In this tutorial, you do the following:

Create an Elastic Beanstalk application using custom python code from AWS and make changes to the application code as well.

We will add more things as we go on in this tutorial too.

[What is Elastic Beanstalk?](#_plxgvfdg6qo0)

[Create an Elastic Beanstalk application](#_b88apdpip744)

### What is Elastic Beanstalk?

Elastic Beanstalk is an AWS service that makes it easy to deploy and manage web applications. It provides a platform to deploy and run applications in multiple programming languages, such as Java, .NET, PHP, Node.js, Python, Ruby, and Go, and provides preconfigured platforms to simplify the setup and configuration of your applications.

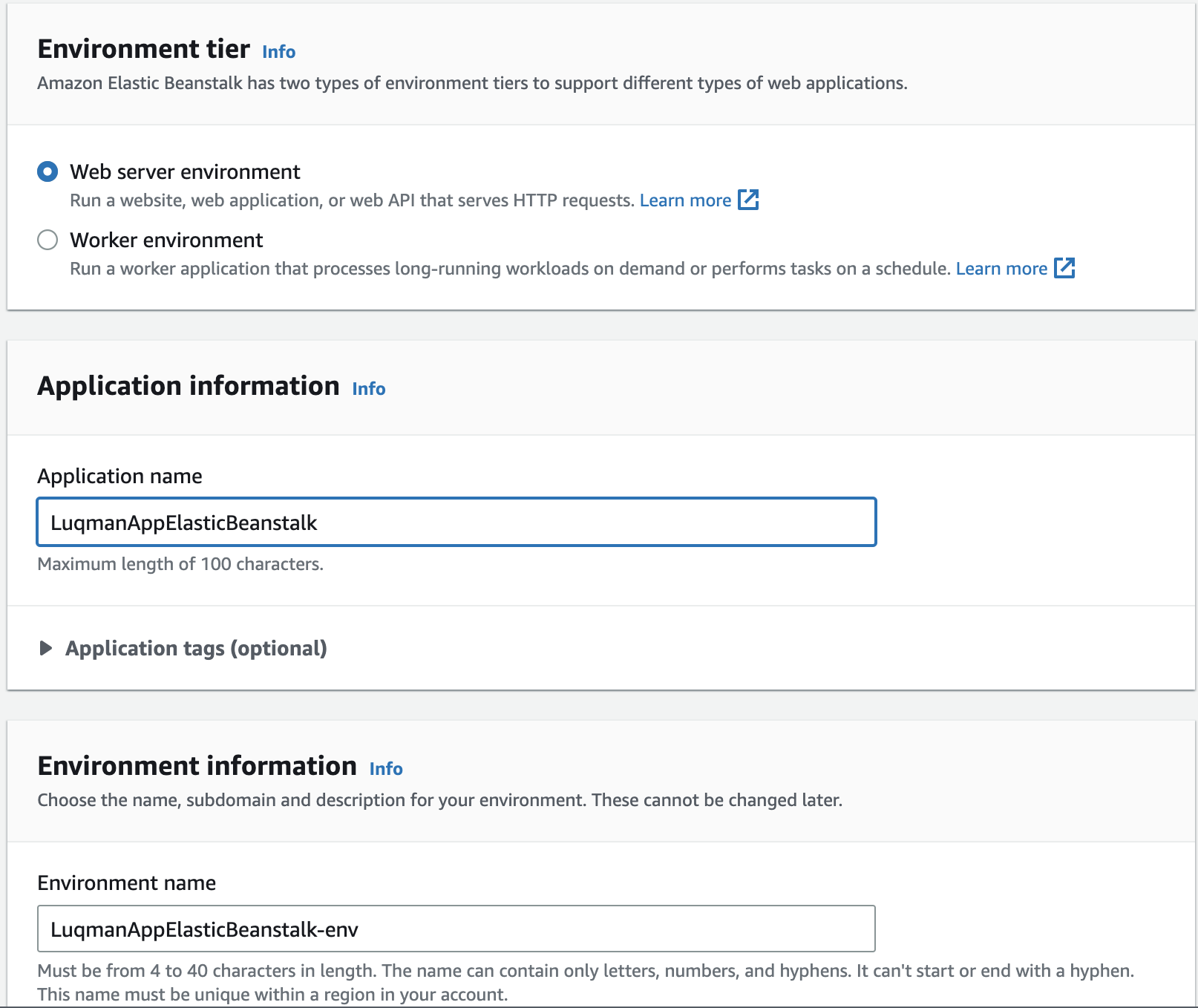
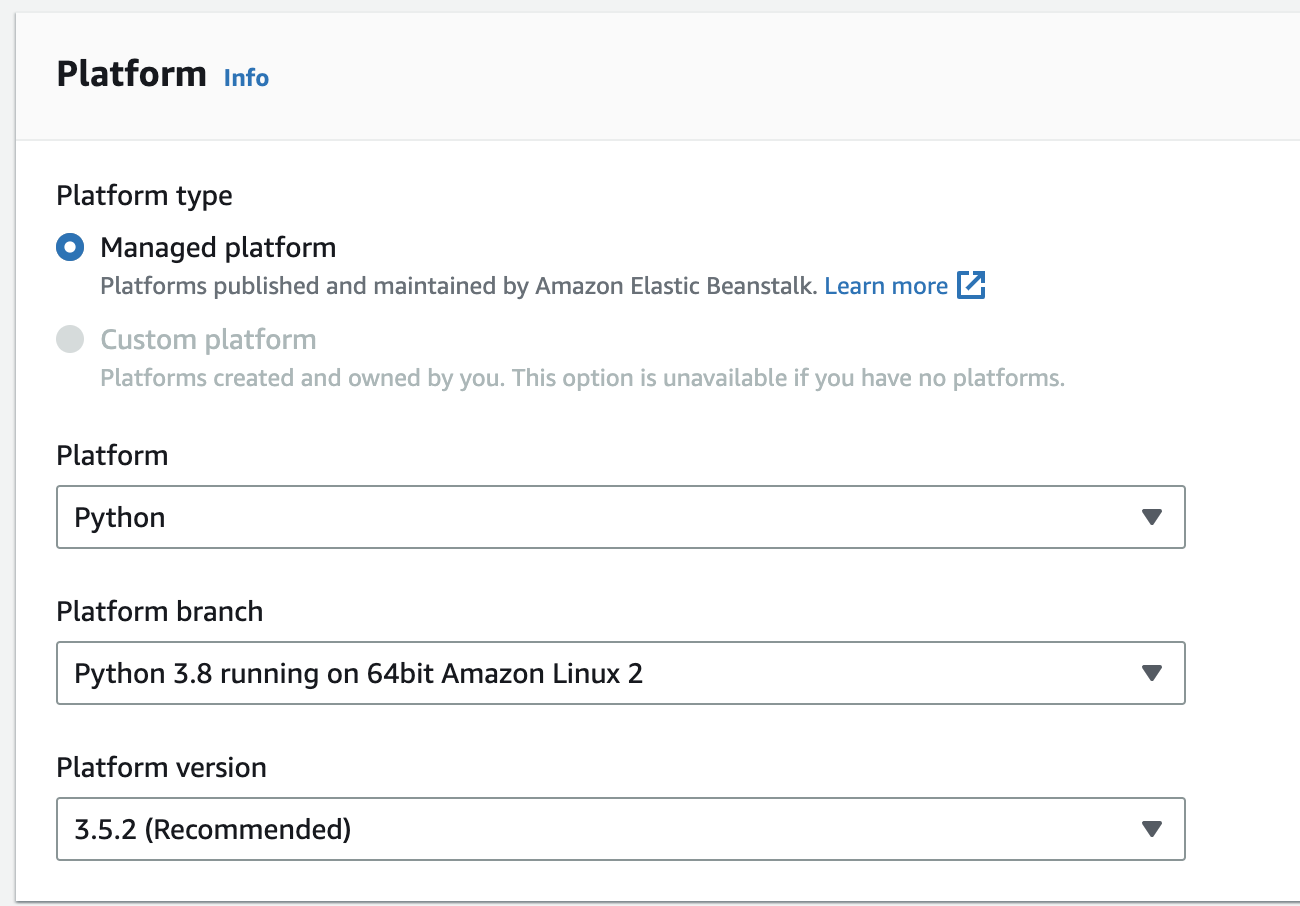
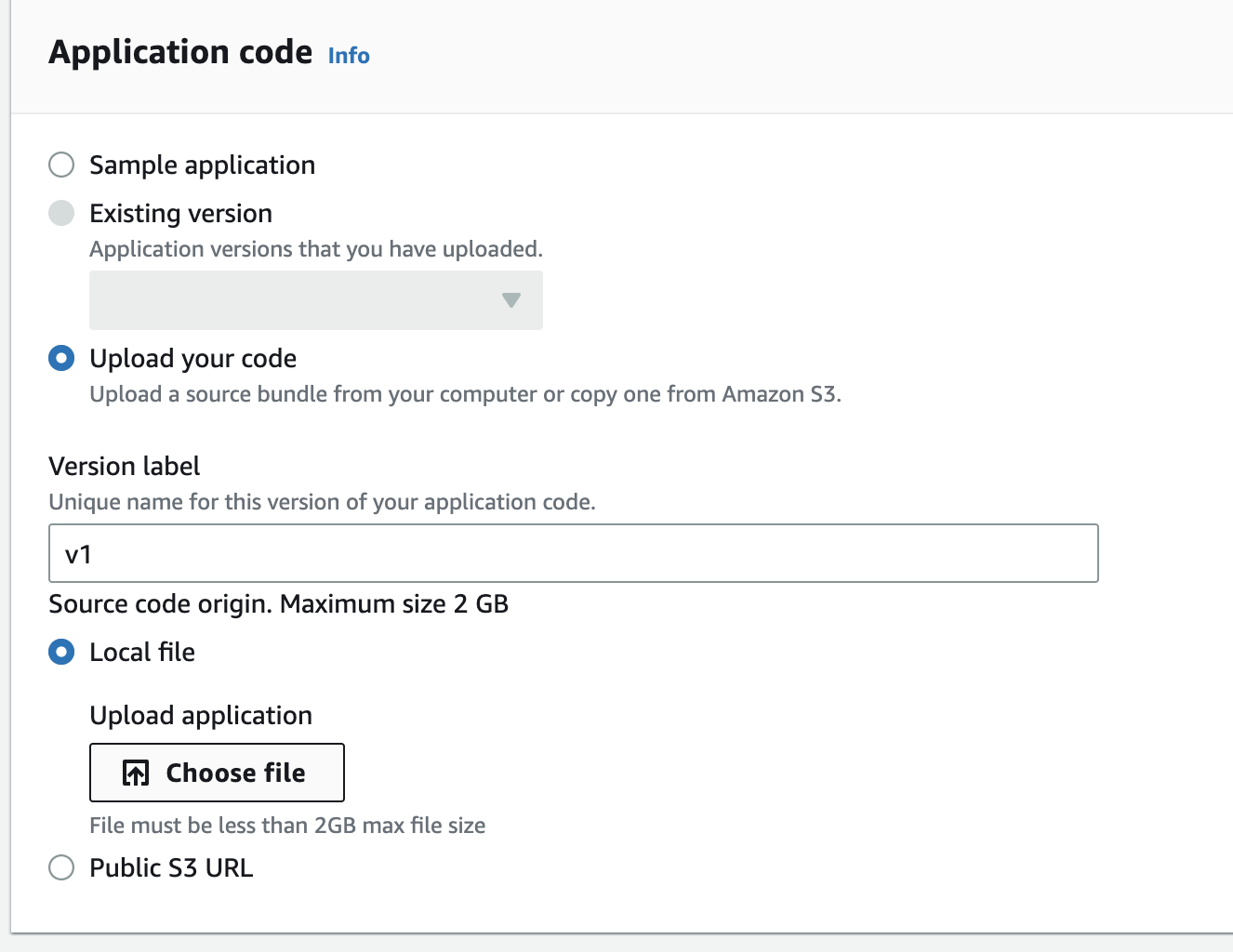
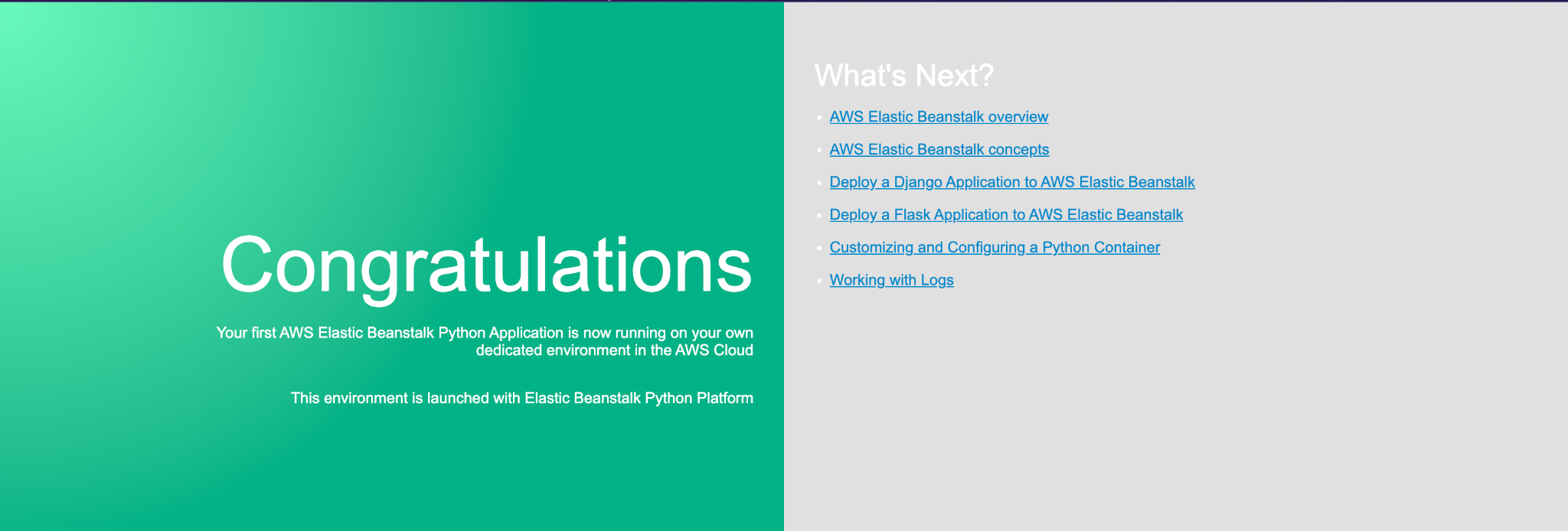
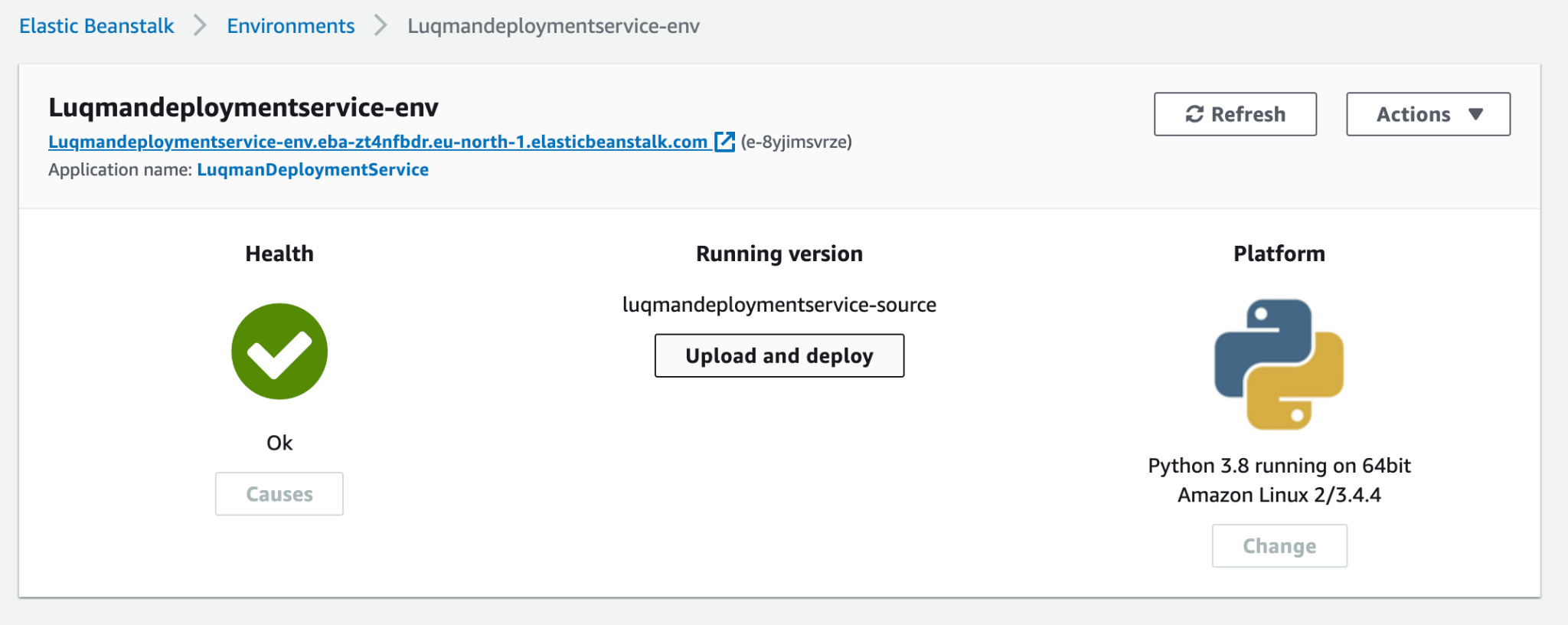
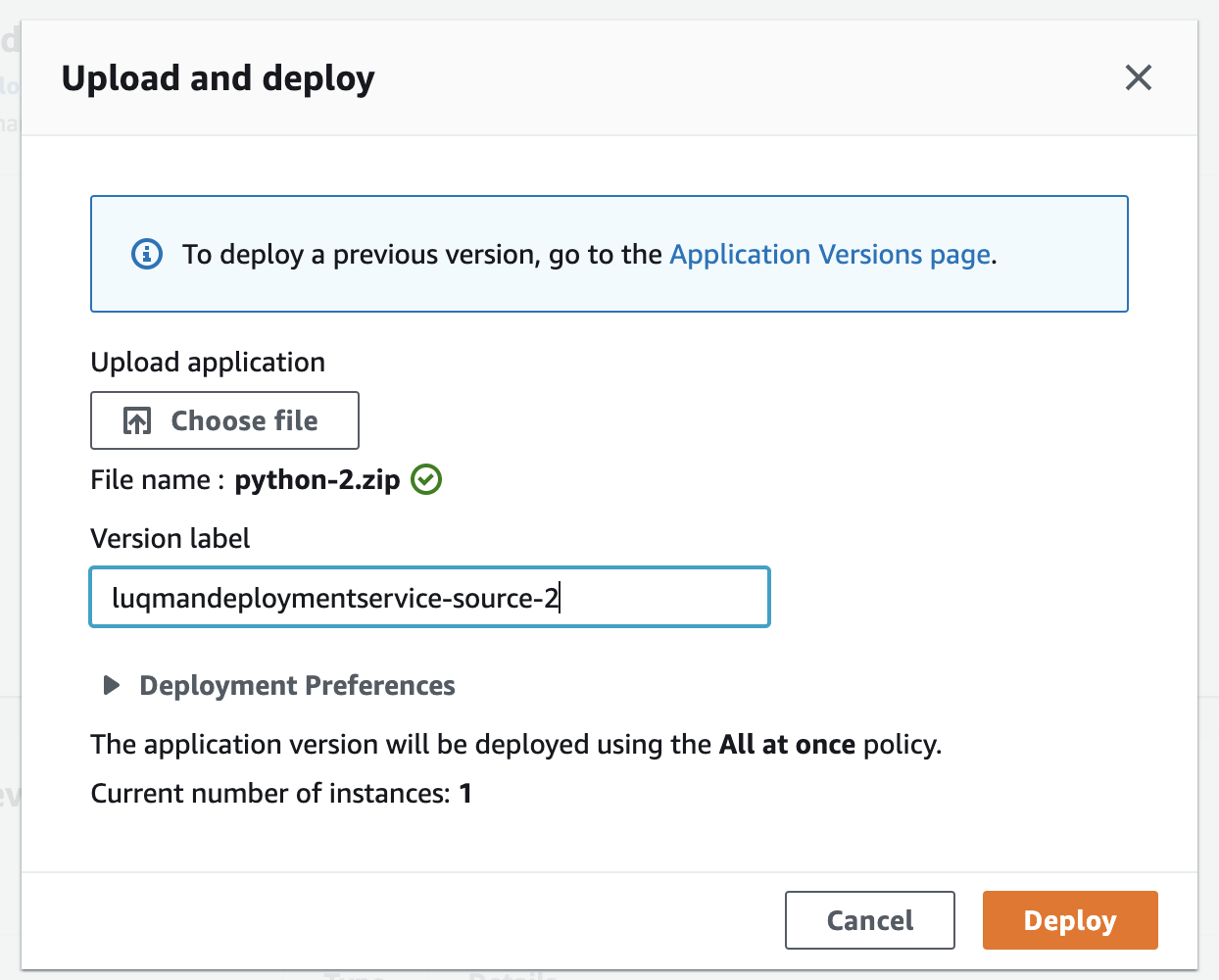
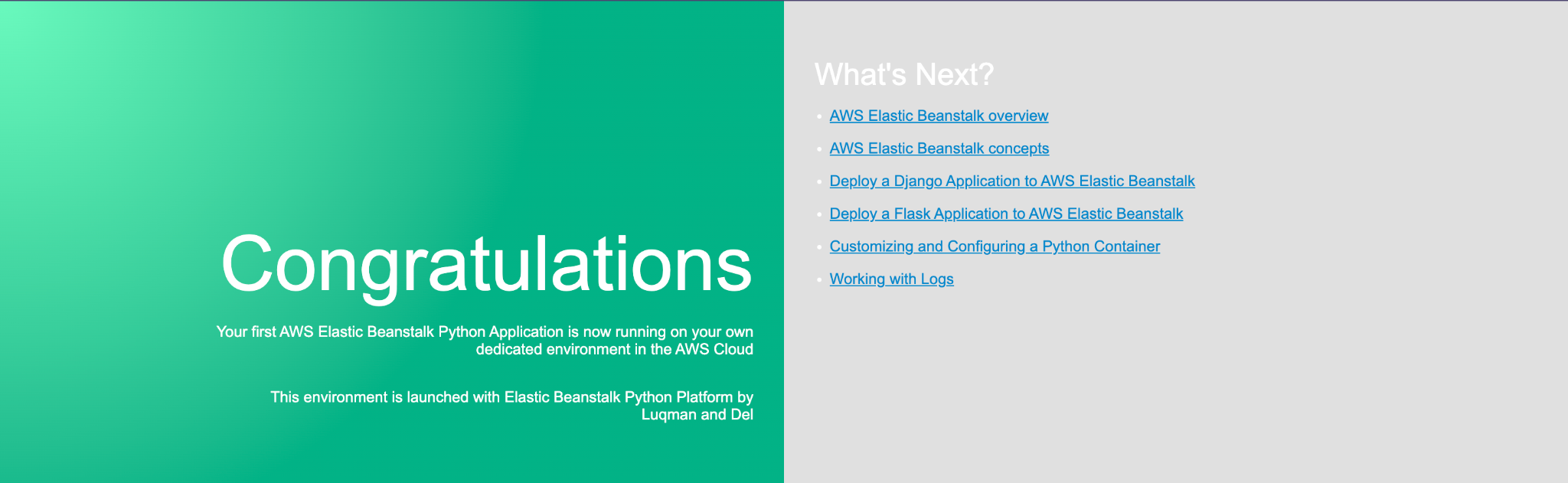
With Elastic Beanstalk, you can quickly and easily deploy your web applications to AWS, without worrying about the underlying infrastructure. Elastic Beanstalk handles the deployment, scaling, and monitoring of your application, while also providing the ability to customize and control the infrastructure as needed.

Elastic Beanstalk supports a wide range of AWS services, such as EC2, S3, RDS, and more, making it easy to integrate your application with other AWS services. It also provides a simple and intuitive web interface and command-line interface, as well as APIs and SDKs, to manage your applications and environments.

When to use Elastic Beanstalk?

* **Rapid prototyping and development**: If you are building a web application and want to get it up and running quickly, Elastic Beanstalk provides preconfigured platforms that can simplify the setup and configuration of your application, so you can focus on developing your application code.
* **Scaling your application**: Elastic Beanstalk can help you scale your application automatically in response to changing traffic patterns, by adding or removing instances as needed. This can help ensure that your application can handle spikes in traffic without downtime or performance issues.
* **Simplifying deployment**: Elastic Beanstalk provides a simple and automated deployment process, making it easy to deploy new versions of your application code. This can help reduce the time and effort required to deploy your application and reduce the risk of errors or downtime.
* **Reduced management of infrastructure**: Elastic Beanstalk can help you manage the underlying infrastructure for your application, including provisioning and managing resources such as EC2 instances, load balancers, and auto-scaling groups. This can help reduce the complexity of managing your infrastructure and ensure that it is properly configured and optimized.

### Create an Elastic Beanstalk application

* Start with the AWS Console page and head over to the AWS Elastic Beanstalk resource from the console.
* On another tab, open the page <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/GettingStarted.DeployApp.html>. We will use this as a sample code base to deploy our application. Our target today is to deploy a sample code, update the existing code, save those changes, zip the package up and deploy again with a new version.
* On the tab above, download the **python.zip** file.
* Back on the Elastic Beanstalk page in AWS console, click Create to create a new Web App. Give it a unique application name e.g. <Name>AppElasticBeanstalk.
* 
* Under Platform, Choose Platform as Python, leaving other fields as the default values.
* 
* Under Application Code, click **Upload your code**.
* You can keep the version label as the default provided, but you can explore multiple ways to upload the code - either through uploading as a zip file or using a previously created S3 bucket to host your files. You can choose either option based on your preference.
* 
* Once done, instead of clicking Create, click Configure more options.
* Look at all the configurations you can change, including “Rolling updates and deployments”. You can edit this if you want to enable Rolling updates.
* Once done, click Create app and wait for the deployment to be done.
* You should be able to view your website like this:
* 
* In the meantime, unzip your python.zip file that you downloaded from the aws website. We will be making an update to the python file and eventually zipping the folder again to redeploy as a new version in Elastic Beanstalk.
* After you have unzipped your file, open the **application.py** file using a text editor. If you have not installed any, I would recommend using <https://code.visualstudio.com/> too. Feel free to download this onto your machine.
* Scroll down and search for the text “<h1>Congratulations</h1>” in the application.py file. We will be editing some information here.
* Search for the text “**<p>This environment is launched with Elastic Beanstalk Python Platform</p>**” and update this text to something unique e.g. “**<p>This environment is launched with Elastic Beanstalk Python Platform by Luqman and Del</p>**”
* Save your changes on the file.
* Feel free to make any other changes too.
* Once done, compress/ zip up the folder Python again to get a zip file again that contains your changes.
* We will head back to the Elastic Beanstalk application and update the code with a version2 based on your changes earlier. Upload your zip file by clicking Upload and deploy.
* 
* 
* Ensure that you have another version label for your second. Click Deploy when you are ready.
* Wait for the deployment to complete and you should see your new changes in code take effect in this v2 deployment.
* Check the browser for your changes.
* 
* If you are facing errors, you may use this guide for further help - <https://aws.amazon.com/getting-started/hands-on/update-an-app/>
* Once done, terminate the environment.