**Day - 1**

Cloud computing models

* + Deployment model
  + Service model

Deployment model types

-Public

-Private

-Hybrid

Service models types -

* + IAAS-- INFRASTRUCTURE AS A SERVICE
  + PAAS-- PLATFORM AS A SERVICE
  + SAAS-- SOFTWARE AS A SERVICE

Setup Git, Github, visual studio code

Create cloud folder in d drive (if d  drive is not available create in c drive)

Create web folder inside cloud folder

Example:  d:/cloud/web

Open visual studio code  -->>  go to open folder  -->> select the created cloud folder and tick checkbox and select trust authors

Click on new file option and type index.html

In the code , type html and select html5 option -->> u will get a format

In the body of the code, type  <h1> My First Webpage </h1>

iIf there is no ' go live ' option in lower right corner, go to extensions  -->> search for live servers and install it

Go to github.com and login

On the right side, click on + option to create a new repository  and give a name to the new repository

Cloud -->> web folder -->> click on the path and type cmd and press enter

You will be taken to command prompt

In cmd, type the following commands

echo "# lakshmiwebsite" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin  <https://github.com/lakshmi301/lakshmiwebsite.git>

git push -u origin main or push an existing repository from the command line

git remote add origin  <https://github.com/lakshmi301/lakshmiwebsite.git>

git branch -M main

git push-u origin main

 create a new file in vsc -->> type the file name contact and get the default html code

In the default format, modify it

Change <title> document</title>  to <title>contact</title>

In the body, type <h1> Contact us </h1>

Go to goolge maps -->> select a place -->> share option -->>embed code -->> copy the link

Paste this link in  the code below <h1> Contact us </h1>

Go live and u will see the location

Create html files -->> course, contact, faculties, images , video

**Day - 2**

* + Deploying static website on github

Login to github.com

Click on repositories

Select  the repository that u have created yesterday

Click on settings ->> pages (on the left column)

Under branch, change "none" to main and click on save  and then refresh the page

You will receive a link

* + Link every pages from the index.html using navigation bar

Index.html

Course.html

Faculties.html

Images.html

Video.html

Contact.html

Or

sindhivapwebsite  --  clone the repository

Create gitref folder indside your cloud folder

Cd to cloud

Git clone

http:// github.com/raghuprasadkr/sindhivapwebsite.git

**Day 3**

Agenda:

* + webservers
  + download and configure nginex
  + download and configure xampp
  + web server and database server

* + open chrome

2 type xginx download for windows

3)click on the stable version for windows-1.26 version

4)unzip the nginx

5)copy the entire nginx folder put it into c folder

Checking for local host: type localhost in the chrome

[u will get "welcome to nginx"

Then open nginx folder->html->index file

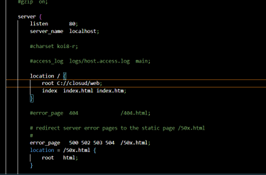
Open index file in visual studio code

Modify the code

Delete the body and write  <h1>I am learning nginx</h1>

Open nginx.conf file in conf folder under nginx folder in visual studio code

Now copy ur folder address[sindhi website folder] and paste it in visual studio code



Now go to chrome and type local host

U shld get ur website

* + Download and configure xampp

Open xampp application and selct start next to apache

If u get error->> click on config and select config and click on apache(httpd.conf)

Ull get a notepad file ->> go to 17 or 18th line ->> listen 80

Change it to listen 8081

Now comeback and click on start

It will now run

Now locate ur xampp folder by clicking on shell next to apache

It will give th elocation of ur  folder either c or d drive

Now go to the xampp location click on htdocs and copy paste ur project file in this folder

Now go to chrome and type example:-  localhost:8081/web

U will get ur website

**Day - 4**

Agenda-

* + Cloud platforms
  + Aws
  + Microsoft Azure
  + Google cloud
  + Heroku- sales force
  + Docker and docker hub

1] AWS

Ec2 - elastic cloud compute

Create virtual server

S3 - simple storage service

RDS - remote data service

SNS - simple notification service

Login to aws

Search for EC2

On right hand side--->> u can see data centre--->>set it to mumbai

Under "Launch an instance"

Type ur name under my server name

Then Select ubuntu

Select Servers r free tier eligible

To use the server it requires a key value pair

Give ur name and Click on Create a key value pair

Give the name of the keyvalue pair

RSA

2 formats--- .pem and .ppk format

It Creates a cryptographic password

It will download the key value pair and ask where to save it

We want to connect to server

So  use SSH and check on all boxes and click on launch instance

Now a new server is created

Click on the instance created example-

([i-0115dd97f1bb865cb](https://ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1" \l "Instances:instanceId=i-0115dd97f1bb865cb))

Now u can see the instance running

Click on the instance and select connect and click on connect

 then u can use linux commands

           now download putty for windows-- 64bit x86 version and install the app

Now to delete the instance

Tick the checkbox and under instance state select stop instance

Once the state changes from stopping to stop -->> tick the checkbox and under instance state click on terminate instance

2] Download docker

Docker desktop  --->>  download  for windows

Sign up  <https://hub.docker.com/>

Containerization

Virtualization

On physical server

We used ec2 to create a virtual server

Docker is used as a container -  box

**Cloud computing day 5**

Agenda:-

* + Docker
  + Docker hub
  + Deploying python flask application

Container- instance of an image

Use image we can create n number of  instance and can the application in any port

Volume- data related to work can be used in volume

Open the docker desktop

under the container -> click on the what is container??

After clicking on what is container -> click on the port 8088-> it will display you a congratulations message

Open the docker hub->type python on the search bar n press enter

On the right hand side copy the command" docker pull python'

Copy that on cmd

It will download python on docker and will be visible in the images

Docker images

Docker container list -->>  type these two command on cmd

Create a python flask application

Flask -- micro web frame

or

django

For java --- servlets/jsp

            Steps:

* + create a python flask folder
  + Create a folder called python flask in the c folder
  + open that folder in visual studio code
  + Go to cmd n type python --version to check for python in ur system
  + Create calculator.py file inside the pythyonflask folder            (click on add folder in visual studio code)
  + Create the following methods
  + add(num1,num2)
  + Sub(num1,num2)
  + Mul(num1,num2)
  + Div(num1,num2)
  + After typing the program in visual studio -->> click on run-->click on run without debugging(ctrl+f5)-->click on the terminal

Flask  -- is  a web server used to run python

Pip install flask command which can be given in cmd

* + Create app.py folder in visual studio
  + Click on run--->then run without debugging
  + [http://127.0.0.1:5000](http://127.0.0.1:5000/) such link will be provided
  + Press ctrl +click on the link n not found url will appear on the screen
  + type [http://127.0.0.1:5000](http://127.0.0.1:5000/) /hello in the url n press enter
  + Route- is the url given to function
  + Similar to the the above code few other route is added into the program

Add a new folder called requirements.txt  --->> type Flask

create a docker folder -->>  Dockerfile

**Cloud computing day:-6**

**Agenda**

* + Push image to docker hub
  + Install python in ec2 instance
  + Connect using putty
  + Deploy flask application

Practicals

Stop previously created flask application in docker--->>delete the image

* + Create pyhtonweb folder
  + Open python web using visual studio code
  + Create a virtual environment
  + Activate virtual environment
  + Install flask
  + Create app.py
  + Run app.py

Go to the docker desktop-->containers delete the pythonflask in container first then go to image n delete the pythonflask

Create a pythonweb folder in c drive--->>open that folder in visual studio code

In the terminal or the cmd of pythonweb type the below syntax "virtualenv venv"

virtualenv <name of the virtual>

Venv <

Where is python install in ur system

Go to the window search --->> type idle--> open the idle-->click on new file-->click three don’t on the right side n click on save as option ---> copy that path n paste it on notepad

Copy the path n open the file explorer n paste it press enter-->> open script n copy that path

Open window search ->>type edit system environment variables--> click on environment variables-->click on user variable-->click on path-->select edit-->click on new-->copy the path n press enter

Repeat the same for system variable

Still if u don’t get--> go to cmd n type pip install virtualenv

Type the virtualenv env in terminal

.\venv\Scripts\activate type the same in terminal-->> if u don’t get --> ctrl + click the link n copy that n open that in the google

Under the powershell -->Set-ExecutionPolicy -ExecutionPolicy RemoteSigned copy this command--> open power shell--> copy the command n press enter-->type Y n enter

Open to visual studio code n type .\venv\Scripts\activate now it gets activated

Go to the visual studio lick on the pythonweb create a new app.py folder

Create dockerfile-->create requirements.txt -->get all the resources from github

docker build-t pythonflask : latest .

Type the above comand in terminal n enter

Open the docker desktop--->click on pythonflask n in optional setting select 5000 as port n run it

 docker tag pythonflask:latest

In cmd login into the docker "docker login"

Docker push raghuprasadkonandur/pythonflask:latest

Docker pull raghuprasadkonandur/pythonflask:latest

Open the aws sign into the account

Click on ec2-->create an instance-->launch the instance--->click on connect-->>  sudo apt update sudo apt install nginx  n link on th address n add http followed by the ip address

**Cloud computing day:-7**

Create a new instance In ec2

Install the docker on ec2 instance

Pulling docker image from the docker hub

Running the image

With instance created on day 6 -  pythonflask

Connect and execute the commands given

Execute till-->>  [Without executing -->>sudo systemctl stop docker]

sudo docker run -p 80:5000 raghuprasadkonandur/pythonflask:latest

Then copy the public ip from below the terminal and paste it in a new tab in google

U can now access the file  using the public ip--> will get a

After using the above command

sudo docker run -p 80:5000 raghuprasadkonandur/pythonflask:latest

Give this command n after this get the container list by using

sudo docker container list

 use this command n get the list of containers

To stop the containers using the follwing the command

syntax - sudo docker stop containerID\_or\_containerName

Steps :

1. create pythonwebapp folder

2. cd pythonwebapp

3. code .

4. open terminal

5. virtualenv venv

6. activate virtual env

.\venv\Scripts\activate

7. install flask

pip install flask

8. create templates folder inside pythonwebapp

Sign up

<form>

name: text box

email : text box

mobile : text box

password : password

button - Sign Up

</form>

jinga templating engine

**Cloud computing - day 8**

End to end project

Go to his git à cloudà dockeràdocker mysqlàprojexct.txtàcopy the link and paste the link in chrome and link on the link-  <https://github.com/makandas/composetutorial>

It will take u to a repository àclone all the files and copy it to a new folder in cloudbca(composetutorial)à go to terminal

Open docker appà delete all containers

Docker compose.ymlàopen terminalàDocker-compose upàgo to localhost:5000\show\_all

For practiceàModify Only app.py file, show\_all.html, update\_addr.html, newaddress.html and try

**Cloud computing-- day 9**

Open the amazon awsà search for rds

after opening the rdsà> click on create databaseà click on easy createà click on mysql under dbs instance size select free tier

Give a name to db instance identifierà keep ,master username  as admin

Select self managed and enter a password in master password

And click on create database

This is PAAS

2]Download and install MySql workbench

Database client -  MySql workbench

We should create workbench and enable it

Click on sindhivapDB à click on view connection details

at connectivity and security à copy the endpoint

click on modify à scroll downà click on additional configuration right after connectivity à under  public access à select publicly access and click continue

open my sql workbench app

now connect ur database in Mumbai center to ur workbench

in the workbench appà click on MySql connection+ à under connection name – sindhivapDBà hostname as endpointàkeep the username as admin and enter the password- admin123 à test connection

if there is an issue à go to rdsàdb instancesà click on ur instanceà click on VPC security groups link àselect the security group IDàunder inbound rulesà edit inbound rulesàadd rulesàchange 0 to 3306à change custom to anywhere IPv4à and click on save rules

also check if ur instance is publicly accessed or not

now test the connection

3]open the workbench app and click on ur database and the create a schema

Give a db name and click on apply and apply

On the left column under urd bà tablesàcreate tableà give a table name àgive column namen product idàtick the check boxes – PK, NN, AIàanother column product name àTICK AIà clicl applyàapplyà finish

Now go to query à type---- select  \* from product:

Enter values and rows by giving the queries and run to get outputs

>