**Setting up an AWS server to run the MJOLNIR pipeline**

**Step-by-step tutorial**

**by Owen S. Wangensteen - July 2024**

**Contents:**

A: Setting up the hardware

A1. Create an account in AWS

A2. Go to the AWS Management Console and launch an instance with EC2

A3. Select and existing key pair or create a new key pair

A4. Launch the instance and start running it

A5. Log in to the server

A6. Resizing instances and EBS storage volumes

A7. Connecting to the AWS server using Filezilla

B: Installing the software needed for metabarcoding in the AWS instance

B1. Conda

B2. Python 2 (and pip**)**

B3. OBITools 2

B4. SWARM, VSEARCH and ecoPCR

B5. Installing R and R libraries

B6. Installing DUFA and the taxonomy databases

**A: Setting up the hardware**

**A1. Create an account in AWS**

Go to <https://aws.amazon.com/> and create an account. You will need a credit card and a mobile phone for authentication. Don’t worry! There is a free tier that will last for the first 12 months. But this includes just the use of a t2.micro or t3.micro instance (with very low computing capacity). Some tutorials for beginners here: <https://aws.amazon.com/getting-started/?nc1=f_cc>

**A2. Go to the AWS Management Console and launch a virtual machine with EC2**

Virtual machines are called "instances" in AWS. You can create a small instance (cheaper) and install all your software there. Then you can resize it to a bigger instance whenever you need larger computing capacity to run your pipelines.

- Go to the AWS Management Console: <https://us-east-2.console.aws.amazon.com/console/home?region=us-east-2>

- Click on "Launch a virtual machine With EC2".

- Select the operating system. I recommend to select the last version of Ubuntu: Ubuntu Server 20.04 LTS (HVM), SSD Volume Type.

- Choose an Instance Type. This choice is crucial for estimating the running costs. Of course, the higher number of cores, RAM and storage selected, the higher the costs. This link leads to the AWS pricing list: [https://aws.amazon.com/ec2/pricing/](https://aws.amazon.com/ebs/pricing/)

There is just a single instance type availabe for free tier: t2.micro in my region, with just 1 core, 1 Gb RAM and 8 Gb EBS-storage.

For metabarcoding we will typically need a server with 32-64 CPU cores, and at least 5 Gb RAM/core. For example, r5n.8xlarge has 32 cores, 256 Gb RAM. This will work.

But, for now, to save money, we can select the free t2.micro and install the software. Then press the "Review & Launch" button and then "Launch".

**A3. Select and existing key pair or create a new key pair**

In this point, we will create a new key pair. This will be needed at all time to access our instance. We will select RSA type, then write a name for the key-pair (e.g. metabar-porto), and then press "download key pair". By default, the pem file is stored in "Downloads" folder. It is a good idea to move this pem file to our local home folder.

**A4. Launch the instance and start running it**

After press the "Launch the instance" button, the instance will be created and it will start to run. The system will give you some advices on how to access your instances, like this:

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out](https://docs.aws.amazon.com/console/ec2/instances/connect/docs) how to connect to your instances.

Here are some helpful resources to get you started

* [How to connect to your Linux instance](https://docs.aws.amazon.com/console/ec2/instances/connect/docs)
* [Learn about AWS Free Usage Tier](https://aws.amazon.com/free/)
* [Amazon EC2: User Guide](https://docs.aws.amazon.com/console/ec2/launchinstance/status/user-guide)
* [Amazon EC2: Discussion Forum](https://forums.aws.amazon.com/forum.jspa?forumID=30)

While your instances are launching you can also

* Create status check alarmsto be notified when these instances fail status checks. (Additional charges may apply)
* [Create and attach additional EBS volumes](https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2" \l "Volumes:)(Additional charges may apply)
* [Manage security groups](https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2" \l "SecurityGroups:)

You can check the status of all your instances in the console:

[https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Home](https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2" \l "Home):

Go to the console, select the instance that is running. You can edit its name (e.g. "metabarcoding"). If you press the "Connect" button you will see several ways to connect. We are going to connect using a SSH client.

**A5. Log into the server**

We are going to connect using the username "ubuntu" which is the default superuser name in AWS Ubuntu servers. Before connecting for the first time, we have to change the attributes of the pem key file to make it private (this needs to be done just once). To do this:

Open an SSH client.

Locate your private key file. The key used to launch this instance is metabar-porto.pem

Run this command, if necessary, to ensure your key is not publicly viewable.

|  |
| --- |
| **chmod 400 metabar-key.pem** |

Connect to your instance using its Public DNS:

ec2-18-117-72-175.us-east-2.compute.amazonaws.com

Example:

|  |
| --- |
| **ssh -i metabar-key.pem ubuntu@ec2-18-117-72-175.us-east-2.compute.amazonaws.com** |

You can also use the IP (it will change every time you start the instance. E.g.

If public IP address is 18.117.72.175

|  |
| --- |
| **ssh -i metabar-key.pem ubuntu@18.117.72.175** |

**A6. Resizing instances and EBS storage volumes**

There is a list of available AWS instance types here: <https://aws.amazon.com/ec2/instance-types>

To resize an AWS instance, follow the instructions here:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/resize-ebs-backed-instance.html>

To increase the storage capacity of your ABS volume follow these instructions: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-modify-volume.html>

**A7. Connecting to the AWS server using Filezilla**

Filezilla is a free software for FTP/SFTP (file transfer). Download and install it in your laptop from here: <https://filezilla-project.org/download.php>

To connect to your running AWS instante, use these instructions: <https://www.zyxware.com/articles/5133/how-to-connect-to-amazon-ec2-using-filezilla>.

In summary:

Select Site Manager and create a new instance:

• Protocol: SFTP

• Host: IP

• Logon: Key File

• User: the user name used to access the EC2 Amazon instance. Example: user03

• Key file: the same .pem file you used for the connection (in the example porto03.pem)

**B. Installing software for metabarcoding**

**B1. Conda**

Python 2 is deprecated and it cannot be directly installed in Ubuntu using apt. But it can be installed from a Conda environment. First we will install Conda using the miniconda installer:

|  |
| --- |
| **mkdir -p ~/miniconda3**  **wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh -O ~/miniconda3/miniconda.sh**  **bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3**  **rm -rf ~/miniconda3/miniconda.sh**  **~/miniconda3/bin/conda init bash** |

**B2. Python 2 (and pip)**

Then we create a Conda environment with Python 2.7 and activate it:

|  |
| --- |
| **conda create -n py27 python=2.7**  **conda activate py27** |

**B3. OBITools 2**

Note that OBITools 2 runs only on Python 2.7 and it can not work on Python 3. So a Python 2.7

environment (which is already deprecated) is required for the installation. OBITools also requires a

particular old version of the sphinx package (v 1.4.8) and it does not work with newer versions.

The following set of commands would work in most systems for installing OBITools 2, based on

recommendations by Frederic Boyer (https://www.biostars.org/p/235898/)

|  |
| --- |
| **pip install sphinx==1.4.8 cython==0.29.21 docutils==0.16**  **wget "https://git.metabarcoding.org/obitools/obitools/repository/archive.tar.gz?ref=master"**  **tar -zxvf "archive.tar.gz?ref=master"**  **cd obitools-master-\***  **python2 setup.py build**  **python2 setup.py install** |

We can return to the base environment using:

|  |
| --- |
| **conda deactivate** |

And we can activate the py27 environment again anytime using:

|  |
| --- |
| **conda activate py27** |

**B4. SWARM, VSEARCH and ecoPCR**

|  |
| --- |
| **sudo apt-get install swarm**  **sudo apt-get install vsearch**  **sudo apt-get install ecopcr** |

**B5 Installing R and R libraries**

We will install R. Then we will install some dependencies that are needed for the devtools library:

|  |
| --- |
| **sudo apt install r-base-core**  **sudo apt-get install libssl-dev libcurl4-openssl-dev libxml2-dev** |

Now we will access R and we will install the Bioconductor’s Biostrings library. Then we will install the devtools library. Then we will use devtools for installing lulu and mjolnir. Finally, we will install the tidyverse suite:

|  |
| --- |
| **sudo R**  **> install.packages("BiocManager")**  **> BiocManager::install("Biostrings")**  **> install.packages("devtools")**  **> library(devtools)**  **> install\_github("tobiasgf/lulu")**  **> install\_github("uit-metabarcoding/MJOLNIR")**  **> install.packages("tidyverse")** |

**Additional notes on installing R libraries:**

There are four different ways to install R libraries (packages) from the different repositories where they can be found.

1.- The most used and standard libraries are available from the CRAN repository (<https://cran.r-project.org/>).

2.- Many libraries for bioinformatics are in Bioconductor (<https://www.bioconductor.org/>).

3.- Other libraries which are in current development can be downloaded and installed directly from GitHub (<https://github.com/qinwf/awesome-R>).

4.- Finally you can install a local library manually if you are given a master source file.

**B5-1 Installing standard libraries from CRAN**

These libraries may be easily installed using the function install.packages() from within R or R-Studio. For example:

|  |
| --- |
| > install.packages(c("CHNOSZ","vegan","MASS"))  > install.packages("seqinr")  > install.packages(c("ape","optparse")) |

**B5-2 Installing libraries from Bioconductor 3.8 (R version > R 3.5.0)**

First, you have to install the R the Bioconductor installer library, which is now called “BiocManager”

|  |
| --- |
| > install.packages("BiocManager") |

Then, to install the core packages of Bioconductor, you have to write:

|  |
| --- |
| > BiocManager::install() |

If you want to install additional libraries from Bioconductor (not included in the core package) you have to use the same function and the name of the library.

|  |
| --- |
| > BiocManager::install("Biostrings")  > BiocManager::install("metagenomeSeq") |

**B5-3 Installing libraries from GitHub**

You can install directly from GitHub libraries which are still in development. But first, you will have to install some development tools from the terminal, before proceed.

|  |
| --- |
| sudo apt-get update  sudo apt-get install -y libssl-dev  sudo apt-get install -y libcurl4-openssl-dev  sudo apt-get install -y libxml2-dev |

Then, you will go into R again and you will install the devtools package from CRAN:

|  |
| --- |
| R  > install.packages("devtools") |

Once this is installed, you need to load the devtools library and then you can use the install\_github("author/package") function. For example:

|  |
| --- |
| > library(devtools)  > install\_github("hadley/dplyr")  > install\_github("ropensci/bold")  > install\_github("ropensci/taxize")  > install\_github("ropensci/rentrez")  > install\_github("ropensci/rfishbase")  > install\_github("cran/vioplot")  > install\_github("cran/XML") |

**B5-4 Installing libraries manually from local master files**

Finally, for installing libraries by hand from the master source file, you can use the install.packages function with the following syntax: install.packages("path\_to\_package\_file", repos = NULL, type="source", dependencies=T). For example, we are going to install the PrimerMiner package.

First, we have to download and unzip the master file:

|  |
| --- |
| wget https://github.com/VascoElbrecht/PrimerMiner/archive/master.zip  unzip ~/Downloads/PrimerMiner-master.zip |

The folder PrimerMiner-master will be created. Then we will go into R and we will install the package from its source:

|  |
| --- |
| R  > install.packages("~/Downloads/PrimerMiner-master/PrimerMiner", repos = NULL, type="source", dependencies=T) |

**B6 Installing DUFA and the taxonomy databases**

|  |
| --- |
| **mkdir ~/taxo**  **cd ~/taxo**  **wget --no-check-certificate 'https://docs.google.com/uc?export=download&id=1U6D7P7feUxpVK-j1aoUdRQVz-wWZr6tG' -O DUFA\_COLR\_20210723.fasta**  **wget --no-check-certificate 'https://docs.google.com/uc?export=download&id=1DHfLgac1P5fl69Qi4agTv6bbmPtBW6TO' -O order.complete.csv**  **wget --no-check-certificate 'https://docs.google.com/uc?export=download&id=1uNewKn25Ca8GECYV9ZyuupPtpcukuxmw' -O genus\_to\_family.csv**  **wget --no-check-certificate 'https://docs.google.com/uc?export=download&id=1WVsLjofkyD8sJ-RmfZBDmVVgfr7EdsJA' -O family\_to\_order.csv**  **wget --no-check-certificate 'https://docs.google.com/uc?export=download&id=1YJv2uqXocDRXf62VK5UaSxNk-MtrOShv' -O taxo\_NCBI\_20210720.ldx**  **wget --no-check-certificate 'https://docs.google.com/uc?export=download&id=1\_v3e27JgqLfQaANI1KQgRx1-e7BcVm-2' -O taxo\_NCBI\_20210720.tar.xz**  **tar -xf taxo\_NCBI\_20210720.tar.xz**  **rm taxo\_NCBI\_20210720.tar.xz** |