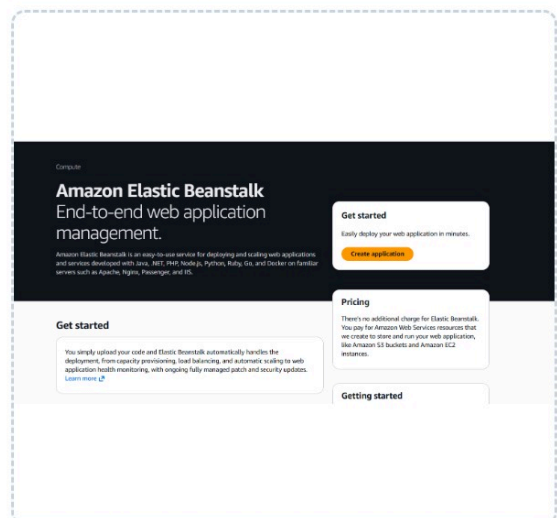


Flask Web App Deployment

AWS Elastic Beanstalk Strategy

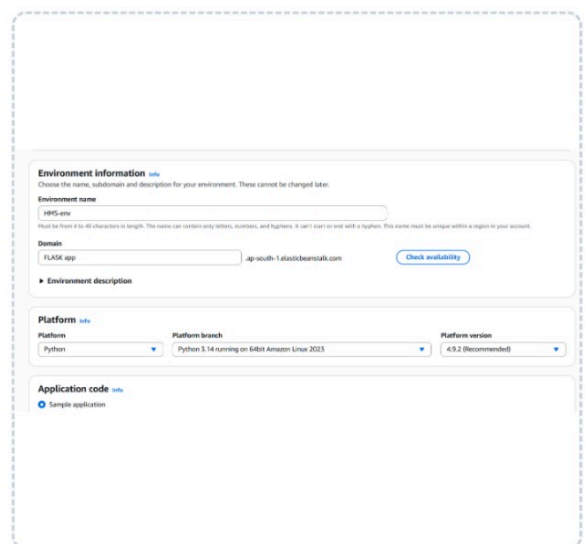
01 Create Application

Initiate the deployment by defining your application name in the AWS Management Console.



02 Environment & Platform

Configure the environment tier and select the Python managed platform version.




The screenshot shows the 'Environment' configuration page in the AWS Lambda console. It is divided into three main sections: 'Environment information', 'Platform', and 'Application code'.

- Environment information:** This section contains fields for 'Environment name' (set to 'HPS-env'), 'Domain' (set to 'FLASK app'), and 'Environment description'. A 'Check availability' button is present next to the domain field.
- Platform:** This section contains three dropdown menus: 'Platform' (set to 'Python'), 'Platform branch' (set to 'Python 3.14 running on 64bit Amazon Linux 2023'), and 'Platform version' (set to '4.9.2 (Recommended)').
- Application code:** This section contains a radio button labeled 'Sample application'.

03 Application Code

Upload your source bundle as a ZIP file ensuring it contains 'application.py'.



The screenshot shows a web form titled "Application code" with a "help" link. It contains several sections: "Sample application" with a radio button and a note about application version; "Upload your code" with a radio button and a note about uploading a source bundle; "Version label" with a text input field and a "Save" button; "Source code origin" with a radio button and a note about maximum size; "Upload application" with a "Choose file" button; and "File name: Rank app.zip" with a note about file size and a "Public: US URL" option.

Application code [help](#)

☐ Sample application
Existing version:
Application version that you have uploaded.

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

Version label
Unique name for this version of your application code.

Save

Source code origin: Maximum size 500 MB

☒ Local file

Upload application

Choose file

File name: Rank app.zip
File must be less than 500MB and less than 100 files

☐ Public: US URL

04 Version Label & Presets

Define a unique version label and choose a configuration preset.

The screenshot shows the 'Create new role' page in the AWS IAM console. The 'Version label' section has a text input field with 'test' entered. Below it, the 'Source code origin' is set to 'Local file'. The 'Upload application' section shows a file named 'test-app.zip' with a size of 1000 bytes. The 'Presets' section is expanded, showing 'Configuration presets' with 'Single instance (free tier eligible)' selected. Other options include 'Single instance (using spot instance)', 'High availability', 'High availability (using spot and on-demand instances)', and 'Custom configuration'.

Version label
Enter a name for this version of your application code.

test

Source code origin. Maximum size 500 MB

☒ Local file

Upload application

☒ Choose file

☒ File name: test-app.zip
File must be less than 1000000 bytes (1 MB)

☐ Public S3 URL

Presets [help](#)

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

☒ Single instance (free tier eligible)

☐ Single instance (using spot instance)

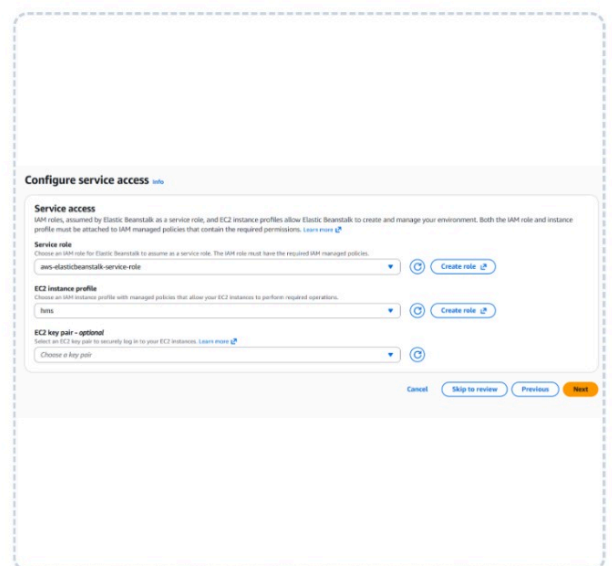
☐ High availability

☐ High availability (using spot and on-demand instances)

☐ Custom configuration

05 Service Access (IAM)

Assign necessary IAM roles and EC2 instance profiles for permissions.



The screenshot shows the 'Configure service access' page in the AWS IAM console. The page is titled 'Configure service access' with a small 'info' link. Below the title, there is a section 'Service access' with a description: 'IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)'. The page is divided into three sections: 'Service role', 'EC2 instance profile', and 'EC2 key pair - optional'. Each section has a dropdown menu to select a role or profile and a 'Create role' or 'Create profile' button. The 'Service role' dropdown is set to 'aws-elasticbeanstalk-service-role'. The 'EC2 instance profile' dropdown is set to 'None'. The 'EC2 key pair - optional' dropdown is set to 'Choose a key pair'. At the bottom right, there are four buttons: 'Cancel', 'Skip to review', 'Previous', and 'Next'.

Configure service access [info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role
Choose an IAM role for Elastic Beanstalk to assume as a service role. The IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role [Create role](#)

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

None [Create profile](#)

EC2 key pair - optional
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair [Create key pair](#)

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

06 VPC and Subnets

Set up network isolation by selecting a VPC and designating subnets.

Choose a subnet in each AZ for the instances that run your application, to avoid exposing your instances to the internet, run your instances in private subnets and add a load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances. [Learn more](#)

VPC
Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. [Learn more](#)

vpc-05173c395afe96255 (172.31.0.0/16) [Create VPC](#)

Public IP address
Assign a public IP address to the Amazon EC2 instances in your environment.

☒ Enable

Instance subnets
Filter instance subnets

Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/> ap-south-1c	subnet-0592b9fa0c36186fe	172.31.16.0/20	
<input checked="" type="checkbox"/> ap-south-1b	subnet-0a999fa0009b650	172.31.0.0/20	
<input type="checkbox"/> ap-south-1a	subnet-0a52607d49f0ab2a2	172.31.32.0/20	

07 Database Setup

Integrate an RDS instance for persistent relational storage.

Database settings.
Choose an engine and instance type for your environment's database.

Engine

Engine version

Instance class

Storage
Choose a number between 5 GB and 1024 GB.
 GB

Username

Password

08 Configure Environment

Customize instance types and auto-scaling triggers.

Review [info](#)

Step 1: Configure environment [Edit](#)

Environment information

Environment tier	Application name
Web server environment	test
Environment name	Application code
test-env	test-app.zip
Platform	
aws-elasticbeanstalk-ubuntu-1: platform/Python 3.14 running on 64bit Amazon Linux 2023/4.9.2	

09 Review & Finish

Verify configuration details and execute the launch.

The screenshot shows the AWS CloudFormation console for an environment named 'flask_form'. At the top, there's a search bar with the text 'flask_form' and a button 'Create a new environment'. Below the search bar, there's a table with columns: Environment name, Health, Date created, Last modified, URL, Running versions, Platform, Platform state, and Tier name. The table contains one entry for 'flask_form-env' with a green 'HEALTHY' status. The 'URL' column shows a link to 'flask-form-env-272acmcs-est-2-9g6ic6uamh.com'. The 'Running versions' column shows 'flask_form-source'. The 'Platform' column shows 'Python 3.7 running on 64bit Amazon Linux 2'. The 'Platform state' column shows 'HEALTHY'. The 'Tier name' column shows 'WebServer'.

Environment name	Health	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
flask-form-env	HEALTHY	2020-12-22 12:21:05 UTC+0530	2020-12-22 12:28:51 UTC+0530	flask-form-env-272acmcs-est-2-9g6ic6uamh.com	flask_form-source	Python 3.7 running on 64bit Amazon Linux 2	HEALTHY	WebServer

Conclusion & Next Steps

✓ **Production Ready:** Scalable managed infrastructure.

↻ **Easy Updates:** Simple version management.

📈 **Monitoring:** Health dashboards and CloudWatch logs.

Deployment Successful!