2:DescribeImages",
        "ec2:CancelSpotInstanceRequests",
        "ec2:DescribeInstances",
        "ec2:TerminateInstances",
        "ec2:RequestSpotInstances",
        "ec2:CreateTags",
        "ec2:DescribeSpotInstanceRequests",
        "ec2:DescribeSubnets",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeInstanceStatus"
      ],
      "Resource": "*"
    }
  ]
}
```

6. When you are finished, choose **Review policy**.

7. On the **Review policy** page, enter *PM_RUNFUSER_POLICY*. Review the policy **Summary** to see the permissions that are granted by your policy. Then choose **Create policy** to save your work.
8. In the navigation pane, choose **Roles**, and then choose **Create role**.
9. For **Select type of trusted entity**, choose **AWS service**.
10. For **Choose the service that will use this role**, choose **EC2**. Then choose **Next: Permissions**.
11. In the list of policies, select the **PM_RUNFUSER_POLICY** policy. You can use the Filter menu to filter the list of policies.
12. Choose **Next: Tags**.
13. Choose **Next: Review**.
14. In the **Role name**, enter *PM_RUNFUSER_ROLE*.
15. Review the role and then choose **Create role**.

2.2. Launching an Initial Instance and Creating Storage

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Amazon EC2 console dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the following AMI from the **AWS Marketplace** list: **NVIDIA Gaming PC - Windows Server 2019**, and then choose **Select**.
4. On the **Choose an Instance Type** page, select the **g4dn.xlarge** type, and choose **Next: Configure Instance Details**.

   Note: Only instance types that match the NVIDIA AMI are available. If you want to use a different instance type, select a different AMI in step 2.2.3, e.g., Windows Server 2019. Make sure to install any graphics drivers that are required for your instance.

   Note: Using a GPU instance is recommended for normal operation of PhotoMesh Master over RDP connection. For productions with hundreds of fusers, other types can be used such as g4dn.8xlarge, which has faster network performance. For more information, see Instance Types in Amazon's AWS documentation.
5. On the **Configure Instance Details** page, enter the following information:

   a. **Network**: Choose the **PM VPC** VPC created in step 1.1.4.

   b. **IAM role**: Choose the **PM_RUNFUSER_ROLE** created in step 2.1.14.

   c. **Enable termination protection**: Select the check box.

   d. **Network Interfaces**: Set the **Primary IP** of **eth0** to *10.0.0.10*.

   e. Choose **Next: Add Storage**.
6. On the **Add Storage** page, enter the following information:

   a. Keep the default volume types.

   b. Click **Add New Volume**. Then set **Size** based on your expected need (E.g. 100 GiB). This is the EBS storage that will hold your data, projects, PM files, etc.

   Note: The EBS volumes will incur additional charges. For more information see Amazon EBS Volumes in Amazon's AWS documentation.

c. **Volume Type:** Choose a volume type based on your expected need (e.g., General Purpose SSD (gp2)). For more information, see Amazon EBS Volume Types.

d. Choose **Next: Add Tags**.

7. On the **Add Tags** page, tag your instance to help you identify it in the Amazon EC2 console after you launch it. Enter the following information:

   a. Select **Add Tag.**

   b. **Key**: *Name.*

   c. **Value**: *PM Master.*

   d. Click **Next: Configure Security Group.**

8. On the **Configure Security Group** page, do the following:

   a. Choose **Select an existing security group**.

   b. Select the **PM_SG** group.

   c. Choose **Review and Launch**.

9. On the **Review Instance Launch** page, check the details of your instance, and make any necessary changes by clicking the appropriate Edit link. When you are ready, choose **Launch**.

10. In the **Select an existing key pair or create a new key pair** dialog box, create a new key pair.

    a. Choose **Create a new key pair**.

    b. **Key pair name**: *PM_KeyPair*.

    c. Choose **Download Key Pair**.

    Note: Store the file in a secure and accessible location since you will need the contents of the private key to connect to your instance after it is launched.

    d. Choose **Launch Instances**.

11. To launch your instance, select the acknowledgment check box, and then choose **Launch Instances**.

2.3. Connecting to the Instance

See: Connecting to Your Windows Instance Using RDP in Amazon's AWS documentation for more information.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/
2. In the navigation pane, choose **Instances.**
3. Select the instance, and then choose **Connect**.
4. In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).
5. Choose **Choose File** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into the contents box.
6. Choose **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect To Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.

7. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.
8. Choose **Download Remote Desktop File**. Your browser prompts you to either open or save the .rdp file. Either option is fine. When you have finished, you can choose **Close** to dismiss the **Connect To Your Instance** dialog box.
9. You may get a warning that the publisher of the remote connection is unknown. Choose **Connect** to connect to your instance.
10. When prompted, log in to the instance, using the administrator account for the operating system and the password that you recorded or copied previously. If your **Remote Desktop Connection** already has an administrator account set up, you might have to choose the **Use another account** option and enter the user name and password manually.
11. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Choose **Yes** or **Continue** to continue.
12. A Windows **Networks** message is displayed asking if "you want to allow your PC to be discoverable by other PCs and devices on this network?" Click **Yes**.
13. After you connect, we recommend that you change the administrator password from the default value. You change the password while logged on to the instance itself, just as you would on any other Windows Server.

    Note: Due to the Remote Desktop Protocol (RDP), the recommended method for changing the password is to press CTRL+ALT+END, and then select **Change a password**.

2.4. Customizing the Instance

See: Making an Amazon EBS Volume Available for Use on Windows and Windows GPU Instances in Amazon's AWS documentation for more information.

1. While connected to the Master instance using RDP, start **Windows File Explorer**.
2. Start the Disk Management utility. On the taskbar, open the context (right-click) menu for the Windows logo and choose **Disk Management**. The Initialize Disk dialog box is displayed.
3. Select a partition style, and choose **OK**.
4. Open the context (right-click) menu for the right panel for Disk 1 and choose **New Simple Volume**. Complete the wizard with the default settings.
5. Give share permissions with **Full Control** to **Everyone** on the *D* drive.
6. Install PhotoMesh using the standard installation to *D:\PhotoMesh*.
7. Create the folder *D:\PMWorkingFolder,* and then then the subfolder *A* directly below it, so that you have the directory structure: *D:\PMWorkingFolder\A*.
8. Create the folder *D:\PMProjects*.
9. Start PhotoMesh from "\\10.0.0.10\D\PhotoMesh\PhotoMesh.exe".
10. Click the **PhotoMesh** button, and then click **Options (F9)**.
11. Change the **Working Folder** to *\\10.0.0.10\D\PMWorkingFolder\A*.
12. Click **OK** and then **close** PhotoMesh.

Step 3 Create a Fuser AMI (Linux)

Creating a Linux AMI for a fuser computer is a multi-part step that entails: launching an instance from an existing AMI, customizing the instance, and finally creating a new AMI from the instance.

To create a fuser AMI, follow the steps below. For more information, **see**: Getting Started with Amazon EC2 Linux Instances in Amazon's AWS documentation.

Linux fuser AMI's are generally recommended over Windows AMI's since Linux instances are more cost effective to run. If for various other considerations, however, you want to create an AMI for a Windows fuser instance, see: "Create a Fuser AMI (Windows) for information."

3.1. Launching an Initial Instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Amazon EC2 console dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the following AMI from the **Quick Start** list: **Ubuntu Server 18.04 LTS 64-bit (x86)**, and then choose **Select**.
4. On the **Choose an Instance Type** page, select the **g4dn.xlarge** type, and choose **Next: Configure Instance Details**.

   Note: The instance type selected here depends on the fusers you intend to use to build the project. The **g4dn.xlarge** instance type should be used to create an AMI for G4 and G3 instances. To create an AMI for G2 instances, select the **g2** type.
5. On the **Configure Instance Details** page, enter the following information:

   a. **Network**: Choose the **PM VPC** VPC created in step 1.1.4.

   b. Choose **Next: Add Storage**.
6. On the **Add Storage** page, enter the following:

   a. Change the default **Root** volume **Size** to at least *16* GiB.

   Note: The EBS volumes will incur additional charges. For more information see Amazon EBS Volumes in Amazon's AWS documentation.

   b. Choose **Next: Add Tags**.
7. On the **Add Tags** page, enter the following information:

   a. Select **Add Tag.**

   b. **Key**: *Name.*

   c. **Value**: *PM Linux Fuser Initial.*

   d. Choose **Next: Configure Security Group** when you are done.
8. On the **Configure Security Group** page:

   a. Choose **Select an existing security group**.

   b. Select the **PM_SG** group.

   c. Choose **Review and Launch**.
9. On the **Review Instance Launch** page, review the details of your instance, and make any necessary changes by clicking the appropriate Edit link. When you are finished, choose **Launch**.
10. In the **Select an existing key pair or create a new key pair** dialog box:

    a. Choose **Choose an existing pair**.

b. **Key pair name**: *PM_KeyPair*.

c. Choose **Launch Instances**.

11. To launch your instance, select the acknowledgment check box, and then choose **Launch Instances**.

3.2. Convert Your Private Key Using PuTTYgen

See: Connecting to Your Linux Instance from Windows Using PuTTY in Amazon's AWS documentation for more information.

Note: For information on additional ways to connect to your Linux instance, see: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html

1. Download and install PuTTY from the PuTTY download page. If you already have an older version of PuTTY installed, we recommend that you download the latest version. Be sure to install the entire suite.
2. From the **Start** menu, choose **All Programs**, **PuTTY**, **PuTTYgen**.
3. Under **Type of key to generate**, choose **RSA**. If you're using an older version of PuTTYgen, choose **SSH-2 RSA**.
4. Choose **Load**. By default, PuTTYgen displays only files with the extension .ppk. To locate your .pem file, choose the option to display files of all types.
5. Select your .pem file for the key pair that you specified when you launched your instance and choose **Open**. PuTTYgen displays a notice that the .pem file was successfully imported. Choose **OK**.
6. To save the key in the format that PuTTY can use, choose **Save private key**. PuTTYgen displays a warning about saving the key without a passphrase. Choose **Yes**.

   Note: A passphrase on a private key is an extra layer of protection. Even if your private key is discovered, it can't be used without the passphrase. The downside to using a passphrase is that it makes automation harder because human intervention is needed to log on to an instance, or to copy files to an instance.

7. Specify the same name for the key that you used for the key pair (for example, *PM_KeyPair*) and choose **Save**. PuTTY automatically adds the .ppk file extension.
   Your private key is now in the correct format for use with PuTTY. You can now connect to your instance using PuTTY's SSH client.

3.3. Connecting to the Instance

1. Start PuTTY (from the **Start** menu, choose **All Programs**, **PuTTY**, **PuTTY**).
2. In the **Category** pane, choose **Session** and complete the following fields:

   a. In the Host Name box, to connect using your instance's public DNS, enter *ubuntu@public_dns_name*.
   You can get the public DNS for your instance using the Amazon EC2 console. Check the **Public DNS (IPv4)** column. If this column is hidden, choose the **Show/Hide** icon and select **Public DNS (IPv4)**.

   b. Ensure that the **Port** value is *22*.

   c. Under **Connection type**, select **SSH**.

3. (Optional) You can configure PuTTY to automatically send 'keepalive' data at regular intervals to keep the session active. This is useful to avoid disconnecting from your instance due to session inactivity. In the **Category** pane, choose **Connection**, and then enter the

required interval in the **Seconds between keepalives** field. For example, if your session disconnects after 10 minutes of inactivity, enter *180* to configure PuTTY to send keepalive data every 3 minutes.

4. In the **Category** pane, expand **Connection**, expand **SSH**, and then choose **Auth**. Complete the following:
   a. Choose **Browse**.
   b. Select the .ppk file that you generated for your key pair and choose **Open**.
   c. (Optional) If you plan to start this session again later, you can save the session information for future use. Under **Category**, choose **Session**, enter a name for the session in **Saved Sessions**, and then choose **Save**.
   d. Choose **Open**.
5. If this is the first time you have connected to this instance, PuTTY displays a security alert dialog box that asks whether you trust the host to which you are connecting. Choose **Yes**. A window opens and you are connected to your instance.

3.4. Customizing the Instance

See: Linux Accelerated Computing Instances in Amazon's AWS documentation for more information.

1. To download and install the Nvidia drivers and update the operating system, in the PuTTY terminal window, type:
   a. *wget http://us.download.nvidia.com/tesla/440.33.01/NVIDIA-Linux-x86_64-440.33.01.run*
   b. *sudo apt-get update*
   c. *sudo apt-get -y install gcc make xserver-xorg libglu1-mesa-dev freeglut3-dev mesa-common-dev libxmu-dev libxi-dev mesa-utils x11vnc xfce4*
   d. *chmod +x NVIDIA-Linux-x86_64-440.33.01.run*
   e. *sudo ./NVIDIA-Linux-x86_64-440.33.01.run*
      Click **OK** when prompted to do so during the installation.
   f. *sudo reboot*
      The following message is displayed: "Remote side unexpectedly closed network connection". Click **OK**.
2. Right-click the title bar of the PuTTY window, and select **Restart Session**.
3. To configure the GPU and start the X window system, in the PuTTY terminal window, type:
   a. *nvidia-xconfig --query-gpu-info | grep BusID*
      Copy the value that is outputted. E.g., if the output is "PCI BusID : PCI:0:30:0", select and copy "*PCI:0:30:0*".
   b. *sudo nvidia-xconfig -a --virtual=1280x1024 --busid=<BUSID>*
      Where <BUSID> is replaced with the value from 2c above, e.g., sudo nvidia-xconfig -a --virtual=1280x1024 --busid=**PCI:0:30:0**.

      Note: If a warning is displayed that the xconfig file was not located/opened, click **OK**.
   c. *sudo /usr/bin/X :0 &*
   d. *export DISPLAY=:0*
4. (Optional) To configure VNC in order to verify X runs properly, in the PuTTY terminal window type:

a. *x11vnc –storepasswd*
   Then type and verify the password

b. *y*
   To confirm writing the password to /home/Ubuntu/.vnc/passwd.

c. *x11vnc -rfbauth ~/.vnc/passwd &*

d. *xfce4-session &*

5. (Optional) Connect with VNC client using your instance's public DNS and the password set in step 3.4.4.a above.
   You can get the public DNS for your instance using the Amazon EC2 console. Check the **Public DNS (IPv4)** column. If this column is hidden, choose the **Show/Hide** icon and select **Public DNS (IPv4)**.
6. To install and configure Wine, do the following:
   a. Open the **Start** menu and search for **cmd** to start a command prompt.
   b. Type *pscp -i <.ppk file> <wineinst.tar full path> ubuntu@<fuser IP>:/home/ubuntu/wineinst.tar*
      Where <.ppk file> is replaced with a full path to the ppk file created in step 3.2; <wineinst.tar full path> is replaced with the full path to the wineinst.tar that can be downloaded from here; and <fuser IP> is replaced with the Public DNS (IPv4) of the instance.
   c. In the PuTTY terminal window, type:
      i. *tar -xvf wineinst.tar*
      ii. *sh fuser.sh*

      Note: This process can take a few minutes to complete.
7. Edit the file **rc.local-awsQG-Template.txt** that can be downloaded from here. It is recommended to edit the file in Notepad++ or another Unix-format text editor:
   - Replace <pass> with the password you set in step 3.2.13.

   Note: The master server IP 10.0.0.10 and the fuser path "/mnt/mount1/PhotoMesh/Fuser" are both written in the file. If you configured your environment differently in the previous steps, replace these values in the file with the values you set. You can also use rc.local-Template.txt as a general template and update the fields marked with angle brackets <>.
8. Rename the file "rc.local-awsQG-Template.txt" to "rc.local" (no txt extension).
9. Copy the "rc.local" file to the instance:
   a. Open the **Start** menu and search for **cmd** to start a command prompt.
   b. Type *pscp -i <.ppk file> <rc.local full path> ubuntu@<fuser IP>:/home/ubuntu/rc.local*
      Where <.ppk file> is replaced with a full path to the ppk file created in step 3.2; <rc.local full path> is replaced with the file renamed in step 3.4.8; and <fuser IP> is replaced with the Public DNS (IPv4) of the instance.
      E.g., *pscp -i c:\ PM-AWS-Files\PM_KeyPair.ppk C:\ PM-AWS-Files\rc.local ubuntu@123.123.123.123:/home/ubuntu/rc.local*
   c. In the PuTTY terminal window, type:
      i. *sudo mv rc.local /etc*
      ii. *sudo chmod +x /etc/rc.local*

iii. *sudo reboot*

10. Right-click the title bar of the PuTTY window, and select **Restart Session**.
11. In the PuTTY terminal window, type *ps aux* | *grep PhotoMeshFuser.exe* | *grep -v grep* | *grep -v wine*
    If the PhotoMesh fuser runs properly, one line will be returned, e.g. "ubuntu   1344 0.8 1.3 2445832 217412 ?   Sl   13:19   0:02 Z:\mnt\mount1\PhotoMesh\Fuser\PhotoMeshFuser.exe"

3.5. Saving a Fuser AMI

See: Creating an Amazon EBS-Backed Linux AMI in Amazon's AWS documentation for more information.

1. In the navigation pane, choose **Instances** and select the **PM Linux Fuser Initial** instance. Choose **Actions**, **Image**, and **Create Image**.
2. In the **Create Image** dialog box, specify values for the following fields, and then choose **Create Image**.
   - **Image name**: *PM Linux Fuser Image*.
3. While your AMI is being created, you can choose **AMIs** in the navigation pane to view its status. Initially, this is pending. After a few minutes, the status should change to available.
4. Record the AMI ID of the created AMI, or copy it to the clipboard. You need this AMI ID for PhotoMesh to launch fuser instances.
5. In the navigation pane, choose **Instances** and select the **PM Linux Fuser Initial** instance.
6. Choose **Actions**, select **Instance State**, and then choose **Terminate**.

Step 4 Create a Fuser AMI (Windows)

Creating a Windows AMI for a fuser computer is a multi-part step that entails: launching an instance from an existing AMI, customizing the instance, and finally creating a new AMI from the instance.

Note: It is generally recommended to create **Linux** fuser AMI's since Linux instances are more cost effective to run. See "Create a Fuser AMI (Linux)" for more information.

To create a fuser AMI, follow the steps below. For more information, **see**: Getting Started with Amazon EC2 Windows Instances in Amazon's AWS documentation.

4.1. Launching an Initial Instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the Amazon EC2 console dashboard, choose **Launch Instance**.
3. On the **Choose an Amazon Machine Image (AMI)** page, choose the following AMI from the **AWS Marketplace** list: **NVIDIA Gaming PC - Windows Server 2019**, and then choose **Select**.
4. In the **NVIDIA Gaming PC - Windows Server 2019** dialog box, choose **Continue**.
5. On the **Choose an Instance Type** page, select the **g4dn.2xlarge** type, and choose **Next: Configure Instance Details**.

   Note: Only instance types that match the NVIDIA AMI are available. If you want to use a different instance type, select a different AMI in step 4.1.3, e.g., Windows Server 2019. Make sure to install any graphics drivers that are required for your instance.

6. On the **Configure Instance Details** page, enter the following information:
    a. **Network**: Choose the **PM VPC** VPC created in step 1.1.4.
    b. Choose **Next: Add Storage**.
7. On the **Add Storage** page, enter the following:
    a. Keep the default volume types.
    b. Choose **Next: Add Tags**.
8. On the **Add Tags** page, enter the following information:
    a. Select **Add Tag.**
    b. **Key**: *Name.*
    c. **Value**: *PM Fuser Initial.*
    d. Choose **Next: Configure Security Group** when you are done.
9. On the **Configure Security Group** page:
    a. Choose **Select an existing security group**.
    b. Select the **PM_SG** group.
    c. Choose **Review and Launch**.
10. On the **Review Instance Launch** page, review the details of your instance, and make any necessary changes by clicking the appropriate Edit link. When you are finished, choose **Launch**.
11. In the **Select an existing key pair or create a new key pair** dialog box:
    a. Choose **Choose an existing pair**.
    b. **Key pair name**: *PM_KeyPair*.
    c. Choose **Launch Instances**.
12. To launch your instance, select the acknowledgment check box, and then choose **Launch Instances**.

4.2. Connecting to the Instance

See: Connecting to Your Windows Instance Using RDP in Amazon's AWS documentation for more information.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/
2. In the navigation pane, choose **Instances**.
3. Select the instance, and then choose **Connect**.
4. In the **Connect To Your Instance** dialog box, choose **Get Password** (it will take a few minutes after the instance is launched before the password is available).
5. Choose **Choose File** and navigate to the private key file you created when you launched the instance. Select the file and choose **Open** to copy the entire contents of the file into the contents box.
6. Choose **Decrypt Password**. The console displays the default administrator password for the instance in the **Connect To Your Instance** dialog box, replacing the link to **Get Password** shown previously with the actual password.

7. Record the default administrator password, or copy it to the clipboard. You need this password to connect to the instance.
8. Choose **Download Remote Desktop File**. Your browser prompts you to either open or save the .rdp file. Either option is fine. When you have finished, you can choose **Close** to dismiss the **Connect To Your Instance** dialog box.
9. You may get a warning that the publisher of the remote connection is unknown. Choose **Connect** to connect to your instance.
10. When prompted, log in to the instance, using the administrator account for the operating system and the password that you recorded or copied previously. If your **Remote Desktop Connection** already has an administrator account set up, you might have to choose the **Use another account** option and enter the user name and password manually.
11. Due to the nature of self-signed certificates, you may get a warning that the security certificate could not be authenticated. Choose **Yes** or **Continue** to continue.
12. A Windows **Networks** message is displayed asking if "you want to allow your PC to be discoverable by other PCs and devices on this network?" Click **Yes**.
13. After you connect, we recommend that you change the administrator password from the default value to the same password set for the master computer in step 2.3.13. You change the password while logged on to the instance itself, just as you would on any other Windows Server.

    Note: Due to the Remote Desktop Protocol (RDP), the recommended method for changing the password is to press CTRL+ALT+END, and then select **Change a password**.

4.3. Customizing the Instance

See: Windows Accelerated Computing Instances in Amazon's AWS documentation for more information.

1. While connected to the fuser instance using RDP, start **Windows File Explorer**.

    Note: Make sure the *PM Master* instance is started.
2. Map a network drive P to \\10.0.0.10\D.

    Note: You must use the Private IP address (and not Public IP or computer name) that was defined in step 2.2.5.
3. Start **PhotoMesh Fuser** from "\\10.0.0.10\D\PhotoMesh\fuser\PhotoMeshFuser.exe".
4. In the **Select Folder** dialog box, browse to the "\\10.0.0.10\D\PMWorkingFolder", select "A" folder, and click **Select Folder**.
5. Open the **Start** menu and search for **cmd** to start a command prompt.
6. Type *control userpasswords2*.
7. Select the **Administrator** user and clear the **Users must enter a user name and password to use this computer** checkbox, and click **OK**.
8. Enter and confirm the password you set in step 4.2.13.
9. Start the **Task Scheduler**
    a. Right-click **Task Scheduler Library** and select **Create Task**.
    b. **General** > **Name**: *Start PM Fuser*.
    c. Triggers:

    i. Click **New**.
    ii. **Begin the task**: Select **At log on**.
    iii. **Settings**: Specific user.
    iv. Click **OK**.

  d. Actions:
    i. Click **New**.
    ii. **Program/script**: \\10.0.0.10\d\PhotoMesh\fuser\PhotoMeshFuser.exe.
    iii. Click **OK**.

  e. Settings:
    i. Select **if the task fails, restart every**, and keep the default **1 minute** and **3 times**.

  f. Click **OK**.

  g. Right-click **Task Scheduler Library** and select **Create Task**.

  h. **General** > **Name**: *Start Cloud Fuser Watchdog*.

  i. Triggers:
    i. Click **New**.
    ii. **Begin the task**: Select **At log on**.
    iii. **Settings**: Specific user.
    iv. Click **OK**.

  j. Actions:
    i. Click **New**.
    ii. **Program/script**: \\10.0.0.10\d\PhotoMesh\fuser\PhotoMeshFuser.exe.
    iii. Click **OK**.

  k. Settings:
    i. Select **if the task fails, restart every**, and keep the default **1 minute** and **3 times**.

  l. Click **OK**.

4.4. Saving a Fuser AMI

See: Creating an Amazon EBS-Backed Windows AMI in Amazon's AWS documentation for more information.

1. In the navigation pane, choose **Instances** and select the **PM Fuser Initial** instance. Choose **Actions**, **Image**, and **Create Image**.
2. In the **Create Image** dialog box, specify values for the following fields, and then choose **Create Image**.
    - **Image name**: *PM Fuser Image*.
3. While your AMI is being created, you can choose **AMIs** in the navigation pane to view its status. Initially, this is pending. After a few minutes, the status should change to available.
4. Record the AMI ID of the created AMI, or copy it to the clipboard. You need this AMI ID for PhotoMesh to launch fuser instances.
5. In the navigation pane, choose **Instances** and select the **PM Fuser Initial** instance.
6. Choose **Actions**, select **Instance State**, and then choose **Terminate**.

Step 5 Build a PhotoMesh Project

5.1. Building a PhotoMesh Project Using the Master Instance

To use the configured environment to produce PhotoMesh projects, follow the steps below:

1. Connect to the **PM Master** instance.

   Note: If the PM Master instance is not already running, choose **Actions**, select **Instance State**, and then choose **Start**.

2. Start **PhotoMesh**, and **create** a project.

   Note: The project must be created in a location that is accessible to both master and fuser instances. It is recommended to organize all projects and sources in the \\10.0.0.10\D\PMProjects\ directory.

3. On the **Home** tab, in the **Build** group, click **Build**, then enter the Build Parameters and click **Build**.

4. In the **PhotoMesh Build Manger** dialog box, select **Automatically launch AWS fuser instances**.

5. In the **AWS Cloud Settings** dialog, enter the following information:

   a. **Fuser AMI ID**: Choose the AMI ID recorded in step 3.5.4 for Linux instances or 4.4.4 for Windows instances.

   b. **Maximum Instances**: Set the number based on the number of EC2 spot instances you want to run.

   c. **Instance Type (Linux/Windows)**: Choose an instance type, e.g., g4dn.2xlarge or g4dn.4xlarge based on your requirements.

      Note: The spot instances will incur additional charges. For more information see: Amazon EC2 Spot Instance Pricing in Amazon's AWS documentation.

   d. **Price Per Instance**: Set the price based on your requirements.

      Note: The spot instances will incur additional charges. For more information see Amazon EC2 Spot Instance Pricing in Amazon's AWS documentation.

   e. Review the other parameters in the **AWS Cloud Setting** dialog. For more information see the PhotoMesh User Guide.

   f. Click **OK**.

6. Click **Build**.

   Note: A message that no available fusers are currently running is displayed. If you want to start a fuser on the master machine to take advantage of its computing resources, click **Yes**. If you are running many spot fuser instances, however, it is recommended to click **No** to free up computing resources for management of the build and file server tasks.

   Note: If a message is displayed that "PhotoMesh's fuser request was rejected by AWS, since it exceeded the instance limit", you need to request a limit increase for "Spot instance requests". **See**: Amazon EC2 Service Limits in Amazon's AWS documentation for more information.

7. In the **PhotoMesh Build Manger** dialog box, you can monitor the creation and automatic use of the automatically launched fuser spot instances. You can also monitor the creation of your spot instances by choosing **Instances** in the navigation pane, and review the custom tags

added to PhotoMesh launched spot instances: Build start time, Owner, Project, Type, and Working folder.

8. When processing is complete, stop the PM Master instance.

   Note: Do not terminate the master instance; only stop and start as needed.

   Note: It is recommended to verify that the spot instances are terminated according to the settings defined in the **AWS Cloud Settings** dialog, particularly the first time they are terminated.

5.2. Creating an IAM User for Running PhotoMesh Project Using an On-Premises Master

To allow PhotoMesh Master running on On-Premises computer to launch EC2 instances, create an IAM user and assign permissions.

See: Creating IAM Users (Console) in Amazon's AWS documentation for more information.

1. Open the Amazon IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users** and then choose **Add user**.
3. Type the user name for the new user.
4. Select the **Programmatic access** check box. Then choose **Next: Permissions**.
5. On the **Set Permissions** page, choose **Attach existing policies to user directly**.
6. In the list of policies, select the **PM_RUNFUSER_POLICY** policy. You can use the Filter menu to filter the list of policies.
7. Choose **Next: Tags**.
8. Choose **Next: Review**.
9. Review the role and then choose **Create user**.
10. To view the users' access keys (access key IDs and secret access keys), choose **Show** next to each password and access key that you want to see. To save the access keys, choose **Download .csv** and then save the file to a safe location.

    Note: This is your only opportunity to view or download the secret access keys. Save the user's new access key ID and secret access key in a safe and secure place. You will not have access to the secret keys again after this step.

5.3. Building a PhotoMesh Project Using an On-Premises Master

To use the configured environment to produce PhotoMesh projects using a PhotoMesh that is installed on an On-Premises computer while utilizing AWS fuser instances, follow the steps below:

1. Make sure the **PM Master** instance is running.

   Note: If the PM Master instance is not already running, choose **Actions**, select **Instance State**, and then choose **Start**. The PM Master must be running to serve as a file server for the EBS storage holding the PM files, data, and projects.

2. Make sure your On-Premises computer has PhotoMesh installed, and has file access to the same IP address that was defined in step 2.2.5.

   Note: File access can be achieved in different ways, e.g. VPN connection. Consult AWS documentation, support, your IT person, or Skyline support for more information.

3. Start **PhotoMesh**, and **create** a project.

Note: The project must be created in a location that is accessible to both master and fuser instances. It is recommended to organize all projects and sources in the \\10.0.0.10\D\PMProjects\ directory.

4. On the **Home** tab, in the **Build** group, click **Build**, then enter the Build Parameters and click **Build**.
5. In the **PhotoMesh Build Manger** dialog box, select **Automatically launch AWS fuser instances**.
6. In the **AWS Security** dialog, enter your AWS credentials created in step 5.2.
7. In the **AWS Cloud Settings** dialog, enter the following information:
   a. **Fuser AMI ID**: Choose the AMI ID recorded in step 3.5.4 for Linux instances or 4.4.4 for Windows instances.

   b. **Maximum Instances**: Set the number based on the number of EC2 spot instances you want to run.

   c. **Price Per Instance**: Set the price based on your requirements.

   Note: The spot instances will incur additional charges. For more information see Amazon EC2 Spot Instance Pricing in Amazon's AWS documentation.

   d. Review the other parameters in the **AWS Cloud Setting** dialog. For more information see the PhotoMesh User Guide.

   e. Click **OK**.
8. Click **Build**.

   Note: A message is displayed informing you that no available fusers are currently running and asking if you want to start a local fuser. If you want to start a fuser on the master machine to take advantage of its computing resources, click **Yes**. If you are running many spot fuser instances, however, it is recommended to click **No** to free up computing resources for management of the build and file server tasks.

   Note: If a message is displayed that "PhotoMesh's fuser request was rejected by AWS, since it exceeded the instance limit", you need to request a limit increase for "Spot instance requests". **See**: Amazon EC2 Service Limits in Amazon's AWS documentation for more information.
9. In the **PhotoMesh Build Manger** dialog box, you can monitor the creation and automatic use of the automatically launched fuser spot instances. You can also monitor the creation of your spot instances by choosing **Instances** in the navigation pane, and review the custom Tags added to PhotoMesh launched spot instances: Build start time, Owner, Project, Type, and Working folder.
10. When processing is complete, stop the PM Master instance.

    Note: Do not terminate the master instance; only stop and start as needed.

    Note: It is recommended to verify that the spot instances are terminated according to the settings defined in the **AWS Cloud Settings** dialog, particularly the first time they are terminated.