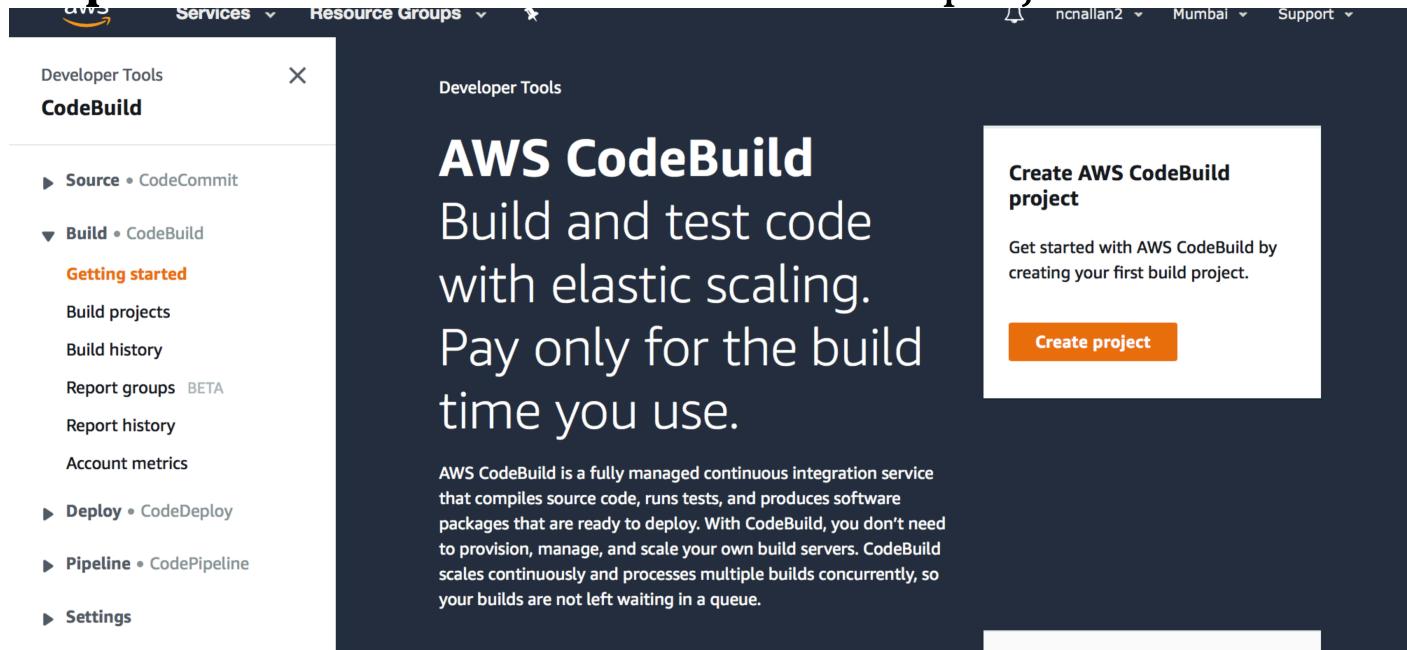


Exercise 04.1:Building a sample project using CodeBuild

Step 1: Go to CodeBuild and click Create project



Step 2: Click create build project and enter/select the following properties

Project name: my-first-webpage

Source: CodeCommit

Repository: my-first-webpage

Reference type: Branch

Branch: master

Environment Image: Managed Image

Operating System: Amazon Linux 2

Runtimes: Standard

Image: Last one

Artifacts:

None

Click 'Create Build Project'

Step 8: Click ‘Start Build’ and then Click “Start Build”. Make sure the Status is “Successful”

The screenshot shows the AWS CodeBuild console. At the top, there is a green banner with the message "Build started" and "You have successfully started the following build: my-first-webpage:6ecd3ab2-411b-487b-9566-d45a4dd51a71". Below the banner, the navigation path is shown: Developer Tools > CodeBuild > Build projects > my-first-webpage > my-first-webpage:6ecd3ab2-411b-487b-9566-d45a4dd51a71. The main title of the build is "my-first-webpage:6ecd3ab2-411b-487b-9566-d45a4dd51a71". Below the title are two buttons: "Stop build" (gray) and "Retry build" (orange). A "Build status" section follows, containing a table with three columns: Status (Succeeded), Initiator (root), and Build ARN (arn:aws:codebuild:ap-south-1:039121988859:build/my-first-webpage:6ecd3ab2-411b-487b-9566-d45a4dd51a71).

Step 9: Go through the buildSpec.yaml

Ref: <https://docs.aws.amazon.com/codebuild/latest/userguide/build-spec-ref.html>

Step 11: Go to CodeCommit and in the index.html remove “Congratulations” and put “Error”. Commit the changes and start a new Build. This time the build should fail as the build is checking for a word “Congratulations” in the script

Ref: <https://docs.aws.amazon.com/codebuild/latest/userguide/build-env-ref-env-vars.html>

Passing Environmental variables to CodeBuild

Step 1: Add a printenv statement in buildspec.yml of the codecommit project and submit the changes.

```
3 phases:
4   install:
5     runtime-versions:
6       nodejs: 10
7     commands:
8       - printenv
9       - echo "installing something"
L0   pre_build:
L1     commands:
L2       - echo "we are in the pre build phase"
L3     . . .
```

Step1.1:

Replace the “Error” in index.html with “Congratulations” to make our build successful.

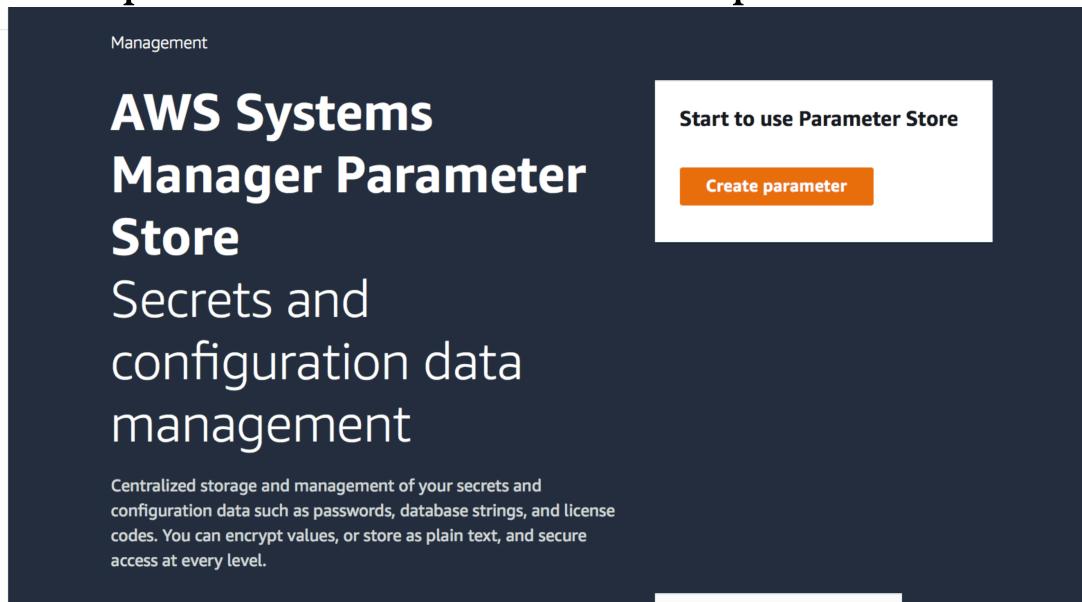
Step 1.2 :In CodeBuild, Click Start Build of an existing code build project and go to the environment variables section at the bottom and observe the System defined Env variables

Step2: Add a couple of parameters as shown below

▼ Environment variables override

Environment variables			
Name	Value	Type	
DB_URL	dburl.com	Plaintext	<input type="button" value="Remove"/>
DBURL_PASSWORD	/Prod/DB_PASSWORD	Secrets Manager	<input type="button" value="Remove"/>
<input type="button" value="Add environment variable"/>			
<input type="button" value="Create parameter"/>			

Step 3: Type SSM in services and open AWS Systems Manager in a new window. Find parameter store in the left panel and click it



Step 4: Click Create Parameter and enter name = /Prod/DBPassword , select Secure String and value ="mypassword". Click Create parameter

Step 5: Provide the access to ssm in IAM policy . Add a policy named “AmazonSSMReadOnlyPolicy” to role that is already created during the build project creation process.

Attach Permissions

Policy name		Type	Used as
<input type="checkbox"/>	AmazonSSMFullAccess	AWS managed	None
<input type="checkbox"/>	AmazonSSMMaintenanceWindowRole	AWS managed	None
<input type="checkbox"/>	AmazonSSMManagedInstanceCore	AWS managed	None
<input checked="" type="checkbox"/>	AmazonSSMReadOnlyAccess	AWS managed	None
<input type="checkbox"/>	AWSResourceAccessManagerFullAccess	AWS managed	None
<input type="checkbox"/>	AWSResourceAccessManagerReadOnlyAccess	AWS managed	None
<input type="checkbox"/>	AWSResourceAccessManagerResourceShareParticipantAccess	AWS managed	None

Cancel

Attach policy

Step 6: Come back to CodeBuild and click Start Build. You should be able to see the DB Parameters in the build logs

```
94 PATH=~/dotnet/tools:/go/bin:/usr/local/go/bin:/php/bin:/usr/local/php/bin:/root/.lo
 :/usr/local/bin:/root/.rbenv/shims:/usr/local/rbenv/bin:/usr/local/rbenv/shims:/usr/l
 in:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/codebuild/user/bin
95 MAVEN_HOME=/opt/maven
96 SRC_DIR=/usr/src
97 GPG_KEYS=CBAF69F173A0FEA4B537F470D66C9593118BCCB6 F38252826ACD957EF380D39F2F7956BC5DA
98 PHP_CPPFLAGS=-fstack-protector -fpic -fpie -O2
99 AWS_DEFAULT_REGION=ap-south-1
100 POWERSHELL_DOWNLOAD_SHA=7605F347F543880A90C1F67305C802562384A4DCDA9E797D501E7BBF67464
101 PWD=/codebuild/output/src902148760/src/git-codecommit.ap-south-
1.amazonaws.com/v1/repos/pharmacy-app
102 DB_PASSWORD=***
103 CODEBUILD_BUILD_IMAGE=aws/codebuild/amazonlinux2-aarch64-standard:1.0
104 DOTNET_SDK_VERSION=3.0.100
105 DOCKER_18_PATH=/usr/local/bin/docker18
106 JAVA_HOME=/usr/lib/jvm/java-1.8.0-amazon-corretto.aarch64
107 GRADLE_DOWNLOADS_SHA256=14cd15fc8cc8705bd69dcfa3c8fefb27eb7027f5de4b47a8b279218f76895
5.4.1\n336b6898b491f6334502d8074a6b8c2d73ed83b92123106bd4bf837f04111043 4.10.3
108 CODEBUILD_FE_REPORT_ENDPOINT=https://codebuild.ap-south-1.amazonaws.com/
109 LANG=C.UTF-8
110 JDK_VERSION=11.0.4.11.1
111 CODEBUILD_KMS_KEY_ID=arn:aws:kms:ap-south-1:039121988859:alias/aws/s3
```

Uploading Artifacts to S3:

<https://docs.aws.amazon.com/codebuild/latest/userguide/build-spec-ref.html>

Step 1: Refer to the artifacts section of the above reference URL and copy the section in to the buildspec.yml. This will upload the artifacts to pre-defined bucket in S3

Step 2: Go to CodeCommit and click your repository and edit the buildspec.yml. Add the following highlighted code at the end of the file. Commit the changes

```
1 version: 0.2
2
3 phases:
4   install:
5     runtime-versions:
6       nodejs: 10
7     commands:
8       - printenv
9       - echo "installing something"
10  pre_build:
11    commands:
12      - echo "we are in the pre build phase"
13  build:
14    commands:
15      - echo "we are in the build block"
16      - echo "we will run some tests"
17      - grep -Fq "Congratulations" index.html
18  post_build:
19    commands:
20      - echo "we are in the post build phase"
21 artifacts:
22   files:
23     - '**/*'
24   name: webapp-artifacts
25
```

Step 3: Go to CodeBuild and select the build. Click Edit and select Artifacts

Step 4: In a separate web browser tab go to S3 and create a new bucket(ex: **mycodebuilds3artifacts**) to store the artifacts

Step 5: Refresh the code build page to find the newly created bucket under the field “Bucket name” and select the bucket

Step 6: Select Zip format for Artifacts packaging and click Update Artifacts button

Step 7: Click the start build button and monitor the logs. You should see in S3 the artifacts uploaded to S3 bucket

				(UTC+5:30)	(UTC+5:30)
BUILD	✔ Succeeded	-	<1 sec	Jul 17, 2020 11:30 AM (UTC+5:30)	Jul 17, 2020 11:30 AM (UTC+5:30)
POST_BUILD	✔ Succeeded	-	<1 sec	Jul 17, 2020 11:30 AM (UTC+5:30)	Jul 17, 2020 11:30 AM (UTC+5:30)
UPLOAD_ARTIFACTS	✔ Succeeded	-	<1 sec	Jul 17, 2020 11:30 AM (UTC+5:30)	Jul 17, 2020 11:30 AM (UTC+5:30)
FINALIZING	✔ Succeeded	-	4 secs	Jul 17, 2020 11:30 AM	Jul 17, 2020 11:30 AM

Artifacts in S3 Bucket

mycodebuilds3artifacts

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with tabs: Overview (selected), Properties, Permissions, Management, and Access points. Below the navigation bar is a search bar with placeholder text "Type a prefix and press Enter to search. Press ESC to clear." Underneath the search bar are several action buttons: Upload, Create folder, Download, and Actions. To the right of these buttons, it shows the region "Asia Pacific (Mumbai)" and a refresh icon. The main content area displays a table of artifacts. The table has columns: Name, Last modified, Size, and Storage class. There is one item listed: "my-code-build-project" was last modified on Jul 17, 2020 at 11:30:33 AM GMT+0530, has a size of 8.9 KB, and is stored in the Standard storage class. The table also includes header rows for sorting and filtering.

	Name	Last modified	Size	Storage class
<input type="checkbox"/>	my-code-build-project	Jul 17, 2020 11:30:33 AM GMT+0530	8.9 KB	Standard

CloudWatch Logs / Metrics/ Events:

Step1: Go to CloudWatch service and click LogGroups. Select the codebuild project and you can see the logs for each build.

Step2: Click Metrics in CloudWatch. You can see Code Build metrics on the right side. Navigate and see varios details of metrics

The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a sidebar with links like Insights, Metrics (which is selected), Events, Rules, Event Buses, ServiceLens, Service Map, Traces, Container Insights (BETA), Resources, and Performance Monitoring. The main area has tabs for All metrics, Graphed metrics (selected), Graph options, and Source. A search bar is present. Below it, a section titled '122 Metrics' lists four categories: CodeBuild (28 Metrics), EBS (18 Metrics), EC2 (34 Metrics), and Events (5 Metrics).

Step 3: Click Events and click Get started. Click Schedule and select a frequency

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

Event Pattern [i](#) Schedule [i](#)

Fixed rate of Minutes

Cron expression

[Learn more](#) about CloudWatch Events schedules.

► Show sample event(s)

Step 4: Click Add target on the right and select Code Build Projects. Click the Learn More link to find the format of the ARN of build project which is of the form `arn:aws:codebuild:region-ID:account-ID:project/project-name` and click Configure. Your build will now be scheduled as per the frequency

Step 5: Select the Event pattern on the left . Select Code Build in Service name and select the Event type. On the right panel add the required target to invoke like SNS or Lambda

The screenshot shows the 'Create rule' interface for AWS Lambda. It has two main sections: 'Event Source' on the left and 'Targets' on the right.

Event Source: This section allows you to define an event pattern or a schedule. It includes a dropdown for 'Build event pattern to match events by service', a 'Service Name' dropdown set to 'CodeBuild', and an 'Event Type' dropdown set to 'CodeBuild Build State Change'. There are also options for 'Any state' or 'Specific state(s)'. At the bottom, there are buttons for 'Event Pattern Preview', 'Copy to clipboard', and 'Edit'.

Targets: This section lists targets for the event. It currently shows an 'SNS topic' target named 'my-codecommit-topic'. A button labeled 'Add target*' is available to add more targets. A note says: 'Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.'

Code Commit Pull requests with Code Build Ref:

<https://aws.amazon.com/blogs/devops/validating-aws-codecommit-pull-requests-with-aws-codebuild-and-aws-lambda/>