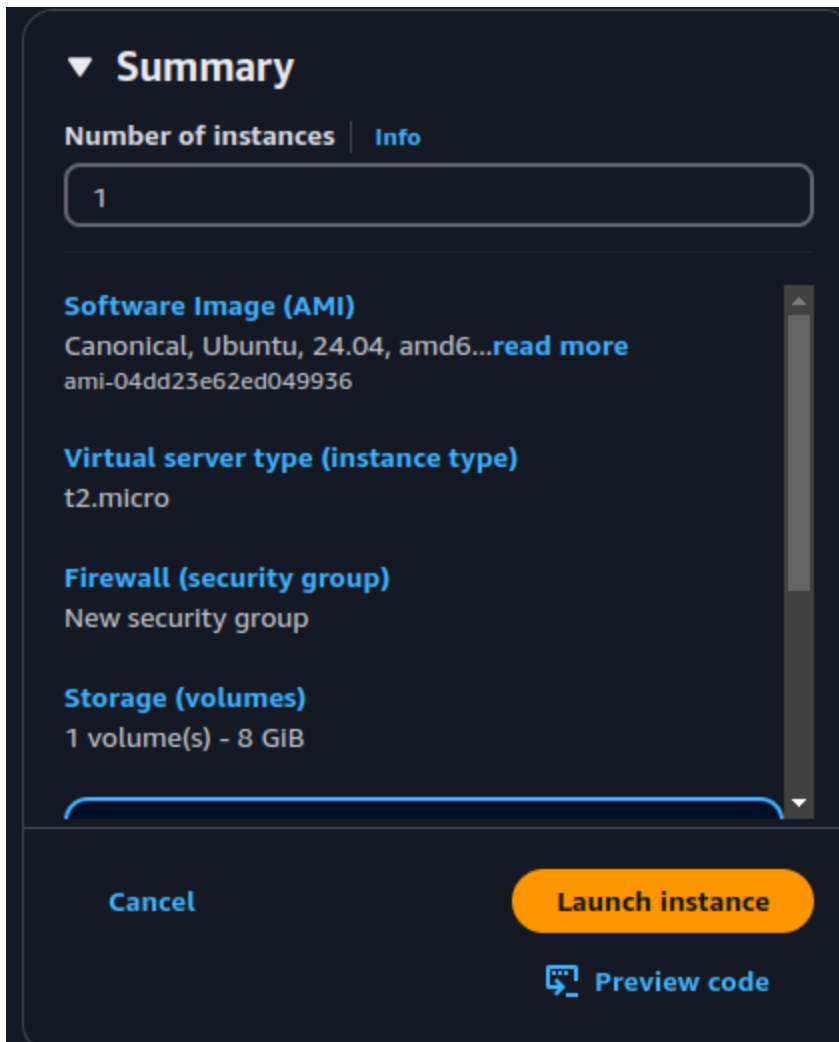


# DOCUMENTATION

## Deploying Todo List on EC2 using PHP

### Step 1: Launch an Ubuntu EC2 Instance

1. Log in to the **AWS Console** and navigate to the EC2 Dashboard.
2. Launch an Instance:
  - **AMI:** Select **Ubuntu Server 24.04 LTS**.
  - **Instance Type:** `t2.micro` (Free Tier Eligible).
  - **Key Pair:** Create or use an existing key pair.
  - **Security Group:**
    - Allow **HTTP (Port 80)** from 0.0.0.0/0.
    - Allow **SSH (Port 22)** from your IP.
3. Launch the Instance.



The screenshot shows the 'Launch Instance' summary page in the AWS console. It features a dark theme with a sidebar on the left and a main content area. The sidebar has a 'Summary' section with a dropdown arrow. The main content area has a 'Number of instances' input field set to '1'. Below this, there are four sections: 'Software Image (AMI)' showing 'Canonical, Ubuntu, 24.04, amd64...read more' with the ID 'ami-04dd23e62ed049936'; 'Virtual server type (instance type)' showing 't2.micro'; 'Firewall (security group)' showing 'New security group'; and 'Storage (volumes)' showing '1 volume(s) - 8 GiB'. At the bottom, there are three buttons: 'Cancel', 'Launch instance' (in orange), and 'Preview code' (with a code icon).

▼ Summary

Number of instances | Info

1

**Software Image (AMI)**  
Canonical, Ubuntu, 24.04, amd64...[read more](#)  
ami-04dd23e62ed049936

**Virtual server type (instance type)**  
t2.micro

**Firewall (security group)**  
New security group

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

Cancel Launch instance Preview code

Instances (1) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

< 1 >

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
<input type="checkbox"/>	Project-3	i-00418f63fee992020	<span>Running</span>	t2.micro	Initializing	<a href="#">View alarms +</a>	us-west-2

---

## Step 2: Connect to the EC2 Instance

1. SSH into the instance:

```
ssh -i "your-key-file.pem" ubuntu@<INSTANCE_PUBLIC_IP>
```

### Connect to instance Info

Connect to your instance i-00418f63fee992020 (Project-3) using any of these options

EC2 Instance Connect Session Manager **SSH client** EC2 serial console

**Instance ID**  
i-00418f63fee992020 (Project-3)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this Instance is project-3-key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
`chmod 400 "project-3-key.pem"`
4. Connect to your Instance using its Public DNS:  
ec2-35-88-193-167.us-west-2.compute.amazonaws.com

✔ Command copied

```
ssh -i "project-3-key.pem" ubuntu@ec2-35-88-193-167.us-west-2.compute.amazonaws.com
```

**Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

```
System load:  0.3          Processes:           118
Usage of /:   22.9% of 6.71GB Users logged in:        0
Memory usage: 22%         IPv4 address for enX0: 172.31.9.111
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
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-172-31-9-111:~$
```

## 2. Update the system:

```
sudo apt update -y
sudo apt upgrade -y
```

```
ubuntu@ip-172-31-9-111:~$ sudo apt update -y
sudo apt upgrade -y
```

## Step 3: Install Apache, PHP, and MySQL

### 1. Install Apache:

```
sudo apt install apache2 -y
```

```
ubuntu@ip-172-31-9-111:~$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

### 2. Install PHP and Required Modules:

```
sudo apt install php libapache2-mod-php php-mysql -y
```

```
ubuntu @ session #1: sshd[1003]
ubuntu@ip-172-31-9-111:~$ sudo apt install php libapache2-mod-php php-mysql -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

### 3. Install MySQL:

```
sudo apt install mysql-server -y
```

```
ubuntu@ip-172-31-9-111:~$ sudo apt install mysql-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

### 4. Secure MySQL Installation:

```
sudo mysql_secure_installation
```

- Set a root password and follow the prompts to secure MySQL.

---

## Step 4: Set Up the Database

### 1. Log into MySQL:

```
sudo mysql -u root -p
```

```

All done!
ubuntu@ip-172-31-9-111:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

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>

```

## 2. Create a Database:

```

CREATE DATABASE todo;
EXIT;

```

```

Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

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> CREATE DATABASE todo;
Query OK, 1 row affected (0.01 sec)

mysql> EXIT;
Bye
ubuntu@ip-172-31-9-111:~$

```

## 3. Import the `todo.sql` File:

- Clone the GitHub repository:

```

sudo apt install git -y
git clone https://github.com/ali-azgar-rakib/ToDo-list-with-php.git

```

```

ubuntu@ip-172-31-9-111:~$ sudo apt install git -y
git clone https://github.com/ali-azgar-rakib/ToDo-list-with-php.git

```

- Navigate to the repository directory:

```
cd Todo-list-with-php
```

- Import the `todo.sql` file into the `todo` database:

```
sudo mysql -u root -p todo < todo.sql
```

```
ubuntu@ip-172-31-9-111:~$ cd Todo-list-with-php/  
ubuntu@ip-172-31-9-111:~/Todo-list-with-php$ sudo mysql -u root -p todo < todo.sql  
Enter password:  
ubuntu@ip-172-31-9-111:~/Todo-list-with-php$
```

---

## Step 5: Deploy the Application

### 1. Move the Application Files:

- Move all files from the cloned repository to the Apache web directory:

```
sudo mv * /var/www/html/
```

### 2. Set Permissions:

- Ensure Apache has access to the files:

```
sudo chmod -R 755 /var/www/html/
```

### 3. Restart Apache:

```
sudo systemctl restart apache2
```

```
ubuntu@ip-172-31-9-111:~$ sudo mv * /var/www/html/  
ubuntu@ip-172-31-9-111:~$ sudo chmod -R 755 /var/www/html/  
ubuntu@ip-172-31-9-111:~$ sudo systemctl restart apache2  
ubuntu@ip-172-31-9-111:~$
```

---

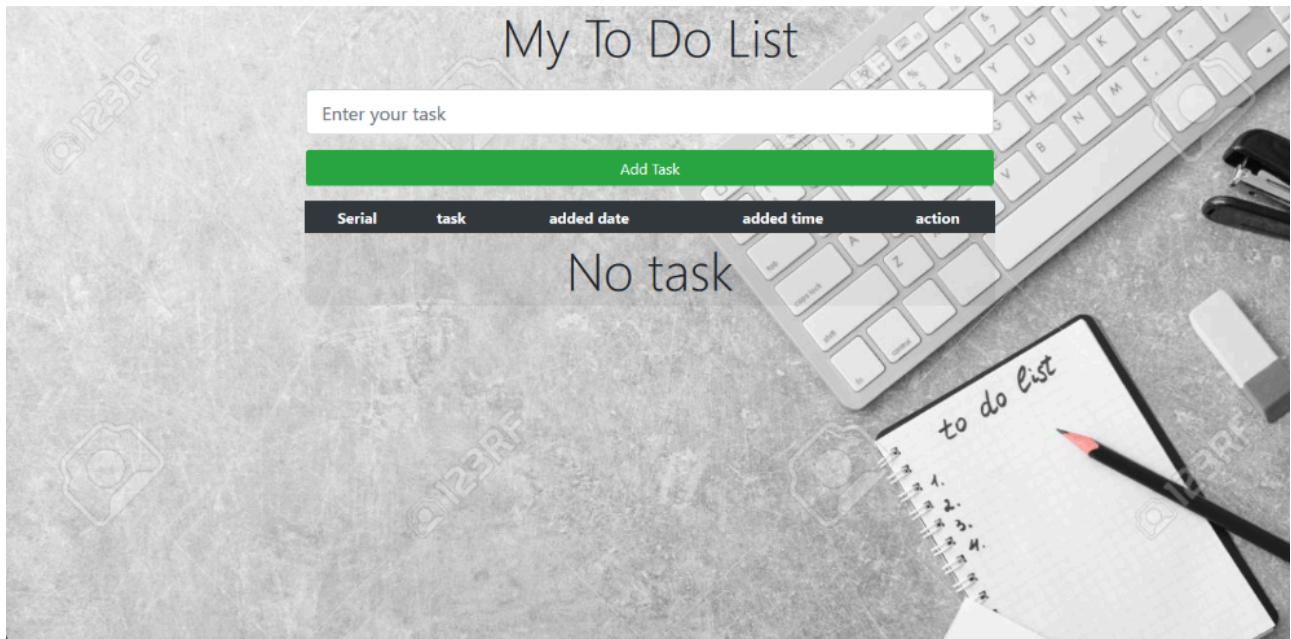
## Step 6: Test the Application

1. Open your browser and navigate to:

`http://<INSTANCE_PUBLIC_IP>`

## 2. What You Should See:

- A working To-Do List application with options to:
  - Add tasks.
  - View tasks.
  - Edit tasks.
  - Delete tasks.



---

## Step 7: Secure and Clean Up

1. **Restrict SSH Access:**
  - Update your EC2 Security Group to allow SSH only from your IP address.
2. **Stop or Terminate the Instance (if not needed).**

---

## Conclusion

You've successfully deployed a PHP-based To-Do List application on an EC2 instance! You can now manage tasks, edit to-do items, and access the application from anywhere using the public IP (or your custom domain if configured).

