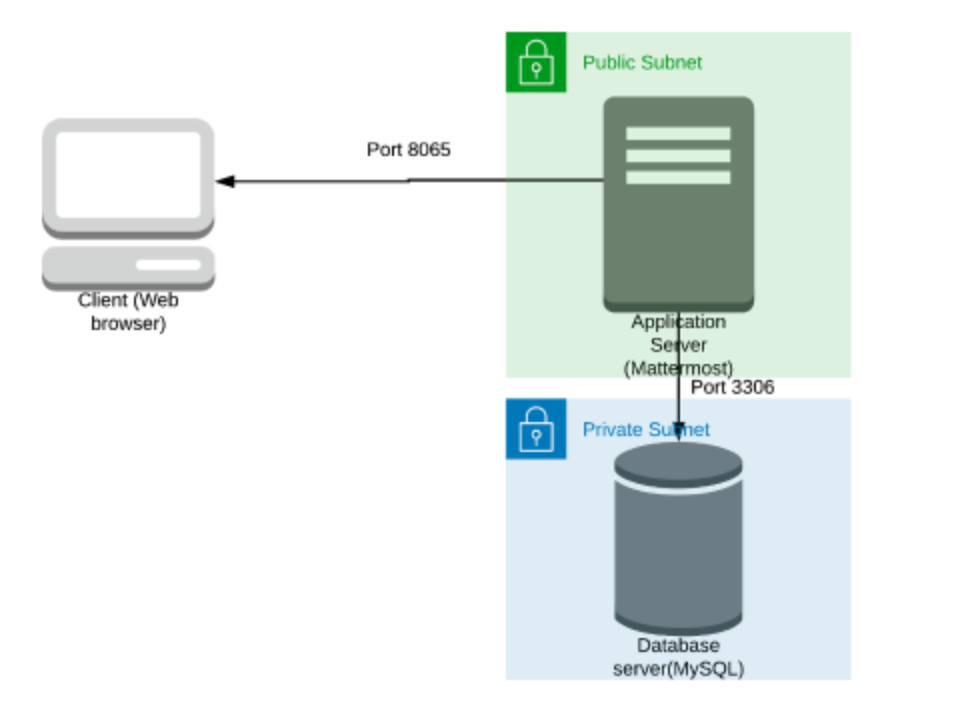
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Declaration** |  | | | | |
| Questions in this exercise are intentionally complex and could be convoluted or confusing. This is by design and to simulate real life situations where customers seldom give crystal clear requirements and ask unambiguous questions. | | | | | |
|
|
| I have read the above statement and agree to these conditions | | | | | |
| I AGREE | ANUBHAV CHATURVEDI | | | | |
| <Enter your name above this line to indicate that you are in agreement> | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Instructions** |  |  |  |  |  |
| Every screenshot requested in this workbook is compulsory and carries 0.5 marks | | | | | |
| Your AWS account ID must be clearly visible in every screenshot using the AWS console; missing id or using someone else's id is not permitted. Such cases will be considered as plagiarism and severe penalty will be imposed. | | | | | |
| All screenshots must be in the order mentioned under "Expected Screenshots" for every step | | | | | |
| DO NOT WAIT UNTIL THE LAST MINUTE. | | | | | |
| The file should be renamed in the format BATCH\_FIRSTNAME\_LASTNAME\_PROJECT1.  For example: IITR\_FSD\_VIJAY\_DWIVEDI\_PROJECT1.docx | | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Resource Clean Up** | |  |  |  |  |
| Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you’ve completed your lab or project is always necessary. This is true whether you’re doing a lab or implementing a project at your workplace. | | | | | |
| After completing the lab, make sure to delete each resource created in reverse chronological order. | | | | | |

**Scenario**

Team communication and instant messaging solutions are an integral part of any business environment today. As of 2020, the total number of users of Slack and Microsoft Teams exceeded 20 million.

Some organizations might have compliance policies in place which do not allow them to use services managed by third parties. They will prefer solutions that can be managed and hosted on servers controlled by them. The same will extend to communication solutions as well.

**Architecture diagram**

****

|  |  |
| --- | --- |
| **Architecture Implementation** | |
| 1 | Implement 2 different subnets (one public and the other private) in a custom VPC |
| 2 | Install and configure MySQL on an Amazon Linux 2 instance on the private subnet using the instructions provided. (Hint: Use a bastion host and a NAT gateway) |
| 3 | Install and configure Mattermost on an Amazon Linux 2 instance on the public subnet using the provided instructions. |
| 4 | Configure the security groups to allow the ports as shown in the architecture. |
| 5 | Test the installation by accessing the IP of the public instance in a browser via the port 8065. |

**Step 1: VPC and Subnet Creation**

|  |  |  |
| --- | --- | --- |
| Step number | a |  |
| Step name | Creation of VPC |  |
| Instructions | 1) Navigate to VPC using the Services button at the top of the screen  2) Select "Your VPCs" on the left side of the screen  3) Click on "Create VPC"  4) Enter the following fields :  Name: Project 1 VPC  IPv4 CIDR Block : 10.0.0.0/16  The rest of the options can be ignored  5) Select "Create VPC"  6) Select the VPC and click on Actions->Edit DNS hostnames  7) Enable DNS hostnames and click on Save |  |
| Expected screenshots | 1. Created VPC with properties visible |  |
|  | |  |

|  |  |
| --- | --- |
| Step number | b |
| Step name | Creation of public subnet |
| Instructions | 1) Navigate to VPC->Subnets  2) Click on "Create Subnet"  3) Enter the following fields  Name tag : Public Subnet  VPC : Select the Project 1 VPC  IPv4 CIDR block : 10.0.1.0/24  The other options can be ignored  4) Click on Create  5) Once the subnet has been created, select the subnet and click on Actions->Modify Auto-assign IP settings  6) Enable the option "Auto assign IPv4" and select Save |
| Expected screenshots | 1. Subnet Creation screen |
|  | |

|  |  |
| --- | --- |
| Step number | c |
| Step name | Creation of private subnet |
| Instructions | 1) Navigate to VPC->Subnets  2) Click on "Create Subnet"  3) Enter the following fields  Name tag : Private Subnet  VPC : Select the Project 1 VPC  IPv4 CIDR block : 10.0.2.0/24  The other options can be ignored  4) Click on Create |
| Expected screenshots | 1. Subnet Creation screen |
|  | |

**Step 2 : Internet Gateway and VPC**

|  |  |
| --- | --- |
| Step number | a |
| Step name | Creation and Configuration of Internet Gateway |
| Instructions | 1) Navigate to VPCs->Internet Gateway  2) Click on "Create Internet Gateway"  3) Enter the name tag "Project 1 Internet Gateway" and click on "Create Internet Gateway"  4) After the gateway is created, select it and click on Actions->Attach to VPC  5) Select the Project 1 VPC and click on "Attach Internet Gateway" |
| Expected screenshots | 1. Creation of Internet Gateway |
|  | |

|  |  |
| --- | --- |
| Step number | b |
| Step name | Creation of public route table |
| Instructions | 1) Navigate to VPC -> Route Tables and click on Create Route table  2) Enter the name tag "Public Route Table", select the Project 1 VPC from the dropdown and click on Create  3) Once the route table is created, select it and select the Routes tab below the list of route tables  4) Click in Edit Routes and add the following route (Don't edit the existing one)  - Destination : 0.0.0.0/0  - Target : Select Internet Gateway and the select the Project 1 Internet Gateway  Click on Save Routes  5) Select the Subnet Associations tab and click on Edit Subnet Associations  6) Select the Public Subnet from the list and click on Save |
| Expected screenshots | 1. Route list of the route table 2. Subnet Associations of the route table |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step number | c |  |  |  |
| Step name | Creation of NAT gateway |  |  |  |
| Instructions | 1) Navigate to VPC using the Services button at the top of the screen  2) Select NAT Gateway at the left side of the screen  3) Click on Create NAT Gateway  - Deploy it in the public subnet  - Connectivity type : Public  - Allocate an elastic IP by clicking on “Allocate Elastic IP”  4) Click on “Create NAT Gateway” to create the gateway |  |  |  |
| Expected screenshots | 1. NAT gateway creation details 2. Gateway after creation |  |  |  |
|  |  |  |  |  |
|  | | | | |

|  |  |
| --- | --- |
| Step number | d |
| Step name | Creation of private route tables |
| Instructions | 1) Navigate to VPC -> Route Tables and click on Create Route table  2) Enter the name tag "Private Route Table", select the Project 1 VPC from the dropdown and click on Create  3) Once the route table is created, select it and select the Routes tab below the list of route tables  4) Click in Edit Routes and add the following route (Don't edit the existing one)  - Destination : 0.0.0.0/0  - Target: Select NAT Gateway and select the NAT Gateway created in the previous step  Click on Save Routes  5) Select the Subnet Associations tab and click on Edit Subnet Associations  6) Select the private Subnet from the list and click on Save |
| Expected screenshots | 1. Route list of the route table 2. Subnet association of the route table |
|  | |

**Step 3 : Creation of database and application servers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step number | a |  |  |  |
| Step name | Creation of application server |  |  |  |
| Instructions | 1) Navigate to EC2 using the Services button at the top of the screen  2) Select Instances at the left side of the screen  3) Click on Launch Instance  - Select the AMI Amazon 2 Linux  - Select the instance type t2.micro  - Select Network as "Project 1 VPC" and subnet as "Public Subnet"  - For the security group, open the ports 80,443, 22 and 8065 for source set to "Anywhere"  4) Launch the instance after creating a new pem file and downloading it | | | |
| Expected screenshots | 1. AMI used 2. Instance configuration screen 3. Security group rules 4. Instance after creation |  |  |  |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step number | b |  |  |  |
| Step name | Creation of database server |  |  |  |
| Instructions | 1) Navigate to EC2 using the Services button at the top of the screen  2) Select Instances at the left side of the screen  3) Click on Launch Instance  - Select the AMI Amazon 2 Linux  - Select the instance type t2.micro  - Select Network as "Project 1 VPC" and subnet as "Private Subnet"  - For the security group, open the ports 80, 443,22 and 3306 for source set to "Anywhere"  4) Launch the instance by selecting the same pem file created in the previous step | | | |
| Expected screenshots | 1. AMI used 2. Instance configuration screen 3. Security group rules 4. Instance after creation |  |  |  |
|  | | | | |

**Step 4: Application and Database Installation and Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step number | a |  |  |  |
| Step name | Installation and configuration of MySQL |  |  |  |
| Instructions | 1) Copy the database pem file into the application server using the below command  *scp -i <application server pem file> <database server pem file > ec2-user@<application server public IP>:/home/ec2-user*  2) Log into the application server using SSH/Putty  3) From the application server, log into the database server using the pem file copied in step 1and the private IP address of the database server with the following command  *ssh -i <database server pem file> ec2-user@<private IP of database server>*  4) Enter the following commands to install and configure MySQL on the database server  *sudo yum update* *wget http://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm sudo yum localinstall mysql57-community-release-el7-9.noarch.rpm -y*  *sudo yum install mysql-community-server -y --nogpgcheck*  *sudo systemctl start mysqld.service*  Run the below command to retrieve a temporary password for MySQL *sudo grep 'temporary password' /var/log/mysqld.log | rev | cut -d" " -f1 | rev | tr -d "."*  Log in to MySQL with the below command and enter the above password when prompted *mysql -u root -p*  Enter the below command after you login to MySQL  *ALTER USER 'root'@'localhost' IDENTIFIED BY 'Password42!';*  Type ‘exit’ into the MySQL prompt and press Enter to exit out of the MySQL environment. Enter the below commands to complete the setup. Ignore any warning messages you receive.  *wget https://d6opu47qoi4ee.cloudfront.net/install\_mysql\_linux.sh*  *chmod 777 install\_mysql\_linux.sh*  *sudo ./install\_mysql\_linux.sh*  5) Type *exit* to exit the database server and go back to the application server | | | |
| Expected screenshots | 1. Installation of MySQL 2. Retrieving the temporary password 3. Executing the provided script |  |  |  |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step number | b |  |  |  |
| Step name | Installation and configuration of Mattermost |  |  |  |
| Instructions | 1) Enter the following commands after logging into the application server via SSH to install and configure Mattermost  *wget https://d6opu47qoi4ee.cloudfront.net/install\_mattermost\_linux.sh*  *sudo yum install dos2unix -y*  *sudo dos2unix install\_mattermost\_linux.sh*  *chmod 700 install\_mattermost\_linux.sh*  *sudo ./install\_mattermost\_linux.sh <private IP of MySQL server>*  Example : sudo ./*install\_mattermost\_linux* 173.65.34.7  *sudo chown -R mattermost:mattermost /opt/mattermost*  *sudo chmod -R g+w /opt/mattermost*  *cd /opt/mattermost*  *sudo -u mattermost ./bin/mattermost*  2) Check whether the server has been successfully deployed by navigating to the following URL in your web browser. The web page might take a couple of minutes to load.  <public IP of the application server>:8065 | | | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |  |  |  |
| Expected screenshots | 1. Executing the script 2. Starting the Mattermost server 3. Accessing the application via web browser |  |  |  |
|  | | | | |

**Step 5: Answer the following questions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Answer the following questions** | | | |  |
| Q1 | What is the default setting for DNS hostnames when a new VPC is created? | | |  |
|  | a) Enabled |  |  |  |
|  | b) Disabled |  |  |  |
|  | c) Can be set during VPC creation |  |  |  |
|  | d) Depends on the region used |  |  |  |
|  | Enter your answer here | b) |  |  |
|  |  |  |  |  |
| Q2 | What is the term used for the machine when we use it to log into the database server? | | |  |
|  | a) Bastion Host |  |  |  |
|  | b) NAT Gateway |  |  |  |
|  | c) Tunnel Interface |  |  |  |
|  | d) SSH Gateway |  |  |  |
|  | Enter your answer here | a) |  |  |
|  |  |  |  |  |
| Q3 | The database server security group in this exercise has to keep port 3306 open. Which protocol uses this port to communicate? | | |  |
|  | a) HTTPS |  |  |  |
|  | b) RDP |  |  |  |
|  | c) TCP |  |  |  |
|  | d) SCP |  |  |  |
|  | Enter your answer here | c) |  |  |
|  |  |  |  |  |
| Q4 | Which port is being used by Mattermost to communicate with the client application | | |  |
|  | a) 8080 |  |  |  |
|  | b) 80 |  |  |  |
|  | c) 443 |  |  |  |
|  | d) 8065 |  |  |  |
|  | Enter your answer here | d) |  |  |
|  |  |  |  |  |
| Q5 | Which of the following is a reason why we cannot set the CIDR block for the public subnet to 10.0.2.0/16, assuming the values for the other CIDR blocks are the same as mentioned in the instructions? | | |  |
|  | a) CIDR block overlaps with existing block |  |  |  |
|  | b) CIDR block is not a valid CIDR |  |  |  |
|  | c) CIDR block does not fall within the VPC |  |  |  |
|  | d) There is no reason, this is a perfectly valid CIDR |  |  |  |
|  | Enter your answer here | c) |  |  |
|  |  |  |  |  |
| Q6 | Assume that you have been asked to create 3 EC2 instances - application server, the database server and NAT instance. Each of these instances have their own security groups with a set of ports to be kept open. One of those ports is entirely unnecessary for the given architecture to function. Which of the ports given in the option below could it be? | | |  |
|  | a) Port 22 on the NAT instances |  |  |  |
|  | b) Port 3306 on the database server |  |  |  |
|  | c) Port 443 on the NAT instance |  |  |  |
|  | d) Port 22 on the application server |  |  |  |
|  | Enter your answer here | a) |  |  |
|  |  |  |  |  |
| Q7 | Describe the steps you would take to increase security of the servers you have deployed so that they are not reachable from external sources | | |  |
|  |  | | |  |
|  | To strengthen the security of the deployed servers and ensure they remain isolated from external access, I would take the following steps:   1. **Refine Security Group Rules:**    * Carefully adjust the security group rules for both the application and database servers. Limit inbound connections to only essential traffic, such as SSH, and specify source IP ranges that correspond to internal or trusted networks. 2. **Utilize a Private Subnet for the Database:**    * Move the database server to a private subnet, removing its direct exposure to the internet. This ensures that the database server is shielded from external access. 3. **Remove Public IP from Application Server:**    * If external access to the application server is unnecessary, eliminate the public IP address assignment. This can be achieved by relying solely on a private IP or implementing Network Address Translation (NAT) to control outbound traffic. 4. **Implement a Bastion Host for SSH Access:**    * Set up a bastion host in the public subnet and configure SSH access through this intermediary. This adds an extra layer of protection and restricts direct SSH access to the servers. 5. **Configure Network ACLs:**    * Leverage Network Access Control Lists (NACLs) to manage inbound and outbound traffic at the subnet level. Specify only essential protocols and ports, and regularly review and update these rules. 6. **Establish VPN or DirectConnect:**    * Implement a Virtual Private Network (VPN) or AWS Direct Connect to establish a secure connection between trusted networks and the AWS VPC. This ensures that server access is limited to users within these trusted networks. 7. **Enforce Multi-Factor Authentication (MFA) for SSH:**    * Enhance the security of SSH access by enabling multi-factor authentication. This requires users to provide a second form of verification in addition to their passwords. 8. **Regular Security Audits and Updates:**    * Conduct routine security audits to identify and address vulnerabilities. Keep the servers up-to-date by applying the latest security patches for the operating system, applications, and server software. 9. **Implement Encryption for Data:**    * Enable encryption for both data in transit and at rest. Use protocols like TLS for encrypting web traffic and technologies like AWS EBS encryption for securing stored data. 10. **Set Up Monitoring and Logging:**     * Deploy monitoring and logging solutions to track and analyze server activities. Configure alerts for suspicious activities, failed login attempts, or any deviations from normal patterns.   Adopting these measures contributes to an ongoing and robust security posture, ensuring that the servers remain resilient against potential threats. Regular reviews and updates to security measures are imperative to address evolving risks and challenges. | | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Q8 | Describe the steps required to deploy the given application in an autoscaling environment | | |  |
|  |  |  |  |  |
|  | Deploying an application in an autoscaling environment involves setting up infrastructure that can dynamically adjust to varying workloads. Below are the steps to deploy the given application in an autoscaling environment:   1. **Design for Statelessness:**    * Ensure that the application architecture supports statelessness. Each instance of the application should be able to handle requests independently, allowing for easy horizontal scaling. 2. **Containerization (Optional):**    * Consider containerizing the application using technologies like Docker. Containerization provides consistency across different environments and simplifies the deployment process. 3. **Create Amazon Machine Images (AMIs):**    * Prepare a custom Amazon Machine Image for the application servers. This AMI should include the necessary dependencies, configurations, and the application code. 4. **Set Up an Application Load Balancer (ALB):**    * Create an Application Load Balancer to distribute incoming traffic across multiple instances. This ensures that the load is evenly distributed and provides high availability. 5. **Configure Auto Scaling Groups:**    * Set up Auto Scaling Groups (ASG) for both the application and database servers. Define policies for scaling in and scaling out based on metrics like CPU utilization, request latency, or custom application metrics. 6. **Define Launch Configuration:**    * Specify a launch configuration for the Auto Scaling Group, referencing the custom AMI created earlier. This configuration includes details such as instance type, security groups, and user data scripts. 7. **Database Scaling (Optional):**    * If the database server is part of the autoscaling environment, consider using Amazon RDS with read replicas to handle database scaling. This ensures that the database layer can also dynamically adapt to changing workloads. 8. **Implement Health Checks:**    * Configure health checks for the Auto Scaling Group. The health checks determine the health of each instance, and instances that fail health checks are replaced automatically. 9. **Monitoring and Alarming:**    * Set up CloudWatch alarms to monitor key metrics such as CPU utilization, network traffic, and application response times. Define alarms to trigger Auto Scaling actions when thresholds are breached. 10. **Integration with AWS Elastic Load Balancer (ELB):**     * Integrate the Auto Scaling Group with the Application Load Balancer to ensure that new instances automatically register with the load balancer and are included in the distribution of incoming traffic.   By following these steps, the given application in an autoscaling environment that adapts dynamically to changing workloads, providing scalability, high availability, and efficient resource utilization. | | |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |
| --- | --- |
| **Grades distribution** |  |
| MCQs | 6 (1 mark each) |
| Subjective questions | 20 marks (10+10) |
| Implementation screenshots | 24 marks (1 marks each) |
| Total | 50 marks |