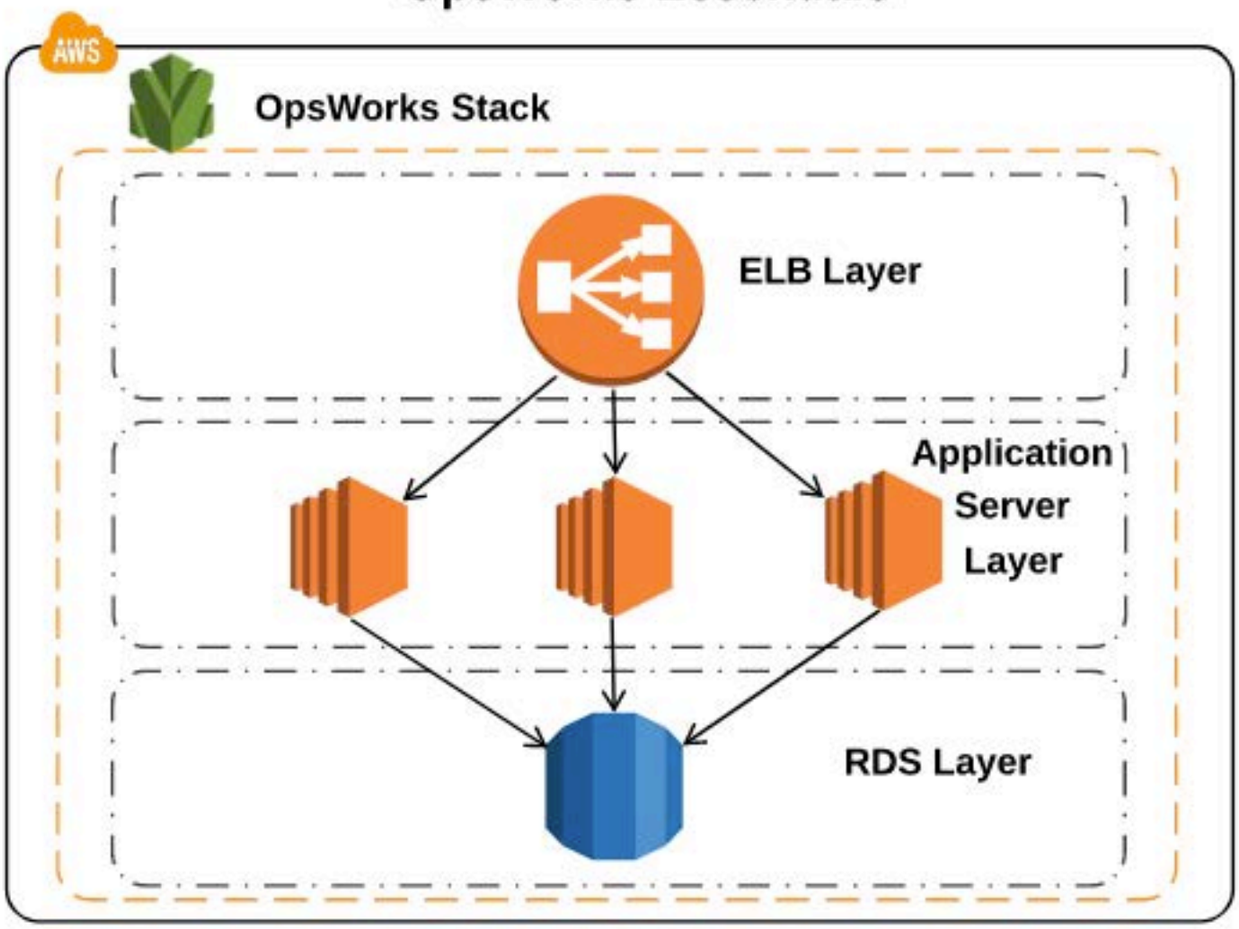
Choose a Topic

Application Deployments On OpsWorks

Scenario: AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet. Chef and Puppet are automation platforms that allow you to use code to automate the configurations of your servers. OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed, and managed across your Amazon EC2 instances or on-premises compute environments.

OpsWorks can be seen as a Platform as a Service, however, unlike most PaaS, you have full control over the OS, instance count, and can make changes directly to the deployment mechanisms.

OpsWorks Essentials



Choose a Topic

X

AWS

OpsWorks Essentials - continued

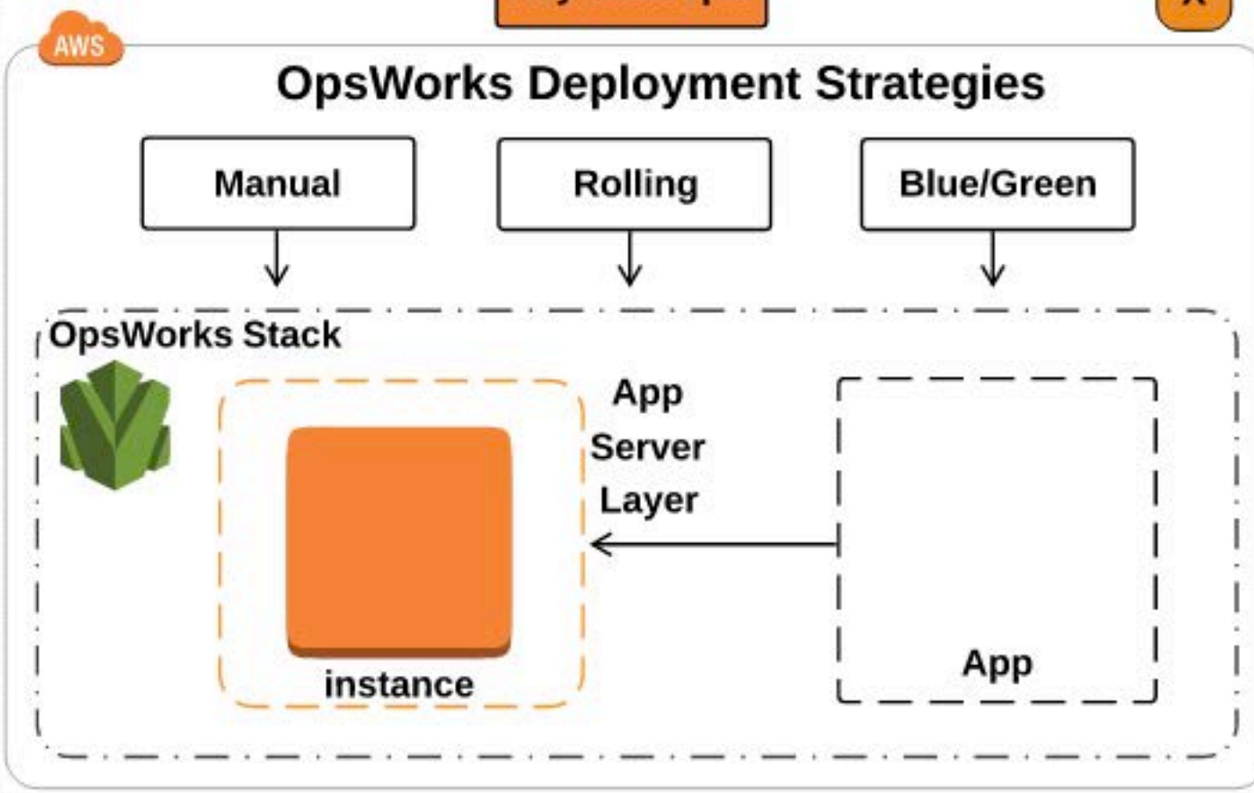
- What is OpsWorks?
  - Provides a flexible way to create and manage resources for our applications, as well as the applications themselves.
  - Create a stack of resources and manage those resources collectively in different layers. These layers can have built-in or custom Chef recipes.
  - Use OpsWorks to:
    - Automate Deployments
    - Monitor Deployments
    - Maintain Deployments
  - OpsWorks provides abstraction from the underlying infrastructure while still giving plenty of control. OpsWorks can provide more customization than Elastic Beanstalk. OpsWorks is better for longer application lifecycles (EB for shorter app lifecycle).
  - Uses Chef, which is an open source tool that automates infrastructure by turning it in to code.
- OpsWorks Anatomy - Stacks, Layers, Instances, Applications
  - Stacks - A set of resources we want to manage as a group. Use Case: can build a stack for dev, staging, production.
  - Layers - Used to represent and configure components of a stack. Example: A layer for ELB, a layer for web app, and a layer for DB.
  - Instances - Must be associated with at least one layer. Can run instances as 24/7, load-based or time-based.
  - Apps - Deployed to the application layer through a source code repository like GIT, SVN, or even S3. We can deploy an app to a layer and have OpsWorks execute recipes to prepare instances for the application.
- OpsWorks Recipes - Created using the Ruby language and based of the Chef deployment software.
  - Custom recipes can customize different layers in an application.
  - Recipes are run at certain pre-defined events within a stack: Setup, Configure, Deploy, Undeploy, Shutdown.

The DevOps Doctrine

Choose a Topic

Key Concepts

X



Deployment Strategies

- Application version consistency - Updating the repo will automatically update new instances but not running instances. To avoid this inconsistency:
  - Avoid updating the master branch of the repo. Instead use tagging to explicitly define the approved source version.
  - Use S3 by packaging code in an archive and use as the source for apps and cookbooks. Versioning can be enabled for versioning and enable rollback.
- **Manual Deploy** - Can run the 'Deploy' command for apps and 'Update Custom Cookbooks' for cookbooks.
  - Pro: Fastest way to deploy
  - Con: An issue with the new app version can cause downtime.
  - We can rollback up to 4 previous versions with OpsWorks. We can also use the undeploy command to remove all versions of the app from instances.
- **Rolling Deployments** - Deploy updates in multiple phases.
  - Pros - can prevent downtime and does not require doubling resources.
  - Cons - Failed deployments reduce capacity and require re-deployment to affected instances.

Choose a Topic

X

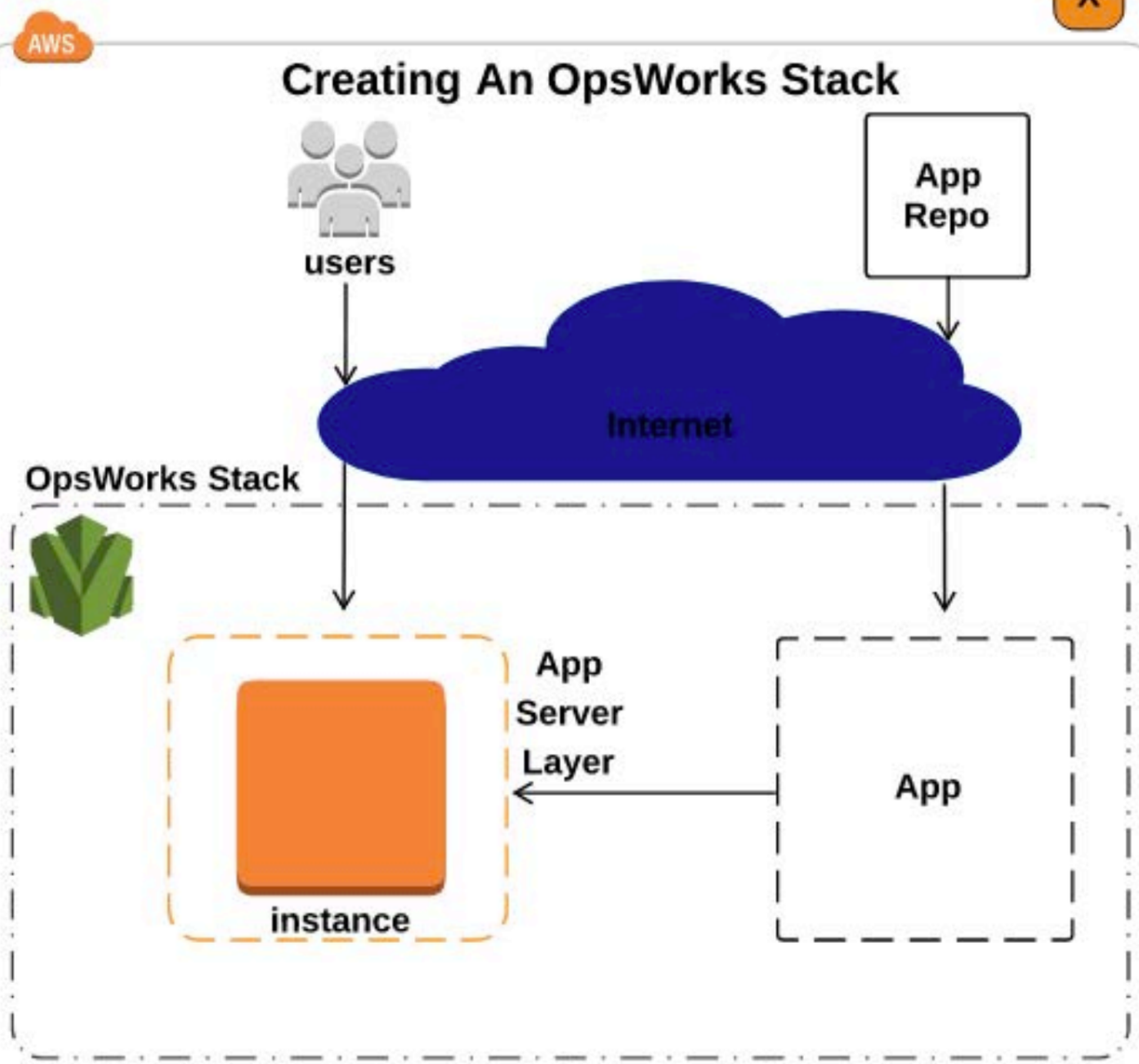
AWS

OpsWorks Essentials - continued

- What is OpsWorks?
  - Provides a flexible way to create and manage resources for our applications, as well as the applications themselves.
  - Create a stack of resources and manage those resources collectively in different layers. These layers can have built-in or custom Chef recipes.
  - Use OpsWorks to:
    - Automate Deployments
    - Monitor Deployments
    - Maintain Deployments
  - OpsWorks provides abstraction from the underlying infrastructure while still giving plenty of control. OpsWorks can provide more customization than Elastic Beanstalk. OpsWorks is better for longer application lifecycles (EB for shorter app lifecycle).
  - Uses Chef, which is an open source tool that automates infrastructure by turning it in to code.
- OpsWorks Anatomy - Stacks, Layers, Instances, Applications
  - Stacks - A set of resources we want to manage as a group. Use Case: can build a stack for dev, staging, production.
  - Layers - Used to represent and configure components of a stack. Example: A layer for ELB, a layer for web app, and a layer for DB.
  - Instances - Must be associated with at least one layer. Can run instances as 24/7, load-based or time-based.
  - Apps - Deployed to the application layer through a source code repository like GIT, SVN, or even S3. We can deploy an app to a layer and have OpsWorks execute recipes to prepare instances for the application.
- OpsWorks Recipes - Created using the Ruby language and based of the Chef deployment software.
  - Custom recipes can customize different layers in an application.
  - Recipes are run at certain pre-defined events within a stack: Setup, Configure, Deploy, Undeploy, Shutdown.

Choose a Topic

X



Creating An OpsWorks Stack

Step 1 - Create Stack
Step 2 - Add Layers
Step 3 - Add Instances
Step 4 - Add Apps
Step 5 - Deploy Apps