

If you are new to Amazon Web Services, this simple guide will be a great tool to help you learn how to launch EC2 instances into the cloud. EC2 instances are probably one of the most common and most used services in the AWS Cloud. EC2 instances are virtual machines and servers which customers can easily request and provision in the cloud.

There are many OS that can be used, such as Ubuntu, Windows, Linux and more. Instances also come in many types, with differing prices, compute power, RAM, processors and other options in order to best suit your needs.

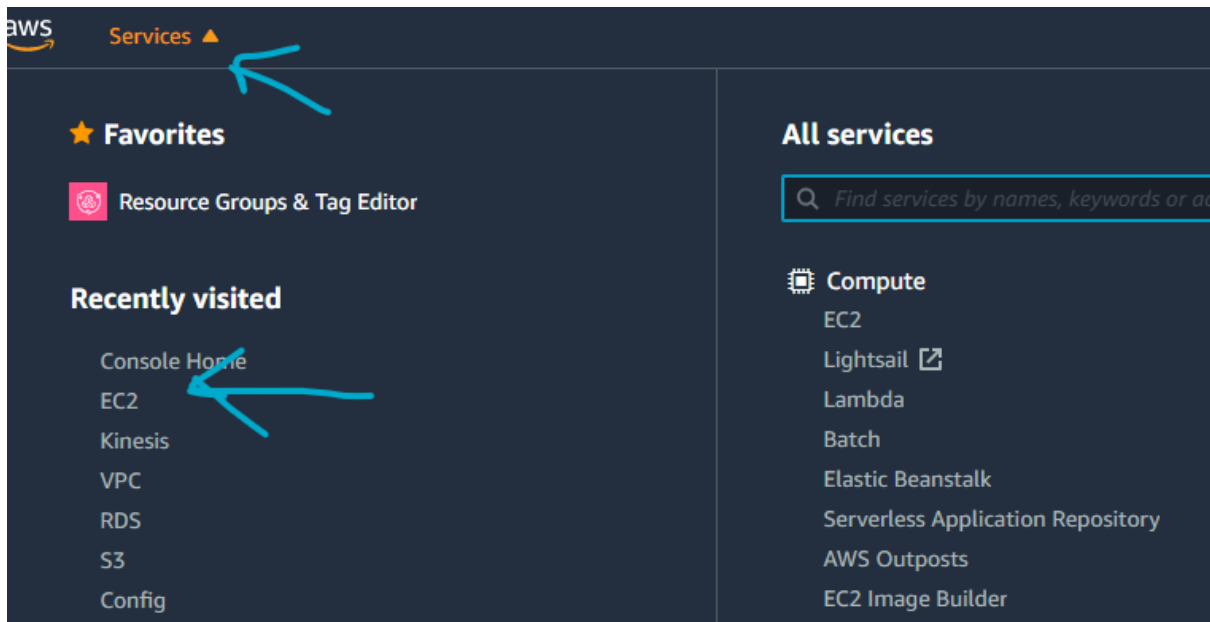
You can use EC2 instances to host websites, web applications, applications, services and pretty much anything that needs computing power. Here, I will cover 3 ways to easily and quickly launch EC2 instances into the cloud.

AWS Management Console

The AWS Management Console provides a web-based GUI for you to easily build and add your cloud services easily. This is one of the most common and easiest ways to launch an EC2 instance, due to a GUI being provided for your use.

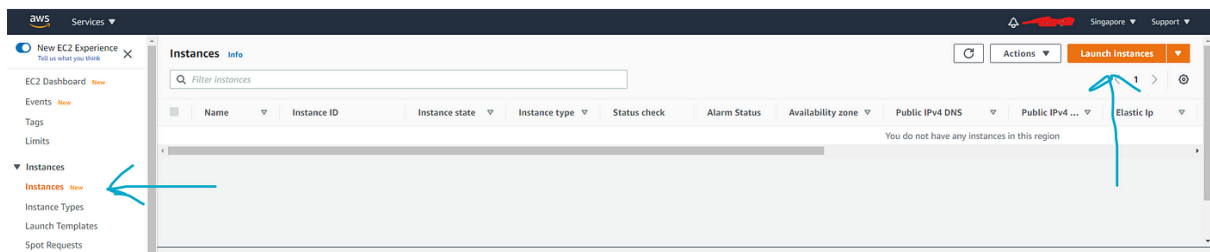
Sign in to the AWS Management Console.

Click on “**Services**” in the top left corner and click “**EC2**”.



Services -> EC2

Select “**Instances**” in the left bar and click “**Launch an Instance**” on the right side of the screen.



Instances -> Launch an Instance

For the next steps, fill in the following information for this example and click next as required.

- **Choose an AMI:** Amazon Linux 2
- **Instance Type:** t2.micro
- **Instance Details:** Click Next

- **Storage:** Click Next
- **Tags:** Click Next
- **Security Group:** Click Review + Launch

At this point, you should see the review screen as seen below. You may ignore the warning for now. Click **“Launch”**.

Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, launch-wizard-3, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

▼ AMI Details [Edit AMI](#)

Free tier eligible **Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-015a6758451df3cb9**
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

▼ Security Groups [Edit security groups](#)

Security group name launch-wizard-3
Description launch-wizard-3 created 2020-10-25T20:02:57.881+08:00

Type (i)	Protocol (i)	Port Range (i)	Source (i)	Description (i)
SSH	TCP	22	0.0.0.0/0	

▶ Instance Details [Edit instance details](#)

▶ Storage [Edit storage](#)

▶ Tags [Edit tags](#)

[Cancel](#) [Previous](#) [Launch](#)

Choose **“Proceed without a key pair”** for this example and tick the box to acknowledge. Click **“Launch Instances”**.

Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

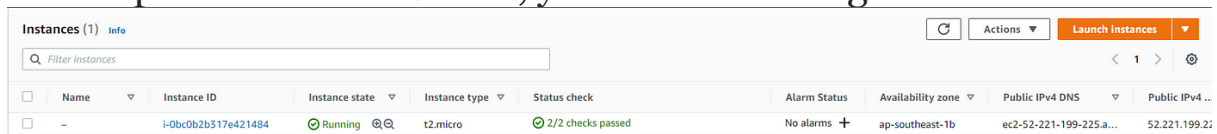
Proceed without a key pair ▼

☒ I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI.

[Cancel](#) [Launch Instances](#)

You will see that your instances are now launching. Click View instances to go back to the Instances tab. Give the instance some time to boot.

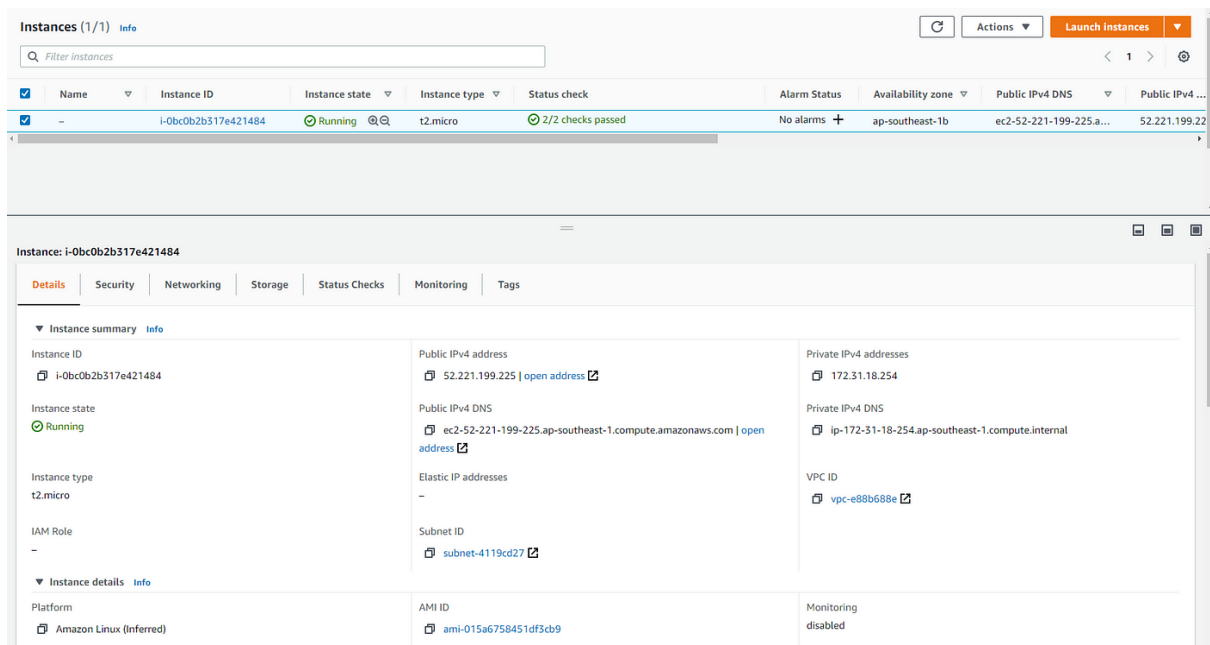
You should now see that an EC2 instance is running and initializing status checks. When the status check completes and shows “2/2 checks passed” as seen below, you can start using the instance.



Instances (1) Info									
Filter instances									
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone	Public IPv4 DNS	Public IPv4 ...
<input type="checkbox"/>	-	i-0bc0b2b317e421484	Running	t2.micro	2/2 checks passed	No alarms +	ap-southeast-1b	ec2-52-221-199-225.a...	52.221.199.22

Status Checks Passed and Instance is ready to use

When you click on the check box on the left of the instance, you can see information about the instance you just launched. This includes Public IP, networking information, status checks, tags, monitoring details and more.



Instances (1/1) Info									
Filter instances									
<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone	Public IPv4 DNS	Public IPv4 ...
<input checked="" type="checkbox"/>	-	i-0bc0b2b317e421484	Running	t2.micro	2/2 checks passed	No alarms +	ap-southeast-1b	ec2-52-221-199-225.a...	52.221.199.22

Instance: i-0bc0b2b317e421484		
Details	Security	Networking
▼ Instance summary Info		
Instance ID i-0bc0b2b317e421484	Public IPv4 address 52.221.199.225 open address	Private IPv4 addresses 172.31.18.254
Instance state Running	Public IPv4 DNS ec2-52-221-199-225.ap-southeast-1.compute.amazonaws.com open address	Private IPv4 DNS ip-172-31-18-254.ap-southeast-1.compute.internal
Instance type t2.micro	Elastic IP addresses -	VPC ID vpc-e88b688e
IAM Role -	Subnet ID subnet-4119cd27	
▼ Instance details Info		
Platform Amazon Linux (Inferred)	AMI ID ami-015a6758451df3cb9	Monitoring disabled

Instance Details

Congratulations, you have launched your EC2 instance using the AWS Management Console!

AWS CLI (Command Line Interface)

If you prefer typing into a terminal instead of using a GUI, you can choose to use the AWS CLI. AWS CLI is actually preferred for running scripts and launching multiple EC2 instances easily without having to go through the setup wizard on the Management Console multiple times. However, you do need to have AWS CLI configured and set up. If you haven't done that, you can refer to my guide [here](#).

If you do already have AWS CLI installed and configured, you are ready to go! It is a much shorter process of launching an EC2 instance using AWS CLI compared to the AWS Management Console, especially if you are launching multiple instances.

To launch instances using AWS CLI, you require certain information, such as `ami-id` you wish to use, the `sg-id` security group id you want to use, and the `subnetid` that you are launching the EC2 instance into. You can get this information from the AWS Management Console or with other AWS CLI commands.

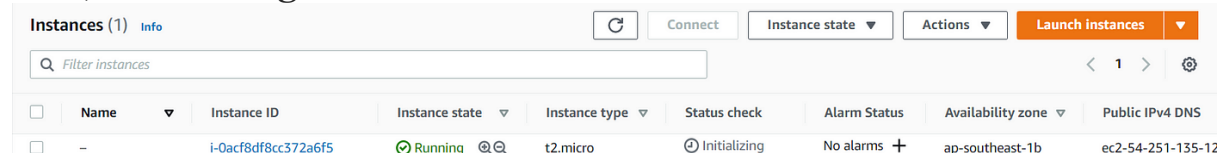
For example, I will use the following information:

- `ami-id`: `ami-od9dacead6e617656` (Ubuntu image by AWS)
- `sg-id`: `sg-od369779` (My default security group id)
- `subnet-id`: `subnet-4119cd27` (Default subnet in ap-southeast-1b)

There command and output we will get is:

```
$ aws ec2 run-instances --image-id ami-0d9dacead6e617656 --count 1 --instance-type t2.micro --security-group-ids sg-0d369779 --subnet-id subnet-4119cd27{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-0d9dacead6e617656",
      "InstanceId": "i-0acf8df8cc372a6f5",
      "InstanceType": "t2.micro",
      "LaunchTime": "2020-11-01T10:48:10+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "ap-southeast-1b",
        "GroupName": "",
        "Tenancy": "default"
      },
      .....
    }
  ]
}
```

As you can see, it shows up on my AWS Management Console as well, confirming that the launch was successful:



Instances (1) Info								
<input type="text" value="Filter instances"/>								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone	Public IPv4 DNS
<input type="checkbox"/>	-	i-0acf8df8cc372a6f5	Running	t2.micro	Initializing	No alarms	ap-southeast-1b	ec2-54-251-135-12

Using AWS SDK

AWS provides many SDKs for the various programming languages for you to manage and access your AWS resources. Supported languages include Python, Ruby, Javascript and many more. For this tutorial, I will only be covering AWS SDK for python, but you can use whichever is your preferred language.

If you don't have python installed, you can download it [here](#). After installing python, you need to install boto3, which is the AWS SDK for Python. Run `pip install boto3` to quickly install it.

Once all that is set up, we can use this Python script to launch an EC2 instance:

```
import boto3
ec2 = boto3.resource('ec2') # create a new EC2 instance
instances = ec2.create_instances(ImageId='ami-0d9dacead6e617656', MinCount=1, MaxCount=1, InstanceType='t2.micro')
```

Conclusion

AWS provides many different ways to help you easily launch EC2 instances in the cloud regardless of your requirement. Need to quickly launch 1 instance for testing? I would use the AWS Management Console. However, AWS CLI and AWS SDK would be preferred for batch launches and automation scripts.

That is just my preference, use whatever is best for you!

The above examples are just very quick tutorials on how to launch EC2 instances in different ways, there are more parameters and factors that you can add as well, based on your needs. For example, you could use user data scripts or the `keypair` parameter for AWS CLI or AWS SDK.

Happy building!