## **Success Criteria for Elastic Load Balancer Provisioning with CloudFormation across Environments**

**Here are some success criteria for your team's effort to provision Elastic Load Balancers (ELBs) using CloudFormation across development, higher environment, and production environments:**

**Functional:**

* **Successful deployment of ELBs in all environments:** Each environment should have an operational ELB configured according to its respective parameters and security controls.
* **Traffic routing:** ELBs should route traffic to the corresponding backend instances or resources as intended.
* **Scalability:** ELBs should be able to handle expected traffic loads and scale automatically if needed.
* **Health checks:** ELBs should perform health checks on backend instances and automatically remove unhealthy ones from the pool.

**Non-functional:**

* **Code reusability:** CloudFormation templates should be modular and reusable across environments with minimal changes.
* **Version control:** CloudFormation templates should be version-controlled for easy tracking and rollback.
* **Deployment process:** The deployment process should be automated and repeatable using CI/CD pipelines.( if possible but not mandatory at this stage)
* **Security compliance:** ELBs should be configured to comply with applicable security standards and controls for each environment.
* **Testing:** ELBs should be thoroughly tested in each environment before deployment to production.

**Steps for Provisioning ELB using CloudFormation:**

1. **Define environment-specific parameters:** Identify the parameters that need to be different for each environment, such as instance types, security groups, health check configurations, and ALB listeners.
2. **Develop modular CloudFormation templates:** Create separate templates for common resources like VPCs, subnets, and security groups, and then reference them in the ELB template. Use parameters and conditional logic to handle environment-specific configurations.
3. **Implement security controls:** Configure ELBs with appropriate security controls, such as access control lists (ACLs), security groups, and certificate validation. Consider different security requirements for each environment.
4. **Testing:**Develop automated tests that verify the functionality and health of ELBs in each environment.
5. **Documentation:** Document the CloudFormation templates, deployment process, and security controls for future reference.

## **Official References from AWS:**

* **AWS CloudFormation Documentation:** <https://docs.aws.amazon.com/cloudformation/>
* **AWS Elastic Load Balancing:** <https://docs.aws.amazon.com/elasticloadbalancing/>
* **AWS CloudFormation Best Practices:** <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/best-practices.html>
* **AWS Well-Architected Framework:** <https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>

## **Additional Tips:**

* Consider using AWS CloudFormation Designer to visually create and edit your CloudFormation templates.
* Utilize CloudFormation parameters to make your templates more flexible and reusable.
* **Leverage existing cloud formation template in AWS Account**
* Monitor your ELBs using Amazon CloudWatch and set up alarms for potential issues.