Enterprise Architecture Proposal: Amazon ECS with Fargate (ap-south-1)

1. Executive Overview

This document outlines a production-grade serverless container architecture using Amazon ECS with Fargate in the ap-south-1 (Mumbai) region. Fargate eliminates EC2 server management, provides automatic scaling, and offers a fully managed container runtime environment.

2. High-Level Architecture Flow

- Users access application via CloudFront (optional)
- CloudFront forwards traffic to Application Load Balancer (ALB)
- ALB routes traffic to ECS Service (Fargate tasks)
- ECS Tasks run containerized application (2 vCPU / 4GB each)
- Application connects to RDS (2 databases) and ElastiCache
- Logs sent to CloudWatch Logs

3. Estimated Monthly Cost – Fargate (ap-south-1)

Component	Configuration	Estimated Monthly Cost (Approx INR)
Fargate Compute	3 tasks (2 vCPU, 4GB each)	■18,000 – ■22,000
Application Load Balancer	1 ALB	■1,800 – ■2,200
ECR Storage	~10GB images	■400 – ■700
CloudWatch Logs	App + container logs	■1,000+
CloudFront (Optional)	Traffic-based pricing	Variable

4. Key Benefits of Fargate

- No EC2 server management
- Automatic scaling at container level
- Pay only for CPU and memory consumed
- High availability across AZs
- Integrated IAM roles per task
- Reduced operational overhead

5. CI/CD Flow (Jenkins → ECR → ECS Fargate)

- Developer pushes code to Git
- Jenkins builds ROOT.war
- Docker image built and tagged
- Image pushed to Amazon ECR
- Task definition updated with new image
- Rolling deployment triggered

Deployment Command Example:

```
aws ecs update-service --cluster prod-cluster --service prod-service --force-new-deployment
```

6. Sample Fargate Task Definition

```
{
  "family": "app-task",
  "requiresCompatibilities": ["FARGATE"],
  "networkMode": "awsvpc",
  "cpu": "2048",
  "memory": "4096",
  "executionRoleArn": "arn:aws:iam::account:role/ecsTaskExecutionRole",
  "containerDefinitions": [
    {
      "name": "app-container",
      "image": "<account>.dkr.ecr.ap-south-1.amazonaws.com/app:1.0",
      "portMappings": [{ "containerPort": 8080 }],
      "essential": true,
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/app",
          "awslogs-region": "ap-south-1"
        }
      }
    }
  ]
}
```

7. Scaling & Auto Rollback

- Target Tracking Scaling (CPU 60%)
- Min tasks: 2, Max tasks: 6
- Deployment Circuit Breaker enabled
- Automatic rollback on failed health checks

8. Load Balancer Configuration

- ALB in public subnet
- Target type: IP
- Health check path: /health
- HTTPS listener (443)
- ACM certificate attached

9. CloudFront Configuration

- Origin: ALB DNS name
- Viewer Protocol Policy: Redirect HTTP to HTTPS
- Origin Protocol: HTTPS only
- Attach ACM certificate in us-east-1
- Optional WAF integration

10. Security & Monitoring

- Fargate tasks in private subnets
- RDS & ElastiCache in private subnets
- IAM roles for tasks
- AWS Secrets Manager for credentials
- CloudWatch Container Insights enabled