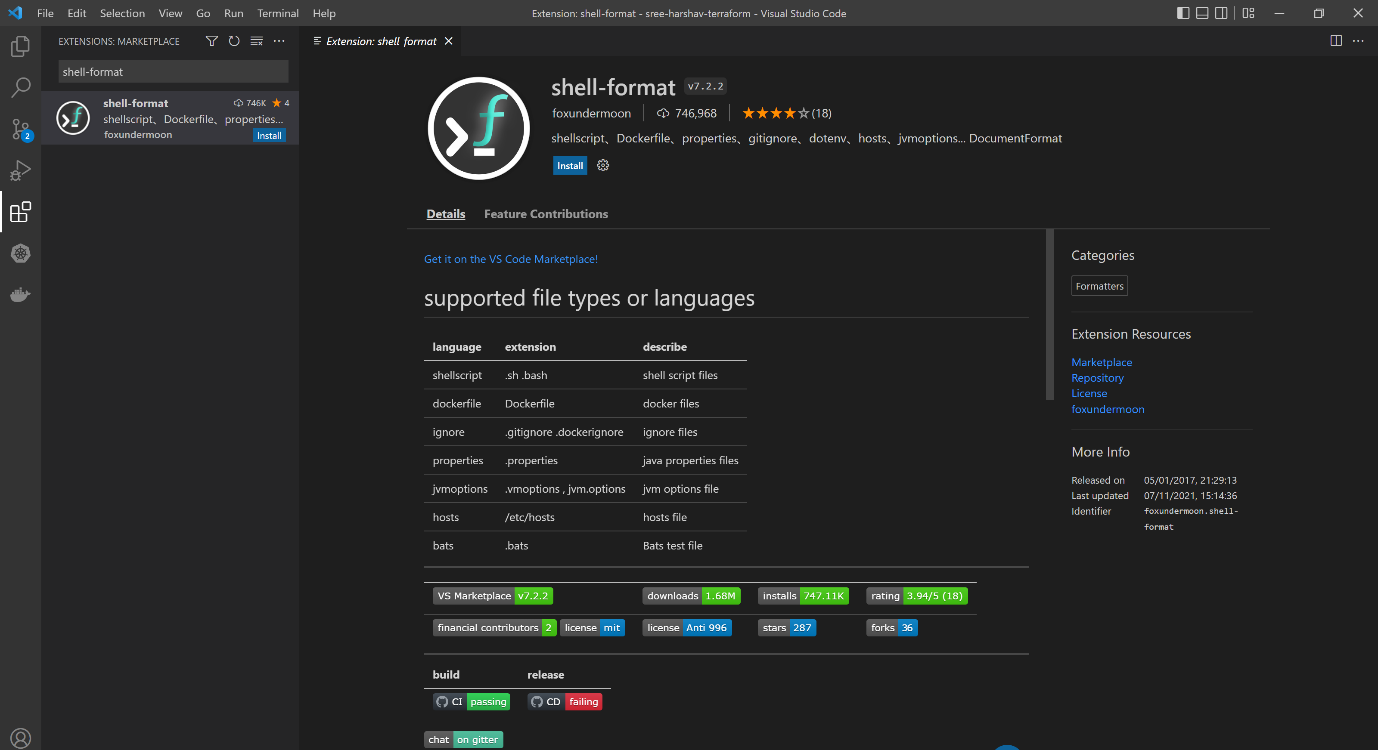
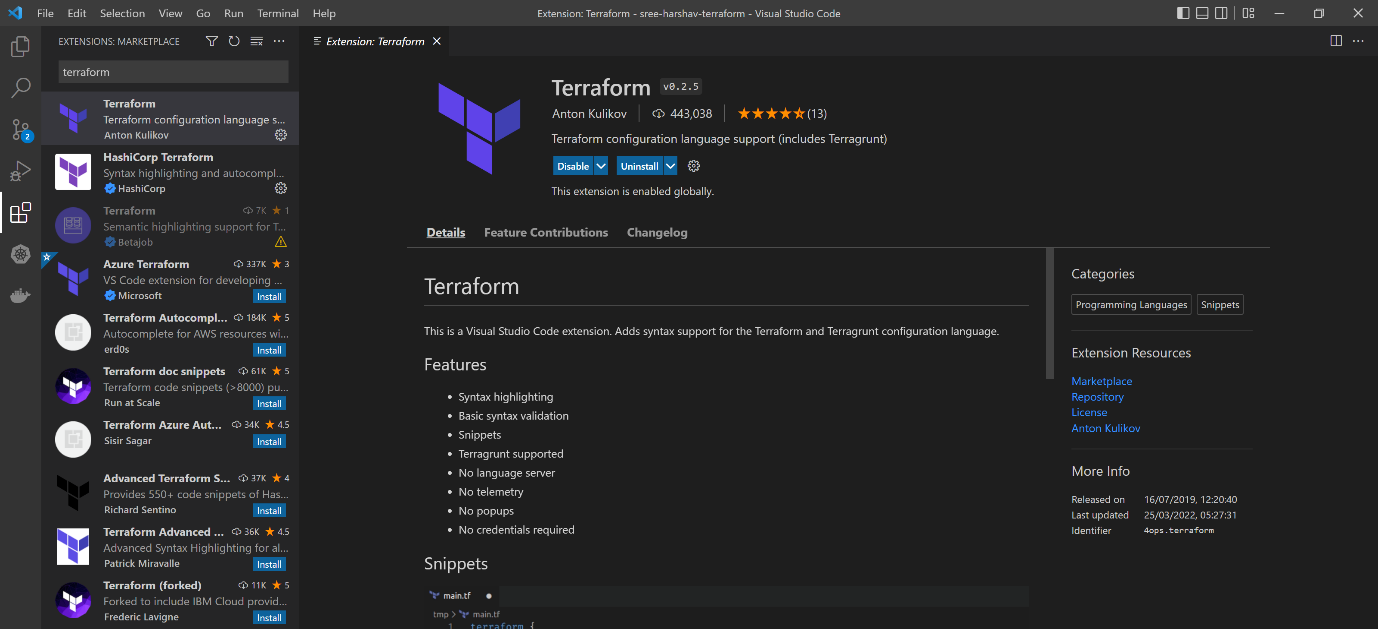
**15.DevOps-B24-Terraform-Part-1**

**Prerequisites**

--- download shell-format plugin.



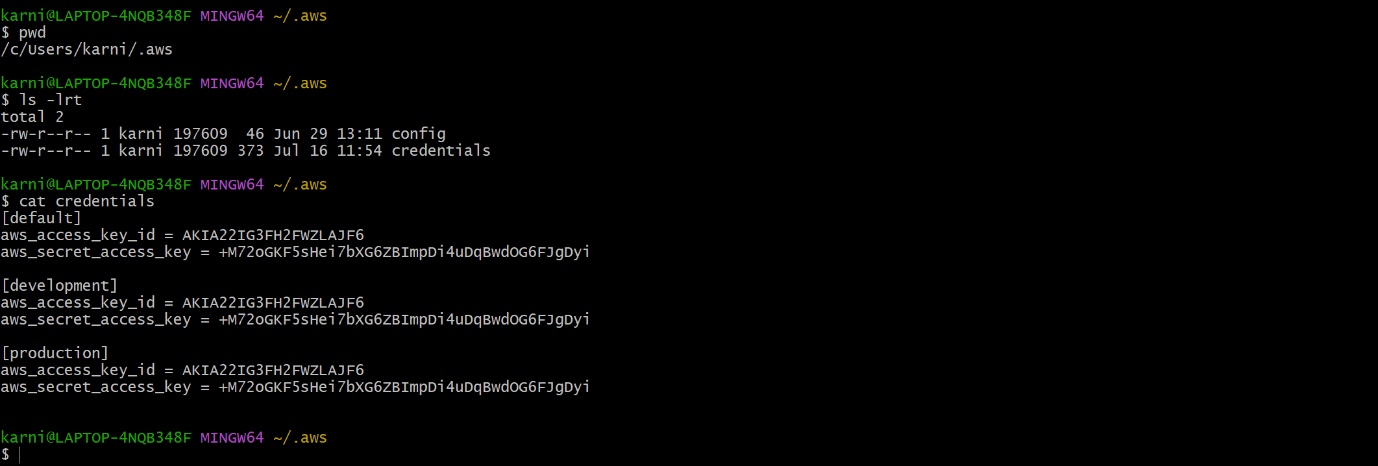
--- download terraform Anton Kulikov.



**Configure 2 aws accounts for terraform**

--- Profile Reference - <https://registry.terraform.io/providers/hashicorp/aws/latest/docs>

--- after executing aws cli on windows machine, you want to manage multiple aws accounts with terraform.



--- **note** – here I mention default, development, and production profile. These profile I will use in different terraform projects

**Provider block**

--- Profile Reference - <https://registry.terraform.io/providers/hashicorp/aws/latest/docs>

provider "aws" {

  region= "us-east-1"

  profile = "production"

}

--- **note** – now the terraform will use production credentials.

**aws\_vpc resource**

--- Reference - <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/vpc>

resource "aws\_vpc" "vpc1" {

  cidr\_block           = "10.0.0.0/16"

  enable\_dns\_hostnames = true

  tags = {

    Name = "vpc1"

    Env  = "production"

  }

}

resource "aws\_vpc" "vpc2" {

  cidr\_block           = "10.0.0.0/16"

  enable\_dns\_hostnames = true

  tags = {

    Name = "vpc2"

    Env  = "production"

  }

}

resource "aws\_vpc" "vpc3" {

  cidr\_block           = "10.0.0.0/16"

  enable\_dns\_hostnames = true

  tags = {

    Name = "vpc3"

    Env  = "production"

  }

}

--- **note** – if you see resource anywhere in the terraform then it means is that we are telling the terraform to create resource.

--- **note** – terraform identifies the resource by its name, here it is identifying the vpc by its name.

**aws\_vpc resource output**

output "vpc\_id1" {

    value = "${aws\_vpc.vpc1.id}"

}

output "vpc\_arn1" {

    value = "${aws\_vpc.vpc1.arn}"

}

output "vpc\_id2" {

    value = "${aws\_vpc.vpc2.id}"

}

output "vpc\_arn2" {

    value = "${aws\_vpc.vpc2.arn}"

}

output "vpc\_id3" {

    value = "${aws\_vpc.vpc3.id}"

}

output "vpc\_arn3" {

    value = "${aws\_vpc.vpc3.arn}"

}

**# Correct the indentation of terraform .tf files**

--- terraform fmt

**# Check errors in syntax**

--- terraform validate

**# Dry run, it will not execute but it will tell you what happened when you execute the terraform.**

--- terraform validate

**# Execute terraform**

--- terraform apply -auto-approve

**# Destroy infrastructure**

--- terraform destroy -auto-approve

**# List the terraform resources**

--- terraform state list

**# Destroy a particular target**

--- terraform destroy -target aws\_vpc.vpc3 -auto-approve

--- note – when we say terraform state list then it will list out all the resources created by terraform so, we will list the resources names while destroying the particular target.