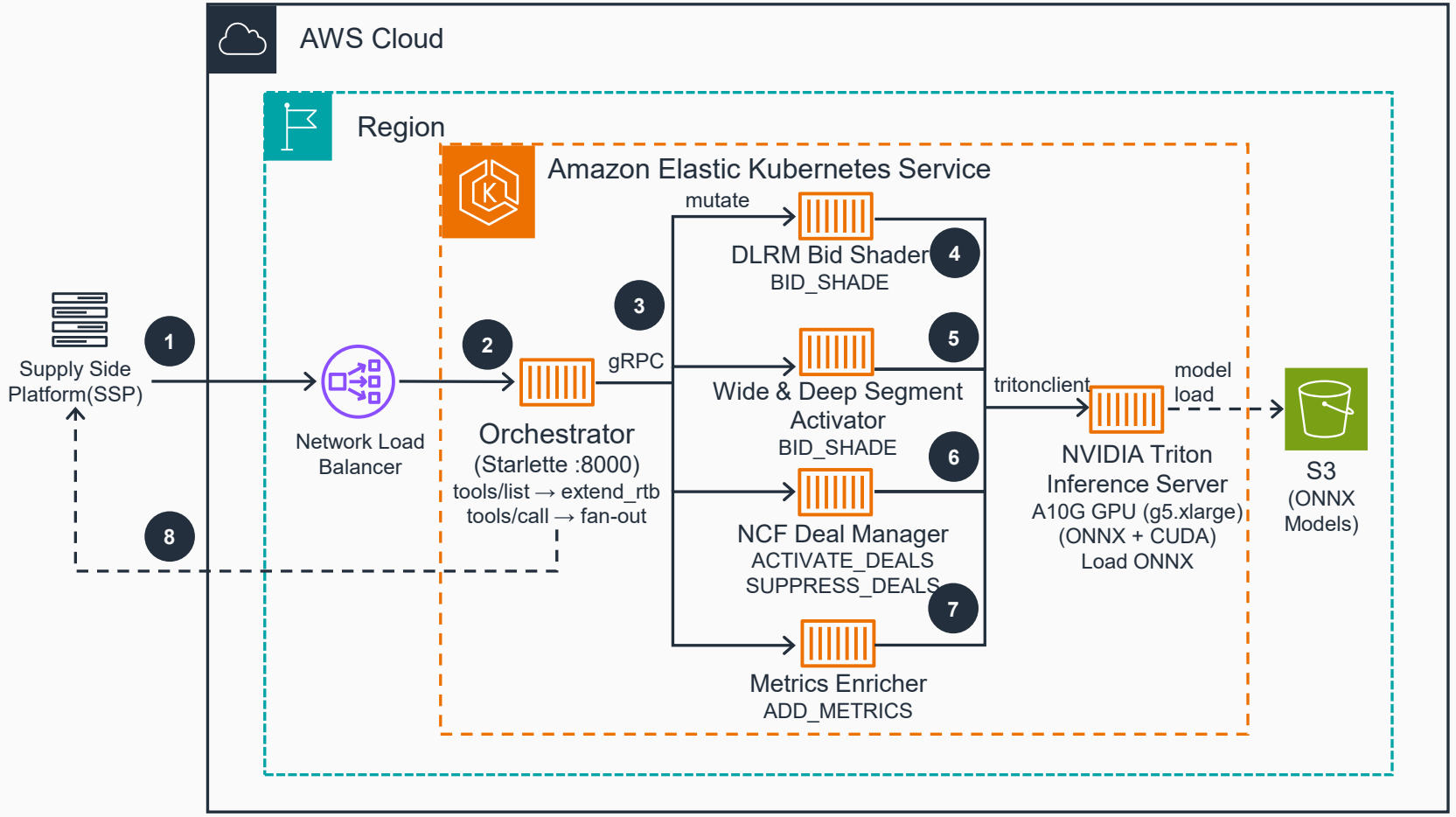
Guidance for Accelerator-Optimized Agentic Bidding on AWS

This architecture shows how accelerator-optimized compute can enable agentic real-time bid-stream mutations for OpenRTB auctions using Triton Inference Server on Amazon EKS with GPU-accelerated inference.



1. The Requester – for example a Supply Side Platform (SSP) sends OpenRTB bid request with JSON Web Token (JWT) tokens for session access.
2. The request routes through an **Network Load Balancer** to the **Amazon Elastic Kubernetes Service (EKS)** cluster's Orchestrator container.
3. The Orchestrator on CPU nodes verifies the JWT token against Cognito's JSON Web Key Set (JWKS) endpoint, then fans out the request to all 4 ARTF containers (Deep Learning Recommendation Model (DLRM), Wide & Deep, Neural Collaborative Filtering (NCF), and Metrics Enricher) in parallel.
4. The NVIDIA model containers (DLRM, Wide & Deep, NCF) call Triton Inference Server via tritonclient Python library, which runs GPU-accelerated inference using Open Neural Network Exchange (ONNX) Runtime with Compute Unified Device Architecture(CUDA) Execution Provider on A10G GPUs.
5. DLRM predicts click-through rate and computes an optimal shaded bid price. Wide & Deep scores user-segment affinities and activates audience segments above threshold.
6. NCF scores user-deal relevance to activate high-affinity deals and suppress poor matches.
7. The rule-based Metrics Enricher adds viewability and brand-safety scores.
8. The Orchestrator merges all mutations from the four containers into a single OpenRTB response and returns it to the requester.

Reviewed for technical accuracy June 15, 2026
© 2026, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Reference Architecture