Acme Manufacturing Cloud Migration: Proof of Concept (POC) Deployment

This document provides step-by-step instructions for deploying a Proof of Concept (POC) application to AWS. The goal is to validate the chosen architecture, identify potential challenges, and refine the migration process before migrating critical applications. This POC will utilize a non-critical application selected for its relatively low impact and complexity. Remember to always consult the official AWS documentation for the most up-to-date information and best practices.

I. Application Selection:

1. **Identify POC Application:** Select a non-critical application suitable for migration. Consider factors such as size, complexity, dependencies, and data volume. The application should be small enough to be migrated quickly and easily, minimizing the risk and disruption if issues are encountered.
2. **Document Dependencies:** Document all application dependencies, including databases, external services, and other systems.
3. **Data Subset:** If the application uses a significant amount of data, consider migrating only a subset of the data for the POC.

II. AWS Environment Setup:

(Assume the core AWS infrastructure (VPC, subnets, internet gateway, NAT gateway, security groups, IAM roles) has already been set up as detailed in the main AWS infrastructure document.)

A. EC2 Instance Launch (Tsakani):

1. **AMI Selection:** Choose an appropriate Amazon Machine Image (AMI) based on the application's requirements (operating system, programming language, etc.).
2. **Instance Type Selection:** Select an appropriate EC2 instance type based on the application's performance requirements (CPU, memory, storage). Consider starting with a smaller instance type and scaling up if needed.
3. **Security Group Assignment:** Assign the appropriate security group (e.g., Acme-Web-SG) to the instance, allowing only necessary inbound and outbound traffic.
4. **IAM Role Assignment:** Assign the Acme-EC2-Role IAM role to the instance, granting only necessary permissions.
5. **Key Pair:** Use a pre-existing key pair or create a new one to access the instance via SSH.
6. **Launch Instance:** Launch the EC2 instance and wait for it to reach the "running" state.
7. **Connectivity Test:** Verify SSH connectivity to the launched instance.

B. Application Deployment (Tsakani):

1. **Application Package:** Prepare the application for deployment (e.g., create a deployable artifact or package).
2. **Deployment Method:** Choose an appropriate deployment method (e.g., SSH, deployment tools).
3. **Deployment Execution:** Deploy the application to the EC2 instance. Ensure that all necessary dependencies are installed and configured correctly.
4. **Configuration:** Configure the application as needed, including database connections and other settings.

C. Database Migration (Bushy):

1. **Database Selection:** If the POC application uses a database, choose the appropriate database service (e.g., RDS, DynamoDB) and create an instance.
2. **Data Subset:** Migrate a subset of the data to the new database instance.
3. **Connection Configuration:** Configure the application to connect to the newly created database instance.

III. Testing (Tsakani, Yamkelani, Lusanda):

1. **Functional Testing:** Verify that all core application functions are working correctly in the AWS environment.
2. **Performance Testing:** Conduct performance testing to assess response time and resource utilization.
3. **Security Testing:** Perform basic security testing to identify any potential vulnerabilities.
4. **Data Validation:** Verify that the data migrated to the new database (if applicable) is accurate and complete.

IV. Documentation:

1. **Detailed Steps:** Document all steps taken during the deployment and testing process, including commands executed, configurations applied, and any issues encountered.
2. **Test Results:** Document all test results, including performance metrics, error logs, and any identified vulnerabilities.
3. **Lessons Learned:** Document any lessons learned during the POC to inform the broader migration strategy.

V. Rollback Plan:

Define a rollback plan in case of failure. This plan should detail the steps necessary to revert the application and data to its pre-migration state.

This document provides a high-level overview. Refer to the official AWS documentation and specific application documentation for detailed instructions. Always prioritize security and

conduct thorough testing at each step. Any deviations from this plan must be properly documented and approved.