**ELASTIC BEANSSTALK**

Easy and quick deployment of application in AWS

**Note:**

a) In general (other provider) in PAAS client don’t have control on the server

b) But, in AWS (i.e PAAS) Client have full control on the EC2 instances which is launched beanstalk

c)EBS handles EC2 instances (OS) behalf us

d)EBS is PAAS == Application + Data

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

No need to do deployment manually

**Two ways we can deploy application**

* 1.EC2 🡪 Manual configuration
* 2.Elastic Beanstalk 🡪 Everything automatically

**AWS Elastic Beanstalk supports the following languages and development stacks**

* .NET Core on Linux
* .NET on Windows Server
* Docker
* Glassfish
* Go
* Java
* Node.js
* PHP for HTTP server
* Python for HTTP server
* Ruby
* Tomcat for JAVA application

**Key Words**

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

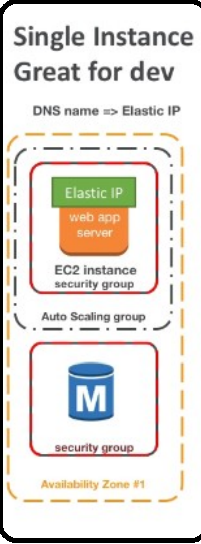
* **Application**: as a logical container for the project
* **Version**: which is deployable build of the application executable
* **Configuration template:** This contains configuration information for both the Beanstalk environment and for the product
* **Environment:** Combines a version with a configuration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

We have Two types of Deployment models

* 1.Single Instance
* 2.HA Deployment

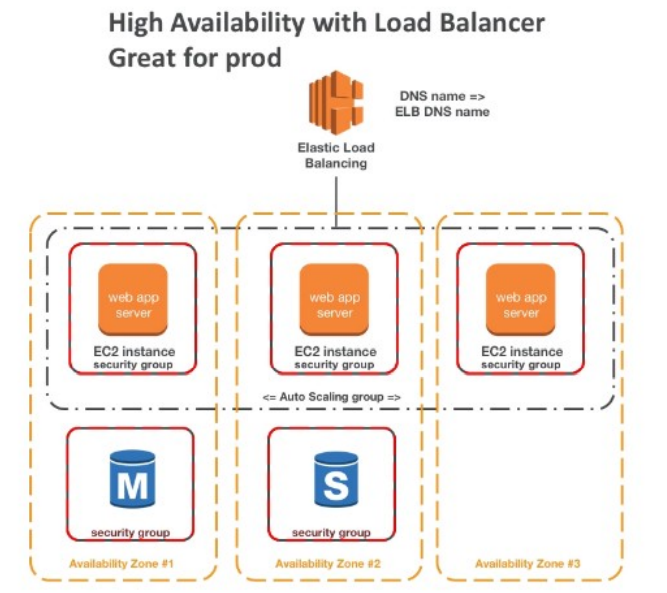
**Single Instance:**

* If you want application run on single instance (i.e One EC2 instance)
* No High Availability
* Single instance also called as spot pre-set
* Recommended for Development and testing



**HA Deployment:**

* If we choose HA Deployment for application, it will automatically create Load Balancer, Automatically Launch EC2 instance like Launch Configuration
* HA Deployment is also called on demand pre-set
* Recommended for production server



**EBS Architecture**

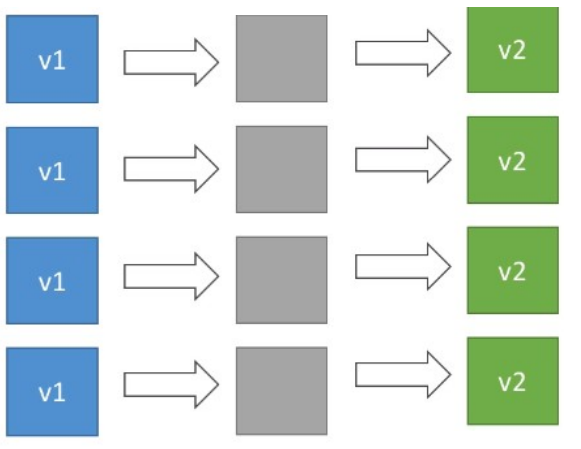
* Create application (not actual application it is folder)
* Inside application need to create environment
* Each environment has its own application and url
* One application will have multiple environment (i.e HA Deployment
* Beanstalk will have its own deployment options

**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

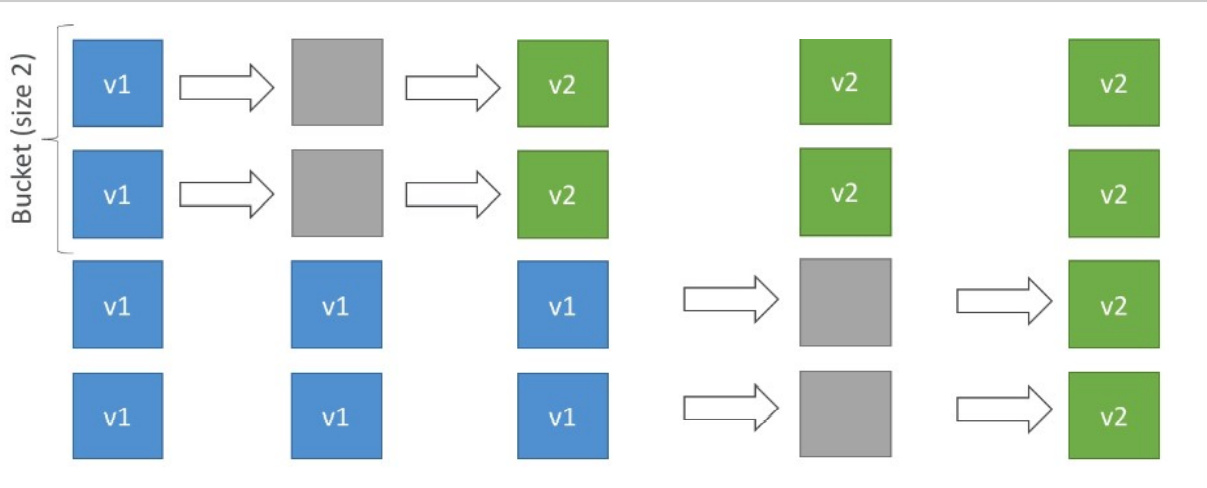
**All at Once (Deploy all in one go)**

* Fastest deployment
* Update all instance at once
* Application has downtime
* Great for quick iterations in development environment
* No additional cost



**Rolling**

* Application is running below capacity
* Update the instances one by one
* Can set the bucket size
* Application at some point of time running both versions simultaneously
* No Additional cost
* Long Deployment



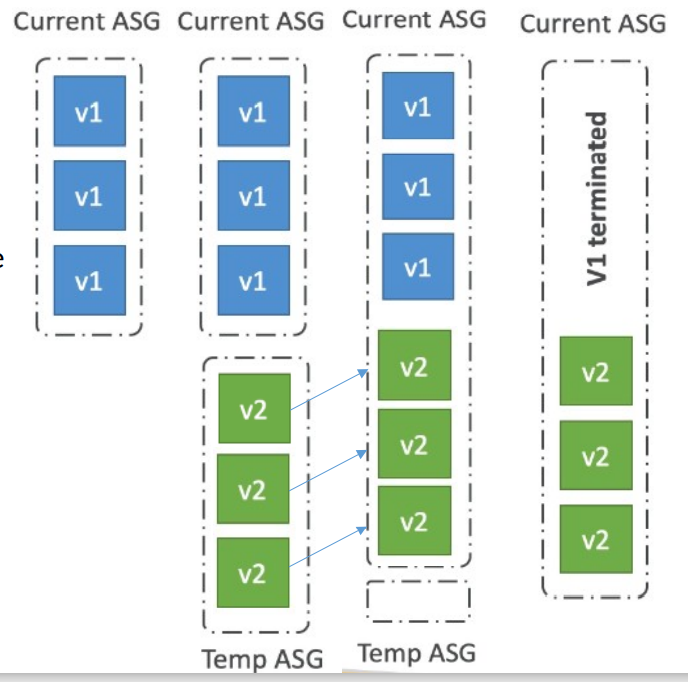
**Rolling with Additional batches**

* Application is running at capacity
* Create a new instance and update it batch basis
* Can set the bucket size
* Application running both versions simultaneously
* Small additional cost
* Additional batch is removed at the end of deployment
* Longer deployment and good for production



**Immutable**

* Zero downtime
* Create a new temp ASG, once everything ready move all the instances.
* 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



**Blue / Green Deployment**

* 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

