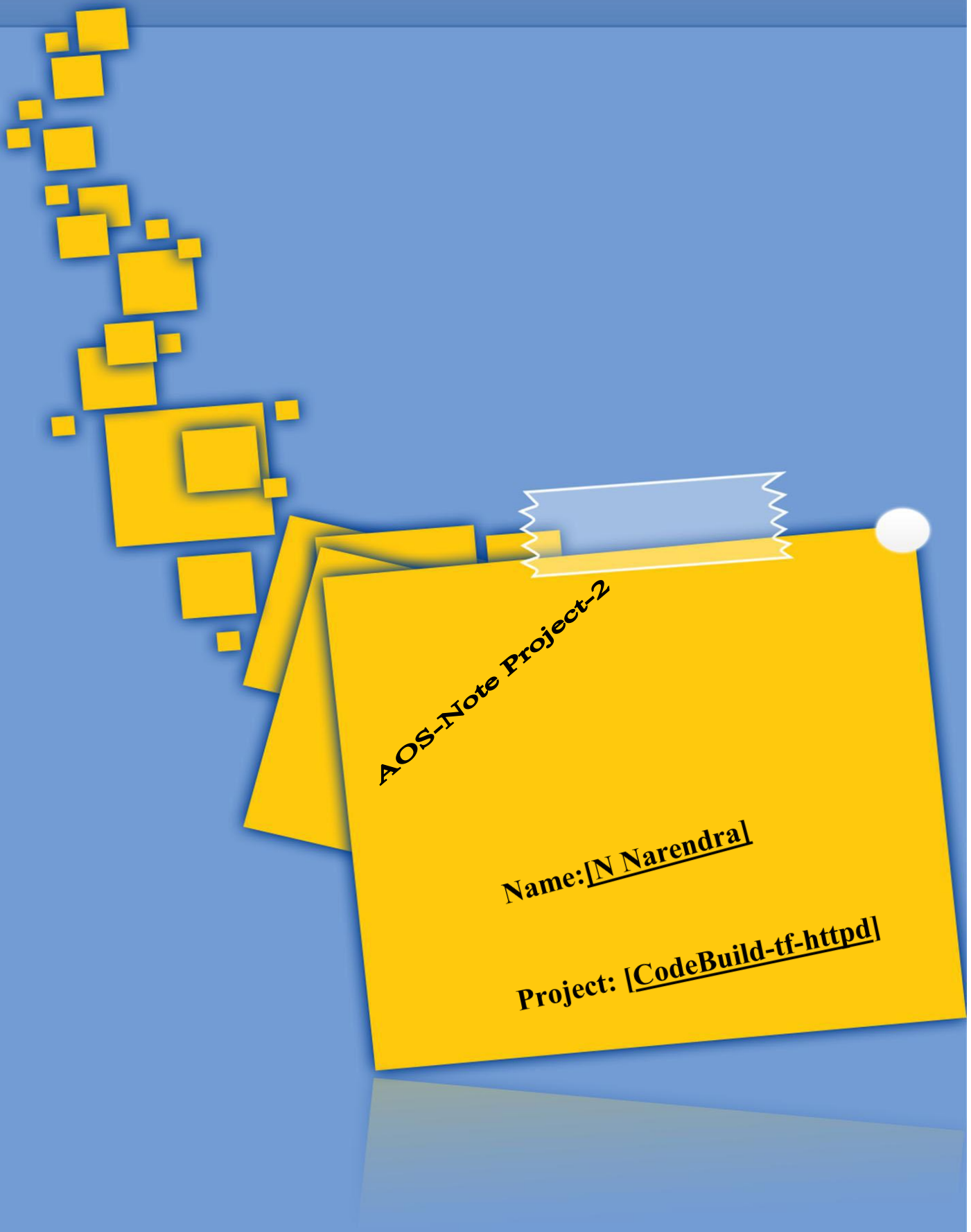
AOS-Note Project-2
Name:[N Narendra]
Project: [CodeBuild-tf-httpd]

Project: **Hosting Website thru Code Build using Terraform & Shell Scripts**
[CodeBuild + Terraform + Shell Scripts (.sh)]
Reference Video: https://youtube.com/playlist?list=PL184oVW5ERMALkQpNuSLMAYpPGLkRYpxN

Step 1: pre-set up in Github to clone Pvt Repos with SSH (thru powershell/cmd)
Step 2: Create IAM user (Administrator Access)
Step 3: Create and Clone Github Repo and add needed Terraform (.tf) files
Step 4: Create S3 Bucket to store .terraform-state file
Step 5: Create Shell Script (.sh) files
Step 6: Create Environment Variable
Step 7: Create buildspec.yml file
Step 8: Create Personal Access Token to access Github repos into CodeBuild
Step 9: Create CodeBuild Project
Step 10: Clean Up all infra by run Terraform Destroy in CodeBuild Project

Task: Thru Terraform

1. ec2.tf:

```
# configured aws provider with proper credentials
provider "aws" {
  region  = "us-east-1"
  profile = "te****"
}

# store the terraform state file in s3
terraform {
  backend "s3" {
    bucket  = "aosnote-terraform-state-bucket"
    key     = "build/terraform.tfstate"
    region  = "us-east-1"
    profile = "ter*****"
  }
}

# create default vpc if one does not exit
resource "aws_default_vpc" "default_vpc" {

  tags = {
    Name = "default vpc"
  }
}

# use data source to get all avalablility zones in region
data "aws_availability_zones" "available_zones" {}

# create default subnet if one does not exit
resource "aws_default_subnet" "default_az1" {
  availability_zone = data.aws_availability_zones.available_zones.names[0]

  tags = {
    Name = "default subnet"
  }
}

# create security group for the ec2 instance
resource "aws_security_group" "ec2_security_group" {
  name        = "ec2 security group"
  description = "allow access on ports 80 and 22"
  vpc_id      = aws_default_vpc.default_vpc.id

  ingress {
    description = "http access"
```

```
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    description = "ssh access"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = -1
    cidr_blocks = ["0.0.0.0/0"]
  }
  tags = {
    Name = "ec2 security group"
  }
}
# use data source to get a registered amazon linux 2 ami
data "aws_ami" "amazon_linux_2" {
  most_recent = true
  owners      = ["amazon"]
  filter {
    name   = "owner-alias"
    values = ["amazon"]
  }
  filter {
    name   = "name"
    values = ["amzn2-ami-hvm*"]
  }
}
# launch the ec2 instance and install website
resource "aws_instance" "ec2_instance" {
  ami                    = data.aws_ami.amazon_linux_2.id
  instance_type          = "t2.micro"
  subnet_id              = aws_default_subnet.default_az1.id
  vpc_security_group_ids = [aws_security_group.ec2_security_group.id]
  key_name               = "nar*****"
  user_data              = file("install_techmax.sh")

  tags = {
    Name = "techmax server"
  }
}
# print the url of the server
output "ec2_public_ipv4_url" {
  value = join("", ["http://", aws_instance.ec2_instance.public_ip])
}
```

2. install-httpd.sh:

```
#!/bin/bash
sudo su
yum update -y
yum install -y httpd
cd /var/www/html
wget https://github.com/azeezsalu/techmax/archive/refs/heads/main.zip
unzip main.zip
cp -r techmax-main/* /var/www/html/
rm -rf techmax-main main.zip
systemctl enable httpd
systemctl start httpd
```

3. **apply-terraform.sh**: **[from 3 to 6 files should be in cicd folder]**

```
#!/bin/bash
set -eu        [# fail on any error]
cd ..          [# go back to the previous directory]
terraform init [# initialize terraform]
terraform apply -auto-approve       [# apply terraform]
# terraform destroy -auto-approve   [# destroy terraform]
```

4. **install-terraform.sh**:

```
#!/bin/bash
set -eu        [# fail on any error]
# install yum-config-manager to manage your repositories
sudo yum install -y yum-utils

# use yum-config-manager to add the official HashiCorp Linux repository
sudo yum-config-manager --add-repo
https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

# install terraform
sudo yum -y install terraform

# verify terraform is installed
terraform --version
```

5. **configure-named-profile.sh**:

```
#!/bin/bash
set -eu        [# fail on any error]
# configure named profile
aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID --profile $PROFILE_NAME
aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY --profile
$PROFILE_NAME
aws configure set region $AWS_REGION --profile $PROFILE_NAME

# verify that profile is configured
aws configure list --profile $PROFILE_NAME
```

6. **buildspec.yml**:

```
version: 0.2

phases:
  install:
    runtime-versions:
      python: 3.x

  pre_build:
    commands:
      - cd cicd # change directory
      - chmod +x install-terraform.sh configure-named-profile.sh apply-terraform.sh
# make files executable
      - ./install-terraform.sh # install terraform
      - ./configure-named-profile.sh # configure named profile

  build:
    commands:
      - ./apply-terraform.sh
```