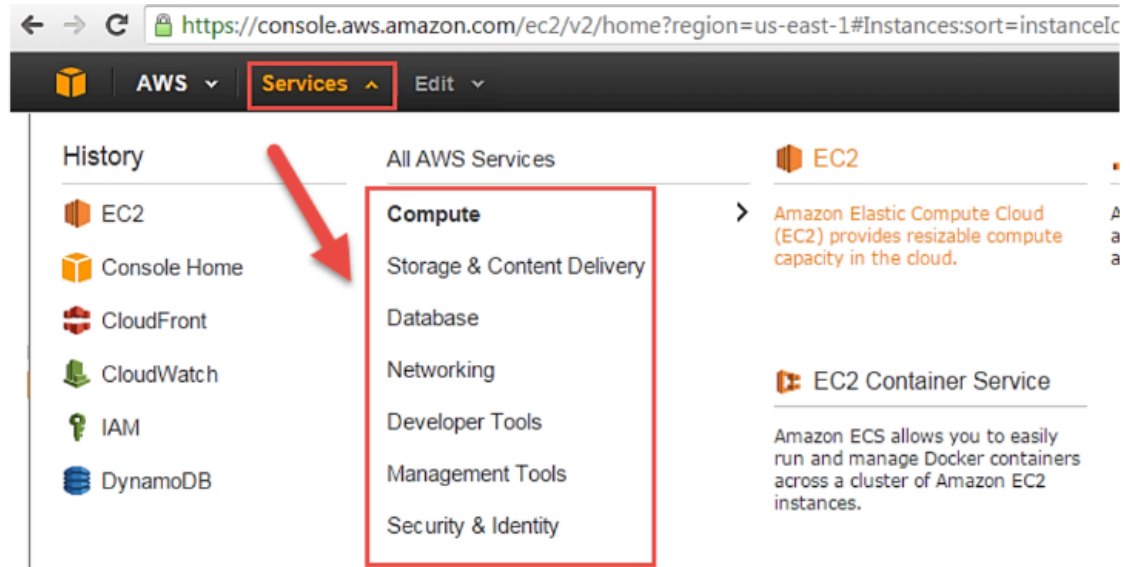


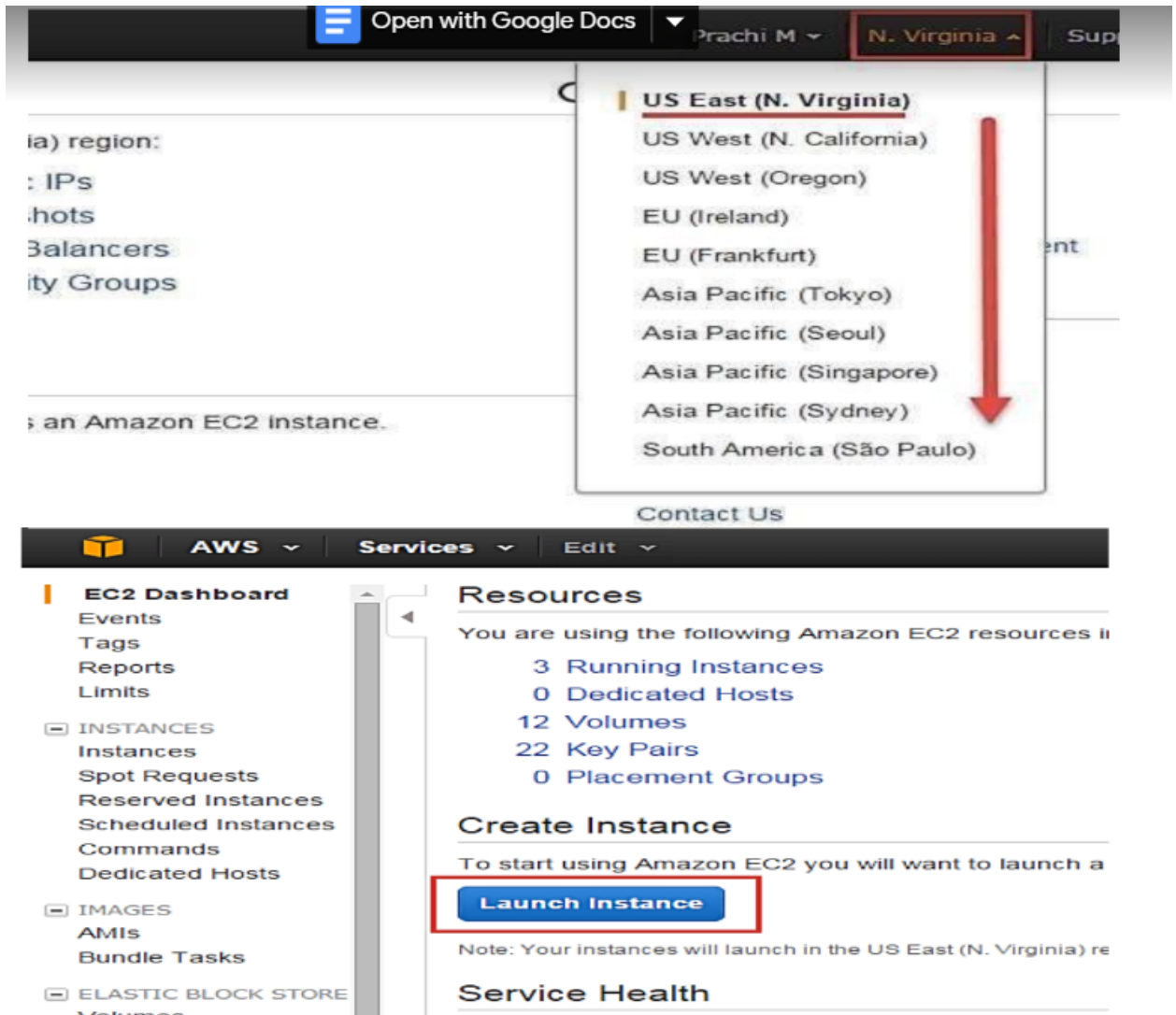
Cloud computing Practical

2. Creating Amazon EC2 instance

- Login to your AWS account and go to the AWS Services tab at the top left corner.
- For creating an EC2 instance, we have to choose Computeà EC2 as in the next step.



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



- Instance creation wizard page will open as soon as you click 'LaunchInstance'. Choose AMI
- 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.
- 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

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

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 on-demand capacity, and more.

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

Purchasing option ⓘ ☐ Request Spot Instances

Network ⓘ **Subnet** ⓘ

Auto-assign Public IP ⓘ

IAM role ⓘ None [Create new IAM role](#)

Subnet dropdown options:

- vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC
- Launch into EC2-Classical
- vpc-621a5e07 (172.20.0.0/16) | POC_vpc
- vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC**
- vpc-8452bce0 (172.20.0.0/16) | POC_vpc
- vpc-823e39e7 (172.22.0.0/16) | TVPC
- vpc-4c51bf28 (10.0.0.0/16) | POC_vpc3

[Create new VPC](#) [Create new subnet](#)

- 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.
- 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.
-

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 instance, and more.

Number of instances Launch into Auto Scaling Group

Purchasing option ☐ Request Spot Instances

Network Create new VPC

Subnet Create new subnet
251 IP Addresses available

Auto-assign Public IP Use subnet setting (Disable)

IAM role Enable Disable Create new IAM role

Shutdown behavior

Step 3: Configure Instance Details

IAM role Create new IAM role

Shutdown behavior

Enable termination protection ☒ Protect against accidental termination

Monitoring ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy Shared - Run a shared hardware instance
Shared - Run a shared hardware instance
Dedicated - Run a Dedicated instance
Dedicated host - Launch this instance on a Dedicated host

Network Interfaces | 117

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 lower pricing, assign an access management role to the instance, and more.

Number of instances 1 [Launch into Auto Scaling Group](#)

Purchasing option 1 ☐ Request Spot instances

Network 1 [Create new VPC](#)

Subnet 1 [Create new subnet](#)
251 IP Addresses available

Auto-assign Public IP 1

IAM role 1 [Create new IAM role](#)

Shutdown behavior 1

Enable termination protection 1 ☒ Protect against accidental termination

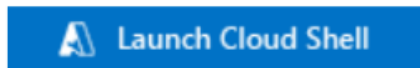
Monitoring 1 ☐ Enable CloudWatch detailed monitoring
Additional charges apply

Tenancy 1
Shared instances run on shared hardware. Dedicated instances run on dedicated hardware.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

1. Creating static web app in Azure

- https://azure.microsoft.com/en-us/free/?ref=microsoft.com&utm_source=microsoft.com&utm_medium=docs&utm_campaign=visualstudio
- Go to <https://shell.azure.com>, or select the Launch Cloud Shell button to open Cloud Shell in your browser.



Select the **Cloud Shell** button on the menu bar at the upper right in the [Azure portal](#).



To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl+Shift+V** on Windows and Linux or by selecting **Cmd+Shift+V** on macOS.
4. Select **Enter** to run the code.

Download the sample

In the Cloud Shell, create a quickstart directory and then change to it.

```
mkdir quickstart
```

```
cd $HOME/quickstart
```

-
- Next, run the following command to clone the sample app repository to your quickstart directory.
git clone <https://github.com/Azure-Samples/html-docs-hello-world.git>

Create a web app

Change to the directory that contains the sample code and run the az webapp up command. In the following example, replace <app_name> with a unique app name. Static content is indicated by the --html flag.

Bash

```
cd html-docs-hello-world
```

```
az webapp up --location westeurope --name <app_name> --html
```

The az webapp up command does the following actions:

- Create a default resource group.
- Create a default app service plan.
- Create an app with the specified name.
- [Zip deploy](#) files from the current working directory to the web app.

This command may take a few minutes to run. While running, it displays information similar to the following example

-

Output

```
{
  "app_url": "https://<app_name>.azurewebsites.net",
  "location": "westeurope",
  "name": "<app_name>",
  "os": "Windows",
  "resourcegroup": "appsvc_rg_Windows_westeurope",
  "serverfarm": "appsvc_asp_Windows_westeurope",
  "sku": "FREE",
  "src_path": "/home/<username>/quickstart/html-docs-hello-world ",
  <JSON data removed for brevity. >
}
```

Make a note of the resourceGroup value. You need it for the [clean up resources](#) section.

Browse to the app

In a browser, go to the app URL: `http://<app_name>.azurewebsites.net`.

- The page is running as an Azure App Service web app.
- Update and redeploy the app
In the Cloud Shell, type `nano index.html` to open the nano text editor. In the `<h1>` heading tag, change "Azure App Service - Sample Static HTML Site" to "Azure App Service", as shown below.
- Save your changes and exit nano. Use the command `^O` to save and `^X` to exit.
You'll now redeploy the app with the same `az webapp up` command.
- Bash
- `az webapp up --location westeurope --name <app_name> --html`
Once deployment has completed, switch back to the browser window that opened in the `Browse to the app` step, and refresh the page.
- Clean up resources
In the preceding steps, you created Azure resources in a resource group. If you don't expect to need these resources in the future, delete the resource group by running the following command in the Cloud Shell. Remember

that the resource group name was automatically generated for you in the create a web app step.

Bash

- `az group delete --name appsvc_rg_Windows_westeurope`

3. Google App engine

1> <https://cloud.google.com/sdk/docs/install>

2> `gcloud components install app-engine-java` or `gcloud components install app-engine-python`

3> Create a new project or use an existing one in the Google Cloud Console:
<https://console.cloud.google.com/project>. (note the project id)

4> `gcloud config set project YOUR_PROJECT_ID`

5> Visit the App Engine API page in the Google Cloud Console:
<https://console.cloud.google.com/apis/library/appengine.googleapis.com>.

Enable the API for your project.

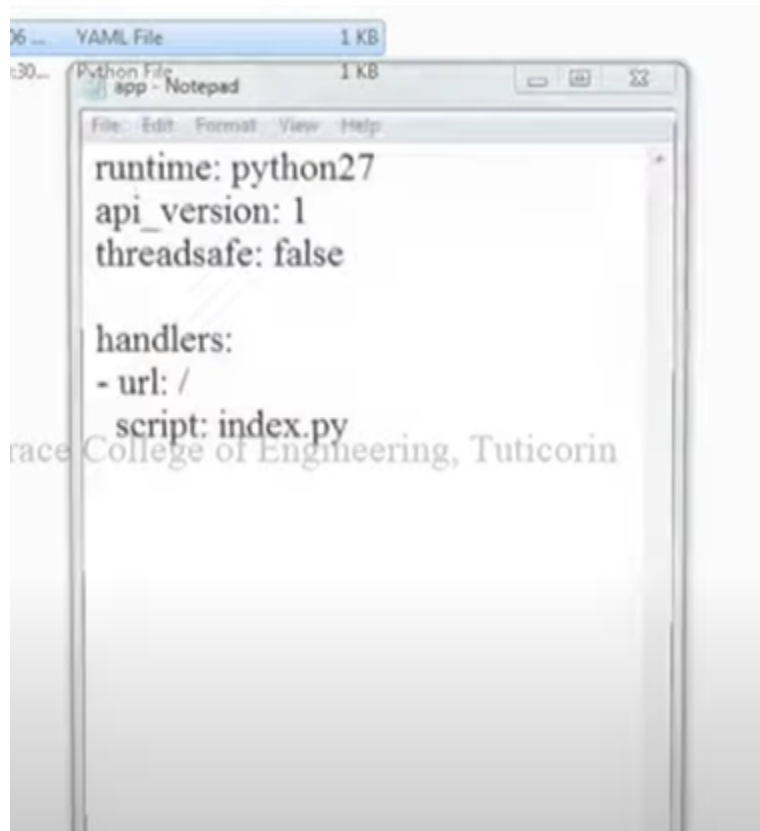
6> Create an `app.yaml` file in the root directory of your app.

Specify the necessary configurations in the `app.yaml` file, such as the runtime, environment variables, scaling settings, etc.

Refer to the App Engine documentation for your specific language and runtime for details on the `app.yaml` configuration:

<https://cloud.google.com/appengine/docs>.

Example yaml file:



And index.py :

```
print("Hello World")
```

Or download from this

git clone <https://github.com/GoogleCloudPlatform/python-docs-sample>

```
cd python-docs-samples/appengine/standard_python37/hello_world
```

```
7> gcloud app deploy app.yaml
```

or

```
google-cloud-sdk\bin\dev_appserver.py "<project file location>"
```

Or through the console>

Go to google cloud console

1. CREATE A NEW PROJECT
2. ACTIVATE GOOGLE CLOUD SHELL
3. ENABLE APP ENGINE ADMIN API
4. CREATE APP ENGINE
5. DEPLOY APPLICATION
6. DISABLE APPLICATION

```
gcloud projects list
```

```
gcloud config set project newapplication-272410
```

```
git clone https://github.com/GoogleCloudPlatform/python-docs-sample
```

```
cd python-docs-samples/appengine/standard_python37/hello_world
```

1 gcloud config set project <project id >

2 gcloud app create

Select location

3 git clone <https://github.com/SumitB007/gcloudpython>

4 cd gcloudpython

5 gcloud app deploy app.yaml

3. Virtualization using KVM

1. sudo apt update

2. sudo virt-manager // is a desktop user interface for managing virtual machines through libvirt

3. If not installed then run this - sudo apt install virt-manager

4. `sudo apt install cpu-checker` // libvirt is an open-source API, daemon and management tool for managing platform virtualization. It can be used to manage KVM,

5. `sudo kvm-ok`

6. `sudo apt install qemu qemu-kvm libvirt-bin bridge-utils virt-manager`

7. `sudo service libvirtd start`

8. `sudo update-rc.d libvirtd enable`

9. `service libvirtd status`

10. `Sudo virt-manager`

New VM

Create a new virtual machine

Step 2 of 5

Locate your install media

☐ Use CDROM or DVD

☒ Use ISO image:

/home/student/Desktop/window 7 s-2 pr

Browse...

Choose an operating system type and version

OS type: Generic

Version: Generic

Cancel

Back

Forward

New VM

Create a new virtual machine

Step 1 of 5

Enter your virtual machine details

Name: ap

Connection: localhost (QEMU)

Warning: KVM is not available. This may mean the KVM package is not installed, or the KVM kernel modules are not loaded. Your virtual machines may perform poorly.

Choose how you would like to install the operating system

☒ Local install media (ISO image or CDROM)

☐ Network Install (HTTP, FTP, or NFS)

☐ Network Boot (PXE)

☐ Import existing disk image

Cancel

Back

Forward

New VM

Create a new virtual machine

Step 3 of 5

Choose Memory and CPU settings

Memory (RAM): 1024 MB

Up to 3957 MB available on the host

CPU(s): 1

Up to 4 available

Cancel

Back

Forward

New VM

Create a new virtual machine

Step 4 of 5

☒ Enable storage for this virtual machine

☒ Create a disk image on the computer's hard drive

8.0 GB

☒ Allocate entire disk now

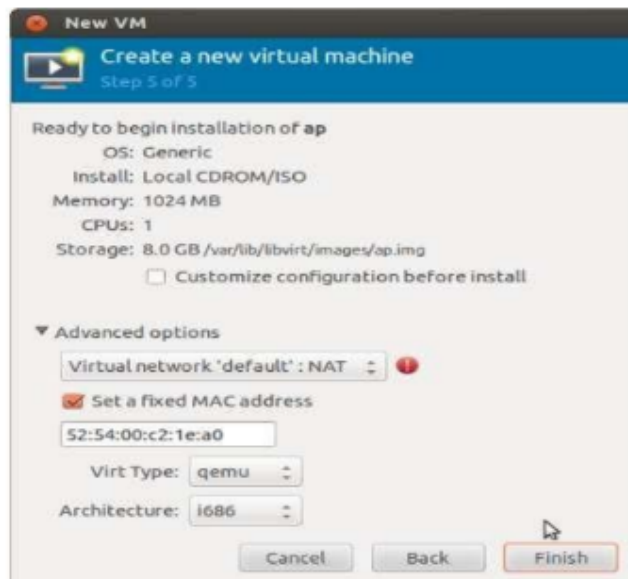
☐ Select managed or other existing storage

Browse...

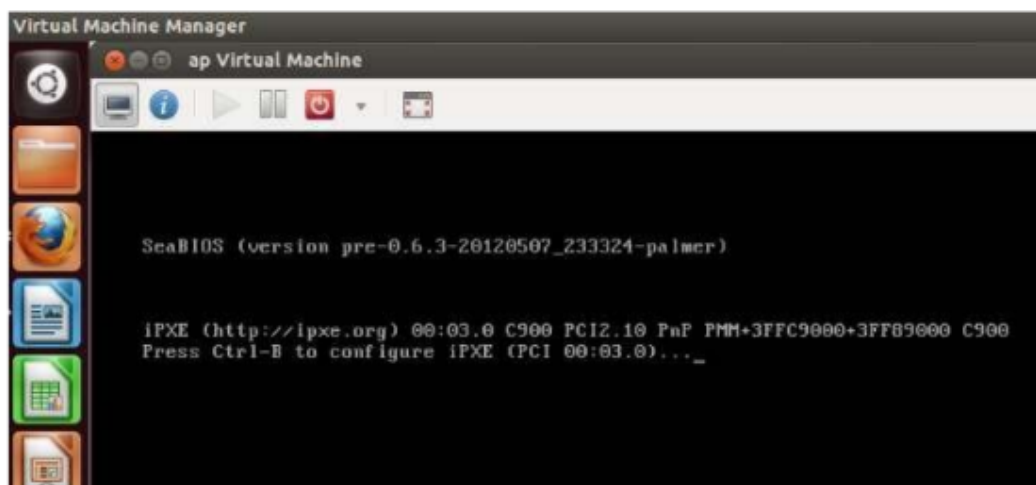
Cancel

Back

Forward



Step 7 : Install windows operating system on virtual machine



Extra commands:

11. `sudo vi /etc/netplan/50-cloud-init.yaml`
12. `sudo netplan apply` //a utility for easily configuring networking on a linux system.
13. `sudo netplan --debug apply`
14. `sudo networkctl status -a` // Creating a bridge for virtual machine reCreating a bridge for virtual machines
15. `Ifconfig` ///displays the current configuration for a network interface when no optional parameters are supplied

16. sudo virt-manager

4. Creating application on Salesforce using apex

1. Create new account:

<https://developer.salesforce.com/signup>

2. After signup, login using following URI

<https://login.salesforce.com/>

3. Go to Developer Console for writing Program from setup

Then file new class

Create class with anyname example stud

Then write

```
public static void main(){  
system.debug('Hello World');  
}
```

Debug and run

5. custom application salesforce

Go to object manager

Create

Custom object

Library management system

Give name example student

Plural name students

Description issues book

Save and new

Book

Books

add new book

Save

**Create fields in book object book)name is already present add
Author_name, publisher**

Then go to setup setup

Search tabs

Go to tabs

New custom object for book and student

Select book select tab style

Next next save

Same for student

Go to setup again

Search app

Go to app app manager

New lightning app

Name description image if you want

**Next next next add your tabs student and book you can add extra inbuilt
tabs also then next**

Add system administrator and then save and finish

**Go to home app launcher search for your application name
And you will see your tabs there add entries to your tab**