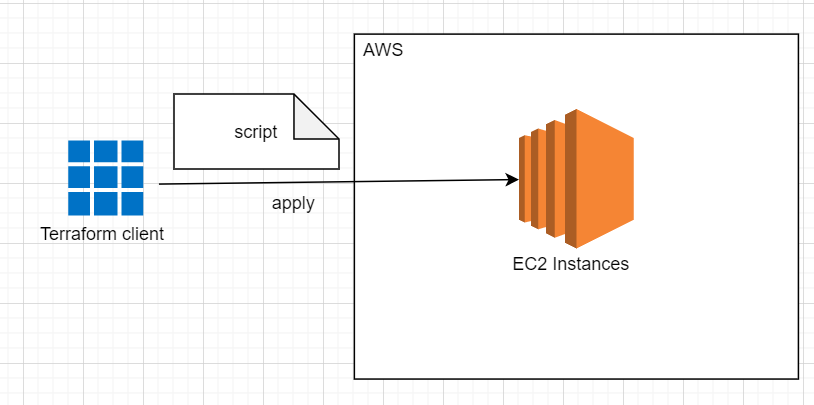
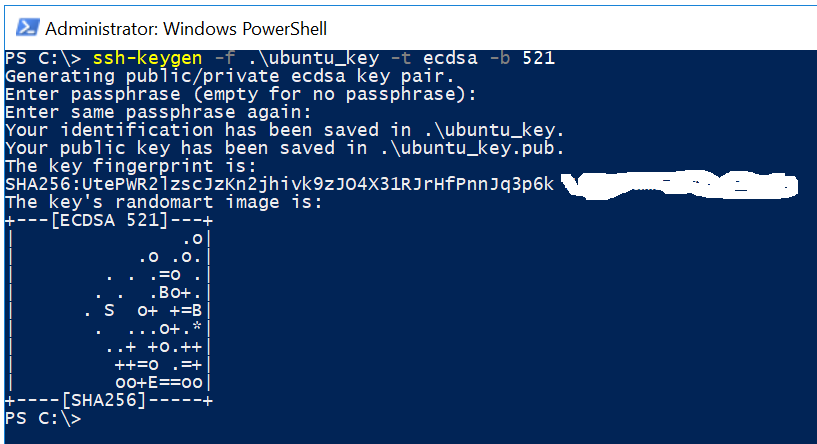
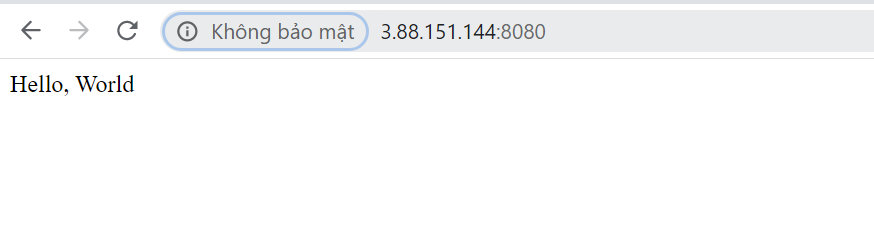
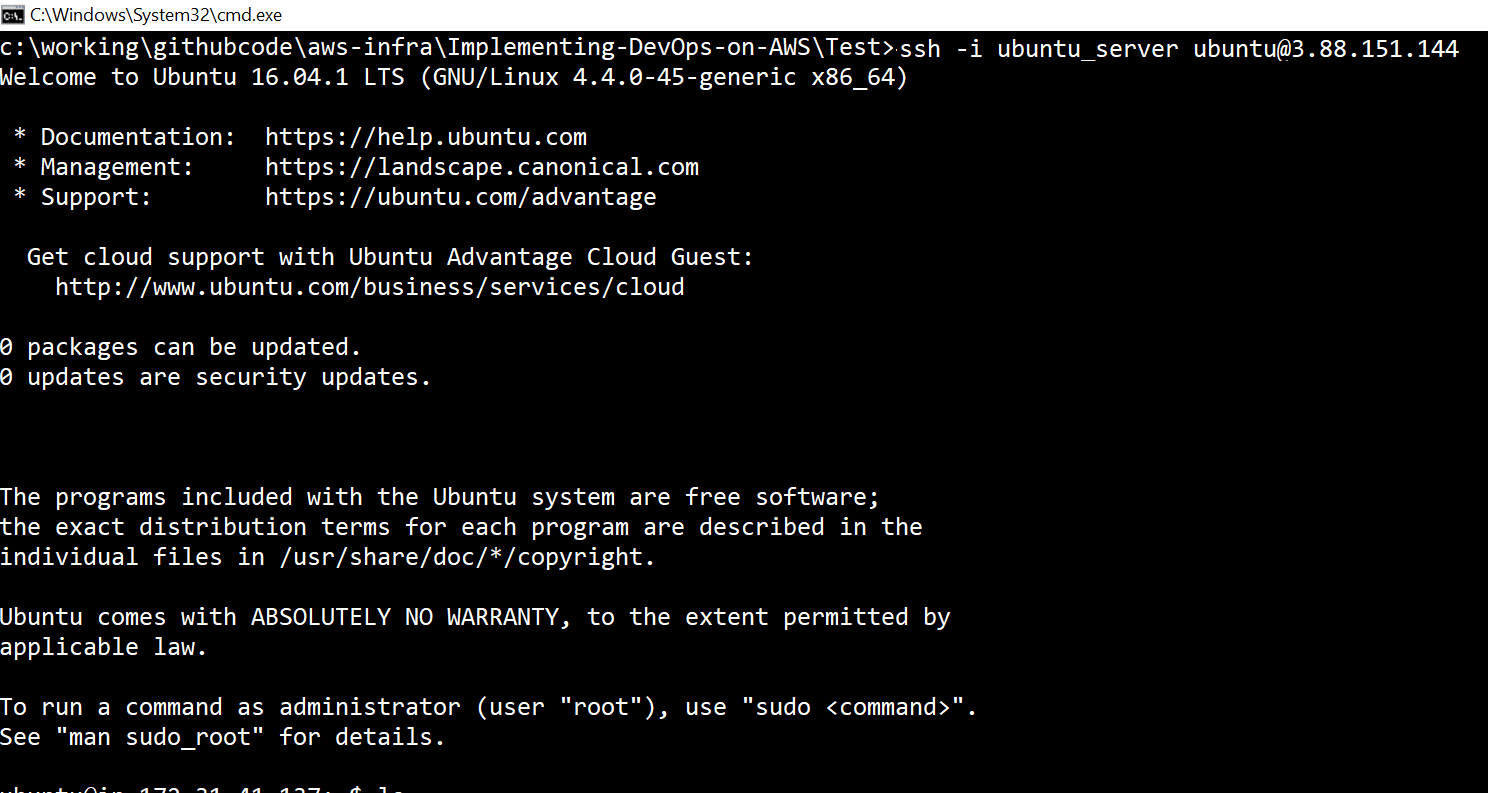
Helloworld Web Application Deployment with EC2 Instance.

1. **Deployment Architecture**  
     
     
     
   The script above includes the following works:  
   1. Authentication with AWS IAM  
   2. Create ec2 ubuntu instances with:  
    - defined region  
    - default VPC  
    - defined Security group  
    - created ssh key  
    - default dynamic public ip  
    - open port 8080  
    - execute init shell script
2. **Set up Infrastructure**  
   1. Create SSH Key pair for the creating EC2. We will use the public key to create the EC2 instance and use private key to connect to it for checking.  
   Open power shell and generate SSH key pair with empty passphrase as below.   
     
   2. In order for Terraform to be able to make changes in your AWS account, you will need to set the AWS credentials for the IAM user you created earlier as the environment variables AWS\_ACCESS\_KEY\_ID and AWS\_SECRET\_ACCESS\_KEY in cmd console:  
   set AWS\_ACCESS\_KEY\_ID=(your access key id)  
   set AWS\_SECRET\_ACCESS\_KEY=(your secret access key)  
   3. Create terraform files:   
   - resource.tf : describe resources are created  
   - variables.tf: declaring variables which are used in resources.tf  
   - terraform.tfvars: set value for variable which are declared in the variables.tf  
   - init.sh: init script when the server starts  
     
   variables.tf:  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   terraform.tfvars:  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   init.sh:  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   4. Run Terraform:  
    - terraform init  
    - terraform plain -out=out  
    - terraform apply “out”  
   5. Check result.  
   - Check web server   
   - Check SSH  
    

variable "aws-region" {

type = string

description = "AWS region"

}

variable "aws-from-port" {

type = number

}

variable "aws-to-port" {

type = number

}

variable "aws-ssh-to-port" {

type = number

}

variable "aws-ssh-to-port" {

type = number

}

variable "aws-public-key" {

type = string

description = "SSH public key"

}

variable "aws-ec2-instance" {

type = string

}

variable "aws-ec2-instance-type" {

type = string

}

aws-region = "us-east-1"

aws-from-port=8080

aws-to-port=8080

aws-ssh-to-port=22

aws-ssh-from-port=22

aws-public-key="ssh-rsa  EAD+mlam21@VNPF0Z4J30"

aws-ec2-instance="ami-40d28157"

aws-ec2-instance-type="t2.micro"

#Aws cloud provider

provider "aws" {

version = "~> 3.0"

region = var.aws-region

}

#open port 8080 for the inbound rule.

#So request from outside can route to the server

resource "aws\_security\_group" "instance" {

name = "terraform-example-instance"

ingress {

from\_port = var.aws-from-port

to\_port = var.aws-to-port

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress {

from\_port = var.aws-ssh-to-port

to\_port = var.aws-ssh-to-port

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"]

}

}

#Declare public ssh key. public\_key value is get from the ubuntu\_key.pub

resource "aws\_key\_pair" "deployer" {

key\_name = "deployer-key"

public\_key = var.aws-public-key

}

#Create a ubuntu (ami = "ami-40d28157")

#and configuration for the server (instance\_type = "t2.micro")

#these value can be checked on AWS

resource "aws\_instance" "helloworld" {

ami = var.aws-ec2-instance

vpc\_security\_group\_ids = ["${aws\_security\_group.instance.id}"]

instance\_type = var.aws-ec2-instance-type

key\_name = aws\_key\_pair.deployer.id

tags = {

Name = "terraform helloworld"

}

#init script is executed when the server starts.

user\_data = file("./init.sh")

}