

AWS Architecting & Ecosystem

- AWS Architecting & Ecosystem
 - Well Architected Framework General Guiding Principles
 - AWS Cloud Best Practices - Design Principles
 - Well Architected Framework 6 Pillars
 - * 1. Operational Excellence
 - * 2. Security
 - * 3. Reliability
 - * 4. Performance Efficiency
 - * 5. Cost Optimization
 - * 6. Sustainability
 - AWS Well-Architected Tool
 - AWS Right Sizing
 - AWS Ecosystem - Free Resources
 - * AWS Ecosystem - AWS Support
 - AWS Marketplace

Well Architected Framework General Guiding Principles

- Stop guessing capacity needs.
- Test systems at production scale.
- Automate to facilitate architectural experimentation.
- Allow for evolutionary architectures based on changing requirements.
- Drive architectures using data.
- Improve through game days by simulating applications for flash sale days.

AWS Cloud Best Practices - Design Principles

- **Scalability:** Scale both vertically and horizontally.
- **Disposable Resources:** Servers should be disposable and easily configured.
- **Automation:** Utilize serverless, infrastructure as a service, and auto-scaling.
- **Loose Coupling:** Break monolithic applications into smaller, loosely coupled components to prevent cascading failures.
- **Services, Not Servers:** Use managed services, databases, and serverless options instead of just EC2.

Well Architected Framework 6 Pillars

1. Operational Excellence

- Ability to run and monitor systems for business value and improve supporting processes.
- **Design Principles:**
 - Perform operations as code (Infrastructure as code).
 - Automate the creation of annotated documentation.
 - Make frequent, small, reversible changes.
 - Refine operations procedures frequently and ensure team familiarity.
 - Anticipate failure.
 - Learn from all operational failures.

2. Security

- Ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.
- **Design Principles:**
 - Implement a strong identity foundation (centralized privilege management, principle of least privilege, IAM).
 - Enable traceability (integrate logs and metrics with systems).
 - Apply security at all layers (edge network, VPC, load balancer, instances, OS, application).
 - Automate security best practices.
 - Protect data in transit and at rest (encryption, tokenization, access control).
 - Keep people away from data (reduce or eliminate direct access).
 - Prepare for security events (run incident response simulations, use automation).

3. Reliability

- Ability to recover from disruptions, dynamically acquire resources, and mitigate misconfigurations or transient network issues.

- **Design Principles:**
 - Test recovery procedures using automation.
 - Automatically recover from failures.
 - Scale horizontally to increase availability.
 - Stop guessing capacity (use Auto Scaling).
 - Manage change with automation.

4. Performance Efficiency

- Efficiently use computing resources to meet system requirements and maintain efficiency as demands change.
- **Design Principles:**
 - Democratize advanced technologies (use services).
 - Go global in minutes (deploy in multiple regions).
 - Use serverless architectures.
 - Experiment frequently.
 - Be aware of all AWS services (mechanical sympathy).

5. Cost Optimization

- Deliver business value at the lowest price point.
- **Design Principles:**
 - Adopt a consumption model (pay for what you use).
 - Measure overall efficiency (use CloudWatch).
 - Stop spending on data center operations (focus on projects).
 - Analyze and attribute expenditure (use tags to measure ROI).
 - Use managed services to reduce costs.

6. Sustainability

- Minimize environmental impacts of running cloud workloads.
- **Design Principles:**
 - Understand your impact (establish performance indicators).
 - Set sustainability goals for each workload.
 - Maximize utilization (right size workloads).
 - Anticipate and adopt new efficient technologies.
 - Use managed services to automate sustainability best practices.
 - Reduce downstream impact (minimize energy/resources for services).

AWS Well-Architected Tool

- Free tool to review architectures against the 6 pillars and adopt best practices.
- **How it works:**
 - Select your workload and answer questions.
 - Review answers against the 6 pillars.
 - Obtain advice: videos, documentation, reports, and dashboards.

AWS Right Sizing

- Match instance types and sizes to workload performance and capacity requirements at the lowest cost.
- Right sizing involves starting small and scaling up easily, continuously adjusting after cloud onboarding, and using tools like CloudWatch, Cost Explorer, and Trusted Advisor.

AWS Ecosystem - Free Resources

- **AWS Blogs:** AWS Blogs
- **AWS Forums:** AWS Forums
- **AWS Whitepapers & Guides:** AWS Whitepapers & Guides
- **AWS Quick Starts:** AWS Quick Starts
 - Automated, gold-standard deployments in the AWS Cloud.
 - Examples: WordPress on AWS, leveraging CloudFormation.
- **AWS Solutions:** AWS Solutions
 - Vetted technology solutions for the AWS Cloud.
 - Example - AWS Landing Zone (secure, multi-account environment).

AWS Ecosystem - AWS Support

DEVELOPER	BUSINESS	ENTERPRISE
Business hours email access to Cloud Support Associates	24x7 phone, email, and chat access to Cloud Support Engineers	Access to a Technical Account Manager (TAM)
General guidance: < 24 business hours	Production system impaired: < 4 hours	Concierge Support Team (for billing and account best practices)
System impaired: < 12 business hours	Production system down: < 1 hour	Business-critical system down: < 15 minutes

AWS Marketplace

- Digital catalog with thousands of software listings from independent software vendors.
- Examples:
 - Custom AMIs, CloudFormation templates, SaaS, containers.
- Purchases go into your AWS bill.
- You can sell your own solutions on the AWS Marketplace.