Architecture diagrams for retail

Personalization in Magento eCommerce on AWS for CPG

Personalization in Magento eCommerce on AWS for CPG: Architecture diagrams for retail

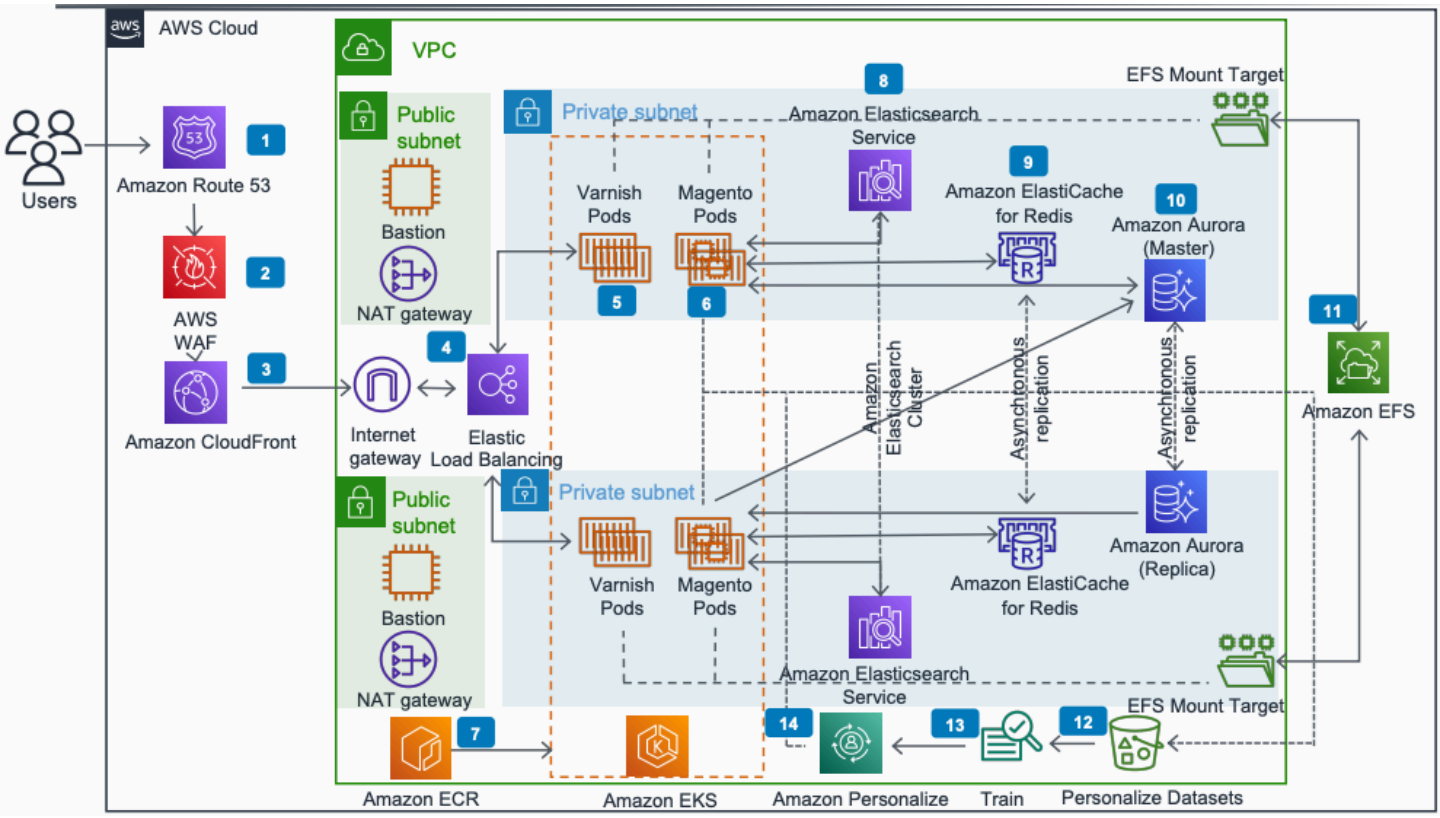
Table of Contents

Personalization in Magento eCommerce on AWS for CPG

Publication date: **May 20, 2021 (Diagram history)**

With this architecture, you can deploy containerized Magento ecommerce software on AWS with personalization for consumer packaged goods (CPG). Product recommendations based on user behavior increase conversions and customer engagement. You use Amazon Personalize for ML-based recommendations and Amazon Elastic Kubernetes Service for container orchestration.

Magento personalization diagram



The following steps describe the architecture:

1. Amazon Route 53 provides DNS configuration.
2. AWS WAF protects Magento against common web exploits.
3. Amazon CloudFront speeds up the distribution of static and dynamic web content as a content delivery network (CDN).
4. Elastic Load Balancing (Application Load Balancer) distributes traffic across Varnish pods running in Amazon EKS through a Kubernetes ingress across multiple Availability Zones.

5. Varnish Cache runs as Kubernetes pods on Amazon EKS. The Enterprise version provides better features for cloud backends and cache-purge across dynamic hosts.
6. Magento Open Source or Commerce edition runs as Kubernetes pods on Amazon EKS. Each pod includes the Magento application, Nginx web server, and PHP.
7. Amazon Elastic Container Registry hosts the container images for both Varnish and Magento.
8. Amazon OpenSearch Service provides a managed search solution for the Magento catalog.
9. Amazon ElastiCache for Redis provides a caching layer for the database.
10. Use Amazon Aurora or Amazon RDS for the relational database. These provide high availability and multi-AZ configuration.
11. Use Amazon EFS to access shared configuration across Varnish and shared media assets across Magento pods.
12. Use the Amazon Personalize extension for Magento to export user historical data into an Amazon Simple Storage Service bucket.
13. Amazon Personalize trains on the historical data exported from Magento. It creates a custom solution and campaign as a private ML model hosted in your AWS account.
14. After activating the Amazon Personalize model, it displays product recommendations to end users. Real-time data interaction indicators let Amazon Personalize learn from user actions such as adding products to carts or wish lists.

Further reading

For additional information, see the following resources:

- AWS Architecture Icons
- AWS Architecture Center
- AWS Well-Architected

Diagram history

To receive updates about this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date

Initial publication	Reference architecture diagram first published.	May 20, 2021

RSS subscription requirement

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.