AWS ElasticBeanStalk

What is Elastic Beanstalk

- With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without worrying about the infrastructure that run those applications.

AWS Elastic Beanstalk supports the following languages and development stacks.

- Apache tomcat for java applications
- Apache HTTP server for PHP applications
- Apache HTTP Server for Python applications
- Nginx or Apache HTTP server for Node.js applications
- Passenger or puma for Ruby applications
- Microsoft IIS 7.5, 8.0 and 8.5 for .NET applications
- Java SE
- Docker
- GO

Get Started in Three Easy Steps

Select a Platform

Upload an Application or Use a Sample

Run it!

Start Now by Selecting Your Platform

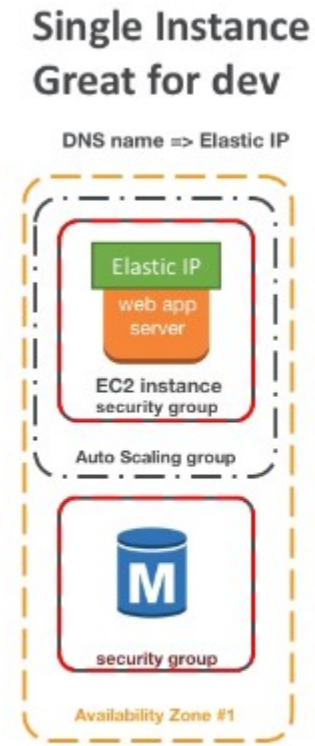
php Ruby Windows Server 2012 node …and more

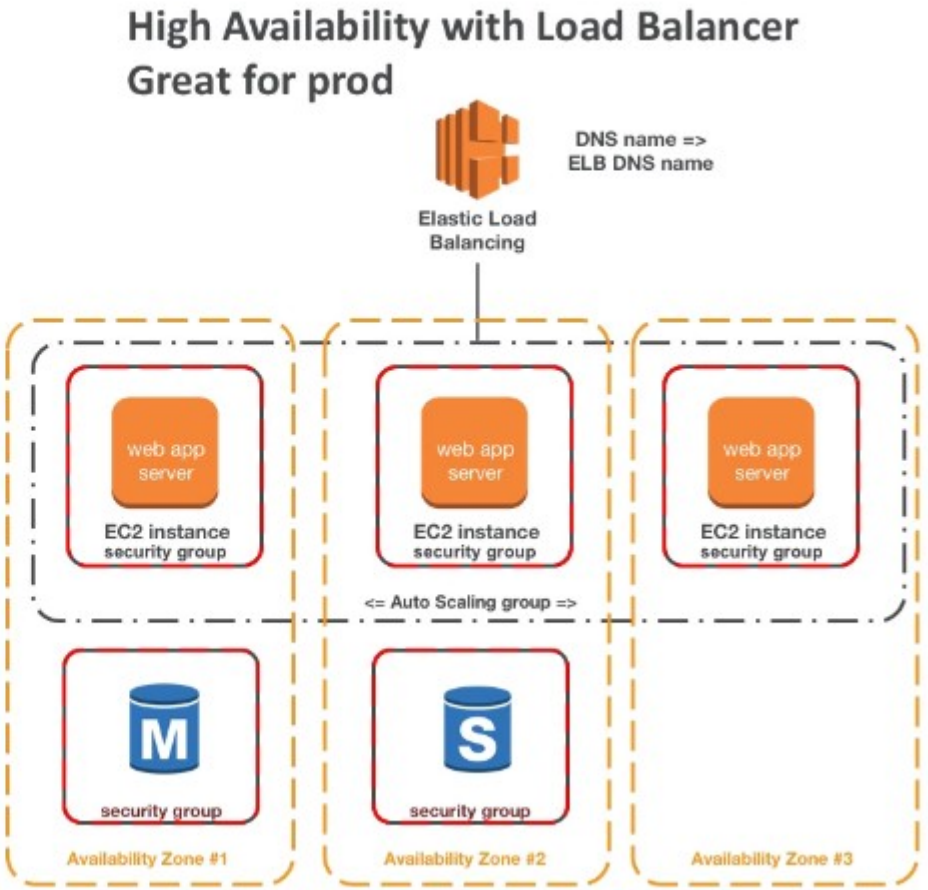
Key Words

Application Deployment requires a number of components to be defined as follows

- **Application**: as a logical container for the project
- **Version**: which is a deployable build of the application executable.
- **Configuration template**: This contains configuration information for both the Beanstalk environment and for the product.
- **Environment**: Combines a version with a configuration and deploys them

BeanStalk Deployment Modes



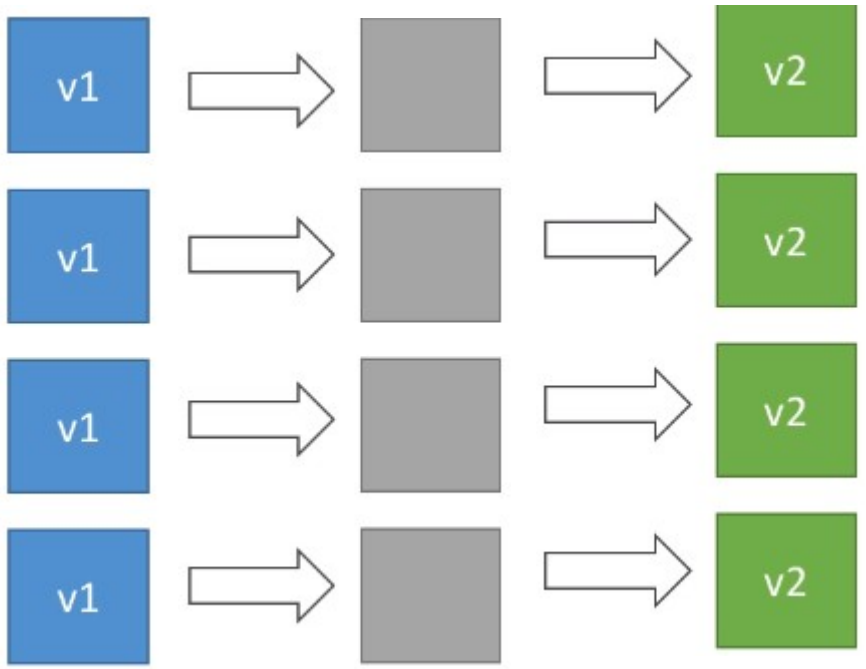


Beanstalk Deployment options for update

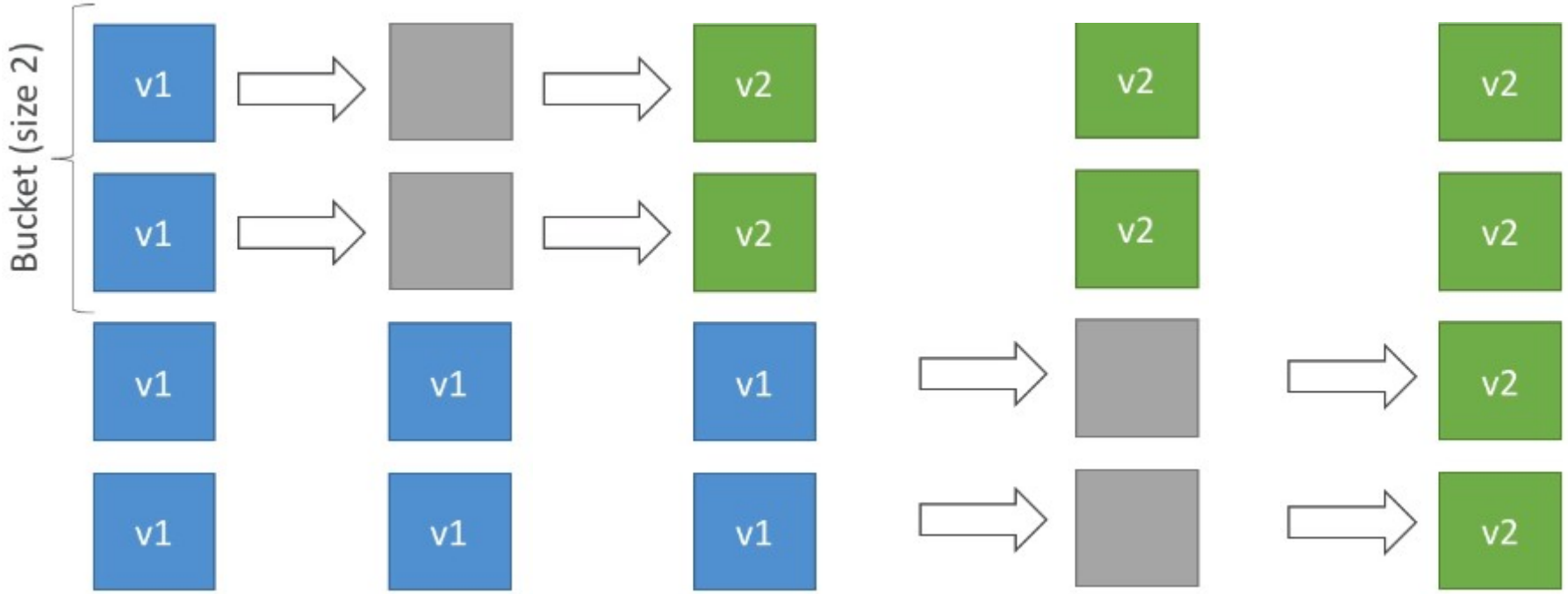
- **All at Once (Deploy all in one go)** – Fastest, but instances aren't available to serve traffic for a bit (downtime)
- **Rolling**: Update a few instances at a time(bucket), and then move onto the next bucket once the first bucket is healthy
- **Rolling with additional batches**: Like Rolling, but 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 versions to these instances, and then swaps all the instance when everything is healthy

Beanstalk Deployment – All at once

- Fastest deployment
- Application has downtime
- Great for quick iterations in development environment
- No additional cost

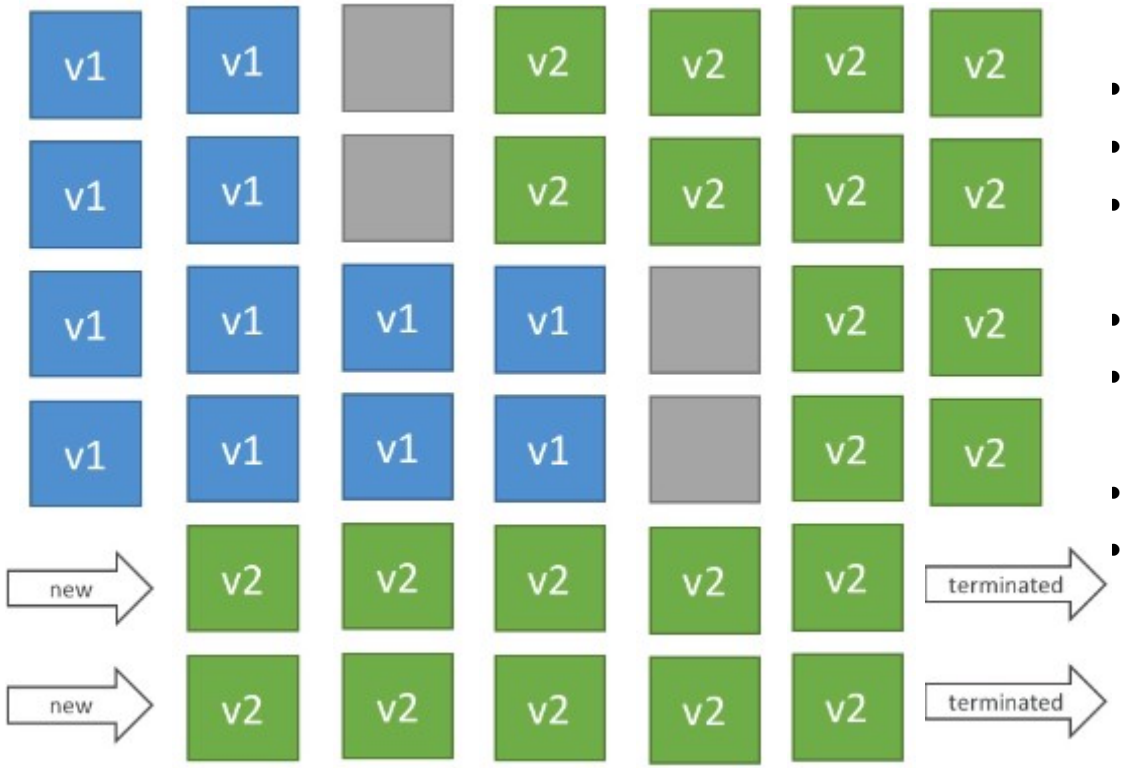


Beanstalk Deployment – Rolling



- Application is running below capacity
- Can set the bucket size
- Application at some point of time running both versions simultaneously
- No Additional cost
- Long Deployment

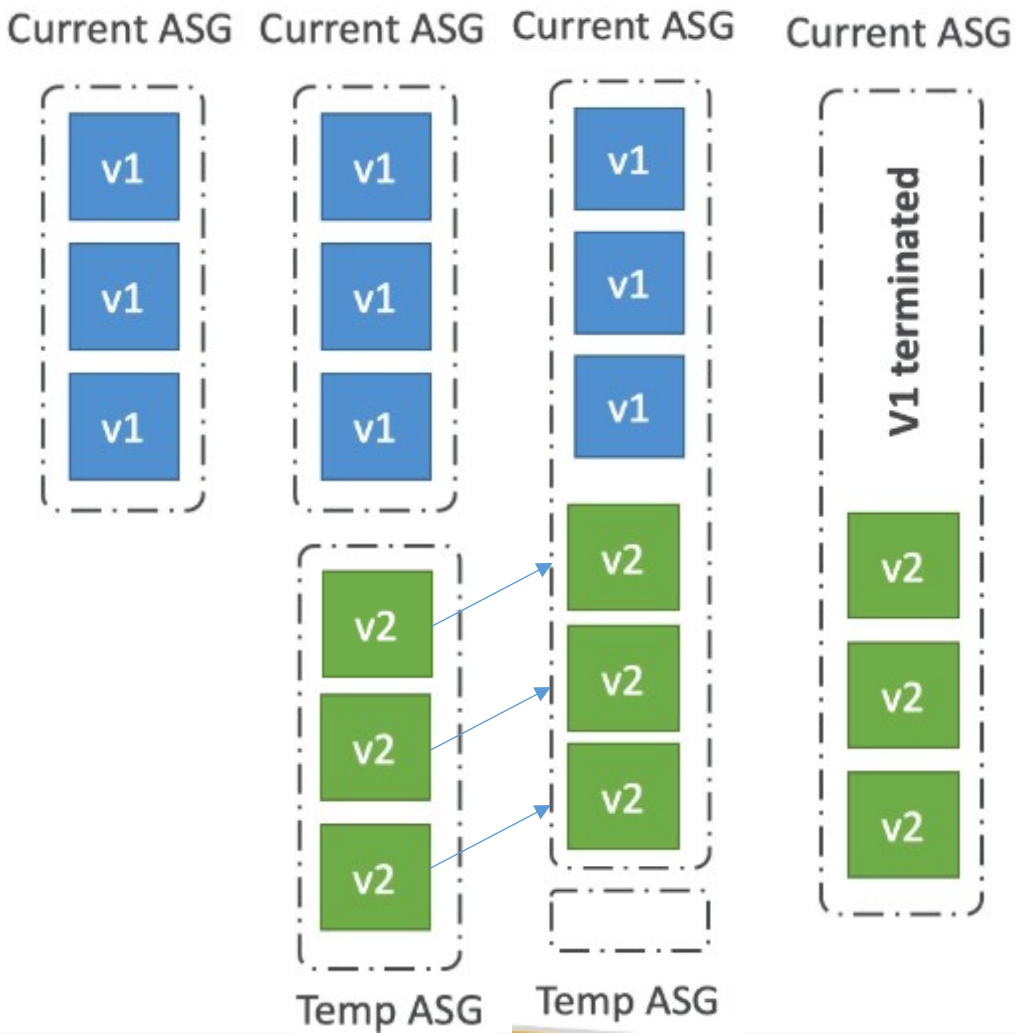
Beanstalk Deployment – Rolling with additional batches



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

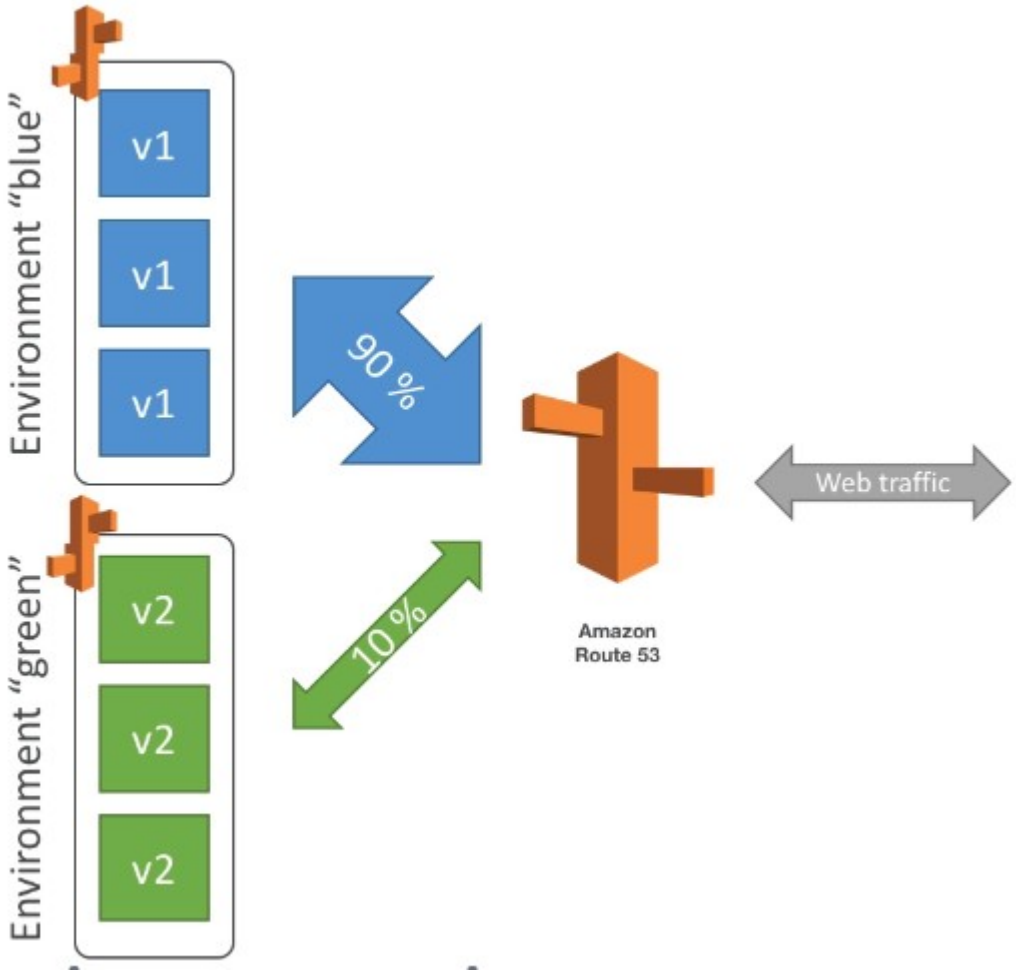
Beanstalk Deployment – Immutable

- Zero downtime
- New code is deployed to new instances, on a temp ASG
- High cost, double capacity
- Longest deployment
- Quick rollback in case of failure (Just terminate new ASG)
- Great for Production



Beanstalk Deployment – Blue/Green

- Not a direct feature of beanstalk
- Zero downtime and release facility
- Create a new “stage” env and deploy v2 there
- The new environment(green) can be validated independently and roll back if issues.
- Using Beanstalk Swap URLs when down with testing environment.
- Manual process



In one shot

Deployment Methods

Method	Impact of Failed Deployment	Deploy Time	Zero Downtime	No DNS Change	Rollback Process	Code Deployed To
All at once	Downtime	🕒	X	✓	Manual Redeploy	Existing instances
Rolling	Single batch out of service; any successful batches prior to failure running new application version	🕒 🕒 †	✓	✓	Manual Redeploy	Existing instances
Rolling with additional batch	Minimal if first batch fails, otherwise, similar to **Rolling**	🕒 🕒 🕒 †	✓	✓	Manual Redeploy	New and existing instances
Immutable	Minimal	🕒 🕒 🕒 🕒	✓	✓	Terminate New Instances	New instances
Blue/green	Minimal	🕒 🕒 🕒 🕒	✓	X	Swap URL	New instances

Only Once -- Fasak

- **All at once**: Update all instance at once
- **Rolling**: Update the instances one by one.
- **Rolling with additional batch**: Create a new instance and update it batch basis.
- **Immutable**: Create a new temp ASG, once everything ready move all the instances.

How Beanstalk Deploys applications

- EC2 has a base AMI(can configure)
- EC2 gets the new code of the app
- EC2 resolves the app dependencies (Can take a while)
- Apps get swapped on the EC2 instances

Resolving the dependencies can take a long time!

We can use Golden AMI to fix the problem

Golden AMI

- If your application has a lot of application or OS dependencies, and you want to deploy as quickly as possible, you should use GOLDEN AMI

- Golden AMI = Standardize company-specific AMI with
  - Package OS dependencies
  - Package App dependencies
  - Package company-wide software

  A *golden AMI* is an AMI that contains the latest security patches, software, configuration, and software agents that you need to install for logging, security maintenance, and performance monitoring. You can build and deploy golden AMIs in your environment, but the AMIs quickly become dated as new vulnerabilities are discovered.

Golden AMI

- By using a Golden AMI to deploy a beanstalk (in combination of blue/green new ASG deployment), our application won’t need to resolve the dependencies or a long time to configure

Exam Tips

- With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without worrying about the infrastructure that run those applications.
- You simply upload your code/application, and elastic beanstalk automatically handles the details of capacity provisioning , load balancing , scaling and application health monitoring.
- Beanstalk can put application logs into CloudWatch logs.
- You can manage the application, AWS will manage the underlying Infrastructure
- Know the different deployment modes
- You are not responsible for running patches etc