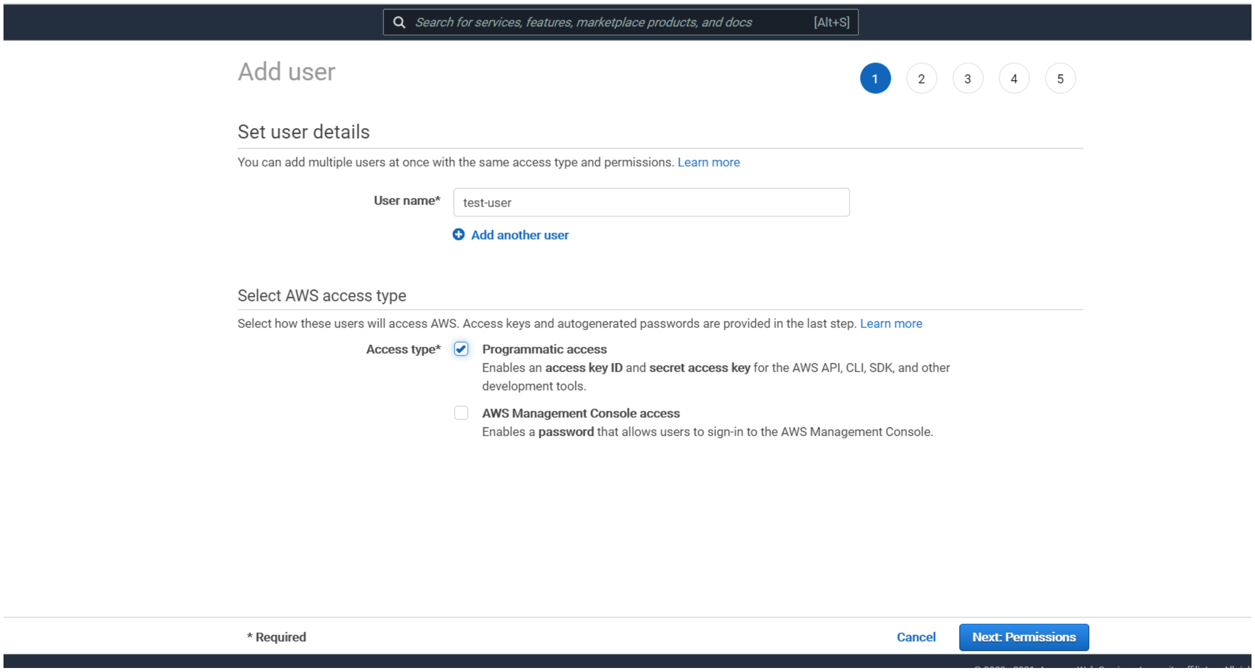
Deployment 4

Instructions:

Step 1: Create an IAM user called “Jenkins-user”. Give the “Jenkins-user” Administrator ElasticBeanStalk access. Once you added your permission policy, you will need to copy and paste the Jenkins-user Access key and Secret key in a text document for later use:

- Create Jenkins-user


Search for services, features, marketplace products, and docs
[Alt+S]
1
2
3
4
5
Add another user
Cancel
Next: Permissions

Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* test-user

Select AWS access type

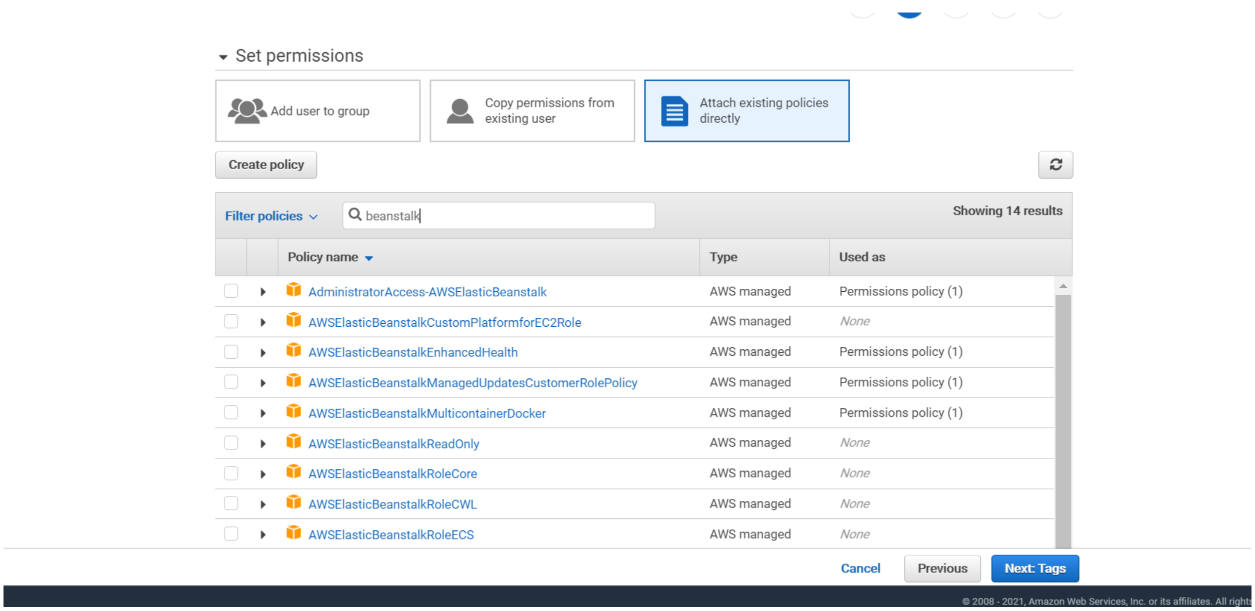
Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

Access type* ☑ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

□ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

- Select the Administrator Access AWSEB



Set permissions

Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Filter policies

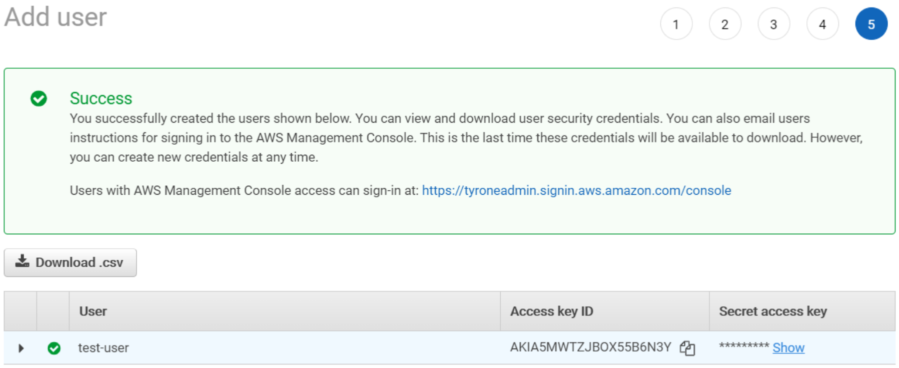
beanstalk

Showing 14 results

		Policy name	Type	Used as
☐	▸	AdministratorAccess-AWSElasticBeanstalk	AWS managed	Permissions policy (1)
☐	▸	AWSElasticBeanstalkCustomPlatformforEC2Role	AWS managed	*None*
☐	▸	AWSElasticBeanstalkEnhancedHealth	AWS managed	Permissions policy (1)
☐	▸	AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	AWS managed	Permissions policy (1)
☐	▸	AWSElasticBeanstalkMulticontainerDocker	AWS managed	Permissions policy (1)
☐	▸	AWSElasticBeanstalkReadOnly	AWS managed	*None*
☐	▸	AWSElasticBeanstalkRoleCore	AWS managed	*None*
☐	▸	AWSElasticBeanstalkRoleCWL	AWS managed	*None*
☐	▸	AWSElasticBeanstalkRoleECS	AWS managed	*None*

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights

- Below is where you'll copy the access key and secret access key



Add user

Success

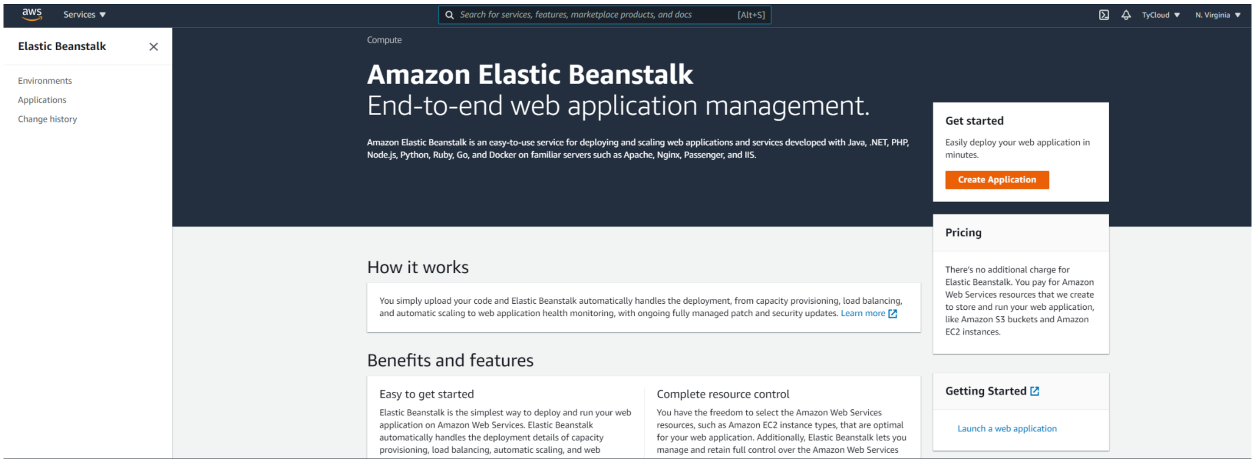
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: https://tyroneadmin.signin.aws.amazon.com/console

		User	Access key ID	Secret access key
▸	✓	test-user	AKIA5MWTZJBOX55B6N3Y	********* Show

Step 2: Create your Elastic BeanStalk by entering the application name, key, value, platform, and application code (see below):

- Enter the AWSEB setup by selecting "create application"



Amazon Elastic Beanstalk
End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

Get started

Easily deploy your web application in minutes.

Pricing

There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

How it works

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. Learn more

Benefits and features

Easy to get started

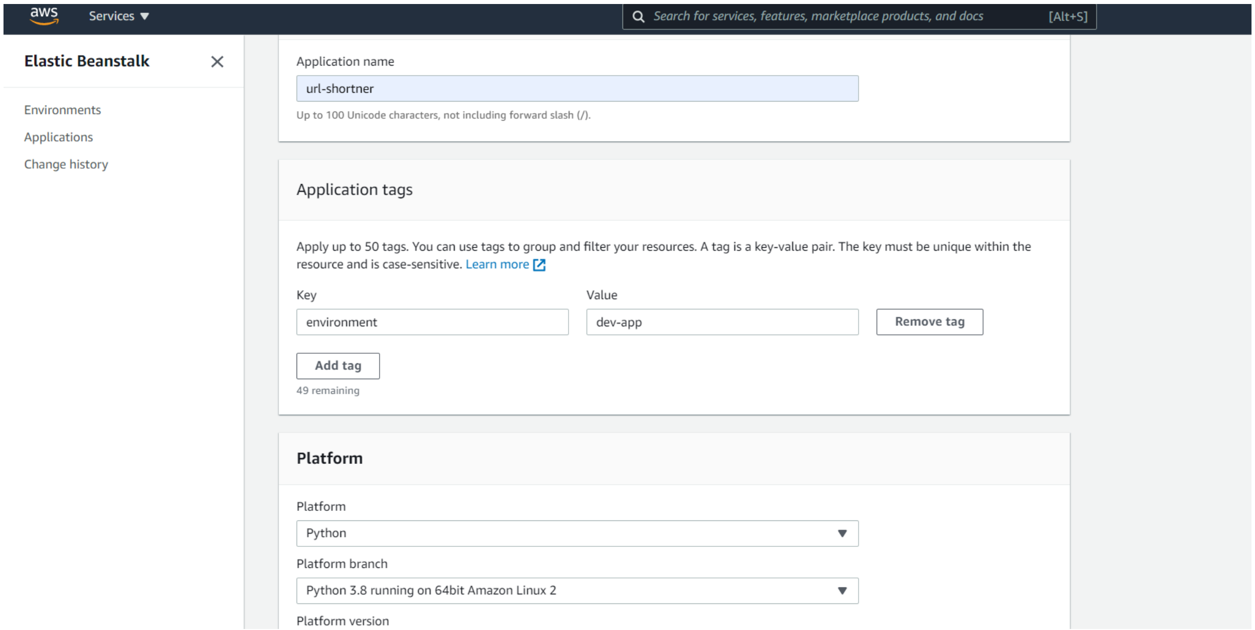
Elastic Beanstalk is the simplest way to deploy and run your web application on Amazon Web Services. Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, automatic scaling, and web

Complete resource control

You have the freedom to select the Amazon Web Services resources, such as Amazon EC2 instance types, that are optimal for your web application. Additionally, Elastic Beanstalk lets you manage and retain full control over the Amazon Web Services

Getting Started

- Enter your application name, application tag, platform and select sample code


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
Elastic Beanstalk
Environments
Applications
Change history

Application name

url-shortner

Up to 100 Unicode characters, not including forward slash (/).

Application tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. Learn more

Key	Value	
environment	dev-app	Remove tag

49 remaining

Platform

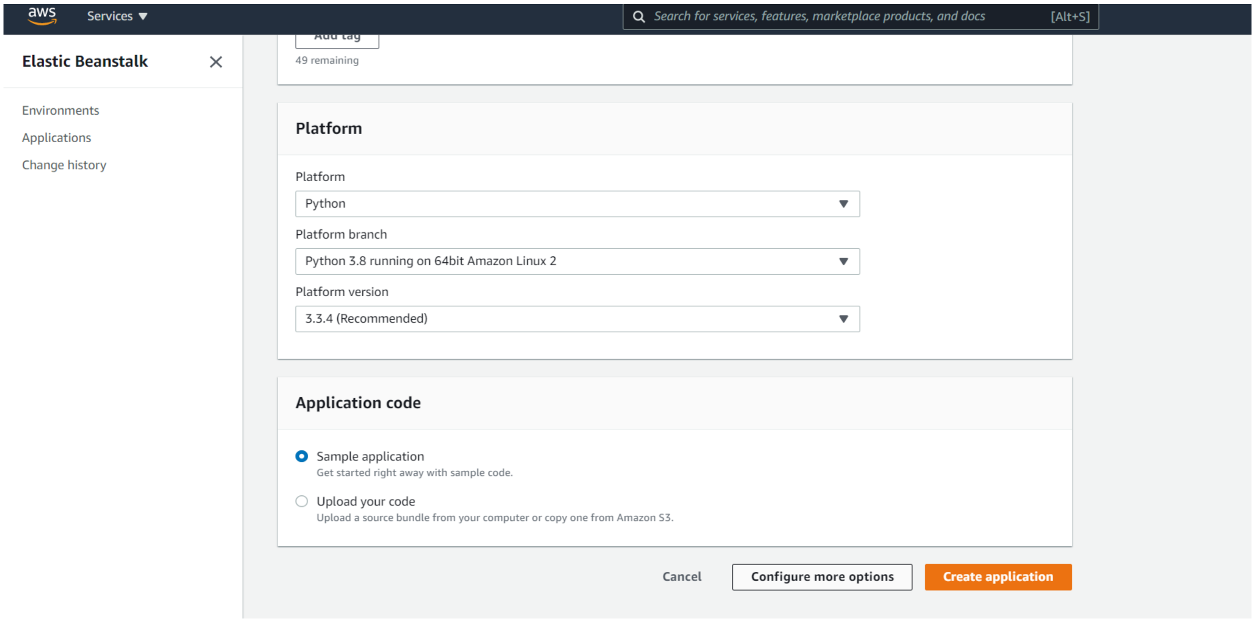
Platform

Python ▼

Platform branch

Python 3.8 running on 64bit Amazon Linux 2 ▼

Platform version



49 remaining

Platform

Platform: Python

Platform branch: Python 3.8 running on 64bit Amazon Linux 2

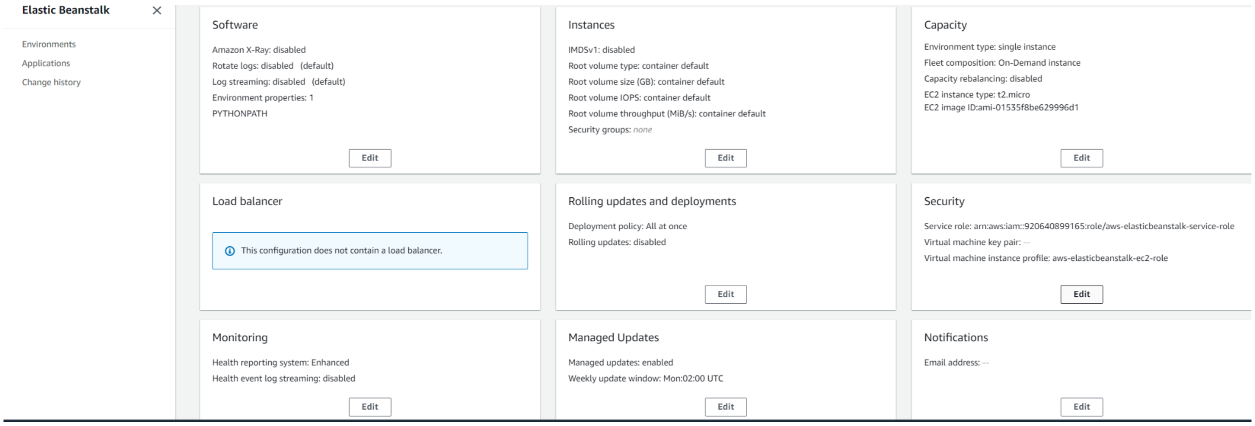
Platform version: 3.3.4 (Recommended)

Application code

● Sample application
Get started right away with sample code.

○ Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

OPTIONAL(adding key pair to EC2): Select configure more options from the above image, select security and then add your key pair to ssh into beanstalk EC2



Software

Amazon X-Ray: disabled
Rotate logs: disabled (default)
Log streaming: disabled (default)
Environment properties: 1
PYTHONPATH

Instances

IMDSv1: disabled
Root volume type: container default
Root volume size (GB): container default
Root volume IOPS: container default
Root volume throughput (MiB/s): container default
Security groups: *none*

Capacity

Environment type: single instance
Fleet composition: On-Demand instance
Capacity rebalancing: disabled
EC2 instance type: t2.micro
EC2 image ID:ami-01535f8be629996d1

Load balancer

This configuration does not contain a load balancer.

Rolling updates and deployments

Deployment policy: All at once
Rolling updates: disabled

Security

Service role: arn:aws:iam::920640899165:role/aws-elasticbeanstalk-service-role
Virtual machine key pair: --
Virtual machine instance profile: aws-elasticbeanstalk-ec2-role

Monitoring

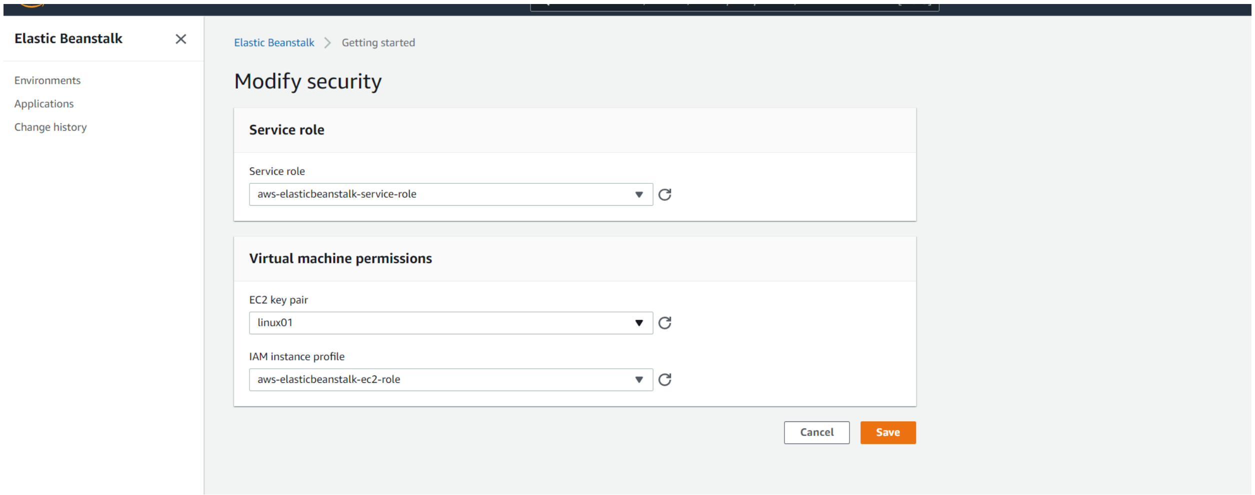
Health reporting system: Enhanced
Health event log streaming: disabled

Managed Updates

Managed updates: enabled
Weekly update window: Mon:02:00 UTC

Notifications

Email address: --



Modify security

Service role

Service role: aws-elasticbeanstalk-service-role

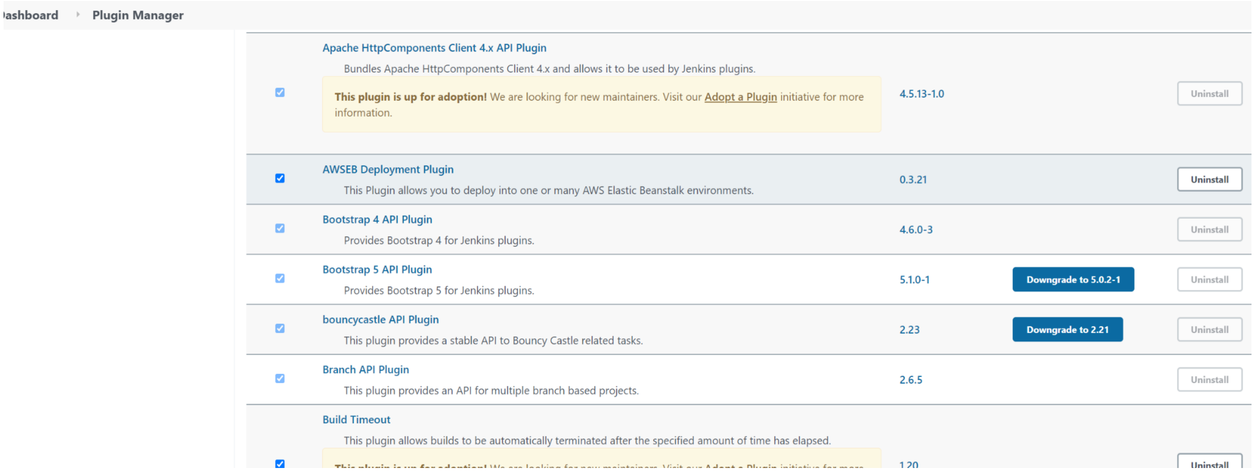
Virtual machine permissions

EC2 key pair: linux01

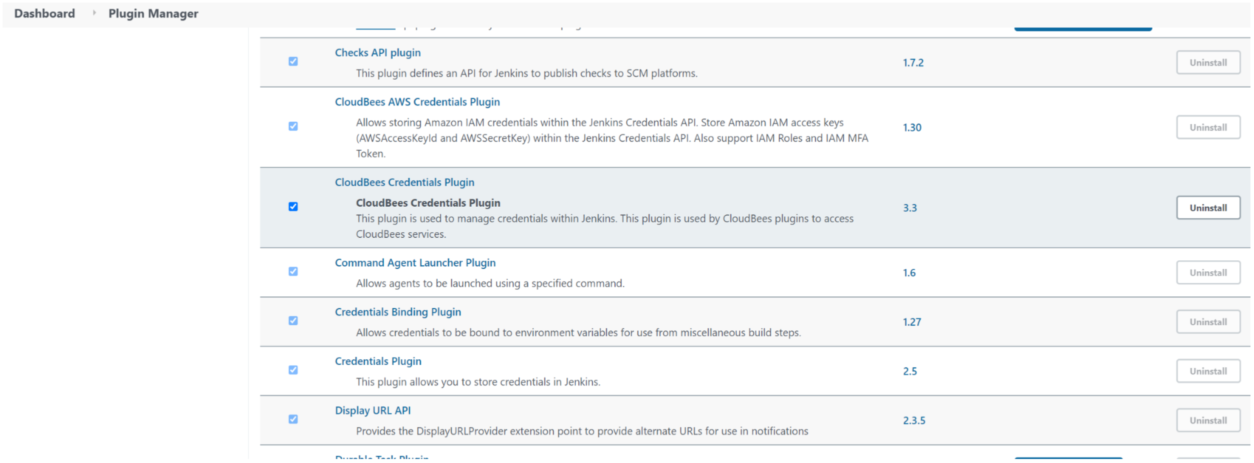
IAM instance profile: aws-elasticbeanstalk-ec2-role

Step 3: Turn on your Jenkins server and download the required plugins below:

- Required plugins
  - AWSEB Deployment Plugin
  - CloudBees Credentials Plugin



☑	Apache HttpComponents Client 4.x API Plugin Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins. **This plugin is up for adoption!** We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	4.5.13-1.0		Uninstall
☑	AWSEB Deployment Plugin This Plugin allows you to deploy into one or many AWS Elastic Beanstalk environments.	0.3.21		Uninstall
☑	Bootstrap 4 API Plugin Provides Bootstrap 4 for Jenkins plugins.	4.6.0-3		Uninstall
☑	Bootstrap 5 API Plugin Provides Bootstrap 5 for Jenkins plugins.	5.1.0-1	Downgrade to 5.0.2-1	Uninstall
☑	bouncycastle API Plugin This plugin provides a stable API to Bouncy Castle related tasks.	2.23	Downgrade to 2.21	Uninstall
☑	Branch API Plugin This plugin provides an API for multiple branch based projects.	2.6.5		Uninstall
☑	Build Timeout This plugin allows builds to be automatically terminated after the specified amount of time has elapsed. **This plugin is up for adoption!** We are looking for new maintainers. Visit our Adopt a Plugin initiative for more	1.20		Uninstall

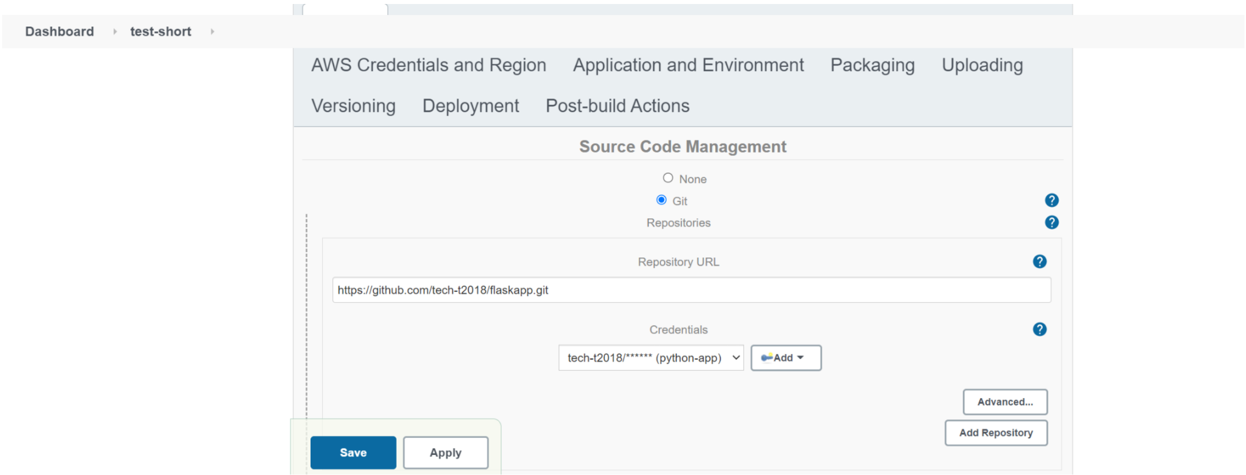


Dashboard › Plugin Manager

	Name	Version	
☑	**Checks API plugin** This plugin defines an API for Jenkins to publish checks to SCM platforms.	1.7.2	Uninstall
☑	**CloudBees AWS Credentials Plugin** Allows storing Amazon IAM credentials within the Jenkins Credentials API. Store Amazon IAM access keys (AWSAccessKeyId and AWSSecretKey) within the Jenkins Credentials API. Also support IAM Roles and IAM MFA Token.	1.30	Uninstall
☑	**CloudBees Credentials Plugin** CloudBees Credentials Plugin This plugin is used to manage credentials within Jenkins. This plugin is used by CloudBees plugins to access CloudBees services.	3.3	Uninstall
☑	**Command Agent Launcher Plugin** Allows agents to be launched using a specified command.	1.6	Uninstall
☑	**Credentials Binding Plugin** Allows credentials to be bound to environment variables for use from miscellaneous build steps.	1.27	Uninstall
☑	**Credentials Plugin** This plugin allows you to store credentials in Jenkins.	2.5	Uninstall
☑	**Display URL API** Provides the DisplayURLProvider extension point to provide alternate URLs for use in notifications	2.3.5	Uninstall

<u>Step 4:</u> Create a freestyle project with the following configurations set:

- Select git for source code manager



Source Code Management

○ None

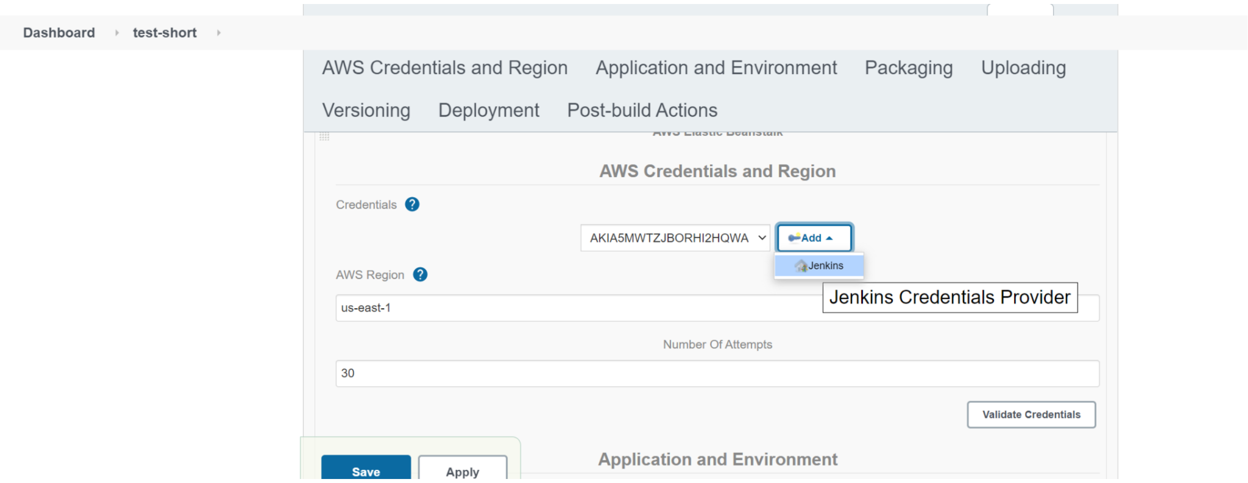
● Git

Repositories

Repository URL: https://github.com/tech-t2018/flaskapp.git

Credentials: tech-t2018/****** (python-app)

- Scroll down to post actions and select deploy to ElasticBeanstalk.
  - First enter AWS credentials (the ID field can be anything):

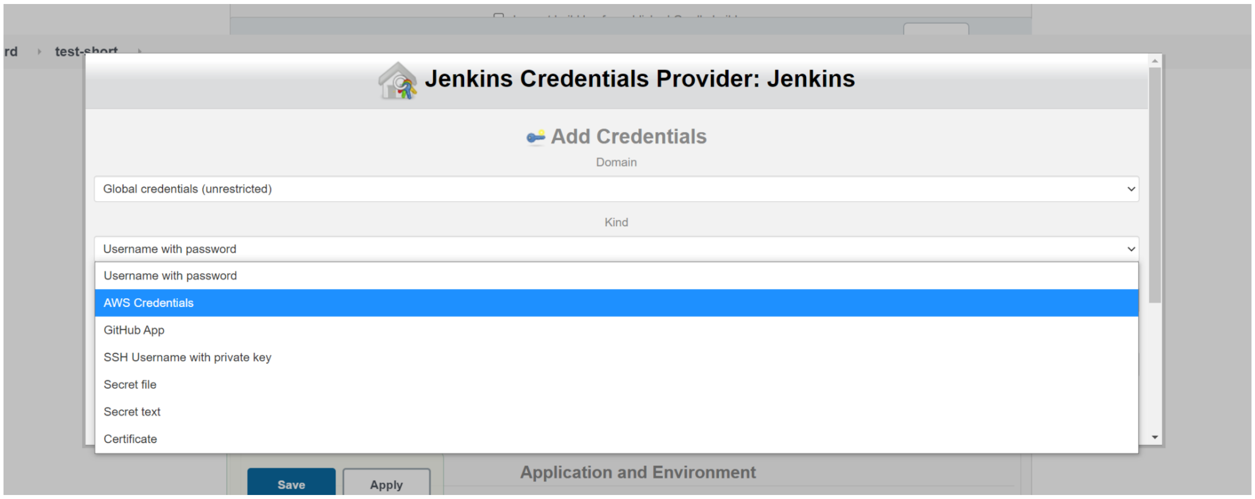


AWS Credentials and Region

Credentials: AKIA5MWTZJBORHI2HQWA

AWS Region: us-east-1

Number Of Attempts: 30



Jenkins Credentials Provider: Jenkins

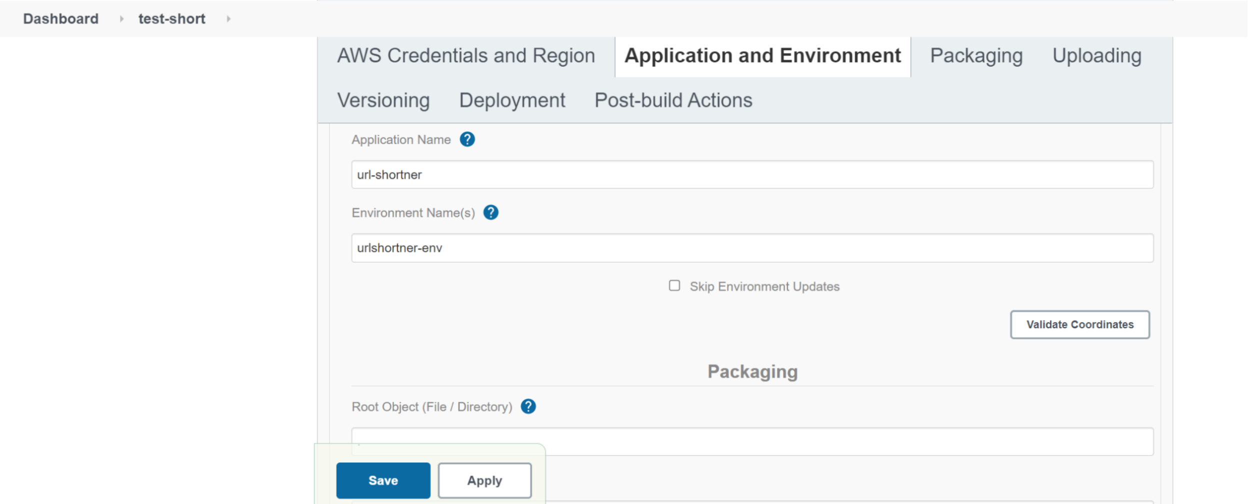
Add Credentials

Domain: Global credentials (unrestricted)

Kind: Username with password

- Second select your AWS credentials, enter your region and select validate credentials

- Third: Enter the application name and environment name. Once entered, validate coordinates



Application Name

url-shortner

Environment Name(s)

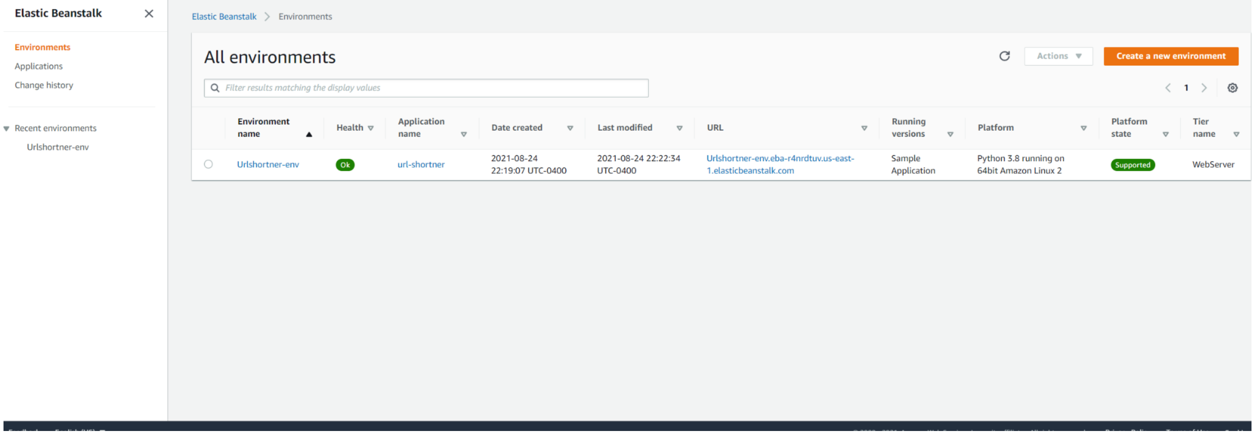
urlshortner-env

□ Skip Environment Updates

Packaging

Root Object (File / Directory)

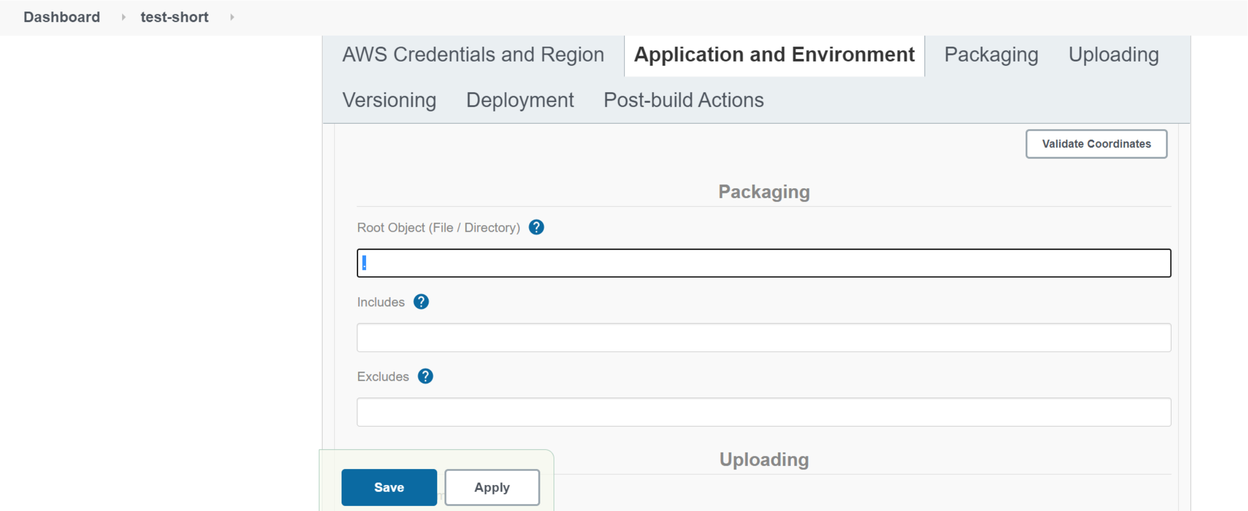
- Location of application name and environment is on the Environments page of AWS



All environments

	Environment name	Health	Application name	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
○	Urlshortner-env	Ok	url-shortner	2021-08-24 22:19:07 UTC-0400	2021-08-24 22:22:34 UTC-0400	Urlshortner-env.eba-r4nrdtuv.us-east-1.elasticbeanstalk.com	Sample Application	Python 3.8 running on 64bit Amazon Linux 2	Supported	WebServer

- Fourth: In the Packaging section, enter a period in the Root Object field



Packaging

Root Object (File / Directory)

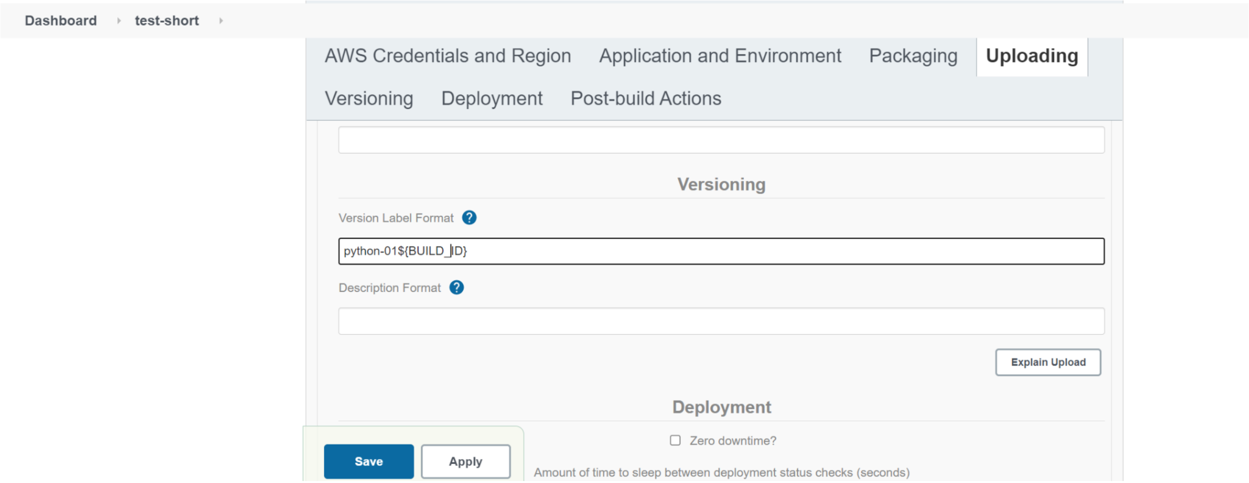
.

Includes

Excludes

Uploading

- Finally: Under the Version Label Format enter what you see below



Versioning

Version Label Format

python-01${BUILD_ID}

Description Format

Deployment

□ Zero downtime?

Amount of time to sleep between deployment status checks (seconds)

Save and start the build to deploy your application to AWSEB.