**AWS Setup Instructions for Deploying a Web Server**

The next part of this guide specifically describes how the web server is to be set up on AWS to facilitate the project of the Ozmart Retail Group’s cloud migration. With the help of AWS reliable and growing cloud solutions, Ozmart plans to achieve its strategy of high application availability, security, and performance.

**1. Prerequisites**

Before starting the deployment, ensure the following prerequisites are met:

* **AWS Account:** AWS has to be active and could also be a paid account for best results to be obtained. If you don’t have one, create one at the AWS portal.
* **IAM User Permissions:** Make sure you have necessary permissions or devise and obtain ample permissions that enables you create resources in AWS such as EC2 instances, VPCs, and security groups.
* **Web Server Files:** Gather all the files that will be run on your web server such as HTML files, CSS files, JavaScript files, server-side scripts and others.

**2. Creating an EC2 Instance**

1. **Log in to AWS Management Console**:
   * Navigate to [console.aws.amazon.com](https://console.aws.amazon.com/) and log in with your AWS credentials.
2. **Launch an EC2 Instance**:
   * Perhaps starting from the AWS Management Console, navigate to Services and then Click on EC2 under Compute.
   * To perform the setup, click on Launch Instance.
3. **Choose an Amazon Machine Image (AMI)**:
   * Select an AMI based on the operating system you need for your web server. Common choices include:
     + Amazon Linux 2 AMI is considered as the most suitable option due to compatibility and optimizations for the AWS.
     + Ubuntu Server 20. 04 LTS for a rock-solid Linux operating system.
     + Microsoft Windows Server 2019 Base, applications created using the Windows base complied the same.
4. **Choose an Instance Type**:
   * Specify an instance type that provides needed performance level and is cost effective. For the simplest example of the web server t2. micro which is eligible for 25000 free tier or t3. micro is sufficient.
5. **Configure Instance Details**:
   * **Number of Instances:** Keep as 1 if you do not require more for whatever purpose it is being used for.
   * **Network:** You can choose Create new VPC to configure a new Virtual Private Cloud for your Amazon Web Services (or select an existing Virtual Private Cloud).
   * **Subnet:** Select a subnet within the created VPC that falls under the public zone for the purpose of accessing the internet.
   * **Auto-assign Public IP:** Do this to make your instance accessible from the internet.
   * **IAM Role:** If your Web server has to access other AWS services you need to provide an IAM role which has the relevant permissions.
   * **Shutdown Behavior:** Select Stop to prevent From stopping the instance by mistake.
6. **Add Storage**:
   * **Root Volume:** Standard options should be enough (for instance, 8 GiB SSD).
   * **Additional Volumes:** You can add more volumes if your application demand more storage space.
7. **Configure Security Group**:
   * To create firewall rules, develop a new security group.
   * Allow inbound traffic for:
     + HTTP (Port 80) for web traffic –/w – According to my understanding the following are the ports currently been used;
     + HTTP (port 80) which is HTTP secure or HTTPS (port 443) for secured web traffic.
     + SSH (port 22) or RDP (port 3389) based on your choice of AMI, for administrator purpose.
   * **Outbound Rules:** There isn’t a general rule, but often it is allowed and a rule allows all traffic going out.
8. **Review and Launch**:
   * Review all your setting. Click on ‘Launch’ and choose an existing key pair or create a new one especially for SSH purposes.
   * After filling all the above information, click on the Launch Instances button to confirm.

**3. Configuring the Web Server**

1. **Connect to Your EC2 Instance**:
   * **SSH Access (Linux):** To access the instances, you can command line or terminal using the public IP and the key pair ever created.

bash

Copy code

ssh -i "your-key-pair.pem" ec2-user@your-ec2-public-ip

* + **RDP Access (Windows):** Connect using RDP client with the help of the IP address provided to the server and the account details.

1. **Install Web Server Software**:
   * **Linux**:
     + Update the package repository:

bash

Copy code

sudo yum update -y # Amazon Linux

sudo apt update -y # Ubuntu

* + - Install Apache or Nginx:

bash

Copy code

sudo yum install httpd -y # Amazon Linux

sudo apt install apache2 -y # Ubuntu

sudo systemctl start httpd # Start Apache

sudo systemctl enable httpd # Enable Apache to start on boot

* + **Windows**:
    - Add the Web Server (IIS) role without necessarily using the Add Roles Wizard.

1. **Deploy Your Web Application**:
   * **Linux**:
     + Transfer your web server files to the /var/www/html directory:

bash

Copy code

sudo cp -r /path/to/your/web/files/\* /var/www/html/

* + **Windows**:
    - Place your web application files in the C:\inetpub\wwwroot directory.

1. **Test the Web Server**:
   * Open a web browser and navigate to the public IP address of your EC2 instance (http:/>Your EC2 Public IP (your-EC2-Public-IP) Depending upon this you should be able to see your ‘web application’ or ‘default web server page’.
2. **Secure Your Server**:
   * **Install SSL/TLS Certificates**:
     + Some of these include Let’s Encrypt or AWS Certificate Manager (ACM) that assist in installing secure websites with HTTPS.
   * **Configure Firewalls and Security Groups**:
     + Check that security group rules are set up to be as tight as possible allowing only admitted traffic.

**4. Monitoring and Scaling**

1. **Enable CloudWatch Monitoring**:
   * Utilize the CloudWatch service of Amazon to check on your instance’s health, including CPU usage, disk space and network bandwidth.
2. **Set Up Auto Scaling**:
   * Define different Auto Scaling Groups to launch or terminate EC2 instances in order to provide better performance and minimize the expenditure.

Thus, the Ozmart Retail Group can achieve the organization’s goals of deploying a web server on AWS and utilizing cloud solutions for efficient business activity.