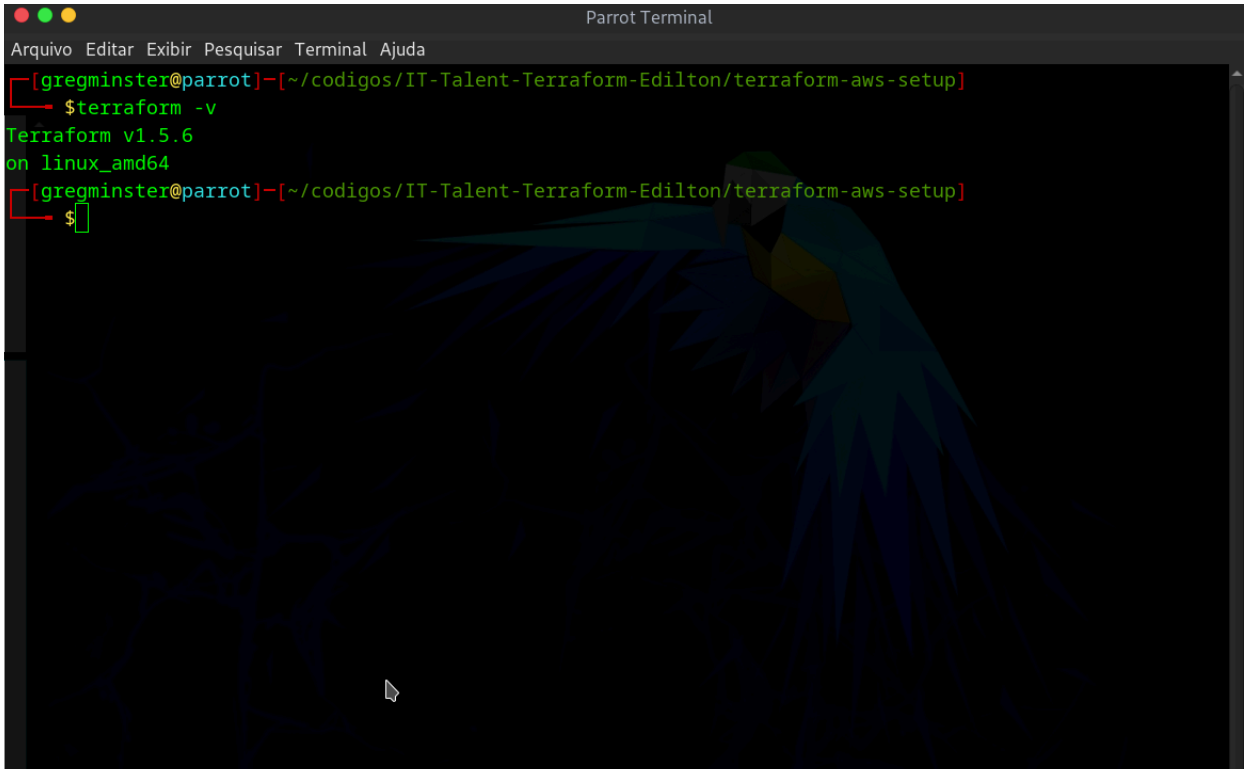
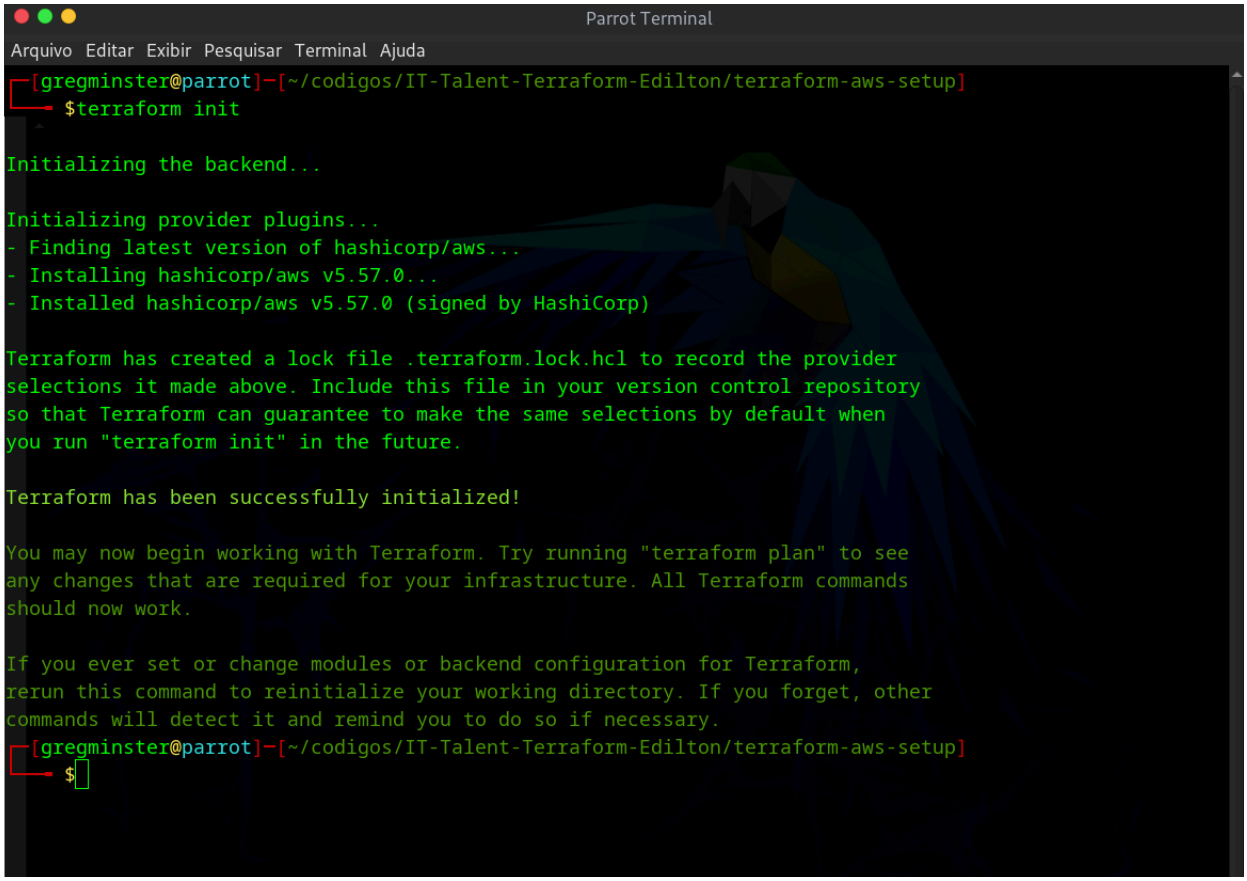
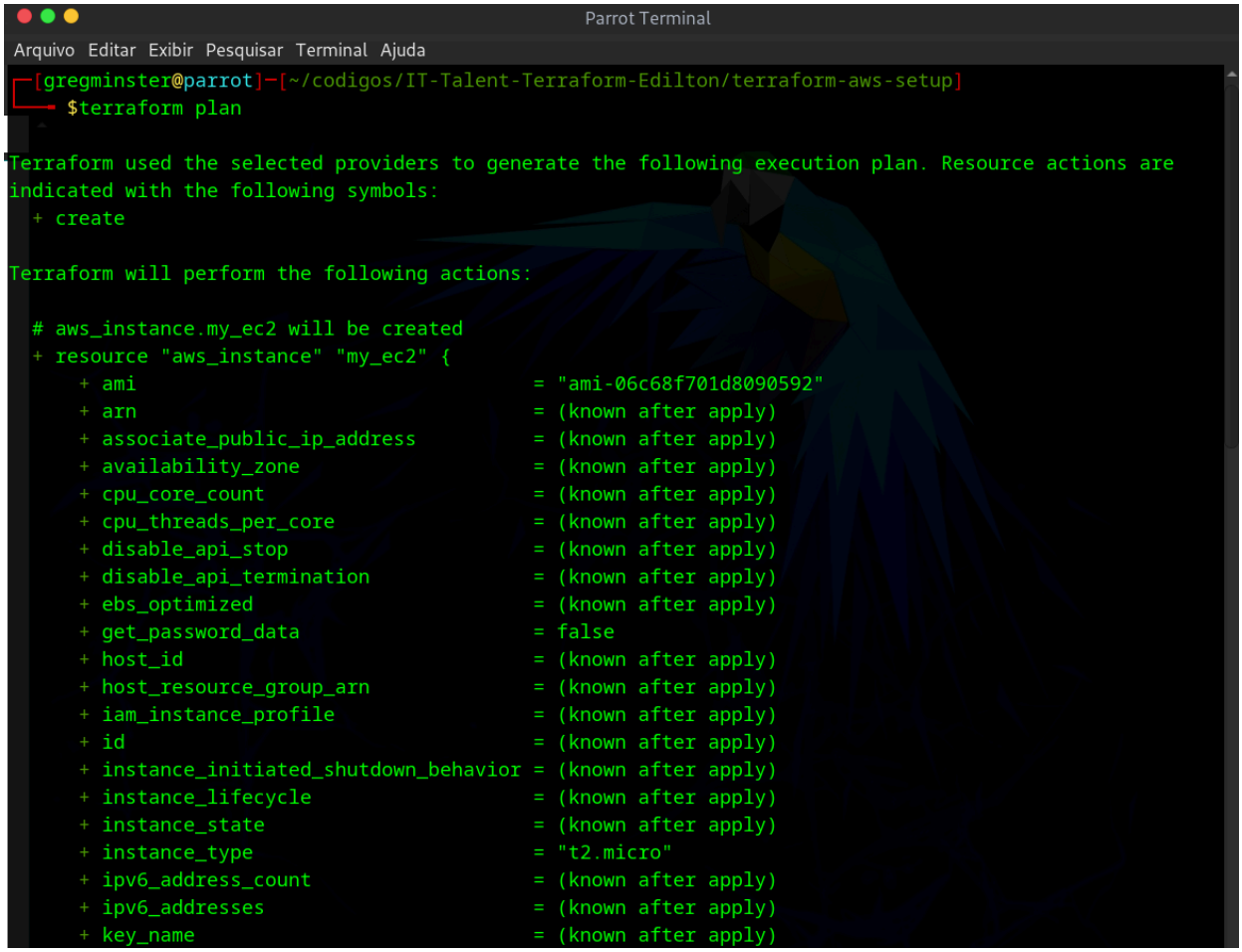
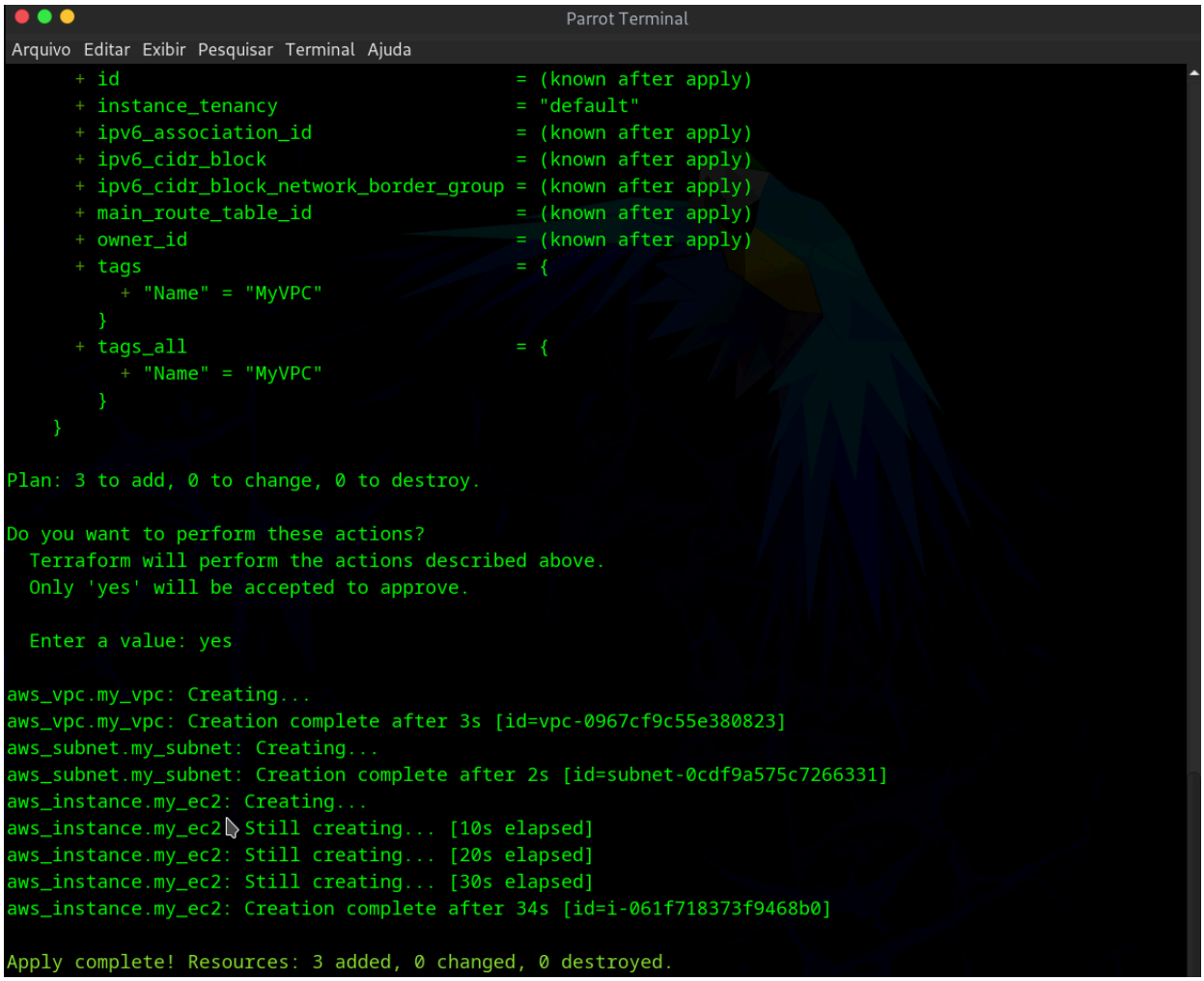
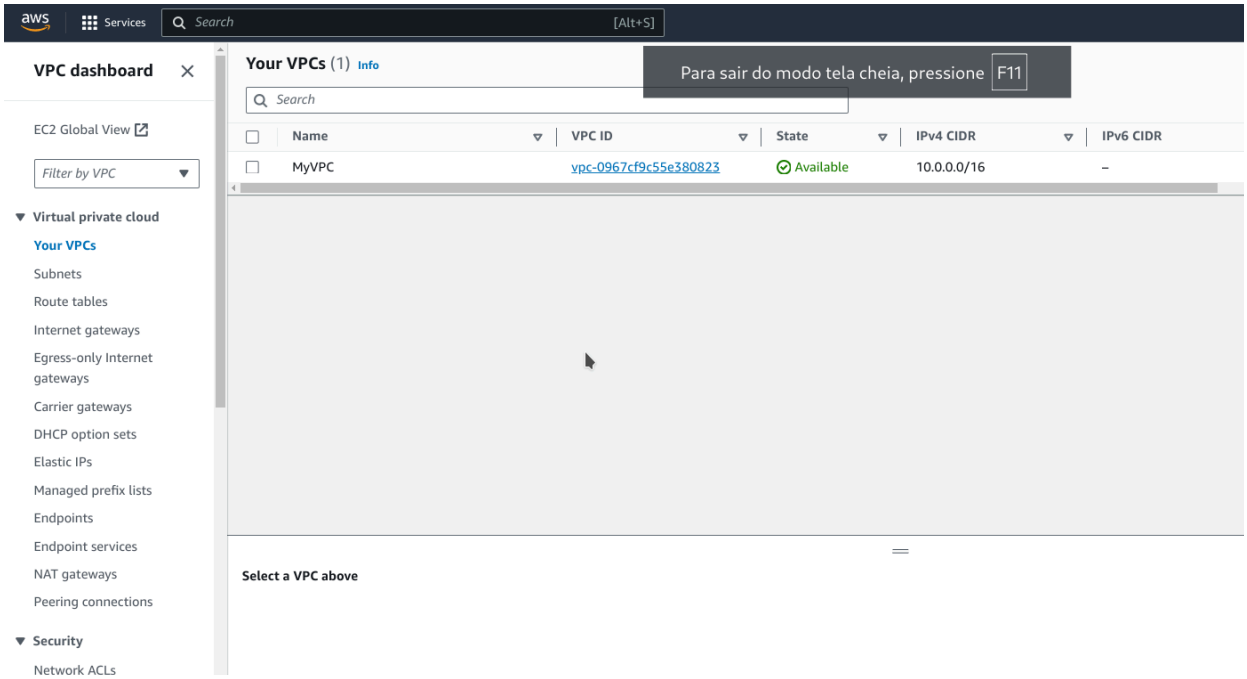
# SCREENSHOTS DA ATIVIDADE COM TERRAFORM - edilton jr

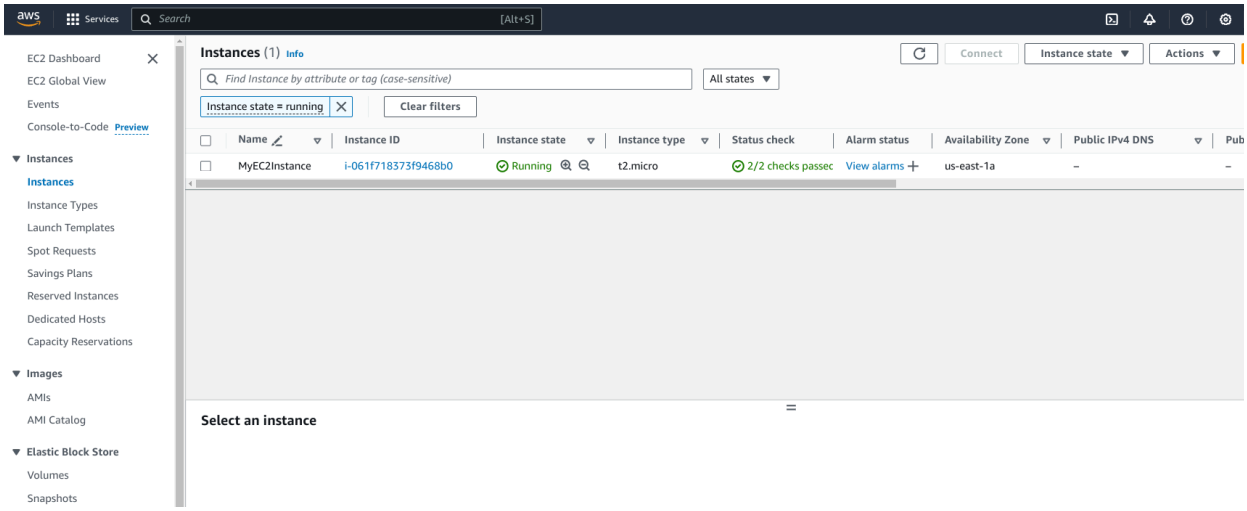

Parrot Terminal
Arquivo Editar Exibir Pesquisar Terminal Ajuda
[gregminster@parrot]-[~/codigos/IT-Talent-Terraform-Edilton/terraform-aws-setup]
$terraform -v
Terraform v1.5.6
on linux_amd64
[gregminster@parrot]-[~/codigos/IT-Talent-Terraform-Edilton/terraform-aws-setup]
$

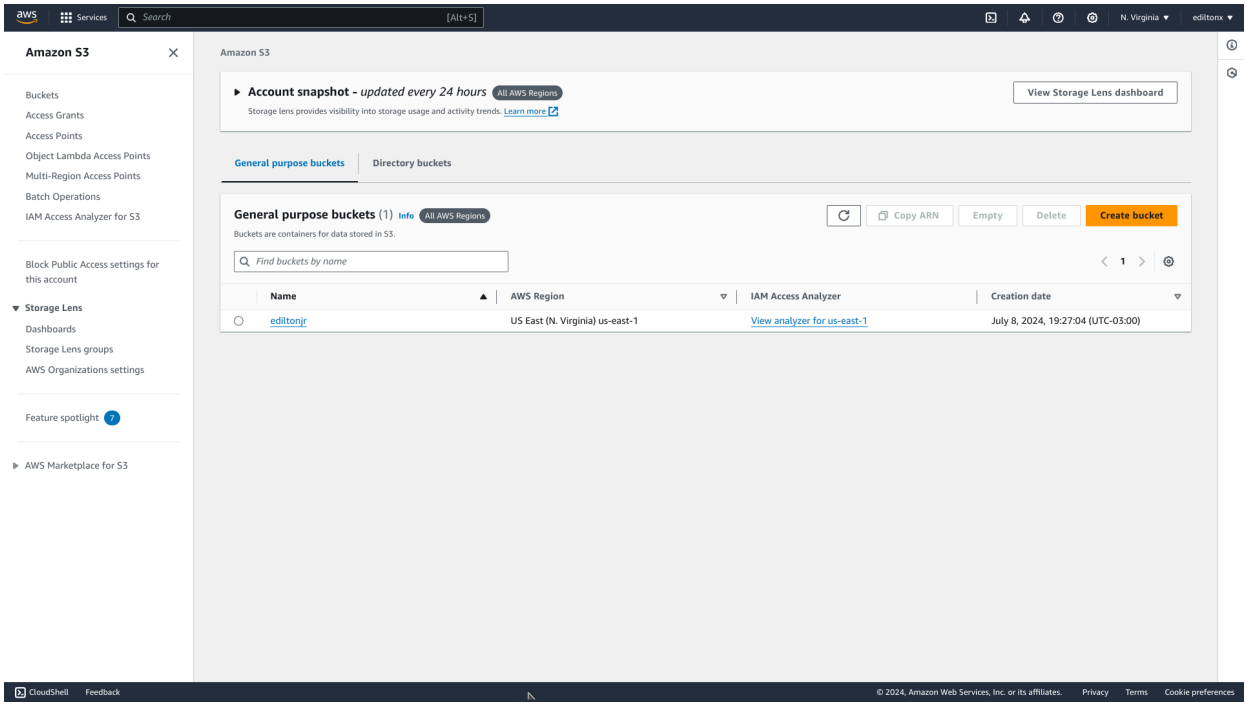

Parrot Terminal
Arquivo Editar Exibir Pesquisar Terminal Ajuda
[gregminster@parrot]-[~/codigos/IT-Talent-Terraform-Edilton/terraform-aws-setup]
$terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.57.0...
- Installed hashicorp/aws v5.57.0 (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.
[gregminster@parrot]-[~/codigos/IT-Talent-Terraform-Edilton/terraform-aws-setup]
$

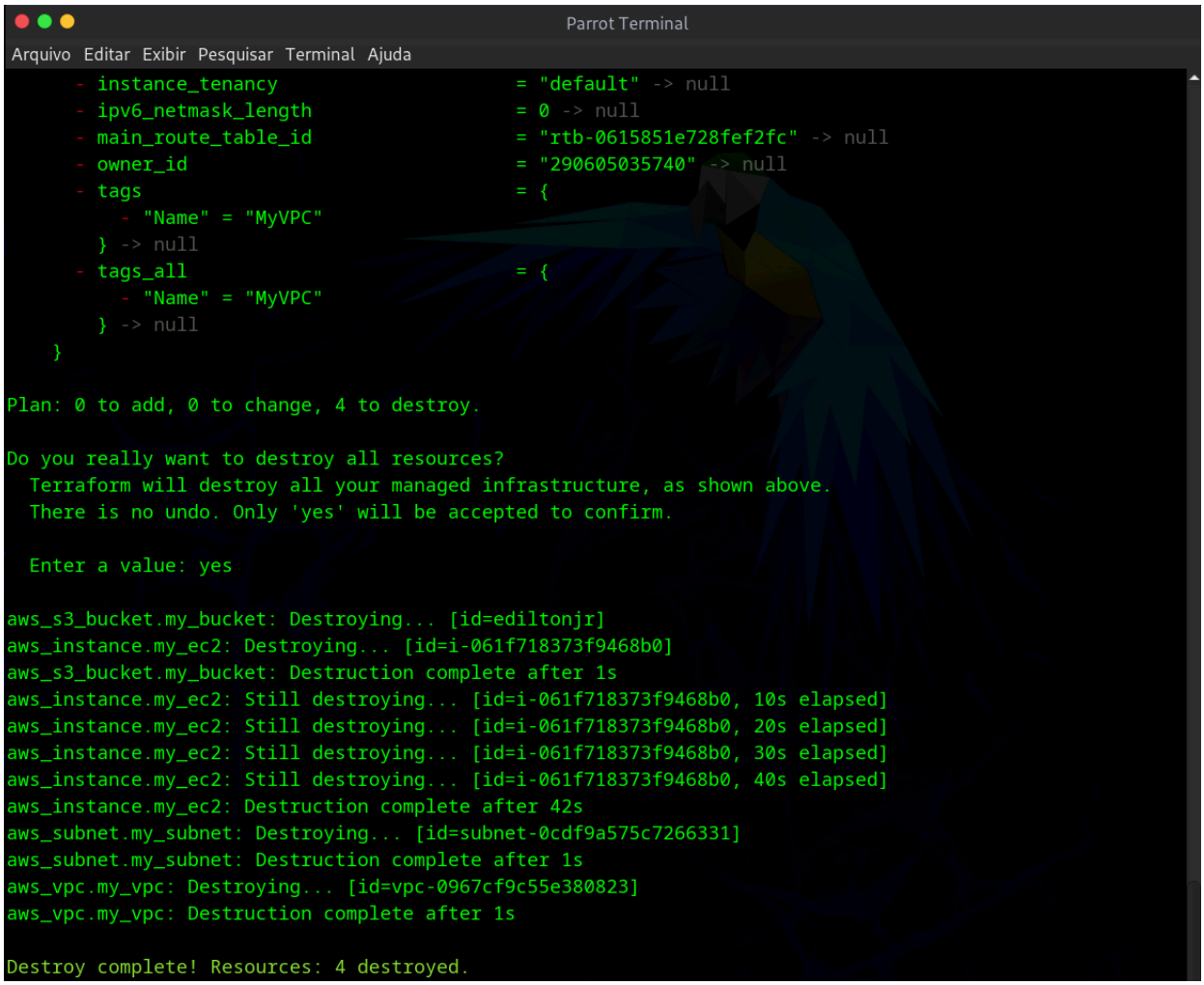

Parrot Terminal
Arquivo Editar Exibir Pesquisar Terminal Ajuda
[gregminster@parrot]-[~/codigos/IT-Talent-Terraform-Edilton/terraform-aws-setup]
$terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
Terraform will perform the following actions:
# aws_instance.my_ec2 will be created
+ resource "aws_instance" "my_ec2" {
+ ami = "ami-06c68f701d8090592"
+ arn = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone = (known after apply)
+ cpu_core_count = (known after apply)
+ cpu_threads_per_core = (known after apply)
+ disable_api_stop = (known after apply)
+ disable_api_termination = (known after apply)
+ ebs_optimized = (known after apply)
+ get_password_data = false
+ host_id = (known after apply)
+ host_resource_group_arn = (known after apply)
+ iam_instance_profile = (known after apply)
+ id = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle = (known after apply)
+ instance_state = (known after apply)
+ instance_type = "t2.micro"
+ ipv6_address_count = (known after apply)
+ ipv6_addresses = (known after apply)
+ key_name = (known after apply)


Parrot Terminal
Arquivo Editar Exibir Pesquisar Terminal Ajuda
+ id = (known after apply)
+ instance_tenancy = "default"
+ ipv6_association_id = (known after apply)
+ ipv6_cidr_block = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id = (known after apply)
+ owner_id = (known after apply)
+ tags = {
+ "Name" = "MyVPC"
}
+ tags_all = {
+ "Name" = "MyVPC"
}
}
Plan: 3 to add, 0 to change, 0 to destroy.
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value: yes
aws_vpc.my_vpc: Creating...
aws_vpc.my_vpc: Creation complete after 3s [id=vpc-0967cf9c55e380823]
aws_subnet.my_subnet: Creating...
aws_subnet.my_subnet: Creation complete after 2s [id=subnet-0cdf9a575c7266331]
aws_instance.my_ec2: Creating...
aws_instance.my_ec2: Still creating... [10s elapsed]
aws_instance.my_ec2: Still creating... [20s elapsed]
aws_instance.my_ec2: Still creating... [30s elapsed]
aws_instance.my_ec2: Creation complete after 34s [id=i-061f718373f9468b0]
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.


aws
Services
Search
[Alt+S]
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only Internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
Your VPCs (1) Info
Search
Para sair do modo tela cheia, pressione F11
Name
VPC ID
State
IPv4 CIDR
IPv6 CIDR
MyVPC
vpc-0967cf9c55e380823
Available
10.0.0.0/16
–
Select a VPC above


Services
Search
[Alt+S]
EC2 Dashboard
EC2 Global View
Events
Console-to-Code Preview
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Instances (1) Info
Find Instance by attribute or tag (case-sensitive)
All states
Instance state = running
Clear filters
Connect
Instance state
Actions
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
MyEC2Instance
i-061f718373f9468b0
Running
t2.micro
2/2 checks passec
View alarms
us-east-1a
Select an instance


Services
Search
[Alt+S]
N. Virginia
editonx
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 7
AWS Marketplace for S3
Amazon S3
Account snapshot - updated every 24 hours
All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets (1) Info
All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
IAM Access Analyzer
Creation date
editonjr
US East (N. Virginia) us-east-1
View analyzer for us-east-1
July 8, 2024, 19:27:04 (UTC-03:00)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Parrot Terminal
Arquivo Editar Exibir Pesquisar Terminal Ajuda
- instance_tenancy = "default" -> null
- ipv6_netmask_length = 0 -> null
- main_route_table_id = "rtb-0615851e728fef2fc" -> null
- owner_id = "290605035740" -> null
- tags = {
- "Name" = "MyVPC"
} -> null
- tags_all = {
- "Name" = "MyVPC"
} -> null
}
Plan: 0 to add, 0 to change, 4 to destroy.
Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.
Enter a value: yes
aws_s3_bucket.my_bucket: Destroying... [id=ediltonjr]
aws_instance.my_ec2: Destroying... [id=i-061f718373f9468b0]
aws_s3_bucket.my_bucket: Destruction complete after 1s
aws_instance.my_ec2: Still destroying... [id=i-061f718373f9468b0, 10s elapsed]
aws_instance.my_ec2: Still destroying... [id=i-061f718373f9468b0, 20s elapsed]
aws_instance.my_ec2: Still destroying... [id=i-061f718373f9468b0, 30s elapsed]
aws_instance.my_ec2: Still destroying... [id=i-061f718373f9468b0, 40s elapsed]
aws_instance.my_ec2: Destruction complete after 42s
aws_subnet.my_subnet: Destroying... [id=subnet-0cdf9a575c7266331]
aws_subnet.my_subnet: Destruction complete after 1s
aws_vpc.my_vpc: Destroying... [id=vpc-0967cf9c55e380823]
aws_vpc.my_vpc: Destruction complete after 1s
Destroy complete! Resources: 4 destroyed.