Amazon Elastic Beanstalk

Problems on AWS for a Developer

- Managing/Creation Infrastructure
- Code Deployment
- Configuring databases
- Configuring load balancers
- Auto Scaling
- Monitoring
- Logging

Amazon Elastic Beanstalk

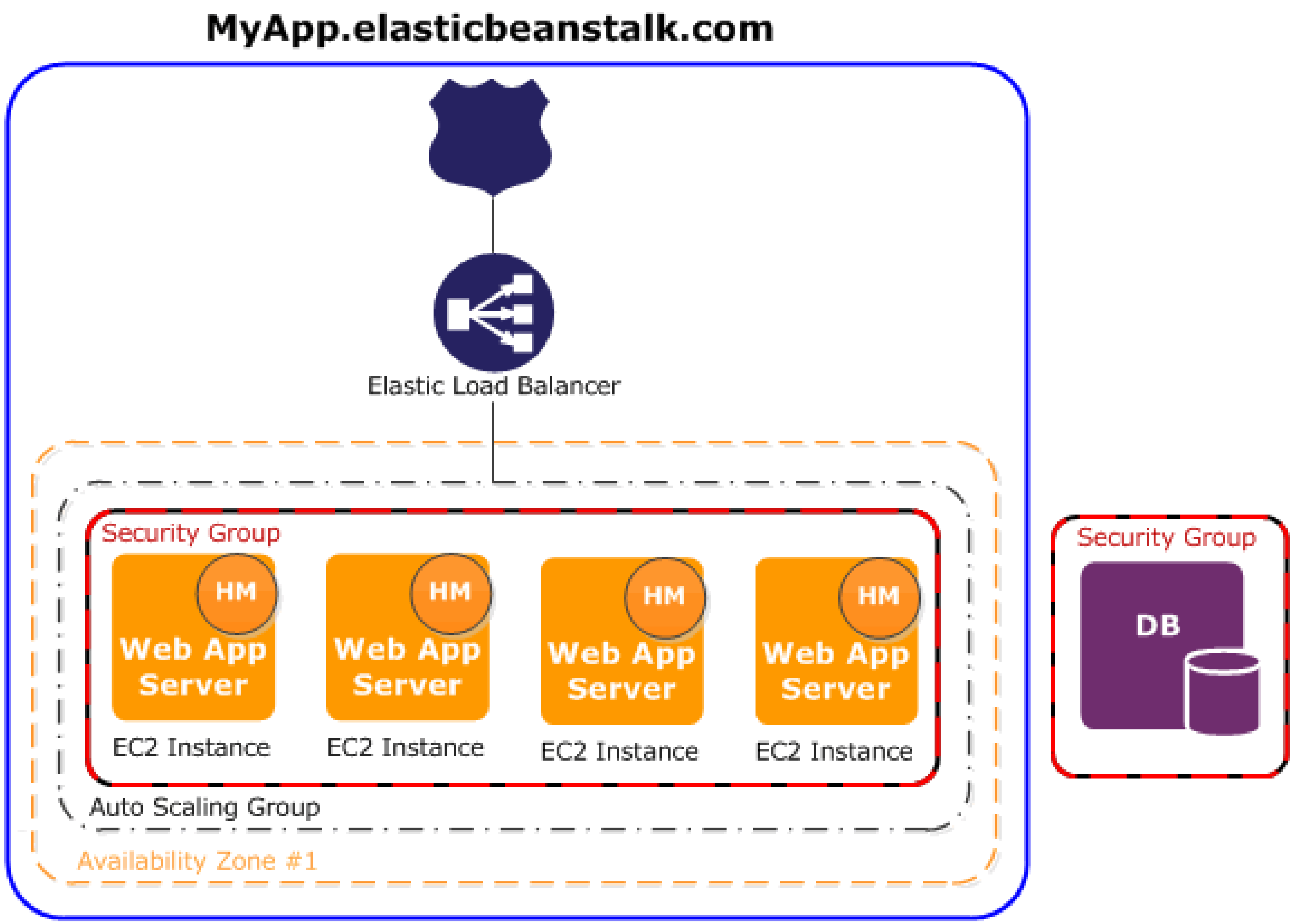
- AWS Elastic Beanstalk is the fastest and simplest way to get an application up and running on AWS.
- Upload your application code, and the service automatically handles all of the details, such as
    - Resource provisioning
    - Load balancing
    - Auto Scaling
    - Monitoring.
- You can quickly deploy and manage applications on the AWS cloud
- Reduces management complexity

AWS Elastic Beanstalk Overview

- Fully Managed Service
  - Instance configuration / OS is handled by Beanstalk
  - Deployment strategy is configurable but performed by Elastic Beanstalk
- Just the application code is the responsibility of the developer
- Three architecture models:
  - Single Instance deployment: good for dev
  - LB + ASG: great for production or pre-production web applications
  - ASG only: great for non-web apps in production (workers, etc..)

AWS Elastic Beanstalk

- Elastic Beanstalk has three components
    - Application
    - Application version: each deployment gets assigned a version
    - Environment name (dev, test, prod…): free naming
- You deploy application versions to environments and can promote application versions to the next environment
- Rollback feature to previous application version
- Full control over lifecycle of environments


MyApp.elasticbeanstalk.com
Elastic Load Balancer
Security Group
HM
Web App Server
EC2 Instance
HM
Web App Server
EC2 Instance
HM
Web App Server
EC2 Instance
HM
Web App Server
EC2 Instance
Auto Scaling Group
Availability Zone #1
Security Group
DB

Elastic Beanstalk

- Support for many platforms:
  - Go
  - .NET
  - Node.js
  - PHP
  - Python
  - Ruby
  - Packer Builder
  - Docker

Elastic Beanstalk Deployment

- Elastic Beanstalk supports environments as
    - Single Instance environments
    - High Availability with Load Balancer

Beanstalk Deployment Options

- **All at once**
  - Deployment fastest, but having little downtime
- **Rolling**
  - Deploy the new version in batches.
  - Each batch is taken out of service during the deployment phase, reducing your environment's capacity by the number of instances in a batch.
- **Rolling with additional batch**:
  - Deploy the new version in batches
  - first launch a new batch of instances to ensure full capacity during the deployment process
- **Immutable**:
  - Deploy the new version to a fresh group of instances
  - swaps all the instances when everything is healthy

Deployment Options - All at once

- Fastest deployment
- Application has a little downtime
- Great for quick iterations in development environment
- No additional cost
- rollback would take time in case of any issues

Deployment Options - Rolling

- Application is running below capacity
- Can set the bucket size
- Application is running both versions simultaneously
- No additional cost
- Long deployment

Deployment Options - Rolling with additional batch

- Application is running at capacity
- Can set the bucket size
- Application is running both versions simultaneously
- Small additional cost
- Additional batch is removed at the end of the deployment
- Longer deployment
- Good for prod env

Deployment Options - Immutable

- Zero downtime
- New Code is deployed to new instances on a temporary ASG
- High cost, double capacity
- Longest deployment
- Quick rollback in case of failures (just terminate new ASG)
- Good for prod env

Method	Impact of failed deployment 🕒	Deploy time	Zero downtime	No DNS change	Rollback process	Code deployed to
All at once	Downtime		No	Yes	Manual redeploy	Existing instances
Rolling	Single batch out of service; any successful batches before failure running new application version	†	Yes	Yes	Manual redeploy	Existing instances
Rolling with an additional batch	Minimal if first batch fails; otherwise, similar to Rolling	†	Yes	Yes	Manual redeploy	New and existing instances
Immutable	Minimal		Yes	Yes	Terminate new instances	New instances
Traffic splitting	Percentage of client traffic routed to new version temporarily impacted	††	Yes	Yes	Reroute traffic and terminate new instances	New instances
Blue/green	Minimal		Yes	No	Swap URL	New instances

Beanstalk Lifecycle Policy

- Elastic Beanstalk can store at most 1000 application versions
- If you don’t remove old versions, you won’t be able to deploy anymore
- To phase out old application versions, use a lifecycle policy
  - Based on time (old versions are removed)
  - Based on space (when you have too many versions)
- Versions that are currently used won’t be deleted
- Option not to delete the source bundle in S3 to prevent data loss

Elastic Beanstalk Extensions

- A zip file containing our code must be deployed to Elastic Beanstalk
- All the parameters set in the UI can be configured with code using files
- Requirements:
  - in the .ebextensions/ directory in the root of source code
  - YAML / JSON format
  - .config extensions (example: logging.config)
  - Ability to add resources such as RDS, ElastiCache, DynamoDB, etc…
- Resources managed by .ebextensions get deleted if the environment goes away