

Setting up a remote AWS instance for U-Net Segmentation

Revision: 12/15/2022 by Madi McElliott and Jeff Beamish

This protocol describes how to set up a remote computer (called an “Instance”) that is equipped with a GPU and processing tools to efficiently perform U-Net training and segmentations used in other aspects of protocols “Analysis” and “Training.”

Note: this protocol utilizes several third-party and/or commercial applications including Amazon Web Services (AWS), PuTTY, and FileZilla. We do not endorse the specific applications and the tasks described below can be performed using a variety of alternatives. However, these applications were utilized for this manuscript and so detailed instructions are provided to allow for replication of our work. Please note that these services are continuously updating so details in the instructions below may need to be adapted to new versions of the software.

BEFORE STARTING:

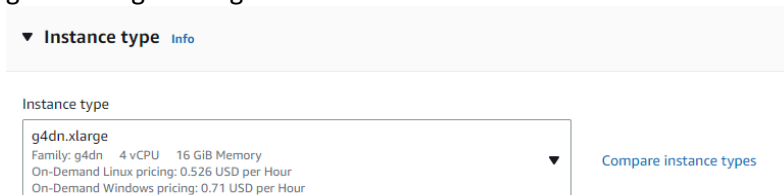
You must have the “U-Net Segmentation” plugin installed in FIJI. If it is not installed, go to Help→”Update...” then select “Manage update sites” and select “U-Net Segmentation” which will add this package to FIJI. You will have to close and restart FIJI after adding.

You will need to locate (or download) at least one base model (which includes TWO files, one ending in “...modeldef.h5” and one ending in “...caffemodel.h5 files”, for example for the DAPI model: “DAPI.modeldef.h5” and “DAPI.caffemodel.h5”). For training it is best to start with a model that most closely replicates the new target of interest. For analysis, select the model that was pre-trained and validated for your target of interest.

To follow these instructions, you will need to install PuTTY, PuTTYgen, and FileZilla. As above, there are many alternatives to these programs for knowledgeable users.

STEP ONE: Start an Amazon Web Services (AWS) EC2 instance:

- 1) Login to AWS
- 2) Select “EC2”
- 3) Select “Launch Instance”
- 4) Under “Application and OS Images (Amazon Machine Image)” search for “Deep Learning Base AMI (Ubuntu 18.04) Version 44.0”. Use the results that appears in “Community AMIs”. NOTE: newer Deep Learning Base AMIs in Ubuntu have newer versions of “cuda” and “cudnn” that are not compatible with the available U-Net code, so use Version 44 or older.
- 5) Under “Instance Type” chose “g4dn.xlarge”. You will have to click the “g4dn” instance type. This is the only type that can support “cuda” and other add ons as implemented by the U-Net software. Note this type is the most affordable and we have found it to be adequate for all our needs, but faster speeds can be obtained with g4dn.2xlarge or larger

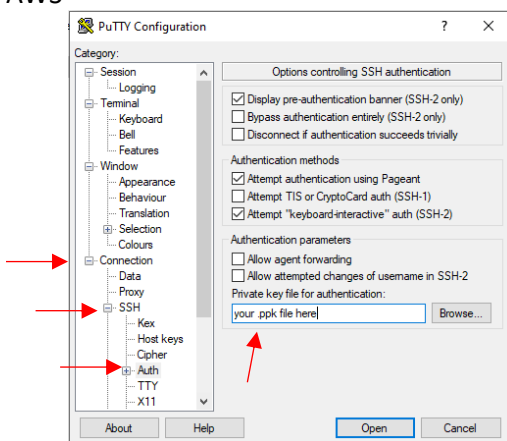


- 6) Under “Key pair (login),” select a previously saved “*.pem” key (you only need to make a new key once per region). If you do not have a Key pair already created, see the supplemental instructions below “CREATING A KEY PAIR”
- 7) Under “Network Settings” you can keep the defaults or add additional security if desired.
- 8) Under “Configure storage” you can use the defaults.

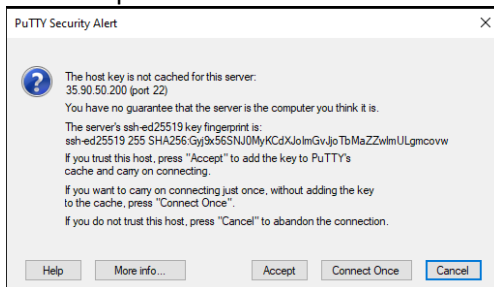
- 9) Under “Advanced Details” select “Request Spot Instances”. This is not required. The “spot” instance significantly reduces the cost. However, it is not guaranteed to be available, and it can be terminated if AWS needs the hardware for other users. Using an “On demand” instance will be available and guaranteed to continue running, but costs more.
- 10) Click the “Launch instance” button.

STEP TWO: Configure the Ubuntu EC2 Instance to run U-Net

- 1) Go to “EC2 Dashboard” → “Instances” → Instance ID (the one created above): Copy “Public IPv4 address”
- 2) After the instance is running (which will take about 2-5 min), log onto the instance using PuTTY (<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>)
- 3) Under “Session” within PuTTY, paste the Public IP into “Host Name (or IP address)”. Use port 22 (which is the default in **STEP ONE -- 7**).
- 4) Under “Connection” then under “SSH” (you will need to expand it), select “Auth”
- 5) Under “Authentication Parameters” click “Browse...” to in the field for “Private key file for authentication:”. Select the previously created “*.ppk” key file that corresponds to the “*.pem” file used to generate “Instance” within AWS



- 6) Select “Open”. After the instances starts to launch, click “Accept” the PuTTY Security Alert:



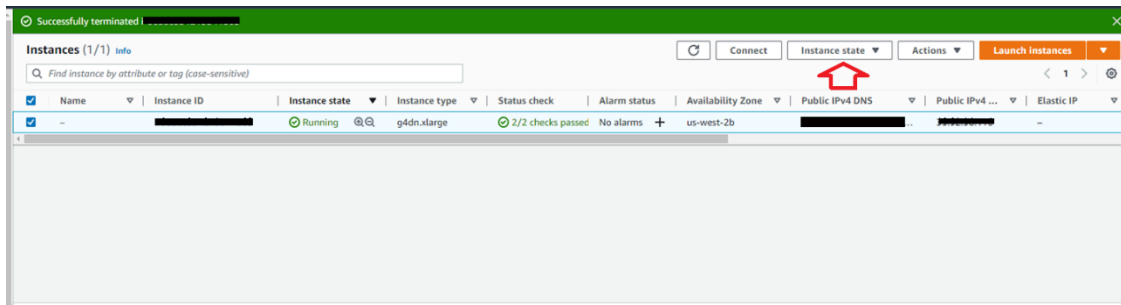
- 7) After the new terminal opens, enter “ubuntu” after “Login As: “
- 8) Paste the following code (copy then right click in the PuTTY terminal at the cursor to paste) into the terminal. This code loads and installs the U-Net software on the instance and configures the instance to direct the ImageJ plugin to the correct local library locations when needed. This code was taken from U-Net documentation.

```
wget https://lmb.informatik.uni-freiburg.de/resources/opensource/unet/caffe_unet_package_18.04_gpu_cuda10_cudnn7.tar.gz
tar -xvzf caffe_unet_package_18.04_gpu_cuda10_cudnn7.tar.gz
echo "export PATH=$PATH:/home/ubuntu/caffe_unet_package_18.04_gpu_cuda10_cudnn7/bin" | cat - ~/.bashrc > tmp
echo "export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/ubuntu/caffe_unet_package_18.04_gpu_cuda10_cudnn7/lib:/home/ubuntu/caffe_unet_package_18.04_gpu
_cuda10_cudnn7/extlib:/usr/local/cuda-10.0/lib64" | cat - tmp > ~/.bashrc
```

- 9) The instance is now configured. Close PuTTY.

STEP THREE: Upload model data

Before deploying the automated segmentation processing macros, all required model weight files (that is, all ...caffemodel.h5 files) must be on the EC2 instance. Upload the model data (...caffemodel.h5) to the AWS instance using

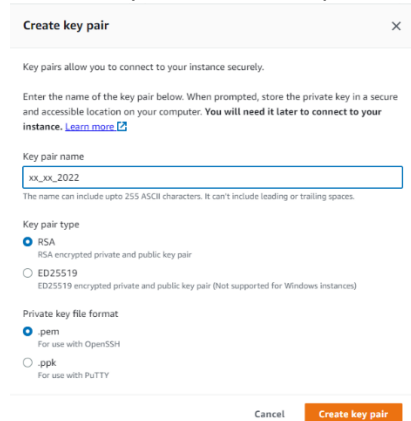


14) From here, you may proceed to “Training” protocol **STEP SIX**

SUPPLEMENTAL STEP: CREATING A KEY PAIR

This step describes how to create a key pair in **STEP ONE - 6** above and then used in several later steps. You only need to do this one time unless you change your AWS region.

1. In **STEP ONE - 6** above, rather than selecting a previous key, select “Create new key pair.”
 - a. Supply a “Key pair name”: anything that will be identifiable later. Note the location where you save it as you will use this key later.
 - b. Key pair type = “RSA”
 - c. Private key file format = “.pem”



The screenshot shows the 'Create key pair' dialog box in AWS. It has a title bar 'Create key pair' with a close button. Below the title bar, it says 'Key pairs allow you to connect to your instance securely.' followed by instructions: 'Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#).' There is a text input field for 'Key pair name' containing 'xx_xx_2022'. Below the field, it says 'The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.' There are two sections: 'Key pair type' with 'RSA' selected (radio button) and 'ED25519' as an option; and 'Private key file format' with '.pem' selected (radio button) and '.ppk' as an option. At the bottom, there are 'Cancel' and 'Create key pair' buttons.

2. Create a corresponding “*.ppk” file for use with FileZilla and PuTTY that matches the “*.pem” file in #1 above. (<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>).

To generate this:

- a. Launch “PuTTYgen”
- b. Under, “Type of key to generate:”, select RSA.
- c. Select “Load” then change to show “All Files (*.*)”. Find the “*.pem” file that you generated earlier and select OK.
- d. Select “Save private key”. The U-Net automation is not compatible with passphrases, so we have not used these.
- e. Enter a name for the “*.ppk” key. We have used to the same file root as the “*.pem” key to keep it clear which keys go together. Save this file in the same location as the “*.pem” key to make it easier to find later.
- f. Select Save