

AWS Continuous Integration

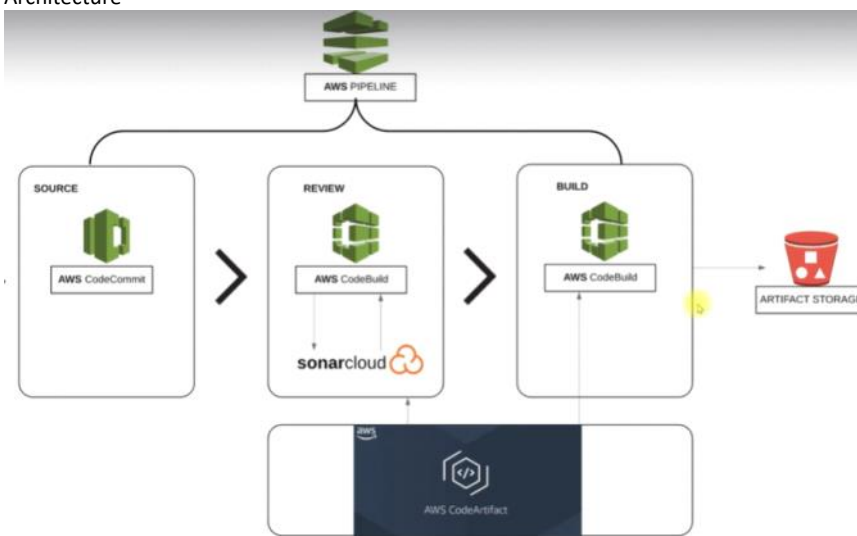
09 June 2024 10:12

Continuous Integration servers have extra overhead to manage the servers. With cloud services, we can set up these services that will have short MTTR, fault isolation, agile, no human intervention, and no ops

AWS Services used

- CodeCommit
- CodeArtifact
- CodeBuild
- CodePipeline
- CodeDeploy
- SonarCloud
- Checkstyle
- CodePipeline
- SNS
- S3

Architecture



Flow of execution

- Login to AWS account
- Code commit
 - o Create codecommit repo
 - o Create IAM user with codecommit policies for Code commit
 - o Generate ssh keys locally
 - o Exchange keys with IAM user
 - o Push source code from github repo to codecommit repository and push
- Code artifact
 - o Create IAM user with code artifact access
 - o Install AWS CLI configure
 - o Export auth token
 - o Update settings.xml file in source code top level directory with below details
 - o Update pom.xml file with repo details
- Sonar cloud
 - o Create sonar cloud account
 - o Generate token
 - o Create SSM parameters with sonar details
 - o Create Build project
 - o update codebuild role to access SSM parameterstore
- Create notifications for sns
- Build project
 - o Update pom.xml with artifact version with timestamp
 - o Create variables in SSM ==> parameterstore
 - o Create build project
 - o Update codebuild role to access SSM parameterstore
- Create pipeline
 - o Codecommit
 - o Testcode
 - o Build

- Deploy to s3 bucket
- Test pipeline

Note: in this project, it is assumed that one has to the following

- AWS account with IAM user with administrative access.
- Sonarcloud
- Github account with git installed on the user's CLI

Step 1: create CodeCommit repo and switch from github repo to codecommit

- I logged into my AWS account and created codecommit repo

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

Repository name

100 characters maximum. Other limits apply.

Description - optional

1,000 characters maximum

Tags

[Add tag](#)

► Additional configuration

AWS KMS key

☐ Enable Amazon CodeGuru Reviewer for Java and Python - optional

Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository.

A service-linked role will be created in IAM on your behalf if it does not exist.

[Cancel](#) [Create](#)

- Create IAM user with custom policy for codecommit access

Specify user details

Step 1: Specify user details

Step 2: [Set permissions](#)

Step 3: [Review and create](#)

User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, -, @, _ - (hyphen)

☐ Provide user access to the AWS Management Console - optional

If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center](#).

❗ If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

[Cancel](#) [Next](#)

Set permissions

Step 1: [Specify user details](#)

Step 2: Set permissions

Step 3: [Review and create](#)

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☐ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

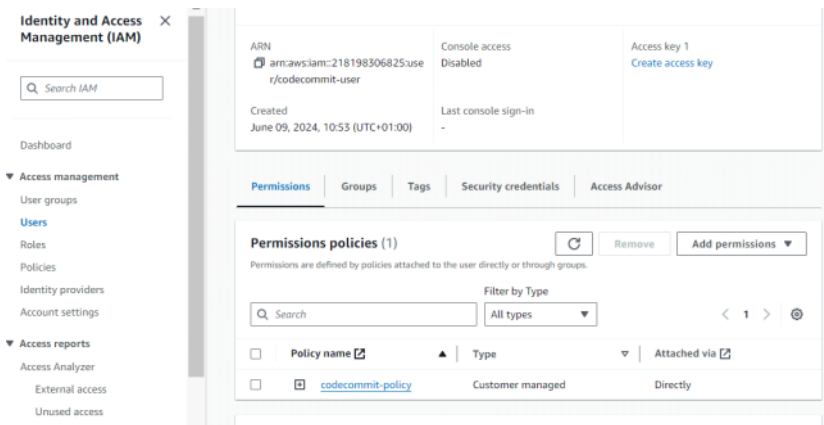
Permissions policies (1203)

Choose one or more policies to attach to your new user.

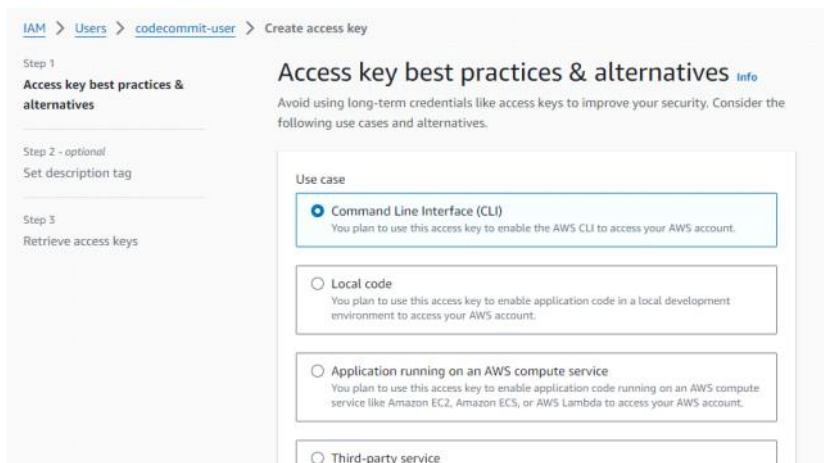
Filter by Type

[Refresh](#) [Create policy](#)

- Attach policies by giving specific commit policy to the user. When prompted, specify the ARN of the account by specifying the name of the codecommit repo (myproject-repo, in this case). Select the policy and attach it the codecommit-user



- After the creation of the user, Create SSH keys by going to the Security Credential tab of the user, scrolling down and clicking the *create access key* button.
- Give CLI access. Agree to understanding the risk at the bottom of the page and click Next



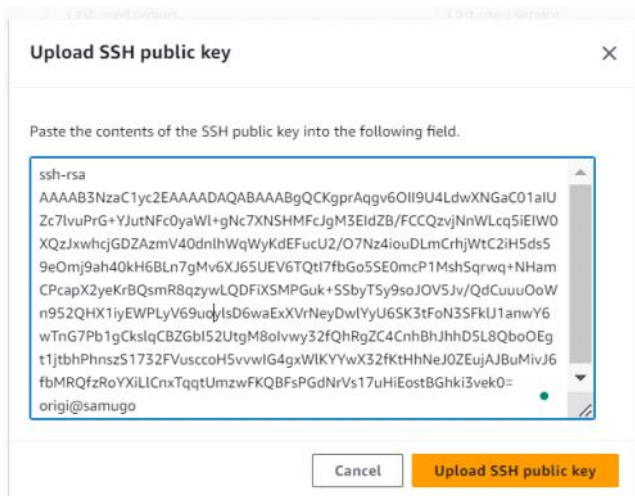
- On the next screen, I just clicked on create access key and later downloaded the access key
- I then configured my terminal with the access key and secret keys downloaded from the previous step

```
origi@samugo MINGW64 /
$ aws configure
AWS Access Key ID [*****EQOD]: AKIATFTNB6AE6NVX63XO
AWS Secret Access Key [*****LMPH]: qgjbpXoaEfNKuJtQf5IuA4vGa4T62A98ni
u384AH
Default region name [eu-west-2]: us-east-1
Default output format [json]: json
```

- Next I generated the ssh keys using the ssh-keygen command

```
origi@samugo MINGW64 /
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/origi/.ssh/id_rsa): /c/Users/origi
/.ssh/myproject-repo_rsa
The key fingerprint is: saved in /c/Users/origi/.ssh/myproject-repo_rsa
SHA256:AzA3S/Mm8ahijyYiqLRSt4PrMdBVBGDF1WJBXUCl0cc origi@samugo
The key's randomart image is: saved in /c/Users/origi/.ssh/myproject-repo_rsa
+---[RSA 3072]-----+
|.o.o= B+=++o
|. . .=.0ooo
|..*.=E
|. .+.
|o o . S
|oo.+ .
|Booo..
|*=o o
|ooo. .
+---[SHA256]-----+
```

- I then cat the public key, copied and saved it in the code commit repo



- Next I created an ssh config file to be used when accessing different git repos. For this repo, the content of this config references the repo as thus, where the User is the access key created in IAM and the identity file, the path to the private ssh key:

```
Host git-codecommit.*.amazonaws.com
  User APKATFTNB6AE2C6PSN7R
  IdentityFile ~/.ssh/myproject-repo_rsa
```

This verifies connectivity:

```
origi@samugo MINGW64 ~/OneDrive - University of the People/it_trainings/aws_solutions_architect/cloud-project-5/vprofile-project (ci-aws)
$ ssh git-codecommit.us-east-1.amazonaws.com
The authenticity of host 'git-codecommit.us-east-1.amazonaws.com (52.94.233.146)' can't be established.
RSA key fingerprint is SHA256:eLMY1j0DKA4uvDZcl/KgtIayZANwX6t8+8isPtotBoY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'git-codecommit.us-east-1.amazonaws.com' (RSA) to the list of known hosts.
You have successfully authenticated over SSH. You can use Git to interact with AWS CodeCommit. Interactive shells are not supported. Connection to git-codecommit.us-east-1.amazonaws.com closed by remote host.
Connection to git-codecommit.us-east-1.amazonaws.com closed.
```

The next major step is to migrate my github repo to codecommit

- Created a folder and cloned the github repo (<https://github.com/hkhcoder/vprofile-project.git>) in the folder
- Checked out all the branches using bash script

```
origi@samugo MINGW64 ~/OneDrive - University of the People/it_trainings/aws_solutions_architect/cloud-project-5/vprofile-project (ci-aws)
$ git branch -a
* ci-aws
main
remotes/origin/HEAD -> origin/main
remotes/origin/aws-LiftAndShift
remotes/origin/aws-refactor
remotes/origin/cd-aws
remotes/origin/ci-aws
remotes/origin/ci-jenkins
remotes/origin/containers
remotes/origin/docker
remotes/origin/main
remotes/origin/terraform-eks

origi@samugo MINGW64 ~/OneDrive - University of the People/it_trainings/aws_solutions_architect/cloud-project-5/vprofile-project (ci-aws)
$ git branch -a | grep remotes | grep -v HEAD | cut -d / -f3 > /tmp/br

origi@samugo MINGW64 ~/OneDrive - University of the People/it_trainings/aws_solutions_architect/cloud-project-5/vprofile-project (ci-aws)
$ for i in `cat /tmp/br`; do git checkout $i; done
Switched to a new branch 'aws-LiftAndShift'
branch 'aws-LiftAndShift' set up to track 'origin/aws-LiftAndShift'.
Switched to a new branch 'aws-refactor'
branch 'aws-refactor' set up to track 'origin/aws-refactor'.
Switched to a new branch 'cd-aws'
branch 'cd-aws' set up to track 'origin/cd-aws'.
Switched to branch 'ci-aws'
Your branch is up to date with 'origin/ci-aws'.
Switched to a new branch 'ci-jenkins'
branch 'ci-jenkins' set up to track 'origin/ci-jenkins'.
Switched to a new branch 'containers'
branch 'containers' set up to track 'origin/containers'.
Switched to a new branch 'docker'
branch 'docker' set up to track 'origin/docker'.
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
Switched to a new branch 'terraform-eks'
branch 'terraform-eks' set up to track 'origin/terraform-eks'.
```

```

origi@samugo MINGW64 ~/OneDrive - University of the People/it_trainings/aws_solutions_architect/cloud-project-5/vprofile-project (terraform-eks)
$ git branch -a
  aws-LiftAndShift
  aws-refactor
  cd-aws
  ci-aws
  ci-jenkins
  containers
  docker
  main
* terraform-eks
remotes/origin/HEAD -> origin/main
remotes/origin/aws-LiftAndShift
remotes/origin/aws-refactor
remotes/origin/cd-aws
remotes/origin/ci-aws
remotes/origin/ci-jenkins
remotes/origin/containers
remotes/origin/docker
remotes/origin/main
remotes/origin/terraform-eks

```

- Switch remote repository -->
 - o git remote set-url origin ssh://commit-url OR
 - o git rm origin and git add origin ssh://commit-url

```

origi@samugo MINGW64 ~/OneDrive - University of the People/it_trainings/aws_solutions_architect/cloud-project-5/vprofile-project (terraform-eks)
$ git remote set-url origin ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/myproject-repo

origi@samugo MINGW64 ~/OneDrive - University of the People/it_trainings/aws_solutions_architect/cloud-project-5/vprofile-project (terraform-eks)
$ git remote -v
origin  ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/myproject-repo (fetch)
origin  ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/myproject-repo (push)

```

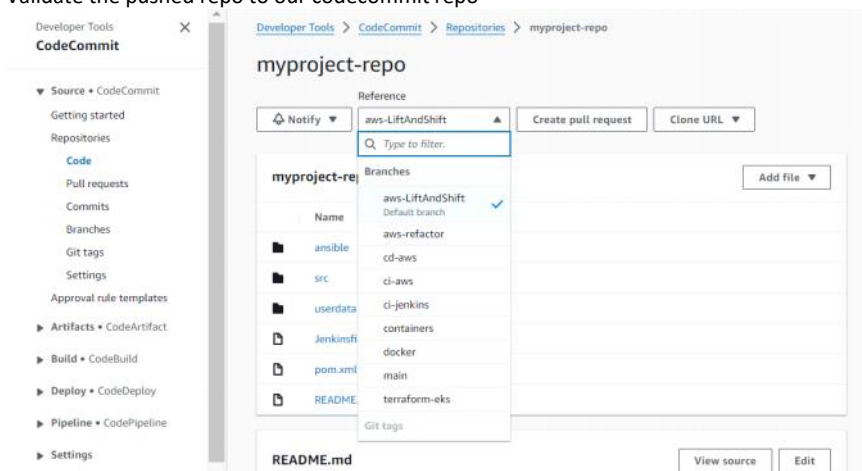
- Push checked out branches to new repo

```

origi@samugo MINGW64 ~/OneDrive - University of the People/it_trainings/aws_solutions_architect/cloud-project-5/vprofile-project (terraform-eks)
$ git push origin --all
Enumerating objects: 487, done.
Counting objects: 100% (487/487), done.
Delta compression using up to 8 threads
Compressing objects: 100% (277/277), done.
Writing objects: 100% (487/487), 7.69 MiB | 2.16 MiB/s, done.
Total 487 (delta 186), reused 487 (delta 186), pack-reused 0
remote: Validating objects: 100%
To ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/myproject-repo
* [new branch]      aws-LiftAndShift -> aws-LiftAndShift
* [new branch]      aws-refactor -> aws-refactor
* [new branch]      cd-aws -> cd-aws
* [new branch]      ci-aws -> ci-aws
* [new branch]      ci-jenkins -> ci-jenkins
* [new branch]      containers -> containers
* [new branch]      docker -> docker
* [new branch]      main -> main
* [new branch]      terraform-eks -> terraform-eks

```

Validate the pushed repo to our codecommit repo



Step 2: create code artifact repository

Code artifact repo is used to store dependencies for the build tools, like maven. The reason for doing this that many projects will have requirements to use dependencies they control rather than downloading them from the internet. Even if one is not using aws for the build process but using jenkins, you can still point the build tool to code artifact repo

- Search for codeartifact on the aws console and select *create repository*.

The screenshot shows the 'Create repository' wizard in the AWS CodeArtifact console. It is at Step 1 of 3. The 'Repository name' field is filled with 'myproject-maven-artifact'. Below it, a 'Repository description - optional' text area is empty. At the bottom, the 'Public upstream repositories - optional' dropdown menu is open, showing 'maven-central-store' as the selected option. The 'Next' button is highlighted in orange.

- I chose maven central store (since the build process will be done with maven)
- Next, I selected that the repo is for my own account. I also gave a domain and clicked on Next --> Create.

The screenshot shows the 'Domain' configuration page. It explains that a domain is a grouping of repositories. Under 'AWS account', 'This AWS account (218198306825)' is selected. A message states 'No domains in this account' and suggests creating a domain. The 'Domain name' field is filled with 'iamugo'. The 'Domain URL' is automatically generated as 'iamugo-218198306825.d.codeartifact.us-east-1.amazonaws.com'. At the bottom, the 'Next' button is highlighted in orange.

The variable, CODEARTIFACT_AUTH_TOKEN will be exported

Step 3: create sonarcloud account with github and configure with

-

The screenshot shows the SonarCloud web interface. The user 'Samson Egbuchulam' is logged in, with a public profile. The navigation menu includes 'Projects', 'Quality Profiles', 'Rules', 'Quality Gates', 'Members', 'Billing & Upgrade', and 'Administration'. The 'Security' tab is not visible in the current view.

- I went to the *Security* tab and generated a token

Security

If you want to enforce security by not providing credentials of a real SonarCloud user to invoke web services, you can provide a User Token as a replacement of the user login. security of your installation by not letting your analysis user's password going through yc

Generate Tokens

 Generate Token

- Copy the token to a safe location
- Click on the *Organizations* tab to access your organization. If you don't have an organization, create one manually by clicking on the '+' sign at the top right corner of the page.
- From the same '+' sign, click *Analyze New Project*

Analyze projects

Manual setup is not recommended, and leads to missing features like appropriate setup of your project or analysis feedback in the Pull Request. We recommend to [import your projects](#)

Organization

[Create another organization](#)

Display Name * ?

Up to 255 characters

Project Key * ?

Up to 400 characters. All letters, digits, dash, underscore, period or colon.

Project visibility

☒ Public

Anyone will be able to browse your source code and see the result of your analysis.

☐ Private **PAID PLAN**

Only members of the organization will be able to browse your source code and see the result of your analysis.

- On the next screen, I chose 'Previous Version' to enable the analysis happen on any new version of the code. Click create
- Next I copied the Project key and Organization key to my notepad

AWS Parameter store

AWS Parameter stores are used to store the values of sensitive variables away from the public. This is because I am going to be using these variables in the project and storing them in the code pose a serious threat; hence the reason why I am using parameter store. Secrets manager could be used but it costs a lot.

The parameters are mentioned in the build_spec file

The Parameter Store is a service under the AWS Secrets Manager .

- Search for parameter store on the AWS console. Create new parameter
- Create entries in the parameter store, each for LOGIN, HOST, Organization and Project. The type is string for all except LOGIN which has a type of Secure String

| | |
|-----------------------|--|
| Sonar build-spec file | Sample of a parameter in parameter store |
|-----------------------|--|

The screenshot displays the configuration of an AWS CodeBuild project. On the left, the 'sonar_buildspec.yml' file is shown with the following content:

```

1 version: 0.2
2 env:
3   parameter-store:
4     LOGIN: LOGIN
5     HOST: HOST
6     Organization: Organization
7     Project: Project
8     #CODEARTIFACT_AUTH_TOKEN: CODEARTIFACT_AUTH_TOKEN
9   phases:
10    install:
11      runtime-versions:
12        java: corretto11
13      commands:

```

On the right, the AWS Systems Manager Parameter Store 'Organization' parameter details are shown:

| Name | Description |
|--|--|
| Organization | - |
| ARN | Data type: text |
| am:aws:ssm:us-east-1:218198306825:parameter/Organization | Last modified user: am:aws:iam::218198306825:user/ugochi |
| Tier: Standard | Last modified date: Sun, 09 Jun 2024 18:35:24 GMT |
| Type: String | Version: 1 |
| Value: myprojectorg | |

Below the parameter details, a table lists 'My parameters':

| Name | Tier | Type | Last modified |
|----------------------------------|----------|--------------|-------------------------------|
| /cdk-bootstrap/hnb659fds/version | Standard | String | Sun, 02 Jun 2024 10:42:55 GMT |
| HOST | Standard | String | Sun, 09 Jun 2024 18:40:33 GMT |
| LOGIN | Standard | SecureString | Sun, 09 Jun 2024 18:44:42 GMT |
| Organization | Standard | String | Sun, 09 Jun 2024 18:35:24 GMT |
| Project | Standard | String | Sun, 09 Jun 2024 18:43:59 GMT |

Step 4: Edit pom.xml, buildspec.yml and settings.xml files with the right information from coder artifact and start CodeBuild project.

CodeBuild is the start of the pipeline to be used.

- The first step is to edit the pom.xml file with the right information
 - o I went to Code artifact and copied the url of the artifact repo. This is used to edit the url section of the build repository in the pom.xml file

The screenshot shows the 'pom.xml' file with the following XML structure:

```

204 <goals>
205   <goal>report</goal>
206 </goals>
207 </execution>
208 </executions>
209 </plugin>
210
211 </plugins>
212 </build>
213 <repositories>
214   <repository>
215     <id>codeartifact</id>
216     <name>codeartifact</name>
217     <url>https://iamugo-218198306825.d.codeartifact.us-east-1.amazonaws.com/maven/maven-central-store/</url>
218   </repository>
219 </repositories>
220 </project>
221

```

- o I also edited the settings file with the url of the artifact

The screenshot shows the 'settings.xml' file with the following XML structure:

```

<profiles>
  <profile>
    <id>default</id>
    <repositories>
      <repository>
        <id>codeartifact</id>
        <url>https://iamugo-218198306825.d.codeartifact.us-east-1.amazonaws.com/maven/maven-central-store/</url>
      </repository>
    </repositories>
  </profile>
</profiles>
<activeProfiles>
  <activeProfile>default</activeProfile>
</activeProfiles>
<mirrors>
  <mirror>
    <id>codeartifact</id>
    <name>domainname--maven-central-store</name>
    <url>https://iamugo-218198306825.d.codeartifact.us-east-1.amazonaws.com/maven/maven-central-store/</url>
  </mirror>
</mirrors>

```

I had to ensure the sonar_buildspec.yml file be in the root directory of the code repo. I renamed it to buildspec.yml

- I then edited the settings.xml file to reflect the right auth_token gotten from code artifact


```

phases:
  install:
    runtime-versions:
      java: corretto11
    commands:
      - cp ./settings.xml /root/.m2/settings.xml
      - export CODEARTIFACT_AUTH_TOKEN=`aws codeartifact get-authorization-token --domain iamugo --domain-owner 21
pre_build:

```

- Saved and pushed the changes to the code commit repo

- Start the CodeBuild project

Project name
myproject-code-analysis

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

► Additional configuration
Description, Build badge, Concurrent build limit, tags

Source Add source

Source 1 - Primary

Source provider
AWS CodeCommit

Repository
myproject-repo

Reference type
Choose the source version reference type that contains your source code.
☒ Branch
☐ Git tag
☐ Commit ID

Branch
Choose a branch that contains the code to build.
ci-aws

Commit ID - optional
Choose a commit ID. This can shorten the duration of your build.

- o The role created during the creation of the build process will not have access to the parameter store. I had to edit the role after the creation of the build process to add a policy for parameter store access.
- o I used ubuntu as the operating system and chose the aws/codebuild/standard:5.0 as the image
- o I also enabled CloudWatch logs
- Next was to edit the roles to add read access for artifact and parameter store to the CodeBuild roles.
- Started the build process

myproject-code-analysis:a91f6d03-e20d-4bf8-9bde-7bedcb0613b7

| Build status | | | | |
|--------------------------------|-----------|--|--|--|
| Status | Initiator | Build ARN | | |
| In progress | ugochi | arn:aws:codebuild:us-east-1:21d/myproject-code-analysis:a91f6d0~7bedcb0613b7 | | |
| Start time | End time | Build number | | |
| Jun 9, 2024 8:40 PM (UTC+1:00) | - | 1 | | |

| Build logs | Phase details | Reports | Environment variables | Build details | Resource utilization |
|-----------------|---------------|---------|-----------------------|--------------------------------|----------------------|
| Name | Status | Context | Duration | Start time | |
| SUBMITTED | Succeeded | - | <1 sec | Jun 9, 2024 8:40 PM (UTC+1:00) | |
| QUEUED | Succeeded | - | <1 sec | Jun 9, 2024 8:40 PM (UTC+1:00) | |
| PROVISIONING | Succeeded | - | 3 secs | Jun 9, 2024 8:40 PM (UTC+1:00) | |
| DOWNLOAD_SOURCE | Succeeded | - | 11 secs | Jun 9, 2024 8:40 PM (UTC+1:00) | |
| INSTALL | Succeeded | - | 16 secs | Jun 9, 2024 8:40 PM (UTC+1:00) | |
| PRE_BUILD | Succeeded | - | 22 secs | Jun 9, 2024 8:40 PM (UTC+1:00) | |
| BUILD | - | - | - | Jun 9, 2024 8:41 PM (UTC+1:00) | |

- The build job takes a while to build but in the end, success

Build logs

Succeeded

Start time: 13 minutes ago

```

22631 [WARNING] * src/test/java/com/visualpathit/account/model/est/MoleTest.java
22632 [WARNING] * src/main/webapp/WEB-INF/views/registration.jsp
22633 [WARNING] * src/main/java/com/visualpathit/account/controller/UserController.java
22634 [WARNING] * src/main/webapp/WEB-INF/views/userUpdate.jsp
22635 [WARNING] * src/test/java/com/visualpathit/account/setup/StandaloneMvcTestViewResolver.java
22636 [WARNING] * src/main/webapp/WEB-INF/appconfig-security.xml
22637 [WARNING] * src/main/java/com/visualpathit/account/service/UserDetailsServiceImpl.java
22638 [WARNING] * src/main/java/com/visualpathit/account/service/SecurityServiceImpl.java
22639 [WARNING] This may lead to missing/broken features in SonarCloud
22640 [INFO] CPD Executor 7 files had no CPD blocks
22641 [INFO] CPD Executor Calculating CPD for 24 files
22642 [INFO] CPD Executor CPD calculation finished (done) | time=27ms
22643 [INFO] Analysis report generated in 2250ms, dir size=669 KB
22644 [INFO] Analysis report compressed in 126ms, zip size=209 KB
22645 [INFO] Analysis report uploaded in 906ms
22646 [INFO] ANALYSIS SUCCESSFUL, you can find the results at: ***/dashboard?id=***
22647 [INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
22648 [INFO] More about the report processing at ***/api/ce/task?id=AYtgoH1z1b88W9NagN
22649 [INFO] Sensor cache published successfully
22650 [INFO] Analysis total time: 49.385 s
22651 [INFO] -----
22652 [INFO] BUILD SUCCESS
22653 [INFO] -----
22654 [INFO] Total time: 01:00 min
22655 [INFO] Finished at: 2023-10-24T07:42:05Z
22656 [INFO] -----
22657

```

- Next, I built the artifact and stored it in an s3 bucket. To do this:
 - o I edited the build_spec file with the right CODEARTIFACT_AUTH_TOKEN (line11) from code artifact. Save and commit

```

aws-files > ! build_buildspec.yml
1 version: 0.2
2 #env:
3   #parameter-store:
4     #CODEARTIFACT_AUTH_TOKEN: CODEARTIFACT_AUTH_TOKEN
5 phases:
6   install:
7     runtime-versions:
8       java: corretto11
9     commands:
10      - cp ./settings.xml /root/.m2/settings.xml
11      - export CODEARTIFACT_AUTH_TOKEN="aws codeartifact g
12 pre_build:
13   commands:
14     - apt-get update
15     - apt-get install -y jq
16     - wget https://d1cdn.apache.org/maven/maven-3/3.9.4/
17     - tar xzvf apache-maven-3.9.4-bin.tar.gz
18     - ln -s apache-maven-3.9.4 maven
19 build:
20   commands:
21     - mvn clean install -DskipTests
22 artifacts:
23   files:
24     - target/**/*.*war
25 discard-paths: yes

```

- o This build job produces the artifact which I will send to the s3 bucket
- o Created a build job for this build_buildspec.yml file. This is same as above but this time, I ensured that the file path (aws-files/build_buildspec.yml) was specified in the buildspec path
- o I had to add the code artifact read policy to the role
- o Start the build job

Step 5: Create S3 bucket to store the artifact and SNS notification for the build pipeline

- Go to the s3 service and created a bucket
- I also created a folder to store the artifact

The screenshot shows the Amazon S3 console. On the left, the 'General purpose buckets' section lists several buckets, including 'myproject-s3-bkt' in the 'us-east-1' region. On the right, the details for 'myproject-s3-bkt' are shown. The 'Objects' tab is active, displaying a list of objects. The first object is 'pipeline-artifact', which is a folder. The console interface includes various navigation and action buttons like 'Copy S3 URI', 'Download', and 'Upload'.

- Next I created an SNS topic

Create topic

Details

Type [Info](#)
Topic type cannot be modified after topic is created

☐ FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

☒ Standard

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

myproject-pipeline-notifications

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

- Created a subscription (email) to the topic. I received a mail to confirm my subscription

Create subscription

Details

Topic ARN

arn:aws:sns:us-east-1:218198306825:myproject-pipeline-notifications

Protocol

The type of endpoint to subscribe

Email

Endpoint

An email address that can receive notifications from Amazon SNS.

samson.egbichulam@gmail.com

After your subscription is created, you must confirm it. [Info](#)

- Confirmed the email using the link sent.

Step 6: create pipeline

- Go to code pipeline. Click create pipeline

Add source stage

Source

Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

AWS CodeCommit

Repository name

Choose a repository that you have already created where you have pushed your source code.

vprofile-code-repo

Branch name

Choose a branch of the repository

ci-aws

Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

☒ Amazon CloudWatch Events (recommended)

Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

☐ AWS CodePipeline

Use AWS CodePipeline to check periodically for changes

Output artifact format

Choose the output artifact format

- I gave the settings seen above, clicked Next
- On the next page, I chose the AWS code commit repo as the build provider and the branch and opted to trigger the pipeline from CloudWatch Events instead. I skipped the deploy section. Create pipeline
- As soon as the pipeline is created, it begins to run. I stopped (Stop Execution button)

Developer Tools > CodePipeline > Pipelines > vprofile-CI-Pipeline

vprofile-CI-Pipeline

Notify Edit Stop execution Clone pipeline Release changes

Source Succeeded

Pipeline execution ID: 90e23e04-cad6-4b44-8f11-a29c0a3f3c7

Source

AWS CodeCommit

Succeeded - Just now

f2944a2d

f2944a2d Source: codeartifact token

Disable transition

Build Stopping

Pipeline execution ID: 90e23e04-cad6-4b44-8f11-a29c0a3f3c7

- I then edited the pipeline to include a stage for the code analysis (action group: the code build job; project name: the code analysis)
- Deployment to s3 is another stage I added at the end of the pipeline (input artifact is the code build job, the product key is the name of the folder - pipeline-artifact I created in s3. I ensured the check mark for *extract before deploy* is checked.
- Saved pipeline
- In the pipeline settings, I created a notification rule to publish to the sns topic I created earlier. For this project, I selected all the events but in real life, not all the notifications will be selected.

Create notification rule

Notification rules set up a subscription to events that happen with your resources. When these events occur, you will receive notifications sent to the targets you designate. You can manage your notification preferences in Settings. [Info](#)

Notification rule settings

Notification name

myproject-ci-notification

Detail type

Choose the level of detail you want in notifications. [Learn more about notifications and security](#)

☒ Full

Includes any supplemental information about events provided by the resource or the notifications feature.

☐ Basic

Includes only information provided in resource events.

Events that trigger notifications

Select none

Select all

Action execution

☒ Succeeded

☒ Failed

☒ Canceled

Stage execution

☒ Started

☒ Succeeded

☒ Resumed

Pipeline execution

☒ Failed

☒ Canceled

☒ Started

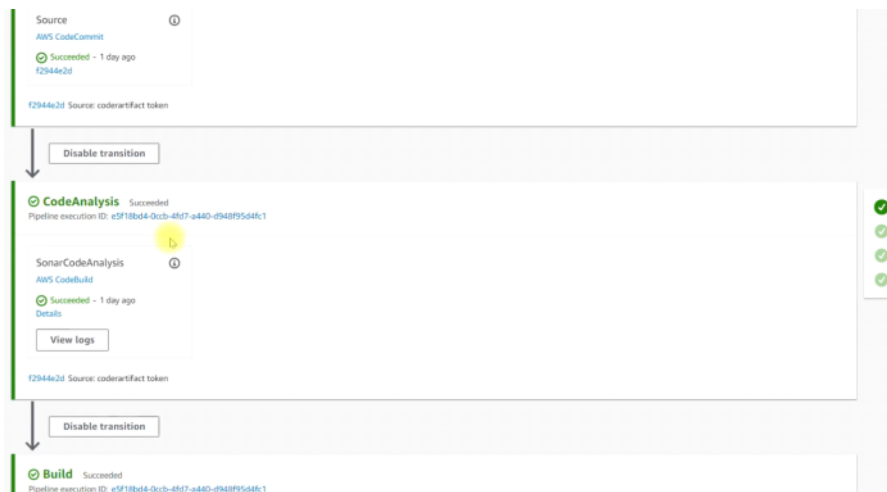
Manual approval

☒ Failed

☒ Needed

☒ Succeeded

- After everything, I clicked on Release change. Another way to test the pipeline is to trigger the pipeline by making a code commit.
- Everything is tied together and the pipeline works!!!



Challenges:

- The decision of which CI tool to use was something I had to consider. Using Jenkins or any other CI tool would have accomplished the job, but would also have come with the complexity of managing the server. I chose AWS CodeBuild because it is a managed service and I only pay for the build time, not for setting it up. In contrast, Jenkins will require me to run an EC2 instance continuously for as long as I need the server running, which is not cost effective. For an organization's build, Jenkins will be better due to the number of build processes that will be going on
- Deciding which image to use during the creation of the code build process. Version 5 was used because it had jdk11, which I needed for this project.