

1

Introduction to Elastic Beanstalk

► Introduction to Elastic Beanstalk

What is Elastic Beanstalk ?

Türkçe manası, fasulye sıırığı demektir.

Elastic Beanstalk'a bir application koyuyorsunuz sonra o fasulye sıırığı gibi büyüyor benzetmesi ile bu ismi almıştır.

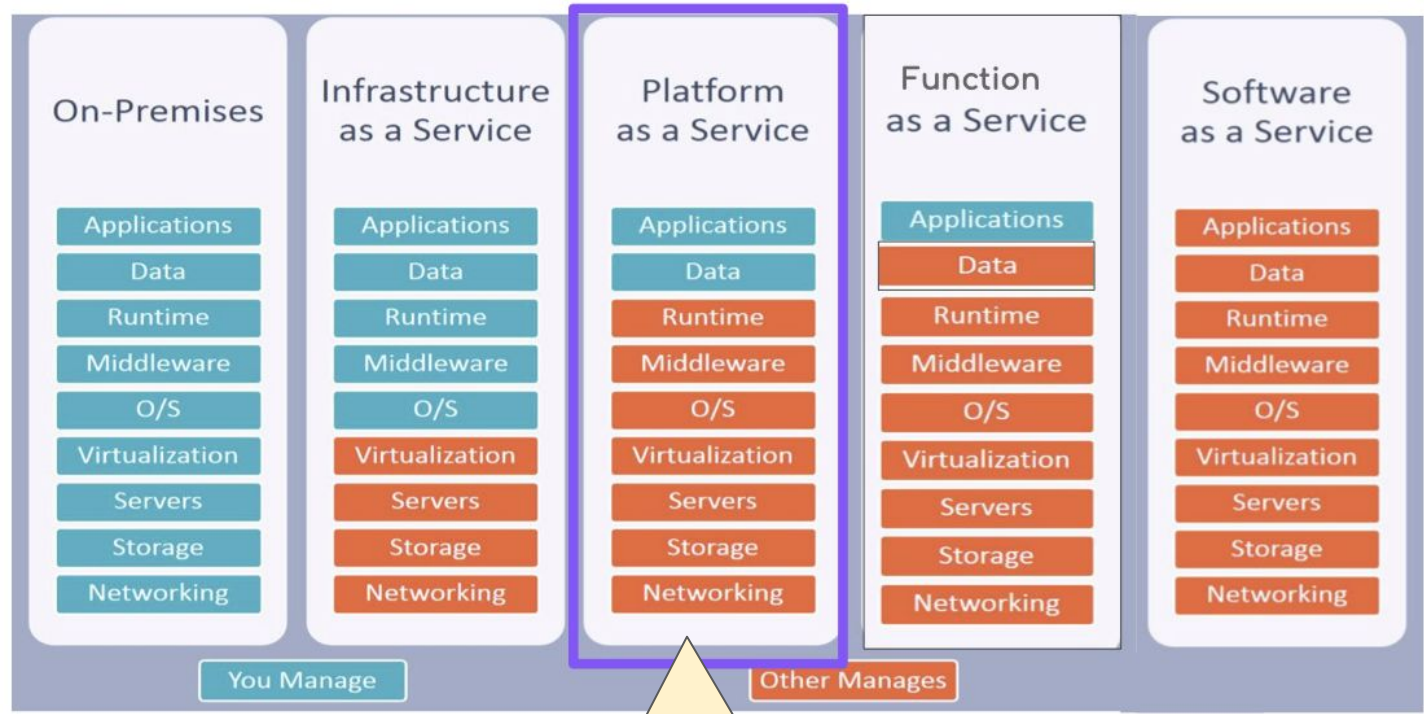
AWS Elastic Beanstalk web uygulamalarını ve hizmetlerini dağıtmak ve ölçeklendirmek için kullanılan bir hizmettir.

AWS tarafından uygulama mimarimizi kurmak için sunulan bir düzenleme hizmetidir.

Bir çok programlama dilleri ve platformları ile uyumludur.

Horoko isimli bir platform var, bunlar tamamen Elastic Beanstalk yapıyor. Piyasada oldukça bilindik. Interview larda duyabilirsiniz.



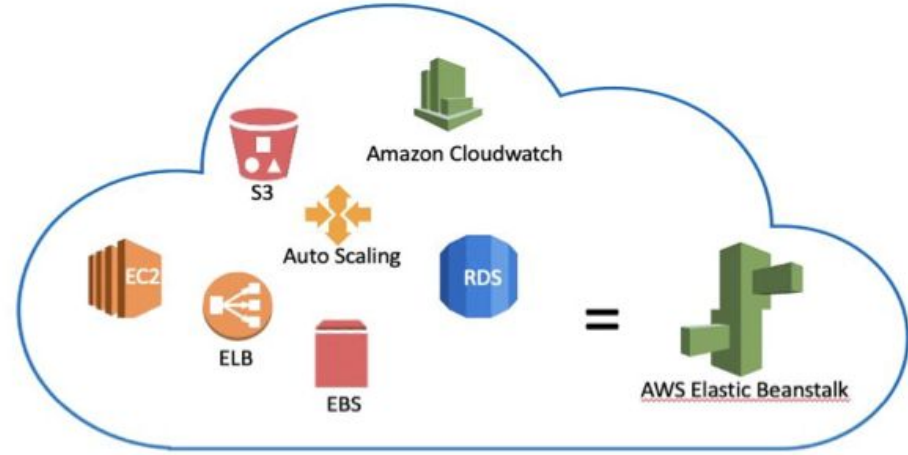


Elastic Beanstalk PAAS dır. Bunların hepsini oluşturup ayağa kaldırıyor ve programı hazır hale getiriyor.

Introduction to Elastic Beanstalk

Why AWS Elastic Beanstalk?

Normalde biz bir VPC içerisinde bir application kaldırırken buradaki EC2, S3, ASG, RDS, vs. tasarlayıp ortaya koymamız gerekiyor. Ancak Elastic Beanstalk, bunların tamamını hazır hale getiriyor ve kapattığımızda da hepsini tekrar kaldırıyor. Bu işlem background da CloudFormation stack üzerinden dönüyor.



- Kapasite sağlamanın ayrıntılarını otomatikleştirir.
- Yük dengeleme yapar.
- Otomatik ölçeklendirme yapar
- Uygulamanın yüklenip çalışmasını sağlar.
- Automates the details of capacity provisioning,
- Load balancing,
- Auto scaling,
- Application deployment,

Why AWS Elastic Beanstalk?

- Automates management tasks:
 - **Yönetim görevlerini otomatikleştirir.**
- Monitoring, **İzleme**
- Version deployment, **Versiyon yükleme**
- Health check **Kaynakların sağlık taraması**
- Log **Kayıt alma**



Monitoring



Health Check



Bu tasklari otomatize ettiđi için bize herşeyin içinde olduđu güzel bir monitoring sunuyor. Yeni herşeyi derli toplu görmüş oluyoruz.



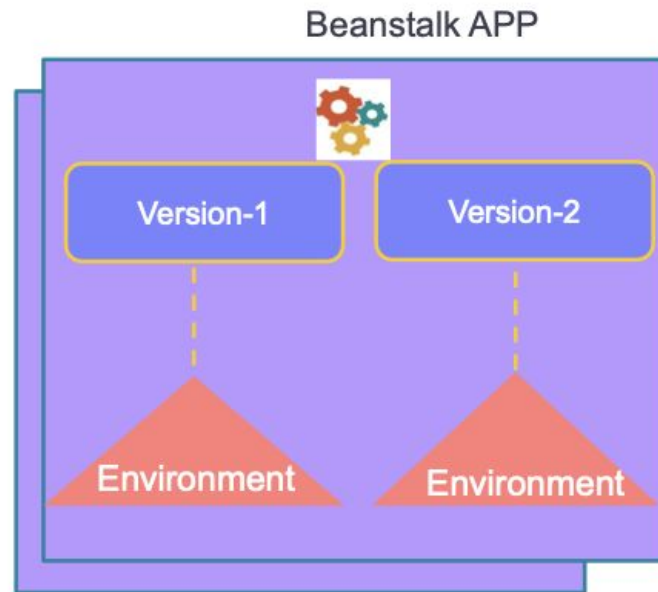
Basic Concepts of Elastic Beanstalk

Application

Sistemin tamamına Application diyoruz.
Application, Elastic Beanstalk bileşenlerinin mantıksal bir koleksiyonudur.
Tüm bileşenleri kapsar.

Application version

Bir WEB uygulaması için konuşlandırılabilir kodun özel etikeli yenilenmesidir.
Application Version'ı bir WAR File olarak düşünebiliriz. Her versiyonda yeni bir "zip" dosyası yüklüyor buda application version oluyor



Not: Bizim burada yüklediğimiz application, Application Version, sistemin tamamı da Application oluyor.

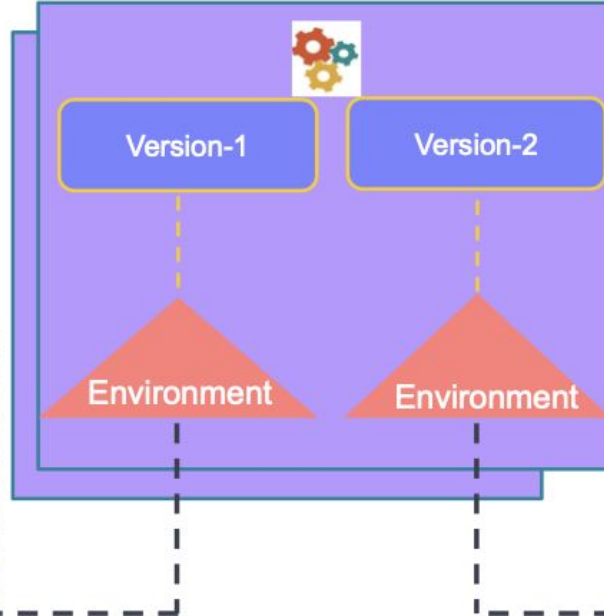


Basic Concepts of Elastic Beanstalk

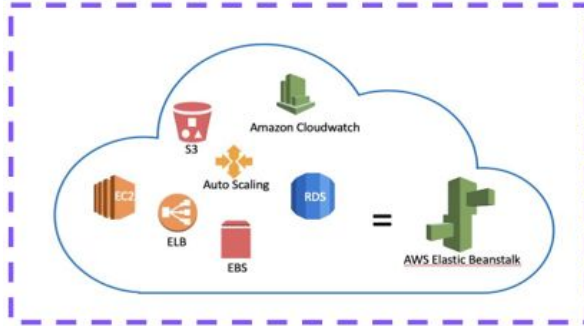
Burada iki ayrı Enviroment ve iki ayrı Version var.



WAR File



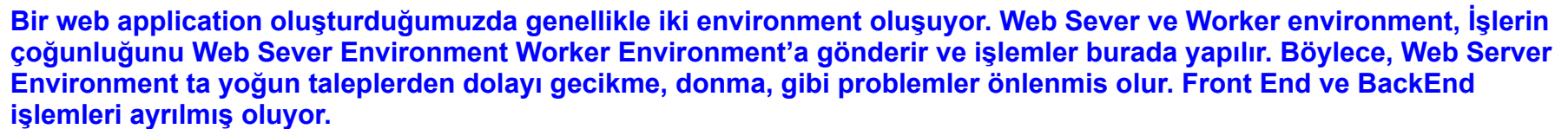
Beanstalk APP

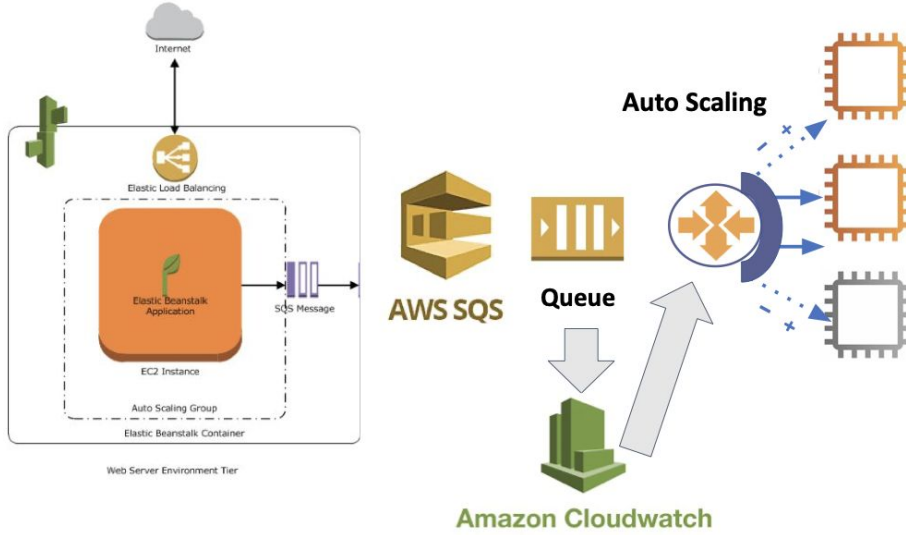


Environment

Environment, bir application version'ı çalıştıran bir AWS kaynakları topluluğudur.
Aynı anda bir environment ta yalnızca tek bir application version çalışabilir.

**Elastic Beanstalk'da
Application kısmında
Platform menüsünden
Environment türünü
seçebiliyoruz ve Elastic
Beanstalk bunu otomatik
oluşturuyor.**





`ApproximateNumberOfMessagesVisible`

Web server dan gelen istekleri QUEUE (Kuyruk) sıralamaya koyarak Auto Scaling Group a yönlendiriyor. Cloudwatch da, bu talepleri gözlemleyerek, talep arttıkça, duruma göre Auto Scaling Group'a EC2 sayısını artırıp azaltmasını söylüyor.

Platform:

Desteklenen platform türleri

Platform

Platform

Java ▼

Platform branch

Corretto 11 running on 64bit Amazon Linux 2 ▼

Platform version

3.0.3 (Recommended) ▼

Supported platform versions

- Docker
- Multicontainer Docker
- Preconfigured Docker
- Go
- Java SE
- Tomcat
- .NET Core on Linux
- .NET on Windows Server
- Node.js
- PHP
- Python
- Ruby

Summary of Terms / Concepts

Concept	What it Means
Application	Logical collection of Elastic Beanstalk components required for a working deployment
Application Version	A labelled version of an application (e.g. 1.0, 1.1, 2.0, etc...)
Environment	A set of AWS resources running a specific application version (e.g. DEV, TEST, PROD)
Environment Tier	The type of application that an environment runs (either Web or Worker)
Platform	Combination of OS, programming language, web server - i.e. the “technology stack”

Application: Elastic Beanstalk'ın çalışma ortamı(environment) tarafından mecbur koşulan bileşenlerinin mantıksal olarak bir araya gelmesine denir.

Application Version: Uygulamaların versiyonunu belirtir.

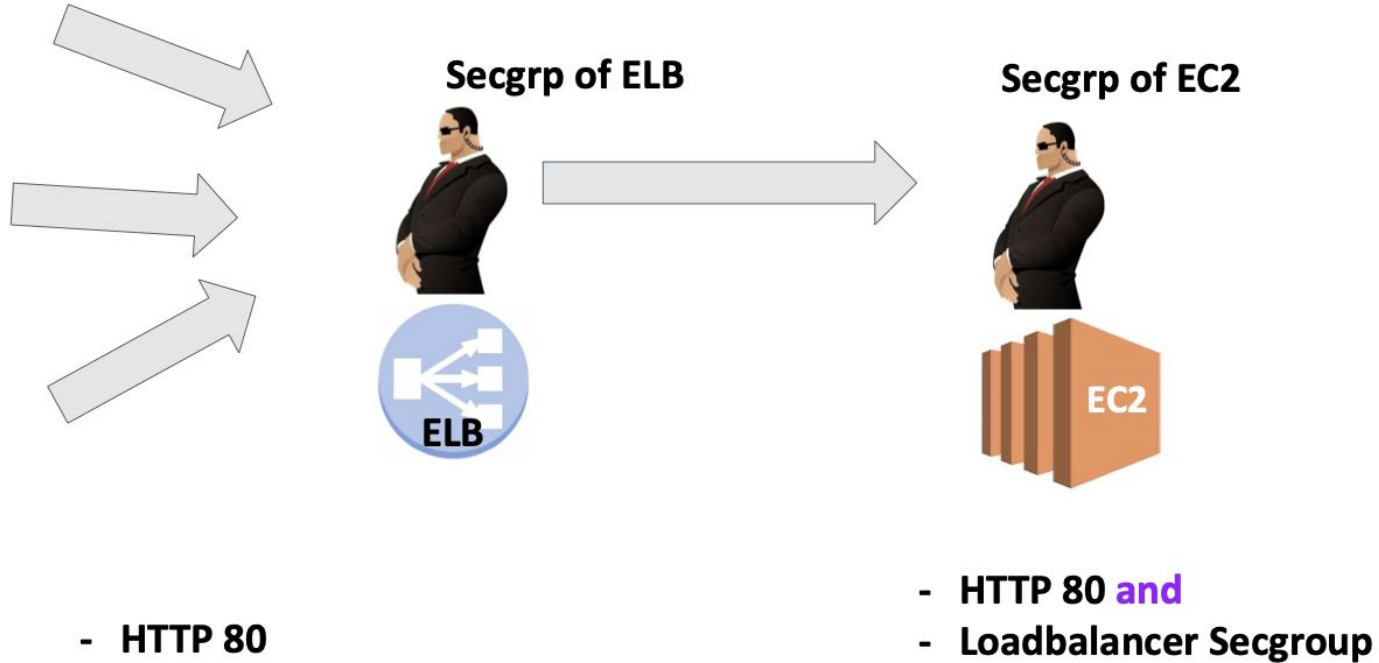
Environment: Uygulamanın belli bir versiyonunu çalıştıran AWS kaynaklar grubudur.

Environment Tier: Bir environment'ın çalıştırdığı programın tipidir.

Platform: İşletim sistemi, programlama dili ve Web Server'ın kombinasyonuna denir.

ELB(Elastic Load Balancer)'ın security group'u üzerinden gelen trafiğe izin vermiş oluyor.

EC2 nun security group'u, sadece ELB üzerinden gelen trafiğe açık olduğu için, EC2 nun public IP si ile direkt bir bağlantı yapmamıza müsaade etmiyor.



Hem ekstra bir Security, hem de trafiğin tek bir kaynak üzerinden aktarılması için, EC2 larda ELB security group kullanılır.

Deployment Metodları

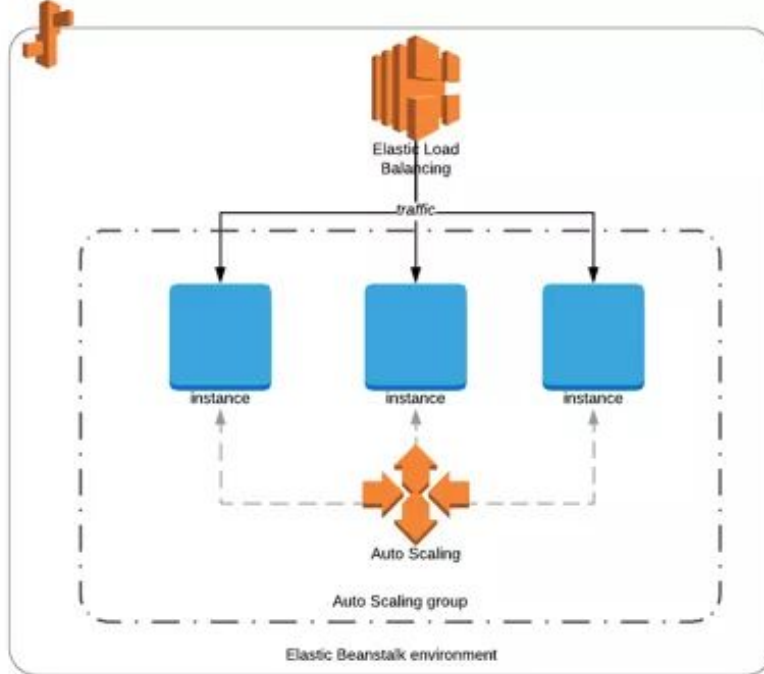
Deployment policy – Deployment seçenekleri:

- **All at once** – Yeni sürümü tüm instancelara aynı anda yüklüyor(deploy). Yükleme(deploy) gerçekleşirken ortamınızdaki tüm instance'lar kısa bir süre için hizmet dışı kalır.
- **Rolling** – Yeni sürümü gruplar halinde yüklüyor. Her toplu iş, dağıtım aşamasında hizmet dışı bırakılır ve ortamınızın kapasitesi bir toplu iş (batch) içindeki instance sayısı kadar azalır.
- **Rolling with additional batch** – Yeni sürümü gruplar halinde yüklüyor, ancak dağıtım işlemi sırasında tam kapasiteyi sağlamak için önce yeni bir instance grubunu başlatır.
- **Immutable** – Değişmez bir güncelleme gerçekleştirerek yeni sürümü yeni bir instance grubuna yüklüyor.
- **Blue/Green deployments** - Önce mevcut sistemi clone layıp, yeni sistem çalışır hale gelince, trafiği yeniye yönlendiriyor.
- **Traffic splitting** – Yeni sürümü yeni bir instance grubuna yüklüyor ve gelen trafiği mevcut uygulama sürümü ile yenisi arasında geçici olarak bölüyor.

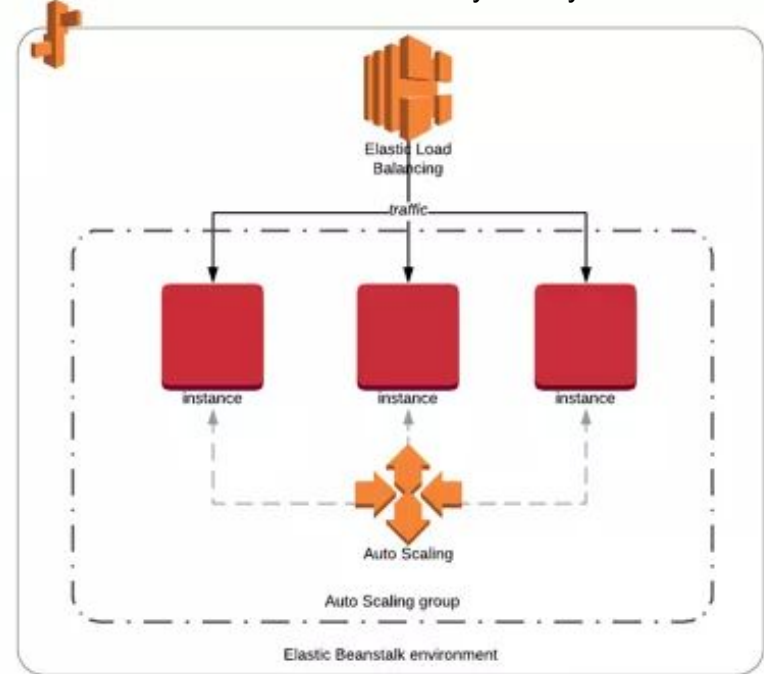
All at once – Yeni sürümü tüm instancelara aynı anda kurar(deploy). Kurulum(deploy) gerçekleşirken ortamınızdaki tüm instancelar kısa bir süre için hizmet dışı kalır.

Mavi işlemin devam ettiği, **Yeşil** başarılı olduğunu ve bittiğini, **Kırmızı** da başarısız olduğunu gösterir

All at once Üç application'ı aynı anda güncelleme yapıyor. **Avantajı, çok hızlı olması, dezavantajı ise, bir sorun olduğunda hepsinde olur ve sistem ilerlemez.** Çalışan durumdaki instance'a yeni versiyonu aniden yüklemeye çalıştığımızdan dolayı, işlem başarılı olsa bile arada kesinti mutlaka olmaktadır. **Test ve Development** ortamında kullanılması tavsiye ediliyor.



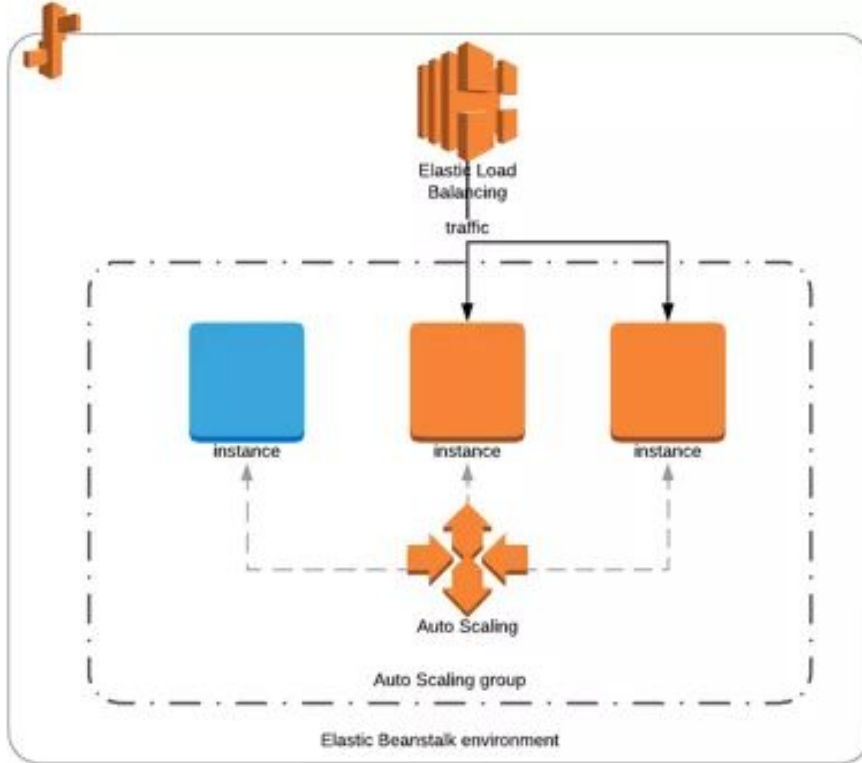
1. Deployment starts in all instances



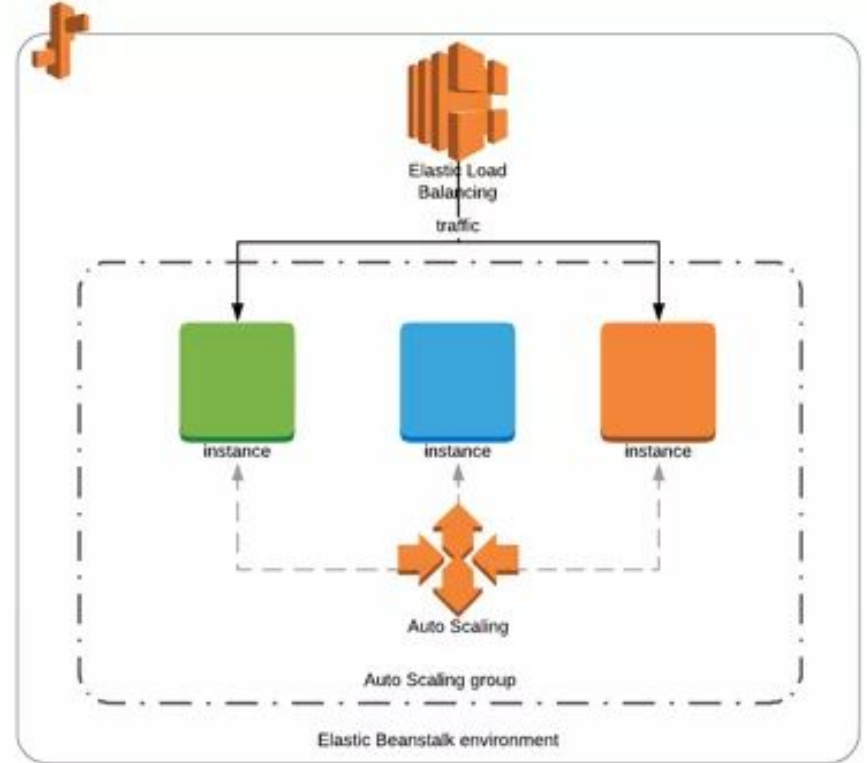
2. In case of failure, all fails!

Rolling – Yeni sürümü gruplar halinde yükler. Her toplu iş, yükleme aşamasında hizmet dışı bırakılır ve ortamınızın kapasitesi bir toplu iş (batch) içindeki instance sayısı kadar azalır.

Önce 1. Instance yeşil olunca 2. Instance geçiyor, sonra 2. Bitince 3.ye geçiyor.



1. Deployment starts in the first batch.

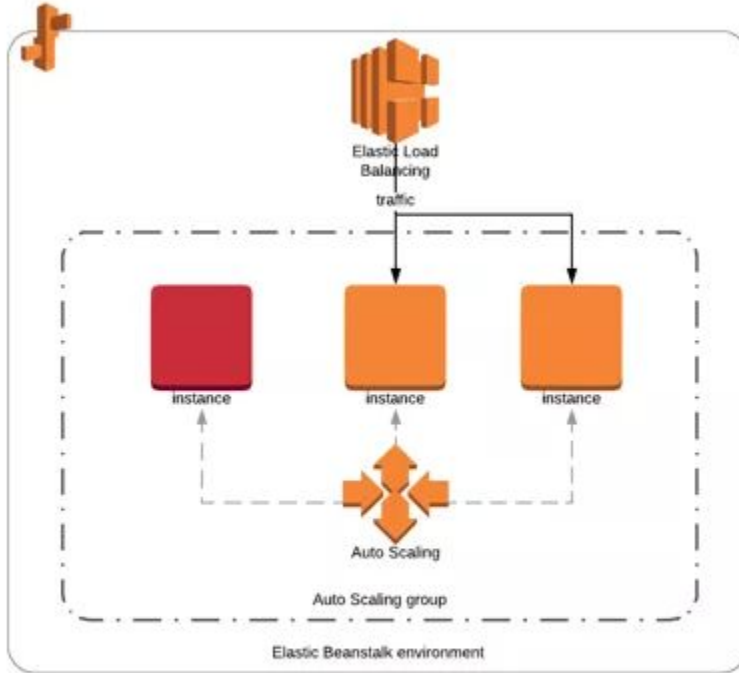


2. Deployment continues with the next batch.

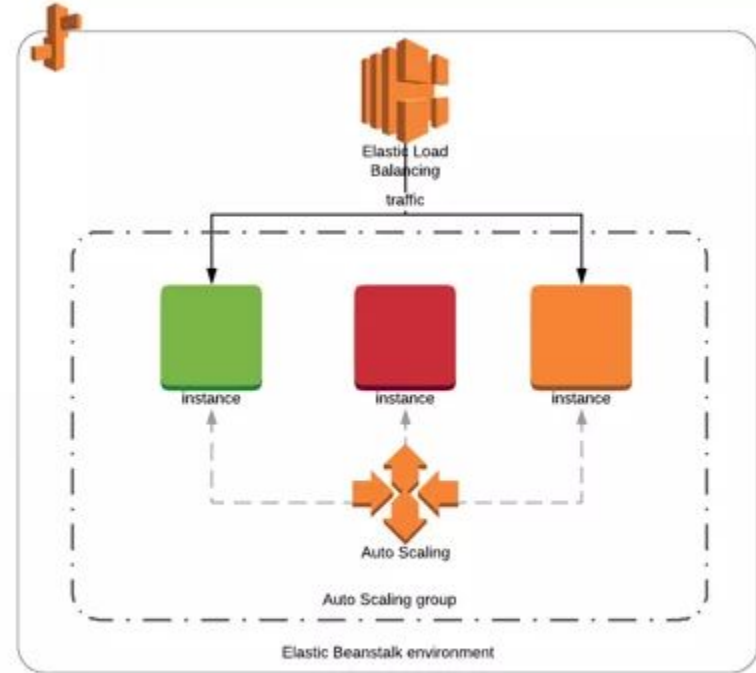
Fail durumunda ise:

1. Başarısız olursa trafiği 2.ve 3. instancelara yönlendirebiliriz. Bunlar eski versiyon ile devam eder ve Sistemde aksama olmaz.

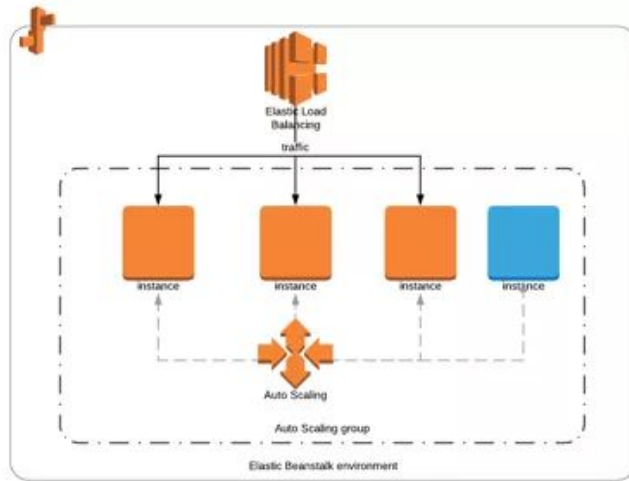
1.başarılı fakat 2.si başarısız olursa, 1. Instance Versiyon 1 de kalırken, 3. Intance eski versiyonda kalır ve sistem ilerlemez. Yani 1 yeni ve 1 eski versiyon kalmış olur.



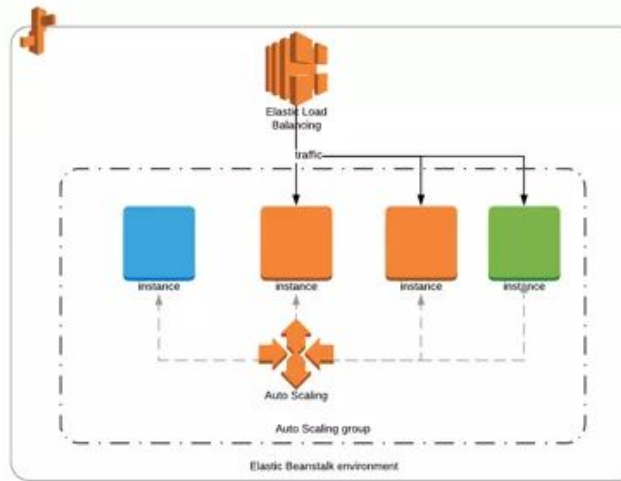
In case of the first batch failure, other instances continue to serve the previous version.



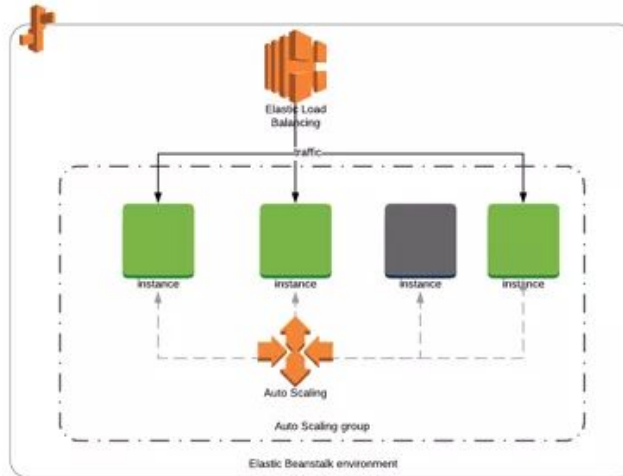
In case of the second batch failure, there are two versions served.



1. Deployment starts by launching new instances for the first batch.



2. Deployment continues with the next batch.



3. After the final batch, excess instance is terminated.

Rolling with additional batch

Bu değişimi mevcut durum üzerinde yapmıyor, yeni bir instance üzerinde yapıyor.

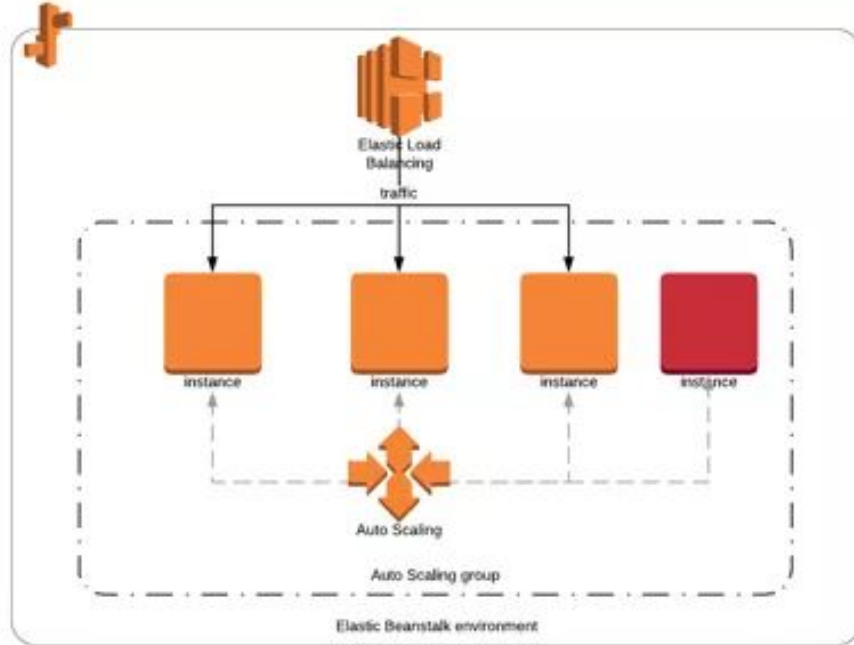
Yeni eklenen 4. Instance başarılı olursa diğerlerini de buna göre update ediyor.

3 instance da başarılı olursa, 4.yü siliyor.

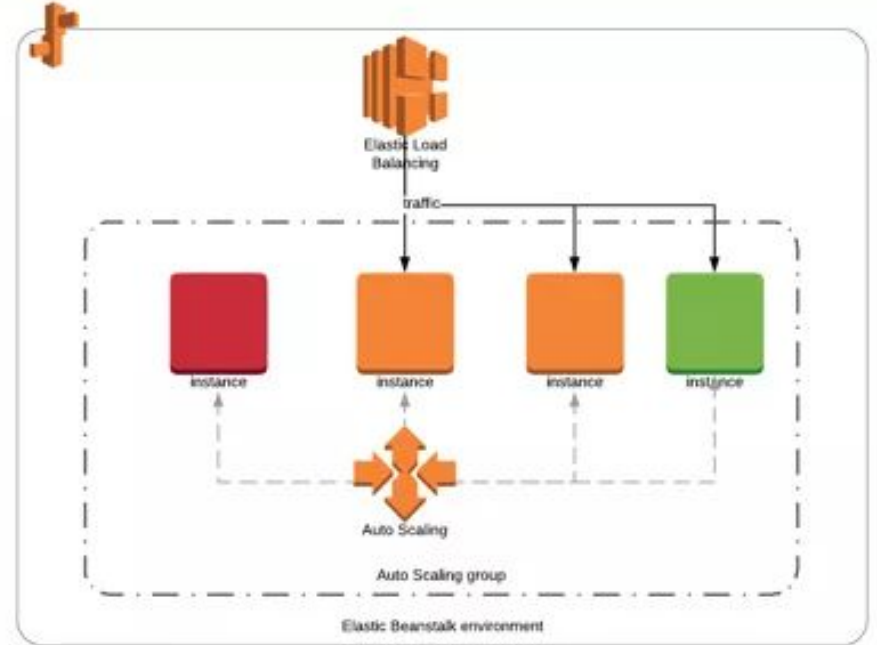
Rolling with additional batch * BAŞARISIZ OLURSA

Yeni eklenen instance başarısız olursa, 3 tane eski versiyon olan instancelarla devam etmiş oluyor.yeni örnek sonlandırılır.

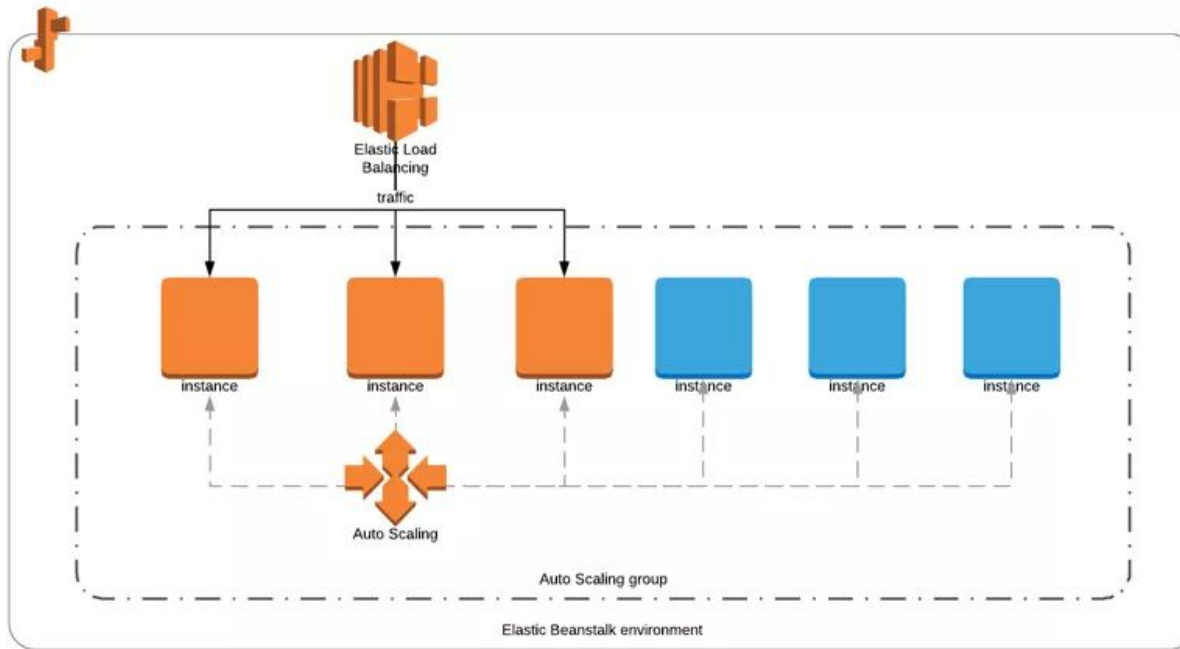
Yeni oluşturulan instance başarılı olup, 1. Instance Başarısız olursa, işlem yarıda kesileceği için aynı anda iki versiyon olma durumu olur.



In case of the first batch failure, the new instance is terminated. The fleet size is not effected.



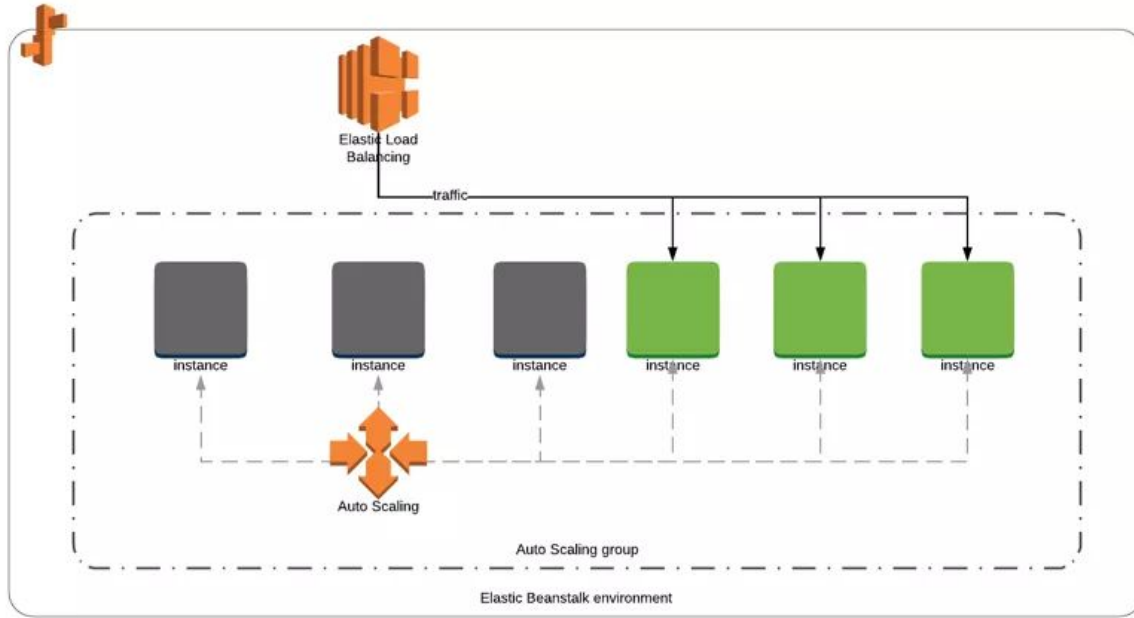
If the second batch fails, there will be two versions running as in rolling deployments without additional batch.



1. Deployment starts by duplicating the instances and deploying the app in new instances

Immutable - (Risk almıyorsunuz)

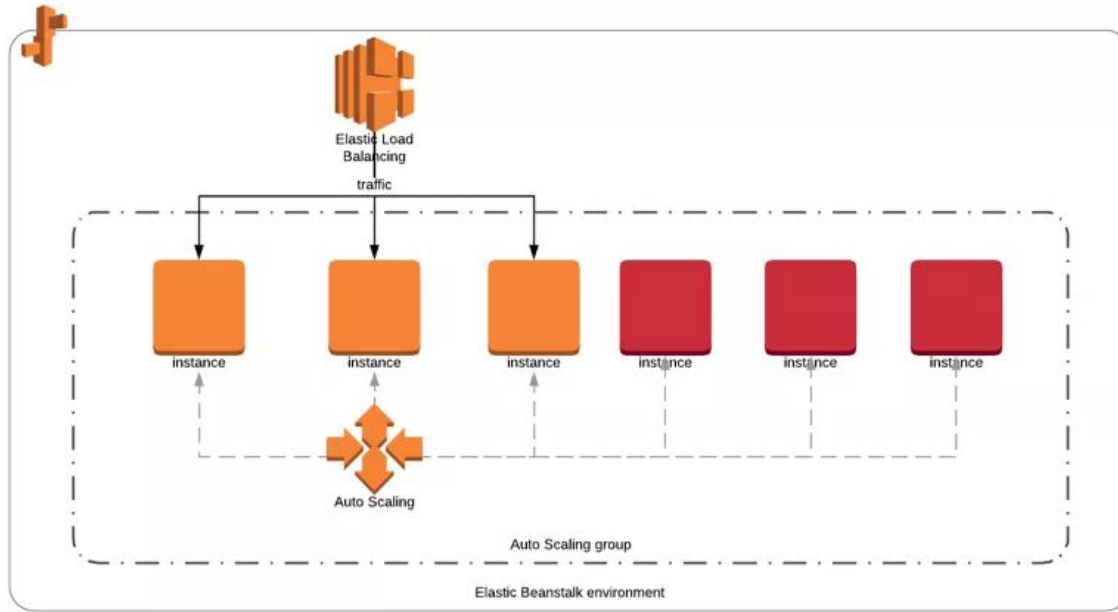
3 instance varsa 3'ünü de kopyalıyor ve yüklemeyi yeni instancelara yaparken sistemde kesinti olmuyor.



2. After deployment succeeds, older instances are terminated.

Immutable Başarılı Olursa:

Yeni eklenen 3 instance da başarılı olursa onları aktive ediyor ve 3 tane eski versiyon olan instanceları kaldırıyor. Sistemde kesinti olmuyor



In case of failure, the new instances is terminated and traffic is not effected.

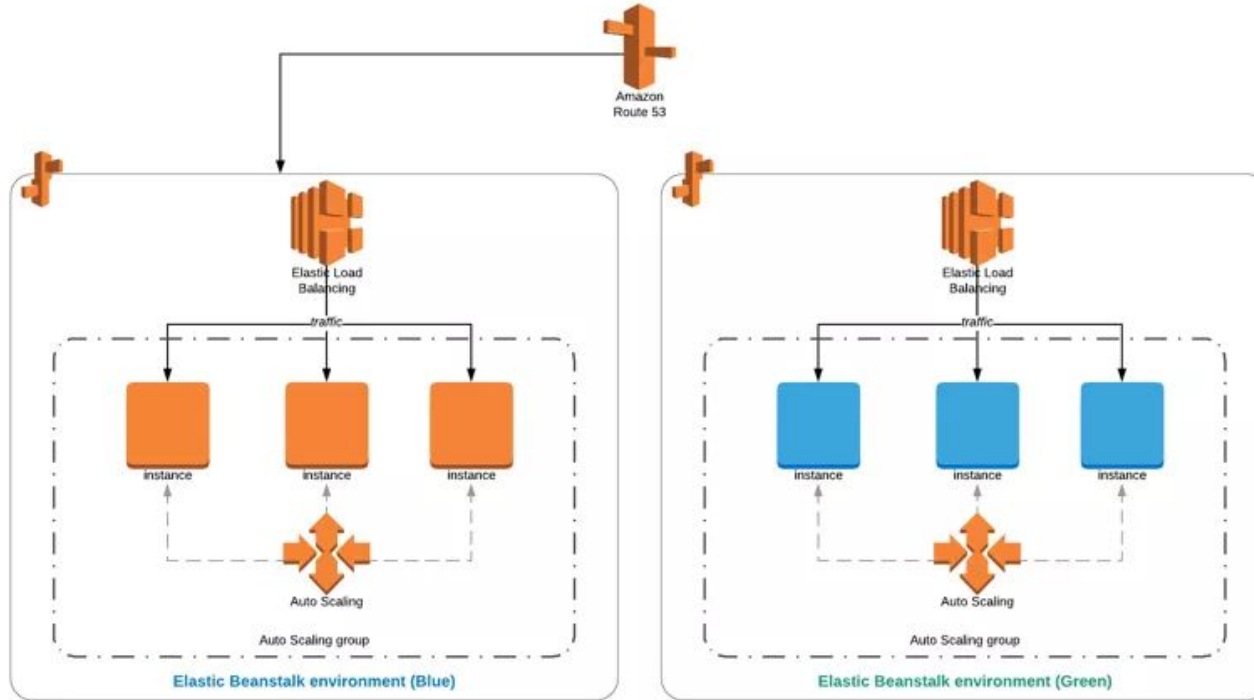
Immutable Fail Olursa:

Yeni eklenen 3 instance da başarısız olursa 3 tane eski versiyon olan instanceları devam ediyor. Kesinti oluşmuyor.

Blue/Green deployments

Burada Cluster'ın aynısını birebir kopyalıyor. Bu durumda iki tane Load Balancer ortaya çıkıyor. Trafik'in mevcut cluster'a akmasını sağlayan Route53'deki **ALIAS** kaydı olmasıdır.

Yeni Cluster başarılı olduğunda Route53'deki **ALIAS** kaydını değiştirip, yeni Load Balancer'a yönlendiriyoruz.



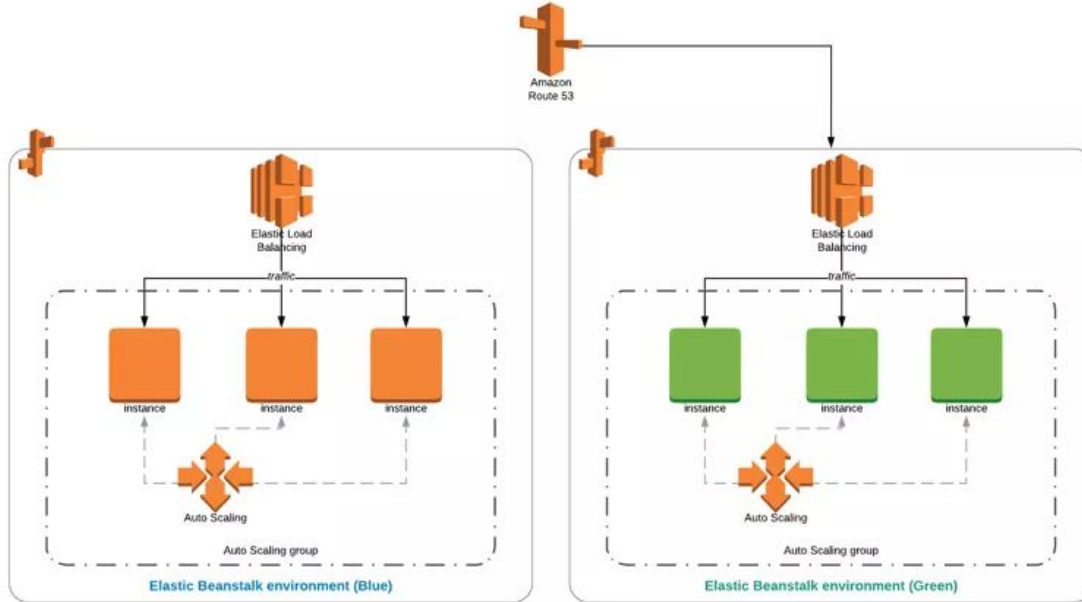
Old environment continues to serve the traffic during deployments on the cloned environment. In case of failure, it will not be effected.

Deployments are run on the cloned environment

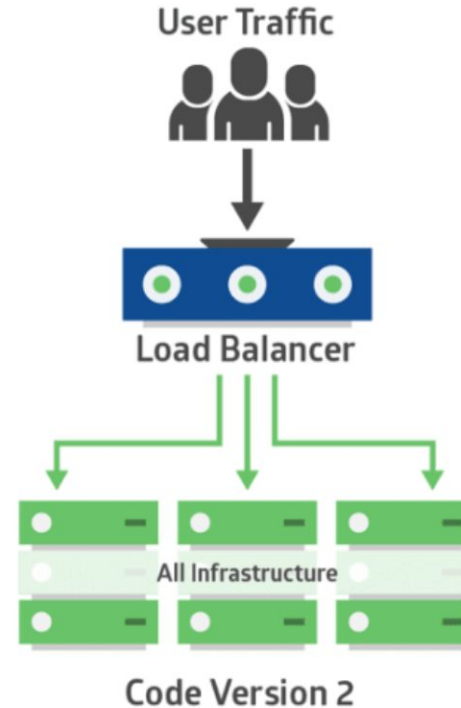
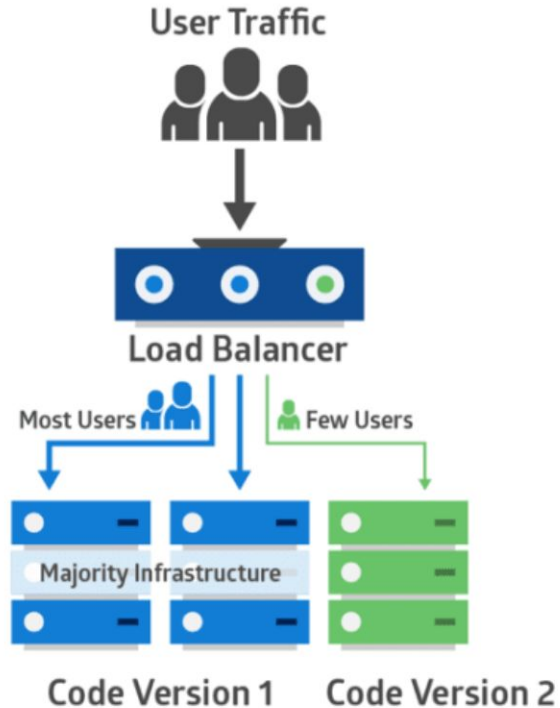
Blue/Green deployments

Burada Cluster in aynısını birebir kopyalıyor. Bu durumda iki tane Load Balancer ortaya çıkıyor. Trafığın mevcut cluster'a akmasını sağlayan Route53 deki **ALIAS** kaydı olmasıdır.

Yeni Cluster başarılı olduğunda Route53 deki **ALIAS** kaydını değiştirip, yeni Load Balancer'a yönlendiriyoruz.



Traffic splitting – Elastic Beanstack, trafiği, yeni instancelara sağlıklı yüklemeler yapıldıkça aşama aşama versiyonun olduğu instancelara yönlendiriyor. Aynı anda iki versiyon olmuş oluyor.



Congratulations!

Application Updated!

Congratulations!

Application Updated!

V1

V2- All at once

V3- Immutable

V4- Add Additional Batch

**Yükleyeceğimiz
Versiyonlarımızın
Ekran Görüntü
Başlıkları**