



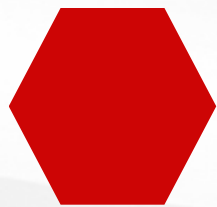
CASE STUDY: E-COMMERCE PORTAL DEPLOYMENT

Problem Statement

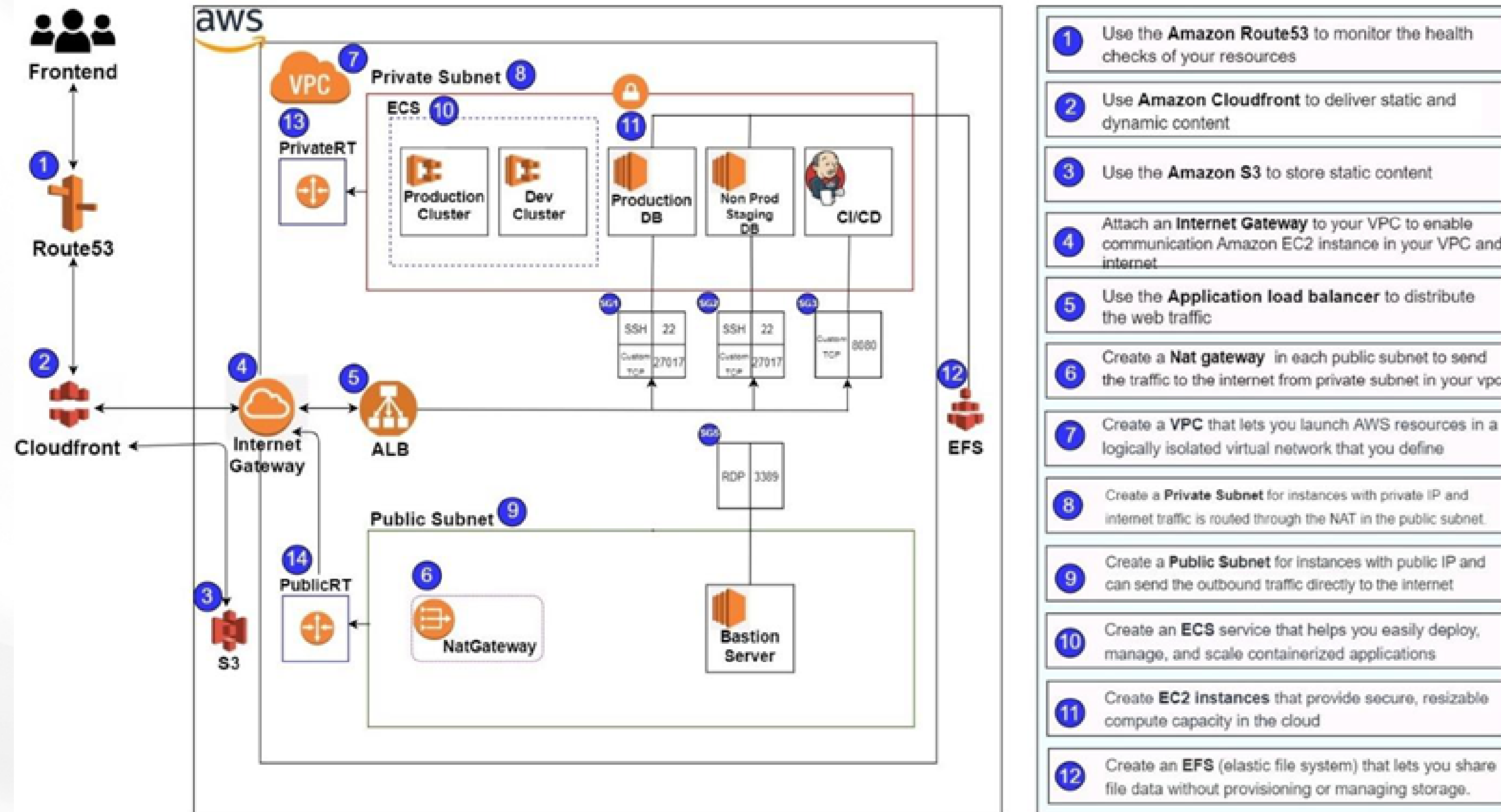
- This is for e-Commerce Marketplace company for software vendors and retailers to showcase and sell their software products. As an startup they wanted to build infrastructure which can scale with demand and very secure. They also need an automated build deploy pipeline for faster deployment.

Solution:

- Defining the target deployment and technical architecture
- Setting up AWS cloud infrastructure for new application in AWS ECS, Mongo DB hosted on Ec2.
- Setting up build and deployment pipeline using Jenkins for microservices and web application deployment
- Integration code analysis tool Sonar, Integration Testing Cypress etc. in the CI/CD pipeline.
- Monitoring the cloud infrastructure and build the Dashboard using Grafana, CloudWatch and other services.
- Log aggregation using Elastic Search



Deployment Diagram



- 1 Use the **Amazon Route53** to monitor the health checks of your resources
- 2 Use **Amazon Cloudfront** to deliver static and dynamic content
- 3 Use the **Amazon S3** to store static content
- 4 Attach an **Internet Gateway** to your VPC to enable communication Amazon EC2 instance in your VPC and internet
- 5 Use the **Application load balancer** to distribute the web traffic
- 6 Create a **Nat gateway** in each public subnet to send the traffic to the internet from private subnet in your vpc
- 7 Create a **VPC** that lets you launch AWS resources in a logically isolated virtual network that you define
- 8 Create a **Private Subnet** for instances with private IP and internet traffic is routed through the NAT in the public subnet.
- 9 Create a **Public Subnet** for instances with public IP and can send the outbound traffic directly to the internet
- 10 Create an **ECS** service that helps you easily deploy, manage, and scale containerized applications
- 11 Create **EC2 instances** that provide secure, resizable compute capacity in the cloud
- 12 Create an **EFS** (elastic file system) that lets you share file data without provisioning or managing storage.

- SG2**
- SSH for bastion
 - SSH access to non prod MongoDB
 - Mongo access to ECS services
 - Mongo access to the bastion

- SG3**
- Allow Jenkins access to load balancer

- SG5**
- Bastion server security group allows access to the user to connect over here to access other servers

13

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat

- SG2**
- SSH access to the user
 - Mongo access to the local network
 - Mongo access to the developer
 - ICMP to the local user

- SG4**
- SSH access for bastion
 - Mongo access to the bastion
 - Mongo access to local network

We have created EC2 instance for database and Jenkin machine

14

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw