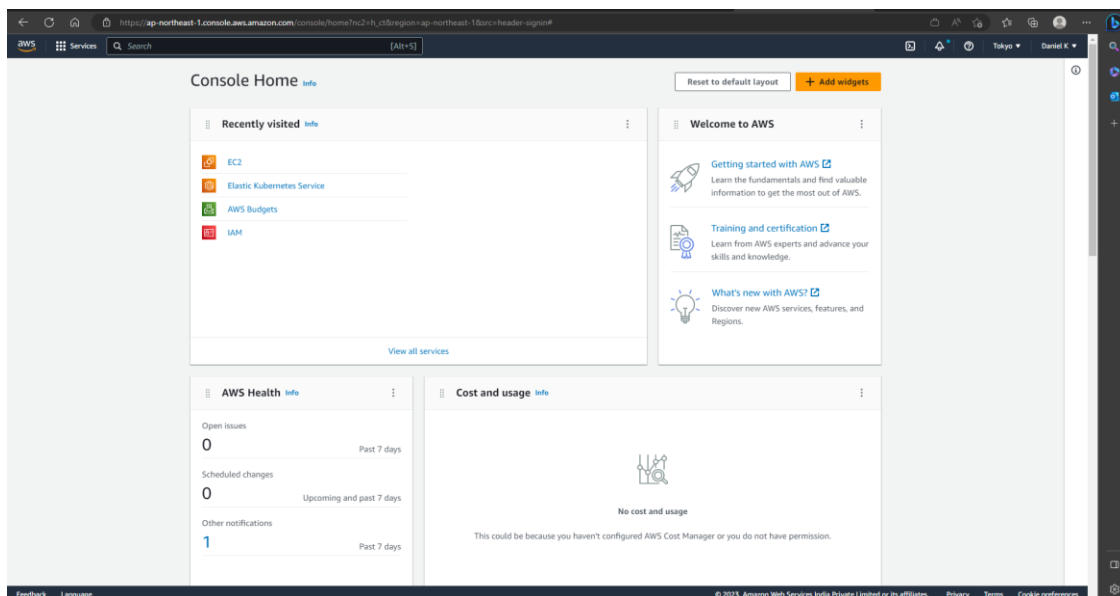


Devops Assignments- Module 9- Introduction to AWS DevOps Services

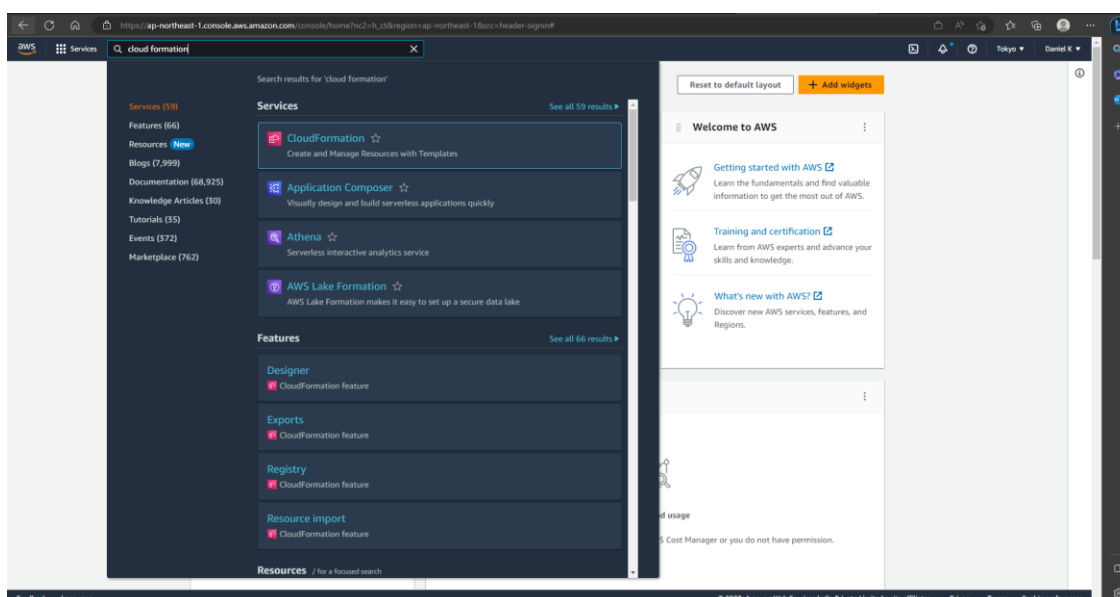
Assignment 1 - Create a Stack using an AWS CloudFormation.docx

In this Assignment we will use a pre- configured sample template that creates an Amazon EC2 instance and installs WordPress with a local MySQL database for storage on it.

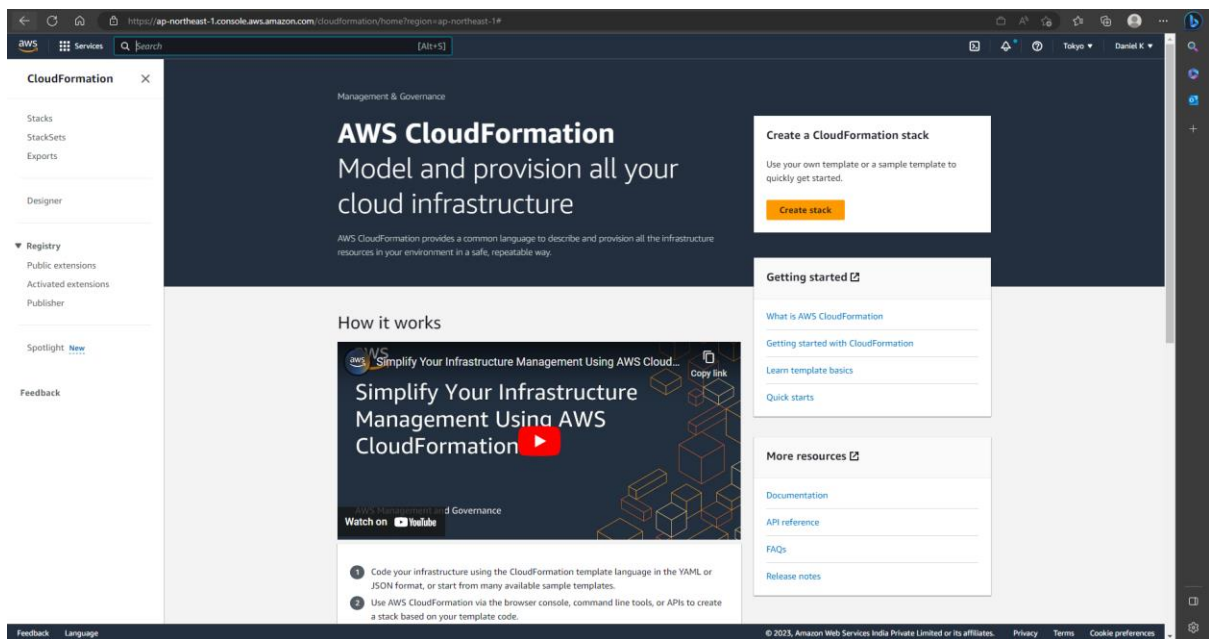
1. Create CloudFormation Stack using template that declares the resources for a WordPress blog, creates a WordPress blog as a stack, monitor the stack creation process, examine the resources on the stack, and then delete the stack:
 - Sign into AWS console



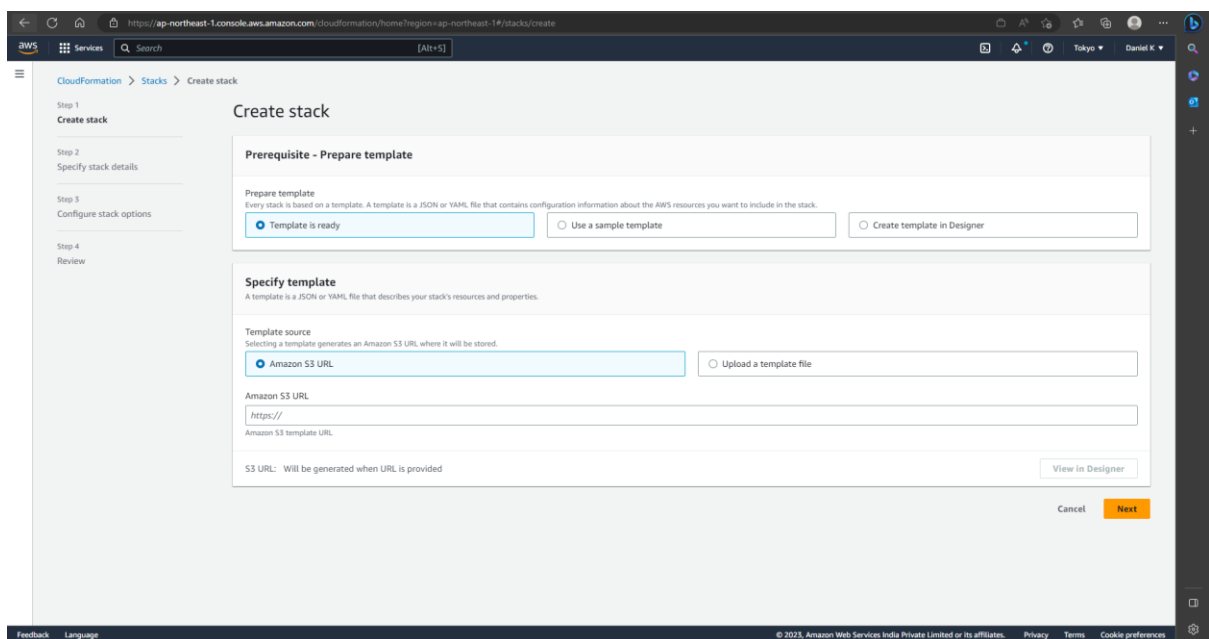
- In AWS console search box search cloud formation.



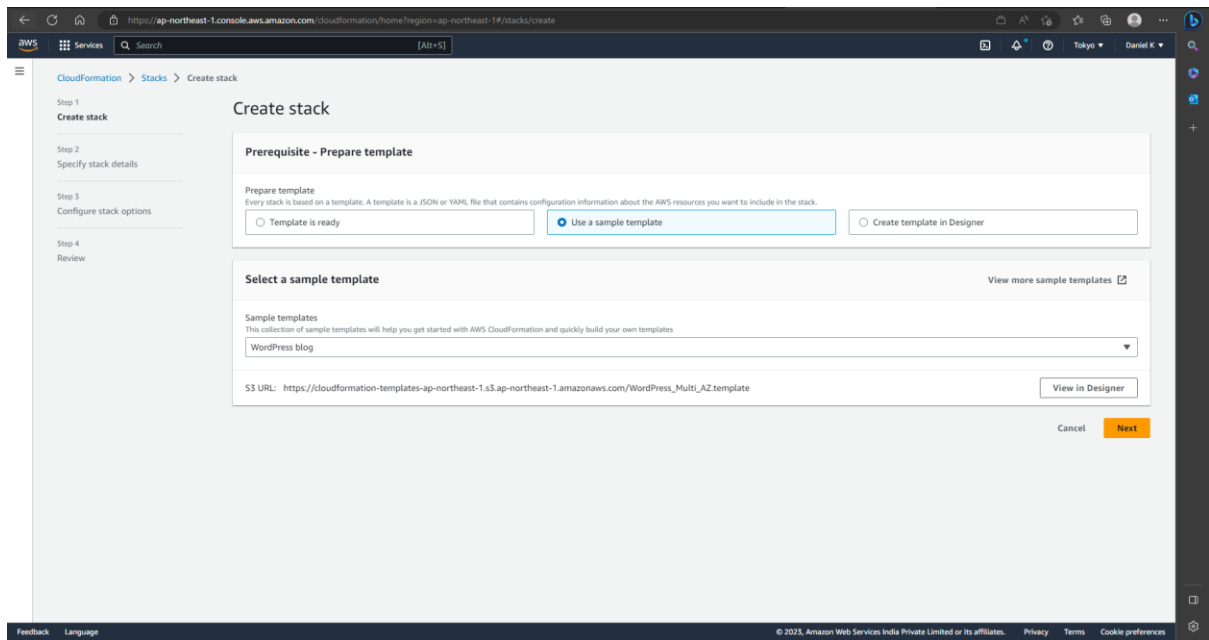
- Click on CloudFormation and It will open a CloudFormation page



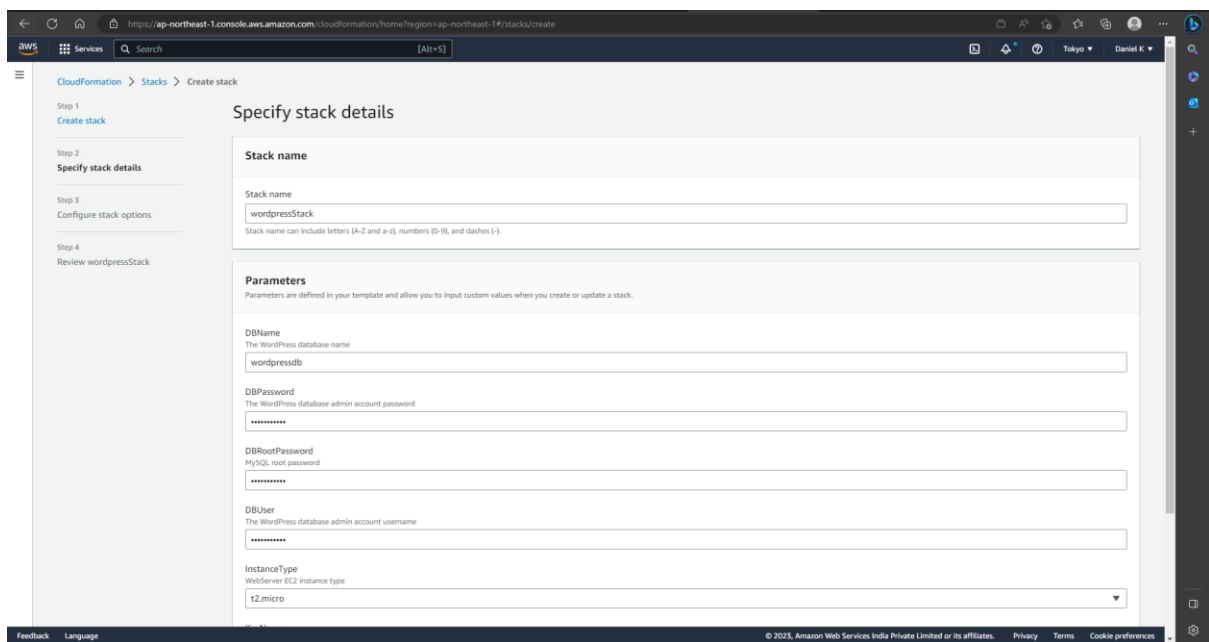
- Click on create stack for Creating a new Stack



- Select → Use a sample template → under sample template select wordpress blog → Click on next.



- Under stack details give name for stack → Enter DB password → select instance type, key pair and all details → click on next



- Under configure stack options → review tags, IAM role → click next

Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review wordpressStack

Configure stack options

Tags
You can specify tags (key-value pair) to apply to resources in your stack. You can add up to 50 unique tags for each stack.

No tags associated with the stack.

[Add new tag](#)
You can add 50 more tags

Permissions

IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name: [Remove](#)

Stack failure options

Behavior on provisioning failure
Specify the roll back behavior for a stack failure. [Learn more](#)

☒ Roll back all stack resources
Roll back the stack to the last known stable state.

☐ Preserve successfully provisioned resources
Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.

Advanced options
You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name: [Remove](#)

Stack failure options

Behavior on provisioning failure
Specify the roll back behavior for a stack failure. [Learn more](#)

☒ Roll back all stack resources
Roll back the stack to the last known stable state.

☐ Preserve successfully provisioned resources
Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.

Advanced options
You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

► **Stack policy**
Defines the resources that you want to protect from unintentional updates during a stack update.

► **Rollback configuration**
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back.

► **Notification options**

► **Stack creation options**

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

- Next review all the details → Click on submit

Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review wordpressStack

Review wordpressStack

Step 1: Specify template [Edit](#)

Template

Template URL
https://cloudformation-templates-ap-northeast-1.s3.ap-northeast-1.amazonaws.com/WordPress_Single_Instance.template

Stack description
AWS CloudFormation Sample Template WordPress_Single_Instance: WordPress is web software you can use to create a beautiful website or blog. This template installs WordPress with a local MySQL database for storage. It demonstrates using the AWS CloudFormation bootstrap scripts to deploy WordPress. ****WARNING**** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.

Step 2: Specify stack details [Edit](#)

Parameters (7)

Key	Value
DBName	wordpressdb
DBPassword	*****
DBRootPassword	*****
DBUser	*****
InstanceType	t2.micro
KeyName	jerk

There is no stack policy defined

Rollback configuration

Monitoring time
-

CloudWatch alarm ARN
-

Notification options

SNS topic ARN
No notification options
There are no notification options defined

Stack creation options

Timeout
-

Termination protection
Disabled

► Quick-create link

Create change set

Cancel Previous **Submit**

- Here it will create stack with one ec2 instance as we can see in resource

CloudFormation > Stacks > wordpressStack

Stacks (1)

Filter by stack name

Active View nested

Stacks

wordpressStack
2023-03-29 16:44:11 UTC+0530
CREATE_IN_PROGRESS

wordpressStack

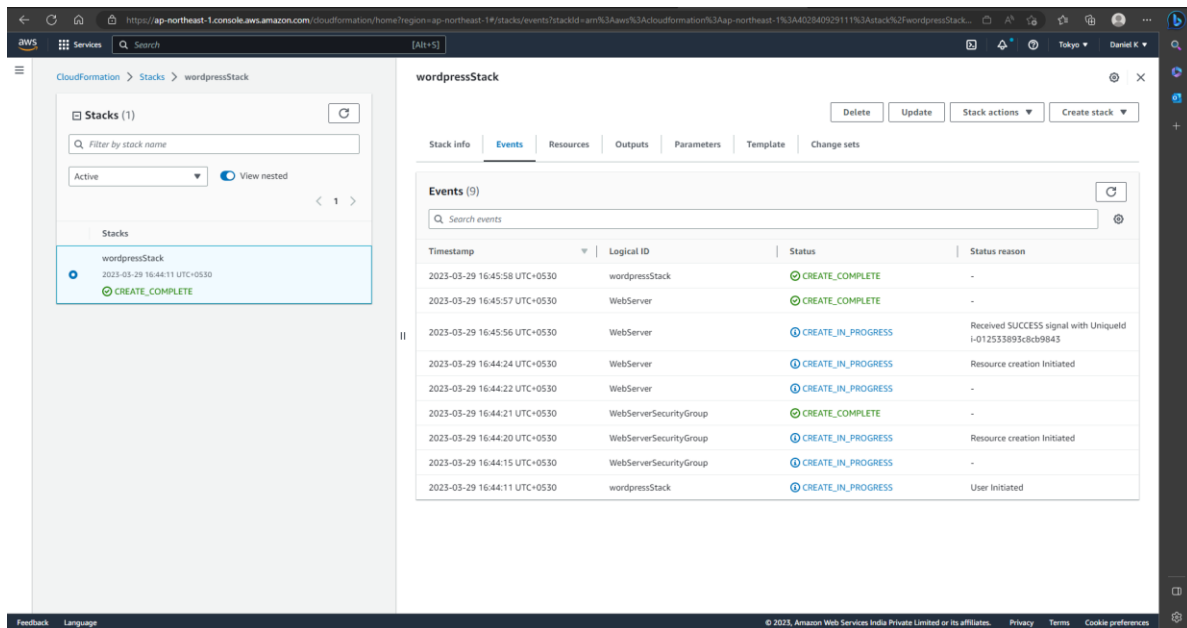
Delete Update Stack actions Create stack

Stack info Events **Resources** Outputs Parameters Template Change sets

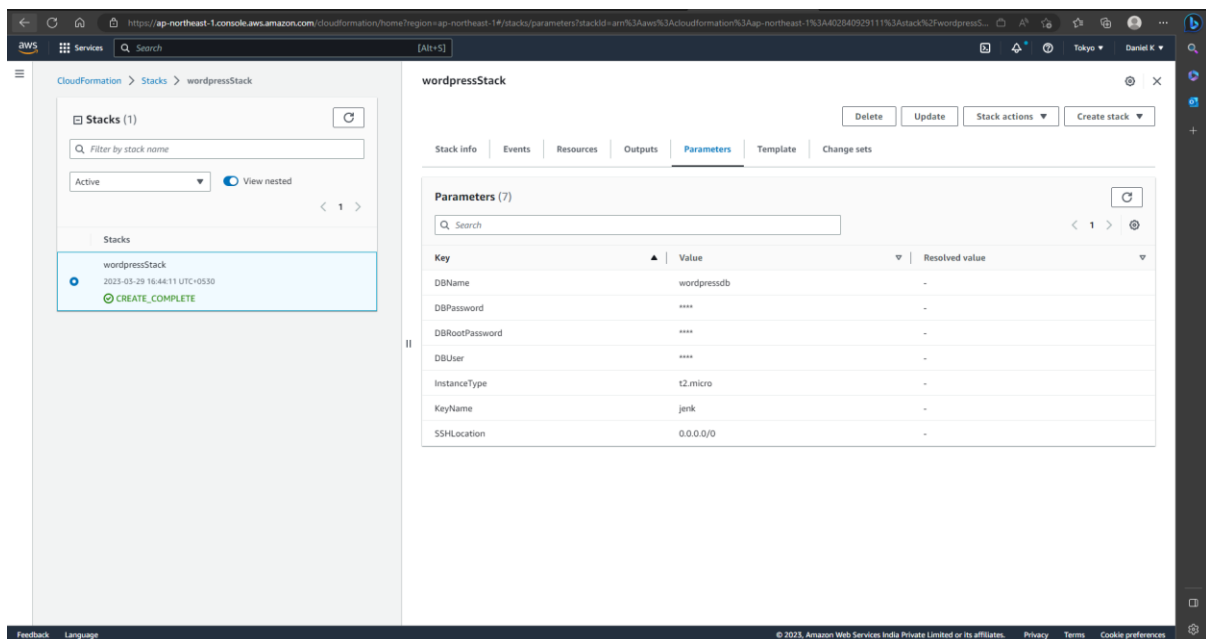
Resources (2)

Logical ID	Physical ID	Type	Status	Module
WebServer	i-012533895dcb9843	AWS::EC2::Instance	CREATE_COMPLETE	-
WebServerSecurityGroup	wordpressStack-WebServerSecurityGroup-1E597GZ7H647X	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-

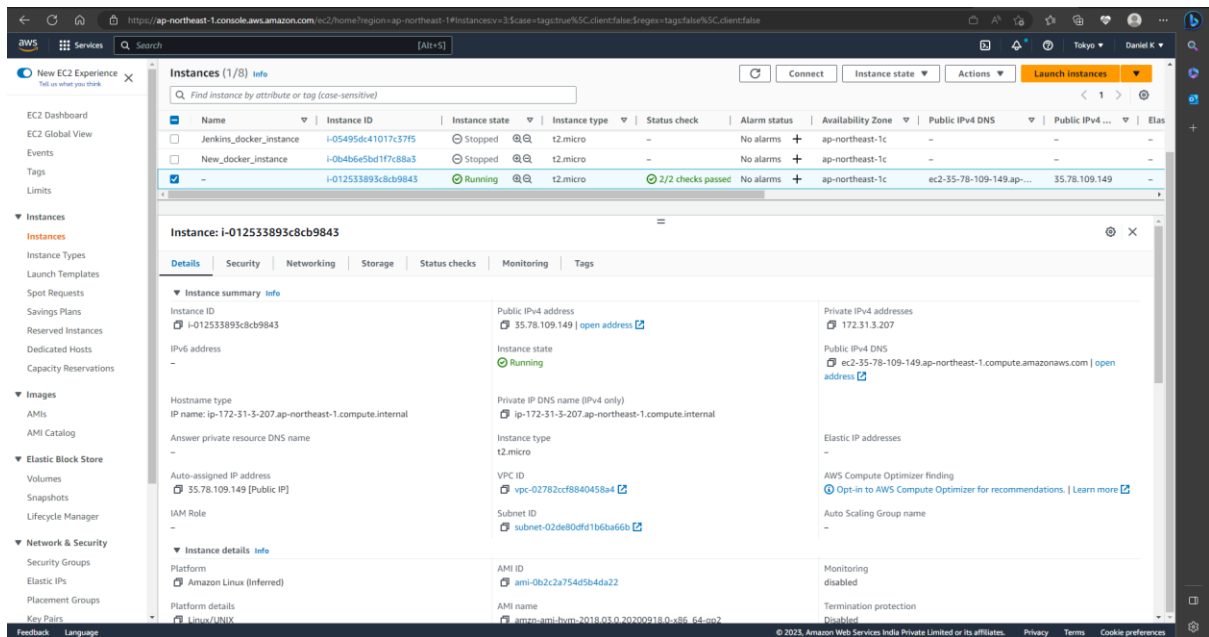
- Under events it shows the details about wordpress



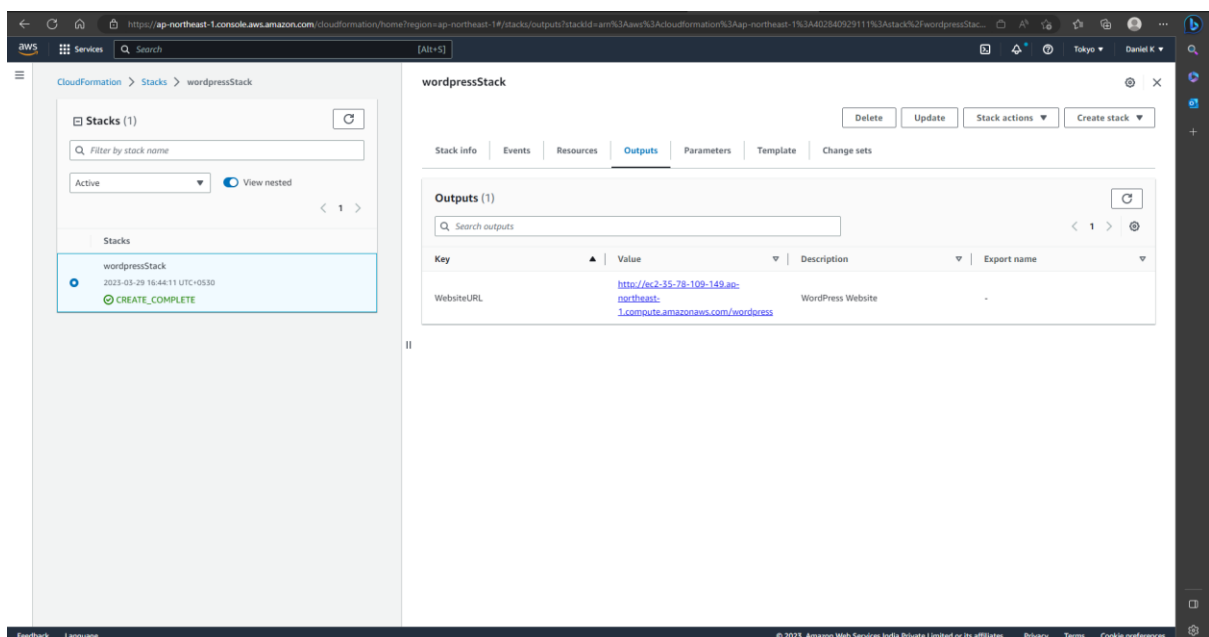
- In parameter tab it will show the provided data as DB name, instance type, keypair etc.



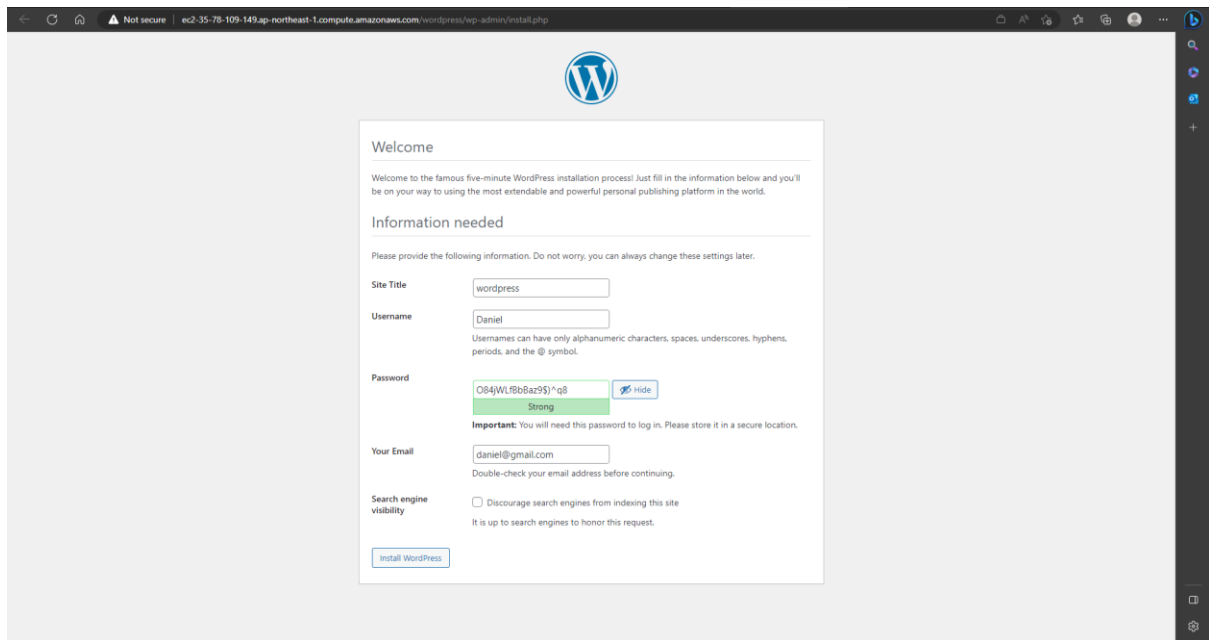
- An instance will be created



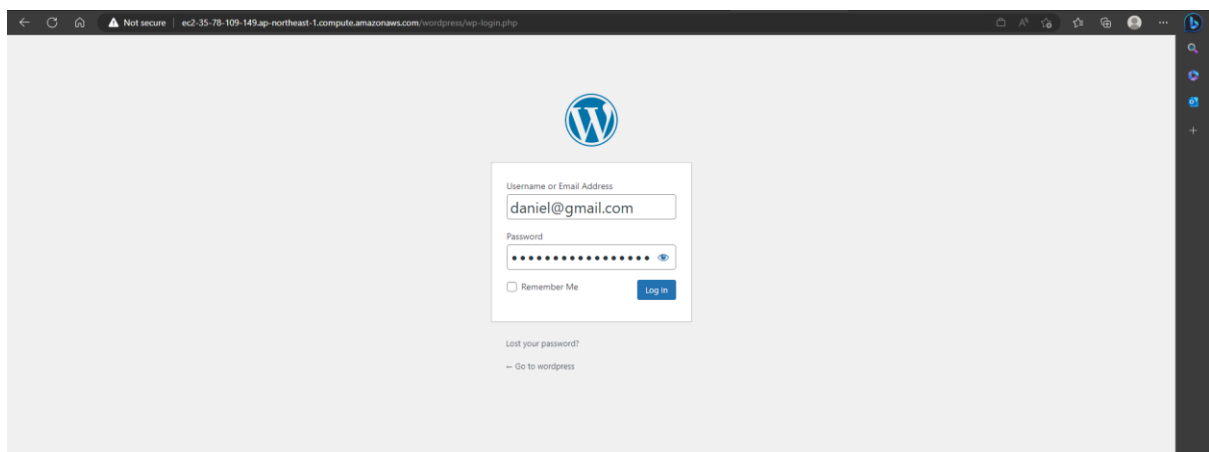
- In output tab stack is created on URL.



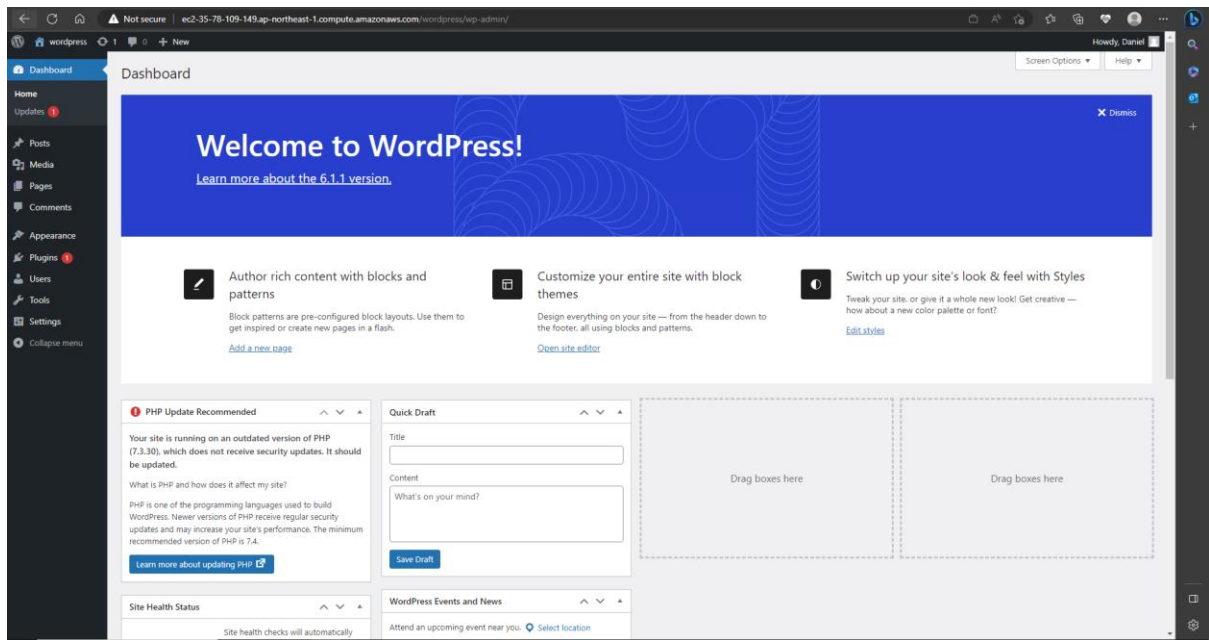
- Click on the URL or open in another page → It opens installation page of wordpress.
- Fill all details and click on install wordpress.



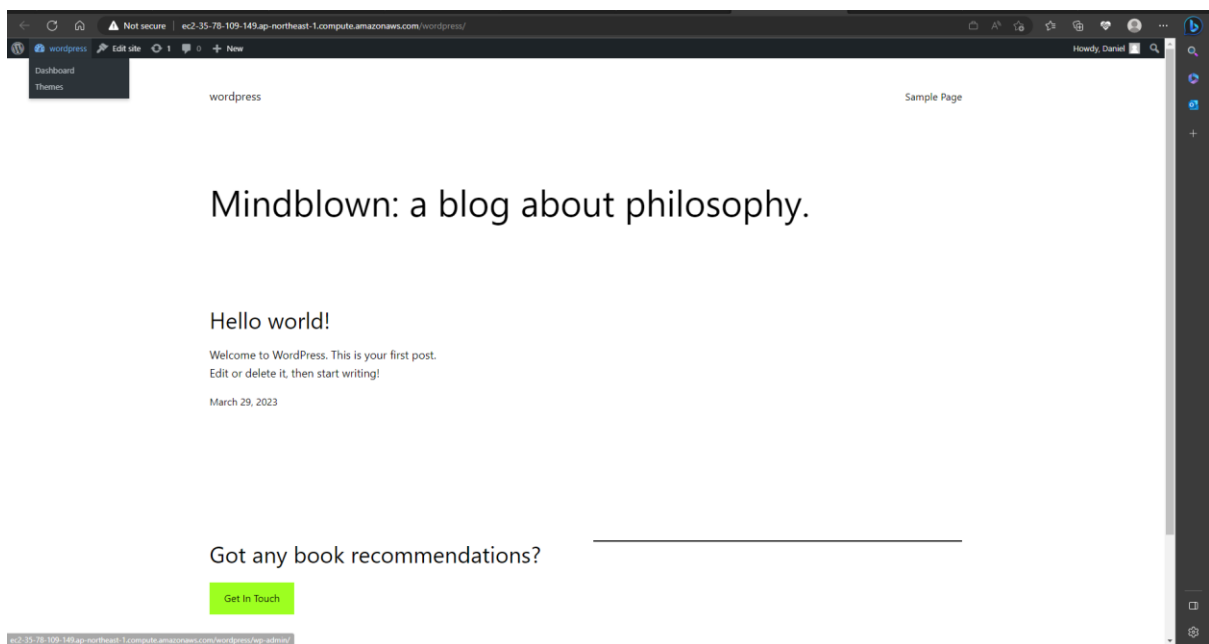
- After installing, Login in to wordpress



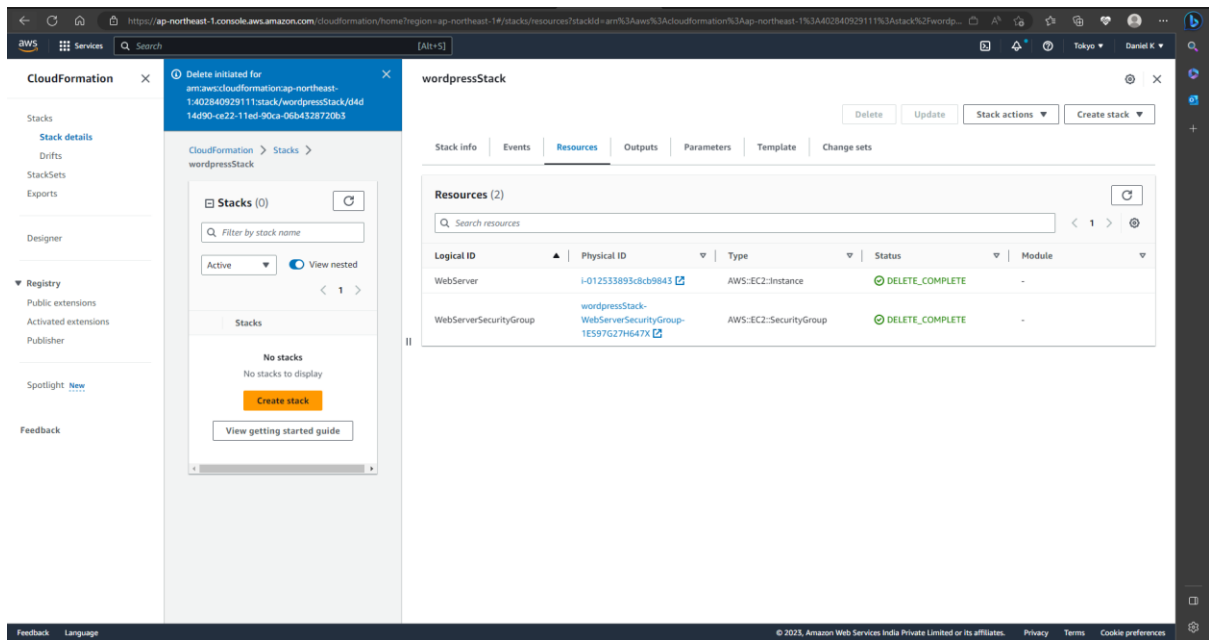
- Wordpress Stack has been successfully created.



- Close the current tab URL and go to the cloudformation stack, again open the URL in browser it will show the provided wordpress name.



- Deleting the stack, it will delete the created ec2 instance as we can see in resource.



- Instance is also deleted when we delete the stack.

