# Launch an EC2 instance

[**PDF**](https://docs.aws.amazon.com/pdfs/AmazonRDS/latest/UserGuide/rds-ug.pdf#CHAP_Tutorials.WebServerDB.LaunchEC2)[**RSS**](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/rdsupdates.rss)

Create an Amazon EC2 instance in the public subnet of your VPC.

To launch an EC2 instance

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the upper-right corner of the AWS Management Console, choose the AWS Region where you want to create the EC2 instance.
3. Choose EC2 Dashboard, and then choose Launch instance, as shown following.  
   
                           EC2 Dashboard
                       
4. Make sure you have opted into the new launch experience.
5. Under Name and tags, for Name, enter **tutorial-ec2-instance-web-server**.
6. Under Application and OS Images (Amazon Machine Image), choose Amazon Linux, and then choose the Amazon Linux 2 AMI. Keep the defaults for the other choices.  
   
                           Choose an Amazon Machine Image
                       
7. Under Instance type, choose t2.micro.
8. Under Key pair (login), choose a Key pair name to use an existing key pair. To create a new key pair for the Amazon EC2 instance, choose Create new key pair and then use the Create key pair window to create it.  
   For more information about creating a new key pair, see [Create a key pair](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html#create-a-key-pair) in the *Amazon EC2 User Guide for Linux Instances*.
9. Under Network settings, set these values and keep the other values as their defaults:
   * For Allow SSH traffic from, choose the source of SSH connections to the EC2 instance.  
     You can choose My IP if the displayed IP address is correct for SSH connections.  
     Otherwise, you can determine the IP address to use to connect to EC2 instances in your VPC using Secure Shell (SSH). To determine your public IP address, in a different browser window or tab, you can use the service at [https://checkip.amazonaws.com](https://checkip.amazonaws.com/). An example of an IP address is 203.0.113.25/32.  
     In many cases, you might connect through an internet service provider (ISP) or from behind your firewall without a static IP address. If so, make sure to determine the range of IP addresses used by client computers.  
     **Warning**If you use 0.0.0.0/0 for SSH access, you make it possible for all IP addresses to access your public instances using SSH. This approach is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, authorize only a specific IP address or range of addresses to access your instances using SSH.
   * Turn on Allow HTTPs traffic from the internet.
   * Turn on Allow HTTP traffic from the internet.
10. 
                            Configure Instance Details
                        
11. Leave the default values for the remaining sections.
12. Review a summary of your instance configuration in the Summary panel, and when you're ready, choose Launch instance.
13. On the Launch Status page, shown following, note the identifier for your new EC2 instance, for example: i-03a6ad47e97ba9dc5.  
    
                            Launch Status
                        
14. Choose View all instances to find your instance.
15. Wait until Instance state for your instance is Running before continuing.
16. Complete [Create a DB instance](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateDBInstance.html).

# Install a web server on your EC2 instance

Connect to your EC2 instance and install the web server.

To connect to your EC2 instance and install the Apache web server with PHP

1. Connect to the EC2 instance that you created earlier by following the steps in [Connect to your Linux instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html).
2. Get the latest bug fixes and security updates by updating the software on your EC2 instance. To do this, use the following command.  
   **Note**The -y option installs the updates without asking for confirmation. To examine updates before installing, omit this option.
3. sudo yum update -y
4. After the updates complete, install the PHP software using the amazon-linux-extras install command. This command installs multiple software packages and related dependencies at the same time.
5. sudo amazon-linux-extras install php8.0 mariadb10.5
6. If you receive an error stating sudo: amazon-linux-extras: command not found, your instance wasn't launched with an Amazon Linux 2 AMI. You might be using the Amazon Linux AMI instead. You can view your version of Amazon Linux using the following command.
7. cat /etc/system-release
8. For more information, see [Updating instance software](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-updates.html).
9. Install the Apache web server.
10. sudo yum install -y httpd
11. Start the web server with the command shown following.
12. sudo systemctl start httpd
13. You can test that your web server is properly installed and started. To do this, enter the public Domain Name System (DNS) name of your EC2 instance in the address bar of a web browser, for example: http://ec2-42-8-168-21.us-west-1.compute.amazonaws.com. If your web server is running, then you see the Apache test page.  
    If you don't see the Apache test page, check your inbound rules for the VPC security group that you created in [Tutorial: Create a VPC for use with a DB instance (IPv4 only)](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateVPC.html). Make sure that your inbound rules include one allowing HTTP (port 80) access for the IP address to connect to the web server.  
    **Note**The Apache test page appears only when there is no content in the document root directory, /var/www/html. After you add content to the document root directory, your content appears at the public DNS address of your EC2 instance. Before this point, it appears on the Apache test page.
14. Configure the web server to start with each system boot using the systemctl command.
15. sudo systemctl enable httpd

## **Connect your Apache web server to your DB instance**

Next, you add content to your Apache web server that connects to your Amazon RDS DB instance.

To add content to the Apache web server that connects to your DB instance

1. While still connected to your EC2 instance, change the directory to /var/www and create a new subdirectory named inc.

cd /var/www

mkdir inc

1. cd inc

vi dbinfo.inc

<?php

define('DB\_SERVER', 'db\_instance\_endpoint');

define('DB\_USERNAME', 'tutorial\_user');

define('DB\_PASSWORD', 'master password');

define('DB\_DATABASE', 'sample');

?>

cd /var/www/html

vi SamplePage.php

<?php include "../inc/dbinfo.inc"; ?>

<html>

<body>

<h1>Sample page</h1>

<?php

/\* Connect to MySQL and select the database. \*/

$connection = mysqli\_connect(DB\_SERVER, DB\_USERNAME, DB\_PASSWORD);

if (mysqli\_connect\_errno()) echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

$database = mysqli\_select\_db($connection, DB\_DATABASE);

/\* Ensure that the EMPLOYEES table exists. \*/

VerifyEmployeesTable($connection, DB\_DATABASE);

/\* If input fields are populated, add a row to the EMPLOYEES table. \*/

$employee\_name = htmlentities($\_POST['NAME']);

$employee\_address = htmlentities($\_POST['ADDRESS']);

if (strlen($employee\_name) || strlen($employee\_address)) {

AddEmployee($connection, $employee\_name, $employee\_address);

}

?>

<!-- Input form -->

<form action="<?PHP echo $\_SERVER['SCRIPT\_NAME'] ?>" method="POST">

<table border="0">

<tr>

<td>NAME</td>

<td>ADDRESS</td>

</tr>

<tr>

<td>

<input type="text" name="NAME" maxlength="45" size="30" />

</td>

<td>

<input type="text" name="ADDRESS" maxlength="90" size="60" />

</td>

<td>

<input type="submit" value="Add Data" />

</td>

</tr>

</table>

</form>

<!-- Display table data. -->

<table border="1" cellpadding="2" cellspacing="2">

<tr>

<td>ID</td>

<td>NAME</td>

<td>ADDRESS</td>

</tr>

<?php

$result = mysqli\_query($connection, "SELECT \* FROM EMPLOYEES");

while($query\_data = mysqli\_fetch\_row($result)) {

echo "<tr>";

echo "<td>",$query\_data[0], "</td>",

"<td>",$query\_data[1], "</td>",

"<td>",$query\_data[2], "</td>";

echo "</tr>";

}

?>

</table>

<!-- Clean up. -->

<?php

mysqli\_free\_result($result);

mysqli\_close($connection);

?>

</body>

</html>

<?php

/\* Add an employee to the table. \*/

function AddEmployee($connection, $name, $address) {

$n = mysqli\_real\_escape\_string($connection, $name);

$a = mysqli\_real\_escape\_string($connection, $address);

$query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";

if(!mysqli\_query($connection, $query)) echo("<p>Error adding employee data.</p>");

}

/\* Check whether the table exists and, if not, create it. \*/

function VerifyEmployeesTable($connection, $dbName) {

if(!TableExists("EMPLOYEES", $connection, $dbName))

{

$query = "CREATE TABLE EMPLOYEES (

ID int(11) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

NAME VARCHAR(45),

ADDRESS VARCHAR(90)

)";

if(!mysqli\_query($connection, $query)) echo("<p>Error creating table.</p>");

}

}

/\* Check for the existence of a table. \*/

function TableExists($tableName, $connection, $dbName) {

$t = mysqli\_real\_escape\_string($connection, $tableName);

$d = mysqli\_real\_escape\_string($connection, $dbName);

$checktable = mysqli\_query($connection,

"SELECT TABLE\_NAME FROM information\_schema.TABLES WHERE TABLE\_NAME = '$t' AND TABLE\_SCHEMA = '$d'");

if(mysqli\_num\_rows($checktable) > 0) return true;

return false;

}

?>

<http://ec2-55-122-41-31.us-west-2.compute.amazonaws.com/SamplePage.php>.

Note make sure you will add private IP add in RDS security group inbound rule for mysql