1. Ensure that the virtual machines have been created and are running without any errors after the provisioning.

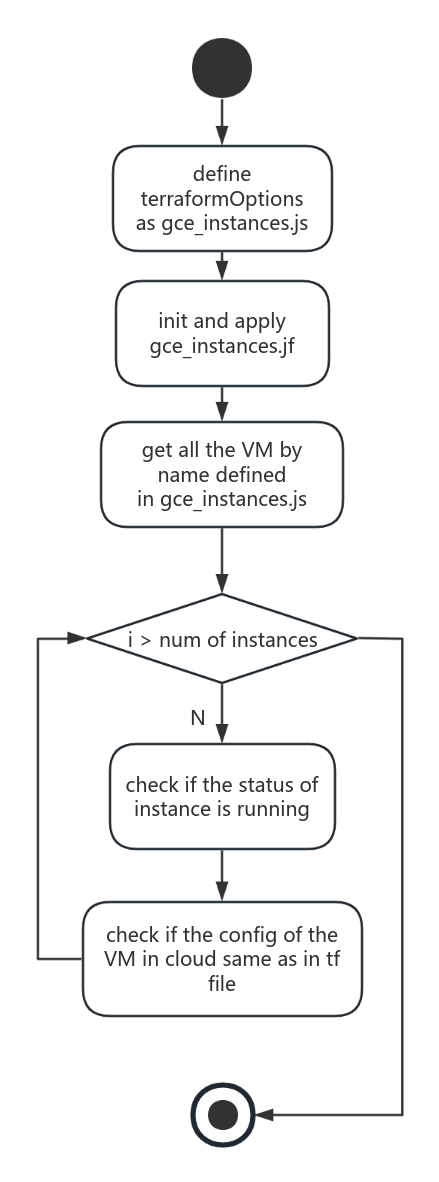
Step 1:Use Terratest to Init and Apply Terraform for VMs

Step 2:Retrieve All VMs Using FetchInstanceE(): Utilize the `FetchInstanceE()` function from `github.com/gruntwork-io/terratest/modules/gcp` to fetch all Compute Instances under the project, one by one.

Step 3:Verify VMs Running Status: Use `assert.Equal(t, "RUNNING", instance.Status, "Instance %s is not running", instance.Name)` to check if each VM is running correctly.

Step 4:Ensure Configuration Consistency: Additionally, you can utilize the objects returned by `FetchInstanceE()` to verify if the VM configurations match those defined in the `gce\_instances.tf` file.

Step 5:Defer destroy of Terraform resources after the test



1. Ensure that the virtual machines have internet connectivity.

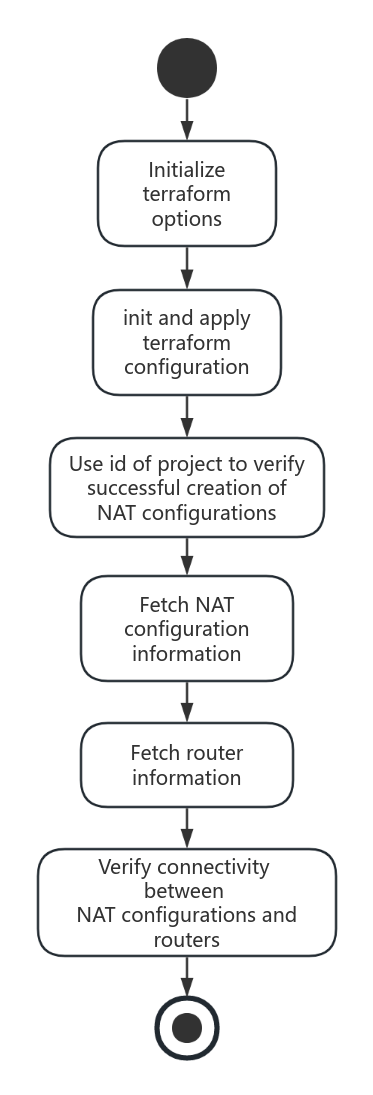
Step 1:Use Terratest to Init and Apply Terraform for Could NAT

Step 2:Verify Successful Creation of NAT Configurations:assertNATConfiguration()

Step 3: Fetch NAT Configuration Information: gcp.FetchNATConfigE()

Step 4: Verify Connectivity between NAT Configurations and Routers:FetchRouterE()

assert.Equal(t, router.Name, natConfig.Router, "NAT configuration %s is not connected to router %s", natName, routerName)



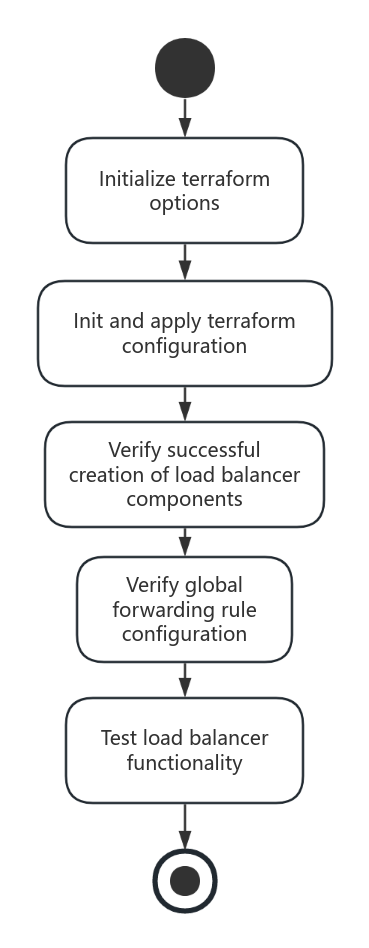
1. Ensure that the load balancer is provisioned and working properly

Step 1:Use Terratest to Init and Apply Terraform for Could NAT

Step 2: Verify Successful Creation of Load Balancer Components

Step 3: Verify Global Forwarding Rule Configuration

Step 4: Test Load Balancer Functionality:perform tests to verify that the load balancer is functioning as expected, such as accessing the load balancer's external IP address and observing traffic distribution to backend instances.

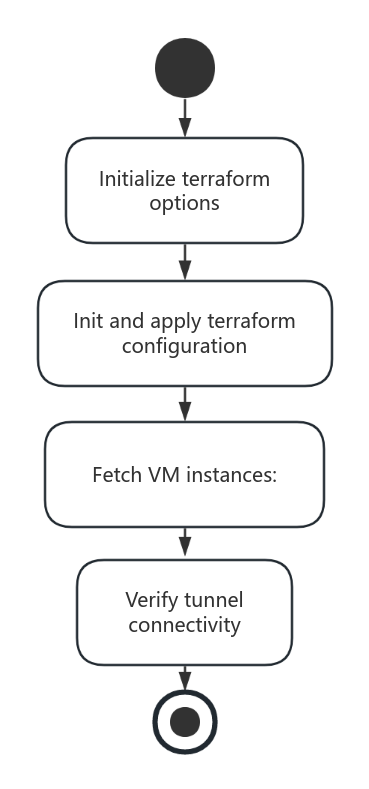


1. Ensure that an end user can tunnel into the VM using IAP.

Step 1:Use Terratest to Init and Apply Terraform for Could NAT

Step 2:Fetch VM Instances

Step 3:Verify Tunnel Connectivity assert.NotNil(t, instance.NetworkInterfaces[0].AccessConfigs[0].NatIP)



1. Ensure that the system is resilient.

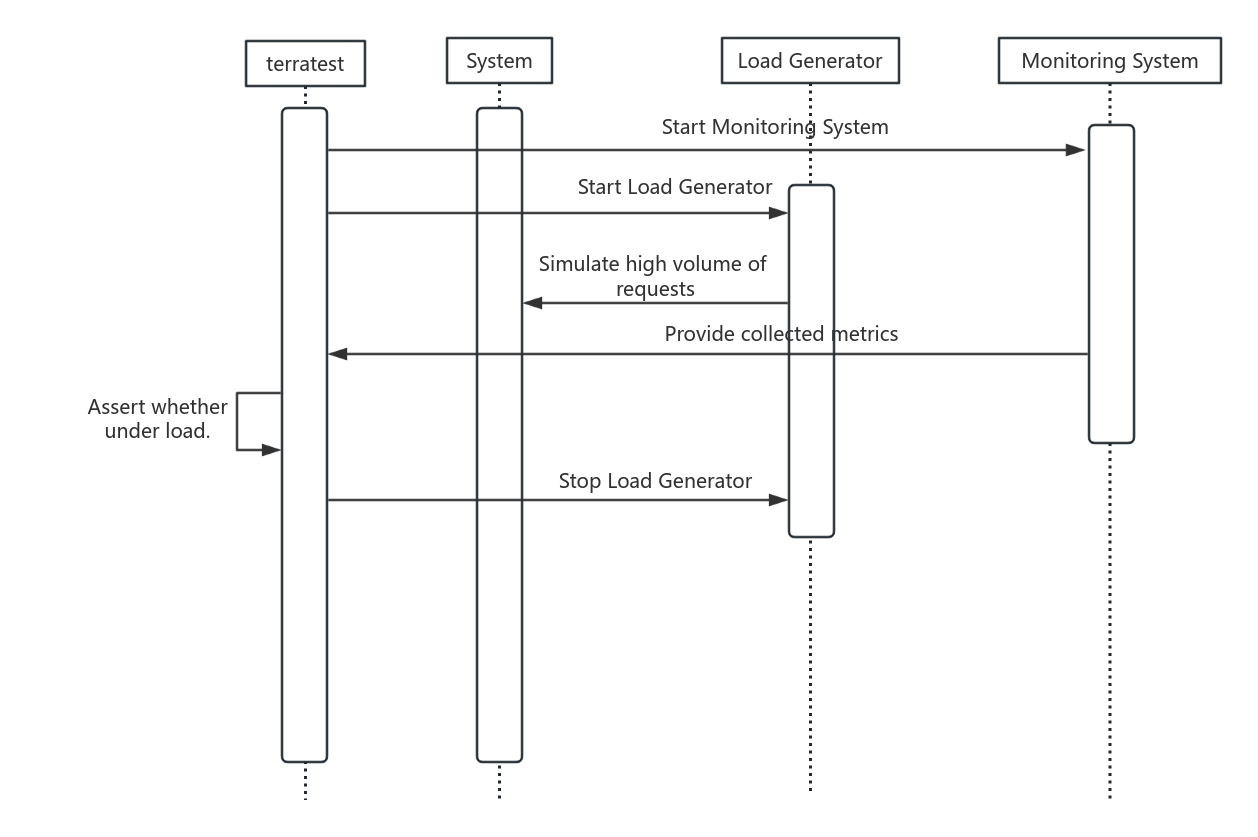
a. Load Testing:

- Start a load generator to simulate a high volume of concurrent requests.

- Monitor system metrics such as response time and throughput under the load.

- Assert that the system response time and throughput are within acceptable ranges.

- Stop the load generator.



b. Failure Recovery Testing:

- Simulate a failure scenario, such as stopping a virtual machine instance.

- Allow some time for the system to detect and recover from the failure.

- Check if the system automatically recovers from the failure.

- Recover the affected resources to ensure other tests are not impacted.

