



## **Introduction to Cloud Service Providers**

Cloud Service providers are vendors who provide Information Technology (IT) as a service over the Internet. Cloud computing is a term which is used for storing and accessing data over the internet. It doesn't store any data on the hard disk of your PC. Cloud companies helps you to access your data from a remote server

Cloud computing companies' services range from full application development platforms to servers, storage, and virtual desktops. Here is a handpicked cloud service providers list. This cloud provider list contains various types of cloud computing services that are available in the market.

As a whole, the top 10 cloud service providers globally in 2022 are Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), Alibaba Cloud, Oracle Cloud, IBM Cloud (Kyndryl), Tencent Cloud, OVHcloud, DigitalOcean, and Linode (owned by Akamai).





## Amazon Web Services (AWS)

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers including the fastest-growing startups, largest enterprises, and leading government agencies are using AWS to lower costs, become more agile, and innovate faster.

### Setting up AWS Environment

In this practical sheet, we will take you through setting up your AWS account and development environment. This will allow you to interact with your AWS account and provision any resources you need for building a system programmatically.

#### Step by step how create AWS free tire Account

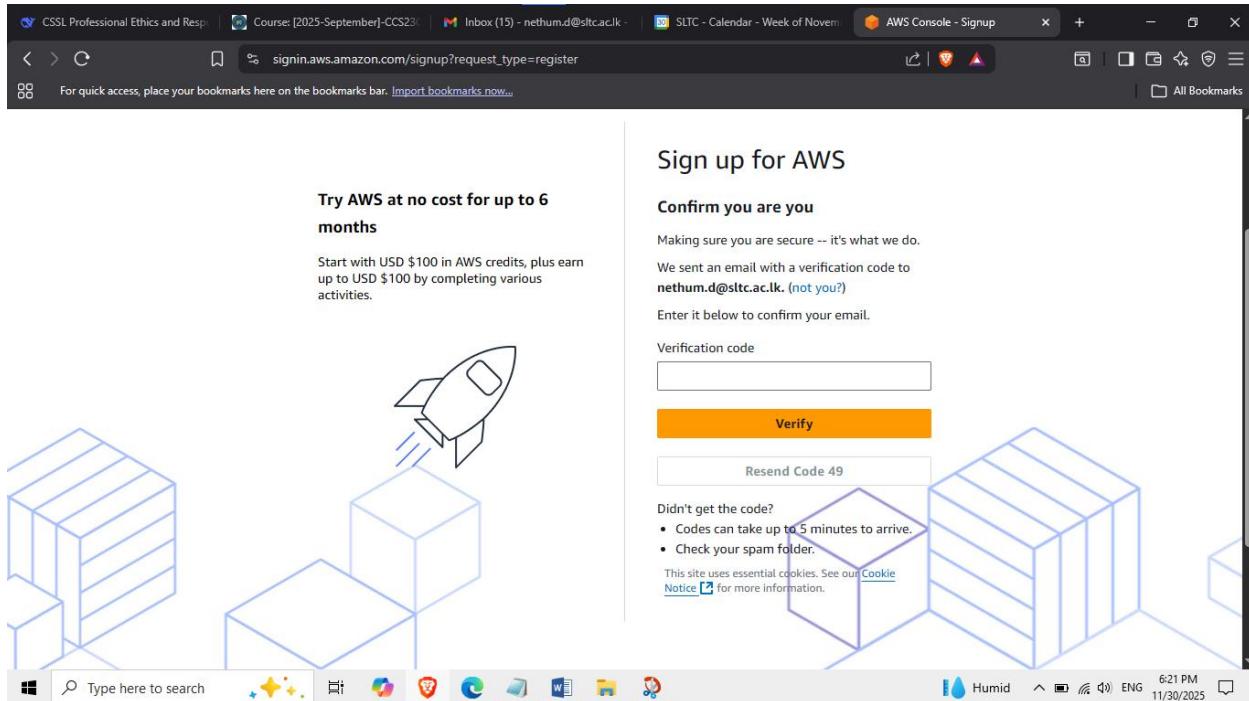
An AWS Free tire account is the starting point to allow provisioning infrastructure. In this step, we will cover how to set up your account.

<https://www.youtube.com/watch?v=Q6eMTgUDPXg&t=2s>

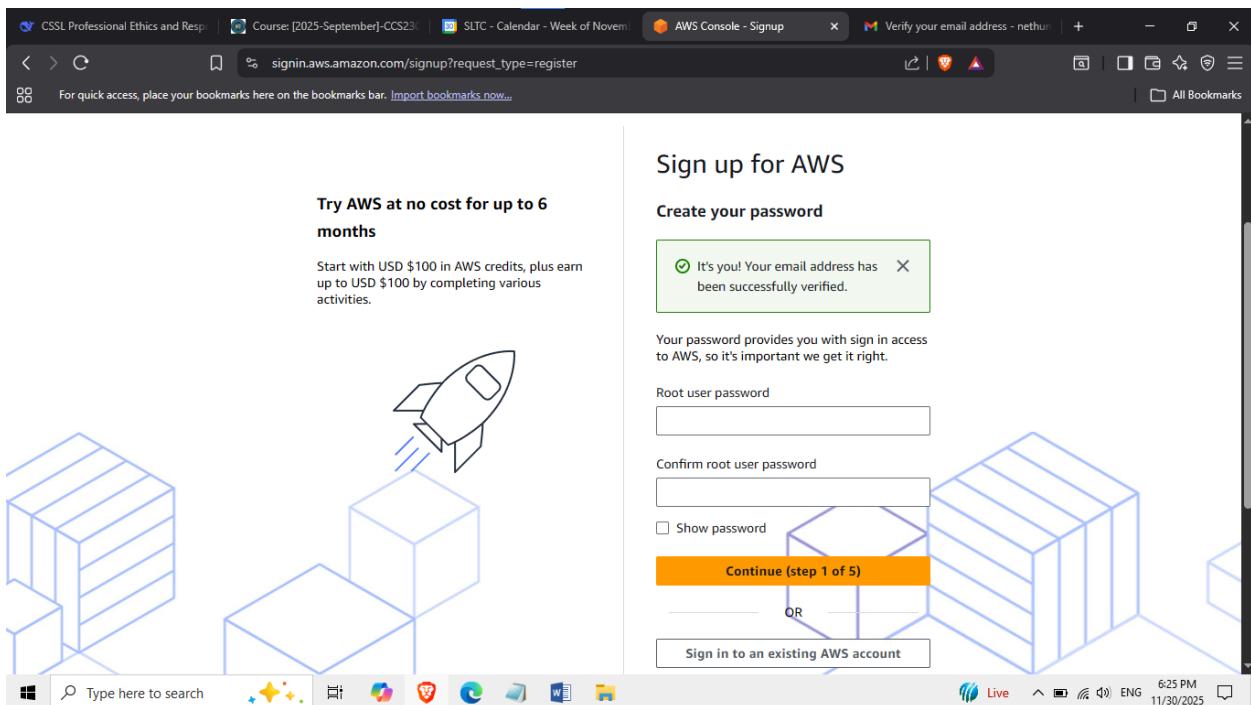
Step 1 – Visit the official AWS website and set up an email account

The screenshot shows a Windows desktop with a browser window open to the AWS Sign-up page. The URL in the address bar is [signin.aws.amazon.com/signup?request\\_type=register](https://signin.aws.amazon.com/signup?request_type=register). The page title is "Sign up for AWS". It features a "Try AWS at no cost for up to 6 months" section with a description about credits and activities, accompanied by a rocket launching from a stack of cubes illustration. On the right, there are fields for "Root user email address" (nethum.d@sltca.ac.lk), "AWS account name" (Nethum), and a "Verify email address" button. Below this is an "OR" link and a "Sign in to an existing AWS account" section with a "Security check" step. The desktop taskbar at the bottom includes icons for File Explorer, Task View, Edge, and other Microsoft applications. The system tray shows the date (11/30/2025), time (6:19 PM), battery status, and network connection.

## Step 2 – Get Email Verification code and put your AWS account.



## Step 3 - Set a password for the root user of your AWS account



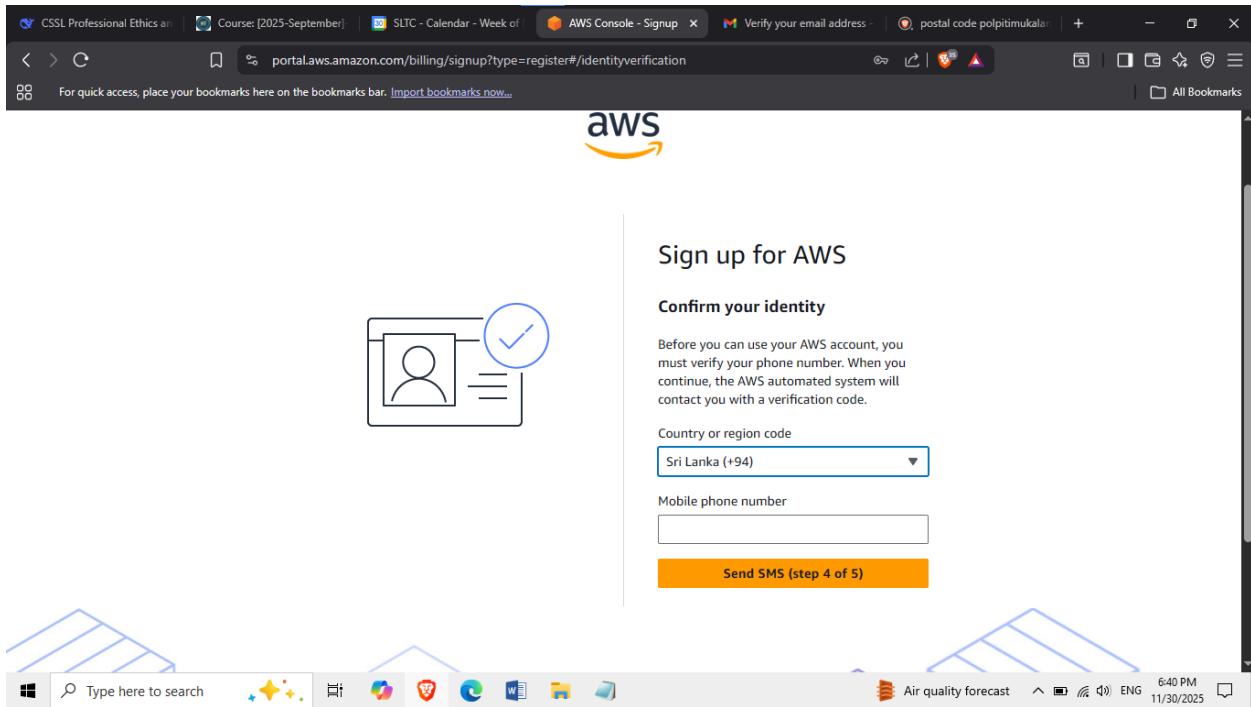
## Step 4 – Choose the AWS Free Tier account and provide your personal details

The screenshot shows the 'Sign up for AWS' page. On the left, there's a section titled 'Earn additional AWS credits' with a description and a graphic of a dollar bill. On the right, under 'Contact Information', there are fields for 'Full Name', 'Country Code' (set to +1), 'Phone Number' (222-333-4444), 'Country or Region' (United States), 'Address line 1', and 'Address line 2'. The Windows taskbar at the bottom includes icons for search, file explorer, and various Microsoft applications.

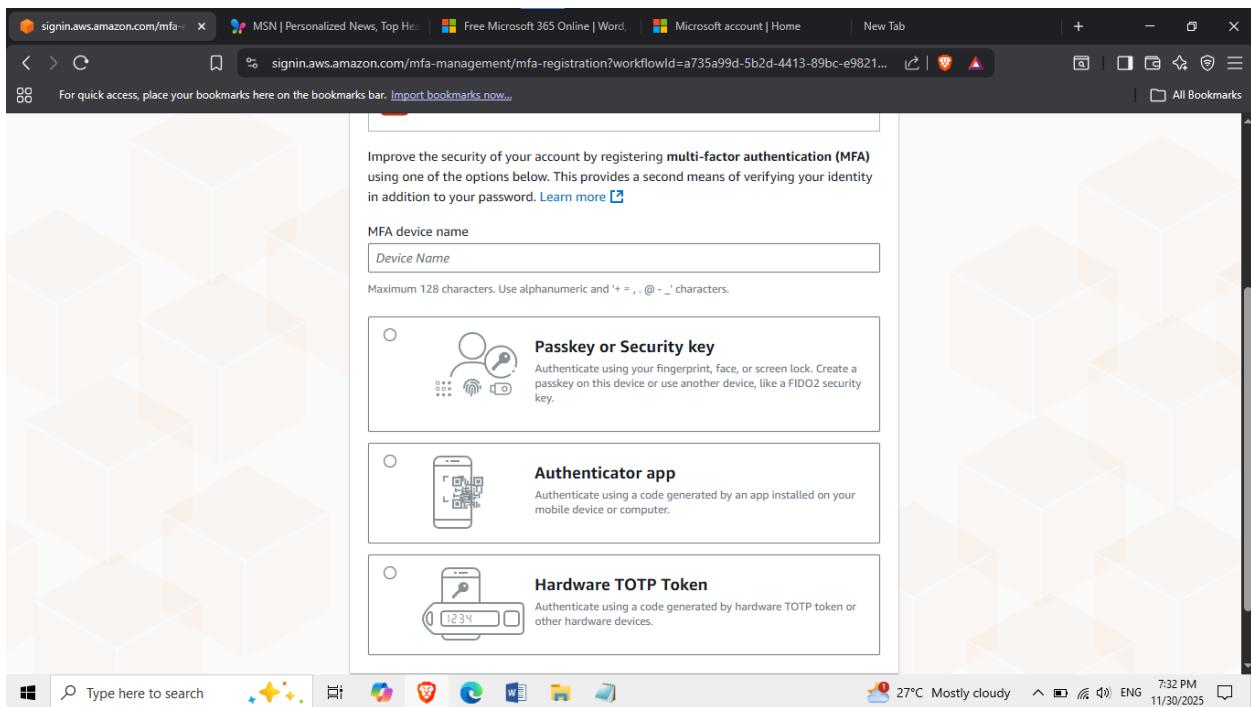
## Step 5 – Enter your billing details, including your credit card information.

The screenshot shows the 'Billing Information' page. On the left, there's a box explaining the verification process and a graphic of a shield with a checkmark. On the right, there are fields for 'Credit or Debit card number' (with an error message: 'The credit card number is required.'), 'Expiration date' (Month and Year dropdowns), 'Security code' (CVV/CVC), 'Cardholder's name', and 'Billing address' (with an option to 'Use my contact address'). The Windows taskbar at the bottom includes icons for search, file explorer, and various Microsoft applications.

Step 6 – Put your phone number and verify the identity details.



Step 7 – To secure your AWS account, connect an authenticator app to enable Multi-Factor Authentication (MFA).

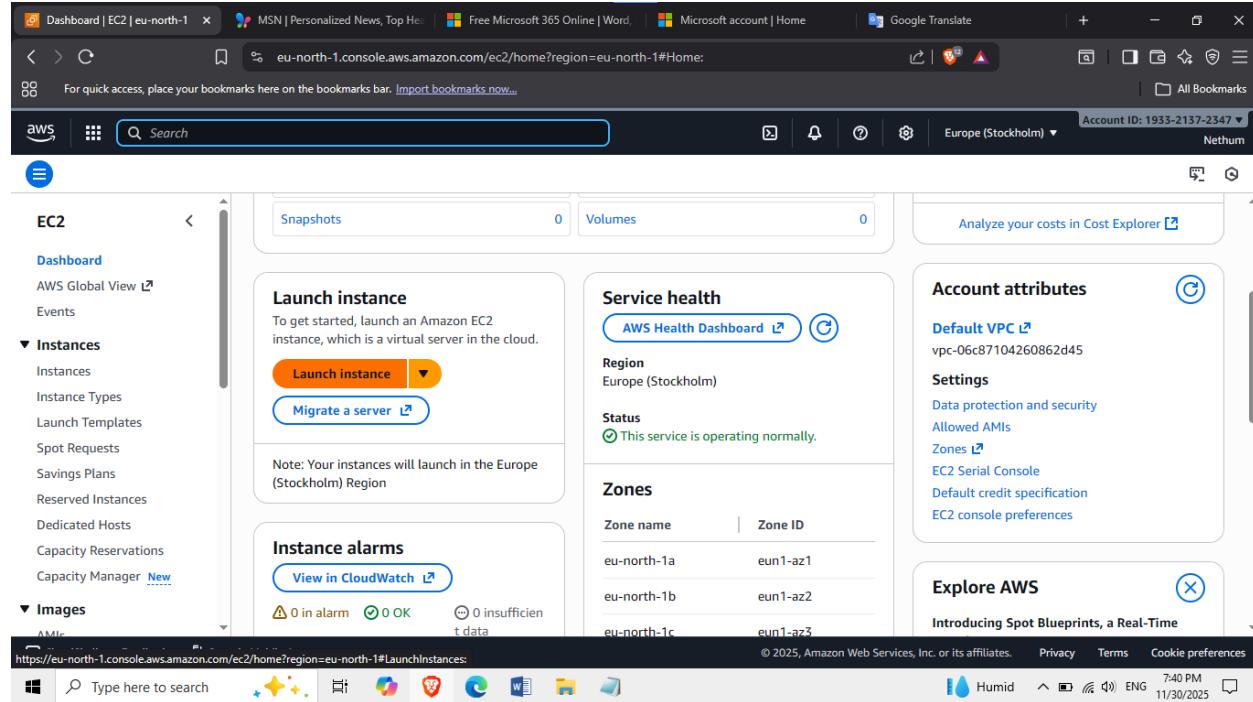


## Launch EC2 Instance

Amazon EC2 (Elastic Compute Cloud) is a web service from Amazon Web Services (AWS) that provides secure, resizable compute capacity in the cloud.

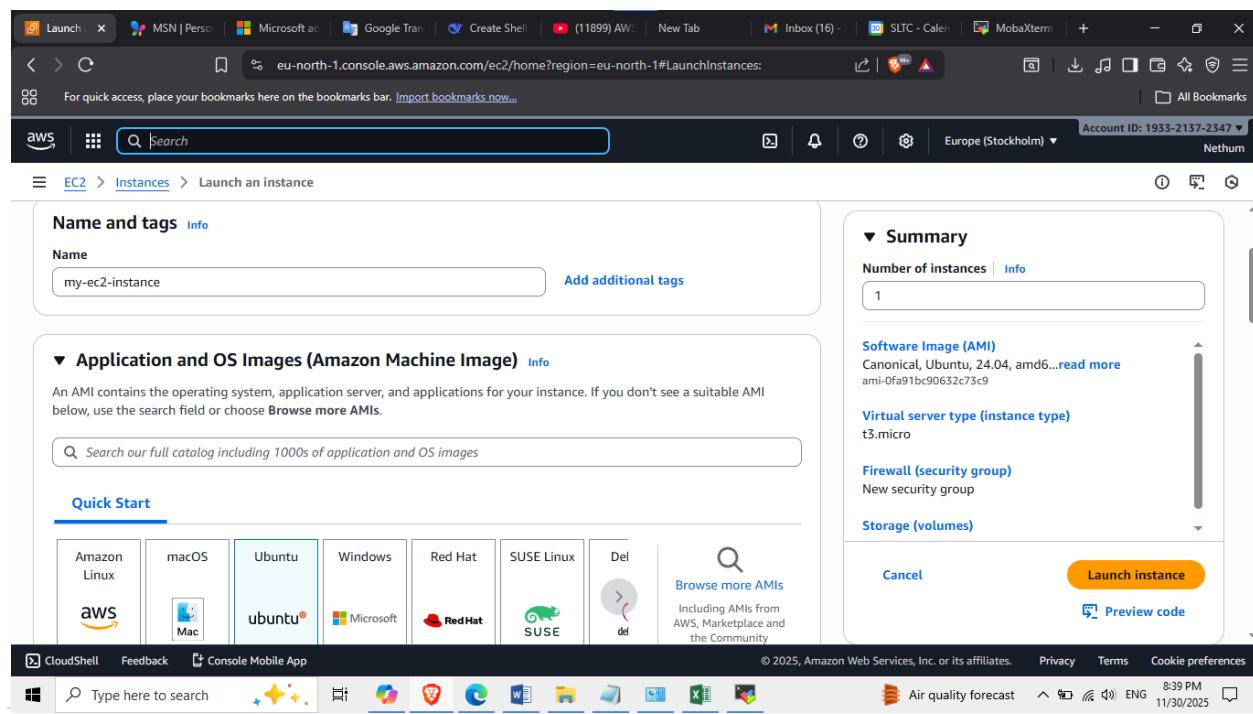
<https://www.youtube.com/watch?v=2zeoNC4cdTA&t=3s>

Step 1 – Search Ec2 Instance in search bar and launch Instance.



The screenshot shows the AWS EC2 Instances dashboard. On the left sidebar, under 'Instances', there is a 'Launch instance' button. The main area displays 'Service health' (Region: Europe (Stockholm), Status: This service is operating normally) and 'Zones' (eu-north-1a, eu-north-1b, eu-north-1c). To the right, there's an 'Account attributes' section and an 'Explore AWS' section. The browser address bar shows the URL: <https://eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#LaunchInstances>.

Step 2 – Assign a name to the EC2 instance and select its operating system (e.g., Ubuntu).



The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' section, the name 'my-ec2-instance' is entered. In the 'Application and OS Images (Amazon Machine Image)' section, 'Ubuntu' is selected. Other options include Amazon Linux, macOS, Windows, Red Hat, SUSE Linux, and Del. A 'Quick Start' section shows icons for various operating systems. On the right, there's a 'Summary' section with 'Number of instances: 1', 'Software Image (AMI)' (Canonical, Ubuntu, 24.04, amd64...read more), 'Virtual server type (instance type)' (t3.micro), 'Firewall (security group)' (New security group), and 'Storage (volumes)'. At the bottom right is a 'Launch instance' button. The browser address bar shows the URL: <https://eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#LaunchInstances>.

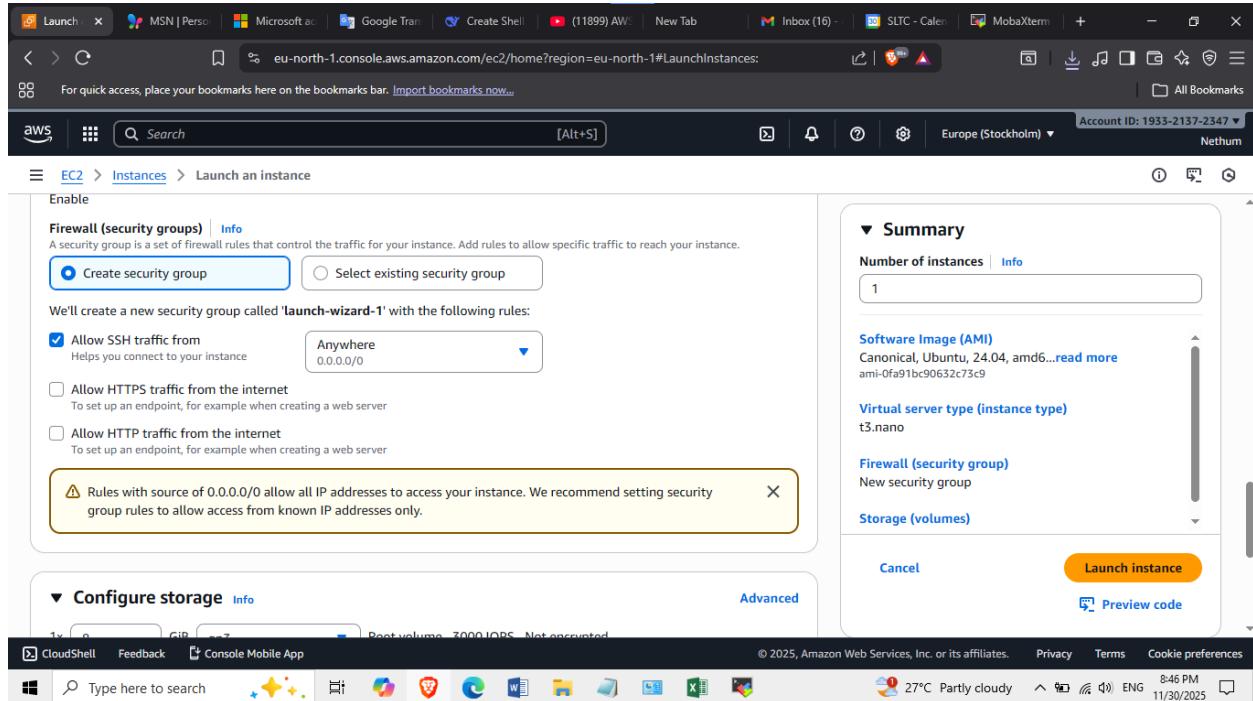
Step 3 – Provide Instance type, When you choose Free tire account choose recommended type. (T2.micro).

The screenshot shows the AWS EC2 Instances Launch an instance page. In the 'Instance type' section, 't3.nano' is selected. The details for t3.nano are shown: Family: t3, 2 vCPU, 0.5 GiB Memory, Current generation: true. On-Demand Ubuntu Pro base pricing: 0.0089 USD per Hour. On-Demand Windows base pricing: 0.01 USD per Hour. On-Demand SUSE base pricing: 0.0054 USD per Hour. On-Demand Linux base pricing: 0.0054 USD per Hour. Below this, a note says 'Additional costs apply for AMIs with pre-installed software'. To the right, there's a 'Summary' section with a 'Launch instance' button.

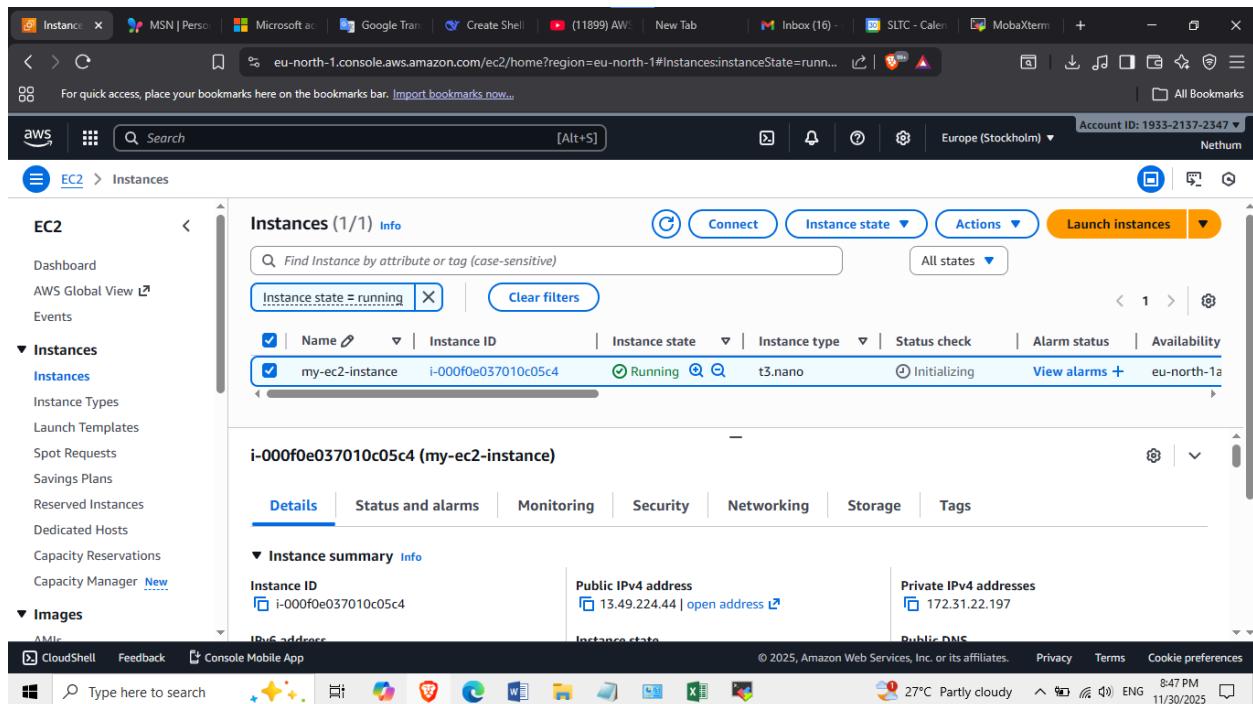
Step 4 - Generate a key pair to connect to the VM remotely. If you are using PuTTY, choose the .ppk

The screenshot shows the 'Create key pair' dialog box overlaid on the EC2 Instances Launch an instance page. The 'Key pair name' field contains 'myloginaccess'. Under 'Key pair type', 'RSA' is selected. Under 'Private key file format', '.pem' is selected. The 'Create key pair' button is highlighted in orange. The background shows the same instance configuration as the previous screenshot.

## Step 5 – Click Allow SSH traffic



## Step 6 – After completed above stages, then launch the EC2 Instance.



## Step 7 – EC2 Instance Details,

The screenshot shows the AWS Management Console interface for an EC2 instance. The left sidebar is collapsed, and the main content area displays the 'Instance summary' for the instance ID `i-000f0e037010c05c4`, which is named `my-ec2-instance`. The summary includes the following details:

- Instance ID:** `i-000f0e037010c05c4`
- IPv6 address:** -
- Hostname type:** IP name: `ip-172-31-22-197.eu-north-1.compute.internal`
- Answer private resource DNS name:** IPv4 (A)
- Auto-assigned IP address:** `13.49.224.44` [Public IP]
- Public IPv4 address:** `13.49.224.44` | [open address](#)
- Instance state:** `Running`
- Private IP DNS name (IPv4 only):** `ip-172-31-22-197.eu-north-1.compute.internal`
- Instance type:** `t3.nano`
- VPC ID:** `vpc-06c87104260862d45`
- Private IP4 addresses:** `172.31.22.197`
- Public DNS:** `ec2-13-49-224-44.eu-north-1.compute.amazonaws.com` | [open address](#)
- Elastic IP addresses:** -
- AWS Compute Optimizer finding:** [Opt-in to AWS Compute Optimizer for recommendations](#)

The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time.

## Connect Remote Server (EC2 Instance) using SSH Key (Private Key)

This process allows you to securely log in to your Amazon EC2 instance in the cloud from your local computer.

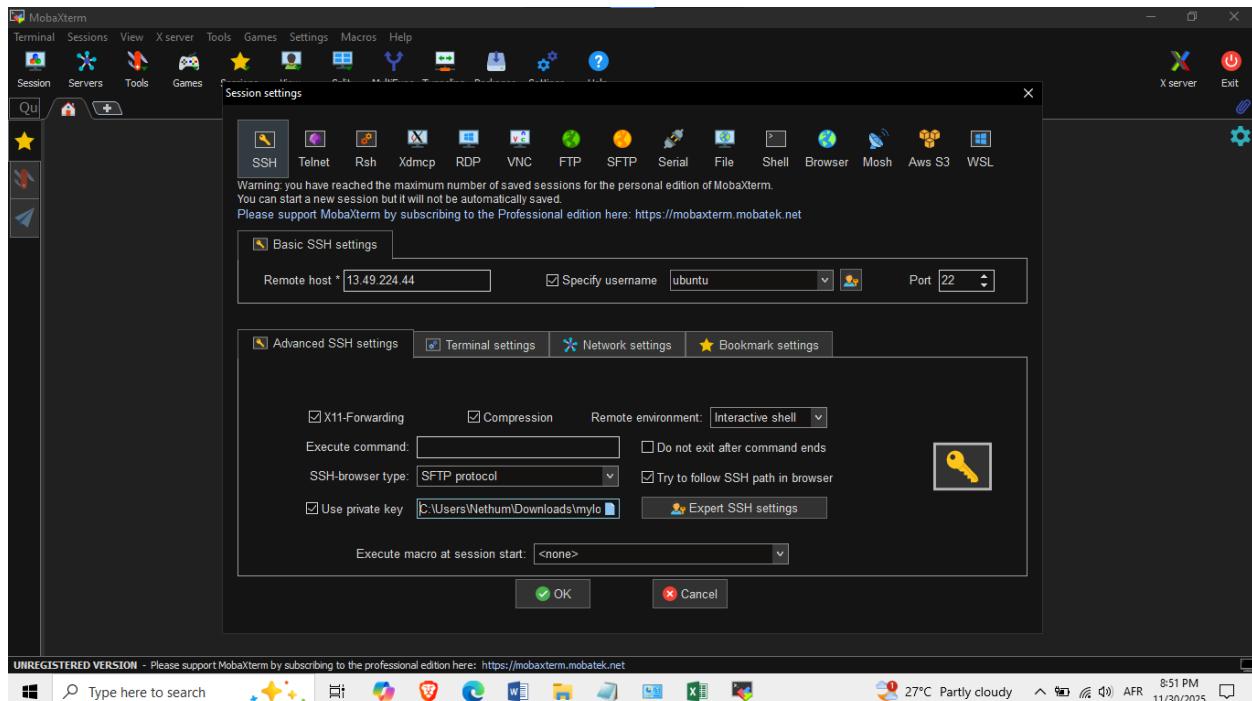
### Step 1 – Downloaded and Open MobaXterm Software Then Open SSH – <https://mobaxterm.mobatek.net/download-home-edition.html>

The screenshot shows the MobaXterm software interface. A 'Session settings' dialog box is open, specifically the 'SSH' tab. The configuration fields include:

- Remote host:** (empty text input field)
- Specify username:** (empty dropdown menu)
- Port:** `22` (dropdown menu)

Below these fields are tabs for 'Advanced SSH settings', 'Terminal settings', 'Network settings', and 'Bookmark settings'. At the bottom of the dialog are 'OK' and 'Cancel' buttons. The background of the main MobaXterm window shows a session titled 'Secure Shell (SSH) session' with a small key icon.

Step 2 – To connect, enter your EC2 instance's IPv4 address, use "ubuntu" as the username, and provide the path to your downloaded key pair file.



Step 3 – After done above part and press OK button then automatically connect to remote Virtual machine.

```
13.49.224.44 (ubuntu)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Qu [2 - 13.49.224.44 (ubuntu)] + [ ]
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Sun Nov 30 15:22:22 UTC 2025
System load: 0.0 Temperature: -273.1 C
Usage of /: 25.9% of 6.71GB Processes: 113
Memory usage: 50% Users logged in: 0
Swap usage: 0% IPV4 address for ens5: 172.31.22.197

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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.

/usr/bin/xauth: file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-22-197:~$ ubuntu@ip-172-31-22-197:~$ ubuntu@ip-172-31-22-197:~$ ubuntu@ip-172-31-22-197:~$ 
```

## Create simple HTML code and connect to the web server via EC2 Instance

- Create and open html file in to your EC2 instance.  
Code :- [Sudo vi welcome.html](#)
- Copy and paste to HTML code inside the welcome.html folder

```
<!DOCTYPE html>
<html>
<head>
    <title>AWS Welcome CCS2302-Cloud Computing Fundamentals</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin: 100px;
            background: #232f3e;
            color: white;
        }
        .welcome {
            font-size: 2.5em;
            color: #ff9900;
            margin-bottom: 30px;
        }
        .info {
            background: #334155;
            padding: 15px;
            margin: 10px;
            border-radius: 5px;
            display: inline-block;
        }
    </style>
</head>
<body>
    <div class="welcome">Hello Welcome AWS</div>
    <div class="info" id="dateInfo">Loading date...</div>
    <div class="info" id="instanceInfo">My Student ID Is " "</div>

    <script>
        document.getElementById('dateInfo').textContent = 'Current date: ' + new
Date().toLocaleString();
        // EC2 metadata would be fetched here in a real EC2 environment
    </script>
```

- ```
</body>
</html>
```
- Serve the HTML file using a web server (Apache2 web server)
 

```
# Install Apache
sudo apt update
sudo apt install -y apache2
```

```
# Copy HTML file to web directory
sudo cp welcome.html /var/www/html/
```

```
# Start Apache
sudo systemctl start apache2
sudo systemctl enable apache2
```
  - Installed apache and check the status
 

```
sudo systemctl status apache2
```

```
13.49.224.44 (ubuntu)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Qu 2: 13.49.224.44 (ubuntu) × +
X server Exit

Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-22-197:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-11-30 15:32:09 UTC; 28s ago
     Docs: http://httpd.apache.org/docs/2.4/
Main PID: 198 (apache2)
   Tasks: 55 (limit: 411)
      Memory: 6.0M (peak: 7.8M)
        CPU: 42ms
       CGroup: /system.slice/apache2.service
           ├─198 /usr/sbin/apache2 -k start
           ├─2200 /usr/sbin/apache2 -k start
           └─2201 /usr/sbin/apache2 -k start

Nov 30 15:32:09 ip-172-31-22-197 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Nov 30 15:32:09 ip-172-31-22-197 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-172-31-22-197:~$
```

UNREGISTERED VERSION - Please support Mobaxterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

- Go to EC2 Instances and add these Inbound rules,
  - # Type: SSH, Port: 22, Source: 0.0.0.0/0 (or your IP)
  - # Type: HTTP, Port: 80, Source: 0.0.0.0/0
  - # Type: HTTPS, Port: 443, Source: 0.0.0.0/0

The screenshot shows the AWS EC2 Security Groups console. A green success message at the top states: "Inbound security group rules successfully modified on security group (sg-055c094b24da8806e | launch-wizard-1)". Below it, the "Details" section shows the owner (193321372347) and counts of inbound (3) and outbound (1) rules. The "Inbound rules" tab is selected, displaying three rules:

Security group rule ID	IP version	Type	Protocol	Port range
sgr-0d63bffa8c1db0ab0	IPv4	HTTPS	TCP	443
sgr-0599c50283db0371d	IPv4	HTTP	TCP	80
sgr-0a68872b390488466	IPv4	SSH	TCP	22

- Open your Browser and copy IPV4 address and navigate this;  
<http://your-ec2-public-ip/welcome.html>

The screenshot shows a browser window displaying a welcome page for an AWS EC2 instance. The URL in the address bar is "Not secure 13.49.224.44/welcome.html". The page content includes:

- A cloud icon at the top.
- The text "Hello Welcome AWS: My Id is = "put your SID"".
- A "Current Date & Time" section showing "Loading...".
- An "EC2 Instance Information" section showing "Instance ID: Not available (run on EC2)" and "Availability Zone: Not available (run on EC2)".
- A "System Status" section showing "Active".

## AWS S3 Bucket Creation

An AWS S3 bucket is a logical container in Amazon Web Services (AWS) Simple Storage Service (S3) used to store and manage objects, which are files along with their associated metadata and a unique key identifier. Each object must reside within a bucket, as it cannot exist independently. Buckets are designed to be globally unique in name and are associated with a specific AWS region. They provide scalable, durable, and secure object-based storage, capable of holding an unlimited amount of data.

Azure also provides a similarly diverse set of storage services. serves as the primary object storage solution, suitable for storing large amounts of unstructured data such as documents, media files, and application installers.

## Setting up AWS Environment



## Step by step create a S3 bucket

- ❖ Log into the AWS Console, navigate to the S3 service, and click "Create bucket."
  - AWS Account Login - <https://aws.amazon.com/console/>

A screenshot of the AWS S3 'Create a bucket' page. The top navigation bar shows the AWS logo, search bar, and account information (Account ID: 1933-2137-2347, Europe (Stockholm)). The main content area has a dark background with white text. On the left, there's a section titled 'Amazon S3' with the sub-headline 'Store and retrieve any amount of data from anywhere'. Below this is a note: 'Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.' On the right, there's a 'Create a bucket' button. A callout box above it says: 'Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.' At the bottom, there are two sections: 'How it works' (with a screenshot of the AWS console) and 'Pricing' (with a note about no minimum fees and a link to the AWS Simple Monthly Calculator).

- ❖ Created a S3 bucket -> when you create it choose a unique name (e.g., my-website-123)

The screenshot shows the AWS S3 console with a green success message at the top: "Successfully created bucket 'nethum-s3-test1'. To upload files and folders, or to configure additional bucket settings, choose View details." Below this, the "General purpose buckets" section lists one bucket: "nethum-s3-test1" (Europe (Stockholm) eu-north-1). The "Create bucket" button is visible. On the right, there are "Account snapshot" and "External access summary" cards.

- ❖ Navigate the S3 bucket and click upload,

The screenshot shows the AWS S3 console for the bucket "nethum-s3-test1". The "Objects" tab is selected, showing 0 objects. A prominent "Upload" button is located at the bottom center. The page includes tabs for Objects, Metadata, Properties, Permissions, Metrics, Management, and Access Points.

- ❖ Add the simple html base file in to the bucket, (Copy the code below to create an HTML file)

```
<!DOCTYPE html>
<html>
<head>
    <title>My S3 Website</title>
    <style>
        /* Simple CSS */
        body {
            font-family: Arial, sans-serif;
            background-color: #f5f5f5;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
            text-align: center;
        }

        .container {
            background-color: white;
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 2px 10px rgba(0,0,0,0.1);
            max-width: 500px;
        }

        h1 {
            color: #FF9900; /* AWS orange */
            margin-bottom: 20px;
        }

        p {
            color: #333;
            font-size: 18px;
            margin-bottom: 20px;
        }

        button {
            background-color: #FF9900;
            color: white;
            border: none;
            padding: 10px 20px;
            border-radius: 5px;
        }
    </style>

```

```

        cursor: pointer;
        font-size: 16px;
    }

    button:hover {
        background-color: #e68a00;
    }

    .hidden {
        display: none;
    }

    .message {
        background-color: #f0f8ff;
        padding: 10px;
        border-radius: 5px;
        margin-top: 20px;
        border-left: 4px solid #FF9900;
    }

```

</style>

</head>

<body>

<div class="container">

<h1>Hello from AWS S3!</h1>

<p>My static website is now live.</p>

<button id="toggleButton">Click for more info</button>

<div id="moreInfo" class="hidden">

<p>This site is hosted on Amazon S3, a scalable cloud storage service.</p>

</div>

<div id="message" class="message hidden"></div>

</div>

<script>

```

// Simple JavaScript
document.addEventListener('DOMContentLoaded', function() {
    const toggleButton = document.getElementById('toggleButton');
    const moreInfo = document.getElementById('moreInfo');
    const messageDiv = document.getElementById('message');

    toggleButton.addEventListener('click', function() {
        // Toggle the hidden info
        if (moreInfo.classList.contains('hidden')) {

```

```

        moreInfo.classList.remove('hidden');
        toggleButton.textContent = 'Show less';

        // Update the message
        messageDiv.textContent = 'Amazon S3 provides secure, durable, and scalable
object storage.';

        messageDiv.classList.remove('hidden');
    } else {
        moreInfo.classList.add('hidden');
        toggleButton.textContent = 'Click for more info';
        messageDiv.classList.add('hidden');
    }
});

// Change background color on mouse move
document.addEventListener('mousemove', function(e) {
    const x = e.clientX / window.innerWidth;
    const y = e.clientY / window.innerHeight;

    // Very subtle color change
    document.body.style.backgroundColor = `rgb(${245 + x * 10}, ${245 + y * 10},
245)`;

});

// Simple page load animation
const container = document.querySelector('.container');
container.style.opacity = '0';
container.style.transform = 'translateY(20px)';

setTimeout(() => {
    container.style.transition = 'opacity 0.5s, transform 0.5s';
    container.style.opacity = '1';
    container.style.transform = 'translateY(0)';
}, 100);
});

</script>
</body>
</html>

```

The screenshot shows the AWS S3 console with a green success message at the top: "Upload succeeded". Below it, a summary table shows the destination as "s3://nethum-s3-test1" with 1 file successfully uploaded (Succeeded) and 0 files failed (Failed). A "Files and folders" tab is selected, displaying a table with one item: "S3.html" (text/html, 4.0 KB, Status: Succeeded).

The screenshot shows the AWS S3 console with the "Upload" configuration screen. It displays a table of files to be uploaded, showing "S3.html" (text/html, 4.0 KB). Below this, the "Destination" section shows "s3://nethum-s3-test1" and includes a "Destination details" section. The "Permissions" section is also visible.

- ❖ If unable to code run in S3 bucket, To set the bucket policy to public, go to the Permissions tab, find the Bucket policy section, click Edit, and add the following policy (ensuring you replace BUCKET\_NAME).
  - Copy and paste under the policy,

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::BUCKET_NAME/*",
      "Condition": {
        "StringEquals": {
          "aws:Referer": "https://your-domain.com"
        }
      }
    }
  ]
}
```

The screenshot shows the AWS S3 Bucket Policy editor. At the top, there's a navigation bar with the AWS logo, a search bar, and account information (Account ID: 1933-2137-2347, Europe (Stockholm)). Below the navigation is a breadcrumb trail: Amazon S3 > Buckets > nethum-s3-test1 > Edit bucket policy. The main area is titled "Bucket policy". It displays a JSON policy block with line numbers 1 through 12. Lines 1-11 define a single statement, and line 12 shows a closing brace. To the right of the policy is a sidebar with a "Policy examples" button. A modal window is open over the sidebar, titled "Edit statement". Inside the modal, it says "Select a statement" and "Select an existing statement in the policy or add a new statement." At the bottom of the modal is a blue button labeled "+ Add new statement". The bottom of the screen shows the standard AWS footer with links for CloudShell, Feedback, Console Mobile App, and various AWS services.

- ❖ Copy the Bucket website endpoint URL and access the website.

