AWS OpsWorks Deployment Strategies

# AWS OpsWorks Deployment Strategies

#### **NOTE: Advanced Topic required for DevOps Professional Exam Only**

## All at Once Deployment

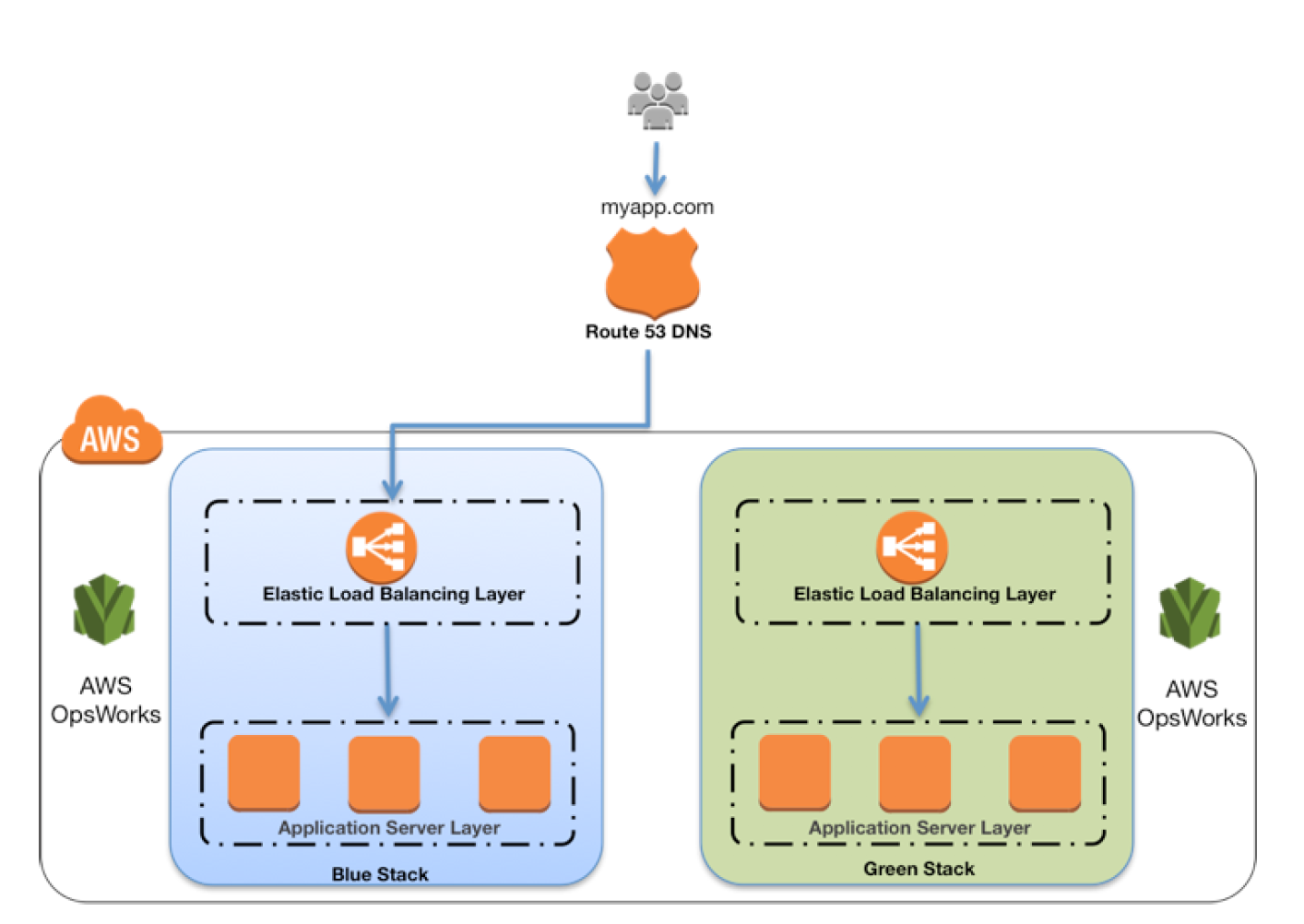
* OpsWorks Stacks does not automatically deploy updated code to online instances, and needs to be done manually
* [Deploy command](https://docs.aws.amazon.com/opsworks/latest/userguide/workingapps-deploying.html) (for apps) or [Update Custom Cookbooks command](https://docs.aws.amazon.com/opsworks/latest/userguide/workingstacks-commands.html) (for cookbooks) helps deploy the update to every instance concurrently
* Approach is simple and fast, but leads to a downtime incase of error
* OpsWorks allows rollback to restore previously deployed app version
* By default, AWS OpsWorks Stacks stores the five most recent deployments, which allows you to roll back up to four versions

## Rolling Deployment

* A rolling deployment updates an application on a stack’s online application server instances in multiple phases.
* With each phase, a subset of the online instances can be updated and verified to be successful before starting the next phase.
* In case of any issues, the instances running the old app version can continue to handle incoming traffic until the issues are resolved.
* Steps to perform Rolling deployment
  + Deploy the app on a single application server instance.
  + The instance can be deregistered from the load balancer, to prevent it from serving traffic
  + Verify the app is working fine
  + Deploy the update on the remainder of instances

## Blue Green Deployment

* Blue Green deployment can be achieved using separate stack for each phase of the application’s lifecycle.
* Different stacks are sometimes referred to as environments like development, staging, production etc.
  + Blue environment is the production stack, which hosts the current application.
  + Green environment is the staging stack, which hosts the updated application.
* Development and testing can be performed on stacks, which are not publicly accessible, and when ready the traffic can be switched.
* **Steps for Blue Green deployment with OpsWorks Stacks stacks, in conjunction with Route 53 and a pool of ELB load balancers**
  + Attach unused ELB from the pool to the green stack’s application server layer
  + After all of the green stack’s instances have passed the ELB health check, the weights in Route 53 can be changed to route traffic gradually from Blue to Green stack.
  + Once the Green stack works fines and is ready to handle all traffic
  + Detach the load balancer from the old blue stack’s application server layer and return it to the pool
  + Blue stack can be retained for some time, so that if any issues the update can be rolled back by reversing the procedure to direct incoming traffic back to the old blue stack

                        OpsWorks Blue Green Deployment

## AWS Certification Exam Practice Questions

1. You company runs a complex customer relations management system that consists of around 10 different software components all backed by the same Amazon Relational Database (RDS) database. You adopted AWS OpsWorks to simplify management and deployment of that application and created an AWS OpsWorks stack with layers for each of the individual components. An internal security policy requires that all instances should run on the latest Amazon Linux AMI and that instances must be replaced within one month after the latest Amazon Linux AMI has been released. AMI replacements should be done without incurring application downtime or capacity problems. You decide to write a script to be run as soon as a new Amazon Linux AMI is released. Which solutions support the security policy and meet your requirements? Choose 2 answers
   1. Assign a custom recipe to each layer, which replaces the underlying AMI. Use AWS OpsWorks life-cycle events to incrementally execute this custom recipe and update the instances with the new AMI. (AMI cannot be updated using recipes)
   2. **Create a new stack and layers with identical configuration, add instances with the latest Amazon Linux AMI specified as a custom AMI to the new layer, switch DNS to the new stack, and tear down the old stack.** (Blue-Green Deployment)
   3. Identify all Amazon Elastic Compute Cloud (EC2) instances of your AWS OpsWorks stack, stop each instance, replace the AMI ID property with the ID of the latest Amazon Linux AMI ID, and restart the instance. To avoid downtime, make sure not more than one instance is stopped at the same time. (Instances cannot be updated by updating the AMI id and needs to be launched anew)
   4. Specify the latest Amazon Linux AMI as a custom AMI at the stack level, terminate instances of the stack and let AWS OpsWorks launch new instances with the new AMI. (Would result in downtime)
   5. **Add new instances with the latest Amazon Linux AMI specified as a custom AMI to all AWS OpsWorks layers of your stack, and terminate the old ones.**(Disposable Rolling deployment)
2. A company has developed a Ruby on Rails content management platform. Currently, OpsWorks with several stacks for dev, staging, and production is being used to deploy and manage the application. Now the company wants to start using Python instead of Ruby. How should the company manage the new deployment?
   1. Update the existing stack with Python application code and deploy the application using the deploy life-cycle action to implement the application code.
   2. **Create a new stack that contains a new layer with the Python code. To cut over to the new stack the company should consider using Blue/Green deployment**
   3. Create a new stack that contains the Python application code and manage separate deployments of the application via the secondary stack using the deploy lifecycle action to implement the application code.
   4. Create a new stack that contains the Python application code and manages separate deployments of the application via the secondary stack