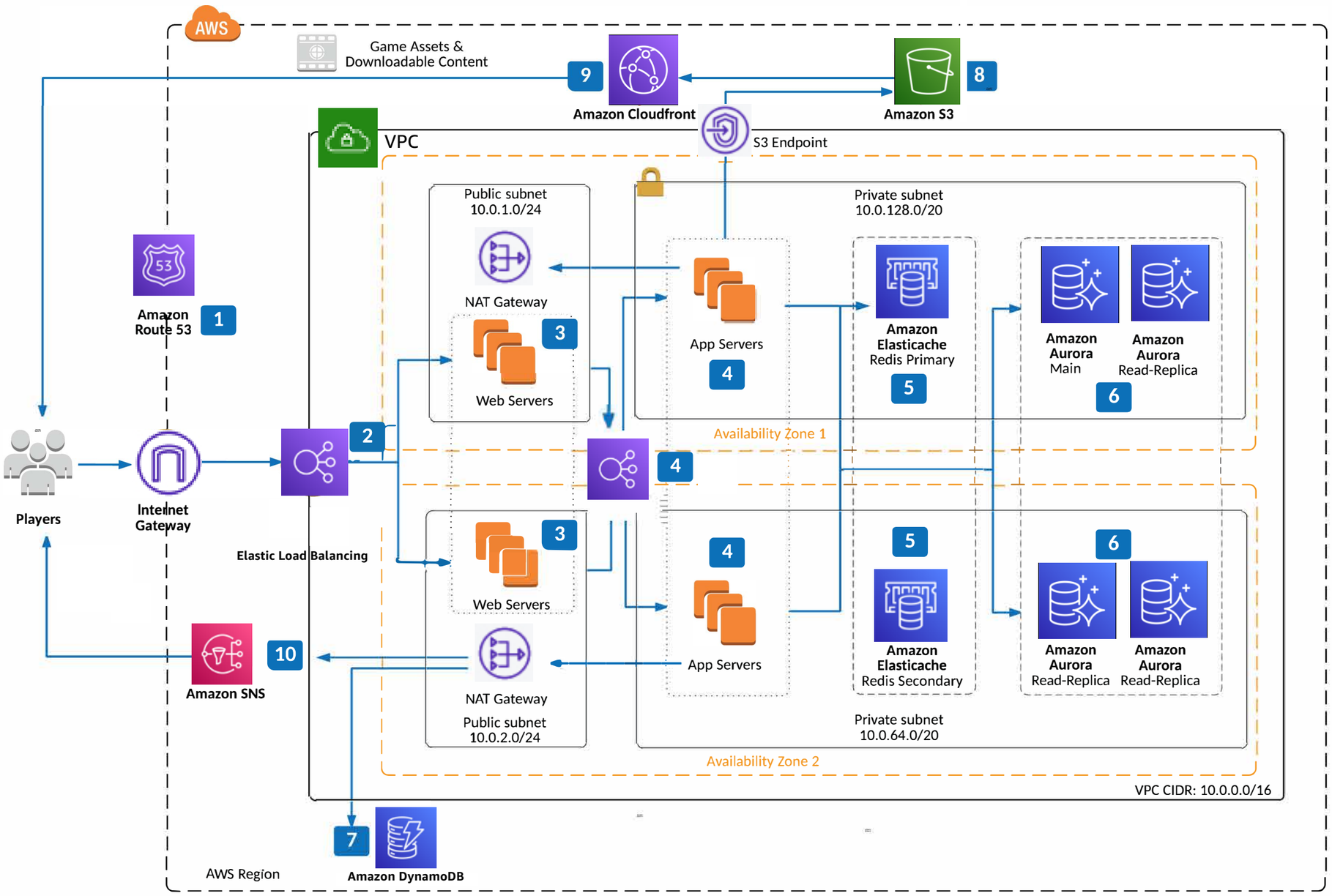
Asynchronous Online Gaming

Highly Available, Scalable & Elastic to Support Millions of Players

This architecture is intended for Mobile & Online Games. These workloads are a natural fit for running on Amazon Web Services, due to unexpected traffic patterns & highly dem&ing request rates. AWS provides the flexibility to start small & power up your architecture in response to your players. Scale up & scale down your architecture to make sure you are only paying for resources that are driving the best experience for your game. Use our managed services for popular caching & database technologies, & leverage this architecture that captures the best practices of some of the largest games running on AWS today.



1. Leverage our 100% SLA for **Amazon Route53** to make sure your players are always able to discover your service endpoints. Use the in-built routing policies to route users based on latency or geography.
2. Route users to your backend using an **Elastic Load Balancing** that scales automatically for incoming traffic. Keep your players data secure in transit via HTTPS & by leveraging the SSL termination capabilities of the ELB.
3. Launch your Web Servers running on **Amazon Elastic Compute Cloud** (EC2) in an Auto Scaling Group that spans Multiple Availability Zones. We recommend the M4 instance type with Enhanced Networking & EBS Optimized enabled.
4. If you separate your app tier from your web tier, leverage an internal ELB. This ELB provides additional benefits of added security by residing in a private subnet & making sure no external traffic overwhelms your app tier.
5. **Amazon Elasticache** for Redis provides a fully managed solution that enhances robustness & reduces the cost of installing, operating & maintaining a highly available & scalable Redis cluster. Leverage Multi-AZ Elasticache in your game to provide automated disaster recovery & a scalable tier with read replicas if needed.
6. **Amazon Aurora**, a MySQL compatible database provides very high read & write throughput, up to 64TB 6-way replicated storage & up to 15 low latency read replicas in Multi-AZ. We recommend Aurora as a fast, scalable robust database if you are using a relational database like MySQL. Gaming customers have seen 2-3x reduction in cost after migration to Aurora.
7. Your game will also benefit from our high speed, low latency managed No-SQL database, **Amazon DynamoDB** that provides predictable performance & scalability.
8. Use **Amazon Simple Storage Service** (S3) to store your game assets, DLC & log files
9. generated by your servers. As your user base grows geographically use Amazon Cloudfront as a globally distributed cache for content.
10. For Push Notifcations, use our managed **Amazon Simple Notification Service** (SNS) service with out of the box support for Apple, Google, Amazon & Windows platforms.

Reviewed for technical accuracy March 24, 2021

AWS Reference Architecture