



AWS Elastic Beanstalk

Challenges faced by developers

- ⌚ Multiple components need to be created and configured while deploying a three-tier architecture
- 🌐 You might need to deploy multiple applications
- ⚙️ Each application may consist of several environments like test, dev and prod
- ⌚ Maintain multiple application versions of each application at the same time
- 💻 Complexities involved while managing infrastructure
- 📈 Scaling concerns
- ✳️ Achieve consistency across different applications and environments
- 💻 Developers want their code to run

Elastic Beanstalk

 Elastic Beanstalk is a developer centric view of deploying an application on AWS

 You can quickly deploy and manage applications

 No need to learn about the infrastructure that runs those applications

 Reduced management complexity without restricting choice or control

 You simply upload your application

 Capacity provisioning, load balancing, scaling and application health monitoring



Elastic Beanstalk **Basic Components**

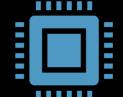
Application, Application Version and Environment

Application

A logical collection of Elastic Beanstalk components, including:

- Environments
- Versions
- Environment Configurations

Application Version

-  An application source bundle or a deployable code (e.g. a Java .war file)
-  An application version is part of an application
-  Applications can have many versions and each application version is unique

Environment

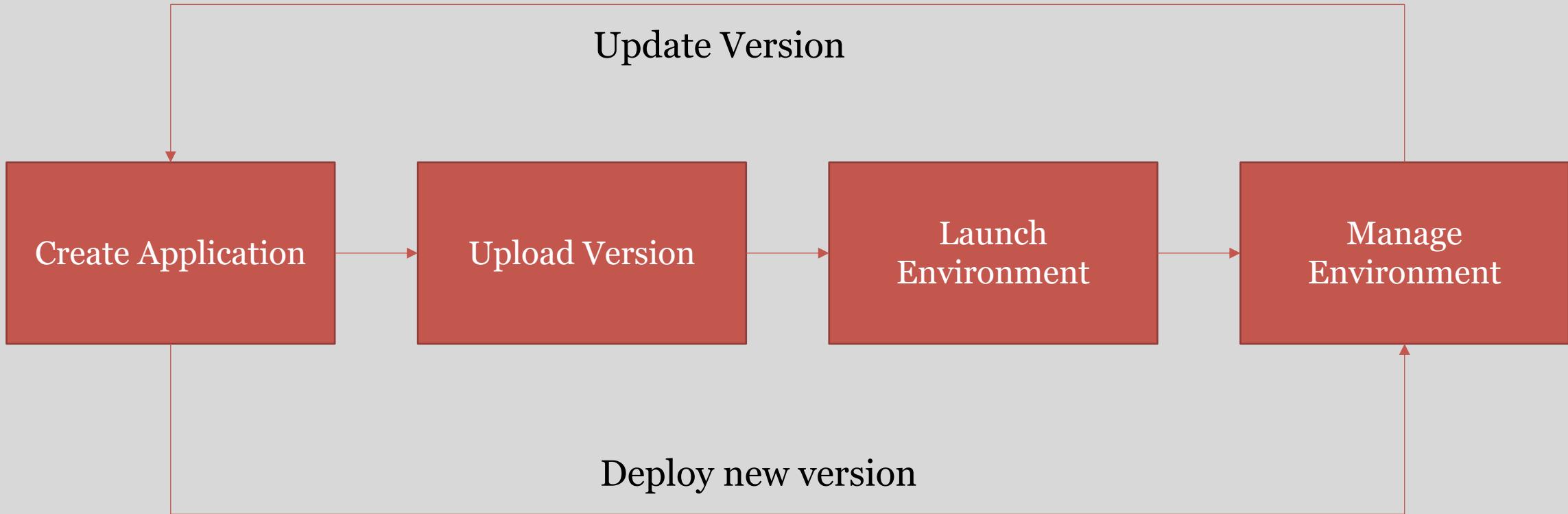
A collection of AWS resources running an application version

Each Environment runs only one application version at a time

You can run the same application version or different application versions in many environments simultaneously

When you create an Environment, Elastic Beanstalk provisions the resources needed to run the application version you specified

How It Works?



Pricing



There is no additional charge for Elastic Beanstalk

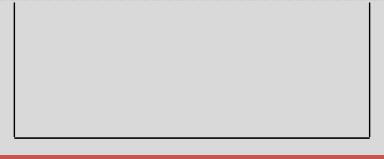


You pay only for the underlying AWS resources that your application consumes



LAUNCHING AN ADVANCED OR PROD ENVIRONMENT

With ‘High Availability’ configuration preset



Deployment Options or Policies

All at once, Rolling, Rolling with additional batches, Immutable & Blue/Green Deployment

Beanstalk Deployment Options for Updates



All at once (deploy all in one go): fastest, but instances aren't available to serve traffic for a bit (downtime)



Rolling: update a few instances at a time (bucket), and then move onto the next bucket once the first bucket is healthy

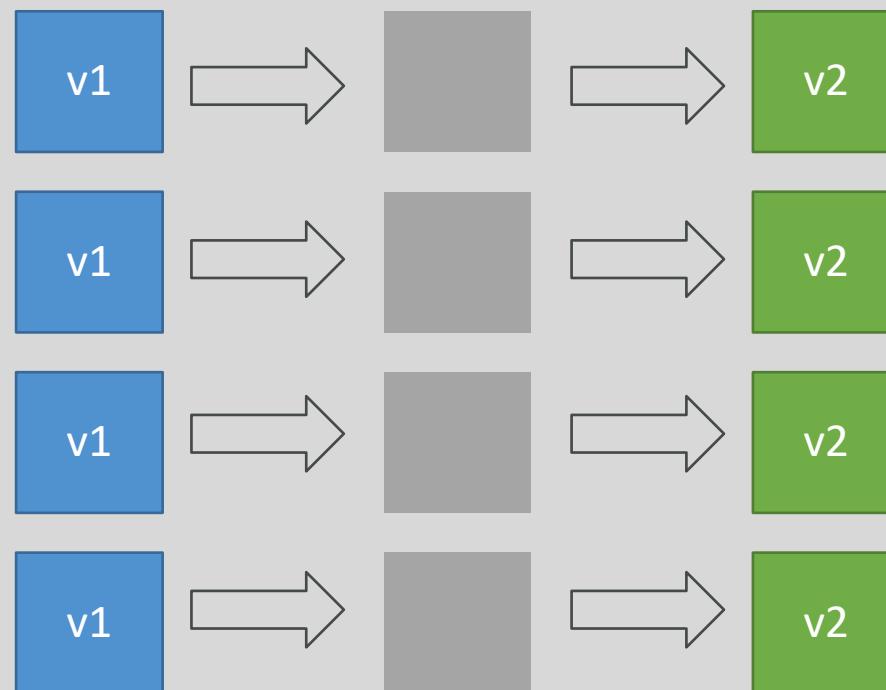


Rolling with additional batches: like rolling, but spins up new instances to move the batch (so that the old application is still available)



Immutable: spins up new instances in a new ASG, deploys version to these instances, and then swaps all the instances when everything is healthy

All at once



All at once



Fastest deployment



Application has downtime

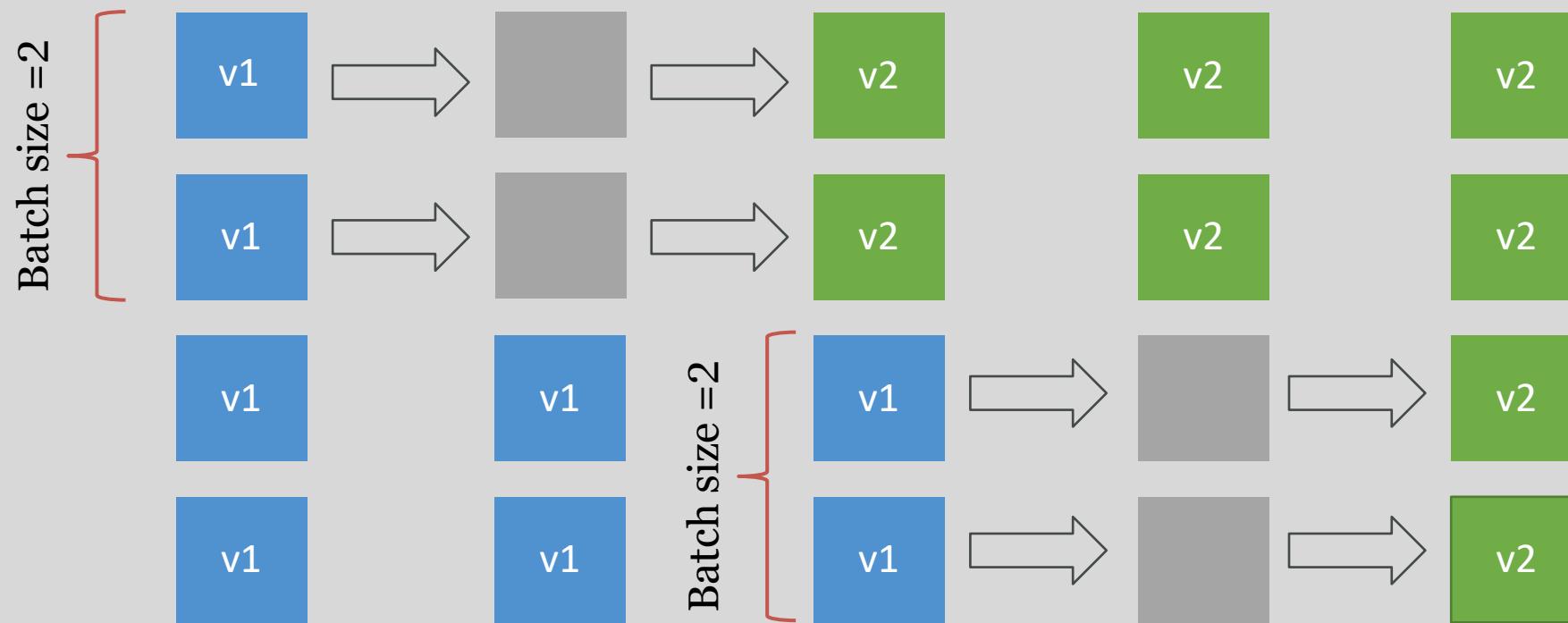


Great for quick iterations in development environment



No additional cost

Rolling



Rolling

Application is running below capacity

Can set the bucket size

Application is running both versions simultaneously

No additional cost

Long deployment

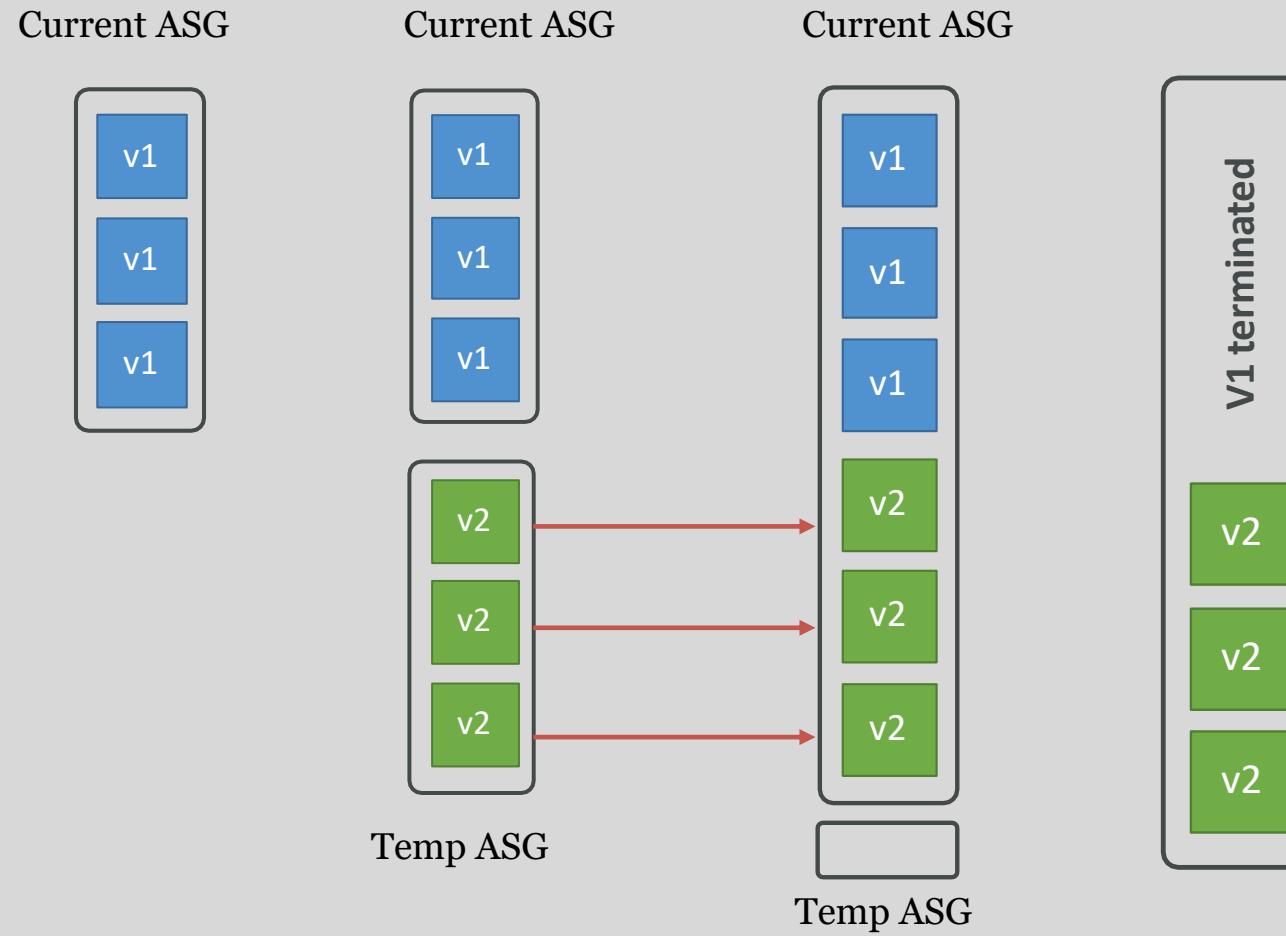
Rolling with additional batches



Rolling with additional batches

- Application is running at capacity
- Can set the batch 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

Immutable



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)

 Great for prod