Terraform tutorial (aws provider doc: https://registry.terraform.io/providers/hashicorp/aws/latest/docs)

1. Install **AWS CLI version 2** from here: https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html
2. Confirm successful installation with: *aws --version*
3. Install **Terraform** from here (more information here: https://developer.hashicorp.com/terraform/tutorials/aws-get-started/aws-build): https://developer.hashicorp.com/terraform/downloads?ajs_aid=01c33e34-bc03-49dc-b85d-8d6460253a82&product_intent=terraform
4. Confirm successful installation: *terraform –v*
5. Select IAM from AWS console
6. Select User from AWS console
7. Select create access key and save it into a file.
8. Configure aws access method with this command from bash:

```
aws configure
AWS Access Key ID [None]: ********
AWS Secret Access Key [None]: *******
```

9. Create a new directory: *mkdir terra_files*
10. Enter into new directory: *cd terra_files*
11. Open Visual Studio Code and create terraform file: code *terra_main.tf* and *variables.tf*.
12. Create key-pair: *ssh-keygen.exe -t rsa -b 4096*
13. Run the command: *terraform init*
14. Run the command: *terraform plan*
15. Run the command: terraform apply
16. Access via SSH with:
    ssh -i .\terra_key_03 ec2-user@34.242.254.130

```
# terra_main.tf

terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.16"
    }
  }

  required_version = ">= 1.2.0"
}

provider "aws" {
  region  = "eu-west-1"
}

resource "aws_security_group" "sg" {
  description = "test sg for terraform"
  vpc_id      = "vpc-0282dfd7758da30e5"
  dynamic "ingress" {
    for_each = var.security_groups
    content {
      description = ingress.value["name"]
      from_port   = ingress.value["from_port"]
      to_port     = ingress.value["to_port"]
      protocol    = ingress.value["protocol"]
      cidr_blocks = ingress.value["cidr_blocks"]
    }
  }
  egress {
    from_port        = 0
    to_port          = 0
    protocol         = "-1"
    cidr_blocks      = ["0.0.0.0/0"]
    ipv6_cidr_blocks = ["::/0"]
  }
}

# Generated with command: ssh-keygen.exe -t rsa -b 4096
resource "aws_key_pair" "deployer" {
  key_name   = "terra_key"
  public_key = file("./terra_key_03.pub")
```

```
}

resource "aws_instance" "app_server" {
  ami = var.ec2.os_type == "linux" ? var.linux_ami : var.ubuntu_ami
  instance_type = var.ec2.instance_type
  associate_public_ip_address = true
  vpc_security_group_ids      = [aws_security_group.sg.id]
  #key_name = "terra_key"
  key_name = aws_key_pair.deployer.id
  root_block_device {
    delete_on_termination = true
    encrypted             = false
    volume_size           = var.ec2.volume_size
    volume_type           = var.ec2.volume_type
  }

  tags = {
    Name = var.instance_name
  }
}
```

```
#variables.tf

variable "instance_name" {
  description = "Value of the Name tag for the EC2 instance"
  type        = string
  default     = "GS_instance"
}

variable "ubuntu_ami" {
  description = "ubuntu ami"
  type        = string
  default     = "ami-04ff9e9b51c1f62ca"
}
variable "linux_ami" {
  description = "linux ami"
  type        = string
  default     = "ami-04f7efe62f419d9f5"
}

variable "ec2" {
  description = "The attribute of EC2 information"
```

```
  type = object({
    name              = string
    os_type           = string
    instance_type     = string
    volume_size       = number
    volume_type       = string
    availability_zone = string
  })
  default = {
    instance_type     = "t2.micro"
    name              = "ppshein"
    os_type           = "linux"
    volume_size       = 20
    volume_type       = "gp3"
    availability_zone = "eu-west-1"
  }
}

variable "security_groups" {
  description = "The attribute of security_groups information"
  type = list(object({
    name        = string
    from_port   = number
    to_port     = number
    protocol    = string
    cidr_blocks = list(string)
  }))
  default = [
        {
            from_port   = 22
            name        = "Office Wifi CIDR Range"
            protocol    = "tcp"
            to_port     = 22
            cidr_blocks = ["0.0.0.0/0"] # you can replace with your
office wifi outbount IP range
        },
        {
            from_port   = 8086
            name        = "InfluxDB"
            protocol    = "tcp"
            to_port     = 8086
            cidr_blocks = ["0.0.0.0/0"]
        }]
```

```
}
```

Docker installation in AWS Linux AMI

1. Accedere via SSH: ssh -i .\terra_key.pem ec2-user@3.252.232.176
2. Install Docker with the following command:

```
sudo yum update -y

sudo yum -y install docker

sudo service docker start

sudo usermod -a -G docker ec2-user

sudo systemctl enable docker

sudo docker version
```

1. Install InfluxDB container:

```
$ docker run -p 8086:8086 -v myInfluxVolume:/var/lib/influxdb2 influxdb:latest
```