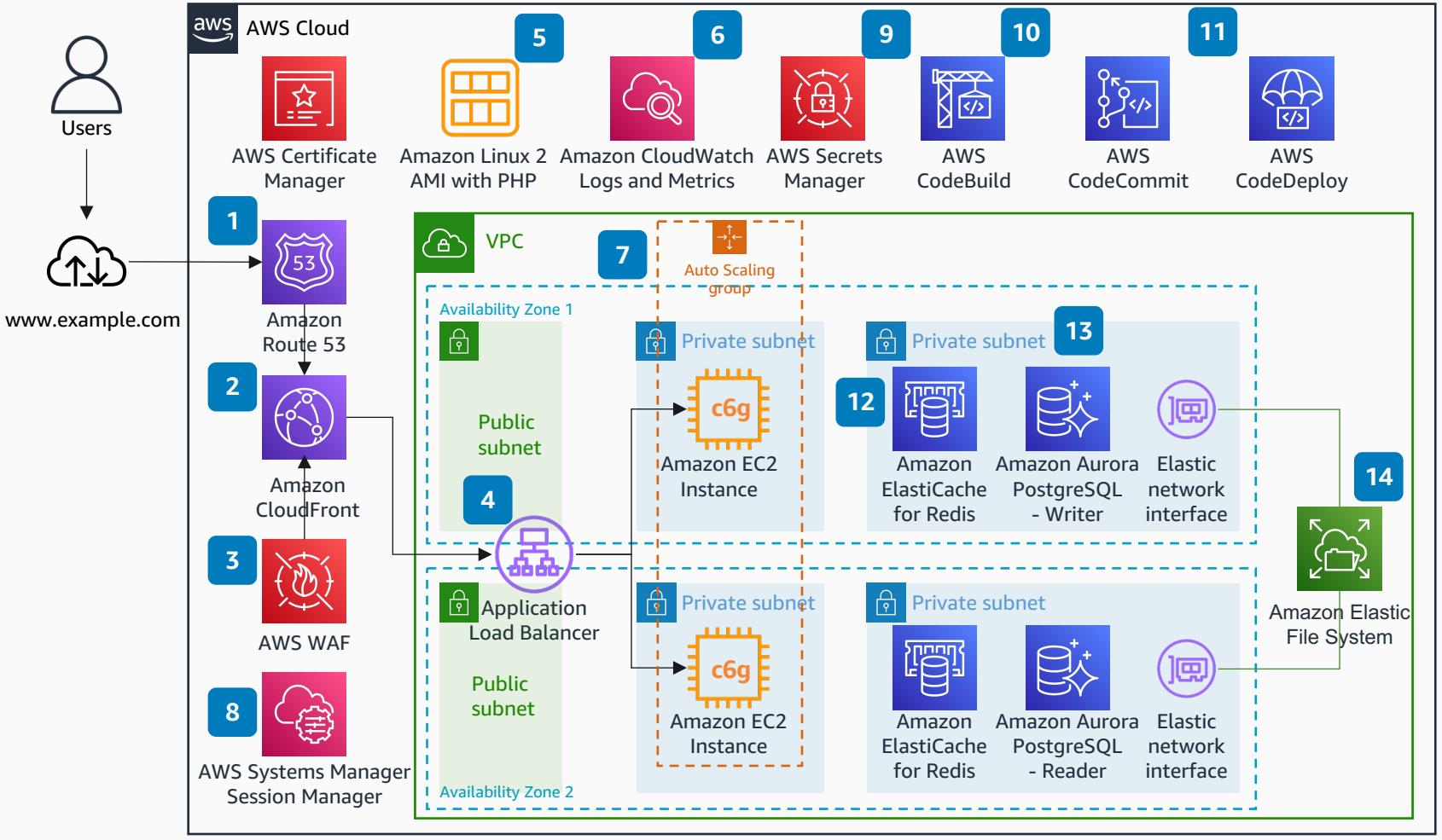
Scaling PHP Applications on AWS

Running highly-available, performant and secure PHP applications



AWS Reference Architecture

1. **Amazon Route 53** routes end-user requests resolving Domain Name Service (DNS).
2. **Amazon CloudFront** caches content and accelerates delivery, leveraging global points of presence. **CloudFront** also handles SSL ending, integrating with **Amazon Certificate Manager**, which automatically creates and renews SSL certificates with no cost.
3. **AWS Web Application Firewall** integration with **CloudFront** and **Application Load Balancer** mitigates OWASP top 10 application vulnerabilities.
4. The **Application Load Balancer** routes HTTP/S requests to **EC2** instances running on private subnets.
5. An **Amazon Linux 2 AMI** contains the PHP and other needed binaries, such as the **AWS SDK** for PHP.
6. The **Amazon CloudWatch** Agent installed on the **Amazon Linux 2 AMI** streams application logs, additional host-level metrics, and custom business metrics.
7. **Amazon EC2 Auto Scaling** manages the instance launch based on metrics such as CPU and memory. It uses **Amazon Graviton** instances for cost optimization.
8. Using **Systems Manager Session Manager**, connect to **EC2** instances with web-based sessions on the **AWS Console**. There is no need for key pairs of SSH ports to be open.
9. Database credentials are securely stored on **AWS Secrets Manager**. Using the **AWS SDK** for PHP, the application code retrieves the credentials stored on **Secrets Manager** though an **IAM** Role.
10. Application code is safely stored on **AWS CodeCommit** using the familiar GitHub command line interface (CLI).
11. **AWS CodePipeline** implements continuous integration and continuous deployment (CI/CD), orchestrating code deployment using an **AWS CodeDeploy** hook that triggers when new **EC2** instances are launched.
12. **Amazon ElastiCache** for Redis caches session data.
13. **Amazon Aurora** Multi-AZ enables high-availability. Application connects via DNS endpoint that handles failover automatically in case of failure. The **Aurora** reader endpoint handles read operations, offloading **Aurora** writer instance load.
14. **Amazon Elastic File System (Amazon EFS)** stores and shares web content with the **Auto Scaling** group.

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.