CI/CD Pipeline Investigations

Objective

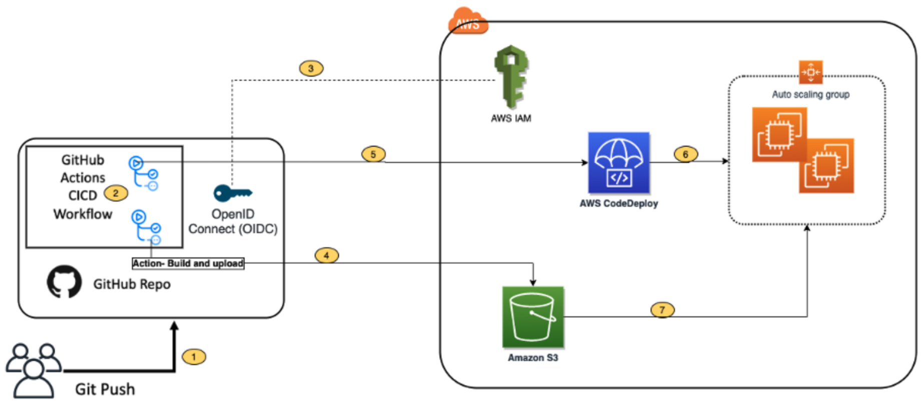
Utilize GitHub and AWS to build a CI/CD pipeline for building and deploying the Validation Hub application on EC2 instances in an AutoScaling group.

1. GitHub stores original code and GitHub Actions is in charge of building, testing, packaging, and deploying the code
2. AWS CodeDeploy automates the application deployments to AWS EC2 instances

Investigations

Steps

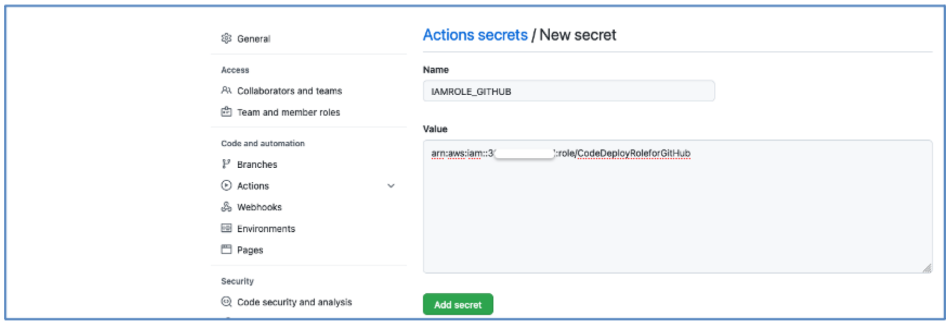
The following diagram illustrates the architecture for the solution:



1. Developer commits code changes from their local repo to the GitHub repository. In this post, the GitHub action is triggered manually, but this can be automated.
2. GitHub action triggers the build stage.
3. GitHub's Open ID Connector (OIDC) uses the tokens to authenticate to AWS and access resources.
4. GitHub action uploads the deployment artifacts to Amazon S3.
5. GitHub action invokes CodeDeploy.
6. CodeDeploy triggers the deployment to Amazon EC2 instances in an Autoscaling group.
7. CodeDeploy downloads the artifacts from Amazon S3 and deploys to Amazon EC2 instances.

1. Deploy AWS resources using infrastructure as code (IaC) services. Resources include VPC, IAM roles, CodeDeploy, S3 bucket, AutoScaling group with EC2 instances. We have two options:
    a. AWS CloudFormation: more convenient but only compatible with AWS
    b. Terraform: platform-agnostic
2. Update deploy.yml for GitHub Actions workflow
    a. Amazon S3 bucket

    b. region
3. Update the S3 bucket in after-install.sh
4. Setup Github secrets
   IAM OpenID Connect (OIDC) is used to let GitHub Actions access CodeDeploy and Amazon S3 bucket.
   Fill the IAM role Arn into GitHub Actions secrets

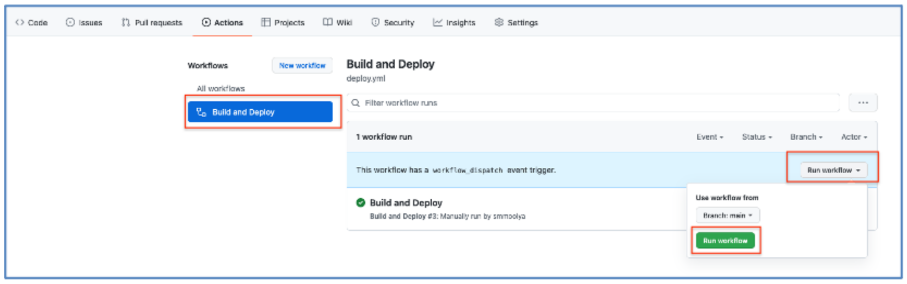


Actions secrets / New secret

Name: IAMROLE_GITHUB

Value: arn:aws:iam::3[redacted]:role/CodeDeployRoleforGitHub

5. Integrate CodeDeploy with GitHub
6. Trigger the GitHub Actions workflow manually

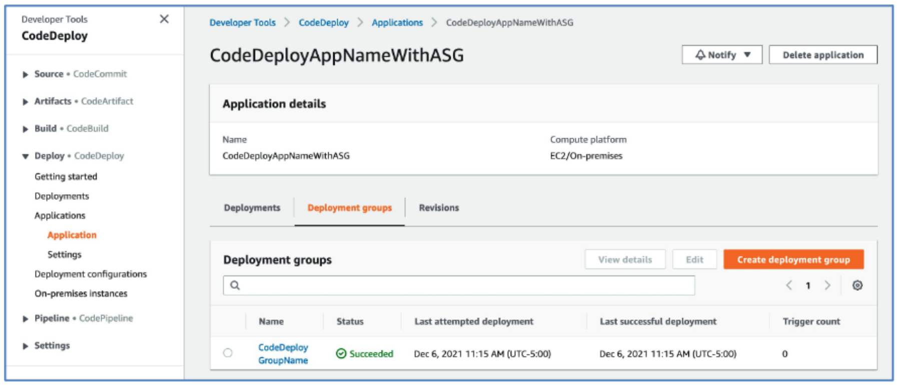


Build and Deploy
deploy.yml

1 workflow run

This workflow has a `workflow_dispatch` event trigger.

Build and Deploy
Build and Deploy #3: Manually run by smmoolya



CodeDeployAppNameWithASG

Application details

Name	Compute platform
CodeDeployAppNameWithASG	EC2/On-premises

Deployment groups

Name	Status	Last attempted deployment	Last successful deployment	Trigger count
CodeDeploy GroupName	Succeeded	Dec 6, 2021 11:15 AM (UTC-5:00)	Dec 6, 2021 11:15 AM (UTC-5:00)	0

7. (Optional) Automate the deployment on Git push

Notes

Development deployment and production deployment should be separated. We need two CI/CD pipelines.

References

1. https://aws.amazon.com/blogs/devops/integrating-with-github-actions-ci-cd-pipeline-to-deploy-a-web-app-to-amazon-ec2/

Implementations

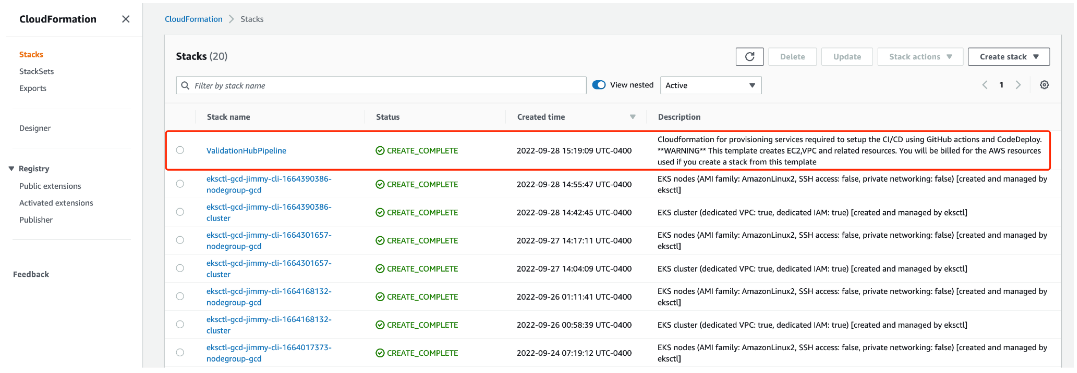
Questions

1. IAM permission
    a. application IAM role (CodeDeploy and deploy instances); it is used in the deployment group of CodeDeploy
    b. GitHub IAM role (access codedeploy and S3)
2. The maximum number of VPCs has been reached.
   Which existing VPC should we use?
3. Backend framework suggestions
    a. Python/Django (easy to implement and much documentation support)

Reproduction

We try to reproduce the procedure in the tutorial in our AWS account.

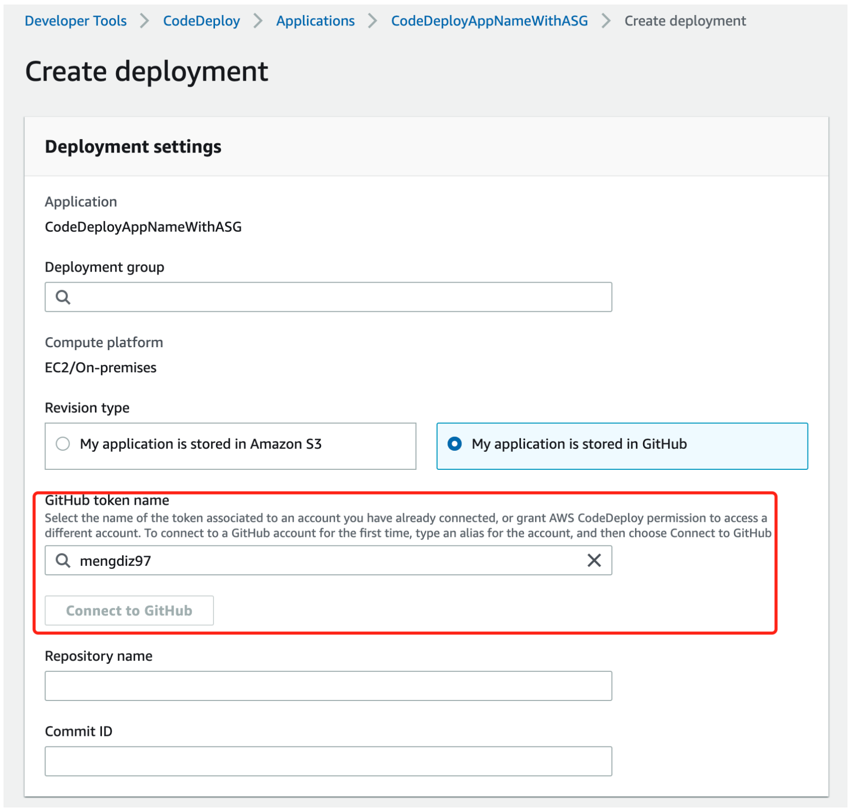
1. Revise configuration files in the example to resolve some issues
   Check the commit.
2. Deploy AWS resources via CloudFormation



	Stack name	Status	Created time	Description
○	ValidationHubPipeline	CREATE_COMPLETE	2022-09-28 15:19:09 UTC-0400	Cloudformation for provisioning services required to setup the CI/CD using GitHub actions and CodeDeploy. **WARNING** This template creates EC2,VPC and related resources. You will be billed for the AWS resources used if you create a stack from this template
○	eksctl-gcd-jimmy-cli-1664390386-nodegroup-gcd	CREATE_COMPLETE	2022-09-28 14:55:47 UTC-0400	EKS nodes (AMI family: AmazonLinux2, SSH access: false, private networking: false) [created and managed by eksctl]
○	eksctl-gcd-jimmy-cli-1664390386-cluster	CREATE_COMPLETE	2022-09-28 14:42:45 UTC-0400	EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]
○	eksctl-gcd-jimmy-cli-1664301657-nodegroup-gcd	CREATE_COMPLETE	2022-09-27 14:17:11 UTC-0400	EKS nodes (AMI family: AmazonLinux2, SSH access: false, private networking: false) [created and managed by eksctl]
○	eksctl-gcd-jimmy-cli-1664301657-cluster	CREATE_COMPLETE	2022-09-27 14:04:09 UTC-0400	EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]
○	eksctl-gcd-jimmy-cli-1664168132-nodegroup-gcd	CREATE_COMPLETE	2022-09-26 01:11:41 UTC-0400	EKS nodes (AMI family: AmazonLinux2, SSH access: false, private networking: false) [created and managed by eksctl]
○	eksctl-gcd-jimmy-cli-1664168132-cluster	CREATE_COMPLETE	2022-09-26 00:58:39 UTC-0400	EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]
○	eksctl-gcd-jimmy-cli-1664017373-nodegroup-gcd	CREATE_COMPLETE	2022-09-24 07:19:12 UTC-0400	EKS nodes (AMI family: AmazonLinux2, SSH access: false, private networking: false) [created and managed by eksctl]

3. Update configuration files with S3 bucket name and region
   Check the commit.
4. Set GitHub secrets

5. Integrate CodeDeploy with GitHub



Developer Tools > CodeDeploy > Applications > CodeDeployAppNameWithASG > Create deployment

Create deployment

Deployment settings

Application
CodeDeployAppNameWithASG

Deployment group

Compute platform
EC2/On-premises

Revision type
My application is stored in Amazon S3 [✓ My application is stored in GitHub]

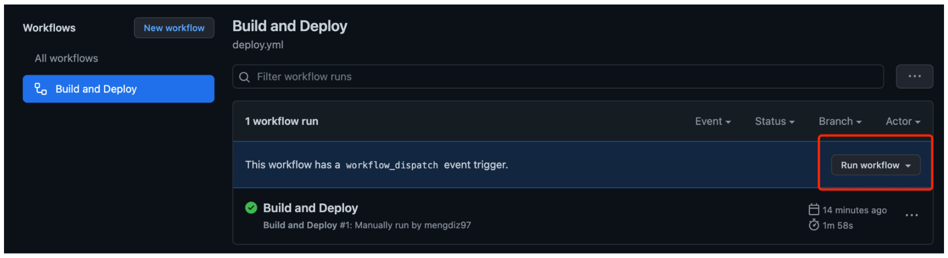
GitHub token name
Select the name of the token associated to an account you have already connected, or grant AWS CodeDeploy permission to access a different account. To connect to a GitHub account for the first time, type an alias for the account, and then choose Connect to GitHub
mengdiz97

Connect to GitHub

Repository name

Commit ID

6. Trigger the GitHub Actions workflow manually



Workflows | New workflow
All workflows
Build and Deploy

Build and Deploy
deploy.yml

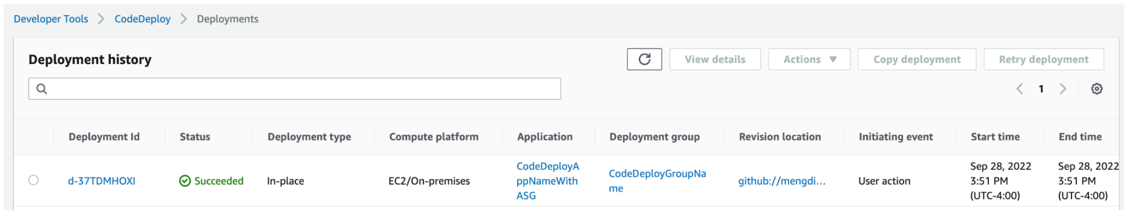
Filter workflow runs

1 workflow run — Event, Status, Branch, Actor

This workflow has a `workflow_dispatch` event trigger. Run workflow

✓ Build and Deploy
Build and Deploy #1: Manually run by mengdiz97 — 14 minutes ago, 1m 58s

The deployment succeeded and we can access the web application via the url



Developer Tools > CodeDeploy > Deployments

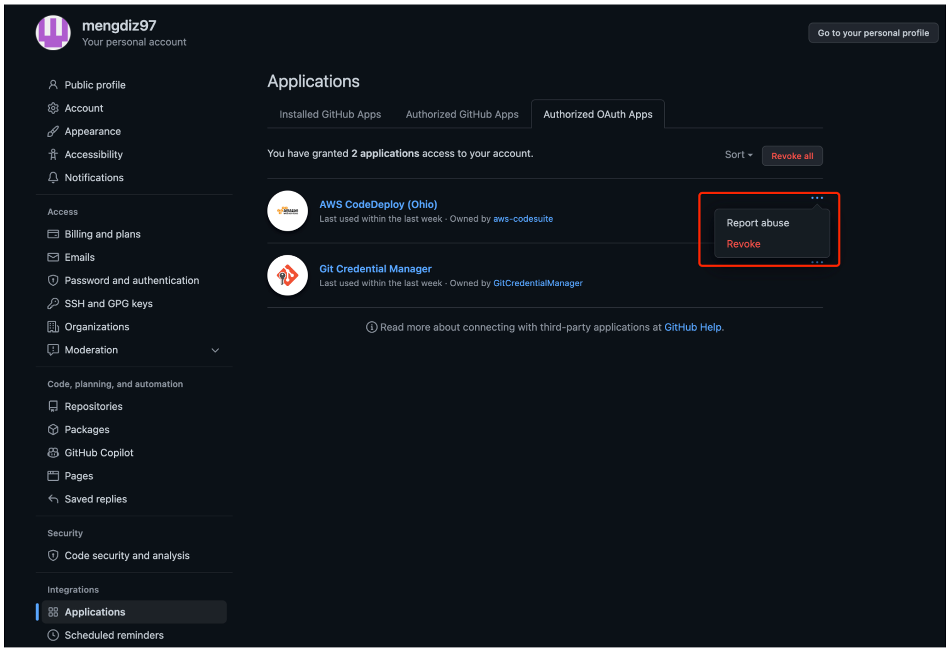
Deployment history

View details | Actions | Copy deployment | Retry deployment

Deployment Id	Status	Deployment type	Compute platform	Application	Deployment group	Revision location	Initiating event	Start time	End time
d-37TDMHOXI	✓ Succeeded	In-place	EC2/On-premises	CodeDeployAppNameWithASG	CodeDeployGroupName	github://mengdi...	User action	Sep 28, 2022 3:51 PM (UTC-4:00)	Sep 28, 2022 3:51 PM (UTC-4:00)

7. Clean up
   a. empty S3 bucket
   b. delete stack on CloudFormation console

c. revoke interaction between CodeDeploy and GitHub



Applications

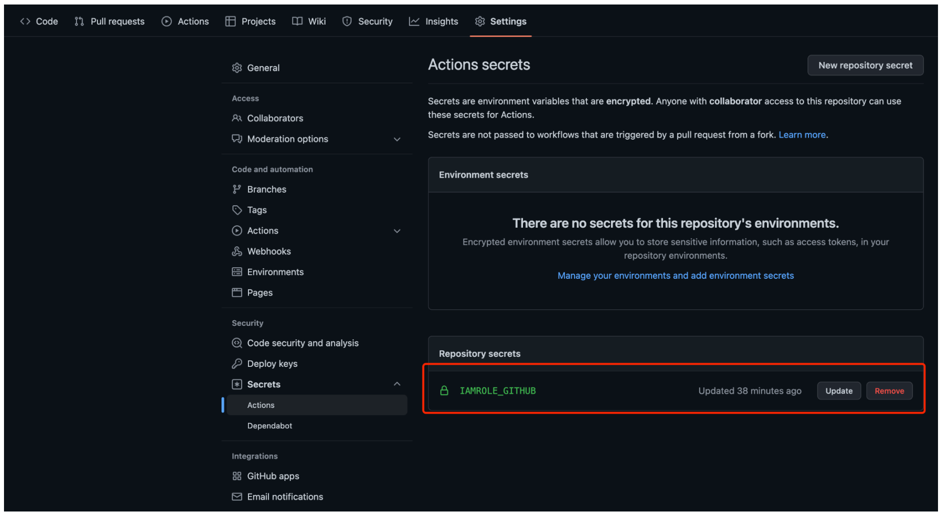
You have granted **2 applications** access to your account.

AWS CodeDeploy (Ohio)
Last used within the last week · Owned by aws-codesuite

Git Credential Manager
Last used within the last week · Owned by GitCredentialManager

Read more about connecting with third-party applications at GitHub Help.

d. delete GitHub secret



Actions secrets

Secrets are environment variables that are **encrypted**. Anyone with **collaborator** access to this repository can use these secrets for Actions.

Secrets are not passed to workflows that are triggered by a pull request from a fork. Learn more.

Environment secrets

There are no secrets for this repository's environments.

Encrypted environment secrets allow you to store sensitive information, such as access tokens, in your repository environments.

Manage your environments and add environment secrets

Repository secrets

`IAMROLE_GITHUB` Updated 38 minutes ago

Useful resources

1. GitHub Actions (related files in .github/):
   https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions
2. Actions Marketplace:
   https://github.com/marketplace?type=actions
3. AppSpec hooks (related files in aws/scripts/ and appspec.yml)
   https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-hooks.html
4. CloudFormation template (related files in cloudformation/)

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-guide.html

Next steps

Step	Effort estimation	Potential risk
Incorporate the pipeline with our project implementation	2 story points	Much time needed to read documentations to adapt configuration files
Deploy two pipelines for development and production respectively	1 story point	

Migration to Python/Django Application

1. CloudFormation template
    a. Utilize Ubuntu 18.04 AMI because CodeDeploy agent is not compatible with Ubuntu 20.04 (https://stackoverflow.com/questions/62286857/aws-codedeploy-agenten-on-ubuntu-20-0lts-ruby-errors) and change LaunchConfig accordingly (install codedeploy agent and SSM agent)
2. Github Workflow deploy.yml
    a. zip code and upload to S3 following https://www.freecodecamp.org/news/how-to-setup-a-ci-cd-pipeline-with-github-actions-and-aws/
3. appspec hooks
    a. appspec.yml destination https://docs.aws.amazon.com/codedeploy/latest/userguide/reference-appspec-file-structure-files.html
    b. after_install.sh
    Secrets downloaded from S3 currently following https://stackoverflow.com/questions/30490745/how-to-specify-sensitive-environment-variables-at-deploy-time-with-elastic-beans
    Do not use MySQL database because MySQL is not in the same VPC currently
    c. application_start.sh
    Cannot `source ~/.bashrc` in bash scripts (https://askubuntu.com/questions/64387/cannot-successfully-source-bashrc-from

-a-shell-script) and need to use another way (https://stackoverflow.com/questions/19331497/set-environment-variables-from-file-of-key-value-pairs)
Utilize `screen` command to run a background process for the Django server and may use docker later
(https://stackoverflow.com/questions/67453523/run-a-new-shell-process-for-django-runserver-command-on-aws-codedeploy)

4. change to use AWS secrets manager
    a. give instances the permission to access AWS secrets manager
    https://docs.aws.amazon.com/secretsmanager/latest/userguide/auth-and-access_examples.html
    b. retrieve secrets in Django settings.py
    notice that a string is returned instead of a json object
    notice expectation raising

    debug in codedeploy to exit with error immediately
    https://stackoverflow.com/questions/35448125/how-to-make-aws-codedeploy-return-an-error-when-some-of-appspec-hooks-fails
5. connect the instance to the database
config in the launch config (assign RDSSecurityGroup to the instance)
6. start application
    a. screen command with log output
    https://stackoverflow.com/questions/14208001/save-screen-program-output-to-a-file
    b. ip address
    should listen to all ipv4 addresses (0.0.0.0); otherwise only listen on localhost (127.0.0.1) and ip forwarding has to be used (visit localhost:8080 on local machine and forward the request to the server)

```
python3 manage.py runserver 0.0.0.0:8080
```

    c. allowed_hosts
    need to add allowed host
    https://stackoverflow.com/questions/57545934/you-may-need-to-add-u127-0-0-1-to-allowed-hosts

    server ip & allowed hosts
    https://stackoverflow.com/questions/16676314/should-server-ip-address-be-in-allowed-hosts-django-setting

Optimizations

1. use and connect with existing VPC, subnets, and database
2. use a pre-deployed S3 bucket to avoid so many commits

TODO

Instances in the auto scaling group are currently deployed in public subnets to make essential environment configuration possible.

Discussion

1. secrets
   https://stackoverflow.com/questions/62731665/django-is-it-ok-to-load-secrets-passwords-dynamic-values-from-a-cloud-serv
   a. option 1: upload to S3 and download to instance
   b. option 2: AWS secrets manager (safest way)
   c. option 3: .env file with S3
2. Elastic Beanstalk
   Seems like a higher-level abstraction than Auto-scaling group and CodeDeploy (https://stackoverflow.com/questions/47217570/automation-using-aws-elastic-beanstalk-vs-aws-codedeploy)

   https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-django.html#python-django-configure-for-eb

   https://dev.to/rmiyazaki6499/deploying-a-production-ready-django-app-on-aws-1pk3#setting-up-the-project-on-the-remote-server