```
demo-2 Variables (need to add security group manually)
  terraform apply -vars AWS_REGION=us-west2
  git clone https://github.com/maheshkharwadkar/terraform-course.git
  cd terraform-course
  git checkout terraform-0.12
  cd demo-2
  terraform init
  vi terraform.tfvars
  ssh-keygen -f mykey
  vi vars.tf
  vi script.sh
  terrafrom plan
  terraform apply

https://docs.aws.amazon.com/cli/latest/userguide/install-linux.html
demo-3 Remote-State
  dnf install python3-pip
  vi ~/.bash_profile
  export PATH=~/.local/bin:$PATH
  source ~/.bash_profile
  pip3 install awscli --upgrade --user
  aws --version
  aws configure

cat ~/.aws/credentials



demo-18-interpolation
  terraform init
  ssh-keygen -f mykey
  terraform apply
  terraform apply -var ENV=dev

demo-built-in-functions
terraform console
  replace("hello this is a string", "e", "!")
  "the server launched at ${timestamp()}"
  list("subnet-1", "subnet-2", "subnet-3")
  list(",", "subnet-1, subnet-2, subnet-3")
  element(list("subnet-1", "subnet-2", "subnet-3"), 0)
  element(list("subnet-1", "subnet-2", "subnet-3"), 1)
  element(list("subnet-1", "subnet-2", "subnet-3"), 2)
  slice(list("subnet-1", "subnet-2", "subnet-3"), 0, 2)
  join(",", slice(list("subnet-1", "subnet-2", "subnet-3"), 0, 2))
  map("eu-west-1", "ami-1", "us-east-1", "ami-2")
  lookup(map("eu-west-1", "ami-1", "us-east-1", "ami-2"), "us-east-1")

demo-18b-Project-Structure-demo
  cd dev
  terraform init
  ssh-keygen -f mykey
  terrafrom apply

ECR/Demo
  Redhat Terraform Machine
```

```
  cd docker-demo-1 (Create ECR Repository)
    terraform init
        terraform apply

  Ubuntu (docker Machine)
    git clone https://github.com/wardviaene/docker-demo.git
    cd docker-demo
        apt install python3-pip
        apt install awscli
        aws --version
        aws configure
        docker build -t my_ecr_repo:1 .
        `aws ecr get-login`[Not working]
        aws ecr get-login --region region --no-include-email
        docker login -u AWS -p password
https://aws_account_id.dkr.ecr.us-east-1.amazonaws.com

    docker push 604024828336.dkr.ecr.us-east-2.amazonaws.com/myapp:1

ECS demo-1
  cd docker-demo-2
  copy terraformstate file from ECR demo
  Edit vars.tf
        region
        ECS AMI -
https://docs.aws.amazon.com/AmazonECS/latest/developerguide/ecs-
optimized_AMI.html
  edit vpc.tf
    subnets changes az as per region
  terraform init
  ssh-keygen -f mykey
  terraform apply
  ssh ec2 instance (ec2-user_)
    cat /etc/ecs/ecs.config
        ps aux | grep agent
  docker ps
  curl localhost:3000
  los dir /var/logs/ecs/*



EKS Demo

edit eks-workers.tf
  line no. 29 (t2.medium)

curl -LO https://storage.googleapis.com/kubernetes-
release/release/$(curl -s https://storage.googleapis.com/kubernetes-
release/release/stable.txt)/bin/linux/amd64/kubectl
chmod +x kubectl
sudo mv kubectl /usr/local/bin

wget https://github.com/kubernetes-sigs/aws-iam-
authenticator/releases/download/v0.3.0/heptio-authenticator-
aws_0.3.0_linux_amd64
chmod +x heptio-authenticator-aws_0.3.0_linux_amd64
sudo mv heptio-authenticator-aws_0.3.0_linux_amd64
```

```
/usr/local/bin/heptio-authenticator-aws



terraform output kubeconfig
  copy kubeconfig

mkdir ~/.kube
vi ~/.kube/config

terraform output config-map-aws-auth

vi config-map-aws-auth.yml
kubectl apply -f config-map-aws-auth.yml

kubectl get nodes
```