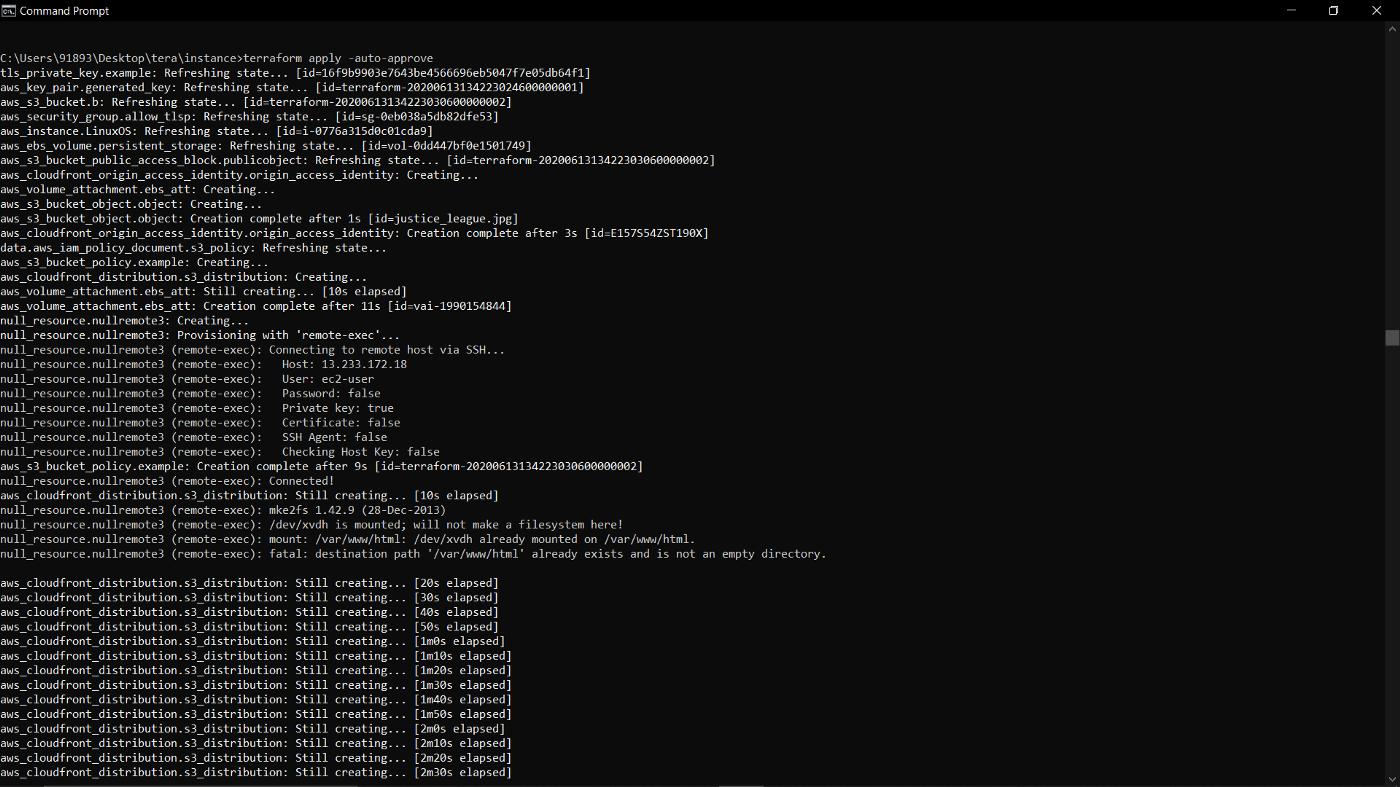
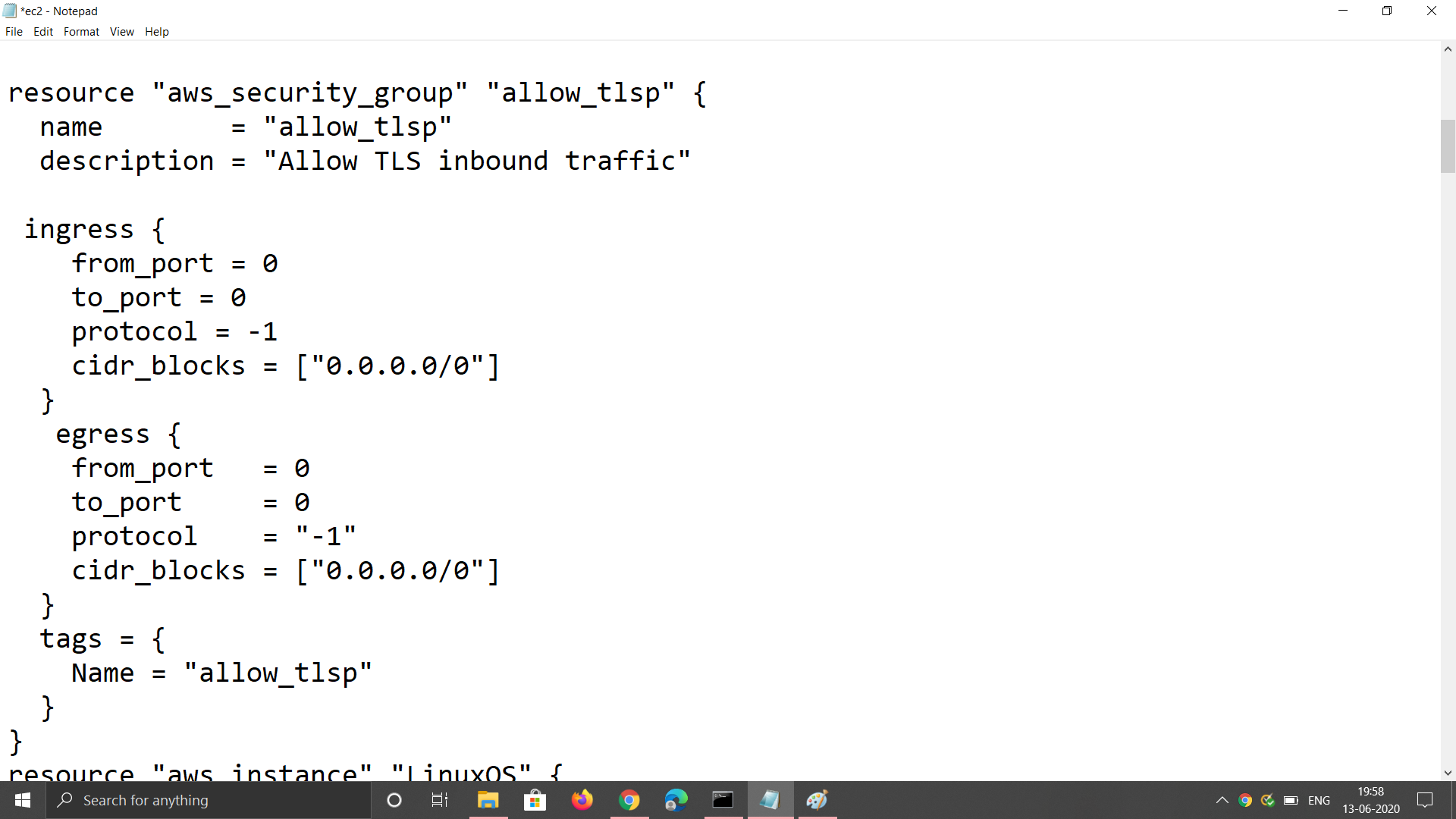
|  |
| --- |
| provider "aws" { |
|  | region = "ap-south-1" |
|  | profile = "aman" |
|  | } |
|  |  |
|  | #create security group |
|  |  |
|  | resource "aws\_security\_group" "allow\_tls" { |
|  | name = "launch-wizard-8" |
|  | description = "Allow http and ssh" |
|  | //vpc\_id = "${aws\_vpc.main.id}" |
|  |  |
|  | ingress { |
|  | description = "SSH Port" |
|  | from\_port = 22 |
|  | to\_port = 22 |
|  | protocol = "tcp" |
|  | cidr\_blocks = ["0.0.0.0/0"] |
|  | } |
|  | ingress { |
|  | description = "HTTPD port" |
|  | from\_port = 80 |
|  | to\_port = 80 |
|  | protocol = "tcp" |
|  | cidr\_blocks = ["0.0.0.0/0"] |
|  | } |
|  | ingress { |
|  | description = "local host" |
|  | from\_port = 8080 |
|  | to\_port = 8080 |
|  | protocol = "tcp" |
|  | cidr\_blocks = ["0.0.0.0/0"] |
|  | } |
|  | ingress { |
|  | description = "Custom tcp" |
|  | from\_port = 81 |
|  | to\_port = 81 |
|  | 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 = "allow\_tls" |
|  | } |
|  | } |
|  |  |
|  | # launch ec2 instance |
|  |  |
|  | resource "aws\_instance" "web" { |
|  | ami = "ami-0447a12f28fddb066" |
|  | instance\_type = "t2.micro" |
|  | key\_name = "mykey" |
|  | security\_groups = [ "launch-wizard-2" ] |
|  |  |
|  | connection { |
|  | type = "ssh" |
|  | user = "ec2-user" |
|  | private\_key = file("C:/Users/user/Downloads/mykey.pem") |
|  | host = aws\_instance.web.public\_ip |
|  | } |
|  |  |
|  | provisioner "remote-exec" { |
|  | inline = [ |
|  | "sudo yum install httpd php git -y", |
|  | "sudo systemctl restart httpd", |
|  | "sudo systemctl enable httpd" |
|  | ] |
|  | } |
|  |  |
|  | tags = { |
|  | Name = "linuxOS" |
|  | } |
|  |  |
|  | } |
|  |  |
|  | #create ebs volume |
|  |  |
|  | resource "aws\_ebs\_volume" "ebs\_volume1" { |
|  | availability\_zone = aws\_instance.web.availability\_zone |
|  | size = 1 |
|  |  |
|  | tags = { |
|  | Name = "linux\_volume\_1" |
|  | } |
|  | } |
|  |  |
|  | # attach ebs volume |
|  |  |
|  | resource "aws\_volume\_attachment" "ebs\_att" { |
|  | device\_name = "/dev/sdh" |
|  | volume\_id = aws\_ebs\_volume.ebs\_volume1.id |
|  | instance\_id = aws\_instance.web.id |
|  | force\_detach = true |
|  | } |
|  |  |
|  |  |
|  | # mount harddisk |
|  |  |
|  | resource "null\_resource" "harddisk" { |
|  | depends\_on = [ |
|  | aws\_volume\_attachment.ebs\_att, |
|  | ] |
|  |  |
|  | connection { |
|  | type = "ssh" |
|  | user = "ec2-user" |
|  | private\_key = file("C:/Users/user/Downloads/mykey.pem") |
|  | host = aws\_instance.web.public\_ip |
|  | } |
|  |  |
|  | provisioner "remote-exec" { |
|  | inline = [ |
|  | "sudo mkfs.ext4 /dev/xvdh", |
|  | "sudo mount /dev/xvdh /var/www/html", |
|  | "sudo rm -rf /var/www/html/\*", |
|  | "sudo git clone https://github.com/square/square.github.io.git /var/www/html/" |
|  | ] |
|  | } |
|  |  |
|  | } |
|  |  |
|  | #create a s3 bucket |
|  | resource "aws\_s3\_bucket" "bucket" { |
|  | depends\_on = [ |
|  | null\_resource.harddisk, |
|  | ] |
|  | bucket = "my-tf-test-bucket-myweb-2" |
|  | //acl = "private" |
|  | acl ="public-read" |
|  | tags = { |
|  | Name = "web\_bucket" |
|  | } |
|  | force\_destroy = true |
|  | //resource "aws\_s3\_bucket\_object" "s3object"{ |
|  | // bucket="my-tf-test-bucket-myweb-2" |
|  | //key="The\_Adventure\_Of\_Sherlock\_Holmes.png" |
|  | //source="C:/Users/user/Desktop/The\_Adventure\_Of\_Sherlock\_Holmes.png" |
|  | //} |
|  |  |
|  | provisioner "local-exec" { |
|  |  |
|  | command = "aws s3 sync C:/Users/user/Desktop/new s3://my-tf-test-bucket-myweb-2 --acl public-read --profile aman " |
|  |  |
|  | } |
|  |  |
|  | } |
|  |  |
|  |  |
|  | #creating CloudFront |
|  |  |
|  | locals { |
|  | s3\_origin\_id = "myS3Origin" |
|  | } |
|  |  |
|  | resource "aws\_cloudfront\_distribution" "s3\_distribution" { |
|  | origin { |
|  | domain\_name = aws\_s3\_bucket.bucket.bucket\_regional\_domain\_name |
|  | origin\_id = local.s3\_origin\_id |
|  | } |
|  |  |
|  | enabled = true |
|  | is\_ipv6\_enabled = true |
|  | comment = "Some comment" |
|  | default\_root\_object = "index.html" |
|  |  |
|  | default\_cache\_behavior { |
|  | allowed\_methods = ["DELETE", "GET", "HEAD", "OPTIONS", "PATCH", "POST", "PUT"] |
|  | cached\_methods = ["GET", "HEAD"] |
|  | target\_origin\_id = local.s3\_origin\_id |
|  |  |
|  | forwarded\_values { |
|  | query\_string = false |
|  |  |
|  | cookies { |
|  | forward = "none" |
|  | } |
|  | } |
|  |  |
|  | viewer\_protocol\_policy = "allow-all" |
|  | min\_ttl = 0 |
|  | default\_ttl = 3600 |
|  | max\_ttl = 86400 |
|  | } |
|  |  |
|  | # Cache behavior with precedence 0 |
|  | ordered\_cache\_behavior { |
|  | path\_pattern = "/content/immutable/\*" |
|  | allowed\_methods = ["GET", "HEAD", "OPTIONS"] |
|  | cached\_methods = ["GET", "HEAD", "OPTIONS"] |
|  | target\_origin\_id = local.s3\_origin\_id |
|  |  |
|  | forwarded\_values { |
|  | query\_string = false |
|  | headers = ["Origin"] |
|  |  |
|  | cookies { |
|  | forward = "none" |
|  | } |
|  | } |
|  |  |
|  | min\_ttl = 0 |
|  | default\_ttl = 86400 |
|  | max\_ttl = 31536000 |
|  | compress = true |
|  | viewer\_protocol\_policy = "redirect-to-https" |
|  | } |
|  |  |
|  | # Cache behavior with precedence 1 |
|  | ordered\_cache\_behavior { |
|  | path\_pattern = "/content/\*" |
|  | allowed\_methods = ["GET", "HEAD", "OPTIONS"] |
|  | cached\_methods = ["GET", "HEAD"] |
|  | target\_origin\_id = local.s3\_origin\_id |
|  |  |
|  | forwarded\_values { |
|  | query\_string = false |
|  |  |
|  | cookies { |
|  | forward = "none" |
|  | } |
|  | } |
|  |  |
|  | min\_ttl = 0 |
|  | default\_ttl = 3600 |
|  | max\_ttl = 86400 |
|  | compress = true |
|  | viewer\_protocol\_policy = "redirect-to-https" |
|  | } |
|  |  |
|  | price\_class = "PriceClass\_200" |
|  |  |
|  | restrictions { |
|  | geo\_restriction { |
|  | restriction\_type = "whitelist" |
|  | locations = ["IN", "CA", "GB", "DE"] |
|  | } |
|  | } |
|  |  |
|  | tags = { |
|  | Environment = "production" |
|  | } |
|  |  |
|  | viewer\_certificate { |
|  | cloudfront\_default\_certificate = true |
|  | } |
|  | } |
|  |  |
|  |  |
|  | output "myos\_ip" { |
|  | value = aws\_instance.web.public\_ip |
|  | } |
|  |  |
|  |  |
|  | resource "null\_resource" "nulllocal2" { |
|  | depends\_on = [ |
|  | null\_resource.harddisk, |
|  | ] |
|  | provisioner "local-exec" { |
|  | command = "start chrome ${aws\_instance.web.public\_ip}" |
|  | } |
|  | }  Outputs of my laptop: Task1 |



Scenario1



Scenario 2



Scenario3



Scenario 4



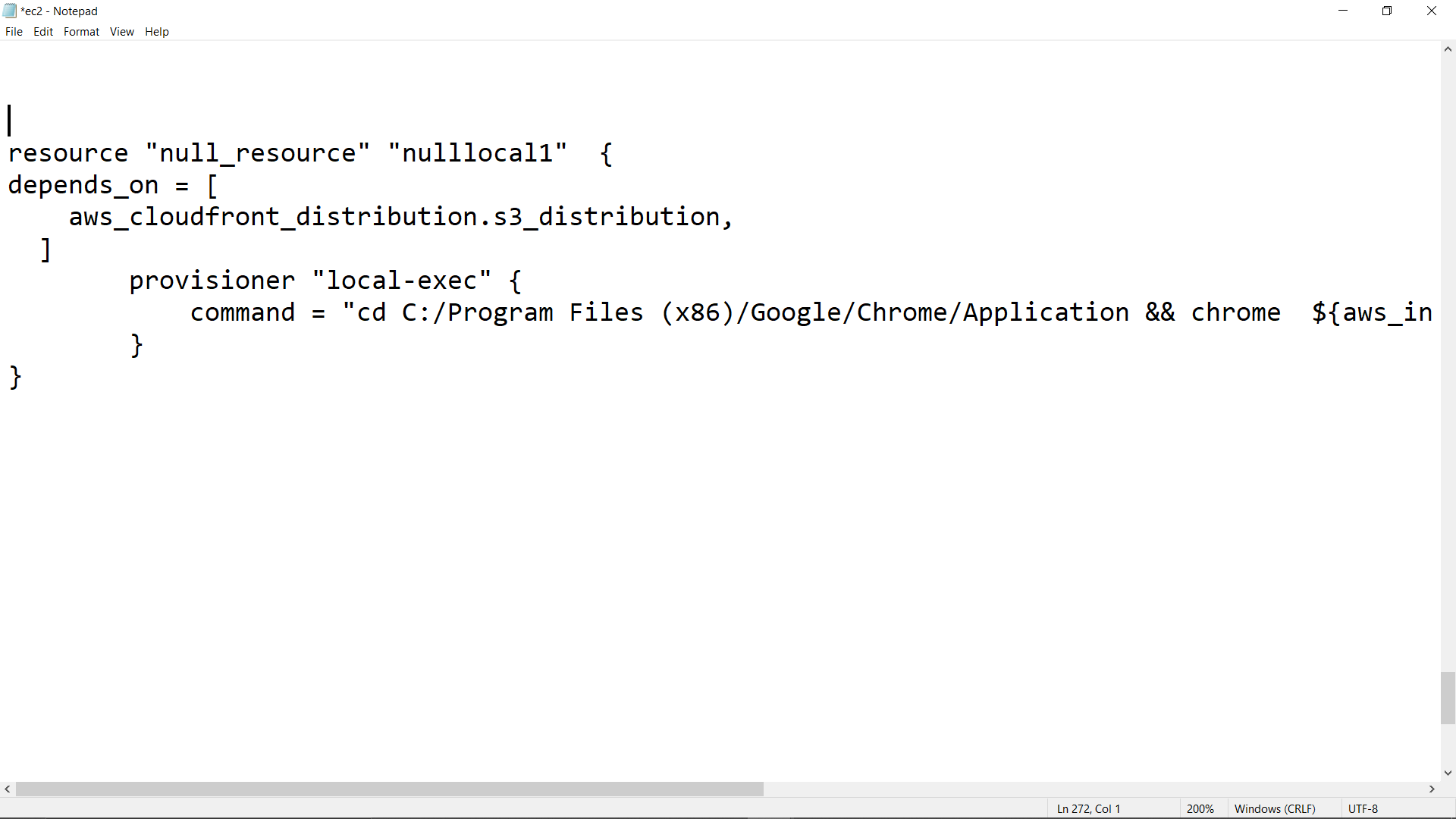
Scenario 5



Scenario 6



Scenario 7



Scenario 8