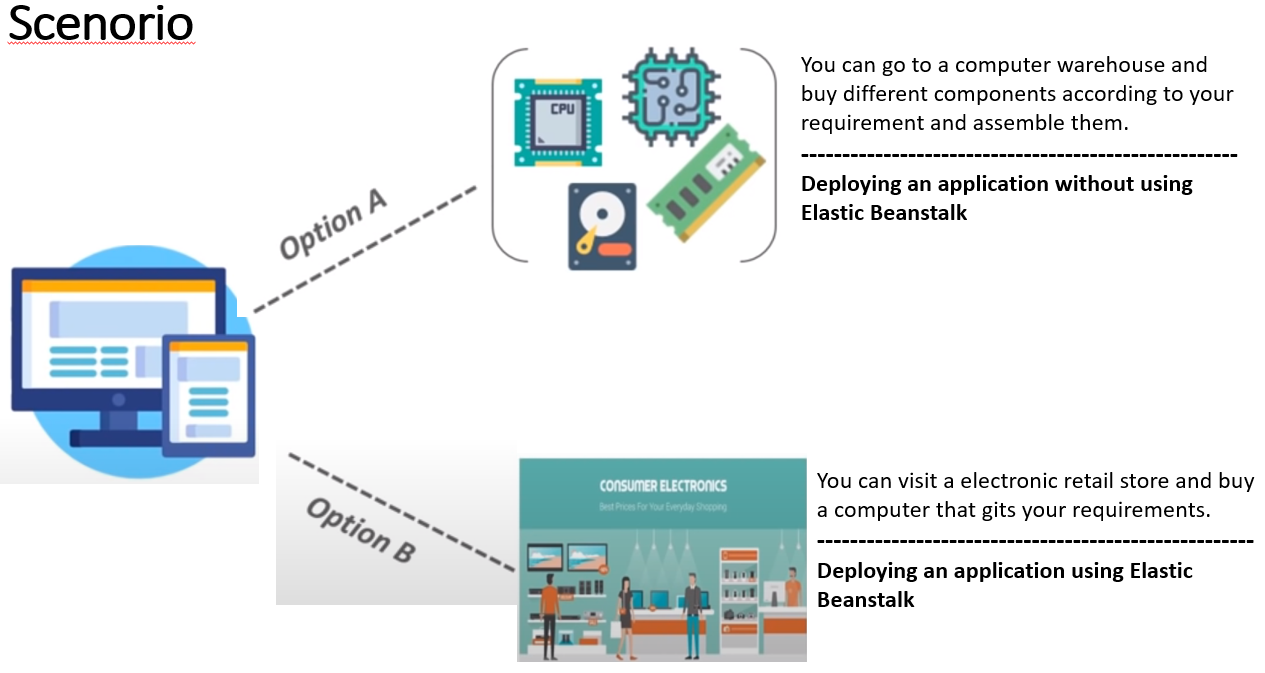
Contents

[Overview 1](#_Toc67635884)

# Overview

* Elastic Beanstalk is a PAAS for deploying and scaling web applications developed in many popular languages: Java, .NET, PHP, Node.js, Python, Ruby, Go and Docker onto widely used application server platforms like Apache Tomcat, Nginx, Passenger, and IIS.
* Developers can focus on writing code and don’t need to worry about any of the underlying infrastructure needed to run the application.
* It is basically a provisioning service
* You upload the code and Elastic Beanstalk will handle deployment, capacity provisioning, Load Balancing, auto-scaling and application health.
* You retain full control of the underlying AWS resources powering your application
* There is no additional charge for Elastic Beanstalk -you pay only for the AWS resources required to store and run your applications (e.g EC2 instances and S3 buckets)



* Elastic Beanstalk supports several options for processing deployments
  + All at once
  + Rolling
  + Rolling with additional Batch
  + Immutable
* All at Once Deployment Updates:
  + Deploys the new version to all instances simultaneously
  + All of your instances are out of service while the deployment takes place
  + You will experience an outage while the deployment is taking place – not ideal for mission-critical production systems
  + If the update fails, you need to rollback the changes by re-deploying the original version to all your instances
* Rolling Deployment Updates:
  + Deploys the new version in batches
  + Each batch of instances is taken out of service while the deployment takes place
  + Your environment capacity will be reduced by the number of instances in a batch while the deployment takes place
  + Not ideal for performance sensitive systems
  + If the update fails, you are going to perform an additional rolling update to roll back the changes
* Rolling with Additional Batch Deployment Updates:
  + Launches an additional batch of instances
  + Deploys the new version in batches
  + Maintains full capacity during the deployment process
  + If the update fails, you need to perform an additional rolling update to roll back the changes
* Immutable Deployment Updates:
  + Deploys the new version to a fresh group of instances in their own new autoscaling group
  + When the new instances pass their health checks, they are moved to your existing auto scaling group and finally the old instances are terminated
  + Maintains full capacity during the deployment process
  + The impact of a failed update is far less, and the rollback process requires only terminating the new auto scaling group
  + Preferred option for Mission critical production systems

