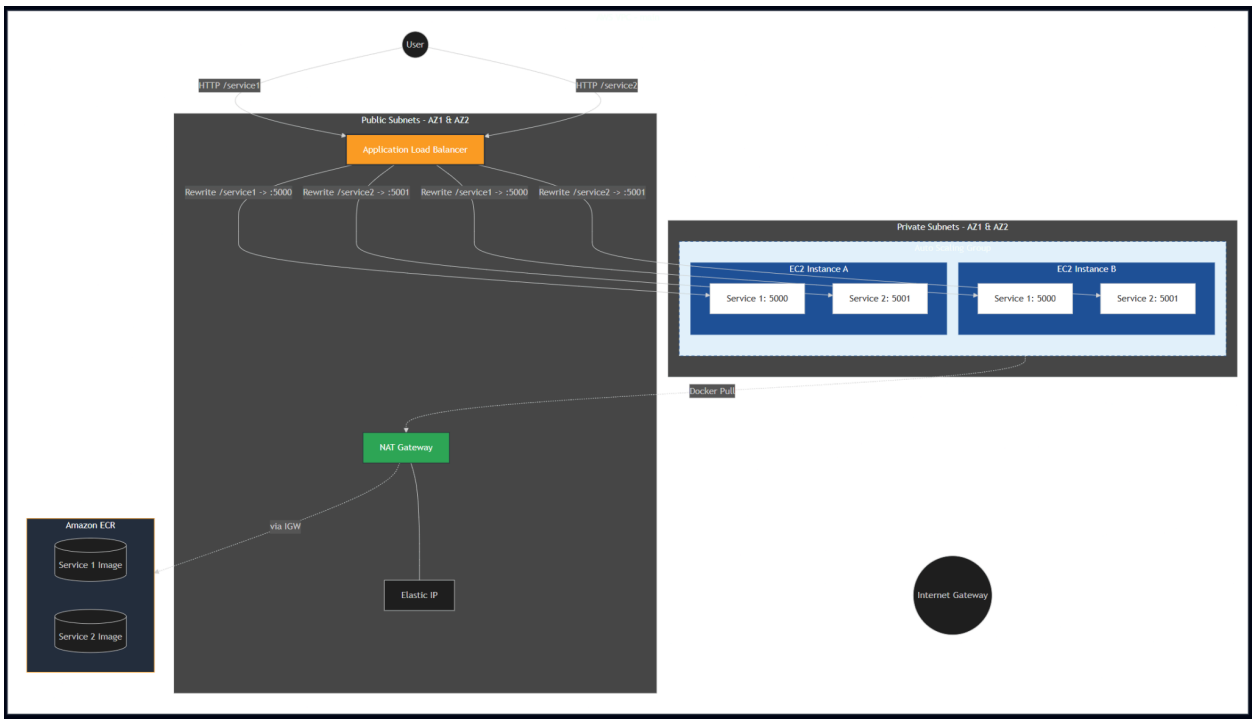
Deliverables Documentation

This project implements a secure, scalable multi-tier cloud architecture on AWS, fully automated through Terraform Infrastructure as Code (IaC). The foundation consists of a custom VPC with network isolation achieved through separate public and private subnets across multiple Availability Zones, ensuring compute resources remain protected from direct internet exposure. Traffic is managed by an Application Load Balancer (ALB) featuring path-based routing to containerized services hosted on EC2 instances within an Auto Scaling Group (ASG). Security is prioritized using least-privilege IAM roles and AWS Systems Manager (SSM) for terminal access, eliminating the need for traditional SSH keys. The deployment cycle concludes with an automated Python validation suite that leverages Boto3 and health-check logic to verify infrastructure integrity and service availability, providing a robust "quality gate" for the entire stack.

Github repo link:
https://github.com/jpranay08/terraform-aws-validated-compute

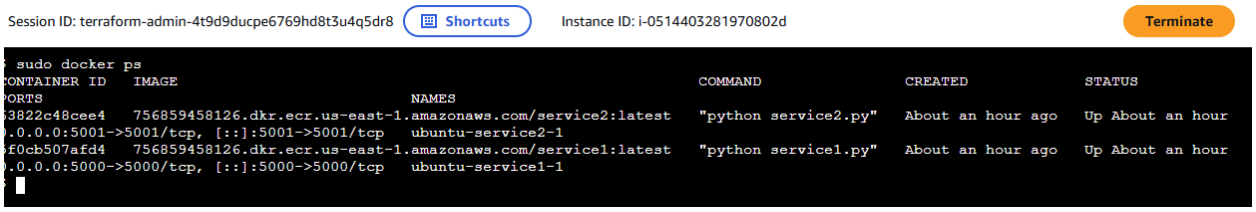
Architecture Diagram:



Deliverables

Docker PS

Docker ps on instance 1



Docker ps on instance 2

ALB DNS : devops-poc-alb-2125543003.us-east-1.elb.amazonaws.com

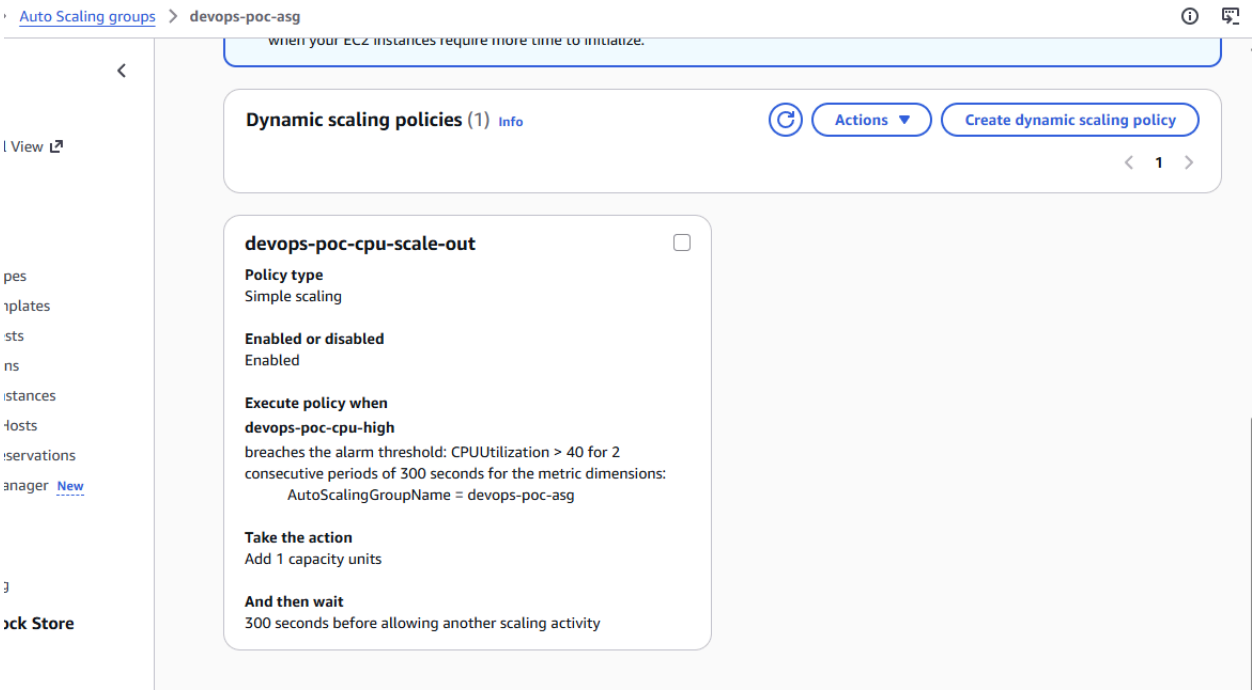
Curl for service 1 output

```
C:\Users\Pranay Chowdary>curl devops-poc-alb-2125543003.us-east-1.elb.amazonaws.com/service1/
{"message":"Hello from Service 1","user_info":"No user info provided"}

C:\Users\Pranay Chowdary>curl devops-poc-alb-2125543003.us-east-1.elb.amazonaws.com/service1/health
{"status":"healthy"}

C:\Users\Pranay Chowdary>curl devops-poc-alb-2125543003.us-east-1.elb.amazonaws.com/service1/service1
{"message":"Hello from Service 1","user_info":"No user info provided"}
```

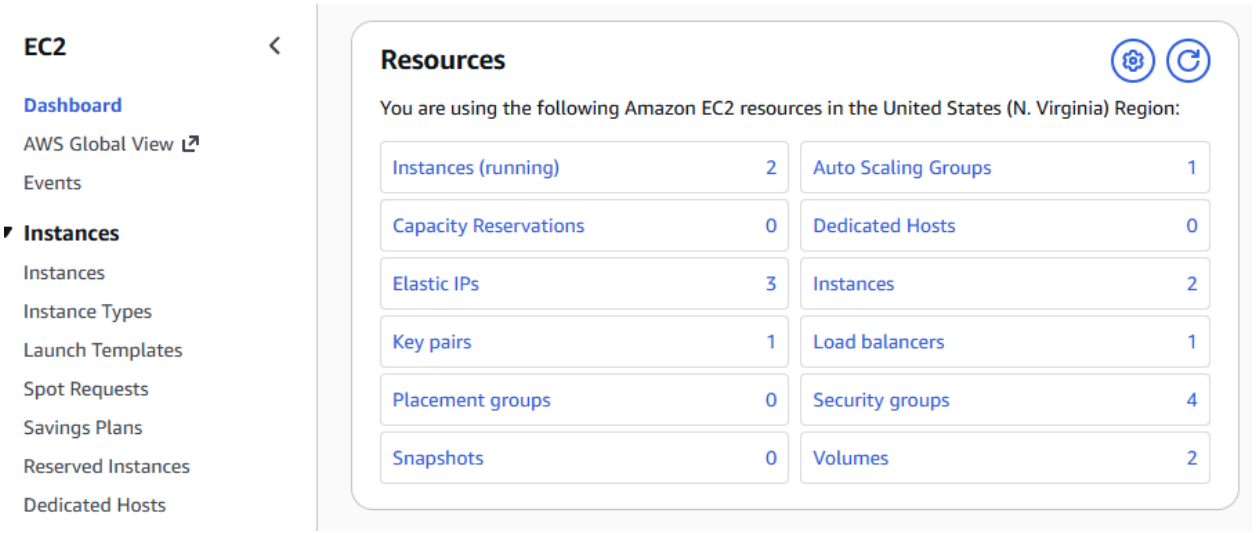
Curl for service2 output

```
C:\Users\Pranay Chowdary>curl devops-poc-alb-2125543003.us-east-1.elb.amazonaws.com/service2
{"message":"Hello from Service 2","user_info":"No user info provided"}

C:\Users\Pranay Chowdary>curl devops-poc-alb-2125543003.us-east-1.elb.amazonaws.com/service2/health
{"status":"healthy"}

C:\Users\Pranay Chowdary>curl devops-poc-alb-2125543003.us-east-1.elb.amazonaws.com/service2/service2
{"message":"Hello from Service 2","user_info":"No user info provided"}
```

Auto Scaling Group Scaleout policy



AWS EC2 Dashboard proof of running the specified resources.



Validation Script Running Proof After terraform apply

```
F:\phynxlabsDevops\devops-poc [master +17 ~0 -0 !]> terraform apply -replace="terraform_data.deployment_validation"


terraform_data.deployment_validation (local-exec): --- Starting Validation for ALB: devops-poc-alb-2125543003.us-east-1.elb.amazonaws.com ---
terraform_data.deployment_validation (local-exec): [1/3] Waiting for ALB and containers to warm up...
terraform_data.deployment_validation (local-exec):   [SUCCESS] All endpoints are responding!

terraform_data.deployment_validation (local-exec): [2/3] Running detailed endpoint checks...
terraform_data.deployment_validation (local-exec):   [OK] Service1 health check passed
terraform_data.deployment_validation (local-exec):   [INFO] Service1 Metric: index=0
terraform_data.deployment_validation (local-exec):   [INFO] Service1 Metric: service1=0
terraform_data.deployment_validation (local-exec):   [OK] Service2 health check passed
terraform_data.deployment_validation (local-exec):   [INFO] Service2 Metric: index=0
terraform_data.deployment_validation (local-exec):   [INFO] Service2 Metric: service2=0

terraform_data.deployment_validation (local-exec): [3/3] Checking ASG Infrastructure...
terraform_data.deployment_validation (local-exec):   [INFO] ASG 'devops-poc-asg': 2 healthy instance(s) found

terraform_data.deployment_validation (local-exec): Deployment Verified Successfully
terraform_data.deployment_validation: Creation complete after 5s [id=2dfd8369-9a0b-1cdc-98ad-63b0d0cf51ea]

Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
F:\phynxlabsDevops\devops-poc [master +17 ~0 -0 !]>
```

Docker-compose.yml used in the launch template user data.

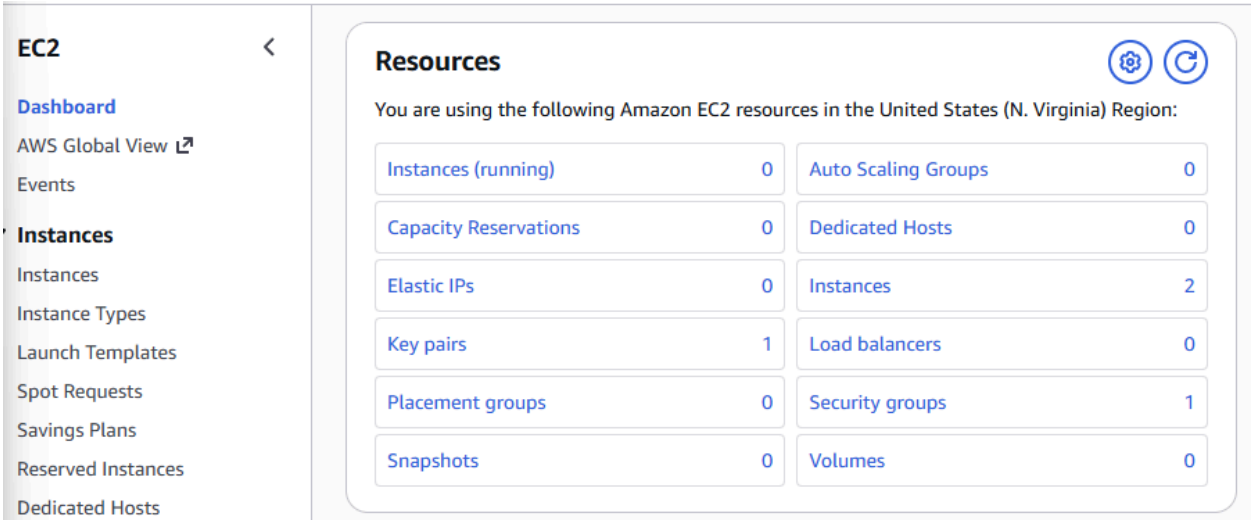
```
cat <<EOT > /home/ubuntu/docker-compose.yml
services:
  service1:
    image: "${var.account_id}.dkr.ecr.us-east-1.amazonaws.com/service1:latest"
    ports:
      - "5000:5000"
    restart: always
  service2:
    image: "${var.account_id}.dkr.ecr.us-east-1.amazonaws.com/service2:latest"
    ports:
      - "5001:5001"
    restart: always
```

Output for Terraform Destroy.

```
  # terraform_data.deployment_validation will be destroyed
  - resource "terraform_data" "deployment_validation" {
      - id               = "2dfd8369-9a0b-1cdc-98ad-63b0d0cf51ea" -> null
      - triggers_replace = {
          - alb_dns = "devops-poc-alb-2125543003.us-east-1.elb.amazonaws.com"
          - asg_id  = "devops-poc-asg"
        } -> null
    }

Plan: 0 to add, 0 to change, 31 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:
```

```
aws_lb_listener_rule.service1_rule: Destruction complete after 1s
aws_iam_role_policy_attachment.ecr_attach: Destruction complete after 1s
aws_iam_role_policy_attachment.ssm_attach: Destruction complete after 1s
aws_lb_listener_rule.service2_rule: Destruction complete after 1s
aws_lb_listener.http: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:756859458126:listener/app/
aws_cloudwatch_metric_alarm.cpu_high: Destruction complete after 1s
aws_autoscaling_policy.cpu_scale_out: Destroying... [id=devops-poc-cpu-scale-out]
aws_lb_listener.http: Destruction complete after 0s
aws_lb.app: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:756859458126:loadbalancer/app/devops
aws_route_table_association.public_assoc[0]: Destruction complete after 1s
aws_route_table_association.private_assoc[0]: Destruction complete after 1s
aws_route_table_association.private_assoc[1]: Destruction complete after 1s
aws_route_table.private: Destroying... [id=rtb-0ba50110fcede41cf]
aws_route_table_association.public_assoc[1]: Destruction complete after 1s
aws_route_table.public: Destroying... [id=rtb-09896b9175fd1babd]
aws_autoscaling_policy.cpu_scale_out: Destruction complete after 0s
aws_autoscaling_group.asg: Destroying... [id=devops-poc-asg]
aws_route_table.private: Destruction complete after 1s
aws_nat_gateway.nat: Destroying... [id=nat-08d92a6629f19b69c]
aws_route_table.public: Destruction complete after 1s
aws_internet_gateway.igw: Destroying... [id=igw-02e6434ece3f73a6a]
```

```
aws_subnet.private[1]: Destroying... [id=subnet-08658ea7f9bec9bcc]
aws_lb_target_group.service1: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:756859458126:targetgroup/devops-poc-tg1/5267aa4020d827ba]
aws_lb_target_group.service2: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:756859458126:targetgroup/devops-poc-tg2/4387df87d2135cac]
aws_subnet.private[0]: Destroying... [id=subnet-01233209345b96e40]
aws_launch_template.app: Destroying... [id=lt-078e12bf5d6d6be66]
aws_lb_target_group.service1: Destruction complete after 1s
aws_lb_target_group.service2: Destruction complete after 1s
aws_launch_template.app: Destruction complete after 1s
aws_iam_instance_profile.combined_profile: Destroying... [id=ec2-combined-profile]
aws_security_group.ec2_sg: Destroying... [id=sg-01a2f9b4d5f0be922]
aws_subnet.private[1]: Destruction complete after 1s
aws_subnet.private[0]: Destruction complete after 1s
aws_iam_instance_profile.combined_profile: Destruction complete after 1s
aws_iam_role.ec2_combined_role: Destroying... [id=ec2-microservices-role]
aws_security_group.ec2_sg: Destruction complete after 1s
aws_security_group.alb_sg: Destroying... [id=sg-01873a9be3706fdd7]
aws_iam_role.ec2_combined_role: Destruction complete after 0s
aws_security_group.alb_sg: Destruction complete after 1s
aws_vpc.main: Destroying... [id=vpc-0989bd393132dc42f]
aws_vpc.main: Destruction complete after 1s

Destroy complete! Resources: 31 destroyed.
F:\phynxlabsDevops\devops-poc [master ≡]>
```



Last 2 images shows the destroyed infrastructure proof using terraform.