- **Docker**
  - Content
    1. Docker Architecture
    2. Docker Volumes
    3. Docker Image
    4. Docker Networks
    5. Docker Compose

  - Duration: 3 days
  - Tasks: https://github.com/salmarefaie/Docker-labs

- **kubernetes**
  - Content
    1. Deployment
    2. Replicaset
    3. Namespace
    4. Kumtomuzation
    5. Ingrees
    6. Service
    7. Service account
    8. Cronjob
    9. Role
    10. Configmap
    11. Secret
    12. Stateful set
    13. Deamon set
    14. Volume
    15. Taint and tolerations

  - Duration: 5 days

- **Terraform**
  - Content
    1. Terraform basics
    2. Terraform commands
    3. Terraform state
    4. Terraform remote state & state lock

5. Terraform Provisioner
6. Terraform import
7. Terraform modules
8. Terraform workspaces
9. Datasources
10. For each & count
11. Life cycle rules

- Duration: 3 days
- Tasks: https://github.com/salmarefaie/terraform

- **Ansible**
  - Content
    1. Ansible inventory
    2. Ansible playbook
    3. Ansible modules
    4. Ansible variables
    5. Ansible conditions
    6. Ansible loops
    7. Ansible roles
    8. Ansible galaxy

  - Duration: 2 days
  - Tasks:
    1. Deploy django app using ansible on machine and the other install apache , change port, custom page
       https://github.com/salmarefaie/ansible-labs

    2. Build infrastructure using terraform & Install nexus and sonarqube using ansible
       https://github.com/salmarefaie/Ansible-Terraform-Task

- **Jenkins**
  - Duration: 2 days
  - Tasks: https://github.com/salmarefaie/jenkins-labs
    1- install jenkins with docker image
    2- install role based authorization plugin
    3- create new user

4- create read role and assign it to the new user
5- create free style pipeline and link it to private git repo(inside it create directory and create file with "hello world")

##################
1- create declarative in jenkins GUI pipeline for your own repo to do "ls"
2- create scripted in jenkins GUI pipeline for your own repo to do "ls"
3- create the same with jenkinsfile in your branches as multibranch pipeline
4- configure jenkins image to run docker commands on your hos docker daemon
5- create CI/CD for this repo https://github.com/mahmoud254/jenkins_nodejs_example.git
##################
1- create docker file to build image for jenkins slave
2- create container from this image and configure ssh
3 from jenkins maste create new node with the slave container
4- integrate slack with jenkins
5- send slack message when stage in your pipeline is successful
6- install audit logs plugin and test it
7- fork the following repo https://github.com/mahmoud254/Booster_CI_CD_Project and add dockerfile to run this django app and use github actions to build the docker image and push it to your dockerhub

- **Apache and nginx**
    - Duration: 2 days

- **Helm**
    - Duration: 2 days

- **AWS**
    - Content
        1. Iam

2. Aws network (vpc - SG - subnets - nacl - nat - internet gateway - routing)
3. Ec2 (types - ami - logs - ssh - volumes)
4. Bastion host
5. Load balancer
6. Auto scaling
7. S3
8. Lambda
9. Dynamo db
10. ECS
11. EKS

- Duration: 5 days

- **GCP**
  - Content
    1. Resources Hierarchy
    2. APIs & Billing
    3. Google Cloud SDK
    4. Identity and Access management (IAM)
    5. Networking (vpc - subnets - nat - firewall - load balancing - iap - routing)
    6. GKE
    7. Google Compute Engine (VMs)
    8. Instance groups
    9. Storage and databases
    10. GCR

  - Duration: 5 days
  - Project: https://github.com/salmarefaie/final-task-gcp

    It is a simple infrastructure which have 1 vpc and 2 subnets. First subnet is called management subnet which contains private vm and nat gateway. Second subnet is called restricted subnet which contains private standard GKE cluster and private control plane. Only management subnet can connect to the cluster. Service accounts are created and attached to vm and cluster with least privillage. Python application with redis are deployed and their images are pushed to GCR. Deployment is exposed to public internet with public http load balancer.

- **Admin1**
  - Content
    1. Linux components
    2. uname - date - cal
    3. File system
    4. Changing Directories
    5. Listing Directory Contents
    6. File Naming
    7. Viewing File Content
    8. Metacharacters
    9. File & Dir. Manipulation
    10. Man
    11. Grup
    12. Vi
    13. Gedit
    14. Environment Variables
    15. Alias
    16. Processes
    17. Shell jobs: kill - sleep ….
    18. Searching
    19. Redirection
    20. Pipeline
    21. Tee
    22. String processing: wc - diff - grep - cut- sort - tr
    23. Hard link & soft link
    24. Archiving
    25. Compression
    26. Yum
    27. find

  - Duration: 5 days

- **Bash script**
  - Content
    1. Awk
    2. Sed
    3. Conditional expressions
    4. Relational Operators

5. Loops
6. Variables
7. Reading User Input
8. logical operations
9. Select
10. Case
11. Array
12. Functions

- Duration: 3 days
- Project: DBMS using bash script
  https://github.com/salmarefaie/DBMS_BashScript

● Final project

- Create a simple infrastructure on aws to make secure cluster eks using terraform.
- Deploy and Configure Jenkins on EKS.
- Deploy python app application on EKS cluster using CI/CD jenkins pipeline.
  https://github.com/salmarefaie/ITI-Final-Task-DevOps