Acme Manufacturing Cloud Migration: AWS Console Step-by-Step Guide

This guide provides detailed step-by-step instructions for each team member, using examples and screenshots to illustrate the process.

Yanga - Network Configuration & Security

1. Create Virtual Private Cloud (VPC)

- **Step 1:** Log in to the AWS Management Console.
- **Step 2:** Navigate to **VPC** in the services menu.
- **Step 3:** Click **Create VPC**.
- **Step 4:** Enter a VPC name (e.g., "Acme-VPC").
- **Step 5:** Choose a CIDR Block (e.g., 10.0.0.0/16).
- **Step 6:** Select **Default VPC settings**.
- **Step 7:** Click **Create VPC**.

[Screenshot: VPC Creation]

2. Create Subnets

- **Step 1:** In the VPC dashboard, click **Subnets**.
- **Step 2:** Click **Create subnet**.
- **Step 3:** Select the previously created VPC ("Acme-VPC").
- **Step 4:** Choose a Subnet name (e.g., "Acme-Public-Subnet").
- **Step 5:** Specify a CIDR Block (e.g., 10.0.0.0/24) for the subnet.
- **Step 6:** Choose Availability Zone based on your desired location (e.g., us-east-1a).
- **Step 7:** Select **Public subnet**.
- **Step 8:** Click **Create subnet**.
- **Step 9:** Repeat steps 2-8 for another subnet ("Acme-Private-Subnet") in a different Availability Zone (e.g., us-east-1b).

[Screenshot: Subnet Creation]

3. Configure Network Address Translation (NAT)

- **Step 1:** In the VPC dashboard, click **NAT Gateways**.
- **Step 2:** Click **Create NAT Gateway**.
- **Step 3:** Choose the VPC ("Acme-VPC") and the subnet ("Acme-Public-Subnet") where the NAT Gateway will be located.
- **Step 4:** Select a NAT Gateway subnet ID (e.g., subnet-xxxxxxxxx).
- **Step 5:** Click **Create NAT Gateway**.

[Screenshot: NAT Gateway Creation]

4. Create Security Groups

- **Step 1:** In the VPC dashboard, click **Security groups**.
- **Step 2:** Click **Create security group**.
- **Step 3:** Enter a security group name (e.g., "Acme-Web-SG").
- **Step 4:** Choose the VPC ("Acme-VPC").
- **Step 5:** Add **inbound rules** based on your application requirements:
    - **HTTP/HTTPS** from **0.0.0.0/0** for public-facing applications.
    - **SSH** from specific IP addresses for administrative access (e.g., your IP address).
    - **Other protocols** as needed.
-
- **Step 6:** Add **outbound rules** based on your application's outbound network traffic (e.g., allow outbound access to the internet).
- **Step 7:** Click **Create security group**.
- **Step 8:** Repeat steps 2-7 for another security group ("Acme-Database-SG") with appropriate inbound and outbound rules.

[Screenshot: Security Group Creation]

5. Configure IAM Roles

- **Step 1:** Navigate to **IAM** in the services menu.
- **Step 2:** Click **Roles**.
- **Step 3:** Click **Create role**.
- **Step 4:** Choose **AWS service** as the type of trusted entity and select the **EC2** service.
- **Step 5:** Click **Next: Permissions**.
- **Step 6:** Search for and select the **AmazonEC2FullAccess** policy.
- **Step 7:** Click **Next: Review**.
- **Step 8:** Enter a role name (e.g., "Acme-EC2-Role") and review the configuration.
- **Step 9:** Click **Create role**.
- **Step 10:** Repeat steps 2-9 to create IAM roles for ECS ("Acme-ECS-Role") and Lambda ("Acme-Lambda-Role") with appropriate permissions.

[Screenshot: IAM Role Creation]

6. Configure KMS Keys

- **Step 1:** Navigate to **KMS** in the services menu.
- **Step 2:** Click **Create key**.
- **Step 3:** Enter a key name (e.g., "Acme-Encryption-Key").
- **Step 4:** Choose **Symmetric** as the key type.
- **Step 5:** Choose **Encrypt/Decrypt** as the key usage.
- **Step 6:** Create a **key policy** to control access to the key.
- **Step 7:** Click **Create key**.

[Screenshot: KMS Key Creation]

Tsakani - Compute & Application Hosting

1. Launch EC2 Instances

- **Step 1:** Navigate to **EC2** in the services menu.
- **Step 2:** Click **Launch Instance**.
- **Step 3:** Choose an **AMI** based on your operating system and application requirements (e.g., Amazon Linux 2 AMI).
- **Step 4:** Select an **instance type** (e.g., t2.micro for small workloads, m4.large for demanding applications).
- **Step 5:** Configure the **network settings**:
    - Choose the created VPC ("Acme-VPC").
    - Select the appropriate subnet ("Acme-Public-Subnet" for public-facing instances, "Acme-Private-Subnet" for private instances).
    - Assign the appropriate security group (e.g., "Acme-Web-SG").
-
- **Step 6:** Configure **storage**:
    - Attach an EBS volume if persistent storage is required.
    - Choose a volume size and volume type (e.g., gp2 for general purpose).
-
- **Step 7:** Configure **other settings** (e.g., key pair, user data).
- **Step 8:** Click **Launch**.

[Screenshot: EC2 Instance Launch]

2. Configure ECS Cluster

- **Step 1:** Navigate to **ECS** in the services menu.
- **Step 2:** Click **Create cluster**.
- **Step 3:** Enter a cluster name (e.g., "Acme-ECS-Cluster").
- **Step 4:** Choose **EC2 Linux** as the launch type.
- **Step 5:** Select the created VPC ("Acme-VPC").
- **Step 6:** Choose an instance type (e.g., t2.micro) and configure the number of instances.
- **Step 7:** Click **Create**.

[Screenshot: ECS Cluster Creation]

3. Deploy Containerized Applications

- **Step 1:** In the ECS dashboard, click **Task Definitions**.
- **Step 2:** Click **Create new task definition**.
- **Step 3:** Choose a family name (e.g., "Acme-Web-Task").
- **Step 4:** Select a **container definition** and specify the container image (e.g., nginx:latest), port mappings, and other settings.

- **Step 5:** Click **Add container** if you need to deploy multiple containers within the same task.
- **Step 6:** Click **Create**.
- **Step 7:** Once the task definition is created, click **Tasks** and then **Create**.
- **Step 8:** Select the created task definition and the cluster ("Acme-ECS-Cluster").
- **Step 9:** Click **Create**.

[Screenshot: ECS Task Definition Creation & Deployment]

4. Configure Lambda Functions

- **Step 1:** Navigate to **Lambda** in the services menu.
- **Step 2:** Click **Create function**.
- **Step 3:** Choose **Author from scratch**.
- **Step 4:** Enter a function name (e.g., "Acme-Data-Processing").
- **Step 5:** Select **Runtime** based on your preferred programming language (e.g., Python 3.9).
- **Step 6:** Configure the **memory** and **timeout** settings based on your function requirements.
- **Step 7:** Click **Create function**.
- **Step 8:** Write the function code using the chosen runtime.
- **Step 9:** Configure **event triggers** for the Lambda function (e.g., S3 event, API Gateway endpoint).

[Screenshot: Lambda Function Creation & Code Editing]

Bushy - Data Storage & Backup

1. Create S3 Buckets

- **Step 1:** Navigate to **S3** in the services menu.
- **Step 2:** Click **Create bucket**.
- **Step 3:** Enter a bucket name (e.g., "acme-manufacturing-data").
- **Step 4:** Choose a region (e.g., us-east-1).
- **Step 5:** Configure access control settings (e.g., block public access) and encryption options (e.g., server-side encryption with KMS) based on your security requirements.
- **Step 6:** Click **Create bucket**.

[Screenshot: S3 Bucket Creation]

2. Configure RDS Database

- **Step 1:** Navigate to **RDS** in the services menu.
- **Step 2:** Click **Create database**.
- **Step 3:** Select the engine (e.g., MySQL, PostgreSQL) and version.
- **Step 4:** Enter a DB instance identifier (e.g., "Acme-Database").

- **Step 5:** Choose a database size and storage type (e.g., General Purpose (SSD) for standard workloads).
- **Step 6:** Configure network settings:
  - Choose the created VPC ("Acme-VPC").
  - Select the appropriate subnet ("Acme-Private-Subnet" for private databases).
  - Assign the appropriate security group ("Acme-Database-SG").
-
- **Step 7:** Click **Create database**.

[Screenshot: RDS Database Creation]

3. Configure DynamoDB Table

- **Step 1:** Navigate to **DynamoDB** in the services menu.
- **Step 2:** Click **Create table**.
- **Step 3:** Enter a table name (e.g., "Acme-Products").
- **Step 4:** Define the primary key (e.g., "ProductId") and attributes (e.g., "ProductName", "Price") for the table.
- **Step 5:** Choose a provisioned throughput based on your expected read and write capacity (e.g., 10 read capacity units, 5 write capacity units).
- **Step 6:** Click **Create table**.

[Screenshot: DynamoDB Table Creation]

4. Configure EBS Volumes

- **Step 1:** In the EC2 dashboard, navigate to **Volumes**.
- **Step 2:** Click **Create Volume**.
- **Step 3:** Choose a volume type (e.g., gp2) and size.
- **Step 4:** Select a volume zone if necessary.
- **Step 5:** Click **Create Volume**.
- **Step 6:** Once the volume is created, you can attach it to an EC2 instance using the **Attach Volume** option in the EC2 instance details page.

[Screenshot: EBS Volume Creation]

5. Configure Glacier Archives

- **Step 1:** Navigate to **Glacier** in the services menu.
- **Step 2:** Click **Create vault**.
- **Step 3:** Enter a vault name (e.g., "Acme-Archives").
- **Step 4:** Choose a region for the vault (e.g., us-east-1).
- **Step 5:** Click **Create vault**.
- **Step 6:** Once the vault is created, you can upload data to the vault using the AWS CLI or SDK.

[Screenshot: Glacier Vault Creation]

Yamkelani - Performance & Optimization

1. Configure CloudFront CDN

- **Step 1:** Navigate to **CloudFront** in the services menu.
- **Step 2:** Click **Create distribution**.
- **Step 3:** Choose **Origin domain** (e.g., S3 bucket, EC2 instance).
- **Step 4:** Configure **distribution settings** (e.g., caching behavior, error pages).
- **Step 5:** Click **Create distribution**.

[Screenshot: CloudFront Distribution Creation]

2. Configure ElastiCache

- **Step 1:** Navigate to **ElastiCache** in the services menu.
- **Step 2:** Click **Create cache cluster**.
- **Step 3:** Choose **engine** (e.g., Redis) and **version**.
- **Step 4:** Enter a cache cluster name (e.g., "Acme-Cache").
- **Step 5:** Configure **cache nodes** and **instance type** based on your performance requirements.
- **Step 6:** Choose a **cache subnet group**.
- **Step 7:** Click **Create cache cluster**.

[Screenshot: ElastiCache Cluster Creation]

3. Configure Route 53

- **Step 1:** Navigate to **Route 53** in the services menu.
- **Step 2:** Click **Hosted zones**.
- **Step 3:** Click **Create hosted zone**.
- **Step 4:** Enter a **domain name** (e.g., "acmemanufacturing.com").
- **Step 5:** Click **Create**.
- **Step 6:** Create **record sets** for the domain, specifying the type (e.g., A, CNAME) and target (e.g., EC2 instance, S3 bucket).

[Screenshot: Route 53 Hosted Zone & Record Set Creation]

4. Monitor with CloudWatch

- **Step 1:** Navigate to **CloudWatch** in the services menu.
- **Step 2:** Click **Metrics**.
- **Step 3:** Choose **Create metric**.
- **Step 4:** Enter a **metric name** (e.g., "EC2-CPU-Utilization").
- **Step 5:** Select a **namespace** (e.g., AWS/EC2).

- **Step 6:** Configure the **metric dimensions** (e.g., InstanceId).
- **Step 7:** Click **Create metric**.
- **Step 8:** Create **alarms** based on specific metric thresholds, triggering notifications or actions when certain conditions are met (e.g., CPU utilization above 80%).

[Screenshot: CloudWatch Metric & Alarm Creation]

Lusanda - Security & Operations

1. Implement GuardDuty

- **Step 1:** Navigate to **GuardDuty** in the services menu.
- **Step 2:** Click **Get Started**.
- **Step 3:** Select the **regions** where you want to enable GuardDuty (e.g., us-east-1).
- **Step 4:** Click **Start monitoring**.

[Screenshot: GuardDuty Enablement]

2. Automate with CloudFormation

- **Step 1:** Navigate to **CloudFormation** in the services menu.
- **Step 2:** Click **Create stack**.
- **Step 3:** Choose **Template source** (e.g., Upload a template file).
- **Step 4:** Specify the **stack name** and **parameters**.
- **Step 5:** Click **Create stack**.

[Screenshot: CloudFormation Stack Creation]

3. Audit with CloudTrail

- **Step 1:** Navigate to **CloudTrail** in the services menu.
- **Step 2:** Click **Create trail**.
- **Step 3:** Enter a **trail name** (e.g., "Acme-CloudTrail").
- **Step 4:** Choose a **bucket** for storing the trail logs (e.g., "acme-manufacturing-data" S3 bucket created earlier).
- **Step 5:** Configure **event selection** (e.g., all API calls, data events).
- **Step 6:** Click **Create trail**.

[Screenshot: CloudTrail Creation]

4. Manage with Systems Manager

- **Step 1:** Navigate to **Systems Manager** in the services menu.
- **Step 2:** Click **Instances**.
- **Step 3:** Click **Manage instances** for a specific instance.

- **Step 4:** Configure **instance management** settings (e.g., software patching, inventory collection).
- **Step 5:** Implement **automation documents** to automate tasks (e.g., system updates, security configurations).

[Screenshot: Systems Manager Instance Management & Automation]

Conclusion:

By following these detailed step-by-step instructions, each team member can successfully complete their assigned tasks for Acme Manufacturing's cloud migration.

Remember:

- **Thorough documentation is essential:** Every step should be documented, including configurations, scripts, and encountered challenges.
- **Regular testing is critical:** Frequent testing of the infrastructure and applications ensures a smooth migration and avoids potential problems.

This comprehensive guide provides a solid foundation for the team's success in implementing a secure and efficient cloud environment for Acme Manufacturing.