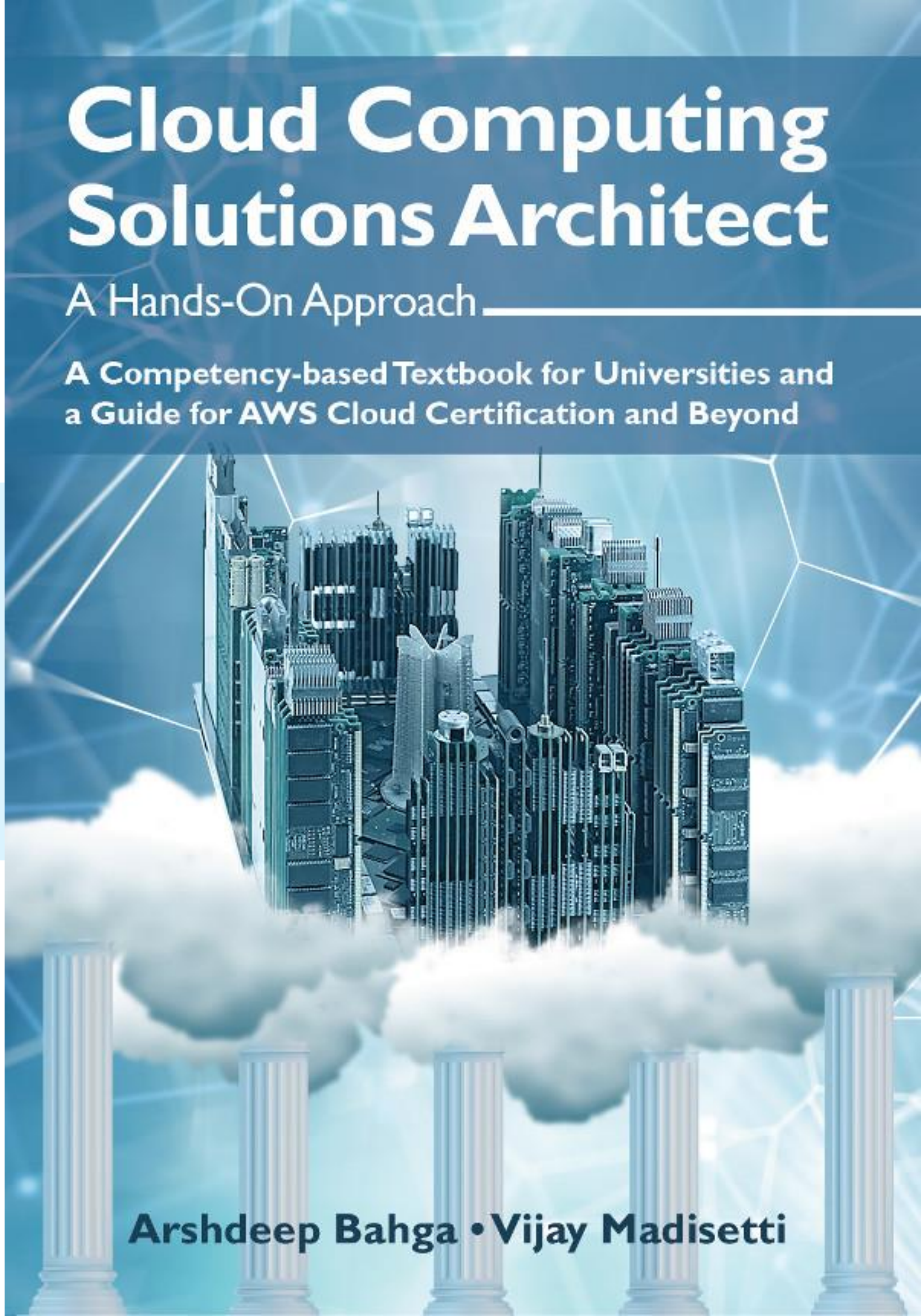
Chapter 18

Applying the Reliability Pillar



Cloud Computing Solutions Architect

A Hands-On Approach

A Competency-based Textbook for Universities and a Guide for AWS Cloud Certification and Beyond

Arshdeep Bahga • Vijay Madisetti

Book website: www.hands-on-books-series.com

Reliability Pillar

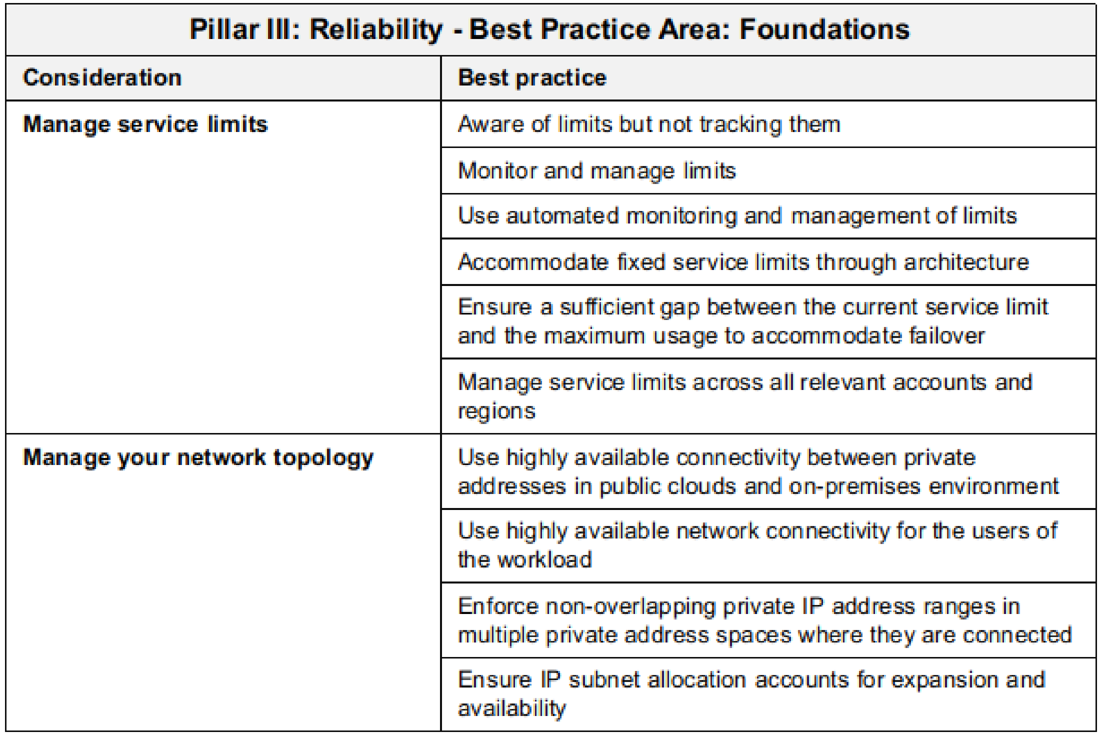
- The Reliability pillar includes the ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues.
- Within the Reliability pillar, there are three best practice areas:
    - Foundations
    - Change Management and Failure
    - Management

Design Principles for Reliability Pillar

- Test how your system would fail and validate the recovery procedures.
- Monitor your system for key performance indicators to track for failures and automatically trigger recovery procedures to recover from failure.
- Scale horizontally (by adding multiple small resources) rather than vertically (using one large resource) to avoid a single point of failure.
- Automate the addition and removal of resources to meet the demand instead of guessing the required capacity.
- Automate changes to be done to your infrastructure.

Best Practice Area: Foundations

- The Foundations best practice area highlights the importance of having a well-planned foundation and monitoring in place to achieve reliability.
- To manage service limits, you should be aware that default service limits exist to prevent accidental provisioning of more resources. Monitor usage of services and implement tools to alert you when a service limit is about to be reached.
- You should ensure that there is a sufficient gap between the current service limit and the maximum usage to accommodate failover as a failed resource may still be counted against limits until it is successfully terminated.
- If your application exists in multiple environments such as a public clouc or on-premises data center, you must use highly available connectivity between the environments.
- You should also use highly available network connectivity for the users of your application.
- If your application spans multiple environments (VPCs or on-premises data center), the IP address ranges of the environments should not overlap.
- The VPC IP address ranges must have sufficient room for future expansion.

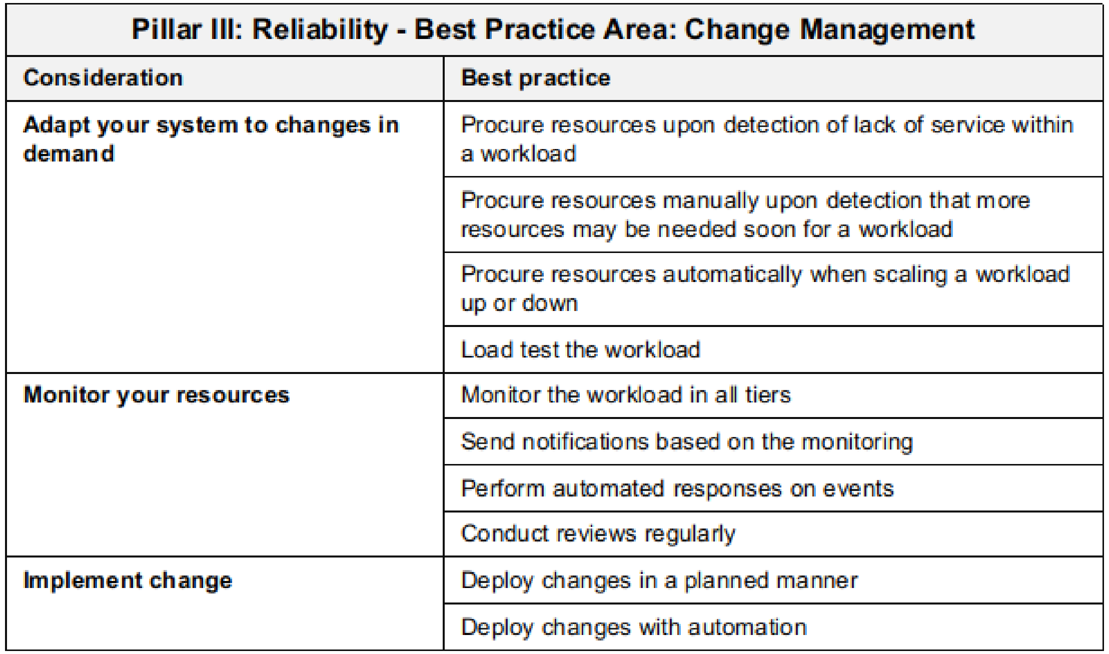


Pillar III: Reliability - Best Practice Area: Foundations

Consideration	Best practice
Manage service limits	Aware of limits but not tracking them
	Monitor and manage limits
	Use automated monitoring and management of limits
	Accommodate fixed service limits through architecture
	Ensure a sufficient gap between the current service limit and the maximum usage to accommodate failover
	Manage service limits across all relevant accounts and regions
Manage your network topology	Use highly available connectivity between private addresses in public clouds and on-premises environment
	Use highly available network connectivity for the users of the workload
	Enforce non-overlapping private IP address ranges in multiple private address spaces where they are connected
	Ensure IP subnet allocation accounts for expansion and availability

Best Practice Area: Change Management

- The Change Management best practice area highlights the importance of monitoring and planning proactively for changes in demand to avoid capacity issues or SLA breaches.
- To ensure that your system adapts to changes in demand, you should either manually scale resources or use a service which scales automatically.
- Load test your application to ensure that it can meet the workload requirements.
- To monitor your resources, use logs and metrics, and send notifications when thresholds are crossed.
- Perform automated actions in the event of failures.
- To implement change, deploy changes in a planned and automated manner.

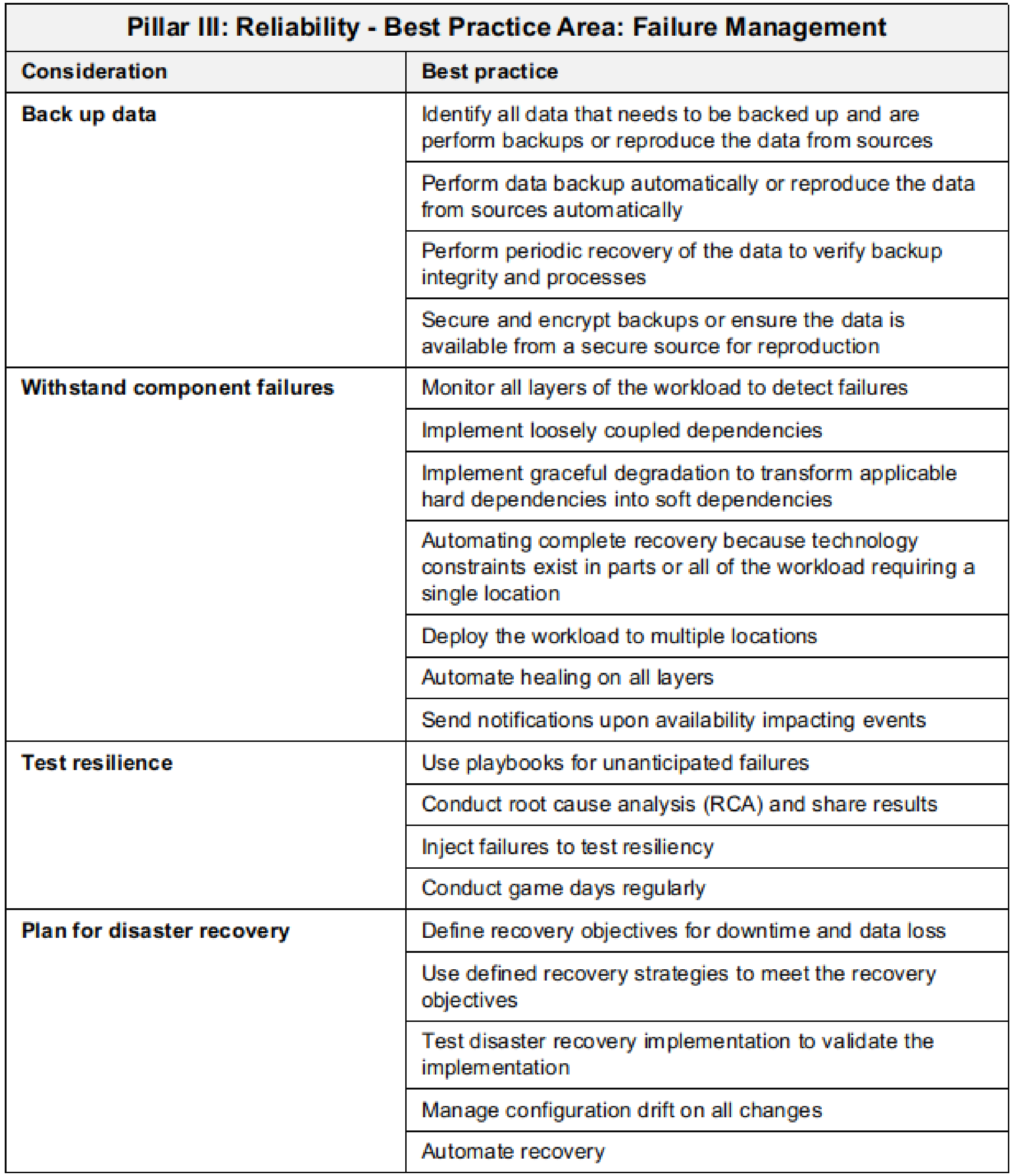


Pillar III: Reliability - Best Practice Area: Change Management

Consideration	Best practice
Adapt your system to changes in demand	Procure resources upon detection of lack of service within a workload
	Procure resources manually upon detection that more resources may be needed soon for a workload
	Procure resources automatically when scaling a workload up or down
	Load test the workload
Monitor your resources	Monitor the workload in all tiers
	Send notifications based on the monitoring
	Perform automated responses on events
	Conduct reviews regularly
Implement change	Deploy changes in a planned manner
	Deploy changes with automation

Best Practice Area: Failure Management

- The Failure Management best practice area highlights the importance of frequent and automated testing of systems and recovery processes so that you can recover all your data and continue to serve your customers, even in the face of sustained problems.
- Perform automated backups of all important data or ensure that the data can be generated from the source.
- Validate your backup processes by performing periodic recovery of data.
- All backups must be secured and encrypted.
- To ensure that your system can withstand component failures, continuously monitor the system health, and report any performance degradation or failures.
- Implement loose coupling between components so that in the event of any failure of a component, the other dependent components can continue to serve requests in a degraded manner.
- Deploy your system across multiple availability zones and regions. Setup automated healing actions for all layers of your system.
- Send notification on the detection of failure events even if they were healed automatically. To test the resilience of your system, use playbooks for unanticipated failures.
- Review each failure event to identify the root cause of failure.
- To plan for disaster recovery, define your recovery time objective (RTO) and recovery point objective (RPO).
- Regularly test your disaster recovery strategies to ensure that RTO and RPO are met.

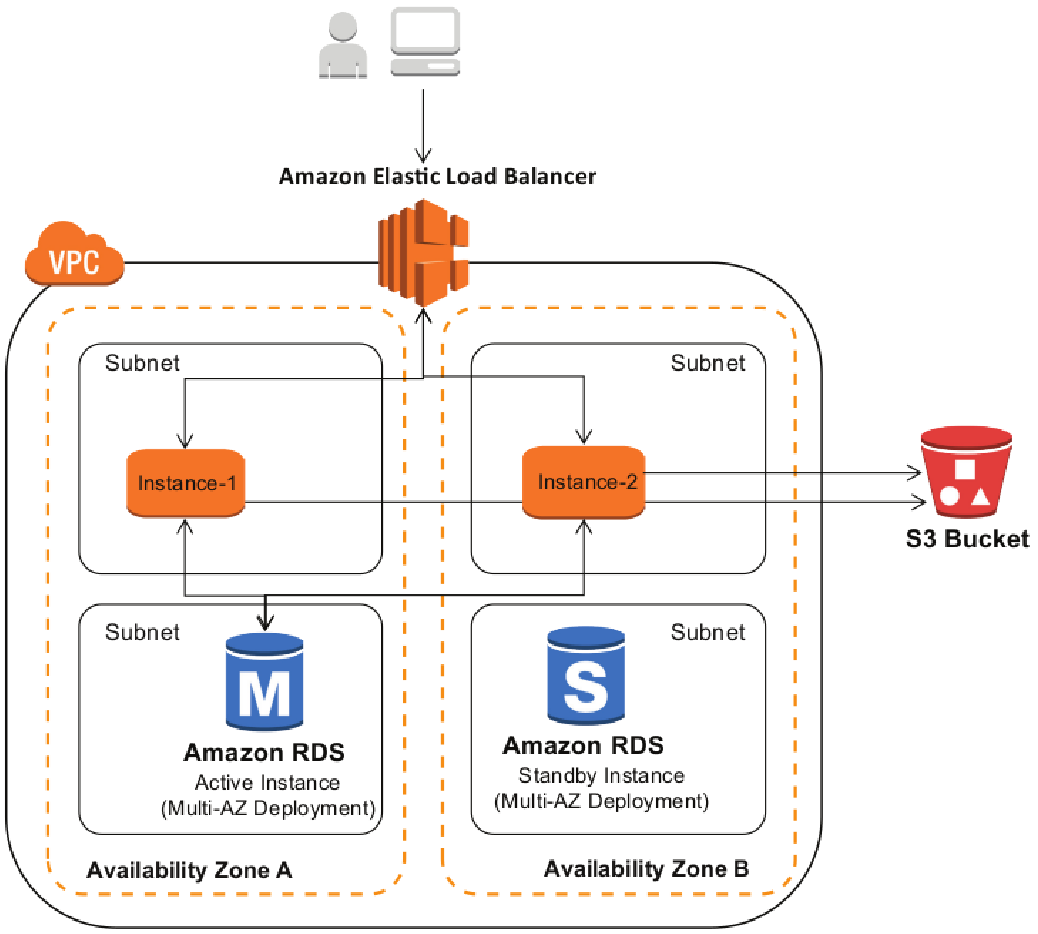


Pillar III: Reliability - Best Practice Area: Failure Management

Consideration	Best practice
Back up data	Identify all data that needs to be backed up and are perform backups or reproduce the data from sources
	Perform data backup automatically or reproduce the data from sources automatically
	Perform periodic recovery of the data to verify backup integrity and processes
	Secure and encrypt backups or ensure the data is available from a secure source for reproduction
Withstand component failures	Monitor all layers of the workload to detect failures
	Implement loosely coupled dependencies
	Implement graceful degradation to transform applicable hard dependencies into soft dependencies
	Automating complete recovery because technology constraints exist in parts or all of the workload requiring a single location
	Deploy the workload to multiple locations
	Automate healing on all layers
	Send notifications upon availability impacting events
Test resilience	Use playbooks for unanticipated failures
	Conduct root cause analysis (RCA) and share results
	Inject failures to test resiliency
	Conduct game days regularly
Plan for disaster recovery	Define recovery objectives for downtime and data loss
	Use defined recovery strategies to meet the recovery objectives
	Test disaster recovery implementation to validate the implementation
	Manage configuration drift on all changes
	Automate recovery

Recipe for Reliability Pillar

- With this recipe, we make the photo gallery application more reliable by replacing the single point of failures and introducing redundancy in the application and database tiers.
- To make the application highly available and reliable, we setup EC2 instances for the application servers in separate availability zones and then place the instances under an Elastic Load Balancer (ELB).
- Further, we set up a Multi-AZ deployment for the database with a standby instance to provide high availability and automatic failover.



Amazon Elastic Load Balancer
VPC
Subnet
Subnet
Instance-1
Instance-2
S3 Bucket
Subnet
Subnet
Amazon RDS
Active Instance
(Multi-AZ Deployment)
Amazon RDS
Standby Instance
(Multi-AZ Deployment)
Availability Zone A
Availability Zone B