## Solution Approach explained

* A private isolated network which would best suit Symbiosis’s 2 tier architecture needs. In order to meet their internal SLAs, they require a highly available solution as well.
  + A private VPC needs to be constructed with two subnets for two tier architecture
* Symbiosis being a B2C company, would typically like its web applications to be accessible over the internet and thus handle HTTP traffic.
  + This suggests that one subnet should be public facing & another should be private
  + It also means that we have to have NAT Gateway in public subnet so that private subnet can’t be accessed directly from internet
* The database tier should have restricted access (not open to HTTP) and allow traffic only through the web tier.
  + It tells us that security group for database instance in private subnet should have only database port open
* They would like to reduce the administrative burden of managing their SQL database and require a managed database for their SQL engine in the proposed solution. They need the database to be highly available.
  + It means that we should be using a managed service from AWS end. For RDBMS, we have AWS RDS which can be configured for this use case
  + Since they want high availability so RDS will be configured for master instance & read replica instance
* Currently they experience medium to high traffic on their network. How can they scale their application to meet demand?
  + Autoscaling will be configured and hence need target group & application load balancer
* Effective distribution of load
  + Load balancer level traffic distribution policy has to be set
* Recommend logging metrics collection and monitoring
  + CloudWatch needs to be configured
* Recommend a DevOps pipeline
  + Pipeline has been put in repo
* Recommend types of testing methodology and tools (if applicable) Suggest software packaging, distribution and deployment mechanism
  + SonarQube
  + Unit testing
  + Nexus
  + These tools can be used for testing packaging & dependency management
* What would you suggest for Infrastructure as code, testing and deploying changes
  + Terraform is suggested for infrastructure automation & Jenkins is suggested for devOps pipeline execution which will also take care of testing & deploying changes to production