

AMAZON WEB SERVICE EC2(AMAZON ELASTIC COMPUTE CLOUD) WALKTHROUGH

- What is it?

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud.

- What does it do?

EC2 is a service on AWS that allows users to rent virtual computers to run their own computer applications according to their business needs. The user can launch virtual servers, configure security and networking, and manage storage for this virtual machine just as someone would for a real on-premises server. It reduces the need to estimate traffic by providing means to scale up or down as per the demand and pay as you go.

- How does it work?

One can choose from a variety of pre-built, templated Amazon Machine Images (AMI). Users can also make their own AMI, which will include all custom libraries, data, programs, and pertinent configuration settings. User can also customize settings with Amazon EC2 by adjusting network access and security. After that, user decides which AMI instances they want to run and whether they want to run in several locations. User can boot, terminate, and monitor as many instances as necessary once user is ready to begin. The flexibility of the service enables the reduced costs associated with a "pay for what you use" payment approach. User can manage how many resources are being used at any one time and one can quickly expand his/her VM environment to meet utilization spikes or dips. Traditional on-premises hosting provides one with a certain number of resources for a specific period. As a result, users seldom cannot respond rapidly to a change in use as they would in an environment like the dependable, secure, and elastic Amazon EC2 environment.

- Why do we need it?

One may instantly create virtual computers using Amazon Elastic Compute Cloud (EC2) with a little initial starting expenditure in infrastructure. One can host a range of software on an on-demand infrastructure, from basic websites to enterprise-grade web applications. Acts as plug

and play. Spot pricing can reduce hosting expenses by 80–90%. Offers choices for auto-scaling and load balancing, develop fault resistant architecture.

Companies can choose EC2 accelerated computing instances if a business wants intensive processing and GPU capacity for gaming, animation, machine-learning, AI, etc.

No pre-existing cost incurred so the user can “pay as go” for only the resources used by the user and nothing else as opposed to on-premises servers. Super easy and convenient to spin up a server in any geographical location in the world without compromising access and security. This makes it super reliable, accessible and secure.

- Use Cases?

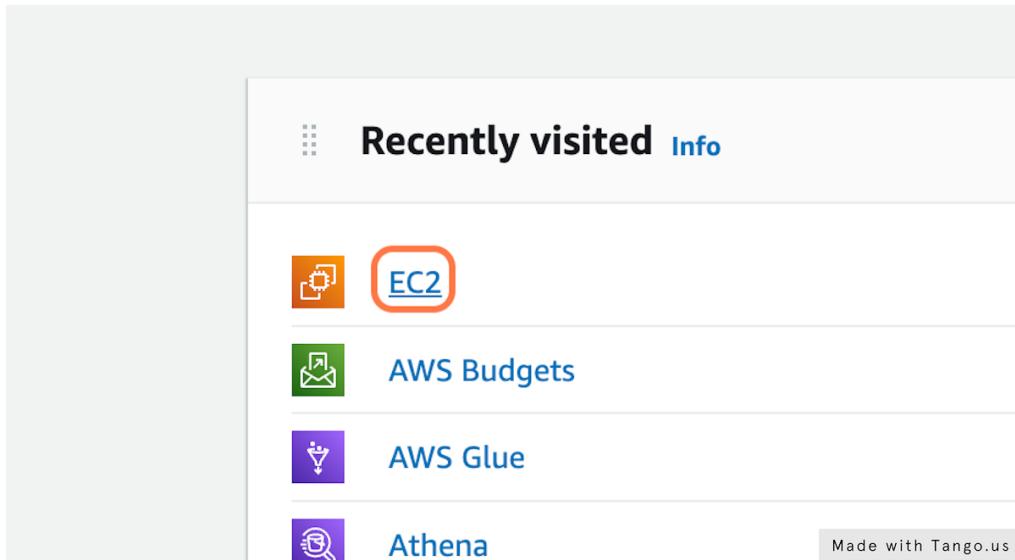
If suddenly a website starts trending on the internet and the number of users increases exponentially, the EC2 server can balance the load automatically and scale as per needed. The business will only pay for the resources needed so it is more cost effective too. If the business had an on-premises server, there is a high chance that the server will overload due to the sudden increase in requests and crash. This can cause the business huge losses. As pose to, if the business decides to invest in more on-premises servers during an off season, those servers won’t be utilized, and the business will still have to pay for the hardware and software cost of it. This is again a loss of money for the business.

Toyota Research Institute example use case: <https://aws.amazon.com/blogs/machine-learning/toyota-research-institute-accelerates-safe-automated-driving-with-deep-learning-at-a-global-scale-on-aws/?p=pm&c=ec2&z=6>

Demonstration:

[**Workflow with Amazon-EC2 \(Workflow Link embedded\)**](#)

1. After logging in to the AWS account, [Go to AWS Management Console](#)
2. Click on EC2



3. Click on Launch instance

The screenshot shows the 'Launch instance' wizard for AWS EC2. On the left, a sidebar lists 'Instances' (with 'Instances' and 'New' options), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances' (with 'New' option), 'Dedicated Hosts', and 'Capacity Reservations'. Below that is a 'Images' section with 'AMIs' (with 'New' option) and 'AMI Catalog'. On the right, the main panel has sections for 'Placement groups' (0) and 'Security groups'. A large orange button labeled 'Launch instance ▾' is highlighted. Below it are three smaller buttons: 'Launch instance Migrate a server' (with a checkmark icon), 'Launch instance from template', and a note stating 'Note: Your instances will launch in the US West (N. California) Region'. A 'Made with Tango.us' watermark is at the bottom right.

4. Click on Name and type desired named for instance.

EC2 > Instances > Launch an instance

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.

Search our full catalog including 1000s of application and OS images

▼ Summary

Number of instances

Software Image (AMI)
Amazon Linux ami-018d291c

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s)

Free tier

Made with Tango.us

5. Click on Quick Start

The screenshot shows the AWS EC2 'Quick Start' interface. At the top, there's a search bar and navigation links. Below it, a section titled 'Application and OS Images (Amazon Machine Image)' provides a brief description of what an AMI is. A search bar is available to find specific AMIs. The main area displays several OS options: Amazon Linux, Ubuntu, Windows, Red Hat, SUSE Linux, and a link to 'Browse more AMIs'. A specific AMI, 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type', is highlighted with a box. This AMI is described as 'Free tier eligible' and includes details like its AMI ID (ami-018d291ca9fffc002f), architecture (64-bit (x86)), and root device type (ebs). Below this, there's a 'Description' section for the selected AMI. On the right side, a 'Summary' panel shows the configuration: 1 instance, AMI 'Amazon Linux 2 Kernel 5.10 AMI...', instance type 't2.micro', and storage '1 volume(s) - 8 GiB'. A 'Free tier' callout box is present. At the bottom, there are 'Cancel' and 'Launch instance' buttons.

Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Made with Tango.us

6. From the "QuickStart" choose your OS image file

For the purpose of this tutorial, we will use Amazon Linux (Eligible for Free-Tire)

Search for services, features, blogs, docs, and more [Option+S]

launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

Free tier eligible

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220805.0 x86_64 HVM gp2

Architecture

AMI ID

64-bit (x86)

Verified provider

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI... [read more](#)

ami-018d291ca9ffc002f

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 not included in the 750 hours

Cancel Launch instance

Instance type [Info](#)

7. Click on Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type(Free tire eligible)

8. Select AMI (Amazon Machine Image) type

Click on "Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type" (Free Tire Eligible)

Search for services, features, blogs, docs, and more [Option+S]

launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q |

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

Free tier eligible

Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type

Free tier eligible

Amazon Linux 2 LTS with SQL Server 2017 Standard

Amazon Linux 2 with .NET 6, PowerShell, Mono, and MATE Desktop Environment

Amazon Linux 2 LTS with SQL Server 2019 Standard

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

Free tier eligible

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220805.0 x86_64 HVM gp2

Architecture

AMI ID

64-bit (x86)

Verified provider

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI... [read more](#)

ami-018d291ca9ffc002f

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 not included in the 750 hours

Cancel Launch instance

Instance type [Info](#)

9. Select Desired Instance Type

The instance type is the hardware configuration of the EC2 instance. Click on t2.micro (Free Tire eligible)

Architecture: 64-bit (x86) AMI ID: ami-018d291ca9ffc002f Verified provider

Instance type [Info](#)

Instance type

- t2.micro Family: t2 1 vCPU 1 GiB Memory Free tier eligible
- t1.micro Family: t1 1 vCPU 0.612 GiB Memory Free tier eligible
- t2.nano Family: t2 1 vCPU 0.5 GiB Memory Free tier eligible
- t2.micro Family: t2 1 vCPU 1 GiB Memory Free tier eligible
- t2.small Family: t2 1 vCPU 2 GiB Memory

Compare instance types [Create new key pair](#) [Edit](#) [Launch Instance](#)

Selected key pair before you launch

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI... [read more](#)

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, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel [Launch Instance](#)

Made with Tango.us

10. Click on Create new key pair (If not created initially)

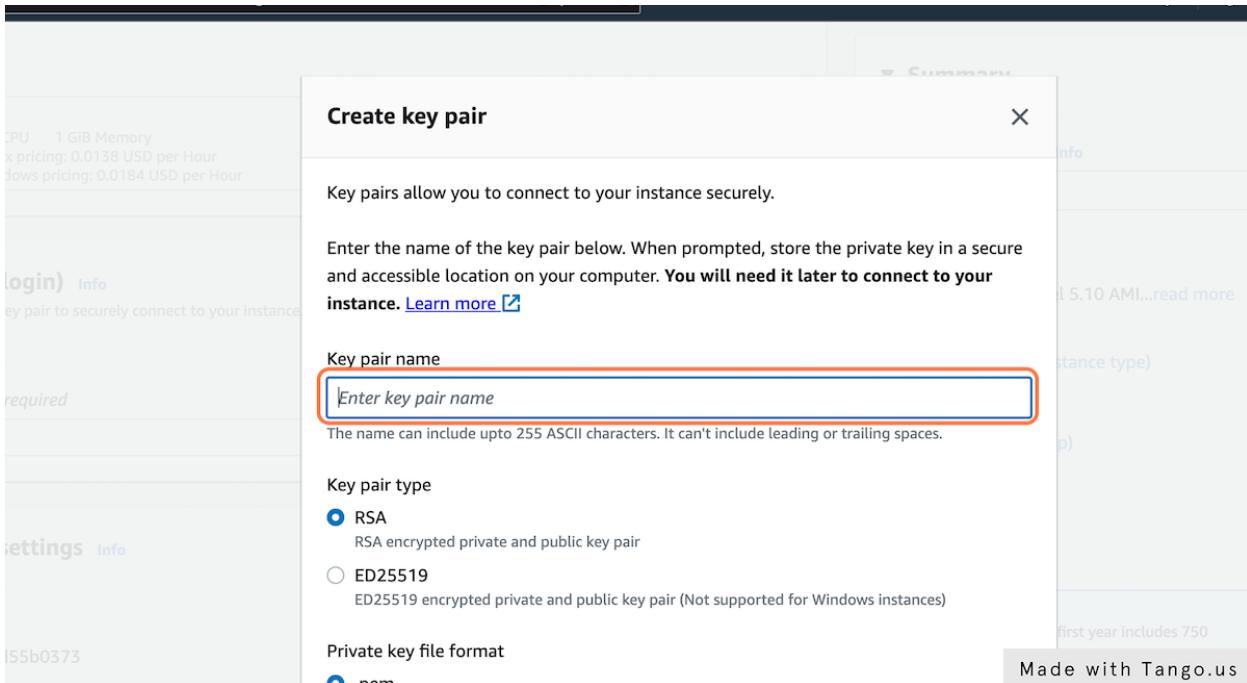
that you have access to the selected key pair before you launch

[Create new key pair](#)

Software |
Amazon L
ami-018d291ca9ffc002f
Virtual ser
t2.micro
Firewall (s
New secur
Storage (v
Edit

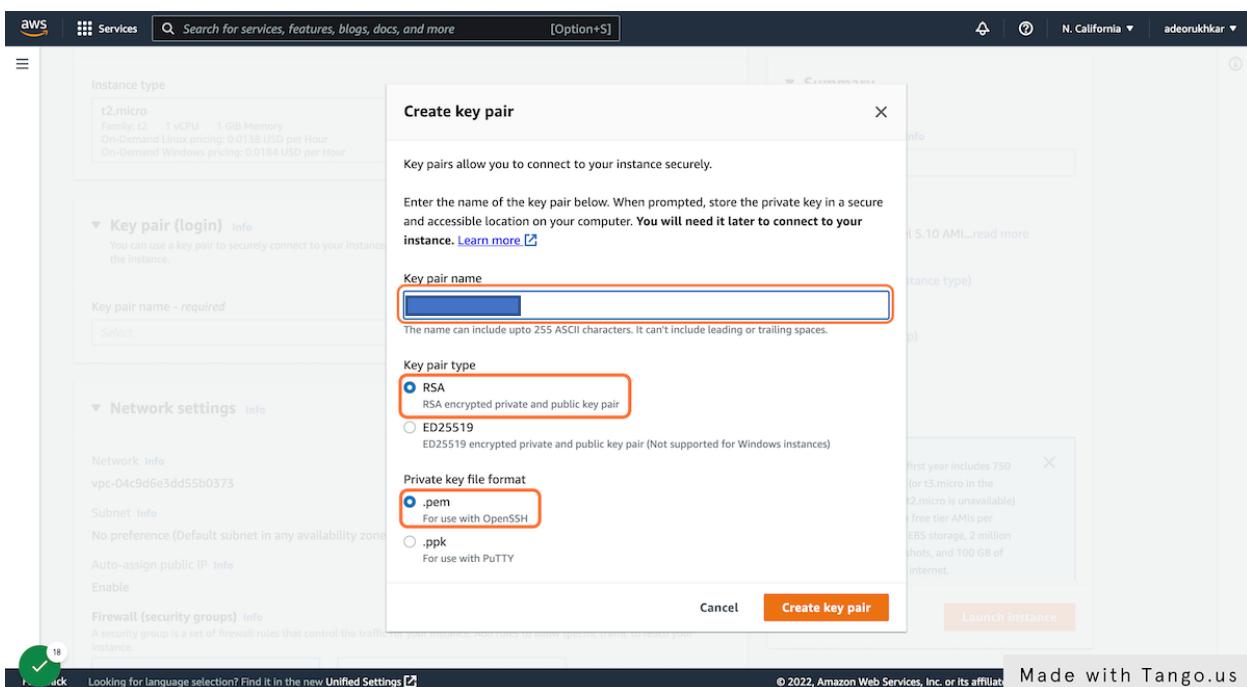
Made with Tango.us

11. Click on Key pair name to type the desired name for Key pair



12. Type the desired name

Let the Key pair type be RSA (by default) and "Private key file format" be .pem (default)



13. Click on Create key pair

nd public key pair (Not supported for Windows instances)

first year includes 750 (or t3.micro in the first year) free tier AMIs per month, 100 GB of EBS storage, 2 million API calls, 100 snapshots, and 100 GB of data transfer to the internet.

Cancel

Create key pair

Launch ins

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved. Made with Tango.us

14. The encrypted key pair will be downloaded on your local machine

15. Next, navigate to the Firewall (security groups)

16. Click on Firewall (security groups) Info

Hit “Create security group” if it has not been created before creating the instance.

vpc-04c9abesdadcc5b05/5

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable

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.

Create security group Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

Allow SSH traffic from
Helps you connect to your instance

Anywhere
0.0.0.0/0

Made with Tango.us

17. Edit the rules for the security group created(Can be done later to)

0.0.0.0/0 is not usually safe and recommended.
Using it just for the assignment

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.

Allow SSH traffic from
Helps you connect to your instance Anywhere 0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage Info Advanced

1x 8 GiB gp2 Root volume

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

Add new volume

Number of instances Info 1

Software Image (AMI) Amazon Linux 2 Kernel 5.10 AMI...read more ami-018d291ca9ffcc02f

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, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance

© 2022, Amazon Web Services, Inc. or its affiliates Made with Tango.us

18. Let other settings be default (Just for the assignment)

0.0.0.0/0 is not usually safe and recommended.
Using it just for the assignment

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.

Allow SSH traffic from
Helps you connect to your instance Anywhere 0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage Info Advanced

1x 8 GiB gp2 Root volume

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

Add new volume

Number of instances Info 1

Software Image (AMI) Amazon Linux 2 Kernel 5.10 AMI...read more ami-018d291ca9ffcc02f

Virtual server type (instance type) t2.micro

Firewall (security group) New security group

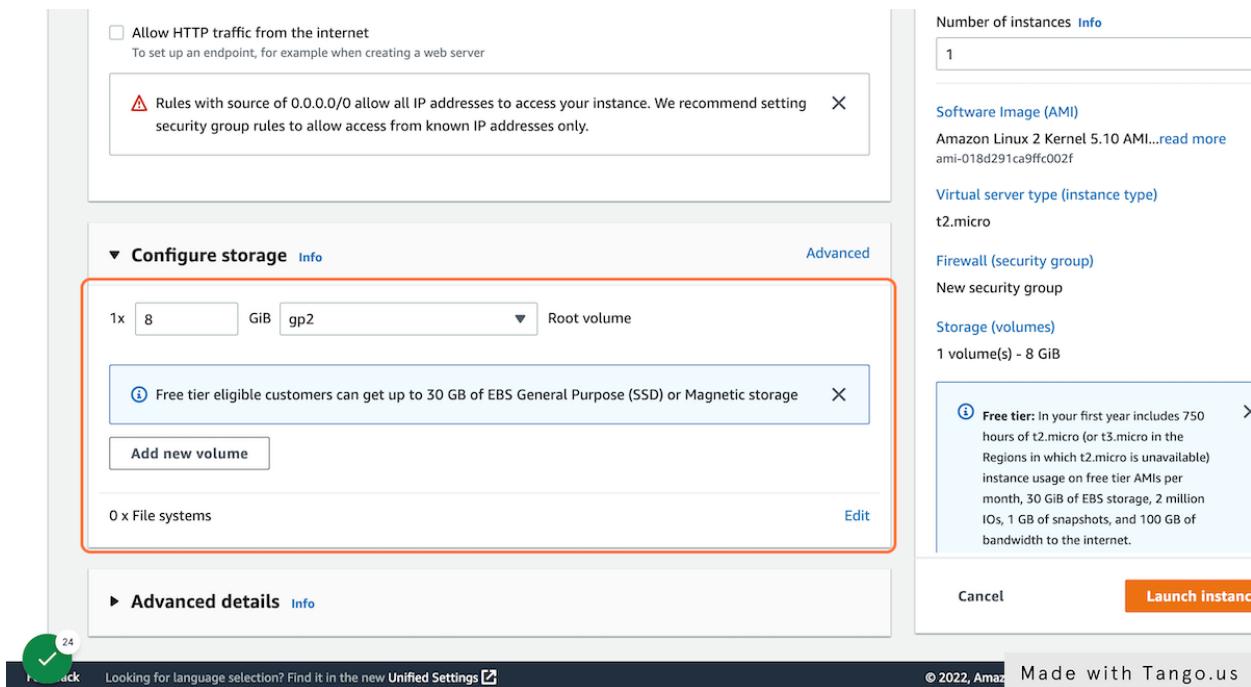
Storage (volumes) 1 volume(s) - 8 GiB

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, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

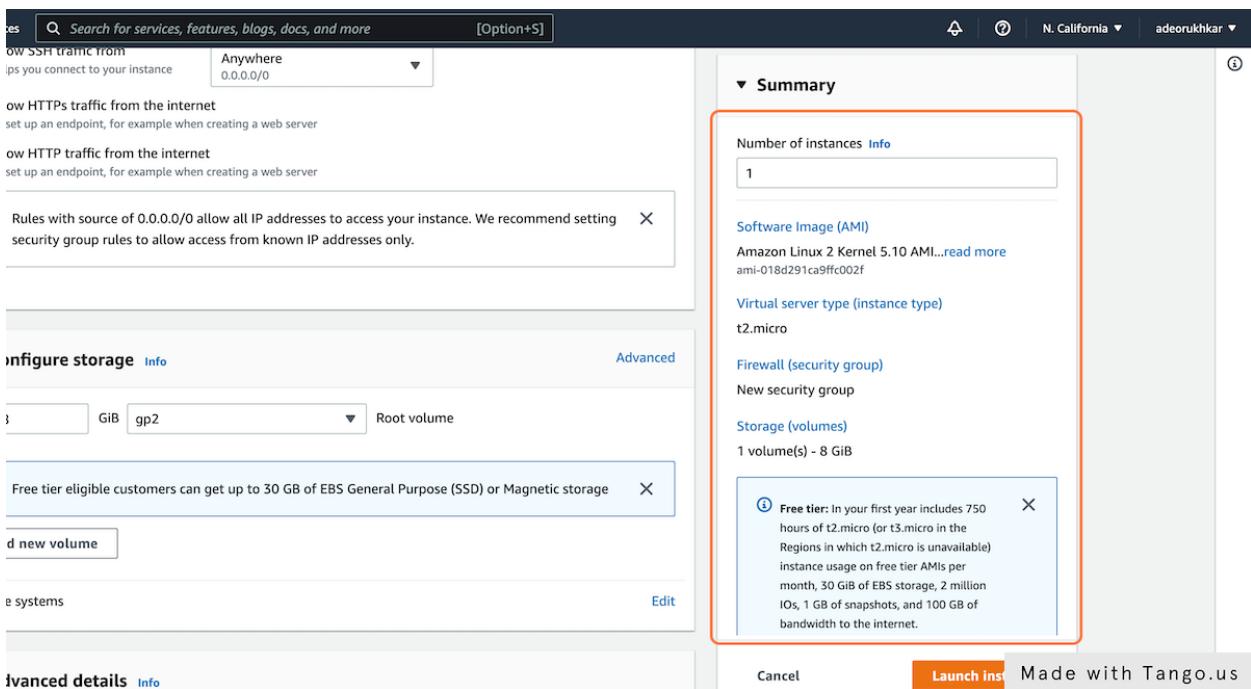
Cancel Launch instance

© 2022, Amazon Web Services, Inc. or its affiliates Made with Tango.us

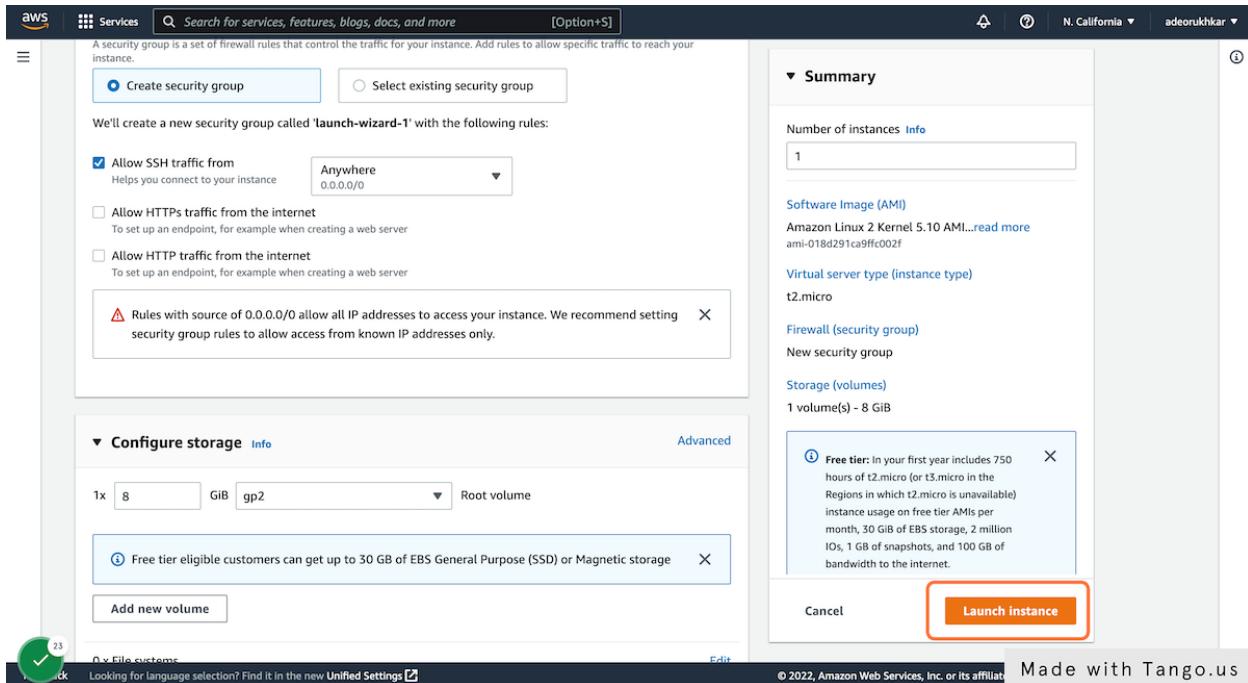
19. Default settings for "Configure storage"



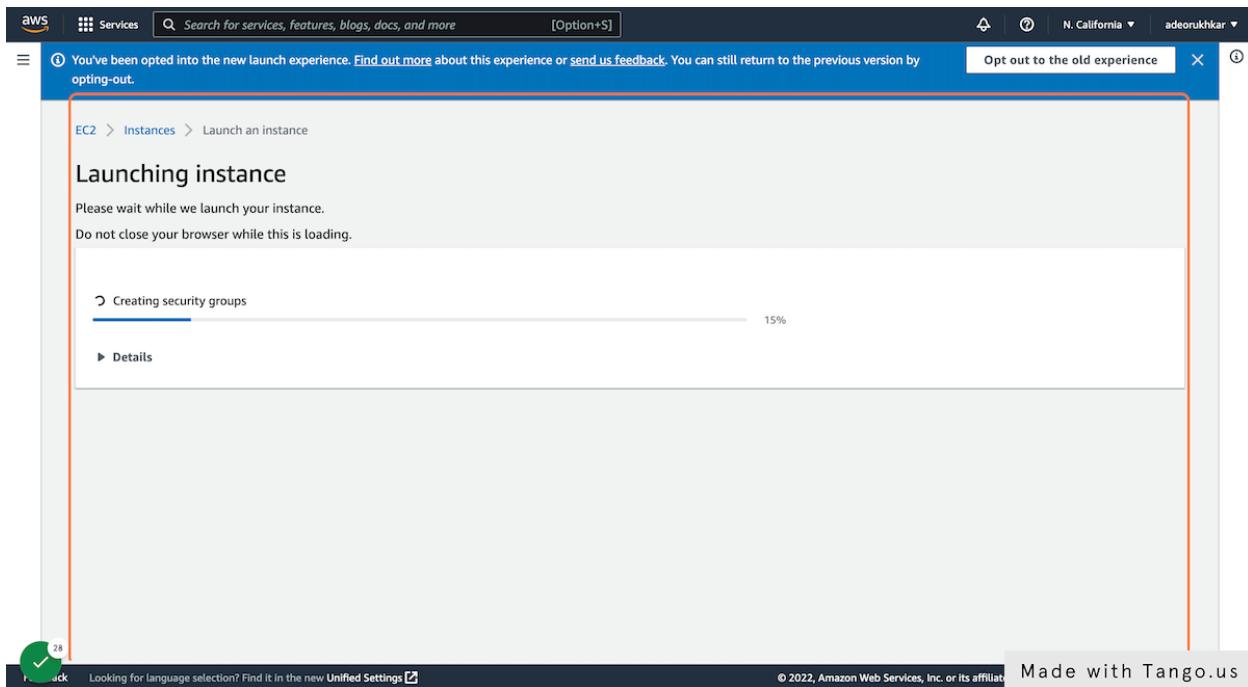
20. A summary of the instance that will be created



21. After reviewing the instance settings, hit "Launch Instance"



22. A progress bar will show the progress of the instance creation process



23. Once the instance is created, a "Success" command will prompt

The screenshot shows the AWS EC2 Instances Launch an instance page. At the top, there's a success message: "Success" and "Successfully initiated launch of instance (i-00d5b3c9ca5b3e689)". Below this, there's a "Launch log" section which is currently collapsed. The "Next Steps" section includes links to "Get notified of estimated charges", "How to connect to your instance", and "View more resources to get you started". At the bottom right, there's a "View all instances" button.

24. Click on Launch log

This can be done to check the detailed instance creation process

The screenshot shows the same AWS EC2 Instances Launch an instance page as before, but the "Launch log" section is now expanded. It displays the following log entries:

Initializing requests	Succeeded
Creating security groups	Succeeded
Creating security group rules	Succeeded
Launch initiation	Succeeded

The rest of the page, including the "Next Steps" section, remains the same.

25. Click on "View all Instances" to see all the history of EC2 instances created

Success
Successfully initiated launch of instance (i-00d5b3c9ca5b3e689)

[Launch log](#)

Next Steps

Get notified of estimated charges
[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier)

How to connect to your instance
Your instance is launching and it might be a few minutes until it is in the running state, when it will be ready for you to use
Click [View Instances](#) to monitor your instance's status. Once your instance is in the 'running' state, you can connect to it from the Instances screen. Find out [how to connect to your instance](#)

[View more resources to get you started](#)

[View all instances](#)

26. A dashboard of all EC2 instances

S

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/>	i-00d5b3c9ca5b3e689	Running	t2.micro	Initializing	No alarms	us-west-1c

Instance summary

Instance state

Instance:

Details			Security	Networking	Storage	Status checks	Monitoring	Tags
Instance summary Info								
Instance ID	Public IPv4 address		Private IPv4 addresses					
i-00d5b3c9ca5b3e689 (ec2_MSIS_Aakash_Dev)	open address		172.31.29.198					
IPv6 address	Instance state		Public IPv4 DNS					
-	Running		1.compute.amazonaws.com open address					
Hostname type	Private IP DNS name (IPv4 only)		Elastic IP addresses					
IP name: ip-172-31-29-198.us-west-1.compute.internal	ip-172-31-29-198.us-west-1.compute.internal		-					
Answer private resource DNS name	Instance type		AWS Compute Optimizer finding					
IPv4 (A)	t2.micro		Opt-in to AWS Compute Optimizer for recommendations.					
Auto-assigned IP address	VPC ID							
54.153.49.126 [Public IP]	vpc-04c9d6e3dd55b0373							

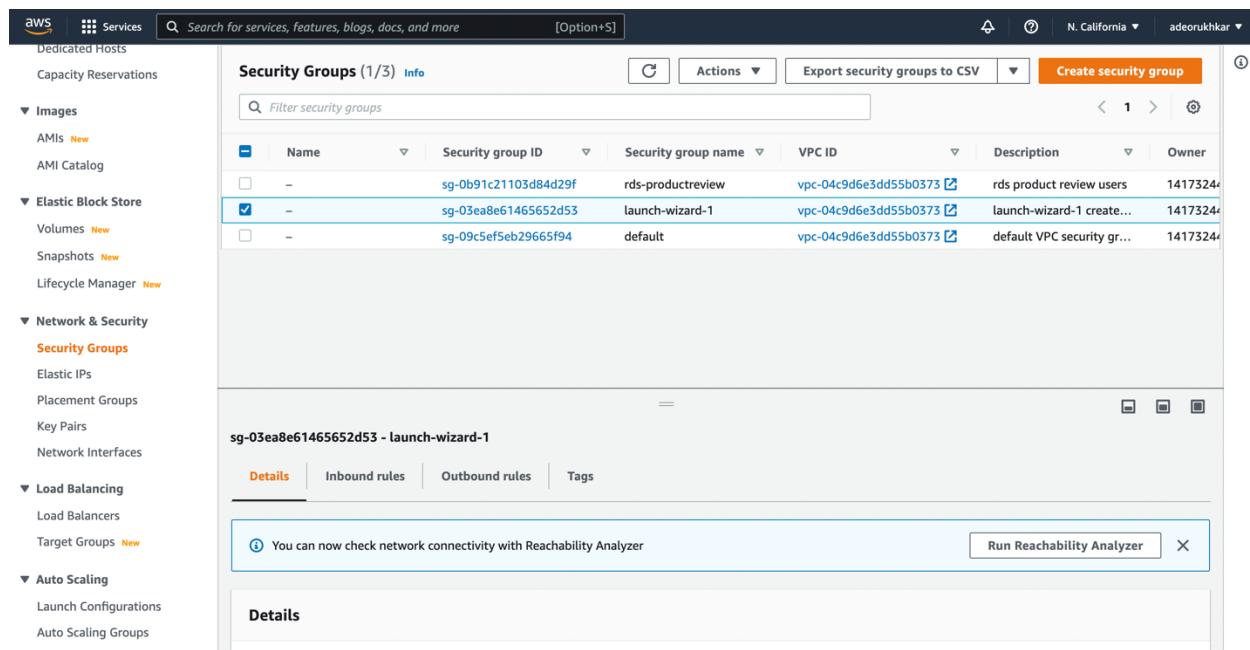
Once the instance is created, there are several ways to connect to it:

- **Linux or MacOS**
 - SSH Client
 - EC2 Instance Connect
 - AWS System Manager Session Manager
- **Windows OS**
 - PuTTY
 - SSH Client
 - AWS System Manager Session Manager
 - Windows Subsystem for Linux

For this assignment, I will be using the SSH Client method to access the EC2 instance

Before we try and connect, we need to make sure our EC2 will accept any connections

1. Within EC2 go to security groups and hit the checkbox of the security group tied to our instance. Here, “launch-wizard-1”



The screenshot shows the AWS Management Console with the Services menu open. Under Network & Security, the Security Groups option is selected. The main pane displays a list of security groups with the following details:

Name	Security group ID	VPC ID	Description	Owner
-	sg-0b91c21103d84d29f	vpc-04c9d6e3dd55b0373	rds-productreview	rds product review users 1417324
<input checked="" type="checkbox"/> -	sg-03ea8e61465652d53	vpc-04c9d6e3dd55b0373	launch-wizard-1	launch-wizard-1 create... 1417324
-	sg-09c5ef5eb29665f94	vpc-04c9d6e3dd55b0373	default	default VPC security gr... 1417324

Below the table, the details for the selected security group, "sg-03ea8e61465652d53 - launch-wizard-1", are shown. The Inbound rules tab is selected, displaying the following configuration:

- Protocol: TCP
- Port range: 22
- Source: 0.0.0.0/0

A note at the bottom says, "You can now check network connectivity with Reachability Analyzer" with a "Run Reachability Analyzer" button.

2. Here we can see the security configuration of our instance.
The connection type is “SSH”
Protocol is “TCP” (Transmission Control Protocol)
Port range is “22”
Source is “0.0.0.0/0”
As discussed earlier, this source is not recommended for a production environment

The screenshot shows the AWS EC2 Security Groups page. The security group 'sg-03ea8e61465652d53 - launch-wizard-1' is selected. The 'Details' section displays the following information:

Security group name	Security group ID	Description	VPC ID
launch-wizard-1	sg-03ea8e61465652d53	launch-wizard-1 created 2022-09-01T19:31:30.065Z	vpc-04c9d6e3dd55b0373
Owner	Inbound rules count	Outbound rules count	
141732448389	1 Permission entry	1 Permission entry	

Below the details, there are tabs for 'Inbound rules' (selected), 'Outbound rules', and 'Tags'. A message indicates: 'You can now check network connectivity with Reachability Analyzer' with buttons to 'Run Reachability Analyzer' or close it.

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range
sgr-0787e20e6d84643fa	IPv4	SSH	TCP	22	

3. Getting the public username and DNS name to connect to the instance

According to AWS's documentation, the default username for Amazon Linux 2 is: **ec2-user**
 Public IPV4: 54.153.49.126
 Public IPV4 DNS: ec2-54-153-49-126.us-west-1.compute.amazonaws.com

4. Locate the private key and set the permission

Locate the key pair file with the .pem(converted as .cer file on MacOS) extension that was downloaded during the creation of the instance. (Refer STEP 12-14 in instance creation)

In the macOS terminal, navigate to the .cer file folder

```
ITinfrastructure -- zsh -- 111x35
[(base)] aakashdeorukhkar@Aakashs-MacBook-Pro MSIS % cd ITinfrastructure
[(base)] aakashdeorukhkar@Aakashs-MacBook-Pro ITinfrastructure % ls
Architecting the Cloud_ Design Decisions for Cloud Computing Service Models ( PDFDrive ).pdf
EC2.docx
EmergingTechnologies-Omnicloud.pages
aakash_ec2_keys.cer
~$EC2.docx
(base) aakashdeorukhkar@Aakashs-MacBook-Pro ITinfrastructure %
```

5. Since we are using the SSH client on a macOS computer to connect to the Linux instance, we use the following command to set the permissions of the private key file so that only we can read it.

```
>> aakashdeorukhkar@Aakashs-MacBook-Pro ITinfrastructure % chmod 400
aakash_ec2_keys.cer
```

- In a terminal window of the local machine, we use the **ssh** command to connect to the instance. We do this by specifying the path and file name of the private key (.cer in our case), the username for your instance, and the public DNS name or IPv6 address for your instance.

Command for us will be as follows:

Template:

ssh -i /path/key-pair-name.pem instance-user-name@instance-public-dns-name

Replace the template placeholders with our configuration as follows:

>> ssh -i ./akash_ec2_keys.cer user_name@instance_public_dns_name

Output:

```
--|_ _--|_)  
_|| ( _/_ / Amazon Linux 2 AMI  
---\_\_---|  
  
https://aws.amazon.com/amazon-linux-2/  
3 package(s) needed for security, out of 7 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-29-198 ~]$ █
```

When we run a who am I command in the EC2 Instance:

[ec2-user@ip-172-31-29-198 ~]\$ whoami

Output: ec2-user

```
[ [REDACTED] ]$ whoami  
ec2-user  
[ [REDACTED] ]$ █
```

Use “exit” command to leave the EC2 terminal and return to local machine terminal

We are inside the EC2 instance now. Users can now transfer all the necessary files from the local machine to the Amazon Linux EC2 created.

This concludes the setting up for an EC2 instance and then connecting to it for file transfer with SSH.

Transferring a testTransfer.txt file from local machine to EC2 instance created

One way to transfer files between your local computer and a Linux instance is to use the secure copy protocol (SCP).

Template:

```
scp -i /path/key-pair-name.pem /path/my-file.txt ec2-user@instance-public-dns-name:path/
```

Replace the values with the relevant values for our instance

```
>> scp -i ./akash_ec2_keys.cer ./testTransfer.txt user_name@instance-public-dns-name:path/
```

We are keeping the “path/” field empty to store the transferring file in root directory(Not usually recommended)

```
testTransfer.txt                                         100%   90    4.1KB/s  00:00  
(base) aakashdeorukhkar@Aakashs-MacBook-Pro ITinfrastructure %
```

The file has successfully transferred to the Amazon Linux system:

```
Last login: Thu Sep  1 21:26:58 2022 from c-73-162-223-76.hsd1.ca.comcast.net  
--| ( --|_ )  
_| ( _ /     Amazon Linux 2 AMI  
---| \___| ___|  
  
https://aws.amazon.com/amazon-linux-2/  
3 package(s) needed for security, out of 7 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-29-198 ~]$ exitls  
-bash: exitls: command not found  
[|] $ ls  
[|] $ pwd  
/home/ec2-user  
[|] $ ls  
testTransfer.txt  
[|] $
```