AWS Web Hosting Design

Production-ready architecture overview

Zoltan Burgermeiszter @ 01/10/2025

Executive Summary

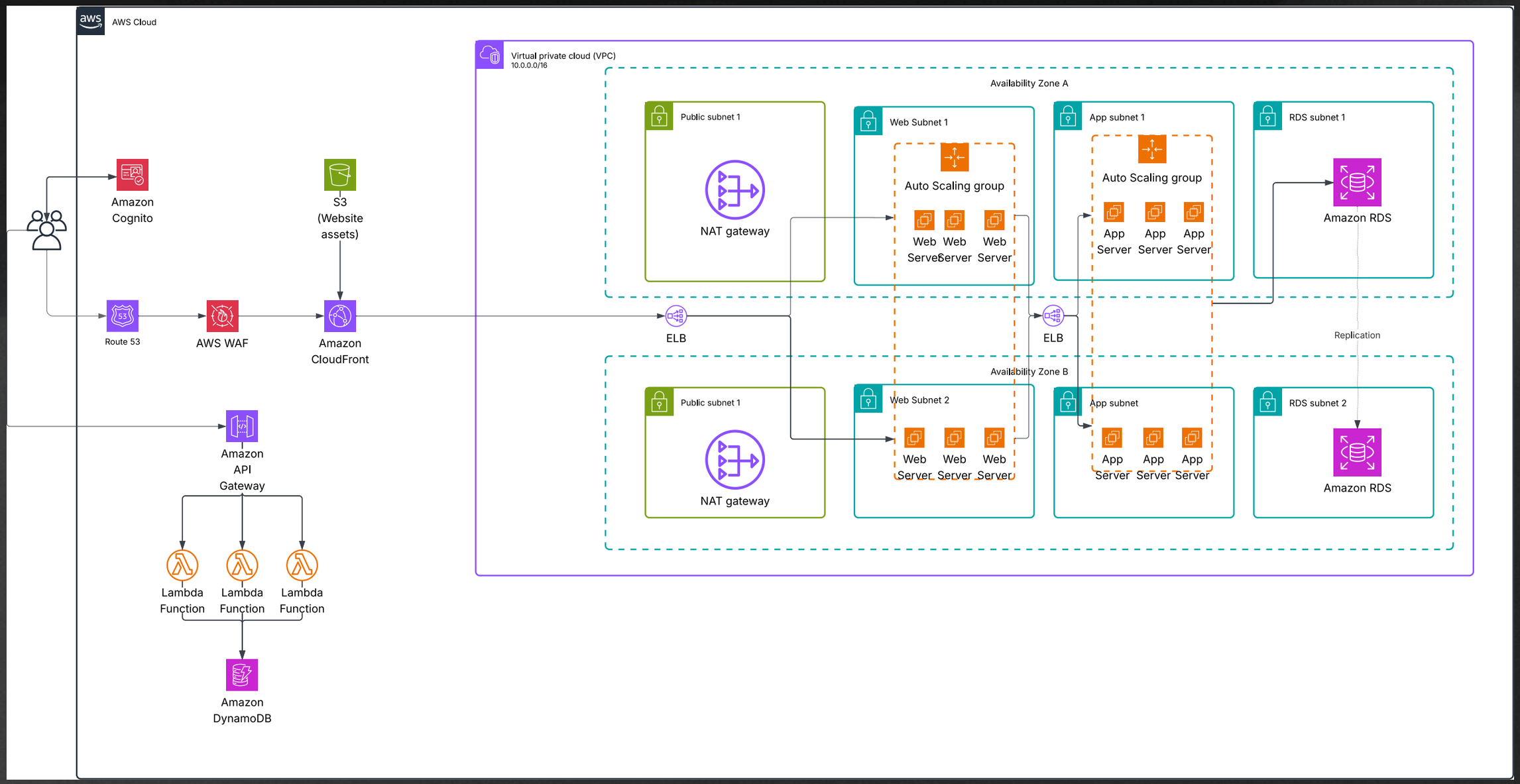
Executive Summary

Business priorities

- Reliability - Always available website, even during traffic spikes
- Stability - Secure, resilient, and scalable
- Transparency - Observability, cost control
- Managed services - Reduced operational overhead
- Smart investment - Performance and cost efficiency balance
- Future-proof - Grow with business needs
- Automation - Build, test, deployment pipelines

Architecture Overview

Architecture Overview



Architecture Overview

The building blocks

- Compute
- Networking
- Storage, databases
- Infrastructure-as-Code, CI/CD

Architecture Overview

Compute

- ECS Fargate (serverless) for containers
  - Pros: Serverless containers, predictable pricing, simple integration with ALB and IAM. Ideal for web apps and microservices.
  - Cons: Less control than EKS
- EKS for advanced workloads
  - Pros: Industry-standard Kubernetes ecosystem, portability, operators and custom schedulers.
  - Cons: Higher operational overhead, higher complexity
- Lambda for event-driven or infrequent tasks
  - Pros: Instant scaling, pay-per-execution
  - Cons: Concurrency limits, slow cold start, runtime limit for long-running processes
- EC2 for legacy or special workloads
  - Pros: Full root access, necessary for legacy apps or per-instance licenses
  - Cons: Higher operational cost, patching requirements

Architecture Overview

Networking

- VPC - Multi-AZ, public subnet (load-balancers, NAT Gateway), private subnet (applications, databases), Security Groups
- Route 53 - DNS - weighted/failover routing, healthchecks
- ACM - TLS certs for CloudFront or ALB
- CDN and Edge computing - reduced latency
    - CloudFront - caching
    - CloudFront Functions for lightweight header manipulation, redirects, and JWT auth.
    - Lamda@Edge for dynamic routing with longer runtime limits, more memory, AWS SDK and network access.
- Load Balancing
    - ELB - simple load balancing for EC2s
    - ALB - load balancing based on request content
- WAF, Shield - Advanced threat mitigation (GeoIP blocking, DDoS prevention, bots handling).
- Optional - Transit Gateway for multi-account networking, VPC Peering, and Direct Connect

Architecture Overview

Storage, databases

- S3 - for static files - public or private, multiple storage tiers for cost optimisation (Glacier)
- RDS Aurora - serverless, scalable, Postgres/MySQL compatible database
- DynamoDB for non-relational data

Architecture Overview

Infrastructure-as-Code, CI/CD

- Terraform code in Git
    - Audit trail for changes
    - Code review by the team
        - Roll-forward in a new commit to revert infra changes
- CI/CD - Github Actions
    - Automated code validation and sanity checks
    - Automated deployment
    - Minimised human mistakes and config errors

Scaling

Cost-efficiency at all times

- Auto Scaling
  - ECS Tasks
  - EKS HorizontalPodAutoscaler
  - EC2 Auto-Scaling Groups
  - Lambdas
  - Database read replicas or Aurora Serverless
- Queues
  - SQS for spreading asynchronous background load over time

Observability

Cut through the noise

- CloudWatch - AWS service
  - Metrics - Dashboards, alerting, AWS/custom metrics
  - Logs - Centralised, structured JSON
  - On-call - CloudWatch integration with PagerDuty or Slack
- Datadog - Third-party observability service
  - More user-friendly
  - Higher cost

Cost, Resilience and Security

The balance

- Cost
  - Use a carefully selected scalable mix of serverless and EC2 (reserved or spot) instances savings plans.
  - S3 lifecycle rules for inexpensive storage and deletion beyond regulatory requirements
  - Use caching for frequently requested content
- Resilience
  - Multi-AZ architecture for optimised networking costs and increased fault tolerance
  - Regular Disaster Recovery drills - failover, backup testing
- Security
  - Automated patching
  - Encryption at rest - S3, EBS, RDS
  - Encryption in-transit - ACM - TLS certs for CloudFront and Load Balancers
  - IAM - Fine-grained service roles, least-privilege principle
  - Secrets Manager - Database credentials and sensitive config
  - Rate limiting - API Gateway, Load Balancers, WAF
  - WAF, Shield - Advanced threat mitigation (GeoIP blocking, DDoS and bot prevention).

Challenges and Risks

Challenges and Risks

Minimise the unavoidable

- EKS adds complexity and operational overhead.
- Serverless costs may spike at scale.
- Lambda cold-starts impact latency.
- Security misconfigurations (S3 bucket exposure, firewall misconfigurations)
- Software supply chain security - npm audit, Snyk.

Challenges and Risks

Factors that affect the plan

- Expected traffic profile
  - Requests-per-second
  - Peak concurrency
  - Growth rate
- Application architecture
  - Containerised
  - Cloud-native
  - Legacy/special (server-based)
  - Budget constraints - CapEx vs OpEx preference

Closing

Closing

Why this design?

A pragmatic balance of managed AWS services for reduced operational burden and more complex or legacy workload hosting on Kubernetes and EC2 servers.

Security, observability, scaling and cost control are top priorities while providing flexibility for changes in business requirements.