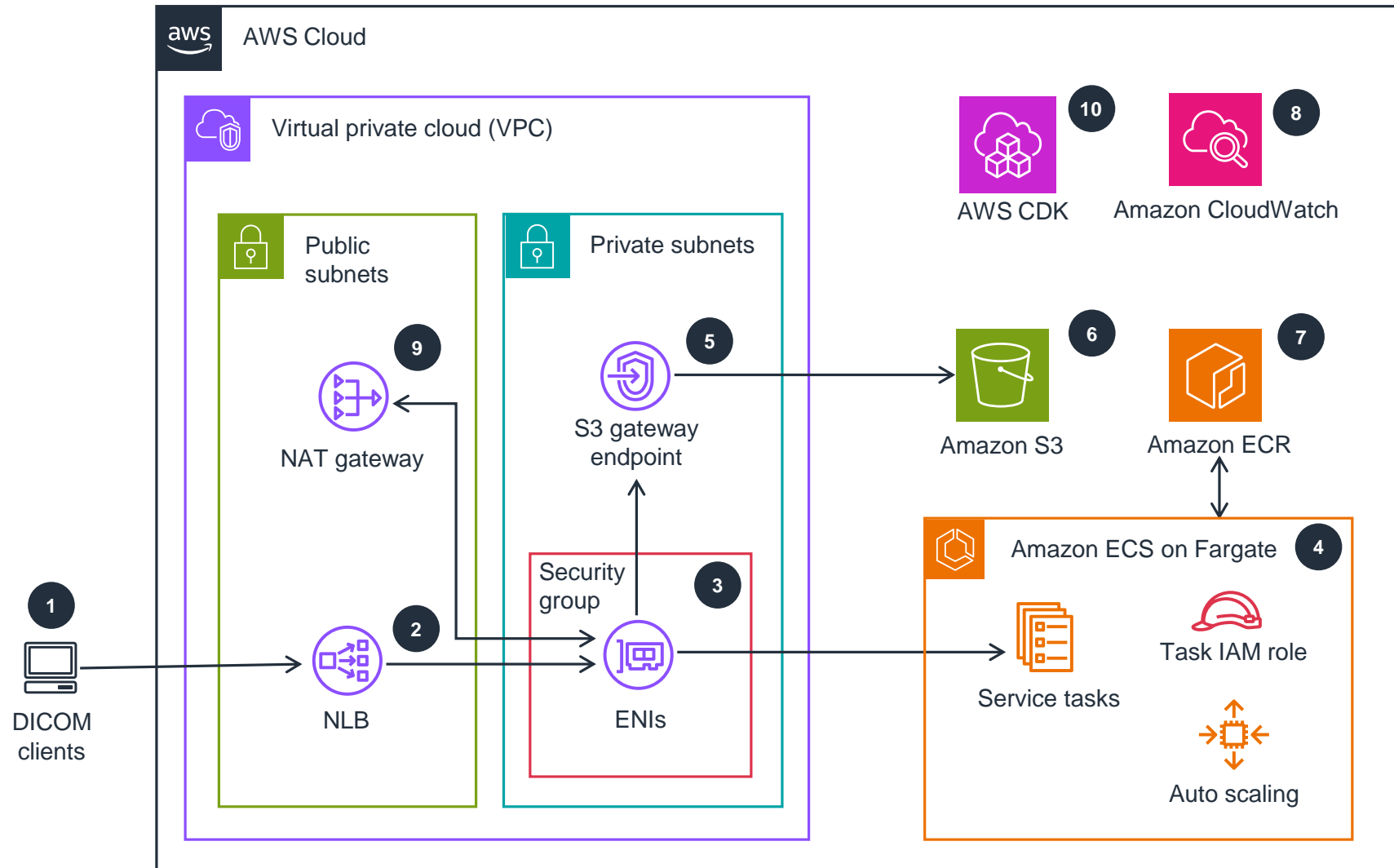


# Guidance for Receiving Digital Imaging and Communications in Medicine (DICOM) Images in Amazon S3

This architecture diagram shows how to deploy containerized DIMSE services using AWS services.



- 1 Clients connect to the application through a **Network Load Balancer (NLB)** and send DIMSE C-STORE requests to the backend service.
- 2 The **NLB** is deployed to a public subnet. Targets are one or more **Amazon Elastic Container Service (Amazon ECS)** on **AWS Fargate** task elastic network interfaces (ENIs) in a private subnet.
- 3 An **Amazon Virtual Private Cloud (Amazon VPC)** security group allows fine-grained network access control to the application.
- 4 Connections are passed to **Amazon ECS** on **Fargate** service tasks. Service task scaling is controlled through an **AWS Auto Scaling** rule, allowing the service to adjust its capacity according to demand. Container permissions are granted through the task **AWS Identity Access and Management (IAM)** role, avoiding the use of hard-coded credentials.
- 5 Tasks access **Amazon Simple Storage Service (Amazon S3)** using reliable and private connectivity provided by a **VPC gateway endpoint**.
- 6 Received DICOM images are stored in **Amazon S3**. Optionally, DICOM metadata can also be extracted and stored.
- 7 Container definitions are stored in **Amazon Elastic Container Registry (Amazon ECR)** and are used by **Amazon ECS** on **Fargate** during deployment.
- 8 **Amazon CloudWatch** collects container logs and metrics.
- 9 The Guidance components access the internet using a **VPC NAT gateway**, avoiding the need to provision public IP addresses for all tasks.
- 10 The Guidance is deployed as infrastructure-as-code using **AWS Cloud Development Kit (AWS CDK)**.

