Question 1)

Prepare a survey on the various deployment models that are currently trending in the industries. Prepare a poster having public, private, hybrid, community clouds. Use your creative strengths to be innovative in preparing the survey.

5 Hybrid Trends 2022 by IBM

1. **Open-source clouds for specific use cases**

   IBM now provides open hybrid cloud will allow firms to manage not only their current public and private clouds, but those they may add in the future. Open hybrid cloud will enable scaling and app deployment across public and private clouds for greater consistency and efficiency.

2. **Finding unified solutions for security and compliance**

   Also, IBM started working with a managed services provider (MSP) for security needs. Establishing zero-trust policies and solutions. Merging DevOps best practices with on premises strengths like network-based visibility.

3. **Cloud retooling**

   Cloud bursting is a strategic design move in which a private cloud can "burst" into a public cloud during peak operational hours when it requires additional resources. The concept has been around for many years, but as recently as this spring IBM used cloud bursting to optimize electronic design automation (EDA).

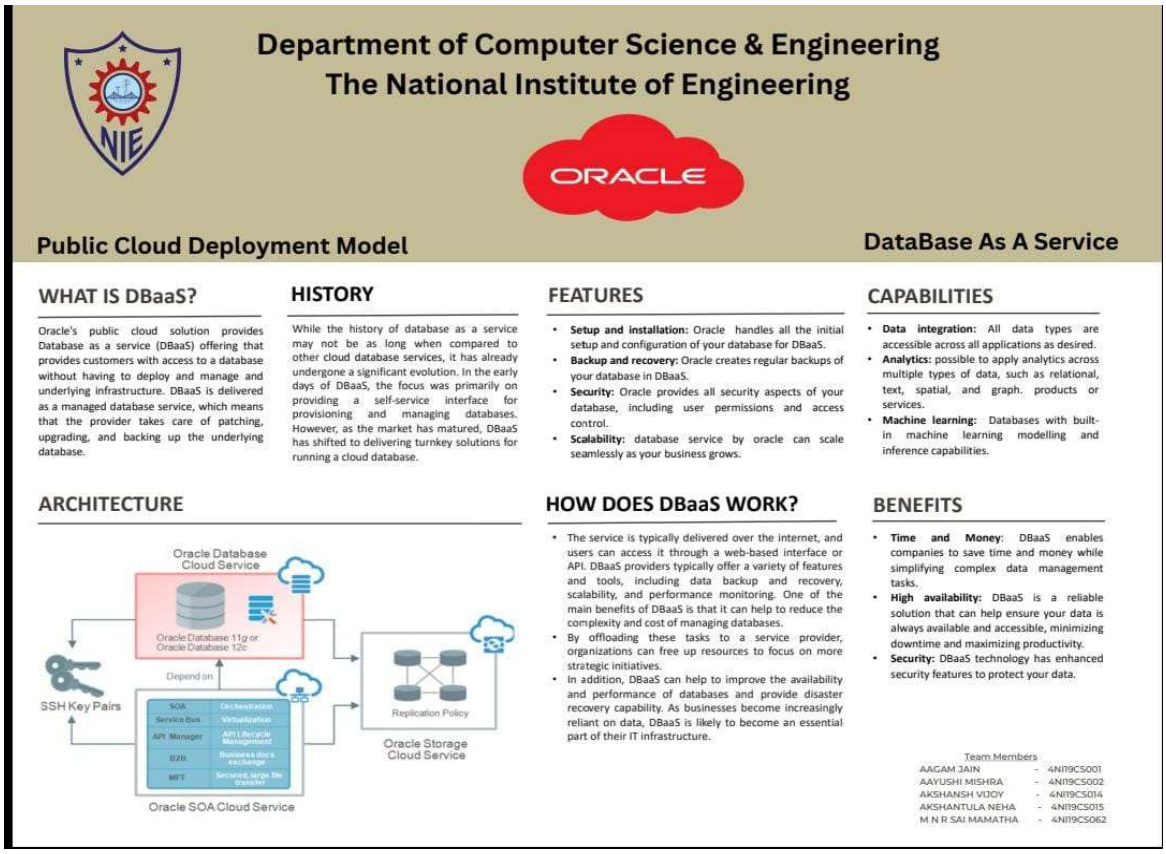
4. **IBM and Microsoft are collaborating to support a hybrid cloud approach to mainframe application modernization**

   This approach leverages the core strengths and attributes of the IBM mainframe together with the extensive cloud services available on Microsoft Azure. A hybrid strategy that includes IBM zSystems and Microsoft Azure can allow for rapid development leveraging an agile DevOps approach, to make it easier to access mainframe applications and share information, and to address skills gaps with open tooling and IT automation. Together, IBM and Microsoft can help accelerate mainframe application modernization, to improve agility, optimize costs and lower risk.

5. **Businesses want a platform that enables AI and automation.**

   AI and automation offer incredible promise for businesses—helping them increase efficiency, profitability and business value while optimizing on cost and driving revenue growth. IBM provide a platform for AI—where the system brings in machine learning. There are different ways to take advantage of AI and automation, and organizations want a hybrid cloud platform that supports it.

Poster:



Question 2)

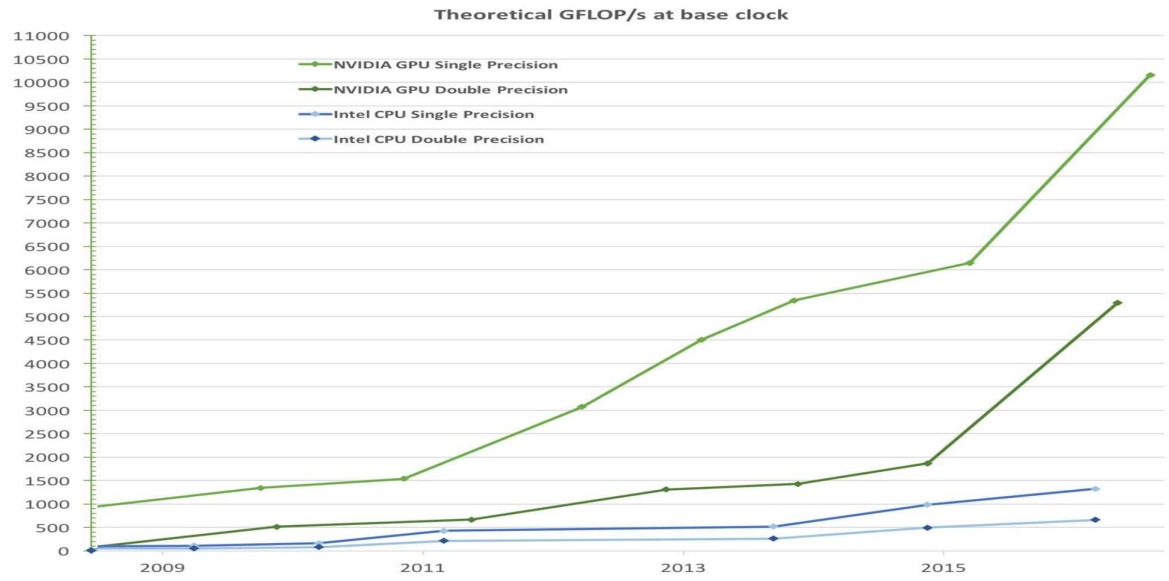
Compare CPU and GPU chips in terms of their strengths and weaknesses. In particular, discuss the trade offs between power efficiency, programmability and performance. Also compare various MPP architectures in processor selection, performance target, efficiency and packaging constraints.

CPU vs GPU

Parameter	AMD Radeon™ RX 5600 XT Drivers & Support	Intel® Core™ i9 Processors
Introduction	The AMD Radeon RX 5600 XT is a fast upper mid-range gaming graphics card for desktops.	Intel Core i9 is a line of Intel CPUs introduced in May 2017.
Transistor Count	Transistor Count: 10.3 B	2.95B transistors with a technology node size of 7nm
Cores and Threads	2304 Cores	8 cores and 16 threads
Memory	Memory Type: GDDR6 and max memory size: 6 GB	128 GB DDR4-2666
Cache	L2 Cache: 3 MB	16 MB Intel Smart Cache
Image		

Performance Analysis: CPU vs GPU

Latency and throughput are the variables taken into consideration for performance comparison. A Central Processing Unit (CPU) is a latency-optimized general-purpose processor that is designed to handle a wide range of distinct tasks sequentially, while a Graphics Processing Unit (GPU) is a throughput-optimized specialized processor designed for high-end parallel computing. This is the core difference between CPUs and GPUs. CPUs are geared for latency or finishing a task as quickly as feasible. GPUs are optimized for throughput, as in, they are slow, but they operate on bulks of data at once.



Programmability: CPU vs GPU

A GPU is essential for the fast, graphics-intensive rendering of the gaming world. Real-time rendering of complex 3D visuals and special effects demands very powerful processing. While GPUs can process data several orders of magnitude faster than a CPU due to massive parallelism. In addition to rendering video, GPUs also perform exceptionally well when performing machine learning, risk modeling, and financial simulations. Additionally, GPUs are employed in the mining of cryptocurrencies like Bitcoin and Ethereum.

However, CPUs are more adaptable than GPUs. A GPU cannot manage all of a computer's inputs and outputs as a CPU can, which has a huge and diverse set of instructions. The CPU is appropriate for a wide range of workloads, especially those where per-core performance or latency are crucial factors. The CPU, a potent execution engine, concentrates its fewer cores on specific tasks and on finishing things swiftly. This makes it particularly well-suited for tasks like running databases and serial computing.

Massively Parallel Processing

Massively parallel is the term for using a large number of computer processors to simultaneously perform a set of coordinated computations in parallel. It is a storage system made to manage the coordinated execution of program activities by numerous processors. With each CPU using its own operating system and memory, this coordinated processing can be applied to different sections of a program.

There are two types of MPP database architecture: grid computing and computer clustering.

Grid Computing:

Grid Computing can be defined as a network of computers working together to perform a task that would rather be difficult for a single machine. All machines on that network work under the same protocol to act as a virtual supercomputer.

Computer Clustering:

Cluster computing refers to a set of computers or devices that work together so that they can be viewed as a single system.

Efficiency: Grid and Cluster

Grid:

Grid computing attempts to achieve high computational performance by non-traditional means. It is about leveraging your available resources and idle processor cycles to solve a problem while at the same time maximizing efficiency and reducing your total cost of ownership. The main advantage of grid computing is that it allows one to share computer resources across networks.

This can both increase the computational power available to programs and reduce the number of machines needed by an organization. It allows for linking a large number of low-cost machines together, rather than spending a large amount of money on a single machine or supercomputer with a larger processing capability. It also allows for applications to be more easily scaled since additional machines can be added to the grid.

Cluster:

A computer cluster helps to solve complex operations more efficiently with much faster processing speed and high data integrity. Hence, cluster computing provides an efficient, relatively inexpensive, unconventional solution since the connecting group of the systems as computer clusters are much cheaper as compared to mainframe computers. The added benefit of Cluster Computing is that it can be expanded easily by adding an additional desktop workstation to the system.

Performance Target:

Grid:

Through volunteer computing, this technology has been used to solve computationally demanding scientific, mathematical, and academic problems. It is also used in commercial enterprises for a variety of applications, including seismic analysis, drug discovery, economic forecasting, and back office data processing in support of e-commerce and Web services.

Cluster:

Ideally, a cluster functions as if it were a single system. A user accessing the cluster does not need to know whether the system is a cluster or an individual machine. A cluster is designed to minimize latency and prevent bottlenecks in node-to-node communication as well as to provide for load balancing and high availability. These characteristics are common to a multitude of technical and scientific applications. Its use includes weather forecasting, fluid dynamics, drug design, and rendering farms, to name a few.

Packaging constraints:

Grid:

The nodes are slackly bundled in grid computing. Standard LANs or WANs are typically used to connect the nodes of a slack cluster.

Cluster:

Cluster nodes can be arranged slackly or compactly. The nodes of a compact cluster are not connected to peripherals and are tightly packed on one or more racks that are located in a room. A small cluster can make use of a proprietary, high-bandwidth, low-latency communication network. The nodes in a slack cluster are connected to their customary peripherals and may be dispersed throughout the world.

Question 3)

Use of MOSIX for cluster computing. Check with the open literature on current features that have been claimed by designers and developers in supporting linux clusters, gpu clusters, multi clusters and even virtualized clouds. Discuss the advantages and short coming from user perspective.

MOSIX:

MOSIX is a cluster management system targeted for parallel computing on Linux clusters and multi-cluster private clouds. MOSIX Supports automatic resource discovery and dynamic workload distribution. It provides users and applications with the illusion of running on a single computer with multiple processors.

MOSIX is a proprietary distributed operating system. The MOS for UNIX (MOSIX) is a multi-computer Operating System with decentralized management. MOSIX is suited to run compute intensive and applications with small to moderate amounts of I/O over fast, secure networks, in a trusted environment (where all remote nodes are trusted). A typical situation is of several private clusters, each connected internally by Infiniband and externally by Ethernet, forming a private (intra-organization) cloud.

MOSIX is based on Unix and provides a single-systems image as if using one computer with multiple CPUs. It geared to reduce the management complexity of users. The user does not have to login all the time. Also user's do not need to login or copy files to remote nodes. Also there is no need to link applications with special libraries. MOSIX has limited support for shared-memory. MOSIX creates a virtual computer, featuring automatic load balancing by migrating processes from heavily loaded nodes to less used node.

Various kinds of clusters:

- **MOSIX for Linux Clusters**

  MOSIX runs as a virtualization layer in the Linux environment. This layer provides SSI to users and applications along with runtime Linux support. The system runs applications in remote nodes as though they were run locally. It supports both sequential and parallel applications, and can dis-cover resources and migrate software processes

transparently and automatically among Linux nodes. MOSIX can also manage a Linux cluster or a grid of multiple clusters.

Flexible management of a grid allows owners of clusters to share their computational resources among multiple cluster owners. Each cluster can still preserve its autonomy over its own clusters and its ability to disconnect its nodes from the grid at any time. This can be done without disrupt-ing the running programs. A MOSIX-enabled grid can extend indefinitely as long as trust exists among the cluster owners. The condition is to guarantee that guest applications cannot be modified while running in remote clusters. Hostile computers are not allowed to connect to the local network.

- **Single Cluster**

A MOSIX cluster is a set of connected servers and workstations, called "nodes", which are administrated by a single owner and run the same version of MOSIX. In a MOSIX cluster, each node maintains information about availability and the state of the resources of all the nodes. Nevertheless, MOSIX processes can run in remote clusters while still using the environment provided by their respective private home clusters. MOSIX2 is most suitable for running compute intensive applications with low to moderate amount of input/output (I/O). Tests of MOSIX2 show that the performance of several such applications over a 1 Gbit/s campus grid is nearly identical to that of a single cluster.

- **Multi Cluster**

A MOSIX multi-cluster (also called "an intra-organizational multi-cluster") is a collection of private MOSIX clusters that run the same version of MOSIX and are configured to work together. A MOSIX multi-cluster usually belongs to the same organization, but each cluster may be administrated by a different owner or belongs to a different group. The cluster-owners are willing to share their computing resources at least some of the time, but are still allowed to disconnect their clusters from the multi-cluster at any time. In a MOSIX multi-cluster, each node maintains information about availability and the state of the resources of all the nodes in all the connected clusters. Different clusters may (or may not) have a shared environment such as a common NFS file system. From the user's perspective, MOSIX transforms such a multi-cluster into a

single cluster by preserving the user's local run-time environment. In MOSIX multi-clusters there is usually a high degree of trust, i.e., a guarantee that applications are not viewed or tampered with when running in remote clusters. Other possible safety requirements are a secure network and that only authorized nodes, with identifiable IP addresses, are included.

- **MOSIX Cloud**

A MOSIX cloud is a collection of entities such as MOSIX clusters; MOSIX multi-clusters; Linux clusters (such as a group of Linux servers); individual workstations and Virtual Machines (VM). Each entity may possibly run a different version of Linux or MOSIX. In a MOSIX cloud, different entities are usually administrated by different owners and rarely share any file systems (such as NFS). In this cloud, nodes in each entity are aware of one or more nodes in other entities, including their IP addresses and services they are willing to provide, but there is no on-going automatic flow of information between entities. In a MOSIX cloud, users can launch applications from their workstations or a private home-cluster, on target nodes of other entities. These applications have access to files on nodes of these entities, while still allowing the applications to access files on their launching node. This is accomplished by the MOSIX Reach Clouds (MOSRC), which allows applications to run in remote nodes, without the need to copy files to/from remote clusters.

Features of MOSIX:

- Provides a single-system image– Users can login on any node and do not need to know where their programs run– No need to modify or link applications with special libraries– No need to copy files to remote nodes.
- Automatic resource discovery and workload distribution– Load-balancing by process migration– Migrating processes from slower to faster nodes– Migrating processes from nodes that run out of free memory.
- Migratable sockets for direct communication between migrated processes.
- Provides a secure run time environment (sandbox) for guest processes.

- Supports checkpoint and recovery.
- Supports 64-bit x86 architectures.
- Includes tools for automatic installation and configuration.
- Includes an on-line monitor.
- Guest processes can move from one cluster to another.
- Clusters can be shared symmetrically or asymmetrically.
- Cluster owner can assign different priorities to guest processes from other clusters.
- Supports disruptive configurations:
  1. Clusters can join or leave the multi-cluster cloud at any time.
  2. Guest processes move out before disconnecting a cluster.

ADVANTAGES	**DISADVANTAGES**
1.The MOSIX cluster is designed to run like an SMP machine.	1.Its periodic approach of information dissemination.
2. It utilizes a "fork and forget" method of dealing with large nodes.	2.Waste of network bandwidth and CPU time.
3.MOSIX handles the optimizing of the resources across multiple machines.	3.Increased communication and virtualization overheads.
4. Decentralized Control and Autonomy	4.Does not improve performance of intensive I/O processes.
5. Flood Control	5.Does not improve performance of non-computational applications, such as web or mail servers.
6.Memory Ushering	

	6.Does not support HA and Linux-system calls.

Question 4)

Install xen on ubuntu machine in 2 methods – from the binary code or from source code. Compare installation guides for 2 methods used. Describe the dependencies of utilities and packages along with troubleshooting tips.

COMPILING AND INSTALLING XEN FROM SOURCE

1) Obtaining the Xen Project Source Code

The primary ways to obtain the Xen Project source code for a stable release are via the release tarballs or by cloning from the appropriate Mercurial source repository.

- **Release Tarballs:**

The latest Xen Project releases are linked to from The Xen.org download page

- **Git:**

The source code repositories are hosted using the Git version control system on xenbits. To clone the source first install Git using your distro's package manager. Then execute the following command:

$ git clone URL

Where URL is the URL of the repository you wish to clone. In our case we should use:

$ git clone git://xenbits.xen.org/xen.git

2) Building from Source

- **Updated /sbin/installkernel on Linux**

Linux distributions shipping with grub2 will need to ensure that their /sbin/installkernel script, which has to be provided by each Linux distribution, copies the the kernel configuration upon a custom kernel install time. The requirement for the config file comes from upstream grub2 /etc/grub.d/20_linux_xen which will only add xen as an instance to your grub.cfg if and only if it finds in your config file either of:

CONFIG_XEN_DOM0=y

CONFIG_XEN_PRIVILEGED_GUEST=y

- **Build Dependencies**

Xen Project uses several external libraries and tools.

Build Dependencies - Ubuntu

Under Debian / Ubuntu (and derived distributions) install the build-essential package:

apt-get install build-essential

you also need to install these additional debs:

apt-get install bcc bin86 gawk bridge-utils iproute libcurl3 libcurl4-openssl-dev bzip2 module-init-tools transfig tgif

apt-get install texinfo texlive-latex-base texlive-latex-recommended texlive-fonts-extra texlive-fonts-recommended pciutils-dev mercurial

apt-get install make gcc libc6-dev zlib1g-dev python python-dev python-twisted libncurses5-dev patch libvncserver-dev libsdl-dev libjpeg62-turbo-dev

apt-get install iasl libbz2-dev e2fslibs-dev git-core uuid-dev ocaml ocaml-findlib libx11-dev bison flex xz-utils libyajl-dev

apt-get install gettext libpixman-1-dev libaio-dev markdown pandoc

One useful shortcut can be to use your distributions package manager to install all the prerequisite packages is to install those packages which are noted as being required to build the distribution's own Xen packages.

apt-get build-dep xen

3) Configure

From Xen Project 4.2 onwards, the software uses the commonly used autoconf tool to provide compile time configurability of the toolstack. This allows some control of what features are built into Xen Project, as well as compile time sanity checking. To configure Xen Project, simply run the provided configure script:

$ ./configure

To see the various options run the configure script with --help

e.g.: $ ./configure --help

Optional Features:

--disable-option-checking ignore unrecognized --enable/--with options

--disable-FEATURE do not include FEATURE (same as --enable-FEATURE=no)

--enable-FEATURE[=ARG] include FEATURE [ARG=yes]

--enable-githttp Download GIT repositories via HTTP (default is DISABLED)

- **Use http:// Rather Than git:// to Clone Additional Repositories**

When building the software from Mercurial, the build system will automatically clone several additional repositories from the network. Some of these repositories use the version control system which uses its own protocol on a specific port. Sometimes this causes issues due to firewalls etc blocking the git port. This can be worked around by instructing the Xen build system to clone such repositories using a less efficient HTTP based protocol:

$ ./configure --enable-githttp

- **Library Installation Directory**

Xen Project 4.2 onwards defaults to installing libraries into /usr/lib by default and from 4.3 onwards defaults to installing to /usr/local/lib by default.

Users on systems which use /usr/local/lib64 for 64-bit libraries should use the --libdir option.
$ ./configure --libdir=/usr/local/lib64

Failure to do this usually results in errors about libraries not found or using older versions of the libraries which will likely not work.

- **Systemd**

If the target system uses systemd, do not forget to enable it:

$ ./configure --enable-systemd

- **Python Prefix and Module Layout**

On some distros (e.g. Debian and Ubuntu) Xen Project may install the python parts of the code into the wrong place (See Debian bug #693721). Therefore it is necessary to set PYTHON_PREFIX_ARG=--install-layout=deb:

$ cat .config PYTHON_PREFIX_ARG=--install-layout=deb

4) Build & Install

To build all components (hypervisor, tools, docs, stubdomains, etc) you can use the dist target.
$ make dist

If you wish to just (re)build a single component you can use the appropriate dist-COMPONENT target:

$ make dist-xen
$ make dist-tools
$ make dist-docs
$ ... etc ...

All of the above targets will build and install the appropriate component into the dist subdirectory but not actually install onto the system.

To install onto the local machine simply call the install target (as root):

make install

As with dist you can also install individual components using the appropriate install-COMPONENT target:

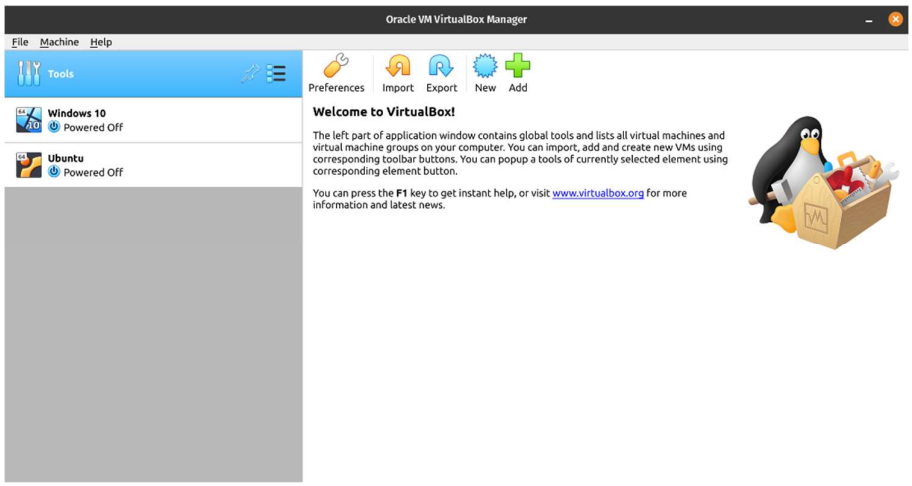
make install-xen
make install-tools
make install-docs
... etc ..

Question 5)

Install any 5 VM's of your choice and network all of them together and you should be able to access internet on any of VMs. Document the procedure with appropriate screenshots.

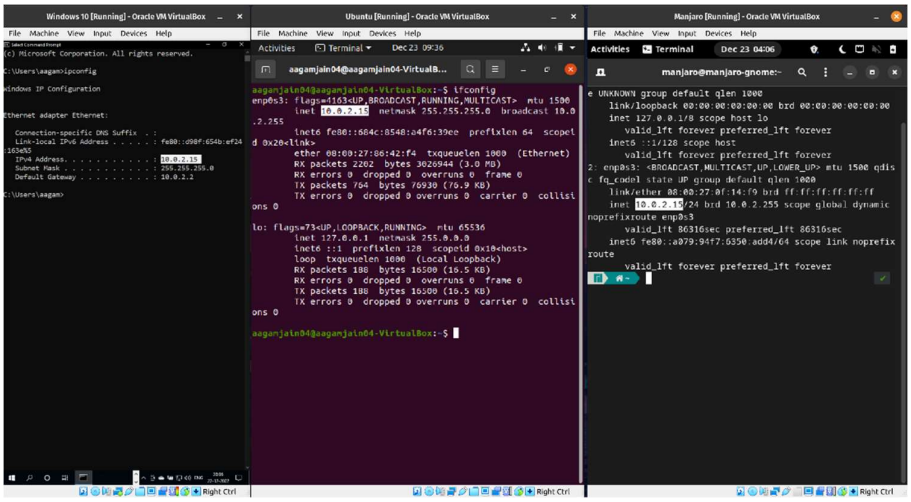
Steps :

1) Install virtual machine.

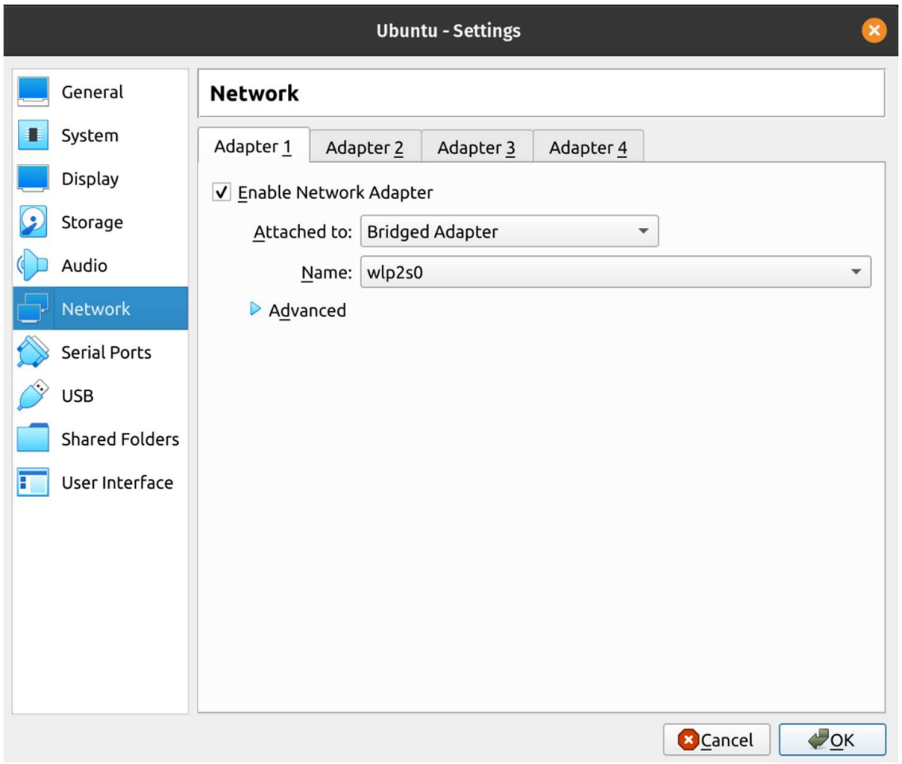


2) Download Iso file of different operating system and launch them on virtual box.

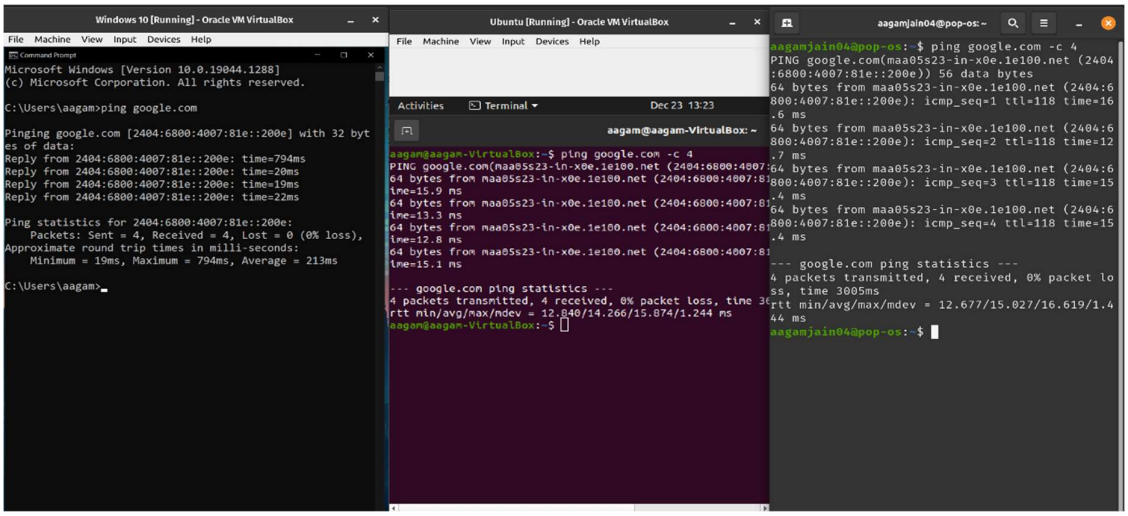
3) Check IP address of each operating system using command ifconfig/ipconfig in the terminal.
   Each OS will have same address which is 10.0.2.15



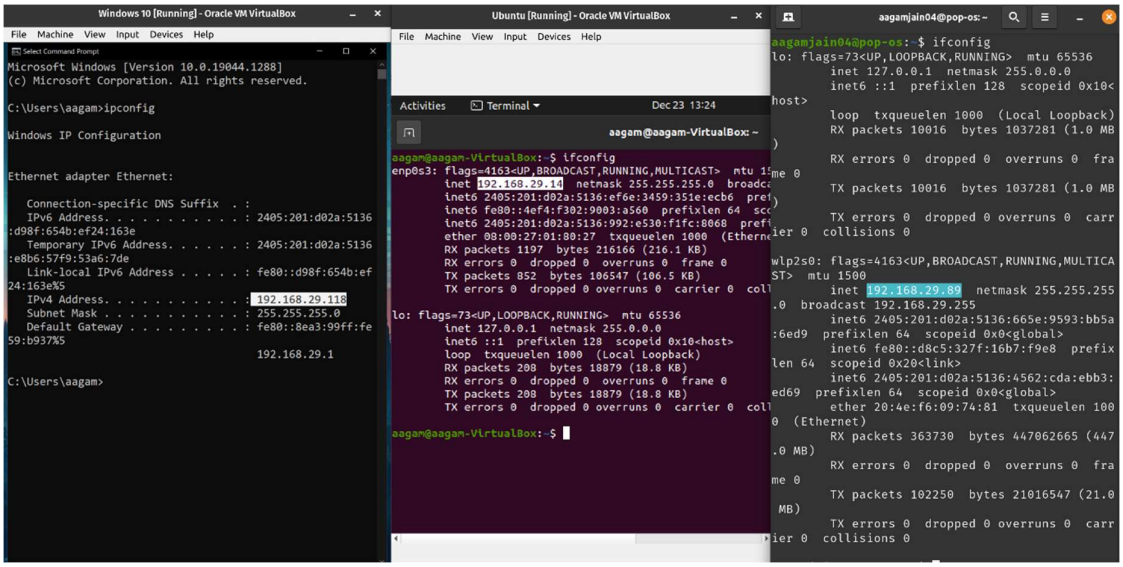
4) In the network tab change NAT network to Bridge Adaptor so that each OS will have its own unique IP address.

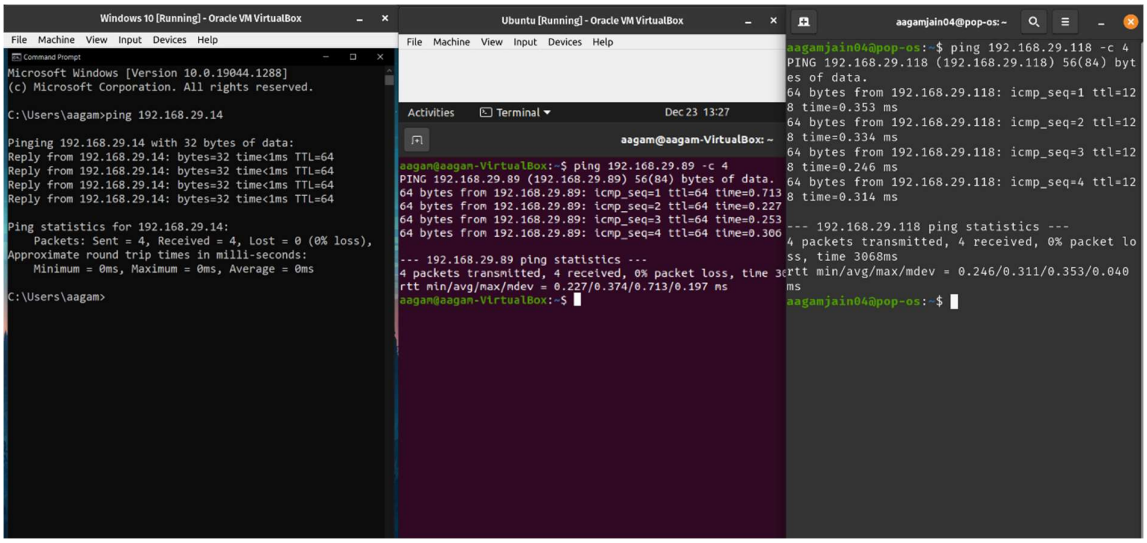


5) To make sure each OS is having internet access, type ‘ping google.com’ in the terminal and make sure packets are received.



6) To check if OS's can talk to each other ping each other using IP address and make sure packets are received.



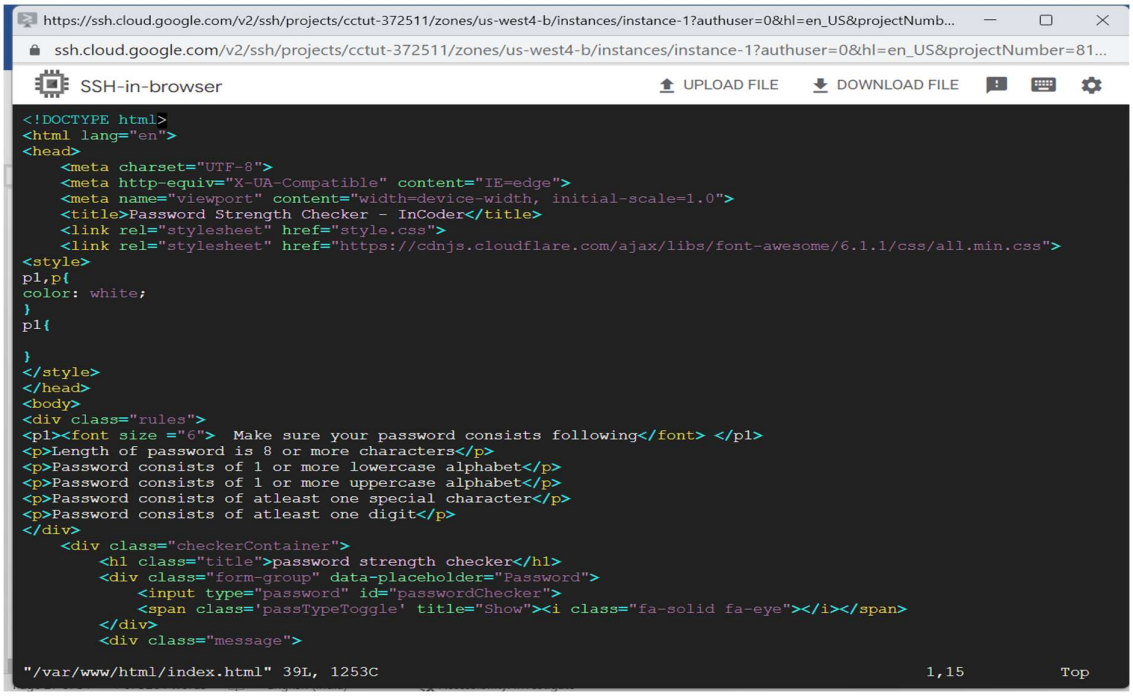

Windows 10 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Command Prompt
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.
C:\Users\aagam>ping 192.168.29.14
Pinging 192.168.29.14 with 32 bytes of data:
Reply from 192.168.29.14: bytes=32 time<1ms TTL=64
Reply from 192.168.29.14: bytes=32 time<1ms TTL=64
Reply from 192.168.29.14: bytes=32 time<1ms TTL=64
Reply from 192.168.29.14: bytes=32 time<1ms TTL=64
Ping statistics for 192.168.29.14:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\aagam>
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Dec 23 13:27
aagam@aagam-VirtualBox: ~
aagam@aagam-VirtualBox:~$ ping 192.168.29.89 -c 4
PING 192.168.29.89 (192.168.29.89) 56(84) bytes of data.
64 bytes from 192.168.29.89: icmp_seq=1 ttl=64 time=0.713
64 bytes from 192.168.29.89: icmp_seq=2 ttl=64 time=0.227
64 bytes from 192.168.29.89: icmp_seq=3 ttl=64 time=0.253
64 bytes from 192.168.29.89: icmp_seq=4 ttl=64 time=0.306
--- 192.168.29.89 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3
rtt min/avg/max/mdev = 0.227/0.374/0.713/0.197 ms
aagam@aagam-VirtualBox:~$
aagamjain04@pop-os: ~
aagamjain04@pop-os:~$ ping 192.168.29.118 -c 4
PING 192.168.29.118 (192.168.29.118) 56(84) byt
es of data.
64 bytes from 192.168.29.118: icmp_seq=1 ttl=12
8 time=0.353 ms
64 bytes from 192.168.29.118: icmp_seq=2 ttl=12
8 time=0.334 ms
64 bytes from 192.168.29.118: icmp_seq=3 ttl=12
8 time=0.246 ms
64 bytes from 192.168.29.118: icmp_seq=4 ttl=12
8 time=0.314 ms
--- 192.168.29.118 ping statistics ---
4 packets transmitted, 4 received, 0% packet lo
ss, time 3068ms
rtt min/avg/max/mdev = 0.246/0.311/0.353/0.040
ms
aagamjain04@pop-os:~$

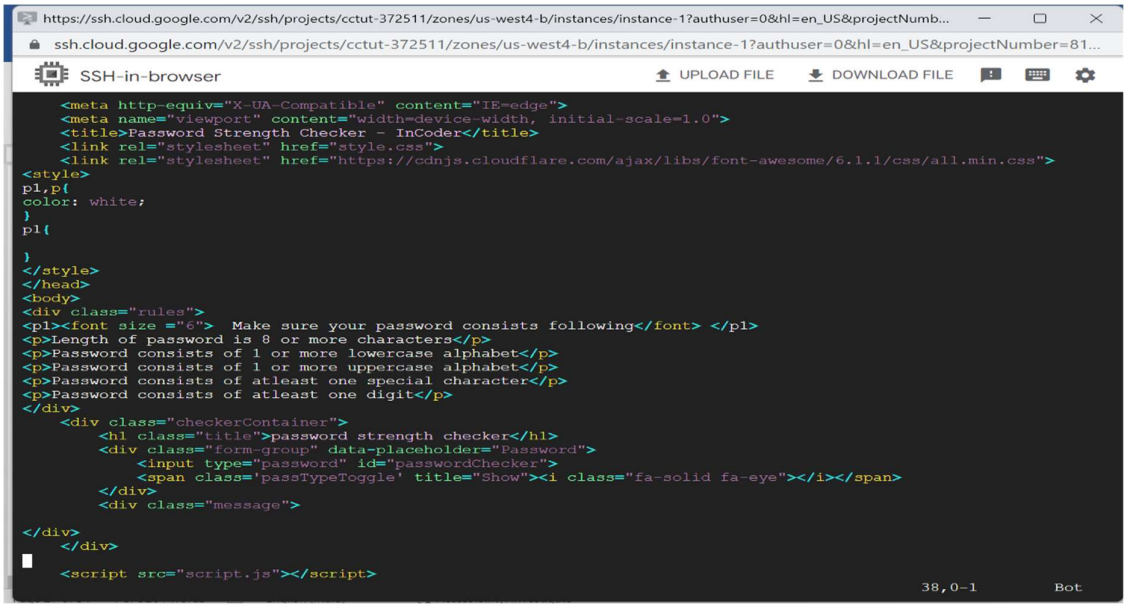
Question 6)

Write any web or high performance application of your choice on google cloud platform. Document the code with supporting screenshots.

Code:

Index.html:





style.css:

https://ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumber=81...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
@import url("https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap");

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}

body {
  display: flex;
  height: 100vh;
  align-items: center;
  justify-content: center;
  background-color: rgb(0 0 0 / 85%);
}

.checkerContainer {
  width: 25rem;
  margin: 1rem;
  display: flex;
  align-items: center;
  border-radius: 0.6rem;
  flex-direction: column;
  justify-content: center;
  background: rgb(0 0 0 / 35%);
  box-shadow: 0 0 20px rgb(76 76 76 / 33%);
}

.checkerContainer .title {
  color: #ccc;
  margin-top: 0.8rem;
  margin-bottom: 0.8rem;
  font-size: clamp(1rem, 4vw, 1.6rem);
```

https://ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&

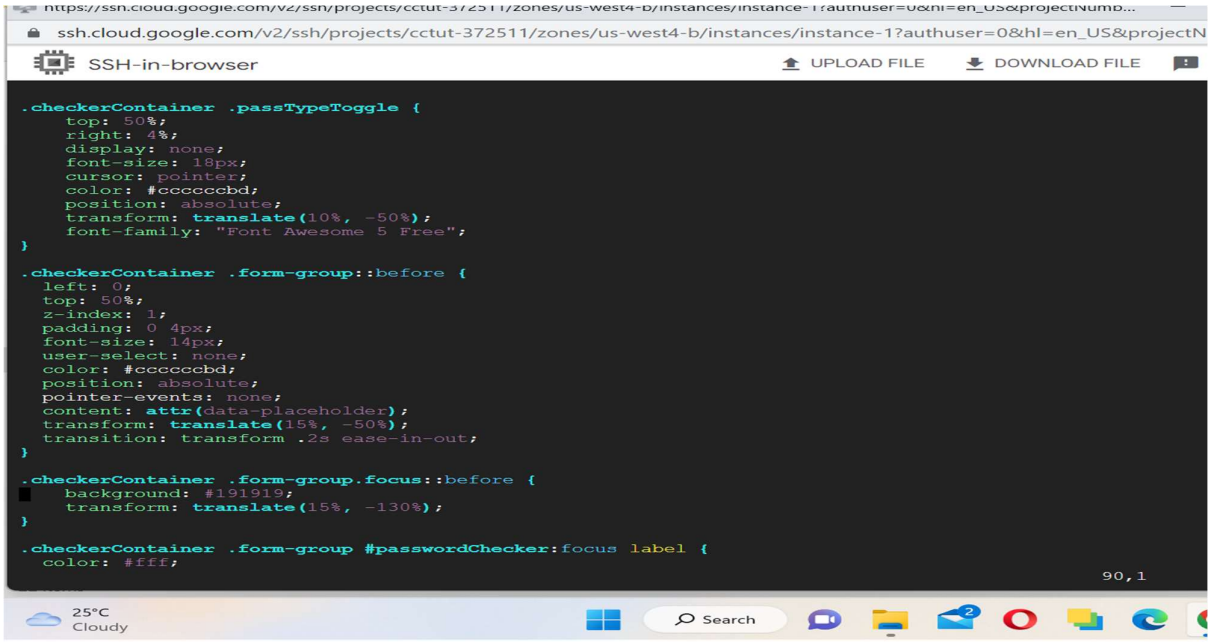
SSH-in-browser UPLOAD FILE

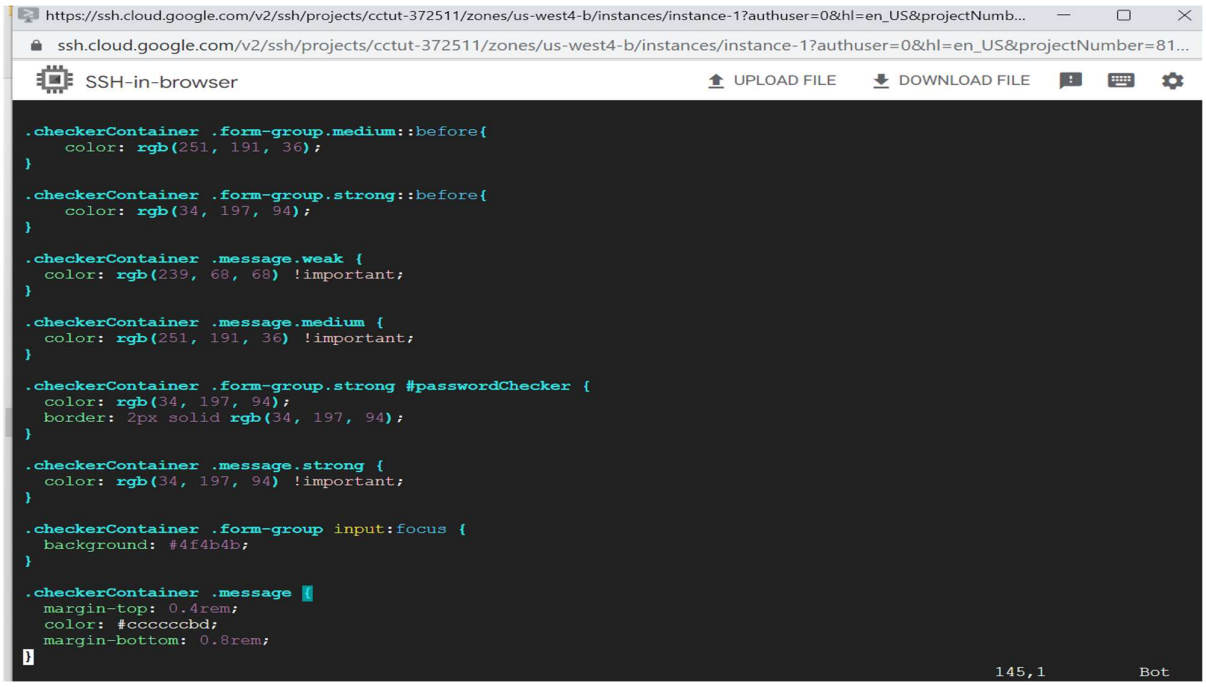
```
  margin-bottom: 0.8rem;
  font-size: clamp(1rem, 4vw, 1.6rem);
}

.checkerContainer .form-group {
  height: 2.6rem;
  display: flex;
  position: relative;
  align-items: center;
  justify-content: center;
  width: calc(100% - 4rem);
}

.checkerContainer .form-group #passwordChecker {
    width: 100%;
    height: 100%;
    outline: none;
    font-size: 1rem;
    margin-top: 0.2rem;
    color: #cccccbd;
    position: relative;
    letter-spacing: 1px;
    border-radius: 0.4rem;
    background: transparent;
    padding: 0 2rem 0 0.5rem;
    border: 3px solid #3b3b3b;
    transition: background 0.19s ease-in-out;
}

.checkerContainer .passTypeToggle {
    top: 50%;
    right: 4%;
    display: none;
    font-size: 18px;
    cursor: pointer;
```


ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectN
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
.checkerContainer .passTypeToggle {
top: 50%;
right: 4%;
display: none;
font-size: 18px;
cursor: pointer;
color: #ccccccbd;
position: absolute;
transform: translate(10%, -50%);
font-family: "Font Awesome 5 Free";
}
.checkerContainer .form-group::before {
left: 0;
top: 50%;
z-index: 1;
padding: 0 4px;
font-size: 14px;
user-select: none;
color: #ccccccbd;
position: absolute;
pointer-events: none;
content: attr(data-placeholder);
transform: translate(15%, -50%);
transition: transform .2s ease-in-out;
}
.checkerContainer .form-group.focus::before {
background: #191919;
transform: translate(15%, -130%);
}
.checkerContainer .form-group #passwordChecker:focus label {
color: #fff;
90,1
25°C
Cloudy
Search


https://ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumb...
ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumber=81...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
.checkerContainer .form-group.medium::before{
color: rgb(251, 191, 36);
}
.checkerContainer .form-group.strong::before{
color: rgb(34, 197, 94);
}
.checkerContainer .message.weak {
color: rgb(239, 68, 68) !important;
}
.checkerContainer .message.medium {
color: rgb(251, 191, 36) !important;
}
.checkerContainer .form-group.strong #passwordChecker {
color: rgb(34, 197, 94);
border: 2px solid rgb(34, 197, 94);
}
.checkerContainer .message.strong {
color: rgb(34, 197, 94) !important;
}
.checkerContainer .form-group input:focus {
background: #4f4b4b;
}
.checkerContainer .message {
margin-top: 0.4rem;
color: #ccccccbd;
margin-bottom: 0.8rem;
}
145,1
Bot

https://ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumb...

ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumber=81...

SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
.checkerContainer .form-group #passwordChecker:focus label {
  color: #fff;
}

.checkerContainer .form-group.weak #passwordChecker {
  color: rgb(239, 68, 68);
  border: 2px solid rgb(239, 68, 68);
}

.checkerContainer .form-group.weak::before{
    color: rgb(239, 68, 68);
}

.checkerContainer .form-group.medium #passwordChecker {
  color: rgb(251, 191, 36);
  border: 2px solid rgb(251, 191, 36);
}

.checkerContainer .form-group.medium::before{
    color: rgb(251, 191, 36);
}

.checkerContainer .form-group.strong::before{
    color: rgb(34, 197, 94);
}

.checkerContainer .message.weak {
  color: rgb(239, 68, 68) !important;
}

.checkerContainer .message.medium {
  color: rgb(251, 191, 36) !important;
}
                                                    122,1         83%
```

script.js

https://ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumb...

ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumber=81...

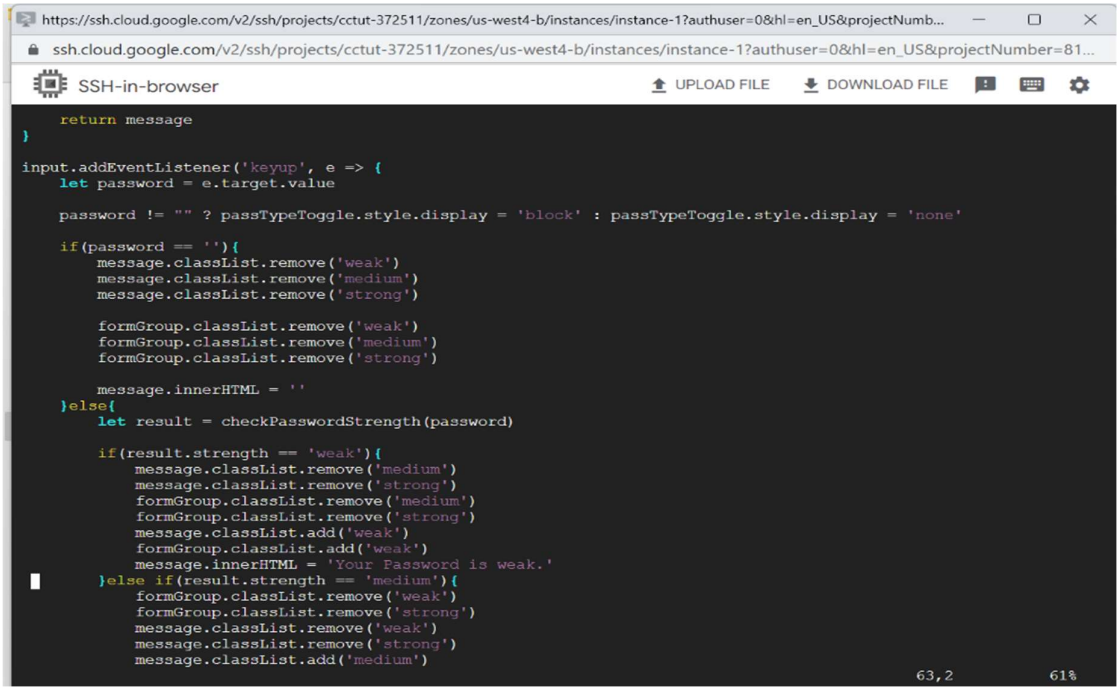
SSH-in-browser UPLOAD FILE DOWNLOAD FILE

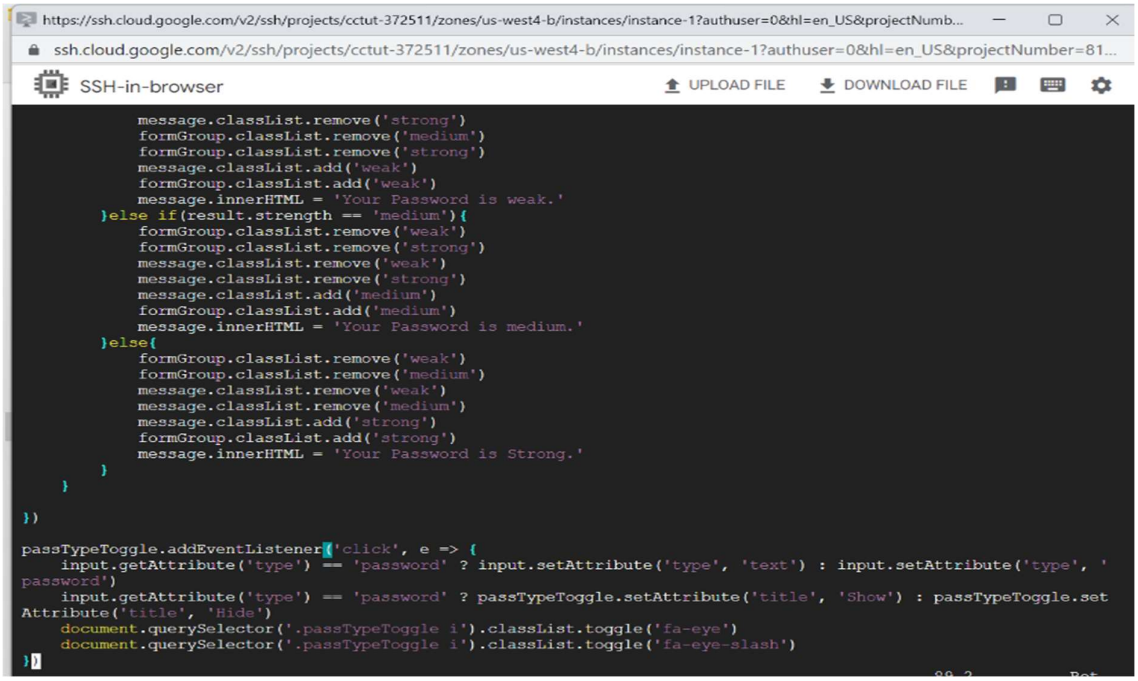
```
let input = document.querySelector('#passwordChecker')
let formGroup = document.querySelector('.form-group')
let message = document.querySelector('.message')
let passTypeToggle = document.querySelector('.passTypeToggle')
let strongPassword = new RegExp('(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[^A-Za-z0-9])(?=.{8,})')
let mediumPassword = new RegExp('((?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[^A-Za-z0-9])(?=.{6,}))|((?=.*[a-z])(?=.
*[A-Z])(?=.*[^A-Za-z0-9])(?=.{8,}))')

document.body.addEventListener('click', function (e) {
    if (input.contains(e.target)) {
        formGroup.classList.add('focus')
    } else {
        if(input.value == ''){
            formGroup.classList.remove('focus')
        }
    }
});

let checkPasswordStrength = (password) => {
    let message = {}

    if(strongPassword.test(password)) {
        message = {
            strength : 'strong'
        }
    } else if(mediumPassword.test(password)) {
        message = {
            strength : 'medium'
        }
    } else {
        message = {
            strength : 'weak'
        }
    }
    return message
                                                    1,2           Top
```


https://ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumb...
ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumber=81...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
return message
}
input.addEventListener('keyup', e => {
let password = e.target.value
password != "" ? passTypeToggle.style.display = 'block' : passTypeToggle.style.display = 'none'
if(password == ''){
message.classList.remove('weak')
message.classList.remove('medium')
message.classList.remove('strong')
formGroup.classList.remove('weak')
formGroup.classList.remove('medium')
formGroup.classList.remove('strong')
message.innerHTML = ''
}else{
let result = checkPasswordStrength(password)
if(result.strength == 'weak'){
message.classList.remove('medium')
message.classList.remove('strong')
formGroup.classList.remove('medium')
formGroup.classList.remove('strong')
message.classList.add('weak')
formGroup.classList.add('weak')
message.innerHTML = 'Your Password is weak.'
}else if(result.strength == 'medium'){
formGroup.classList.remove('weak')
formGroup.classList.remove('strong')
message.classList.remove('weak')
message.classList.remove('strong')
message.classList.add('medium')
63,2
61%

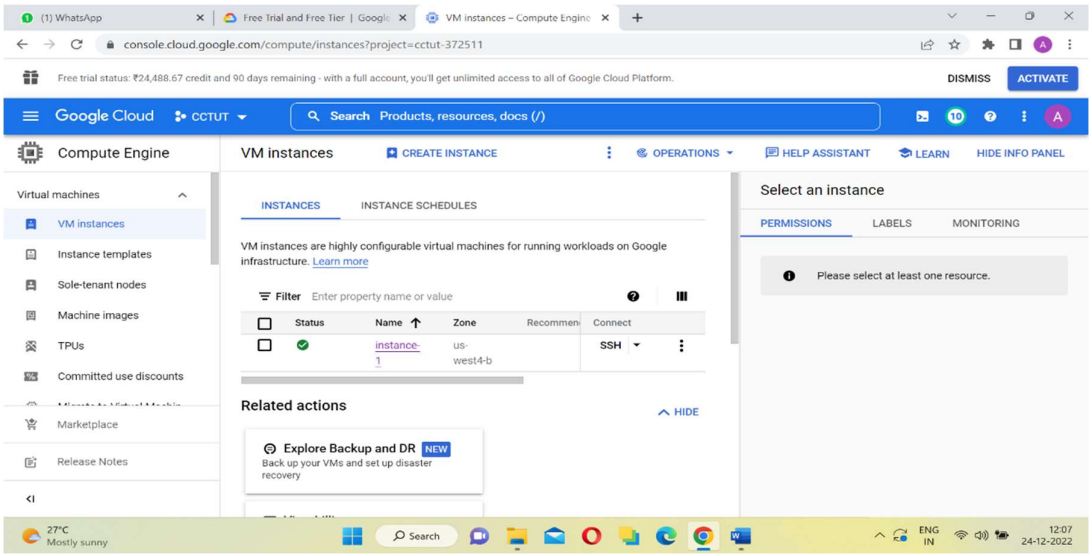

https://ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumb...
ssh.cloud.google.com/v2/ssh/projects/cctut-372511/zones/us-west4-b/instances/instance-1?authuser=0&hl=en_US&projectNumber=81...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
message.classList.remove('strong')
formGroup.classList.remove('medium')
formGroup.classList.remove('strong')
message.classList.add('weak')
formGroup.classList.add('weak')
message.innerHTML = 'Your Password is weak.'
}else if(result.strength == 'medium'){
formGroup.classList.remove('weak')
formGroup.classList.remove('strong')
message.classList.remove('weak')
message.classList.remove('strong')
message.classList.add('medium')
formGroup.classList.add('medium')
message.innerHTML = 'Your Password is medium.'
}else{
formGroup.classList.remove('weak')
formGroup.classList.remove('medium')
message.classList.remove('weak')
message.classList.remove('medium')
message.classList.add('strong')
formGroup.classList.add('strong')
message.innerHTML = 'Your Password is Strong.'
}
}
})
passTypeToggle.addEventListener('click', e => {
input.getAttribute('type') == 'password' ? input.setAttribute('type', 'text') : input.setAttribute('type', '
password')
input.getAttribute('type') == 'password' ? passTypeToggle.setAttribute('title', 'Show') : passTypeToggle.set
Attribute('title', 'Hide')
document.querySelector('.passTypeToggle i').classList.toggle('fa-eye')
document.querySelector('.passTypeToggle i').classList.toggle('fa-eye-slash')
})

Access for the source code:

https://github.com/akshantula-neha/Cloud-Computing.git

Steps to deploy the above code on Google Cloud Platform:

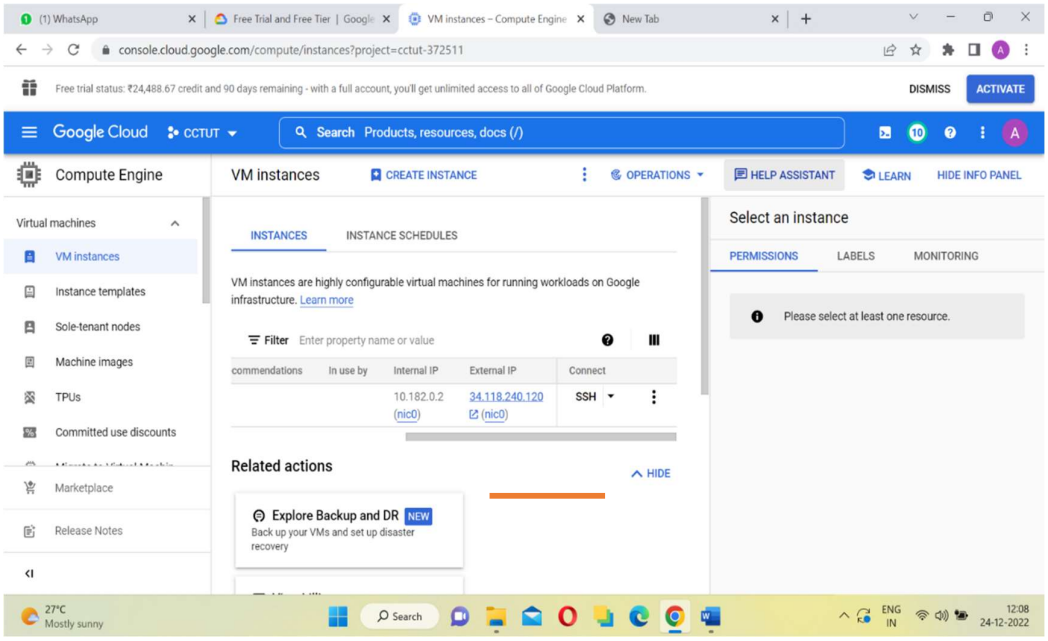
- Create a google cloud account
- Create a new project and a VM instance with suitable configurations if dynamic website



is to be created. Else buckets can be used to create static website.

- Open SSH terminal and execute following commands:
  - **sudo su** - to get into root user.
  - **apt –get install apache2 server** - installing most widely used apache server.
  - **vi /var/www/html/ index.html** -open index.html file and replace it with the code of website.
  - Save code and exit from the vi editor.

- Go back to instances on gcp and copy the corresponding external ip addresss of the particular instance being used.



- Paste the ip address as the url on any web browser to access the website.