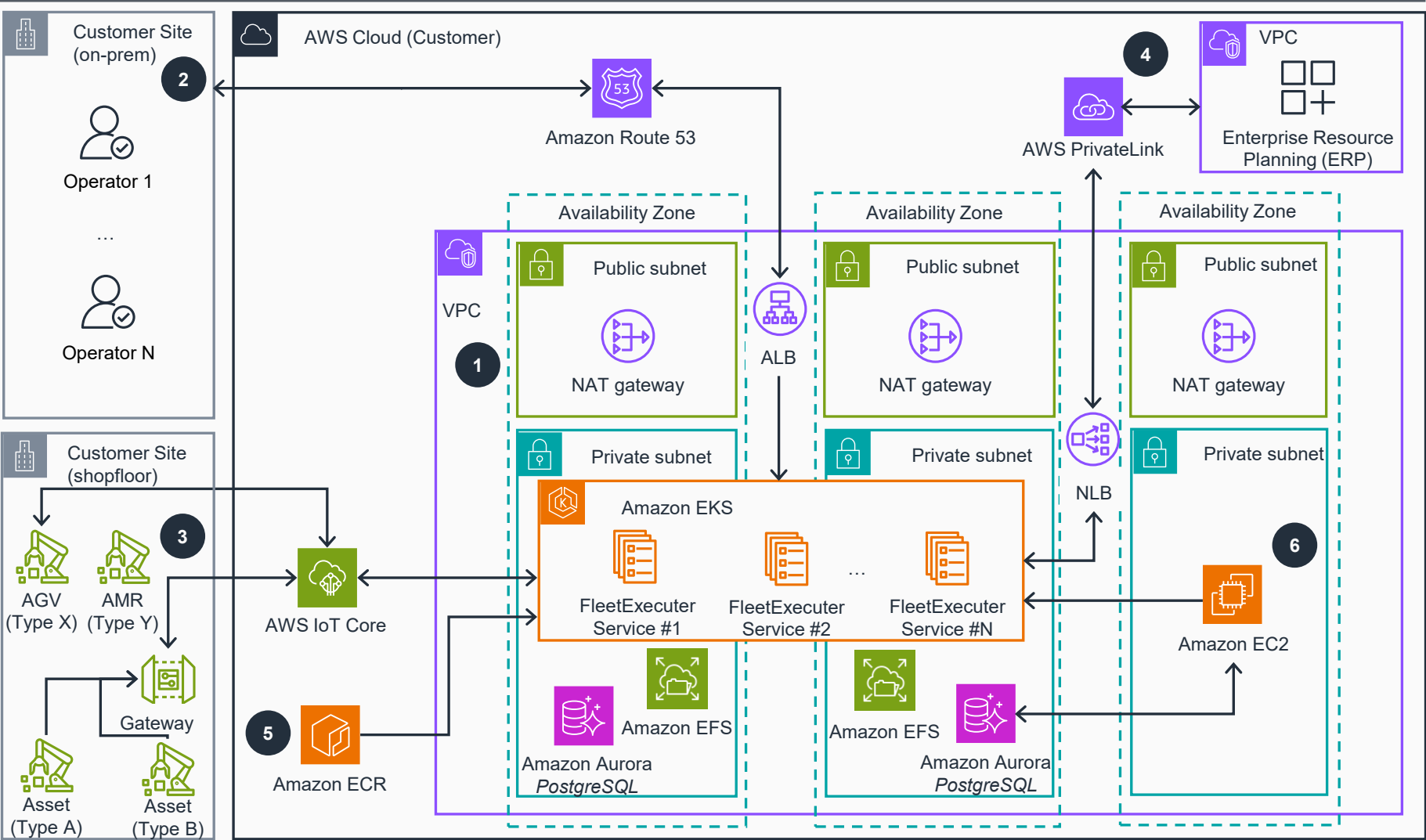


Guidance for MHP FleetExecuter on AWS

Central Primary Control for Autonomous Mobile Robot Fleets and Intralogistic Assets

This architecture diagram shows data flow between MHP FleetExecuter, customer shopfloor AGVs, AMRs, equipment, and enterprise systems through AWS. MHP FleetExecuter is available as customer-deployable software or fully managed service by MHP.



- 1 MHP FleetExecuter's main logic runs in containers on **Amazon Elastic Kubernetes Service (Amazon EKS)** across two Availability Zones (AZ). **Amazon Aurora** (PostgreSQL) stores application data with replicas. **Amazon Elastic File Service (Amazon EFS)** provides StatefulSet storage with mount targets in both AZs. NAT Gateways enable internet access, and **Amazon Route 53** handles the DNS.
- 2 Operators access the FleetExecuter Service through the web browser, routed via **Amazon Route 53** to the internet-facing **Application Load Balancer (ALB)**, then to **Amazon EKS** services.
- 3 AGVs and AMRs from different vendors communicate with FleetExecuter via **AWS IoT Core** using MQTT. An IoT gateway manages shopfloor equipment and communicates with **AWS IoT Core** via MQTT. Alternatively, customers can host the MQTT broker within the **Amazon EKS** cluster.
- 4 Customers integrate their Enterprise Resource Planning (ERP) systems with FleetExecuter through **AWS PrivateLink**. ERP systems in separate VPCs connect securely via an internal **Network Load Balancer (NLB)** routing to **Amazon EKS**. The ERP system creates transport orders and receives real-time status updates.
- 5 **Amazon Elastic Container Registry (Amazon ECR)** stores MHP FleetExecuter Service container images and supporting services. **Amazon EKS** worker nodes pull images from **Amazon ECR** during deployments.
- 6 A bastion host running in an **Amazon Elastic Compute Cloud (Amazon EC2)** instance in a separate AZ enables administrative access to **Amazon EKS**, **Amazon Aurora** database, and **Amazon EFS** via **AWS Systems Manager**.

