Steps With ScreenShots

1. **Clone repo**
    - git clone https://github.com/sukrity1/terraform-gitops.git

```
macold@Macs-MacBook-Pro devops-take-home: % git clone https://github.com/sukrity1/terraform-gitops.git

Cloning into 'terraform-gitops'...
remote: Enumerating objects: 104, done.
remote: Counting objects: 100% (104/104), done.
remote: Compressing objects: 100% (81/81), done.
remote: Total 104 (delta 14), reused 100 (delta 11), pack-reused 0 (from 0)
Receiving objects: 100% (104/104), 24.20 KiB | 4.03 MiB/s, done.
Resolving deltas: 100% (14/14), done.
macold@Macs-MacBook-Pro devops-take-home: %
```

2. **Bootstrap Dev Environment**
    - cd terraform-gitops/terraform/bootstrap
    - aws login (Best practice to test from local is to use aws configure sso)
    - terraform init

```
macold@Macs-MacBook-Pro bootstrap % terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (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.
macold@Macs-MacBook-Pro bootstrap %
```

    - terraform plan -var-file=tfvars/dev.tfvars

```
Plan: 4 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + dynamodb_table_arn  = (known after apply)
  + dynamodb_table_name = "tf-locks-dev"
  + s3_bucket_arn       = (known after apply)
  + s3_bucket_id        = (known after apply)
```

- infracost breakdown --path . --terraform-var-file="tfvars/dev.tfvars"

```
macold@Macs-MacBook-Pro bootstrap % infracost breakdown --path . --terraform-var-file="tfvars/dev.tfvars"
INFO Autodetected 1 Terraform project across 1 root module
INFO Found Terraform project main at directory . using Terraform var files tfvars/dev.tfvars

Project: main

 Name                                        Monthly Qty  Unit                Monthly Cost

 aws_dynamodb_table.tf_lock
 ├─ Write request unit (WRU)                 Monthly cost depends on usage: $0.000000625 per WRUs
 ├─ Read request unit (RRU)                  Monthly cost depends on usage: $0.000000125 per RRUs
 ├─ Data storage                             Monthly cost depends on usage: $0.25 per GB
 ├─ On-demand backup storage                 Monthly cost depends on usage: $0.10 per GB
 ├─ Table data restored                      Monthly cost depends on usage: $0.15 per GB
 └─ Streams read request unit (sRRU)         Monthly cost depends on usage: $0.0000002 per sRRUs

 aws_s3_bucket.tf_state
 └─ Standard
    ├─ Storage                               Monthly cost depends on usage: $0.023 per GB
    ├─ PUT, COPY, POST, LIST requests        Monthly cost depends on usage: $0.005 per 1k requests
    ├─ GET, SELECT, and all other requests   Monthly cost depends on usage: $0.0004 per 1k requests
    ├─ Select data scanned                   Monthly cost depends on usage: $0.002 per GB
    └─ Select data returned                  Monthly cost depends on usage: $0.0007 per GB

 OVERALL TOTAL                                                                $0.00

*Usage costs can be estimated by updating Infracost Cloud settings, see docs for other options.

──────────────────────────────
4 cloud resources were detected:
• 2 were estimated
• 2 were free

| Project | Baseline cost | Usage cost* | Total cost |
|---|---|---|---|
| main | $0.00 | - | $0.00 |
```

- terraform apply -var-file=tfvars/dev.tfvars

```
Plan: 4 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + dynamodb_table_arn  = (known after apply)
  + dynamodb_table_name = "tf-locks-dev"
  + s3_bucket_arn       = (known after apply)
  + s3_bucket_id        = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

3. **Setup Dev Environment Infra**
   - cd terraform-gitops/terraform/resources
   - eval $(aws configure export-credentials --profile default --format env)

- terraform init -backend-config=environments/dev/dev.config

```
macold@Macs-MacBook-Pro resources % terraform init -backend-config=environments/dev/dev.config

Initializing the backend...

Initializing modules...

Initializing provider plugins...
- Reusing previous version of hashicorp/external from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/null from the dependency lock file
- Using previously-installed hashicorp/external v2.3.5
- Using previously-installed hashicorp/aws v5.100.0
- Using previously-installed hashicorp/null v3.2.4

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.
```

- terraform plan -var-file=environments/dev/dev.tfvars

```
Plan: 28 to add, 0 to change, 1 to destroy.

Changes to Outputs:
  + alb_dns_name = (known after apply)
```

- infracost breakdown --path . --terraform-var-file="environments/dev/dev.tfvars"

```
macold@Macs-MacBook-Pro resources % infracost breakdown --path . --terraform-var-file="environments/dev/dev.tfvars"
 INFO Autodetected 1 Terraform project across 1 root module
 INFO Found Terraform project main at directory . using Terraform var files environments/dev/dev.tfvars

Project: main

 Name                                                    Monthly Qty  Unit                    Monthly Cost

 module.vpc.aws_nat_gateway.main
 ├ NAT gateway                                                   730  hours                         $32.85
 └ Data processed                                  Monthly cost depends on usage: $0.045 per GB

 module.alb.aws_lb.main
 ├ Application load balancer                                     730  hours                         $16.43
 └ Load balancer capacity units                    Monthly cost depends on usage: $5.84 per LCU

 module.ecs.aws_ecs_service.main
 ├ Per GB per hour                                               0.5  GB                             $1.62
 └ Per vCPU per hour                                            0.25  CPU                            $7.39

 module.dns_records.aws_route53_zone.primary
 └ Hosted zone                                                     1  months                         $0.50

 module.cloudwatch.aws_cloudwatch_log_group.main
 ├ Data ingested                                   Monthly cost depends on usage: $0.50 per GB
 ├ Archival Storage                                Monthly cost depends on usage: $0.03 per GB
 └ Insights queries data scanned                   Monthly cost depends on usage: $0.005 per GB

 module.dns_records.aws_route53_record.main["web-CNAME"]
 ├ Standard queries (first 1B)                     Monthly cost depends on usage: $0.40 per 1M queries
 ├ Latency based routing queries (first 1B)        Monthly cost depends on usage: $0.60 per 1M queries
 └ Geo DNS queries (first 1B)                      Monthly cost depends on usage: $0.70 per 1M queries

 module.ecr.aws_ecr_repository.main
 └ Storage                                         Monthly cost depends on usage: $0.10 per GB

 OVERALL TOTAL                                                                                   $58.79

*Usage costs can be estimated by updating Infracost Cloud settings, see docs for other options.

──────────────────────────────────
32 cloud resources were detected:
• 8 were estimated
• 24 were free

| Project | Baseline cost | Usage cost* | Total cost |
|---|---|---|---|
| main | $59 | - | $59 |
```

- terraform apply -var-file=environments/dev/dev.tfvars

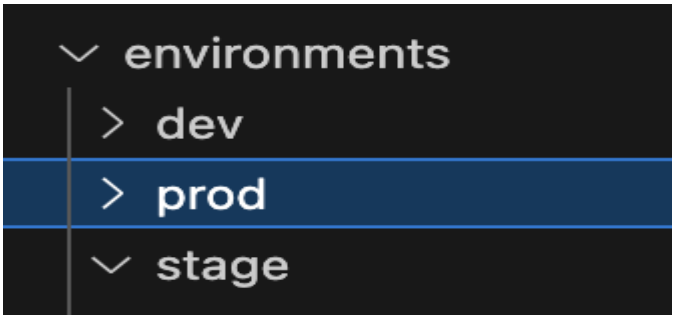
```
Plan: 28 to add, 0 to change, 1 to destroy.

Changes to Outputs:
  + alb_dns_name = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```

Note :- setup stage/prod Infra just by using configs from their respective env folders



Above steps were tested locally. Github workflow terraform.yml can be used for automatic infra updates if there are any changes in the terraform folder. I will successfully run once credentials & github secrets are configured.

4. Setup CI/CD

From Local (Tested Successfully):-

- **Build Once :-**

  docker build -t java-api -f docker/Dockerfile .

```
macold@Macs-MacBook-Pro terraform-gitops % docker build -t java-api -f docker/Dockerfile .
 [+] Building 3.4s (18/18) FINISHED                                                                                    docker:desktop-linux
  => [internal] load build definition from Dockerfile                                                                                  0.0s
  => => transferring dockerfile: 940B                                                                                                  0.0s
  => [internal] load metadata for docker.io/library/maven:3.8.5-openjdk-17                                                             2.0s
  => [internal] load metadata for docker.io/library/amazoncorretto:17-alpine                                                           1.8s
  => [internal] load .dockerignore                                                                                                     0.0s
  => => transferring context: 2B                                                                                                       0.0s
  => [build 1/5] FROM docker.io/library/maven:3.8.5-openjdk-17@sha256:3a9c30b3af6278a8ae0007d3a3bf00fff80ec3ed7ae4eb9bfa1772853101549b  0.0s
  => => resolve docker.io/library/maven:3.8.5-openjdk-17@sha256:3a9c30b3af6278a8ae0007d3a3bf00fff80ec3ed7ae4eb9bfa1772853101549b        0.0s
  => [stage-1 1/6] FROM docker.io/library/amazoncorretto:17-alpine@sha256:a75407e86b53bb4f1c309b86c2682705ea8148b41c54fd6e529a207c89542ac1  0.0s
  => => resolve docker.io/library/amazoncorretto:17-alpine@sha256:a75407e86b53bb4f1c309b86c2682705ea8148b41c54fd6e529a207c89542ac1      0.0s
  => [stage-1 3/6] ADD https://download.newrelic.com/newrelic/java-agent/newrelic-agent/current/newrelic-agent.jar /app/newrelic.jar  1.2s
  => [internal] load build context                                                                                                     0.0s
  => => transferring context: 2.88kB                                                                                                   0.0s
  => CACHED [stage-1 2/6] WORKDIR /app                                                                                                 0.0s
  => CACHED [stage-1 3/6] ADD https://download.newrelic.com/newrelic/java-agent/newrelic-agent/current/newrelic-agent.jar /app/newrelic.jar  0.0s
  => CACHED [stage-1 4/6] RUN chmod 644 /app/newrelic.jar                                                                              0.0s
  => CACHED [stage-1 5/6] RUN addgroup -S spring && adduser -S spring -G spring                                                        0.0s
  => CACHED [build 2/5] WORKDIR /app                                                                                                   0.0s
  => CACHED [build 3/5] COPY pom.xml .                                                                                                 0.0s
  => CACHED [build 4/5] COPY src ./src                                                                                                 0.0s
  => CACHED [build 5/5] RUN mvn clean package -DskipTests                                                                              0.0s
  => CACHED [stage-1 6/6] COPY --from=build /app/target/*.jar app.jar                                                                  0.0s
  => exporting to image                                                                                                                0.1s
  => => exporting layers                                                                                                               0.0s
  => => exporting manifest sha256:26f410b76d3aec588222f597e3565df71a0121b4f44e6cf39e95e57e515d470e                                     0.0s
  => => exporting config sha256:b4c1c8355d158eb041a182dacbe2752f4732d409dd6cdc966ebc52957d9f696c                                       0.0s
  => => exporting attestation manifest sha256:e70f85d87ee19d8dddfa4e7bc83188fd735e612afcfe6127ad70d58ba77cb66e                         0.0s
  => => exporting manifest list sha256:87ed9cff0067cdd73a93f768c5fd40e274db0a63f1d0f69f031a95ccae246fc8                                0.0s
  => => naming to docker.io/library/java-api:latest                                                                                    0.0s
  => => unpacking to docker.io/library/java-api:latest                                                                                 0.0s
```

- **Deploy linearly on all env :-**

  - [ ] export NEW_RELIC_LICENSE_KEY="<XXX>"

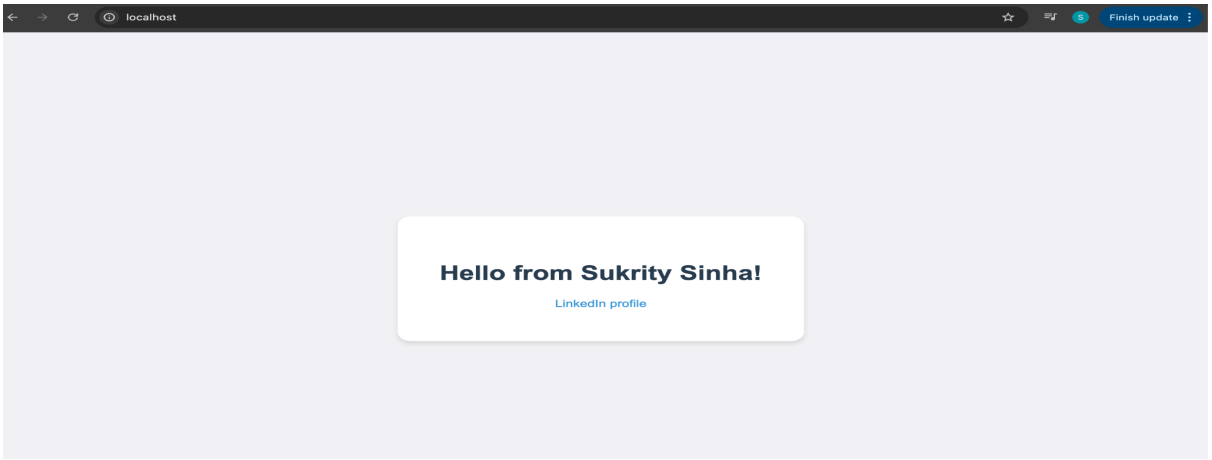
  - [ ] docker run -e NEW_RELIC_LICENSE_KEY="${NEW_RELIC_LICENSE_KEY}" --name java-app -p 80:8080 my-java-api

App Startup Logs

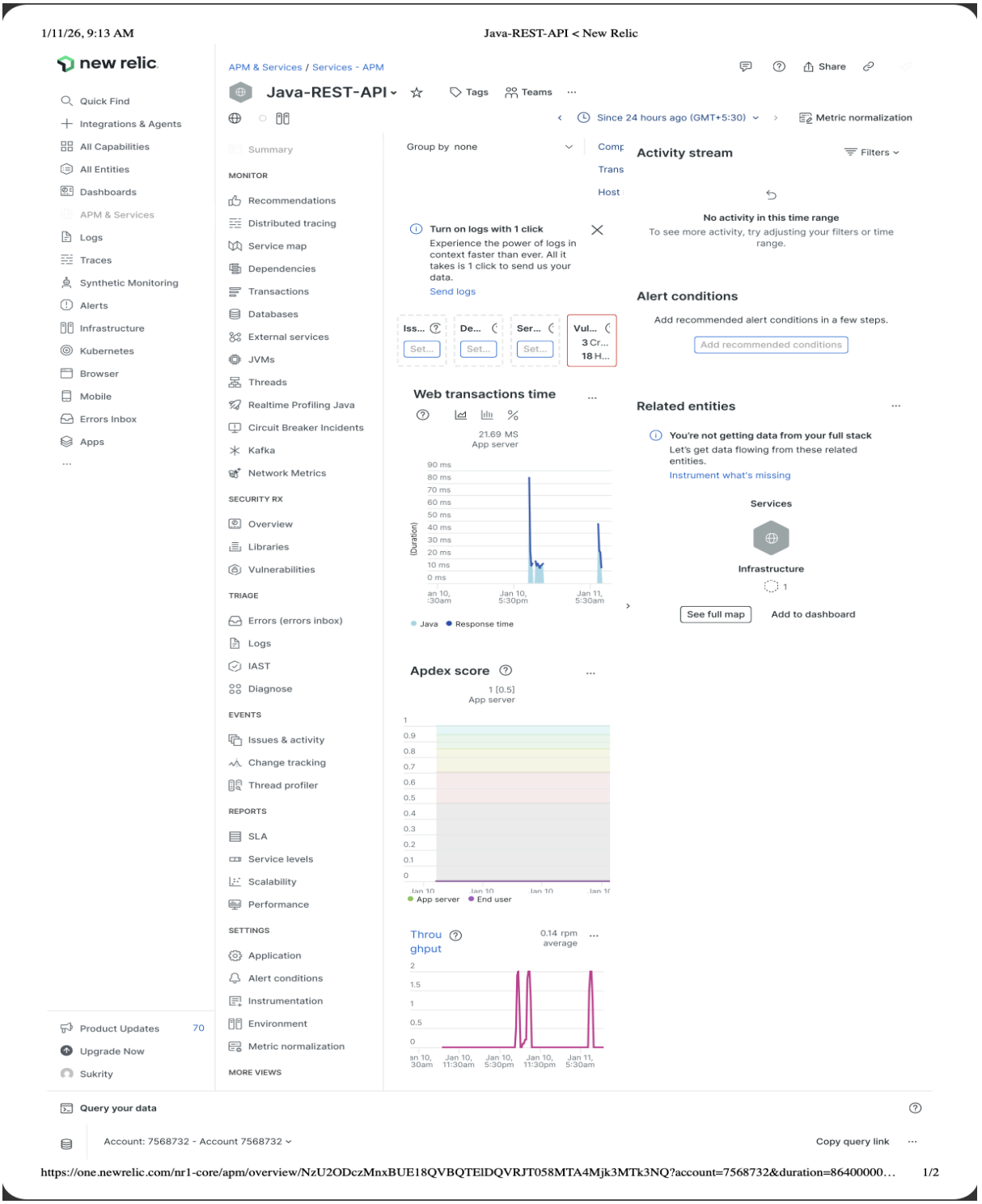
```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.1.5)

2026-01-11T03:38:53.062Z  INFO 1 --- [           main] com.example.DemoApplication              : Starting DemoApplication v0.0.1-SNAPSHOT using Java 1
7.0.17 with PID 1 (/app/app.jar started by spring in /app)
2026-01-11T03:38:53.085Z  INFO 1 --- [           main] com.example.DemoApplication              : No active profile set, falling back to 1 default prof
ile: "default"
2026-01-11T03:38:54.371Z  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2026-01-11T03:38:54,392+0000 [1 1] com.newrelic INFO: Server Info: Apache Tomcat/10.1.15
2026-01-11T03:38:54.506Z  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2026-01-11T03:38:54.507Z  INFO 1 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.15]
2026-01-11T03:38:54.643Z  INFO 1 --- [           main] o.a.c.c.c.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2026-01-11T03:38:54.644Z  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed
in 1493 ms
2026-01-11T03:38:55.250Z  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context p
ath ''
2026-01-11T03:38:55.262Z  INFO 1 --- [           main] com.example.DemoApplication              : Started DemoApplication in 2.731 seconds (process run
ning for 5.809)
2026-01-11T03:38:56,757+0000 [1 39] com.newrelic INFO: Host name is 05df95446e4d, display host 05df95446e4d for application Java-REST-API
2026-01-11T03:38:56,766+0000 [1 39] com.newrelic INFO: New Relic JFR Monitor is disabled: JFR config has not been enabled in the Java agent.
2026-01-11T03:38:58,066+0000 [1 39] com.newrelic INFO: Collector redirection to collector.newrelic.com:443
2026-01-11T03:38:59,431+0000 [1 39] com.newrelic INFO: Max payload size is 1,000,000 bytes
2026-01-11T03:38:59,431+0000 [1 39] com.newrelic INFO: Agent run id: BVoj3SmAH_ueAHN9XEEwb11pYxuQAAIBACdHIQEAAAABAgRAjNdHBd-VRG5NJLeICMn09f00a6MS1PYq_G
gu1O-iiE1KiI4DAAY4LjI1LjEADDA1ZGY5NTQ0NmU02ZAANSmF2YS1SRVNULUFQSO
2026-01-11T03:38:59,432+0000 [1 39] com.newrelic INFO: Agent 1@05df95446e4d/Java-REST-API connected to collector.newrelic.com:443
2026-01-11T03:38:59,433+0000 [1 39] com.newrelic INFO: Reporting to: https://rpm.newrelic.com/accounts/7568732/applications/1082971975
2026-01-11T03:38:59,434+0000 [1 39] com.newrelic INFO: Using default collector host: collector.newrelic.com
2026-01-11T03:38:59,434+0000 [1 39] com.newrelic INFO: Using default metric ingest URI: https://metric-api.newrelic.com/metric/v1
2026-01-11T03:38:59,434+0000 [1 39] com.newrelic INFO: Using default event ingest URI: https://insights-collector.newrelic.com/v1/accounts/events
2026-01-11T03:38:59,436+0000 [1 39] com.newrelic INFO: The root sampler was configured to use the adaptive sampler type.
2026-01-11T03:38:59,436+0000 [1 39] com.newrelic INFO: The remote_parent_sampled sampler was configured to use the adaptive sampler type.
2026-01-11T03:38:59,436+0000 [1 39] com.newrelic INFO: The remote_parent_not_sampled sampler was configured to use the adaptive sampler type.
2026-01-11T03:38:59,439+0000 [1 39] com.newrelic INFO: Using default collector host: collector.newrelic.com
2026-01-11T03:38:59,439+0000 [1 39] com.newrelic INFO: Using default metric ingest URI: https://metric-api.newrelic.com/metric/v1
2026-01-11T03:38:59,439+0000 [1 39] com.newrelic INFO: Using default event ingest URI: https://insights-collector.newrelic.com/v1/accounts/events
2026-01-11T03:38:59,444+0000 [1 39] com.newrelic INFO: The root sampler was configured to use the adaptive sampler type.
2026-01-11T03:38:59,444+0000 [1 39] com.newrelic INFO: The remote_parent_sampled sampler was configured to use the adaptive sampler type.
2026-01-11T03:38:59,444+0000 [1 39] com.newrelic INFO: The remote_parent_not_sampled sampler was configured to use the adaptive sampler type.
2026-01-11T03:39:00,822+0000 [1 39] com.newrelic INFO: Real user monitoring is enabled for application Java-REST-API. Auto instrumentation is enabled.
2026-01-11T03:39:00,823+0000 [1 39] com.newrelic INFO: Updating shared Adaptive Sampler sampling target to 120
2026-01-11T03:39:00,824+0000 [1 39] com.newrelic INFO: New Relic JFR Monitor is disabled: JFR config has not been enabled in the Java agent.
```

App UI



NewRelic APM



From Github Actions :- Will be successful post terraform apply which will deploy ecr and all environments infra.