This manual provides detailed, step-by-step instructions with examples for using the AWS Management Console to execute the Acme Manufacturing cloud migration project. All team members must follow these procedures precisely. Screenshots should accompany each step in a separate document (not included here). Always consult official AWS documentation for the most up-to-date information. Prioritize security best practices and adhere to the principle of least privilege in all IAM configurations.

I. Project Setup & Infrastructure (Yanga & Lusanda)

1. **VPC Creation:**
   - **Action:** In the EC2 service, select "VPCs," then "Create VPC."
   - **Example:** Use CIDR block 10.0.0.0/16, IPv6 CIDR block 2001:db8::/56. Name: Acme-VPC-Prod.
2.
3. **Subnet Creation:**
   - **Action:** In the EC2 service, select "Subnets," then "Create subnet."
   - **Example:** Create two subnets:
     - Acme-Public-Subnet-US-East-1a (CIDR: 10.0.1.0/24, AZ: us-east-1a, Public).
     - Acme-Private-Subnet-US-East-1b (CIDR: 10.0.2.0/24, AZ: us-east-1b, Private).
   -
4.
5. **Internet Gateway:**
   - **Action:** In the EC2 service, select "Internet Gateways," then "Create internet gateway."
   - **Example:** Name: Acme-IGW-Prod. Attach to Acme-VPC-Prod.
6.
7. **NAT Gateway:**
   - **Action:** In the EC2 service, select "NAT Gateways," then "Create NAT Gateway."
   - **Example:** Select Acme-VPC-Prod, subnet Acme-Public-Subnet-US-East-1a. Instance type: natgateway-standard.
8.
9. **Route Table Configuration:**
   - **Action:** In the EC2 service, select "Route Tables." Create route tables for public and private subnets.
   - **Example:**
     - **Public Route Table:** Add route to the internet gateway (target: 0.0.0.0/0).
     - **Private Route Table:** No internet gateway route; only routes within the VPC.
   -
10.
11. **Security Group Creation:**

- **Action:** In the EC2 service, select "Security Groups," then "Create security group."
- **Example:**
  - **Acme-Web-SG:** Allow inbound HTTP (port 80), HTTPS (port 443), SSH (port 22 from your IP). Allow outbound all traffic.
  - **Acme-Database-SG:** Allow inbound only from Acme-Web-SG on the database port.
-

12.
13. **IAM Role Creation (Example: EC2 Role):**
    - **Action:** In the IAM service, select "Roles," then "Create role."
    - **Example:** Select "AWS service" as the trusted entity, choose "EC2". Attach a policy allowing only necessary EC2 actions (e.g., a custom policy limiting actions to specific instance types and AMIs within your VPC). Name: Acme-EC2-Role.
14.

II. Application Deployment (Tsakani)

A. Deploying a Node.js Application to EC2:

1. **Launch Instance:** Launch an EC2 instance (t2.medium) in Acme-VPC-Prod, Acme-Private-Subnet-US-East-1b, security group Acme-Web-SG, IAM role Acme-EC2-Role.
2. **Connect via SSH:** Connect using your key pair.
3. **Install Node.js:** sudo yum install nodejs npm
4. **Clone Repo:** git clone <your_repo_url>
5. **Install Dependencies:** npm install
6. **Start Application:** npm start

B. Deploying a Python Application to Elastic Beanstalk:

1. **Create Elastic Beanstalk Application:** In Elastic Beanstalk, create an application (e.g., OrderProcessingApp).
2. **Create Environment:** Choose Python 3.9, t3.micro instance type, security group Acme-Web-SG.
3. **Deploy:** Specify the source code repository (e.g., GitHub). Elastic Beanstalk will handle the deployment.

III. Data Migration (Bushy)

A. Migrating MySQL Database to RDS:

1. **Create RDS Instance:** In RDS, create a MySQL DB instance. Choose a DB instance identifier (e.g., Acme-OrderDB), DB engine version (e.g., MySQL 8.0), instance class (e.g., db.t3.medium), VPC (Acme-VPC-Prod), subnet (Acme-Private-Subnet-US-East-1b), security group (Acme-Database-SG).

2. **Import Data:** After the instance is created, use the AWS console or tools like AWS DMS to import your existing MySQL database.

B. Uploading Data to S3:

1. **Create S3 Bucket:** In S3, create a bucket (e.g., acme-manufacturing-data). Configure appropriate access control and encryption.
2. **Upload Data:** Use the AWS console or AWS CLI to upload data files.

C. DynamoDB Table Creation:

1. **Create DynamoDB Table:** In DynamoDB, create a table (e.g., ResumeData).
2. **Define Primary Key:** Specify the primary key (e.g., resumeId). Add other attributes as needed (e.g., name, email, skills).
3. **Provisioned Capacity:** Set appropriate read and write capacity units.

IV. Resume Processing (Tsakani, Bushy, Yamkelani) *(Refer to diagram)*

This section is complex; adapt to your actual implementation. This is a high-level example.

1. **Lambda Function (Python):** Create a Lambda function (e.g., ProcessResume).
2. **S3 Trigger:** Configure an S3 trigger to invoke this function whenever a file is uploaded to a specific S3 bucket.
3. **Textract Integration:** Use the boto3 library in your Lambda function to call the Textract API to extract data from the resume.
4. **DynamoDB Write:** Write the extracted data to your DynamoDB table.
5. **Error Handling & Logging:** Implement robust error handling and logging using CloudWatch Logs.

V. Monitoring and Optimization (Yamkelani)

1. **CloudWatch Metrics:** In CloudWatch, create custom metrics for relevant services (e.g., Lambda function invocations, DynamoDB throughput).
2. **Alarms:** Create alarms to notify you of performance issues. For example, an alarm that triggers if Lambda function errors exceed a threshold.

VI. Security and Operations (Lusanda)

1. **GuardDuty:** Enable GuardDuty for your VPC.
2. **CloudTrail:** Create a CloudTrail trail to log all API activity in your account. Configure logging to an S3 bucket.

VII. Post-Migration Activities (Yanga)

- **Testing:** Test the entire pipeline thoroughly, ensuring data integrity and application functionality.

- **Documentation:** Create comprehensive documentation covering all aspects of the migration.

This detailed manual provides a structured approach. Remember to meticulously document each step, including configurations, commands, and any issues encountered. Regular testing and security reviews are crucial. Close collaboration is key. Always refer to the official AWS documentation for complete details and best practices.