MSDS 7346
Cloud Computing
Midterm Exam

MidTerm Exam

This is a midterm exam for MSDS 7346, Cloud Computing. This is a hands-on exercise that builds on what we have done in the class. Please submit screenshots of all of the steps along with configurations. If there is something that you cannot make work, please provide details around steps that you have taken to resolve the issue.

Purpose

The purpose of this exercise is to get hands-on experience working with AWS, more specifically I am looking for you to create a fully functional configuration of Application Servers, Load Balancer, RDS, and S3. We have worked with all of these pieces during our class with load balancer as the only exception.. This lab brings all of the pieces of a puzzle together.

Application Servers – are used to host application such as your website and provide service to external users.

Load Balancer – is used to distribute traffic between application server nodes. In this exercise you will configure classic load balancer that will direct user request in a round robin fashion (alternate between different EC2 instances.)

RDS – is used to store data. In this exercise we will not be storing or retrieving any data as we will not host any application. The objective is connectivity of these building blocks.

S3 – object storage is created for the purpose of backup.

Description

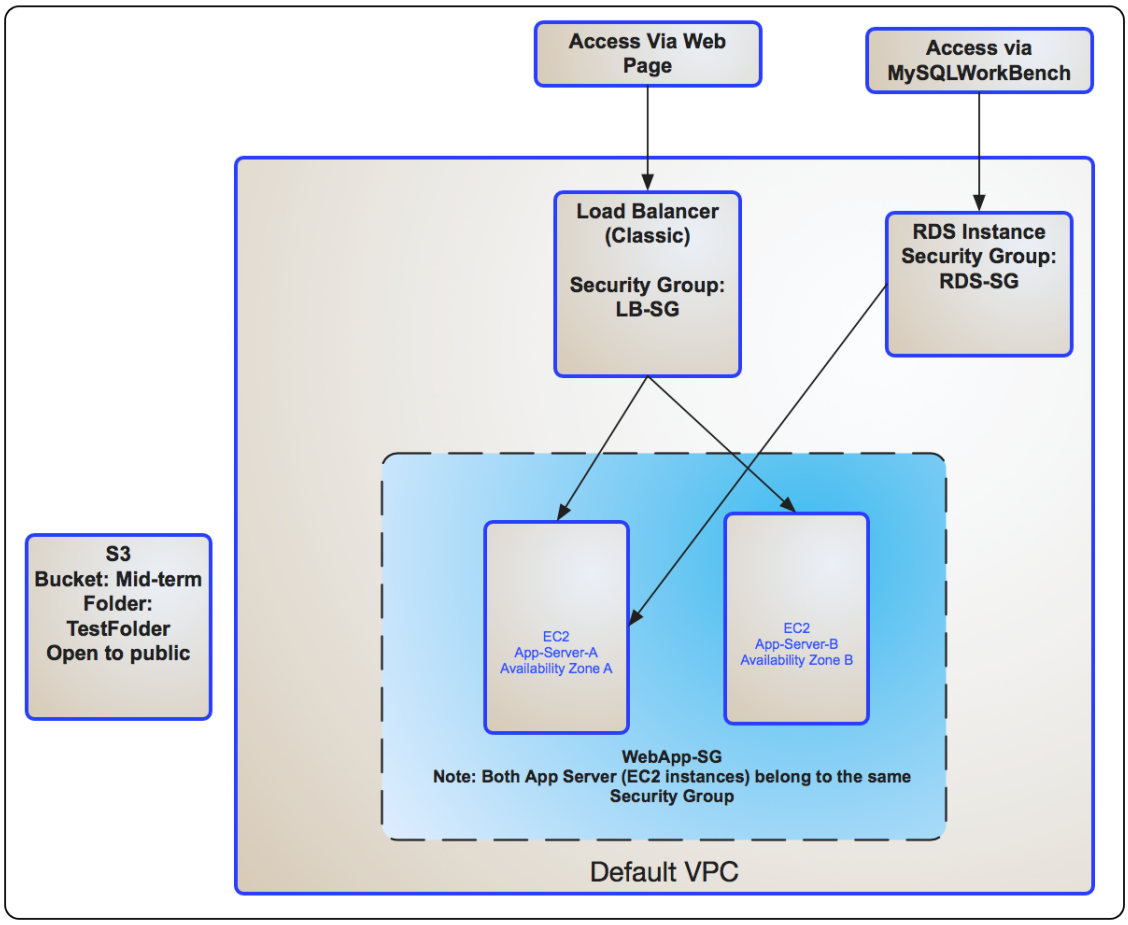
The following are some of the steps that you need to do to complete this lab. This in no way intended to be a complete list, but I tried to provide as much explanation as possible. You are expected to have this complete system working. Towards the end of this document, you will find screenshots from my AWS account to give you some idea of different configurations. In this mid-term, not only you would setup AWS, but will also install a ”Hello World” HTML file, simulating a web application on EC2 instances. This web app will be accessed via browser using the DNS address of the load balancer. This will demonstrate how user gets access to an web application running on EC2 instance and in addition provides insight into how the load balancer works. You also see a MySQLWorkbench on your local machine and this is simply there to test connection to the RDS instance and also to validate if your database command on EC2. Figure 1 below should give you an the architecture of the final configuration would look like.

1) **EC2**
   - Create two EC2 instances (use free tier) for each of these two use the default VPC (Virtual Private Cloud).
   - Create a new Security Group which will allow all traffic from any source. You will need to only create this when you launch the first instance, for the second instance you will select existing Security Group.
   - Create these instances in different availability zones. To make it easy, name them App-Server-A for instance in availability zone A and App-Server-B for availability zone B.
   - Verify that you are able to connect to both EC2 instances from your machine using ssh.

2) **Security Group**
   - Create a new Security Group which will allow all traffic from any source. You will need to only create this when you launch the first instance, for the second instance you will select existing Security Group.

3) **Availability Zone**

Fig 1 – Architecture of Configuration in AWS



- Each region, as we discussed in the class, by default has three zones.
- For example if you logged into US West 2 region then you will have us-west-2a, us-west-2b, us-west-2c availability zones available to you.
- Create both of the EC2 instances in two different availability zones.
- To make it easy to remember, name each of these EC2 instances as App-Server-A for instance in availability zone A and App-Server-B for availability zone B.

4) **Web Application**

- Now that you have verified these instances. Install httpd on your EC2 instances.
- Ye need this to run a basic html file (simulating a Web Application).
- Use following instructions as a guideline:
  - Log into EC2 instances one at a time via SSH
  - Once you are in EC2 intance use ”yum” to install httpd:
    sudo yum install -y httpd
  - install the index.html (provided seprately) in /var/www/html directory
  - Modify the # sign in the file to reflect the instance number For example ”Mid-term MSDS 7336, displaying App Server #” should be either A for App-Server-A and B for App-Server-B
  - Now you start the server:
    sudo service httpd start
- At this time if you take the dns address of your EC2 instance and put that in the browser you would see the ”Mid-term ... ” message.
  - **NOTE**: If you do not see this message then make sure httpd is running and your file is in the correct directory.

5) **Load Balancer**

- Configure Classic Load Balancer using the default VPC. You should allow all http traffic

- Add both nodes (instances) to the Load Balancer
- Make sure health check is setup.
- You should see the status of both instances go from OutOfService to InService. It takes couple of minutes for the status to change.
- Put the DNS name that you find under "Description" tab of the Load Balancer in a browser you will see the above message from html file display.
- Refresh your browser and you will see it go to the next EC2 instance. Keep refreshing it to see different messages.

6) **RDS**
   - Configure RDS intance
   - Once RDS is configured, go to one of the EC2 instance and install MySQL client as it does not come installed by default. (Note: Only do this on one EC2 instance)
   - Use "yum" command to do that. Exact syntax as I showed earlier for httpd except now it would be MySQL.
   - Login to the MySQL using the following command:
     mysql -h rds.endpoint.aws.com -u username -p
   - One you are there create a database using the create database command
   - Now go to the MySQLWorkbench connect to the RDS and you should see that database.

Submission

The following is what you need to submit.

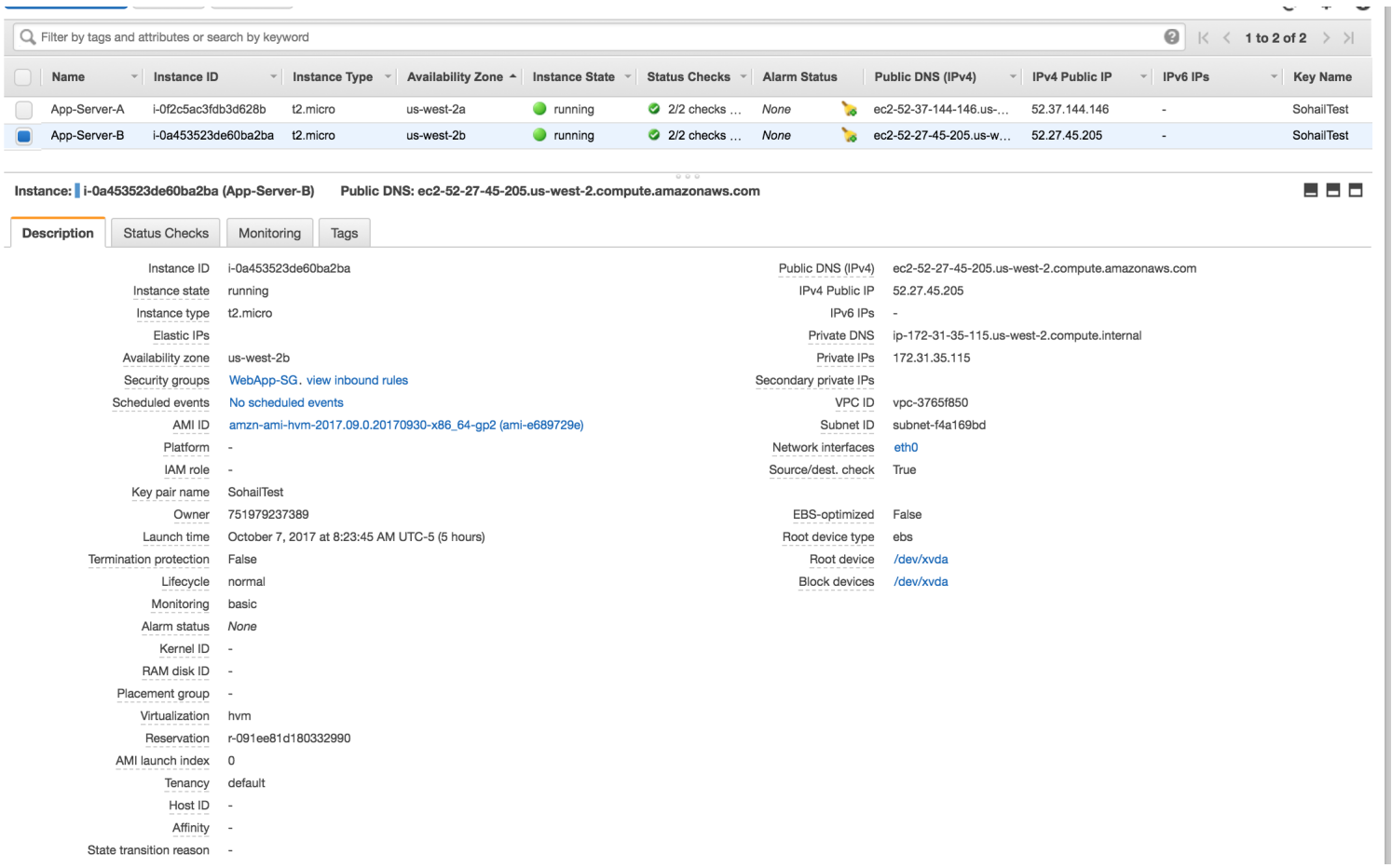
- Provide few bullet points or a short paragraph of what you were able to work, if you had problems or couldn't get some things to work please provide brief explanation here.
- Submit screenshots of all of the different configurations depicting the architecture picture in Figure 1. I also want to see the screenshot your web application (html page) showing different application server names. Arrange these screenshots with proper labels and explanation for each screenshot to make is clear for the grader.
- If you still run into any problems when it is time to submit, please explain what steps you have taken and provide screenshots for all of those steps.
- Good Luck and Enjoy!!

Due Date

- The mid-term is due at **6:30pm Central time on Wednesday October 18, 2017**.
- **Note**: Unless agreed upon ahead of time, there will be no extension to this. **NO** submission via email. No grade will be provided for late submission.

Screenshots

Application Server Configuration



Filter by tags and attributes or search by keyword — 1 to 2 of 2

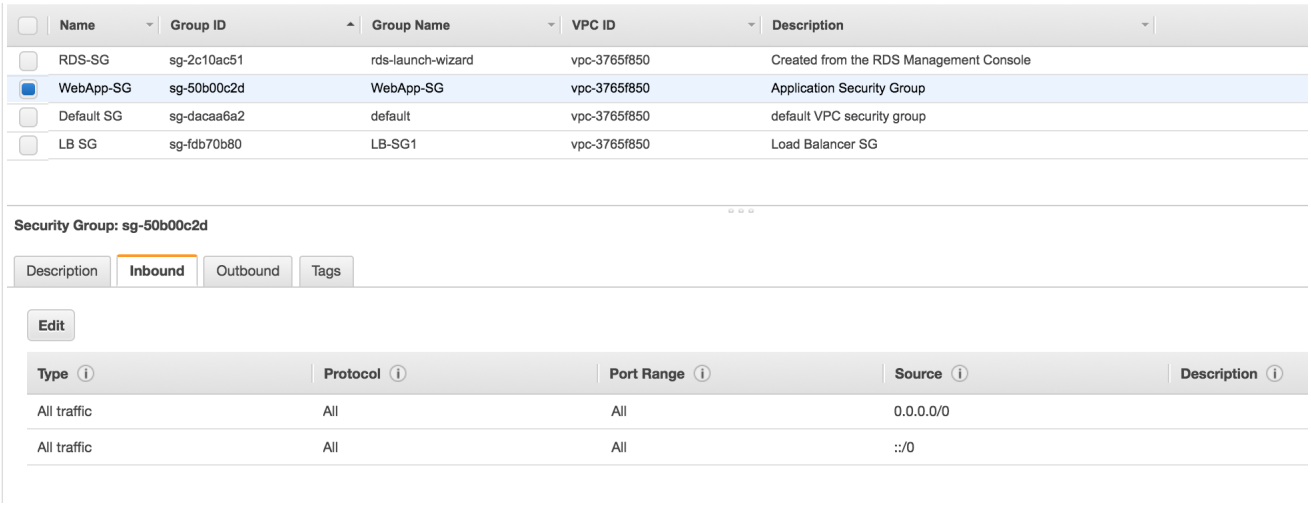
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name
App-Server-A	i-0f2c5ac3fdb3d628b	t2.micro	us-west-2a	running	2/2 checks …	None	ec2-52-37-144-146.us-…	52.37.144.146	-	SohailTest
App-Server-B	i-0a453523de60ba2ba	t2.micro	us-west-2b	running	2/2 checks …	None	ec2-52-27-45-205.us-w…	52.27.45.205	-	SohailTest

Instance: i-0a453523de60ba2ba (App-Server-B) **Public DNS: ec2-52-27-45-205.us-west-2.compute.amazonaws.com**

Description | Status Checks | Monitoring | Tags

Instance ID	i-0a453523de60ba2ba	Public DNS (IPv4)	ec2-52-27-45-205.us-west-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	52.27.45.205
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-35-115.us-west-2.compute.internal
Availability zone	us-west-2b	Private IPs	172.31.35.115
Security groups	WebApp-SG. view inbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-3765f850
AMI ID	amzn-ami-hvm-2017.09.0.20170930-x86_64-gp2 (ami-e689729e)	Subnet ID	subnet-f4a169bd
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True
Key pair name	SohailTest		
Owner	751979237389	EBS-optimized	False
Launch time	October 7, 2017 at 8:23:45 AM UTC-5 (5 hours)	Root device type	ebs
Termination protection	False	Root device	/dev/xvda
Lifecycle	normal	Block devices	/dev/xvda
Monitoring	basic		
Alarm status	None		
Kernel ID	-		
RAM disk ID	-		
Placement group	-		
Virtualization	hvm		
Reservation	r-091ee81d180332990		
AMI launch index	0		
Tenancy	default		
Host ID	-		
Affinity	-		
State transition reason	-		

AppServer Security Group



Name	Group ID	Group Name	VPC ID	Description
RDS-SG	sg-2c10ac51	rds-launch-wizard	vpc-3765f850	Created from the RDS Management Console
WebApp-SG	sg-50b00c2d	WebApp-SG	vpc-3765f850	Application Security Group
Default SG	sg-dacaa6a2	default	vpc-3765f850	default VPC security group
LB SG	sg-fdb70b80	LB-SG1	vpc-3765f850	Load Balancer SG

Security Group: sg-50b00c2d

Description | Inbound | Outbound | Tags

Edit

Type	Protocol	Port Range	Source	Description
All traffic	All	All	0.0.0.0/0	
All traffic	All	All	::/0	

Load Balancer Security Group

Name	Group ID	Group Name	VPC ID	Description
RDS-SG	sg-2c10ac51	rds-launch-wizard	vpc-3765f850	Created from the RDS Management Console
WebApp-SG	sg-50b00c2d	WebApp-SG	vpc-3765f850	Application Security Group
Default SG	sg-dacaa6a2	default	vpc-3765f850	default VPC security group
LB SG	sg-fdb70b80	LB-SG1	vpc-3765f850	Load Balancer SG

Security Group: sg-fdb70b80

Description | **Inbound** | Outbound | Tags

Edit

Type	Protocol	Port Range	Source	Description
All traffic	All	All	0.0.0.0/0	

Load Balancer Instances

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monito
Class-LB	Class-LB-817349688.us-wes…		vpc-3765f850	us-west-2a, us-west-2b…	classic	October 7, 2017 at 7:41:19 A…	

Load balancer: Class-LB

Description | **Instances** | Health Check | Listeners | Monitoring | Tags

Connection Draining: Enabled, 300 seconds (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
i-0a453523de60ba2ba	App-Server-B	us-west-2b	InService	Remove from Load Balancer
i-0f2c5ac3fdb3d628b	App-Server-A	us-west-2a	InService	Remove from Load Balancer

Edit Availability Zones

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
us-west-2a	subnet-9eb80ef9	172.31.16.0/20	1	Yes	Remove from Load Balancer
us-west-2b	subnet-f4a169bd	172.31.32.0/20	1	Yes	Remove from Load Balancer
us-west-2c	subnet-52487c0a	172.31.0.0/20	0	No (Availability Zone contains no healthy targets)	Remove from Load Balancer

Load Balancer Health Check

Filter: Search 1 to 1 of 1

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monito
Class-LB	Class-LB-817349688.us-wes…		vpc-3765f850	us-west-2a, us-west-2b…	classic	October 7, 2017 at 7:41:19 A…	

Load balancer: Class-LB

Description | Instances | **Health Check** | Listeners | Monitoring | Tags

Ping Target HTTP:80/index.html
Timeout 5 seconds
Interval 30 seconds
Unhealthy threshold 2
Healthy threshold 10

Edit Health Check