

Elastic Beanstalk

Deploy and Scale Web Application

Content Prepared By: Chandra Lingam, Cotton Cola Designs LLC

For Distribution With AWS Certification Course Only

Copyright © 2017 Cotton Cola Designs LLC. All Rights Reserved.

All other registered trademarks and/or copyright material are of their respective owners



Elastic Beanstalk

Automatic Provisioning of Infrastructure and Software

Preconfigured for load balancing, auto-scaling, monitoring

Full control of underlying resources

No additional charge for Elastic Beanstalk

Language Support

- Java
- PHP
- .NET
- Node.js
- Python
- Ruby

and different Container Types for each Language

Comparison

- Elastic Beanstalk - Easy Solution for web apps and web services
- CloudFormation
 - Building-block Service that allows you to build and manage any AWS resource
 - Require you to author a template in JSON/YAML
 - Application deployment can be cumbersome
- OpsWorks
 - Powerful end-to-end solution. Scripting in Ruby
 - Complete application lifecycle from resource provisioning, configuration management, deployment, updates, monitoring, access control

Elastic Beanstalk Concepts

Concept	Description
Application	Logical Collection of Elastic Beanstalk components
Application Version	Labeled version of a deployable code
Environment	Resources provisioned to run a single application version
Environment Tier	Two types of environments: <ul style="list-style-type: none">• Web server environment to handle http requests• Worker Environment to process SQS messages
Environment Configuration	Collection of parameters and settings to manage the resources
Configuration Template	Starting point for creating a new environment configuration

Elastic Beanstalk Workflow

Elastic Beanstalk Workflow

- Create Application
- Upload Code - Application Version
- Launch Environment
- Manage Environment

Permissions

- Elastic Beanstalk [Service Role](#)
 - Used for AWS resource management on your behalf
 - Monitoring resources
- Elastic Beanstalk [Instance Profile](#) – EC2 Instance IAM Role
 - Used by instance to log to S3
 - Upload Debug data to AWS X-Ray
 - ...

Demo – Node.js and DynamoDB

- Web and database as part of single EB Bundle
- Create Application: "demoEB"
- Upload Code. Call it Application Version: "1.0"
- Create environment
- Test
- Delete Environment - lost the database table !!

Demo – Node.js and DynamoDB

- [Manage DB Table outside of application EB bundle](#)
- Launch a new environment with version 1.2 of bundle
- Test
- Delete environment

Source Bundle

- Single Zip file
- Single WAR file
- Max size 512 MB
- Cannot contain a parent directory in the source bundle.
Subdirectories are supported

Deployment Options

Deployment Option	Description
All at once	All instances are updated at the same time
Rolling	Updates are performed in batches. Old version and new version running in the environment until all instances are updated
Rolling with additional batch	Maintains full capacity by launching additional instances. When deployment completes, additional instances are terminated
Immutable	Full set of new instances for new version. Old instances are terminated after successful deployment

Blue/Green Deployment

Eliminate downtime using Blue/Green Deployment

Blue - Production running old version

Green - New environment running new version

When Green deployment is successful, simply swap the CNAMEs of two environments using “Swap Environment URLs” option. Green now becomes the new Blue Production environment

Platform Updates

- AWS releases periodic updates to Elastic Beanstalk Platform
 - Software Component (AMIs, Tools, Elastic Beanstalk Scripts)
 - Configuration Component (Default settings applicable)
- Manual Update
- Managed Updates - automatically upgrades to latest version during scheduled maintenance window.
 - Only patches and minor version updates are supported
 - Major version changes are not automatically applied