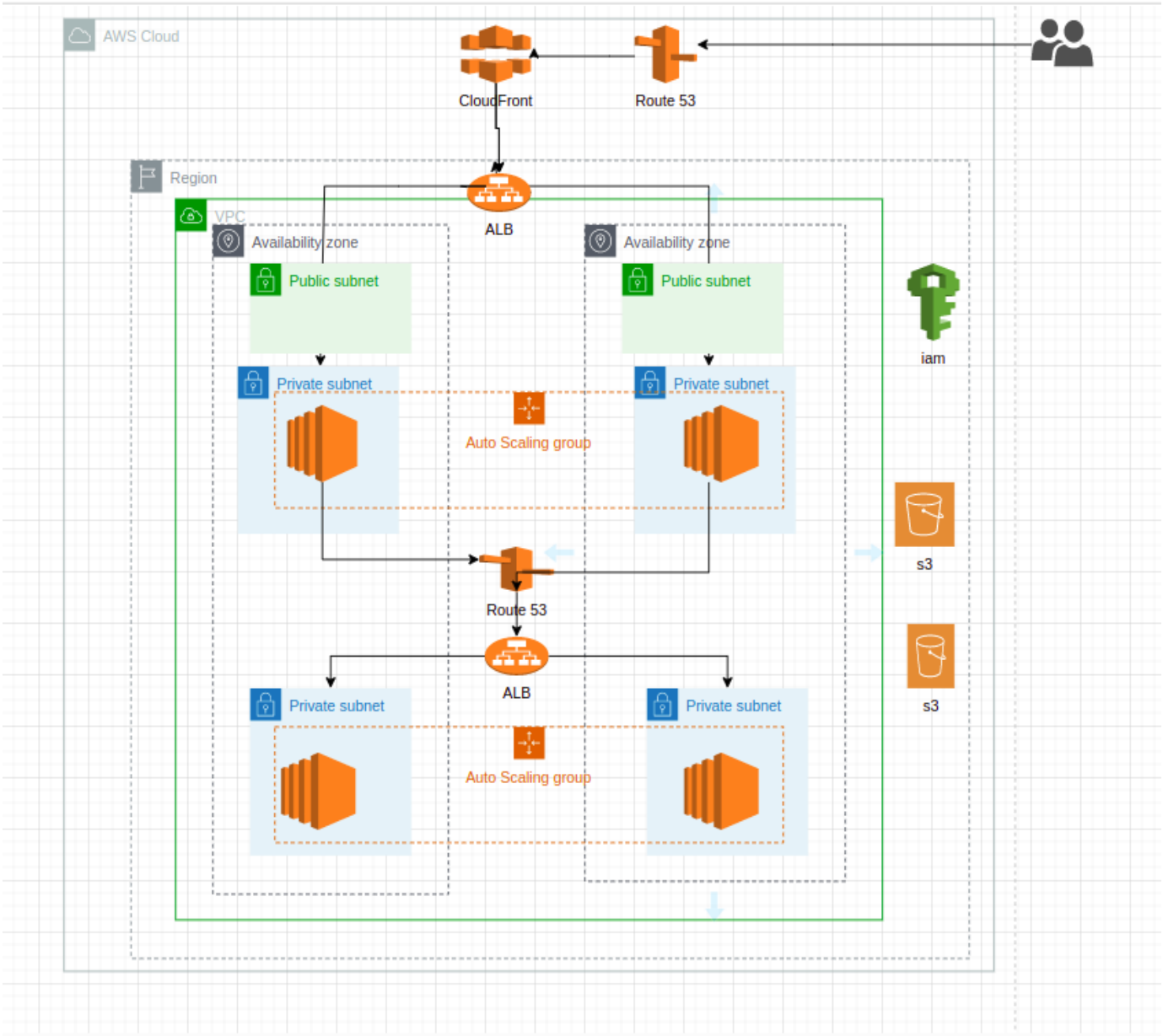
UAT- Infrastructure-



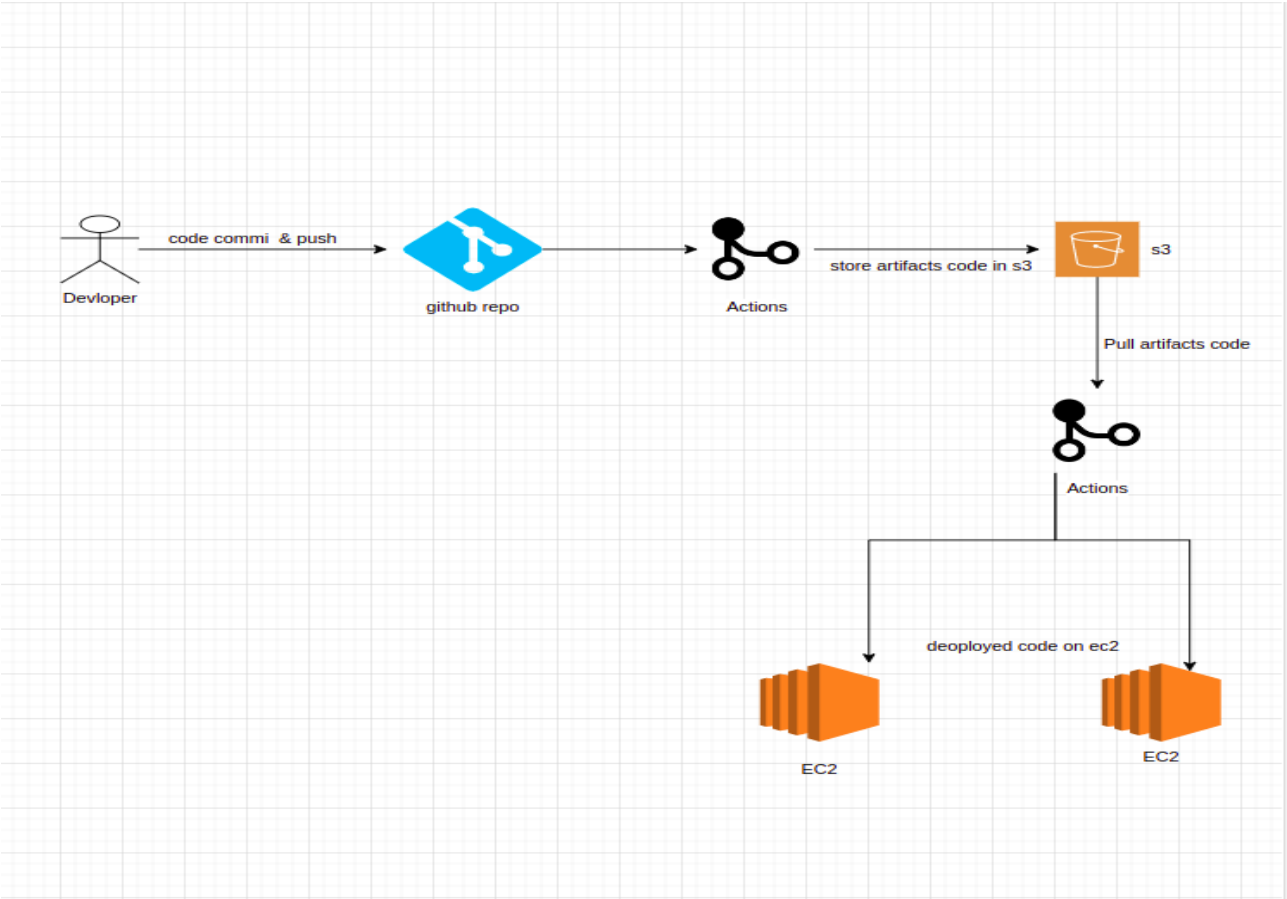
```
admin1@system:~/Downloads/task/project2_terraform_code$ terraform state list
module.asg.aws_autoscaling_group.my-asg
module.asg.aws_launch_template.my-launch-tamplate
module.asg-2.aws_autoscaling_group.my-asg
module.asg-2.aws_launch_template.my-launch-tamplate
module.cdn.aws_cloudfront_distribution.my_web_site
module.lb.aws_lb.dev-lb
module.lb.aws_lb.dev-lb-2
module.lb.aws_lb_listener.dev-lb-attach
module.lb.aws_lb_listener.dev-lb-attach-2
module.lb.aws_lb_target_group.dev-tg
module.lb.aws_lb_target_group.dev-tg-2
module.nw.data.aws_availability_zones.available
module.nw.aws_internet_gateway.dev-igw
module.nw.aws_route_table.dev-rt
module.nw.aws_route_table_association.dev-rta["pub_sub-1"]
module.nw.aws_route_table_association.dev-rta["pub_sub-2"]
module.nw.aws_subnet.dev-pub-snet["pub_sub-1"]
module.nw.aws_subnet.dev-pub-snet["pub_sub-2"]
module.nw.aws_subnet.dev-pvt-snet["pvt_sub-1"]
module.nw.aws_subnet.dev-pvt-snet["pvt_sub-2"]
module.nw.aws_vpc.dev-vpc
module.route53.aws_route53_record.aws_53
module.route53.aws_route53_record.lb_2_record
module.route53.aws_route53_zone.internal_53
module.route53.aws_route53_zone.primary
module.s3.aws_s3_bucket.s3_bucket
module.s3.aws_s3_bucket_acl.nginx
module.sg.aws_security_group.sg["lb-sg"]
module.sg-2.aws_security_group.sg["asg-nginx-sg"]
module.sg-3.aws_security_group.sg["lb-tomcat-sg"]
module.sg-4.aws_security_group.sg["asg-tomcat-sg"]
admin1@system:~/Downloads/task/project2_terraform_code$
```

terraform state list

Infrasture setup using terraform

- set-up two Route 53 for routhing traffic first one send traffic to CDN and second one attached with back end ALB.
- set-up cloud front for low latancey so that user get fast response and set alias of CDN in route 53.
- Setting up two applications Load Balancer first one is frnt-end and second one is back-end and its send the traffic to the web serve
- We would be making 2 Separate target Group running different Types of Instances to Run .zip and .war file separately
- Target Would have an ASG enabled so that they can scale accordingly and would be always up and running.
- Set up two s3 to store artifact file and terraform logs and backup file
- all instances created in pvt subnet for security.
- Set-up 4 security groups for frnt end LB ,nginx ,back-end LB and tomcat.
- Created two golden ami and used in while creating launch tamplate.

CI/CD STEP :-



CI-CD workflow

- Developer commits code changes from their local repo to the GitHub repository. In this post, the GitHub action is triggered manually, but this can be automated.
- GitHub action triggers the build stage.
- GitHub action uploads the deployment artifacts to Amazon S3.
- GitHub action invokes CodeDeploy.
- Github actions downloads the artifacts from Amazon S3 and deploys to Amazon EC2 instances