Implement Advanced CloudWatch Monitoring for a Web Server

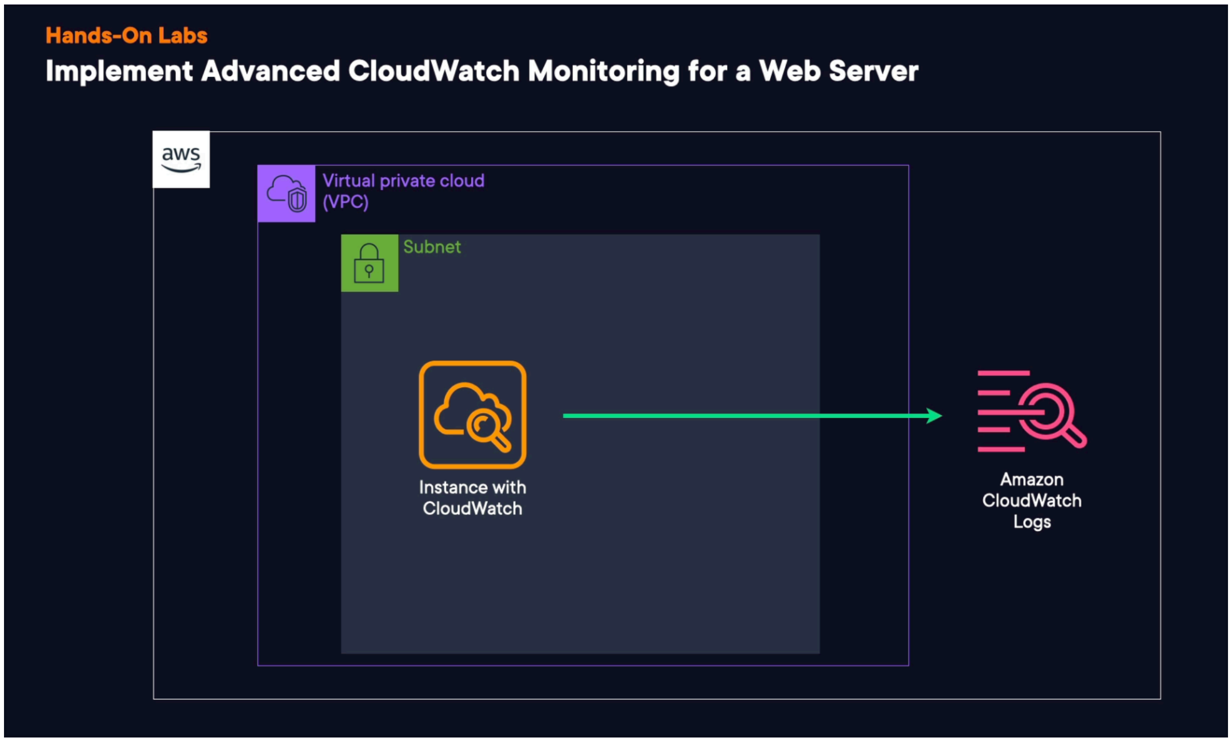
The objective of this project is to utilize CloudWatch Logs, a robust and scalable service, for centralizing logs generated across your systems, applications, and AWS services. Within this lab, you will learn to set up an EC2 instance, enabling it to transmit its Apache web server error logs directly to CloudWatch Logs. Through configuring the necessary agent and subsequently verifying the log streaming process in the CloudWatch Logs console, you'll gain hands-on experience in efficiently managing log data. By the conclusion of this lab, you'll have the expertise to install the CloudWatch Logs agent and adeptly configure it to seamlessly stream logs to the CloudWatch Logs service.

Shall we begin? Yoohooo!

I want to inform you that I am using some resources from Cloud Guru, and I extend special thanks to them!

Before conducting this project, ensure that you have one instance running.

The initial step in our project is to create a diagram


Hands-On Labs
Implement Advanced CloudWatch Monitoring for a Web Server
aws
Virtual private cloud (VPC)
Subnet
Instance with CloudWatch
Amazon CloudWatch Logs

First, open the search box and type "EC2".

Once you are in the EC2 dashboard, click on the "Instances Running" section.

Select the checkbox next to your instance and click on the "Connect" button on the right-hand side.

I will provide you with command lines in this GitHub repository which you can use for this project.

Run the following command in your terminal to download the AWS Logs Agent setup file from S3:

```
wget -O awslogs-agent-setup.py https://s3.amazonaws.com/aws-cloudwatch/
downloads/latest/awslogs-agent-setup.py
```

Copy and paste the command into your terminal and press enter. This command will run the agent setup file from S3.

Next, execute our sudo Python code. Copy and paste it into the terminal and press enter to run the code:

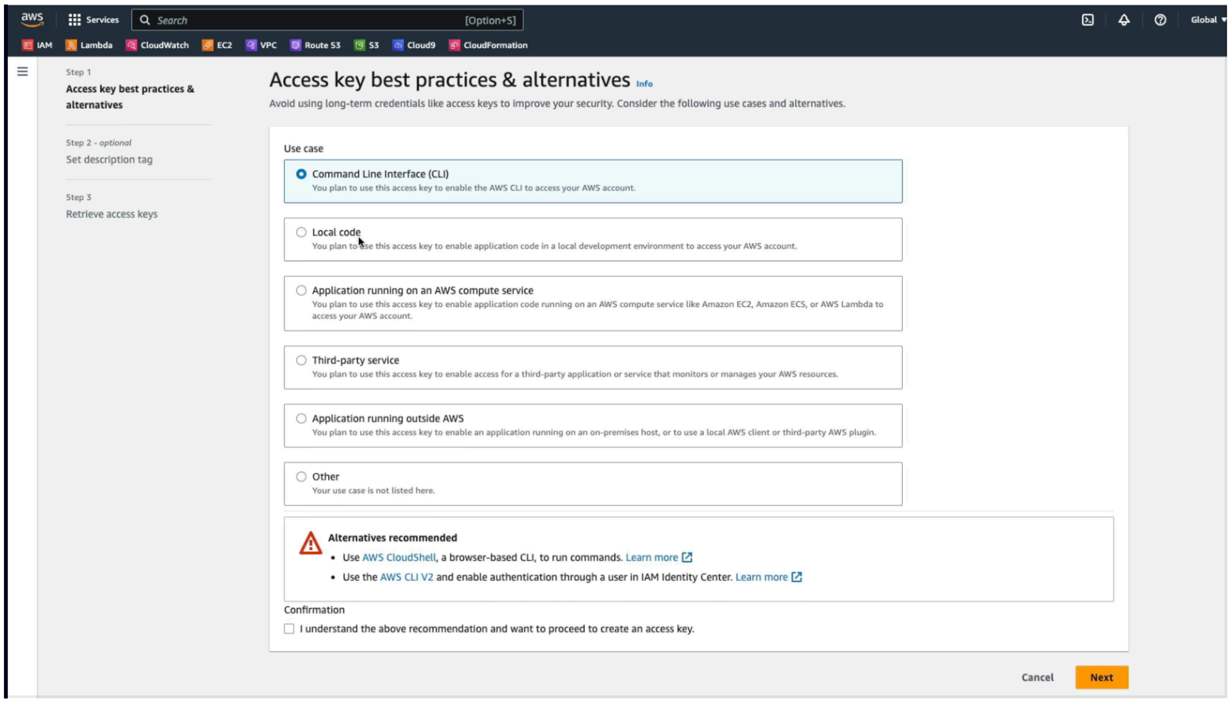
```
sudo python ./awslogs-agent-setup.py --region us-east-1
```

Now, search for "IAM" in the console and click on it.

Once you are in the IAM console, on the left side panel, click on "Users".

Select the user you want to configure. At the bottom of the page, click on "Security Credentials".

Click on "Create Access Key". For our project, we will be using the Command Line Interface (CLI).



Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

- ● Command Line Interface (CLI)
  You plan to use this access key to enable the AWS CLI to access your AWS account.
- ○ Local code
  You plan to use this access key to enable application code in a local development environment to access your AWS account.
- ○ Application running on an AWS compute service
  You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- ○ Third-party service
  You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- ○ Application running outside AWS
  You plan to use this access key to enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.
- ○ Other
  Your use case is not listed here.

Alternatives recommended
- Use AWS CloudShell, a browser-based CLI, to run commands. Learn more
- Use the AWS CLI V2 and enable authentication through a user in IAM Identity Center. Learn more

Confirmation

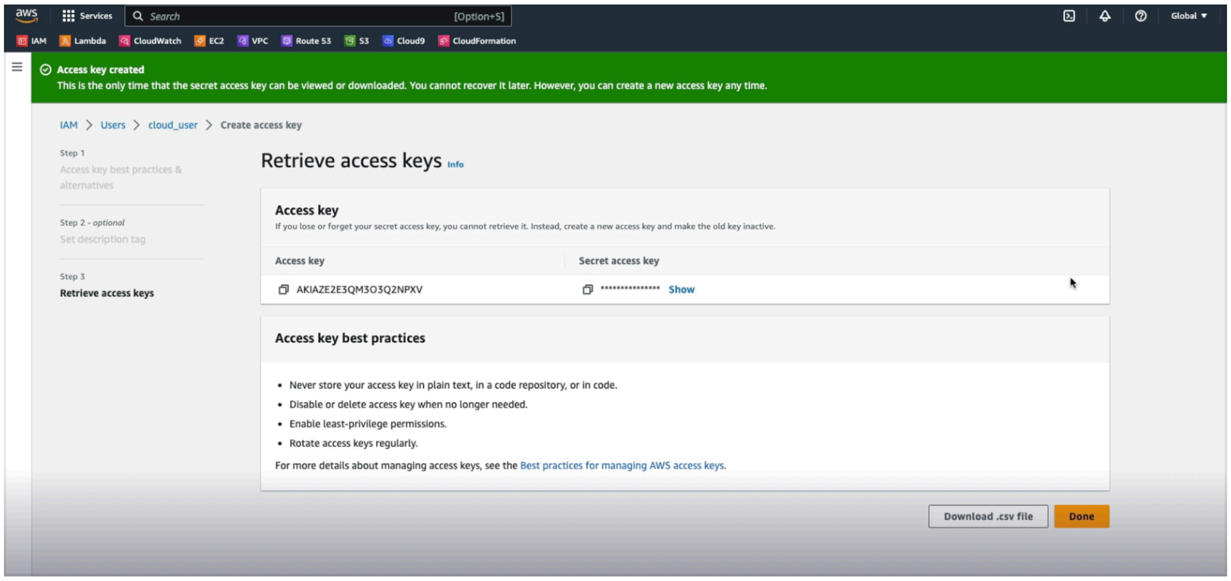
☐ I understand the above recommendation and want to proceed to create an access key.

Now, confirm your action by checking the confirmation checkbox.

Click on the "Next" button.

For our description tag value, name it "clouduseraccesskey".

Congratulations! You have successfully created your access key



Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

IAM > Users > cloud_user > Create access key

Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

Retrieve access keys Info

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAZE2E3QM3O3Q2NPXV	*************** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the Best practices for managing AWS access keys.

First, copy the access key you generated.

Now, return to the EC2 console. Paste the access key value and press enter.

Next, go back to the IAM user page. Click on "Show Secret Access Key" and copy the value.

Return once more to the EC2 console. Paste the secret access key value there. Press enter twice.

For the path, please refer to the provided path guide link: `/var/log/apache2/error.log`. Do not modify anything here for the purpose of this project. Press enter.

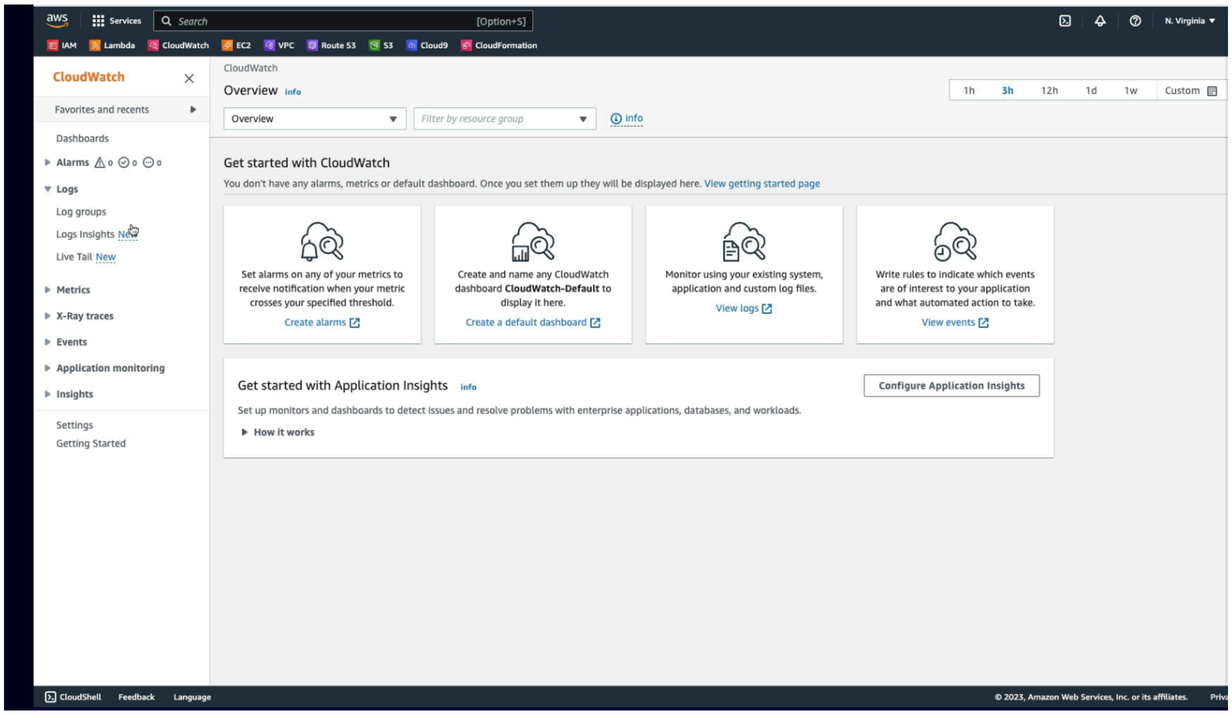
When prompted for the third choice, type "1" and then "n". Press enter.

Now that the agent has been configured, you can check the URL provided.

In the search bar at the top, type "CloudWatch" and press enter.

Right-click on CloudWatch and open it in a new tab.

Now, that the CloudWatch page is open, click on "Log Groups" on the left side.



Get started with CloudWatch

You don't have any alarms, metrics or default dashboard. Once you set them up they will be displayed here. View getting started page

Set alarms on any of your metrics to receive notification when your metric crosses your specified threshold.

Create and name any CloudWatch dashboard **CloudWatch-Default** to display it here.

Monitor using your existing system, application and custom log files.

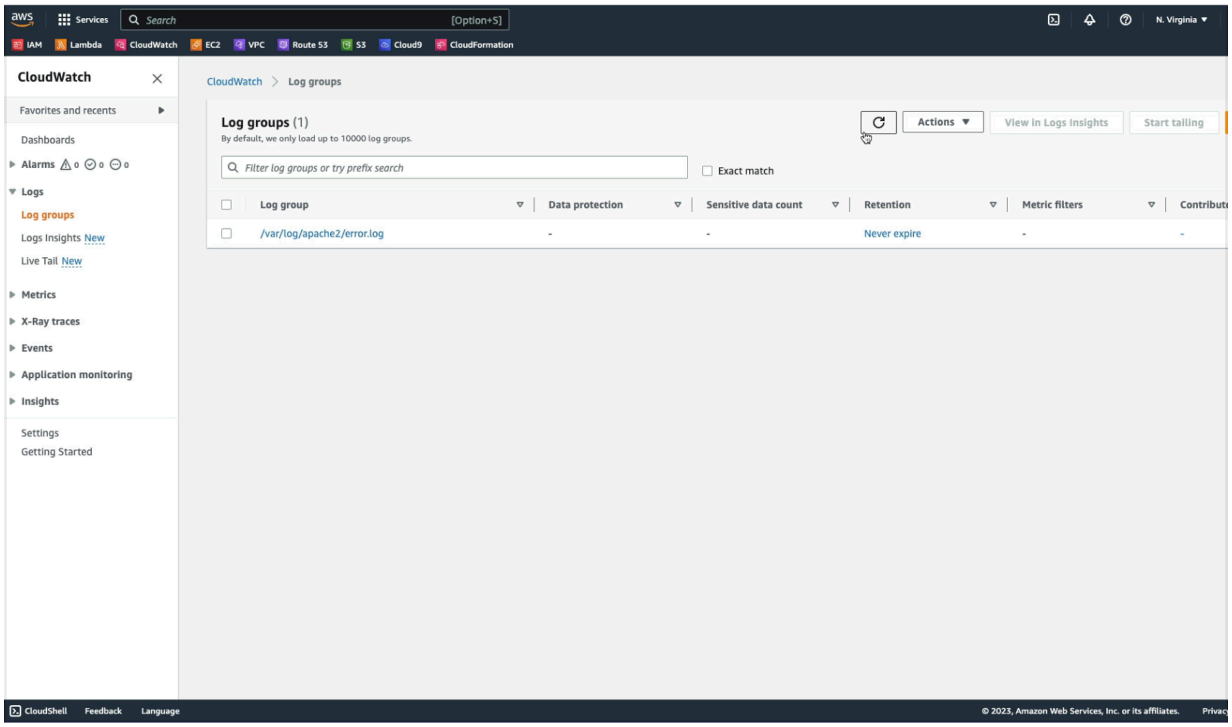
Write rules to indicate which events are of interest to your application and what automated action to take.

Get started with Application Insights info

Set up monitors and dashboards to detect issues and resolve problems with enterprise applications, databases, and workloads.

© 2023, Amazon Web Services, Inc. or its affiliates.

You should now be able to see the newly created log group. If it's not visible, click the refresh button.



Log groups (1)
By default, we only load up to 10000 log groups.

Log group	Data protection	Sensitive data count	Retention	Metric filters	Contribut
/var/log/apache2/error.log	-	-	Never expire	-	-

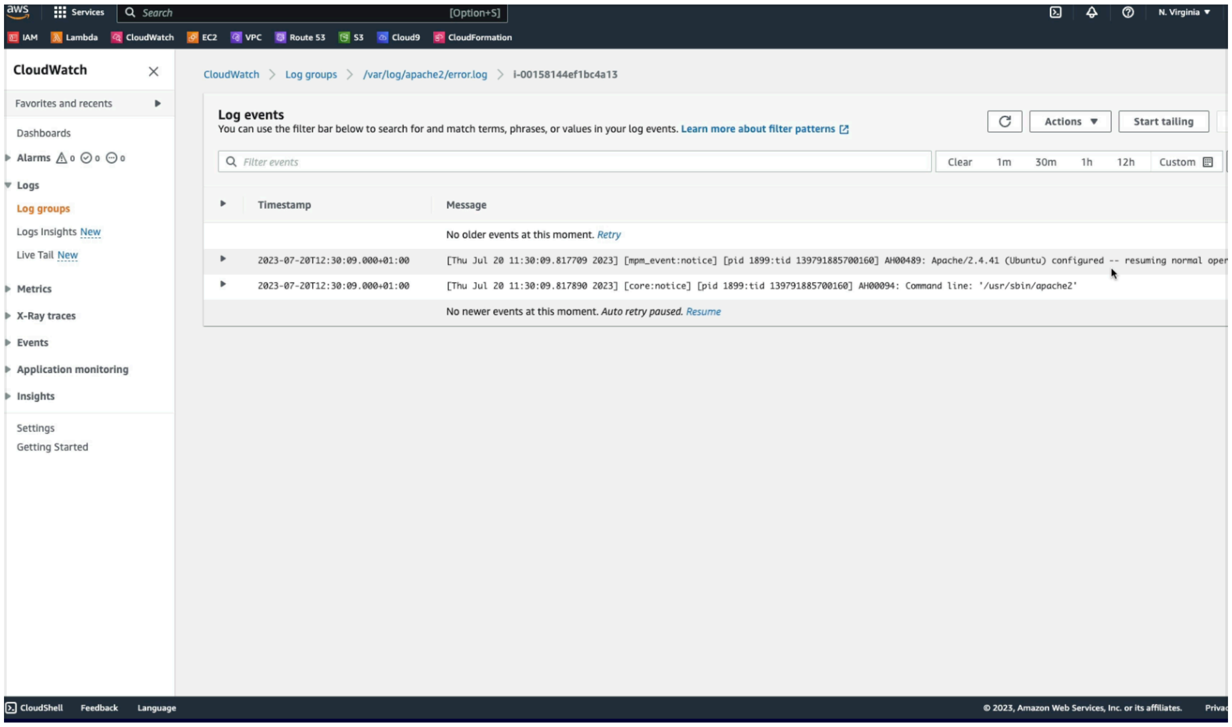
Select the log group that we just created.

Under "Log Stream," click on our log stream.

To verify if we have completed this project correctly, you can compare our log events with the results from the EC2 terminal when we ran this code:

```
sudo python ./awslogs-agent-setup.py --region us-east-1
```

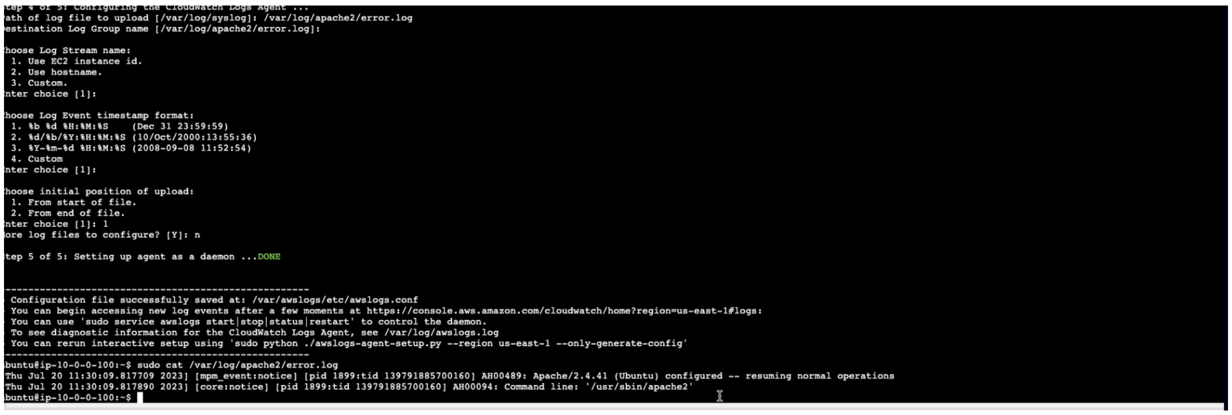
This will help confirm the proper execution of our project.



Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                      Message
                               No older events at this moment. Retry
2023-07-20T12:30:09.000+01:00  [Thu Jul 20 11:30:09.817709 2023] [mpm_event:notice] [pid 1899:tid 139791885700160] AH00489: Apache/2.4.41 (Ubuntu) configured -- resuming normal oper
2023-07-20T12:30:09.000+01:00  [Thu Jul 20 11:30:09.817890 2023] [core:notice] [pid 1899:tid 139791885700160] AH00094: Command line: '/usr/sbin/apache2'
                               No newer events at this moment. Auto retry paused. Resume
```



```
tep 4 of 5: Configuring the CloudWatch Logs Agent ...
ath of log file to upload [/var/log/syslog]: /var/log/apache2/error.log
estination Log Group name [/var/log/apache2/error.log]:

hoose Log Stream name:
 1. Use EC2 instance id.
 2. Use hostname.
 3. Custom.
nter choice [1]:

hoose Log Event timestamp format:
 1. %b %d %H:%M:%S    (Dec 31 23:59:59)
 2. %d/%b/%Y:%H:%M:%S (10/Oct/2000:13:55:36)
 3. %Y-%m-%d %H:%M:%S (2008-09-08 11:52:54)
 4. Custom
nter choice [1]:

hoose initial position of upload:
 1. From start of file.
 2. From end of file.
nter choice [1]: 1
ore log files to configure? [Y]: n

tep 5 of 5: Setting up agent as a daemon ...DONE


------------------------------------------------------
 Configuration file successfully saved at: /var/awslogs/etc/awslogs.conf
 You can begin accessing new log events after a few moments at https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#logs:
 You can use 'sudo service awslogs start|stop|status|restart' to control the daemon.
 To see diagnostic information for the CloudWatch Logs Agent, see /var/log/awslogs.log
 You can rerun interactive setup using 'sudo python ./awslogs-agent-setup.py --region us-east-1 --only-generate-config'
------------------------------------------------------
buntu@ip-10-0-0-100:~$ sudo cat /var/log/apache2/error.log
Thu Jul 20 11:30:09.817709 2023] [mpm_event:notice] [pid 1899:tid 139791885700160] AH00489: Apache/2.4.41 (Ubuntu) configured -- resuming normal operations
Thu Jul 20 11:30:09.817890 2023] [core:notice] [pid 1899:tid 139791885700160] AH00094: Command line: '/usr/sbin/apache2'
buntu@ip-10-0-0-100:~$
```

Congratulations! It sounds like you have successfully launched your project and achieved your goals. Well done! If you have any more questions or if there's anything else I can assist you with, feel free to let me know. I'm here to help!