Client-Ready Documentation: Resilient and Scalable Web Application Deployment on AWS

Executive Summary

This proposal outlines a highly available, scalable, and secure AWS architecture for a web application using VPC, EC2 Auto Scaling, Application Load Balancer (ALB), Elastic File System (EFS), and Route 53. The design aligns with AWS Well-Architected pillars and targets multi-AZ resilience, elastic scale, defense-in-depth security, and operational excellence. Optional enhancements address higher resiliency (multi-Region failover), extreme scale (ALB sharding), and cost visibility.

Business outcomes:

- High availability via multi-AZ architecture and automated healing.
- Elastic scalability with demand-driven Auto Scaling.
- Strong security with private subnets, TLS, least-privilege IAM, and monitoring.
- Clear path to RTO/RPO objectives and optional DNS failover.

Objectives and Success Criteria

- High Availability: Continuous service through AZ disruption, no single point of failure.
- Scalability: Automatic scale-out/in based on load and SLOs.
- Security: Encrypted traffic, least privilege, and network isolation.
- Resilience: Health-driven recovery at each layer and optional cross-Region failover.

Proposed SLOs (to be finalized with client):

- Availability target: ≥99.9% single-Region HA; ≥99.99% with multi-Region failover.
- Performance: P95 latency and throughput targets driven by application profile.
- Recovery: RTO/RPO defined by business continuity needs.

Target Architecture (High-Level)

- Networking: A dedicated VPC spanning at least two Availability Zones (AZs) with public and private subnets per AZ.
- Ingress: Internet-facing Application Load Balancer (ALB) in public subnets, terminating TLS and routing to target groups.
- Compute: EC2 Auto Scaling Group (ASG) across private subnets in multiple AZs; instances are immutable and health-checked.
- Storage: Amazon EFS for shared application files when needed (uploads/assets); stateless services favored by default.
- DNS: Route 53 hosted zone with alias record to ALB; optional health checks and DNS failover for active/passive or active/active designs.
- Egress: NAT Gateways for controlled outbound access by private instances.
- Security: Security groups enforcing least-privilege flows (Internet→ALB only; ALB→App; App→EFS), IAM roles with KMS-based encryption.
- Observability: CloudWatch metrics/alarms, structured logs, access logs, and CloudTrail.

Traffic flow: Client → Route 53 → ALB (TLS) → Target Group → EC2 (private) → EFS (if required) → outbound via NAT as needed.

Design Principles (AWS Well-Architected Alignment)

- Reliability: Multi-AZ distribution, health checks, automatic instance replacement, and DNS-aware routing.
- Performance Efficiency: Horizontal scaling, right-sizing, and load balancing best practices.
- Security: Zero-trust mindset with layered controls, encryption in transit/at rest, and strong identity governance.
- Cost Optimization: Autoscaling, lifecycle policies, instance right-sizing, and tiered storage choices.
- Operational Excellence: Infrastructure as Code (IaC), immutable AMIs, automated deployments, runbooks, and continuous improvement.
- Sustainability: Efficient use of resources via on-demand scaling and elimination of idle capacity.

Detailed Design

Networking

- VPC CIDR: Determined per client IP plan (e.g., 10.0.0.0/16).
- Subnets: At least two AZs; each AZ has one public (ALB/NAT) and one private (EC2/EFS access) subnet.

- Routing: IGW attached; public route tables for ALB/NAT; private route tables for NAT egress.
- NACLs: Restrictive, stateless complements to stateful security groups.

Security Controls

- Security Groups:
  - ALB SG: Inbound 443 (and 80 only for redirect if needed) from Internet; outbound to app port.
  - App SG: Inbound only from ALB SG; outbound to EFS(2049) and required endpoints.
  - EFS SG: Inbound 2049 (NFS) from App SG only.
- IAM: Instance roles scoped to minimum permissions; separate roles for build/deploy/operate; key policies for KMS.
- Encryption: TLS at ALB (ACM certificates), EBS/EFS encryption at rest, encrypted EFS mounts.
- Secrets: Managed via AWS Secrets Manager/Parameter Store with rotation policies.
- Compliance: Logging, audit trails (CloudTrail), and configuration baselines.

Compute and Autoscaling

- Launch Template: Hardened AMI, instance type profile, user data for bootstrap, CloudWatch agent, and EFS mount.
- Auto Scaling Group:
  - Multi-AZ across private subnets; health check type: ELB and EC2.
  - Capacity: min=2 (HA baseline), desired varies by demand, max sized per forecast.
  - Scaling Policies: Target tracking (e.g., CPU%, ALB RequestCountPerTarget) and step policies for predictable spikes.
  - Lifecycle Hooks: For graceful drain/termination and pre-warm steps if applicable.

Load Balancing

- ALB Listeners:
  - 80→Redirect to 443 (optional).
  - 443 with TLS (ACM), modern cipher policy.
- Target Group:
  - Health check path (e.g., /health) with sane thresholds and intervals.
  - Deregistration delay for graceful connection draining.
- Advanced Patterns (optional):
  - ALB sharding for very high TPS/connection counts, fronted by Route 53 weighted records.

Storage

- EFS:
  - Regional file system with per-AZ mount targets.
  - Performance mode: General Purpose (default) or Max I/O for high concurrency.
  - Throughput: Bursting or provisioned based on workload.
  - Access points and POSIX permissions for multi-tenant or microservices scenarios.
- Alternatives:
  - Prefer stateless app nodes with object storage (S3) where feasible.
  - Use RDS/Aurora for relational data (out of scope here but compatible).

DNS and Resilience

- Route 53:
  - Public hosted zone and alias A/AAAA to ALB.
  - Evaluate target health: Enabled for ALB alias.
  - Health checks for failover policies (active-passive) to secondary stack/Region if required.
- Multi-Region Options:
  - Active-Passive: Health-checked failover with RTO minutes.
  - Active-Active: Latency/weighted routing; careful data replication strategy.

Observability and Operations

- Metrics: ALB (RequestCount, 5XX, TargetResponseTime), EC2 (CPU, status), ASG (UnhealthyHostCount), EFS (IO/Tput), Route 53 health status.
- Logs: ALB access logs, application logs, system logs aggregated to CloudWatch; structured logging recommended.
- Alarms & Actions: Autoscaling triggers, error-rate thresholds, latency SLO breach, EFS credit alarms, DNS failover triggers.
- Runbooks: Scale tuning, failover and rollback, incident handling, disaster recovery drills.

Implementation Plan

Phase 1: Design and Readiness

- Confirm non-functional requirements: availability, latency, RTO/RPO, compliance.
- Finalize capacity assumptions: baseline/peak RPS, traffic patterns, data growth.
- Security sign-off: ports, identity, secrets, encryption, audit.

Phase 2: Build Foundation

- Provision VPC, subnets, route tables, IGW, NAT Gateways.
- Create security groups per design.
- Set up EFS with mount targets and access controls.
- Prepare ACM TLS certificates.

Phase 3: Compute & Load Balancing

- Bake hardened AMI (Image Builder/Packer) with app runtime, agents, and bootstrap.
- Create Launch Template and Auto Scaling Group across two or more AZs.
- Configure ALB, listeners, target groups, and health checks.
- Attach ASG to target group and validate registration.

Phase 4: DNS and Access

- Configure Route 53 hosted zone and alias records.
- Optionally implement health checks and failover policies.
- Validate TLS endpoints and HSTS/redirect behavior.

Phase 5: Testing & Optimization

- Functional tests: connectivity, health checks, EFS mount, least-privilege flows.
- Load tests: step/ramp/spike; verify scale-out/in timing and stability.
- Failure drills: instance termination, AZ impairment, and (if configured) DNS failover.
- Tuning: health check thresholds, scaling targets/cooldowns, instance right-sizing, EFS throughput mode.

Phase 6: Documentation & Handover

- As-built diagrams (logical and network), parameter catalogs, and security mappings.
- Runbooks/playbooks, alarm catalogs, and deployment pipelines overview.
- Knowledge transfer and Well-Architected review summary.

Risks and Mitigations

- Bootstrap Latency: Bake AMIs and minimize user-data work; warm pools if necessary.
- Health Check Sensitivity: Balance intervals/thresholds to avoid flapping; use lightweight /health endpoints.
- EFS Performance: Choose correct performance/throughput modes; cache warmup; consider S3 for static assets.

- Sudden Traffic Spikes: Pre-scaling/target tracking tuning; potential ALB sharding for extreme scale.
- Configuration Drift: Enforce IaC, immutable deployments, and automated golden images.

Cost Considerations

- Variable components: EC2 (by instance hours), ALB LCU usage, NAT egress, EFS storage/throughput, data transfer.
- Optimization levers:
    - Right-size instance families and use Auto Scaling aggressively.
    - Consider Savings Plans/Reserved Instances for steady baselines.
    - Reduce NAT usage via VPC endpoints where applicable.
    - Use S3 for static content to offload EFS where feasible.
- Optional: Provide an estimate once traffic assumptions and environment sizing are confirmed.

Deliverables

- Architectural Diagrams and Design Rationale.
- Implementation and Configuration Guide (step-by-step with parameters and IaC structure).
- Performance and Optimization Report (load test results, scaling behavior, recovery timings).
- Project Presentation (objectives, architecture, security, test results, risks/mitigations, roadmap).

Meeting-Ready Slides Outline (10–12 Slides)

1. Executive Summary and Outcomes
2. Requirements and Success Criteria
3. High-Level Architecture Diagram
4. Networking and Security Model
5. Compute and Auto Scaling
6. Load Balancer and Health Checks
7. Storage Strategy (EFS vs. Stateless/S3)
8. DNS and Resilience Options (Single-Region vs. Multi-Region)
9. Observability and Operations
10. Risks and Mitigations
11. Cost Levers and Next Steps
12. Q&A

Next Steps

- Confirm non-functional requirements and SLAs (availability, latency, RTO/RPO).
- Provide traffic and data assumptions for sizing and cost estimation.
- Approve initial PoC scope and timeline:
  - Week 1: Foundation (VPC, SGs, EFS, ACM)
  - Week 2: Compute/ALB/ASG and functional validation
  - Week 3: Load testing, tuning, and failover drills
  - Week 4: Documentation, handover, and roadmap

If desired, this content can be converted into a slide deck and a one-page architecture diagram.