Elastic Beanstalk

Developer problems on AWS

- Managing infrastructure
- Deploying code
- Configuring all the databases, load balancers, etc
- Scaling concerns
- Most web apps have the same architecture (ALB + ASG)
- All the developers want is for their code to run!
- Possibly, consistently across different applications and environments

Elastic Beanstalk – Overview

- Elastic Beanstalk is a developer-centric view of deploying an application on AWS
- It uses all the components we've seen before: EC2, ASG, ELB, RDS, ...
- Managed service
  - Automatically handles capacity provisioning, load balancing, scaling, application health monitoring, instance configuration, ...
  - Just the application code is the responsibility of the developer
- We still have full control over the configuration
- Beanstalk is free but you pay for the underlying instances

Elastic Beanstalk – Components

- **Application**: collection of Elastic Beanstalk components (environments, versions, configurations, ...)
- **Application Version**: an iteration of your application code
- **Environment**
  - Collection of AWS resources running an application version (only one application version at a time)
  - **Tiers**: Web Server Environment Tier & Worker Environment Tier
  - You can create multiple environments (dev, test, prod, ...)

Deployment Lifecycle

1. Create Application
2. Upload Version
3. Launch Environment
4. Manage Environment

You can update the version or deploy a new version at any time.

Elastic Beanstalk – Supported Platforms

- Go
- Java SE
- Java with Tomcat
- .NET Core on Linux

- .NET on Windows Server
- Node.js
- PHP
- Python
- Ruby
- Packer Builder
- Single Container Docker
- Multi-container Docker
- Preconfigured Docker

Web Server Tier vs. Worker Tier

Web Environment

- URL: `myapp.us-east-1.elasticbeanstalk.com`
- Uses an Elastic Load Balancer (ELB)
- Auto Scaling Group across multiple Availability Zones
- EC2 instances act as Web Servers
- Protected by Security Groups

Worker Environment

- Uses an SQS Queue to receive tasks
- EC2 instances act as Workers
- Pulls messages from the SQS queue
- Also spans multiple Availability Zones
- Auto Scaling based on SQS queue length

Notes

- Scale is based on the number of SQS messages
- Web Server Tier can push messages to the SQS queue used by Worker Tier

Elastic Beanstalk Deployment Modes

Single Instance (Great for dev)

- One EC2 instance in a single Availability Zone
- Associated with an Elastic IP
- Connects to a single RDS Master instance

High Availability with Load Balancer (Great for prod)

- Multiple EC2 instances across Availability Zones
- Uses an Application Load Balancer (ALB)
- Auto Scaling Group spans multiple zones
- Connects to RDS with:
    - RDS Master in one AZ
    - RDS Standby in another AZ (for failover)