1 - Installing Homebrew on Mac OS: https://brew.sh
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

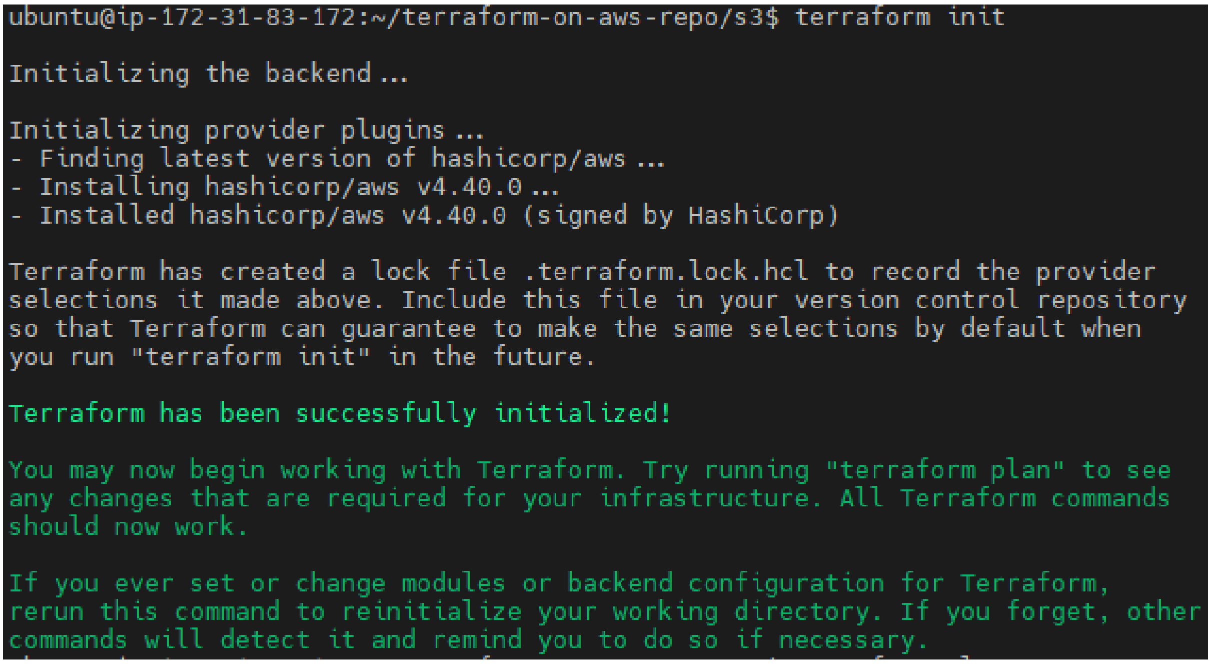
2 - Installing Terraform:
https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli

3 - How to create S3 bucket in AWS using Terraform:
https://www.coachdevops.com/2021/07/terraform-create-s3-bucket-example-how_12.html
What is S3 bucket?
https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html

3.1 - Creating an IAM Role for S3 Access In the Same AWS Account:

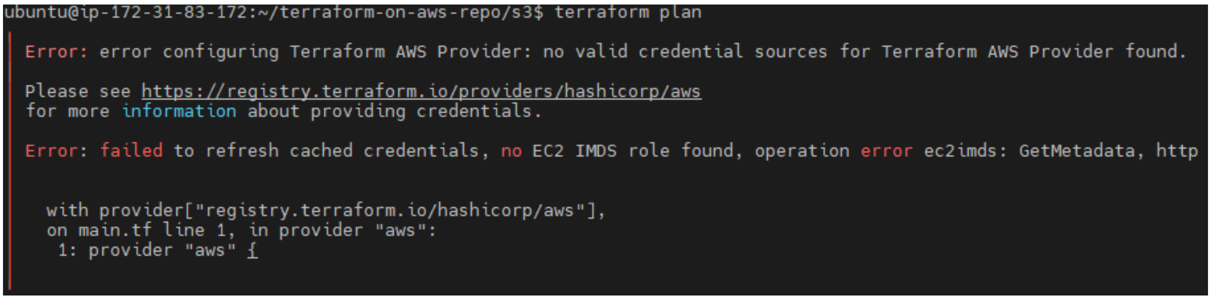
https://www.ibm.com/docs/vi/atcm/1.3.1?topic=cir-creating-iam-role-s3-access-in-same-aws-account

3.2 - terraform init:



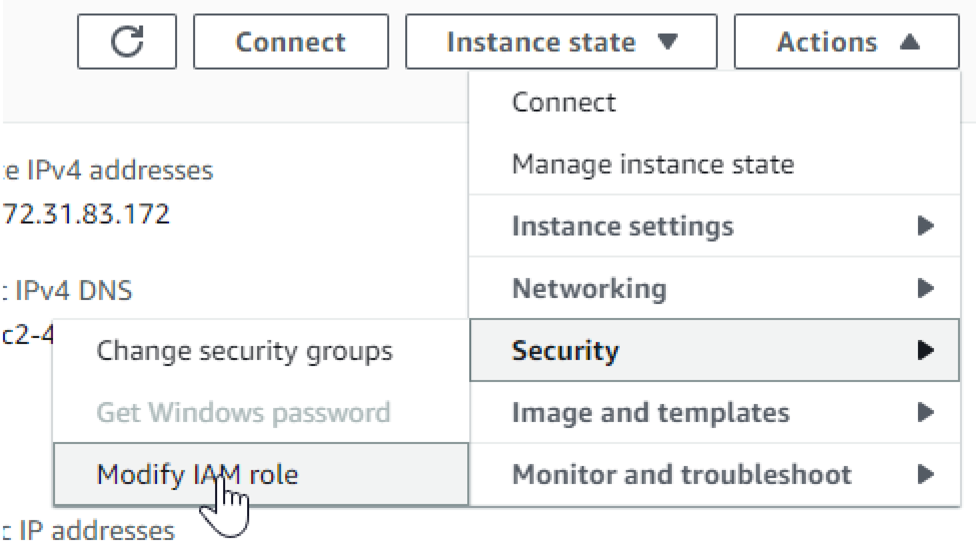
```
ubuntu@ip-172-31-83-172:~/terraform-on-aws-repo/s3$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.40.0...
- Installed hashicorp/aws v4.40.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

3.3 - terraform plan:

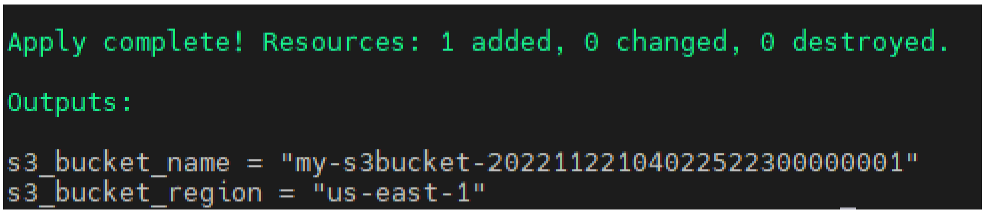


```
ubuntu@ip-172-31-83-172:~/terraform-on-aws-repo/s3$ terraform plan

Error: error configuring Terraform AWS Provider: no valid credential sources for Terraform AWS Provider found.

Please see https://registry.terraform.io/providers/hashicorp/aws
for more information about providing credentials.

Error: failed to refresh cached credentials, no EC2 IMDS role found, operation error ec2imds: GetMetadata, http


  with provider["registry.terraform.io/hashicorp/aws"],
  on main.tf line 1, in provider "aws":
   1: provider "aws" {
```

3.3.1 - Solve:

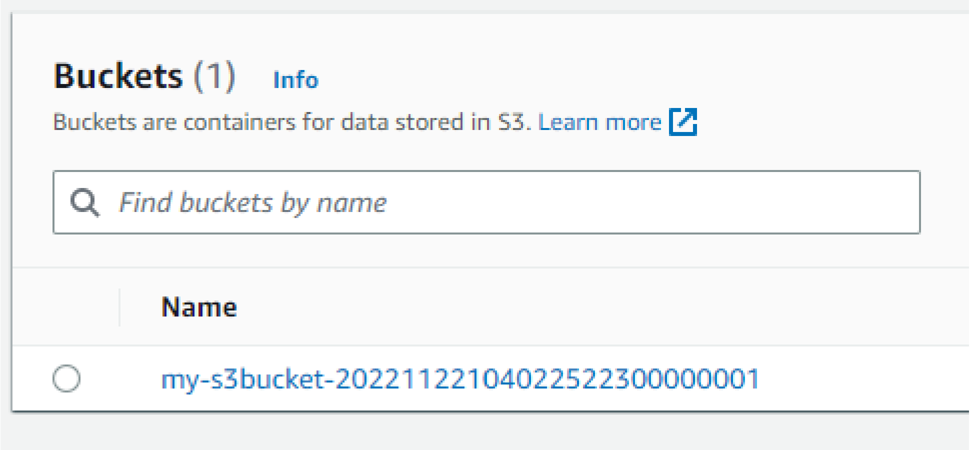


3.4 - terraform apply:



```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

s3_bucket_name = "my-s3bucket-20221122104022522300000001"
s3_bucket_region = "us-east-1"
```

3.5 - result:



Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

	Name
○	my-s3bucket-20221122104022522300000001

4 - How to access S3 Bucket from EC2 Instance using IAM Role:
https://www.youtube.com/watch?v=w4XxfrDI-fM

4.1 - How to create an Nginx instance in AWS using Terraform:
https://awstip.com/how-to-create-an-nginx-instance-in-aws-using-terraform-feb6af12749a

4.2 - Deploy an Nginx Web Server on an EC2 instance with Terraform: https://dev.to/danielfavour/deploy-an-nginx-web-server-on-an-ec2-instance-with-terraform-621

5 - Add an application load balancer to AWS EC2 using Terraform: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/lb

https://skundunotes.com/2022/07/30/add-an-application-load-balancer-to-aws-ec2-using-terraform/

5.1 - Creating ALB listener rules: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/lb_listener_rule

Issues:

```
│ Error: failure configuring LB attributes: InvalidConfigurationRequest: Access Denied for bucket: my-s3bucket-20221123144955009600000001. Please check S3bucket permission
│       status code: 400, request id: 3e5462e3-9461-4e81-98be-43d6365993c0
│
│   with aws_alb.proxy,
│   on main.tf line 35, in resource "aws_alb" "proxy":
│   35: resource "aws_alb" "proxy" {
```

My solution:

```
resource "aws_alb" "alb1" {
  name               = "alb1"
  internal           = "false"  // To be accessed from internet
  security_groups    = [aws_security_group.webserver.id]
  subnets            = [aws_subnet.public-subnet1.id,
aws_subnet.public-subnet2.id]

  enable_deletion_protection = false

  /*
  access_logs {
    bucket  = aws_s3_bucket.lb_logs.bucket
    prefix  = "test-lb"
    enabled = true
  }
  */
```

Terraform destroy.

Terraform appy:

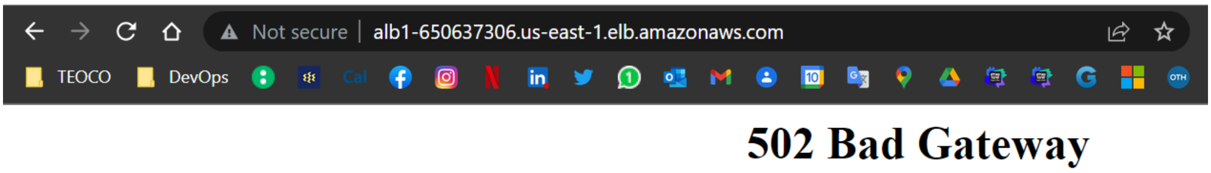
Error:

```
 Error: Error launching source instance: InvalidParameter: Security group
sg-05257c518034e0c14 and subnet subnet-05bcf213054755b48 belong to different
networks.
│       status code: 400, request id: 4d6e94c0-38ca-4824-aa61-aac64b49cb7e
│
│    with aws_instance.webserver-red,
│    on instances.tf line 1, in resource "aws_instance" "webserver-red":
│     1: resource "aws_instance" "webserver-red" {
│
╵

╷
│ Error: Error launching source instance: InvalidParameter: Security group
sg-05257c518034e0c14 and subnet subnet-05bcf213054755b48 belong to different
networks.
│       status code: 400, request id: 1dcdff32-e4f1-4944-a71f-eb35114363ca
│
│    with aws_instance.webserver-blue,
│    on instances.tf line 13, in resource "aws_instance" "webserver-blue":
│    13: resource "aws_instance" "webserver-blue" {
```

Solution:
https://stackoverflow.com/questions/48252650/security-group-and-subnet-belongs-to-different-networks

6 - How to Deploy a Set of EC2 Instances behind an ALB Using Terraform:
https://medium.com/geekculture/how-to-manage-auto-scaling-group-and-load-balancer-with-terraform-9ece263060b5

7 - http://alb1-650637306.us-east-1.elb.amazonaws.com/

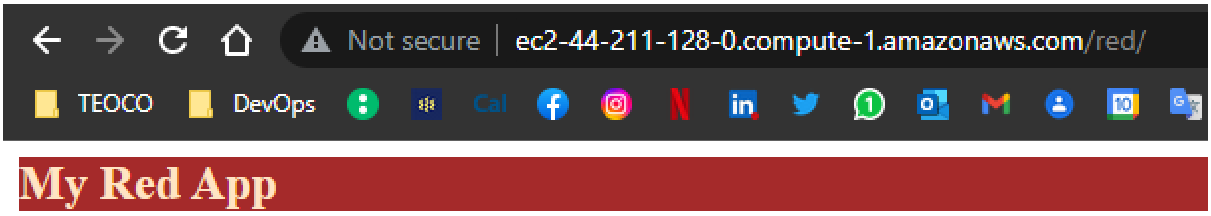


Not secure | alb1-650637306.us-east-1.elb.amazonaws.com

502 Bad Gateway

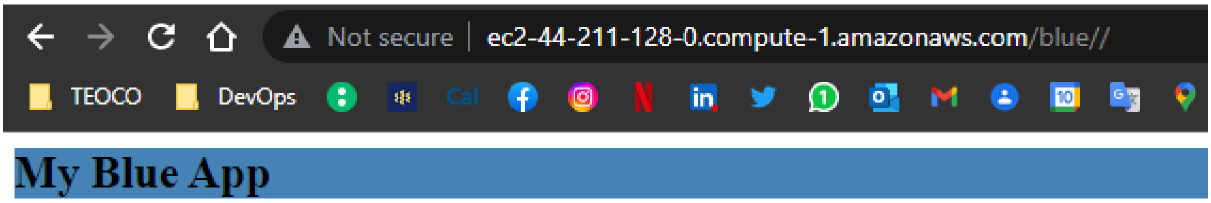
7.1 - Solution:
https://k21academy.com/oracle-ebs-r12-on-cloud/solved-oci-load-balancer-throwing-error-502-bad-gateway/
https://docs.nginx.com/nginx/admin-guide/web-server/serving-static-content/

http://ec2-44-211-128-0.compute-1.amazonaws.com/red/



My Red App

http://ec2-44-211-128-0.compute-1.amazonaws.com/blue/



My Blue App