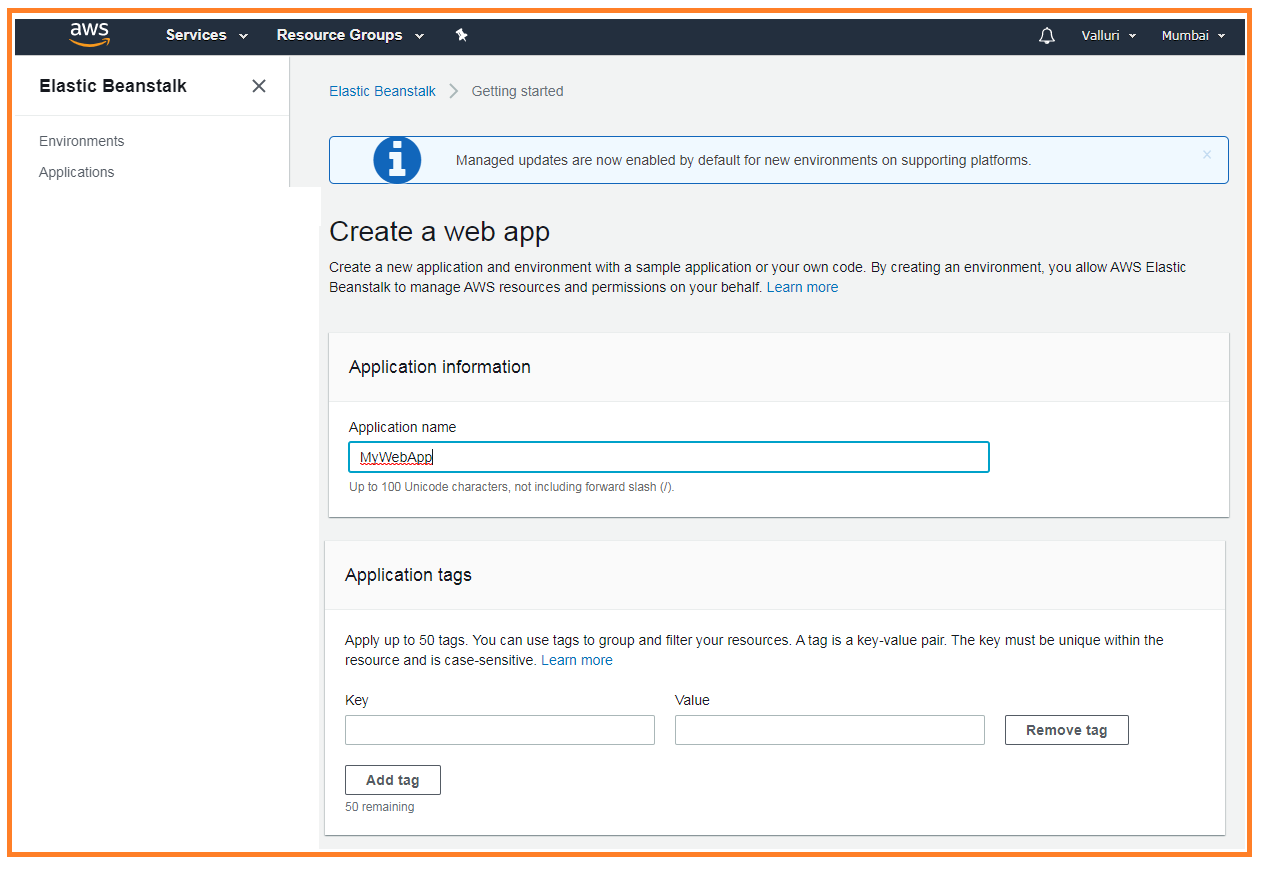
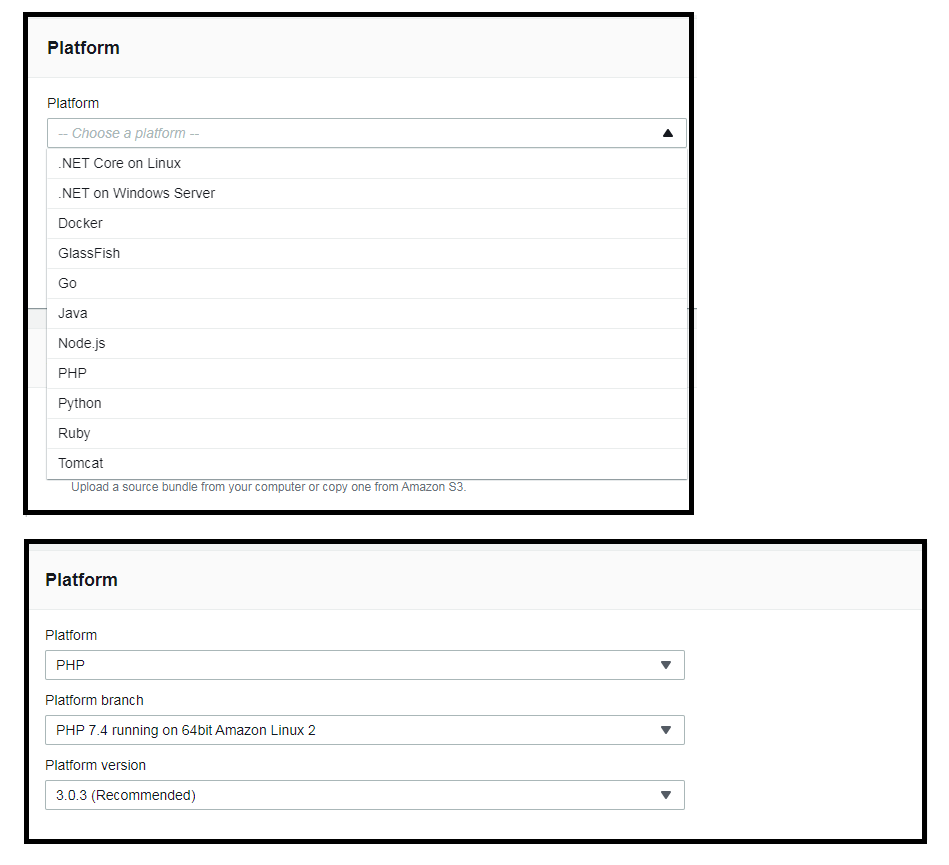
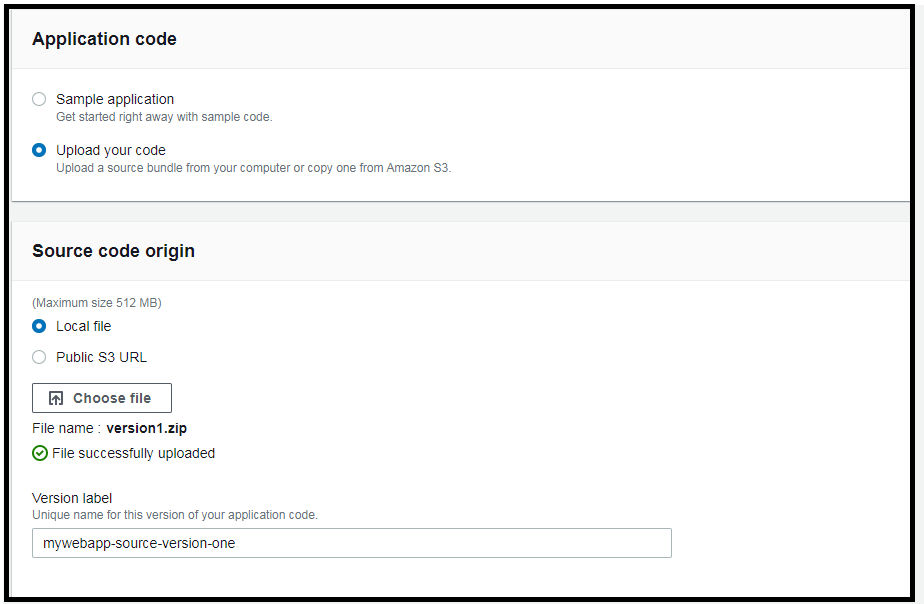
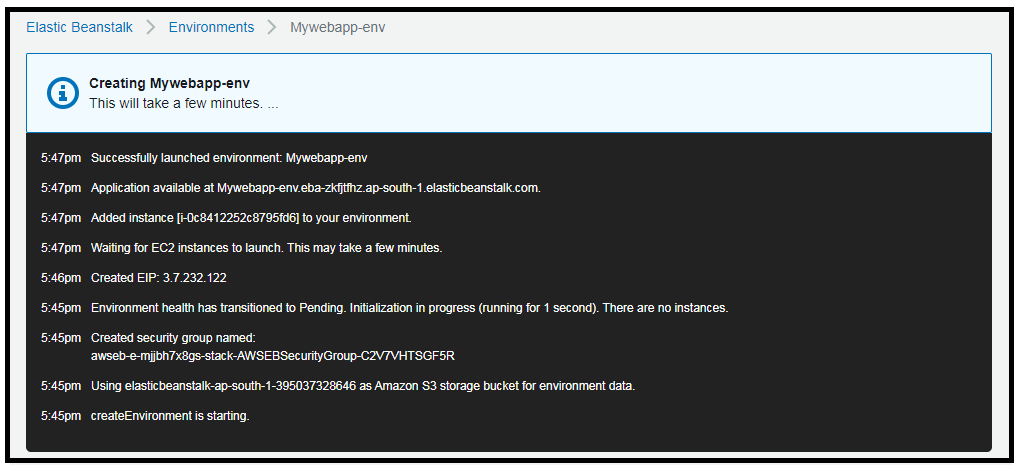
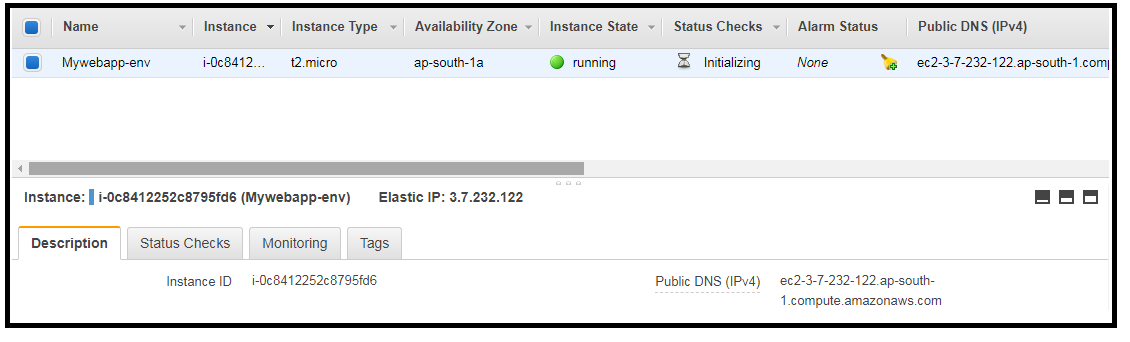
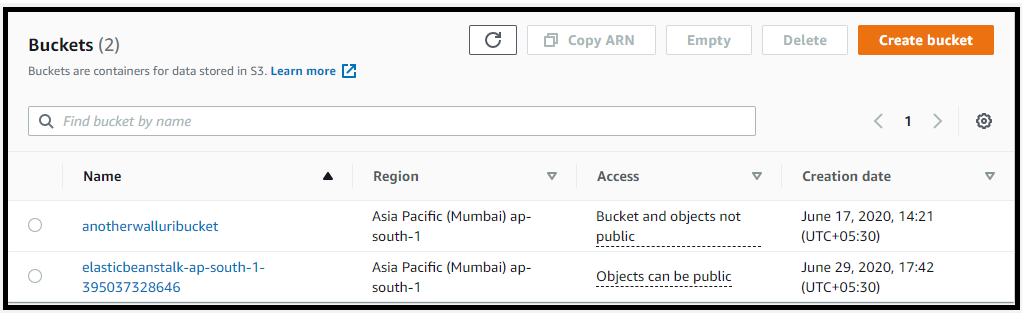
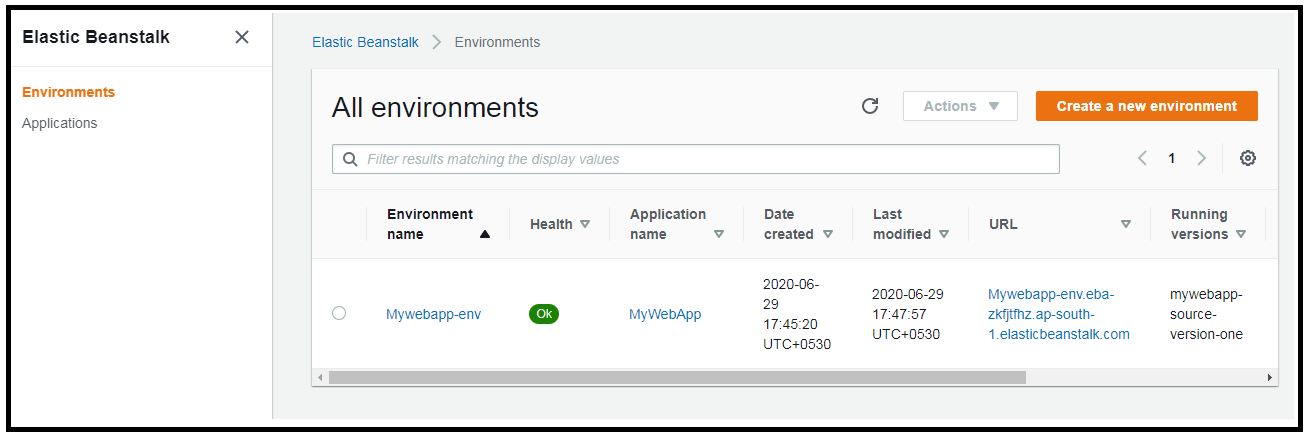
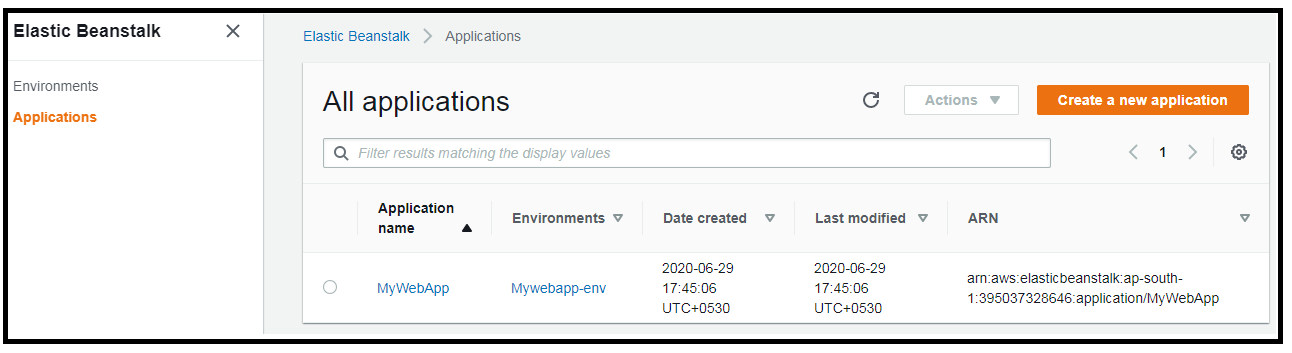
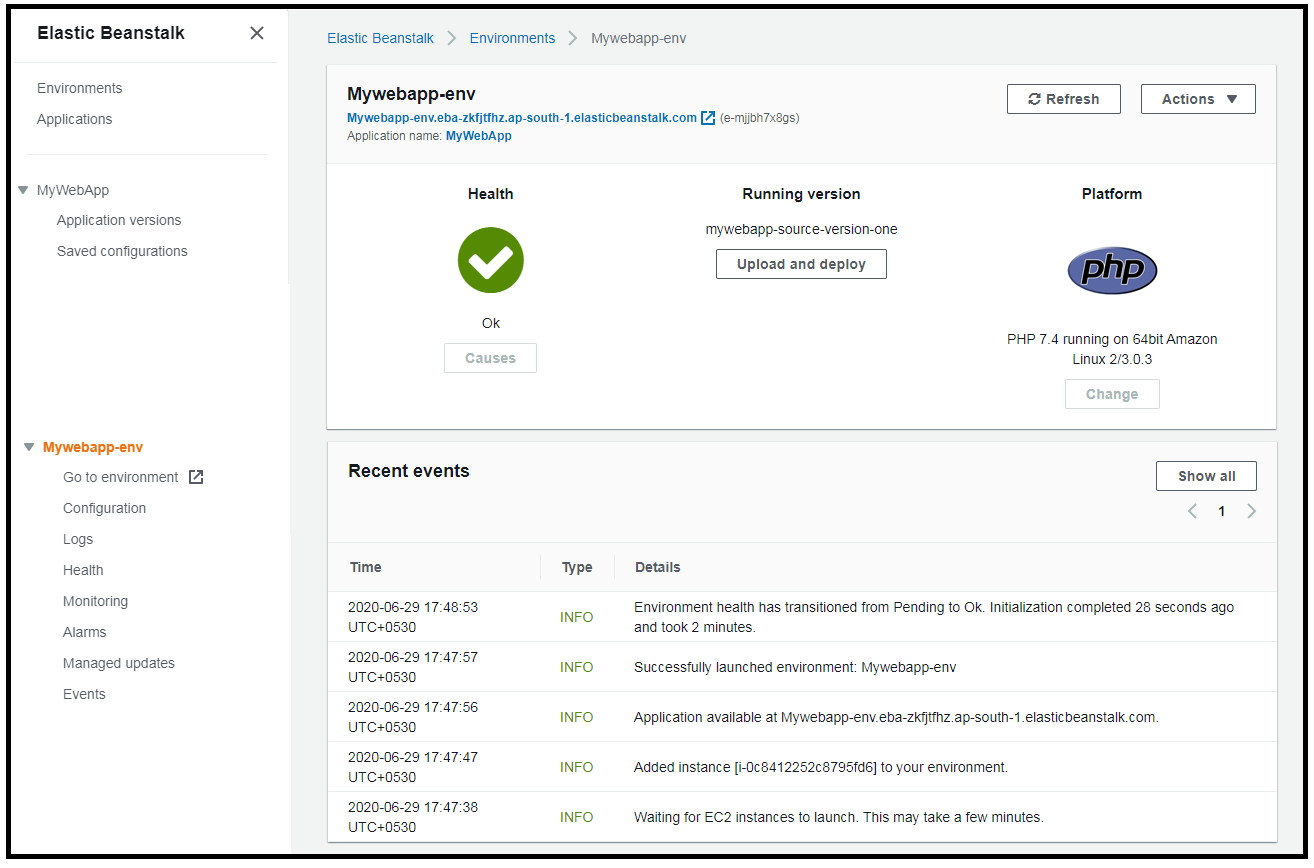
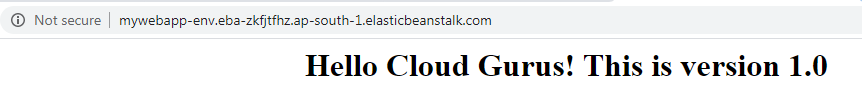
  
  
1. What is Elastic Beanstalk : It is a service for deploying and scaling web applications developed in many popular languages – Java, .net, PHP, NodeJS, Python, Ruby, Go, and Docker ONTO Widely used application server platforms like Apache Tomcat, Nginx, Passenger, Puma and IIS.  
  
It is a provisioning service – similar to cloud formation (scripting your infrastructure) which is JSON based + cloud formation templates.  
  
2. What is the benefit of using this service : Developers can concentrate on writing code and don’t have to worry about the underlying infrastructure. It is integrated with CloudWatch + XRay (Monitoring , metrics and health checks) .

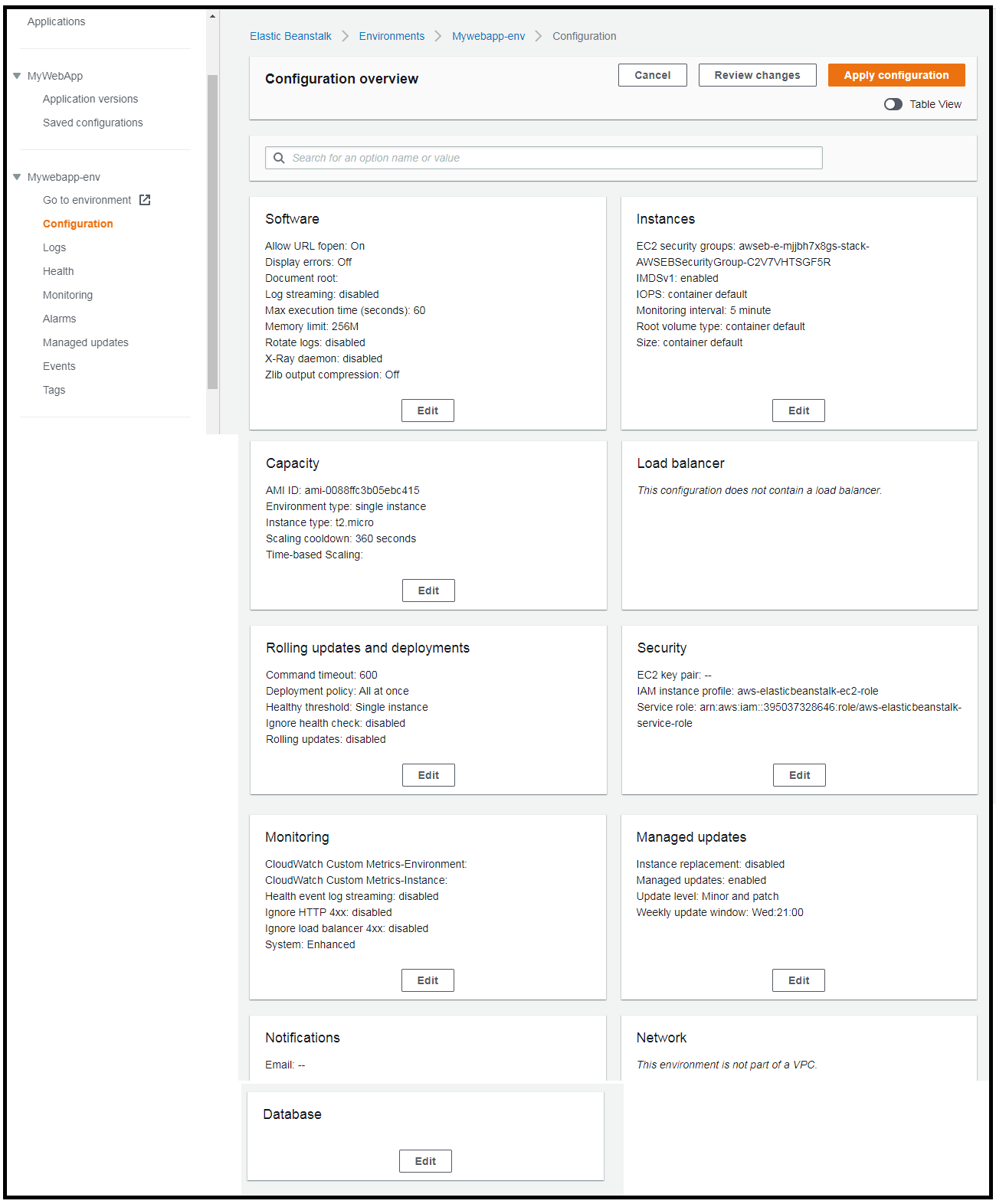
3. What does this service do : We just upload the code and Elastic Beanstalk will handle deployment, capacity provisioning, Load balancing, Auto Scaling and the applications health. We only pay for the aws resources required to store and run your application – EC2, S3(Every version of your code is stored in S3).

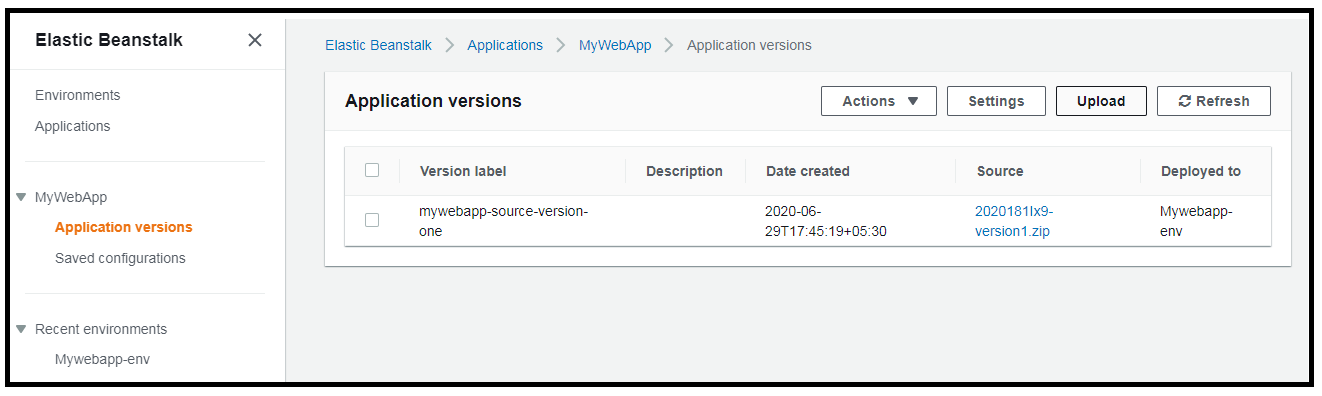
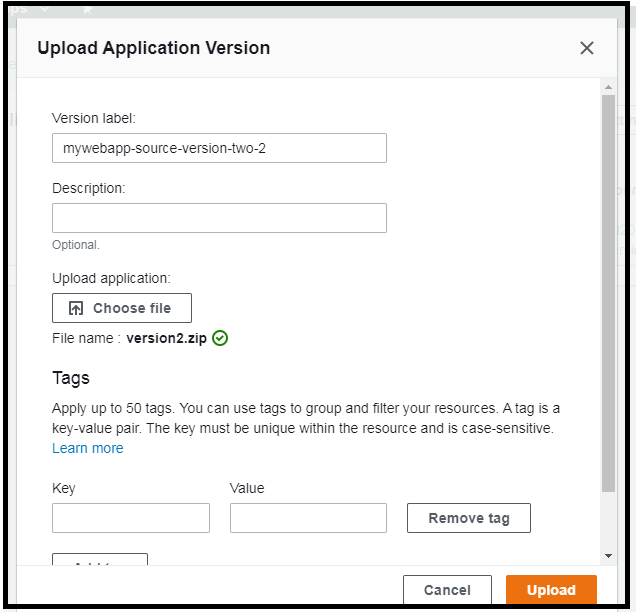
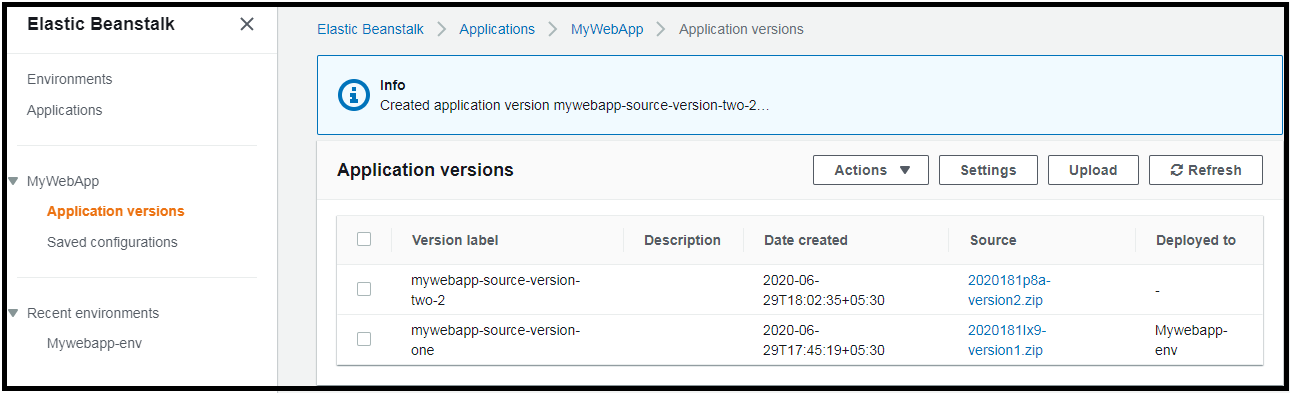
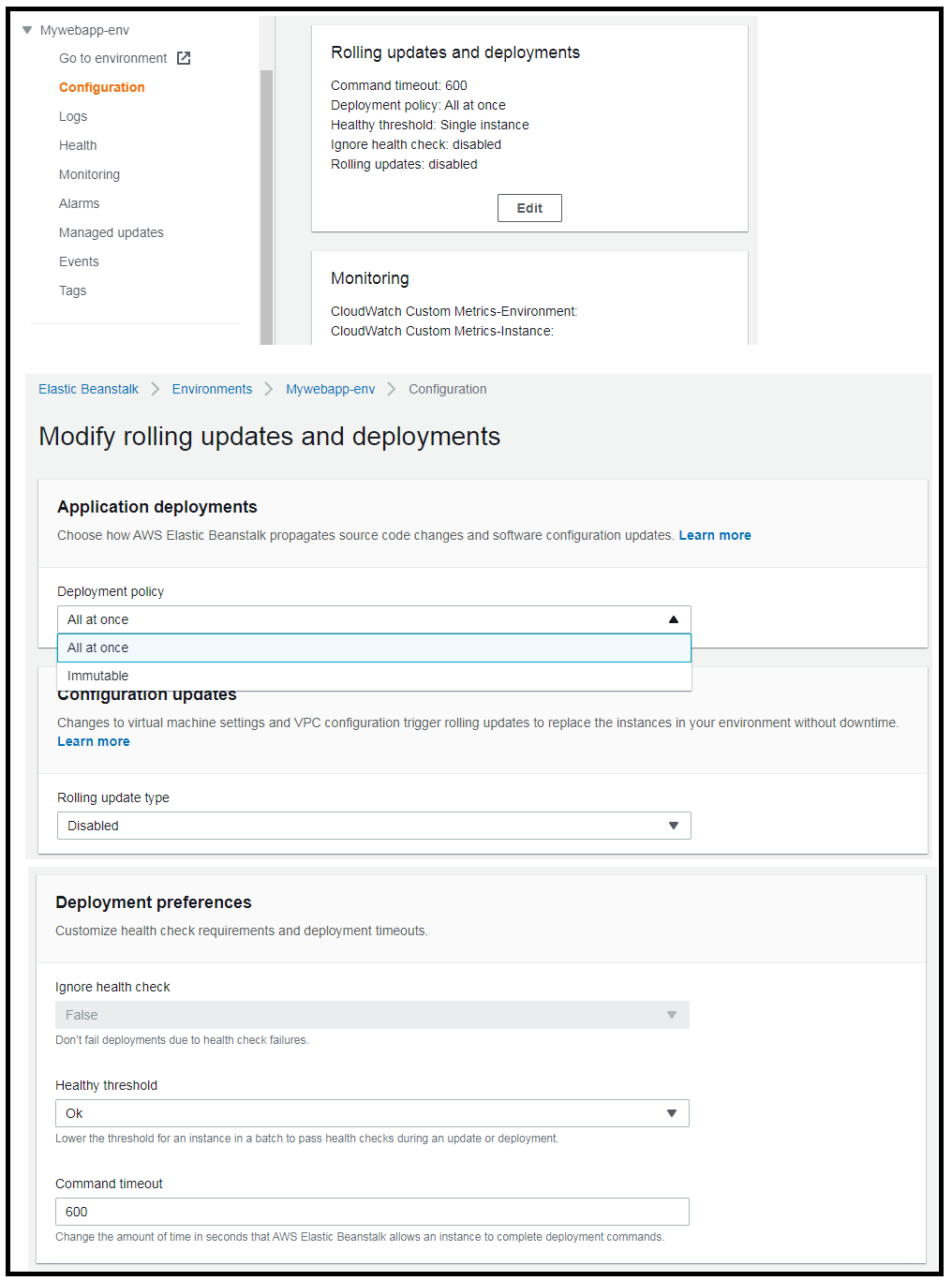
4. LAB : Create an EBStalk app and deploy new version of your app with ‘all at once’ deploy method.  
  
4.1  
  
  
  
  
  
  


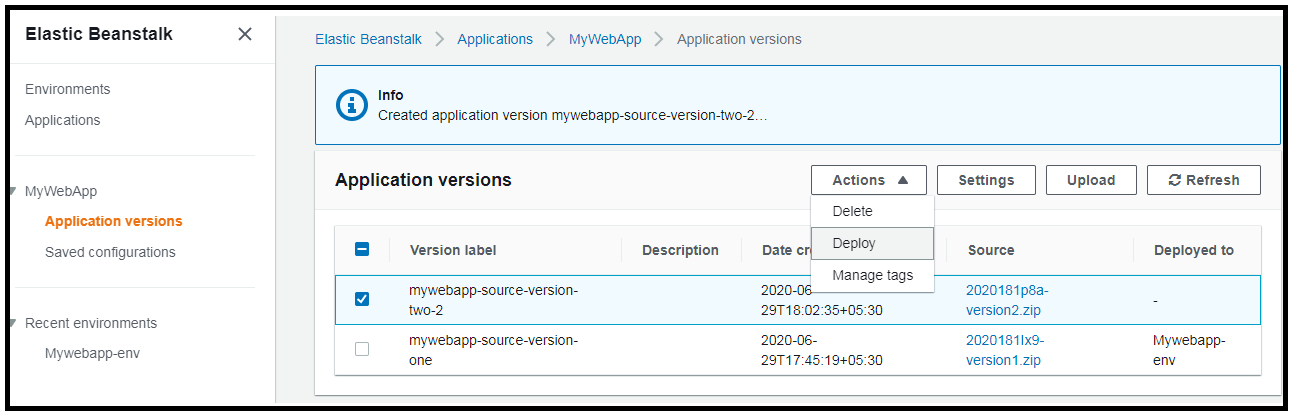
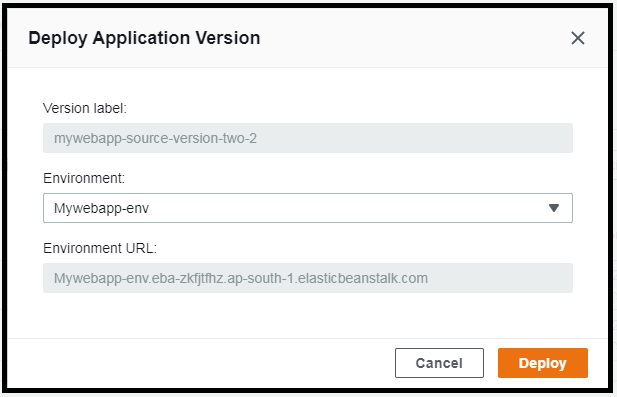
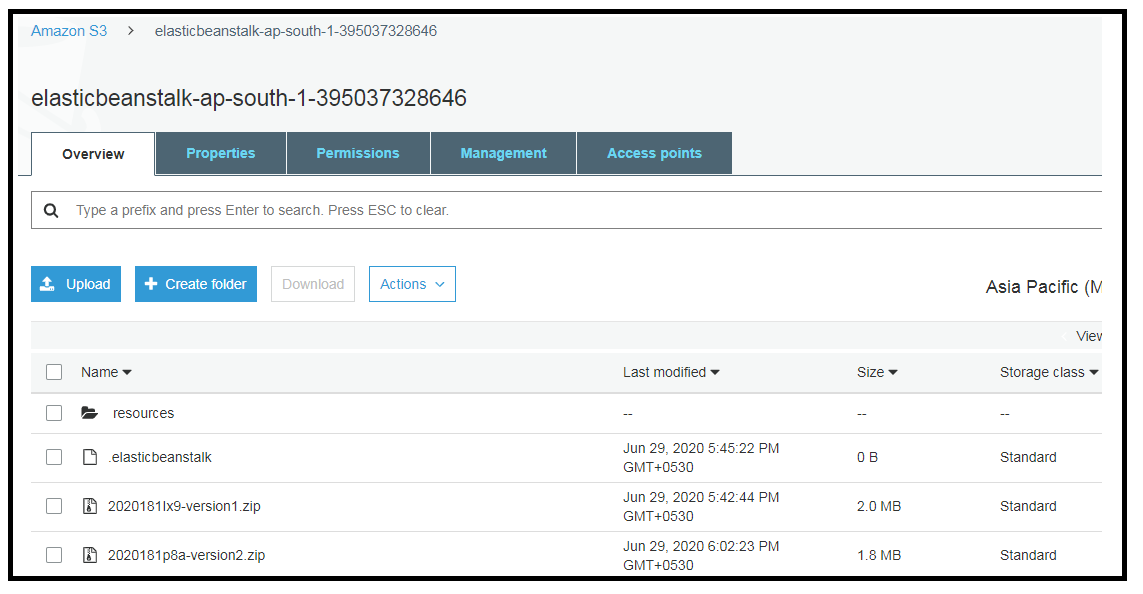
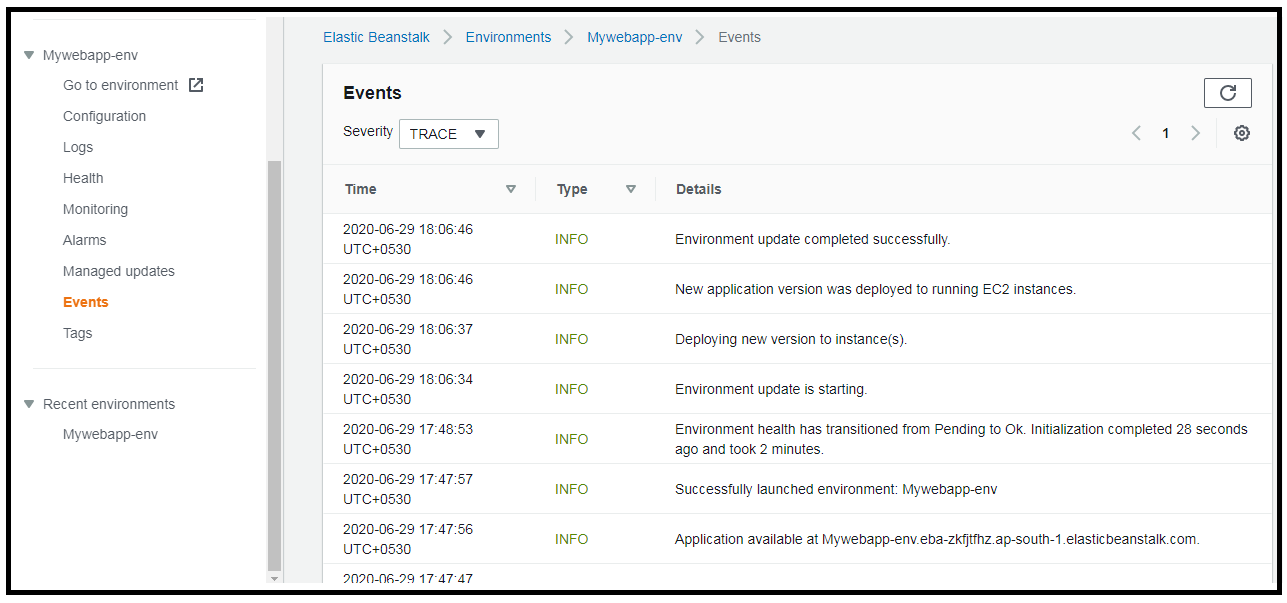


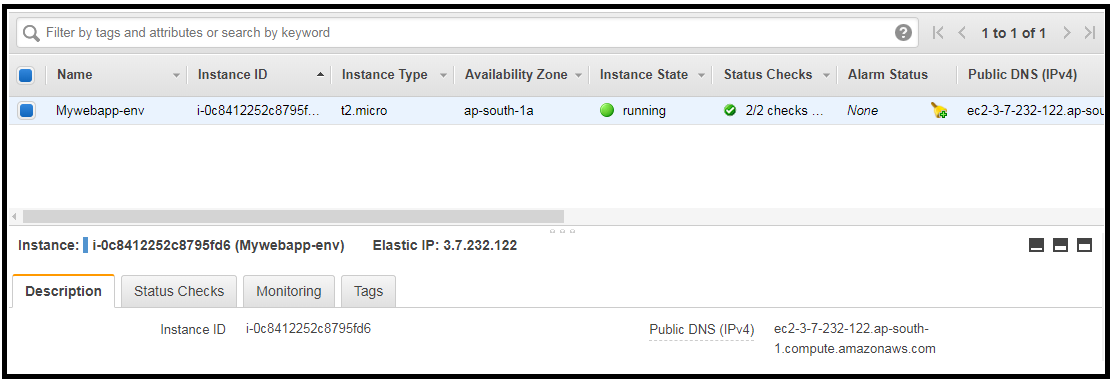
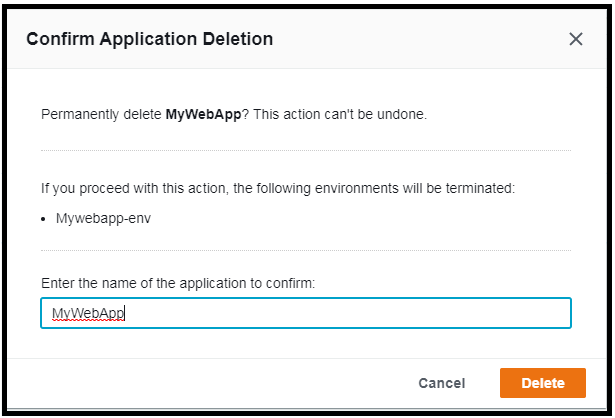
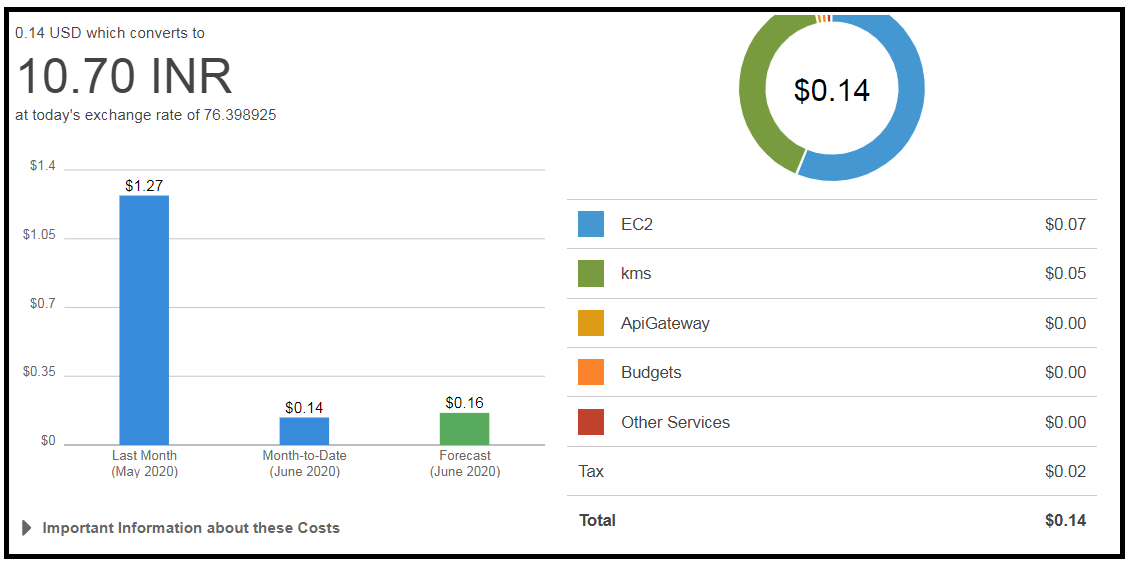
  






5. Updating Elastic Beanstalk.

Elastic BeanStalk Deployment Policy : Below are the options supported by Elastic BeanStalk for processing deployments.  
  
- All at once   
Deploys the new version to **all instances** **simultaneously**. [All of your instances are going to be out of service while the deployment takes place].  
  
You will **experience outage** while the deployment happens and is not ideal for mission critical production systems.  
  
If the **update fails** we need to rollback the changes by re-deploying the original version to all your instances.  
  
- Rolling   
  
Deploys the new version in batches.  
  
Each batch of instances **are taken out of service** while the deployment takes place. In this case the environment capacity will be reduced by the number of instances in a batch while the deployment takes place. This is not ideal for performance sensitive systems.  
  
If the update fails, we must perform an additional rolling update to rollback the changes.  
  
This is not ideal for performance systems, as systems with a ‘batch size’ are taken out of service.  
  
- Rolling with additional batch  
An additional batch of instances are launched + deploys the new version in batches. This maintains full capacity during deployment process.  
  
If the update fails we have to perform an additional rolling update to roll back the changes.  
  
No Downtime you maintain full capacity - This is the option.

Can afford downtime – Either ‘All at once’ or ‘Rolling update’  
  
  
- Immutable

This deploys a 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 autoscaling group, And then the old instances are terminated.

This maintains the full capacity during the deployment process.

The impact of a failed update is far less and the rollback process requires only terminating the new autoscaling group.

Best used for mission critical production systems.

6. AWS CLI