# AWS Beanstalk

## 목차

[AWS Beanstalk 1](#_Toc464029767)

[목차 1](#_Toc464029768)

[１ Beanstalk란 2](#_Toc464029769)

[가) Beanstalk 2](#_Toc464029770)

[나) (참조 : https://aws.amazon.com/ko/elasticbeanstalk/?nc2=h\_m1 ) 2](#_Toc464029771)

[２ Beanstalk 생성 2](#_Toc464029772)

[가) Beanstalk 페이지 진입 2](#_Toc464029773)

[나) Application 생성 3](#_Toc464029774)

[다) Environment 생성 4](#_Toc464029775)

[３ Beasntalk 배포 9](#_Toc464029776)

[가) War 파일 생성 9](#_Toc464029777)

[나) Beanstalk 에 War 배포 10](#_Toc464029778)

[다) 배포 기록하기 11](#_Toc464029779)

[４ Beanstalk 버전 되돌리기 12](#_Toc464029780)

[가) Beanstalk 버전 되돌리기 12](#_Toc464029781)

[５ Beanstalk TimeZone 설정 13](#_Toc464029782)

[가) JVM command line options 설정을 한다. 13](#_Toc464029783)

[６ Beanstalk Auto Scaling 설정 15](#_Toc464029784)

[가) Auto Sacling이란 15](#_Toc464029785)

[나) Auto Scaling 설정하기 15](#_Toc464029786)

[다) 그 외의 auto scaling에 대한 내용 : AWS AUTO SCALING 17](#_Toc464029787)

## Beanstalk란

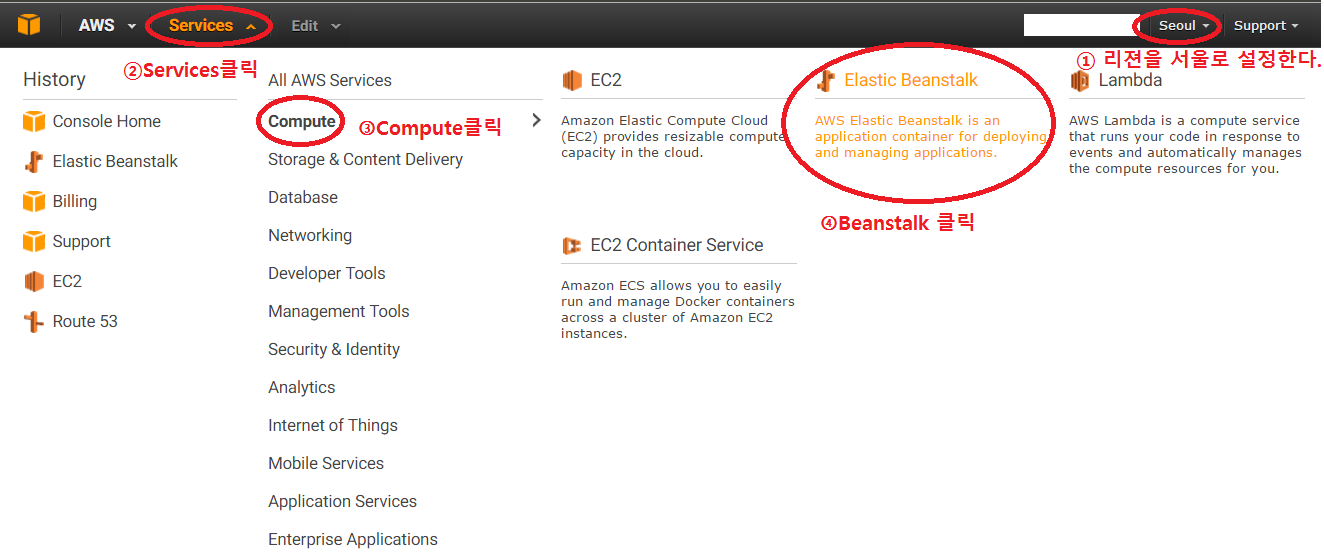
### Beanstalk 용어

* + 1. 코드만 업로드하면 용량 프로비저닝, 로드 밸런싱, 자동 크기 조정, 애플리케이션 상태 모니터링 등등 배포 자동 처리를 해주는 서비스

### (참조 : <https://aws.amazon.com/ko/elasticbeanstalk/?nc2=h_m1> )

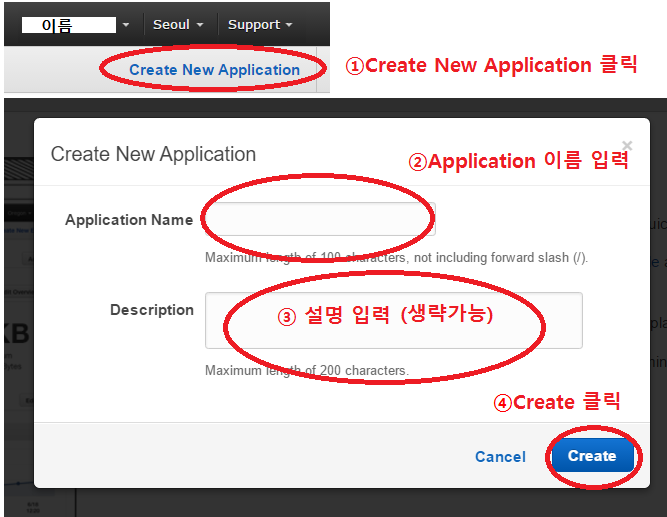
## Beanstalk 생성

### Beanstalk 페이지 진입



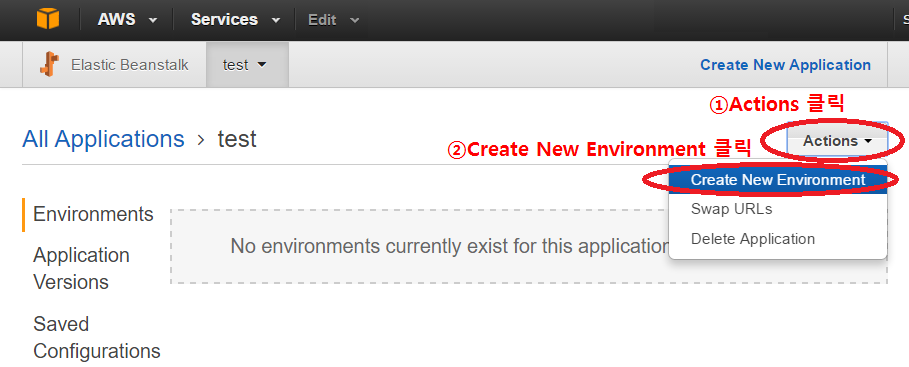
* + 1. 사용할 Region을 선택한다. Region에 따라 서비스 속도 차이가 있을 수 있다. 되도록 실제 서비스 되는 곳의 지역과 가깝게 하는게 좋다.
    2. 상단의Services를 클릭힌다.
    3. 나오는 드롭다운 메뉴 중 Compute를 클릭한다.
    4. 나오는 오른쪽 세부 메뉴의 Elastic Beanstalk 부분을 클릭한다.

### Application 생성

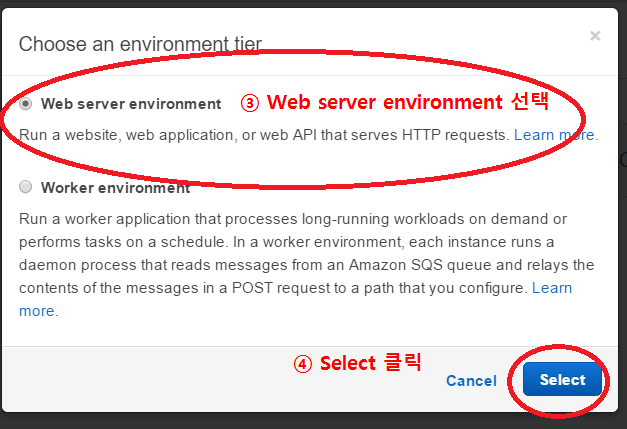


* + 1. 우상단에 위치한 Create New Application을 클릭한다.
    2. Application 이름을 입력한다.
    3. 설명을 입력한다. 생략이 가능하다.
    4. Create 클릭하면 applications 페이지로 이동된다.

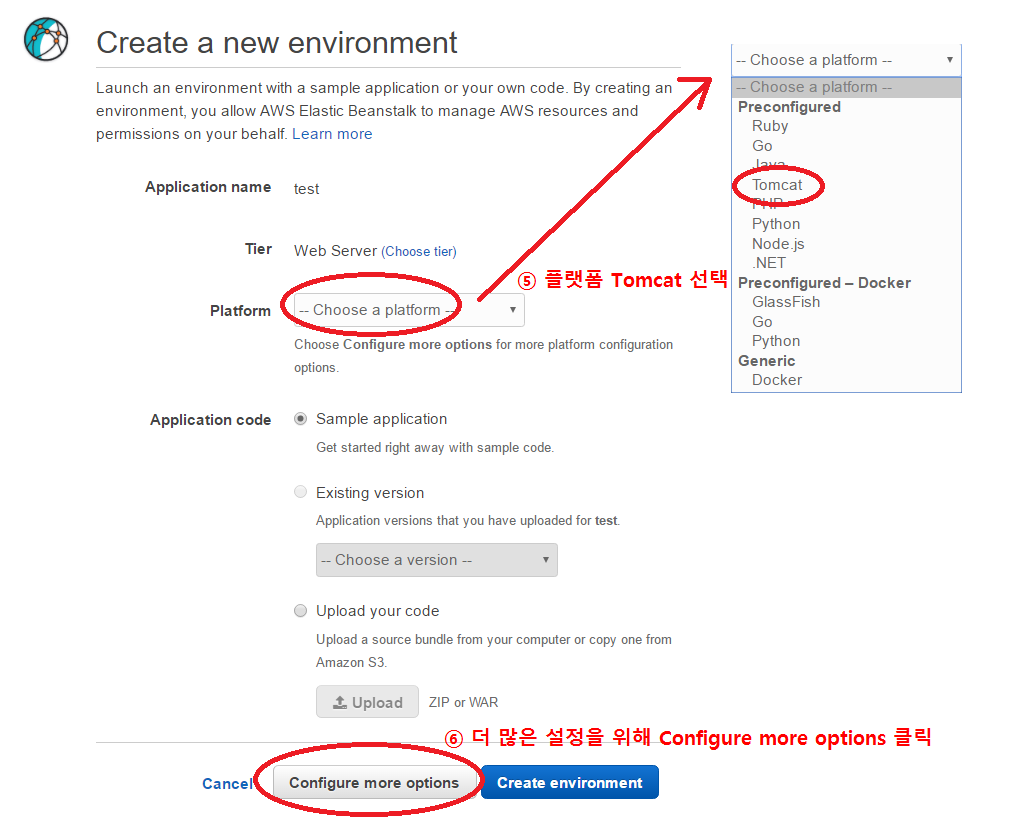
### Environment 생성



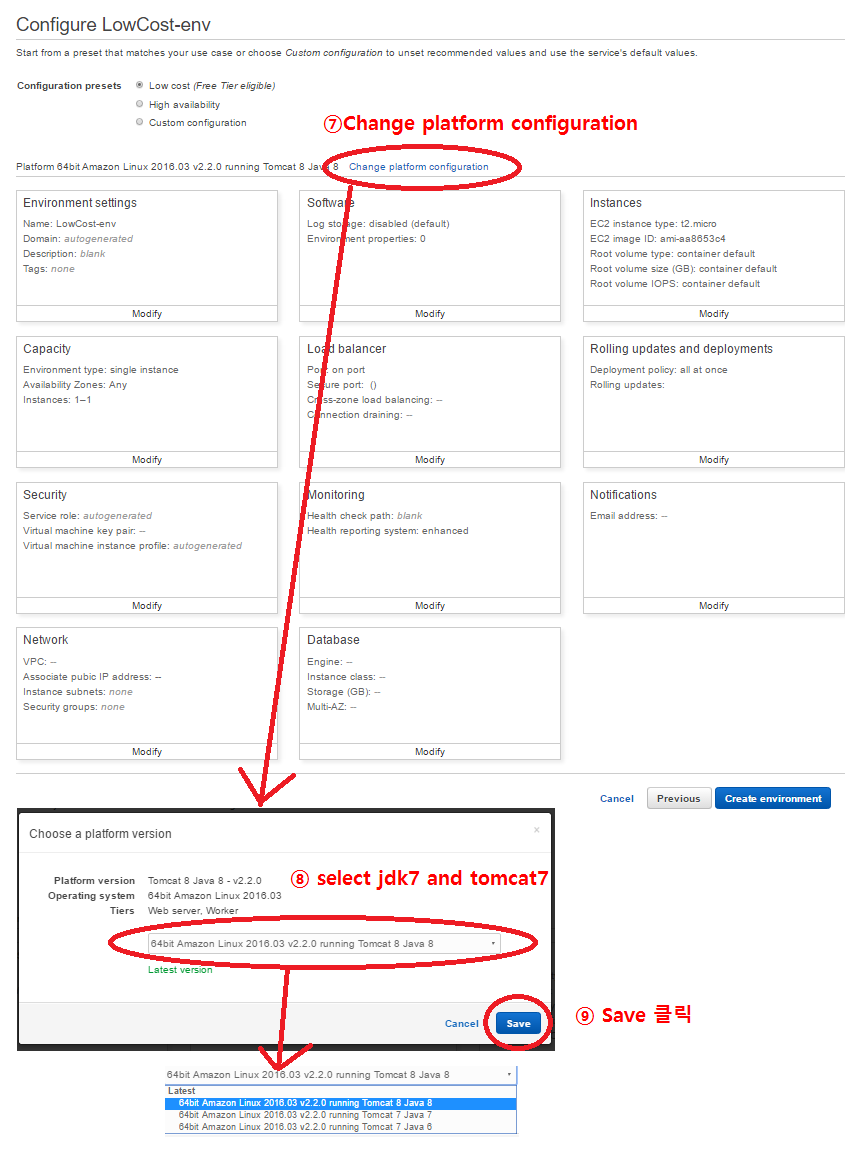
* + 1. Actions를 클릭해 메뉴를 확인한다.
    2. Create New Environment를 생성한다.
       1. (Environment와 배포되는 로드밸런서는 1:1 매칭관계)



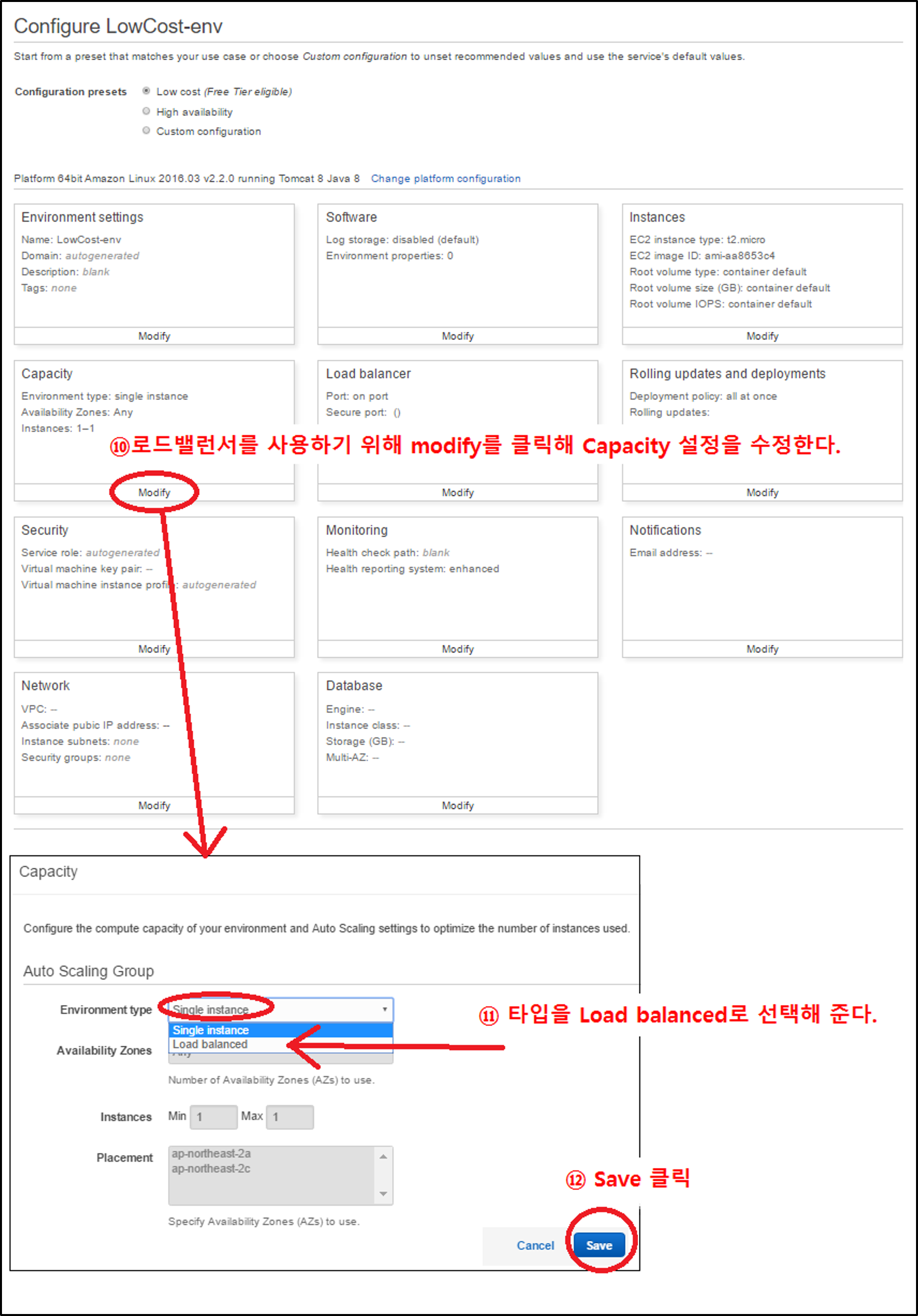
* + 1. Web server environment를 선택한다.
    2. Select 클릭한다.



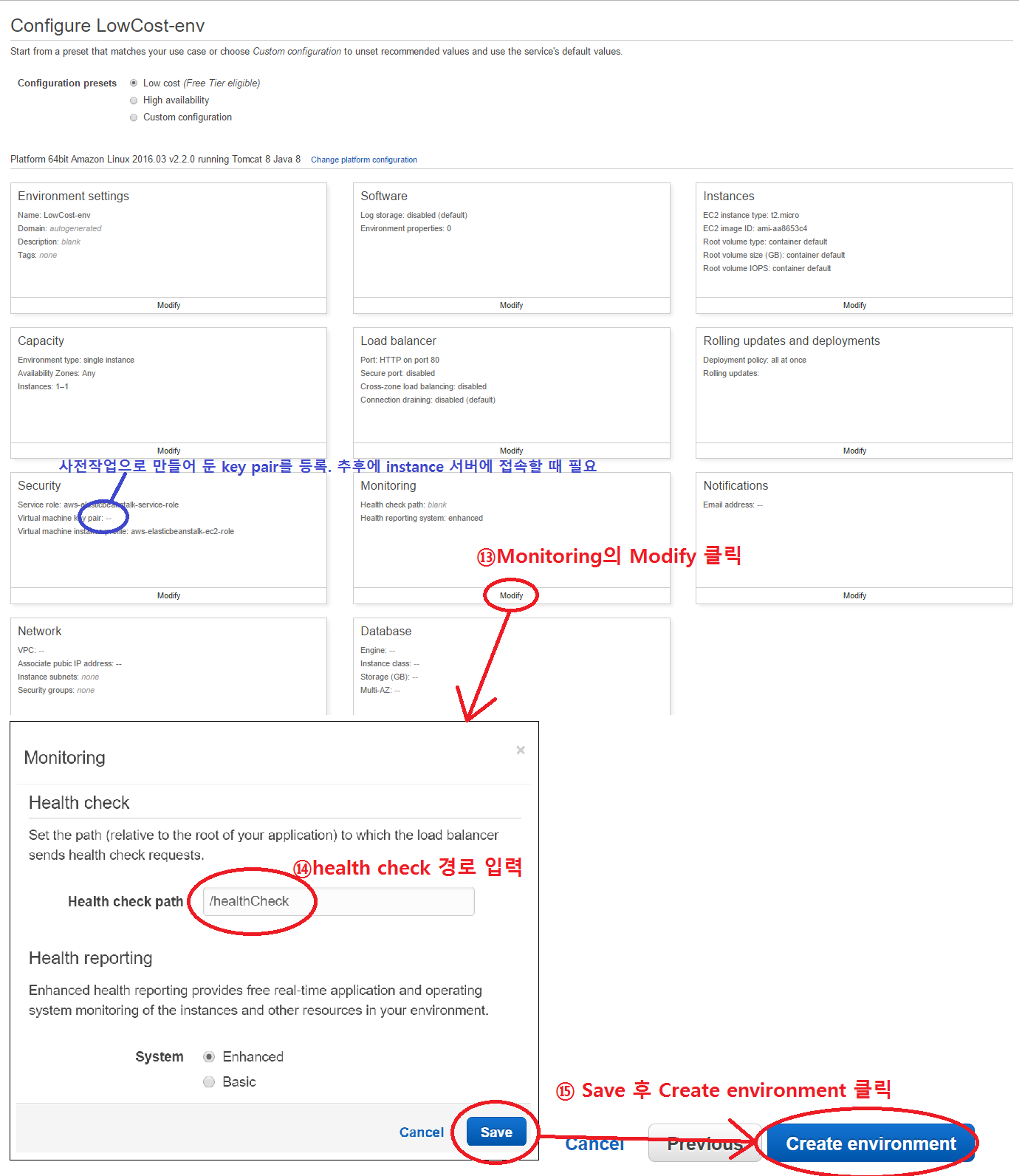
* + 1. 스프링 프레임워크의 플랫폼인 Tomcat을 선택한다.
    2. Tomcat7, Jdk7 선택을 위해 Configuration more options를 클릭한다.



* + 1. Change platform configuration 클릭한다.
    2. 플랫폼 버전을 jdk7과 tomcat7을 골라 선택한다.
    3. Save 클릭하여 변경 내용을 저장한다.



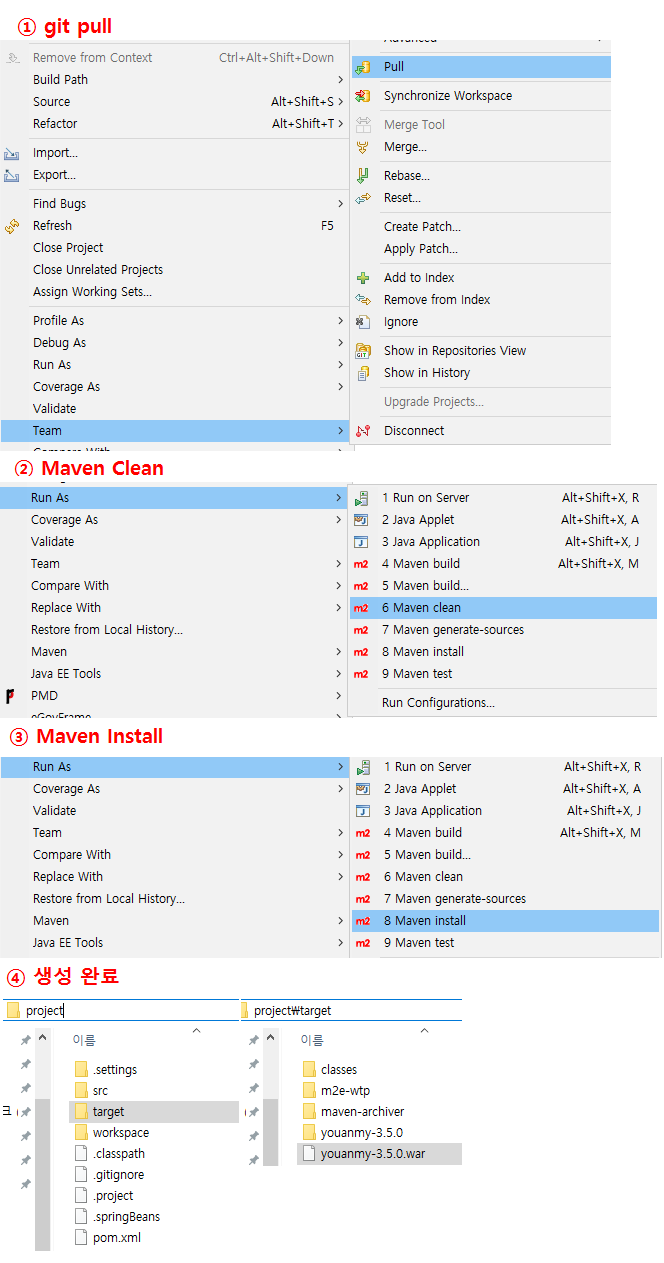
* + 1. 로드 밸런서를 사용하기 위해 capacity의 modify를 클릭한다.
    2. Environment type을 Load balanced로 선택한다.
       1. (참고 : <http://docs.aws.amazon.com/ko_kr/elasticbeanstalk/latest/dg/environments-create-wizard.html>)
    3. Save를 클릭해 저장한다.



* + 1. Health check 설정을 해주기 위해 Monitoring의 modify를 클릭한다.
    2. Health check 전용 경로를 입력한다. 조건은 200 code가 뜨는 get 형식의 http 요청이다. (https 불가) 나중에 https 설정 시 아파치 설정 파일에 이 경로를 입력해야 하므로 기억해 둬야 한다. health부분은 생략하고 프로젝트 배포하고 나서 설정해도 무방하다. 대신 컨트롤러에서 저 url을 매핑시켜줘야한다.
    3. 설정을 저장하고 Create environment를 클릭해 Beanstalk를 생성한다.

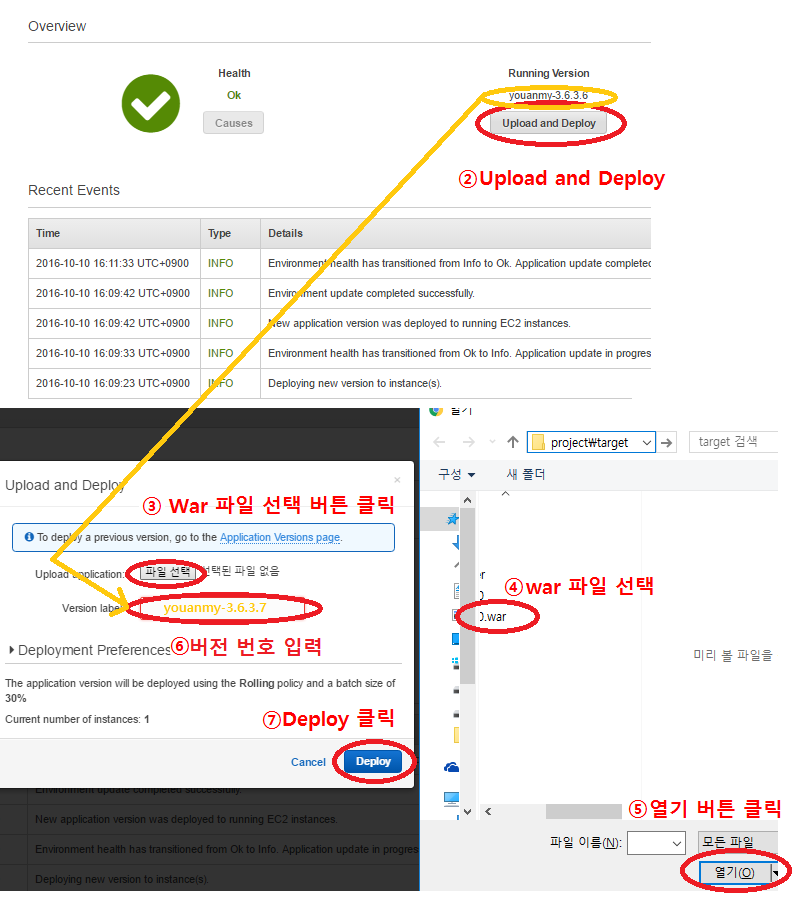
## Beasntalk 배포

### War 파일 생성



* + 1. Git으로부터 최신의 버전을 Pull 받는다.(필요에 의해 생략 가능)
    2. Maven Clean으로 기존의 Target폴더의 War파일을 지워준다.
    3. Maven Install으로 기존의 Target 폴더 안에 새로운 War파일을 생성해준다.
    4. 프로젝트 최 상단 위치의 target폴더 안에 war 파일이 생성 되어있음을 확인한다.

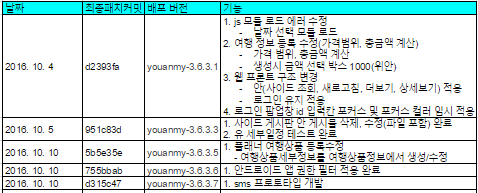
### Beanstalk 에 War 배포

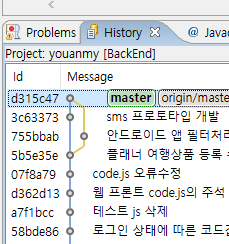
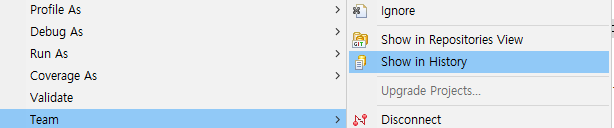


* + 1. [2. 가) Beanstalk 페이지 진입](#_Beanstalk_페이지_진입)을 참고하여 Beanstalk 페이지로 진입한다.
    2. Upload and Deploy를 클릭한다.
    3. 파일을 고르기 위해 파일선택 버튼 클릭한다.
    4. war파일을 선택한다.
    5. 열기를 눌러 선택을 완료한다.
    6. 차례대로 버전 숫자를 지정해 이름을 입력해준다.
    7. Deploy 클릭하여 배포를 완료한다.

### 배포 기록하기

* + 1. 개발문서의 3번째 탭인 서버 패치사항으로 이동한다.



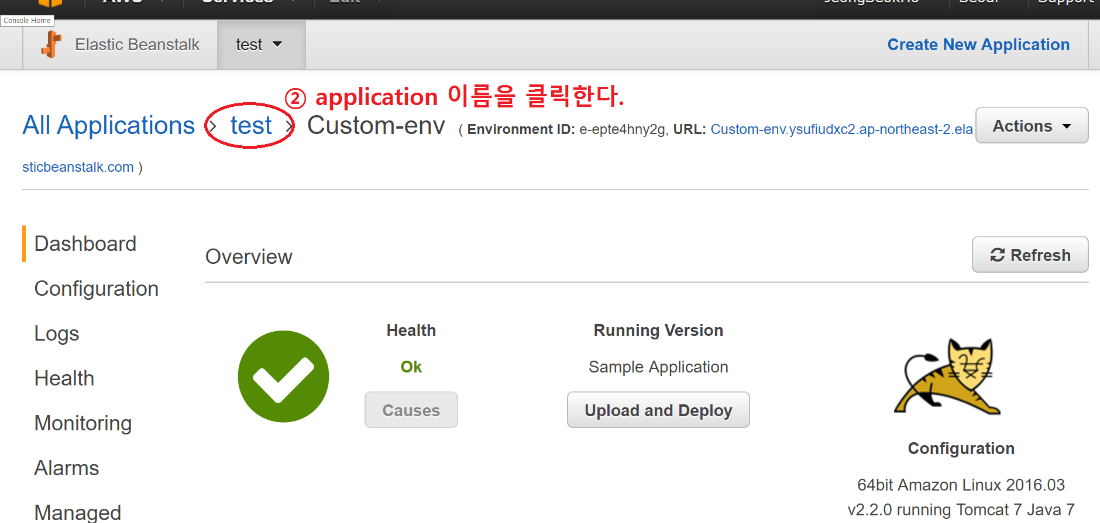


* + 1. 날짜와 git에서 pull받은 후의 가장 최근 커밋 번호(project에서 우클릭 후 Team-> Sho in Hisrosy를 클릭 한 후 가장 오른쪽 값이 커밋 번호)와 배포버전을 쓰고 커밋된 내용의 comment를 보고 패치된 기능을 작성한다.

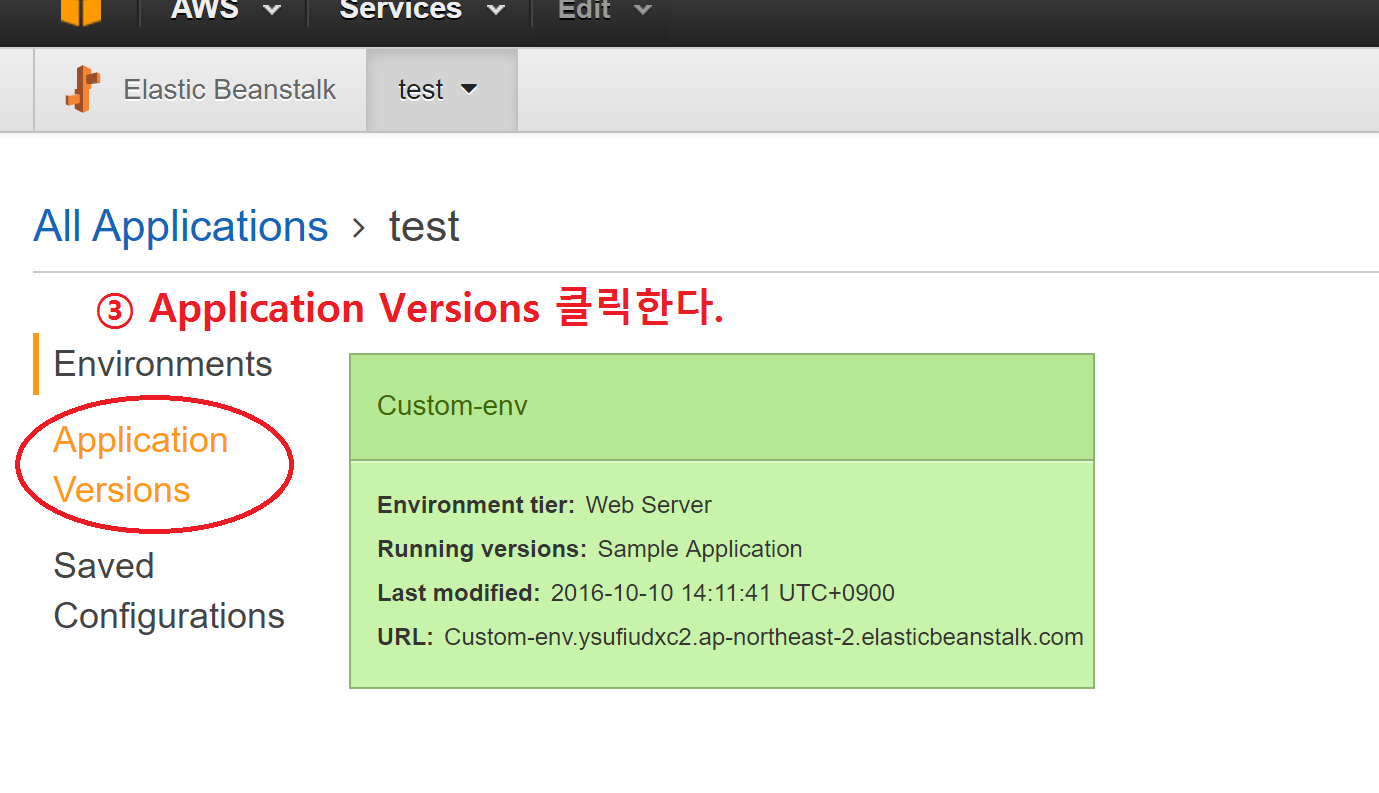
## Beanstalk 버전 되돌리기

### Beanstalk 버전 되돌리기

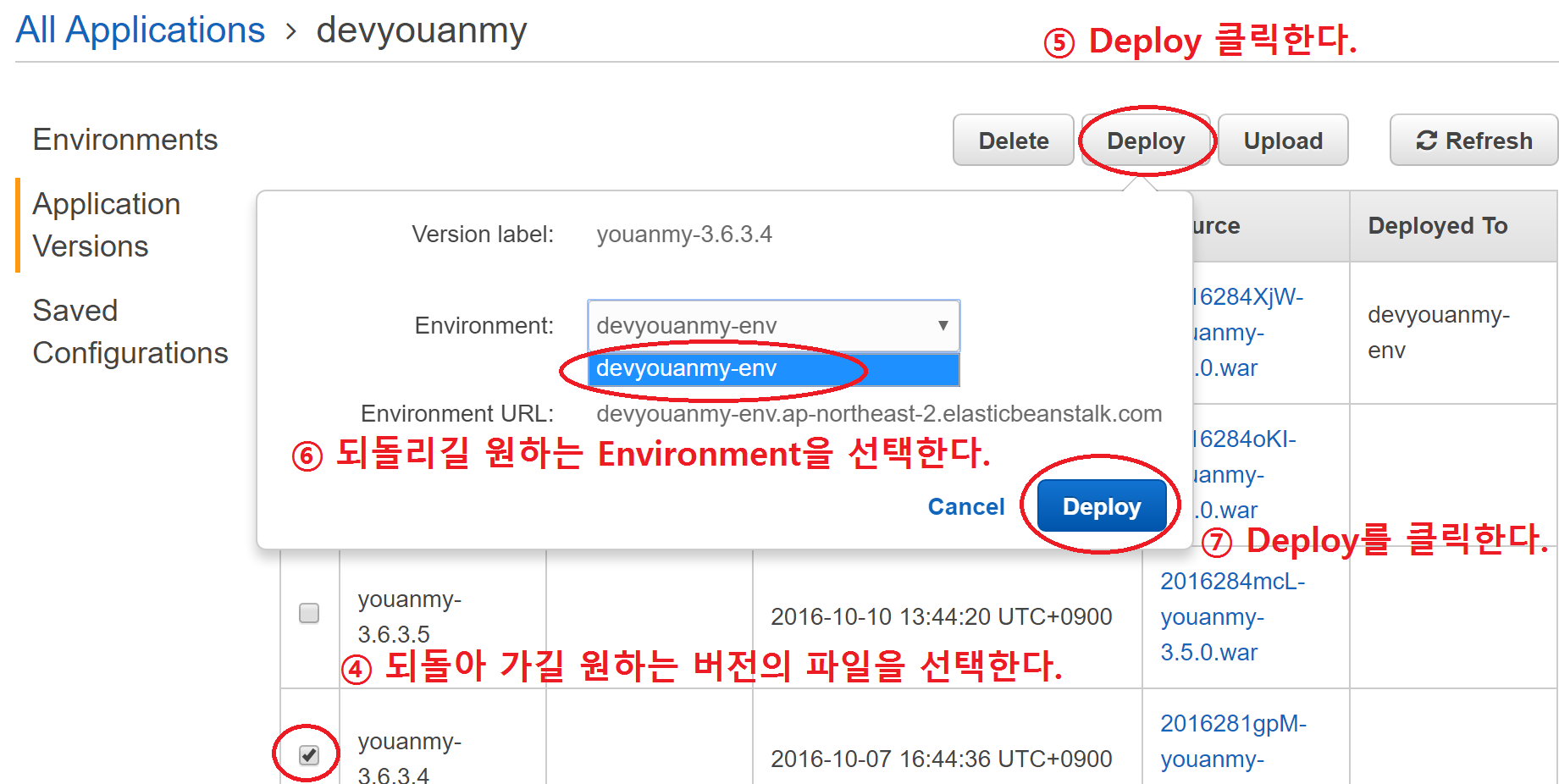
* + 1. [2. 가) Beanstalk 페이지 진입](#_Beanstalk_페이지_진입)을 참고하여 Beanstalk 페이지로 진입한다.



* + 1. Application 이름을 클릭하여 application 페이지로 이동한다.



* + 1. Application Versions를 클릭하여 배포된 파일 리스트 페이지로 이동한다.

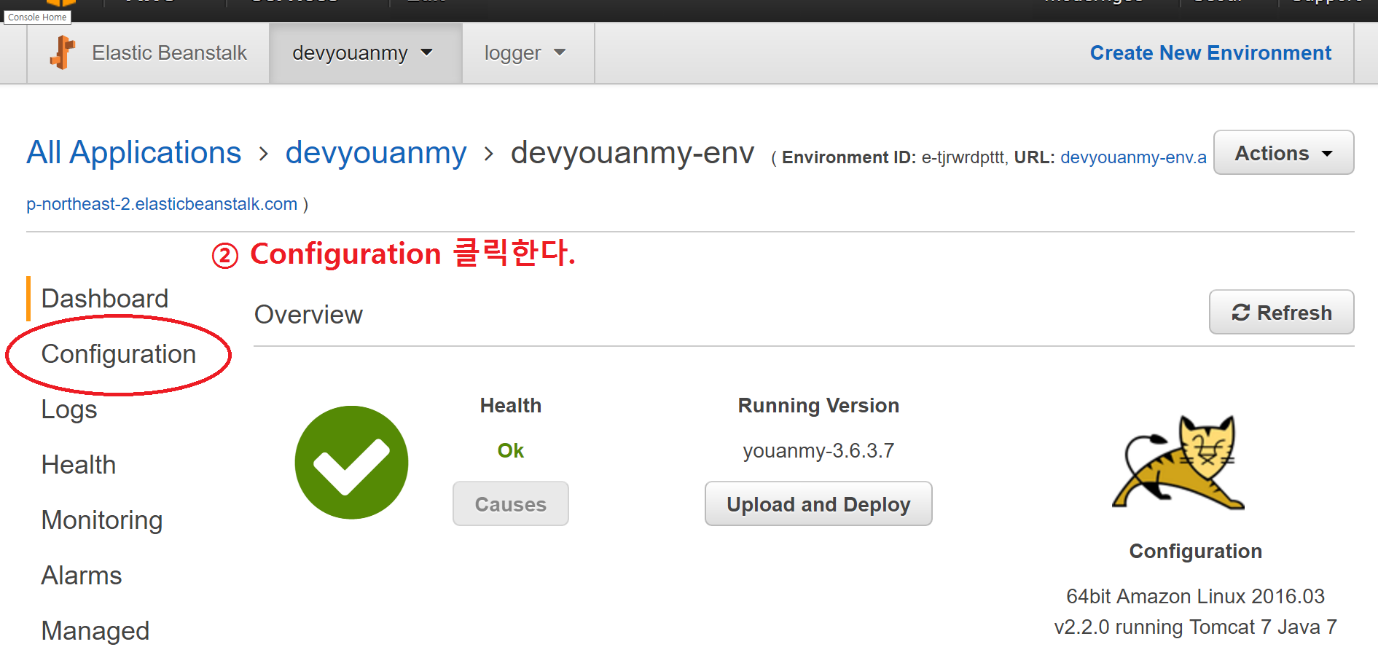


* + 1. 배포 파일을 선택한다.
    2. 회색의 Deploy버튼을 클릭한다.
    3. 뜬 팝업창에서 Environment 선택해 준다.
    4. 파란색의 Deploy를 클릭해 재배포를 시작한다.

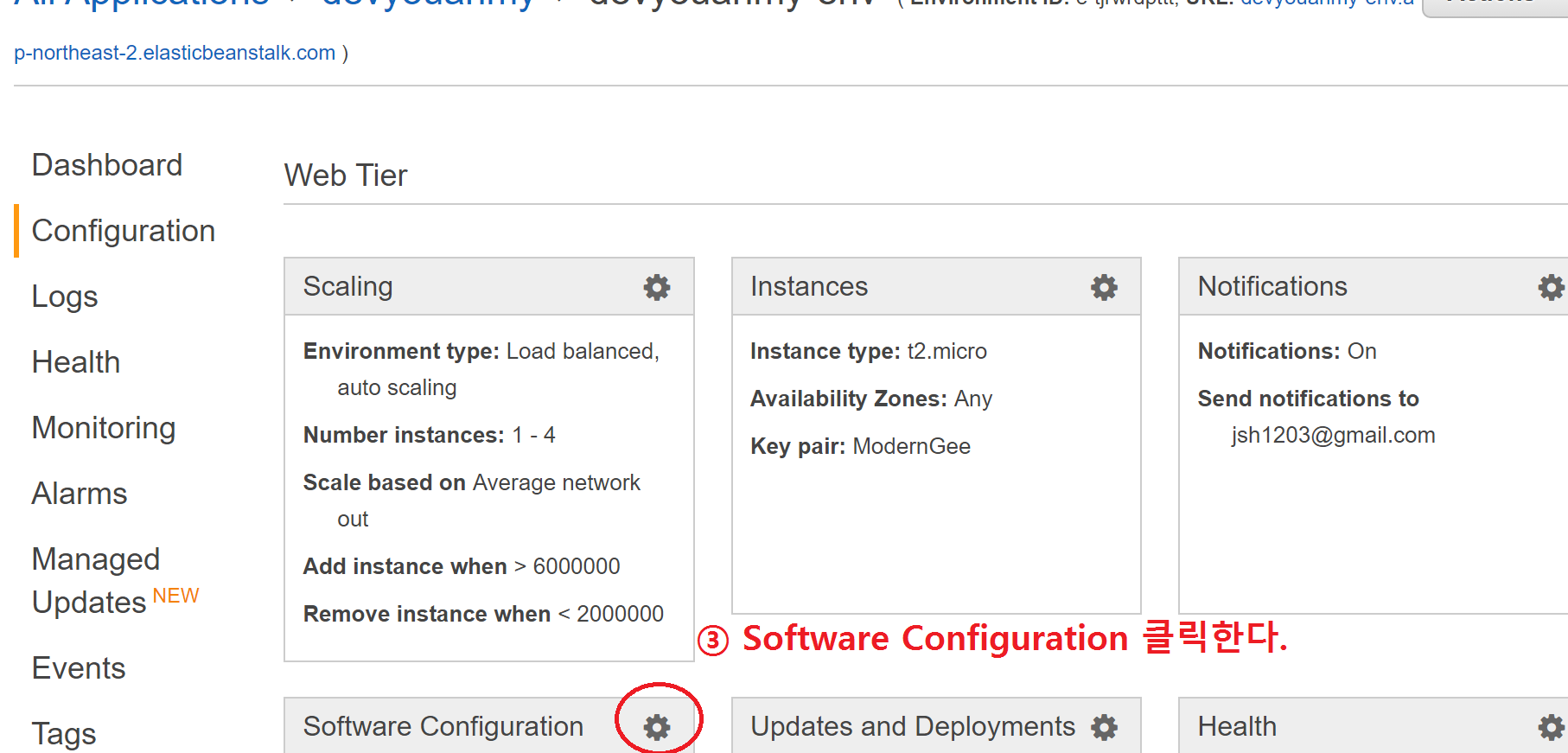
## Beanstalk TimeZone 설정

### JVM command line options 설정을 한다.

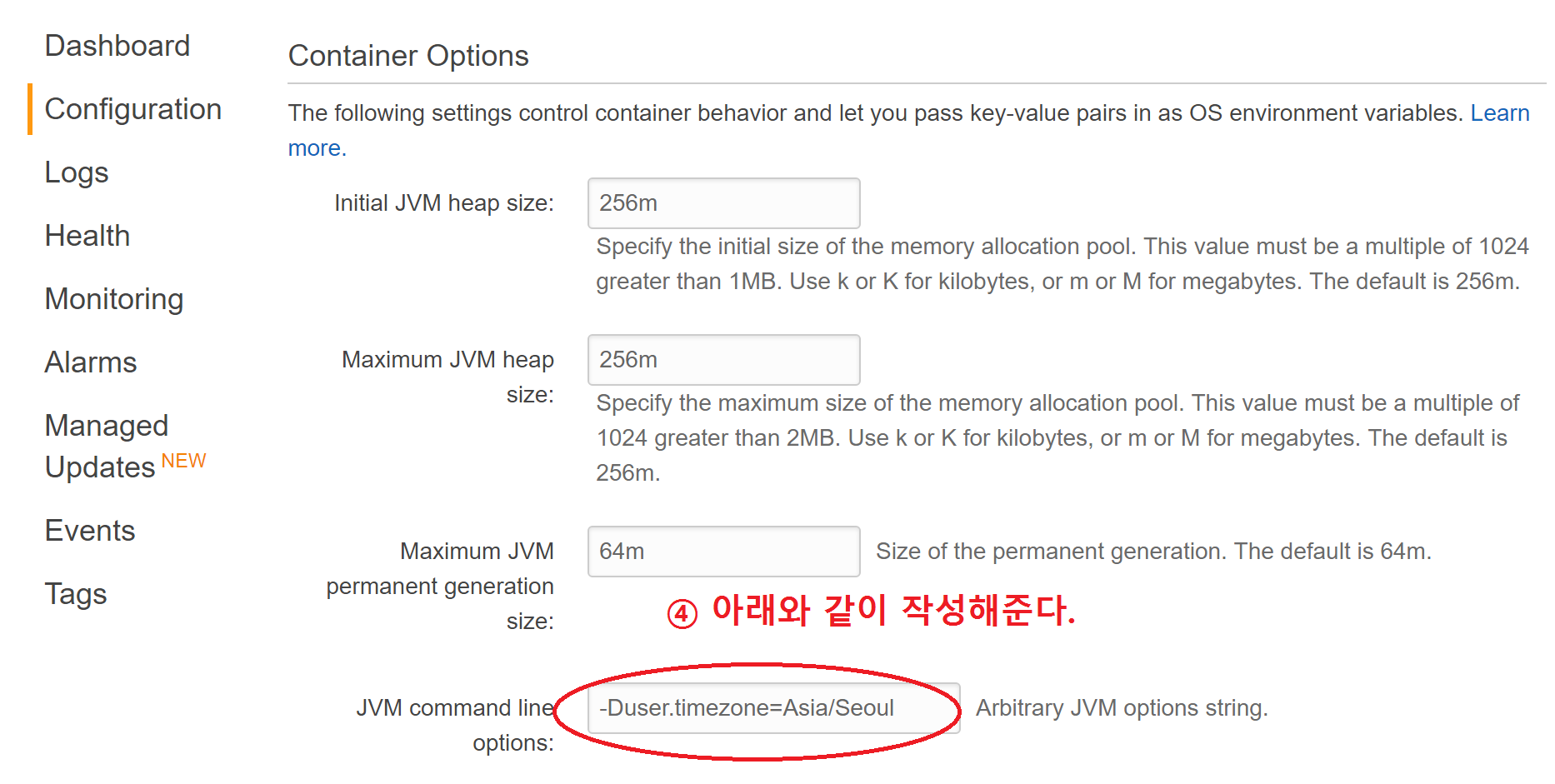
* + 1. [2. 가) Beanstalk 페이지 진입](#_Beanstalk_페이지_진입)을 참고하여 Beanstalk 페이지로 진입한다.



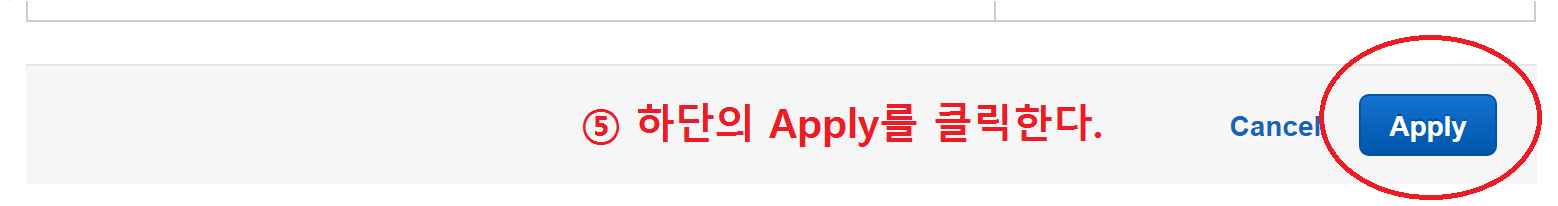
* + 1. Configuration을 클릭하여 설정 페이지로 이동한다.



* + 1. Software Configuration을 클릭하여 컨테이너 설정 페이지로 이동한다.



* + 1. JVM command line options에 “-Duser.timezone=Asia/Seoul” 라고 적는다.



* + 1. 적용시켜 주기 위해 Apply를 클릭한다.

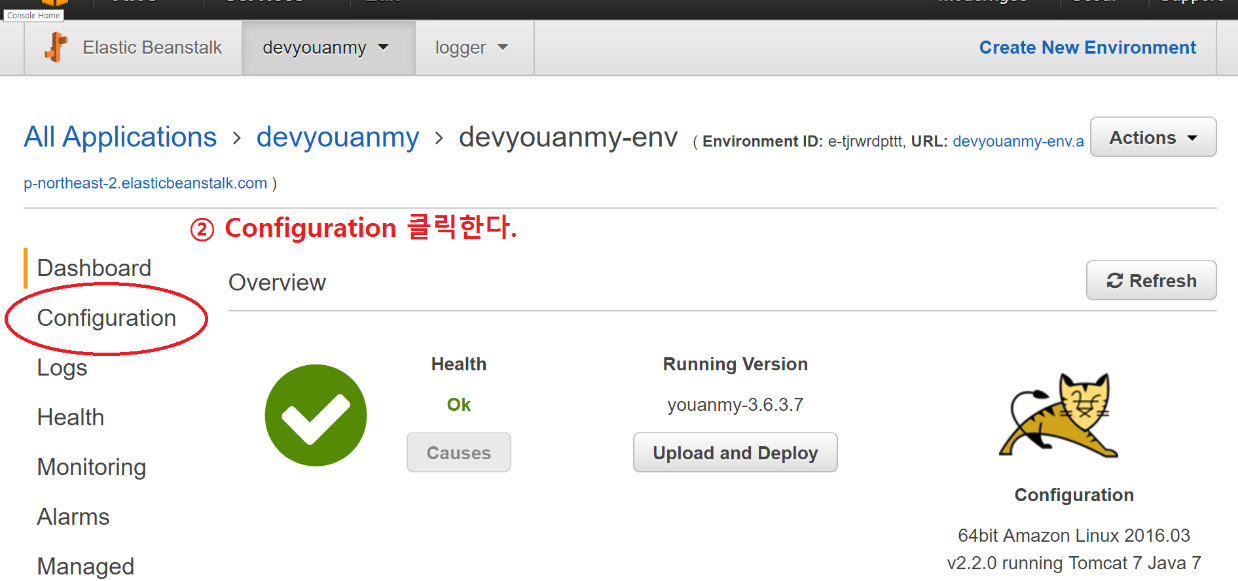
## Beanstalk Auto Scaling 설정

### Auto Sacling이란

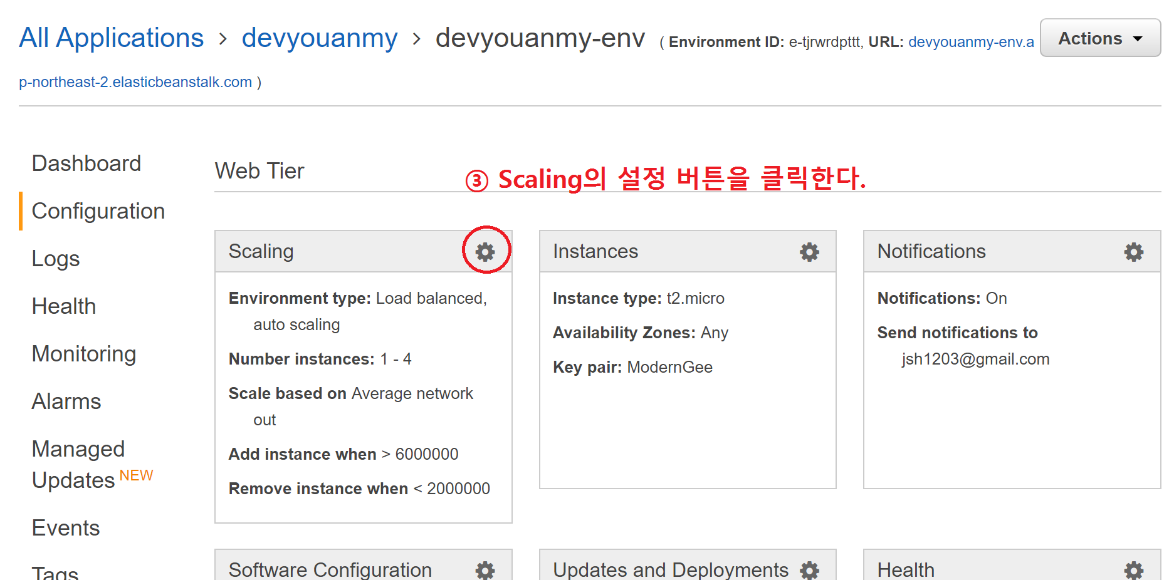
* + 1. 애플리케이션 가용성을 보다 간편하게 관리하게 해주며, Amazon EC2 용량이 사용자가 정의한 조건에 따라 자동으로 확장 또는 축소하게 해주는 아마존 웹 서비스의 일종이다.
    2. (참조 : <https://aws.amazon.com/ko/autoscaling/?nc2=h_m1> )

### Auto Scaling 설정하기

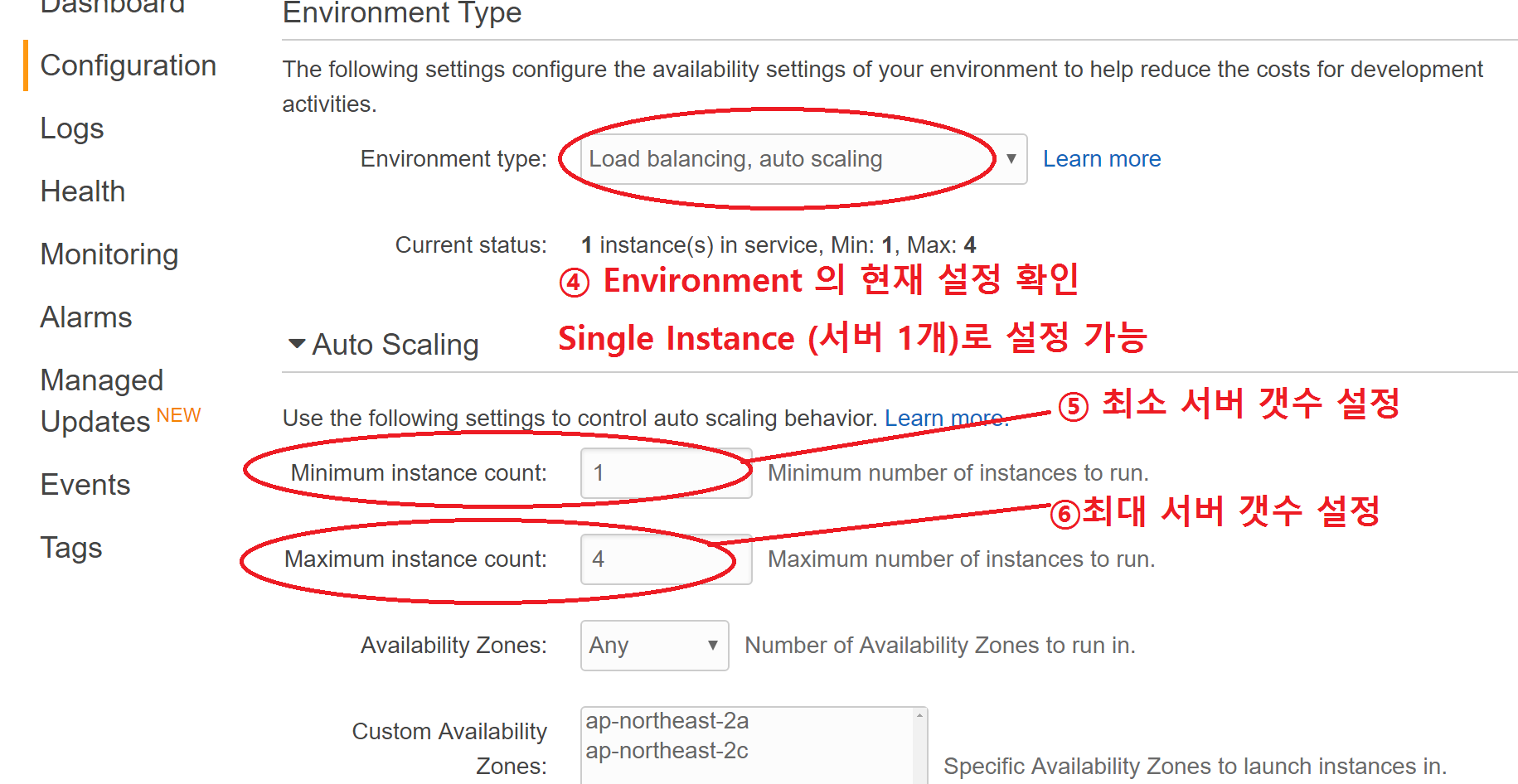
* + 1. [2. 가) Beanstalk 페이지 진입](#_Beanstalk_페이지_진입)를 참고하여 Beanstalk 페이지로 이동한다.



* + 1. Configuration을 클릭하여 설정 페이지로 이동한다.



* + 1. Scaling 설정 버튼을 클릭하여 auto scaling 설정 페이지로 이동한다.

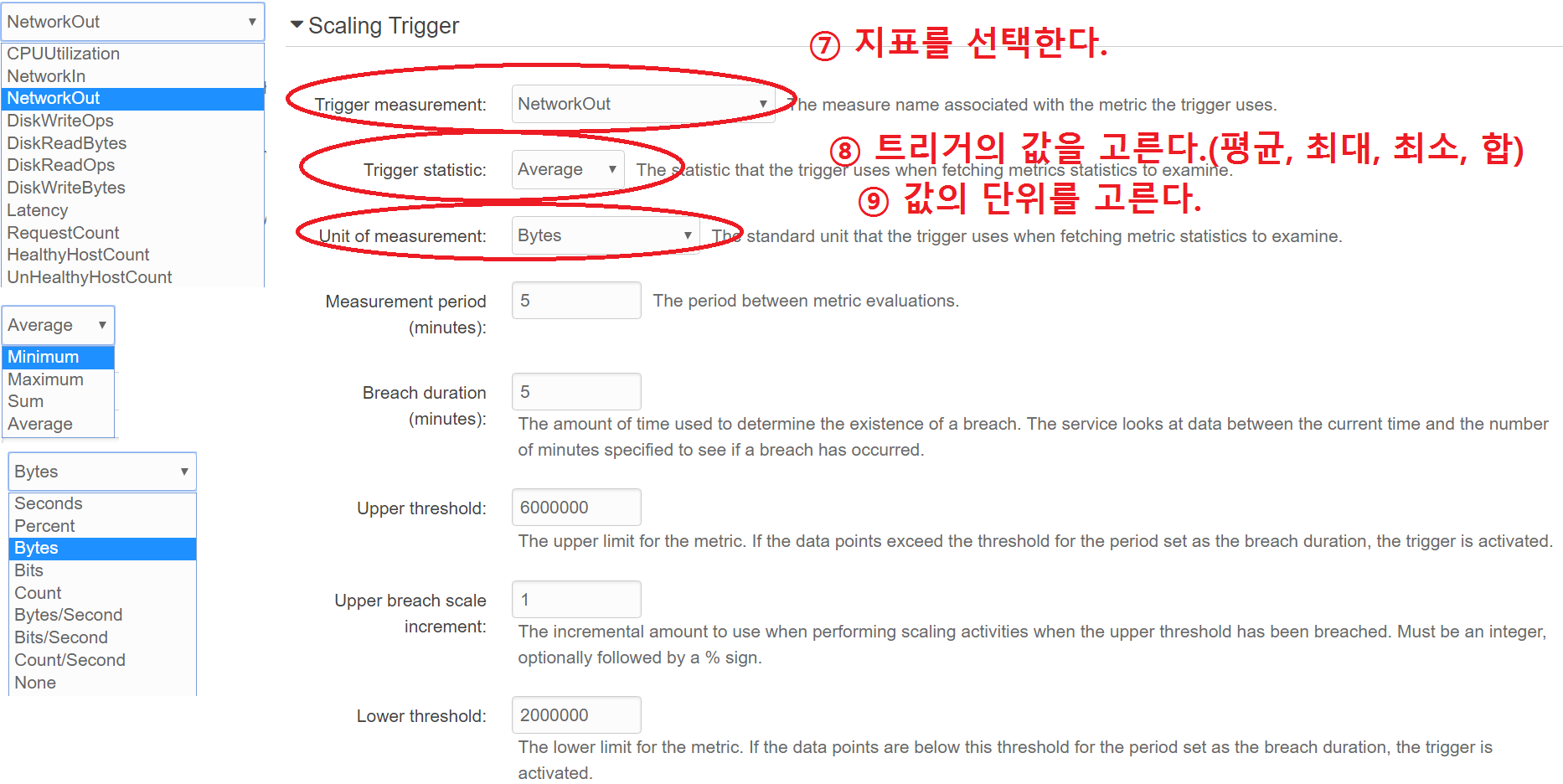


* + 1. Environment의 현재 설정을 확인 할 수 있다.

1 instance(s) in service, Min:1, Max:4 -> 1개의 서버가 서비스 중임을 알 수 있다.

Environment Type을 Single Instance로 바꿀 수도 있다. 대신 Auto Scaling과 https 설정을 바꿔야 한다. 현재 매뉴얼은 Load balancing을 기준으로 작성되어 있기 때문에 [AWS 온라인 설명서](https://aws.amazon.com/ko/documentation/)를 참고해야 한다.

* + 1. 최소 서버 개수를 설정할 수 있다.
    2. 최대 서버 개수를 설정할 수 있다.



* + 1. 지표를 선택한다. 여러 지표를 선택할 수 있다. CPU사용률, 네트워크IO, 디스크 읽기/쓰기 등등. 메모리는 없으나 cloud watch로 지표를 만들어 auto scaling할 수 있다. 이에 대한 내용은 [CloudWatch를 이용한 Auto Scaling](http://docs.aws.amazon.com/ko_kr/autoscaling/latest/userguide/as-instance-monitoring.html)을 참고.
    2. 트리거를 걸 값의 종류를 고를 수 있다. (평균값, 최댓값, 최솟값, 합)
    3. 값의 단위를 고를 수 있다.

### 그 외의 auto scaling에 대한 내용 : [AWS AUTO SCALING](http://docs.aws.amazon.com/ko_kr/autoscaling/latest/userguide/WhatIsAutoScaling.html)