Cloud Computing Architecture

Assignment 2

Creating and deploying Photo Album website onto a simple OCI infrastructure

Due date: 9 AM (AEST), Monday of Week 10

Prerequisite requirements:

- Successfully completed Assignments 1A and 1B.
- Completed all OCI Labs (weeks 7-9)
- Knowledge of using OCI Network Security Groups.

All supporting materials mentioned in this document can be found in the corresponding assignment page on Canvas.

The PHP source code has been provided for this assignment. However, you will need to understand how this code works to be able to modify the missing parts. Each student is supposed to add their own specific information in this code; hence, you must not copy someone else's code.

Objectives

This assignment has the following objectives:

1. Create a secure Virtual Cloud Network (VCN) with appropriate subnets, routing tables and security services.
2. Control access to and from your VCN via an Internet Gateway.
3. Modify the provided PHP code to create a website that stores meta-data information about photos uploaded to Object Storage in a MySQL database. The website should enable the user to search for and display photos using meta-data.
4. Deploy and test your PHP web site on an Apache web server running on an virtual machine instance.
5. Add an aditional layer of security by applying a Network Security Group to the web server.

1. Infrastructure deployment

You will set up a VCN with the structure and services as illustrated in the diagram below.

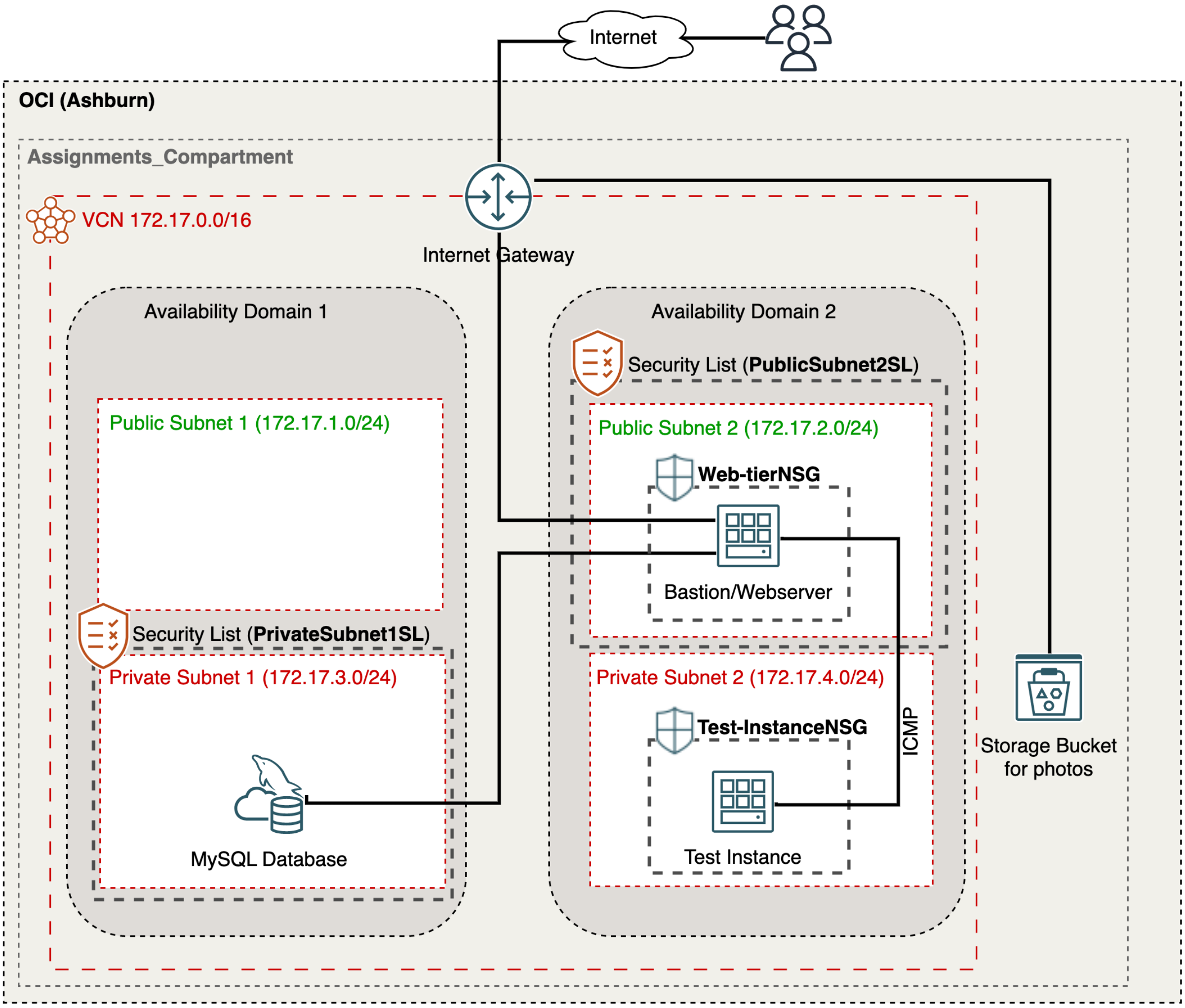
Internet
OCI (Ashburn)
Assignments_Compartment
VCN 172.17.0.0/16
Internet Gateway
Availability Domain 1
Availability Domain 2
Security List (PublicSubnet2SL)
Public Subnet 1 (172.17.1.0/24)
Public Subnet 2 (172.17.2.0/24)
Web-tierNSG
Bastion/Webserver
Security List (PrivateSubnet1SL)
Private Subnet 1 (172.17.3.0/24)
MySQL Database
Private Subnet 2 (172.17.4.0/24)
Test-InstanceNSG
ICMP
Test Instance
Storage Bucket for photos

Figure 1: Architecture Diagram (OCI)

NOTE:

- All resources must be created in "**Assignments_Compartment**" compartment
- Do not use the default VCN (if any). All services should be in your custom VCN. Below are the detailed requirements for each service.

1.1 - VCN:

- Name: [*FirstNameInitial*][*LastName*]*VCN*. For example, if your name is Bill Gates, your VCN would be named "*BGatesVCN*".
- Region: us-ashburn-1
- Two availability domains each with a private and public subnet with suitable CIDR as specified in the diagram above.
- Associate public subnets with a public route table that routes to the Internet Gateway.

- Associate private subnets with a private route table.

NOTE: due to some incompatibility issues, it is recommended to create your VCN manually (use the "**Create VCN**" button in VCN tab). Please do NOT use the "**Start VCN Wizard**" button in OCI dashboard.

1.2 – Network Security Groups

Create the following network security groups, each is associated with each tier shown in the architecture diagram:

NSG Name	Ingress	Egress
Test-InstanceNSG	All traffic from Anywhere	ICMP to Public Subnet 2
Web-tierNSG	HTTP (80) from Anywhere	
	SSH (22) from Anywhere	
	ICMP from Private Subnet 2	ICMP to Private Subnet 2
	TCP (All ports) from Private Subnet 1	

Table 1: Network Security Groups

NOTE: by default, the "Default Security List" is attached to all subnets in Oracle Cloud. To simplify the process, you can REMOVE all Ingress and Egress rules of this default Security list to make NSG rules effective, for Private Subnet 2 of this assignment.

1.3 – Compute (Virtual machines)

You will create two instances, a test instance and a bastion/web server instance.

1.3.1 – Bastion/Web server instance

Your web server must be deployed on an instance in Public Subnet 2. This instance should be configured similar to the instance created in Assignment 1:

- Machine Image: ***Oracle-Linux-8***
- Instance type: ***VM.Standard1.1 (1 OCPU, 7GB Memory)***
  OR
  VM.Standard.E2.1 (1 OCPU, 8GB Memory)
  OR
  VM.Standard3.Flex (1 OCPU, 6GB Memory)
  OR
  VM.Standard.A1.Flex (1 OCPU, 6GB Memory)
- Has Apache web server and other PHP packages installed (bash script provided in the Assignment page).

This instance will host the "Photo Album" web application, which was created in Assignment 1 – more details are in Section 2 of this specification document. This instance will also act as a bastion host for you to SSH into the Test instance, which resides in a private subnet.

NOTE: [WebServer_public_IP_Address] will change every time the webserver instance restarts. To avoid this behavior and to ensure your Webserver public IP address remains persistent, add a Reserved Public IP Address to this instance by allocating a Reserved Public IP address in the same region under the Networking/IP Management section.

Once the IP address is reserved, go to the WebServer instance details and navigate to "**Attached VNICs/IPv4 Addresses**". Click on the 3-dot menu and select "**Edit**". In this menu you should be able to Unassign the attached Public IP address first by choosing "**No public IP**", update and then selecting the "**Reserved public IP**".

1.3.2 – Test instance

This instance will be used for demonstration purposes only. It does not contribute to the functionality of Photo Album website.
You will SSH into the web server and test its bidirectional/two-way connectivity to this Test instance (using "ping" command in Linux). You must be able to establish a connection (ICMP ping) between this instance and the Bastion/Web server instance that are running in two separate subnets.

The configuration of this instance is entirely your choice. This instance does not host the web application.

1.4 – MySQL Database

Your database instance must have the following configs:

- DB engine version:	*MySQL 8.0.41*
- Type:	*Standalone (Development or testing)*
- Shape:	*MySQL2 (ECPU model)*
- HeatWave cluster:	*Disabled*
- Data Storage Size:	*50GB*
- Automatic Backup:	*Disabled*
- Administrator user:	*admin*
- Admin password:	*Oraclecloud@2023*
- Resides in private subnet as per architecture diagram	

NOTE: your database needs to be in a private subnet. Only Public Subnet 2 can access it. However, you need to be able to access your database over the internet so that you can set it up and maintain it. There are several ways to do this. The easiest way is to install phpMyAdmin (a web-based MySQL administration tool) on your web server instance and manage your database through phpMyAdmin's UI. Instructions on how to do this are in *Install_phpMyAdmin_OCI.pdf* file.

Create a database in your database instance with a table called ***photosDB*** that stores meta-data of the photos stored in the object storage bucket. Create a table inside this database that has the following columns:

- photo_title (*varchar(255)* type)
- description (*varchar(255)* type)
- creation_date (*date* type)
- keywords (*varchar(255)* type)
- object_reference (*varchar(255)* type)

1.5 – Security Lists

PublicSubnet2SL:

To add an additional layer of security to your web server, you have been asked to design and deploy a security list (named "PublicSubnet2SL") that limits ICMP and other necessary traffic to the associated instance (Bastion/Webserver). This Security List must follow the least-privilege principle. In other words, irrelevant traffic from irrelevant sources must not be allowed. To be specific, the security list:

- must ALLOW **SSH(22)** [1]traffic from anywhere so that you can access the WebServer instance from your computer.
- must ALLOW **ICMP** traffic *only* from the subnet that contains the Test instance.
- must ALLOW other necessary traffic so that the Photo Album website is fully functional for users from anywhere and can access MySQL database.

NOTE: default "Security List" is attached to all subnets in Oracle Cloud. Ensure this security list is removed from Public Subnet 2.

PrivateSubnet1SL:

This security list is applied to the subnet that MySQL database exists. Add the required security list rules to ensure required instances (e.g. Webserver) can successfully communicate with the database in this subnet in a **least-privilege access model**.

NOTE: default "Security List" is attached to all subnets in Oracle Cloud. Ensure this security list is removed from Private Subnet 1.

2. Functional requirements of Photo Album website

Your Photo Album website must have the following functional requirements.

2.1 – Photo storage

Create an object storage bucket to store your photos. Manually upload some photos onto this bucket that you just created and ensure they have been successfully uploaded.

All objects (photos) in this bucket must become publicly available. This can be done via OCI management console, changing the buckets visibility settings.

NOTE: marks will be deducted if the bucket objects have been individually configured to be publicly available.

[1] *Ideally, SSH(22) traffic should only be allowed from your home network's public IPv4 address range since common users do not need to access the web server. But for simplicity, you can allow SSH from anywhere in this assignment.*

2.2 – Photo meta-data in MySQL Database

The meta-data of the photos stored in the S3 bucket is stored in a database table, which has been created in Section 1.4. You need to populate the table with a few records. Below is an example of a record:

- Photo title:	*Swinburne Logo*
- Description:	*Logo of Swinburne uni*
- Creation date:	*2021-08-09*
- Keywords:	*logo, university*
- Object Ref/URL:	*https://photo-bucket.oraclecloud.com/swinburnelogo.jpg*

NOTE: Object URL = Pre-Authenticated Request, generated by OCI storage bucket.

2.3 – Photo Album website functionality

The website must be able to list all the photos (stored in the bucket) along with their meta-data (stored in the database). The full source code has been provided to you (*photoalbum_OCI_v3.0.zip*). Modify the *constants.php* file in the provided code (carefully read the comments in the file) using available information from the storage bucket and MySQL database that you created in the previous steps.

NOTE: in constants.php file, variables/names that you add must not include space. You may use underscore "_" instead of space.

Examples:

```
define('DB_PHOTO_CREATIONDATE_COL_NAME', 'creationdate');
define('DB_PHOTO_KEYWORDS_COL_NAME', 'keywords_column');
```

The website should be accessible through http://[*WebServer_Public_IP_Address*]/cos80001/photoalbum/album.php if the directory structure in your web server is correctly created.

Testing

Manually upload several photos to the storage bucket and insert their meta-data into the database. Thoroughly test to make sure the photos and their meta-data are correctly displayed.

Ensure the Security List satisfies the additional security requirement, by login into the Test instance (e.g. via SSH) and run a ping to the web server's IP address.

Submission

Make sure your website is functional from the due date - check you can start the web server instance and stop it. (No need to start the Test instance).

Submission is a single PDF document to Canvas. No demonstration is required. The document must contain the following:

1. A **single PDF document**, **maximum 15 pages,** in **IEEE Conference Style** in either one or two column mode submitted to Canvas by the due date.
2. Title page with your name, student ID, and tutorial class.
3. URL of the album.*php* pages on your EC2 so your marker can view your website from their browser using the URL that you have provided (Reserved Public IP address to be used).
4. **If you have not created an instructor user in previous weeks, refer to week 8 lab instrucitons and ensure you have finished the lab, provided IAM admin user and the account details (account/tenancy name) to your tutor for marking purposes.**
5. Well formatted Screenshot(s) of the data records in your database.
6. Well formatted Screenshot(s) of Linux terminal showing you have been able to ping the Web server Instance from your Test instance, or vice versa.
7. Well formated screenshot(s) and a brief explanation for each step that you have taken, problems that you faced and achievements during your deployment for this assignment.
    - Route tables with their route rules and associated subnets
    - MySQL Database configurations/settings
    - Security List Rules
    - Network Security Group Rules
8. Each screenshot must have your OCI Management Console username/student ID visible.
9. All screenshots must:
    - Have a Caption (e.g. Figure 2: MySQL table)
    - Be readable (reasonably high quality)

NOTE:

This assignment is to be completed in your free-tier OCI account, which is accessible through Oracle Cloud Login page. For further information of how to access this environment please refer to your Swinburne Canvas Page “**Accessing OCI Resources**”.

This environment comes with $300 USD credits. It is your responsibility to use and manage this credit correctly to ensure there will be enough remaining credits for all assignments.

Marks will be deducted if your assignment resources are not accessible due to insufficient credits.

Assignment 2 Checklist

Make sure all the following task are successfully completed.

Submission Checklist

Student Name: ………………………………………………

Student Id: ……………………………………………………

Tutorial time: ………………………………………………

Date of submission: ……………………………………

OCI account/tenancy name: ……………………….

Submit to Canvas:

- □ A PDF document file as specified in the Submission section of the assignment specification.
- □ Public URL has been provided in the document.
- □ IAM user has been created and its details have been provided in the document.

Marking Scheme

Infrastructure Requirements		
VCN with 2 public and 2 private subnets	6	
Correct Public and Private Routing tables with correct subnet associations	10	
Network Security groups properly configured and attached.	13	
Security List properly configured and attached	10	
Correct Web server and Test instances running in correct subnets	4	
Database schema as specified (Version, Shape, Type)	10	
Database running in correct subnet(s)	8	
bucket objects publicly accessible, at the bucket level	6	
Functional Requirements		
album.php page displayed from Web server	10	
Provided URL is persistent (Reserved Public IP Association)	10	
Photos loaded from bucket with matching metadata from MySQL DB	6	
Web server instance reachable from Test instance via ICMP	7	
Deductions		
Documentation not as specified or poorly presented (up to minus 100)		
Serious misconfigurations of OCI services being used (up to minus 100)		

Comments