SMART POUR
DESIGN MANUAL

THIRD YEAR UNIFIED PROJECT

TEAM

E/17/122: SHAZNA ISTHIKAR
E/17/153: ODASARA KARUNACHANDRA
E/17/294: MISHEL ROSSMAREE

Deaprtment of Computer Engineering
Unviersity of Peradeniya
2022

Contents

1. Introduction

Smart Pour is a fully automated coffee machine designed for solo or group use. Carefully designed to overcome the downsides of conventional coffee making, this system helps clients improve efficiency and profitability with minimal involvement of labor.

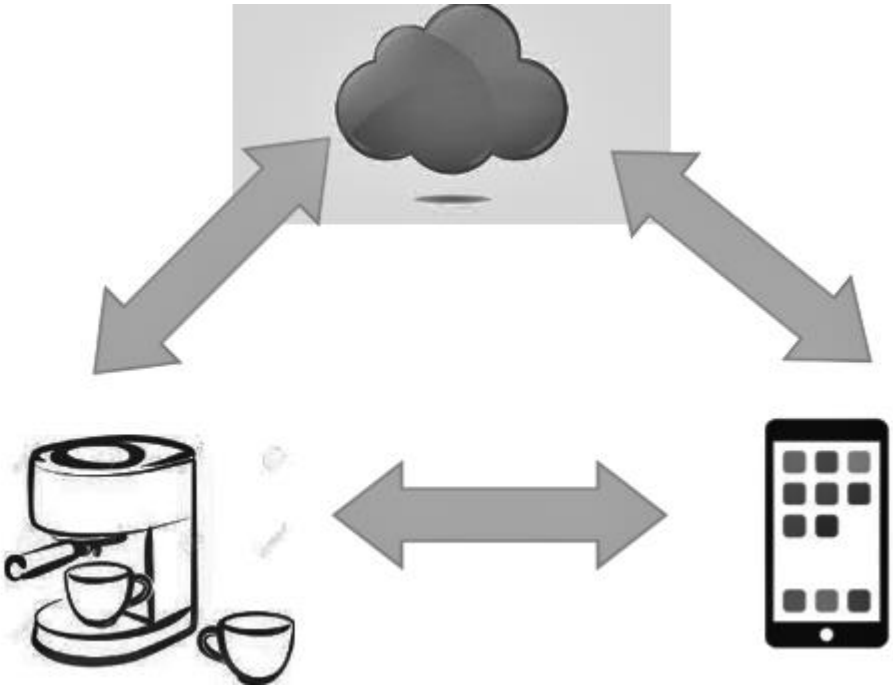
The system consists of two main components; a mobile application to place orders online and track ingredient availability and schedules, favorite recipes, and the Automated Hardware Node (Coffee Machine) .

This Coffee machine is power efficient, easy to use and Fast. The machine can connect to home Wifis and any kind of internet connection. The Coffee machine is powered by NodeMCU which uses a ESP8266 module.

2. Features

- Ability to Schedule coffee making according to user’s need
- Ingredient tracking mechanism
- The coffee making can be customized according to user’s preference
- User friendly controller mobile application to control all operations of the system.
- Emergency manual control switch.
- A management system to keep track of the devices, users, schedules and ingredients

3. System Overview



Mobile Application
An Android mobile application is available for SmartPour coffee machine to allow users to remotely make coffee in a scheduled manner as well. The mobile application sends a notification when the coffee is made and reminders for scheduled coffee.Users can log in to the mobile app anytime to check the availability of the ingredients to make coffee.The app is designed using Flutter.

Web Server

AWS server is used as the web server for the SmartPour System. Signals from sensors are passed to the mobile application as well as control information from mobile applications are passed to the machine and response mechanisms through this server. Database which is used to store favourite recipes, availability of the ingredients and logs of the system is hosted at the server.

SmartPour Machine

The coffee making is done in the machine. The pouring of coffee, ingredient separation, ingredient tracking is identified through sensors. It will consist of wireless sensors such as ultrasonic sensors, reflective optical sensors, valves and servo motors. NodeMCU is used as the microcontroller of the SmartPour and ESP8266 WiFi modules are used for wireless sensors.

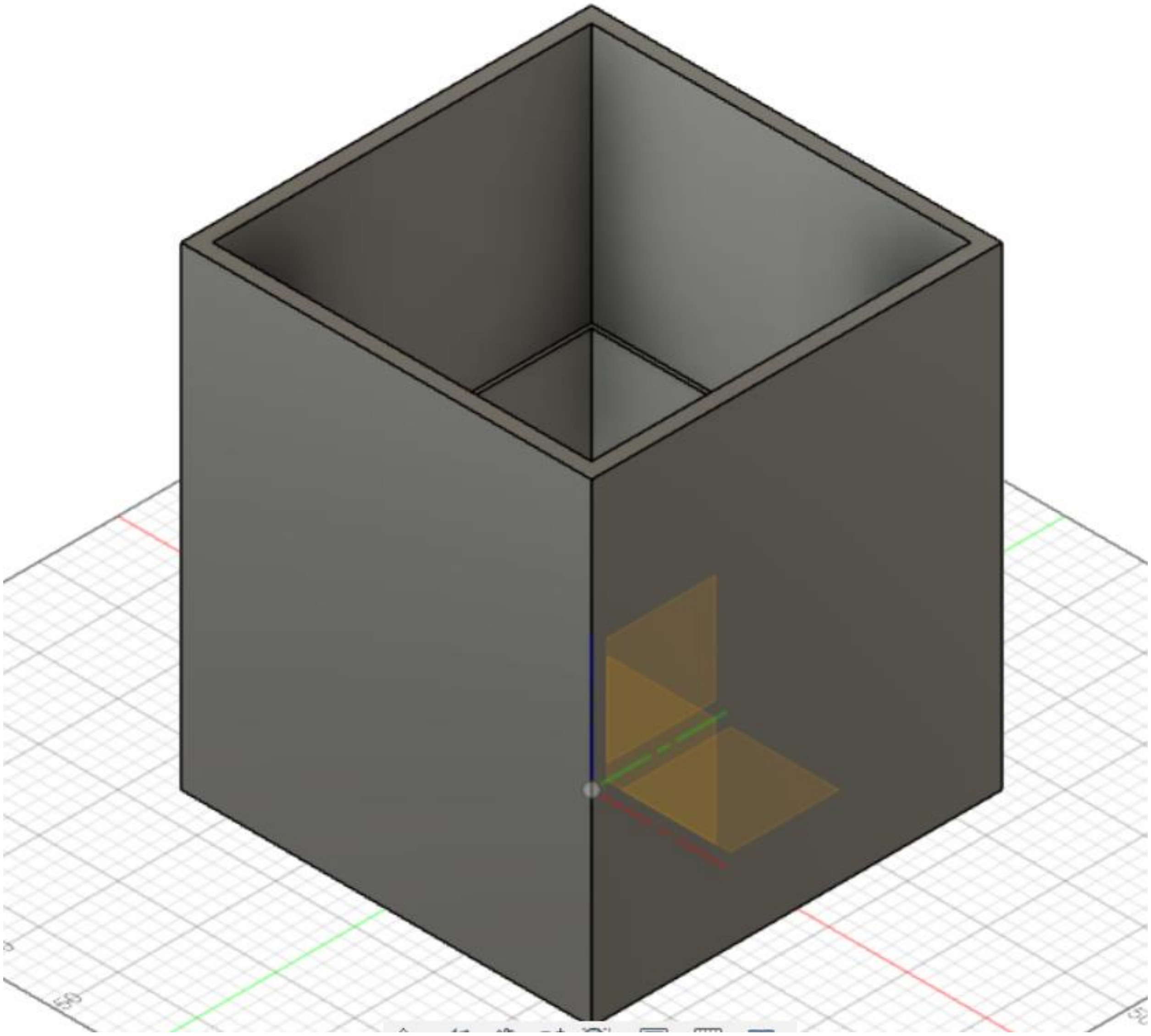
In a nutshell we used,

- **Mobile Application - FlutterDart**
- **RestAPI -SpringBoot**
- **Cloud Server - AWS EC2 Instance with linux Ubuntu OS**
- **Requests- MQTT**

4. Design Overview

The overall Design Looks like the model figure above.The coffee machine is 60 cm in height, 80 cm in width and 80 cm in length. The coffee machine can hold up to 200g, coffee and sugar of each 100g.

4.1. 3D CAD Designs



The 3D cad designs were made to make sure we can have the Laser cuttings according to the dimensions. And to have an idea about the vending machine and how it is going to be there.These 3D designs were made using AutoDesk Fusion .The dimensions were specified as before.

5. Software Deployment

5.1. Web Server

5.1.1. Overview

- Web server is developed using Java Spring Boot with MySQL cloud database deployment.
- MVC architecture is used here, therefore each section of the server is well isolated.
- The server is using http protocol and implemented as a stateless application. It supports POST, GET, PUT, DELETE requests. Hence it is designed to do create, read, update and delete (CRUD) operations.
- The web server is handling user requests with the collaboration of the database. Web server exchanges data between clients and database in JSON format.
- The server employs an internal router to route client requests to required endpoints.
- Apart from this, the server has many Security features, Controllers, Middleware and employs various object model types as well.
- The server uses asynchronous methods to improve efficiency.

5.1.2. Roles in the system

There is only one main roles in our system. Each client with any role must use a unique email address to register on our system.

- User (Customer) – This is our customer. He can register devices to our service. By using our mobile application, he can monitor those devices and receive notifications, warnings, and emails.

5.1.3. Functionalities of the Web Server

1. Login clients

- During a login operation, the client will send email and password in a JSON object such as {email : “user mail”,password: “password”}. The server will cross-reference the data with the database and verify it.
- If the data is wrong, the server will send a response sending a JSON object of errors with a response code of 400.

- If the data matches those of a registered account, then it will create a jwt token with the account id, which is unique to each account, and set a reasonable expiration time. Also, the account name and role are acquired by the database and stored in a JSON object. Then response is sent to the client with that JSON object as body, jwt token as a header, and with 200 response code.

2. Authorization middleware

- Each client request will go through authorization middleware. Since a jwt token created with the unique account id is in request headers, the server verifies and 7 decodes the token to identify the user and role. Then grant the request if the request is within the scope of the client's scope. Otherwise redirected to a warning page.
- Also if the jwt token is expired in the client requests, the client is logged out. Therefore redirected to the login page.
- Each account in the database has a boolean value "isverified". If this value is false, the account is registered but not verified. Therefore the user will be redirected to verification steps.

3. Sending data to clients

- A client sends GET requests to various endpoints in the server. All the responses are sent in JSON.

4. Receiving Data from Devices

- During the registration process, each device will be assigned a device id corresponding to the user's email and device number. Therefore each device will have a unique endpoint. Readings from the devices will come as PUT requests to each endpoint in JSON format bodies with sensor readings. Data will be verified and then stored in the database if it passes the verification. Then 200 response codes will be sent, otherwise, 400 is sent.

5. Security Features

- Password hashing by "bcrypt" library to prevent user information compromises.
- Email verification to filter bot accounts.

- Authorization middleware to limit requests to corresponding role’s scope.
- Setting burst limits using “ddos” library and set “rate limit headers” to prevent DDoS attacks. This blocks the IP addresses which send a burst of requests at the same time and sends 429 responses. Those IP addresses will be blocked for 1 hour.
- Hiding Server information to prevent specially designed attacks to the server.
- Using jwt tokens for client recognition using the “jsonwebtoken” library.

6. Error handling

- In designing the server, user inputs are not to be trusted. They might cause harm to the system if not validated and checked for errors. Also, the data that comes from the devices could be erroneous as well. Therefore the error handling mechanism is integrated into the server.
- When outside data is received, before processing and storing they are validated and checked for errors. In case of no issues, data will be processed and stored accordingly. Otherwise, data will be ignored and the corresponding error message will be sent back.
- When the user inputs are received, a controller method called “handleErrors” will validate the data and generate error messages if there are any.
- When the data from devices are received data size, data type, data integrity, and whether the data value is within the numerical range is checked and stored using a controller method called “readings_put”

7. Requests of each role

Endpoint	**Type**	**Purpose**
/smartpour/user/adduser	Post	To add the new users
/smartpour/user/login	Post	To login the old users with correct username and passwords
/smartpour/machine/addmachine	Post	To add new device
/smartpour/machine/{userID}	Get	To get the machines by ID

/smartpour/recipe/addrecipe	Post	To add a new recipe
/smartpour/recipe/{userId}	Get	To get the recipes by the user
/smartpour/schedule/addschedules	Post	To add the new schedules
/smartpour/schedule/{userId}	Get	To get the recipes by the user

8. Controllers and functionalities

Controller method	**Functionality**
emailvalidator	Validate the input emails
passwordvalidator	Validate the input emails
adduser	Add new users
GetschedulesByUserID	Get the schedules which belongs to the user
_adddvice	Add new devices

9. External Libraries

Library	**Purpose**
http	provides methods to handle typical apps web requests, such as calls to RESTful APIs and image downloads.
JPA	provide a way to map Java objects to database tables.
JWT	self-contained way for securely transmitting information between parties as a JSON object.

5.2. Database Deployment

5.2.1. Overview

- Database deployment is done using MySQL cloud-based database.
- Data validation and error checking methods are included.
- Outside access to the database is limited.
- Data backup is enabled.
- Configure Maven Dependencies

```
<dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>
```

- Configuring MySQL, JPA, and Hibernate- In the resources folder within the src/main folder, open the application.properties file and write the properties below.
    - `spring.datasource.url`: Springboot uses the URL along with the credentials (Here, the MySQL is configured with port 3306) to establish a database connection to our MySQL database.
    - `spring.datasource.username` & `spring.datasource.password` properties are the MySQL database username and password.
    - The default Spring boot port is `8080`.
    - `Springboot.jpa.properties.hibernate.dialect`: The SQL dialect makes Hibernate generate better SQL for the chosen database.

5.2.2. Steps

See 5.5 Cloud Deployment Part

5.2.3. Tables in the Database

Tables are defined as models using Java Spring Boot.

Users :

- userID - Primary Key

- username
- password

Schedules :

- scheduleID - Primary Key
- date
- time
- recipeID
- userID

Machines :

- machineID - Primary Key
- name
- deviceID
- status

Recipe :

- recipeID - Primary Key
- recipeName
- sugar
- coffeepowder

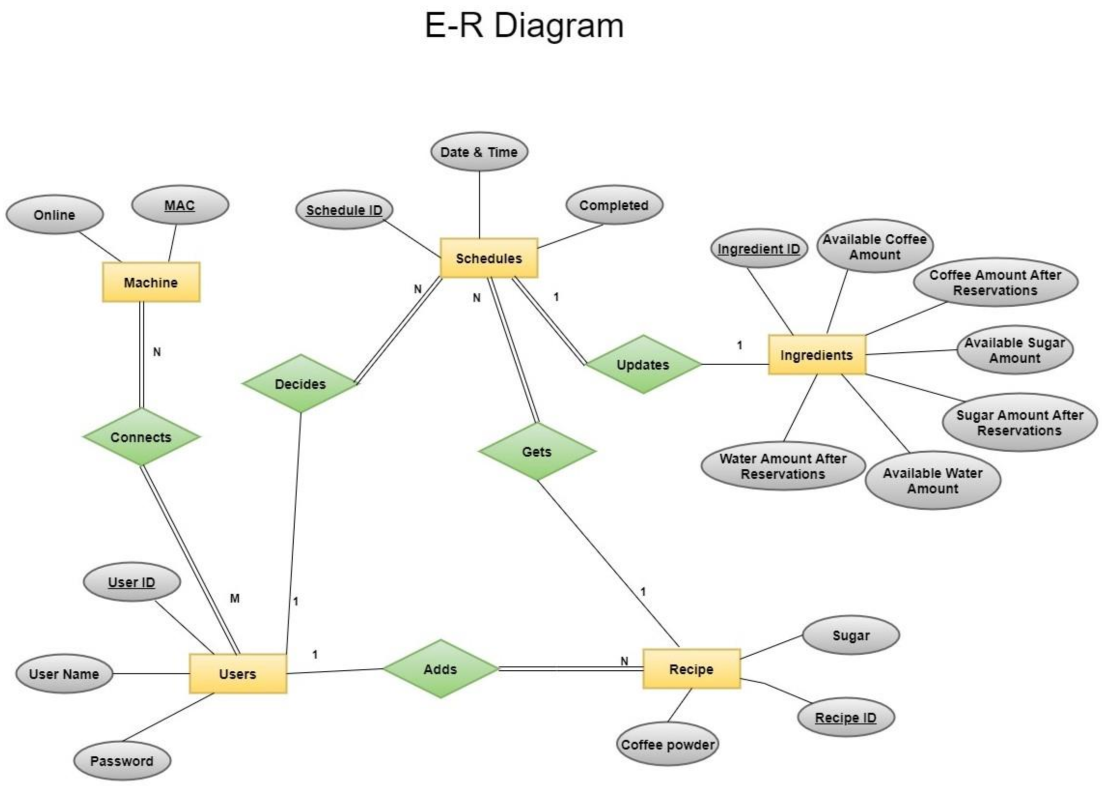
Ingredients :

- ingredientsID - Primary Key
- name

Machine2Ingredients :

- Machine2IngredientsID - Primary Key
- machineID
- ingredientID
- currentAmount
- remainingAmount

5.2.4. ER Diagram



5.3. Mobile Application

5.3.1 Overview

- Mobile application is developed using flutter sdk and dart language.
- Used to access the coffee machine and get the current status of the ingredient amounts
- Notifications and alerts are sent to users when necessary.
- Email and password entered by the user is validated within the app before sending it to the server.

5.3.2. Security of Mobile Application

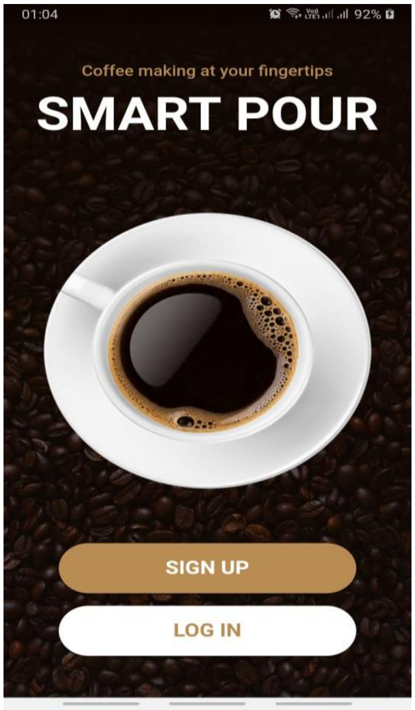
- For the mobile application , there is a login function and after that login function the user will keep the connection with the server through a jwt cookie.
- That cookie will be stored in a shared_preference instance. It is a common library to store non trivial data in mobile applications. Everytime the application makes a

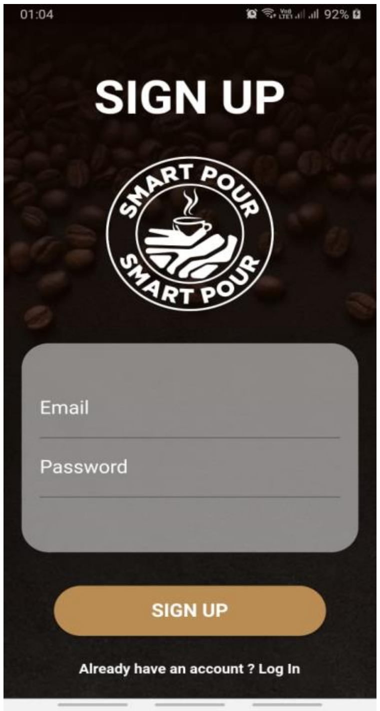
request to the server, the cookie will be retrieved from the instance to make the request and it will be updated after the response.

- Since the mobile app uses the shared preferences, even if we close the app , it will store the cookie until the user logs in again.

5.3.3. Configuration of the Mobile Application

- Go to the play store and download the smartPour application.
- After installing finishes , the user should login by giving the email and password.





5.3.4. Additional Resource Documentation for the Mobile Application .

This contains the documentations of the packages which are used in the mobile application development.

- Flutter documentation https://flutter.dev/docs
- Firebase Documentation https://firebase.google.com/docs/cloud-messaging
- Local_Notifications.dart https://pub.dev/packages/flutter_local_notifications
- http/http.dart https://pub.dev/packages/http
- shared_preferences https://pub.dev/packages/shared_preferences

5.4. Embedded Software

5.4.1. Overview

- One embedded module is used in this product.
- It is used for,
  - To get sensor readings
  - For communication with the internet
- Programming is done using the Arduino language.
- Input validation/comparison done by NodeMCU 8266 chip.
- Device configuration details are stored in ESP 12E chip installed module.
- Communication between the chips is done by serial communication.
- Sensor readings taken by the ESP8266 chip are transmitted to ESP 12E chip.
- ESP 12E sends the data to the web server.

5.4.2. Functionality : ESP 8266 12E Chip(Node MCU)

Libraries Used

Library	**Purpose**
ESP8266HTTPClient	To establish a connection with web server
ESP8266WiFi	Configure WiFi connection
WiFiManager	To store WiFi information

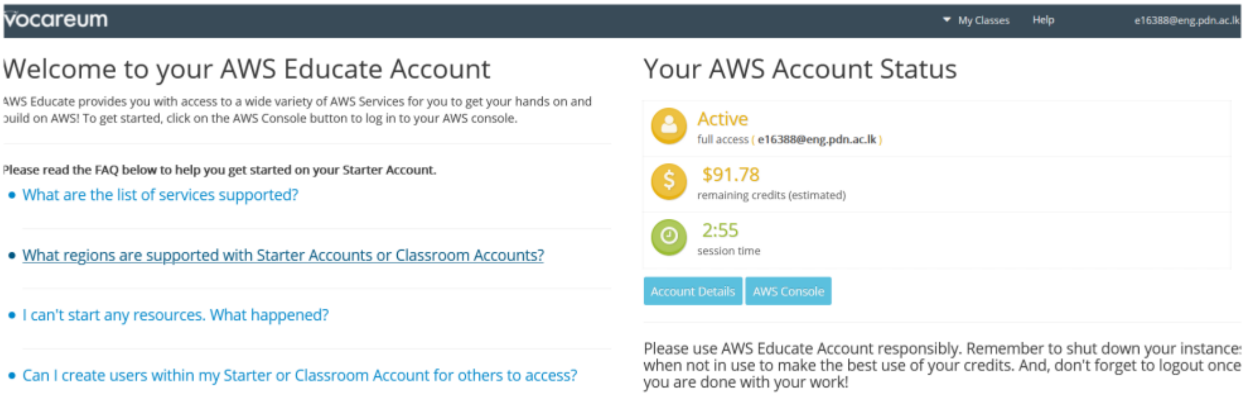
5.5. Amazon Web Services Cloud Deployment

5.5.1. Overview

- Web server is deployed on aws ec2 instance.
- Sever can be accessed via port 80.
- Owners can ssh to the instance via port 22.
- Client requests are handled by the server in ec2 instance.
- Route 53 service is used to add a domain name for our deployment.

5.5.2. Steps to create AWS EC2 Instance

1. Login to your AWS account.



Welcome to your AWS Educate Account

AWS Educate provides you with access to a wide variety of AWS Services for you to get your hands on and build on AWS! To get started, click on the AWS Console button to log in to your AWS console.

Please read the FAQ below to help you get started on your Starter Account.

- What are the list of services supported?
- What regions are supported with Starter Accounts or Classroom Accounts?
- I can't start any resources. What happened?
- Can I create users within my Starter or Classroom Account for others to access?

Your AWS Account Status

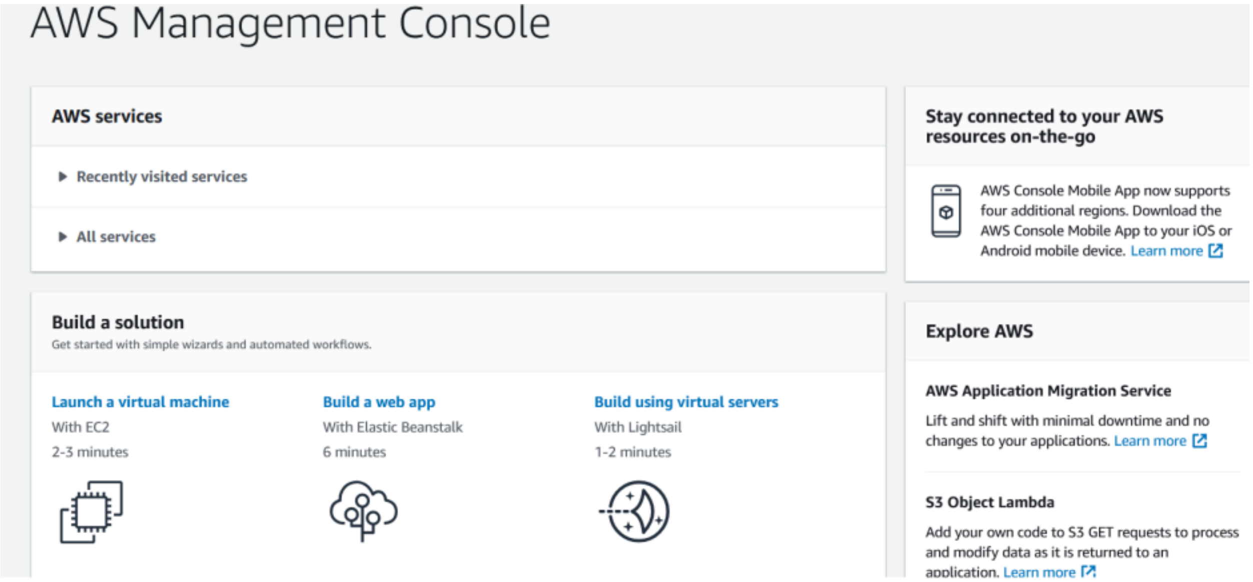
Active
full access (e16388@eng.pdn.ac.lk)

$91.78
remaining credits (estimated)

2:55
session time

Please use AWS Educate Account responsibly. Remember to shut down your instances when not in use to make the best use of your credits. And, don't forget to logout once you are done with your work!

2. Select AWS Console.



AWS Management Console

AWS services

Build a solution

Get started with simple wizards and automated workflows.

Launch a virtual machine
With EC2
2-3 minutes

Build a web app
With Elastic Beanstalk
6 minutes

Build using virtual servers
With Lightsail
1-2 minutes

Stay connected to your AWS resources on-the-go

AWS Console Mobile App now supports four additional regions. Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more

Explore AWS

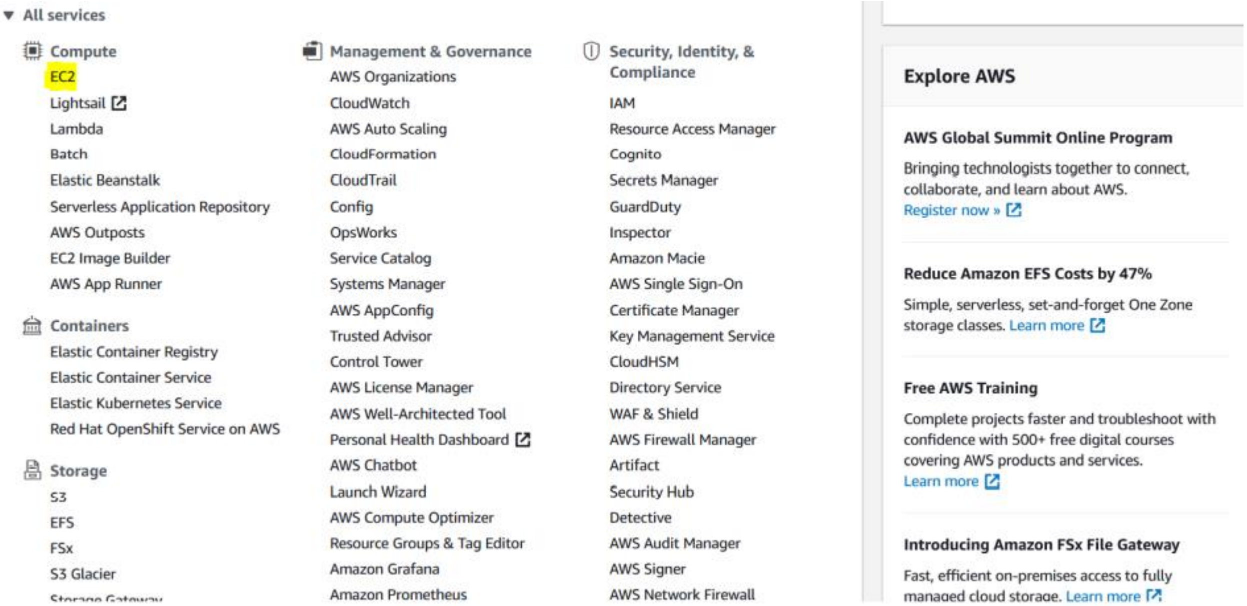
AWS Application Migration Service

Lift and shift with minimal downtime and no changes to your applications. Learn more

S3 Object Lambda

Add your own code to S3 GET requests to process and modify data as it is returned to an application. Learn more

3. Select All services -> EC2.



▼ All services

Compute
EC2
Lightsail
Lambda
Batch
Elastic Beanstalk
Serverless Application Repository
AWS Outposts
EC2 Image Builder
AWS App Runner

Containers
Elastic Container Registry
Elastic Container Service
Elastic Kubernetes Service
Red Hat OpenShift Service on AWS

Storage
S3
EFS
FSx
S3 Glacier
Storage Gateway

Management & Governance
AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
AWS AppConfig
Trusted Advisor
Control Tower
AWS License Manager
AWS Well-Architected Tool
Personal Health Dashboard
AWS Chatbot
Launch Wizard
AWS Compute Optimizer
Resource Groups & Tag Editor
Amazon Grafana
Amazon Prometheus

Security, Identity, & Compliance
IAM
Resource Access Manager
Cognito
Secrets Manager
GuardDuty
Inspector
Amazon Macie
AWS Single Sign-On
Certificate Manager
Key Management Service
CloudHSM
Directory Service
WAF & Shield
AWS Firewall Manager
Artifact
Security Hub
Detective
AWS Audit Manager
AWS Signer
AWS Network Firewall

Explore AWS

AWS Global Summit Online Program

Bringing technologists together to connect, collaborate, and learn about AWS.
Register now »

Reduce Amazon EFS Costs by 47%

Simple, serverless, set-and-forget One Zone storage classes. Learn more

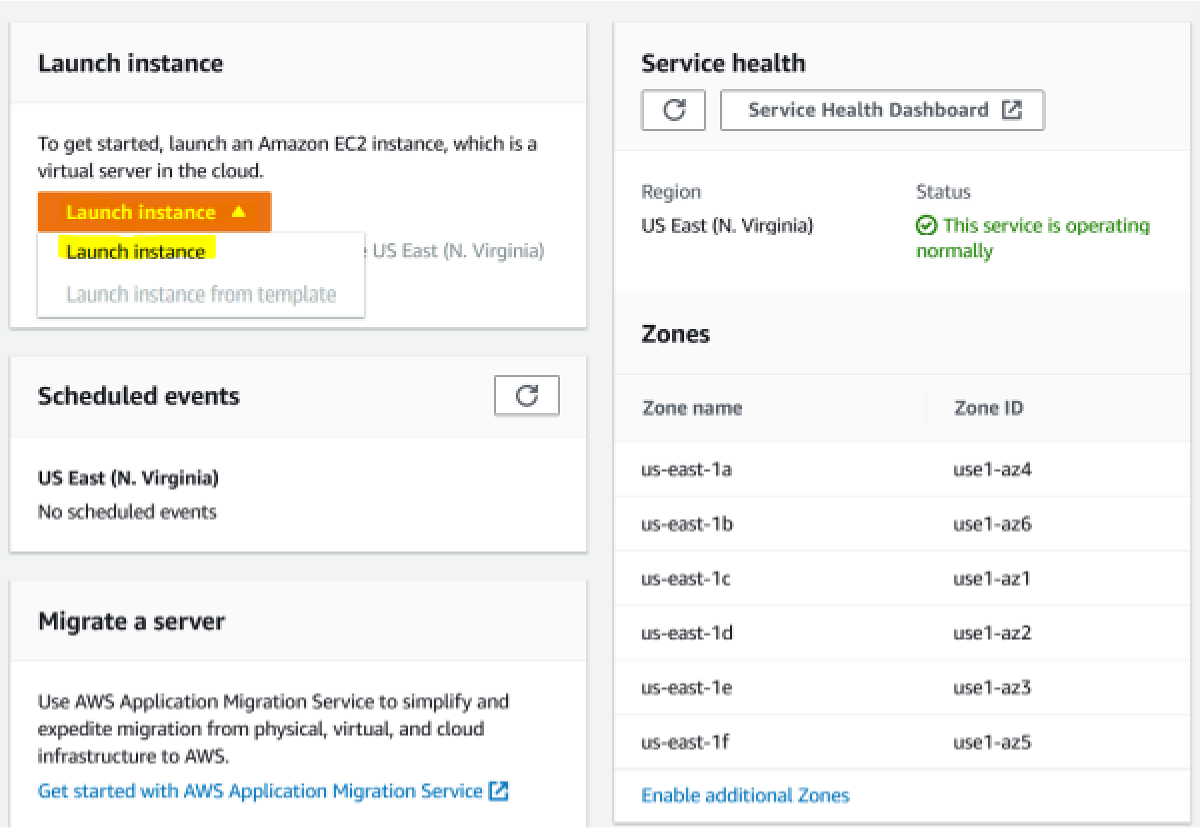
Free AWS Training

Complete projects faster and troubleshoot with confidence with 500+ free digital courses covering AWS products and services.
Learn more

Introducing Amazon FSx File Gateway

Fast, efficient on-premises access to fully managed cloud storage. Learn more

4. Select Launch Instance -> Launch Instance



Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Scheduled events

US East (N. Virginia)
No scheduled events

Migrate a server

Use AWS Application Migration Service to simplify and expedite migration from physical, virtual, and cloud infrastructure to AWS.
Get started with AWS Application Migration Service

Service health

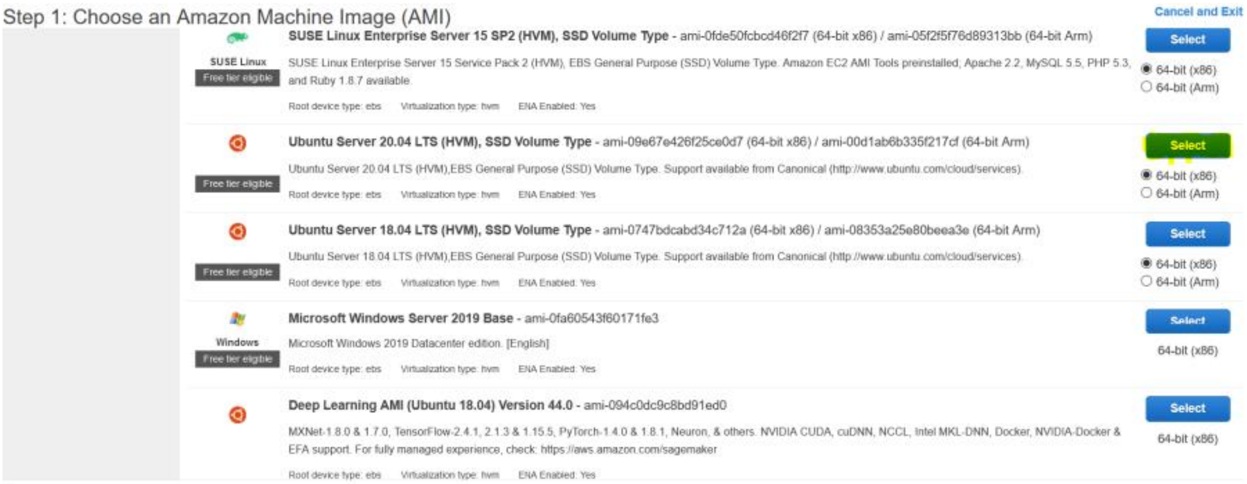
Region: US East (N. Virginia)
Status: This service is operating normally

Zones

Zone name	Zone ID
us-east-1a	use1-az4
us-east-1b	use1-az6
us-east-1c	use1-az1
us-east-1d	use1-az2
us-east-1e	use1-az3
us-east-1f	use1-az5

Enable additional Zones

5. Choose an Amazon Machine Image (AMI). Ideally select a Linux based Machine Image



Step 1: Choose an Amazon Machine Image (AMI)

SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type - ami-0fde50fcbcd46f2f7 (64-bit x86) / ami-05f2f5f76d89313bb (64-bit Arm)

SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-09e67e426f25ce0d7 (64-bit x86) / ami-00d1ab6b335f217cf (64-bit Arm)

Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-0747bdcabd34c712a (64-bit x86) / ami-08353a25e80beea3e (64-bit Arm)

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Microsoft Windows Server 2019 Base - ami-0fa60543f60171fe3

Microsoft Windows 2019 Datacenter edition. [English]

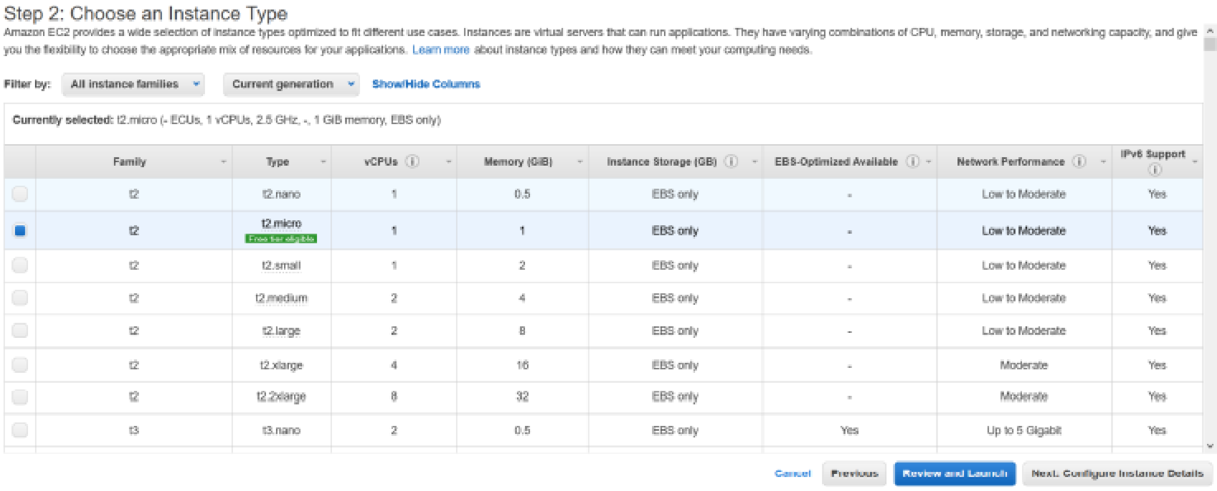
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Deep Learning AMI (Ubuntu 18.04) Version 44.0 - ami-094c0dc9c8bd91ed0

MXNet-1.8.0 & 1.7.0, TensorFlow-2.4.1, 2.1.3 & 1.15.5, PyTorch-1.4.0 & 1.8.1, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA support. For fully managed experience, check: https://aws.amazon.com/sagemaker

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

6. Choose an Instance Type



Step 2: Choose an Instance Type

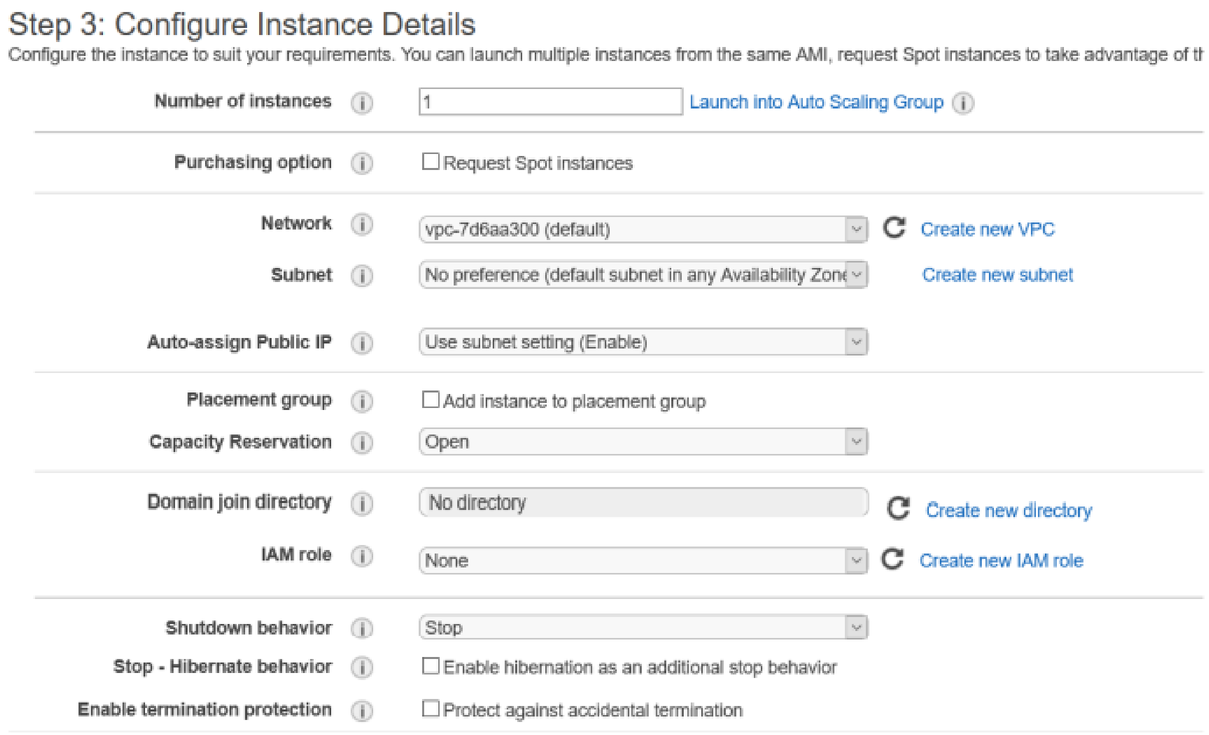
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources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

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

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
□	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
□	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
□	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

7. Configure Instance Details

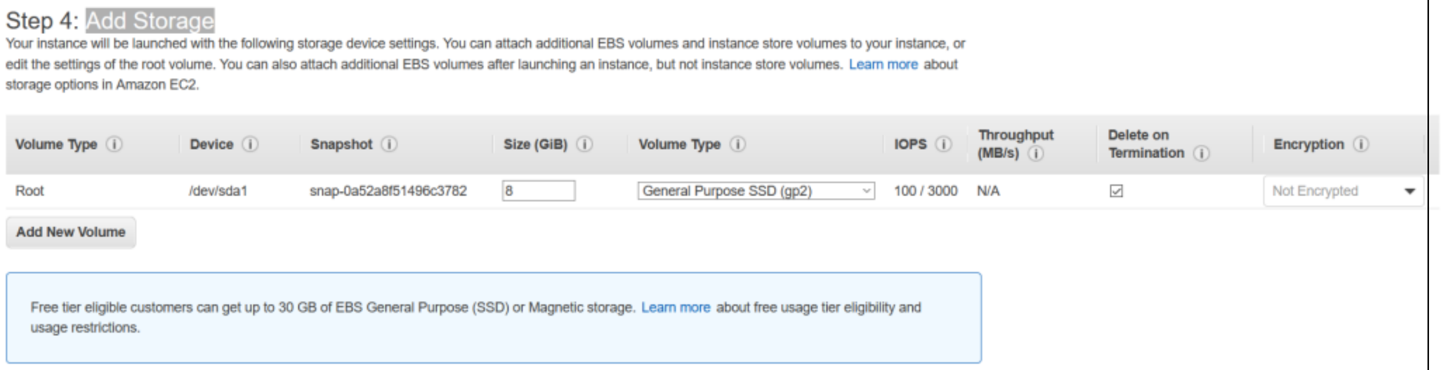


Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of tl

Number of instances	1 — Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances
Network	vpc-7d6aa300 (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
Domain join directory	No directory — Create new directory
IAM role	None — Create new IAM role
Shutdown behavior	Stop
Stop - Hibernate behavior	□ Enable hibernation as an additional stop behavior
Enable termination protection	□ Protect against accidental termination

8. Add Storage as your requirement



Step 4: Add Storage

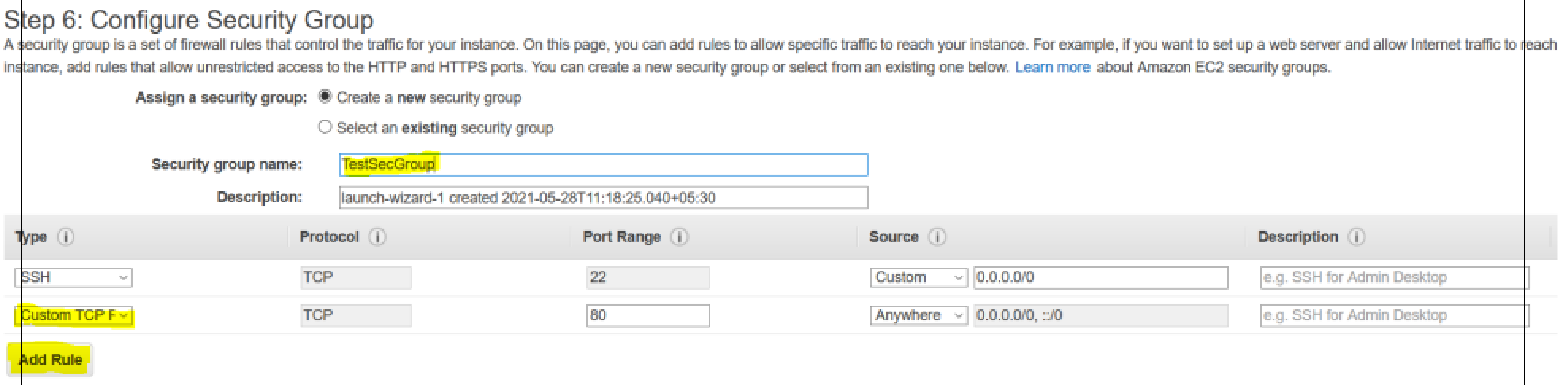
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. 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/sda1	snap-0a52a8f51496c3782	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

Add New Volume

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

9. Configure Security Group.
    a. Create a new Security Group.
    b. Click Add Rule -> add new Custom TCP rule on port 80 and source as given.
    c. Now unauthorized access is restricted according to these rules



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

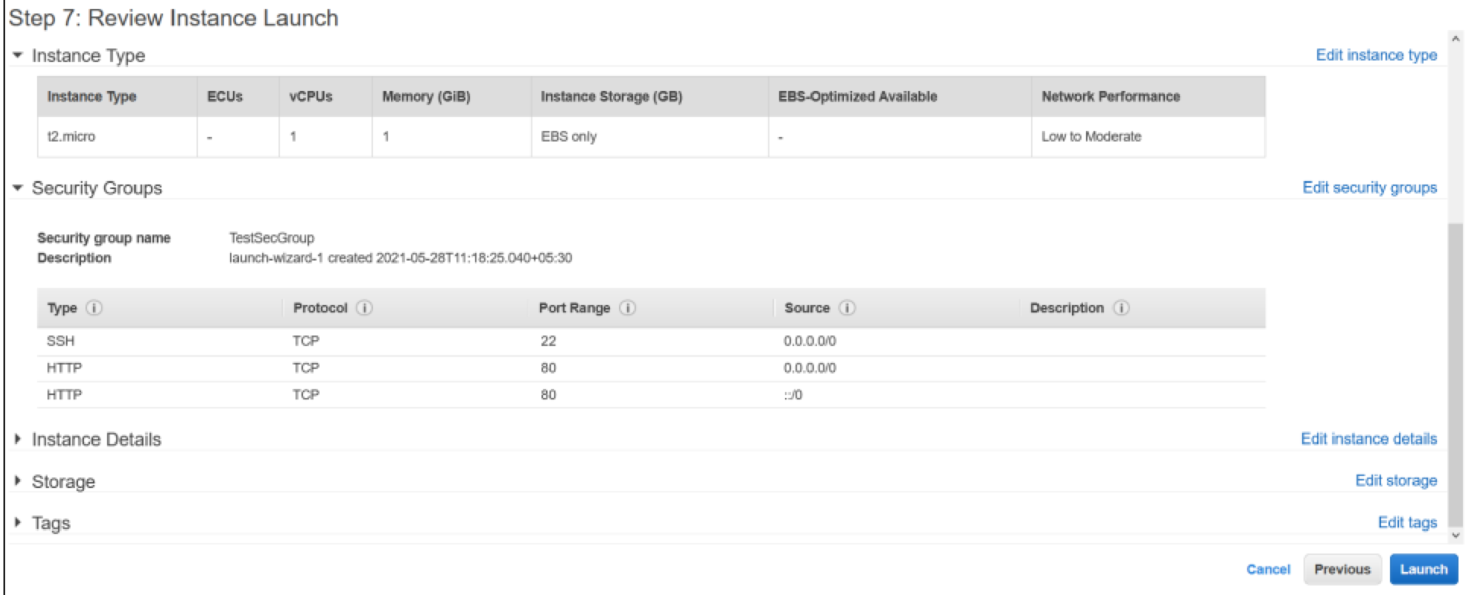
○ Select an **existing** security group

Security group name: TestSecGroup

Description: launch-wizard-1 created 2021-05-28T11:18:25.040+05:30

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	80	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

10. Review Instance Launch. Review your selections and launch the instance.



Step 7: Review Instance Launch

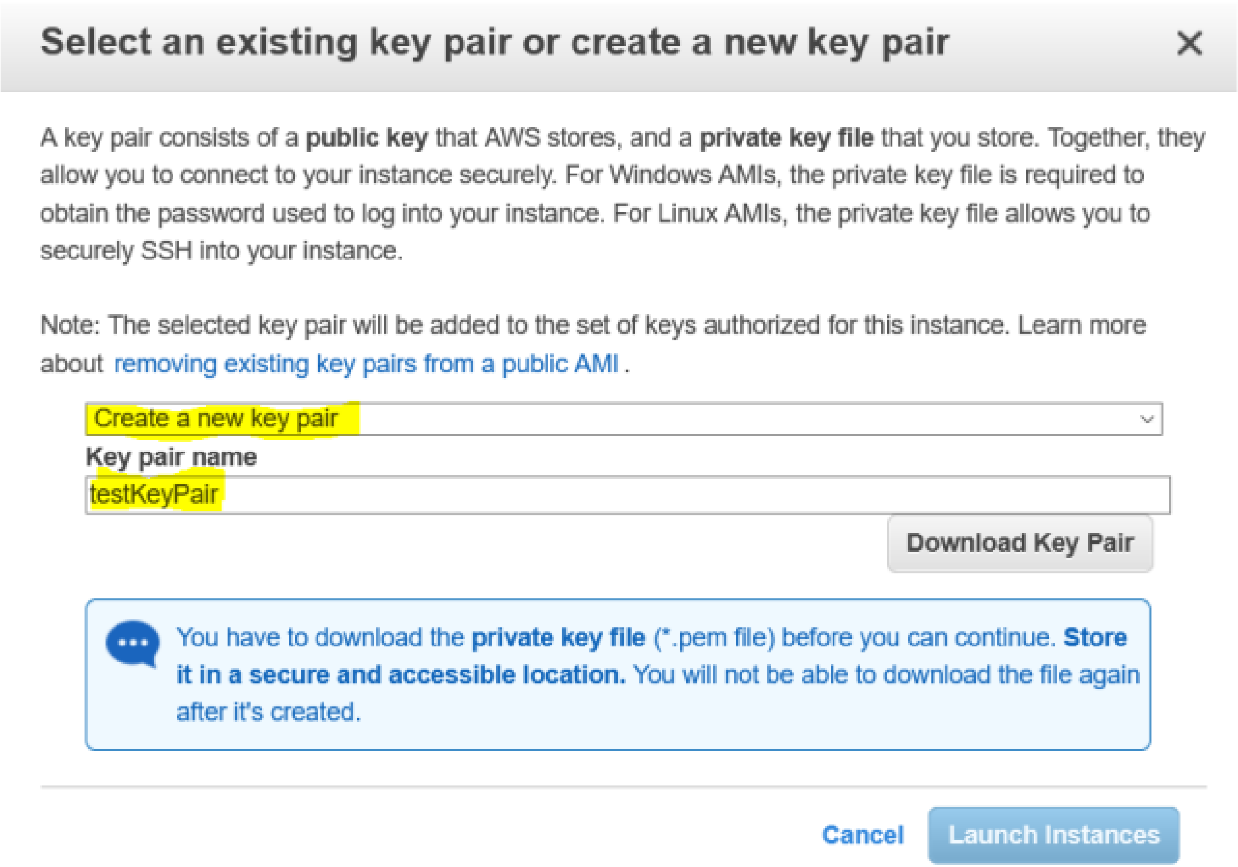
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Security group name TestSecGroup

Description launch-wizard-1 created 2021-05-28T11:18:25.040+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	

11. Create a new Key Pair to ssh into your instance. Then download it. Never lose this file.



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 .

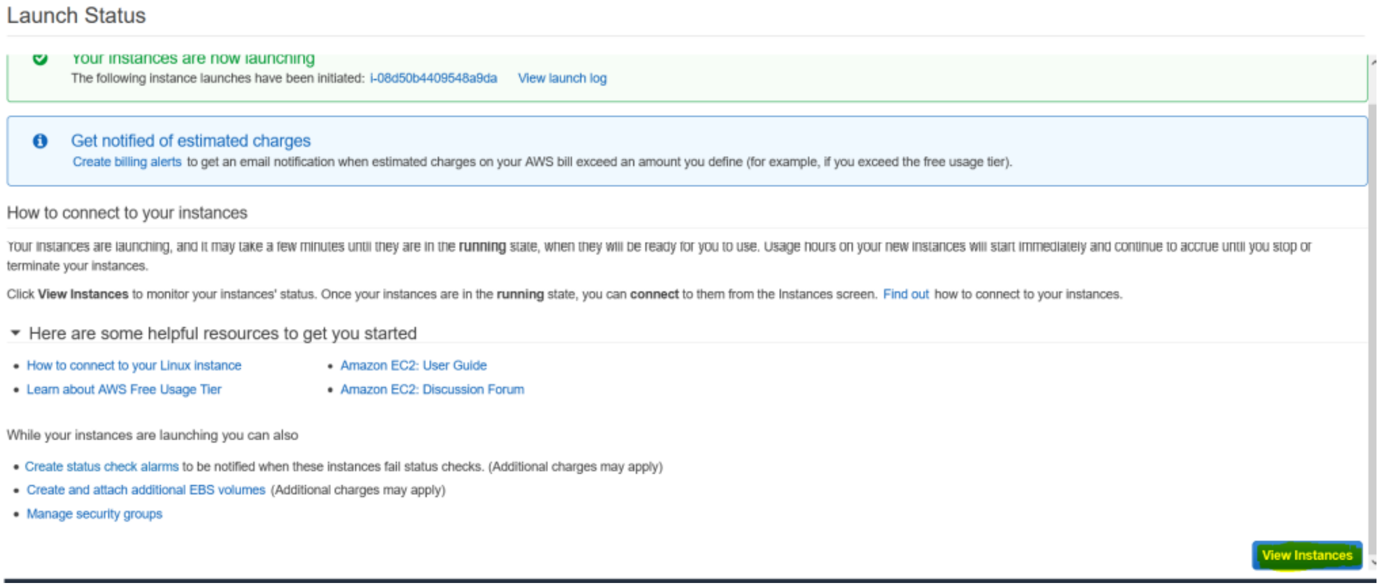
Create a new key pair

Key pair name

testKeyPair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

12. Now your instance is launched. To view your instance. Click on View Instance.



Launch Status

Your instances are now launching

The following instance launches have been initiated: i-08d50b4409548a9da View launch log

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

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 how to connect to your instances.

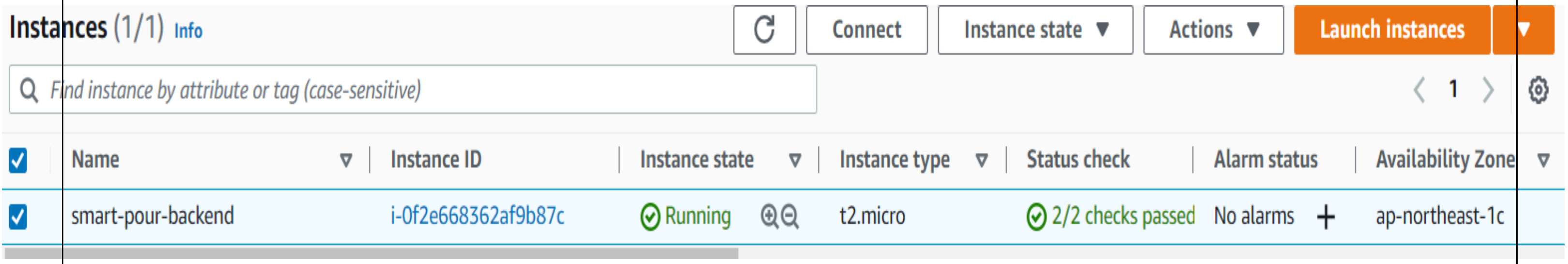
Here are some helpful resources to get you started

- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

While your instances are launching you can also

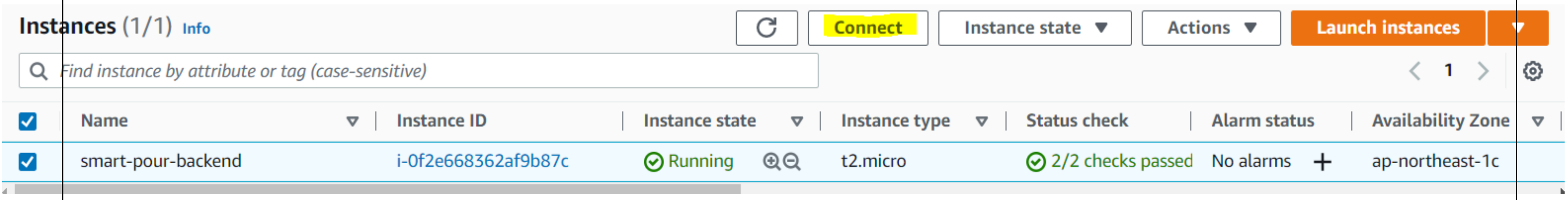
- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes (Additional charges may apply)
- Manage security groups

13. Select your instance



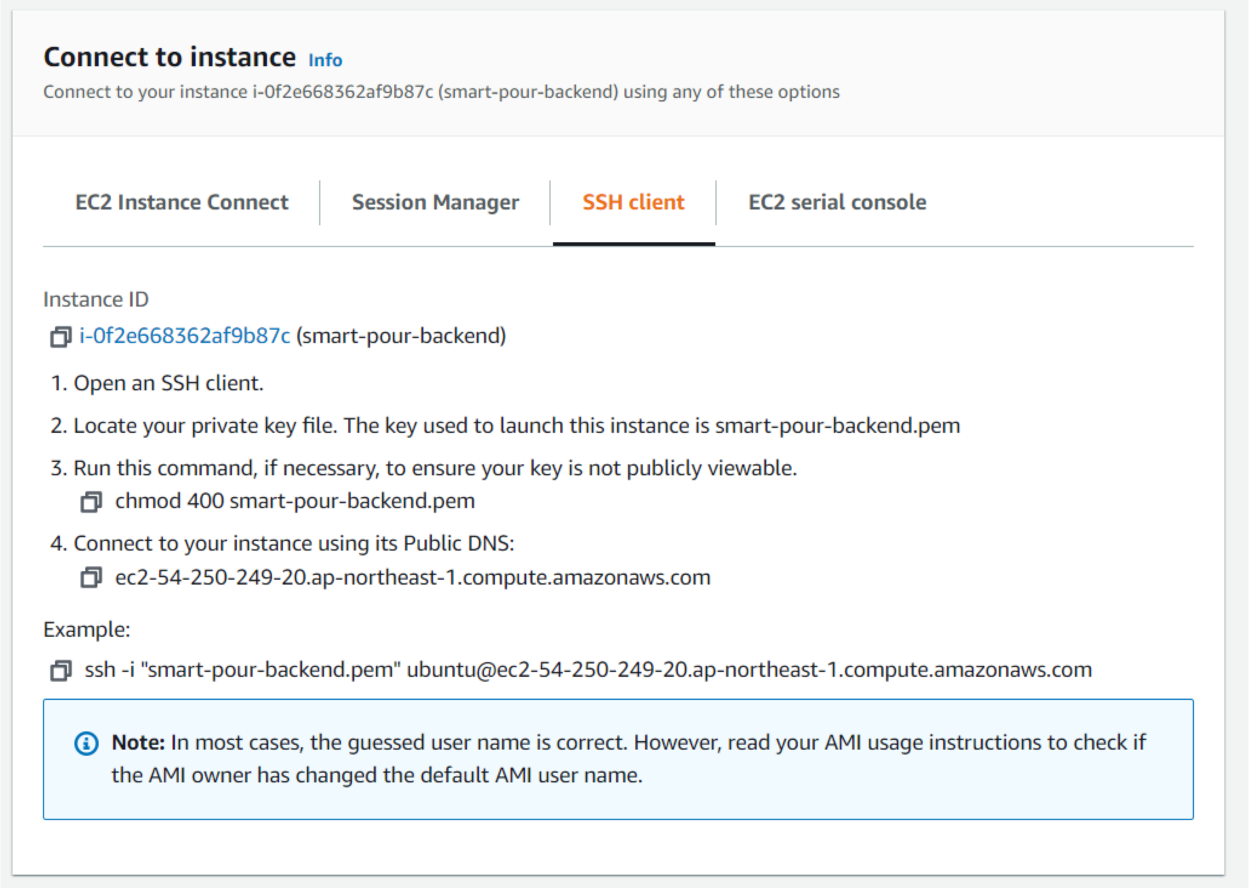
☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	smart-pour-backend	i-0f2e668362af9b87c	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	ap-northeast-1c

14. Select Connect



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	smart-pour-backend	i-0f2e668362af9b87c	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	ap-northeast-1c

15. Copy the highlighted command -> open a terminal in the folder with key pair->run the command->Type "yes".



Connect to instance Info

Connect to your instance i-0f2e668362af9b87c (smart-pour-backend) using any of these options

Instance ID

i-0f2e668362af9b87c (smart-pour-backend)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is smart-pour-backend.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.

```
chmod 400 smart-pour-backend.pem
```

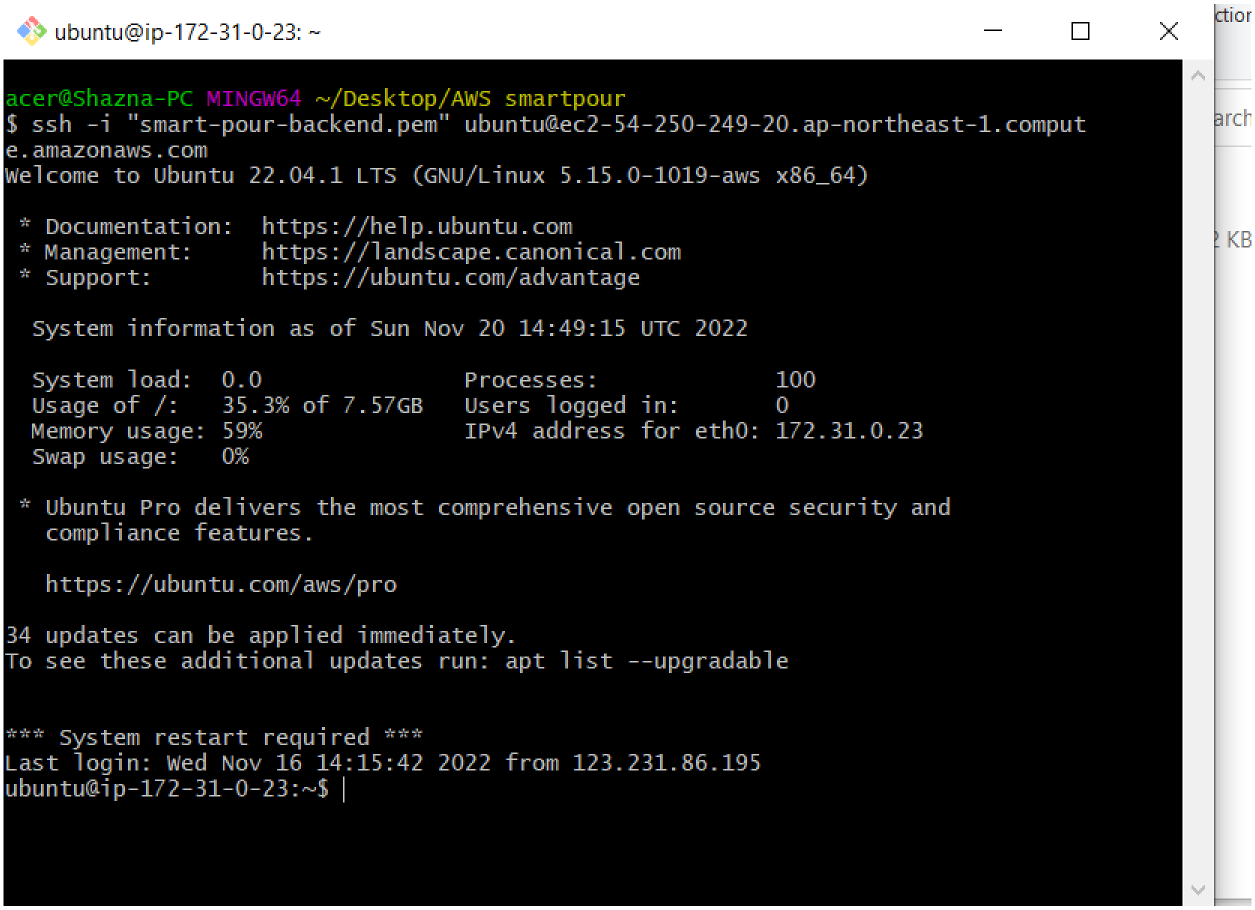
4. Connect to your instance using its Public DNS:

```
ec2-54-250-249-20.ap-northeast-1.compute.amazonaws.com
```

Example:

```
ssh -i "smart-pour-backend.pem" ubuntu@ec2-54-250-249-20.ap-northeast-1.compute.amazonaws.com
```

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.



```
acer@Shazna-PC MINGW64 ~/Desktop/AWS smartpour
$ ssh -i "smart-pour-backend.pem" ubuntu@ec2-54-250-249-20.ap-northeast-1.comput
e.amazonaws.com
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1019-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Nov 20 14:49:15 UTC 2022

  System load:  0.0               Processes:             100
  Usage of /:   35.3% of 7.57GB   Users logged in:       0
  Memory usage: 59%               IPv4 address for eth0: 172.31.0.23
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

34 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


*** System restart required ***
Last login: Wed Nov 16 14:15:42 2022 from 123.231.86.195
ubuntu@ip-172-31-0-23:~$ |
```

16. Run following commands.
    a. sudo apt update
    b. sudo apt install git

5.5.3. Steps to Install MySQL on your AWS EC2 instance.

1. Run following commands.
    a. `sudo apt update`
    b. `sudo apt install mysql-server`
    c. `sudo systemctl status mysql` – to check whether the server is running or not
    d. `sudo mysql` – log in as root.
    e. `mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'your_password_here';`

`mysql> FLUSH PRIVILEGES; -` set a password for your root now

exit and log in with the root credentials.

f. mysql> exit

g. `$ sudo mysql -u root -p`

5.5.4. Steps to deploy spring boot application in AWS EC2 Instance

1. Prepare the final jar file

Run maven build to package the application as a fat jar.

```
mvn clean install
```

Once your build is successful, you will find the jar file in the target folder. Copy the jar file and keep it in your system at a convenient location.

2. Launch an EC2 Instance and keep the key pair handy

Before we deploy our spring boot app, we need to launch an EC2 instance. (Mentioned Above)

3. Copy jar file to AWS EC2

Let's keep the KeyPair and the Jar to be deployed into the same folder.Please note that If you are using a Linux machine you can use scp command out of the box.

Set the permissions of your private key file so that only you can read it. If you don't set the permission then you can not connect to your instance using keypair.

```
chmod 400 DemoKeyPair.pem
```

copy the files using SCP.

```
scp -i ./DemoKeyPair.pem ./demo-0.0.1-SNAPSHOT.jar ec2-user@ec2-34-240-45-168.eu-west-1.compute.amazonaws.com:~
```

4. SSH into the EC2 instance and Install Java 1.8

SSH into your Instance using Instance Connect. (Mentioned Above)

Install Java 1.8

`java -version` - to check whether java is installed

`sudo yum install java-1.8.0 -` if it’s not installed, you can install it using this command.

5. Run the Spring Boot Jar File on EC2

Java is installed in the EC2 instance and our jar file is present in the home directory of the instance. Run the following command.

```
java -jar demo-0.0.1-SNAPSHOT.jar
```

5.5.5. Scalability & Reliability

Reliability

- Email Authentication is there to make sure the email that is given by the user is his/hers and it exists.
- Reset forgot Password to make sure if any of the user forgets his/her credentials they can always get it back using their email.

Scalability

- Multiple users can access the server at the same time
- Pagination for the items available in the coffee machine to make sure the web application Stays Efficient
- MySQL can Easily Manage if the data gets into big data.

5.5.6. Security

- Security of the cloud – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon EC2, see AWS Services in Scope by Compliance Program.
- Security in the cloud – Your responsibility includes the following areas:
    - Controlling network access to your instances, for example, through configuring your VPC and security groups. For more information, see Controlling network traffic.
    - Managing the credentials used to connect to your instances.
    - Managing the guest operating system and software deployed to the guest operating system, including updates and security patches. For more information, see Update management in Amazon EC2.
    - Configuring the IAM roles that are attached to the instance and the permissions associated with those roles. For more information, see IAM roles for Amazon EC2.

- For more information https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security.html

6.0 Hardware Development

6.1. Required Components

Item	Quantity
NodeMCU ESP8266 12E	1
Heater	1
Gear Motor 5V	1
Relay module	3
Solenoid Valves	2
Servo Motor	2
TCRT5000L Reflective Optical Sensor	1
Ultra-sonic Sensor	2
5V power Supply (power packs 2A)	2

6.1.1. Sensors

Servo Motors

The correct amount of sugar and coffee powder (dry ingredients) is seperated from the storing containers to the icing unit.

- Power : 5 V
- Interface : PWM enabled pins (pin 0-16)
- Range : 0° to 180° (360°)

With the PWM signal, the control is,
0 degrees for a pulse width of 1ms
90 degrees with a pulse width of 1.5 ms

180 degrees with a 2 ms pulse width

- Positioning Accuracy : +/- 1°

Solenoid Valves

The water to make the coffee, and final output coffee (liquids) are separated using solenoid valves.

- Power : 12V
- Interface : 12V from the transform and the other terminal to GPIO

Ultra-sonic Sensor

Reliable hardware for measuring short distance with sufficient accuracy is used in here to track the available amount of ingredients in the containers.

- Model: HC-SR04
- Power : 5V
- Interface : Trigger input pin and Echo output pin to suitable GPIO
- pins in NodeMCU
- Maximum Range : 4 meters
- Minimum Range : 2 cm
- Ranging Accuracy : 3mm

TCRT5000L Reflective Optical Sensor

Checking whether the cup is available, only of the cup is there the whole process will start

- Model : TCRT 5000
- Power : 5 V
- Peak operating distance: 2.5 mm
- Operating range : 0.2 mm to 15 mm

6.1.2. Tasks of other components

NodeMCU ESP8266 12E

Provides the connectivity between the device and server. Low cost which reduces the price of the product and low energy consumption

- Operating Voltage : 3.3 V
- Flash Memory : 4 MB
- Available Interfaces:

  3.3 V Power Pin
  16 GPIO Pins

- Built in Units : WIFI module
- Clock speeds : 80MHz
- Programming Language : C

Heater

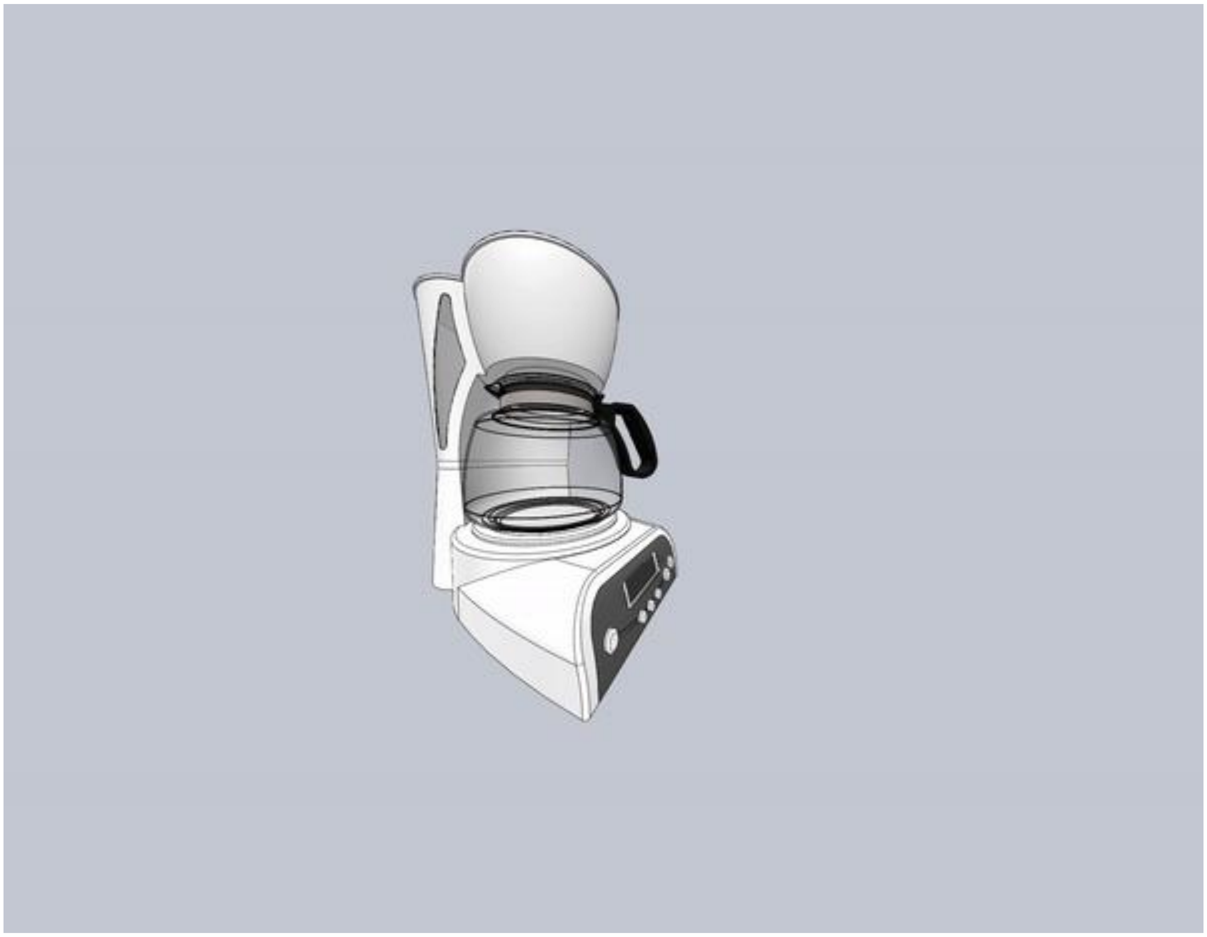
- Power : 230V
- Interface : Connected to a NodeMCU GPIO pin through a relay component

Geat Motor 5V

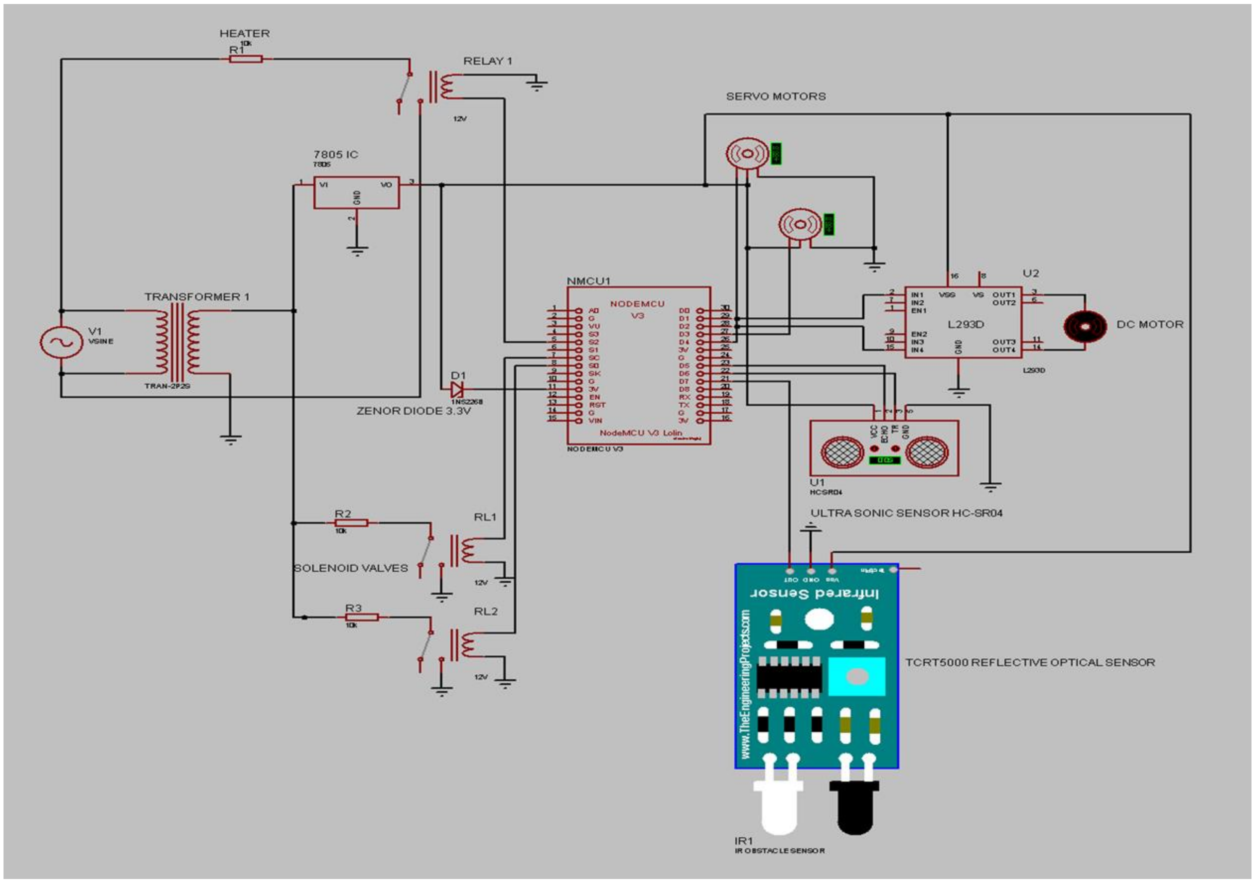
- Power : 5 V
- Interface : Connected to the board via L293D or relay
- Frequency : 200 rpm

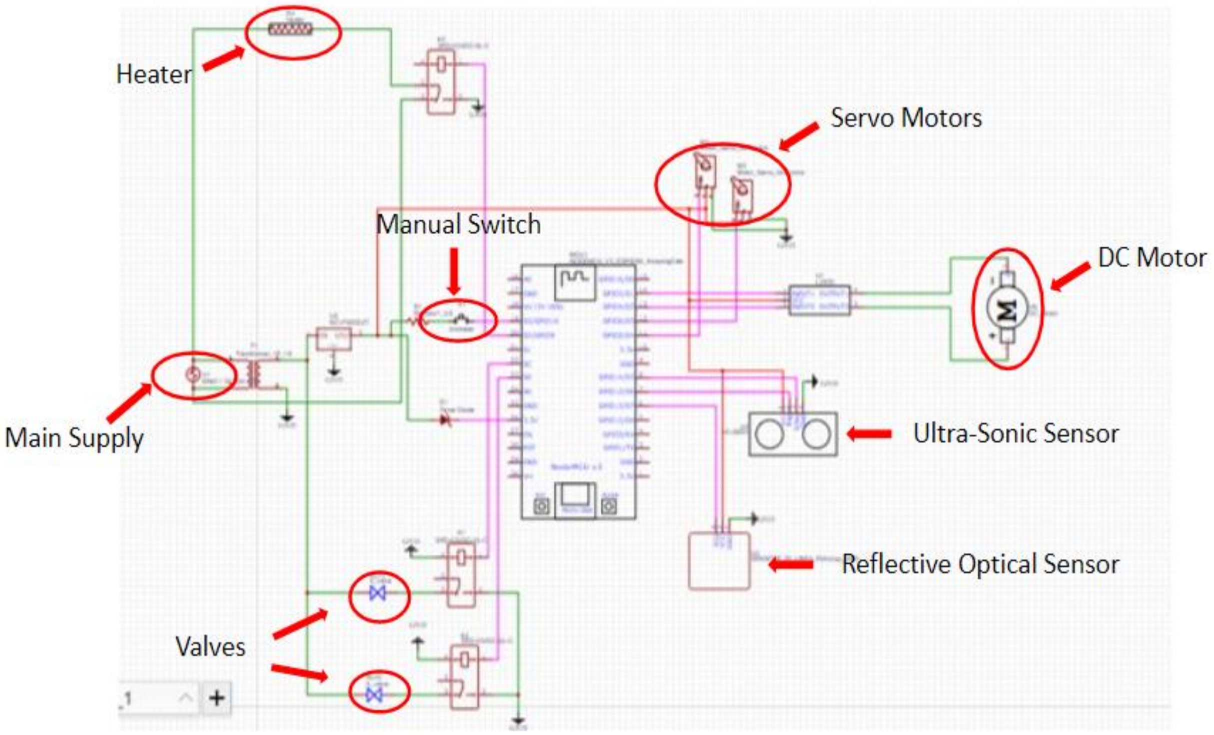
6.2. Designs and Diagrams

6.2.1. Designs for 3D printing



6.2.2. Circuit Diagrams





7. Security, Reliability and Scalability

7.1. Security

7.1.1. Hardware Aspects

- Password Authentication to access the storage unit.
- Using insulators to prevent overheating components due to the boiling unit.

7.1.2. Software Aspects

- Password Encoding Using BCrypt
- Use of JWT authentication
- Security groups
- EC2 security
- Access Management Control

7.2. Reliability.

7.2.1. Hardware Aspects

- Manual mode to operate when there is a network failure.

7.2.2. Software Aspects

- Well-secured Features
- Backups of the database provided by AWS

7.3. Scalability

7.3.1. Hardware Aspects

- The same design can be repeated to make different drinks.

7.3.2. Software Aspects

- Use of smaller, independent packages or modules while coding
- Auto-scaling feature
- Automatically maintain predictable performance at the lowest possible cost.

8. Resources

All the designs and codes can be found on our github repository.
https://github.com/cepdnaclk/e17-3yp-Smart-Pour

END