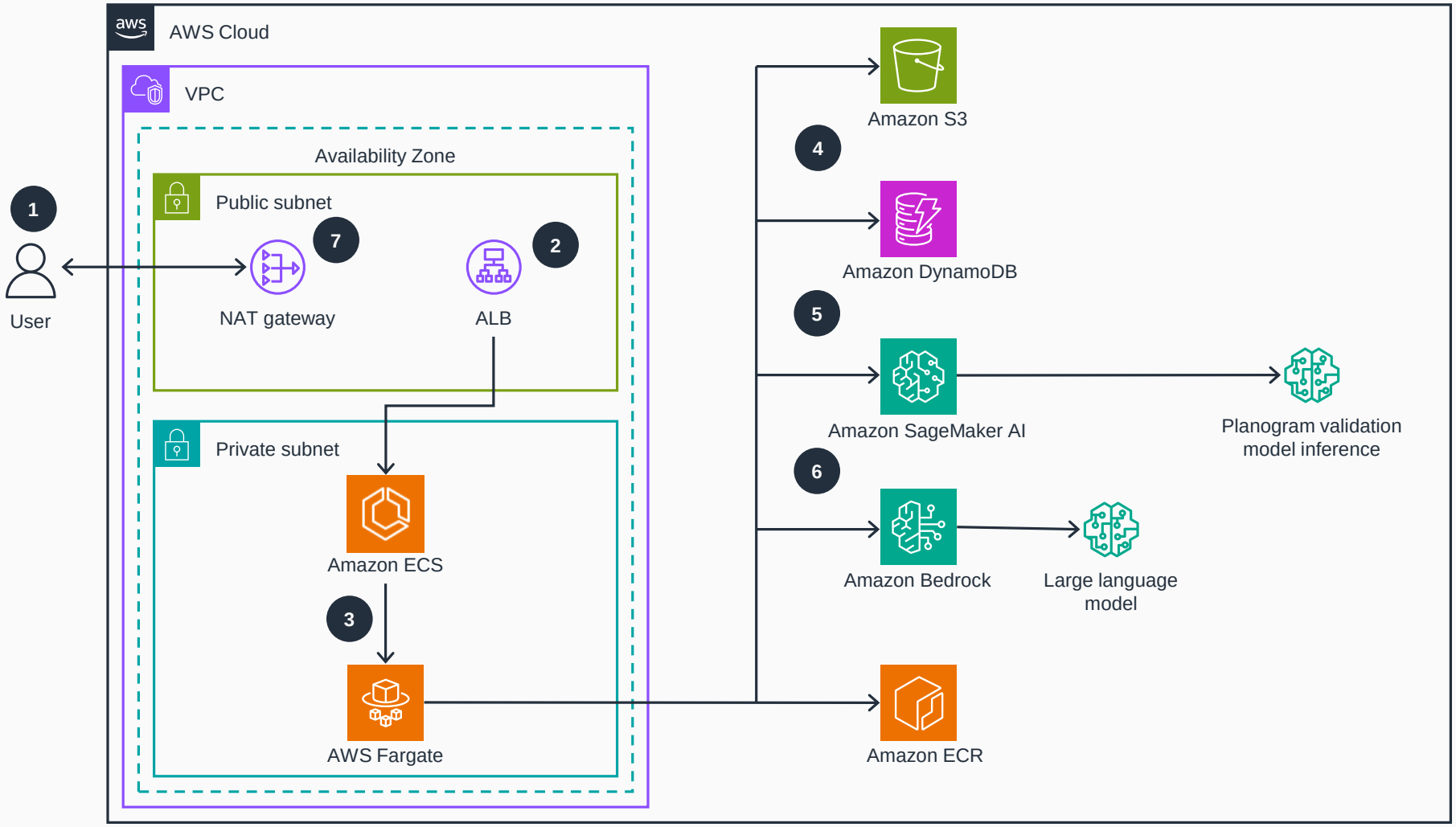
Guidance for Managing Planograms with Amazon Bedrock

This architecture diagram shows how to reduce the time to create retail planograms, enabling planogram generation and validation using generative AI.



1. The user enters a prompt for the planogram to be generated using the application UI. The prompt includes the planogram dimensions and placement instructions. The user can also specify additional rules to be applied for the planogram.

2. An Application Load Balancer (ALB) in the virtual private cloud (VPC) public subnet takes in the user request and distributes the workload evenly to the application deployed on AWS. The security group configuration in **Amazon Elastic Container Service (Amazon ECS)** helps ensure that only inbound traffic from allowed sources is processed.

3. The planogram application takes in requests and processes them in containers, storing container images in **Amazon Elastic Container Registry (Amazon ECR)**. The tasks run using **Amazon ECS**, and **AWS Fargate** acts as the compute engine for **Amazon ECS** tasks, providing a serverless way to run containers without managing servers or clusters.

4. **Amazon Simple Storage Service (Amazon S3)** stores the planogram images generated by the application and other static web assets. The planogram instructions, the underlying supported models, the rules, the templates, and the product metadata persist in **Amazon DynamoDB** tables.

5. **Amazon Bedrock** receives the request to generate the planogram over an API call and invokes the model the user specified in the application UI. The provided prompt is enhanced by the placement rules stored in **DynamoDB**. **Amazon S3** stores the generated planogram for the application UI to display.

6. For validation, the user provides a planogram and a captured image of the shelf. **Amazon SageMaker AI** processes this validation request through API calls using the planogram validation model inference.

7. The application returns the generated planogram and the results of the validation to you using the NAT gateway, where the application UI renders corresponding images pulled from **Amazon S3**.

Reviewed for technical accuracy May 5, 2025

AWS Reference Architecture