## Deployments with AWS Elastic Beanstalk

### 

[**Deployments with AWS Elastic Beanstalk 1**](#_gjdgxs)

[Overview 2](#_t3s0tcfemct2)

[Prerequisites 2](#_30j0zll)

[Step 1: Create Elastic Beanstalk Application 3](#_bwyzwom30ek4)

[Step 2: Create Elastic Beanstalk Environment 3](#_hz4bnez6yfr7)

[Step 3: Verify the Deployment 5](#_ji9fumfc23hu)

[Step 4: Setup New Version for Deployment 6](#_wqowmsdkpfzo)

[Step 5: Scale Out 7](#_sld0mb8u5z3n)

[Step 6: Single-instance Environment 7](#_mtjrvz2ha9el)

[Questions 7](#_dugrckj530pi)

### 

### 

### Overview

This is an activity that introduces you to AWS Elasticbeanstalk. We would use a sample python code base as the application for deployment. We would explore how new versions of an application can be deployed to an environment, and the different configuration options on Elastic Beanstalk service.

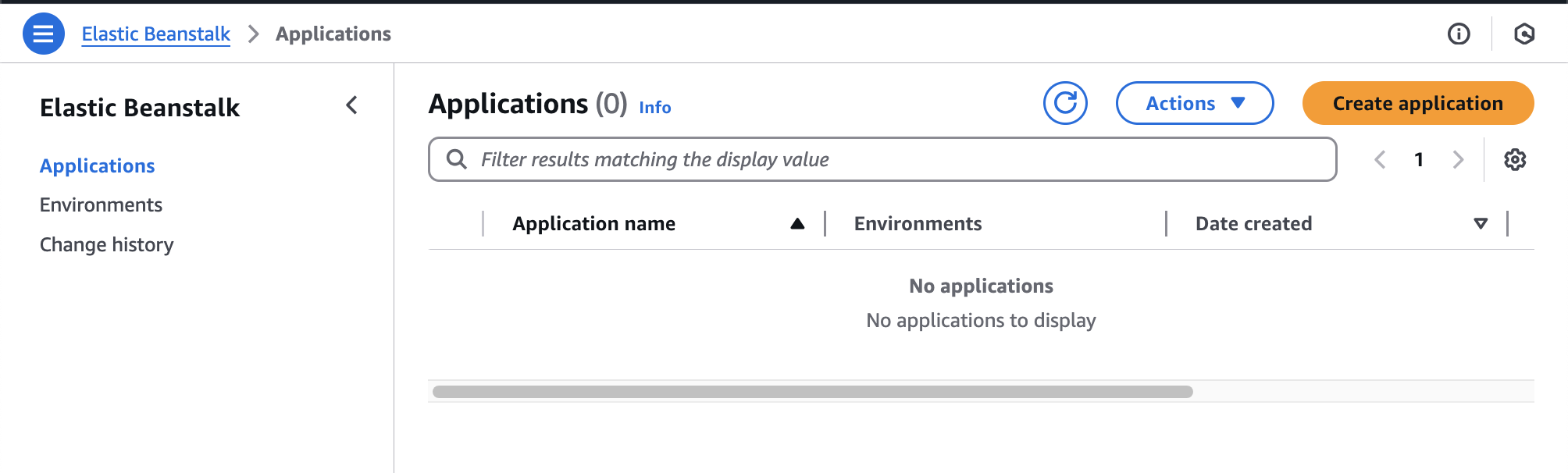
### Prerequisites

* **Existing VPC with public and private subnets** (<https://github.com/KeenGWatanabe/terraform-vpc.git>)
* Access to AWS Elastic Beanstalk service (aws console)
* Sample python package (zip file) from <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/GettingStarted.DeployApp.html>

### 

### Step 1: Create Elastic Beanstalk Application

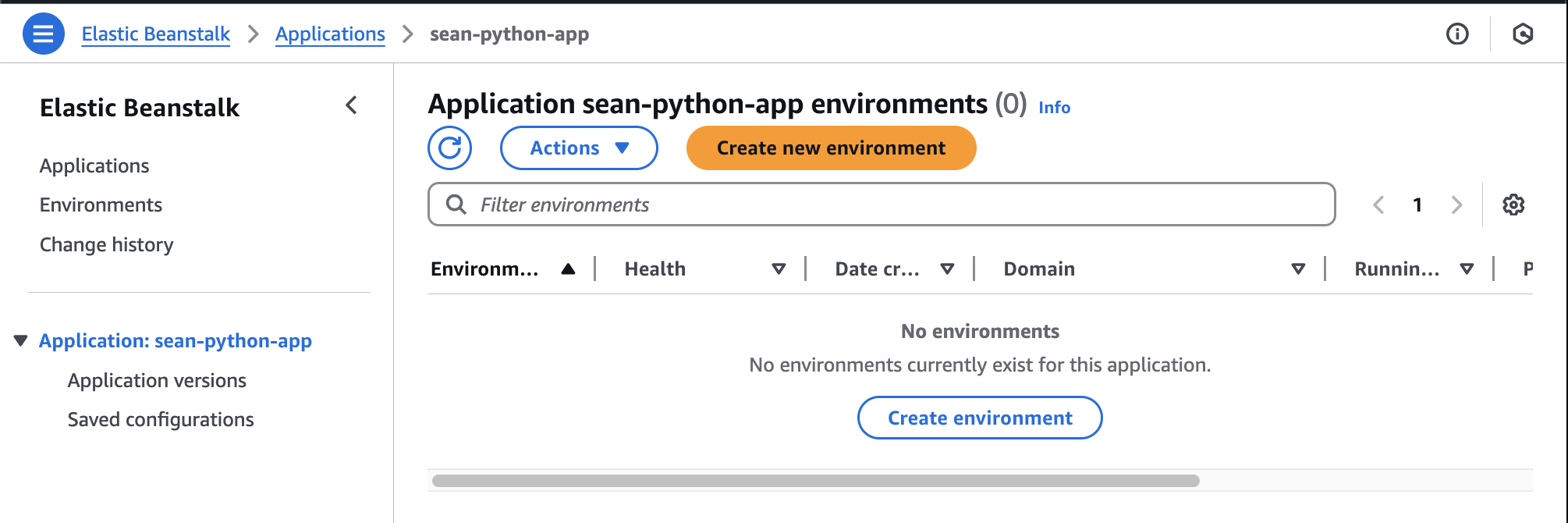
Navigate into Elastic Beanstalk via the console, expand the side menu to enter the applications page.



Create an application with the name <name>-python-app

### Step 2: Create Elastic Beanstalk Environment

Navigate into your application on Elastic Beanstalk to view the environments belonging to your application.



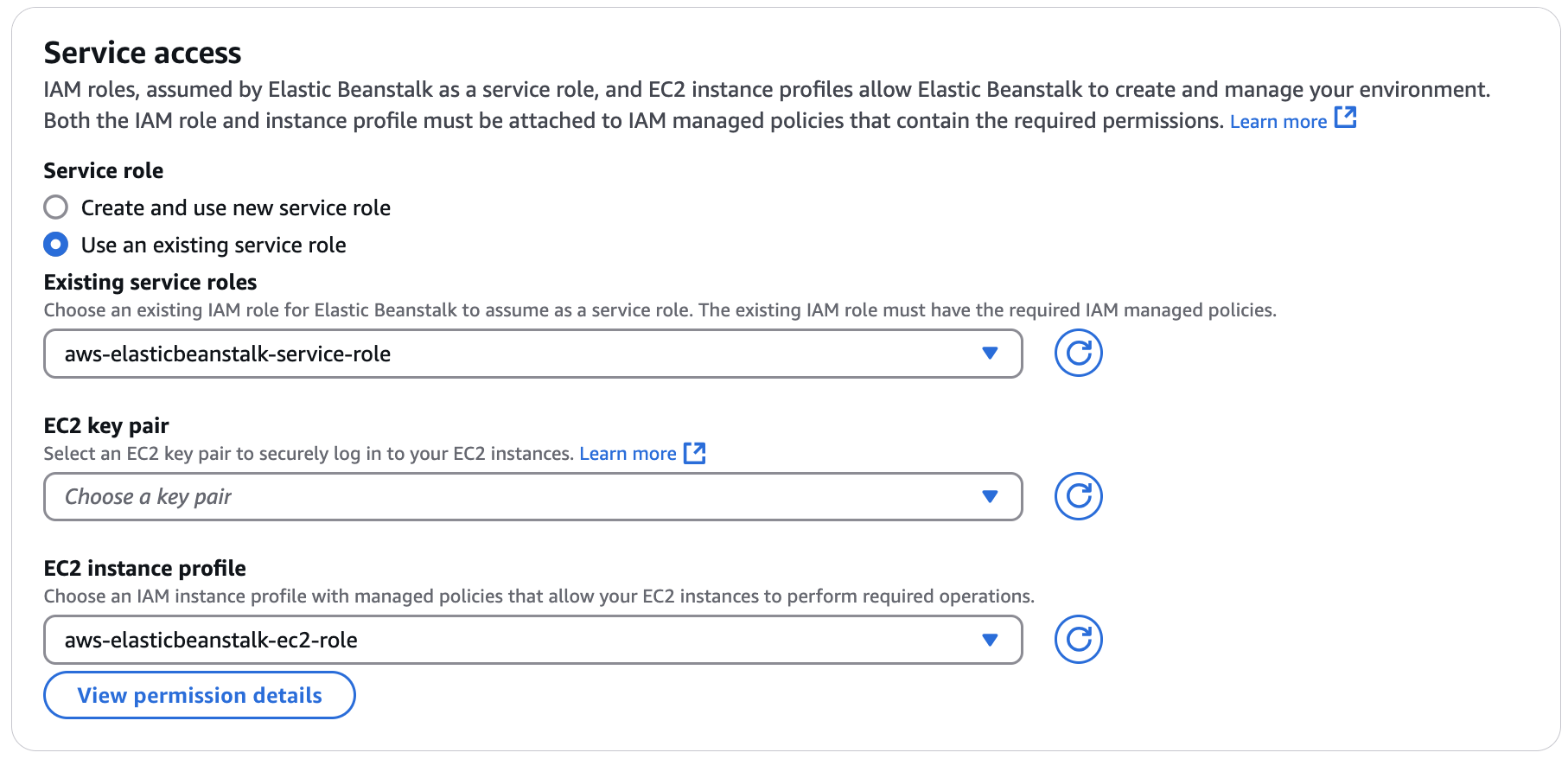
Create an environment.

Configure environments

* Environment tier: **Web server** environment
* Application name: <name>-python-app
* Environment name: <name>-development
* Platform
  1. Type: Managed platform
  2. Platform: Python
* Application code
  1. Upload your code
  2. Version label: 0.0.1
  3. Source code origin: Local file (Upload the sample python zip code)
* Presets: Single instance

Configure service access

Pick the default service roles and EC2 instance profile.



Networking, database, and tags

* VPC: (Pick one) ce9-learner
* Instance settings
  1. **Public IP address: uncheck**
  2. Subnet selection: (Private ones across the AZs)

**tf-ec2-alb (pte and pub subnets config)**

Configure instance traffic and scaling

Besides the following mentioned, leave the rest as default. Do have a look through on the different configuration options available.

Instances

EC2 security groups: Choose according to your vpc;

* Capacity
  1. Environment type: **Load balanced**
  2. Instances min: 1
  3. Instances max: 1
  4. Instance types: **t2.micro (remove the t3)**
* Load balancer network settings
  1. Visibility: **Public**
  2. **Subnet selection: (Public ones across the AZs) remove the pte ones**
* Load balancer type
  1. **Application load balancer**
  2. Dedicated

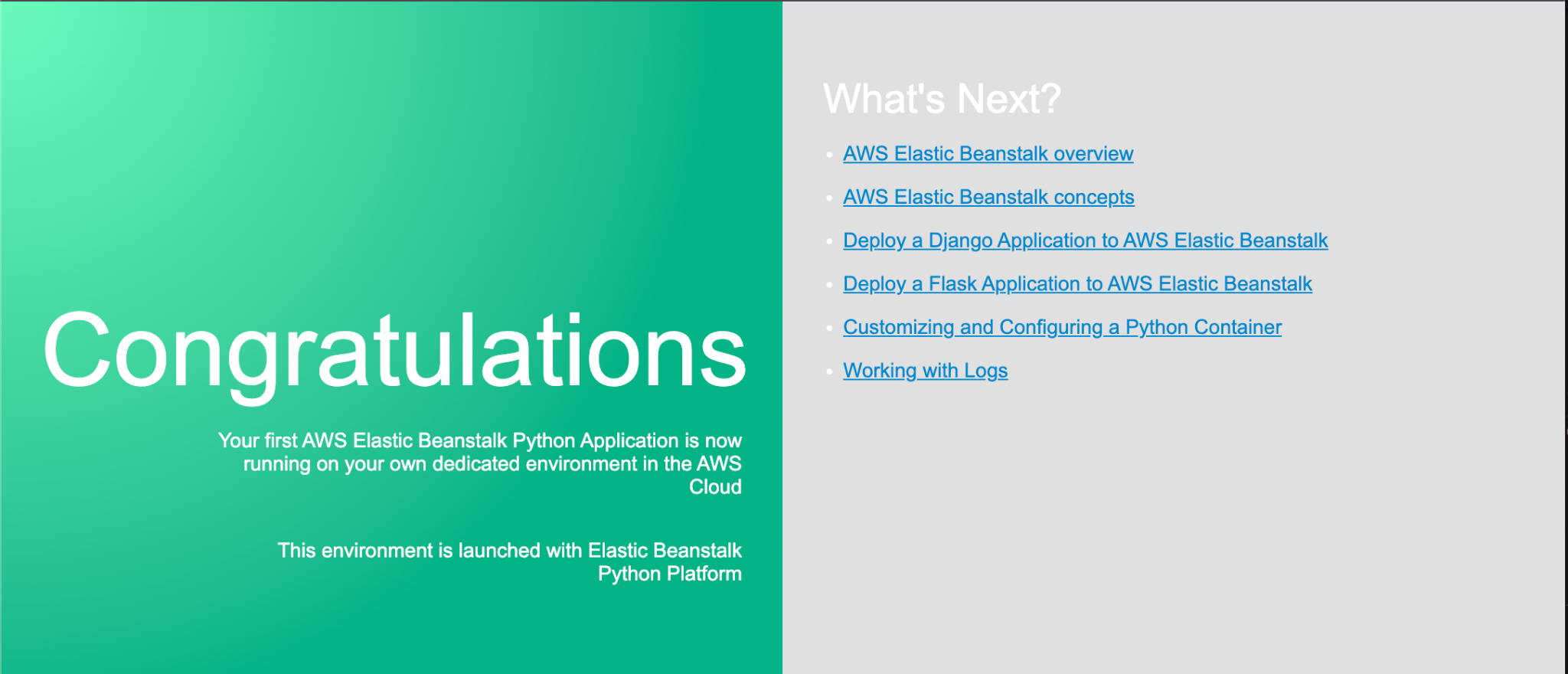
**tf-ec2-alb (application load balancer config)**

Configure updates, monitoring, and logging

Leave as default.  
Check Health status, a few minutes, depically >5min

### Step 3: Verify the Deployment

Once deployment is complete, visit the domain and you’d see this.



### 

### Step 4: Setup New Version for Deployment

Prepare new version

Unzip the application for editing in your IDE. Make the following edits.

Replace

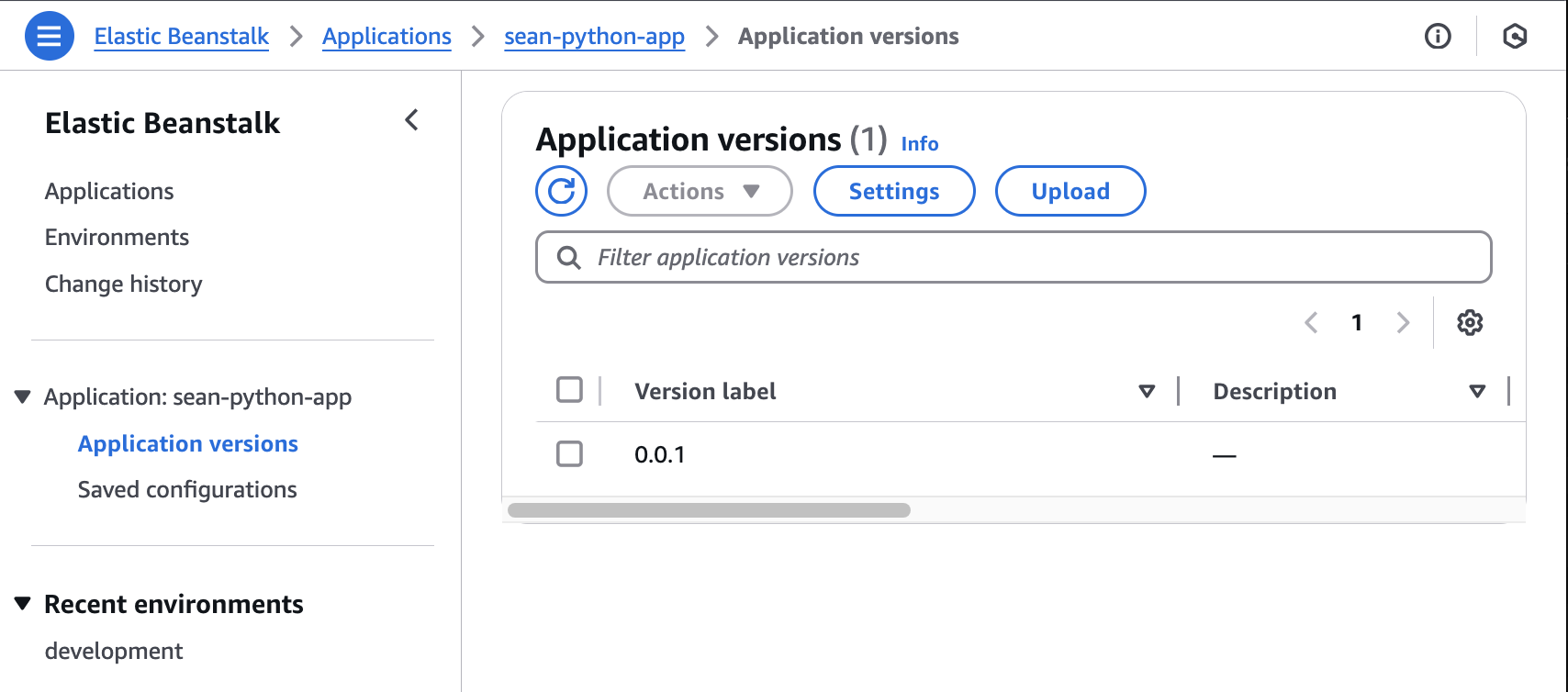
<h1>Congratulations</h1>

With

<h1>Sample app in Development Environment by MYNAME</h1>

Save the code and create a new zip file.

Upload new version



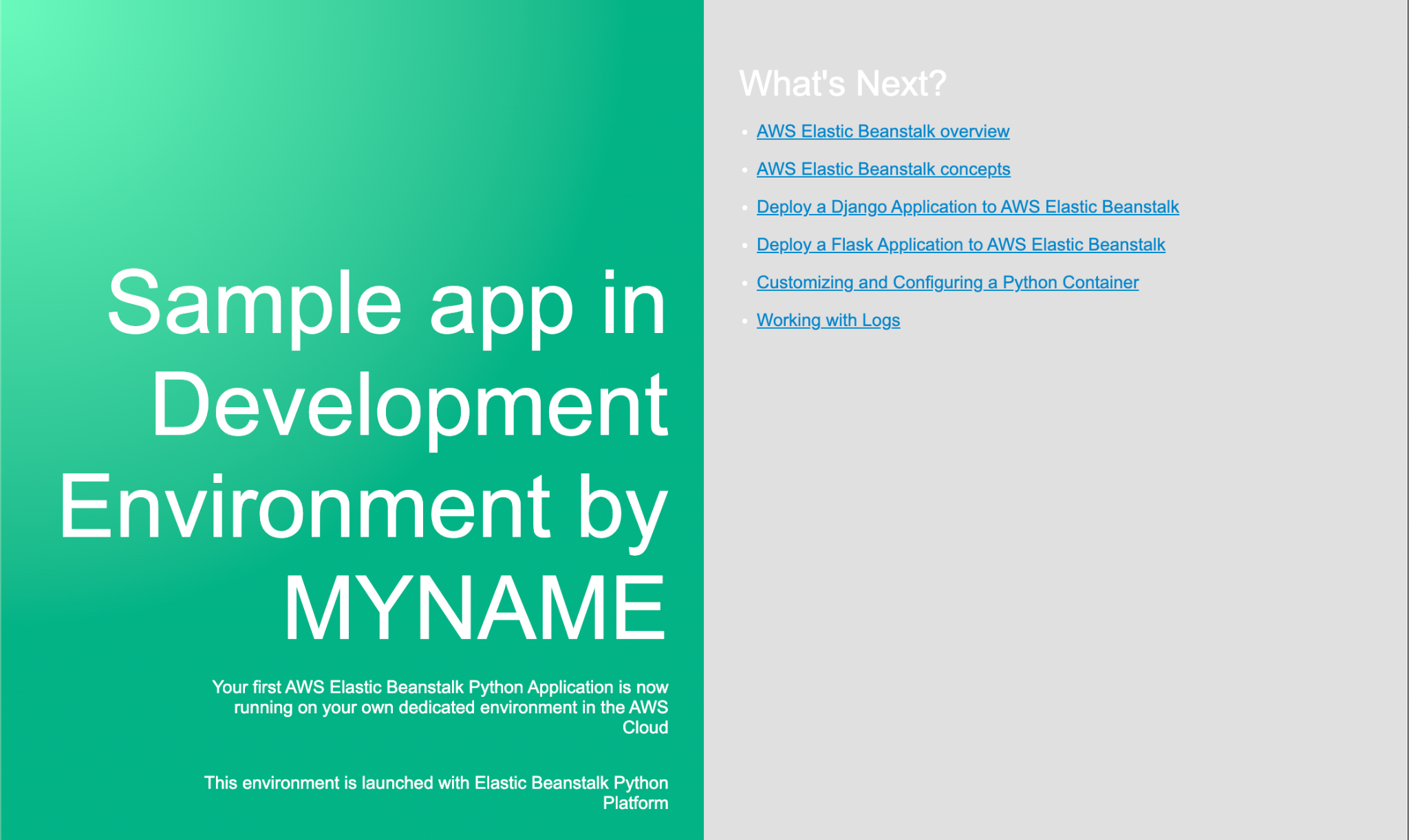
Navigate into your application versions page. Make a new upload with the version label of 0.0.2

Deploy to Development

Select your new version for deployment.

Verify

Check that it has deployed successfully.



If you are facing errors, you may use this guide for further help - <https://aws.amazon.com/getting-started/hands-on/update-an-app/>

### Step 5: Scale Out

Through the Elastic Beanstalk configurations, raise the minimum and maximum instance count to 2.

### Step 6: Single-instance Environment

Let’s save some cost by making the necessary configurations so the application is deployed without a load balancer. The website should still be accessible through the public internet.

### Questions

* How would it look like managing an application across multiple environments in Elastic Beanstalk?
* What is the purpose of the Service Role attached to Elastic Beanstalk? How does it differ from EC2 Instance Profile?
* Can Elastic Beanstalk work with RDS? Does it manage the RDS as well?