

AWS EC2 MULTI-AZ DEPLOYMENT LAB

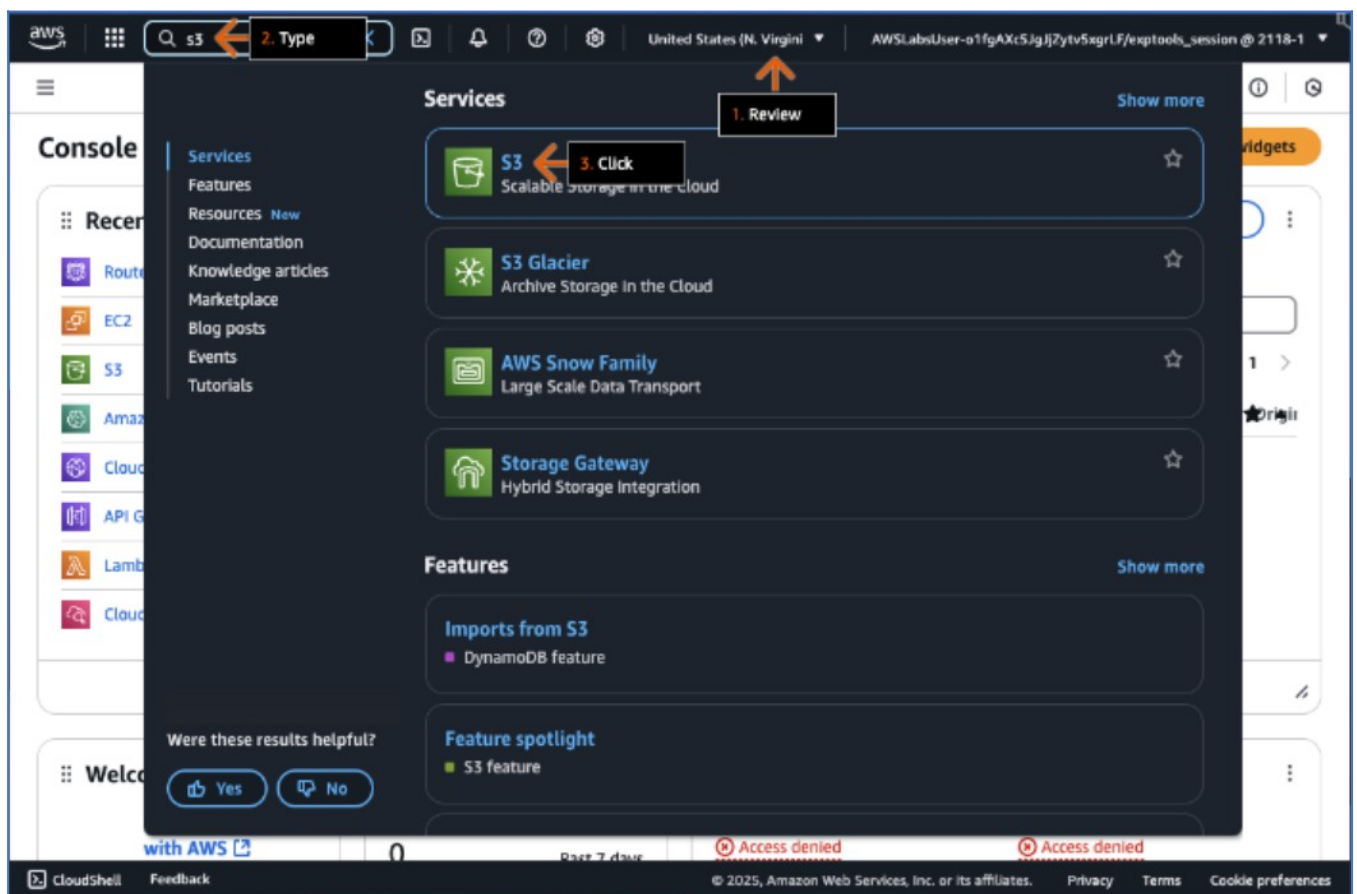
Objectives:

- Launch an Amazon EC2 instance.
- Configure a user data script to display the instance details in a browser.
- Launch a second EC2 instance in a different Availability Zone of the same AWS Region.

Steps / Procedures / Instructions:

In this lab, you will:

- Launch an Amazon EC2 instance.
- Configure a user data script to display the instance details in a browser.
- On the top navigation bar, review the Region selector to confirm that the Region is set to N. Virginia (us-east-1).
- In the Services search box, type: s3
- In the search results, under Services, click S3.



AWS has the concept of a Region, which is a physical location around the world where we cluster data centers. We call each group of logical data centers an Availability Zone.

Each AWS Region consists of a minimum of three, isolated, and physically separate AZs within a geographic area.

Amazon Simple Storage Service (Amazon S3) stores data as objects within buckets.

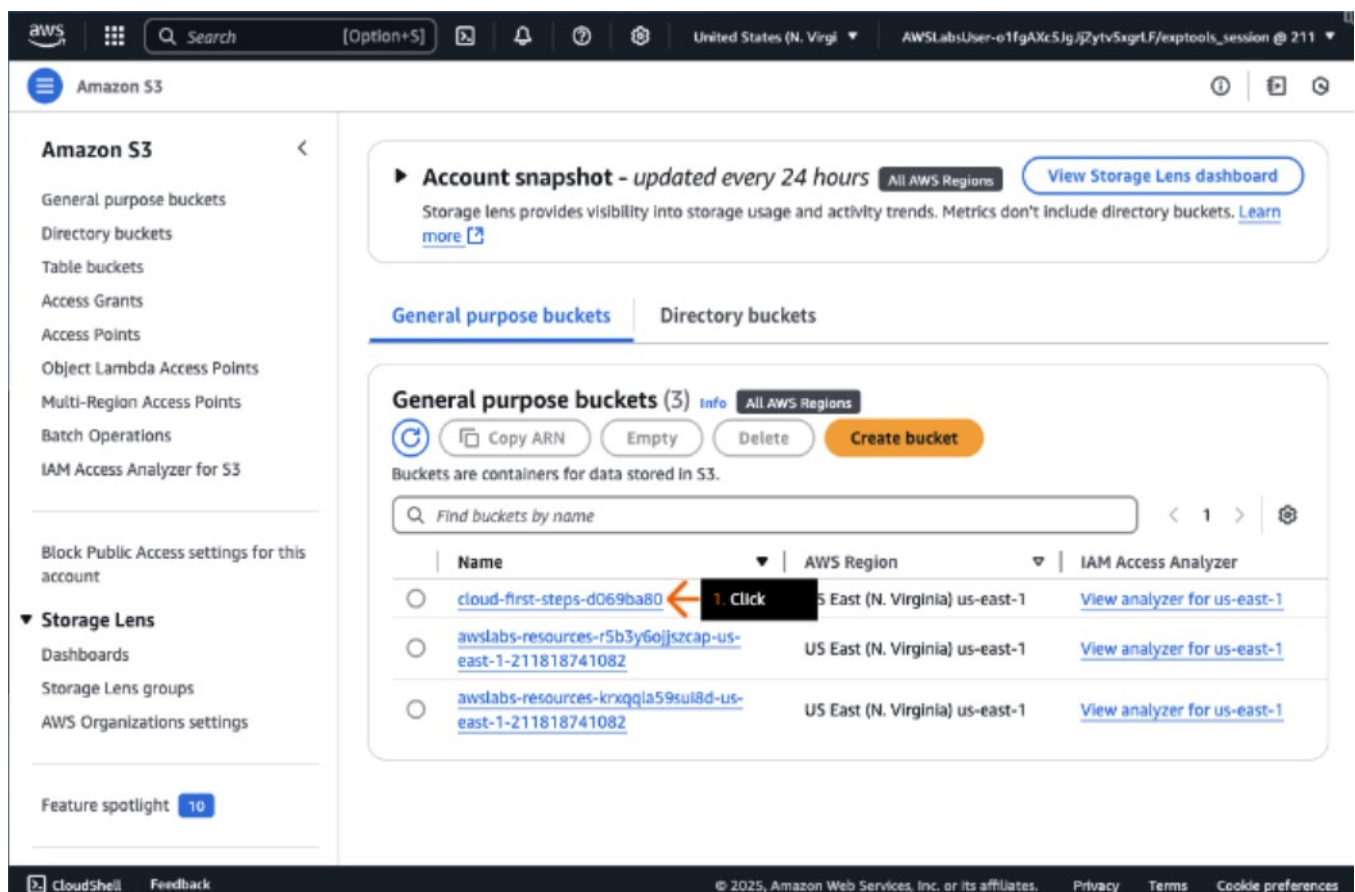
An object is a file and any metadata that describes the file.

A bucket is a container for objects.

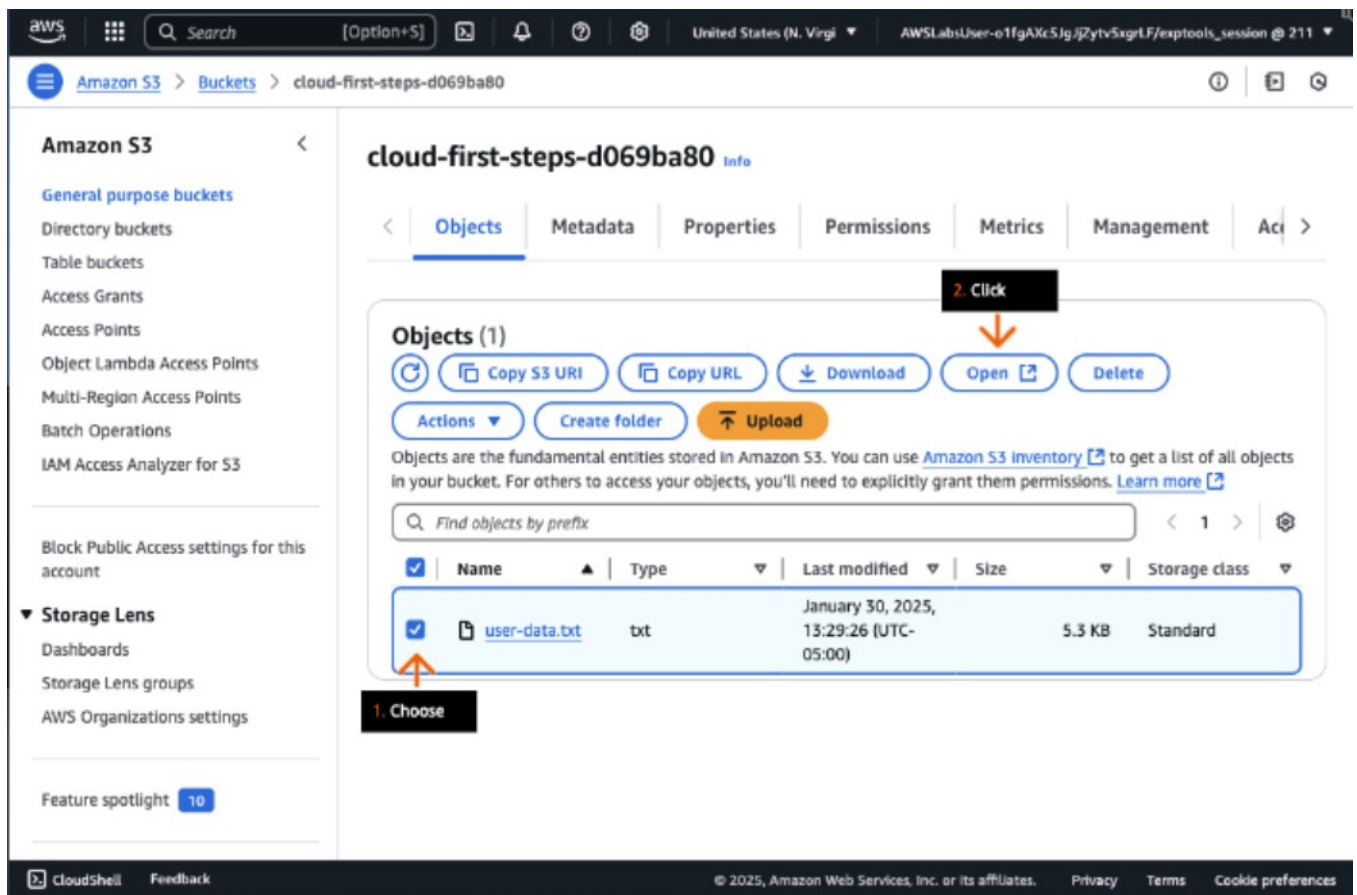
To store your data in Amazon S3, you first create a bucket and specify a bucket name and AWS Region. Then, you upload your data to that bucket as objects in Amazon S3.

Each object has a key (or key name), which is the unique identifier for the object within the bucket.

- On the General purpose buckets tab, click the bucket name that starts with cloud-first-steps-.



- On the Objects tab, choose the check box to select the user-data.txt file.
- Click Open.
 - The user-data.txt file opens in a new browser tab (or window).



- In the new browser tab, review the file contents.
 - This user data script installs and launches a web server, using port 80, to display internal information about the instance.

When you launch an Amazon Elastic Compute Cloud (Amazon EC2) instance, you can pass user data to the instance that is used to perform automated configuration tasks, or to run scripts after the instance starts.

The user data is as follows:

```
#!/bin/bash
sudo yum update -y
sudo yum install -y httpd
sudo yum install -y git
export TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
export META_INST_ID=`curl http://169.254.169.254/latest/meta-data/instance-id -H "X-aws-ec2-metadata-token: $TOKEN"`
export META_INST_TYPE=`curl http://169.254.169.254/latest/meta-data/instance-type -H "X-aws-ec2-metadata-token: $TOKEN"`
```

```

export META_INST_AZ=`curl
http://169.254.169.254/latest/meta-data/placement/availability-zone -H
"X-aws-ec2-metadata-token: $TOKEN"`
cd /var/www/html
echo "<!DOCTYPE html>" >> index.html
echo "<html lang='en'>" >> index.html
echo "<head>" >> index.html
echo "    <meta charset='UTF-8'>" >> index.html
echo "    <meta name='viewport' content='width=device-width, initial-
scale=1.0'>" >> index.html
echo "    <style>" >> index.html
echo "        @import url('https://fonts.googleapis.com/css?
family=Open+Sans&display=swap');" >> index.html
echo "        html {" >> index.html
echo "            position: relative;" >> index.html
echo "            overflow-x: hidden !important;" >> index.html
echo "        }" >> index.html
echo "        * {" >> index.html
echo "            box-sizing: border-box;" >> index.html
echo "        }" >> index.html
echo "        body {" >> index.html
echo "            font-family: 'Open Sans', sans-serif;" >> index.html
echo "            color: #324e63;" >> index.html
echo "        }" >> index.html
echo "        .wrapper {" >> index.html
echo "            width: 100%;" >> index.html
echo "            width: 100%;" >> index.html
echo "            height: auto;" >> index.html
echo "            min-height: 90vh;" >> index.html
echo "            padding: 50px 20px;" >> index.html
echo "            padding-top: 100px;" >> index.html
echo "            display: flex;" >> index.html
echo "        }" >> index.html
echo "        .instance-card {" >> index.html
echo "            width: 100%;" >> index.html
echo "            min-height: 380px;" >> index.html
echo "            margin: auto;" >> index.html
echo "            box-shadow: 12px 12px 2px 1px rgba(13, 28, 39, 0.4);"
>> index.html
echo "            background: #fff;" >> index.html
echo "            border-radius: 15px;" >> index.html
echo "            border-width: 1px;" >> index.html
echo "            max-width: 500px;" >> index.html
echo "            position: relative;" >> index.html
echo "            border: thin groove #9c83ff;" >> index.html
echo "        }" >> index.html
echo "        .instance-card__cnt {" >> index.html
echo "            margin-top: 35px;" >> index.html

```

```

echo "        text-align: center;" >> index.html
echo "        padding: 0 20px;" >> index.html
echo "        padding-bottom: 40px;" >> index.html
echo "        transition: all .3s;" >> index.html
echo "    }" >> index.html
echo "    .instance-card__name {" >> index.html
echo "        font-weight: 700;" >> index.html
echo "        font-size: 24px;" >> index.html
echo "        color: #6944ff;" >> index.html
echo "        margin-bottom: 15px;" >> index.html
echo "    }" >> index.html
echo "    .instance-card-inf__item {" >> index.html
echo "        padding: 10px 35px;" >> index.html
echo "        min-width: 150px;" >> index.html
echo "    }" >> index.html
echo "    .instance-card-inf__title {" >> index.html
echo "        font-weight: 700;" >> index.html
echo "        font-size: 27px;" >> index.html
echo "        color: #324e63;" >> index.html
echo "    }" >> index.html
echo "    .instance-card-inf__txt {" >> index.html
echo "        font-weight: 500;" >> index.html
echo "        margin-top: 7px;" >> index.html
echo "    }" >> index.html
echo "    </style>" >> index.html
echo "    <title>Amazon EC2 Status</title>" >> index.html
echo "</head>" >> index.html
echo "<body>" >> index.html
echo "    <div class='wrapper'>" >> index.html
echo "        <div class='instance-card'>" >> index.html
echo "            <div class='instance-card__cnt'>" >> index.html
echo "                <div class='instance-card__name'>Your EC2
Instance is running!</div>" >> index.html
echo "                <div class='instance-card-inf'>" >> index.html
echo "                    <div class='instance-card-inf__item'>" >>
index.html
echo "                        <div class='instance-card-
inf__txt'>Instance Id</div>" >> index.html
echo "                        <div class='instance-card-inf__title'>"
$META_INST_ID "</div>" >> index.html
echo "                    </div>" >> index.html
echo "                    <div class='instance-card-inf__item'>" >>
index.html
echo "                        <div class='instance-card-
inf__txt'>Instance Type</div>" >> index.html
echo "                        <div class='instance-card-inf__title'>"
$META_INST_TYPE "</div>" >> index.html
echo "                    </div>" >> index.html

```

```

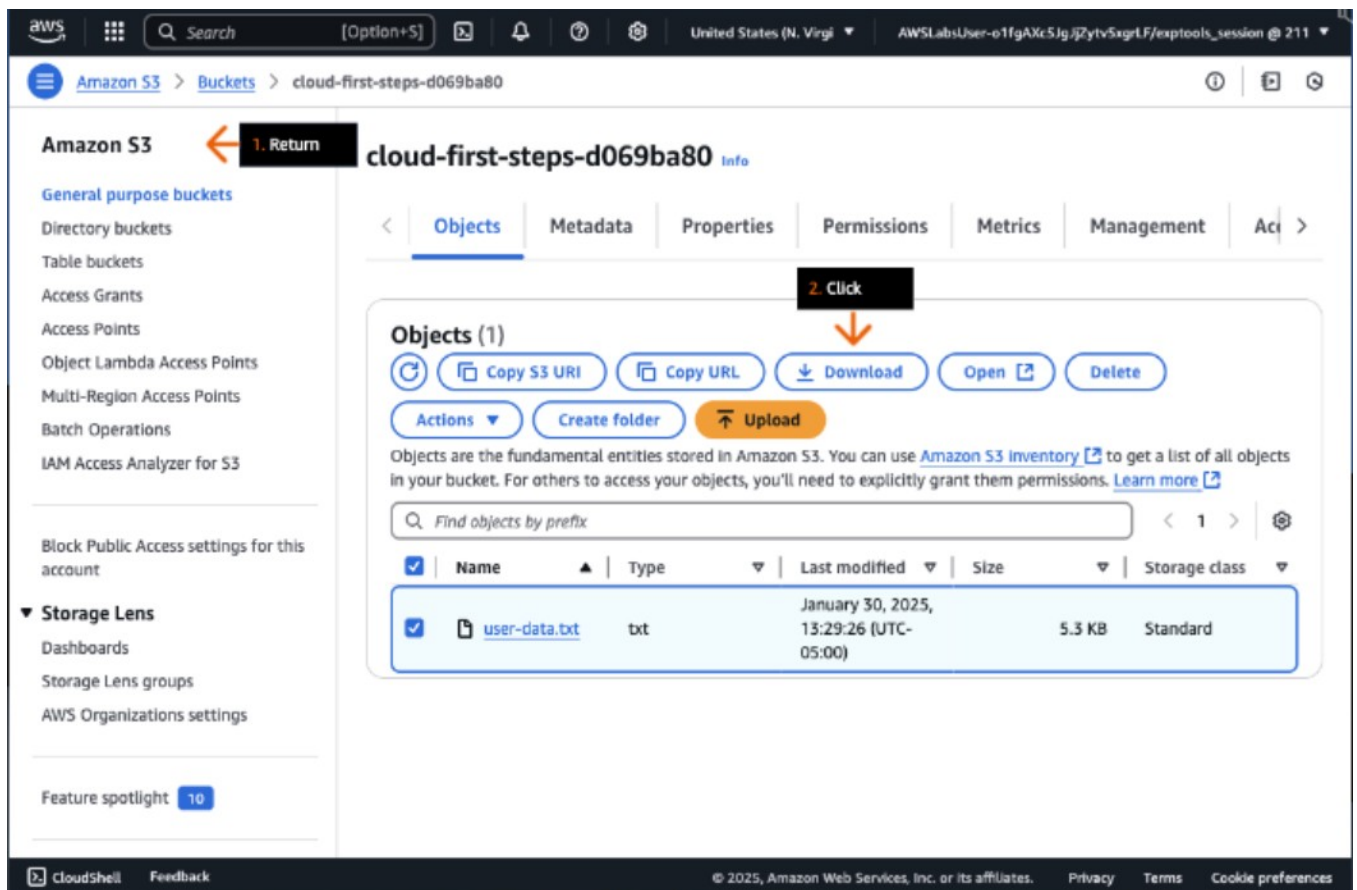
echo "                                <div class="instance-card-inf__item">" >>
index.html
echo "                                <div class="instance-card-
inf__txt">Availability zone</div>" >> index.html
echo "                                <div class="instance-card-inf__title">"
$META_INST_AZ "</div>" >> index.html
echo "                                </div>" >> index.html
echo "                                </div>" >> index.html
echo "                                </div>" >> index.html
echo "                                </div>" >> index.html
echo "</body>" >> index.html
echo "</html>" >> index.html
sudo service httpd start

```

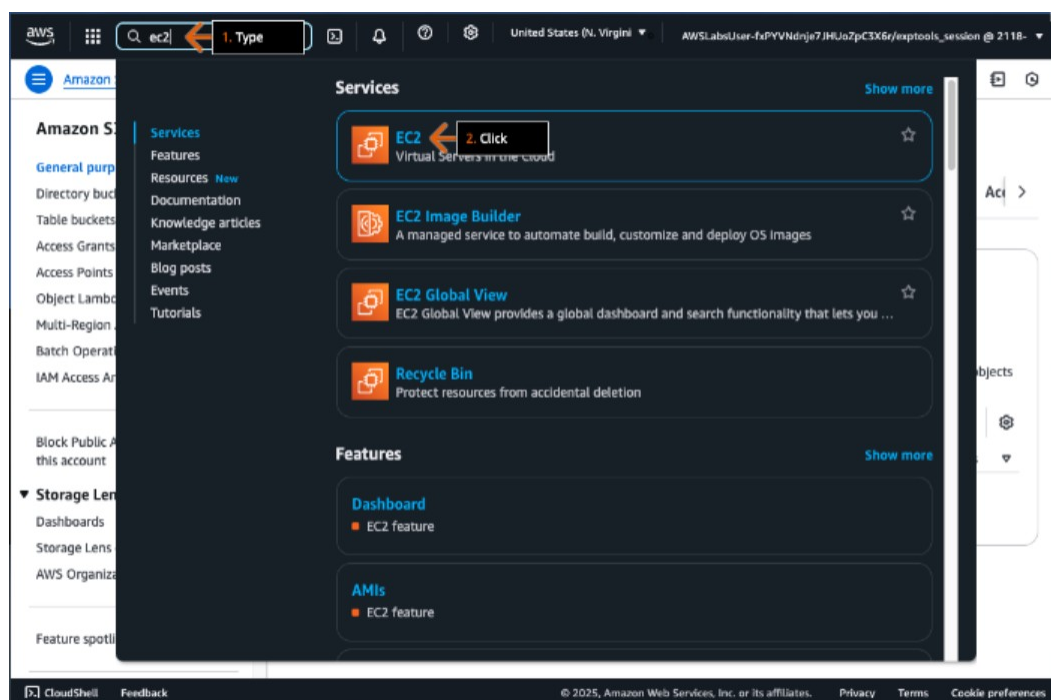
This user data script is designed to automatically set up your EC2 instance when it launches. Think of it as a startup script for the server. Here's what it does:

First, it updates the system software and installs a couple of programs: the **Apache web server** (so the instance can host a website) and **Git** (a version control tool, though not directly used in the web page creation part). Then, it connects to a special service running on the EC2 instance itself to gather information about it, like its unique **Instance ID**, its **Instance Type** (like t2.micro, m5.large, etc.), and the **Availability Zone** it's located in. Finally, it uses this collected information to create a simple HTML webpage right in the web server's default directory. This webpage is designed to display the Instance ID, Instance Type, and Availability Zone in a clear format. After creating the page, it starts the Apache web server so that when you access the EC2 instance's public address in a browser, you will see this generated webpage showing the instance's details.

- Return to the Amazon S3 console in the other browser tab.
- On the Objects tab, to save the user-data.txt file to your device, click Download.



- In the top navigation bar search box, type: **ec2**
- In the search results, under Services, click **EC2**.



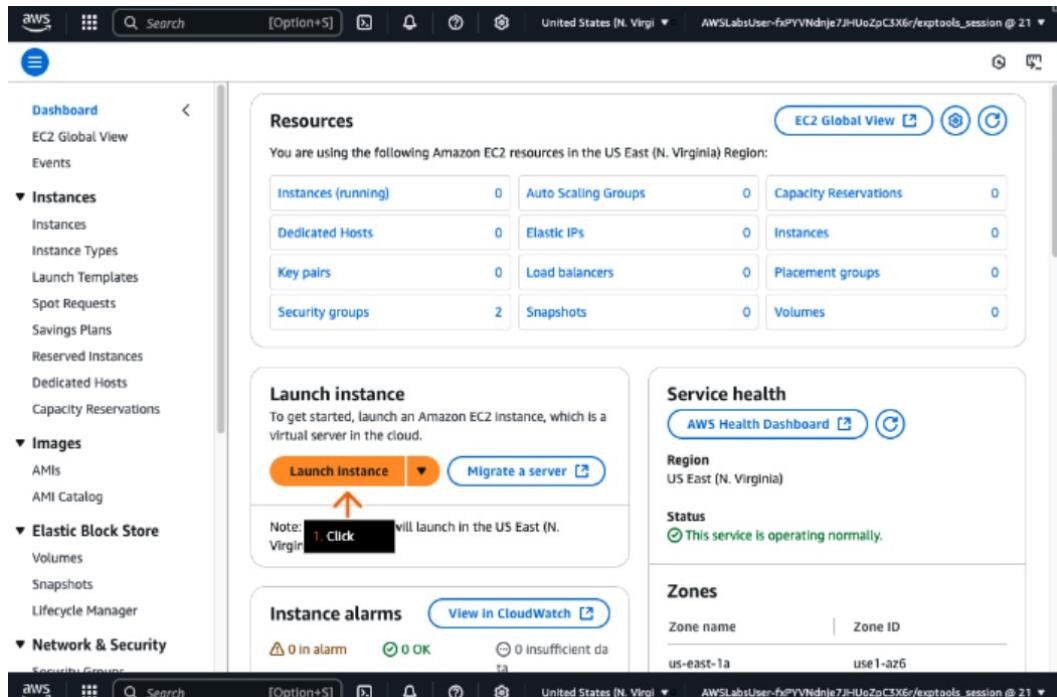
Amazon EC2 provides on-demand, scalable computing capacity in the AWS Cloud.

Using Amazon EC2 reduces hardware costs, so you can develop and deploy applications faster.

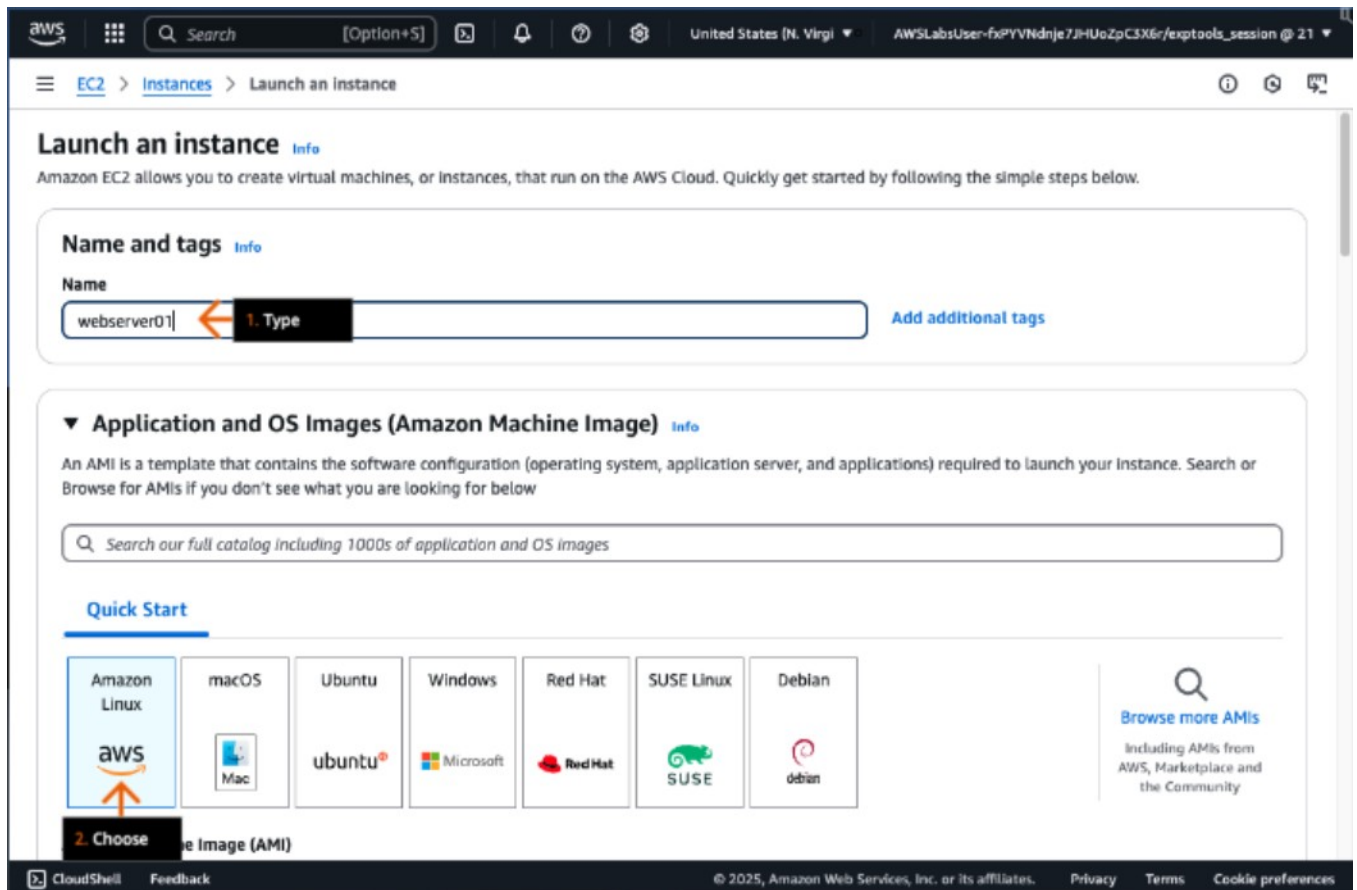
You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage.

An EC2 Instance is a virtual server in the cloud.

- In the Launch instance section, click Launch instance.



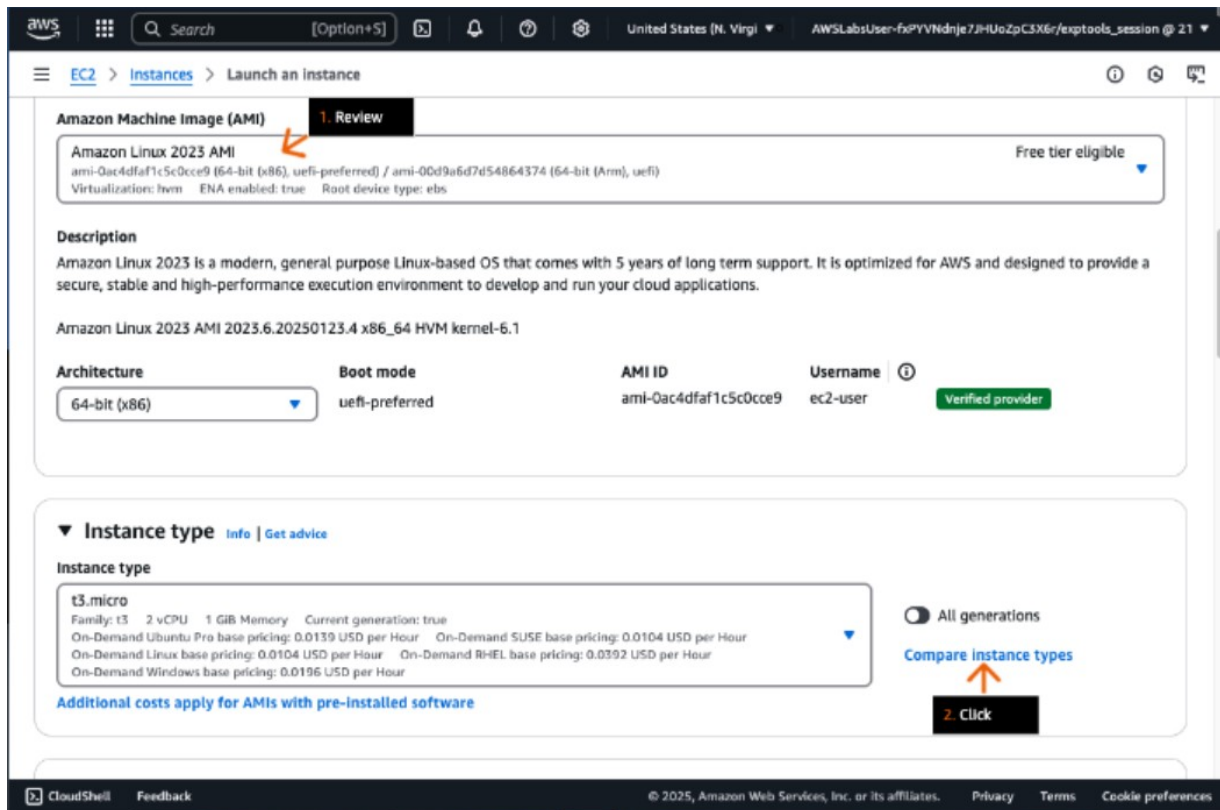
- In the Name and tags section, for Name, type a name that you like, such as webserver01.
- In the Application and OS Images section, under Quick Start, choose Amazon Linux.



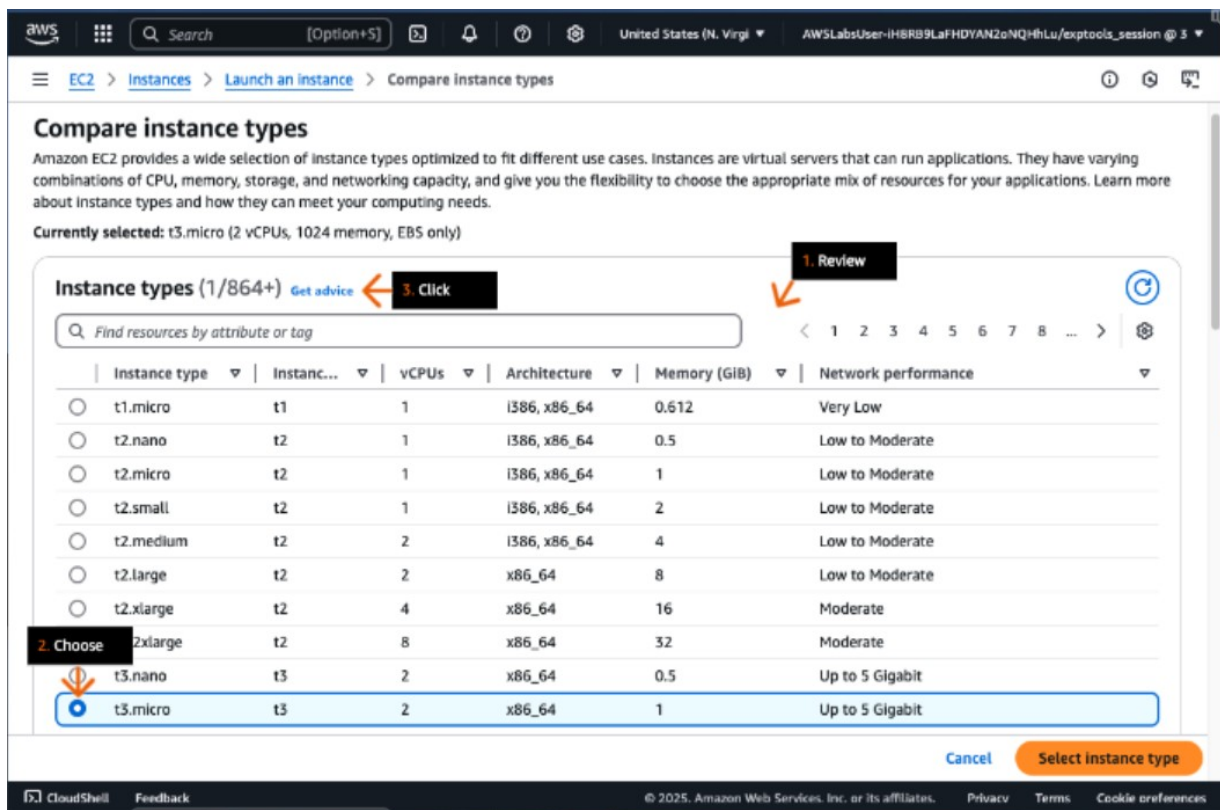
An Amazon Machine Image (AMI) provides the information, such as operating system and software configurations, required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need different configurations.

- For Amazon Machine Image (AMI), on the dropdown list, choose or keep Amazon Linux 2023 AMI.
- For Instance type, click Compare instance types.

When you launch an instance, the instance type that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families based on these capabilities.

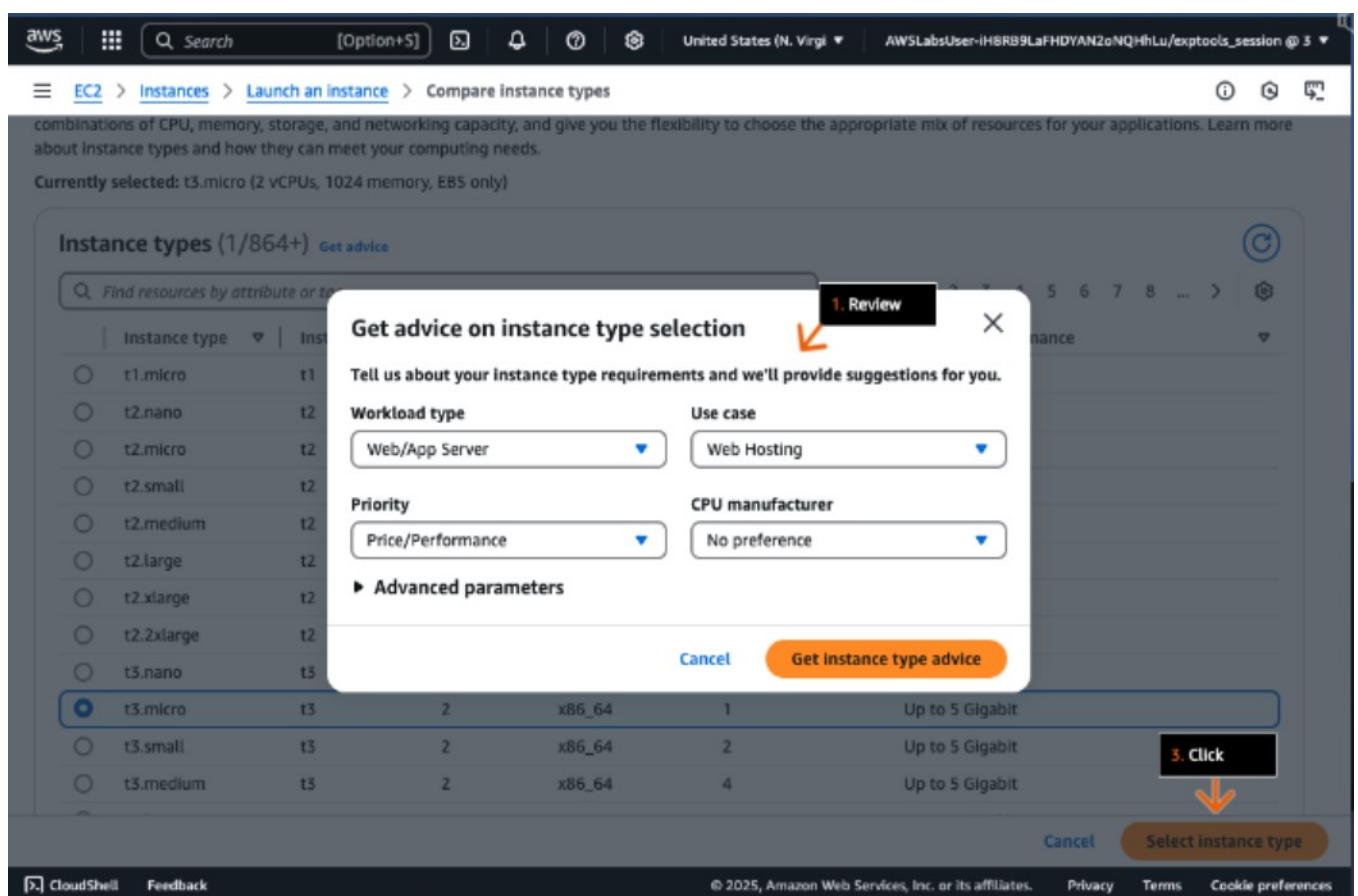


- In the Instance types section, review the available instances and their associated resources.
- Choose t3.micro.
- Click Get advice.

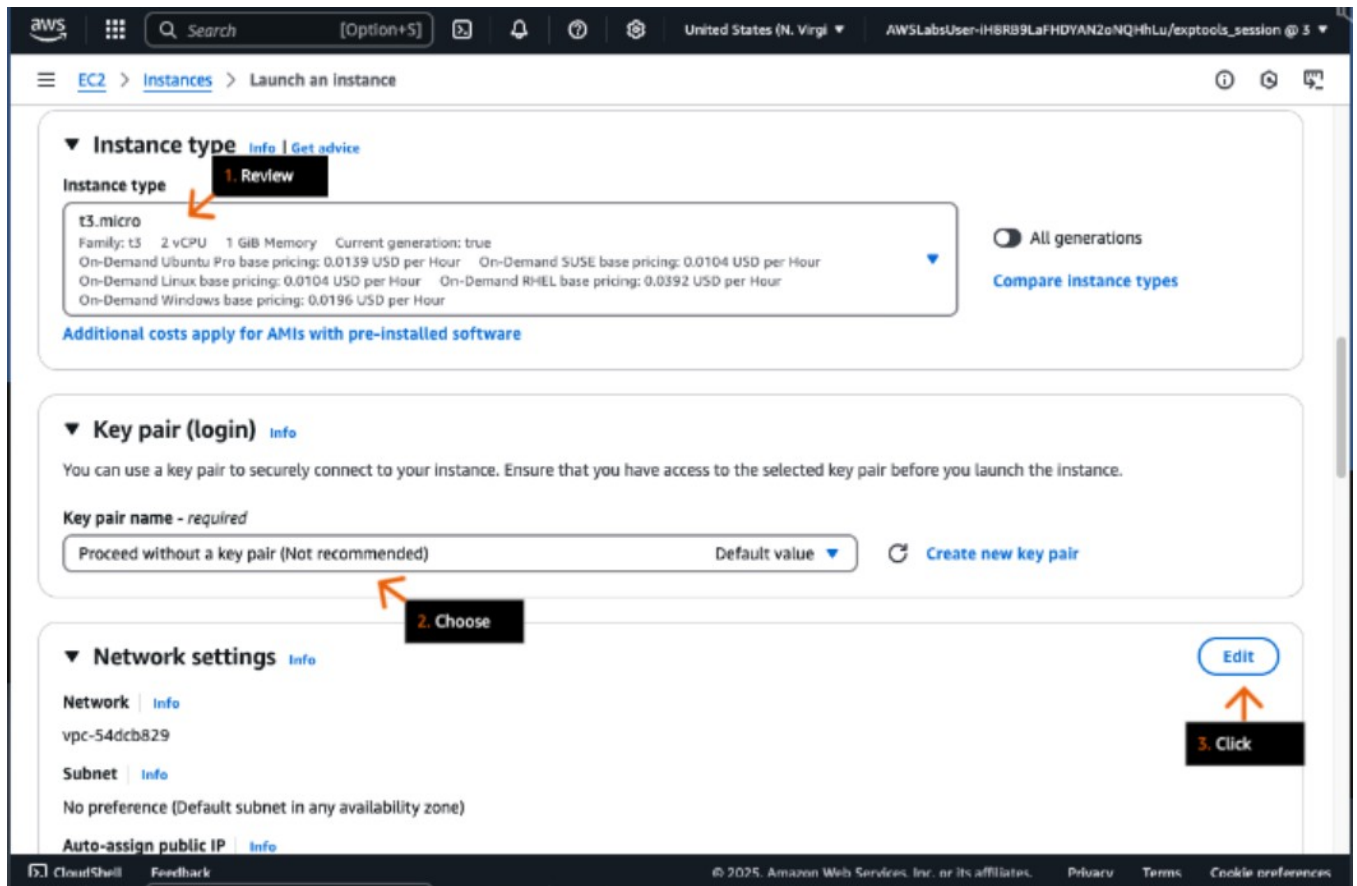


Amazon EC2 provides a wide selection of Instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each Instance type includes one or more instance sizes, so you can scale your resources to the requirements of your target workload.

- In the pop-up box, to see instance type and family recommendations for various workloads, review different instance type requirements.
 - You can use the dropdown lists to experiment with different requirements, clicking Get instance type advice for each new requirement.
- When finished, click Close.
- Click Select instance type.



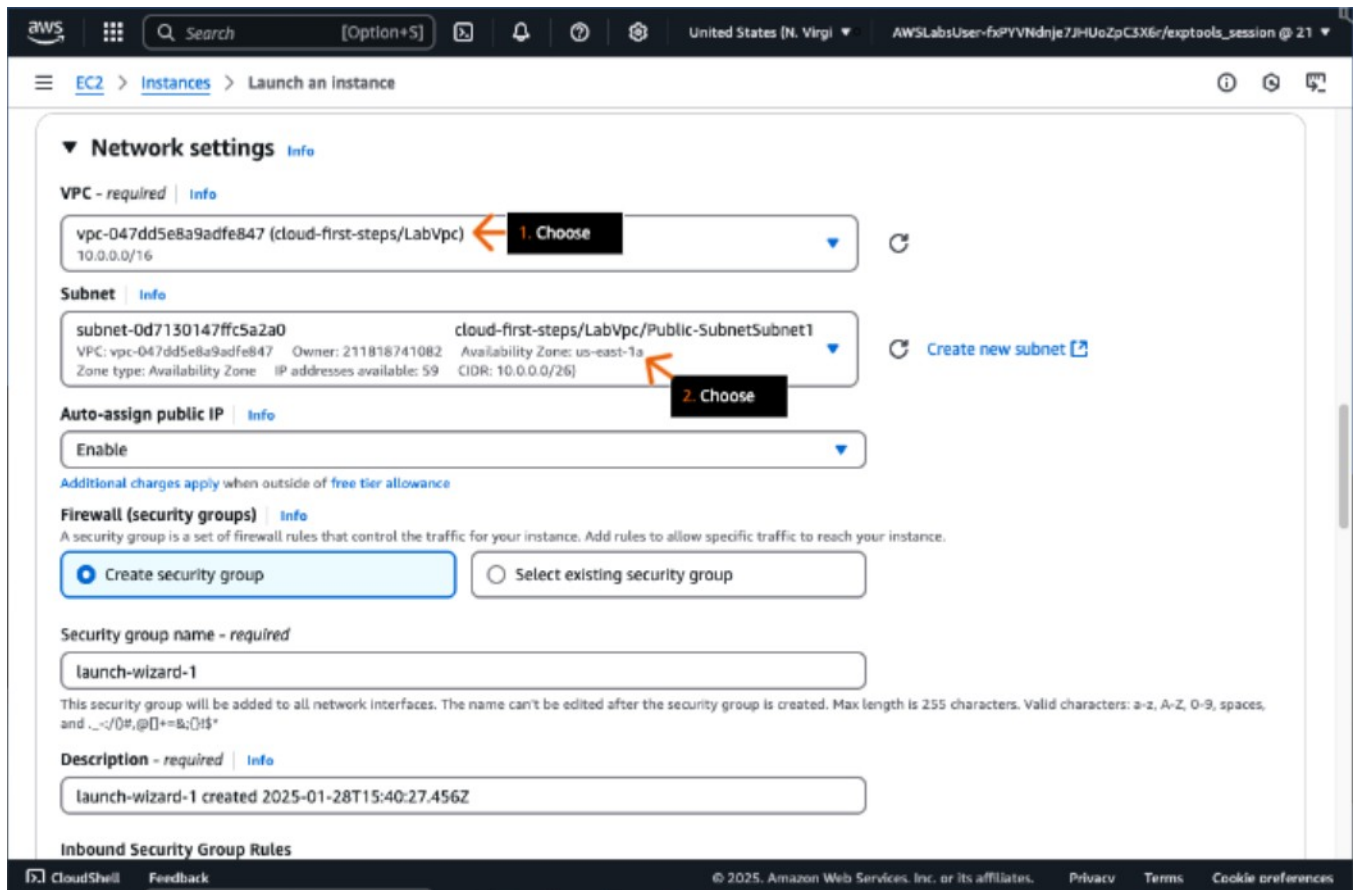
- Review the Instance type to confirm that t3.micro is selected.
- For Key pair name, choose Proceed without a key pair.
- In the Network settings section, click Edit.



Amazon EC2 uses public key cryptography to encrypt and decrypt login information. Public key cryptography uses a public key to encrypt a piece of data, and then the recipient uses their private key to decrypt the data. The public and private keys are known as a key pair.

- For VPC, choose cloud-first-steps/LabVpc.
 - Your solution will fail if you do not choose this VPC.
- For Subnet, choose the subnet in the us-east-1a Availability Zone.

Note the AZ choices on the dropdown list. In the upcoming DIY section of this solution, you must choose the subnet in the other AZ.

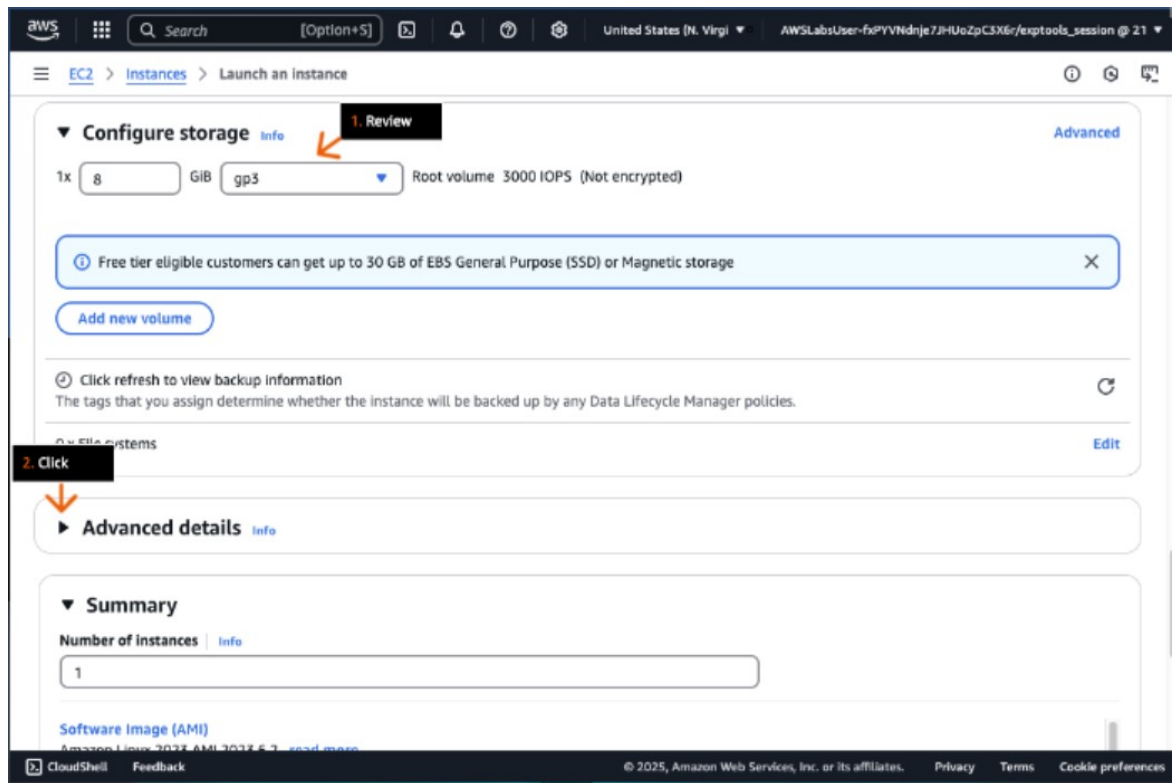


A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. While a VPC resides in an AWS Region, a subnet must reside within a single AZ.

- For Security group name, type: Lab-SG
- For Description, type: HTTP Security Group
- For Type, choose HTTP.
- In the information alert, review the rules notice.
 - The notice shows that port 80 (HTTP) is open to the entire internet. While web servers usually do need to allow access to the public, we recommended that security group rules allow the least amount of access possible.

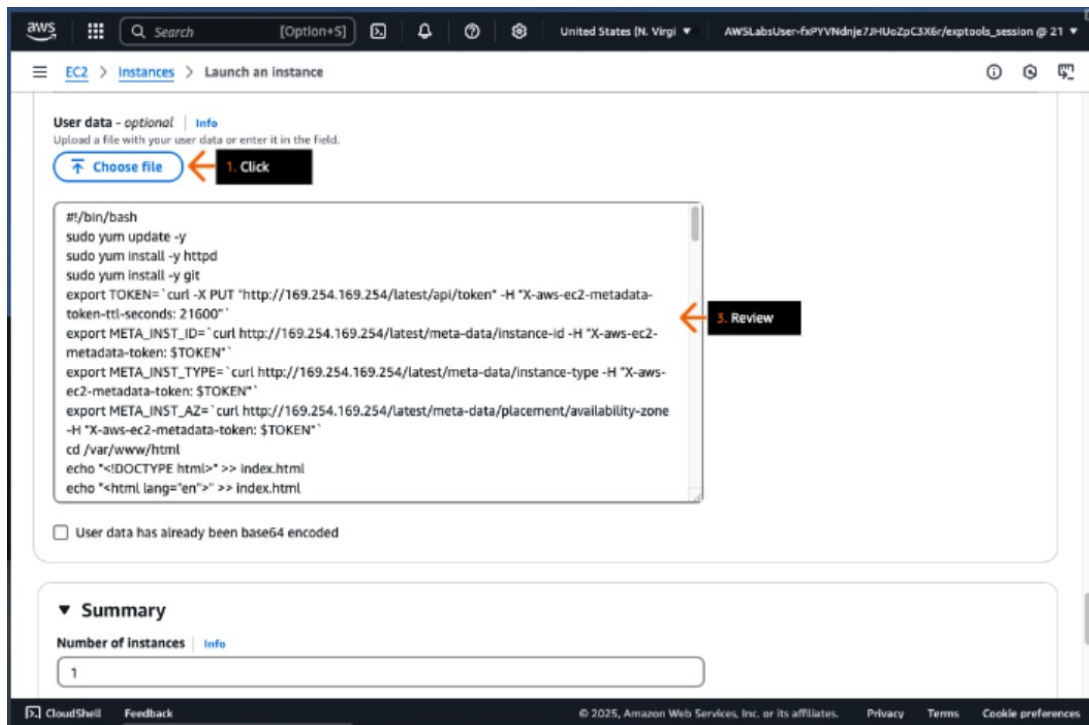
A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you can specify one or more security groups; otherwise, the default security group is used. You can add rules to each security group that allows traffic to or from its associated instances.

- In the Configure storage section, review the default option of an 8 GiB gp3 volume.
- Click to expand Advanced details.



When you launch an instance, the root device volume contains the image used to boot the instance. General Purpose SSD (gp2 and gp3) volumes are backed by solid-state drives (SSDs). They balance price and performance for a wide variety of transactional workloads.

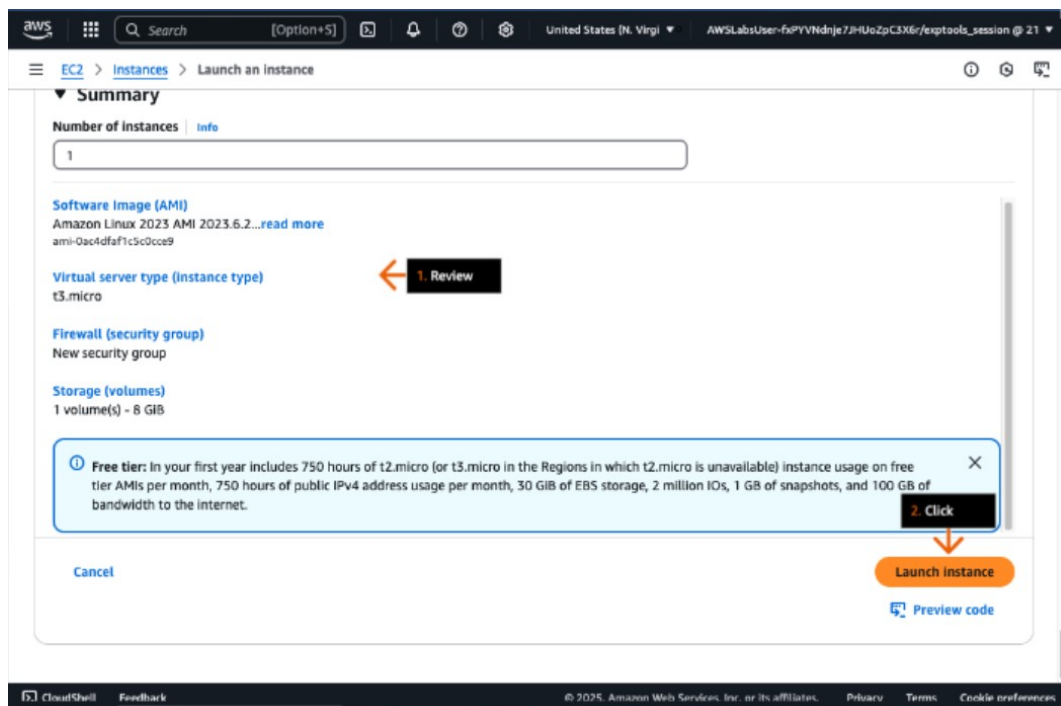
- For User data, click Choose file.
- In the pop-up box , choose the user-data.txt file that you downloaded in an earlier step.
- In the text box, review the file contents.



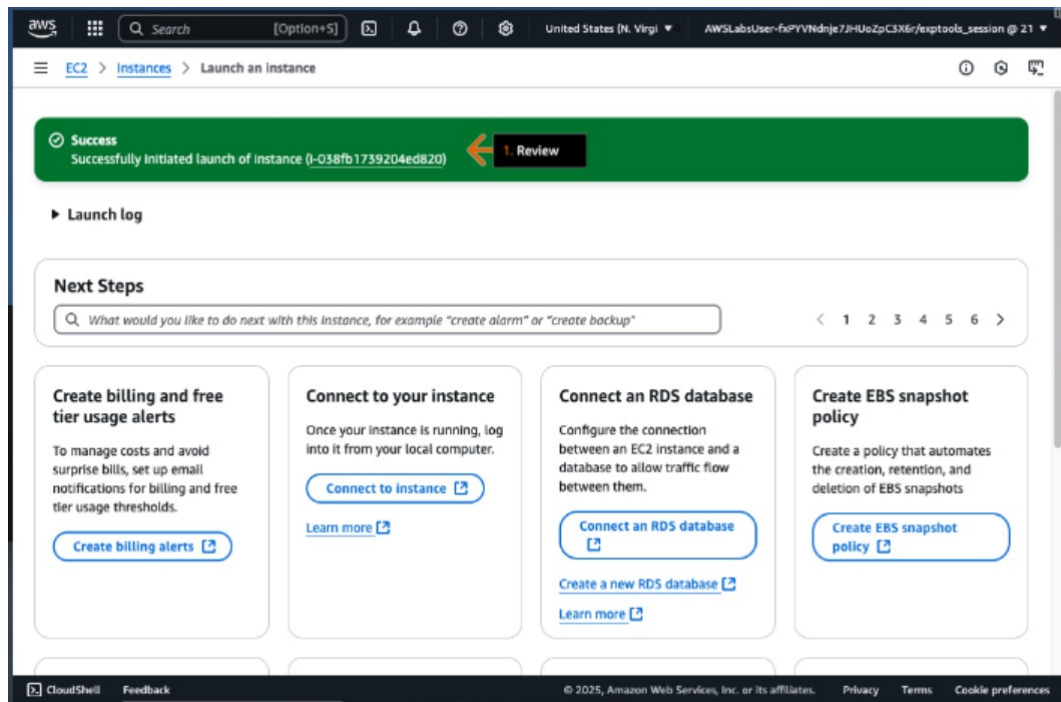
- Review the Summary section.
 - This section, when your browser is fully expanded, floats on the right side.

It's always a good idea to review the instance launch details that you have configured before you deploy the instance.

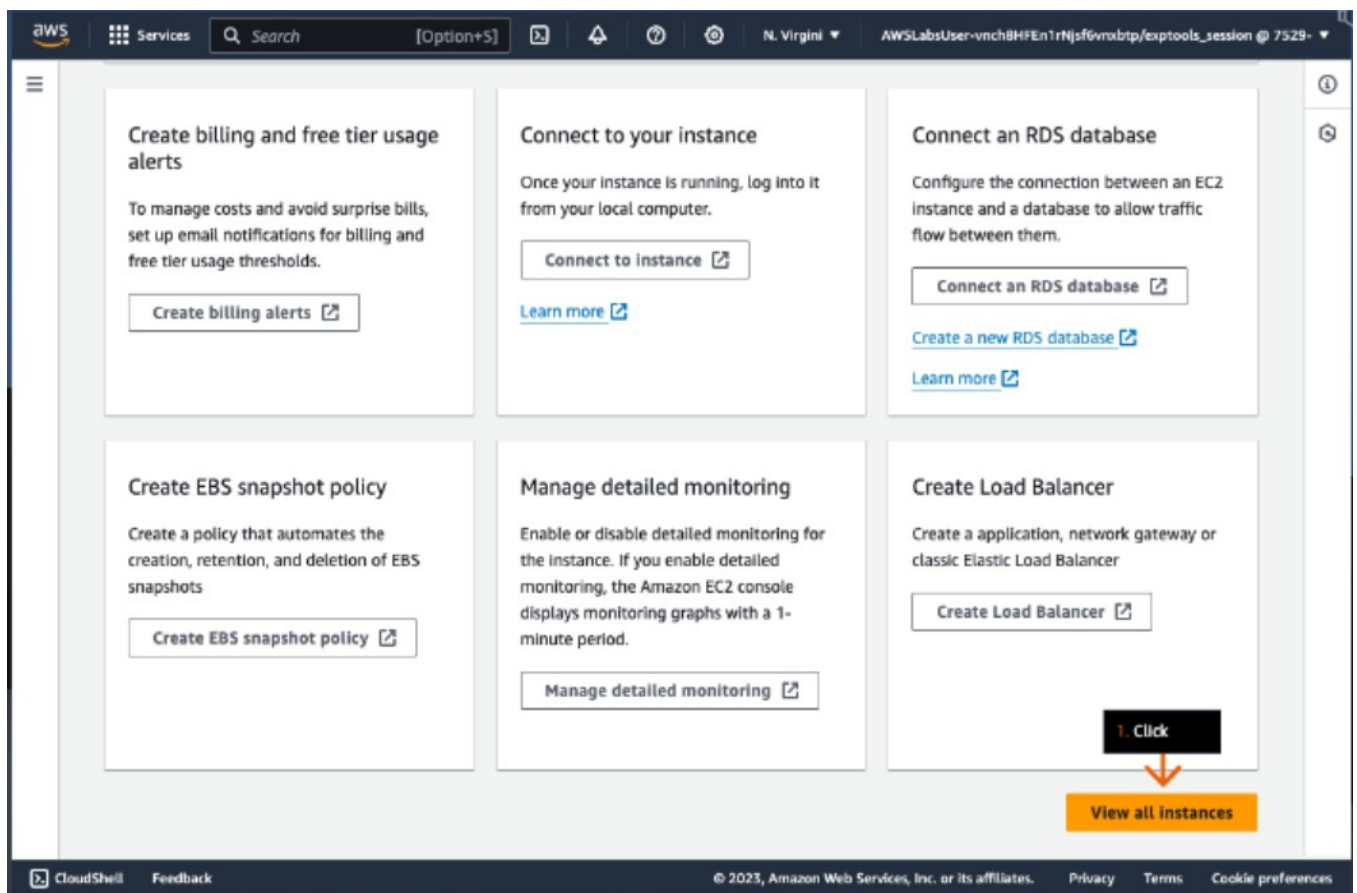
- Click Launch instance.



- In the success alert, review the message

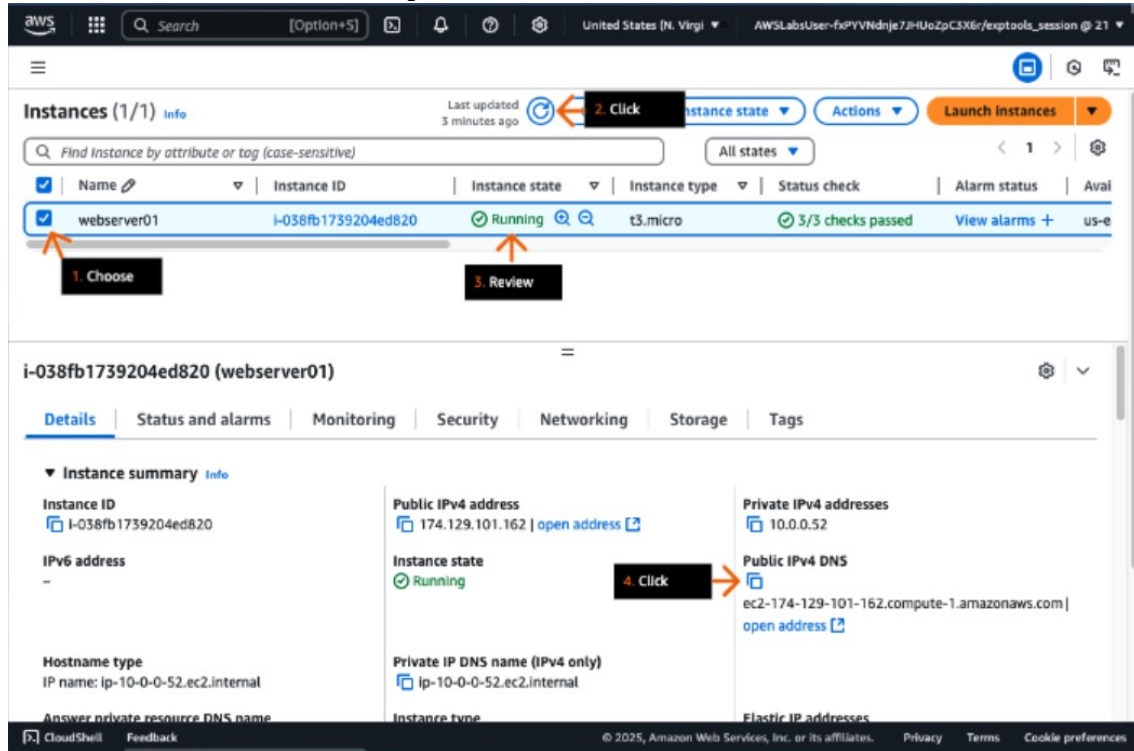


- At the bottom of the page, click View all instances



- In the Instances section, choose the check box to select your EC2 instance.
- After 2-3 minutes, click the section's refresh icon.
- Under Instance state, review to confirm that the state has changed to Running.

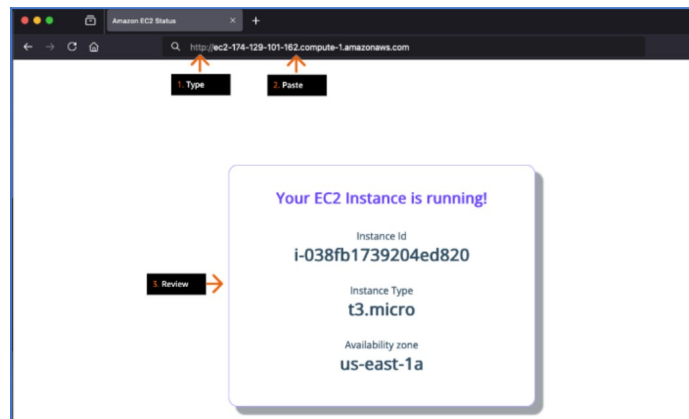
- o The change to Running might take several minutes, and you might need to refresh several times.
- After the instance state is Running, on the Details tab, under Public IPv4 DNS, click the copy icon to copy the provided address.
 - o Do not click the "open address" link.



An instance enters the pending state when it launches for the first time. It changes to a running state when it is ready for use.

- In a new browser tab address bar, type: `http://`
- Next to `http://`, paste the DNS address that you just copied and press Enter.
- On the page, review the details about your instance.
 - o If you see a connection timeout message when opening the webpage, check that the address begins with `http://` and not `https://`.
 - o The public DNS and the security group are used to access the instance details that appear on the webpage. The public DNS record translates a domain name to an IP address. The record informs the browser which server to connect to.

DNS can be configured and customized using Amazon Route 53. Route 53 is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to internet applications by translating names, such as `example.com`, into numeric IP addresses, such as `192.0.2.1`, that computers use to connect to each other.



Launching Web Server 02 in a Different Availability Zone (No Key Pair)

Follow these steps in the AWS Management Console to launch your second web server instance:

- Navigate to the EC2 dashboard.
- Click the **Launch instance** button.
- Assign Name as webserver02.
- **Choose an Amazon Machine Image (AMI):** Select the **same AMI** that you used for webserver01.

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name
[Add additional tags](#)

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents
Quick Start

Amazon Linux
aws

macOS
Mac

Ubuntu
ubuntu

Windows
Microsoft

Red Hat
Red Hat

SUSE Linux
SUSE

Debian
debian

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-0e449927258d45bc4 (64-bit (x86), uefi-preferred) / ami-086a54924e40cab98 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▼

- **Choose an Instance Type:** Select the **same instance type** that you used for webserver01 (e.g., t3.micro).
- **Select an existing key pair or create a new key pair:** In the key pair dialog box, select **"Proceed without a key pair"** from the dropdown menu. You will need to acknowledge that you understand that you cannot connect to the instance via SSH without a key pair. **Note:** Choosing this option means you will not be able to SSH into the instance using a key pair later.

▼ Instance type

Info | Get advice

Instance type

t3.micro

Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour

All generations
Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login)

Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Proceed without a key pair (Not recommended)

Default value ▼

Create new key pair

- **Configure Network Settings:**
 - **VPC:** Ensure the **same VPC** is selected that contains your subnets in us-east-1a and us-east-1b.
 - **Subnet:** This is crucial. Select the **subnet located in us-east-1b**. You should see the Availability Zone listed next to the subnet name or ID.
 - **Auto-assign Public IP:** Ensure this is **Enable** if you want the instance to automatically receive a public IP address upon launch (necessary to access the web page).
- **Configure Security Group:**
 - Choose the option **Select an existing security group**.
 - Select the **same security group** that you assigned to webserver01. This security group should have rules allowing inbound traffic (e.g., HTTP/port 80) from appropriate sources.

▼ Network settings

Info

VPC - required

Info

vpc-01a19a3ae66134763 (cloud-first-steps/LabVpc)

10.0.0.0/16

Subnet

Info

subnet-0943d908c19e82701

cloud-first-steps/LabVpc/Public-SubnetSubnet2

VPC: vpc-01a19a3ae66134763 Owner: 847678444112 Availability Zone: us-east-1b
Zone type: Availability Zone IP addresses available: 58 CIDR: 10.0.0.64/26

Create new subnet

Auto-assign public IP

Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)

Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

Common security groups

Info

Select security groups

Lab-SG sg-08311fb7425263050

VPC: vpc-01a19a3ae66134763

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▶ Advanced network configuration

- **Add Storage.** Accept the default root volume size or configure as needed (same as webserver01 is recommended for consistency).

▼ **Configure storage** [Info](#) Advanced

1x GiB Root volume, 3000 IOPS, Not encrypted

i Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✕

[Add new volume](#)

⌚ Click refresh to view backup information ↻

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

- Scroll down and expand the **Advanced Details** section.
 - **User data:** Paste the **same user data script** you used for webserver01 into the text area.

User data - optional | [Info](#)

Upload a file with your user data or enter it in the field.

[Choose file](#)

```
#!/bin/bash
sudo yum update -y
sudo yum install -y httpd
sudo yum install -y git
export TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
export META_INST_ID=`curl http://169.254.169.254/latest/meta-data/instance-id -H "X-aws-ec2-metadata-token: $TOKEN"`
export META_INST_TYPE=`curl http://169.254.169.254/latest/meta-data/instance-type -H "X-aws-ec2-metadata-token: $TOKEN"`
export META_INST_AZ=`curl http://169.254.169.254/latest/meta-data/placement/availability-zone -H "X-aws-ec2-metadata-token: $TOKEN"`
cd /var/www/html
echo "<!DOCTYPE html>" >> index.html
echo "<html lang='en'>" >> index.html
```

☐ User data has already been base64 encoded

- **Review Instance Launch:** Review all the configuration details to ensure they are correct, especially the AMI, instance type, subnet (in us-east-1b), user data, and security group.
- Click **Launch Instances**.

View your instances in the EC2 console. You should see webserver02 entering the pending state and then transitioning to the running state. Verify that its Availability Zone is us-east-1b.

The screenshot shows the AWS Management Console for EC2 Instances. A table lists two instances: webserver01 and webserver02. webserver02 is selected. Below the table, the 'Monitoring' tab is active, displaying several performance metrics as line graphs: CPU utilization (%), Network in (bytes), Network out (bytes), Network packets in (count), Network packets out (count), Metadata no token (count), CPU credit usage (count), and CPU credit balance (count).

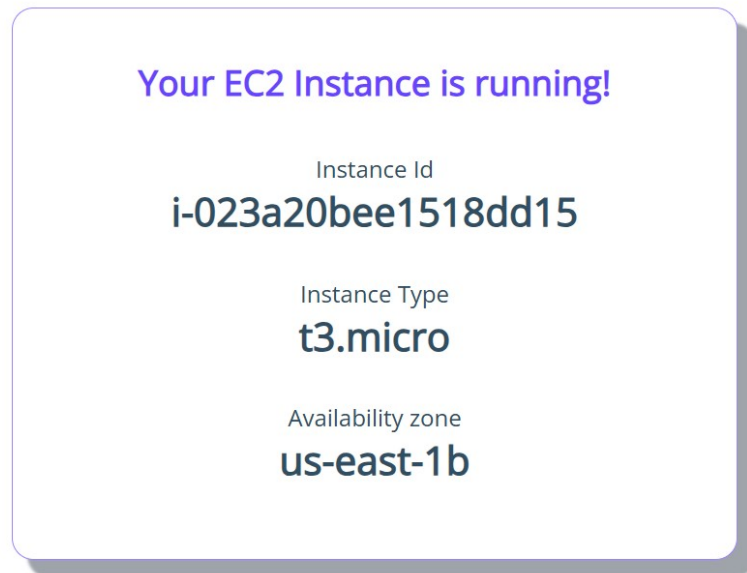
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address
webserver01	i-09ed47f8d12faed86	Running	t3.micro	3/3 checks passed	View alarms +	us-east-1a	ec2-54-152-254-10.compute-1.amazonaws.com	54.152.254.10
webserver02	i-023a20bee1518dd15	Running	t3.micro	3/3 checks passed	View alarms +	us-east-1b	ec2-54-235-25-161.compute-1.amazonaws.com	54.235.25.161

- Select the webserver02 instance by clicking on its row.
- In the details pane at the bottom of the screen, locate the **Public IPv4 DNS** entry.
- **Copy** the Public IPv4 DNS name.
- Open a **new tab** in your web browser.
- **Paste** the copied Public IPv4 DNS name into the address bar and press Enter.

You should see the web page generated by the user data script, displaying the details (Instance ID, Instance Type, and Availability Zone) of webserver02.

The screenshot shows the details for instance i-023a20bee1518dd15 (webserver02). The 'Details' tab is active, displaying instance summary information. Key details include the Public IPv4 address (54.235.25.161), Instance state (Running), Private IP DNS name (ip-10-0-0-125.ec2.internal), Instance type (t3.micro), VPC ID (vpc-01a19a3ae66134763), and Public IPv4 DNS (ec2-54-235-25-161.compute-1.amazonaws.com).

Field	Value
Instance ID	i-023a20bee1518dd15
IPv6 address	-
Hostname type	IP name: ip-10-0-0-125.ec2.internal
Answer private resource DNS name	-
Auto-assigned IP address	54.235.25.161 [Public IP]
Public IPv4 address	54.235.25.161 open address
Instance state	Running
Private IP DNS name (IPv4 only)	ip-10-0-0-125.ec2.internal
Instance type	t3.micro
VPC ID	vpc-01a19a3ae66134763 (cloud-first-steps/LabVpc)
Private IPv4 addresses	10.0.0.125
Public IPv4 DNS	ec2-54-235-25-161.compute-1.amazonaws.com open address
Elastic IP addresses	-
AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more



Conclusion:

You have successfully achieved all the defined objectives for this lab. You have demonstrated the ability to launch Amazon EC2 instances, utilize user data scripts to automate instance configuration and display details in a browser, and successfully launch a second instance in a separate Availability Zone within the same AWS Region. This hands-on experience highlights fundamental skills in deploying automated and more resilient infrastructure on AWS.