

CAPSTONE AWS PROJECT-1

Deploying a Multi-Tier Website Using AWS EC2

Description:

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front 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. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

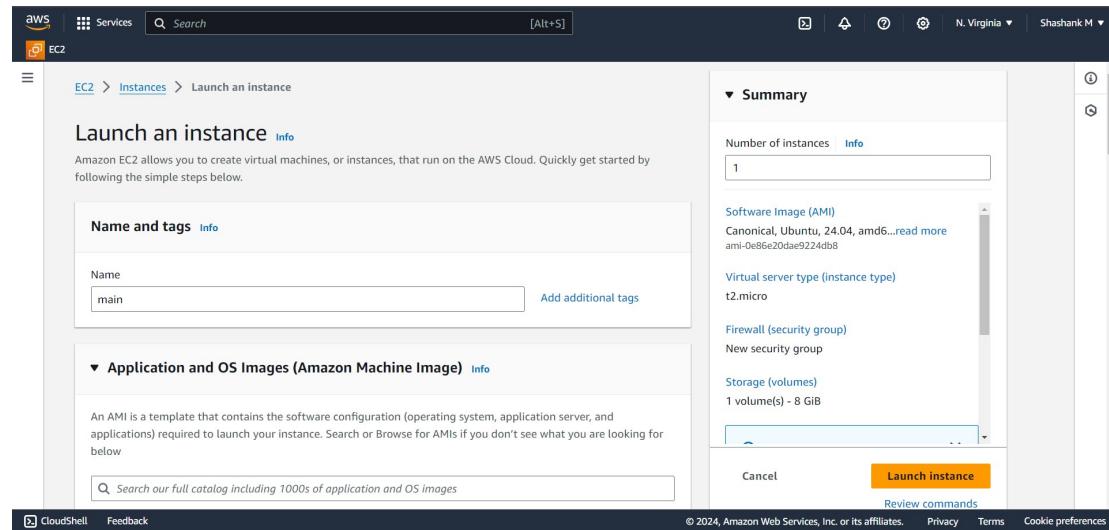
Problem Statement:

Company ABC wants to move their product to AWS. They have the following things set up right now:

1. MySQL DB
2. Website (PHP) The company wants high availability on this product, therefore wants Auto Scaling to be enabled on this website.

Solution:

1. Launch an instance for deployment of php application.



The screenshot shows a terminal window titled "EC2" within the AWS Management Console. The command "sudo apt-get update" has been run, and the output is displayed. The output includes information about available updates, the source of packages, and the progress of the download. At the bottom of the terminal, it shows the public and private IP addresses.

```
aws Services Search [Alt+S] EC2 See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-93-46:~$ sudo apt-get update
Hit: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 [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [380 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]

i-0fb80cafa19d88b28 (main)
PublicIPs: 52.91.203.137 PrivateIPs: 172.31.93.46
```

Update the instance.

2. Install any web server

The screenshot shows a terminal window titled "EC2" within the AWS Management Console. The command "sudo apt-get install apache2" has been run, and the output is displayed. The output shows the package being installed, dependencies being resolved, and the progress of the download. At the bottom of the terminal, it shows the public and private IP addresses.

```
aws Services Search [Alt+S] EC2

ubuntu@ip-172-31-93-254:~$ sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-utils libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libaprilt64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 143 not upgraded.
Need to get 2084 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libaprilt64 amd64 1.7.2-3.lubuntu0.1 [108 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.lubuntu7 [91.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.lubuntu7 [11.2 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.lubuntu7 [9116 B]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 liblua5.4-0 amd64 5.4.6-3build2 [166 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-lubuntu8.4 [1329 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-lubuntu8.4 [163 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-lubuntu8.4 [97.1 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-lubuntu8.4 [90.2 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntul [17.8 kB]
Fetched 2084 kB in 0s (35.4 MB/s)
Preconfiguring packages ...
```



3. Copy the source file from Desktop to Instance (code.zip)

Desktop

```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Shash>cd Downloads

C:\Users\Shash\Downloads>scp -i common.pem code.zip ubuntu@52.91.203.137:/home/ubuntu
Warning: Identity file -r not accessible: No such file or directory.
ubuntu@52.91.203.137: Permission denied (publickey).
lost connection

C:\Users\Shash\Downloads>scp -i common.pem code.zip ubuntu@52.91.203.137:/home/ubuntu
code.zip                                              100%  769KB 287.7KB/s  00:02

C:\Users\Shash\Downloads>
```

Instance

```
Running kernel seems to be up-to-date.

No services need to be restarted.

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-172-31-93-46:~$ ls
code.zip
ubuntu@ip-172-31-93-46:~$
```

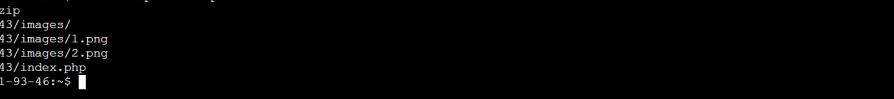
i-0fb0cafaf9d88b28 (main)

PublicIPs: 52.91.203.137 PrivateIPs: 172.31.93.46

4. Install unzip package

The following NEW packages will be installed:
unzip
0 upgraded, 1 newly installed, 0 to remove and 143 not upgraded.
Need to get 175 kB of archives.
After this operation, 384 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main unzip amd64 6.0-28ubuntu4 [175 kB]
Patched 175 kB in 0s (832 kB/s)
(Selecting previously unselected package unzip.
(Reading database ... 68464 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-28ubuntu4_amd64.deb ...
Unpacking unzip (6.0-28ubuntu4) ...
Setting up unzip (6.0-28ubuntu4) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes... [=====] Scanning processes... [=====] Scanning processes... [=====]
Scanning processes... [=] Scanning processes... [=] Scanning processes... [=]
Scanning processes... [===] Scanning processes... [===] Scanning processes... [===]
Scanning processes... [====] Scanning processes... [====] Scanning processes... [====]
Scanning processes... [=====] Scanning processes... [=====] Scanning processes... [=====]

5. Unzip code.zip file



The screenshot shows the AWS Lambda function editor interface. The top navigation bar includes the AWS logo, a 'Services' dropdown, a search bar, and account information for 'N. Virginia' and 'Shashank%20K'. On the left, there's a sidebar with 'EC2' selected. The main area displays the deployment code:

```
ubuntu@ip-172-31-93-46:~$ sudo unzip code.zip
Archive: code.zip
  creating: 1243/images/
  inflating: 1243/images/1.png
  inflating: 1243/images/2.png
  inflating: 1243/index.php
ubuntu@ip-172-31-93-46:~$
```

6. Move the folder to /var/www/html

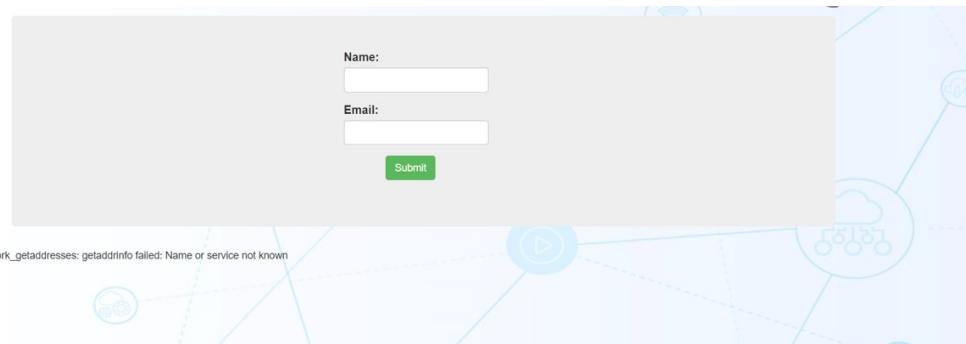
```
ubuntu@ip-172-31-89-214:/1243$ sudo mv * /var/www/html
```

AWS Services Search [Alt+S]

EC2

```
ubuntu@ip-172-31-89-214:/var/www/html$ ls
images index.php
ubuntu@ip-172-31-89-214:/var/www/html$
```

i-074706de80926567d (main)
PublicIPs: 54.173.122.9 PrivateIPs: 172.31.89.214



7. Download the dependencies required for php-mysqli

AWS Services Search [Alt+S]

EC2

```
ubuntu@ip-172-31-89-214:/$ sudo apt-get install php5.6 mysql-client php5.6-mysqli -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'php5.6-mysqli' instead of 'php5.6-mysq'
The following additional packages will be installed:
  debsurypg-archive-keyring libapache2-mod-php5.6 libpcre3 mysql-client-8.0 mysql-client-core-8.0 mysql-common php-common php5.6-cli php5.6-common
  php5.6-client php5.6-mysql php5.6-opcache php5.6-readline
Suggested packages:
  php-pe
The following NEW packages will be installed:
  debsurypg-archive-keyring libapache2-mod-php5.6 libpcre3 mysql-client mysql-client-8.0 mysql-client-core-8.0 mysql-common php-common php5.6
  php5.6-client php5.6-common php5.6-json php5.6-mysql php5.6-opcache php5.6-readline
0 upgraded, 15 newly installed, 0 to remove and 143 not upgraded.
Need to get 6957 kB of archives.
After this operation, 76.9 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libpcre3 amd64 2:8.39-15build1 [248 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.39-0ubuntu0.24.04.2 [2794 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 mysql-common all 5.0+11.0build1 [6746 B]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-8.0 amd64 8.0.39-0ubuntu0.24.04.2 [22.5 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client all 8.0.39-0ubuntu0.24.04.2 [9412 B]
Get:6 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 debsurypg-archive-keyring all 2024.02.05+ubuntu24.04.1+deb.sury.org+1 [4332 B]
Get:7 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php-common all 2:95ubuntu24.04.1+deb.sury.org+1 [13.2 kB]
Get:8 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble/main amd64 php5.6-common amd64 5.6.40-7+ubuntu24.04.1+deb.sury.org+1 [685 kB]
```

i-074706de80926567d (main)
PublicIPs: 54.173.122.9 PrivateIPs: 172.31.89.214

8. Create a database using RDS (mysql) :

Creating database database-1

Your database might take a few minutes to launch. You can use settings from database-1 to simplify configuration of suggested database add-ons while we finish creating your DB for you.

Introducing Aurora I/O-Optimized

Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% costs savings for I/O-intensive applications.

Consider creating a Blue/Green Deployment to minimize downtime during upgrades

You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. [RDS User Guide](#) [Aurora User Guide](#)

Databases (1)

DB identifier	Status	Role	Engine	Region ...	Size
database-1	Creating	Instance	MySQL Co...	-	db.t3.micro

9. Connect DB instance to EC2:

```
ubuntu@ip-172-31-89-214:~$ sudo mysql -h database-1.cfoc2gw02buk.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 28
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2024, 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.

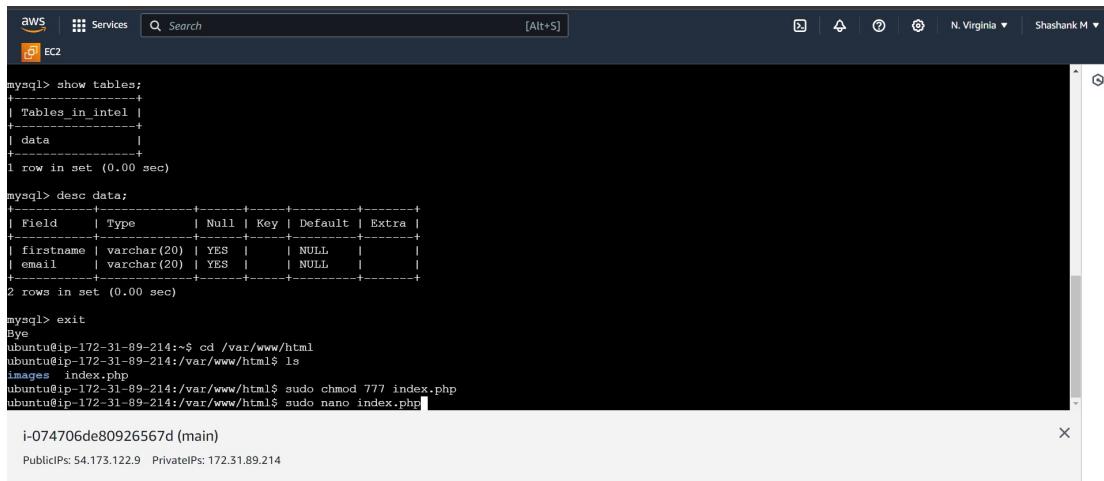
mysql> 
```

i-074706de80926567d (main)

PublicIPs: 54.173.122.9 PrivateIPs: 172.31.89.214

10. Create a database intel:

Create table with : 1. firstname 2. email



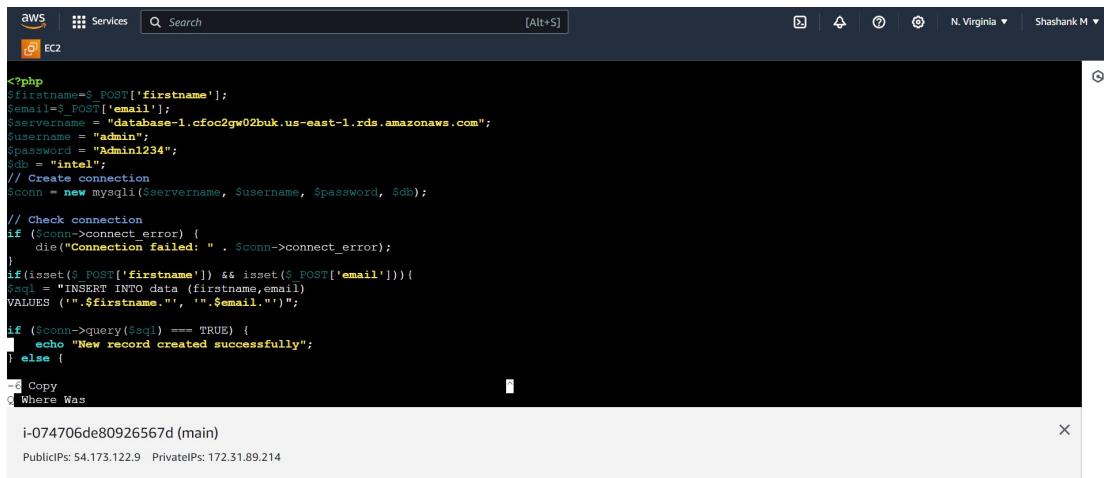
AWS Cloud9 IDE screenshot showing a terminal window with a MySQL session. The session starts with 'show tables;' which returns 'Tables_in_intel' and 'data'. Then 'desc data;' is run, showing two columns: 'firstname' and 'email', both of type 'varchar(20)' with 'YES' as the default value and 'NULL' as the extra value. Finally, 'exit' is run, followed by file operations: 'cd /var/www/html', 'ls', 'sudo chmod 777 index.php', and 'sudo nano index.php'. The bottom status bar shows the instance ID (i-074706de80926567d), public IP (54.173.122.9), and private IP (172.31.89.214).

```
aws Services Search [Alt+S] EC2
mysql> show tables;
+-----+
| Tables_in_intel |
+-----+
| data |
+-----+
1 row in set (0.00 sec)

mysql> desc data;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra   |
+-----+-----+-----+-----+-----+
| firstname | varchar(20) | YES |    | NULL    |          |
| email | varchar(20) | YES |    | NULL    |          |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> exit
Bye
ubuntu@ip-172-31-89-214:~$ cd /var/www/html
ubuntu@ip-172-31-89-214:/var/www/html$ ls
images index.php
ubuntu@ip-172-31-89-214:/var/www/html$ sudo chmod 777 index.php
ubuntu@ip-172-31-89-214:/var/www/html$ sudo nano index.php
i-074706de80926567d (main)
PublicIPs: 54.173.122.9 PrivateIPs: 172.31.89.214
```

11. Connect database endpoint in source file (index.php) and set username and password.



AWS Cloud9 IDE screenshot showing a terminal window with a PHP script. The script connects to a MySQL database using mysqli. It sets variables for host ('database-1.cfcoc2gw02buk.us-east-1.rds.amazonaws.com'), username ('admin'), password ('Admin1234'), and database ('intel'). It then creates a connection object (\$conn) and checks for errors. If successful, it inserts a new record into the 'data' table with 'firstname' and 'email' fields. The bottom status bar shows the instance ID (i-074706de80926567d), public IP (54.173.122.9), and private IP (172.31.89.214).

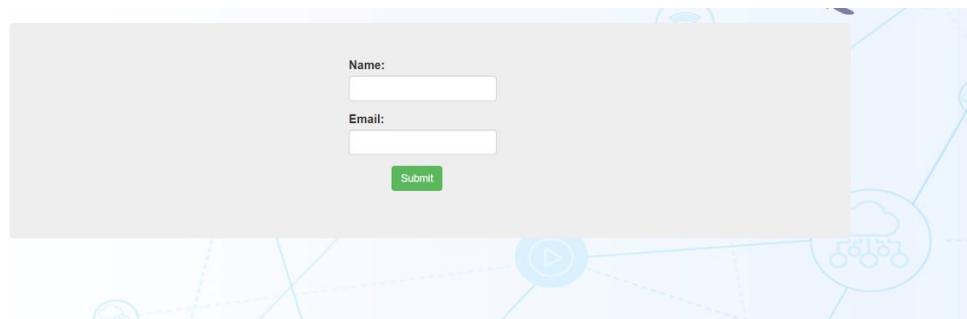
```
<?php
$firstname=$_POST['firstname'];
$email=$_POST['email'];
$servername = "database-1.cfcoc2gw02buk.us-east-1.rds.amazonaws.com";
$username = "admin";
$password = "Admin1234";
$db = "intel";
// Create connection
$conn = new mysqli($servername, $username, $password, $db);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
if(isset($_POST['firstname']) && isset($_POST['email'])){
$sql = "INSERT INTO data (firstname,email)
VALUES ('".$firstname."','".$.$email."')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
}
}
$conn->close();
}

i-074706de80926567d (main)
PublicIPs: 54.173.122.9 PrivateIPs: 172.31.89.214
```

12. Now connectivity issue with database resolved



13. Sample database updated from web page to real database

```
mysql> select * from intel;
ERROR 1146 (42S02): Table 'intel.intel' doesn't exist
mysql> select * from data;
+----+-----+
| firstname | email      |
+----+-----+
| aws       | aws@amazon.com |
+----+-----+
1 row in set (0.00 sec)

mysql> i-074706de80926567d (main)
PublicIPs: 54.173.122.9 PrivateIPs: 172.31.89.214
```

14. Create AMI

The screenshot shows the AWS EC2 console interface. On the left, there's a navigation sidebar with links like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main content area has a title 'Amazon Machine Images (AMIs) (1)'. It includes buttons for 'Create', 'Recycle Bin', 'EC2 Image Builder', 'Actions', and 'Launch Instance from AMI'. A search bar says 'Owned by me' and 'Find AMI by attribute or tag'. Below is a table with columns: Status, Creation date, Platform, Root device type, and Block devices. One entry is shown: Available, 2024/09/29 17:12 GMT+5:30, Linux/UNIX, ebs, /dev/sda1=snap-09259a5b05314d. At the bottom, a modal window titled 'Select an AMI' is open.

15. Create a launch template

The screenshot shows the AWS EC2 Launch Templates page. At the top, a green banner displays the message "Delete Launch Template Request Succeeded". Below this, the "Launch Templates (1) Info" section is shown, containing a table with one item:

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time
lt-0983ad47a45b29462	shashank-project-LT	1	1	2024-09-29T11:47:28.000Z

Below the table, a modal window titled "Select a launch template" is open, listing the same item from the table.

16. Create ASG(Auto Scaling Group) for high availability

The screenshot shows the "Create Auto Scaling group" wizard, Step 1: Choose launch template. The left sidebar lists steps: Step 1 (Choose launch template), Step 2 (Choose instance launch options), Step 3 - optional (Configure advanced options), Step 4 - optional (Configure group size and scaling), Step 5 - optional (Add notifications), Step 6 - optional (Add tags), and Step 7 (Review). The main area is titled "Choose launch template" and contains the following fields:

- Name:** shashank-project-asg
- Auto Scaling group name:** Enter a name to identify the group. The input field contains "shashank-project-asg". A note below states: "Must be unique to this account in the current Region and no more than 255 characters."

At the bottom of the main area, there is a note: "For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023."

The screenshot shows the 'Create Auto Scaling group' wizard in the AWS EC2 console. The first step, 'VPC', is selected. A dropdown menu shows 'vpc-05e80b0e599ac40fa' is chosen. Below it, a list of availability zones and subnets is shown, each with its IP range and status:

- us-east-1a | subnet-008a52a10e98ca283 | 172.31.32.0/20 | Default
- us-east-1b | subnet-01065aa5ba369b14a | 172.31.0.0/20 | Default
- us-east-1c | subnet-09038f5d4bff50a5b | 172.31.80.0/20 | Default
- us-east-1d | subnet-05551150e65fc0749 | 172.31.16.0/20 | Default
- us-east-1e | subnet-0c7a85e2998d1b4ff | 172.31.48.0/20 | Default
- us-east-1f | subnet-05ed24b4757249049 | 172.31.64.0/20 | Default

At the bottom, there are links for CloudShell, Feedback, and a footer with copyright information.

The screenshot shows the 'Auto Scaling groups' list page. It displays one group named 'shashank-project-asg' which is associated with the launch template 'shashank-project-LT'. The group has a current instance count of 2, a minimum of 2, and a maximum of 4. The table includes columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, and Max.

At the bottom, a message states '0 Auto Scaling groups selected'.

2 instances created for the high availability

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store. The main content area displays 'Instances (1/3) Info' with a search bar and filters for Instance state (running). A table lists three instances: 'server' (i-04da4f0e3b29d3ecb), 'server' (i-08502f900d6dc46b5), and 'main' (i-074706de80926567d). The 'main' instance is selected. Below the table is the 'i-074706de80926567d (main)' details pane, which includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab shows the Instance ID (i-074706de80926567d), Public IPv4 address (54.173.122.9), Private IPv4 addresses (172.31.89.214), Public IPv6 DNS (ec2-54-173-122-9.compute-1.amazonaws.com), and Instance state (Running).

17. Create load balancer for traffic on application.

The screenshot shows the 'Load balancer types' section of the AWS CloudFront console. It compares three types: Application Load Balancer (ALB), Network Load Balancer (NLB), and Gateway Load Balancer (GWLB). The ALB diagram shows a client connecting to an ALB, which then routes traffic to Lambda functions or an origin. The NLB diagram shows a client connecting to an NLB, which routes traffic via TCP, UDP, or TLS to an ALB or another NLB. The GWLB diagram shows a client connecting to a GWLB, which routes traffic to third-party virtual appliances. Each type has a detailed description below its diagram.

18. Create a target group to register the target instances for Application Load Balancer

The screenshot shows the 'Create target group' wizard. Step 1: 'Specify group details' is completed. Step 2: 'Register targets' is in progress, showing 'Available instances (3/3)' in a table. The table lists three instances: 'server' (i-08502f900d6dc46b5), 'server' (i-04da4f0e3b29d3ecb), and 'main' (i-074706de80926567d), all in the 'Running' state and associated with the 'default' security group. A note at the bottom states: 'This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.'

Load balancer created successfully

The screenshot shows the AWS EC2 Load Balancers console. A success message at the top states: "Successfully created load balancer: shashank-project-LB. It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks." Below this, the navigation path is EC2 > Load balancers > shashank-project-LB. The main view displays the details of the load balancer "shashank-project-LB". The "Details" section includes the following information:

Load balancer type	Status	VPC	Load balancer IP address type
Application	Active	vpc-05e80b0e599ac40fa	IPv4
Scheme	Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones subnet-05551150e65fc0749 us-east-1d (use1-az4) subnet-008a52a10e98ca283 us-east-1a (use1-az6) subnet-01065aa5ba369b14a us-east-1b (use1-az1)
			Date created September 29, 2024, 17:36 (UTC+05:30)

At the bottom of the page, there are links for CloudShell, Feedback, and a footer with copyright information: © 2024, Amazon Web Services, Inc. or its affiliates.

19. Take the DNS name from load balancer created for target instance.

<http://shashank-project-lb-975489885.us-east-1.elb.amazonaws.com/>

Ready to go:

