



PHASE	STEPS	SAMPLE ACTIVITIES
1. Assessment and Planning	1. Inventory and Dependency Mapping	<ul style="list-style-type: none"> a. Identify all applications and their components. b. Highlight dependencies for a streamlined transition.
	2. Basic Resource Utilization Analysis	<ul style="list-style-type: none"> a. Gather essential performance metrics for swift decision-making. b. Choose AWS instance types based on initial resource considerations.
	3. Basic Security and Compliance Check	<ul style="list-style-type: none"> a. Ensure minimal disruption to existing security measures. b. Implement basic AWS security measures such as security groups and IAM roles.
	4. Streamlined Data Migration Plan	<ul style="list-style-type: none"> a. Assess MySQL database schema. b. Plan for rapid database migration using AWS Database Migration Service (DMS) or similar tools.
2. AWS Environment Setup	1. AWS Account Setup	<ul style="list-style-type: none"> a. Set up AWS account with a focus on rapid deployment.
	2. Network Setup	<ul style="list-style-type: none"> a. Design a simplified VPC architecture for quick implementation. b. Establish basic security groups and Network ACLs.
	3. AM Configuration	<ul style="list-style-type: none"> a. Set up basic IAM roles and policies for applications.
	4. Service Selection	<ul style="list-style-type: none"> a. Choose straightforward AWS services for each component. b. Prioritize speed and flexibility over complexity per requirements.
3. Application Migration	1. Front-end (Angular)	<ul style="list-style-type: none"> a. Package and deploy Angular assets to S3 or use AWS Amplify for rapid deployment.
	2. Backend (.NET)	<ul style="list-style-type: none"> a. Deploy .NET applications on EC2 instances without extensive reconfiguration.
	3. Database Migration	<ul style="list-style-type: none"> a. Set up RDS for MySQL with minimal downtime. b. Utilize efficient methods for data migration without complex transformations.
	4. Configuration Updates	<ul style="list-style-type: none"> a. Expedite the modification of connection strings and configurations. b. Focus on ensuring seamless communication between components.



PHASE	STEPS	SAMPLE ACTIVITIES
4. Testing	1. Essential Unit and Integration Testing	<ul style="list-style-type: none"> a. Prioritize critical testing for individual components b. Ensure basic functionality and communication integrity.
	2. Load Testing	<ul style="list-style-type: none"> a. Simulate essential traffic loads for swift identification of major performance issues.
	3. Basic Security Testing	<ul style="list-style-type: none"> a. Conduct basic security assessments to identify critical vulnerabilities.
5. Deployment	1. Rollout Plan	<ul style="list-style-type: none"> a. Implement a phased rollout strategy for applications
	2. Basic Monitoring and Logging	<ul style="list-style-type: none"> a. Set up basic CloudWatch monitoring and logging. b. Configure essential alarms for key metrics.
6. Post-Migration Optimization	1. Immediate Resource Optimization	<ul style="list-style-type: none"> a. Make adjustments based on initial performance metrics. b. Implement basic auto-scaling where applicable.
	2. Backup and Disaster Recovery Setup	<ul style="list-style-type: none"> a. Establish quick and basic backup procedures based on the provided requirements provided. b. Test the core components of the disaster recovery process.
7. Continuous Monitoring and Iteration	1. Handover to Operations Team	<ul style="list-style-type: none"> a. Transfer key knowledge to the operations team efficiently. b. Ensure access and permissions are set up for immediate management.
	2. Continuous Monitoring	<ul style="list-style-type: none"> a. Establish ongoing monitoring practices for quick issue identification. b. Regularly review logs and performance metrics for basic insights.