

WEEK 4 ASSIGNMENT 1

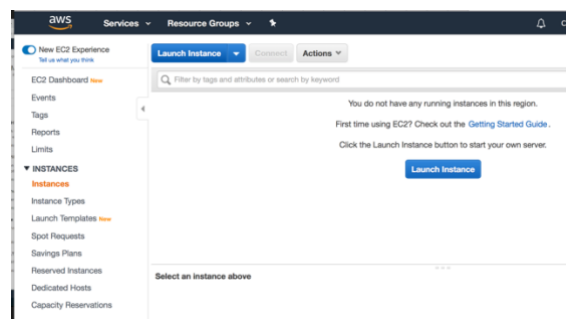
Large-Scale Data Storage Systems – DATA-5400 | Spring 2020

Christina Morgenstern

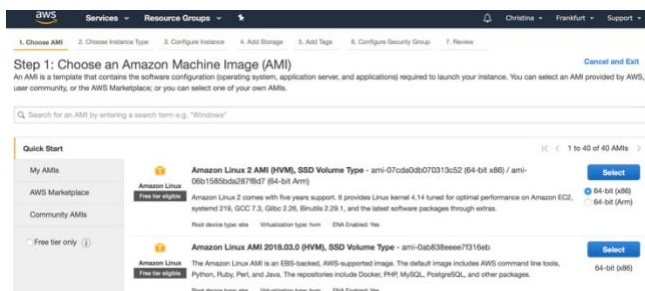
In this assignment, I created a Linux VM in the cloud and ran Linux and some commands on the newly created VM.

Using the Atom editor, I created three text files with .txt extension and saved them in my course directory. The contents of these files are a one-line sentence and the same in each case.

In order to create a Linux VM, I go to my AWS Management Console and select the EC2 service. On the left-hand menu, I choose Instances and then click the blue Launch Instance button, selecting launch instance. As for the region, I stick with the Europe (Frankfurt) eu-central-1 region, which is closest to my home.



In the next step, you have to choose the type of Amazon Machine Image (AMI), a template containing the software configuration required to launch the instance. To create a Linux VM, I selected the Amazon Linux 2, Free tier, instance.



I selected the General Purpose t2.micro type instance with 1 vCPU, 2.5 GHz Intel Xeon and 1 GiB memory of EBS only.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPU, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

Next, further instance details were configured. Like the number of instances which was set to 1.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 [Launch into Auto Scaling Group](#)

Purchasing option: ☐ Request Spot instances

Network: vpc-72cb0918 (default) [Create new VPC](#)

Subnet: No preference (default subnet in any Availability Zone) [Create new subnet](#)

Auto-assign Public IP: ☐ Use subnet setting (Enable)

Placement group: ☐ Add instance to placement group

Capacity Reservation: Open [Create new Capacity Reservation](#)

IAM role: None [Create new IAM role](#)

Shutdown behavior: ☐ Stop [Create new IAM role](#)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

The storage was set to 10 GB.

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	Snapshot: snap-07c0d936ca000c0e5	10	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

Tags allow you to identify instances. I used tags that identify these VMs with the Large Scale Storage course.

Step 5: Add Tags

A tag consists of a name-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (255 characters maximum)	Value (255 characters maximum)	Instances	Volumes
This resource currently has no tags.			

[Add Tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

aws Services Resource Groups Christina Frankfurt Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

[Add Rule](#)

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

aws Services Resource Groups Christina Frankfurt Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instances' security. Your security group, **launch-wizard-1**, is open to the world. Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-07cda0db070313c52

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Bnutils 2.29.1, and the latest software packages through extras.

Root Device Type: etc Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

[Cancel](#) [Previous](#) [Launch](#)

In Step 7, when reviewing the details of the instance, a key pair for authorization of the instance needs to be generated. The key pair file needs to be downloaded as .pem file and was saved on my drive in the same course directory.

aws Services Resource Groups Christina Frankfurt Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Please review your instance launch details.

Improve your instances' security. Your instances may be accessible from any IP address. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-07cda0db070313c52

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Bnutils 2.29.1, and the latest software packages through extras.

Root Device Type: etc Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs
t2.micro	Variable

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. [Learn more about removing existing key pairs from a public AMI.](#)

☒ Choose an existing key pair

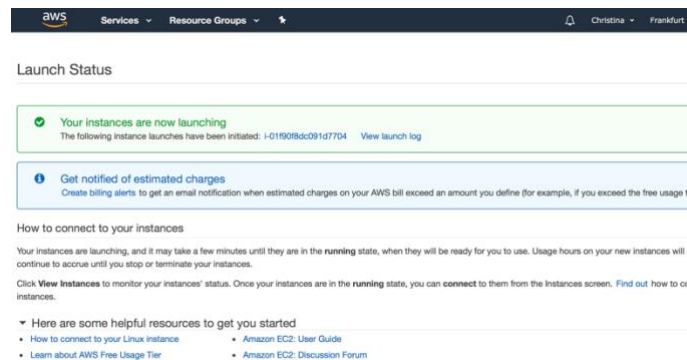
☒ Create a new key pair

☐ Proceed without a key pair

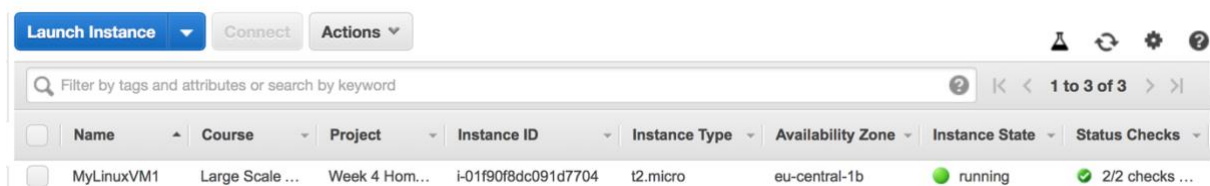
No key pairs found

You don't have any key pairs. Please create a new key pair by selecting the **Create a new key pair** option above to continue.

[Cancel](#) [Launch Instances](#)



I successfully created a Linux VM, called MyLinuxVM1, using AWS EC2 service. Selecting the VM on the left-hand side lets you explore the details of this instance. Under the description tab, you can find the private and public IP addresses.



To connect to MyLinuxVM1, I opened the Terminal window on my Mac Laptop and followed the instructions in the AWS documentation (<https://docs.aws.amazon.com/quickstarts/latest/vmlaunch/step-2-connect-to-instance.html#sshclient>) using SSH:

1. Using the `cd` command, I navigated to the directory containing the private key file and the created text files.
2. The following `chmod` command makes sure that the private key file isn't publicly viewable:
`chmod 400 mykeypair.pem`
3. With the following SSH command, I connected to my instance:
`ssh -i "mykeypair.pem" ec2-user@ec2-35-158-247-96.eu-central-1.compute.amazonaws.com`

These commands can also be found in the EC2 console window upon selecting the instance and pressing the connect button.

```
(base) Christinas-MacBook-Pro:Week_4 Christina$ chmod 400 mykeypair.pem
(base) Christinas-MacBook-Pro:Week_4 Christina$ ssh -i "mykeypair.pem" ec2-user@ec2-35-158-247-96.eu-central-1.compute.amazonaws.com

 _ | _ | _ )
 _ | ( /   /   Amazon Linux 2 AMI
 _ | \ | _ |

https://aws.amazon.com/amazon-linux-2/
7 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-39-22 ~]$
```

```

  _ | _ | _ )
  _ | ( /
  _ | \ | _ |
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
7 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-39-22 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-39-22 ~]$ ls
[ec2-user@ip-172-31-39-22 ~]$

```

Using the SCP client on my macOS, I transferred the previously created text files to the Linux VM. For that, I opened another terminal window and navigated to the directory with the text files and the secure keys. Then using the following general SCP command, the files were copied to the VM:

```
scp -i path/to/key file/to/copy user@ec2-xx-xx-xxx-xxx.compute-1.amazonaws.com:path/to/file
```

Successful transfer of text1.txt file:

```

(base) Christinas-MacBook-Pro:Week_4 Christina$ scp -i mykeypair.pem text1.txt ec2-user@ec2-35-158-247-96.eu-central-1.compute.amazonaws.com:/home/ec2-user
text1.txt
100% 86 2.6KB/s 00:00

```

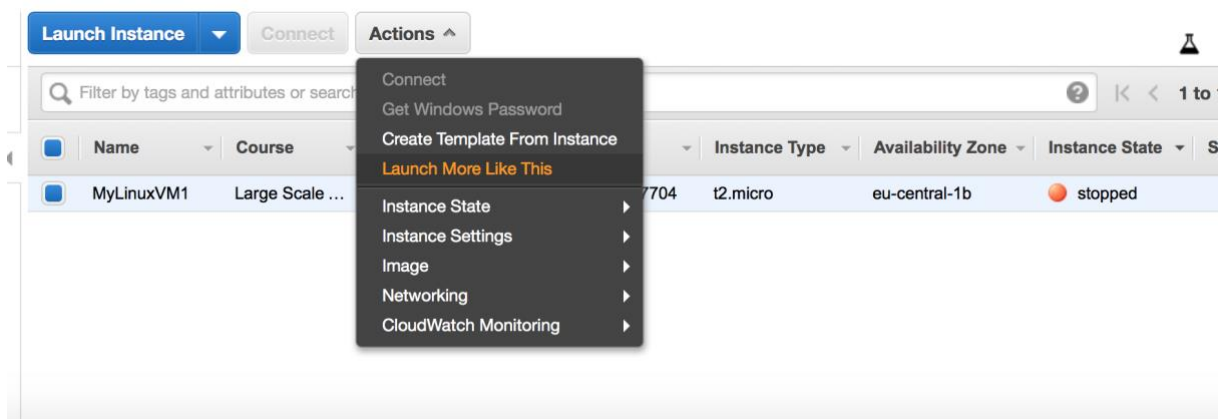
And verification of the transfer of three text files in the terminal window connected to the VM using the `ls` command.

```

[ec2-user@ip-172-31-39-22 ~]$ ls
text1.txt text2.txt text3.txt
[ec2-user@ip-172-31-39-22 ~]$

```

A second Linux virtual machine, named MyLinuxVM2, was created using the Actions button and the Launch More Like This option.



	Name	Course	Project	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
<input type="checkbox"/>	MyLinuxVM1	Large Scale ...	Week 4 Hom...	i-01f90f8dc091d7704	t2.micro	eu-central-1b	running	2/2 checks ...
<input type="checkbox"/>	MyLinuxVM2	Large Scale ...	Week 4 Hom...	i-0f34ee24ffb5229c9	t2.micro	eu-central-1b	running	2/2 checks ...

In a third terminal window, I connected to MyLinuxVM2.

```
Last login: Mon Feb 10 22:52:10 on ttys002
(base) Christinas-MacBook-Pro:~ Christina$ cd /Users/Christina/01_Files/19_Master/studies/02_Course/work/07_LargeScaleDataStorage/Week_4
(base) Christinas-MacBook-Pro:Week_4 Christina$ ssh -i "mykeypair.pem" ec2-user@ec2-35-158-121-11.eu-central-1.compute.amazonaws.com
The authenticity of host 'ec2-35-158-121-11.eu-central-1.compute.amazonaws.com (35.158.121.11)' can't be established.
ECDSA key fingerprint is SHA256:d3Bh+R5wM2HUTSLyF108XGNT5wCFzlt76gvEyc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-35-158-121-11.eu-central-1.compute.amazonaws.com,35.158.121.11' (ECDSA) to the list of known hosts.

 _ _ _ _ _
| | | | |
|_|_|_|_|_| Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
7 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-44-20 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-44-20 ~]$ ls
[ec2-user@ip-172-31-44-20 ~]$
```

Transfer of the keypair file to VM2 (and accidental transfer of text3.txt file, which was immediately removed using the `rm` command.)

```
(base) Christinas-MacBook-Pro:Week_4 Christina$ scp -i mykeypair.pem text3.txt mykeypair.pem ec2-user@ec2-35-158-121-11.eu-central-1.compute.amazonaws.com:/home/ec2-user
text3.txt                                100% 86      2.6KB/s   00:00
mykeypair.pem                          100% 1696    47.9KB/s   00:00
```

Verification that the keypair file resides within the VM2.

```
[ec2-user@ip-172-31-44-20 ~]$ ls
mykeypair.pem
[ec2-user@ip-172-31-44-20 ~]$
```

Transferring the three text files from MyLinuxVM1 to the MyLinuxVM2 using the `scp` command.

```
[ec2-user@ip-172-31-39-22 ~]$ scp -i mykeypair.pem text1.txt ec2-user@ec2-35-158-121-11.eu-central-1.compute.amazonaws.com:/home/ec2-user
text1.txt                                100% 86      72.6KB/s   00:00
[ec2-user@ip-172-31-39-22 ~]$ scp -i mykeypair.pem text2.txt ec2-user@ec2-35-158-121-11.eu-central-1.compute.amazonaws.com:/home/ec2-user
text2.txt                                100% 86      70.5KB/s   00:00
[ec2-user@ip-172-31-39-22 ~]$ scp -i mykeypair.pem text3.txt ec2-user@ec2-35-158-121-11.eu-central-1.compute.amazonaws.com:/home/ec2-user
text3.txt                                100% 86      73.2KB/s   00:00
[ec2-user@ip-172-31-39-22 ~]$
```

Verification that the text files were copied to MyLinuxVM2 using the `ls` command.

```
[ec2-user@ip-172-31-44-20 ~]$ ls
mykeypair.pem text1.txt text2.txt text3.txt
[ec2-user@ip-172-31-44-20 ~]$
```

I tested several Linux commands using MyLinuxVM2, like the following:

`ss` command displays network statistics. I have used the `ss` command without any arguments as well as with different arguments (<https://www.linux.com/tutorials/introduction-ss-command/>).

The `wc` command outputs the number of lines, number of words, number of bytes and the filename (<https://shapeshed.com/unix-wc/>)

```
$ wc text1.txt
1 16 86 text1.txt
```

The `cat` command displays the file contents:

```
[ec2-user@ip-172-31-44-20 ~]$ wc text1.txt
1 16 86 text1.txt
[ec2-user@ip-172-31-44-20 ~]$ cat text1.txt
This is a text file created for week 4 assignment of large scale data storage course.
[ec2-user@ip-172-31-44-20 ~]$
```

The `history` command shows the previously used commands:

```
[ec2-user@ip-172-31-44-20 ~]$ history
1  pwd
2  ls
3  rm text3.txt
4  ls
5  ss
6  ss -l
7  wc text1.txt
8  cat text1.txt
9  history
[ec2-user@ip-172-31-44-20 ~]$
```

The `which` command helps to locate executable files, like where python resides.

```
[ec2-user@ip-172-31-44-20 ~]$ which python
/usr/bin/python
```

Using the `exit` command, the terminal windows were closed, and the instances stopped within the AWS console window.