The architecture model is a multi-layered architecture design which aims to separate between various system’s components. This architecture model provide several benefits:

- It provides a highly-available environment which maintains the system’s availability in case of service interruption. This ensures a zero-downtime environment.

- It provide a crucial level of security. By separating the system’s components, it streamlines the enforcement of: access control, DDOS attacks protection, firewalls management, public/private instances and network-subnets isolation, applications/database level of isolation, to ensure a secure environment.

- This model also helps in implementing a fault tolerant system. Fault tolerance helps against service interruption with maintaining the system’s performance.

- Scalability also is a feature of the multi-layered architecture. This model allows us to scale the system in and out to accommodate traffic surges.

- Load balancing is another feature of this model which allows us to balance users traffic among different servers to combat any overwhelming or unusual traffic that could degrade the application performance.

- In this model, Content Delivery Networks (CDN) provide caching mechanism which help enhancing applications performance as well as offloading some of the traffic from the application servers.

------------------------------------------------------------------------------------------------------

To roll-out an application update with zero-downtime, we could follow the green/blue deployment mechanism. *(or use a cloud service that provide green/blue deployment to automate the below process – example: AWS CodeDeploy)*

Blue/Green deployment steps can be summarized as:

1. Provision a new infrastructure. We could utilize Infrastructure as a Code tools (e.g. Terraform, CloudFormation).

2. Deploy the latest version of the foo.com application. We could utilize configuration management tools such Puppet, Chef to roll-out the new application across all the instances.

3. Now the infrastructure is ready and foo.com is installed, we can perform application-testing to ensure that the new infrastructure is ready to accept traffic.

4. Let the load-balancers start routing the traffic to the newly provisioned infrastructure gradually. Traffic routing could be shifted in percentages.

3. Once the new infrastructure managed to accommodate all the traffic, it becomes safe to terminate the old instances.

AWS services that help in applications deployment and infrastructure

provisioning:

CodeDeploy: To automatically & rapidly release and deploy new updates and avoid application downtime during the deployment process.

OpsWorks: Managed service to run applications with the ability to control the underlying resources.

Elastic Beanstalk: Managed service for code deployment without warring about the underlying infrastructure.

CloudFormation: Infrastructure as a Code tool to unify and control the model, types, and versions of the infrastructure deployment (such as compute, networking, users management).