



Exercise Sheet 1

Distributed Information Systems (Spring Semester 2024)

Anastasiya Merkusova

Raphael Waltenpül

July 4, 2024

Exercise 1.1 Creating EC2 Instances (4 points)

1. Login to the AWS console

You've been invited to participate in a class at AWS Academy . The class is called AWS Academy Learner Lab [72713]. Course role: Student


Name: **Anastasiya Merkushova**
Email: **a.merkushova@unibas.ch**
Username: **none**

You'll need to register with Canvas before you can participate in the class.

Get Started

[Home](#)
[Modules](#)
[Discussions](#)
[Grades](#)

AWS Academy Learner Lab [72713]



AWS Academy Learner Lab provides a long-running sandbox environment for ad hoc exploration of AWS services. Within this class, students will have access to a **restricted set of AWS services**. Not all AWS documentation walk-through or sample labs that operate in an AWS Production account will work in the Learner Lab environment. You will retain access to the AWS resources set up in this environment for the duration of this course. We limit your budget (\$100USD), so you should exercise caution to prevent charges that will deplete your budget too quickly. If you exceed your budget, you will lose access to your environment and lose all of your work.

Each session lasts for 4 hours by default, although you can extend a session to run longer by pressing the start button to reset your session timer. At the end of each session, any resources you created will persist. However, we automatically shut EC2 instances down. Other resources, such as RDS instances, keep running. Keep in mind that we do not stop some AWS features, so they can still incur charges between sessions. For example, an Elastic Load Balancer or a NAT. You may wish to delete those types of resources and recreate them as needed to test your work during a session. You will have access to this environment for the duration of the class that you are enrolled in. When the class ends, your access to the learner lab will also end.

Course and Exercise Information
For course and exercise information, please see the Distributed Information Systems workspace in Adam.

[View Course Stream](#)
[View Course Calendar](#)
[View Course Notifications](#)

To Do
Nothing for now


Recent Feedback
Nothing for now

▼ AWS Academy Learner Lab

[Launch AWS Academy Learner Lab](#)

ALLv1EN-LT113-7... > Modules > AWS Academy Le... > Launch AWS Academy Learner Lab

[Home](#)
[Modules](#)
[Discussions](#)
[Grades](#)

AWS 

Used \$0 of \$100

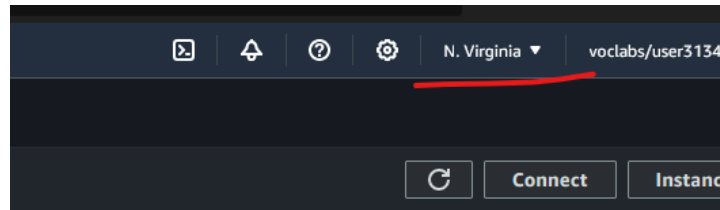
00:00

[Start Lab](#) [End Lab](#) [AWS Details](#) [Readme](#) [Reset](#) [X](#)

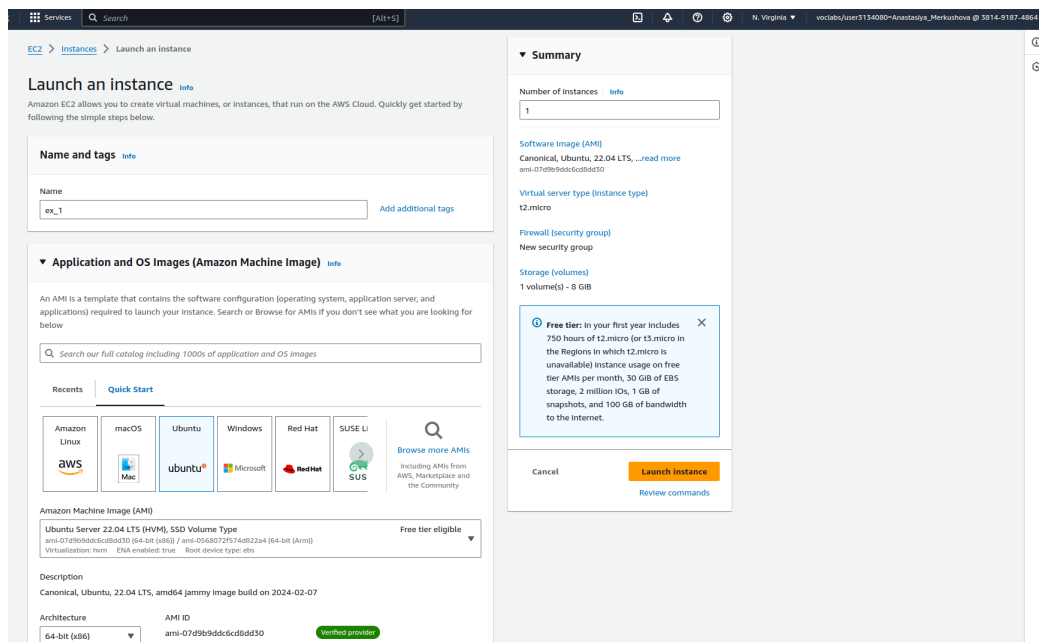
EN-US

Learner Lab

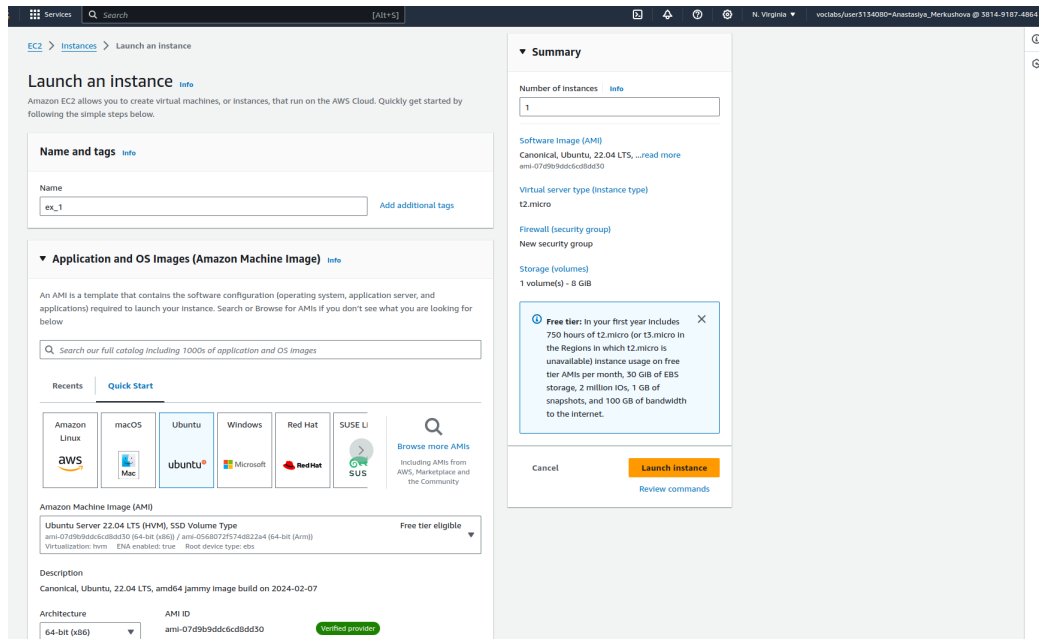
2. Select the US East (N. Virginia) AWS Region.



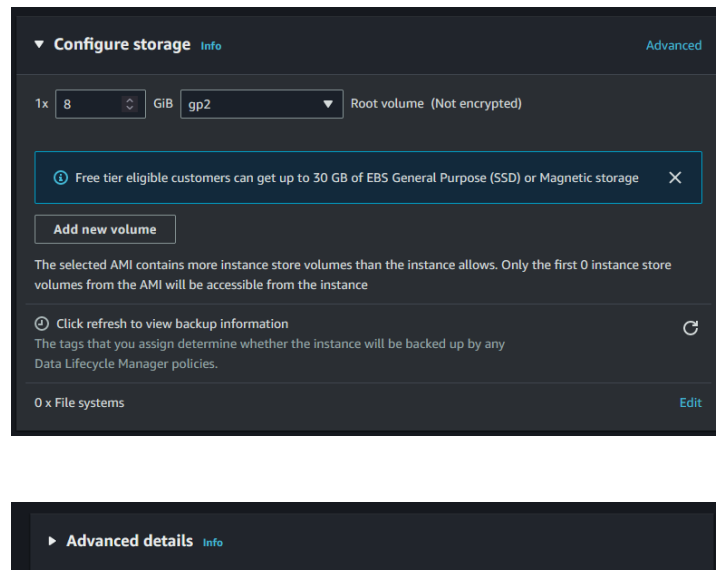
3. Select Services → Compute → EC2 on the top menu, click on the Launch Instance button, select the latest Ubuntu Server AMI, select t2.micro instance type.



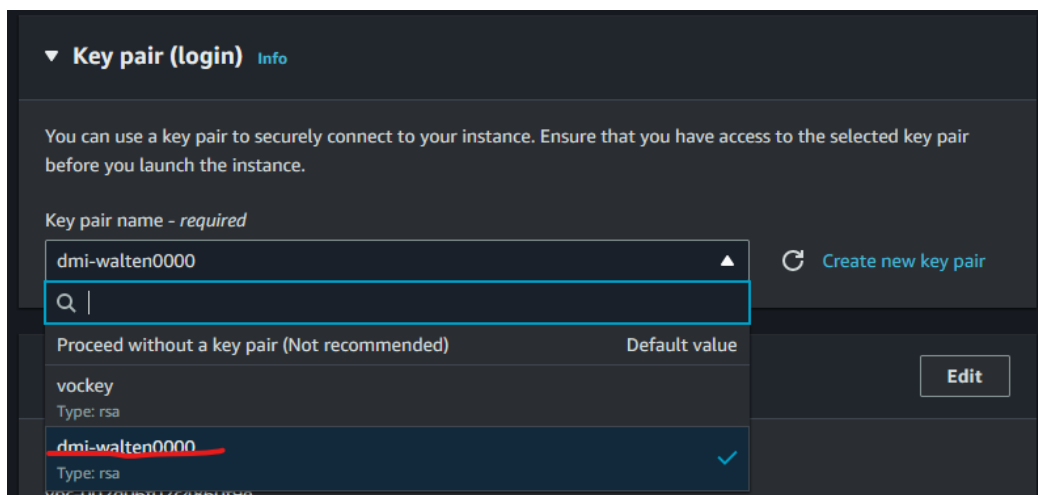
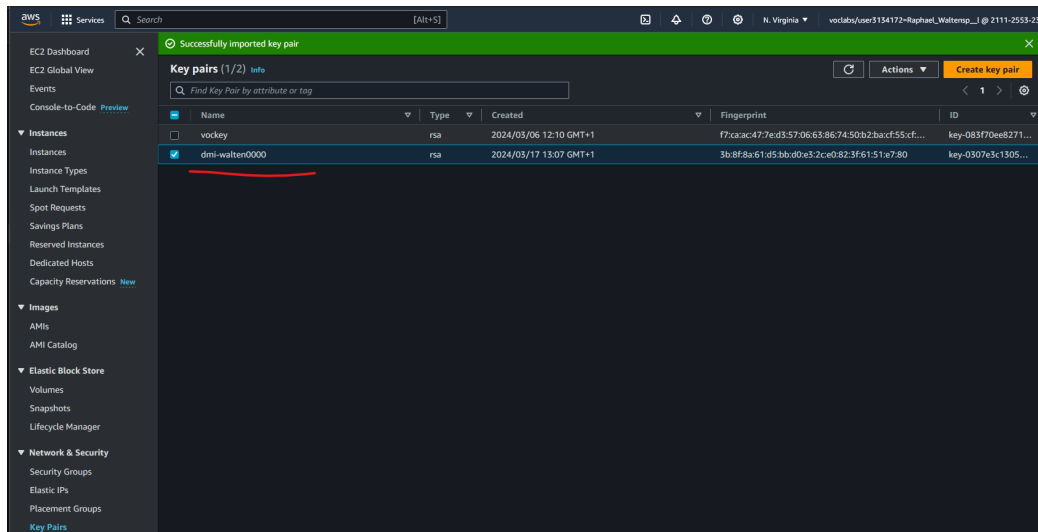
4. Further we set the following network settings for accessing with ssh:



5. The additional configurations were kept by default:



6. Further we imported an ssh key, and adding it to a security group:



7. With that we were able to connection over the shell:

```
(base) PS C:\Users\walten0000> ssh ubuntu@ec2-52-204-230-168.compute-1.amazonaws.com
The authenticity of host 'ec2-52-204-230-168.compute-1.amazonaws.com (52.204.230.168)' can't be established.
ED25519 key fingerprint is SHA256:/k1XJDZLwhxZwuq+t57FydC54aCGAqFDV0H+Nf05wi4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-204-230-168.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1014-aws x86_64)
```

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-31-121:~$
```

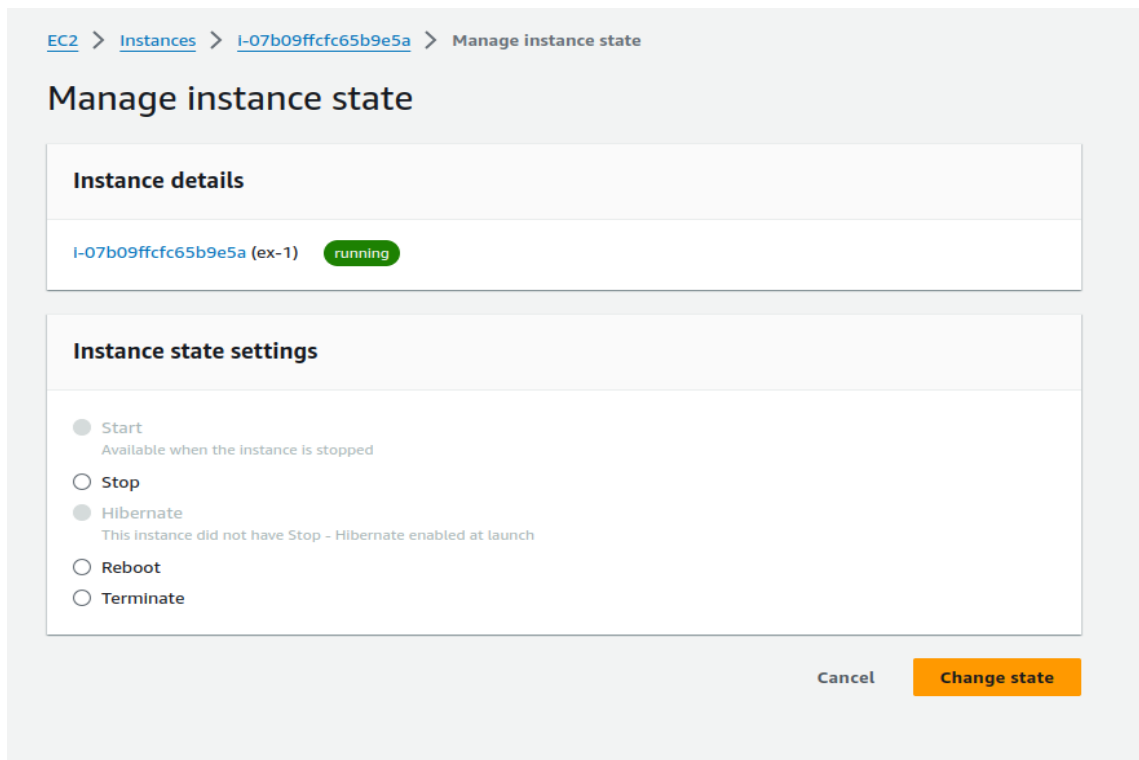
Exercise 1.2 Managing EC2 Instances (2 points)

1. Connect and login to the previously started EC2 instances using SSH:

Listing 1: Connect to the instance

```
1 # ssh -i /path/key-pair-name.pem  
   instance-user-name@instance-public-dns-name  
2 ssh -i "ex_1.pem"  
   ubuntu@ec2-34-238-246-47.compute-1.amazonaws.com
```

2. Use the AWS console or an IDE extension to explore and document the options for managing your instances. There are several options to do:



a) **Create an EC2 Instance:**

This option allows you to launch a new Amazon Elastic Compute Cloud (EC2) instance, specifying details like the Amazon Machine Image (AMI), instance type, key pair, security groups, and other configuration settings.

b) **View Your Instances:**

Once you have launched instances, this option lets you see a list of all your running, stopped, or terminated instances, providing information such as instance ID, state, public IP address, and other details.

c) **Connect to Your Instance:**

After launching an EC2 instance, this option allows you to connect to it using Secure Shell (SSH) for Linux instances or Remote Desktop Protocol (RDP) for Windows instances, requiring the key pair or password associated with the instance.

d) **Reboot Your Instance:**

This option initiates a reboot of your running instance, restarting the operating system without stopping the instance. It can be useful for applying updates or troubleshooting.

e) **Stop Your Instance:**

Stopping an instance puts it into a stopped state, temporarily halting its operation to avoid ongoing charges. The instance can be started again later.

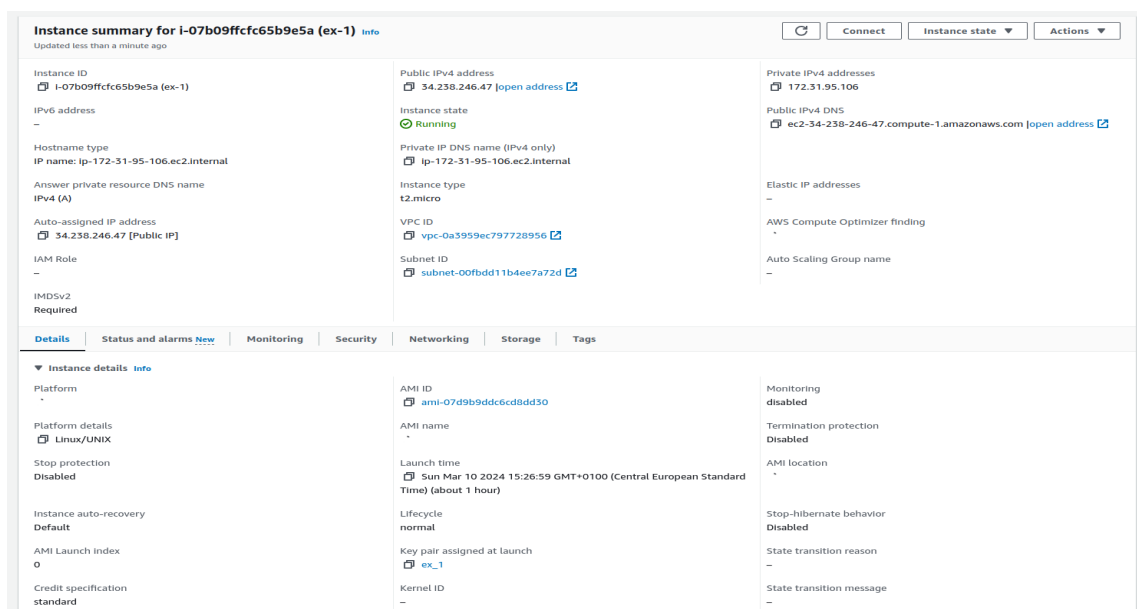
f) **Start Your Instance:**

If you've previously stopped an instance, this option allows you to start it again, rebooting and restoring it to the state it was in when stopped.

g) **Terminate Your Instance:**

This option permanently deletes the selected instance, terminating associated Amazon EBS (Elastic Block Store) volumes by default. This action is irreversible and removes the instance and associated data.

3. Via the instance detail page or via the API you can get the DNS name and the IP address of your instance(s):



Instance summary for i-07b09ffcf65b9e5a (ex-1) Info

Updated less than a minute ago

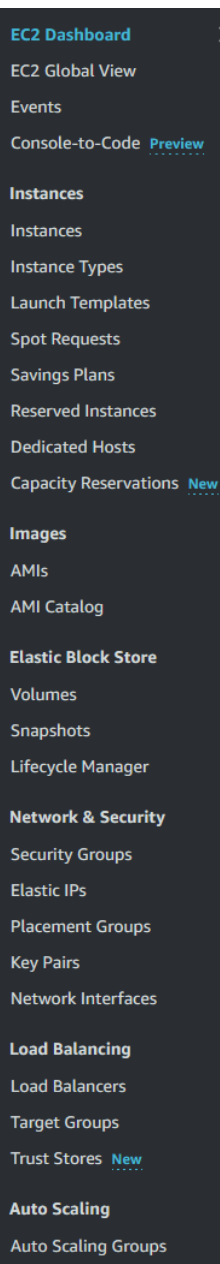
Connect Instance state Actions

Instance ID i-07b09ffcf65b9e5a (ex-1)	Public IPv4 address 34.238.246.47 open address	Private IPv4 addresses 172.31.95.106
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-34-238-246-47.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-95-106.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-95-106.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding -
Auto-assigned IP address 34.238.246.47 [Public IP]	VPC ID vpc-0a3959ec797728956	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-00fbd11b4ee7a72d	
IMDSv2 Required		

Details Status and alarms New Monitoring Security Networking Storage Tags

▼ Instance details Info

Platform Linux/UNIX	AMI ID ami-07d9b9ddc6cd8dd30	Monitoring disabled
Platform details Linux/UNIX	AMI name -	Termination protection Disabled
Stop protection Disabled	Launch time Sun Mar 10 2024 15:26:59 GMT+0100 (Central European Standard Time) (about 1 hour)	AMI location -
Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior Disabled
AMI Launch Index 0	Key pair assigned at launch ex_1	State transition reason -
Credit specification standard	Kernel ID -	State transition message -



4. The sidebar of the EC2 dashboard provides options for manage your instances such as:
- **EC2 Dashboard:** A central place in the AWS Management Console that provides a quick view of EC2 resources including running instances, alarms, volumes, IP addresses, and more.
 - **EC2 Global View:** Allows viewing and managing EC2 resources across multiple regions from a single interface.
 - **Events:** Shows scheduled events related to EC2 instances such as reboots, maintenance, and retirement.
 - **Console-to-Code:** Helps generate SDK code templates for actions performed in the AWS Management Console, simplifying scripting and automation.
 - **Instances:** Overview of instance-related features including virtual computing environments (Instances), configurations (Instance Types), templates (Launch Templates), spot requests, savings plans, reserved instances, dedicated hosts, and capacity reservations.
 - **Images:** AMIs (Amazon Machine Images) and the AMI Catalog for pre-configured templates and operating systems.
 - **Elastic Block Store (EBS):** Covers persistent storage volumes (Volumes), incremental backups (Snapshots), and lifecycle management.
 - **Network & Security:** Includes security groups, elastic IPs, placement groups, key pairs, and network interfaces for comprehensive security and network management.
 - **Load Balancing:** Describes the functionality of load balancers, target groups, and trust stores for application traffic distribution.
 - **Auto Scaling:** Explains auto scaling groups for adjusting the number of EC2 instances automatically to meet application demand.
5. Under the following link one can find a comprehensive overview over the the options for managing instances as endpoint description. [API-EC2-Doc](#).

Exercise 1.3 Create an AMI (5 points)

1. Create an EC2 Instance as described in Question 1 and login to your instance.
2. Install the Apache Web Server on your EC2 Instance via a package management system (apt):

Listing 2: Install Apache

```
1 sudo apt install apache2
```

```
ubuntu@ip-172-31-31-121:~$ ls
ubuntu@ip-172-31-31-121:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support
  ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 2139 kB of archives.
After this operation, 8518 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

3. Create a custom index.html file and deploy it on the Web Server in order to check that the installation was successful.

```
see https://ubuntu.com/esm or run: sudo apt status
Last login: Sun Mar 17 16:10:10 2024 from 131.152.54.27
ubuntu@ip-172-31-31-121:~$ sudo vim /var/www/html/index.html
```

```
ubuntu@ip-172-31-31-121: ~ x + v
!DOCTYPE html
<html>
<head>
<style>
body {
  margin-left: 200px;
  background: #5d9ab2 url("img_tree.png") no-repeat top left;
}

.center_div {
  border: 1px solid gray;
  margin-left: auto;
  margin-right: auto;
  width: 90%;
  background-color: #d0f0f6;
  text-align: left;
  padding: 8px;
}
</style>
</head>
<body>

<div class="center_div">
  <h1>Hello World!</h1>
  <p>This example contains some advanced CSS methods you may not have learned yet.
.</p>
:wa
```

Listing 3: Start Apache

```
1 sudo systemctl start apache2
```

```
ubuntu@ip-172-31-31-121:~$ sudo vim /var/www/index.html
ubuntu@ip-172-31-31-121:~$ sudo systemctl start apache2
ubuntu@ip-172-31-31-121:~$
```

```
ubuntu@ip-172-31-31-121:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-03-17 12:55:58 UTC; 9min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1847 (apache2)
    Tasks: 55 (limit: 1121)
   Memory: 5.1M
      CPU: 62ms
   CGroup: /system.slice/apache2.service
           └─1847 /usr/sbin/apache2 -k start
           └─1849 /usr/sbin/apache2 -k start
           └─1850 /usr/sbin/apache2 -k start

Mar 17 12:55:57 ip-172-31-31-121 systemd[1]: Starting The Apache HTTP Server...
Mar 17 12:55:58 ip-172-31-31-121 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-31-121:~$
```

4. Make sure that your website is reachable from the Internet.

Create security group info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name info
web
Name cannot be edited after creation.

Description info
Allow connection to webserver

VPC info
vpc-002d06f02c4860f9e

Inbound rules info

Type <small>info</small>	Protocol <small>info</small>	Port range <small>info</small>	Source <small>info</small>	Description - optional <small>info</small>	
HTTP	TCP	80	Anywh...	0.0.0.0/0	Delete
HTTP	TCP	80	Anywh...	:::0	Delete
HTTPS	TCP	443	Anywh...	0.0.0.0/0	Delete
HTTPS	TCP	443	Anywh...	:::0	Delete

Add rule

Change security groups [Info](#)

Amazon EC2 evaluates all the rules of the selected security groups to control inbound and outbound traffic to and from your instance. You can use this window to add and remove security groups.

Instance details

Instance ID i-0248fe3c47fa1c212 (aws-ec2--eusnv-u2204-t2m-rsa)	Network interface ID eni-0da52d74e64d55
---	--

Associated security groups

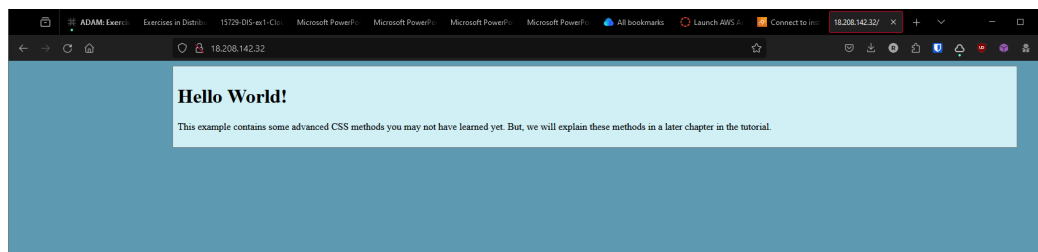
Add one or more security groups to the network interface. You can also remove security groups.

✕ Add security group

Security groups associated with the network interface (eni-0da52d74e64d55)

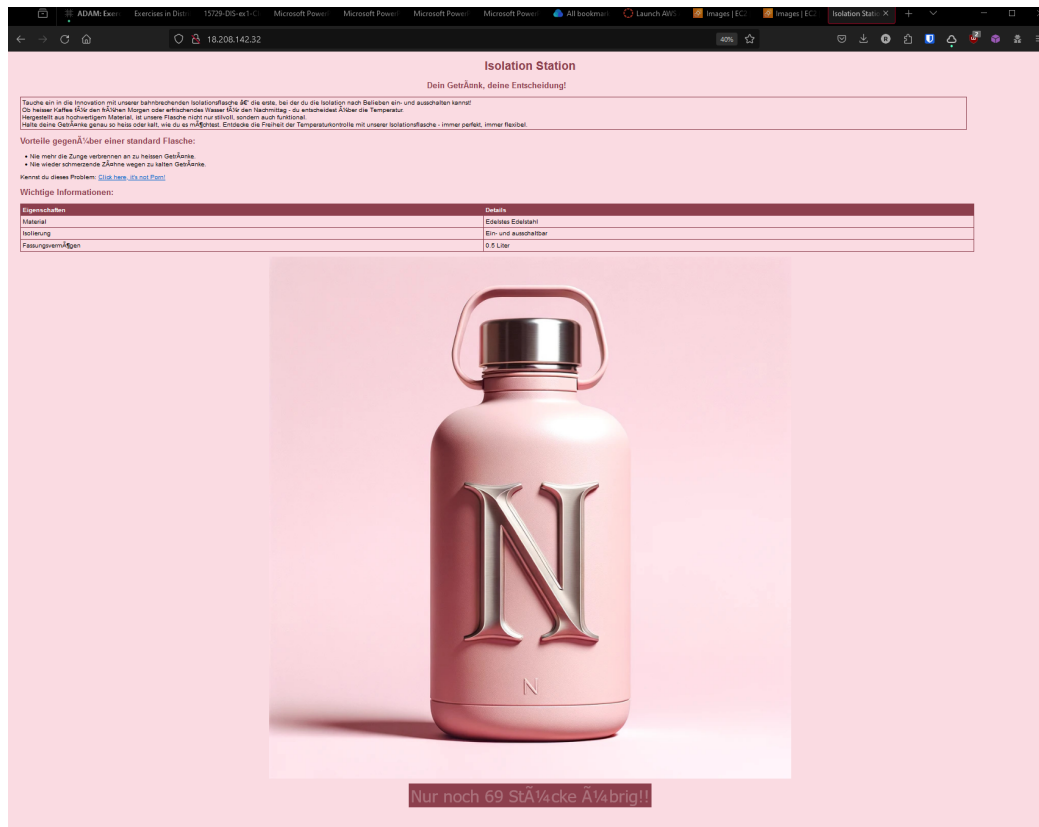
Security group name	Security group ID	
launch-wizard-2	sg-00bea0b5f1c7706a5	Remove
web	sg-0aede324e7fd7d26c	Remove

Cancel Save

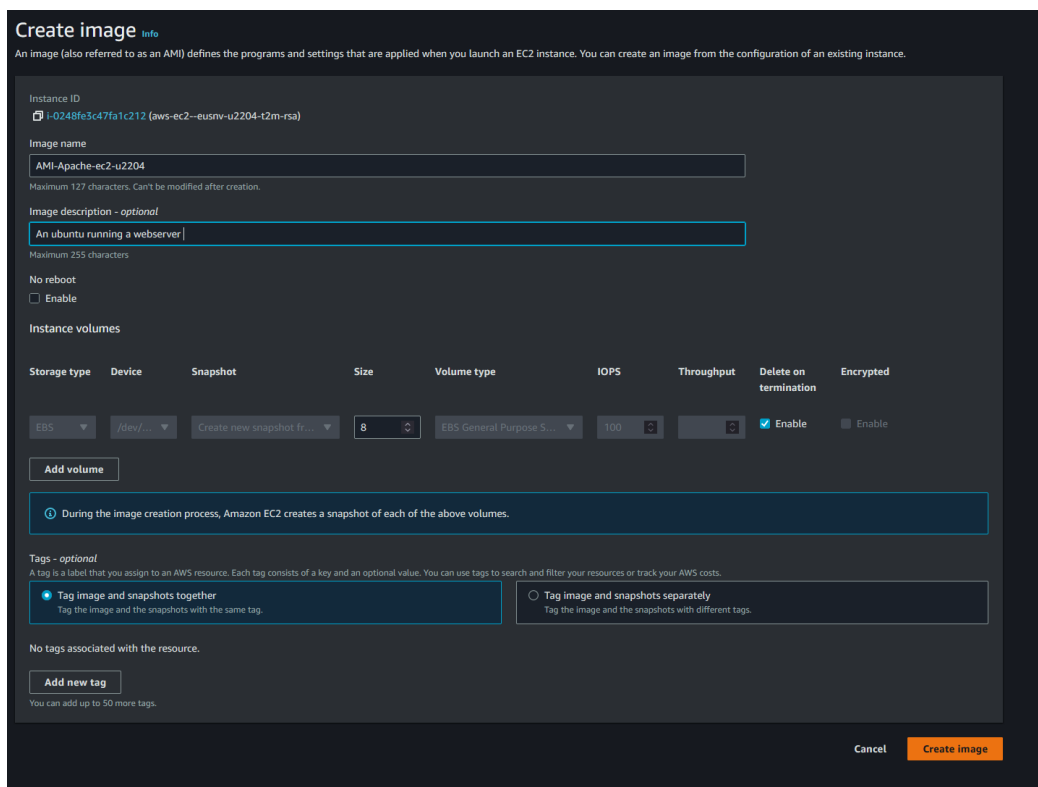
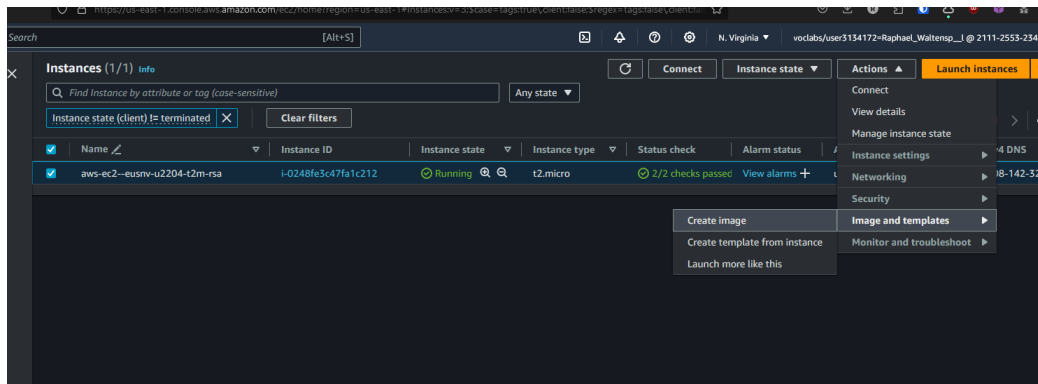


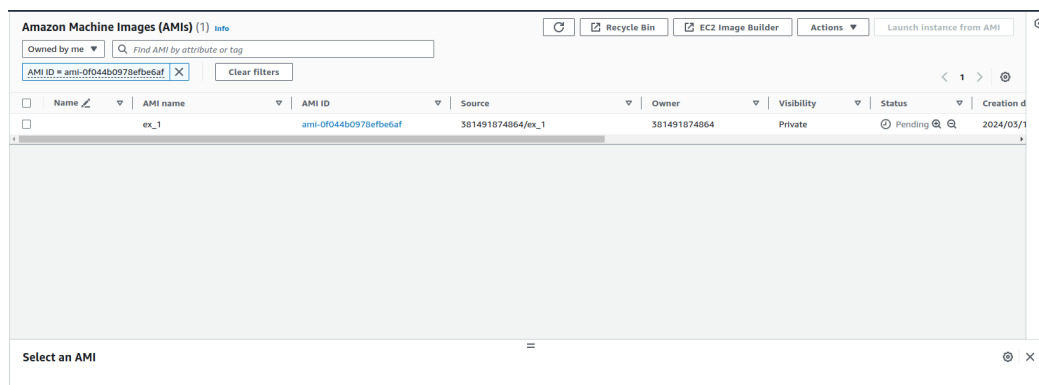
Adding a more complex website from *Informatiklabor* thanks to *Natascha Schmidt*.

```
(base) PS C:\Users\walten0000> scp -r C:\Users\walten0000\Documents\myDoc\Class\WSC_ComputerScience\15729-01 - Distributed Information Systems\Website_Natascha\Website_Natascha
s" ubuntu@ec2-18-208-142-32.compute-1.amazonaws.com:/home/ubuntu
accept.txt 100% 8 0.0KB/s 00:00
index.html 100% 2145 23.0KB/s 00:00
Produkt Foto.jpg 100% 113KB 126.0KB/s 00:00
style.css 100% 1570 16.7KB/s 00:00
(base) PS C:\Users\walten0000>
```



5. Create your own Amazon Machine Image (AMI).





6. Start an EC2 instance using your own AMI. The Apache Web Server should be contained in your EC2 Instance as well as your index.html.

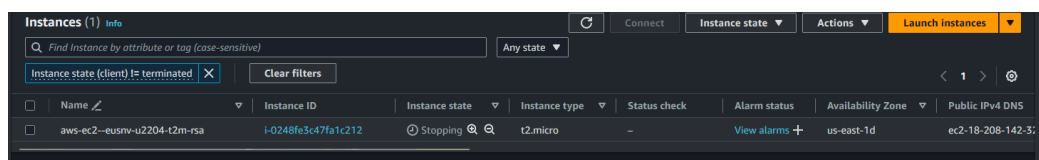


Figure 1: Stop instance

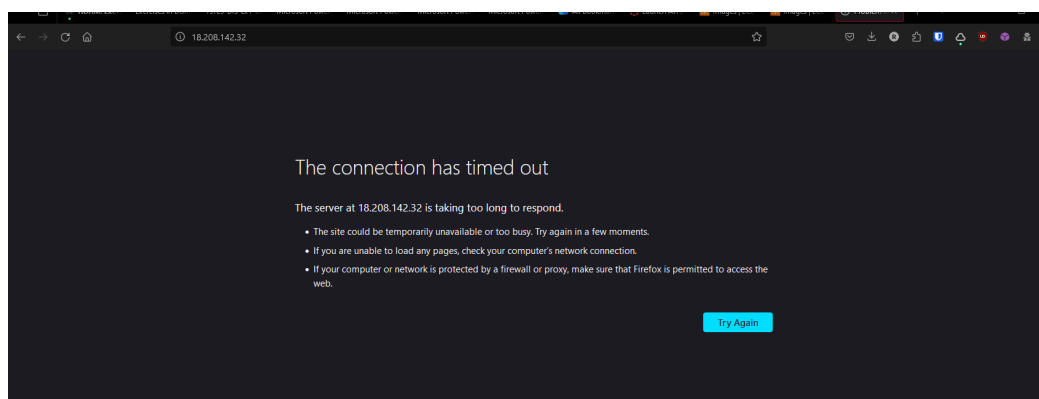


Figure 2: Check connection is down

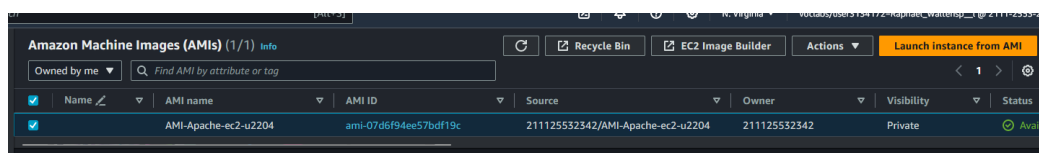


Figure 3: Create instance from ami

[EC2](#) > [Instances](#) > [Launch an instance](#)

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below


[AMI from catalog](#)

[My AMIs](#)

[Quick Start](#)

☒ Owned by me

☐ Shared with me

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

AMI-Apache-ec2-u2204

ami-07d6f94ee57bdf19c

2024-03-17T16:19:19.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Description

An ubuntu running a webserver

Architecture	AMI ID
x86_64	ami-07d6f94ee57bdf19c

Figure 4: Instantiate from ami

The screenshot displays the AWS Management Console configuration page for a new EC2 instance. It is divided into three main sections: Instance type, Key pair (login), and Network settings.

- Instance type:** The selected instance type is `t2.micro`. It is noted as "Free tier eligible". Details include Family: t2, 1 vCPU, 1 GiB Memory, and Current generation: true. Pricing information for On-Demand Windows, SUSE, RHEL, and Linux is provided. A toggle for "All generations" is present, along with a link to "Compare instance types".
- Key pair (login):** A message states that a key pair is required for secure connection. The "Key pair name" dropdown is set to `dmi-walten000`. A "Create new key pair" button is available.
- Network settings:** An "Edit" button is in the top right. The "Network" dropdown is set to `vpc-002d06f02c4860f9e`. The "Subnet" dropdown is set to "No preference (Default subnet in any availability zone)". The "Auto-assign public IP" toggle is set to "Enable". Under "Firewall (security groups)", the "Select existing security group" option is chosen. A list of common security groups is shown, including `launch-wizard-1`, `launch-wizard-2`, and `web`, each with its VPC ID and a close button. A "Compare security group rules" button is at the bottom right.

Figure 5: Instanciate from ami settings

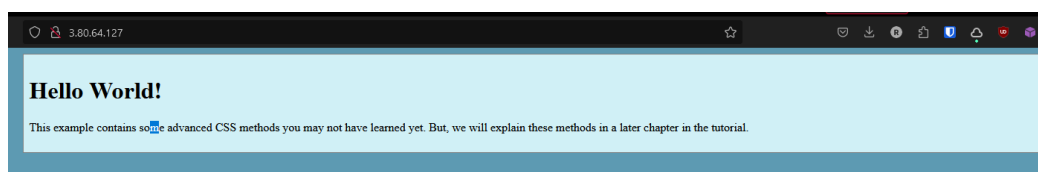


Figure 6: Back online

Exercise 1.4 Cost Model Comparison (6 points)

1. Calculate the cost of the given scenarios.

On-Premise Solution:

Upfront Investment:

- Server cost (2 servers): \$12,000
- SSDs for 2 servers (4 TB): \$800
- On-premise network equipment: \$5,000

Energy Costs:

- Energy consumption for 2 servers per hour: 1500 Watts
- Monthly energy cost: \$432 ($1500W * 24h * 30 \text{ days} * \$0.0004/\text{Watt}$)

Recurring Costs:

- Hardware maintenance cost: \$160 (15% of \$12,800 / 12 months)
- Internet connection: \$120/month

Per month \$712 + \$25000 for the beginning

Cloud Solution:

Compute Instances (AWS: c5.4xlarge):

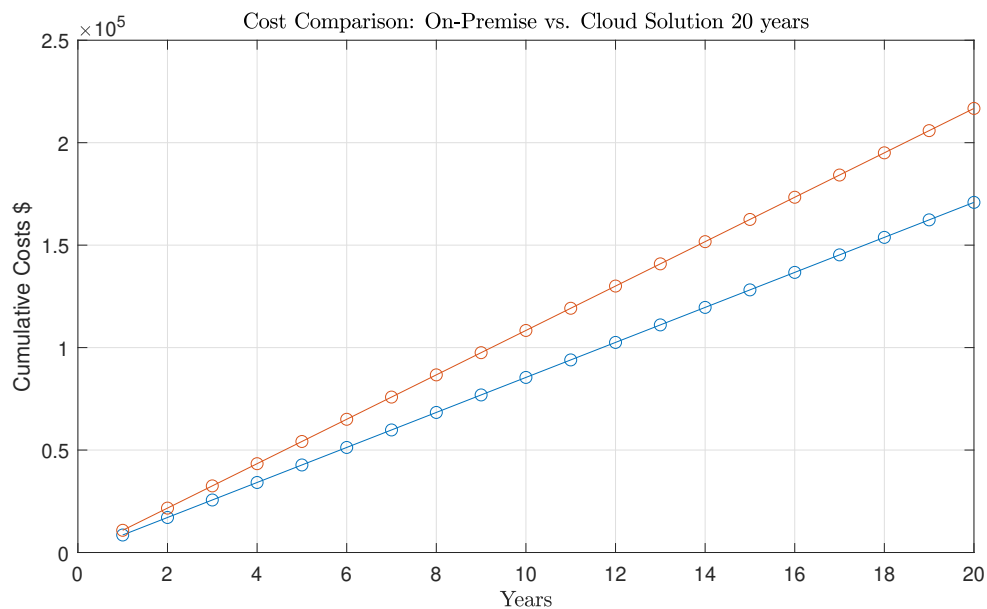
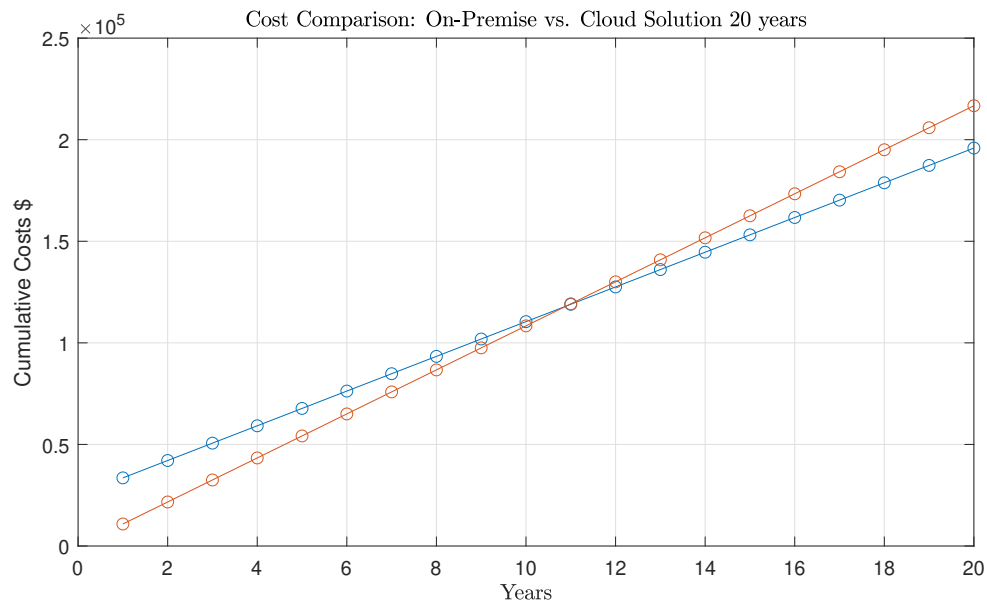
- 2 instances: \$887 ($\$0.616/\text{hour}/\text{instance} * 2 * 24 \text{ hours} * 30 \text{ days}$) - with reservation for 1 year

Storage Costs (AWS: S3):

- Storage: \$11.5 ($500GB * \$0.023/\text{GB}/\text{month}$)
- Data transfer costs: \$4.5 ($100GB \text{ (free)} + 50 * \$0.09/\text{GB} \text{ for data read, } 20GB * \$0.00/\text{GB} \text{ for data write}$)

Per month \$903

2. Compare them
3. Provide a graphic showing the yearly invested sum for both solutions (Is there a breakeven where one solution becomes cheaper than the other? If so, when?)



After 11 years, we would be break even.

4. Which solution do you prefer and why?

We would go with Cloud solution, because market can be unpredictable, and it is rarely worth to buy and SSDs that costs so much for an online shop servers. It will be harder afterwards to resell them, then just stop your Amazon services. Moreover, it is cheaper.

5. List at least three advantages and three disadvantages of the on-premise solution (in comparison to the Cloud solution)

Advantages of On-Premise Solution:

- a) No reliance on external service providers.
- b) Cheaper per month, especially if we would take into account server load (according to some papers average server load is 50%, so we could multiply by this energy consumption)
- c) Greater flexibility in software customization and integration with existing systems

Disadvantages of On-Premise Solution:

- a) Upfront Investment: High initial costs for hardware and network equipment.
- b) Scalability: Scaling up can be slower and more expensive compared to Cloud solutions.
- c) Management: Requires ongoing maintenance and management.

Exercise 1.5 Services (3 points)

Assign the following Amazon Web Services to XaaS categories:

1. EC2 (Elastic Compute Cloud):

- **XaaS Category:** Infrastructure as a Service (IaaS)
- **Explanation:** EC2 provides virtual servers in the cloud, allowing users to run applications. It offers scalable compute capacity, giving users control over the underlying infrastructure (virtual machines).

2. S3 (Simple Storage Service):

- **XaaS Category:** Storage as a Service (STaaS)
- **Explanation:** S3 is a scalable object storage service that allows users to store and retrieve data. It provides storage on demand and is commonly used for backup, archiving, and as a content delivery network.

3. Elastic Beanstalk:

- **XaaS Category:** Platform as a Service (PaaS)
- **Explanation:** Elastic Beanstalk abstracts the underlying infrastructure, providing a platform for deploying and managing applications. It simplifies the process of deploying and scaling web applications without users needing to manage the underlying infrastructure.

4. SES (Simple Email Service):

- **XaaS Category:** Communication as a Service (CaaS)
- **Explanation:** SES is a cloud-based email sending service. It allows users to send emails without managing the underlying email infrastructure. SES is a communication service that fits within the broader category of Communication as a Service (CaaS).

5. RDS (Relational Database Service):

- **XaaS Category:** Database as a Service (DBaaS)
- **Explanation:** RDS simplifies database management tasks by providing a managed relational database service. Users can deploy, scale, and operate relational databases without dealing with the administrative overhead, falling under the category of DBaaS.

6. CloudWatch:

- **XaaS Category:** Monitoring as a Service (MaaS)
- **Explanation:** CloudWatch is a monitoring and management service that provides real-time monitoring for AWS resources. It falls under the category of MaaS as it offers monitoring, logging, and alerting capabilities for various AWS resources.