

Group4 Lab 5- EBS with Code Pipeline

Akilandeshwari, Srinivasan [451036]

Course Code – CLCM 3102

Due Date – Oct 16, 2023

Description

Creating the AWS Elastic Beanstalk which is connected to Code Pipeline to automatically deploy the code from GitHub to AWS Elastic Beanstalk.

Preparation

We need to create:

- GitHub repository
- AWS Elastic Beanstalk
- AWS Code Pipeline

Observations

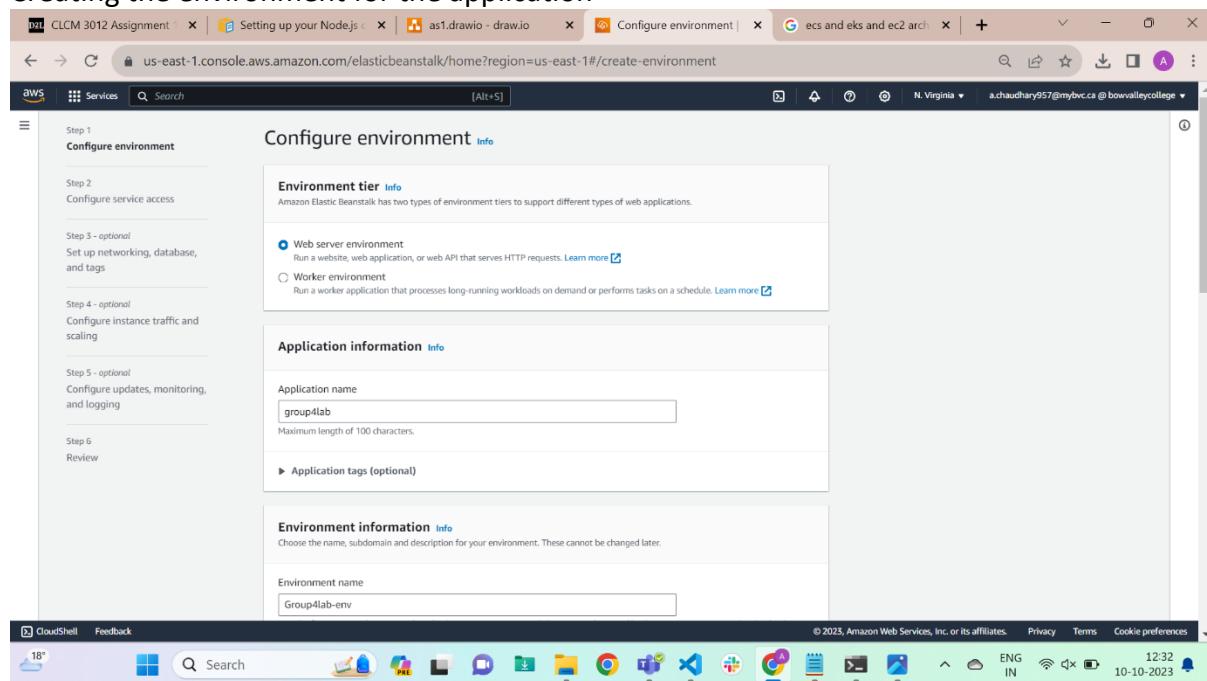
We have provided PHP as a platform in AWS Elastic Beanstalk.

We need to upload all the documents in GitHub.

We skipped the AWS Code Build step in code pipeline.

Screenshots

Creating the environment for the application



Have to select "PHP" platform

The screenshot shows the 'Create Environment' wizard on the AWS Elastic Beanstalk console. The current step is 'Platform info'. The 'Platform type' section is expanded, showing 'Managed platform' selected (with a note about platforms published by Amazon) and 'Custom platform' (disabled). The 'Platform' dropdown is set to 'Node.js' and the 'Platform branch' dropdown is set to 'Node.js 18 running on 64bit Amazon Linux 2023'. The top navigation bar includes tabs for 'CLCM 3012 Assignment', 'Setting up your Node.js', 'as1.drawio - draw.io', 'Configure environment', 'ecs and eks and ec2 arch', and a '+' button. The browser address bar shows the URL: us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/create-environment. The bottom status bar shows the date and time: 10-10-2023 12:33.

The screenshot continues the 'Create Environment' wizard. The 'Application code' section is expanded, showing 'Sample application' selected. The 'Presets' section is also expanded, showing 'Configuration presets' with 'Single instance (free tier eligible)' selected. The top navigation bar and browser details remain the same as the previous screenshot, indicating a continuation of the same session.

Have to create the keypair or linking the existing keypair

The screenshot shows the 'Configure service access' step of the AWS Elastic Beanstalk environment creation wizard. The 'Service role' section has 'Use an existing service role' selected, with 'aws-elasticbeanstalk-service-role' chosen. The 'EC2 key pair' section shows 'newkeyash' selected. The 'EC2 instance profile' section shows 'aws-elasticbeanstalk-ec2-role' selected. At the bottom, there are 'Cancel', 'Skip to review', 'Previous', and 'Next' buttons.

We have selected the default VPC

The screenshot shows the 'Set up networking, database, and tags - optional' step of the AWS Elastic Beanstalk environment creation wizard. Under 'Virtual Private Cloud (VPC)', the 'VPC' dropdown is set to 'vpc-551ddc28 (172.31.0.0/16)'. Under 'Instance settings', the 'Public IP address' checkbox is checked. Under 'Instance subnets', a table lists one subnet: 'us-east-1b' with CIDR '172.31.97.128/25' and name 'RDS-Pvt-subnet-4'. At the bottom, there are 'Cancel', 'Skip to review', 'Previous', and 'Next' buttons.

Public IP address should be activated

The screenshot shows the AWS Elastic Beanstalk configuration interface. The left sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main panel is titled 'Set up networking, database, and tags - optional'. It has two sections: 'Virtual Private Cloud (VPC)' and 'Instance settings'. In the VPC section, it says 'Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console.' A dropdown menu shows 'vpc-551dc28 | (172.31.0.0/16)'. Below it is a 'Create custom VPC' button. In the 'Instance settings' section, it says '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 load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances.' A note says 'Assign a public IP address to the Amazon EC2 instances in your environment.' A checkbox labeled 'Activated' is checked. At the bottom, there's a 'Instance subnets' table:

Availability Zone	Subnet	CIDR	Name
us-east-1b	subnet-001eeab4c...	172.31.97.128/25	RDS-Pvt-subnet-4

Selected one public subnet

The screenshot shows the same AWS Elastic Beanstalk configuration interface as the previous one, but with a different selection in the 'Instance subnets' table. The row for 'us-east-1d' is now highlighted with a blue border, indicating it is selected. The rest of the table rows are unselected.

Availability Zone	Subnet	CIDR	Name
us-east-1b	subnet-001eeab4c...	172.31.97.128/25	RDS-Pvt-subnet-4
us-east-1a	subnet-02e5e81fd...	172.31.96.128/25	RDS-Pvt-subnet-2
us-east-1e	subnet-05599aa1e...	172.31.98.128/25	RDS-Pvt-subnet-6
us-east-1d	subnet-059f58a33...	172.31.98.0/25	RDS-Pvt-subnet-5
us-east-1f	subnet-063376e40...	172.31.97.0/25	RDS-Pvt-subnet-3
us-east-1c	subnet-0c551b868...	172.31.96.0/25	RDS-Pvt-subnet-1
us-east-1f	subnet-2bd0af25	172.31.64.0/20	
<input checked="" type="checkbox"/> us-east-1d	subnet-31e96a10	172.31.80.0/20	
us-east-1b	subnet-5736b008	172.31.32.0/20	
us-east-1c	subnet-6f6aeef09	172.31.0.0/20	
us-east-1e	subnet-93c800a2	172.31.48.0/20	
us-east-1a	subnet-e91246a4	172.31.16.0/20	

At present there is no selection in Database

The screenshots show the 'Choose database subnets' step in the AWS Elastic Beanstalk 'Create Environment' wizard. The first screenshot shows a table of 12 available subnets across various availability zones. The second and third screenshots show the same table, but the 'us-east-1c' subnet has been selected, indicated by a checked checkbox.

Availability Zone	Subnet	CIDR	Name
us-east-1b	subnet-001eab4c...	172.31.97.128/25	RDS-Pvt-subnet-4
us-east-1a	subnet-02e5e81fd...	172.31.96.128/25	RDS-Pvt-subnet-2
us-east-1e	subnet-05599aa1e...	172.31.98.128/25	RDS-Pvt-subnet-6
us-east-1d	subnet-059f58a33...	172.31.98.0/25	RDS-Pvt-subnet-5
us-east-1f	subnet-063376e40...	172.31.97.0/25	RDS-Pvt-subnet-3
us-east-1c	subnet-0c551b868...	172.31.96.0/25	RDS-Pvt-subnet-1
us-east-1f	subnet-2bd0af25	172.31.64.0/20	
us-east-1d	subnet-31e96a10	172.31.80.0/20	
us-east-1b	subnet-5736b008	172.31.32.0/20	
us-east-1c	subnet-6f6aeef09	172.31.0.0/20	
us-east-1e	subnet-97e900-2	172.31.48.0/20	

The screenshot shows the AWS Elastic Beanstalk console with the URL `us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/create-environment`. The current step is "Choose database subnets (12)". A table lists 12 subnets across various availability zones:

	Availability Zone	Subnet	CIDR	Name
1	us-east-1b	subnet-001eeab4c...	172.31.97.128/25	RDS-Pvt-subnet-4
2	us-east-1a	subnet-02e5e81fd...	172.31.96.128/25	RDS-Pvt-subnet-2
3	us-east-1e	subnet-05599aa1e...	172.31.98.128/25	RDS-Pvt-subnet-6
4	us-east-1d	subnet-059f58a33...	172.31.98.0/25	RDS-Pvt-subnet-5
5	us-east-1f	subnet-063376e40...	172.31.97.0/25	RDS-Pvt-subnet-3
6	us-east-1c	subnet-0c551b868...	172.31.96.0/25	RDS-Pvt-subnet-1
7	us-east-1f	subnet-2bd0af25	172.31.64.0/20	
8	us-east-1d	subnet-31e96a10	172.31.80.0/20	
9	us-east-1b	subnet-5736b008	172.31.32.0/20	
10	us-east-1c	subnet-6f6aef09	172.31.0.0/20	
11	us-east-1e	subnet-93c800a2	172.31.48.0/20	
12	us-east-1a	subnet-e91246a4	172.31.16.0/20	

A checkbox labeled "Enable database" is present at the bottom left of the subnet selection area.

The screenshot shows the next step in the wizard: "Restore a snapshot - optional". It includes fields for "Snapshot" (set to "None"), "Database settings" (with dropdowns for "Engine", "Engine version", and "Instance class"), "Storage" (a slider set to 68 GB), and "Availability" (set to "Low (one AZ)").

The screenshot shows the final step of the wizard, which displays a summary of the configuration choices made so far.

The screenshot shows the AWS Elastic Beanstalk 'Create Environment' wizard at Step 3: Set up networking, database, and tags. The page is titled 'Set up networking, database, and tags - optional'. It includes sections for 'Instances' (Info), 'Root volume (boot device)', 'IOPS', 'Throughput', and 'Amazon CloudWatch monitoring'. On the left, a sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 (optional: Set up networking, database, and tags), Step 4 (optional: Configure instance traffic and scaling), Step 5 (optional: Configure updates, monitoring, and logging), and Step 6 (Review). The 'Next Step' button is highlighted in orange.

Instances

Root volume (boot device)

IOPS

Throughput

Amazon CloudWatch monitoring

Monitoring interval

Step 1
Configure environment

Step 2
Configure service access

Step 3 - optional
Set up networking, database, and tags

Step 4 - optional
Configure instance traffic and scaling

Step 5 - optional
Configure updates, monitoring, and logging

Step 6
Review

Instance metadata service (IMDS)
Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. [Learn more](#)

IMDSv1
With the current setting, the environment enables only IMDSv1.
 Deactivated

EC2 security groups
Select security groups to control traffic.

Group name	Group ID
Ash7security	sg-0f753fb7cece7c73a
ash7ubuntu	sg-07e66225661739d6d
ec2-rds-23	sg-0eb7bd4a7693df2
rds-ec2-23	sg-00e45ba03c2eda39e
security77ash	sg-0dd2b4cd9cf11c99
terrashivani	sg-01733ef95e64d0794

Capacity [Info](#)
Configure the compute capacity of your environment and auto scaling settings to optimize the number of instances used.

Auto scaling group

Environment type
Select a single-instance or load-balanced environment. You can develop and test an application in a single-instance environment to save costs and then upgrade to a load-balanced environment when the application is ready for production. [Learn more](#)

Single instance

Instances
1 Min
1 Max

Fleet composition
Spot instances are launched at the lowest available price. [Learn more](#)

On-Demand instance
 Spot instance

Maximum spot price
The maximum price per instance-hour, in USD, that you're willing to pay for a Spot Instance. Setting a custom price limits your chances to fulfill your target capacity using Spot instances.

Default
 Set your maximum price

On-Demand base
The minimum number of On-Demand Instances that your Auto Scaling group provisions before considering Spot Instances as your environment scales out.

The screenshot shows the AWS Elastic Beanstalk 'Create Environment' wizard, Step 3: Configure instance traffic. The configuration page includes:

- On-Demand above base:** A slider set to 0%.
- Capacity rebalancing:** A checkbox labeled 'Turn on capacity rebalancing' is checked.
- Architecture:** A radio button for 'x86_64 - new' is selected. A note states: 'This architecture uses x86 processors and is compatible with most third-party tools and libraries.'
- Instance types:** A dropdown menu 'Choose x86 instance types' shows 't3.micro' and 't3.small' selected.
- AMI ID:** The value 'ami-00482b709fe24a4a' is entered.
- Availability Zones:** A dropdown menu 'Number of Availability Zones (AZs) to use' has 'Any' selected.
- Placement:** A dropdown menu 'Specify Availability Zones (AZs) to use' has 'Choose Availability Zones (AZs)' selected.
- Scaling cooldown:** A slider set to 360 seconds.

At the bottom, there are navigation buttons: 'Cancel', 'Skip to review', 'Previous', and 'Next' (highlighted in yellow).

The image consists of three vertically stacked screenshots of the AWS Elastic Beanstalk configuration interface, specifically the 'Create New Environment' wizard.

Screenshot 1: Configure updates, monitoring, and logging - optional

Monitoring:

- System: Enhanced (selected)
- CloudWatch Custom Metrics - Instance: Choose metrics
- CloudWatch Custom Metrics - Environment: Choose metrics

Health event streaming to CloudWatch Logs:

Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

Log streaming:

Activated (standard CloudWatch charges apply.)

Retention:

7 days

Screenshot 2: Lifecycle

Lifecycle:

Keep logs after terminating environment

Managed platform updates:

Activate managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

Managed updates:

Activated

Weekly update window:

Tuesday at 21:49 UTC

Update level:

Minor and patch

Instance replacement:

If enabled, an instance replacement will be scheduled if no other updates are available.

Activated

Email notifications:

Enter an email address to receive email notifications for important events from your environment. [Learn more](#)

Email:

user@example.com

Screenshot 3: Lifecycle (Continuation)

18°

CloudShell Feedback

CloudShell Feedback

CloudShell Feedback

The image consists of three vertically stacked screenshots of the AWS Elastic Beanstalk console, illustrating the process of creating a new environment.

Screenshot 1: Configuration Step 1

This screenshot shows the initial configuration step. It includes:

- Email notifications:** An email address "user@example.com" is entered.
- Rolling updates and deployments:**
 - Application deployments:** Deployment policy is set to "All at once".
 - Batch size type:** Percentage is selected.
 - Deployment batch size:** 100 instances.
- Configuration updates:** Rolling update type is set to "Deactivated".

Screenshot 2: Configuration Step 2

This screenshot shows the continuation of the configuration process. It includes:

- Deployment preferences:**
 - Ignore health check:** Set to "False".
 - Health threshold:** Set to "Ok".
 - Command timeout:** Set to 600 seconds.
- Platform software:** Configuration for the specific platform.
- Container options:** Proxy server is set to "Nginx".

Screenshot 3: Configuration Step 3

This screenshot shows the final configuration step. It includes:

- Amazon X-Ray:** Configuration for Amazon X-Ray integration.

The image consists of three vertically stacked screenshots of the AWS Elastic Beanstalk console, showing the configuration steps for creating a new environment.

Screenshot 1: X-Ray Configuration

Amazon X-Ray

X-Ray daemon (service charges may apply)
 Activated

S3 log storage

Configure the instances in your environment to upload rotated logs to Amazon S3. Learn more [\[link\]](#)

Rotate logs (standard S3 charges apply)
 Activated

Instance log streaming to CloudWatch logs

Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. Learn more [\[link\]](#)

Log streaming (standard CloudWatch charges apply)
 Activated

Retention 7

Lifecycle Keep logs after terminating envir...

Screenshot 2: S3 Log Storage Configuration

S3 log storage

Configure the instances in your environment to upload rotated logs to Amazon S3. Learn more [\[link\]](#)

Rotate logs (standard S3 charges apply)
 Activated

Instance log streaming to CloudWatch logs

Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. Learn more [\[link\]](#)

Log streaming (standard CloudWatch charges apply)
 Activated

Retention 7

Lifecycle Keep logs after terminating envir...

Screenshot 3: Environment Properties Configuration

Environment properties

The following properties are passed in the application as environment properties. Learn more [\[link\]](#)

No environment properties have been configured.

Add environment property

Cancel Previous Next

The screenshot shows the AWS Elastic Beanstalk console interface for creating a new environment. The top navigation bar includes tabs for 'CLCM 3012 Assignment', 'Setting up your Node.js', 'as1.drawio - draw.io', 'Configure environment', 'ecs and eks and ec2 arch', and a search bar.

Step 1: Configure environment

Environment information

Environment tier	Application name
Web server environment	group4lab
Environment name	Application code
Group4lab-env	Sample application
Platform	arn:aws:elasticbeanstalk:us-east-1::platform/Node.js.18
	running on 64bit Amazon Linux 2023.6.0.1

Step 2: Configure service access

Service access

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 key pair	EC2 instance profile
arn:aws:iam::698668199773:role/aws-elasticbeanstalk-service-role	newkeyshash	aws-elasticbeanstalk-ec2-role

Step 3: Set up networking, database, and tags

Networking, database, and tags

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

Network

VPC	Public IP address	Instance subnets
vpc-551dcc28	true	subnet-31e96a10

Tags

Key	Value
No tags	
There are no tags defined	

Step 4: Configure instance traffic and scaling

Instance traffic and scaling

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options.

Instances

The screenshot shows two stacked windows of the AWS Elastic Beanstalk console.

Top Window: Environment Configuration

This window displays the configuration for an environment named "IMDSv1".

	Value	
Deactivated	EC2 Security Groups	sg-0f753f87cece7c73a
Capacity		
Environment type	Fleet composition	On-demand base
Single instance	On-Demand instance	0
On-demand above base	Capacity rebalancing	Scaling cooldown
0	Deactivated	360
Processor type	Instance types	AMI ID
x86_64	t3.micro,t3.small	ami-00482b709efe24a4a
Availability Zones	Metric	Statistic
Any	NetworkOut	Average
Unit	Period	Breach duration
Bytes	5	5
Upper threshold	Scale up increment	Lower threshold
6000000	1	2000000
Scale down increment		
-1		
Load balancer		
Load balancer visibility	Load balancer subnets	Load balancer type
public	subnet-31e96a10	application

Bottom Window: Step 5: Configure updates, monitoring, and logging

This window shows the configuration for "Step 5: Configure updates, monitoring, and logging".

	Value	
Updates, monitoring, and logging <small>Info</small>		
Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.		
Monitoring		
System	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
enhanced	—	—
Log streaming	Retention	Lifecycle
Deactivated	7	false
Updates		
Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok
Ignore health check	Instance replacement	
false	false	
Platform software		
Lifecycle	Log streaming	Proxy server
false	Deactivated	nginx

The screenshot shows the AWS Elastic Beanstalk configuration interface for creating a new environment. The configuration page includes sections for deployment settings, platform software, and environment properties.

Deployment Settings:

Setting	Value	Type
Managed updates	Activated	
Deployment batch size	100	Deployment batch size type
Command timeout	600	Deployment policy
Ignore health check	false	Health threshold
Instance replacement	false	

Platform Software:

Setting	Value	Proxy server
Lifecycle	false	Deactivated
Log streaming	nginx	
Logs retention	7	Rotate logs
X-Ray enabled	Deactivated	Deactivated
Update level	minor	

Environment Properties:

Key	Value
No environment properties	
There are no environment properties defined	

Buttons at the bottom: Cancel, Previous, Submit.

EBS is created

The screenshot shows the AWS Elastic Beanstalk environment dashboard for 'Group4lab-env-1'. The left sidebar displays navigation options like Applications, Environments, and Configuration. The main content area shows the 'Environment overview' and 'Platform' sections. In the 'Events' tab, four log entries are listed:

Time	Type	Details
October 10, 2023 13:06:15 (UTC-6)	INFO	Created security group named: sg-0f1967602268436a6
October 10, 2023 13:06:04 (UTC-6)	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 2 seconds). There are no instances.
October 10, 2023 13:06:04 (UTC-6)	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 2 seconds). There are no instances.

At the bottom of the dashboard, a green banner indicates: 'Environment successfully launched.'

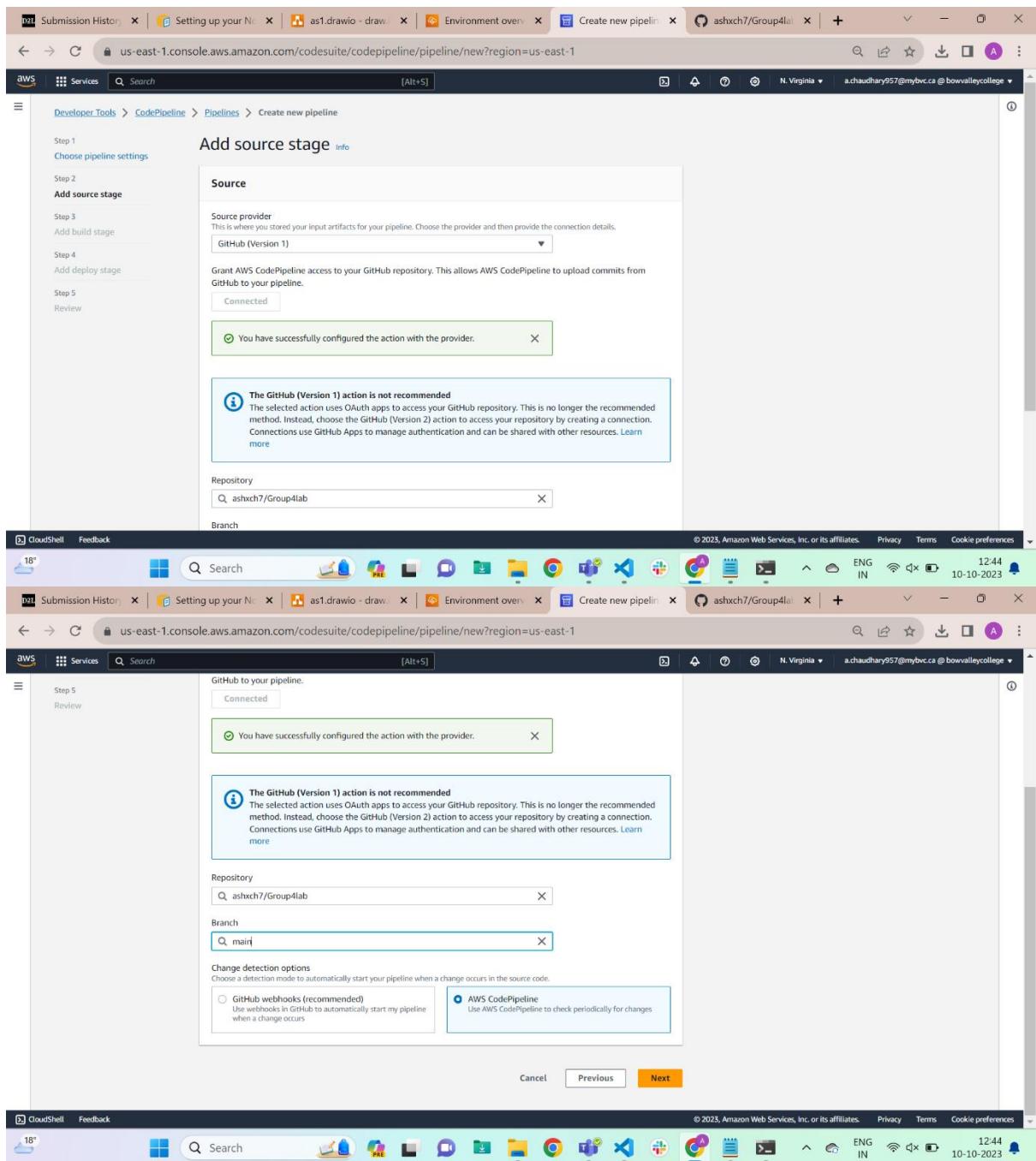
The taskbar at the bottom of the screen shows several open browser tabs and system icons.

Upload all the documents in the Github repository

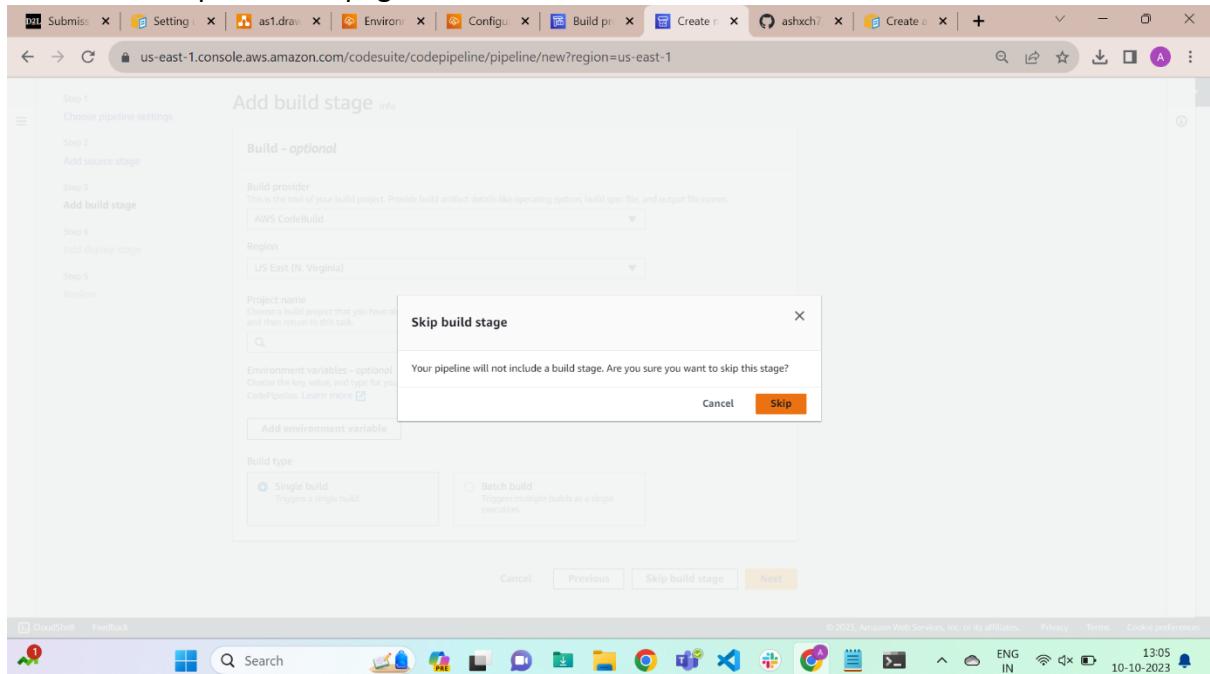
The screenshot shows a Microsoft Edge browser window with several tabs open. The active tab is 'ashxch7/Group4lab' on GitHub. The repository page displays basic information: 1 branch, 0 tags, and 1 commit. The commit details show a file upload from 'ashxch7' at '156ce98' with the message 'Add files via upload'. The commit includes five files: 'index.html', 'india.html', 'singapore.html', 'style.css', and 'venice.html', all added via upload. The repository stats show 1 star, 0 forks, and 0 watching. A note at the bottom encourages adding a README.

In AWS pipeline we are creating the new pipeline to link the Github

The screenshot shows the AWS CodePipeline 'Create new pipeline' wizard. It is on Step 1: 'Choose pipeline settings'. The pipeline name is set to 'group4lab'. Under 'Service role', the 'New service role' option is selected, with the role name 'AWSCodePipelineServiceRole-us-east-1-group4lab'. An 'Advanced settings' button is visible below. At the bottom, there are 'Cancel' and 'Next' buttons, with 'Next' being highlighted.

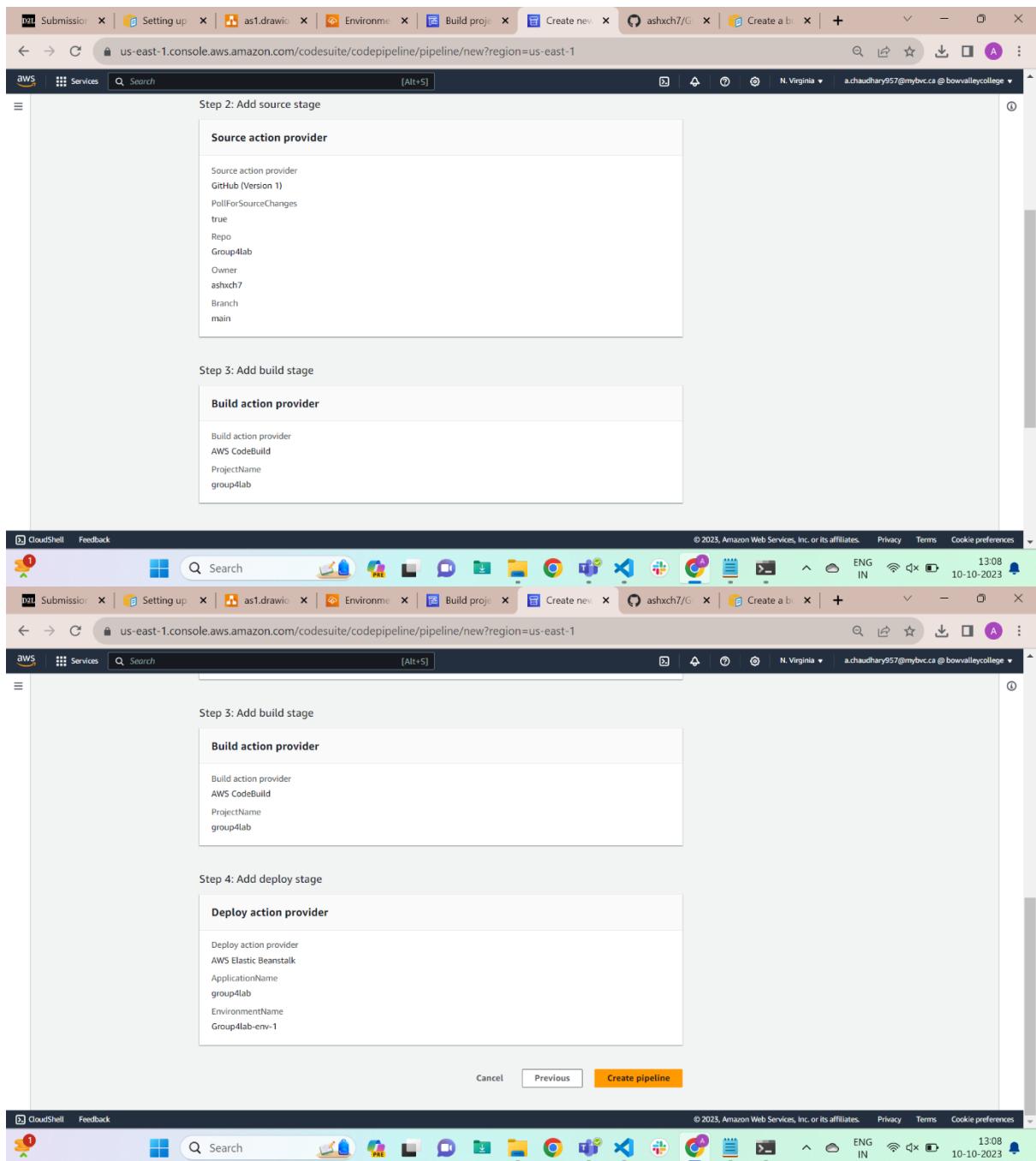


We have to skip the build page



The screenshot shows the 'Add deploy stage' step in the AWS CodePipeline 'Create new pipeline' wizard. The 'Deploy - optional' section is open, showing the configuration for deploying to AWS Elastic Beanstalk. The 'Deploy provider' dropdown is set to 'AWS Elastic Beanstalk'. The 'Region' dropdown is set to 'US East (N. Virginia)'. The 'Application name' field contains 'group4lab'. The 'Environment name' field contains 'Group4lab-env-1'. At the bottom, there are 'Cancel', 'Previous', 'Skip deploy stage', and 'Next' buttons.

The screenshot shows the 'Review' step in the AWS CodePipeline 'Create new pipeline' wizard. It displays the summary of the pipeline settings. The pipeline name is 'group4lab', the artifact location is 'codepipeline-us-east-1-318717957300', and the service role name is 'AWSCodePipelineServiceRole-us-east-1-group4lab'. Below this, the 'Source action provider' section is shown, listing 'GitHub (Version 1)', 'PollForSourceChanges', 'true', and 'Repo'. At the bottom, there are 'Cancel', 'Previous', 'Next', and 'Create pipeline' buttons.



After creating the pipeline, navigating AWS EBS and clicking the domain link, it should open the web page

The image contains two side-by-side screenshots of the AWS Management Console.

Screenshot 1: AWS Elastic Beanstalk Environment Overview

This screenshot shows the "Environment overview" for "Group4lab-env-1". Key details include:

- Health: Ok
- Domain: group4website.us-east-1.elasticbeanstalk.com
- Application name: group4lab
- Platform: PHP 8.2 running on 64bit Amazon Linux 2023/4.0.1
- Running version: -
- Platform state: Supported

The "Events" tab shows two recent events:

- October 10, 2023 13:07:31 (UTC-6): INFO Instance deployment completed successfully.
- October 10, 2023 13:07:26 (UTC-6): INFO Instance deployment: You didn't include a 'composer.json' file in your source bundle. The deployment didn't install

Screenshot 2: AWS CodePipeline Pipeline Overview

This screenshot shows the "Success" status for the pipeline "group4lab2". The pipeline structure is:

- Source (GitHub, Version 1) → Deploy (AWS Lambda, Step Functions)

A message indicates: "This pipeline has a source (Source) that is configured for polling. Migrate (update) your pipeline to the recommended event-based mechanism for change detection. For details, see the [migration guide](#). Alert closing in 10 seconds."

The screenshots show the AWS CodePipeline console interface for a pipeline named "group4lab2".

Screenshot 1 (Top): Pipeline in Progress

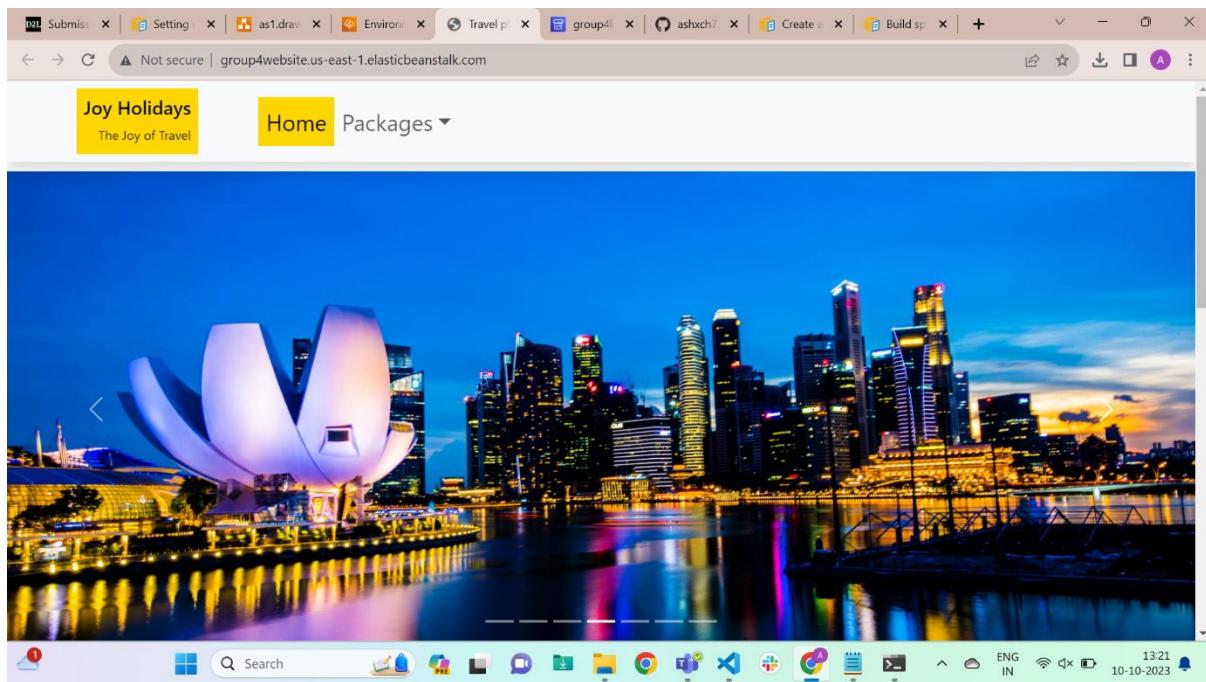
- Source Stage:** In progress. Pipeline execution ID: d2608af8-8b6b-4542-b76d-99c3ed2978ab.
 - Source: GitHub (Version 1)
 - Status: In progress - Just now
- Deploy Stage:** In progress. Pipeline execution ID: 7570ad64-d7e0-408f-a0ed-cb1abaa1ef5.
 - Deploy: AWS Elastic Beanstalk
 - Status: In progress - Just now

Screenshot 2 (Middle): Pipeline Succeeded

- Source Stage:** Succeeded. Pipeline execution ID: 00d98629a-8d35-4350-ac46-00432f2b2428.
 - Source: GitHub (Version 1)
 - Status: Succeeded - 1 minute ago
- Deploy Stage:** Succeeded. Pipeline execution ID: 00d98629a-8d35-4350-ac46-00432f2b2428.
 - Deploy: AWS Elastic Beanstalk
 - Status: Succeeded - Just now

Screenshot 3 (Bottom): Pipeline Succeeded

- Source Stage:** Succeeded. Pipeline execution ID: 00d98629a-8d35-4350-ac46-00432f2b2428.
 - Source: GitHub (Version 1)
 - Status: Succeeded - 1 minute ago
- Deploy Stage:** Succeeded. Pipeline execution ID: 00d98629a-8d35-4350-ac46-00432f2b2428.
 - Deploy: AWS Elastic Beanstalk
 - Status: Succeeded - Just now



What we Learned

Akilandeshwari, Srinivasan: I have learned how to use code pipeline to automate code in AWS Elastic Beanstalk. Moreover, experienced that we have missed the asset folder to upload in the git that leads to missing images and hierarchy in the code, so upload every folder to git. Finally use "Release Changes" for bulk upload in pipeline.

Ambedkar Rani, Subbiah: I started by learning how to set up the PHP platform's Elastic Beanstalk on AWS. Subsequently, I learned how to integrate the code pipeline with EBS to use Glithub for code deployment. If we make any small modifications in Github after the pipeline procedure is finished, they will automatically appear in EBS. On the other hand, we must hit "Release changes" in the code pipeline when we make bulk uploads.

Ashutosh, Chaudhary:

- I learned to create AWS Code Pipeline.
- I learned to deploy code using Code Pipeline.
- I learned to create application and environment in AWS Elastic Beanstalk.

Sorawat, Tanthikun

Learned about how to deploy by using AWS Elastic Beanstalk and observed that when creating a website, need to use PHP for basic HTML and CSS websites. Also, gained some knowledge about how to connect AWS CodePipeline with Elastic Beanstalk from GitHub repository.

Taniya, Kamboj: I now know how to set up AWS Elastic Beanstalk, setup it with a code pipeline, and use Github to deploy the code.

References

<https://aws.amazon.com/>

<https://docs.aws.amazon.com/index.html>