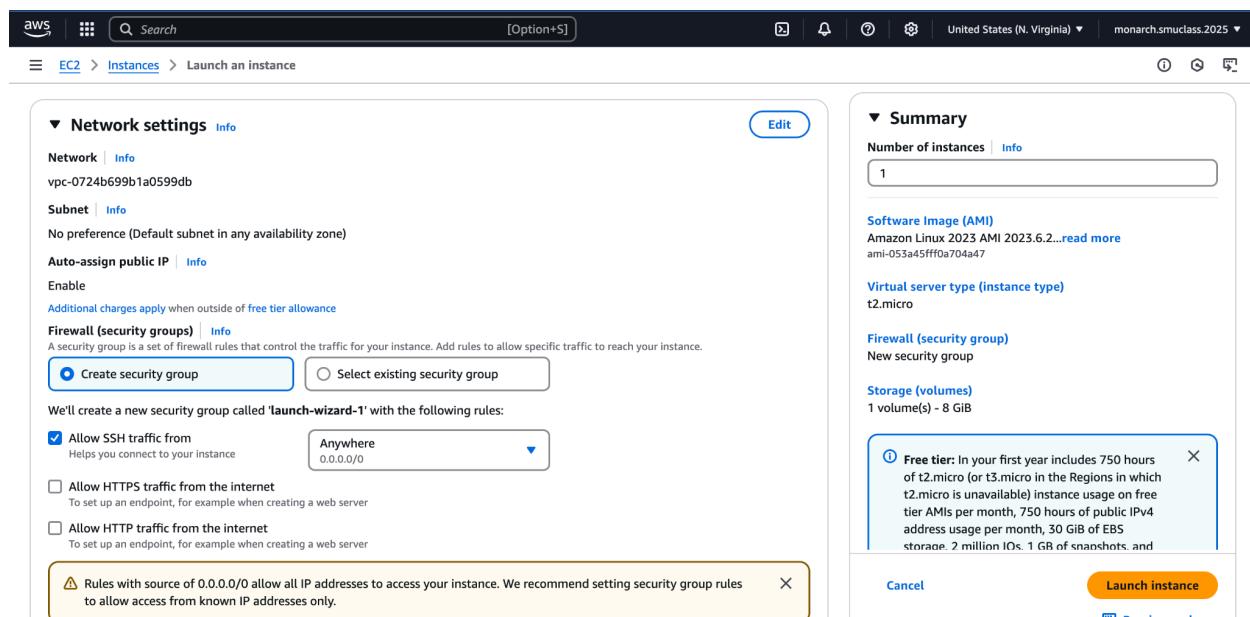


# Lab - 2 AWS | Monarch Nigam

## Exercise 2.1 - Launch & Connect to EC2 Instance

This exercise includes launching an Amazon EC2 instance with a Linux AMI, configuring network and security settings, and ensuring SSH access. After creating or selecting a key pair, the instance is started, and the user connects via SSH using the assigned public IP and key file. All the steps are as follows:

1. From the EC2 Dashboard, click to launch a new instance and select a Linux AMI and instance type. Remember, the t2.micro is Free Tier-eligible if your AWS account is still within its first year.
2. Explore the Network Settings, Configure Storage, and Advanced sections—although the default settings should work fine.
3. In the Network Settings section, make sure there's a rule permitting incoming SSH (port 22) traffic. It should be there by default. You can create your own security group or select an existing group by clicking the Edit button. Default settings will all normally work well here.



4. Before letting you launch the instance, AWS will require you to select—or create—a key pair. Follow the instructions.

The screenshot shows the AWS EC2 Instances Launch wizard. It includes sections for instance type (t2.micro), key pair (MonarchEC2Key), and security group (New security group). The summary section indicates one instance will be launched using the Amazon Linux 2023 AMI.

- Once the instance is launched, you can return to the Instances Dashboard to wait a minute or two until everything is running properly.

The screenshot shows the AWS EC2 Instances dashboard. A single instance named "MonarchEC2Instance" is listed, showing it is currently running. The instance ID is i-0553122c4f38ae76e.

- Click the Actions pull-down menu and select Connect for instructions on how to connect to the instance from your local machine. Note how the SSH username will vary, for instance, between Linux distributions (like Ubuntu or Amazon Linux). Then connect and take a look at your virtual cloud server.

```

Documents - ec2-user@ip-172-31-95-216:~ - ssh -i MonarchEC2Key.pem ec2-user@ec2-54-173-233-226.compute-1.amazonaws.com - 146x44
Last login: Sun Feb 9 18:37:43 on ttys000
monarchnigam@Monarchs-Air ~ % cd Documents/
monarchnigam@Monarchs-Air Documents % cd MonarchEC2Key.pem/
cd: not a directory: MonarchEC2Key.pem/
monarchnigam@Monarchs-Air Documents % chmod 400 "MonarchEC2Key.pem"
monarchnigam@Monarchs-Air Documents % ec2-54-173-233-226.compute-1.amazonaws.com
zsh: command not found: ec2-54-173-233-226.compute-1.amazonaws.com
monarchnigam@Monarchs-Air Documents % ssh -i /Users/monarchnigam/Documents/MonarchEC2Key.pem ec2-54-173-233-226.compute-1.amazonaws.com
The authenticity of host 'ec2-54-173-233-226.compute-1.amazonaws.com (54.173.233.226)' can't be established.
ED25519 key fingerprint is SHA256:DM53noak08UC1NSJK7Ne9c16ttG3rmxAdzxaQvOMSdc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-173-233-226.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
monarchnigam@ec2-54-173-233-226.compute-1.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
monarchnigam@Monarchs-Air Documents % ssh -i "MonarchEC2Key.pem" ec2-user@ec2-54-173-233-226.compute-1.amazonaws.com
'###'          Amazon Linux 2023
~~ \###\
~~ \###\
~~ #/ \-->
~~ \#/
~~ \_ \_ / \
~~ \_ \_ / \
~~ \_ \_ / \
~~ \_ \_ / \
~~ \_ \_ / \
[ec2-user@ip-172-31-95-216 ~]$ 

```

## Exercise 2.2 - Monitor & Change EC2 Instance Type

This exercise will focus on monitoring an EC2 instance's resource usage (CPU, memory, and storage) through the AWS EC2 dashboard. All the steps are as follows:

- With an instance running, open the Instances Dashboard in the EC2 console. Select the instance you're interested in and click the Monitoring tab in the bottom half of the screen. That's a good place to see what percentage of compute and network resources you've been using over the past hours or weeks.

Now pretend that your instance is nearly maxed out and change the instance type as follows.

The screenshot shows the AWS EC2 Instances Dashboard. At the top, there is a search bar and filters for 'Name' and 'Instance ID'. Below the search bar, the instance details are listed: 'MonarchEC2Instance' (i-0553122c4f38ae76e), 'Running', 't2.micro', and '2/2 checks passed'. The 'Monitoring' tab is selected in the navigation bar. Under the 'Monitoring' tab, there are four metrics displayed as line charts: 'CPU utilization (%)', 'Network in (bytes)', 'Network out (bytes)', and 'Network packets in (bytes)'. The CPU utilization chart shows a sharp increase from 1.94% to 3.87% between 22:25 and 23:25. The other three charts show a similar pattern of high values at the end of the day. The bottom of the page includes copyright information and links for privacy, terms, and cookie preferences.

- Stop the instance using the Instance State drop-down menu.

The screenshot shows the AWS EC2 Instances Dashboard after stopping the instance. A green notification bar at the top left says 'Successfully initiated stopping of i-0553122c4f38ae76e'. The instance details now show 'Stopped' instead of 'Running'. The 'Monitoring' tab is still selected, and the same four metrics are shown as line charts. The CPU utilization chart shows a sharp drop from 3.87% to 0% between 22:25 and 23:25. The other three charts show a similar pattern of high values at the end of the day. The bottom of the page includes copyright information and links for privacy, terms, and cookie preferences.

- From the Actions drop-down menu, click Instance Settings and then Change Instance Type (which might only appear after a minute or so or once you refresh the page). Select a new type (although remember that choosing a non-Free Tier instance type might incur costs).

**Change instance type** [Info](#) | [Get advice](#)

You can change the instance type only if the current instance type and the instance type that you want are compatible.

**Instance ID**  
i-0553122c4f38ae76e (MonarchEC2Instance)

**Current instance type**  
t2.micro

**New instance type**  
 [X](#)

EBS-optimized  
EBS-optimized is not supported for this instance type

- Restart the instance and confirm that it's running properly.

**Instance summary for i-0553122c4f38ae76e (MonarchEC2Instance)** [Info](#)

Updated less than a minute ago

<b>Instance ID</b> i-0553122c4f38ae76e	<b>Public IPv4 address</b> –	<b>Private IPv4 addresses</b> 172.31.95.216
<b>IPv6 address</b> –	<b>Instance state</b> Stopped	<b>Public IPv4 DNS</b> –
<b>Hostname type</b> IP name: ip-172-31-95-216.ec2.internal	<b>Private IP DNS name (IPv4 only)</b> ip-172-31-95-216.ec2.internal	<b>Elastic IP addresses</b> –
<b>Answer private resource DNS name</b> IPv4 (A)	<b>Instance type</b> t2.micro	<b>AWS Compute Optimizer finding</b> <a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a>
<b>Auto-assigned IP address</b> –	<b>VPC ID</b> vpc-0724b699b1a0599db	<b>Auto Scaling Group name</b> –
<b>IAM Role</b> –	<b>Subnet ID</b> subnet-08b3a700e1ecece48	<b>Managed</b> false
<b>IMDSv2</b> Required	<b>Instance ARN</b> arn:aws:ec2:us-east-1:160885292183:instance/i-0553122c4f38ae76e	

The Below Snapshot shows the instance is running fine

```

Documents — ec2-user@ip-172-31-95-216:~ — ssh -i MonarchEC2Key.pem ec2-user@ec2-3-95-173-12.compute-1.amazonaws.com — 146x44
monarchnigam@Monarchs-Air Documents % ssh -i "MonarchEC2Key.pem" ec2-user@ec2-3-95-173-12.compute-1.amazonaws.com
'___. _###_          Amazon Linux 2023
~~ \###\_
~~ \###|
~~ \#/ ____ https://aws.amazon.com/linux/amazon-linux-2023
~~ \~'-->
~~ .-. /_
~~ /_/
~/m/'

Last login: Sat Feb 15 23:22:25 2025 from 35.146.115.156
[ec2-user@ip-172-31-95-216 ~]$ uname -a
Linux ip-172-31-95-216.ec2.internal 6.1.127-135.201.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Jan 28 23:19:58 UTC 2025 x86_64 x86_64 x86_64 GNU/L
inux
[ec2-user@ip-172-31-95-216 ~]$ free -h
              total        used        free      shared  buff/cache   available
Mem:       949Mi       125Mi      628Mi       0.0Ki     196Mi      687Mi
Swap:        0B         0B         0B

[ec2-user@ip-172-31-95-216 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M    0  4.0M  0% /dev
tmpfs          475M    0  475M  0% /dev/shm
tmpfs          190M  444K  190M  1% /run
/dev/xvda1      8.0G  1.6G  6.4G  20% /
tmpfs          475M    0  475M  0% /tmp
/dev/xvda128    10M   1.3M   8.7M  13% /boot/efi
tmpfs          95M    0   95M  0% /run/user/1000
[ec2-user@ip-172-31-95-216 ~]$ 

```

### Exercise 2.3 - Choosing a pricing model

To compute the optimal pricing model for your two always-on f1.2xlarge instances and four other instances for 100 hours per year -

**On-Demand Instances:** Best suited for short-term, unpredictable workloads. But two always-on instances with On-Demand pricing will be expensive.

**Reserved Instances (RI):** Most suitable for long-term, constant-state workloads. Since the two always-on instances will be used on a yearly basis, the acquisition of a 1-year or 3-year Reserved Instance would offer cost savings compared to On-Demand pricing.

**Spot Instances:** They are cheaper but not appropriate for always-running loads because AWS may terminate them if they are needed elsewhere. They can be useful for the additional four instances required for 100 hours of a year.

We can go to the Aws pricing calculator and determine the estimated price as shown in the below Screenshots - <https://calculator.aws/#/>

**aws pricing calculator**

Feedback Language: English ▾ Contact Sales ▾ Create an AWS Account

🔍 f1.2xlarge

Instance family [Info](#) vCPUs Memory (GiB) Network performance

Any Instance family Any vCPUs Any Memory (GiB) Any Network Performance

Show only current generation instances.

Instance name	vCPUs	Memory	Network Performance	Storage	On-Demand Hourly Cost	Curren
<input type="radio"/> inf1.2xlarge	8	16 GiB	Up to 25 Gigabit	EBS only	0.362	Yes
<input checked="" type="radio"/> f1.2xlarge	8	122 GiB	Up to 10 Gigabit	1 x 470 NVMe SSD	1.65	Yes

**Payment options**

Estimated commitment price based on the following selections:  
Instance type: **f1.2xlarge** Operating system: **Linux**

**aws pricing calculator**

Feedback Language: English ▾ Contact Sales ▾ Create an AWS Account

<input checked="" type="radio"/> f1.2xlarge	8	122 GiB	Up to 10 Gigabit	1 x 470 NVMe SSD	1.65	Yes	0.7167 (57%)
---	---	---------	------------------	------------------	------	-----	--------------

**Payment options**

Estimated commitment price based on the following selections:  
Instance type: **f1.2xlarge** Operating system: **Linux**

Select the container and options to find your best price

Compute Savings Plans  
One plan that automatically applies to all usage on EC2, Fargate, and Lambda. Up to 66% discount.  
[Learn more](#)

Reservation term  
 1 year  
 3 year

Payment Options  
 No upfront  
 Partial upfront  
 All upfront

EC2 Instance Savings Plans  
Get deeper discount when you only need one instance family and region. Up to 72% discount.  
[Learn more](#)

Reservation term  
 1 year  
 3 year

Payment Options  
 No upfront  
 Partial upfront  
 All upfront

On-Demand  
Maximize flexibility. [Learn more](#)

Expected utilization  
Enter the expected usage of Amazon EC2 instances

Usage  
100

Usage type  
Utilization percent per month

Note that for usage less than 100%, the calculator assumes that the attached EBS volume is deleted while the instance is not running. Keep in mind that volumes are billed as long as they are provisioned in your account, even while attached to a stopped instance.

Spot Instances  
Minimize cost by leveraging EC2's spare capacity. Recommended for fault tolerant and interruption tolerant applications. [Learn more](#)

The historical average discount for f1.2xlarge is 68%

Assume percentage discount for my estimate  
-1

Actual spot instance pricing varies  
With spot instances, you pay the spot price that's in effect for the time period your instance is running

Upfront: 0.00 Monthly: 936.59/Month

Upfront: 0.00 Monthly: 601.16/Month

Instance: 1.65/Hour Monthly: 1204.50/Month

Instance: 1.65/Hour Monthly: 1216.55/Month

## Exercise 2.4 : Create and Launch an AMI Based on an Existing Instance Storage Volume

1. If necessary, launch an instance (following the instructions displayed when you click the Connect menu item at the top) and make at least some token change to the root volume. This could be something as simple as typing `touch test.txt` on a Linux instance to create an empty file.

```
[ec2-user@ip-172-31-95-216 ~]$ touch test.txt
[ec2-user@ip-172-31-95-216 ~]$ ls -l
total 0
-rw-r--r--. 1 ec2-user ec2-user 0 Feb 16 01:10 test.txt
[ec2-user@ip-172-31-95-216 ~]$
```

## 2. Create an image from the instance's volume

The screenshot shows the AWS EC2 AMIs page. On the left, there's a sidebar with 'Instances' selected. The main area displays a table for 'Amazon Machine Images (AMIs)'. One row is shown, representing the newly created AMI:

Name	AMI ID	Source	Owner	Vis
NewImageMonarch	ami-09395179ba7b0108e	160885292183/NewImageMonarch	160885292183	Priv

## 3. Launch an instance from the console and select the new AMI from the My AMIs tab.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'Instances' selected. A green notification bar at the top says: 'Currently creating AMI ami-09395179ba7b0108e from instance i-0553122c4f38ae76e. Check that the AMI status is "Available" before deleting the instance or carrying out other actions related to this AMI.' The main area shows a table for 'Instances (1/1)'. One row is listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
MonarchEC2Instance	i-0553122c4f38ae76e	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c

**Instance summary for i-0553122c4f38ae76e (MonarchEC2Instance)**

Instance ID	i-0553122c4f38ae76e	Public IPv4 address	39.5.173.12   open address
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-172-31-95-216.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-95-216.ec2.internal
Answer private resource DNS name	IPv4 (A)	Instance type	t2.micro
Auto-assigned IP address	39.5.173.12 [Public IP]	VPC ID	vpc-0724b699b1a0599db
IAM Role	-	Subnet ID	subnet-08b3a700e1ecece48
IMDSv2	Required	Instance ARN	arn:aws:ec2:us-east-1:160885292183:instance/i-0553122c4f38ae76e
Operator	-	AWS Compute Optimizer finding	<small>Opt-in to AWS Compute Optimizer for recommendations.</small>
<a href="#">Details</a>   Status and alarms   Monitoring   Security   Networking   Storage   Tags			

**4. Log into the instance and confirm that your previous change has persisted. As we can see from the below snapshot, the test.txt file is still present, which means that the previous change has clearly persisted**

```
[ec2-user@ip-172-31-95-216 ~]$ touch test.txt
[ec2-user@ip-172-31-95-216 ~]$ ls -l
total 0
-rw-r--r--. 1 ec2-user ec2-user 0 Feb 16 01:10 test.txt
[ec2-user@ip-172-31-95-216 ~]$
Broadcast message from root@ip-172-31-95-216.ec2.internal (Sun Feb 16 01:16:18 UTC):
The system will power off now!

Connection to ec2-3-95-173-12.compute-1.amazonaws.com closed by remote host.
Connection to ec2-3-95-173-12.compute-1.amazonaws.com closed.
monarchnigam@Monarchs-Air Documents % ssh -i "MonarchEC2Key.pem" ec2-user@ec2-3-95-173-12.compute-1.amazonaws.com
,
~\_\#\#\#_      Amazon Linux 2023
~~ \_\#\#\#\| 
~~ \#\#\| 
~~ \#/ -- https://aws.amazon.com/linux/amazon-linux-2023
~~   V~' '=>
~~/
~~_.-' /'
~~/_/-' /'
~/m/'

Last login: Sat Feb 15 23:41:26 2025 from 35.146.115.156
[ec2-user@ip-172-31-95-216 ~]$ ls -l
total 0
-rw-r--r--. 1 ec2-user ec2-user 0 Feb 16 01:10 test.txt
[ec2-user@ip-172-31-95-216 ~]$
```

## EXERCISE 2.5 : Creating a Launch Template

1. In the EC2 Dashboard, click Launch Templates.
2. Click the Create Launch Template button.
3. Give the launch template a name such as MyTemplate.
4. Click inside the search field for an image and select the AMI of one of the Ubuntu Server LTS AMIs (make sure the AMI you choose uses the 64-bit x86 architecture and not 64-bit ARM).
5. For Instance Type, select t2.micro.

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with the AWS logo, a search bar, and various icons. Below the navigation bar, the main content area has a heading 'Search results' and a sub-section 'Application and OS Images (Amazon Machine Image)'. A search bar here also contains the text 'Search our full catalog including 1000s of application and OS images'. On the left, there's a sidebar with sections for 'AMI from catalog', 'Recents', and 'Quick Start'. The main panel displays details for an AMI: Name (Ubuntu Server 24.04 LTS (HVM), SSD Volume Type), Description (Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>)), Image ID (ami-04baf1a9cf54c11d0), Username (ubuntu), Catalog (Quick Start AMIs), Published (2025-01-15T09:17:20.000Z), Architecture (x86\_64), Virtualization (hvm), Root device type (ebs), and ENA Enabled (Yes). To the right, there's a 'Summary' section with tabs for 'Software Image (AMI)', 'Virtual server type (instance type)' (set to t2.micro), 'Firewall (security group)' (empty), and 'Storage (volumes)' (1 volume(s) - 8 GB). A callout box highlights the 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom right are 'Cancel' and 'Create launch template' buttons.

6. Under Network Settings, select a security group that allows inbound HTTP access. Create a new security group if necessary.

The below snapshots depict the required network settings while launching the template and the security group that contains all the desired inbound network rules -

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-012468cd9910cabf7	IPv4	SSH	TCP	22	0.0.0.0/0	
sgr-0de83bbbe963fc1d3	IPv4	HTTP	TCP	80	0.0.0.0/0	

## 7. Expand the Advanced Details section and scroll down to the User Data field -

In the below screenshot the User data has been filled with the linux code -

**Metadata response hop limit** | [Info](#)  
2

**Allow tags in metadata** | [Info](#)  
Don't include in launch template

**User data - optional** | [Info](#)  
Upload a file with your user data or enter it in the field.  
[Choose file](#)

```
#!/bin/bash
apt-get update
apt-get install -y apache2
echo "Welcome to my website" > index.html
cp index.html /var/www/html
```

User data has already been base64 encoded

**Summary**

**Software Image (AMI)**  
Ubuntu Server 24.04 LTS (HVM)...[read more](#)  
ami-0404f1fa9cf5c4c11a0

**Virtual server type (instance type)**  
t2.micro

**Firewall (security group)**  
New security group

**Storage (volumes)**  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Create launch template](#)

## 8. Click the Create Launch Template button.

**Success**  
Successfully created MonarchTemplate(lt-0f6e87b92379a818b).

**Actions log**

**Next Steps**

[Launch an instance](#)  
With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or upfront payments. Launch an On-Demand Instance from your launch template.

[Launch instance from this template](#)

[Create an Auto Scaling group from your template](#)  
Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

[Create Auto Scaling group](#)

[Create Spot Fleet](#)  
A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.

[Create Spot Fleet](#) [View launch templates](#)

## 9. Click the Launch Instance From This Template link.

10. Under Source Template Version, select 1 (Default).

11. Click the Launch Instance From Template button.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, and Dedicated Hosts. The main content area has a heading 'Instances (1/1) Info'. It shows a table with one row for 'MonarchInstance' (i-02ca606024ef0315c). The instance is listed as 'Running' with an 'Initializing' status check. It's in the 'us-east-1c' availability zone and has a public IPv4 address of 'ec2-3-84-220-136'. There are buttons for 'Connect', 'Actions', and 'Launch instances'.

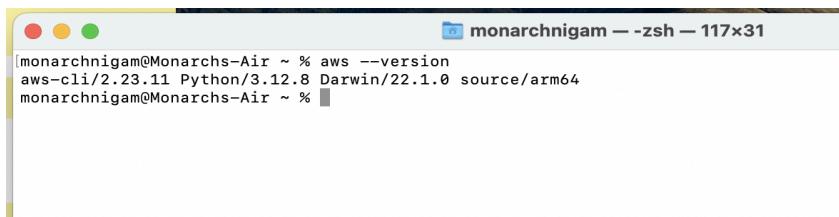
12. After the instance boots, browse to its public IP address. You should see a web page that says "Welcome to my website." The below snapshot clearly depicts the successful output of my running EC2 instance-



13. Terminate the instance when you're done with it.

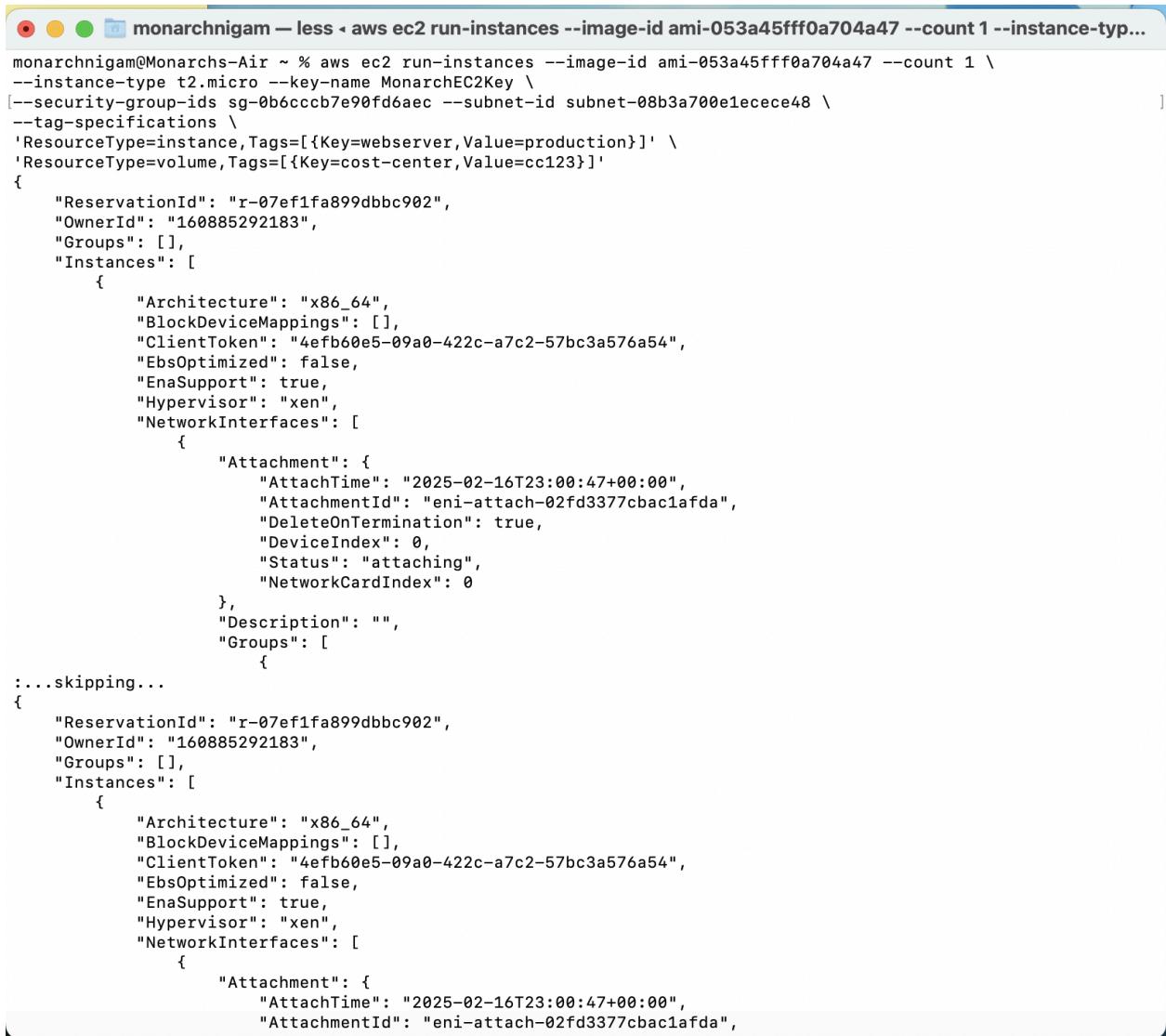
The screenshot shows the AWS EC2 Instances page again. A green success message at the top says 'Successfully initiated termination (deletion) of i-02ca606024ef0315c'. The main content area shows the same instance table as before, but now the instance state is 'Shutting-down'. The status check shows '2/2 checks passed'. At the bottom, there's a details panel for 'i-02ca606024ef0315c (MonarchInstance)' with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

## EXERCISE 2.6 - Install the AWS CLI and Use It to Launch an EC2 Instance - I installed and AWS CLI -



```
[monarchnigam@Monarchs-Air ~ % aws --version
aws-cli/2.23.11 Python/3.12.8 Darwin/22.1.0 source/arm64
monarchnigam@Monarchs-Air ~ % ]
```

Then I launched the EC2 instance from here as shown in the below screenshots



```
monarchnigam — less < aws ec2 run-instances --image-id ami-053a45fff0a704a47 --count 1 --instance-type t2.micro --key-name MonarchEC2Key \
--security-group-ids sg-0b6cccb7e90fd6aec --subnet-id subnet-08b3a700e1ecece48 \
--tag-specifications \
'ResourceType=instance,Tags=[{Key=webserver,Value=production}]' \
'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
{
    "ReservationId": "r-07ef1fa899dbbc902",
    "OwnerId": "160885292183",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "4efb60e5-09a0-422c-a7c2-57bc3a576a54",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-02-16T23:00:47+00:00",
                        "AttachmentId": "eni-attach-02fd3377cbac1afda",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        ...
                    ]
                }
            ],
            "...skipping...
            {
                "ReservationId": "r-07ef1fa899dbbc902",
                "OwnerId": "160885292183",
                "Groups": [],
                "Instances": [
                    {
                        "Architecture": "x86_64",
                        "BlockDeviceMappings": [],
                        "ClientToken": "4efb60e5-09a0-422c-a7c2-57bc3a576a54",
                        "EbsOptimized": false,
                        "EnaSupport": true,
                        "Hypervisor": "xen",
                        "NetworkInterfaces": [
                            {
                                "Attachment": {
                                    "AttachTime": "2025-02-16T23:00:47+00:00",
                                    "AttachmentId": "eni-attach-02fd3377cbac1afda",
                                }
                            }
                        ]
                    }
                ]
            }
        }
    ]
}
```

On checking the console we can see there is the EC2 instance we created from CLI -

The screenshot shows the AWS EC2 Instances page with a single instance listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
	i-0af30121dc6f6ae5b	Running	t2.micro	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1c	ec2-3-83-1

**Please note** - I forgot to capture the full screenshot of the created instance, so below I am attaching the screenshot of the same instance but it has been terminated

The screenshot shows the AWS EC2 Instance summary page for the instance i-0af30121dc6f6ae5b, which is now terminated.

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0af30121dc6f6ae5b	-	-

Details tab (selected):

- AMI ID:** ami-053a45ff0a704a47
- AMI name:** al2023-ami-2023.6.20250211.0-kernel-6.1-x86\_64
- Monitoring:** disabled
- Allowed image:** -
- Platform details:** Linux/UNIX
- Termination protection:** Disabled

## EXERCISE 2.7 Clean Up Unused EC2 Resources

The Below Screenshots shows that I have no Bills to pay as I deleted each and every resource after I was done using them.

The screenshot shows the AWS EC2 Instances page with two terminated instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
	i-0af30121dc6f6ae5b	Terminated	t2.micro	-	<a href="#">View alarms +</a>	us-east-1c	-
	i-01c68dd6cdee4cfef	Terminated	t2.micro	-	<a href="#">View alarms +</a>	us-east-1c	-

Screenshot of the AWS Billing and Cost Management home page.

**Billing and Cost Management**

**Choose billing view [New](#)**

Primary view ▾

**Billing and Cost Management home** [Info](#)

**Cost summary** [Info](#)

Month-to-date cost **\$0.00**  
- compared to last month for same period

Total forecasted cost for current month  
**Data unavailable**

Last month's cost for same time period  
**\$0.00**  
Jan 1 – 16

Last month's total cost  
**\$0.00**

**Cost monitor** [Info](#)

Budgets status  
**Setup required**  
No budget created

Cost anomalies status (MTD)  
**None detected**  
1 monitor(s) active

Screenshot of the AWS Billing and Cost Management Bills page.

**Billing and Cost Management**

**Choose billing view [New](#)**

Primary view ▾

**Bills** [Info](#)

Page refresh time: Sunday, February 16, 2025 at 5:10:09 PM CST

**AWS estimated bill summary** [Info](#)

Total charges and payment information

Account ID <b>160885292183</b>	Billing period <a href="#">Info</a> <b>February 1 - February 28, 2025</b>	Bill status <a href="#">Info</a> <b>Pending</b>
Service provider <b>Amazon Web Services, Inc.</b>	Total in USD <b>USD 0.00</b>	
<b>Estimated grand total:</b> <b>USD 0.00</b>		

**Payment information** [Info](#)

Viewing Amazon Web Services, Inc.

Highest service spend <b>USD 0.00</b>	Trend compared to prior month No data to display.	Highest AWS Region spend <b>USD 0.00</b>	Trend compared to prior month No data to display.
Service name CloudWatch	Region name US East (N. Virginia)		

**Charges by service** **Charges by account** **Invoices** **Savings** **Taxes by service**

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)