

## WEEK 4 ASSIGNMENT 2

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

Christina Morgenstern

In this hands-on assignment, I am creating a Windows VM in the cloud using AWS EC2 service and connect to it using a remote desktop connection.

To launch a new instance, in the EC2 console window, go to Instances in the left-hand menu and press the blue button Launch Instance.

For the Windows VM, I selected the Free tier Microsoft Windows Server 2019 Base.

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

Step 1: Choose an Amazon Machine Image (AMI) Cancel and Exit

**Amazon RDS** consuming database management tasks. With RDS, you can easily deploy **Amazon Aurora**, **MariaDB**, **MySQL**, **Oracle**, **PostgreSQL**, and **SQL Server** databases on AWS. **Aurora** is a MySQL- and PostgreSQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. [Learn more about RDS](#)

[Launch a database using RDS](#)

**Microsoft Windows Server 2019 Base** - ami-0ab54e065e88f7520 Select

**Windows** Microsoft Windows 2019 Datacenter edition. [English] 64-bit (x86)

Free tier eligible Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

As for Instance Type, I selected the General purpose and free tier eligible t2.micro.

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

Step 2: Choose an Instance Type

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

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	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 Free tier eligible	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
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Launch the instance and use the same key pair as previously generated.

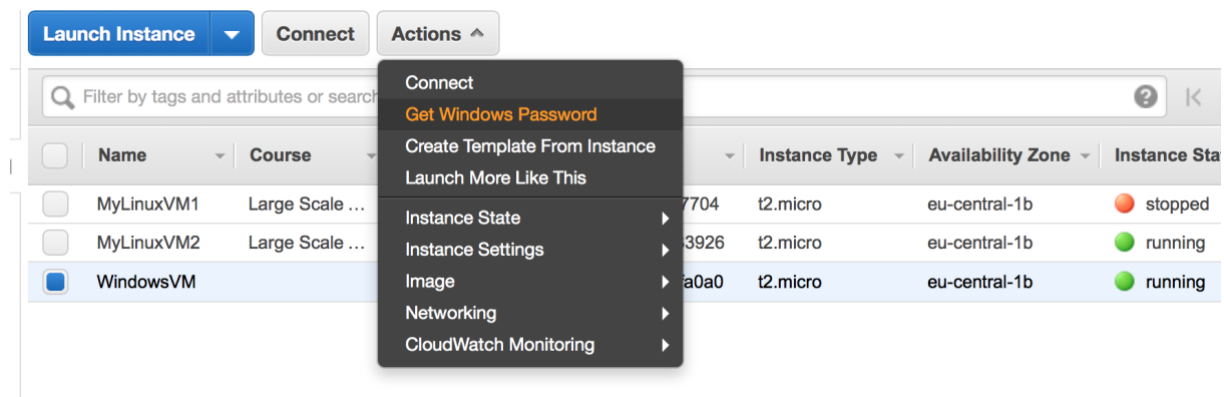
The Windows VM is successfully running in the Europe Frankfurt region.

Launch Instance Connect Actions

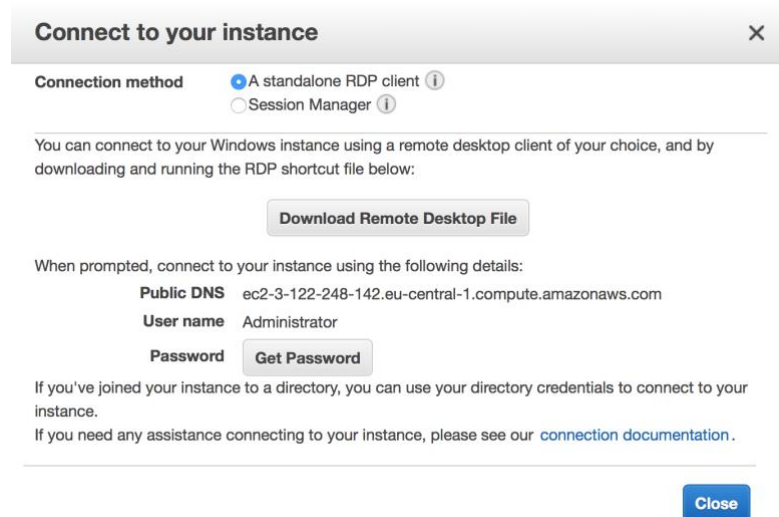
Filter by tags and attributes or search by keyword

	Name	Course	Project	Instance ID	Instance Type	Availability Zone	Instance State
<input type="checkbox"/>	MyLinuxVM1	Large Scale ...	Week 4 Hom...	i-01f90f8dc091d7704	t2.micro	eu-central-1b	stopped
<input type="checkbox"/>	MyLinuxVM2	Large Scale ...	Week 4 Hom...	i-0f34ee24ffb5229c9	t2.micro	eu-central-1b	stopped
<input type="checkbox"/>	WindowsVM	Large Scale ...	Week 4 Hom...	i-017c490e91dcfa0a0	t2.micro	eu-central-1b	running

Connecting to the Windows VM requires a Remote Desktop Client for Mac and a Windows password. The password is created as follows and with the help of the previously obtained key pair.



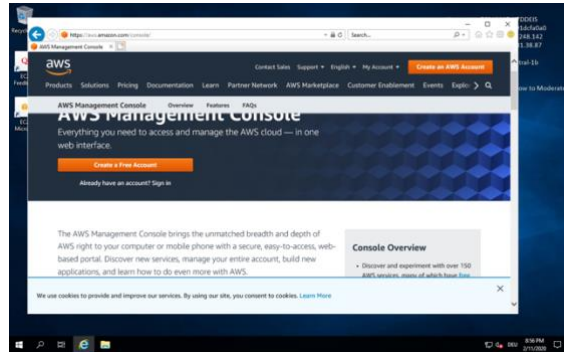
Selecting the Windows VM and pressing Connect opens a window that creates the security credentials for connecting to the VM. I also downloaded the Remote Desktop File.



From the App-Store, I downloaded the following Remote Desktop Client:



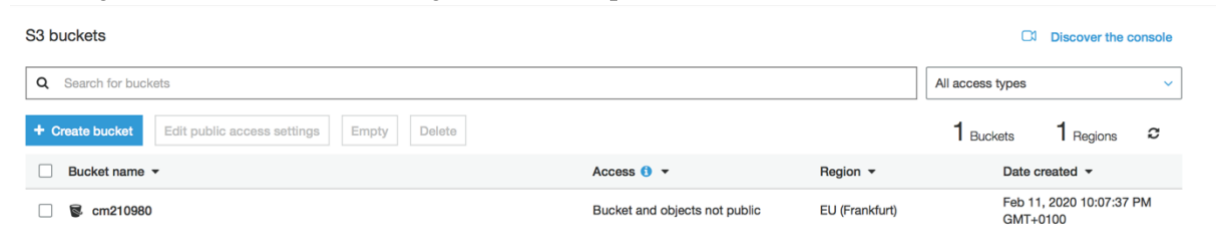
I opened the Remote Desktop File that was downloaded from the EC2 console. This way of connecting to the VM failed. But when starting the Microsoft Remote Desktop app and creating a new desktop through entering the previously generated Administrator and Password, I could connect to the Windows VM. And internet connection was also verified through browsing the web using the Internet Explorer (see for step-by-step guidance <https://thebackroomtech.com/2018/11/08/how-to-connect-to-a-windows-ec2-instance-from-mac/>)



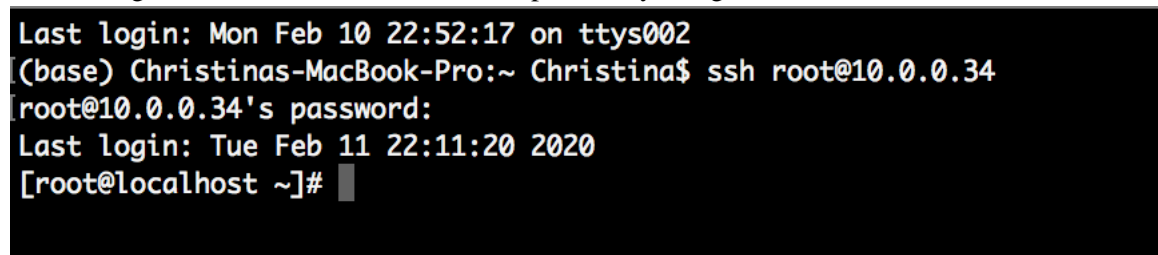
Transferring files to the Windows VM:

Under the Identity and Access Management, I enabled the access and secret keys to make programmatic calls to AWS from AWS CLI.

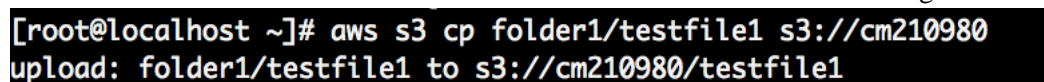
Creating a S3 bucket was done using the S3 desktop service.



Connecting to the CentOS VM was done as previously using SSH.



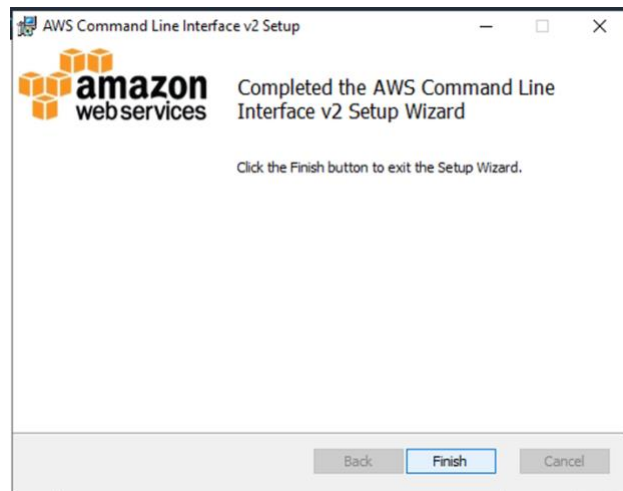
Transfer of a testfile 1 from the CentOS VM to the S3 bucket cm210980 using CLI.



Verification that the testfile shows up in the s3 bucket in the console window.



For downloading the file in the s3 bucket using the AWS CLI, I needed to install the AWS CLI version 2 for windows.



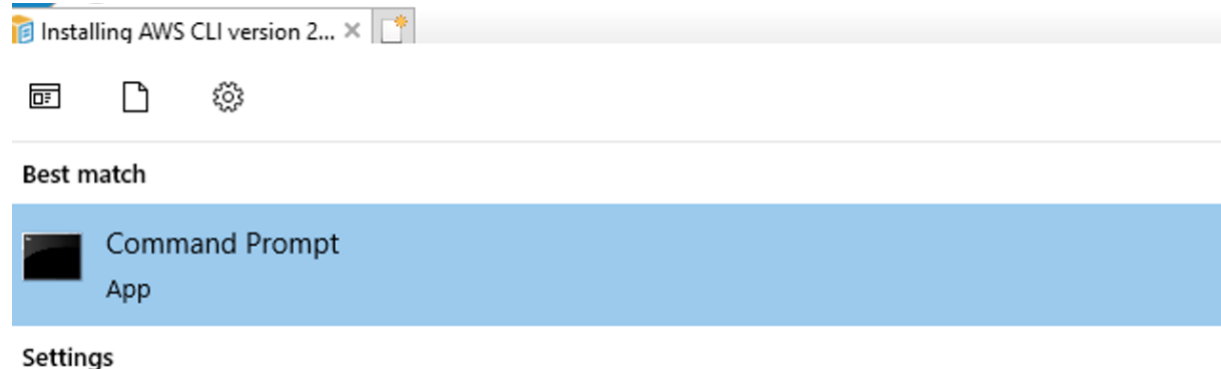
Successful installation of the AWS CLI and verification of version.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.973]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>aws --version
aws-cli/2.0.0 Python/3.7.5 Windows/10 botocore/2.0.0dev4

C:\Users\Administrator>
```

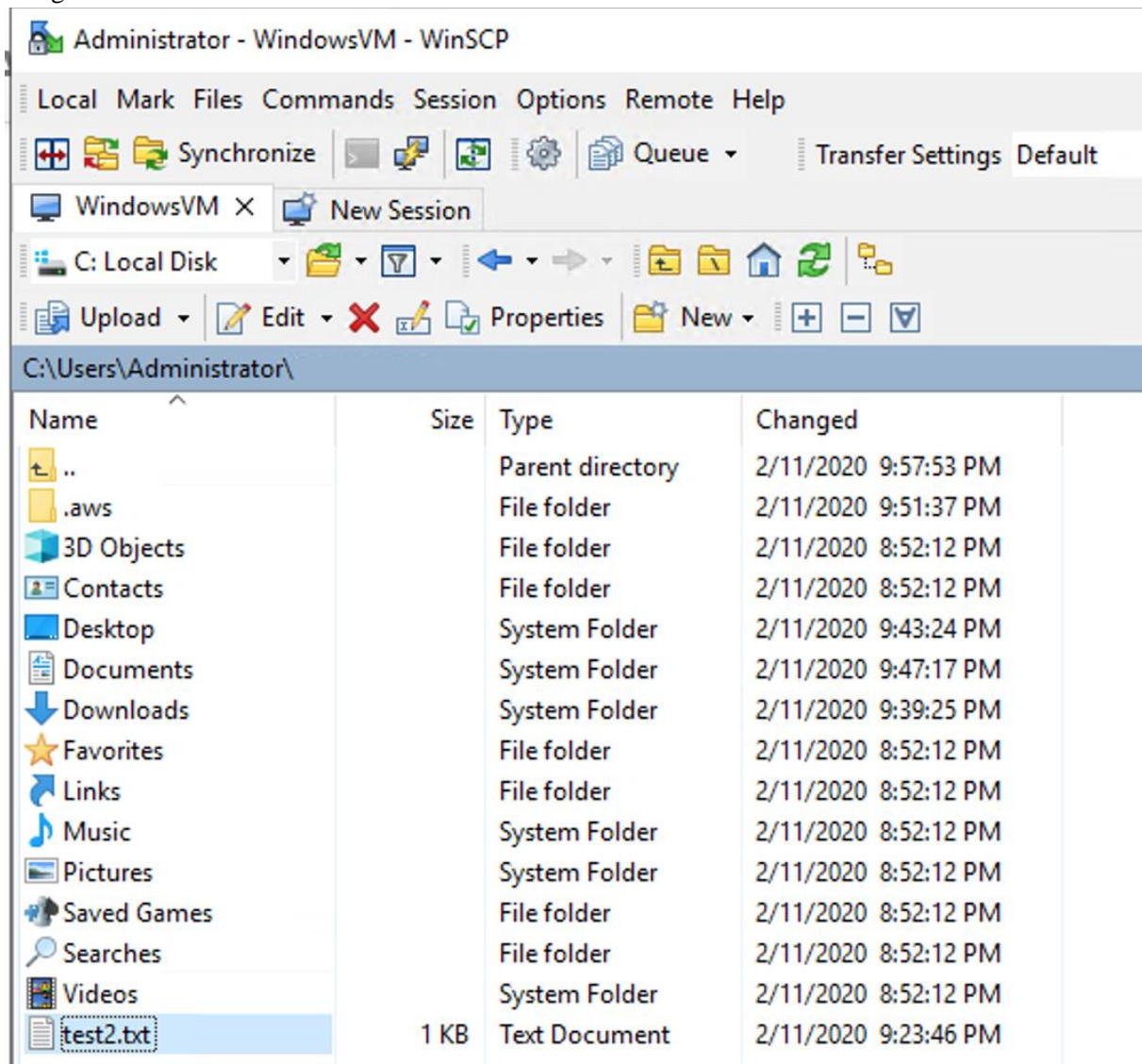
Using the Windows Command Prompt for connecting to the s3 bucket and downloading the file.



After configuring the instance by entering the access key and secret key as well as the preferred location, I could use the AWS CLI commands as previously and downloaded the testfile1 from the cm210980 bucket.

```
C:\Users\Administrator>aws s3 cp s3://cm210980/testfile1 test2.txt
download: s3://cm210980/testfile1 to .\test2.txt
```

Testfile1 appears as test2.txt because during the download it was renamed. I am showing the location of the test2.txt file in the WinSCP window. I have also installed WinSCP on my Windows VM and used it to connect to the s3 bucket. Since I am not a Windows user, I wanted to get the experience of also using WinSCP.



I disabled the keys, deleted the s3 bucket and logged off the VMs.