

# Create VPC:

The screenshot shows the AWS VPC dashboard with the title "Your VPCs (2) Info". On the left, there's a sidebar with "VPC dashboard" and "Virtual private cloud" sections. The main area displays a table with two rows of VPC information:

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
-	vpc-0d0ba0c809cc775dd	Available	Off	172.31.0.0/16	-
vpc-123	vpc-02c5c75f1041980e8	Available	Off	10.0.0.0/16	-

A message at the bottom says "Select a VPC above".

# Create subnets in vpc:

The screenshot shows the AWS Subnets page with the title "Subnets (4/15) Info". The main area displays a table with six rows of subnet information:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
subnet1	subnet-0972e7152c5414616	Available	vpc-02c5c75f1041980e8   vpc...	Off	10.0.128.0/1
-	subnet-023713b21bf48b13c	Available	vpc-0d0ba0c809cc775dd	Off	172.31.16.0/
subnet2	subnet-0f9916d131b040339	Available	vpc-02c5c75f1041980e8   vpc...	Off	10.0.32.0/19
RDS-Pvt-subnet-5	subnet-0f88bde61a89f6b55	Available	vpc-02c5c75f1041980e8   vpc...	Off	10.0.2.128/2
subnet-1	subnet-0ceb288ae1953c8b8	Available	vpc-02c5c75f1041980e8   vpc...	Off	10.0.64.0/18
subnet-2	subnet-02a6976d8e5231559	Available	vpc-02c5c75f1041980e8   vpc...	Off	10.0.16.0/20

## Create route table and associate subnets to route table:

The screenshot shows the AWS Route Tables list. The table has columns: Name, Route table ID, Explicit subnet associa..., Edge associations, Main, and VPC. There are four rows: rtb-09c74db23a4f9f0e3, rtb-12 (selected), rtb-0fa9b448abccffe99, and RDS-Pvt-rt. The rtb-12 row shows 4 subnets under Explicit subnet associations and No under Main. The VPC column shows vpc-02c5c75f1041980e8 | vpc-123. A modal window for rtb-000e8dae301975c64 / rtb-12 is open, showing the Details tab with fields: Route table ID (rtb-000e8dae301975c64), Main (No), Owner ID (536394435750), Explicit subnet associations (4 subnets), and Edge associations (empty).

## Create Internet Gateway and attach to VPC:

The screenshot shows the AWS Internet Gateways list. The table has columns: Name, Internet gateway ID, State, VPC ID, and Owner. There is one row: igw-12 with Internet gateway ID igw-0cc482e07ed462aa8, State Attached, VPC ID vpc-02c5c75f1041980e8 | vpc-123, and Owner 536394435750.

## Create S3 bucket :

The screenshot shows the AWS S3 General purpose buckets list. The table has columns: Name, AWS Region, and Creation date. There is one row: awsbucket-123 with AWS Region US East (N. Virginia) us-east-1 and Creation date July 25, 2025, 10:26:33 (UTC+05:30). The left sidebar shows navigation links for Amazon S3: General purpose buckets, Directory buckets, Table buckets, Vector buckets (Preview), Access Grants, Access Points (General Purpose Buckets, FSx file systems), Access Points (Directory Buckets), Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, and Storage Lens.

# Add object to s3 bucket:

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with 'Amazon S3' and 'General purpose buckets' sections. The main area is titled 'awsbucket-123 info' and shows 'Objects (1)'. A single object, 'image.jpg', is listed with details: Name: image.jpg, Type: jpg, Last modified: July 25, 2025, 10:27:16 (UTC+05:30), Size: 90.3 KB, Storage class: Standard.

# Create EFS:

The screenshot shows the AWS EFS console interface. The main area is titled 'File systems (1)' and lists a single file system named 'efs-123'. The table provides the following details:

Name	File system ID	Encrypted	Total size	Size in Standard	Size in IA	Size in Archive	Provisioned Throughput (MiB/s)
efs-123	fs-Offf390e8c72af943f	Encrypted	6.00 KiB	6.00 KiB	0 Bytes	0 Bytes	-

# Create EC2 Instance:

The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar shows navigation options like Dashboard, EC2 Global View, Events, and Instances. Under Instances, 'Instances' is selected. The main area displays a table titled 'Instances (2) Info' with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. Two instances are listed: 'ec2-123' (Instance ID: i-0419e8a01f3c6cdb1, State: Running, Type: t2.micro, Status: 2/2 checks passed, Alarm: View alarms +, Availability Zone: us-east-1a) and another unnamed instance (Instance ID: i-0d4cf7aab23ffe140, State: Running, Type: t2.micro, Status: 2/2 checks passed, Alarm: View alarms +, Availability Zone: us-east-1a). A search bar at the top allows filtering by attribute or tag.

# Connect to Ubuntu:

The screenshot shows a terminal session in the AWS CloudShell. The terminal window title is 'EC2 Instance Connect'. The session is connected to an Ubuntu 24.04.2 LTS instance (i-0419e8a01f3c6cdb1). The terminal output shows system information, including system load (0.08), memory usage (21%), and swap usage (0%). It also shows the last login details (Fri Jul 25 07:28:28 2025 from 18.206.107.29). The prompt at the bottom right indicates the session is still active.

```
ast login: Fri Jul 25 07:28:28 2025 from 18.206.107.29
buntu@ip-10-0-240-14:~$ sudo apt update
get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
get:4 http://security.ubuntu.com/ubuntu noble-security InRelease
get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [163 kB]
get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7052 B]
get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [28.4 kB]
get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
get:13 829 kB in 1s (959 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 packages can be upgraded. Run 'apt list --upgradable' to see them.
buntu@ip-10-0-240-14:~$
```

```
ubuntu@ip-10-0-240-14:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
  openssh-client openssh-server openssh-sftp-server
  upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
ubuntu@ip-10-0-240-14:~$
```

Install nginx

```
ubuntu@ip-10-0-240-14:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.24.0-2ubuntu7.4).
```

## Check status of nginx:

```
ubuntu@ip-10-0-240-14:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
  Active: active (running) since Fri 2025-07-25 06:41:02 UTC; 3h 36min ago
    Docs: man:nginx(8)
    Process: 547 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 681 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 686 (nginx)
      Tasks: 2 (limit: 1121)
     Memory: 3.0M (peak: 3.3M)
        CPU: 26ms
       CGroup: /system.slice/nginx.service
           ├─686 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
           └─687 "nginx: worker process"
Jul 25 06:41:00 ip-10-0-240-14 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Jul 25 06:41:02 ip-10-0-240-14 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
ubuntu@ip-10-0-240-14:~$
```

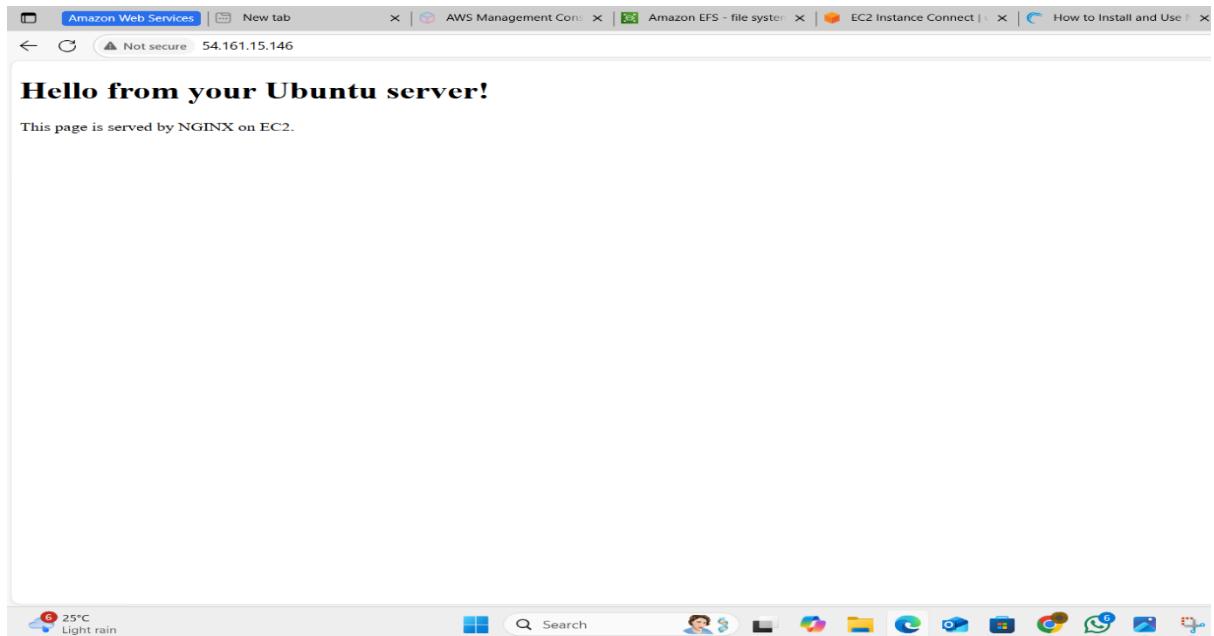
---

# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*



## Mount EFS TO EC2:

```
Service restarts being deferred:  
/etc/init.d/dbus.service  
systemctl restart getty@tty1.service  
systemctl restart networkd-dispatcher.service  
systemctl restart serial-getty@ttyS0.service  
systemctl restart systemd-logind.service  
systemctl restart unattended-upgrades.service  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-10-0-240-14:/var/www/html$ sudo mount -t efs -o tls fs-0ff390e8c72af943f:/ /var/www/html/  
mount: /var/www/html: unknown filesystem type 'efs'.  
     dmesg(1) may have more information after failed mount system call.  
ubuntu@ip-10-0-240-14:/var/www/html$ sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0ff390e8c72af943f.efs.us-east-1.amazonaws.com:/ /var/www/html/  
ubuntu@ip-10-0-240-14:/var/www/html$ df -h  
Filesystem           Size  Used Avail Use% Mounted on  
/dev/root            6.8G  2.2G  4.6G  33% /  
tmpfs               479M    0  479M   0% /dev/shm  
tmpfs               192M  912K  191M   1% /run  
tmpfs                5.0M    0  5.0M   0% /run/lock  
/dev/xvda16          881M  149M  671M  19% /boot  
/dev/xvda15          105M   6.2M  99M   6% /boot/efi  
tmpfs                96M   12K  96M   1% /run/user/1000  
fs-0ff390e8c72af943f.efs.us-east-1.amazonaws.com:/  8.0E    0  8.0E   0% /var/www/html  
ubuntu@ip-10-0-240-14:/var/www/html$
```

## Create RDS and Read Replica:

A screenshot of the AWS RDS console. The top navigation bar includes "Databases (2)", "Group resources", "Actions", "Create database", and a search bar. The main table lists two databases: "database-12" (Primary, MySQL, us-east-1a, db.t4g.micro) and "dbinstance-123" (Replica, MySQL, us-east-1d, db.t4g.micro). The table has columns for DB identifier, Status, Role, Engine, Region, and Size.

## Connect EC2 to RDS:

```
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@ip-10-0-240-14:~$ mysql -h database-12.c834eam4ykh0.us-east-1.rds.amazonaws.com -P 3306 -u admin -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 46  
Server version: 8.0.41 Source distribution  
  
Copyright (c) 2000, 2025, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
4 rows in set (0.01 sec)  
  
mysql> [REDACTED]
```

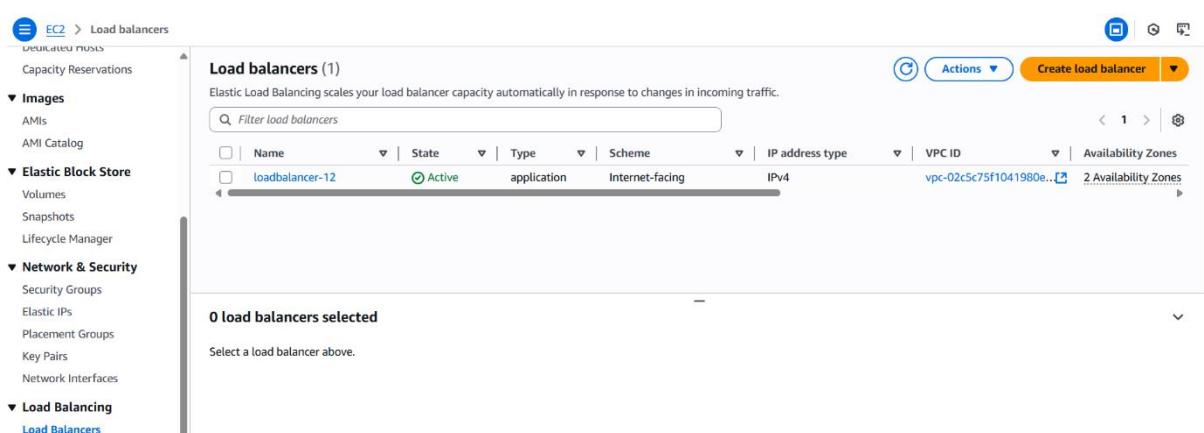
## Create AMI:



The screenshot shows the AWS Lambda console with the following details:

- Amazon Machine Images (AMIs) (1)**: Shows one item named "ami-12".
- Actions**: Includes options like "Recycle Bin", "EC2 Image Builder", and "Launch instance from AMI".
- Filters**: "Owned by me" and a search bar "Find AMI by attribute or tag".
- Table Headers**: Name, AMI name, AMI ID, Source, Owner, and Visibility.
- Data Row**: ami-12, ami-05339af24699f20aa, 536394435750/ami-12, 536394435750, Private.

## Create Load Balancer:



The screenshot shows the AWS Lambda console with the following details:

- Load balancers (1)**: Shows one item named "loadbalancer-12".
- Actions**: Includes options like "Create load balancer".
- Filters**: Filter load balancers.
- Table Headers**: Name, State, Type, Scheme, IP address type, VPC ID, and Availability Zones.
- Data Row**: loadbalancer-12, Active, application, Internet-facing, IPv4, vpc-02c5c75f1041980e.., 2 Availability Zones.

# Create Auto Scaling Group:

The screenshot shows the AWS Auto Scaling Groups console. At the top, there's a header with "Auto Scaling groups (1) Info" and a timestamp "Last updated less than a minute ago". Below the header is a search bar with placeholder text "Search your Auto Scaling groups". The main area displays a table with one row for the group "asg-12". The columns in the table are: Name, Launch template/configuration, Instances, Status, Desired capacity, Min, and Max. The "Name" column shows "asg-12", the "Launch template/configuration" column shows "template-12 | Version Default", and the "Desired capacity" column shows "1". The "Min" and "Max" columns both show "1".

# Create cloudFront:

The screenshot shows the AWS CloudFront Distributions console. On the left, there's a sidebar with navigation links for CloudFront, Distributions, Policies, Functions, Static IPs, VPC origins, SaaS, Telemetry, Reports & analytics, and more. The main area shows a distribution named "cloudfront-123" in a "Standard" configuration. The "General" tab is selected. The "Details" section shows the distribution name "cloudfront-123", the distribution domain name "d1ul00pamspds.m.cloudfront.net", the ARN "arn:aws:cloudfront::536394435750:distribution/E12ZE3SR22PKFV", and the last modified date "July 25, 2025 at 7:45:43 AM UTC". The "Settings" section includes fields for Description, Price class (set to "Use all edge locations (best performance)"), Supported HTTP versions ("HTTP/2, HTTP/1.1, HTTP/1.0"), Alternate domain names (with a "Add domain" button), Standard logging (set to "Off"), Cookie logging (set to "Off"), and Default root object. There's also a "Continuous deployment" section.