

## **Experiment No.6**

### **Aim:**

Sketch out and analyze architecture of Moodle cloud portal and Moodle cloud site and create different entities dynamically.

### **Theory:**

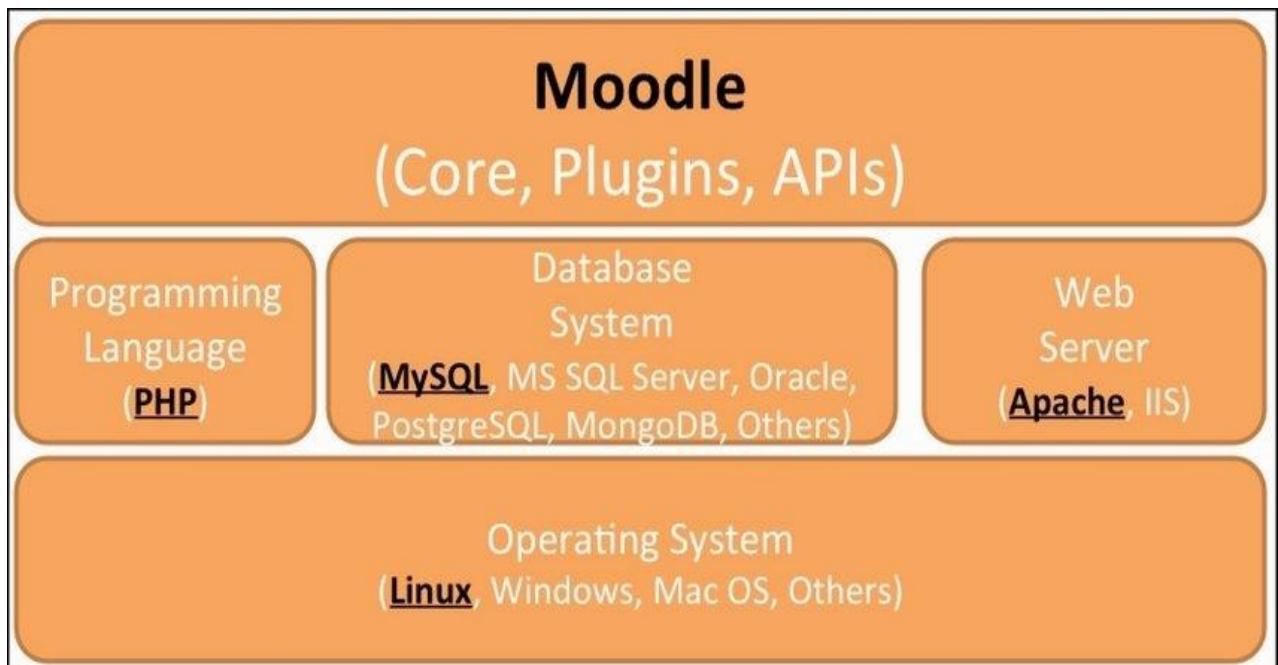
#### 1. Architecture of Moodle cloud/Gnome Cloud :

Moodle is a Learning Management System, Course Management System, or Virtual Learning Environment, depending on which term you prefer. Its goal is to give teachers and students the tools they need to teach and learn. Moodle comes from a background of Social Constructionist pedagogy, however, it can be used to support any style of teaching and learning.

There are other types of software systems that are important for educational institutions, for example ePortfolios, Student Information Systems and Content repositories. Generally, Moodle does not try to re-invent these areas of functionality. Instead, it tries to be the best LMS possible, and then interoperate gracefully with other systems that provide the other areas of functionality. It is, however, perfectly possible

to use Moodle as a stand-alone system, without integrating it with anything else.

Moodle is a web application written in PHP. Moodle is open source. Copyright is owned by individual contributors, not assigned to a single entity, although the company Moodle Pty Ltd in Perth Australia, owned by Moodle's founder Martin Dougiamas, manages the project.



- Like many successful open source systems, Moodle is structured as an application core, surrounded by numerous plugins to provide specific functionality. Moodle is designed to be highly extensible and customizable without modifying the core libraries, as doing so would create problems when upgrading Moodle to a newer version. So when customizing or extending your own Moodle install, always do so through the plugin architecture.
- Plugins in Moodle are of specific types. That is, an authentication plugin and an activity module will communicate with Moodle core using different APIs, tailored to the type of functionality the plugin provides. Functionality common to all plugins (installation, upgrade, permissions, configuration, ...) are, however, handled consistently across all plugin types.

- The standard Moodle distribution includes Moodle core and a number of plugins of each type, so that a new Moodle installation can immediately be used to start teaching and learning. After installation a Moodle site can be adapted for a particular purpose by changing the default configuration option, and by installing add-ons or removing standard plugins. Most add-ons that have been shared publicly are listed in the [Moodle Plugins Directory](#).
- Physically, a Moodle plugin is just a folder of PHP scripts (and CSS, JavaScript, etc. if necessary). Moodle core communicates with the plugin by looking for particular entry points, often defined in the file `lib.php` within the plugin.

## 2. Steps to create Moodle cloud site/Gnome Cloud and create different entities (with screenshots)

- Go to [moodle.com/cloud](https://moodle.com/cloud) where you can get a Moodle Cloud site now. The first thing you want to do is title your LMS. Make sure you're not a robot please, and then just simply fill in the appropriate information. Make sure to read their terms of service, and actually this terms of service is easy and somewhat enjoyable to read. It's not that hard, just a couple paragraphs, I read it on their website, so on here, there's a lot of scrolling, but on their website it's pretty short and sweet. In fact, it doesn't have any confusing terminologies. It's pretty easy to read, so feel free to read that at your convenience. Click continue to go forward.
- Moodle Cloud is actually tied to your smart phone and this helps alleviates spam. It will send you a verification code, so simply go to your phone, see the verification code that you got in SMS from and type it in here below. I've gotten my code and I just type it in here. The next step is to pick a password and make sure that it's strong. I always pick a very strong password so that nobody can hack in. Let's click Save and there we go. That's it. We're all done. You can log into your website.
- Here is what your new Moodle Cloud service looks like. It has a dashboard over here with your admin settings. It has your available

courses, which you can actually take a course on Moodle Cloud, which is pretty cool.

The screenshot shows the MoodleCloud homepage. At the top, there's a navigation bar with the MoodleCloud logo, 'Features', 'Pricing', 'Support', 'Login' (with a dropdown arrow), 'or', and 'Sign up'. Below the header is a large banner with a scenic background of mountains and clouds. The banner contains the text 'Moodle hosting from the people that make Moodle' and a subtext 'Your learning environment with the world's leading open source learning platform Moodle, hosted in the cloud.' A prominent orange button labeled 'Get started for Free!' is visible. To the right of the banner, there's a small cursor icon pointing towards the 'Get started for Free!' button. Below the banner, a section titled 'MoodleCloud is for...' describes the service as suitable for educators, trainers, or anyone needing an online learning environment. A subtext below this states 'Educators, trainers or anyone who needs an online learning environment to teach a class or facilitate learning in any situation.' Further down, a section titled 'Creating your learning environment is easy' includes a testimonial from 'MoodleWorld' with a quote from 'John Doe' about MoodleCloud being the best choice for Moodle hosting. The bottom of the page features a large orange 'Let's get started' button, two smaller buttons for 'Create new account' and 'Log in' (the latter with a cursor icon pointing at it), and a dark footer bar with the 'MoodleWorld' logo.



## Enter your details

This screenshot shows the first step of the account setup process. At the top, there are five tabs labeled STEP 1 through STEP 5. Below them is a form with several input fields:

- First name \*: A text input field containing the letter 'I'.
- Family name \*: An empty text input field.
- Email address \*: An empty text input field.
- Email address confirmation \*: An empty text input field.
- Mobile/Cell number \*: A dropdown menu showing '(201) 555-5555'.
- Your timezone \*: A dropdown menu showing 'Kolkata'.

Below the form, a note states: "We will use this information to verify your account."

At the bottom of the form, there are two more fields: "Organisation name" and "Organisation type". The "Organisation type" field has a dropdown menu with "Select..." option. There are also "Street address" and "City" fields, though they are partially cut off.



## Setup your MoodleCloud site

Here's the exciting bit! Give your site a name and a place to live. Choose wisely - your site name and hosting location cannot be changed later.

This screenshot shows the second step of the setup process. At the top, there are five tabs labeled STEP 1 through STEP 5. Below them is a form with three main sections:

- Site name \***: A text input field containing "my site .moodlecloud.com".
- Choose where your site will be hosted. We recommend choosing the location closest to you and your students. \***: A dropdown menu with "Choose one..." option.
- What best describes how you intend to use MoodleCloud? \***: A dropdown menu with "Choose one..." option.



## Choose a password

You will use this password to login to both your MoodleCloud site and portal.

STEP 1 STEP 2 STEP 3 STEP 4 STEP 5

New password (minimum 5 characters) \*

••••••| I  
Medium

And again, just to make sure \*

Next



MoodleWorld



## Success!

Before you head over to your shiny new Moodle site at [moodleworld.moodlecloud.com](http://moodleworld.moodlecloud.com), we'd like to let you know some important details.



STEP 1 STEP 2 STEP 3 STEP 4 STEP 5

**Portal**

In your [MoodleCloud portal](#) you can check your current packages and payments, billing information and more. To login to the portal, your **username** is your site URL, and your **password** is your Moodle admin password. The portal is accessible from your MoodleCloud site or [moodlecloud.com](http://moodlecloud.com).

**Support**

MoodleCloud hosting is inexpensive because we don't include a lot of support. See our [FAQ](#) for some answers to frequently-asked questions you may have, or [Moodle Docs](#) for deeper information about using Moodle.



MoodleWorld

Mount Orange Admin User

## Site administration

Search

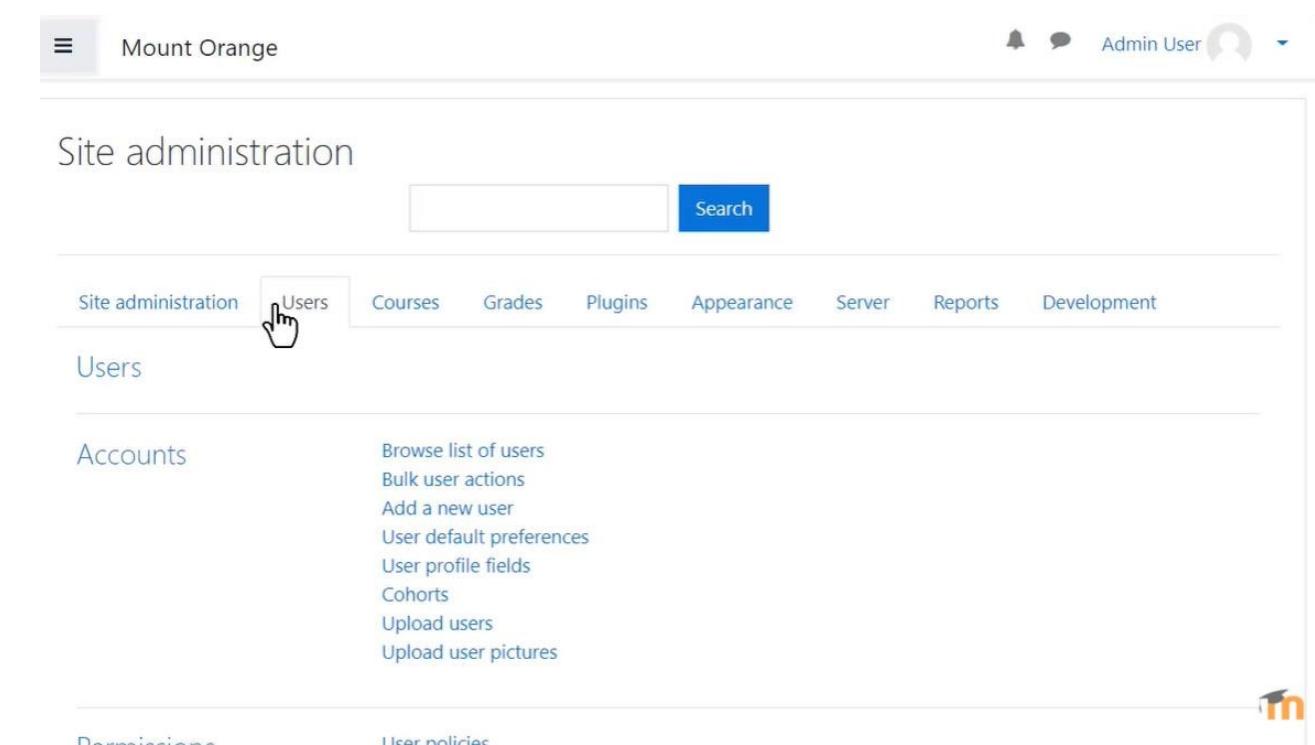
Site administration **Users** Courses Grades Plugins Appearance Server Reports Development

Users

Accounts

- Browse list of users
- Bulk user actions
- Add a new user
- User default preferences
- User profile fields
- Cohorts
- Upload users
- Upload user pictures

Permissions User policies



Mount Orange Admin User

## Upload users

Upload

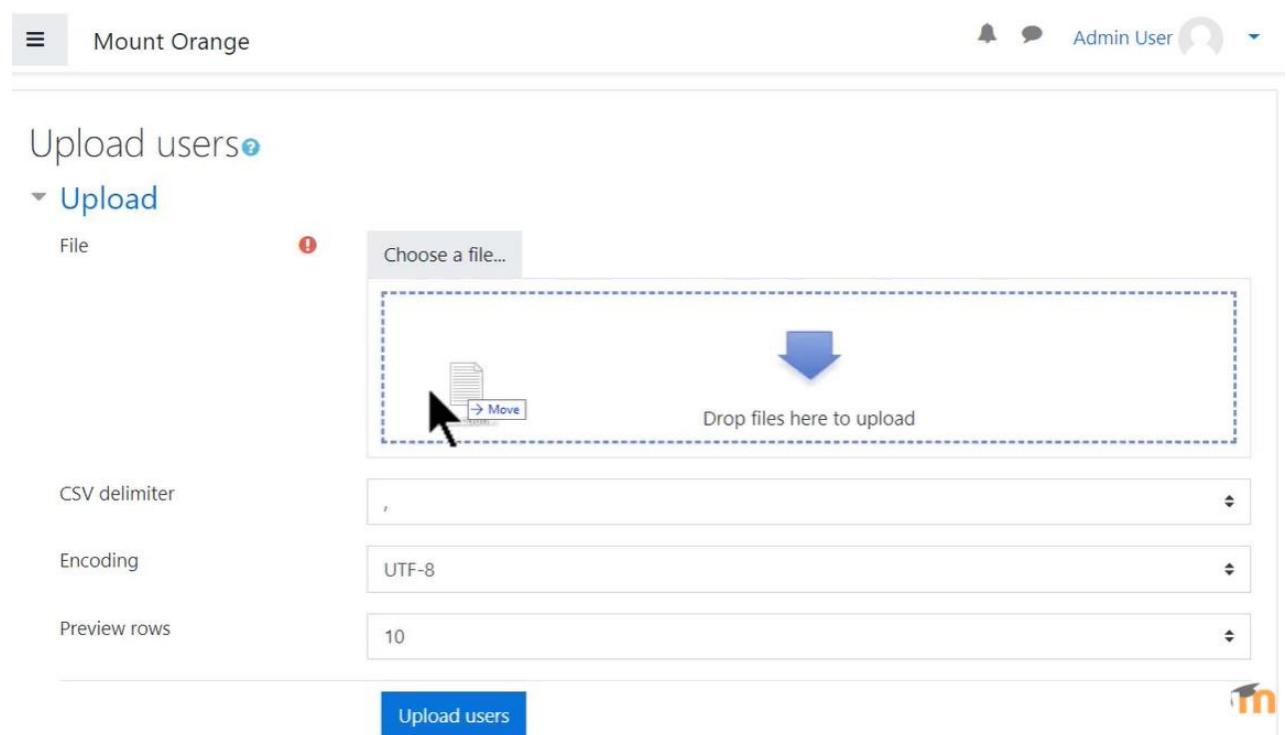
File Choose a file...

CSV delimiter ,

Encoding UTF-8

Preview rows 10

Upload users



Mount Orange

Group ID number

Group description

Enrolment key

Click to enter text  

Hide picture

No

New picture

Choose a file... 

----- Admin Panel -----

Mount Orange

Enrolment key

Click to enter text  

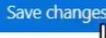
Hide picture

No

New picture

Choose a file... 

teamred.png

Save changes  Cancel 

There are required fields in this form marked .

## **Experiment No. 7**

**Aim** Implement and use sample cloud services with the help of Microsoft Azure  
**Theory:**

Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform. It provides a range of cloud services, including compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications, or run existing applications in the public cloud.

The Azure platform aims to help businesses manage challenges and meet their organizational goals. It offers tools that support all industries -- including e-commerce, finance and a variety of Fortune 500 companies -- and is compatible with open source technologies. This provides users with the flexibility to use their preferred tools and technologies. In addition, Azure offers 4 different forms of cloud computing: infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS) and serverless.

Microsoft charges for Azure on a pay-as-you-go basis, meaning subscribers receive a bill each month that only charges them for the specific resources they have used.

**Points to be included:**

1. Introduction Microsoft Azure

Once customers subscribe to Azure, they have access to all the services included in the Azure portal. Subscribers can use these services to create cloud-based resources, such as virtual machines (VM) and databases

In addition to the services that Microsoft offers through the Azure portal, a number of third-party vendors also make software directly available through Azure. The cost billed for third-party applications varies widely but may involve paying a subscription fee for the application, plus a usage fee for the infrastructure used to host the application.

Microsoft provides five different customer support options for Azure:

- Basic

- Developer
- Standard
- Professional Direct
- Premier

These customer support plans vary in terms of scope and price. Basic support is available to all Azure accounts, but Microsoft charges a fee for the other support offerings. Developer support costs \$29 per month, while Standard support costs \$100 per month and Professional Direct support is \$1000 per month. Microsoft does not disclose the pricing for Premier support.

## 2. Azure SQLdb

Microsoft Azure SQL Database is a managed cloud database provided as part of Microsoft Azure. A cloud database is a database that runs on a cloud computing platform, and access to it is provided as a service. Managed database services take care of scalability, backup, and high availability of the database.

## 3. Steps to create Azure SQL database or Steps to create Azure PostgreSQL database

To create a single database in the Azure portal, this quickstart starts at the Azure SQL page.

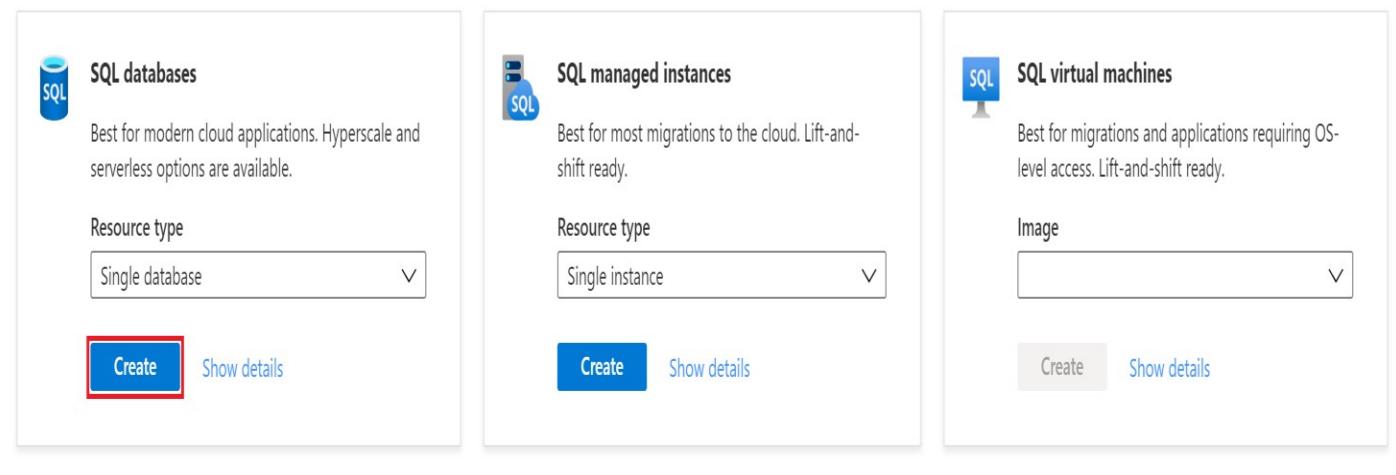
1. Browse to the Select SQL Deployment option page.
2. Under **SQL databases**, leave **Resource type** set to **Single database**, and select **Create**.

## Select SQL deployment option

Microsoft

 Feedback

### How do you plan to use the service?



**SQL databases**  
Best for modern cloud applications. Hyperscale and serverless options are available.

Resource type  
Single database ▾

**Create** Show details

**SQL managed instances**  
Best for most migrations to the cloud. Lift-and-shift ready.

Resource type  
Single instance ▾

**Create** Show details

**SQL virtual machines**  
Best for migrations and applications requiring OS-level access. Lift-and-shift ready.

Image ▾

**Create** Show details

3. On the **Basics** tab of the **Create SQL Database** form, under **Project details**, select the desired **Azure Subscription**.
4. For **Resource group**, select **Create new**, enter *myResourceGroup*, and select **OK**.
5. For **Database name**, enter *mySampleDatabase*.
6. For **Server**, select **Create new**, and fill out the **New server** form with the following values:
  - **Server name:** Enter *mysqlserver*, and add some characters for uniqueness. We can't provide an exact server name to use because server names must be globally unique for all servers in Azure, not just unique within a subscription. So enter something like *mysqlserver12345*, and the portal lets you know if it's available or not.
  - **Server admin login:** Enter *azureuser*.
  - **Password:** Enter a password that meets requirements, and enter it again in the **Confirm password** field.
  - **Location:** Select a location from the dropdown list.

Select **OK**.

7. Leave **Want to use SQL elastic pool** set to **No**.
8. Under **Compute + storage**, select **Configure database**.
9. This quickstart uses a serverless database, so select **Serverless**, and then select **Apply**.

## Configure

X



Looking for basic, standard, premium?

## General Purpose

Scalable compute and storage options

500 - 20,000 IOPS

2-10 ms latency

## Hyperscale

On-demand scalable storage

500 - 204,800 IOPS

1-10 ms latency

## Business Critical

High transaction rate and high resiliency

5,000 - 204,800 IOPS

1-2 ms latency

## Compute tier

## Provisioned

Compute resources are pre-allocated  
Billed per hour based on vCores configured

## Serverless

Compute resources are auto-scaled  
Billed per second based on vCores used

## Compute Hardware

Click "Change configuration" to see details for all hardware generations available including memory optimized and compute optimized options

## Hardware Configuration

## Gen5

up to 40 vCores, up to 120 GB memory  
[Change configuration](#)

## Max vCores



1 vCore

## Min vCores



0.5 vCores

2.02 GB MIN MEMORY 3 GB MAX MEMORY

## Auto-pause delay

The database automatically pauses if it is inactive for the time period specified here, and automatically resumes when database activity recurs. Alternatively, auto-pausing can be disabled.

 Enable auto-pause

Days Hours Minutes

0 1 0

**Apply**

## Cost summary

Gen5 - General Purpose (GP\_S\_Gen5\_1)

Cost per GB

Max storage selected (in GB)

ESTIMATED STORAGE COST / MONTH

COMPUTE COST / VCORE / SECOND<sup>1</sup>

## NOTES

<sup>1</sup> Serverless databases are billed in vCores based on a combination of CPU and memory utilization. Learn more about serverless billing

10. Select **Next: Networking** at the bottom of the page.

Dashboard > Azure SQL > Select SQL deployment option >

## Create SQL Database

Microsoft

provision with smart defaults, or visit each tab to customize. [Learn more](#) ↗

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ  ▼

Resource group \* ⓘ  ▼

[Create new](#)

### Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name \*  ✓

Server \* ⓘ  ▼

[Create new](#)

Want to use SQL elastic pool? \* ⓘ  Yes  No

Compute + storage \* ⓘ

**General Purpose**  
Serverless, Gen5, 1 vCore, 32 GB storage  
[Configure database](#)

**Review + create** **Next : Networking >**

11. On the **Networking** tab, for **Connectivity method**, select **Public endpoint**.
12. For **Firewall rules**, set **Add current client IP address** to **Yes**. Leave **Allow Azure services and resources to access this server** set to **No**.
13. Select **Next: Additional settings** at the bottom of the page.

**Create SQL Database**

Microsoft

Basics Networking Additional settings Tags Review + create

Configure network access and connectivity for your server. The configuration selected below will apply to the selected server 'mysqlserver-12' and all databases it manages. [Learn more](#)

### Network connectivity

Choose an option for configuring connectivity to your server via public endpoint or private endpoint. Choosing no access creates with defaults and you can configure connection method after server creation. [Learn more](#)

Connectivity method \* ⓘ

No access  
 Public endpoint  
 Private endpoint (preview)

### Firewall rules

Setting 'Allow Azure services and resources to access this server' to Yes allows communications from all resources inside the Azure boundary, that may or may not be part of your subscription. [Learn more](#)

Setting 'Add current client IP address' to Yes will add an entry for your client IP address to the server firewall.

Allow Azure services and resources to access this server \*

No Yes

Add current client IP address \*

No Yes

**Review + create** < Previous Next : Additional settings >

14. On the **Additional settings** tab, in the **Data source** section, for **Use existing data**, select **Sample**. This creates an AdventureWorksLT sample database so there's some tables and data to query and experiment with, as opposed to an empty blank database.
15. Optionally, enable Microsoft Defender for SQL.
16. Optionally, set the maintenance window so planned maintenance is performed at the best time for your database.
17. Select **Review + create** at the bottom of the page:

[Home](#) > [servercontoso](#) >

## Create SQL Database

Microsoft

[Basics](#)[Networking](#)[Additional settings](#)[Tags](#)[Review + create](#)

Customize additional configuration parameters including collation & sample data.

### Data source

Start with a blank database, restore from a backup or select sample data to populate your new database.

[Use existing data \\*](#)[None](#)[Backup](#)[Sample](#)

AdventureWorksLT will be created as the sample database.

### Database collation

Database collation defines the rules that sort and compare data, and cannot be changed after database creation. The default database collation is SQL\_Latin1\_General\_CI\_AS. [Learn more](#)

[Collation](#)

SQL\_Latin1\_General\_CI\_AS

### Azure Defender for SQL

Protect your data using Azure Defender for SQL, a unified security package including vulnerability assessment and advanced threat protection for your server. [Learn more](#)

Advanced Data Security costs 15 USD/server/month.

[Enable Azure Defender for SQL \\*](#)[Enable](#)[Not now](#)

### Maintenance window

Select a preferred maintenance window from the drop down. Please note, during a maintenance event, Azure SQL Database are fully available and accessible but some of the maintenance updates require a failover as Azure takes SQL DB instances offline for a short time to apply the maintenance updates. If the database is part of elastic pool, the maintenance configuration of elastic pool will be applied. [Learn more](#)

[Maintenance window](#)

System default

[Review + create](#)< PreviousNext : Tags >

18. On the **Review + create** page, after reviewing, select **Create**.



## Experiment No. 8

**Aim:** Categorize Amazon Web Service (AWS) and implement its various cloud entities using its Cloud Toolbox support.

### Theory:

Best Services offered by AWS are –

1. Amazon Elastic Cloud Compute (EC2)



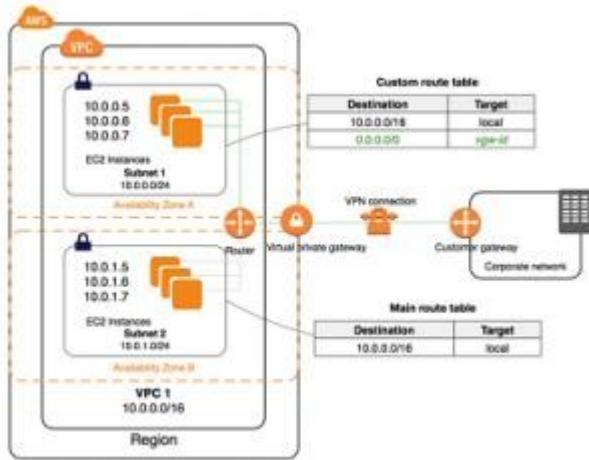
The Amazon EC2 service comes under the compute domain and it provides services that help to compute workloads. Amazon EC2 web interface is used to reduce the expensive physical servers by creating virtual machines. Also, they help in managing different features of the virtual servers such as security, ports, and storage.

### Amazon S3 (Simple Storage Service)



Amazon S3 is categorized under storage domain that provides data storage over the Internet services. Primarily, S3 stores data over the cloud in the form of objects. Amazon S3 stores the data with high security because of its improved infrastructure.

## Amazon Virtual Private Cloud (VPC)



Amazon VPC falls under the Networking domain of AWS which is used to isolate the network infrastructure of user's computer. Every Amazon account holds a unique virtual network that protects the information from being accessed by others.

## Open Source Tools

### Infrastructure Security

#### [cf-signer](#)

Tool for signing and verifying the integrity of CloudFormation templates

#### [domain-protect](#)

scans Amazon Route53 across an AWS Organization for domain records vulnerable to takeover

### Data Protection

#### [Patrolaroid](#)

Patrolaroid snapshots AWS instances and buckets to uncover malware, backdoors, cryptominers, toolkits, and other attacker tomfoolery that you probably don't want in your prod.

### Identity & Access Management

#### [leapp](#)

Leapp is a Cross-Platform Cloud access App, and is designed to manage and secure Cloud Access in multi-account environments.

#### [rpCheckup](#)

An AWS resource policy security checkup tool that identifies public, external account access, intra-org account access, and private resources. It makes it easy to reason about resource visibility across all the accounts in your org

## [\*\*iam-floyd\*\*](#)

AWS | EC2, Systems Manager

AWS IAM policy statement generator with fluent interface.

## [\*\*aws-iamctl\*\*](#)

AWS | IAM

IAMCTL is a tool that you can use to extract the IAM roles and policies from two accounts, compare them, and report out the differences and statistics.

## [\*\*TrailScraper\*\*](#)

A command-line tool to get valuable information out of AWS CloudTrail and a general purpose toolbox for working with IAM policies

## [\*\*SkyWrapper\*\*](#)

CyberArk | IAM

SkyWrapper analyzes behaviors of temporary tokens created in a given AWS account. The tool is aiming to find suspicious creation forms and uses of temporary tokens to detect malicious activity in the account. The tool analyzes the AWS account, and creating an excel sheet includes all the currently living temporary tokens

## [\*\*cloudsplaining\*\*](#)

Salesforce | IAM

An AWS IAM Security Assessment tool that identifies violations of least privilege and generates a risk-prioritized HTML report.

## [\*\*policy\\_sentry\*\*](#)

Salesforce | IAM

IAM least privilege policy generator, auditor, and analysis database.

E

### **Points to be included:**

1. Impact of AWS in Cloud Computing.

### **IT's SECURE:**

AWS has some of the well-recognized certifications like SAS 70 Type II, PCI DSS Level 1, HIPAA, FISMA Moderate and ISO 27001. AWS is a durable and extremely secure technological platform.

Amazon's data centers and services have several layers of physical and operational security to ensure the safety and integrity of your data. AWS conducts regular audits for continued infrastructural security.

### **IT's COST - EFFICIENT**

AWS' Policy of "Pay as you go" pricing, proves to be very cost efficient for many. Thus, you have to make no long term commitments with Amazon. Amazon passes the benefits of cost savings on its

customers in the form of lower prices while they manage and build a global infrastructure at a larger scale.

You can also consume storage and computing power as per your requirement. Without any upfront or minimum expenditure, you can gradually pay as you go. Generally, it is not at all easy to predict the requirements for resources, so you might allocate fewer resources than required and impact the customer satisfaction.

#### ***IT's FLEXIBLE AND OPEN***

AWS as a platform doesn't believe in operating systems and languages. You can select a programming model or development platform that suits your business. Select services (1 or more than 1) and the way you want to use them and how. This flexibility helps you concentrate on innovation rather than infrastructure.

Use programming languages, architectures, operating systems and databases that you are familiar with. This can help you cut some cost, as your IT personnel won't need new skills to pick up. This can help you utilize your overall time to market and help you improve your productivity.

#### ***ELASTIC AND AGILE***

Iterate, Experiment and Innovate quickly with AWS' huge global cloud infrastructure. AWS lets you quickly scale up or down in accordance with your demand. Instead of waiting long months for hardware, using different and new apps can be very useful.

Add or subtract resources in accordance with customer demand and manage costs effectively. You can avoid resource provisioning upfront for projects with short lifetimes and variable consumption rates. You have according to your usage, regardless of whether you require a thousand of virtual servers or only one for 24/7 or only few hours a day.

## 2. Types of AWS services.

### **List of the top AWS services**

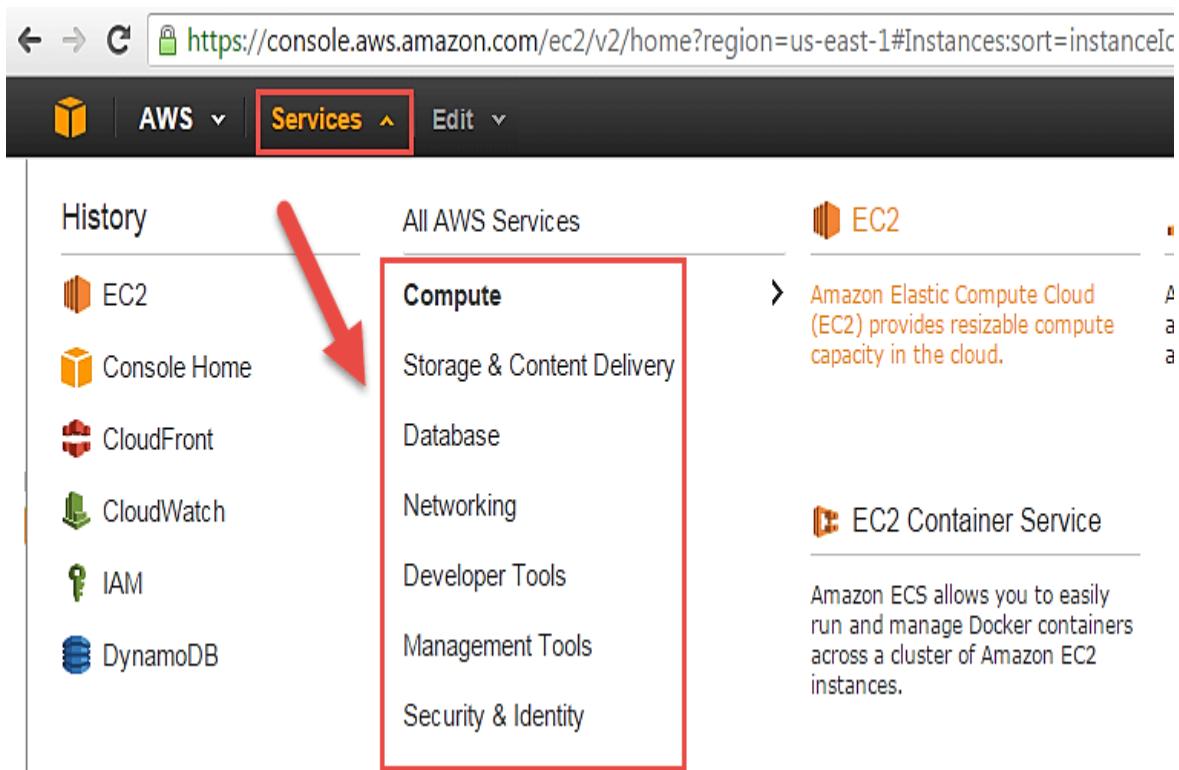
- Amazon EC2 (Elastic Compute Cloud) ...
- Amazon RDS (Relational Database Services) ...
- Bonus Service: Amazon Connect. ...
- Amazon S3 (Simple Storage Service) ...
- Amazon Lambda. ...
- Amazon CloudFront. ...
- Amazon Glacier. ...

- Amazon EBS (Elastic Block Store)

### 3. Steps to create EC2 Instance (Screenshot)

**Step 1)** In this step,

- Login to your AWS account and go to the AWS Services tab at the top left corner.
- Here, you will see all of the AWS Services categorized as per their area viz. Compute, Storage, Database, etc. For creating an EC2 instance, we have to choose Compute & EC2 as in the next step.



- Open all the services and click on EC2 under Compute services. This will launch the dashboard of EC2.

Here is the EC2 dashboard. Here you will get all the information in gist about the AWS EC2 resources running.

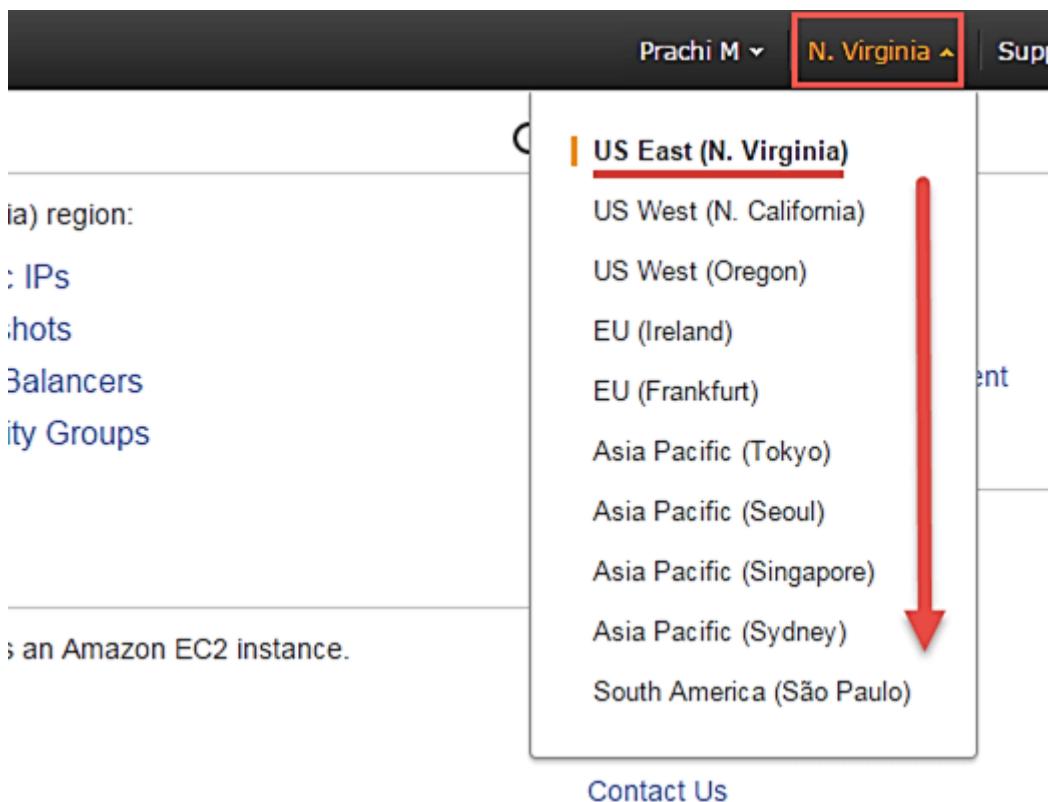
The screenshot shows the AWS EC2 Dashboard. On the left, a sidebar lists navigation options: Events, Tags, Reports, Limits, INSTANCES (with sub-options Instances, Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts), IMAGES (with sub-options AMIs, Bundle Tasks), and ELASTIC BLOCK STORE (with sub-options Volumes, Snapshots). A red arrow points from the 'Events' link in the sidebar to the 'Events' link in the top navigation bar. The main content area is titled 'Resources' and displays a summary of Amazon EC2 resources in the US East (N. Virginia) region:

3 Running Instances	4 Elastic IPs
0 Dedicated Hosts	17 Snapshots
12 Volumes	0 Load Balancers
22 Key Pairs	28 Security Groups
0 Placement Groups	

Below this summary, a message encourages users to try Amazon Simple Queue Service. The 'Create Instance' section contains a large blue 'Launch Instance' button.

**Step 2)** On the top right corner of the EC2 dashboard, choose the AWS Region in which you want to provision the EC2 server.

Here we are selecting N. Virginia. AWS provides 10 Regions all over the globe.



**Step 3)** In this step

- Once your desired Region is selected, come back to the EC2 Dashboard.
- Click on 'Launch Instance' button in the section of Create Instance (as shown below).

The screenshot shows the AWS EC2 Dashboard. The left sidebar has a tree view with 'EC2 Dashboard' selected. Under 'INSTANCES', 'Instances' is selected. Under 'IMAGES', 'AMIs' is selected. Under 'ELASTIC BLOCK STORE', 'Volumes' is selected. The main content area is titled 'Resources' and displays the following statistics:  
3 Running Instances  
0 Dedicated Hosts  
12 Volumes  
22 Key Pairs  
0 Placement Groups

A section titled 'Create Instance' contains the text: 'To start using Amazon EC2 you will want to launch a' followed by a large blue button with white text that says 'Launch Instance'. This button is highlighted with a red rectangular border.

Note: Your instances will launch in the US East (N. Virginia) re

- Instance creation wizard page will open as soon as you click 'Launch Instance'.

#### Choose AMI

**Step 1)** In this step we will do,

1. You will be asked to choose an AMI of your choice. (An AMI is an Amazon Machine Image. It is a template basically of an Operating System platform which you can use as a base to create your instance). Once you launch an EC2 instance from your preferred AMI, the instance will automatically be booted with the desired OS. (We will see more about AMIs in the coming part of the tutorial).
2. Here we are choosing the default Amazon [Linux](#) (64 bit) AMI.

The screenshot shows the AWS EC2 instance creation wizard at Step 1. The top navigation bar includes 'AWS Services' and 'Edit'. On the right, it shows 'Prachi M' (selected), 'N. Virginia', and 'Support'. Below the navigation, a progress bar shows steps 1 through 7. Step 1 is highlighted with a red circle containing the number 1. The main section title is 'Step 1: Choose an Amazon Machine Image (AMI)'. A sub-instruction says: 'An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.' A 'Cancel and Exit' button is on the right. The 'Quick Start' sidebar has tabs for 'My AMIs', 'Amazon Linux' (selected, marked 'Free tier eligible'), and 'Community AMIs'. The 'Amazon Linux' card displays the 'Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-60b6c60a' with a red box around it. It includes a description: 'The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages'. To the right is a 'Select' button with a red circle containing the number 2, and below it is '64-bit'. Navigation arrows at the top right indicate '1 to 22 of 22 AMIs'.

### Choose EC2 Instance Types

**Step 1)** In the next step, you have to choose the type of instance you require based on your business needs.

1. We will choose t2.micro instance type, which is a 1vCPU and 1GB memory server offered by AWS.
2. Click on “Configure Instance Details” for further configurations

AWS Services Edit Prachi M N. Virginia Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

## Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	m4.large	2	8	EBS only	Yes	Moderate

1 2

Cancel Previous Review and Launch Next: Configure Instance Details

- In the next step of the wizard, enter details like no. of instances you want to launch at a time.
- Here we are launching one instance.

### Configure Instance

**Step 1)** No. of instances- you can provision up to 20 instances at a time. Here we are launching one instance.

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Tag Instance   6. Configure Security Group   7. Review

## Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the role to the instance, and more.

Number of instances  Launch into Auto Scaling Group

Purchasing option  Request Spot instances

**Step 2)** Under Purchasing Options, keep the option of ‘Request Spot Instances’ unchecked as of now. (This is done when we wish to launch Spot instances instead of on-demand ones. We will come back to Spot instances in the later part of the tutorial).

1. Choose AMI   2. Choose Instance Type   3. Configure Instance   4. Add Storage   5. Tag Instance   6. Configure Security Group   7. Review

## Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the role to the instance, and more.

Number of instances  Launch into Auto Scaling Group

Purchasing option  Request Spot instances

**Step 3)** Next, we have to configure some basic networking details for our EC2 server.

- You have to decide here, in which VPC (Virtual Private Cloud) you want to launch your instance and under which subnets inside your VPC. It is better to determine and plan this prior to launching the instance. Your AWS architecture set-up should include IP ranges for your subnets etc. pre-planned for better management. (We will see how to create a new VPC in Networking section of the tutorial).
- Subnetting should also be pre-planned. E.g.: If it’s a web server you should place it in the public subnet and if it’s a DB server, you should place it in a private subnet all inside your VPC.

Below,

1. Network section will give a list of VPCs available in our platform.
2. Select an already existing VPC
3. You can also create a new VPC

Here I have selected an already existing VPC where I want to launch my instance.

Screenshot of the AWS EC2 Instance Creation Wizard - Step 3: Configure Instance Details.

The wizard steps are: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance (highlighted), 4. Add Storage, 5. Tag Instance, 6. Configure Security Group, 7. Review.

**Step 3: Configure Instance Details**

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of lower prices, attach data volumes to the instance, and more.

**Number of instances:** 1

**Purchasing option:**  Request Spot instances

**Network** (1)  (3)

**Subnet** (1)  (2)

**Auto-assign Public IP:**

**IAM role:** None

1  
2  
3

**Step 4)** In this step,

- A VPC consists of subnets, which are IP ranges that are separated for restricting access.
- Below,
  1. Under Subnets, you can choose the subnet where you want to place your instance.
  2. I have chosen an already existing public subnet.
  3. You can also create a new subnet in this step.

AWS Services Edit

1. Choose AMI    2. Choose Instance Type    3. Configure Instance    4. Add Storage    5. Tag Instance    6. Configure Security Group    7. Review

## Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the low role to the instance, and more.

**Number of instances** (i)  Launch into Auto Scaling Group (i)

**Purchasing option** (i)  Request Spot instances

**Network** (i)  C Create new VPC

**Subnet** (i)  C Create new subnet

**Auto-assign Public IP** (i) 1 2 3

**IAM role** (i)  C Create new IAM role

**Shutdown behavior** (i)

- Once your instance is launched in a public subnet, AWS will assign a dynamic public IP to it from their pool of IPs.

### Step 5) In this step,

- You can choose if you want AWS to assign it an IP automatically, or you want to do it manually later. You can enable/ disable 'Auto assign Public IP' feature here likewise.
- Here we are going to assign this instance a static IP called as EIP (Elastic IP) later. So we keep this feature disabled as of now.

AWS Services Edit

1. Choose AMI    2. Choose Instance Type    3. Configure Instance    4. Add Storage    5. Tag Instance    6. Configure Security Group    7. Review

## Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the current price, choose a security group, and more.

Number of instances  Launch into Auto Scaling Group [\(i\)](#)

Purchasing option  Request Spot instances

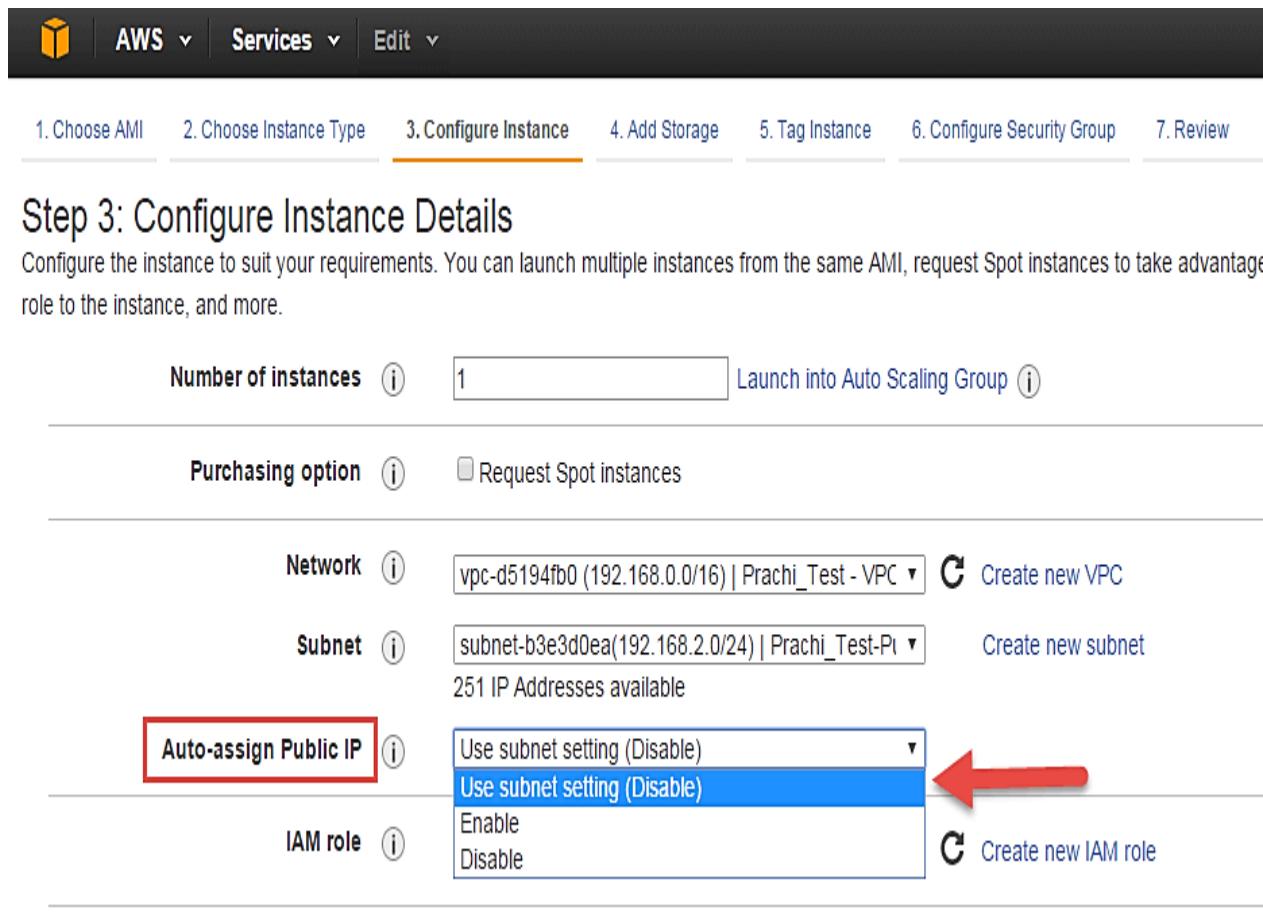
Network [\(i\)](#) vpc-d5194fb0 (192.168.0.0/16) | Prachi\_Test - VPC [\(i\)](#) [C](#) Create new VPC

Subnet [\(i\)](#) subnet-b3e3d0ea(192.168.2.0/24) | Prachi\_Test-Pl [\(i\)](#) [Create new subnet](#)  
251 IP Addresses available

Auto-assign Public IP [\(i\)](#)     

IAM role [\(i\)](#) [\(i\)](#) [C](#) Create new IAM role

Shutdown behavior [\(i\)](#) Stop



**Step 6)** In this step,

- In the following step, keep the option of IAM role ‘None’ as of now. We will visit the topic of IAM role in detail in IAM services.

AWS Services Edit

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

## Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the low role to the instance, and more.

**Number of instances** (i)  Launch into Auto Scaling Group (i)

**Purchasing option** (i)  Request Spot instances

**Network** (i) vpc-d5194fb0 (192.168.0.0/16) | Prachi\_Test - VPC **C** Create new VPC

**Subnet** (i) subnet-b3e3d0ea(192.168.2.0/24) | Prachi\_Test-Pl **Create new subnet**  
251 IP Addresses available

**Auto-assign Public IP** (i) Use subnet setting (Disable)

**IAM role** (i) **None**  **C** Create new IAM role

### Step 7) In this step, you have to do following things

- Shutdown Behavior – when you accidentally shut down your instance, you surely don't want it to be deleted but stopped.
- Here we are defining my shutdown behavior as Stop.

AWS Services Edit

1. Choose AMI    2. Choose Instance Type    3. Configure Instance    4. Add Storage    5. Tag Instance    6. Configure Security Group    7. Review

## Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of t role to the instance, and more.

Number of instances (1) Launch into Auto Scaling Group (i)

Purchasing option (i) Request Spot instances

Network (i) vpc-d5194fb0 (192.168.0.0/16) | Prachi\_Test - VPC (C Create new VPC)

Subnet (i) subnet-b3e3d0ea(192.168.2.0/24) | Prachi\_Test-Pl (Create new subnet)  
251 IP Addresses available

Auto-assign Public IP (i) Use subnet setting (Disable)

IAM role (i) None (C Create new IAM role)

Shutdown behavior (i) Stop (Stop) (Terminate)

Enable termination protection (i)

Monitoring (i) Enable CloudWatch detailed monitoring

**Step 8)** In this step,

- In case, you have accidentally terminated your instance, AWS has a layer of security mechanism. It will not delete your instance if you have enabled accidental termination protection.
- Here we are checking the option for further protecting our instance from accidental termination.

Screenshot of the AWS CloudFormation Step 3: Configure Instance Details page.

The navigation bar at the top shows: AWS Services Edit. Below it is a progress bar with steps: 1. Choose AMI (disabled), 2. Choose Instance Type, 3. Configure Instance (highlighted in orange), 4. Add Storage, 5. Tag Instance, 6. Configure Security Group, 7. Review.

### Step 3: Configure Instance Details

Instance Type: t2.micro (Free tier)

AMI: Amazon Linux 2 (ami-0a9a9a9a9a9a9a9a9)

IAM role: None

Shutdown behavior: Stop

**Enable termination protection**  Protect against accidental termination ←

Monitoring:  Enable CloudWatch detailed monitoring  
Additional charges apply.

**Step 9)** In this step,

- Under Monitoring- you can enable Detailed Monitoring if your instance is a business critical instance. Here we have kept the option unchecked. AWS will always provide Basic monitoring on your instance free of cost. We will visit the topic of monitoring in AWS Cloud Watch part of the tutorial.
- Under Tenancy- select the option if shared tenancy. If your application is a highly secure application, then you should go for dedicated capacity. AWS provides both options.

Screenshot of the AWS CloudFormation Step 3: Configure Instance Details page.

The navigation bar at the top shows: AWS Services Edit. Below it is a progress bar with steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance (highlighted in orange), 4. Add Storage, 5. Tag Instance, 6. Configure Security Group, 7. Review.

### Step 3: Configure Instance Details

Instance Type: t2.micro (Free tier)

AMI: Amazon Linux 2 (ami-0a9a9a9a9a9a9a9a9)

IAM role: None

Shutdown behavior: Stop

**Enable termination protection**  Protect against accidental termination

**Tenancy**  Shared - Run a shared hardware instance ←

Shared - Run a shared hardware instance Selected

Dedicated - Run a Dedicated instance

Dedicated host - Launch this instance on a Dedicated host

Network interfaces:

**Step 10)** In this step,

- Click on ‘Add Storage’ to add data volumes to your instance in next step.

The screenshot shows the AWS CloudFormation console interface for creating a new stack. The top navigation bar includes 'AWS', 'Services', 'Edit', and user information 'Prachi M N. Virginia Support'. Below the navigation is a progress bar with steps 1 through 7: 'Choose AMI', 'Choose Instance Type', 'Configure Instance' (which is selected), 'Add Storage', 'Tag Instance', 'Configure Security Group', and 'Review'. The main content area is titled 'Step 3: Configure Instance Details' with the sub-instruction 'Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.' The configuration fields include:

- Number of instances:** 1
- Purchasing option:** Request Spot instances (unchecked)
- Network:** vpc-d5194fb0 (192.168.0.0/16) | Prachi\_Test - VPC (with 'Create new VPC' button)
- Subnet:** subnet-b3e3d0ea(192.168.2.0/24) | Prachi\_Test-Pl (with 'Create new subnet' button)  
251 IP Addresses available
- Auto-assign Public IP:** Use subnet setting (Disable)
- IAM role:** None (with 'Create new IAM role' button)
- Shutdown behavior:** Stop
- Enable termination protection:** Protect against accidental termination (checked)
- Monitoring:** Enable CloudWatch detailed monitoring (unchecked)  
Additional charges apply.
- Tenancy:** Shared - Run a shared hardware instance

At the bottom right of the configuration area, there are four buttons: 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Add Storage' (with a red arrow pointing to it).

### Add Storage

**Step 1)** In this step we do following things,

- In the Add Storage step, you'll see that the instance has been automatically provisioned a General Purpose SSD root volume of 8GB. ( Maximum volume size we can give to a General Purpose volume is 16GB)
- You can change your volume size, add new volumes, change the volume type, etc.

- AWS provides 3 types of EBS volumes- Magnetic, General Purpose SSD, Provisioned IOPS. You can choose a volume type based on your application's IOPs needs.

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2.](#)

Volume Type (i)	Device (i)	Snapshot (i)	Size (GiB) (i)	Volume Type (i)	IOPS (i)	Delete on Termination (i)	Encrypted (i)
Root	/dev/xvda	snap-a17f1036	8	General Purpose SSD (GP2)	24 / 3000	<input checked="" type="checkbox"/>	Not Encrypted
<input type="button" value="Add New Volume"/>							

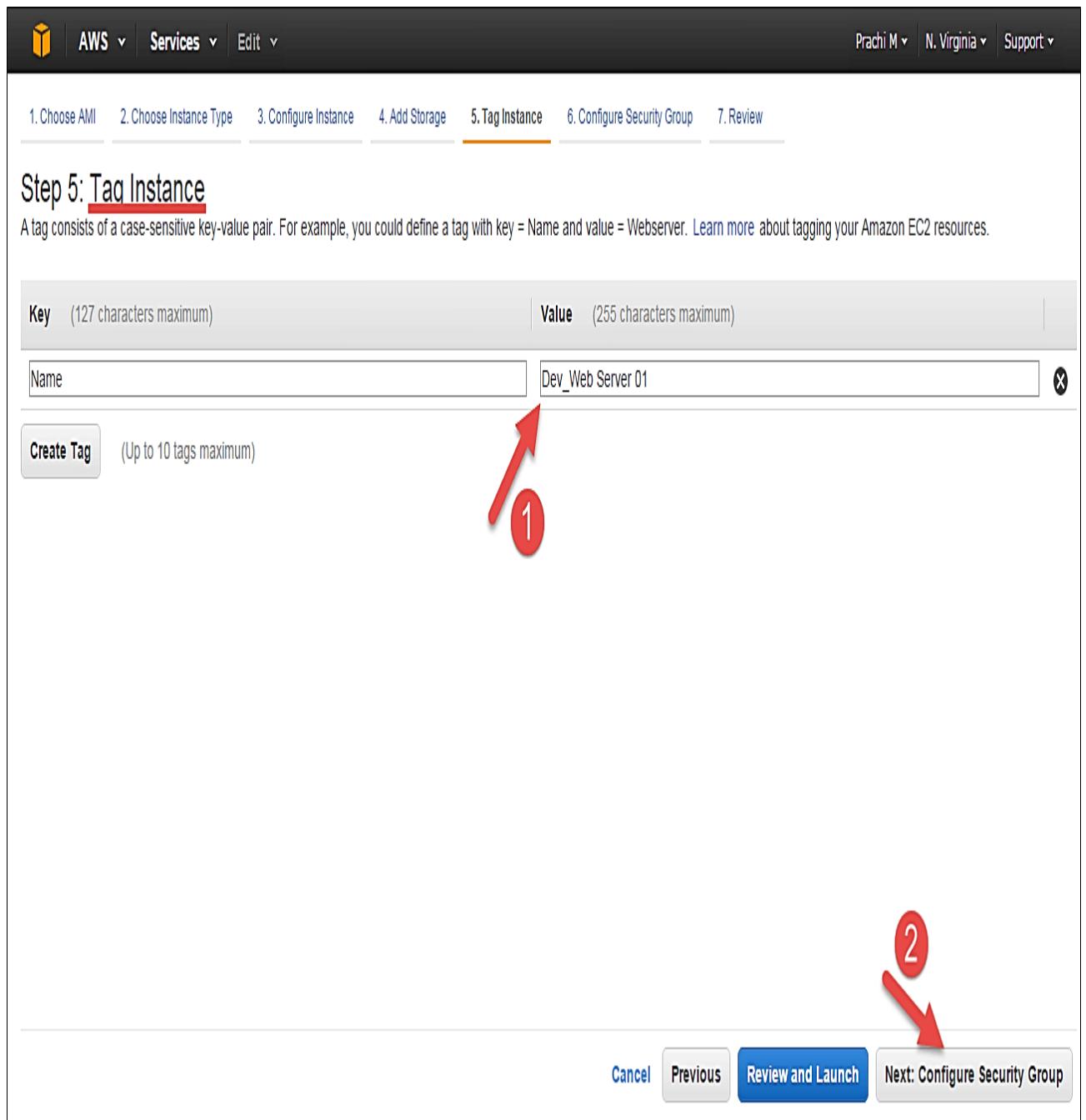
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more about free usage tier eligibility and usage restrictions.](#)

## Tag Instance

### Step 1) In this step

- you can tag your instance with a key-value pair. This gives visibility to the AWS account administrator when there are lot number of instances.
- The instances should be tagged based on their department, environment like Dev/SIT/Prod. Etc. this gives a clear view of the costing on the instances under one common tag.

1. Here we have tagged the instance as a **Dev\_Web server 01**
2. Go to configure Security Groups later



## Configure Security Groups

**Step 1)** In this next step of configuring Security Groups, you can restrict traffic on your instance ports. This is an added firewall mechanism provided by AWS apart from your instance's OS firewall.

You can define open ports and IPs.

- Since our server is a webserver=, we will do following things
  1. Creating a new Security Group
  2. Naming our SG for easier reference
  3. Defining protocols which we want enabled on my instance
  4. Assigning IPs which are allowed to access our instance on the said protocols

5. Once, the firewall rules are set- Review and launch

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below.

Learn more about Amazon EC2 security groups.

Assign a security group:

- Create a new security group 1
- Select an existing security group

Security group name: Web Server SG 2

Description: launch-wizard-7 created 2016-02-03T19:49:12.288+05:30

Type	Protocol	Port Range	Source
SSH	TCP	22	My IP 52.1.77.244/32
HTTP	TCP	80	Anywhere 0.0.0.0/0
HTTPS	TCP	443	Anywhere 0.0.0.0/0

Add Rule 3

Cancel Previous Review and Launch 4 5

## **Experiment No. 9**

**Aim:** Develop any one application in AWS Cloud9(PaaS) using Java or Python/PHP.

### **Theory:**

AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal. Cloud9 comes prepackaged with essential tools for popular programming languages, including JavaScript, Python, PHP, and more, so you don't need to install files or configure your development machine to start new projects. Since your Cloud9 IDE is cloud-based, you can work on your projects from your office, home, or anywhere using an internet-connected machine. Cloud9 also provides a seamless experience for developing serverless applications enabling you to easily define resources, debug, and switch between local and remote execution of serverless applications. With Cloud9, you can quickly share your development environment with your team, enabling you to pair program and track each other's inputs in real time.

**Points to be included:**

### **1. What is AWS Cloud9.**

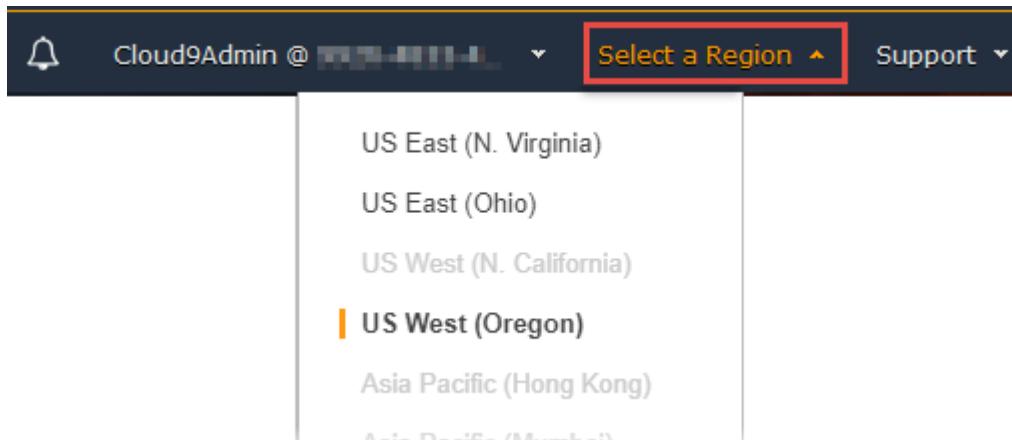
AWS Cloud9 is a cloud-based **integrated development environment (IDE)** that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal.

### **2. All steps to create AWS Cloud9 Environment**

#### **Create an EC2 Environment with the console**

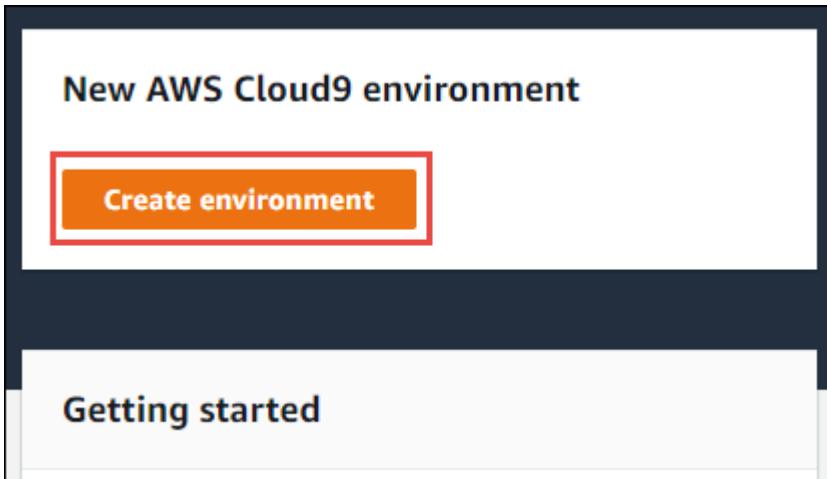
##### **1. Sign in to the AWS Cloud9 console:**

After you sign in to the AWS Cloud9 console, in the top navigation bar, choose an AWS Region to create the environment in. For a list of available AWS Regions, see [AWS Cloud9](#) in the *AWS General Reference*.

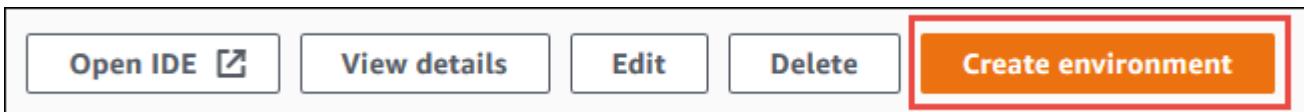


2. Choose the large **Create environment** button in one of the locations shown.

If you have no AWS Cloud9 environments yet, the button is shown on a welcome page.



If you already have AWS Cloud9 environments, the button is shown as follows.



1. On the **Name environment** page, for **Name**, enter a name for your environment. For this tutorial, use `my-demo-environment`.
2. For **Description**, enter something about your environment. For this tutorial, use `This environment is for the AWS Cloud9 tutorial.`
3. Choose **Next step**.
4. On the **Configure settings** page, for **Environment type**, choose one of the following options to create an EC2 backed environment:
  - **Create a new EC2 instance for environment (direct access)** – Launches an Amazon EC2 instance that AWS Cloud9 can connect to directly over SSH.
  - **Create a new no-ingress EC2 instance for environment (access via Systems Manager)** – Launches an Amazon EC2 instance that doesn't require any open inbound ports. AWS Cloud9 connects to the instance through [AWS Systems Manager](#).
- If you select the **access via Systems Manager** option, a service role and an IAM instance profile are automatically created to allow Systems Manager to interact with the EC2 instance on your behalf. You can view the names of both in the **Service role and instance profile for Systems Manager access** section further down the interface. For more information, see [Accessing no-ingress EC2 instances with AWS Systems Manager](#).

For **Instance type**, leave the default choice. This choice has relatively low RAM and vCPUs, which is sufficient for this tutorial.

For **Platform**, choose the type of Amazon EC2 instance that you want: **Amazon Linux 2**, **Amazon Linux**, or **Ubuntu**. AWS Cloud9 creates the instance and then connects the environment to it.

Choose a value for **Cost-saving setting**. When all web browser instances that are connected to the IDE for the environment are closed, AWS Cloud9 waits the amount of time specified and then shuts down the Amazon EC2 instance for the environment.

#### Expand **Network settings (advanced)**.

AWS Cloud9 uses Amazon Virtual Private Cloud (Amazon VPC) to communicate with the newly created Amazon EC2 instance. For this tutorial, we recommend that you don't change the preselected default settings. With the default settings, AWS Cloud9 attempts to automatically use the default VPC with its single subnet in the same AWS account and AWS Region as the new environment.

Add up to 50 tags by supplying a **Key** and a **Value** for each tag. The tags are attached to the AWS Cloud9 environment as resource tags, and are propagated to the following underlying resources: the AWS Cloud Formation stack, the Amazon EC2 instance, and Amazon EC2 security groups. You can find information about tags in Control Access Using AWS Resource Tags in the *IAM User Guide*. Also see the advanced information about tags.

1. Choose **Next step**.
2. On the **Review** page, choose **Create environment**. Wait while AWS Cloud9 creates your environment. This can take several minutes.

After AWS Cloud9 creates your environment, it displays the AWS Cloud9 IDE for the environment.

## 3. Steps to write a program using AWS Cloud9

### Step 1

#### **Create an AWS account**

Create an AWS account or sign in to your existing account.

### Step 2

#### **Set up AWS Cloud9**

Choose the Cloud9 usage pattern that applies to you and set up Cloud9.

#### **Create an AWS Cloud9 environment**

Create your first Cloud9 environment so you can start coding.

#### **Complete a basic tutorial**

Start coding and explore AWS Cloud9.

# Experiment 10

**Aim:** Configure Google App Engine to deploy Python Program application.)

## **Theory:**

Google App Engine is a cloud computing platform as a service for developing and hosting web applications in Google-managed data centers. Applications are sandboxed and run across multiple servers.

## **Points to be included:**

### **1. What is Platform as a Service?**

Platform as a service (PaaS) is a **cloud computing model where a third-party provider delivers hardware and software tools to users over the internet**. Usually, these tools are needed for application development. A PaaS provider hosts the hardware and software on its own infrastructure.

### **2. Benefits and drawbacks of PAAS.**

#### **Advantages of PaaS**

- **Cost Effective:** No need to purchase hardware or pay expenses during downtime
- **Time Savings:** No need to spend time setting up/maintaining the core stack
- **Speed to Market:** Speed up the creation of apps
- **Future-Proof:** Access to state-of-the-art data center, hardware and operating systems
- **Increase Security:** PaaS providers invest heavily in security technology and expertise
- **Dynamically Scale:** Rapidly add capacity in peak times and scale down as needed
- **Custom Solutions:** Operational tools in place so developers can create custom software
- **Flexibility:** Allows employees to log in and work on applications from anywhere

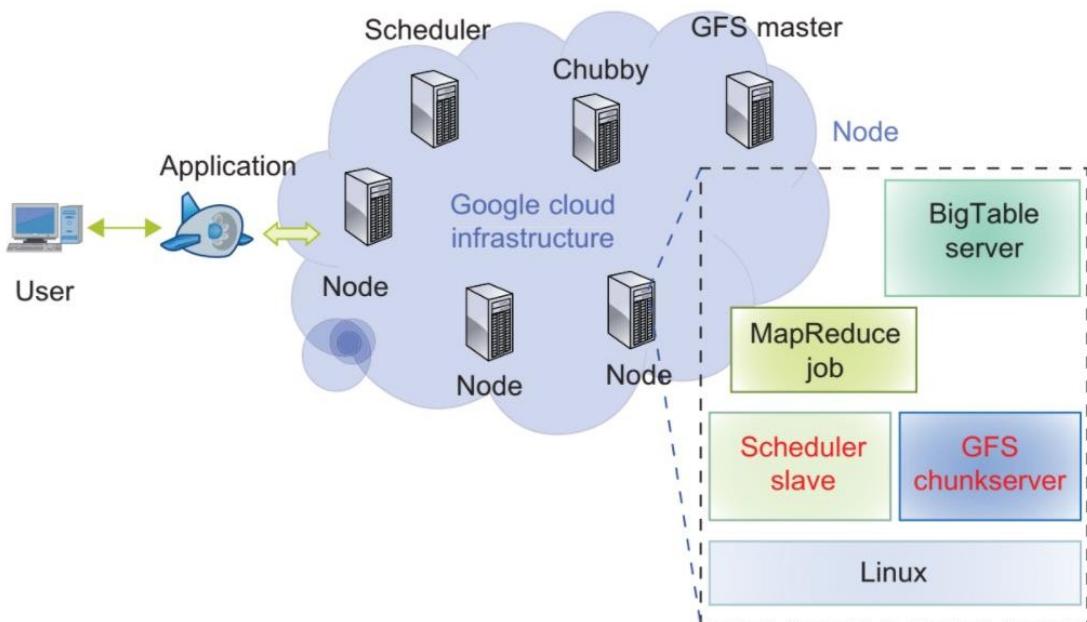
#### **Disadvantages of PaaS**

- Security. All the data of applications are stored inside the provider's cloud database. ...
- Control. Users lack some controls over a PaaS solution. ...
- Reliability. PaaS solutions often face reliability concerns. ...
- Compatibility. Not all the components are cloud enabled. ...
- Integration. ...
- Locked-in Features.

### 3. Features and block diagram of Google App Engine (GAE)

#### Major Features of Google App Engine in Cloud Computing

- Collection of Development Languages and Tools. ...
- Fully Managed. ...
- Pay-as-you-Go. ...
- Effective Diagnostic Services. ...
- Traffic Splitting. ...
- All Time Availability. ...
- Ensure Faster Time to Market. ...
- Easy to Use Platform.



block diagram of Google App Engine (GAE)

### 4. How to develop an application using GAE

In the Google Cloud Console, on the project selector page, select or create a Google Cloud project.

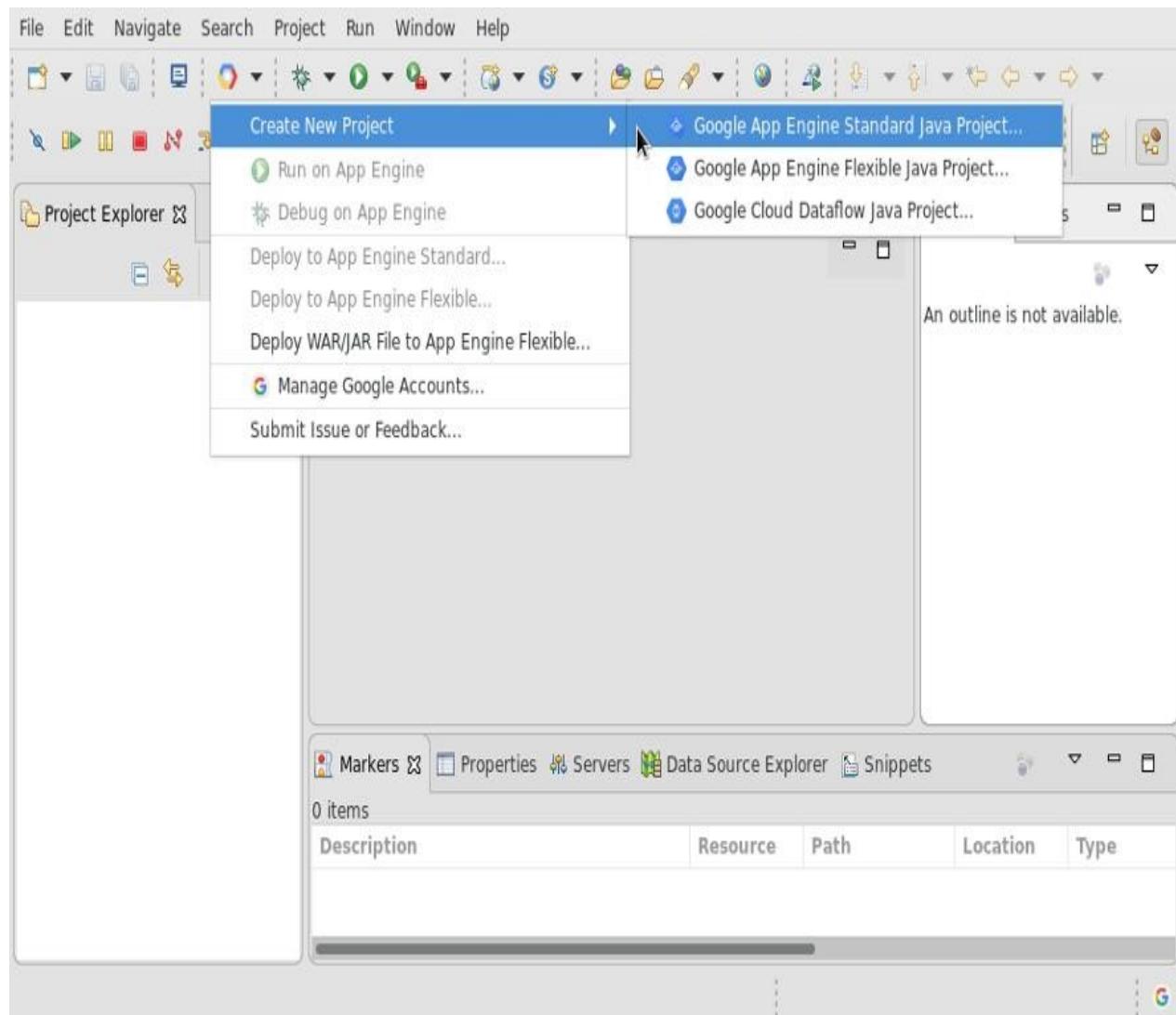
1. Make sure that billing is enabled for your Cloud project.
2. Enable the Cloud Build API.
3. Install and initialize the Cloud SDK.

4. Create an App Engine application for your Cloud project in the Google Cloud Console.
5. Select a region where you want your app's computing resources located.
1. Ensure you have the latest version of Cloud Tools for Eclipse.

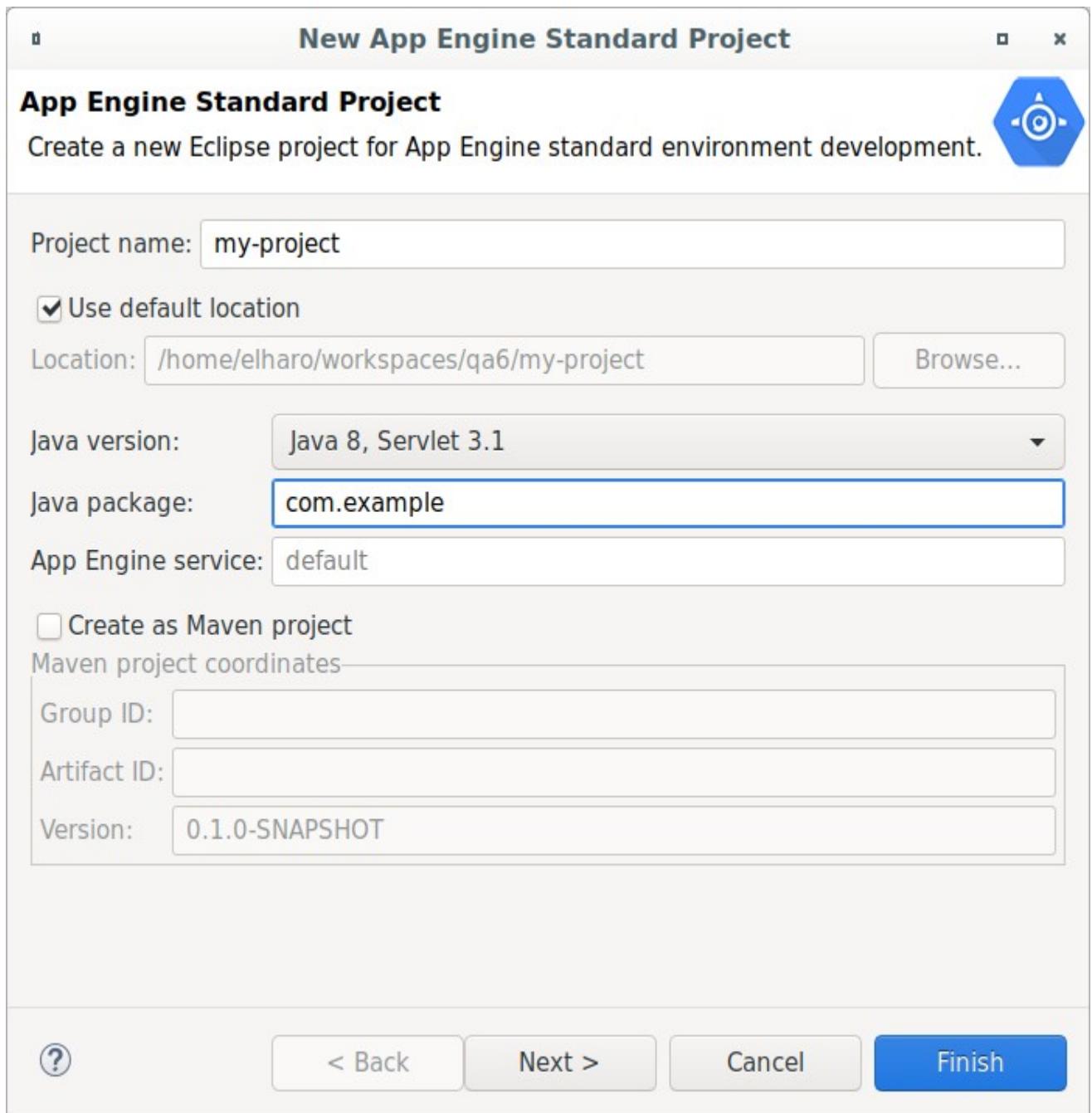
Creating a new Java project in the App Engine standard environment

To create a new project for the App Engine standard environment in Eclipse:

1. Click the **Google Cloud Platform** toolbar button 
2. Select **Create New Project > Google App Engine Standard Java Project**.



3. Enter a **Project name** and (optionally) a Java package.



4. To create a Maven-based App Engine project, select **Create as Maven Project** and enter a Maven **Group ID** and **Artifact ID** of your choosing to set the coordinates for this project. The **Group ID** is often the same as the package name, but does not have to be. The **Artifact ID** is often the same as or similar to the project name, but does not have to be.
5. Click **Next**.
6. Select any libraries you need in the project.
7. Click **Finish**.

The wizard generates a native Eclipse project, with a simple servlet, that you can run and deploy from the IDE.

## **Experiment No. 11**

**Aim:** Design and analyze architecture of Aneka identify different entities to understand the structure of it.

### **Theory:**

The system includes four key components, including **Aneka Master**, **Aneka Worker**, **Aneka Management Console**, and **Aneka Client Libraries**. The Aneka Master and Aneka Worker are both Aneka Containers which represents the basic deployment unit of Aneka based Clouds.

#### **Points to be included:**

##### **1. Introduction of Aneka**

Aneka is the product of Manjarasoft.

Aneka is used for developing, deploying and managing cloud applications.

Aneka can be integrated with existing cloud technologies.

Aneka includes extensible set of APIs associated with programming models like MapReduce.

These APIs supports different types of cloud models like private, public, hybrid cloud.

##### **2. Architecture of Aneka with detail point**

###### **Architecture of Aneka with detail points**

Aneka is a platform and a framework for developing distributed applications on the Cloud. It harnesses the spare CPU cycles of a heterogeneous network of desktop PCs and servers or datacenters on demand. Aneka provides developers with a rich set of APIs for transparently exploiting such resources and expressing the business logic of applications by using the preferred programming abstractions. System administrators can leverage on a collection of tools to monitor and control the deployed infrastructure. This can be a public cloud available to anyone through the Internet, or a private cloud constituted by a set of nodes with restricted access.

The Aneka based computing cloud is a collection of physical and virtualized resources connected through a network, which are either the Internet or a private intranet. Each of these resources hosts an instance of the Aneka Container representing the runtime environment where the distributed applications are executed. The container provides the basic management features of the single node and leverages all the other operations on the services that it is hosting.

The services are broken up into fabric, foundation, and execution services. Fabric services directly interact with the node through the Platform Abstraction Layer (PAL) and perform hardware profiling and dynamic resource provisioning. Foundation services identify the core system of the Aneka middleware, providing a set of basic features to enable Aneka containers to perform specialized and specific sets of tasks. Execution services directly deal with the scheduling and execution of applications in the Cloud.

One of the key features of Aneka is the ability of providing different ways for expressing distributed applications by offering different programming models; execution services are mostly concerned with providing the middleware with an implementation for these models. Additional services such as persistence and security are transversal to the entire stack of services that are hosted by the Container. At the application level, a set of different components and tools are provided to: 1) simplify the development of applications (SDK); 2) porting existing applications to the Cloud; and 3) monitoring and managing the Aneka Cloud.

A common deployment of Aneka is presented at the side. An Aneka based Cloud is constituted by a set of interconnected resources that are dynamically modified according to the user needs by using resource virtualization or by harnessing the spare CPU cycles of desktop machines. If the deployment identifies a private Cloud all the resources are in house, for example within the enterprise. This deployment is extended by adding publicly available resources on demand or by interacting with other Aneka public clouds providing computing resources connected over the Internet.

