

CLOUD DEVOPS (EPAM)

- 20CS3019AA 2022-23 EVEN SEMESTER

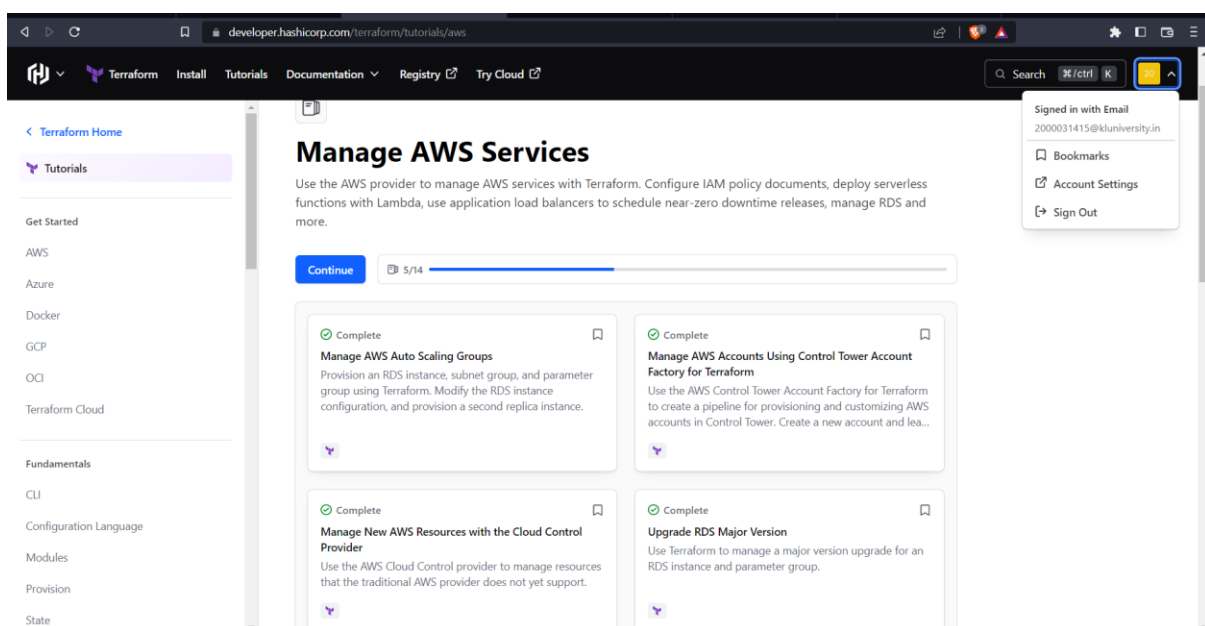
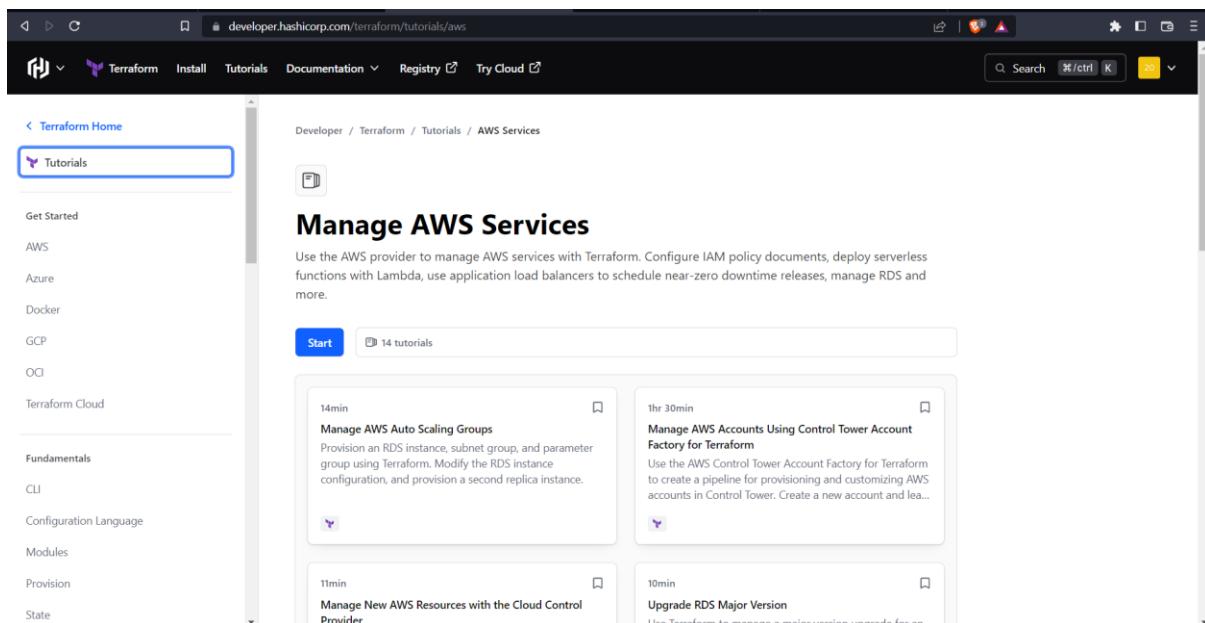
2000031415 Jayanth Krishna Sec: 13

Home assignment – 2

<https://developer.hashicorp.com/terraform/tutorials/aws>

The above link consists of 14 Aws Services.

Every student must go through those all services and submit **terraform** configuration files with at least 4 of the listed services.



Manage AWS Auto Scaling Groups

```
provider "aws" {  
  region = "us-east-2"  
  
  default_tags {  
    tags = {  
      hashicorp-learn = "aws-asg"  
    }  
  }  
}  
  
data "aws_availability_zones" "available" {  
  state = "available"  
}  
  
module "vpc" {  
  source = "terraform-aws-modules/vpc/aws"  
  version = "2.77.0"  
  
  name = "main-vpc"  
  cidr = "10.0.0.0/16"  
  
  azs            = data.aws_availability_zones.available.names  
  public_subnets = ["10.0.4.0/24", "10.0.5.0/24", "10.0.6.0/24"]  
  enable_dns_hostnames = true  
  enable_dns_support   = true  
}  
  
data "aws_ami" "amazon-linux" {  
  most_recent = true  
  owners      = ["amazon"]  
}
```

```
filter {  
  name = "name"  
  values = ["amzn-ami-hvm-*-x86_64-ebs"]  
}  
}
```

```
resource "aws_launch_configuration" "terramino" {  
  name_prefix    = "learn-terraform-aws-asg-"  
  image_id       = data.aws_ami.amazon-linux.id  
  instance_type  = "t2.micro"  
  user_data      = file("user-data.sh")  
  security_groups = [aws_security_group.terramino_instance.id]
```

```
  lifecycle {  
    create_before_destroy = true  
  }  
}
```

```
resource "aws_autoscaling_group" "terramino" {  
  name          = "terramino"  
  min_size      = 1  
  max_size      = 3  
  desired_capacity = 1  
  launch_configuration = aws_launch_configuration.terramino.name  
  vpc_zone_identifier = module.vpc.public_subnets  
  lifecycle {  
    ignore_changes = [desired_capacity, target_group_arns]  
  }  
  tag {  
    key          = "Name"  
    value        = "HashiCorp Learn ASG - Terramino"  
    propagate_at_launch = true  
  }  
}
```

```
resource "aws_lb" "terramino" {  
  name          = "learn-asg-terramino-lb"  
  internal      = false  
  load_balancer_type = "application"  
  security_groups = [aws_security_group.terramino_lb.id]  
  subnets      = module.vpc.public_subnets  
}
```

```
resource "aws_lb_listener" "terramino" {  
  load_balancer_arn = aws_lb.terramino.arn  
  port              = "80"  
  protocol          = "HTTP"  
  
  default_action {  
    type = "forward"  
    target_group_arn = aws_lb_target_group.terramino.arn  
  }  
}
```

```
resource "aws_lb_target_group" "terramino" {  
  name    = "learn-asg-terramino"  
  port    = 80  
  protocol = "HTTP"  
  vpc_id  = module.vpc.vpc_id  
}
```

```
resource "aws_autoscaling_attachment" "terramino" {  
  autoscaling_group_name = aws_autoscaling_group.terramino.id  
  alb_target_group_arn   = aws_lb_target_group.terramino.arn  
}
```

```
resource "aws_security_group" "terramino_instance" {
```

```
name = "learn-asg-terramino-instance"
```

```
ingress {
```

```
    from_port    = 80
```

```
    to_port      = 80
```

```
    protocol     = "tcp"
```

```
    security_groups = [aws_security_group.terramino_lb.id]
```

```
}
```

```
egress {
```

```
    from_port    = 0
```

```
    to_port      = 0
```

```
    protocol     = "-1"
```

```
    security_groups = [aws_security_group.terramino_lb.id]
```

```
}
```

```
vpc_id = module.vpc.vpc_id
```

```
}
```

```
resource "aws_security_group" "terramino_lb" {
```

```
    name = "learn-asg-terramino-lb"
```

```
    ingress {
```

```
        from_port = 80
```

```
        to_port   = 80
```

```
        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"]
```

```
    }
```

```
vpc_id = module.vpc.vpc_id
}
```

Manage AWS Accounts Using Control Tower Account Factory for Terraform

```
module "aft" {
  source = "github.com/aws-ia/terraform-aws-control_tower_account_factory"

  ct_management_account_id  = var.ct_management_account_id
  log_archive_account_id    = var.log_archive_account_id
  audit_account_id          = var.audit_account_id
  aft_management_account_id = var.aft_management_account_id
  ct_home_region            = var.ct_home_region
  tf_backend_secondary_region = var.tf_backend_secondary_region

  vcs_provider          = "github"
  account_request_repo_name = "${var.github_username}/learn-terraform-aft-account-request"
  account_provisioning_customizations_repo_name = "${var.github_username}/learn-terraform-aft-account-provisioning-customizations"
  global_customizations_repo_name = "${var.github_username}/learn-terraform-aft-global-customizations"
  account_customizations_repo_name = "${var.github_username}/learn-terraform-aft-account-customizations"
}
```

Manage New AWS Resources with the Cloud Control Provider

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
    }
  }
}
```

```

    version = "~> 3.0"
  }

  awsgcc = {
    source = "hashicorp/awsgcc"
    version = "~> 0.1.0"
  }
  random = {
    source = "hashicorp/random"
    version = "~> 3.1.0"
  }
}

}

provider "aws" {
  region = var.aws_region
}

resource "aws_kms_key" "terraform" {
  description = "Example key for Cassandra table"
}

resource "random_pet" "keyspace" {
  length = 4
  separator = "_"
}

resource "awsgcc_cassandra_keyspace" "terraform" {
  keyspace_name = random_pet.keyspace.id
}

resource "awsgcc_cassandra_table" "users" {
  keyspace_name = awsgcc_cassandra_keyspace.terraform.keyspace_name
  table_name = "users"

  partition_key_columns = [
    {
      column_name : "id"
      column_type : "int"
    }
  ]
}

```

```

    }
  ]
  regular_columns = [
    {
      column_name : "first_name"
      column_type : "text"
    },
    {
      column_name : "last_name"
      column_type : "text"
    },
    {
      column_name : "email"
      column_type : "text"
    }
  ]

  encryption_specification = {
    encryption_type : "AWS_OWNED_KMS_KEY"
    kms_key_identifier : aws_kms_key.terraform.key_id
  }
}

output "keyspace_name" {
  description = "Name of Cassandra keyspace."
  value      = awscs_cassandra_keyspace.terraform.keyspace_name
}

```

Upgrade RDS Major Version

```

provider "aws" {
  region = var.region

  default_tags {
    tags = {

```



```
    HashiCorpLearnTutorial = "rds-upgrade"
  }
}
}
```

```
data "aws_availability_zones" "available" {}
```

```
module "vpc" {
  source = "terraform-aws-modules/vpc/aws"
  version = "2.77.0"

  name            = "education"
  cidr            = "10.0.0.0/16"
  azs            = data.aws_availability_zones.available.names
  public_subnets = ["10.0.4.0/24", "10.0.5.0/24", "10.0.6.0/24"]
  enable_dns_hostnames = true
  enable_dns_support  = true
}
```

```
resource "random_pet" "name" {
  length = 1
}
```

```
resource "aws_db_subnet_group" "education" {
  name      = "${random_pet.name.id}-education"
  subnet_ids = module.vpc.public_subnets

  tags = {
    Name = "Education"
  }
}
```

```
resource "aws_security_group" "rds" {
  name = "${random_pet.name.id}_education_rds"
```

```
vpc_id = module.vpc.vpc_id
```

```
ingress {  
  from_port = 5432  
  to_port   = 5432  
  protocol  = "tcp"  
  cidr_blocks = ["0.0.0.0/0"]  
}
```

```
egress {  
  from_port = 5432  
  to_port   = 5432  
  protocol  = "tcp"  
  cidr_blocks = ["0.0.0.0/0"]  
}
```

```
tags = {  
  Name = "education_rds"  
}  
}
```

```
resource "aws_db_parameter_group" "education" {  
  name_prefix = "${random_pet.name.id}-education"  
  family      = "postgres13"
```

```
  parameter {  
    name = "log_connections"  
    value = "1"  
  }  
  lifecycle {  
    create_before_destroy = true  
  }  
}
```

```
resource "aws_db_instance" "education" {
```

```
identifier      = "${random_pet.name.id}education"
instance_class  = "db.t3.micro"
allocated_storage = 10
apply_immediately = true
engine          = "postgres"
engine_version  = "13.3"
username        = "edu"
password        = var.db_password
allow_major_version_upgrade = true
db_subnet_group_name    = aws_db_subnet_group.education.name
vpc_security_group_ids  = [aws_security_group.rds.id]
parameter_group_name    = aws_db_parameter_group.education.name
publicly_accessible     = true
skip_final_snapshot     = true
backup_retention_period = 1
}
```