

# VPC WITH NAT GATEWAY

The image consists of three vertically stacked screenshots of the AWS Management Console, specifically the EC2 service.

**Screenshot 1: Launch AWS Academy Learner - Launch an instance [EC2] us-east-1:console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances**

This screenshot shows the initial step of launching an instance. It includes fields for Name (MyPublic), Application and OS Images (Amazon Machine Image), and Instance type (t2.micro). A summary box indicates "Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) used when with free tier AMI, 250 hours per month of public IPv4 address usage, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet."

**Screenshot 2: Launch AWS Academy Learner - Launch an instance [EC2] us-east-1:console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances**

This screenshot shows the continuation of the instance configuration, including the selection of an Amazon Machine Image (Ubuntu Server 24.04 LTS (HVM), SSD Volume Type) and the choice of an instance type (t2.micro). It also shows the addition of a key pair (Key pair (login)).

**Screenshot 3: Launch AWS Academy Learner - Launch an instance [EC2] us-east-1:console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances**

This screenshot shows the final steps of the instance launch process, including the review of the instance details and the "Launch instance" button.

# VPC WITH NAT GATEWAY

The screenshots show the AWS EC2 Launch Instances wizard across three steps, illustrating the configuration of a VPC with a NAT gateway.

**Step 1: Key pair (login)**

- Key pair name: `ec2Key`
- Network settings:
  - Network: `vpc-0736500c2f96a5f90`
  - Subnet: `No preference (Default subnet in any availability zone)`
  - Auto-assign public IP: `Enable`
  - Firewall (security groups):
    - `Create security group`
    - `Select existing security group`
  - Allow SSH traffic from: `Anywhere`
  - Allow HTTP traffic from the internet
  - Allow HTTP traffic from the internet
  - Allow SSH traffic from the internet

**Step 2: Network settings**

  - VPC required: `vpc-0d6544e4713c4c257 (JKVPC)`
  - Subnet: `JK_PublicSubnet`
  - Auto-assign public IP: `Enable`
  - Firewall (security groups):
    - `Create security group`
    - `Select existing security group`
  - Description: `launch-wizard-19 created 2023-03-26T05:15:877Z`
  - Inbound Security Group Rules:
    - Security rule 1 (TCP 22, 0.0.0.0/0)
      - Type: `ssh`
      - Protocol: `TCP`
      - Port range: `22`
      - Source type: `All traffic`
      - Description: `SSH access to your instance`

**Step 3: Summary**

  - Number of instances: `1`
  - Software Image (AMI): `Canonical: Ubuntu, 24.04, amd64, read more`
  - Virtual server type (instance type): `t2.micro`
  - Firewall (security group): `New security group`
  - Storage (volumes): `1 volume(s) - 8 GB`

**Free tier information:** In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) used when free tier AWS services are enabled when used with free tier AWS services. 750 hours per month of public IPv4 address usage, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

**Buttons:** Cancel, Launch Instance, Preview code

# VPC WITH NAT GATEWAY

The image consists of three vertically stacked screenshots of the AWS CloudFront console, illustrating the step-by-step creation of a new distribution.

**Screenshot 1: Initial Distribution Creation Step**

This screenshot shows the first step in creating a new CloudFront distribution. It includes fields for:

- Name:** `CloudFront-Demo`
- Origin:** `http://127.0.0.1:8081`
- Default Cache Behavior:** `Cache based on query string`
- Custom Headers:** `Content-Type`, `Accept`
- Compress:** `On`
- Forward Headers:** `None`
- Smooth Streaming:** `Off`
- HTTP 2.0:** `Off`
- Custom Error Responses:** `None`
- Custom Origin Response Headers:** `None`
- Cache Policy:** `CloudFront Default Cache Policy`
- Restrictions:** `None`
- SSL Certificate:** `None`

A large orange **Create Distribution** button is prominently displayed at the bottom right.

**Screenshot 2: Distribution Configuration Step**

This screenshot shows the configuration step for the newly created distribution. It includes:

- Distribution ID:** `CDW2L9JLQH2V2`
- Default Cache Behavior:** `Cache based on query string`
- Custom Headers:** `Content-Type`, `Accept`
- Compress:** `On`
- Forward Headers:** `None`
- Smooth Streaming:** `Off`
- HTTP 2.0:** `Off`
- Custom Error Responses:** `None`
- Custom Origin Response Headers:** `None`
- Cache Policy:** `CloudFront Default Cache Policy`
- Restrictions:** `None`
- SSL Certificate:** `None`

An orange **Next Step** button is at the bottom right.

**Screenshot 3: Final Distribution Configuration Step**

This screenshot shows the final configuration step for the distribution. It includes:

- Distribution ID:** `CDW2L9JLQH2V2`
- Default Cache Behavior:** `Cache based on query string`
- Custom Headers:** `Content-Type`, `Accept`
- Compress:** `On`
- Forward Headers:** `None`
- Smooth Streaming:** `Off`
- HTTP 2.0:** `Off`
- Custom Error Responses:** `None`
- Custom Origin Response Headers:** `None`
- Cache Policy:** `CloudFront Default Cache Policy`
- Restrictions:** `None`
- SSL Certificate:** `None`

An orange **Finish** button is at the bottom right.

# VPC WITH NAT GATEWAY

The screenshots show the 'Launch Instance' wizard in three stages:

- Step 1: VPC - required**: Selects a VPC (vpc-0d64e4713c4c257) and a Subnet (subnet-083bab542b5e413f). It also sets the instance type to t2.micro and the security group to 'Create new security group' (launch-wizard-19).
- Step 2: Firewall security group**: Adds an inbound rule for SSH (TCP port 22) from Anywhere. A note suggests setting security group rules to allow access from known IP addresses.
- Step 3: Summary**: Shows the selected instance type (t2.micro), storage (1 volume(s) - 8 GB), and software image (Canonical Ubuntu, 24.04, ami-0845ab0a838164a4). The 'Launch Instance' button is highlighted.

The screenshot shows the 'Instances' page with two running instances listed:

Name	Instance ID	Instance State	Type	Status Check
MyPrivate	i-0653ab70982baee	Running	t2.micro	Initializing
MyPublic	i-00b51c207d8e2b0ea	Running	t2.micro	Initializing

# VPC WITH NAT GATEWAY

The screenshots illustrate the step-by-step process of launching an EC2 instance with a VPC and NAT gateway:

- Step 1: Launch an instance - Name and tags**
  - Name: bastianServer
  - Software Image (AMI): Canonical, Ubuntu, 24.04, amd64, ami-08456db438326464
  - Virtual server type (instance type): t2.micro
  - Firewall (security group): New security group
  - Storage (volumes): 1 volume(s) - 8 GB
- Step 2: Application and OS Images (Amazon Machine Image)**
  - Description: Ubuntu Server 24.04 LTS (HVM) SSD Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
  - Architecture: 64-bit (x86)
  - AMI ID: ami-08456db438326464
  - Publish Date: 2023-03-05
  - Username: ubuntu (Verified provider)
- Step 3: Instance type**
  - Instance type: t2.micro
  - Processor: 1 CPU, 1 GB Memory
  - Current generation: true
  - On-Demand Windows base pricing: 0.0162 USD per Hour
  - On-Demand Linux base pricing: 0.0161 USD per Hour
  - On-Demand RHEL base pricing: 0.028 USD per Hour
  - On-Demand SUSE base pricing: 0.0162 USD per Hour
- Step 4: Key pair (login)**
  - Key pair name - required: bastian
  - Create new key pair
- Step 5: Network settings**
  - Network: vpc-0736002c2996a3f90
- Step 6: Summary**
  - Number of instances: 1
  - Software Image (AMI): Canonical, Ubuntu, 24.04, amd64, ami-08456db438326464
  - Virtual server type (instance type): t2.micro
  - Firewall (security group): New security group
  - Storage (volumes): 1 volume(s) - 8 GB
- Step 7: Firewall (security group)**
  - Create security group: launch-wizard-19
  - Allow SSH traffic from: Anywhere (0.0.0.0/0)
  - Allow HTTPS traffic from the internet: Anywhere (0.0.0.0/0)
  - Allow HTTP traffic from the internet: Anywhere (0.0.0.0/0)
- Step 8: Configure storage**
  - Advanced

# VPC WITH NAT GATEWAY

The screenshots illustrate the process of launching an EC2 instance with a VPC and NAT gateway.

**Screenshot 1:** The "Launch an instance" wizard step 1. It shows the "Network settings" section where a VPC is selected (vpc-00d044e47134c2f57). A subnet (jk\_PublicSubnet) is chosen, and the "Auto-assign public IP" option is set to "Enable". A security group is being created ("Create security group"). The "Inbound Security Group Rules" section contains one rule: "Security group rule 1 (TCP, 22, 0.0.0.0/0)" allowing SSH (TCP, port 22) from anywhere. The "Summary" section shows 1 instance and includes a "Free tier" callout about account benefits.

**Screenshot 2:** The "Launch an instance" wizard step 2. It shows the "Inbound Security Group Rules" section with three rules: "Security group rule 1 (TCP, 22, 0.0.0.0/0)", "Security group rule 2 (TCP, 443, 0.0.0.0/0)", and "Security group rule 3 (TCP, 80, 0.0.0.0/0)". Each rule allows traffic from anywhere. The "Summary" section shows 1 instance and includes a "Free tier" callout.

**Screenshot 3:** The "Success" page after launching the instance. It displays the instance ID (i-036cbe5736ae195f5) and provides a "Launch log" link. Below, the "Next Steps" section lists various AWS services and their descriptions, such as "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", "Create EBS snapshot policy", "Manage detailed monitoring", "Create Load Balancer", "Create AWS budget", "Manage CloudWatch alarms", "Disaster recovery for your instances", "Monitor for suspicious runtime activities", "Get instance screenshot", and "Get system log".

# VPC WITH NAT GATEWAY

Instances (1/3) **BastianServer**

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary info

Instance ID: i-05dcbe5736ae395f5 (BastianServer)

IPV4 address

Hostname type: IP name: ip-10-0-0-215.ec2.internal

Auto-assigned IP address: 54.166.247.160 [Public IP]

IAM Role: -

IMDv2

Public IPv4 address: 54.166.247.160 [open address]

Private IP address: 10.0.0.215

Public IPv4 DNS: -

Elastic IP addresses: -

AWS Compute Optimizer finding: C

Auto Scaling Group name: -

Managed

Instance ARN: arn:aws:ec2:us-east-1:085044884422:instance/i-05dcbe5736ae395f5

EC2 Instance Connect Session Manager **SSH client** EC2 serial console

Connect to instance info

Connect to your instance i-05dcbe5736ae395f5 (BastianServer) using any of these options

Instance ID: i-05dcbe5736ae395f5 (BastianServer)

1. Open an SSH client.  
2. Locate your private key file. The key used to launch this instance is bastian.pem  
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
    chmod 400 "bastian.pem"  
4. Connect to your instance using its Public IP:  
    54.166.247.160

Example:  
ssh -i "bastian.pem" ubuntu@54.166.247.160

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



EC2 Instance Connect Session Manager **SSH client**

Putty Configuration

Category: Session

Basic options for your PuTTY session

Specify the destination you want to connect to:

Host Name (or IP address): 54.166.247.160

Port: 22

Connection type: SSH

Session: bastian

Load save or delete a stored session

Save Sessions

Default Settings

Close window on exit:

Always (radio button selected)  Never  Only on clean exit

About Help Open Cancel



# VPC WITH NAT GATEWAY

The screenshot displays the AWS Management Console interface for managing EC2 instances. The main window shows a list of three instances:

- MyPublic**: Status: Running, Instance Type: t2.micro, Elastic IP: 34.229.61.27
- MyPrivate**: Status: Running, Instance Type: t2.micro, Elastic IP: 54.166.247.160
- BastianServer**: Status: Initializing

The **MyPrivate** instance is selected, and its detailed view is shown below:

**Security details**

Name	Owner ID	Last updated
sg-089e991c07db95492 (launch-wizard-18)	00823093548	less than a minute ago

**Inbound rules**

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sr-069568f1131205d6	3306	TCP	0.0.0.0/0	launch-wizard-18	-
-	sr-0456302b7b7977008	22	TCP	0.0.0.0/0	launch-wizard-18	-

The screenshot also includes a PuTTY configuration window and a terminal window showing a Linux command line session on the MyPrivate instance.

# VPC WITH NAT GATEWAY

The screenshot shows the AWS EC2 Security Groups console. A new security group named "sg-089c991c07db95492" is being created via the "launch-wizard-18" path. The "Inbound rules" tab is selected, displaying two rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0693683f1112056de	IPv4	MySQL/Aurora	TCP	3306	0.0.0.0/0	-
-	sgr-0456302b7b767708	IPv4	SSH	TCP	22	0.0.0.0/0	-

The screenshot shows the "Edit inbound rules" wizard. The modified inbound rules are:

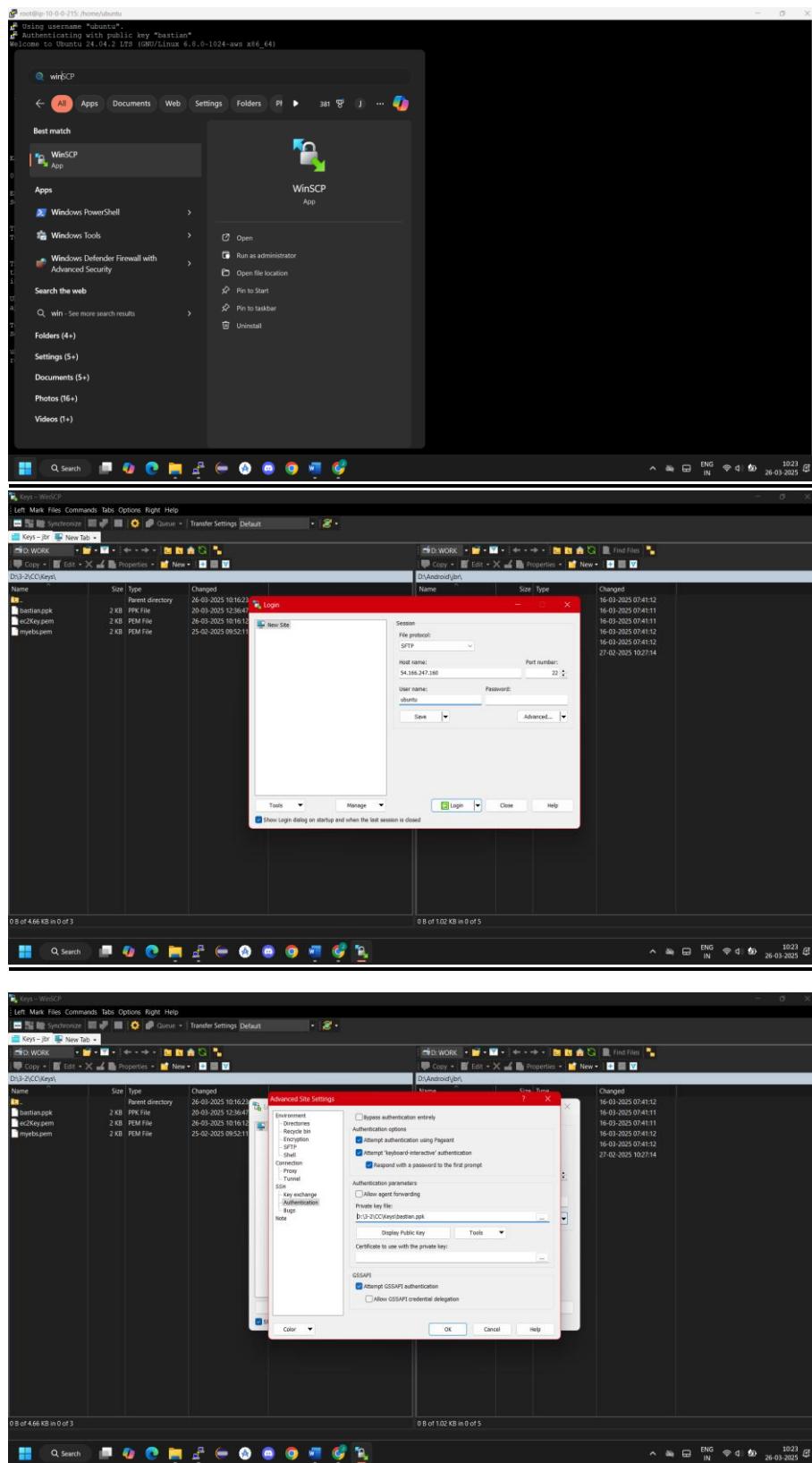
Name	Security group rule ID	Type	Protocol	Port range	Source	Description	
-	sgr-0693683f1112056de	MySQL/Aurora	TCP	3306	Custom	0.0.0.0/0	-
-	sgr-0456302b7b767708	SSH	TCP	22	Custom	10.0.0.215/32	-



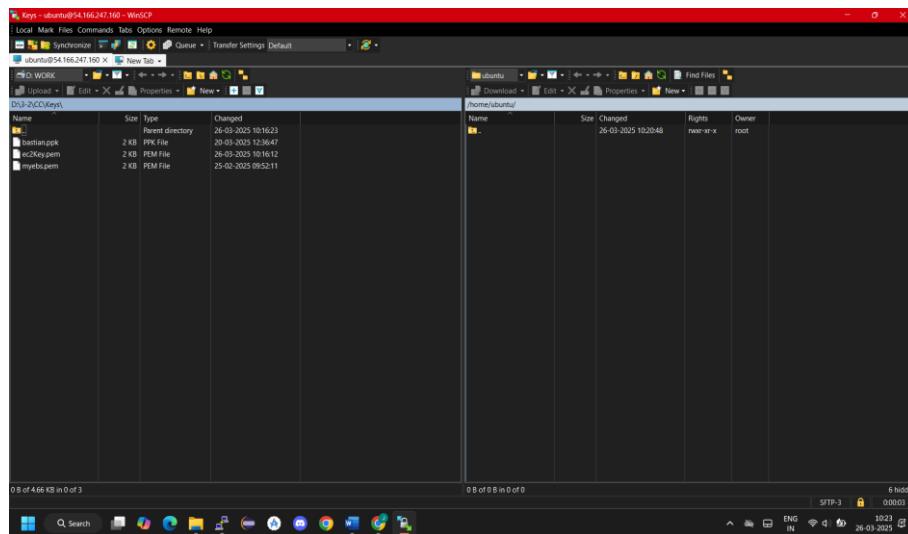
The screenshot shows the AWS EC2 Security Groups console again. The message "Inbound security group rules successfully modified on security group sg-089c991c07db95492 | launch-wizard-18" is displayed. The "Inbound rules" tab is selected, showing the same two rules as before:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0456302b7b767708	IPv4	SSH	TCP	22	10.0.0.215/32	-
-	sgr-0693683f1112056de	IPv4	MySQL/Aurora	TCP	3306	0.0.0.0/0	-

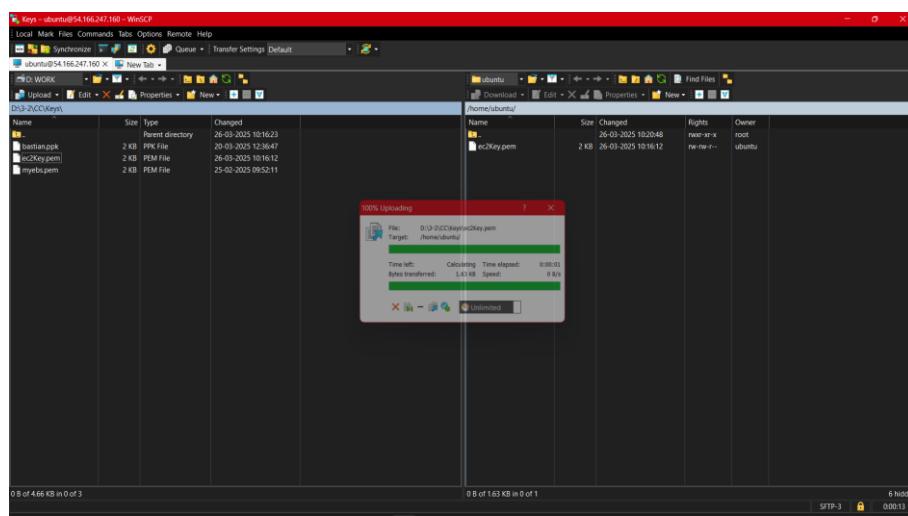
# VPC WITH NAT GATEWAY



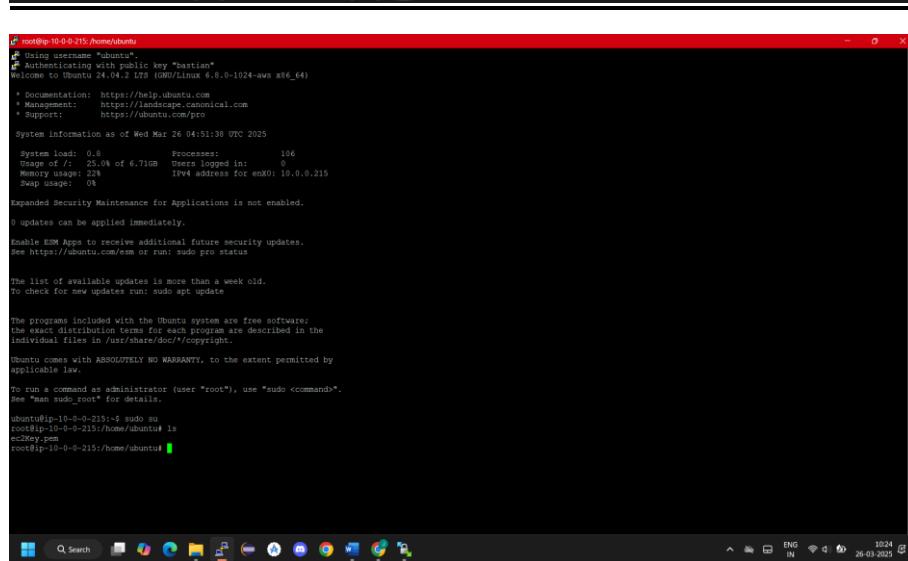
# VPC WITH NAT GATEWAY



The screenshot shows two windows of WinSCP running on a Windows host. Both windows are titled "Keys - ubuntu@54.166.247.160 - WinSCP".  
Left window (Local): Shows a directory structure for "D:\WORK\keysA" containing files: bastianppk (2 KB, PPK File), ec2Key.pem (2 KB, PEM File), and mykey.pem (2 KB, PEM File).  
Right window (Remote): Shows a directory structure for "/home/ubuntu" containing files: Name (2 KB, Changed 26-03-2025 10:20:48) and ec2Key.pem (2 KB, Changed 26-03-2025 10:16:12).  
Bottom status bar: 0.0 of 4.66 KB in 0 of 3, SFTP-3, 00913, ENG IN, 10:23, 26-03-2025.

A "100% Uploading" dialog box is displayed, showing the transfer of "ec2Key.pem" from "D:\WORK\keysA" to "/home/ubuntu". The progress bar is at 100%, with a speed of 0 B/s and a time elapsed of 0:00:01. Other details shown include bytes transferred (1.41 KB) and target path (/home/ubuntu).

The terminal window title is "root@ip-10-0-0-215:~". The session is connected via SSH to an Ubuntu 24.04.2 LTS system.  
Output of "ls /home/ubuntu":

```
root@ip-10-0-0-215:~# ls /home/ubuntu
ec2Key.pem
```

# VPC WITH NAT GATEWAY

The image consists of three vertically stacked screenshots from the AWS Management Console, showing the configuration of a VPC with a NAT gateway.

**Screenshot 1: Connect to instance**  
This screenshot shows the "Connect to instance" dialog for an EC2 instance. It provides instructions for connecting via SSH client, including the instance ID (i-005b4eb7c9820aae) and its private IP (10.0.1.52). It also includes a note about the default AMI username and a command example: `ssh -i "ec2Key.pem" ubuntu@10.0.1.52`.

**Screenshot 2: Ubuntu terminal session**  
A Windows taskbar at the bottom indicates the session is running on a local machine. The terminal window shows a successful SSH connection to the Ubuntu instance. The user runs several commands to verify connectivity and system status, including:

```
ubuntu@ip-10-0-1-52: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-52:~$ sudo su
root@ip-10-0-1-52:~# ssh -i "ec2Key.pem" ubuntu@10.0.1.52
root@ip-10-0-1-52:~# ssh -i "ec2Key.pem" ubuntu@10.0.1.52
The authenticity of host '10.0.1.52 (10.0.1.52)' can't be established.
RSA key fingerprint is SHA256:c1f3W/cqygbSPWfP00cty7X5GhFD0hr1vNsXkn/i.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.52' (ED25593) to the list of known hosts.

Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.6.0-1024-aw9 x86_64)

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

System information as of Wed Mar 26 04:54:49 UTC 2025

  System load: 0.0          Processes:      105
  Usage of /: 25.0% of 6.71GB   Users logged in: 0
  Memory usage: 208          IPv4 address for enx0: 10.0.1.52
  Swap usage: 0B

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/enes 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.

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

ubuntu@ip-10-0-1-52:~$
```

**Screenshot 3: VPC dashboard**  
This screenshot shows the AWS VPC dashboard. It displays various resources such as VPCs, Subnets, Route tables, Internet gateways, and NAT gateways. A prominent feature is the "NAT Gateways" section, which lists two existing NAT gateways. The dashboard also includes sections for Service Health, Settings, Additional Information, and the AWS Network Manager.

# VPC WITH NAT GATEWAY

The screenshots illustrate the process of creating and managing a NAT gateway in the AWS VPC console.

**Screenshot 1: NAT gateways - Create NAT gateway**

This screenshot shows the "Create NAT gateway" wizard. The "Name" field is set to "bastian-gateway". The "Subnet" dropdown is set to "subnet-0b5054884422d7e9 / UK\_PublicSubnet". The "Connectivity type" is set to "Public". The "Elastic IP allocation" dropdown is set to "ephelip-019e4fffb048d856". A tag "Name: bastian-gateway" is added. The "Tags" section shows the tag "Name: bastian-gateway".

**Screenshot 2: NAT gateway - nat-0de3e56dddec7d822d / bastian-gateway**

This screenshot shows the details of the newly created NAT gateway. The "Details" section includes:

- NAT gateway ID:** nat-0de3e56dddec7d822d
- Connectivity type:** Public
- Primary public IPv4 address:** 10.0.0.9
- Subnet:** subnet-0b5054884422d7e9 / UK\_PublicSubnet
- State:** Pending
- Primary private IPv4 address:** 10.0.0.9
- Created:** Wednesday, March 26, 2025 at 10:28:11 GMT+5:30
- Deleted:** -

**Screenshot 3: NAT gateways - Select a NAT gateway**

This screenshot shows the list of NAT gateways. There is one entry: "bastian-gateway" (nat-0de3e56dddec7d822d). The "Actions" dropdown next to it has an option to "Edit".

# VPC WITH NAT GATEWAY

NAT gateway `nat-0d6c56ddc70822d` [bastian-gateway] was created successfully.

Name	NAT gateway ID	Connectivity	State	State message	Primary public IP	Primary private IP	Primary network interface	VPC
<code>bastian-gateway</code>	<code>nat-0d6c56ddc70822d</code>	Public	Pending	-	-	10.0.0.9	<code>eni-05b0263052e45...</code>	<code>vpc-0d64e64715dc2f...</code>

Select a NAT gateway

Name	Route table ID	Explicit subnet association	Edge associations	Main	VPC	Owner ID
<code>rtb-0b1602affa485cb9</code>	-	-	-	Yes	<code>vpc-07768022996a3f90</code>	008283093548
<code>rtb-0679adef79d45680</code>	<code>subnet-0b5034884427de...</code>	-	-	Yes	<code>vpc-0d64e64715dc2f57</code> (VPC)	008283093548

Select a route table

**Create route table**

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.

**VPC**  
The VPC to use for this route table.

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

**Key**  **Value - optional**

You can add up to 50 more tags.

# VPC WITH NAT GATEWAY

The screenshots illustrate the process of creating and configuring a route table in the AWS VPC console.

**Screenshot 1: Route Table Creation**

A screenshot of the AWS VPC console showing the creation of a new route table named "bastian-route". The route table is associated with a specific VPC (vpc-00d4e4f4713c4c2f57) and has a Main entry. A success message indicates that the route was created successfully.

**Screenshot 2: Route Table Configuration**

A screenshot of the AWS VPC console showing the configuration of the "bastian-route" route table. It lists a single route entry: Destination 10.0.0.0/16, Target local, Status Active, and Propagated No. The "Actions" menu is open, showing options like Set main route table, Edit subnet associations, Edit edge associations, Edit propagation, Edit routes, Manage tags, and Delete route table.

**Screenshot 3: Subnet Association**

A screenshot of the AWS VPC console showing the "Edit subnet associations" dialog for the "bastian-route" route table. The "Available subnets" section lists two subnets: JK\_PublicSubnet (subnet-0b0534884422de7e9) and JK\_PrivateSubnet (subnet-0b306a654265e413f). The "Selected subnets" section shows JK\_PrivateSubnet selected. A "Save associations" button is visible at the bottom right.

# VPC WITH NAT GATEWAY

The screenshot shows the AWS VPC console's Route Tables page. A context menu is open over the 'bastian-route' route table, with 'Edit routes' highlighted. The route table details show a single route entry:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	NAT Gateway	-	No
0.0.0.0/0	nat-0de5e56ddc7d822d	-	-

The screenshot shows the 'Edit routes' dialog for the 'bastian-route' route table. A new route entry is being added:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	NAT Gateway	-	No
0.0.0.0/0	nat-0de5e56ddc7d822d	-	-

Buttons at the bottom include 'Add route', 'Cancel', 'Preview', and 'Save changes'.

The screenshot shows the 'Route Table Details' page for the 'bastian-route' route table. The 'Routes' tab displays the following route entries:

Destination	Target	Status	Propagated
0.0.0.0/0	nat-0de5e56ddc7d822d	Active	No
10.0.0.0/16	local	Active	No

# VPC WITH NAT GATEWAY

```
[`ubuntu@ip-10-0-1-52: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-52:~$ sudo su
root@ip-10-0-1-52:~/home/ubuntu ls
etcXkey.pem
root@ip-10-0-1-52:~/home/ubuntu# ssh -l "ec2Key.pem" ubuntu@10.0.1.52
RSA key fingerprint of host '10.0.1.52' (10.0.1.52) can't be established.
RSA25519 key fingerprint is SHA256:c1f3W/cyegpSPW900xtY7kSGhFDshrlvNsXkn/i.
This key is not known by any other names.
RSA25519 key fingerprint matching '(rsa/no)/fingerprint)? yes
Warning: Permanently added '10.0.1.52' (ED255319) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.0.0-1024-aws x86_64)

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

System information as of Wed Mar 26 04:54:49 UTC 2025

  System load: 0.0              Processes:          105
  Usage of /: 25.0% of 6.71GB   Users logged in:    0
  Memory usage: 20M             IPv4 address for eth0: 10.0.1.52
  Swap usage: 0M

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 apt 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.

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

ubuntu@ip-10-0-1-52:~$ sudo apt-get update
[`ubuntu@ip-10-0-1-52: ~
Q Search  Microsoft Edge  File Explorer  This PC  File  Task View  Start  Taskbar  10:32  IN  26-03-2025
```

```
[`ubuntu@ip-10-0-1-52: ~
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [55.0 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [6320 B]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [553 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [103 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [13.5 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [103 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [263 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [164 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [104 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [25.9 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [828 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [168 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [168 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [492 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [55.3 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [103 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [592 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [186 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [376 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7094 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [272 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [16.3 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [15.8 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [164 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Packages [216 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [116 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 kB]
Get:33 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8972 kB]
Get:34 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [7932 kB]
Get:35 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [176 kB]
Get:36 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]
Get:37 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [10300 kB]
Get:38 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [730 kB]
Get:39 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [166 kB]
Get:40 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [214 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [4460 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [3112 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [286 kB]
Fetched 33.0 MB in 8s (4496 kB/s)
ubuntu@ip-10-0-1-52:~$
```