

When we launch the Instance In, AWS cloud. OS will know about what the resource they have inside (RAM, CPU, Hard-Disk, Network), but where the instance is physically available. OS will don't know about this. This information stores inside the **meta-data**.

For **Retrieving** the **meta-data**, we need to **run** this **command** inside the AWS instance
curl <http://169.254.169.254/latest/meta-data/>

```
[root@ip-172-31-39-23 ~]# curl http://169.254.169.254/latest/meta-data/  
ami-id  
ami-launch-index  
ami-manifest-path  
block-device-mapping/  
events/  
hibernation/  
hostname  
identity-credentials/  
instance-action  
instance-id  
instance-life-cycle  
instance-type  
local-hostname  
local-ipv4  
mac  
managed-ssh-keys/  
metrics/  
network/  
placement/  
profile  
public-hostname  
public-ipv4  
public-keys/
```

For Example:-

- I Want to know my Amazon machine Image Id we can see this by

- Command –

- **curl** <http://169.254.169.254/latest/meta-data/ami-id>

```
[root@ip-172-31-39-23 ~]# curl http://169.254.169.254/latest/meta-data/ami-id  
ami-079b5e5b3971bd10d[root@ip-172-31-39-23 ~]# █
```

- I Want to know my Public IP; we can see this by

- Command

- **curl** <http://169.254.169.254/latest/meta-data/public-ipv4>

```
[root@ip-172-31-39-23 ~]# curl http://169.254.169.254/latest/meta-data/public-ipv4  
3.108.228.189[root@ip-172-31-39-23 ~]# █
```

By this way we can access the meta data of our Instance and we can do whatever the things we want.

Now Suppose we need to setup a Webserver, but the challenge is how we can launch the webserver without having the SHELL and also we need to see the Public Ip inside the webpage?

For this there is a concept of **cloud-init**

When the Operating System is started at the boot time whatever we pass inside the cloud-init it will always run.

So how we can do this?

Answer :- For this we need User-data.

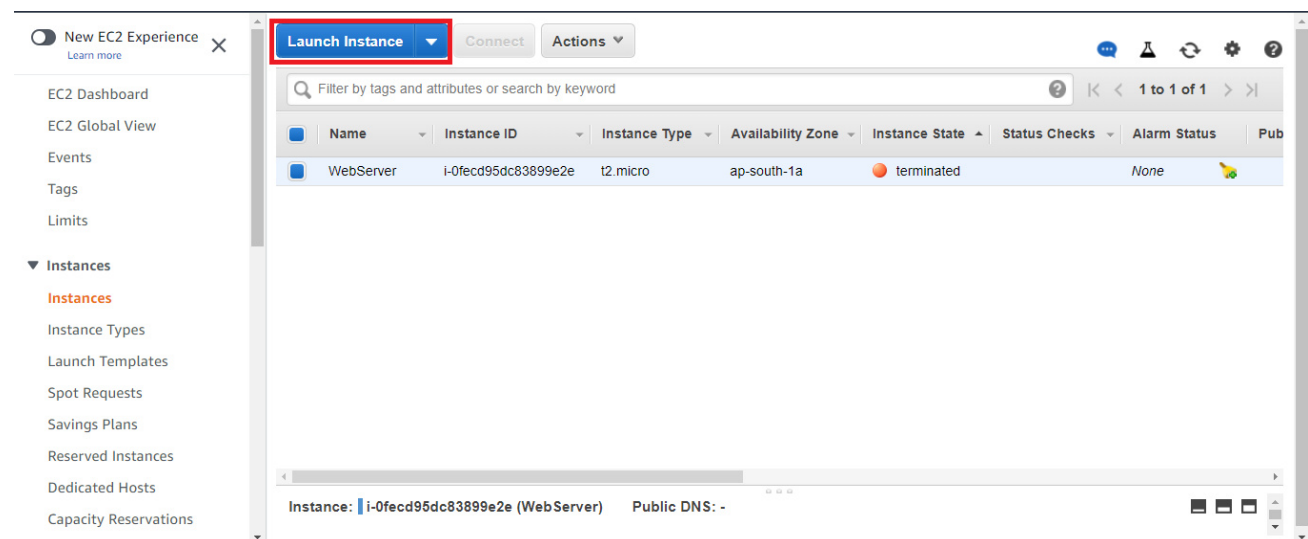
Now the question is What is User-data? And how it works?

User-data:- when the OS is booting at that time, we want to perform some instruction to the OS that time we send the instruction with the help of User-data.

And inside the instruction we are passing our command that will be executed when the cloud-init will run. So, it means OS will contact to cloud-init and cloud-init behind the scene check if any instructions are given by user (user-data) or not. If it is given then it will execute those instructions and whatever the things are written inside the instruction it will executed. If the instructions are not given then it normally boots the OS.

So, lets see this practical.

Step 1 : Click on Launch Instance



The screenshot displays the AWS Management Console interface for the EC2 service. On the left sidebar, the 'Instances' section is expanded, and the 'Launch Instance' button is highlighted with a red rectangular box. The main content area shows a table of EC2 instances. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. A single instance named 'WebServer' is listed with Instance ID 'i-0fec95dc83899e2e', Instance Type 't2.micro', Availability Zone 'ap-south-1a', and Instance State 'terminated'. The status checks are 'None' and the public DNS is not visible. Below the table, there is a summary bar for the selected instance 'i-0fec95dc83899e2e (WebServer)' showing its public DNS as '-'. The top navigation bar includes the 'Launch Instance' button, a 'Connect' button, and an 'Actions' dropdown menu.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
WebServer	i-0fec95dc83899e2e	t2.micro	ap-south-1a	terminated		None	

Step 2:- Choose the AMI, In My case, I am choosing the Free tier

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Choose an Amazon Machine Image (AMI)

Cancel and Exit

Search by Systems Manager parameter


Quick Start 1 to 46 of 46 AMIs

My AMIs

AWS Marketplace

Community AMIs


☐ Free tier only ⓘ

**Amazon Linux**
Free tier eligible

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-079b5e5b3971bd10d (64-bit x86) / ami-0b5794cc5d751f003 (64-bit Arm)
Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 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. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

☒ 64-bit (x86)
☐ 64-bit (Arm)

Select

**Amazon Linux**
Free tier eligible

Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-08f3712c8ca5af75e (64-bit x86) / ami-0494f7473bf694265 (64-bit Arm)
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. This AMI is

☒ 64-bit (x86)
☐ 64-bit (Arm)

Select

Step 3 :- Here I am choosing t2.micro which comes under free tier and click on Next

Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families Current generation Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Step 4:- In this screen go to the bottom and you find a section called user data

Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances ⓘ 1 Launch into Auto Scaling Group ⓘ

Purchasing option ⓘ ☐ Request Spot instances

Network ⓘ vpc-04b9b23f7770065e6 (default) Create new VPC

Subnet ⓘ No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP ⓘ Use subnet setting (Enable)

Hostname type ⓘ Use subnet setting (IP name)

DNS Hostname ⓘ ☒ Enable IP name IPv4 (A record) DNS requests
☒ Enable resource-based IPv4 (A record) DNS requests

Advanced Details

Enclave ⓘ ☐ Enable

Metadata accessible ⓘ Enabled

Metadata version ⓘ V1 and V2 (token optional)

Metadata token response hop limit ⓘ 1

Allow tags in metadata ⓘ Disabled

User data ⓘ ☒ As text ☐ As file ☐ Input is already base64 encoded

(Optional)

Cancel Previous Review and Launch Next: Add Storage

Here in user data we are passing our command for setting up webserver as instruction

Step 3: Configure Instance Details

Metadata version ⓘ V1 and V2 (token optional)

Metadata token response hop limit ⓘ 1

Allow tags in metadata ⓘ Enabled

⚠ Allow tags in metadata enabled

Any instance tags that you add will be available via instance metadata. To prevent this, under **Advanced details**, choose **Disable** from **Allow tags in metadata**.

User data ⓘ ☒ As text ☐ As file ☐ Input is already base64 encoded

```
#!/bin/bash
yum install httpd -y
echo Hello, This is Harsh > /var/www/html/index.html
IP=$(curl http://169.254.169.254/latest/meta-data/public-ipv4)
echo My Public IP is $IP >> /var/www/html/index.html
systemctl start httpd
systemctl enable httpd
```

Cancel Previous Review and Launch Next: Add Storage

Step 5:- Here I am adding storage of 8 GB only and click on next

Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encryption ⓘ
Root	/dev/xvda	snap-003a9e0039d94c184	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypt

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch Next: Add Tags

Step 6:- Give the Tag names to OS and click Next

Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances ⁱ	Volumes ⁱ	Network Interfaces ⁱ	
<input type="text" value="Name"/>	<input type="text" value="WebOS"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="✕"/>
<input type="button" value="Add another tag"/> (Up to 50 tags maximum)					

[Cancel](#)

[Previous](#)

[Review and Launch](#)

[Next: Configure Security Group](#)

Step 7:- Here I don't want to connect to my OS as I mentioned above and directly launch the webserver without going inside the OS. And here I use HTTP protocol to connect my OS from outside world that's why I choose source Anywhere. And click on next

Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group

☐ Select an existing security group

Security group name:

Description:

Type ⁱ	Protocol ⁱ	Port Range ⁱ	Source ⁱ	Description ⁱ	
<input type="text" value="HTTP"/>	<input type="text" value="TCP"/>	<input type="text" value="80"/>	<input type="text" value="Anywhere"/> 0.0.0.0/0, ::/0	<input type="text" value="e.g. SSH for Admin Desktop"/>	<input type="button" value="✕"/>
<input type="button" value="Add Rule"/>					



Warning

You will not be able to connect to this instance as the AMI requires port(s) 22 to be open in order to have access. Your current security group doesn't have port(s) 22 open.

[Cancel](#)

[Previous](#)

[Review and Launch](#)

Step 8:- Here I am choosing the Existing Key Pair and Click on Launch Instance

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. Amazon EC2 supports ED25519 and RSA key pair types.

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](#).

Choose an existing key pair

Select a key pair

AWS_CSA_Training | RSA

☒ I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

Cancel

Launch Instances

Now the Instance is Launch Successfully. Copy the Public Ip and See the Webpage It is Launched

Launch Instance | Connect | Actions

search: i-0de3633879b2675f6 | Add filter

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
WebOS	i-0de3633879b2675f6	t2.micro	ap-south-1b	running	Initializing	None	ec2-3-109-208-179.ap-...

Instance: i-0de3633879b2675f6 (WebOS) | Public DNS: ec2-3-109-208-179.ap-south-1.compute.amazonaws.com

Description | Status Checks | Monitoring | Tags

Instance ID: i-0de3633879b2675f6 | Public DNS (IPv4): ec2-3-109-208-179.ap-south-1.compute.amazonaws.com | Instance state: running | IPv4 Public IP: 3.109.208.179

Instances | EC2 Management Cor | 3.109.208.179

Not secure | 3.109.208.179

Hello, This is Harsh My Public IP is 3.109.208.179

Our Webserver is successfully launched without opening the OS
Thank You