Check list:

- [x] ~~Github repo - https://github.com/dustman7g/tech-challenge-2025~~
- [x] ~~Terraform modules - coalfires modules encouraged~~
    - [x] ~~Ec2 moudule = https://github.com/Coalfire-CF/terraform-aws-ec2/tree/main~~
- [x] ~~Diagram of solution~~
- [x] ~~Readme.md~~
- [x] ~~Deployment steps~~
- [x] ~~Commentary~~
    - [x] ~~Breaking up the subnets to use ALB~~
    - [x] ~~Using Management subnet for ALB since it public~~
    - [x] ~~Adding keys to talk to MGMT server to App servers for testing~~
    - [x] ~~t3.micro instead of t2.micro since t3 is on free tier~~
    - [x] ~~Added VPC endpoint for S3 to download httpd~~
- [x] ~~Network~~
    - [x] ~~1 VPC - 10.1.0.0/16~~
    - [x] ~~3 Subnets~~
        - [x] ~~Application /24 (no internet)~~
        - [x] ~~Management /24 (internet facing)~~
        - [x] ~~Application /24 (no internet)~~
- [x] ~~Compute~~
    - [x] ~~Ec2 in ASG to run linux in the application subnet~~
        - [x] ~~SG Allows from management ec2, allows web traffic from the Application Load balancer and no external traffic~~
        - [x] ~~Script installation of apache~~
        - [x] ~~2 min and 6 max host in ASG~~
        - [x] ~~t2.micro~~
    - [x] ~~1 EC2 running linux in MGMT Subnet~~
        - [x] ~~SG Allows SSH from a single specific IP or network space only~~
        - [x] ~~Can SSH from this instance to ASG~~
        - [x] ~~T2.micro~~
- [x] ~~Supporting Infrastructure~~
    - [x] ~~One ALB that sends web traffic to the ec2's in the ASG~~
- [x] ~~Document in README~~
    - [x] ~~Analysis of your own deployed infra~~
        - [x] ~~What security gaps exist?~~
        - [x] ~~What availability issues?~~
        - [x] ~~Cost Optimization opportunities?~~
        - [x] ~~Operational short commings(no backups and no monitoring)?~~
    - [x] ~~Improvement Plan~~

- [x] ~~List specific changes you 'd make to improve security, resilience, cost, maintainability~~
- [x] ~~Prioritize them(why would you first and why?)~~
- [x] ~~Include at least 2 implemented improvements in code or scripts(tightening SG rules, adding cloud watch alarms, setting bucket polices)~~

- [x] ~~Runbook-style notes~~
  - [x] ~~How would someone else deploy and operate your environment?~~
  - [x] ~~How would you respond to an outage for the EC2 instance?~~

- [x] ~~Overall Deliverables~~
  - [x] ~~All terraform configurations~~
  - [x] ~~Architecture diagram~~
  - [x] ~~README including~~
    - [x] ~~Solution overview~~
    - [x] ~~Deployment instructions~~
    - [x] ~~Design decision and assumptions~~
    - [x] ~~Reference to resources used~~
      - [x] ~~Terraform registry~~
      - [x] ~~Coalfire git~~
    - [x] ~~Assumptions made~~
      - [x] ~~Internet gateway for internet access~~
      - [x] ~~VPC s3 endpoint to download httpd~~
    - [x] ~~Improvement plan with priorities~~
    - [x] ~~Analysis of operational gaps~~

To run terraform plan apply need to add to test in AWS
$env:AWS_ACCESS_KEY_ID = "your access key id"
$env:AWS_SECRET_ACCESS_KEY = "your access key"
$env:AWS_DEFAULT_REGION = "us-east-1"

####Add open SSH
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
####Service might not be running and in a disabled state - enable openSSH service and start it

```
Set-Service "ssh-agent" -StartupType Automatic
Start-Service "ssh-agent"
```

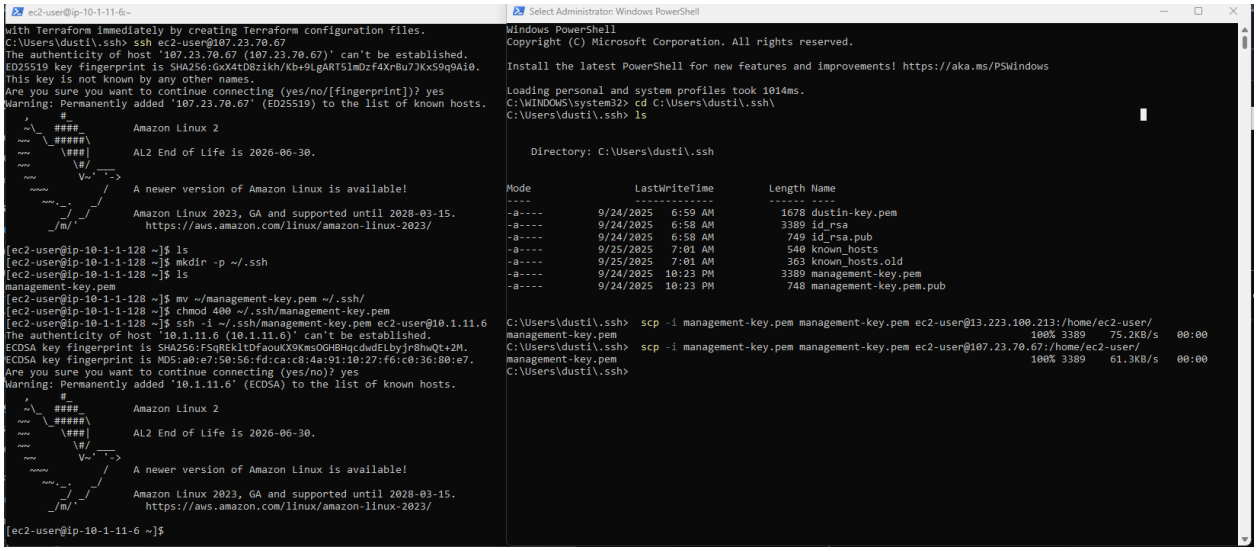
Generate a key pair on your local machine (if you don't have one yet)

On **Windows (PowerShell with OpenSSH installed)**:

```
ssh-keygen -t rsa -b 4096 -f $env:USERPROFILE\.ssh\management-key
```

- `-f $env:USERPROFILE\.ssh\management-key` → saves the files as:
  - `management-key` → private key (keep this safe!)
  - `management-key.pub` → public key (AWS needs this)

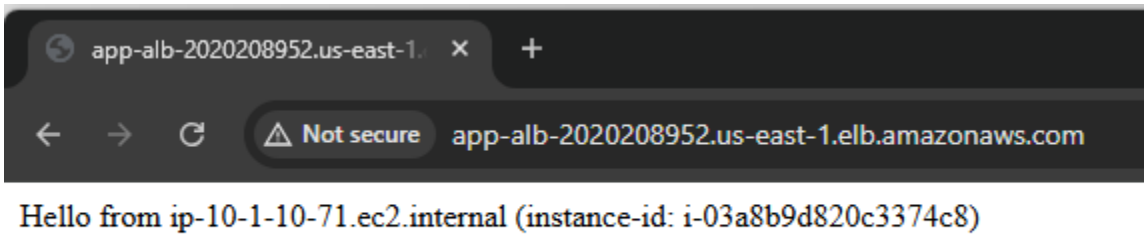
####### Find my public IP
(Invoke-WebRequest -Uri "https://ifconfig.me/ip").Content.Trim()
Update security group for management with your IP
ssh-add C:\users\username\.ssh\management-key.pem
ssh ec2-user@13.223.100.213

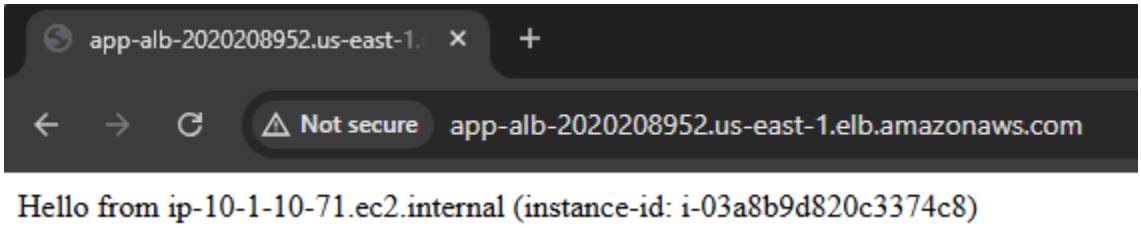


Copy key over to management server from local path
scp -i management-key.pem management-key.pem ec2-user@13.223.100.213:/home/ec2-user/
Make .ssh directory on management server, move mgmt pem key to folder and update permissions
mkdir -p ~/.ssh
mv ~/management-key.pem ~/.ssh/
chmod 400 ~/.ssh/management-key.pem
SSH from MGMT server to ASG/App instance
ssh -i ~/.ssh/management-key.pem ec2-user@10.1.11.6 or 10.1.10.71

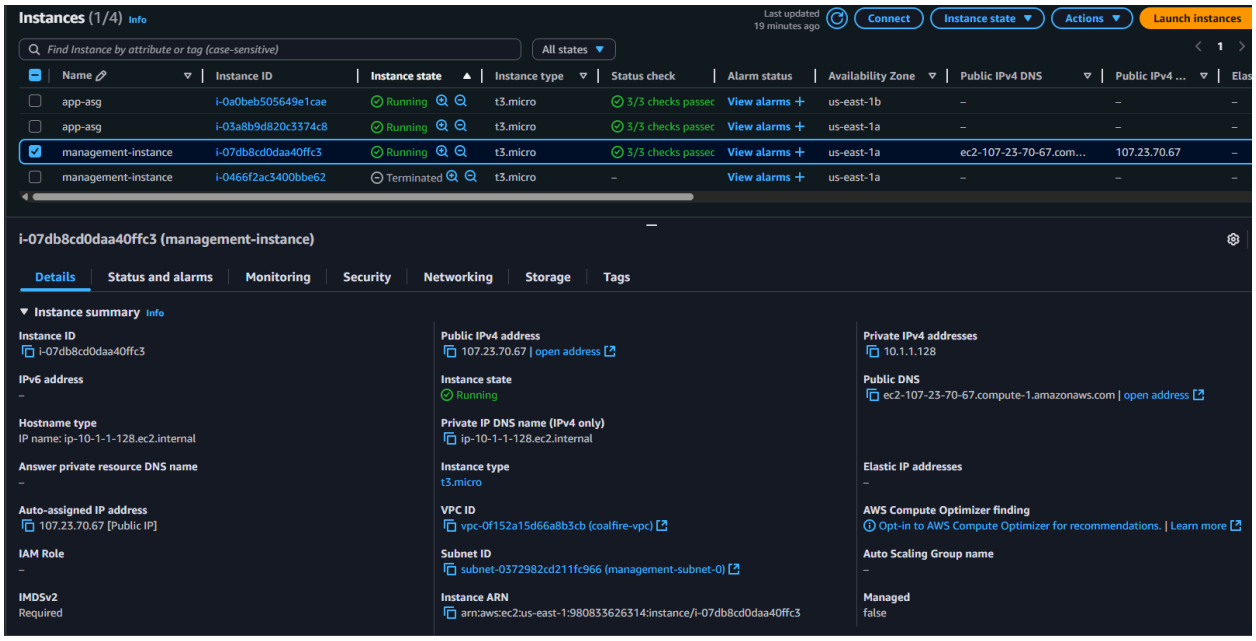
Evidence of Success

http://app-alb-2020208952.us-east-1.elb.amazonaws.com - Success

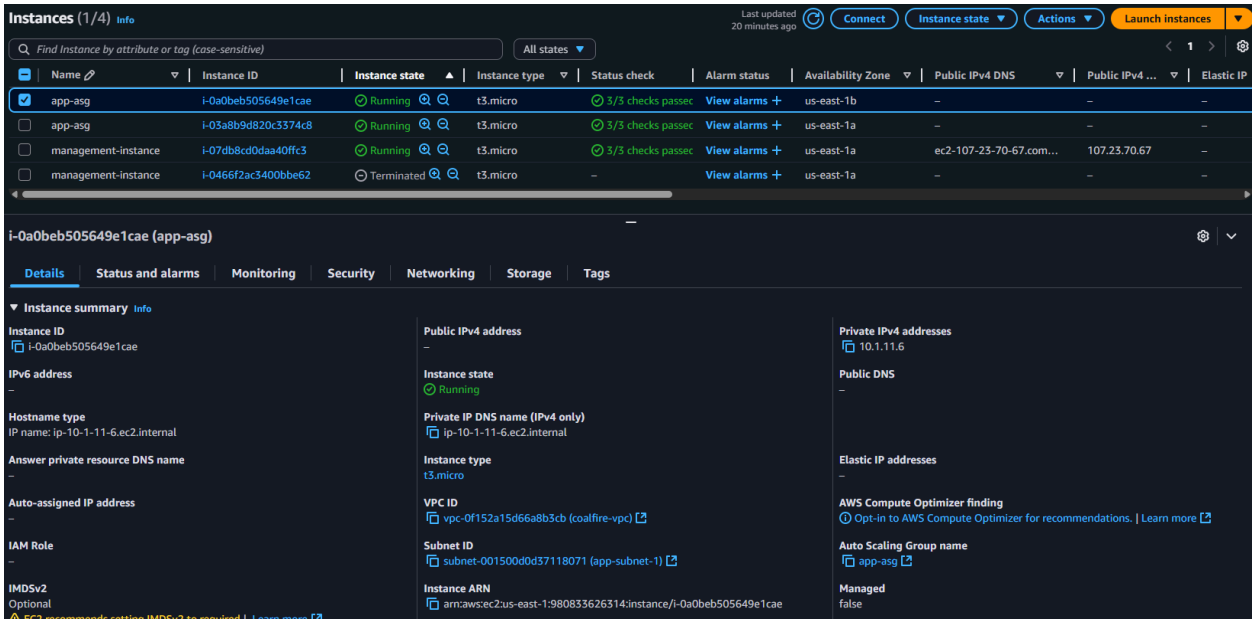




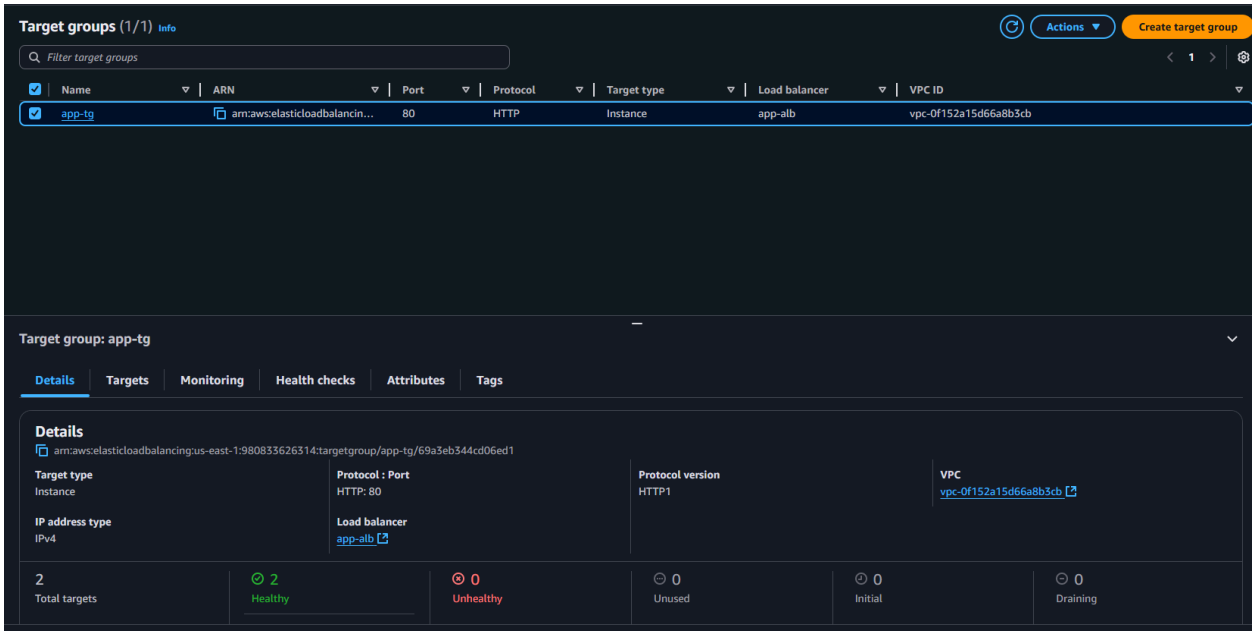
MGMT instance

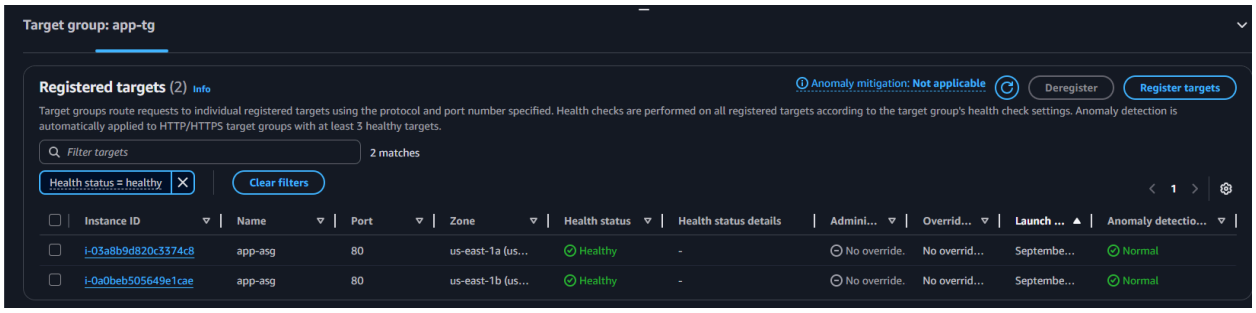


ASG

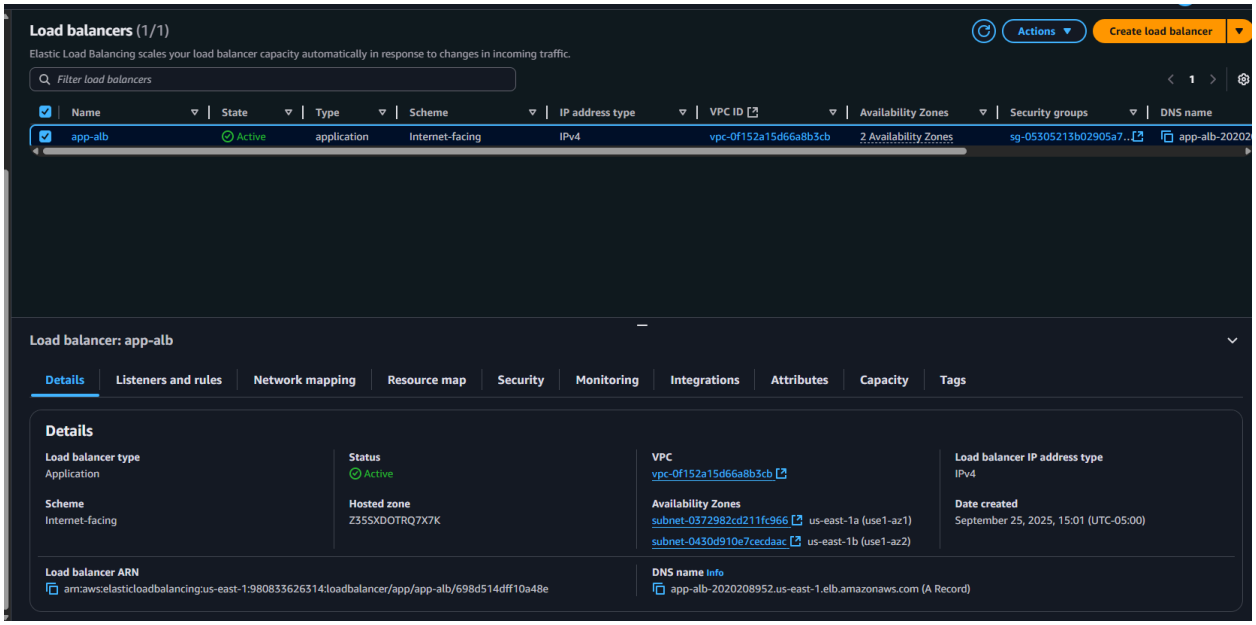


Target groups

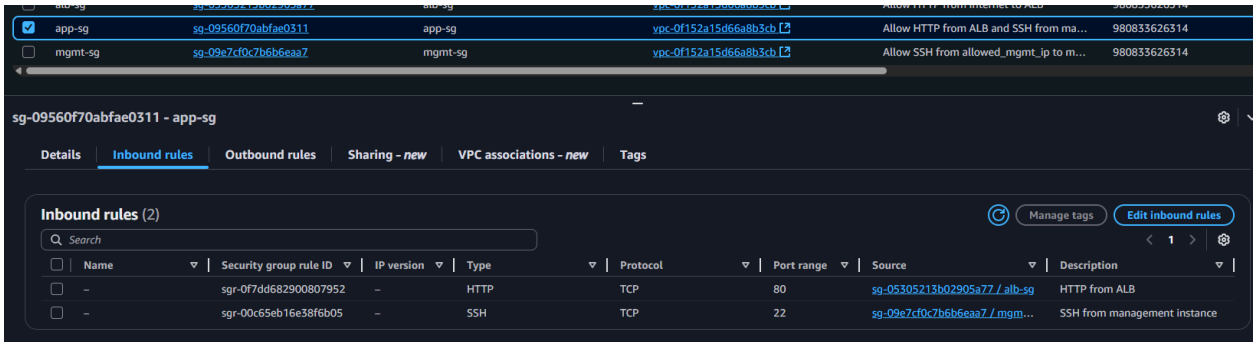


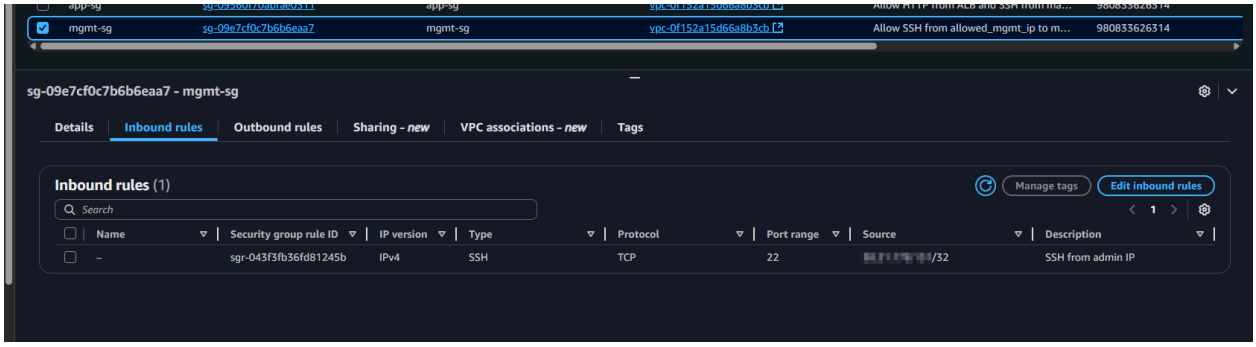


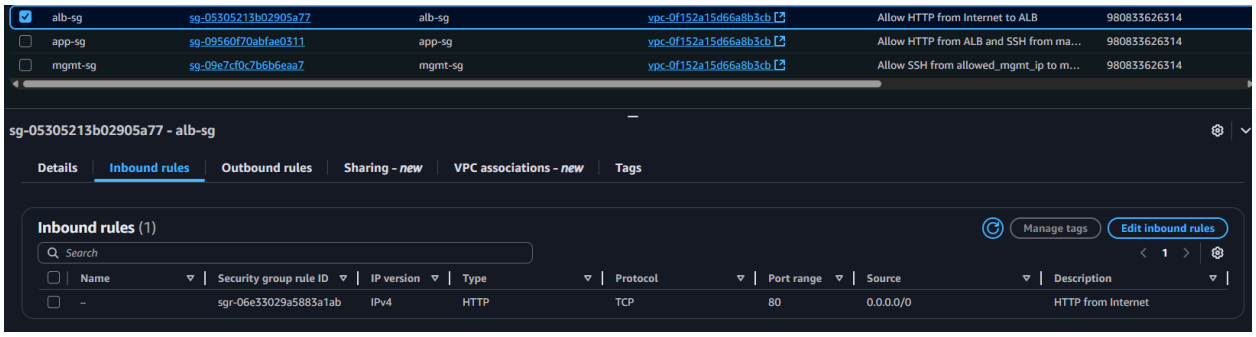
Load Balancer



SGs







Measure-Command {terraform apply -auto-approve | Out-File -FilePath "apply.log" -Encoding utf8}

Days : 0
Hours : 0
Minutes : 3
Seconds : 35
Milliseconds : 556
Ticks : 2155563592
TotalDays : 0.00249486526851852
TotalHours : 0.0598767664444444
TotalMinutes : 3.59260598666667
TotalSeconds : 215.5563592
TotalMilliseconds : 215556.3592

Measure-Command {terraform destroy -auto-approve | Out-File -FilePath "destroy.log" -Encoding utf8}

```
Days              : 0
Hours             : 0
Minutes           : 6
Seconds           : 41
```

```
Milliseconds      : 438
Ticks             : 4014384869
TotalDays         : 0.004646627878356481
TotalHours        : 0.111510690805556
TotalMinutes      : 6.69064144833333
TotalSeconds      : 401.4384869
TotalMilliseconds : 401438.4869
```