

Shopalyst Demo Webapp

AWS CloudFormation Project Documentation

Project Title: Scalable & Secure Web Application Infrastructure using AWS CloudFormation

Submitted To: Shopalyst

Submitted By: Jitheesh PJ

Submission Date: 3rd August 2025

1. Project Objective

To automate the deployment of a secure, scalable, and highly available web application infrastructure using AWS CloudFormation with nested stacks, incorporating best practices for networking, security, monitoring, and auto-scaling.

2. Architecture Overview

Components:

VPC Stack: Custom VPC with 2 Public and 2 Private subnets, Internet Gateway, NAT Gateway, Route Tables

Compute Stack: ALB, Launch Template, Auto Scaling Group, Target Group, EC2 Instance (Amazon Linux 2 with NGINX)

Database Stack: RDS (MySQL) with Multi-AZ, Secret Manager Integration

S3 Bucket: For storing key pair and parameter files

CloudWatch: Logs, Alarms, Auto Scaling Policies, Email Notifications

3. Security Considerations

EC2 is in private subnet, traffic flows through ALB

RDS credentials stored securely in AWS Secrets Manager

IAM roles scoped with least privilege

Key pair securely created and stored in S3 (Keys/)

4. Automation Strategy

Entire deployment managed via master.yaml as root stack

Nested stacks used for modular and reusable templates:

vpc.yaml

compute.yaml

rds.yaml

secrets.yaml (if applicable)

5. Parameter Management

env.json used for parameter values

Referenced in CLI or passed through ParameterFile S3 object

Template URLs for nested stacks optionally passed or defaulted

6. Monitoring and Scaling

CloudWatch Alarms based on CPU Utilization

Scale Out: > 70% for 5 mins

Scale In: < 30% for 5 mins

Email notifications on alarm using SNS

7. Health Checks & ASG Behavior

ASG uses ELB health check

ALB target group checks / on port 80 (served by NGINX)

On failure: unhealthy instance is terminated and replaced

8. User Data (EC2 Bootstrap)

Shell script to install NGINX

Serves a custom index.html with branding (Shopalyst-style)

9. Outputs

ALB DNS Name

RDS Endpoint

RDS Secret ARN

VPC and Subnet IDs

KeyPair location in S3

10. Challenges Faced

Handling SecretManager integration due to password validation rules

Avoiding direct user input during stack creation (used env.json + default param handling)

ASG replacement logic on target group health check

EC2 in private subnet requiring ALB for web access

User data verification (via log and NGINX status)

11. References

AWS Official Documentation:

CloudFormation User Guide

Auto Scaling with ALB

Secret Manager Integration

Forums, GitHub Repos, and AWS Blog Posts

12. Testing Checklist

[x] NGINX accessible via ALB DNS

[x] EC2 recreated on NGINX failure

[x] DB credentials securely stored

[x] ASG triggers scaling based on CPU

[x] Email sent on alarm breach

13. Project Folder Structure

```
|— Project/
|   |— master.yaml
|   |— vpc.yaml
|   |— compute.yaml
|   |— rds.yaml
|
|   |— scripts/
|   |   |— user-data-script.sh
|   |   |— index.html
|   |
|   |— config/
|   |   |— env.json
|   |
|   |— README.md
```

14. Notes

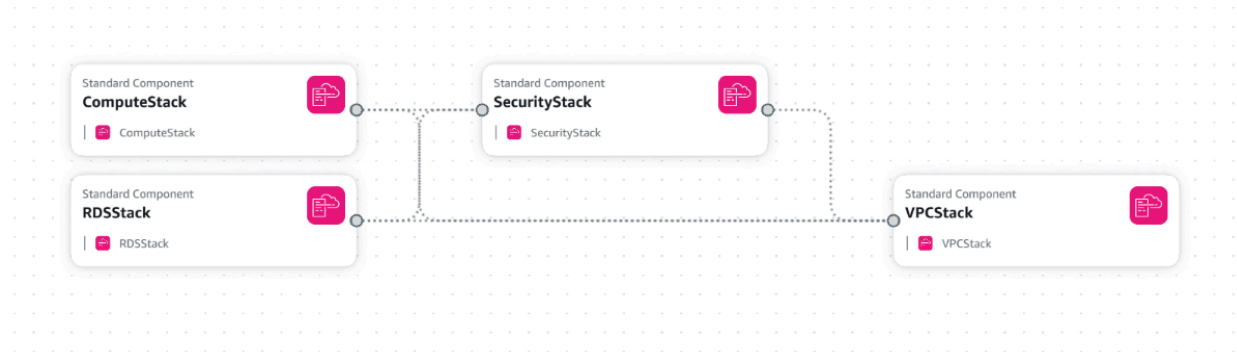
Parameter file should be updated for new resource names or tags

Use CLI or automation tools to trigger deployment

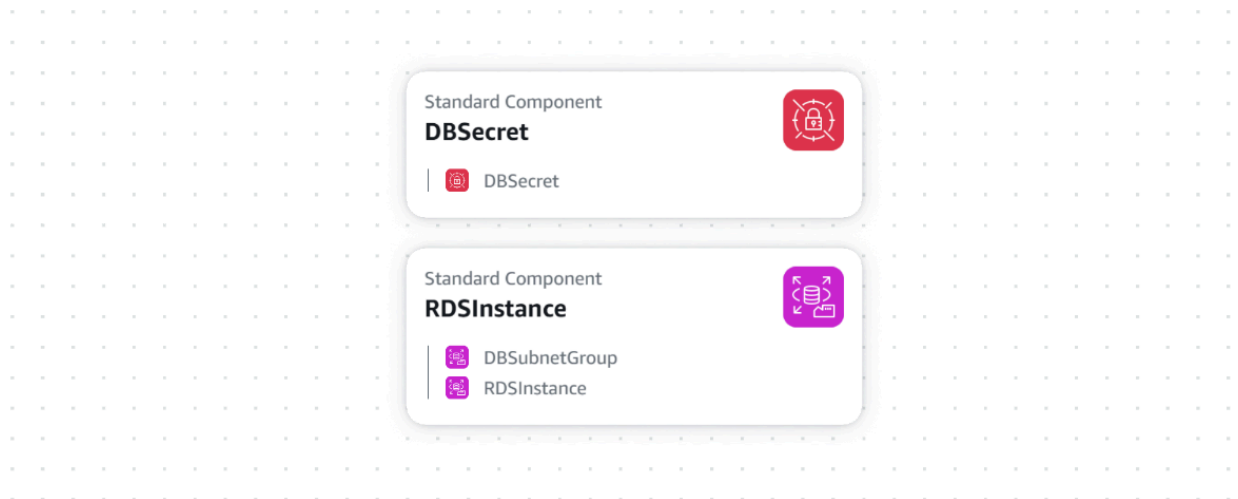
All stacks can be deleted via master.yaml rollback

Architecture Diagram:

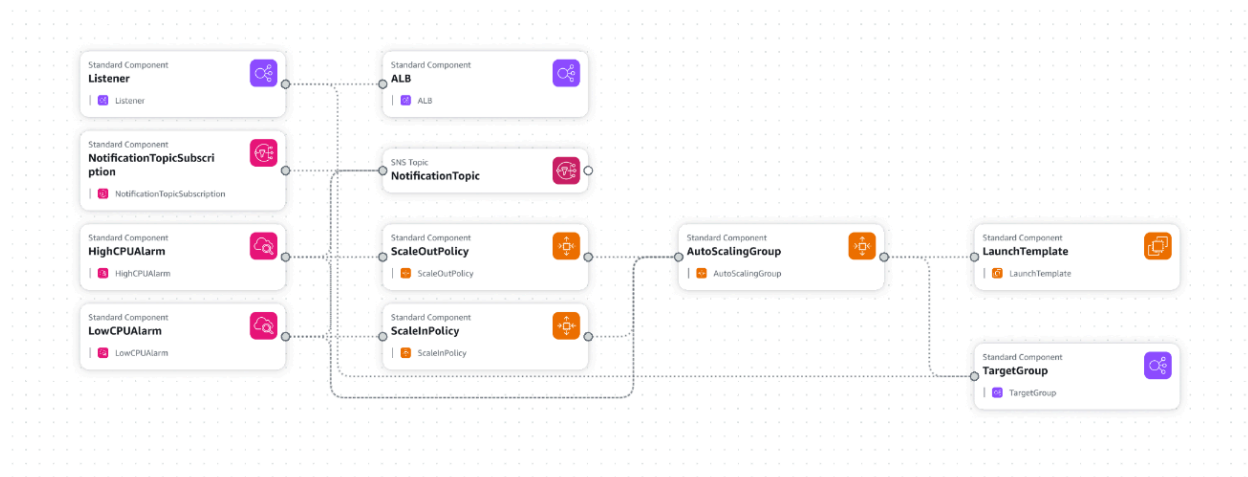
Master Stack : Shopalyst



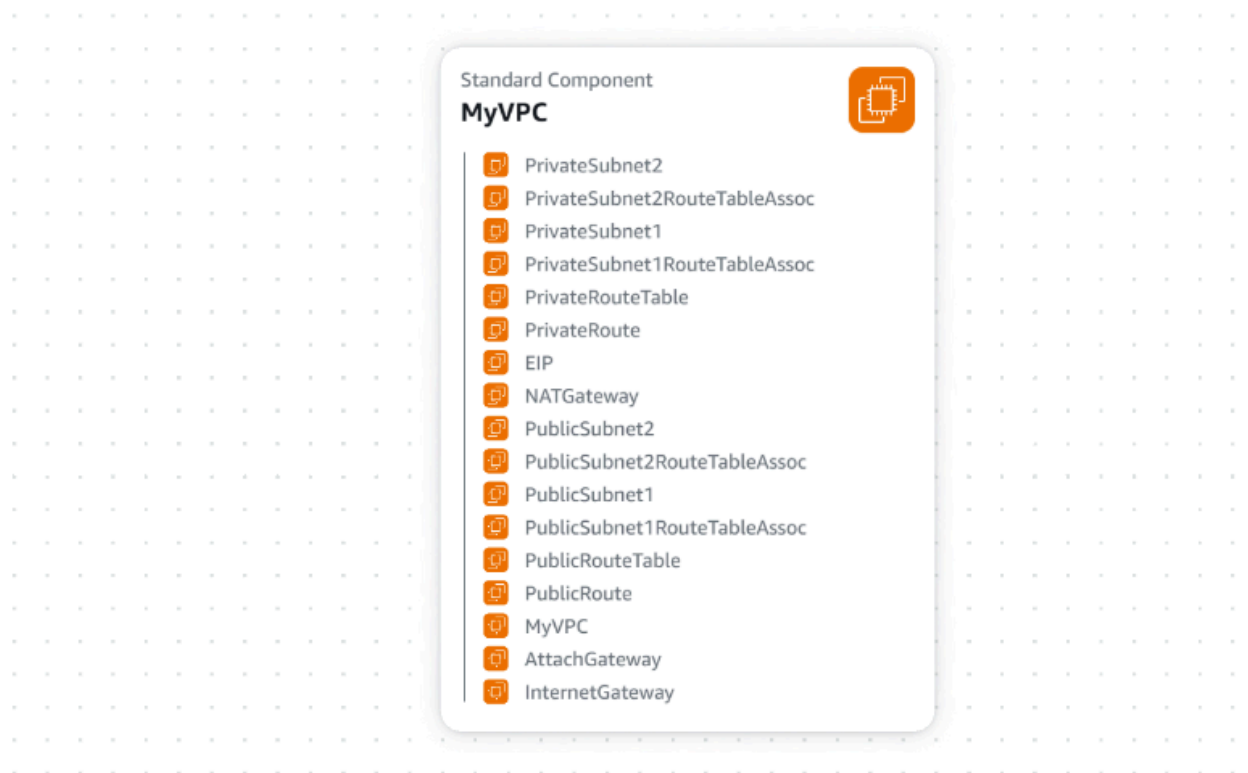
RDS Stack



Compute Stack:



VPC Stack:



Security Stack :

