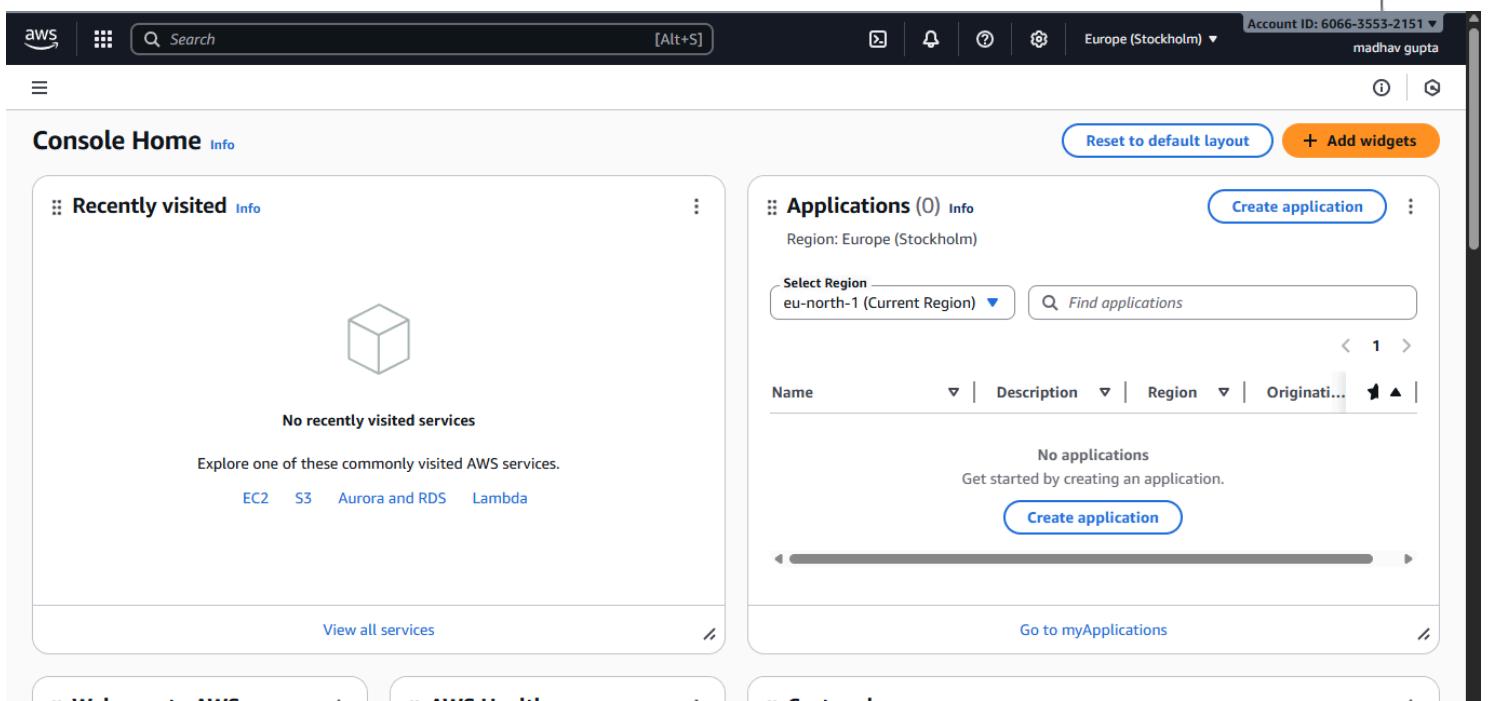


AWS Assignment-1

❖ Task 1: AWS Free Tier Account Creation

Root account



□ Steps:

1. Go to the **AWS Management Console** → <https://aws.amazon.com/console>.
2. Click **Sign in as root user**.
3. Enter the **email address** (used when creating the account).
4. Enter the **root password**.
5. (If MFA is enabled) Enter the **MFA code**.
6. After entering the **MFA code**, root user will be created

❖ Task 2: Root-Equivalent IAM User Creation

The screenshot shows the AWS IAM Users page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' selected. A callout box highlights the creation of a new user: 'IAM user created as:--adminuser'. The main area shows a table of users with one entry: 'adminuser'. The table includes columns for User name, Path, Groups, Last activity, MFA, Password age, and Console. The 'Last activity' column shows '4 minutes' with a checkmark.

□ Steps:

Step 1: Sign in as the Root User

- Go to the [AWS Management Console](#).
- Click “Sign in as root user”.
- Enter the **root email** and **password**.
- (If MFA is enabled, enter the MFA code.)

The screenshot shows the AWS Console Home page. The left sidebar has 'Console Home' selected. The main area features a 'Recently visited' section with a placeholder message 'No recently visited services' and links to EC2, S3, Aurora and RDS, and Lambda. The right side shows the 'Applications' section, which is currently empty ('No applications'). It includes a 'Create application' button and a 'Find applications' search bar.

Step 2: Open the IAM Console

- In the AWS console search bar, type “IAM”.
- Click **IAM (Identity and Access Management)** to open it.

The screenshot shows the AWS IAM Console's 'Console Home' page. On the left sidebar, under 'Recently visited', 'Systems Manager', 'EC2', 'IAM', and 'VPC' are listed. The main content area is titled 'Applications (0) Info' and shows a message: 'Region: Europe (Stockholm)' and 'Select Region: eu-north-1 (Current Region)'. A search bar 'Find applications' is present. Below, there is a table header for 'Name', 'Description', 'Region', and 'Originati...'. A message at the bottom says 'No applications' and 'Get started by creating an application.' with a 'Create application' button.

Step 3: Create a New IAM User

1. In the left sidebar, click **Users → Add users**.
2. Enter a **username** (e.g., admin-user).
3. Under **Select AWS access type**, choose:
 - **Password - AWS Management Console access** (if user needs console access)
 - **Access key - Programmatic access** (if user needs CLI or API access)

Click **Next**.

The screenshot shows the 'Specify user details' step of the IAM User creation wizard. The left sidebar shows steps: Step 1 (selected), Step 2, Step 3, Step 4. The main area has a title 'Specify user details'. It contains a 'User details' section with a 'User name' input field containing 'adminuser'. Below it is a note: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . _ - (hyphen)'. There is a checked checkbox 'Provide user access to the AWS Management Console - optional' with a note: 'In addition to console access, users with SignInLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.' Below this is a 'Console password' section with a radio button for 'Autogenerated password' (selected) and 'Custom password'. A note for 'Custom password' says 'Enter a custom password for the user.' At the bottom, there are two checkboxes: 'Show password' (unchecked) and 'Users must create a new password at next sign-in - Recommended' (checked).

Step 4: Attach Administrator Permissions

1. Choose **Attach policies directly**.
2. Search for and select **AdministratorAccess**.
 - o This gives the user full permissions for all AWS services (like the root user, but still restricted by IAM).
3. Click **Next**.

The screenshot shows the AWS IAM 'Create user' wizard at Step 4: Set permissions. In the 'Permissions options' section, the 'Attach policies directly' radio button is selected. Below this, a table lists 'Permissions policies (1/1419)'. The table includes columns for Policy name, Type, and Attached entities. The row for 'AdministratorAccess' is selected, showing it is an AWS managed - job function policy.

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0
AdministratorAccess	AWS managed - job function	0
AdministratorAccess-Amplify	AWS managed	0

Step 5: Review and Create

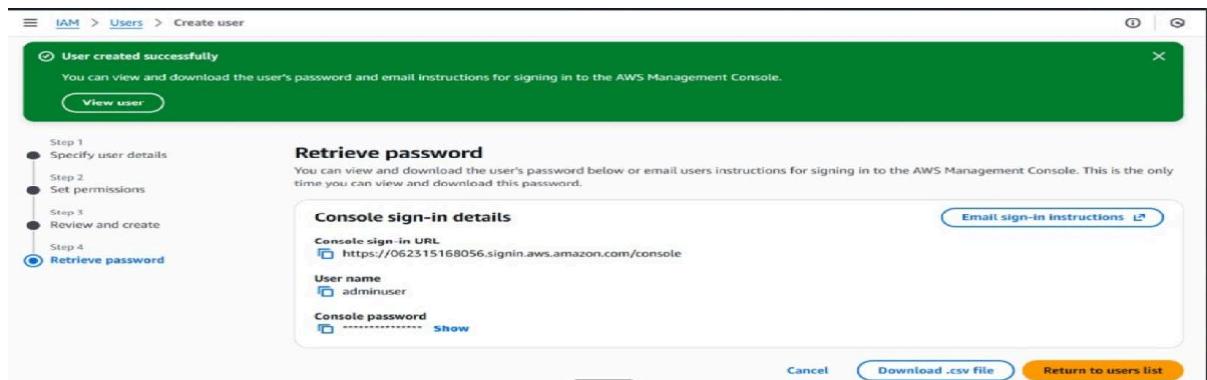
- Review all details.
- Click **Create user**.

AWS will show the **login URL**, **username**, and (if applicable) the **password or access keys**.

The screenshot shows the 'Review and create' step of the 'Create user' wizard. It displays the 'User details' (User name: adminuser) and 'Permissions summary' (Name: AdministratorAccess, Type: AWS managed - job function, Used as: Permissions policy). Below these, there is a 'Tags - optional' section which is currently empty.

Step 6: Secure the Account

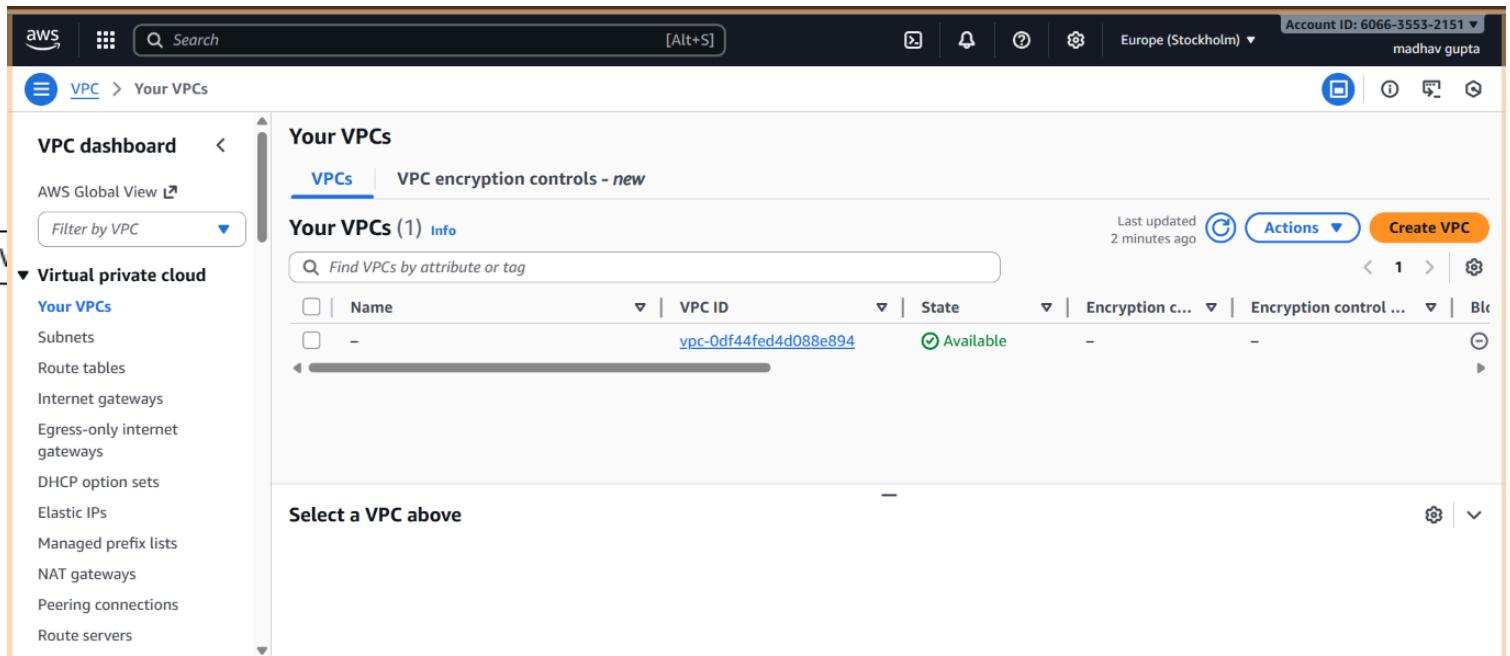
1. **Sign in as the new IAM user** to confirm it works.
2. **Enable MFA (Multi-Factor Authentication):**
 - o Go to **IAM → Users → [your user] → Security credentials**.
 - o Under **Multi-factor authentication (MFA)**, choose **Assign MFA device**.
 - o Use an **authenticator app** (like Google Authenticator).
3. Store credentials securely (not in plaintext).



- Hence, IAM user created as **adminuser**

The screenshot shows the 'Users' list page in the AWS Management Console under the 'Identity and Access Management (IAM)' section. The left sidebar includes 'Dashboard', 'Access management' (with 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Root access management', 'Temporary delegation requests', and 'Access reports' sub-options), and 'Access reports'. The main content area shows a table titled 'Users (1) info' with one entry: 'adminuser'. The table columns include 'User name' (with a value of 'adminuser'), 'Path' (value '/'), 'Group' (value '0'), 'Last activity' (value '-'), 'MFA' (value '-'), 'Password age' (value '4 minutes'), and 'Console' (with a blue circular icon). Action buttons for 'Delete' and 'Create user' are located at the top right of the table.

❖ Task 3: VPC Selection & Region Choice



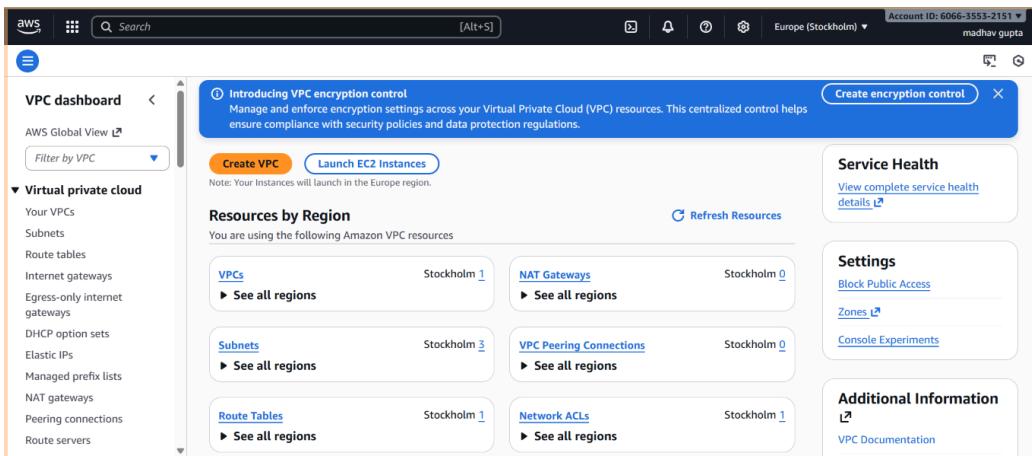
The screenshot shows the AWS VPC Dashboard. The left sidebar is titled "Virtual private cloud" and includes options like "Your VPCs", "Subnets", "Route tables", "Internet gateways", "Egress-only internet gateways", "DHCP option sets", "Elastic IPs", "Managed prefix lists", "NAT gateways", "Peering connections", and "Route servers". The main area is titled "Your VPCs" and shows a table with one row. The table columns are "Name", "VPC ID", "State", "Encryption c...", and "Encryption control ...". The single row has a Name of "-", a VPC ID of "vpc-0df44fed4d088e894", a State of "Available", and other fields empty. A message at the bottom says "Select a VPC above". The top right corner shows account information: "Account ID: 6066-3553-2151", "Region: Europe (Stockholm)", and "User: madhav gupta".

Name	VPC ID	State	Encryption c...	Encryption control ...
-	vpc-0df44fed4d088e894	Available	-	-

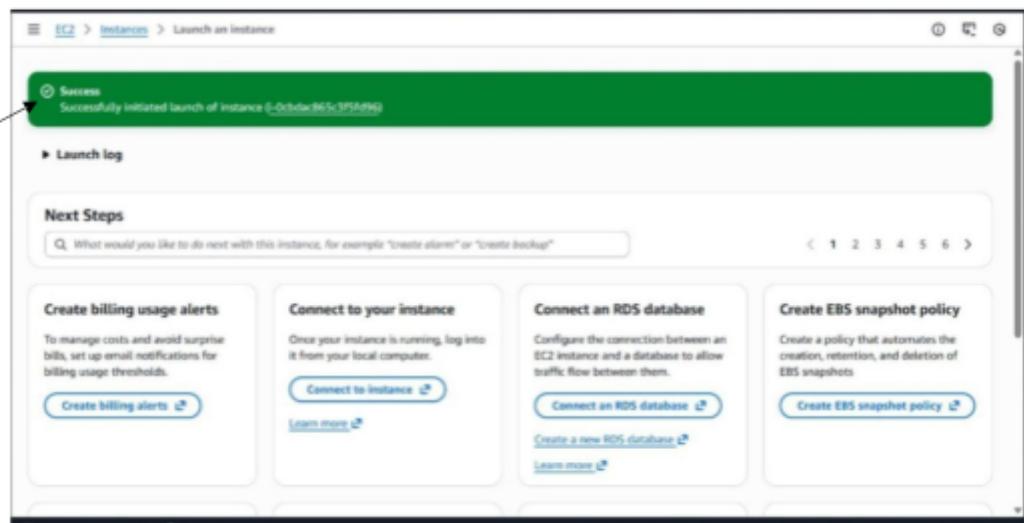
□ Steps:

Steps to Select or Create a VPC

1. In the AWS console, go to the **VPC Dashboard** (search for “VPC”).
2. Click **Your VPCs** in the left sidebar.
3. You’ll see a **Default VPC** (it usually has “Default” in the “Name” column).
4. You can use this VPC directly for EC2, RDS, or other resources — it already has:
 - o Public subnets in each AZ
 - o Internet Gateway attached
 - o Route tables set up



❖ Task 4: Windows Web Server EC2 Instance



□ Steps:

Step 1: Sign in and Select a Region

1. Go to the [AWS Management Console](#).

In the top-right corner, choose your **AWS Region** (e.g., Asia Pacific (Mumbai) ap-south-1).

- o All resources you create will reside in this region.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with 'EC2' selected, followed by 'Dashboard', 'Instances' (with sub-options like Instances, Instance Types, Launch Templates, etc.), and 'Images'. The main area has a 'Resources' section with a table showing counts for various EC2 components. Below it is a 'Launch instance' button. To the right, there's a 'Service health' section with a link to 'AWS Health Dashboard' and a 'Region' dropdown. Further right is an 'EC2 cost' summary with details like 'Date range: Past 6 months', 'Region: Global', 'Costs in your free plan account are covered by credits', 'Credits remaining: \$100 USD', 'Days remaining: 181 (May 21, 2026)', and a note 'Unable to load'. At the bottom right is an 'Account attributes' section with a 'Default VPC' link.

Step 2: Open the EC2 Dashboard

1. In the AWS Console search bar, type “EC2” and select it.
2. Click Instances → Launch instances.

This screenshot shows the 'Launch an instance' wizard. Step 1 is 'Set instance details'. It has sections for 'Name and tags', 'Application and OS Images (Amazon Machine Image)', and 'Quick Start'. In 'Name and tags', the name is set to 'e.g. My Web Server'. In 'Application and OS Images (Amazon Machine Image)', there's a search bar and a list of operating systems including Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. On the right, the 'Summary' section shows 1 instance, the software image (Amazon Linux 2023 AMI 2023.9.2...), virtual server type (t3.micro), firewall (New security group), and storage (1 volume(s) - 8 GiB). At the bottom are 'Cancel', 'Launch instance', and a 'Preview code' link.

Step 3: Configure Basic Details

1. **Name your instance** — e.g., Windows-WebServer.
2. **Choose an Amazon Machine Image (AMI):**
 - Scroll to find **Microsoft Windows Server 2022 Base** (or 2019/2016 if needed).
 - These AMIs come with Windows preinstalled.
3. **Choose Instance Type:**

- For basic setups, select t3.micro or t2.micro (Free Tier eligible).

The screenshot shows two identical instances of the AWS EC2 'Launch an instance' wizard side-by-side. Both instances are set up to launch a single Windows instance.

Left Instance Wizard:

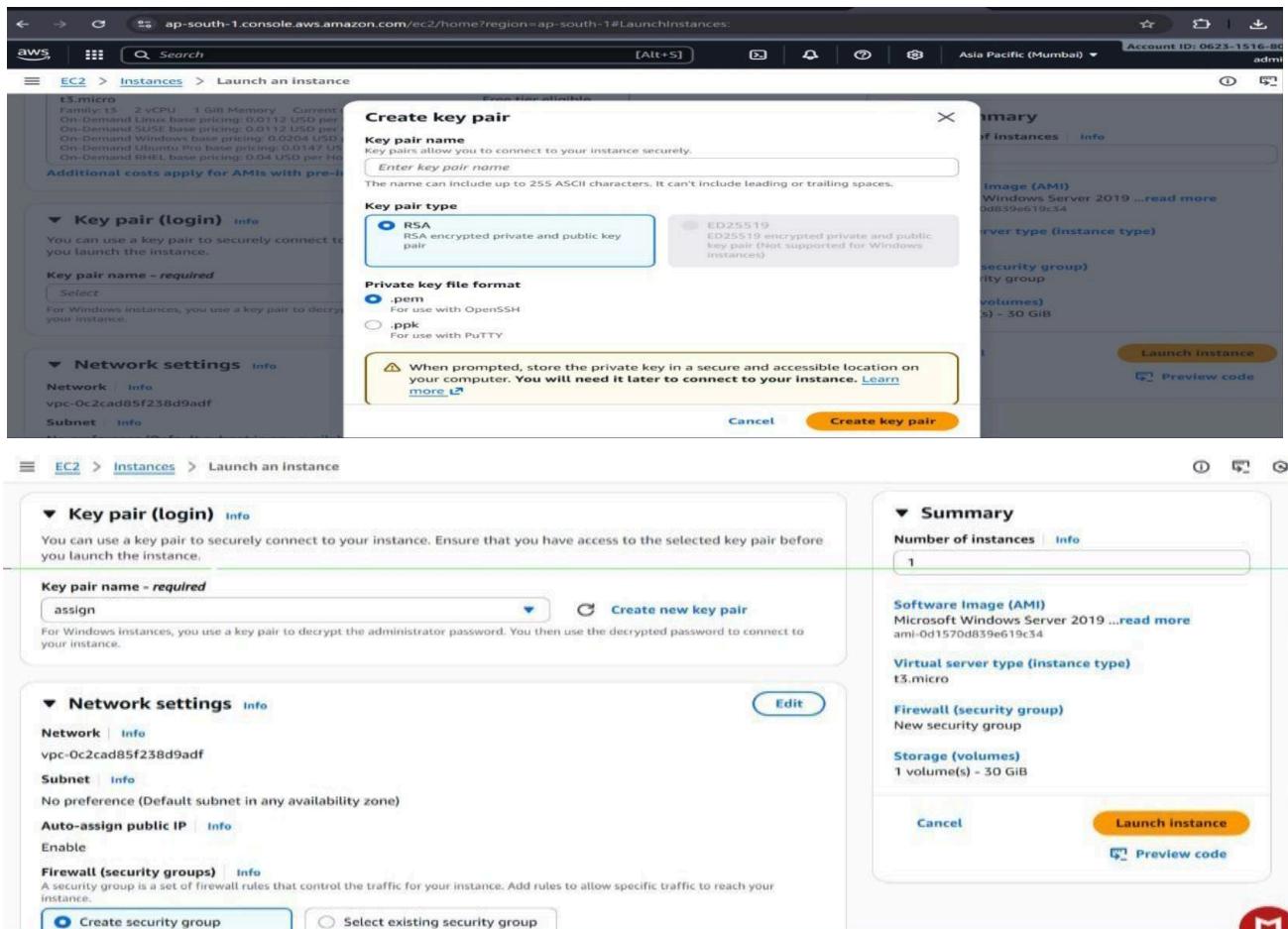
- Name and tags:** Name is set to "Windows-WebServer".
- Application and OS Images (Amazon Machine Image):** A search bar is present, and the "Windows" tab is selected.
- Quick Start:** Shows tabs for Amazon, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Deb.
- Instance type:** t3.micro is selected. It is labeled as "Free tier eligible".
- Key pair (login):** A dropdown menu is open, showing "Select" and "Create new key pair".

Right Instance Wizard:

- Summary:** Number of instances: 1.
- Software Image (AMI):** Microsoft Windows Server 2025.
- Virtual server type (instance type):** t3.micro.
- Firewall (security group):** New security group.
- Storage (volumes):** 1 volume(s) - 30 GiB.
- Buttons:** Cancel, Launch instance, Preview code.

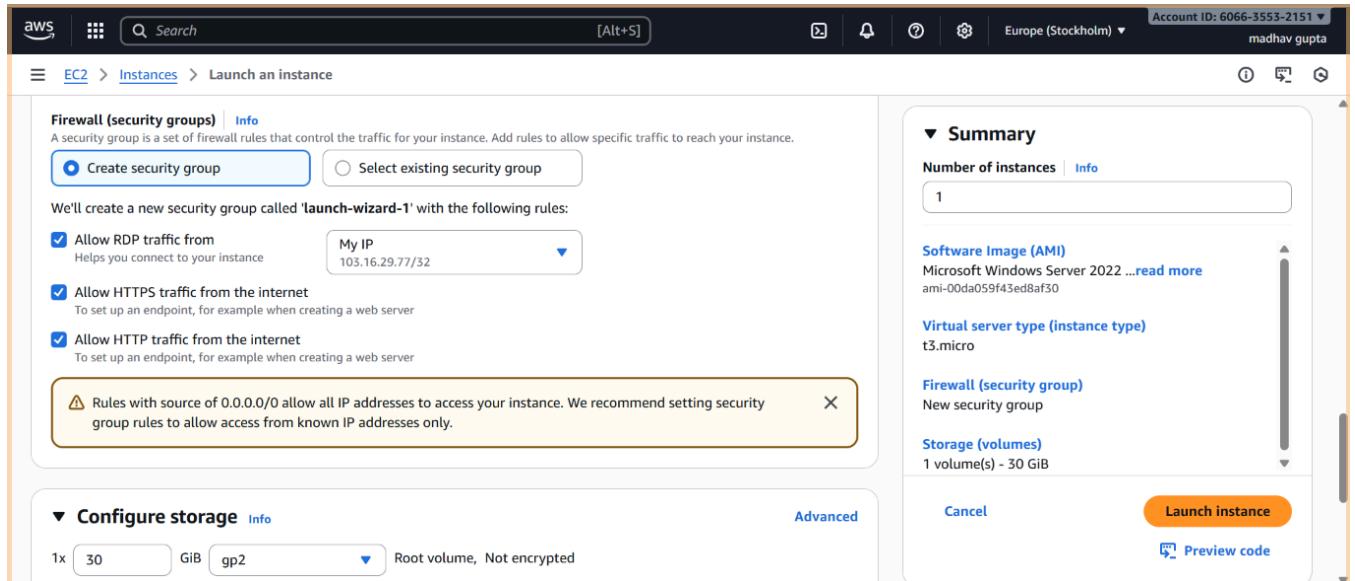
Step 4: Configure Key Pair (Login)

1. Under **Key pair (login)**, choose:
 - **Create new key pair** (if you don't have one yet).
 - Give it a name (e.g., WindowsKeyPair).
 - Choose **Key pair type: RSA**, and **File format: .pem** (for RDP decryption).
 - Click **Create key pair** — it will automatically download the .pem file to your computer.
 - . | Keep it safe — you'll need it to connect later.



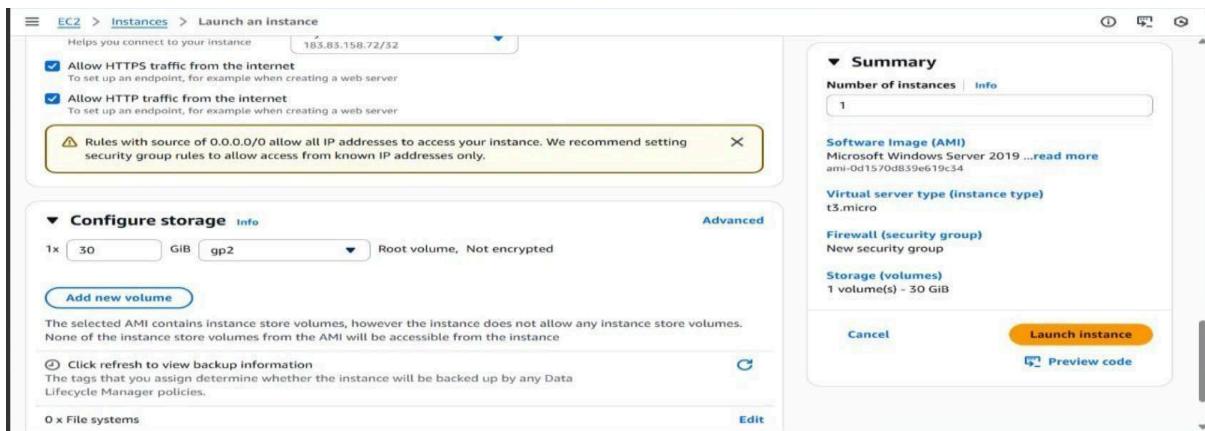
Step 5: Configure Network Settings

1. Under **Network settings**, select your **VPC** and **Subnet** (default VPC is fine).
2. Check **Auto-assign public IP** → *Enable*.
3. Under **Firewall (security group)**:
 - o Choose **Create security group**.
 - o Allow:
 - **RDP (Port 3389)** → Source: My IP (for remote access)
 - **HTTP (Port 80)** → Source: Anywhere (for web traffic)



Step 6: Configure Storage

- Default 30 GB is fine, or increase if needed.
- Ensure volume type is **gp3** (default).



Step 7: Launch Instance

- Review your settings and click **Launch instance**.
- Wait a few moments until the status shows **running**.

Task 5: Systems Manager Fleet Manager Setup

Steps:

Step 1: Create IAM Role for Systems Manager

Fleet Manager requires the instance to have permission to communicate with SSM.

Steps:

1. Go to **IAM → Roles → Create role**.
2. Under **Trusted entity type**, choose **AWS service**.
3. Choose **EC2** → Click **Next**.
4. In **Permissions policies**, search and select:
 - AmazonSSMManagedInstanceCore
5. Click **Next**, name the role:
6. SSM-Managed-Instance-Role
7. Click **Create role**.

The screenshot shows the AWS IAM 'Create role' wizard at Step 3: Name, review, and create. The left sidebar shows the steps: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main area is titled 'Name, review, and create'. It contains 'Role details' with a 'Role name' field set to 'SSMRole' and a 'Description' field containing 'Allows EC2 instances to call AWS services on your behalf.' Below this is a note about character limits. At the bottom, there's a 'Step 1: Select trusted entities' section with a 'Trust policy' tab showing a JSON configuration for EC2 to assume the role.

```
1  "Version": "2012-10-17",
2  "Statement": [
3  {
4  "Effect": "Allow",
5  "Action": [
6  "sts:AssumeRole"
7 ]}
```

Step 2: Attach IAM Role to EC2 Instance

1. Go to EC2 → Instances.
2. Select your instance → Actions → Security → Modify IAM role.
3. Select SSM-Managed-Instance-Role → click Update IAM role.

The screenshot shows the AWS IAM Roles page. A green success message at the top says "Role SSMrole created." Below it, a table lists four roles: AWSServiceRoleForResourceExplorer, AWSServiceRoleForSupport, AWSServiceRoleForTrustedAdvisor, and SSMrole. The SSMrole row shows "AWS Service: ec2" under Trusted entities and "17 minutes ago" under Last activity. On the right, there are sections for "Roles Anywhere" (with options for X.509 Standard and Temporary credentials) and "Access AWS from your non AWS workloads".

The screenshot shows the AWS EC2 Instances page. It displays one instance named "Windows-Web..." with Instance ID "i-05ff2a5f7353d5ada". The instance is listed as "Running" with a status check of "3/3 checks passed". The page includes a search bar, filters for Instance state, type, and status check, and buttons for Connect, Instance state, Actions, and Launch instances. The left sidebar shows navigation links for EC2 (Dashboard, AWS Global View, Events), Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager), Images (AMIs, AMI Catalog), and Elastic Block Store.

Step 3 Verify SSM Agent is Installed

For Amazon Linux 2 or Windows:

- The SSM Agent is preinstalled.

For Other OS (e.g., Ubuntu, CentOS):

1. Connect via SSH or RDP.
2. Run these commands:

The screenshot shows the AWS EC2 Connect to instance page. At the top, it displays the instance ID: i-05ff2a5f7353d5ada (Windows-WebServer). Below this, there are two tabs: 'Session Manager' and 'RDP client', with 'RDP client' being the active tab. Under 'Connection Type', there are two options: 'Connect using RDP client' (selected) and 'Connect using Fleet Manager'. A note indicates that the SSM Agent must be installed and running on the instance for Fleet Manager. Below these options, a link to download the remote desktop file is shown. To the right, there is a 'Username Info' dropdown set to 'Administrator'. At the bottom, there are fields for 'Public DNS' (ec2-13-48-84-95.eu-north-1.compute.amazonaws.com) and 'Password' (with a 'Get password' link).

Step 4: Open Fleet Manager

1. In the AWS Console, go to **Systems Manager**.
2. In the left menu, under **Node Management**, click **Fleet Manager**.
3. You should now see your EC2 instance listed.
 - If it doesn't appear yet, wait a few minutes for SSM to register it.

The screenshot shows the AWS EC2 Connect to instance page for a CentOS instance. The instance ID is i-0cbdac865c3f5fd96 (web server). The 'RDP client' tab is selected. The 'Connection Type' section shows 'Connect using RDP client' (selected) and 'Connect using Fleet Manager'. A note states that joining to a directory allows directory credentials to connect. The 'Username Info' dropdown is set to 'Administrator'. The 'Public DNS' field shows ec2-15-206-77-193.ap-south-1.compute.amazonaws.com. At the bottom, there is a note about joining to a directory and using directory credentials.

The screenshot shows the AWS Systems Manager Fleet Manager interface. At the top, there's a blue header bar with a 'Try out the new AWS Systems Manager unified console' message. Below it, the navigation path is 'Systems Manager > Fleet Manager > Managed nodes'. The main area is titled 'Fleet Manager' with an 'Info' button. Under 'Managed Nodes (1)', there's a table with one row. The table columns include Node ID (i-0cbdac8...), Status (Running), Name (web sever), Platform (Windows), Resource Type (Microsoft ...), Source ID (EC2 insta...), Ping status (Online), and Agent version (3.3.3050.0). There are buttons for 'Filter', 'Report', and 'Node actions'.

Step 5: Manage Instance via Fleet Manager

Now you can manage your EC2 instance **without RDP or SSH**:

From Fleet Manager Dashboard:

- Select your instance → click **Node actions** or instance name. You can now:
- View system info (CPU, memory, network, OS details)
- Browse and edit files
- View event logs
- Manage users
- Run PowerShell/Command Line directly in console
- Access the Windows registry (for Windows)
- Reboot or stop instance remotely

The screenshot shows a modal dialog box titled 'Connect to node'. It says 'You can connect to a maximum of 4 nodes in this view.' Below that, it lists a node: 'web sever i-0cbdac865c3f5fd96'. The 'Authentication type' section has two options: 'User credentials' (selected) and 'Key pair'. The 'Administrator account name' field is set to 'Administrator'. The 'Key pair' section has three options: 'Choose file' (selected), 'Browse your local machine to select the key pair file.', and 'Paste key pair content'. At the bottom, there are links for 'CloudShell', 'Feedback', and 'Console Mobile App', along with copyright information: '© 2025, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

```
PS C:\Users\Administrators> Install-WindowsFeature -name Web-Server
Success Restart Needed Exit Code Feature Result
----- ----- ----- -----
True No NoChangeNeeded {}

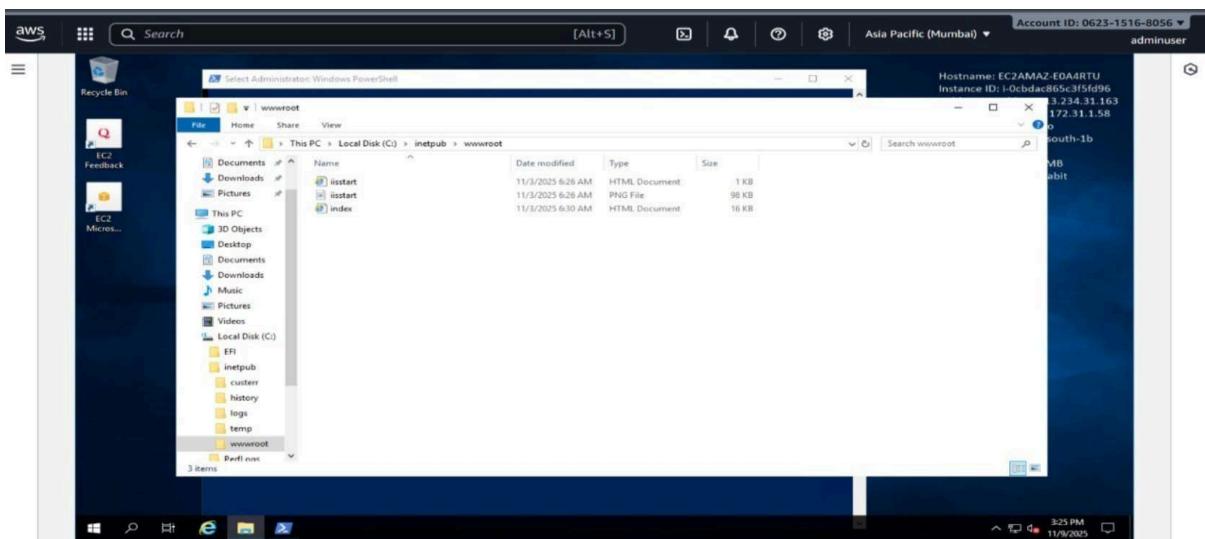
PS C:\Users\Administrators> Install-WindowsFeature -name Web-Common-Http
Success Restart Needed Exit Code Feature Result
----- ----- ----- -----
True No NoChangeNeeded {}

PS C:\Users\Administrators> Install-WindowsFeature -name Web-Http-Errors
Success Restart Needed Exit Code Feature Result
----- ----- ----- -----
True No NoChangeNeeded {}

PS C:\Users\Administrators> Install-WindowsFeature -name Web-Static-Content
Success Restart Needed Exit Code Feature Result
----- ----- ----- -----
True No NoChangeNeeded {}

PS C:\Users\Administrators>
```

Hostname: EC2AMAZ-E0A4RTU
Instance ID: i-0cbdac865c3f5fd96
Public IPv4 address: 13.234.31.163
Private IPv4 address: 172.31.1.58
Instance type: t3.micro
Availability Zone: ap-south-1b
Architecture: AMD64
Total memory: 1024 MB
Network: Up to 5 Gigabit



Final Output:

