EXPERIMENT NO. 2

Aim :To study and Implement Platform as a Service using AWS Elastic Beanstalk/ Microsoft Azure App Service.

Hardware:	Device Name :- computer-ThinkCentre Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3 Memory :- 8.0 Gib Processor :- 12th Gen Intel® CoreTM i5-12400 × 12 Disk Capacity :- 256.1 GB
Software :	Operating System :- LINUX Amazon AWS

Theory : The objective of this experimrnt is to demonstrate the steps to deploy Web applications or Web services written in different languages on AWS Elastic Beanstalk/ Microsoft Azure App Service.

An Amazon EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure.

Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instances and choice of the latest processor, storage, networking, operating system, and purchase model to help you best match the needs of your workload.

We are the first major cloud provider that supports Intel, AMD, and Arm processors, the only cloud with on-demand EC2 Mac instances, and the only cloud with 400 Gbps Ethernet networking.

It offers the best price performance for machine learning training, as well as the lowest cost per inference instances in the cloud. More SAP, high performance computing (HPC), ML, and Windows workloads run on AWS than any other cloud.

Algorithm:

Step1 : Login to AWS console and go to Elastic Beanstalk

Step 2: Click on Create Application

Step 3: Write Application information : Name, Tag,Platform etc.

Step 4: In Application Code: select sample application and then Click on button Create Application
This will take a few minutes.

Step 5: Click on Environments -> Check the health of Environment wait till it becomes ‘OK

Step 6: Click the URL

Step 7: To Delete the application and Environment (Select it and in Action -Delete/Terminate : give conformation)

Conclusion :Thus, in this practical we have studied and implemented Platform as a service (PaaS) using AWS Beanstalk.

EXPERIMENT NO. 1

Aim :To study and Implement Infrasture as a Service using AWS/Microsoft Azure.

Hardware: Device Name :- computer-ThinkCentre
Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3
Memory :- 8.0 Gib
Processor :- 12th Gen Intel® CoreTM i5-12400 × 12
Disk Capacity :- 256.1 GB

Software : Operating System :- LINUX
Amazon AWS

Theory : The objective of this experimrnt is to demonstrate the steps to dcreate and run Virtual machines inside a Public cloud platorm.

An Amazon EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure.

Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instances and choice of the latest processor, storage, networking, operating system, and purchase model to help you best match the needs of your workload.

We are the first major cloud provider that supports Intel, AMD, and Arm processors, the only cloud with on-demand EC2 Mac instances, and the only cloud with 400 Gbps Ethernet networking.

It offers the best price performance for machine learning training, as well as the lowest cost per inference instances in the cloud. More SAP, high performance computing (HPC), ML, and Windows workloads run on AWS than any other cloud.

Amazon EC2 provides the broadest and deepest instance choice to match your workload’s needs. General purpose, compute optimized, memory optimized, storage optimized, and accelerated computing instance types are available that provide the optimal compute, memory, storage, and networking balance for your workloads.

Processors from Intel, AMD, NVIDIA and AWS power these instance types and provide additional performance and cost optimizations.

Local storage and enhanced networking options available with instance types further help optimize performance for workloads that are disk or network I/O bound.

Conclusion : Thus, in this practical we have studied and implemented Infrastructure as a Service using AWS.

EXPERIMENT NO. 3

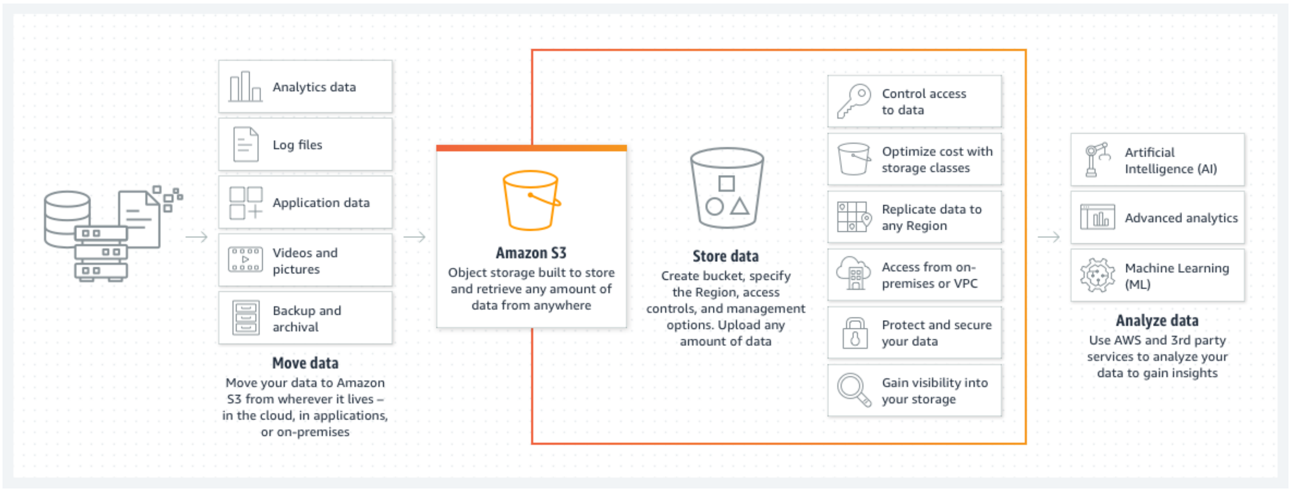
Aim :To study and Implement Storage as a Service using Own Cloud/ AWS S3, Glaciers/ Azure Storage.

Hardware: Device Name :- computer-ThinkCentre
Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3
Memory :- 8.0 Gib
Processor :- 12th Gen Intel® CoreTM i5-12400 × 12
Disk Capacity :- 256.1 GB

Software : Operating System :- LINUX
Amazon AWS

Theory : The objective To understand the concept of Cloud storage and to demonstrate the different types of storages like object storage, block level storages etc. supported by Cloud Platforms like Own Cloud/ AWS S3, Glaciers/ Azure Storage.

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as data lakes, cloud-native applications, and mobile apps. With cost-effective storage classes and easy-to-use management features, you can optimize costs, organize data, and configure fine-tuned access controls to meet specific business, organizational, and compliance requirements.


Analytics data
Log files
Application data
Videos and pictures
Backup and archival
Move data
Move your data to Amazon S3 from wherever it lives – in the cloud, in applications, or on-premises
Amazon S3
Object storage built to store and retrieve any amount of data from anywhere
Store data
Create bucket, specify the Region, access controls, and management options. Upload any amount of data
Control access to data
Optimize cost with storage classes
Replicate data to any Region
Access from on-premises or VPC
Protect and secure your data
Gain visibility into your storage
Artificial Intelligence (AI)
Advanced analytics
Machine Learning (ML)
Analyze data
Use AWS and 3rd party services to analyze your data to gain insights

Algorithm:

Step-1: click on create bucket

Step-2: Give Bucket name & select region for storage

Step-3: Keep object ownership setting as ACLs Disabled as by-default

Step-4: Disable block all public access checkbox

Step-5: Select the checkbox for Turning off block all public access might result in this bucket and the objects within becoming public

Step-6: Keep bucket versioning as disabled and add tags if required.

Step-7: Keep default encryption disabled and click on create bucket button

You can now see the successful creation of your bucket

Step-8: now click on the bucket that you have created

Step-9: You can either create a folder here or upload an existing file in the bucket

Step-10: now click on upload button and click on add files button browse your local machine and select which file you need to upload on S3 next click on upload button at bottom right end

Now you can check the upload status screen

Now click on close button The screen will appear as below

Step-11: Select properties and scroll down to Static website hosting option which is disabled now click on Edit option on right side

Step-12: Enable the radio button and specify the file name in Index document which you have added in S3

Scroll down and save the changes at bottom right Following screen will appear

Step-13: Click on Permissions Tab

Step-14: In bucket policy click on Edit option

Step 15- after clicking on edit button paste the following code in bucket policy

```
{
 "Version": "2012-10-17",
"Statement": [
{ "Sid": "PublicReadGetObject",
"Effect": "Allow",
"Principal": "*",
"Action": [ "s3:GetObject"
],
"Resource": [
"arn:aws:s3:::Bucket-Name/*" ]
 }
 ]
 }
```

Note-Make sure that you add your bucket name in the code above.

Scroll down and click on Save Changes button

Step-16: open your html file and click on Object URL

Step-17: Now for delete files click on checkbox of your file and then click on Delete Button.

Write permanently delete and click on delete object button

Now click on close button

Step-18: now come to Amazon S3 tab and select your bucket and then click on delete button

Write down your bucket name in delete bucket tab and click on delete button at bottom right

You can see that the bucket is deleted.

Create Own Cloud:

Step 1: sudo apt install net-tools
Step 2:ifconfig
Step 3:sudo apt-get update

Install Apache2 HTTP Server

Install Apache2 on Ubuntu by running the following commands:
sudo apt install apache2
After installing Apache2, run the commands below to disable directory listing.
sudo sed -i "s/Options Indexes FollowSymLinks/Options FollowSymLinks/" /etc/apache2/apache2.conf
After install Apache2, the commands below can be used to stop, start and enable Apache2 service to
always start up with the server boots.
sudo systemctl stop apache2.service
sudo systemctl start apache2.service
sudo systemctl enable apache2.service

Step 5: Install MariaDB Server

To install MarisDB run the commands below.
sudo apt-get install mariadb-server mariadb-client -y
The commands below is use to stop, start and enable MariaDB service to always start up when the server boots.

sudo systemctl stop mariadb.service
sudo systemctl start mariadb.service
sudo systemctl enable mariadb.service
After that, run the commands below to secure MariaDB server.
sudo mysql_secure_installationWhen prompted, answer the questions below by following the guide.

- Enter current password for root (enter for none): Just press the Enter
- Set root password? [Y/n]: Y
- New password: Enter password
- Re-enter new password: Repeat password
- Remove anonymous users? [Y/n]: Y
- Disallow root login remotely? [Y/n]: Y
- Remove test database and access to it? [Y/n]: Y
- Reload privilege tables now? [Y/n]: Y

Step 6: Restart MariaDB server
sudo systemctl restart mariadb.service
Install PHP and Related Modules
Run the commands below to add a third party repository and upgrade to PHP 7.1
sudo apt-get install software-properties-common -y
sudo add-apt-repository ppa:ondrej/php
Then update and upgrade to PHP 7.1
sudo apt update

Run the commands below to install PHP 7.1 and related modules..

sudo apt install php7.1 libapache2-mod-php7.1 php7.1-common
php7.1-mbstring php7.1-xmlrpc php7.1-soap php7.1-apcu php7.1-smbclient
php7.1-ldap php7.1-redis php7.1-gd php7.1-xml php7.1-intl php7.1-json

php7.1-imagick php7.1-mysql php7.1-cli php7.1-mcrypt php7.1-ldap
php7.1-zip php7.1-curl -yAfter install PHP 7.1, run the commands below to open FPM PHP default file.

sudo gedit /etc/php/7.1/apache2/php.ini

Then make the change the following lines below in the file and save.
file_uploads = On
allow_url_fopen = On
memory_limit = 256M (line n0.404)
upload_max_filesize = 100M (line no. 824)
display_errors = Off (line no. 477)
date.timezone = Asia/kolkata (line no. 939)

Step 7 : Create OwnCloud Database
Now that you've install all the required packages, continue to start configuring the server.
To connect to MariaDB server, run the commands below.
sudo mysql -u root -p
Then create a database called owncloud
CREATE DATABASE owncloud;
Create a database user called ownclouduser with new password
CREATE USER 'ownclouduser'@'localhost' IDENTIFIED BY
'password_here';
Then grant the user full access to the database.
GRANT ALL ON owncloud.* TO 'ownclouduser'@'localhost' IDENTIFIED
BY 'password_here' WITH GRANT OPTION;
Now, save your changes and exit.
FLUSH PRIVILEGES;
EXIT;

Step 8: Download Latest OwnCloud Release
Download and and extract OwnCloud files into its root directory:
cd /tmp && wget https://download.owncloud.org/community/owncloud-
10.0.8.zip
unzip owncloud-10.0.8.zip
sudo mv owncloud /var/www/html/owncloud/

Step 9: Then run the commands below to set the correct permissions for OwnCloud to function.
sudo chown -R www-data:www-data /var/www/html/owncloud/
sudo chmod -R 755 /var/www/html/owncloud/
Configure Apache2
Configure Apahce2 site configuration file for OwnCloud. This file will control how users access OwnCloud content. Run the commands below to create a new configuration file called owncloud.conf
sudo gedit /etc/apache2/sites-available/owncloud.conf
Then copy and paste the content below into the file and save it. Replace the highlighted in red lines with your own domain name and directory root location.
<VirtualHost *:80>
ServerAdmin admin@example.com
DocumentRoot /var/www/html/owncloud/
ServerName avoiderrors.com

```
ServerAlias www.avoiderrors.com
Alias /owncloud "/var/www/html/owncloud/"
<Directory /var/www/html/owncloud/>
Options +FollowSymlinks
AllowOverride All
Require all granted
<IfModule mod_dav.c>
Dav off
</IfModule>
SetEnv HOME /var/www/html/owncloud
SetEnv HTTP_HOME /var/www/html/owncloud
</Directory>
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Step 10: Save the file and exit.Enable the OwnCloud and Rewrite Module
After configuring the VirtualHost above, enable it by running the commands below

```
sudo a2ensite owncloud.conf (sudo systemctl reload apache2)
sudo a2enmod rewrite(sudo systemctl restart apache2)
sudo a2enmod headers(sudo systemctl restart apache2)
sudo a2enmod env
sudo a2enmod dir
sudo a2enmod mime
```

Restart Apache2

To load all the settings above, restart Apache2 by running the commands below.

```
sudo systemctl restart apache2.service
```

Access Owncloud from the LAN

To access ownCloud from inside the network, We need to know the IP address of the owncloud server and use it to access it by placing it in the search bar as follows:

Step 10.1: Open the terminal and enter the following command:

```
ifconfig
```

Step 10.2: Locate the IP address of the Ubuntu server, it should look something like: 192.x.x.x or 10.x.x.x3.

Step 10.3: Place your LAN IP in the address bar like
this: http://LANIP/owncloud
Here in Create an admin account:
Username: admin or any other username
Password: any password of your choice

Step 10.4 : Below you have to enter database name, database user name and password which you had given earlier while Creating OwnCloud Database.

Step 10.5 : You should then see OwnCloud setup page. You'll be prompted to create admin account and password. Connect to the database using the information you created and continue.

Conclusion : Thus we have studied and Implemented Storage as a Service using Own Cloud/ AWS S3, Glaciers/ Azure Storage.

EXPERIMENT NO. 4

Aim :To study and Implement Database as a Service on SQL/NOSQL databases like AWS RDS, AZURE SQL/ MongoDB Lab/ Firebase.

Hardware: Device Name :- computer-ThinkCentre
Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3
Memory :- 8.0 Gib
Processor :- 12th Gen Intel® CoreTM i5-12400 × 12
Disk Capacity :- 256.1 GB

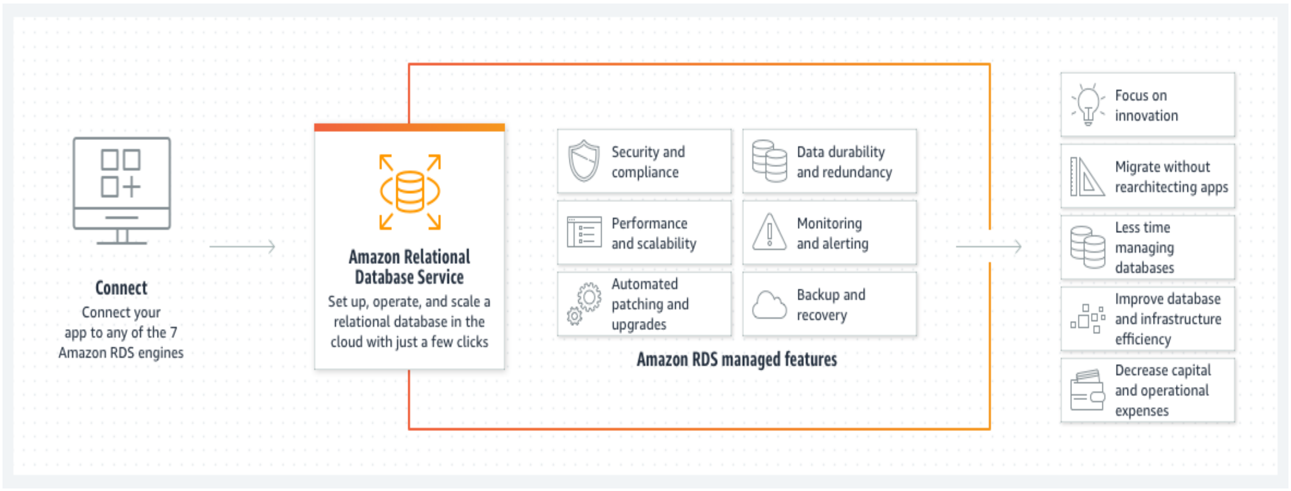
Software : Operating System :- LINUX
Amazon AWS

Theory : The objective TTo know the concept of Database as a Service running on cloud and to demonstrate the CRUD operations on different SQL and NOSQL databases running on cloud like AWS RDS, AZURE SQL/ Mongo Lab/ Firebase.

Amazon Relational Database Service (Amazon RDS) is a collection of managed services that makes it simple to set up, operate, and scale databases in the cloud.

Features:-

- Performance
- Scalability
- Security
- Manageability
- Cost effectiveness
- Developer Productivity
-



Connect
Connect your app to any of the 7 Amazon RDS engines
Amazon Relational Database Service
Set up, operate, and scale a relational database in the cloud with just a few clicks
Security and compliance
Data durability and redundancy
Performance and scalability
Monitoring and alerting
Automated patching and upgrades
Backup and recovery
Amazon RDS managed features
Focus on innovation
Migrate without rearchitecting apps
Less time managing databases
Improve database and infrastructure efficiency
Decrease capital and operational expenses

Algorithm :

Step1 : Login to aws console and search RDS

Step2: Click on to RDS and create database

Step 3: Select standard database

Step 4: Select MySQL and MySQL Community edition

Step 5: In Templates select Free tier

Step 6: Mention database name (default is database1) and username and password

Step 7: Instance is t2.micro

•Rest of things keep default

Step 8: Select Public Acess -Yes

Step 9: Click on to create Database

Step 10: It will take some time

Step 12: Click on to download

- MySQL community download – Microsoft Windows

Step 13: Click on to – No thanks , just download

Step 14: Go to downloads of your machine and install it with default settings

- Check your database is created and status is available

Step 15: Click on to view credential

Step 16: Click on to database

Step 17: Copy Endpoint

Step 18: Go back to workbench

Step 19: Click on to mysql connection

Step 20:
- Paste copied endpoint in Hostname
- Connection Name : databaseShilpa
- Username : admin
- Click on to Test Connection
  Enter admin password.

Step 21.1: Go to vpc security group

Step 21.2: Goto workbench (after giving details click on to Test Connection)

Step 21.3: Click on Ok button

Step 21.4: Go to workbench double click on connection(databaseshilpa)
It will get opened

Step 23: Click on to inbound rules

Step 24: First select Click on to Edit inbound rule add rule select ipv4 --all traffic (add 0.0.0.0./0)
and save Rules (important step to add inbound rule)

Step 22: Goto workbench (after giving details click on to Test Connection)

Click on Ok button
Go to workbench double click on connection(databaseshilpa)
It will get opened

Step 23.1 : Write query and execute
Create database tsec;
Use tsec;
Show tables;

Step 23.2 : Create table for eg:
create table student(roll int, name varchar(10), city varchar(10));

Describe student;

insert into student values(1,'shilpa','thane'); (Perform all CURD) operations)

Step 24:

Now delete the instance (once you have done with it)
Select instance go to action stop instance and then delete instance

Uncheck create final shapshot

Conclusion : Thus, we have successfully studied and implemented Database as a Service on SQL databases like AWS RDS.

EXPERIMENT NO. 5

Aim: To study and Implement Security as a Service on AWS/Azure.

Hardware: Device Name :- computer-ThinkCentre
Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3
Memory :- 8.0 Gib
Processor :- 12th Gen Intel® CoreTM i5-12400 × 12
Disk Capacity :- 256.1 GB

Software : Operating System :- LINUX
Amazon AWS

Theory : The objective to know the concept of Database as a Service running on cloud and to demonstrate the CRUD operations on different SQL and NOSQL databases running on cloud like AWS RDS, AZURE SQL/ Mongo Lab/ Firebase.

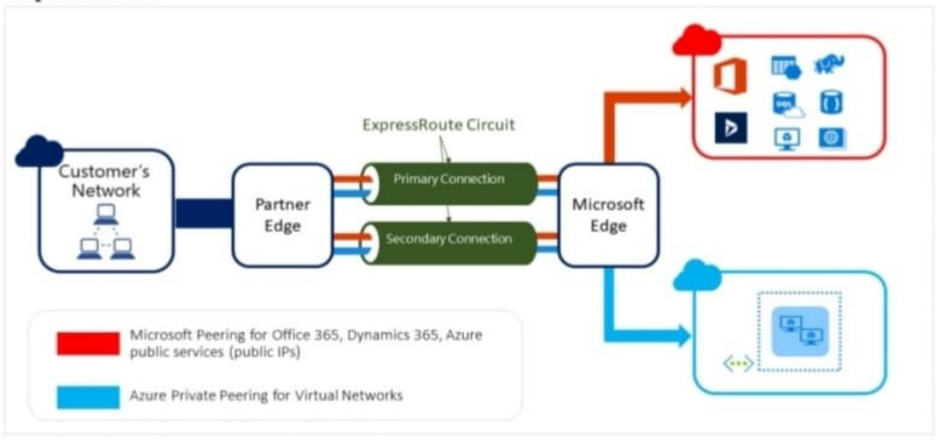
Amazon Relational Database Service (Amazon RDS) is a collection of managed services that makes it simple to set up, operate, and scale databases in the cloud.

Features:-

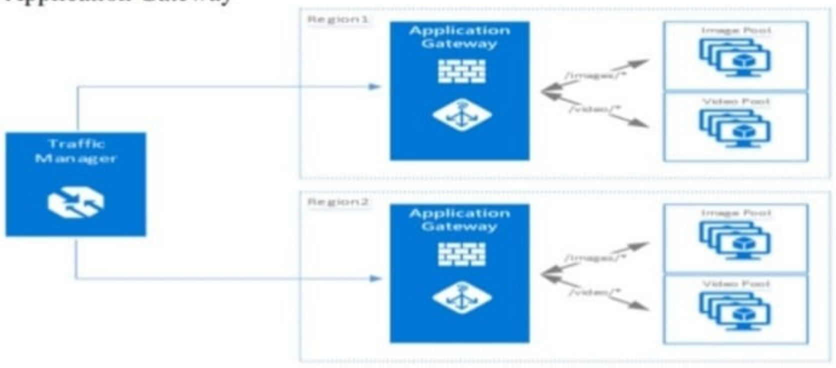
- Performance
- Scalability
- Security
- Manageability
- Cost effectiveness
- Developer Productivity

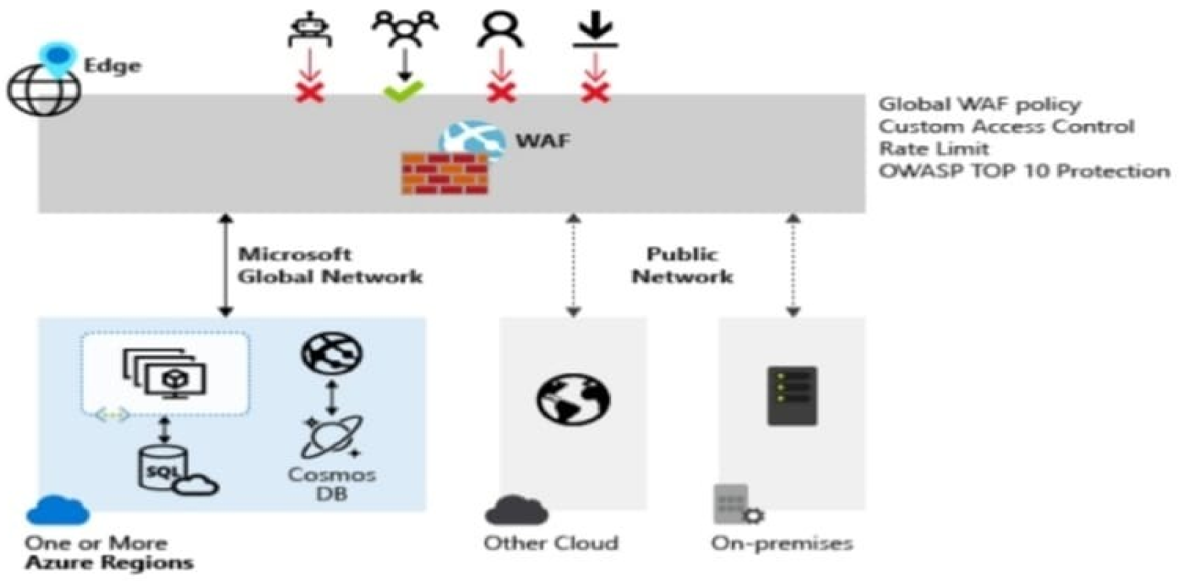
Case study on Implementation Express Route.

Express Route


ExpressRoute Circuit
Customer's Network
Partner Edge
Primary Connection
Secondary Connection
Microsoft Edge
Microsoft Peering for Office 365, Dynamics 365, Azure public services (public IPs)
Azure Private Peering for Virtual Networks

Application Gateway





Conclusion : Thus, in this practical we have studied and Implement Security as a Service on AWS/Azure

Experiment No:06

Aim: To study and Implement Identity and Access Management (IAM) practices on AWS/Azure cloud.

Hardware: Device Name :- computer-ThinkCentre
Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3
Memory :- 8.0 Gib
Processor :- 12th Gen Intel® CoreTM i5-12400 × 12
Disk Capacity :- 256.1 GB

Software : Operating System :- LINUX
Amazon AWS

Theory : The objective is to understand the working of Identity and Access Management IAM in cloud computing and to demonstrate the case study based on Identity and Access Management (IAM) on AWS/Azure cloud platform.

Create a user from AWS Management Console Editor's note: AWS uses the names AWS Management Console and AWS Console interchangeably. To create a user in AWS, we fill out a form and receive an access ID and secret key. At this step, we create a user named cli-user with full access permissions and programmatical access. This user is how we will manage other users later.

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permissions that control which AWS resources users can access. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

Algorithm:

Step 1: Create a user from AWS Management Console Editor's note: AWS uses the names AWS Management Console and AWS Console interchangeably.

Step 2 : In AWS, we fill out a form and receive an access ID and secret key.

Step 3: At this step, we create a user named cli-user with full access permissions and programmatical access. This user is how we will manage other users later.

Step 4: Open the IAM console of the AWS Console in a browser window. 1.Sign in. Sign in as a root user. Provide username and password when prompted.

Step 5: Select the Users menu. Navigate to the Users screen. You'll find it in the IAM dashboard, under the Identity and Access Management (IAM) drop-down menu on the left side of the screen. Click on Users.

Step 6: Add a user. Click on Add User to navigate to a user detail form. Provide all details, such as the username and access type. In this tutorial, we use the name cli-user, and check the Programmatic access box under Access type. This option gives the user access to AWS development tools, such as the command line interface used later in this tutorial. Click on Next: Permissions to continue.

Step 7: Set the user permissions. Click Attach existing policies directly and then filter the policies by keyword: IAM. For this user, select IAMFullAccess from the list of available policies. The IAMFullAccess policy enables this user to create and manage user permissions in AWS. Later in the tutorial, this user will perform AWS IAM operations.

Step 8: Finish the user setup. For this tutorial, we will skip the tags section of user creation and go t o the review page. Check the details of the username, AWS access type and permissions. Then, click Create user.

Step 9: Create the user after verifying the name, access type and permissions are correct.

Step 10: At this point in the tutorial, the user cli-user exists, with the chosen policies applied to the account. AWS provides this user an access key ID and secret access key. Download or copy these keys to a secure place to use later in this tutorial.

Step 11: Set up AWS user credentials in the CLI . Open source AWS CLI tool, available through the cloud provider can be used. AWS CLI enables an admin to use their favourite shell or CLI to interact with AWS services. You can choose any Linux distribution or shell. A Bash shell running on an Ubuntu Linux distribution can be used.

Step 12: Install the command line tool. First, install AWS CLI on your system using the following command in Bash terminal:sudo apt install awscli

Step 13: Once the setup runs, verify the installation by checking the version: aws –version

Step 14: Configure the user with the keys. Run the aws configure command in the shell to quickly set up the access key ID and secret access key obtained from AWS when you created the new user in the IAM console.

Step 15: This step saves your credentials in a local file at path: ~/.aws/credentials and region and output format configs at path: ~/.aws/config file.

Step 16: Now that cli-user with programmatic access is set up, we can use that account to create other users and give them policy-based access through AWS CLI.

Step 17: The next two sections foloow these steps. Create a user and assign permissions .

To create a user using IAM, run the aws iam create-user command in AWS CLI with a username: **aws iam create-user --user-name prateek.**

It creates a new user and shows the user details in the bash console.

Step 18.1 : The command creates a user with the name Prateek and shows details, such as the creation date and user ID. Suppose this user needs to manage EC2 services. To grant this new user EC2 admin rights, start by listing which EC2 policies we can grant.

Step 18.2 : Use the command:

a**ws iam list-policies | grep EC2FullAccess**

Identify the appropriate policy for the user's access level. In this case, it is AmazonEC2FullAccess.

Pass the Amazon Resource Name (ARN) to the following command in **--policy-arn** parameter:

aws iam attach-user-policy --user-name prateek --policy-arn "arn:aws:iam::aws:policy/AmazonEC2FullAccess"

Step 19 : Check user details and list user permissions. Once you create the user and attach the appropriate user policy to them, verify that AWS assigned the appropriate policy by checking the user details. To check the list of IAM users,

run: aws iam list-users

The following command tells AWS to list all attached policies for a user account:

aws iam list-attached-user-policies --user-name prateek

Conclusion: Thus, we have studied Identity and Access Management (IAM) practices on AWS /Azure.

Experiment No:07

Aim: To study and Implement Containerization using Docker.

Hardware: Device Name :- computer-ThinkCentre
Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3
Memory :- 8.0 Gib
Processor :- 12th Gen Intel® CoreTM i5-12400 × 12
Disk Capacity :- 256.1 GB

Software : Operating System :- LINUX

Theory : Docker is the containerization platform that is used to package your application and all its dependencies together in the form of containers to make sure that your application works seamlessly in any environment which can be developed or tested or in production.

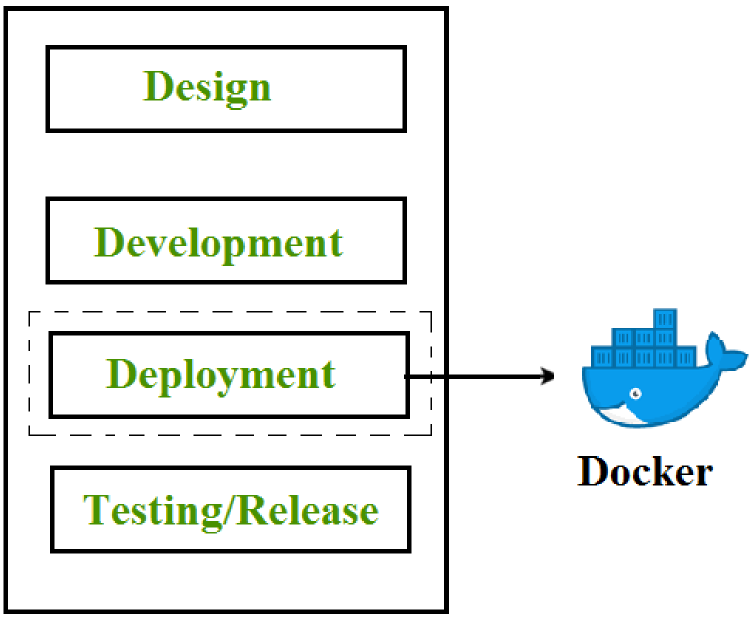
Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.

Containerization is OS-based virtualization that creates multiple virtual units in the userspace, known as Containers.

Containers share the same host kernel but are isolated from each other through private namespaces and resource control mechanisms at the OS level.

Container-based Virtualization provides a different level of abstraction in terms of virtualization and isolation when compared with hypervisors.

Hypervisors use a lot of hardware which results in overhead in terms of virtualizing hardware and virtual device drivers. A full operating system (e.g -Linux, Windows) runs on top of this virtualized hardware in each virtual machine instance.



Algorithm:

Step 1: Create an account on dockerhub

Step 2: Verify email

Step 3: Login to docker hub

Go to https://labs.play-with-docker.com/

Step 4: Login with dockerhub account

Step 5: Click on start

Step 6: Click on add new instance

Step 7: Type command $ docker version

Step 8: Type command $ docker –version

Step 9: Type command $ docker container ls

Step 10: Type command $ docker images ls -a

Step 11: Type command $ docker pull ubuntu

Step 12: Type command $ docker pull mysql

Step 13: Type command $ docker images

Step 14: Type command $ docker container ls

Step 15: Type command $ docker search –filter=starts=100 ubuntu

Step 16: Type command $ docker pull mysql:latest

Step 17: Type command $ docker pull mysql:5.7

Step 18: Type command $ docker run ubuntu

Step 19: Type command $ docker ps -a

Step 20: Type command $ docker run -it ubuntu /bin/bash

Step 21: Type command $ exit

Step 22: Type command $ docker ps -a

Step 23: Type command $ docker run –publish 8080:80 nginx

Step 24: $ docker ps

Step 25: $ docker container stop <container id>

Step 26: $ docker container ls -a

Step 27: $ docker container rm <container id>

Step 28 : $ docker image ls

Step 29 : $ docker image rm <image id>

Step 30: $ docker image ls

Step 31 : $ docker run -d -p 8080:80 nginx

Step 32: $ docker ps

Step 33 : $ docker container top <container id>

Step 34: $ docker container inspect <container id>

Step 35 : $ docker container stats <container id>

Step 36: $ docker container exec -it <container id> “/bin/bash”

Step 37: $ docker run -d –publish 8081:80 –name nginx-container nginx

Step 38: Create your own image

```
$ mkdir myimage
$ cd myimage
```

Step 39: Using editor to create 2 files in created directory

```
  Dockerfile
FROM nginx:latest
WORKDIR /usr/share/nginx/html
COPY index.html index.html
index.html
index.html
```

hello from docker

```
$ docker image build -t nginx-with-html -f Dockerfile .
```

Step 40: $ docker run -d -p 8080:80 nginx-with-html

Conclusion: Thus, we have studied and Implemented Containerization using Docker.

Experiment No:08

Aim: To study and Implement container orchestration using Kubernetes.

Hardware:	Device Name :- computer-ThinkCentre Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3 Memory :- 8.0 Gib Processor :- 12th Gen Intel® CoreTM i5-12400 × 12 Disk Capacity :- 256.1 GB
Software :	Operating System :- LINUX

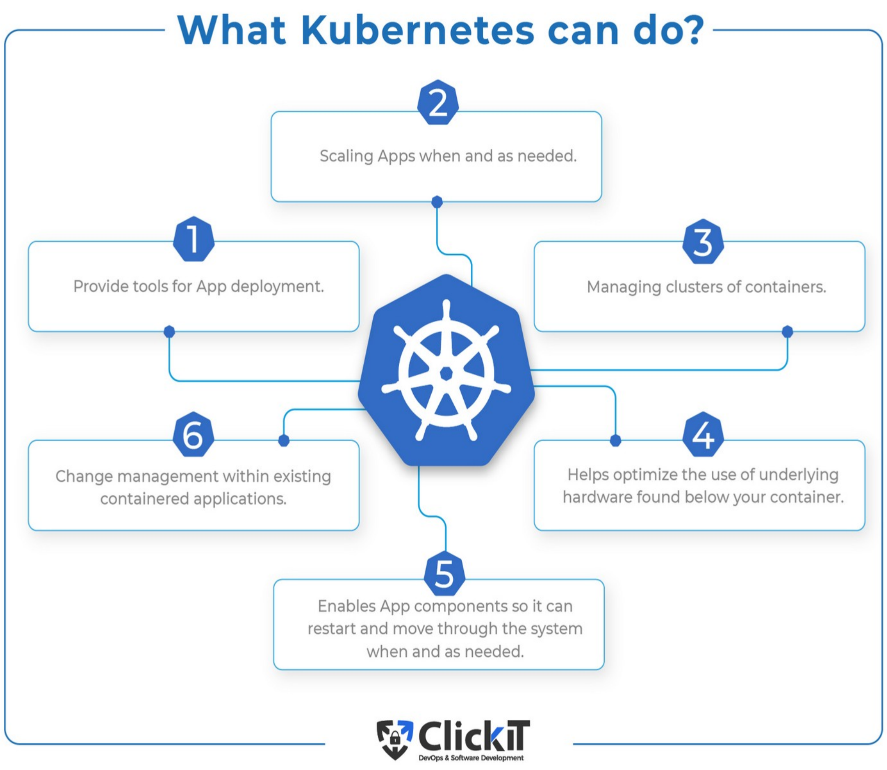
Theory : Container orchestration is the automation of much of the operational effort required to run containerized workloads and services. This includes a wide range of things software teams need to manage a container's lifecycle, including provisioning, deployment, scaling(up and down), networking, load balancing and more.

Container orchestration automates the deployment, management, scaling, and networking of containers. Enterprises that need to deploy and manage hundreds or thousands of Linux Containers and hosts can benefit from container orchestration.

Container orchestration can be used in any environment where you use containers. It can help you to deploy the same application across different environments without needing to redesign it.

Kubernetes is a popular open source platform for container orchestration. It enables developers to easily build containerized applications and services, as well as scale, schedule and monitor those containers.

Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.


What Kubernetes can do?
2 Scaling Apps when and as needed.
1 Provide tools for App deployment.
3 Managing clusters of containers.
6 Change management within existing containered applications.
4 Helps optimize the use of underlying hardware found below your container.
5 Enables App components so it can restart and move through the system when and as needed.
ClickIT DevOps & Software Development

The benefits of Kubernetes are:

- Scalability
- High availability
- Portability
- Security

Conclusion: Thus we have successfully studied and implemented container orchestration using Kubernetes.

Experiment No:09

Aim: To study and implement Hosted Virtualization using VirtualBox& KVM.

Hardware: Device Name :- computer-ThinkCentre
Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3
Memory :- 8.0 Gib
Processor :- 12th Gen Intel® CoreTM i5-12400 × 12
Disk Capacity :- 256.1 GB

Software : Operating System :- LINUX

Theory : Virtualization enables the creation of a virtual machine (VM) that is different from a physical machine but where its physical resources are assigned. It is a great way of reducing the amount of needed physical servers or resources and enables the separation of certain applications on several different machines. This separation helps to troubleshoot specific issues that arise. The kinds of tools one can virtualize are:

- Software.
- Operating system (OS), the most common example of virtualization.
- Storage device.
- Platform.
- Computer network

Kernel-Based Virtual Machine (KVM) is an open source virtualization module directly built into the Linux kernel, enabling Linux OS to function as a Type 1(bare-metal hyperviosr) However, worth noting is that the distinction between Type 1 and Type 2 hypervisors can be blurred with KVM, as it can function as either of the two. Furthermore, it enables the hypervisor to deploy separate virtual machines.

Algorithm:

Hosted Virtualization on Oracle Virtual Box Hypervisor

Step 1: Download Oracle Virtual box from https://www.virtualbox.org/wiki/Downloads

Step 2: Install it in Windows, Once the installation has done open it.

Step 3: Create Virtual Machine by clicking on New

Step 4: Specify RAM Size, HDD Size, and Network Configuration and Finish the wizard

Step 5: To Select the media for installation Click on start and browse for iso file

Step 6: Complete the Installation and use it.

Step 7: To Connect OS to the network change network Mode to Bridge Adapter

The Steps to Create and run Virtual machines in KVM are as follows

Step 7.1 : Check whether CPU has hardware virtualization support. KVM only works if your CPU has hardware virtualization support – either Intel VT-x or AMDV. To determine whether your CPU includes these features, run the following command: #sudo grep -c "svm\|vmx" /proc/cpuinfo

Step 7.2 : Install KVM and supporting packages. Virt-Manager is a graphical application for managing your virtual machines.you can use the kvm command directly, but libvirt and Virt-Manager simplify the process. #sudo apt-get install qemu-kvm libvirt-bin bridge-utils virt-manager

Step 7.3: Create User. Only the root user and users in the libvirtd group have permission to use KVM virtual machines. Run the following command to add your user account to the libvirtd group: #sudo adduser tsec #sudo adduser tsec libvirtd After running this command, log out and log back in as tsec

Step 7.4 : Check whether everything is working correctly. Run following command after logging back in as tsec and you should see an empty list of virtual machines. This indicates that everything is working correctly. #virsh -c qemu:///system list

Step 7.5 : Open Virtual Machine Manager application and Create Virtual Machine #virt-manager

Step 7.6 : Create and run Virtual Machines

Conclusion: Thus, in this experiment we have created and ran virtual machines on hosted hypervisor Oracle VirtualBox and KVM and we can see expected results.

Experiment No:10

Aim: To study and Implement Bare-metal Virtualization using Xen server.

Hardware: Device Name :- computer-ThinkCentre
Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3
Memory :- 8.0 Gib
Processor :- 12th Gen Intel® CoreTM i5-12400 × 12
Disk Capacity :- 256.1 GB

Software : Operating System :- LINUX

Theory : The term bare metal refers to the fact that there is no operating system between the virtualization software and the hardware. The virtualization software resides on the "bare metal" or the hard disk of the hardware, where the operating system is usually installed.

XenServer is a server virtualization platform that offers virtualization performance for virtualized server and client operating systems that nearly matches the performance of a bare metal servers.

Algorithm:

Step 1: Install Xen Server

Step 1.1: Insert Bootable Xen Server CD into CDROM

Step 1.2: press F2 to see the advanced options, Make first boot device as a CDROM from BIOS otherwise press Enter to start installation.

Step 1.3: Select Keyboard Layout

Step 1.4: Press Enter to load Device Drivers

Step 1.5: Press Enter to Accept End user license Agreement

Step 1.6:Select Appropriate disk on which you want to install Xen server

Step 1.7: Select Appropriate installation Media

Step 1.8: Select Additional Packages for installation

Step 1.9: Specify Root password

Step 1.10: Specify IP Address to a Xen Server

Step 1.11: Select Time Zone

3Step 1.12:-:Specify NTP Servers address or use manual time entry then start installation

Step 2: Connect Xen Server to Xen Center Firstly, download the xen center a management utily from xen server by opening the xen severs IP address as a URL on browser. Once Xen center is downloaded, install it.Open Xen center from start menu of Windows. To connect to the XenServer host you configured earlier, click Add a server.

Step 3: Enter the IP address I asked you to take note of earlier. Also enter the password you assigned for your root account. Click Add.

Step 4: One of the first things you want to make sure as you're adding a new XenServer to XenCenter is to save and restore the server connection state on startup. Check the box that will do just that.

Step 5: Once you do that, you will be allowed to configure a master password for all the XenServers you'll be associating with this XenCenter. Click the Require a master password checkbox if that's what you want to do, and then enter your desired master password in the fields provided.

Step 6 : After you click OK, you'll be brought back to the main screen, where you'll see your XenServer already added to XenCenter.

Step 7: Now Before Creating VM we have to Create Storage Repository first which is nothing but shared directory on Xen Center which holds all iso files and which is required to install Operating system on Xen Server its steps are as follows.Right click on Xenserver icon on xen center and click on New SR.

Step 8 : Now Select Windows CIFS library

Step 9 : Specify Storage Repository Name.

Step 10 : Now specify path of shared folder at client side which holds all iso files of os or VM which we are going to install on Xen Server.

Step 11 : At the end Click on finish to create SR.

Step 12 : To check all iso files click on CIFS library and select storage this will show you all iso files.

Step 13 : Installation of UBUNTU Server on Xen Server

Step 13.1 : Right click on Xenserver icon on xen center and select New VM
Step 13.2 : Now select an Operating System to be install here select Ubuntu Lucid Lynx and click on next
Step 13.3 : Now specify Instance Name as ubuntu server
Step 13.4 : Select iso file of Ubuntu server 10.10 to be install
Step 13.5 : Now select hardware for vm i.e. no. of cpu's and memory
Step 13.6 : Select local storage
Step 13.7 : Select network and click on finish.
Step 13.8 : Now go to Console tab to install ubuntu and follow installation Steps.
Step 13.8 : The Xen orchestra provides web based functionality of Xen Center.it provides access to all the VMs with their lifecycle management which are installed over Xen Server.
Step 13.9 : Xen Orchestra (XOA) Portal.

Step 13.10 The Windows XP image running on Xen Orchestra over Google chrome web browser.

Conclusion: Thus, we have understood, studied and Implemented Bare-metal Virtualization using Xen Server.

Experiment No:11

Aim: Design a Web Application hosted on a public cloud platform.

Hardware: Device Name :- computer-ThinkCentre
Hardware Model :- Lenovo ThinkCentre neo 50t Gen 3
Memory :- 8.0 Gib
Processor :- 12th Gen Intel® CoreTM i5-12400 × 12
Disk Capacity :- 256.1 GB

Software : Operating System :- LINUX

Theory : The mini project is based on the concepts of IaaS, PaaS, DbaaS, Storage as a Service Security as a Service, etc.

The students are working on the mini project in groups. A mini project report has to be attached after all the experiments.

Conclusion: Thus we succesfully implemented the mini project by hosting the website over public cloud where the website covers concepts of various Services provided by cloud.