

Smart Cow - React Flask NGINX Cloud Deployment Guide

Task #2

<https://github.com/ryancomia/sc-exercise/tree/main>

Before we begin deploying into AWS, let's create a Workflow to deploy the code in an automated manner.

For this we will use Github Actions. I wrote the following yaml file named deploy.yml

Found here:

<https://github.com/ryancomia/sc-exercise/blob/dev/.github/workflows/deploy.yml>

Steps:

- Checkout the code when a trigger to main branch
- Creates a package of the code and zips it
- Generate a timestamp
- Adds a string into the name with timestamp
- Deploys the package into AWS S3 and triggers an action to deploy it into the Elastic Beanstalk environment

Before we can deploy we need the following:

- AWS Keys
- Application Name
- Application ENV

```
1  name: Deploy to AWS Beanstalk
2  on:
3    push:
4      branches: [ main ]
5  jobs:
6    Deploy:
7      runs-on: ubuntu-latest
8      steps:
9        - name: Checkout Source Code
10         uses: actions/checkout@master
11
12        - name: Generate Deployment Package
13          run: zip -r deployment.zip * -x "**node_modules**.*.git*"
14
15        - name: Create a Timestamp
16          uses: gerred/actions/current-time@master
17          id: current-time
18
19        - name: String Replacement
20          uses: frabert/replace-string-action@v2.0
21          id: format-time
22          with:
23            pattern: '[:\.\.]+'
24            string: '${{ steps.current-time.outputs.time }}'
25            replace-with: '='
26            flags: 'g'
27
28        - name: Deploy to Elastic Beanstalk
29          uses: einaregilsson/beanstalk-deploy@v20
30          with:
31            aws_access_key: ${ secrets.AWS_ACCESS_KEY_ID }
32            aws_secret_key: ${ secrets.AWS_SECRET_ACCESS_KEY }
33            application_name: smartcow-testapp
34            environment_name: Smartcowtestapp-env
35            version_label: "sc-exercise-${{ steps.format-time.outputs.replaced }}"
36            region: ap-southeast-1
37            deployment_package: deployment.zip
```

Ideally we want to create a Group and attach a policy into it.

Note: granting permission to a specific user/service account is a bad practice! =)

IAM

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

IAM

>

User groups

>

Create user group

Create user group

Name the group

User group name

Enter a meaningful name to identify this group.

elastic-beanstalk-deployer-group

Maximum 128 characters. Use alphanumeric and '+=, @- _' characters.

For this exercise:

We will use AWS Elastic Beanstalk admin access for the service account

[IAM](#) > [User groups](#) > elastic_beanstalk-deployer-group

elastic_beanstalk-deployer-group

Summary

User group name	Creation time
elastic_beanstalk-deployer-group	June 10, 2022, 20:12 (UTC+08:00)


- Users
- Permissions
- Access Advisor

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Q

Filter policies by property or policy name and press enter

<input type="checkbox"/>	Policy name ↗	Type
<input type="checkbox"/>	<div><div>+</div><div> AdministratorAccess-AWSElasticBeanstalk</div></div>	AWS managed

Modify the account credential type to access-key programmatic type

User name*

githubaction-eb-serviceaccount

+

Add another user

Access type

Users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Credential type*

☒

Access key - Programmatic access

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

☐


Password - AWS Management Console access

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

Add the created Group for elastic beanstalk deployer

	Group ▼	Attached policies
<input checked="" type="checkbox"/>	elastic_beanstalk-deployer-group	AdministratorAccess-AWSElasticBeanstalk


Finally we can get the API keys and.
The Keys will then be stored into github,
and we can then call this to deploy using
github action






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://779385966398.signin.aws.amazon.com/console>

 Download .csv

	User	Access key ID	Secret access key
	 githubaction-eb-serviceaccount	AKIA3K5YGB	 ***** Show

In AWS bean-stalk, we create a web application that would be used as our deployment environment

I chose Docker as the target platform as we intend it to be a containerized deployment

IAM

Elastic Beanstalk

×

Environments

Applications

Change history

▼ Recent environments

Smartcowtestapp-env

Application information

Application name

smartcow-devops-app

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

Remove tag

Add tag

50 remaining

Platform

Platform

Docker

Platform branch

Docker running on 64bit Amazon Linux 2

Platform version

3.4.16 (Recommended)

Spinning up an EB environment takes about 5-10mins.



Creating Scexerciseapp-env
This will take a few minutes. ...

- 9:02pm Created security group named:
awseb-e-mtq76pt8ia-stack-AWSEBSecurityGroup-1RP6B1JQT1PX9
- 9:01pm Environment health has transitioned to Pending. Initialization in progress (running for 16 seconds). There are no instances.
- 9:01pm Created security group named:
sg-03c9a94892bab8f40
- 9:01pm Created target group named:
arn:aws:elasticloadbalancing:ap-southeast-1:779385966398:targetgroup/awseb-AWSEB-1R11GFHOPJSAC/9f93435f2909bb82
- 9:01pm Using elasticbeanstalk-ap-southeast-1-779385966398 as Amazon S3 storage bucket for environment data.
- 9:01pm createEnvironment is starting.


Now that the environment is up & running, we can get the following:

- Public URI
- Application Name
- Application Env

configuration.


Elastic Beanstalk > Environments > Smartcowdevopsapp-env

Smartcowdevopsapp-env

Smartcowdevopsapp-env.eba-rwbfpzum.ap-southeast-1.elasticbeanstalk.com  (e-fmi3923dmd)

Application name: smartcow-devops-app

Health



Ok

Causes

Recent events

Given that we have the EB load balancer endpoint, I updated the react fetch to use the URI to get api stats

Ideally in a actual PROD setup, we need to have a public DNS record CNAME'd to this load balancer.

However I did try creating a Route 53 record but unfortunately the record propagation is not fast enough. (And give I am left 1 day to complete and document this exercise!, So I resorted to use the LB URI. =)

Note: using a CNAME also allows for a faster blue/green deployment approach. As it is faster to switch a record compared to re-triggering a action redeployment

```
async loadData() {  
  try {  
    const res = await fetch('http://smartcowtestapp-env.eba-mqh2ba2q.ap-southeast-1.elasticbeanstalk.com/stats');  
    const blocks = await res.json();  
    const ram = blocks.ram;  
    const cpu = blocks.cpu;  
  }  
}
```

We can then pass the Application Name and Environment name as a variable in deploy.yml for the action to identify where we want to deploy it.

```
when:
  aws_access_key: ${ secrets.AWS_ACCESS_KEY_ID }
  aws_secret_key: ${ secrets.AWS_SECRET_ACCESS_KEY }
  application_name: smartcow-testapp
  environment_name: Smartcowtestapp-env
  version_label: "sc-exercise-${ steps.format-time.outputs.replaced }"
  region: ap-southeast-1
  deployment_package: deployment.zip
```

CLOUD URI

<http://smartcowdevopsapp-env.eba-rwbfpsum.ap-southeast-1.elasticbeanstalk.com/>

