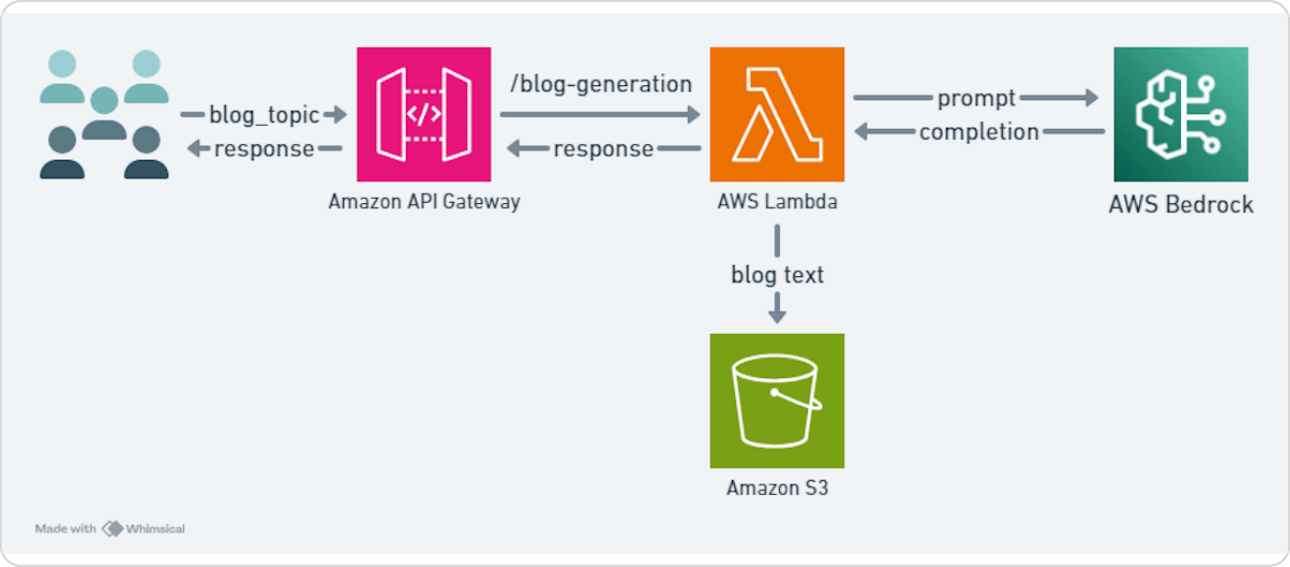
Using Amazon Bedrock

90 Steps View most recent version on Tango.us

Created by	Creation Date	Last Updated
Dasaraju Abhishek Varma	May 14, 2024	May 14, 2024

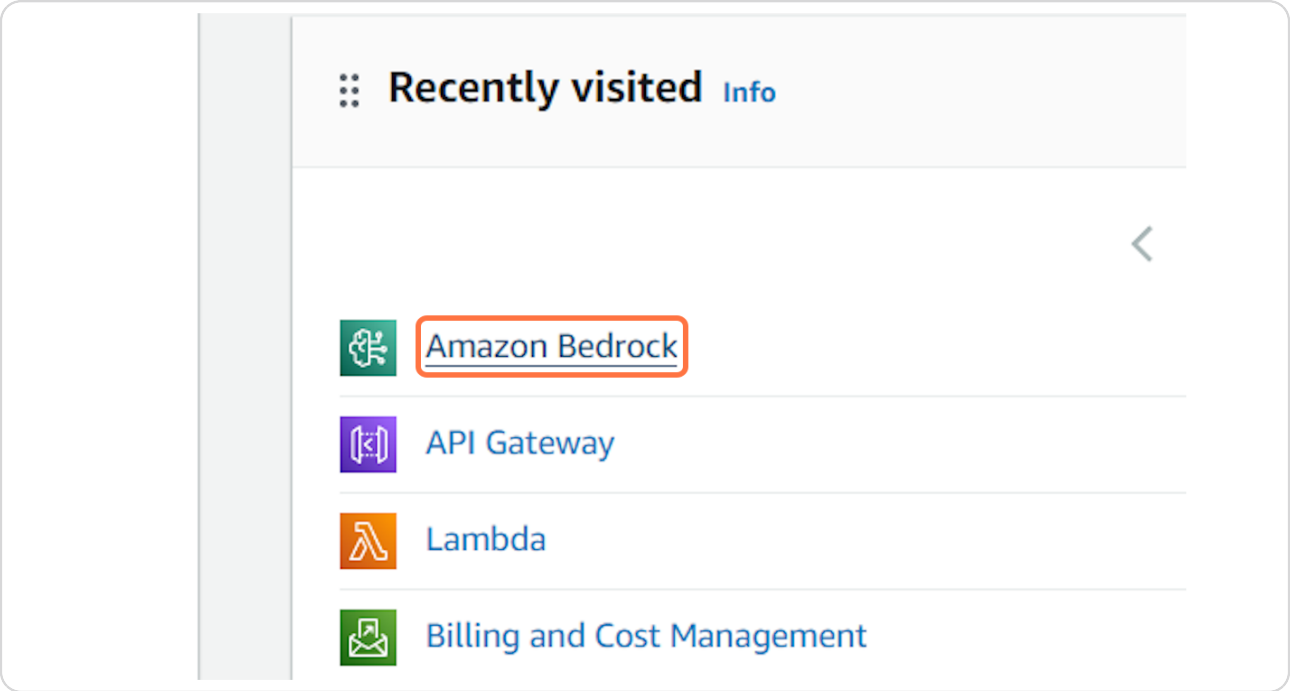
STEP 1

Guide to use AWS ecosystem to build LLM powered applications



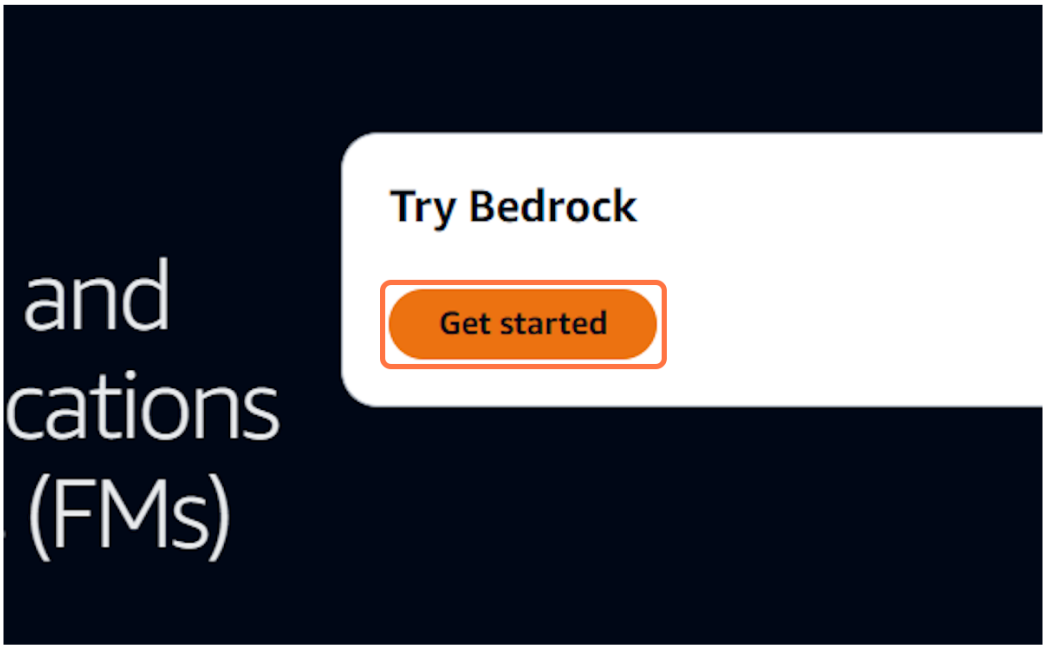
STEP 2

Click on Amazon Bedrock



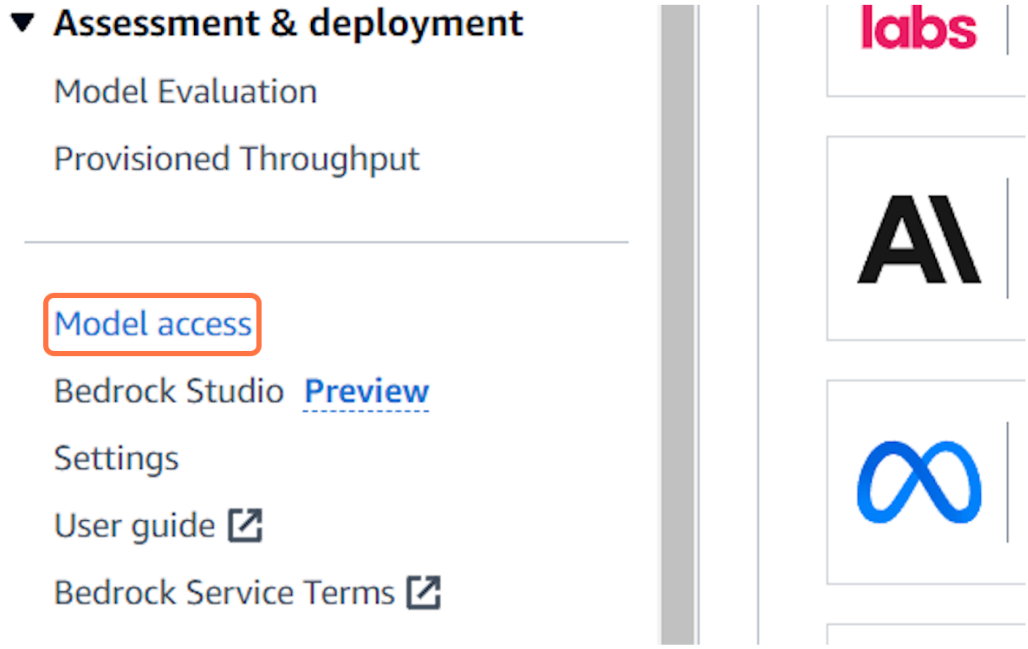
STEP 3

Click on Get started



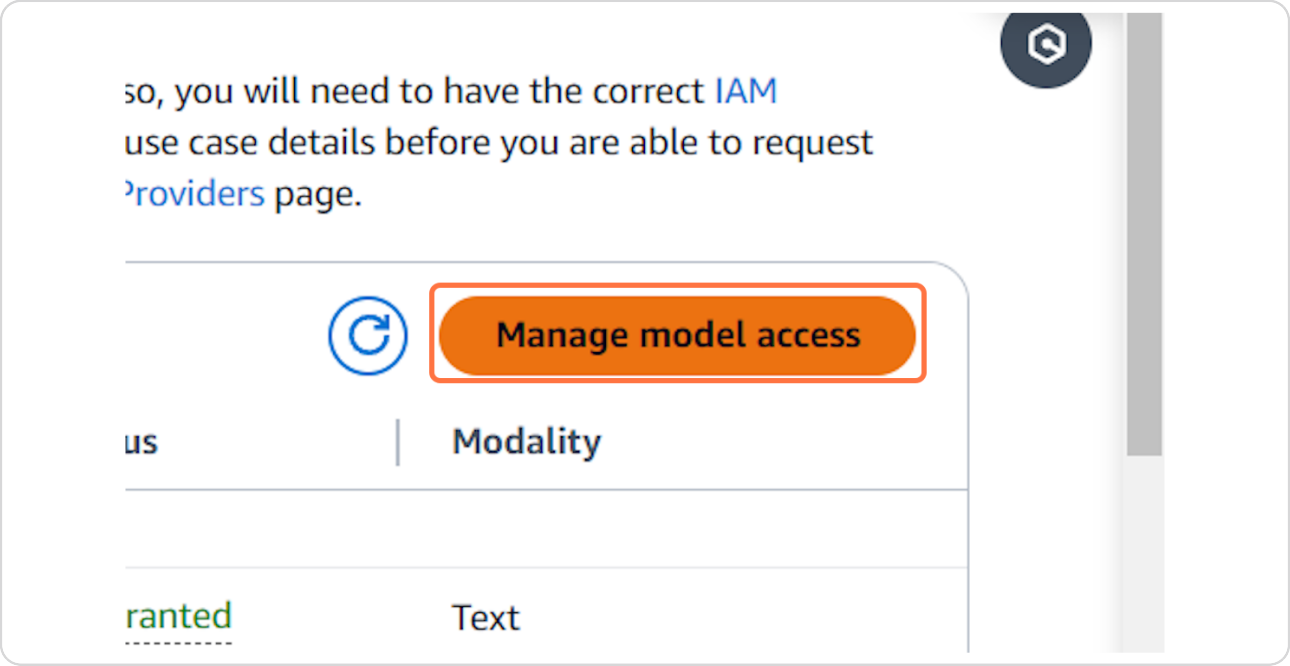
STEP 4

Click on Model access



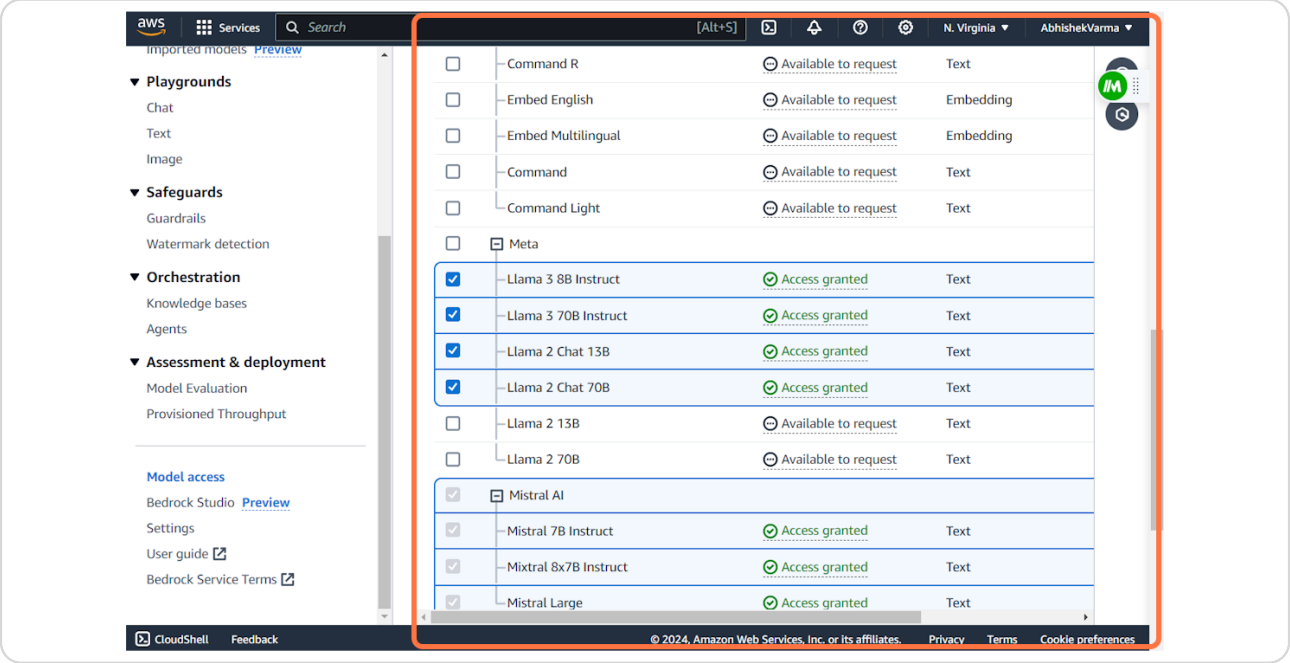
STEP 5

Click on Manage model access



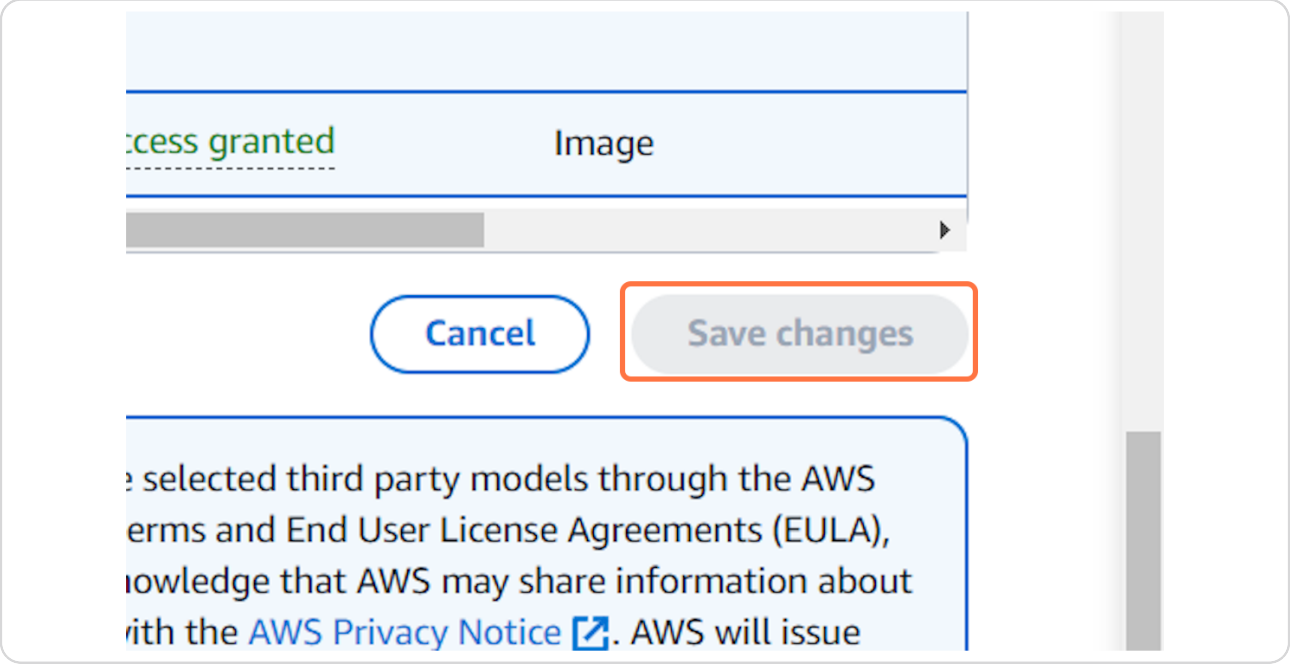
STEP 6

Click on Base models



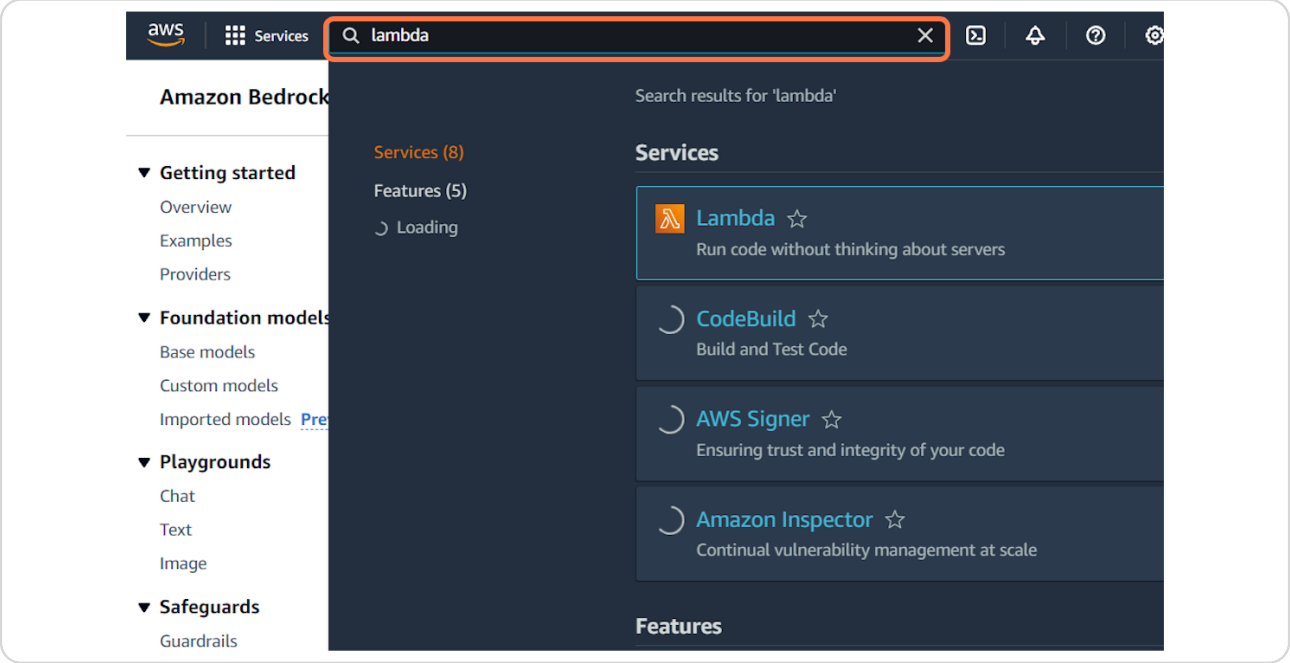
STEP 7

Click on Save changes



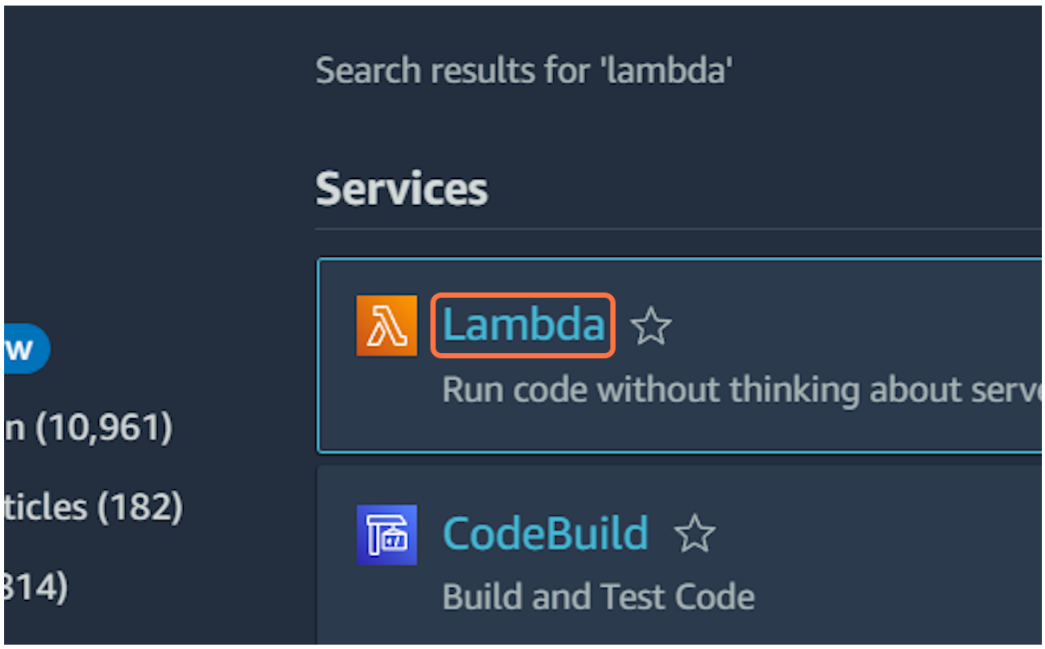
STEP 8

Type "lambda"



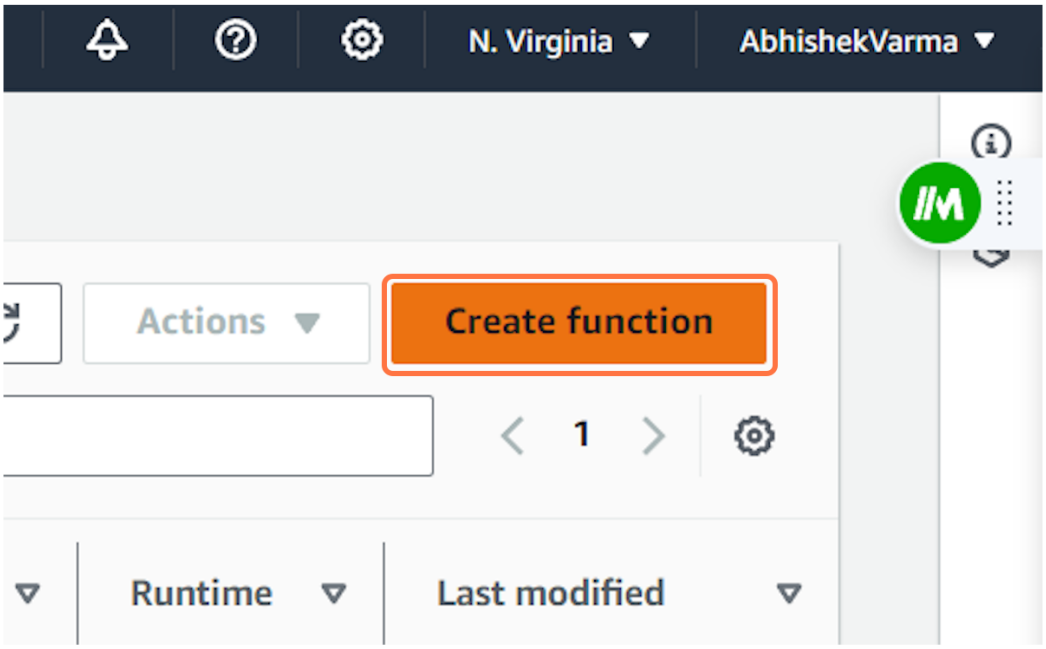
STEP 9

Click on Lambda



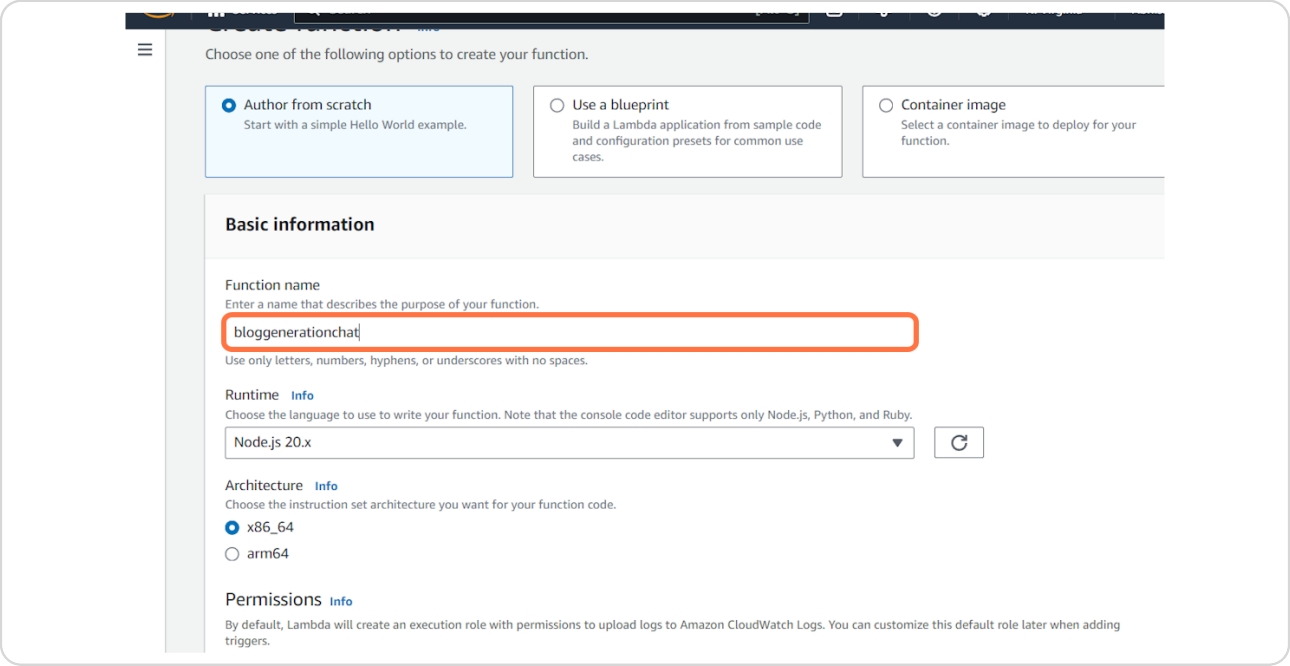
STEP 10

Click on Create function



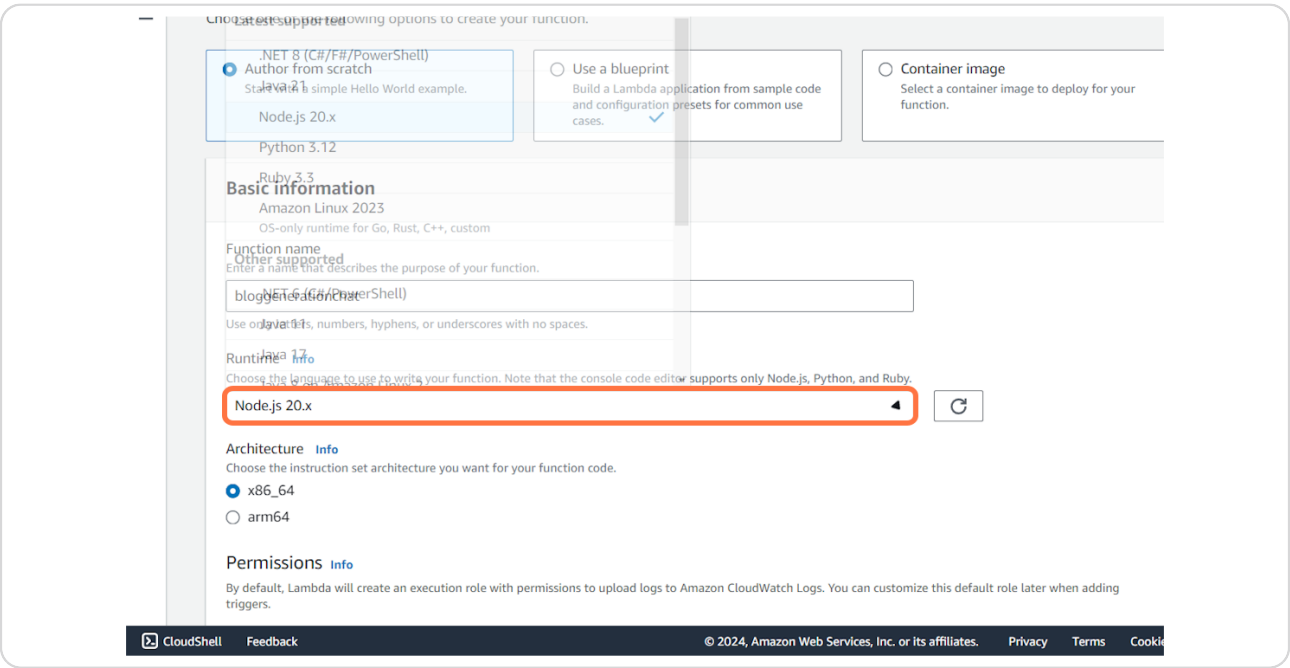
STEP 11

Type "bloggenerationchat"



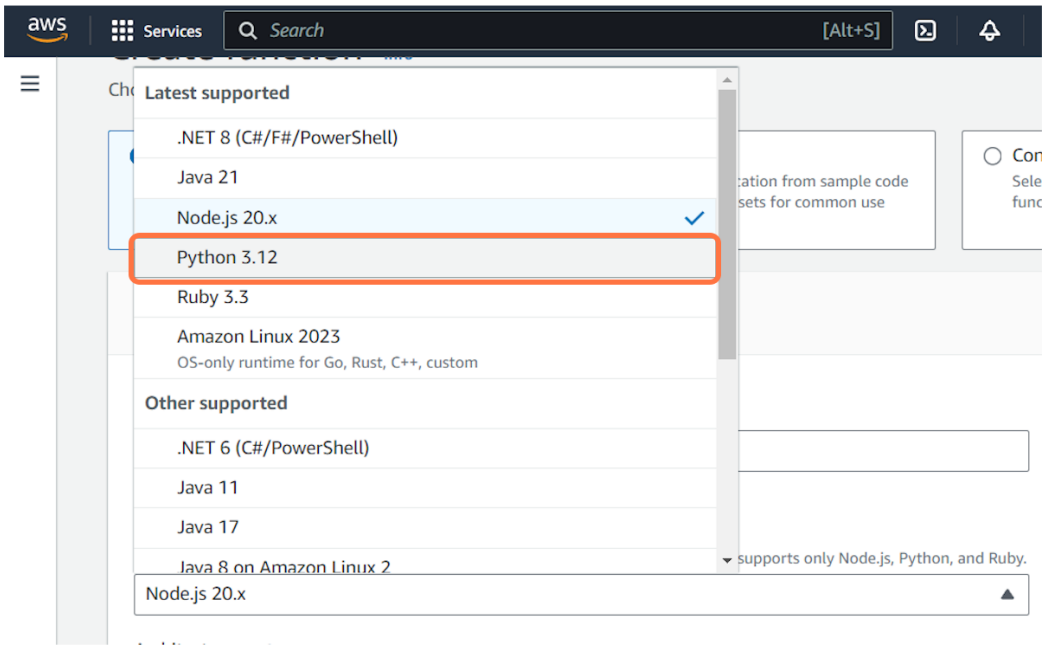
STEP 12

Click on Node.js 20.x



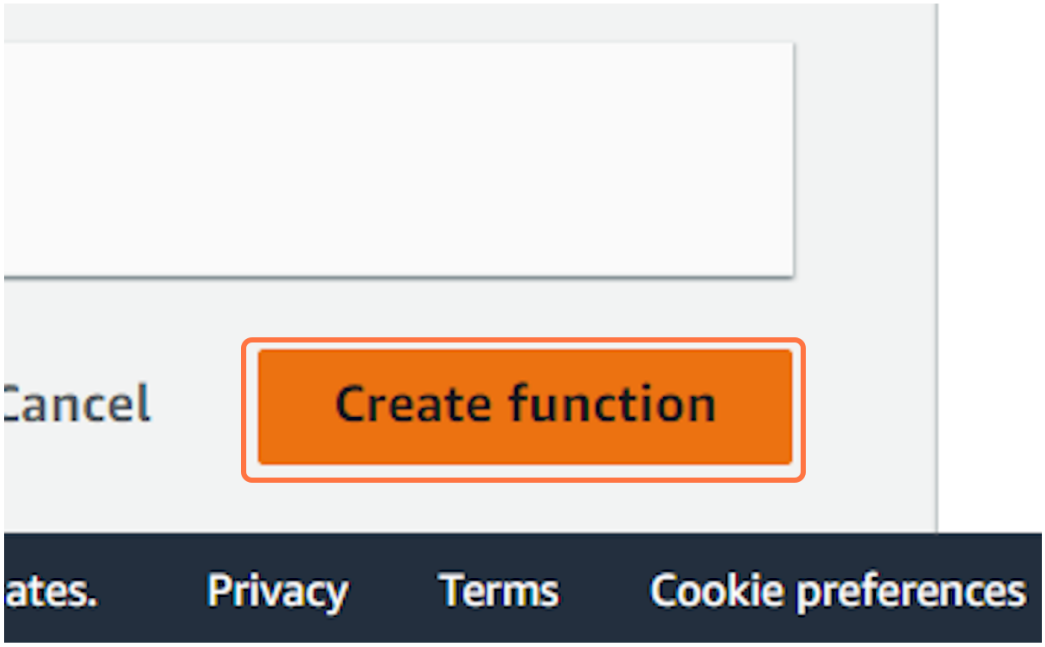
STEP 13

Click on Python 3.12...



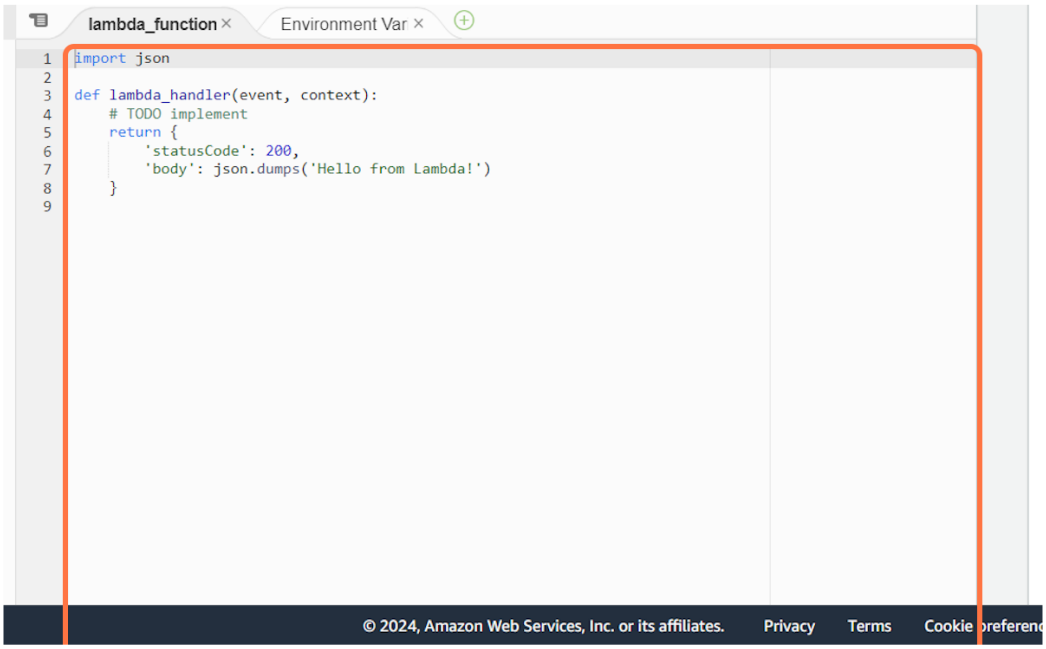
STEP 14

Click on Create function



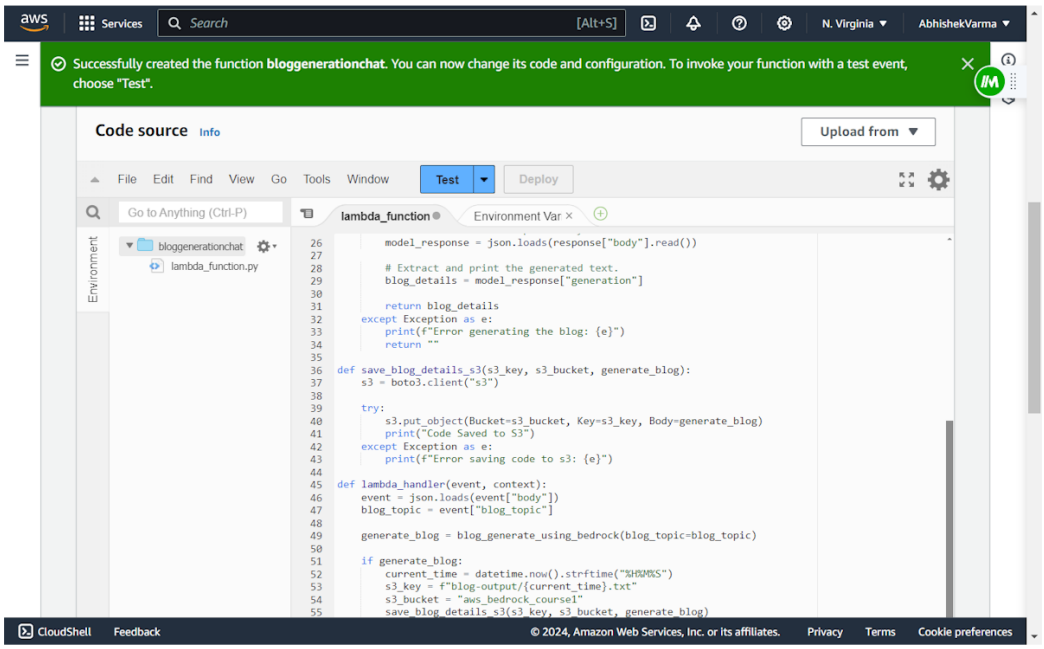
STEP 15

Go to the code tab



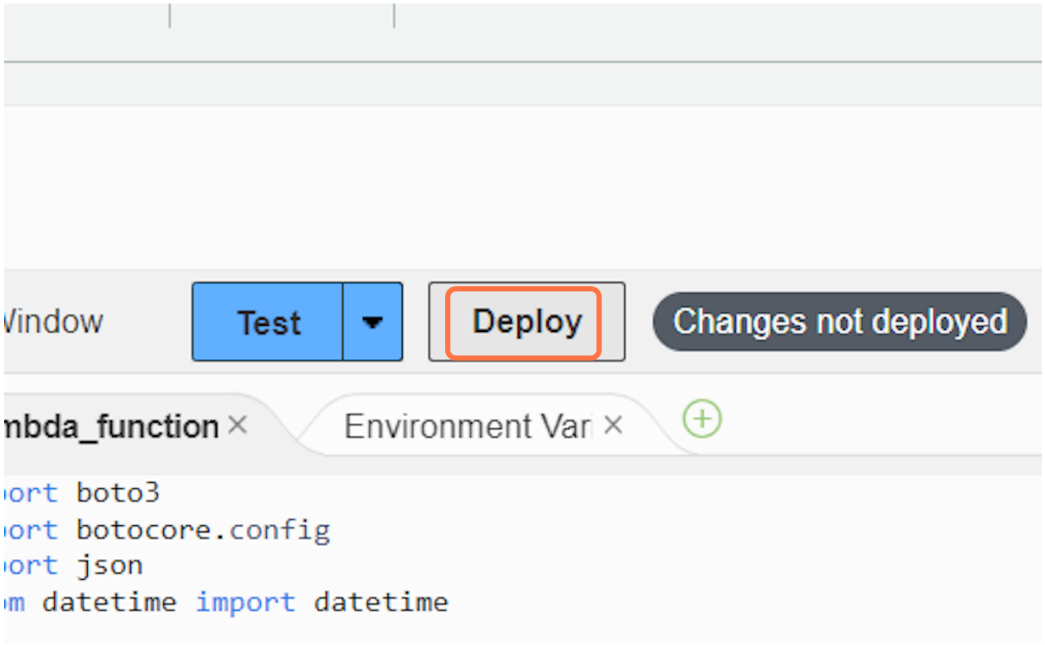
STEP 16

Paste the code in app.py to lambda function



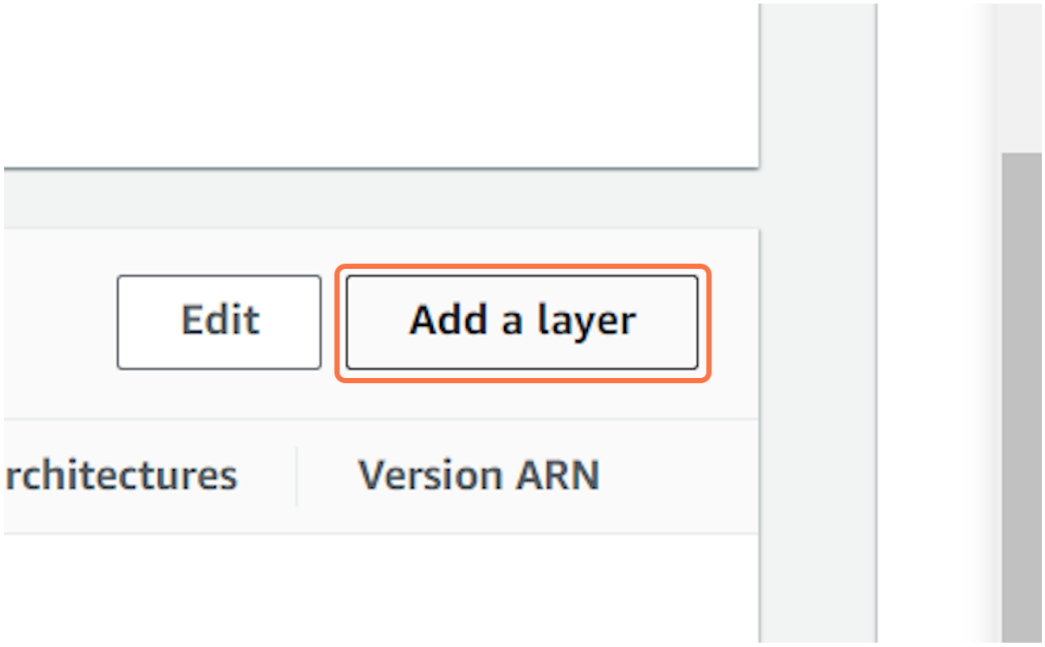
STEP 17

Click on Deploy



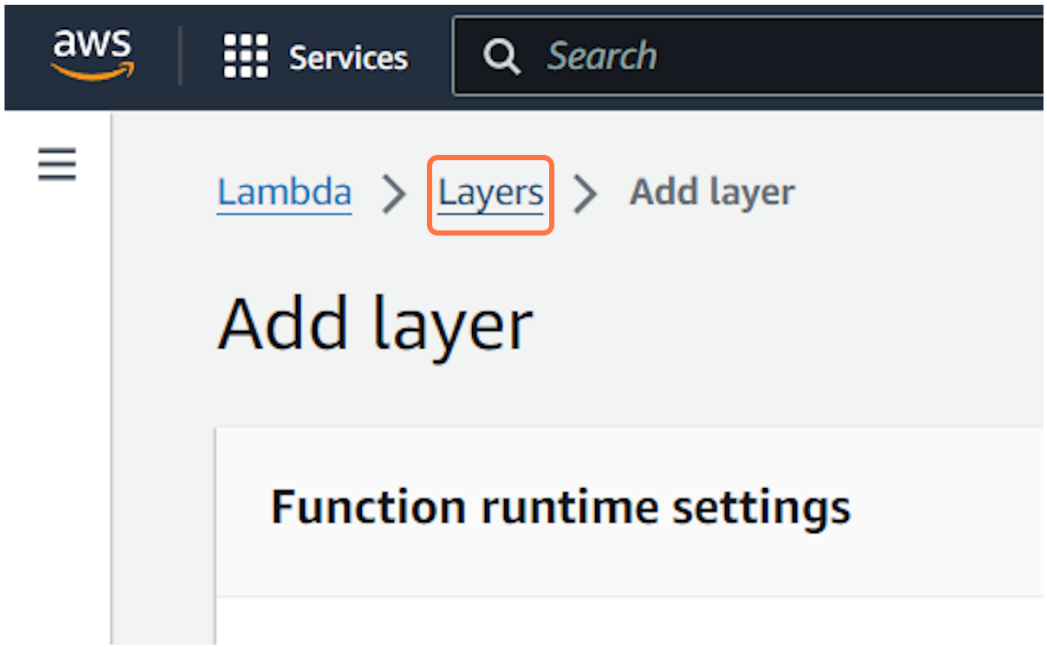
STEP 18

Click on Add a layer



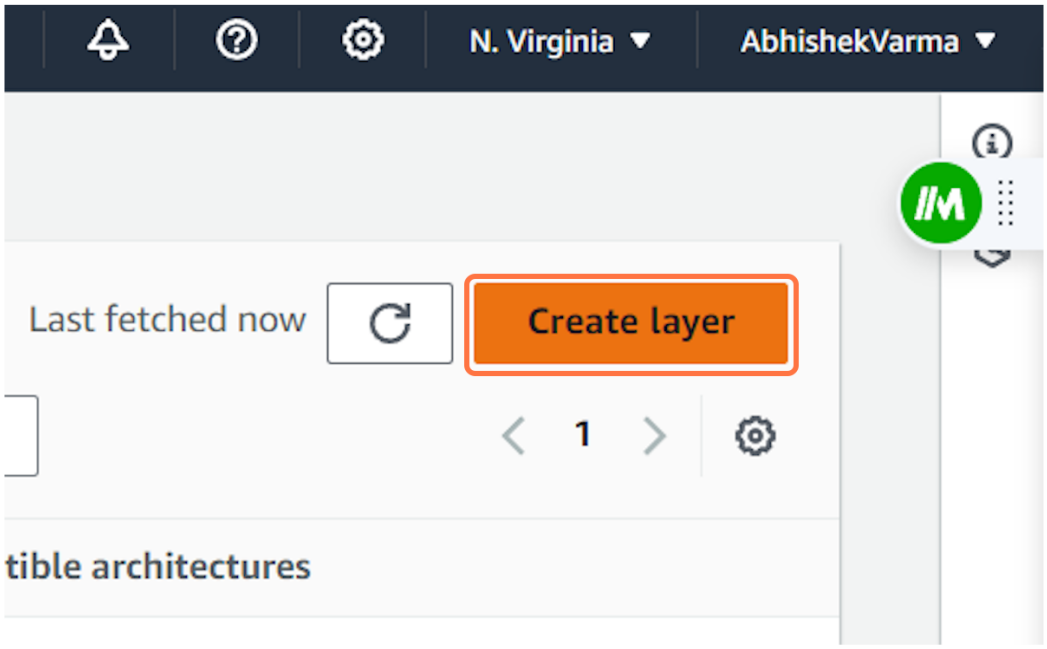
STEP 19

Click on Layers



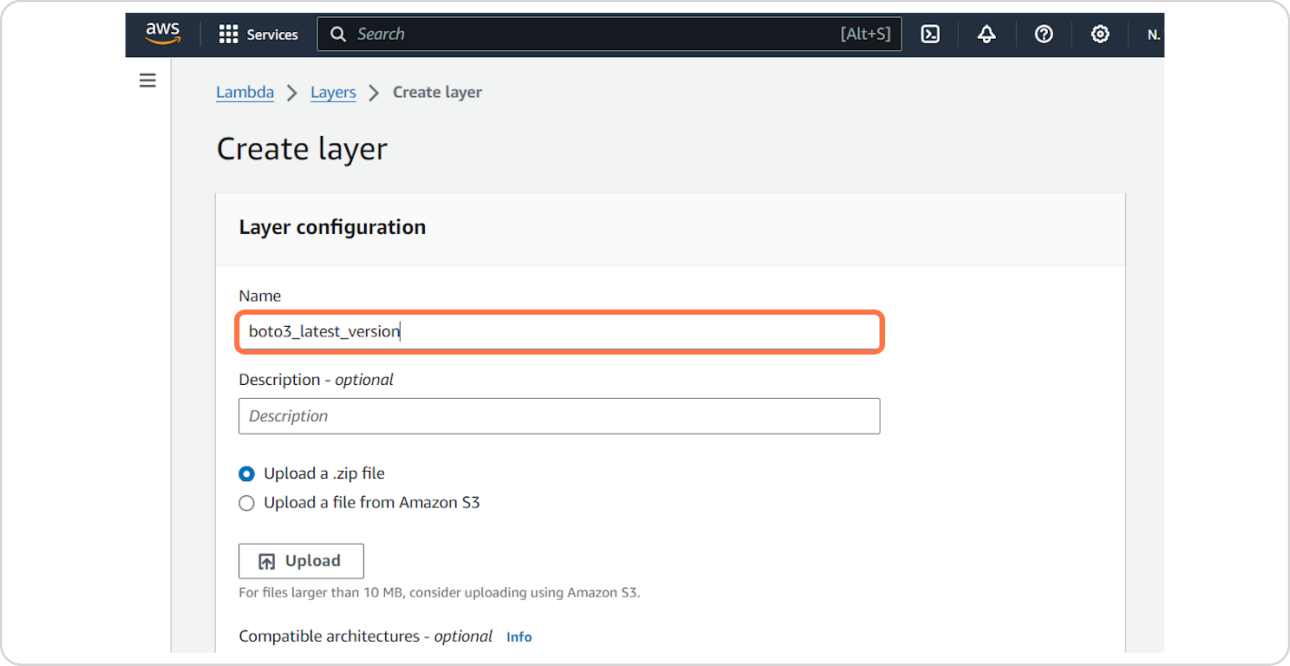
STEP 20

Click on Create layer



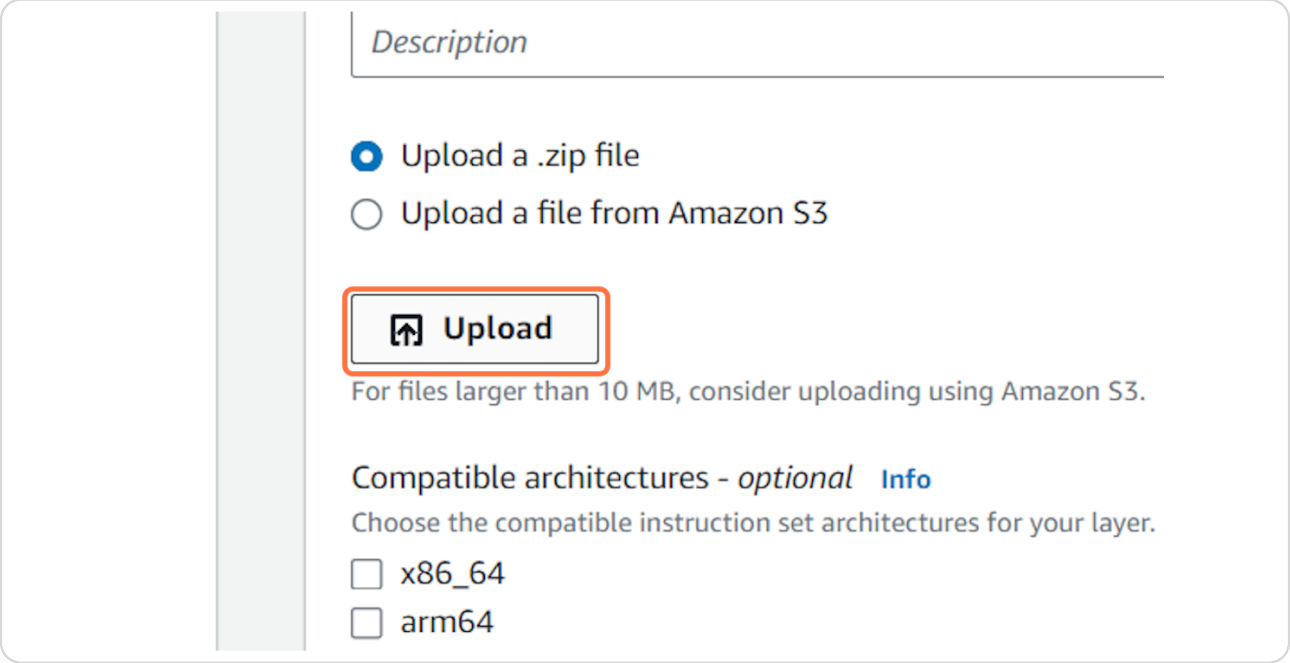
STEP 21

Type "boto3_latest_version"



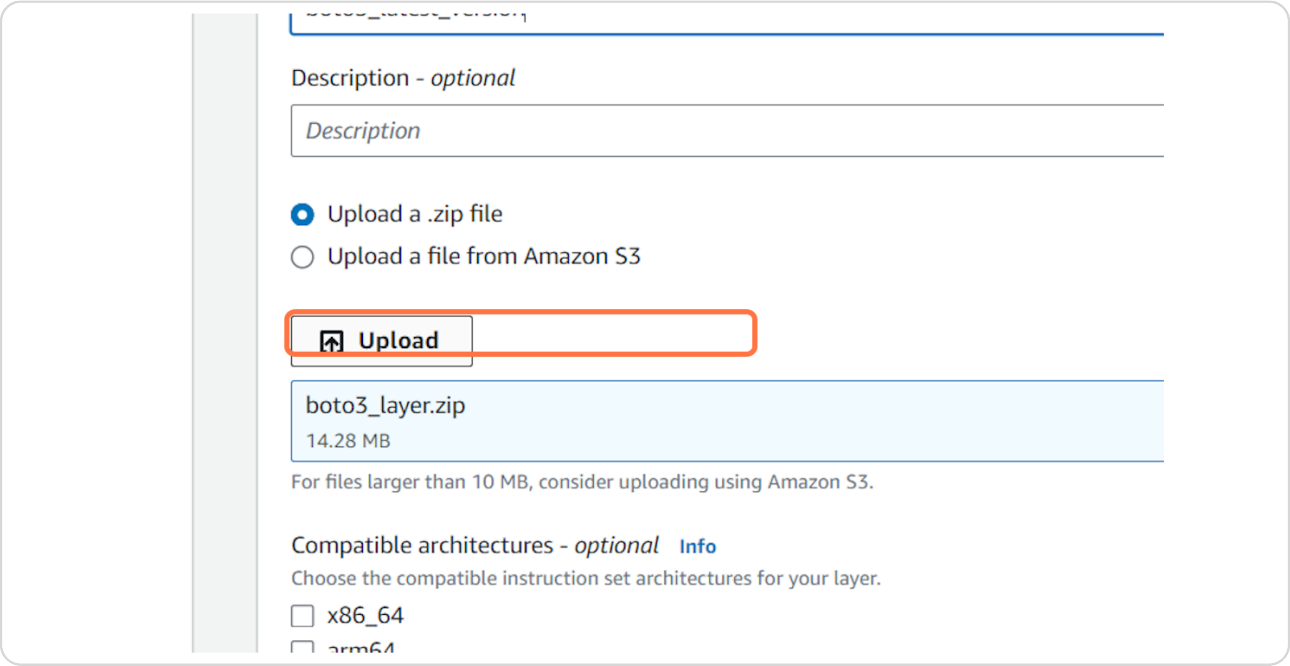
STEP 22

Click on Upload



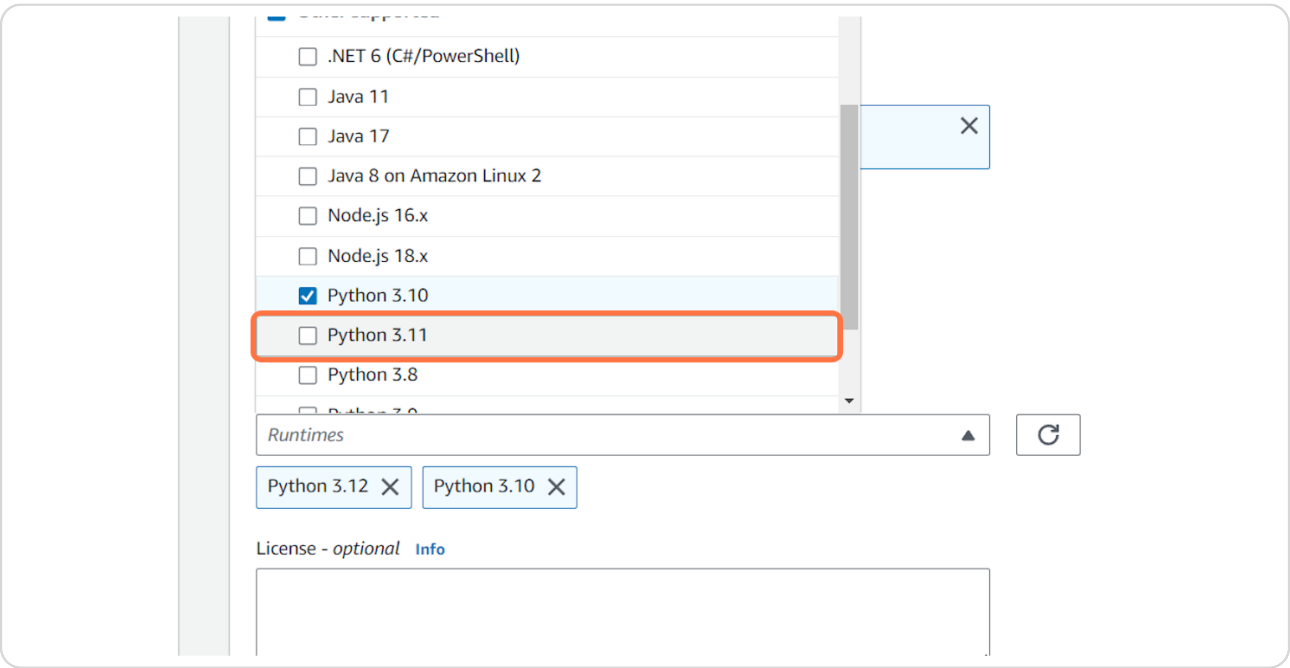
STEP 23

Select boto3_layer.zip file from local



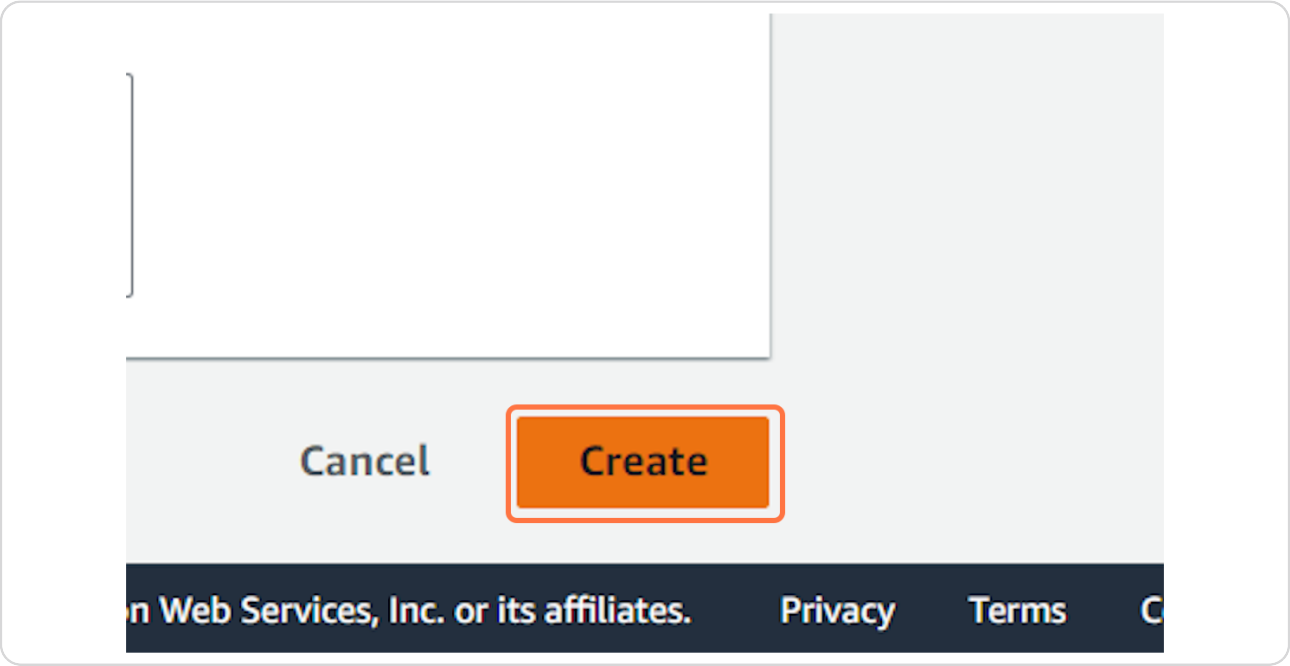
STEP 24

Click on Runtimes and select Python 3.10, 3.11, 3.12



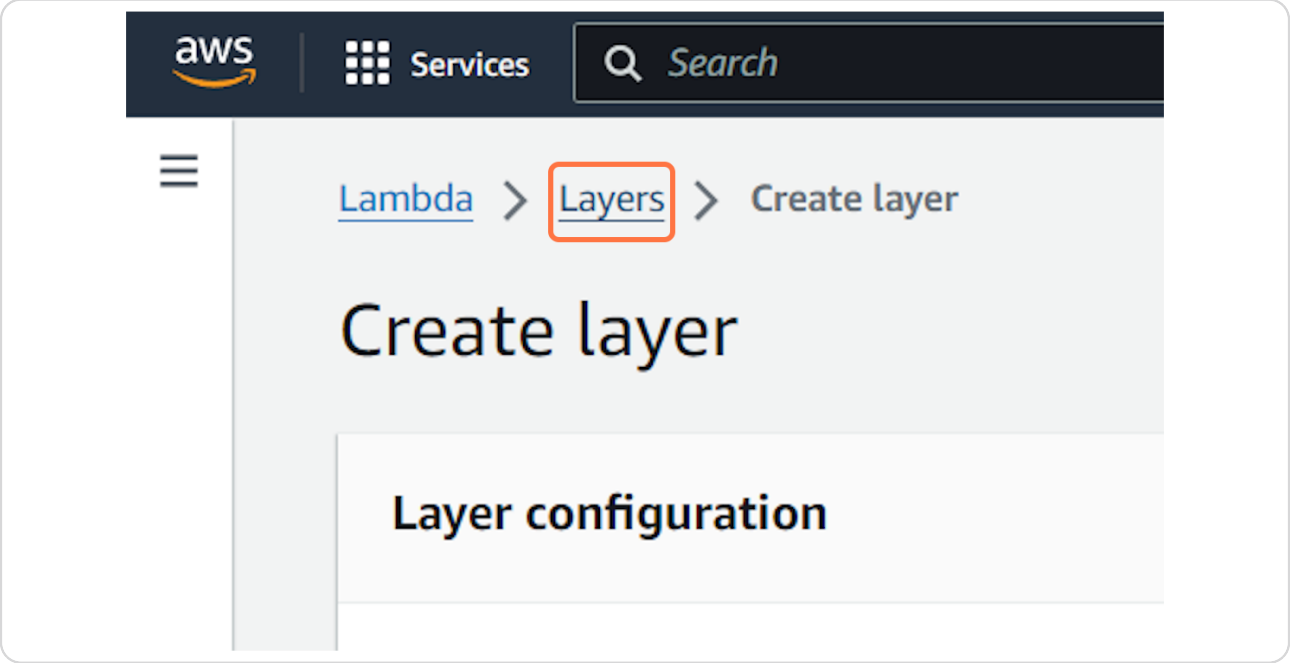
STEP 25

Click on Create



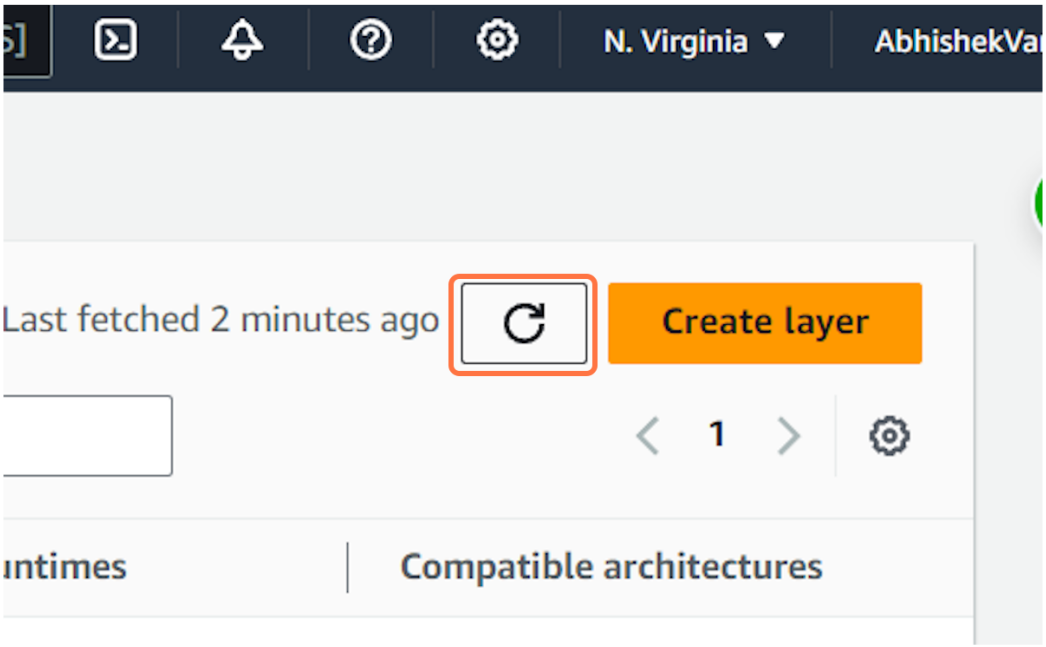
STEP 26

Click on Layers



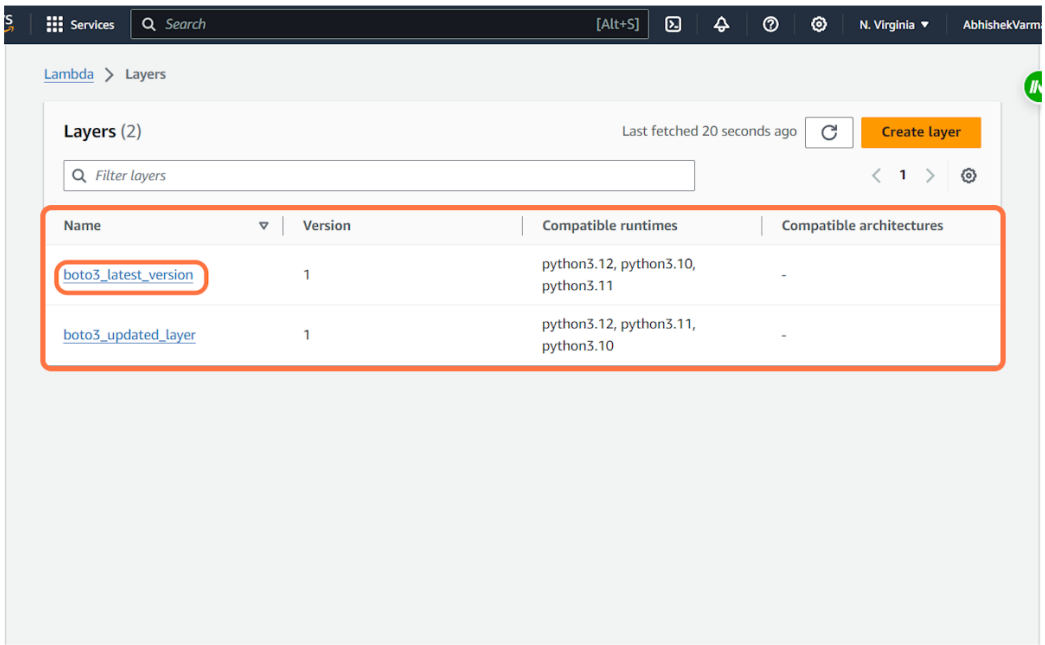
STEP 27

Click on Refresh



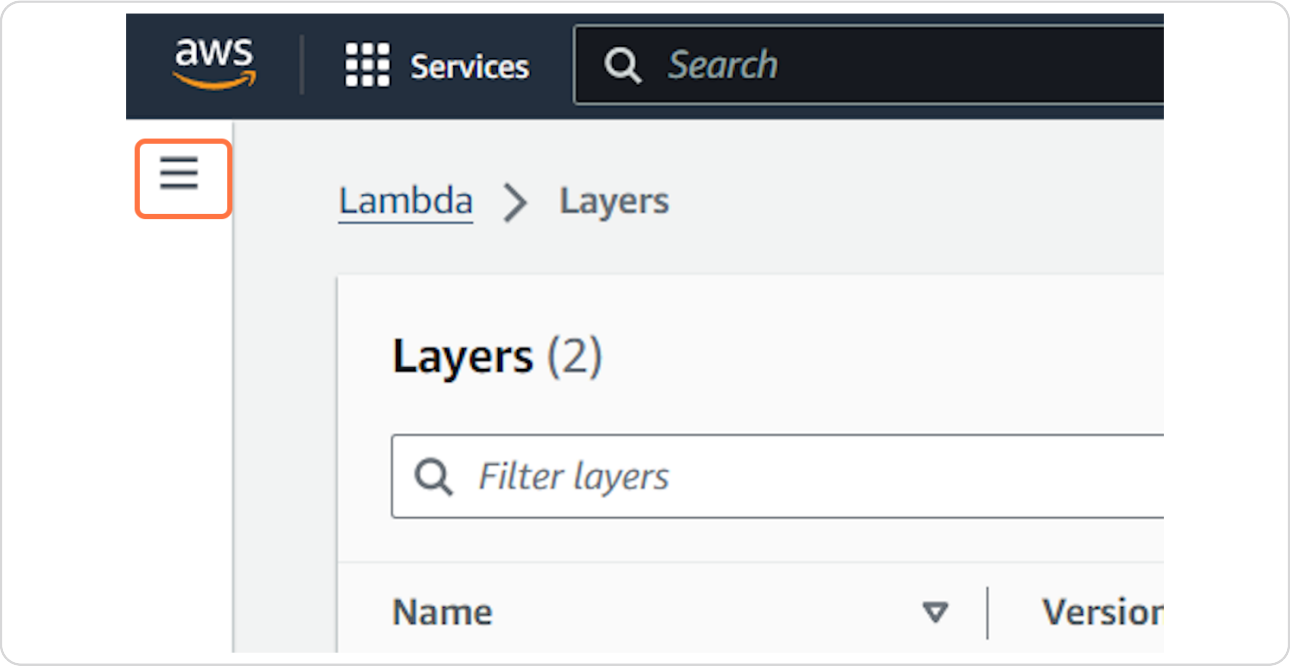
STEP 28

See if the layer created successfully



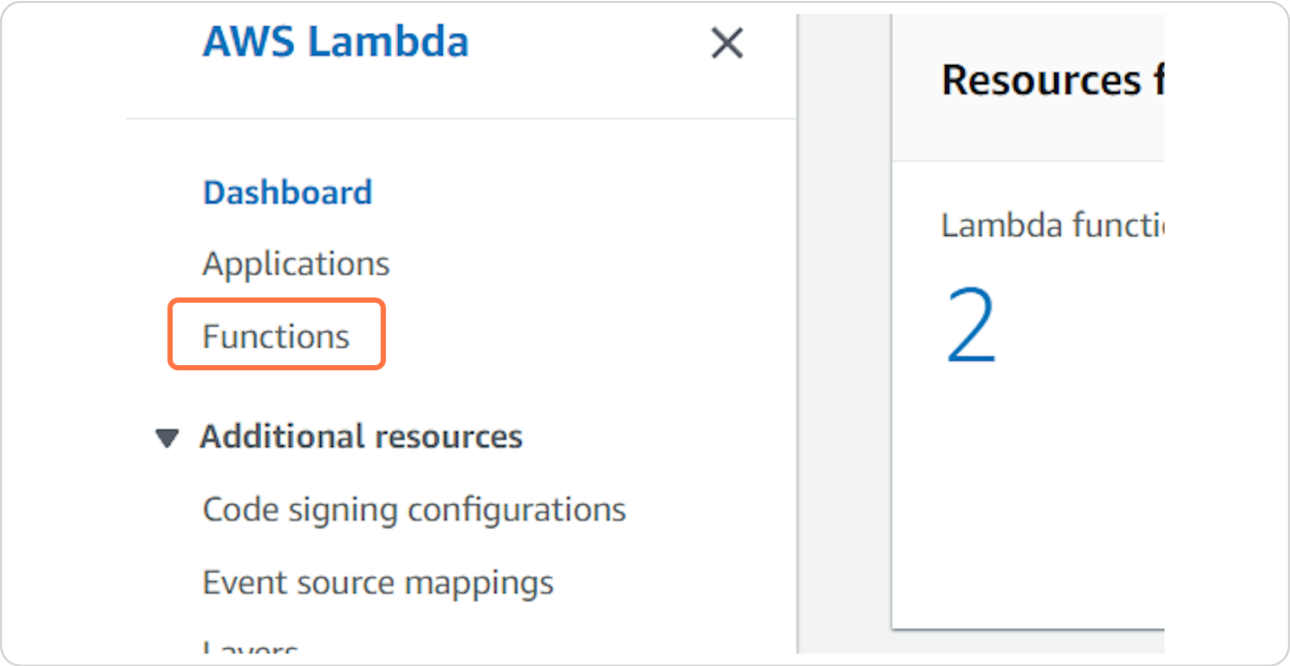
STEP 29

Click on burger menu



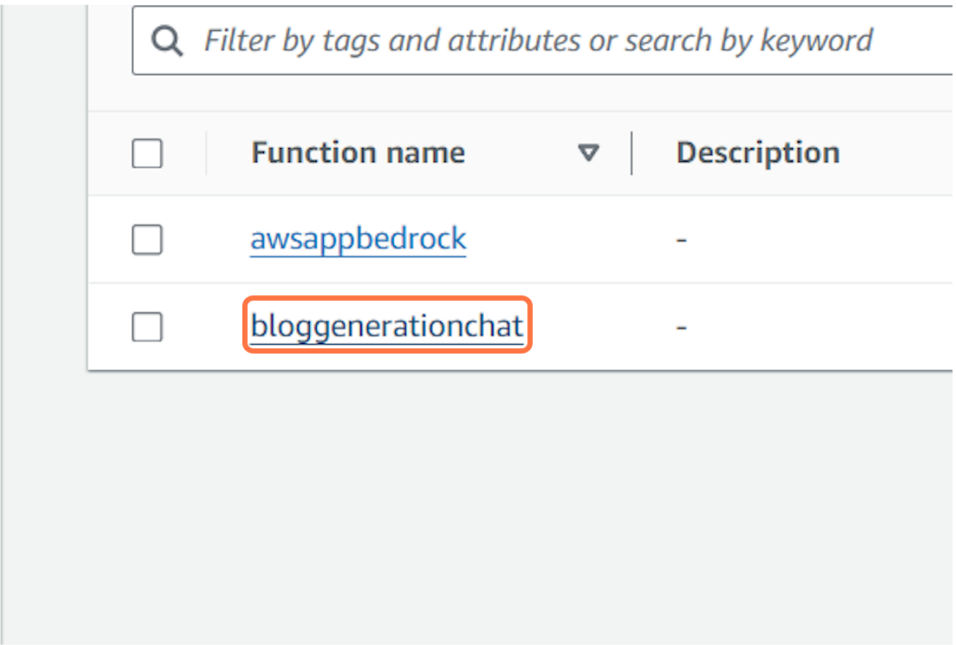
STEP 30

Click on Functions



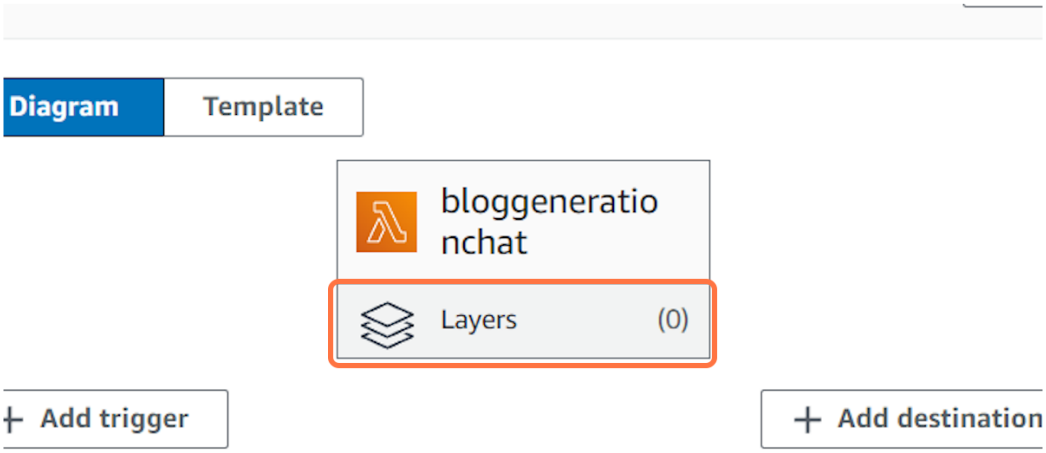
STEP 31

Click on bloggenerationchat



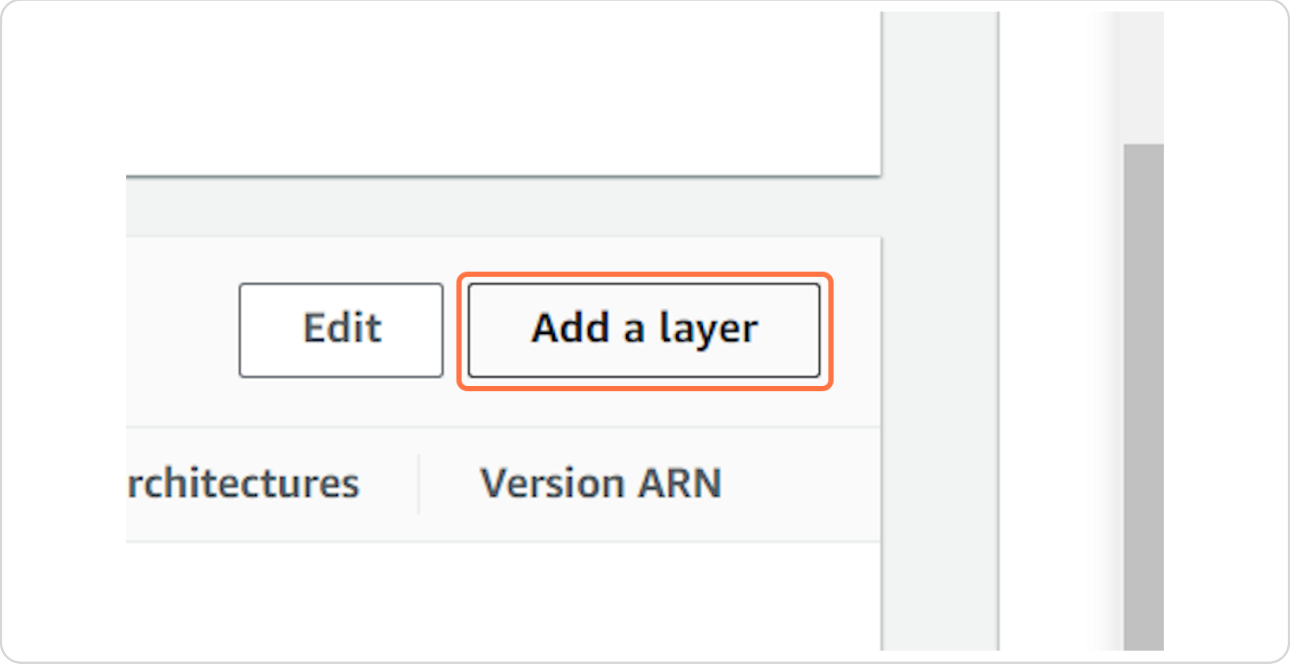
STEP 32

Click on Layers...



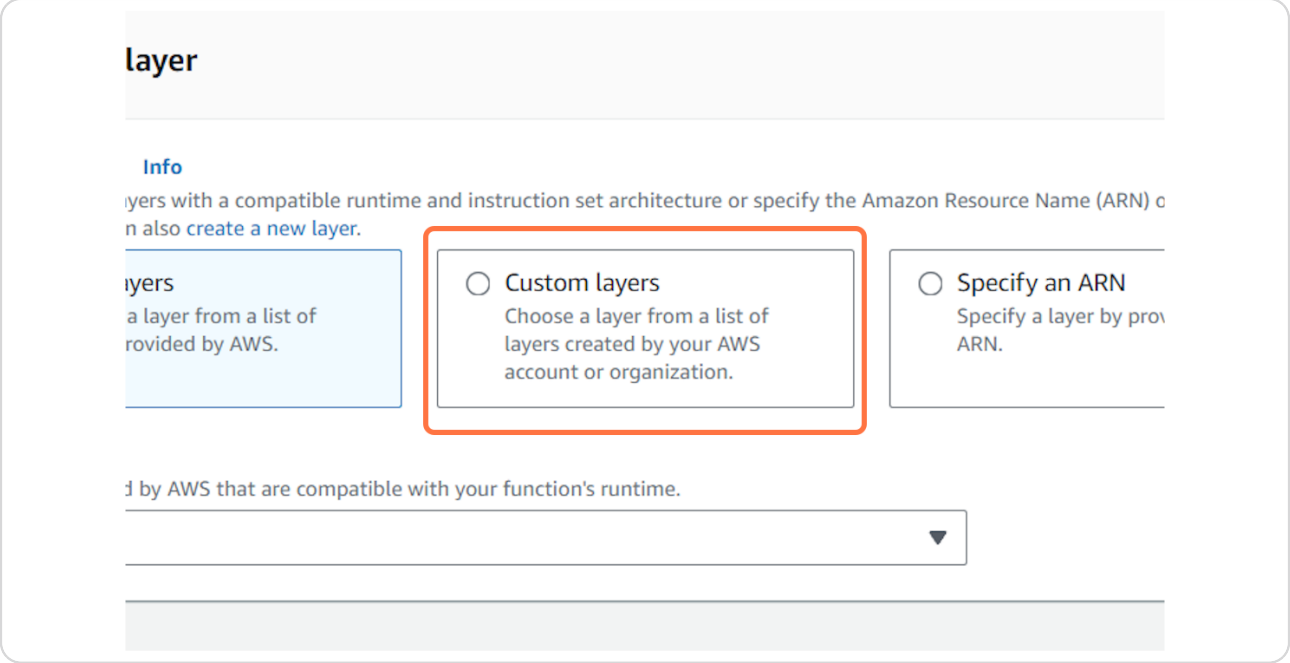
STEP 33

Click on Add a layer



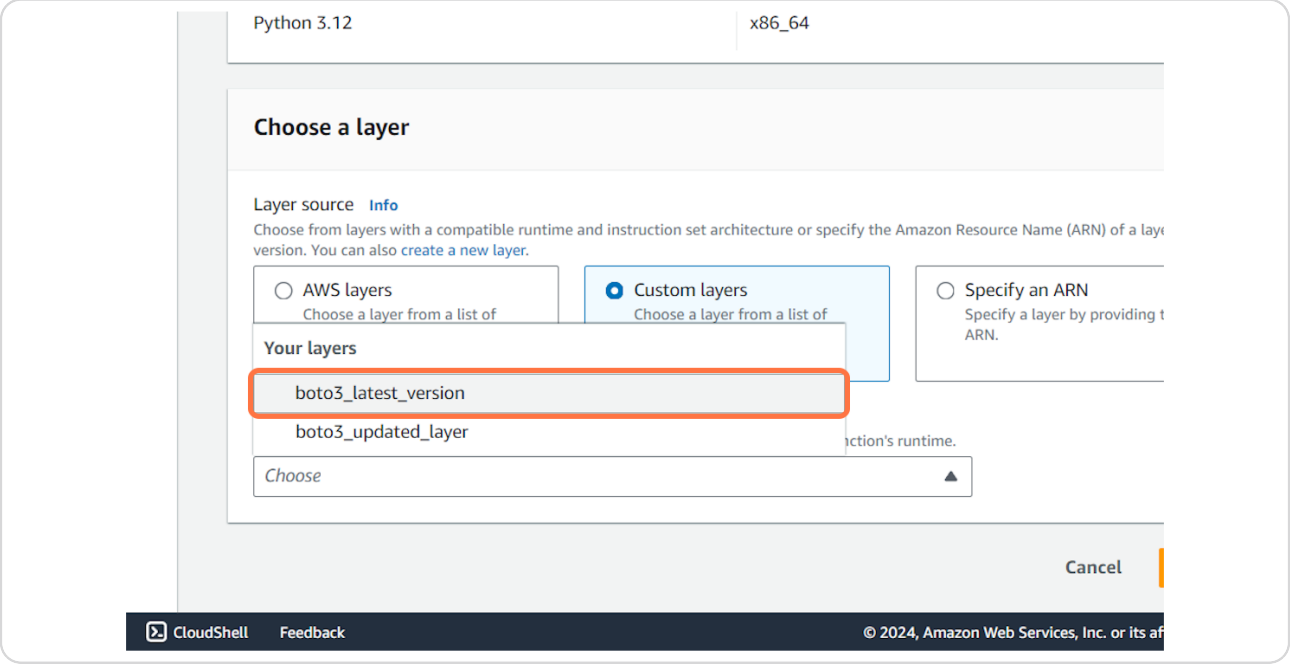
STEP 34

Click on Layer custom layers



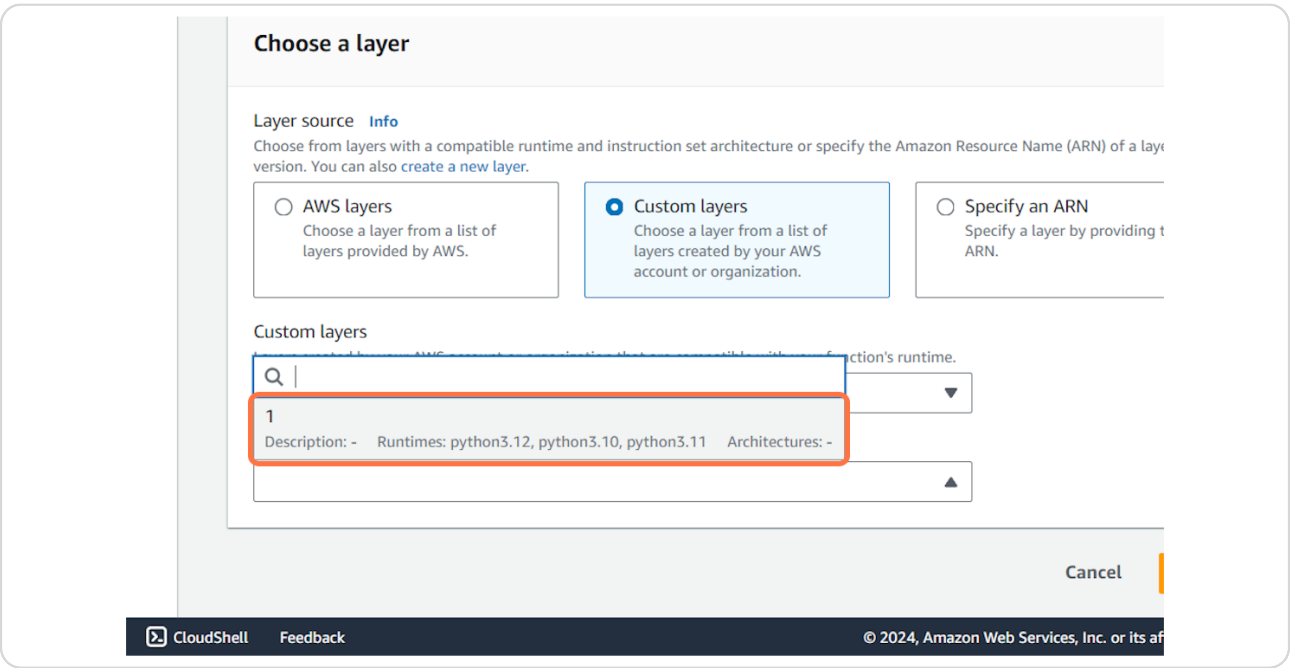
STEP 35

Choose boto3_latest_version



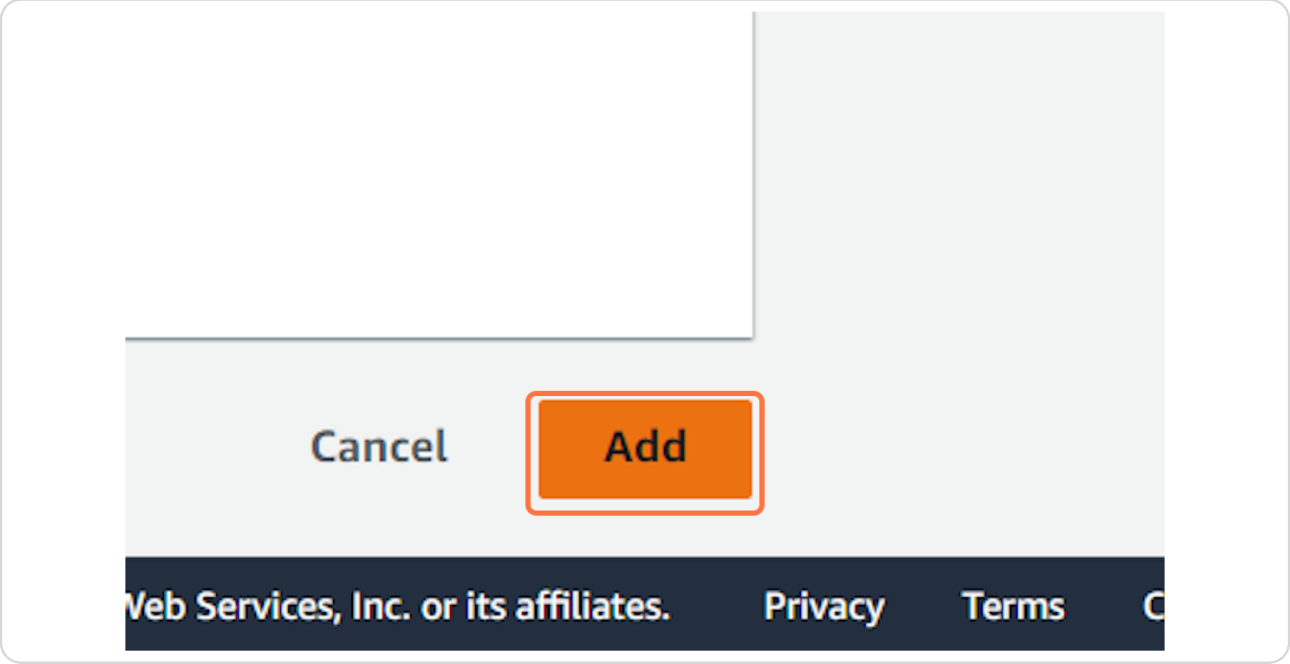
STEP 36

Choose version 1



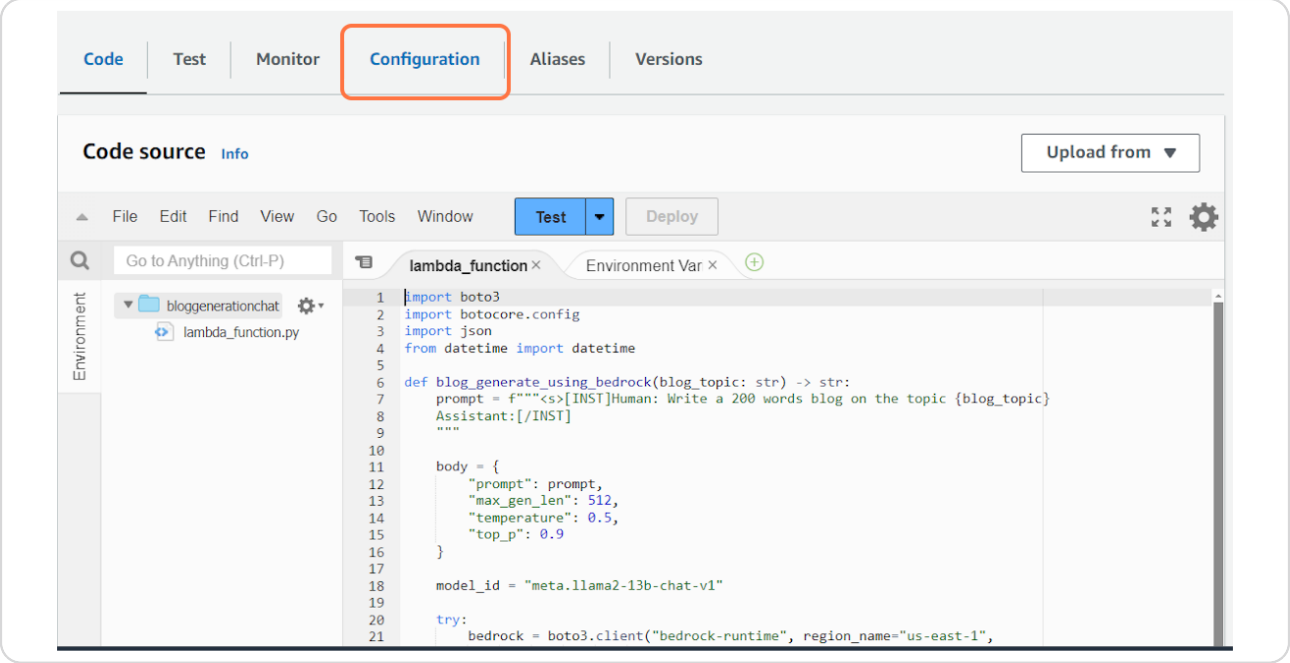
STEP 37

Click on Add



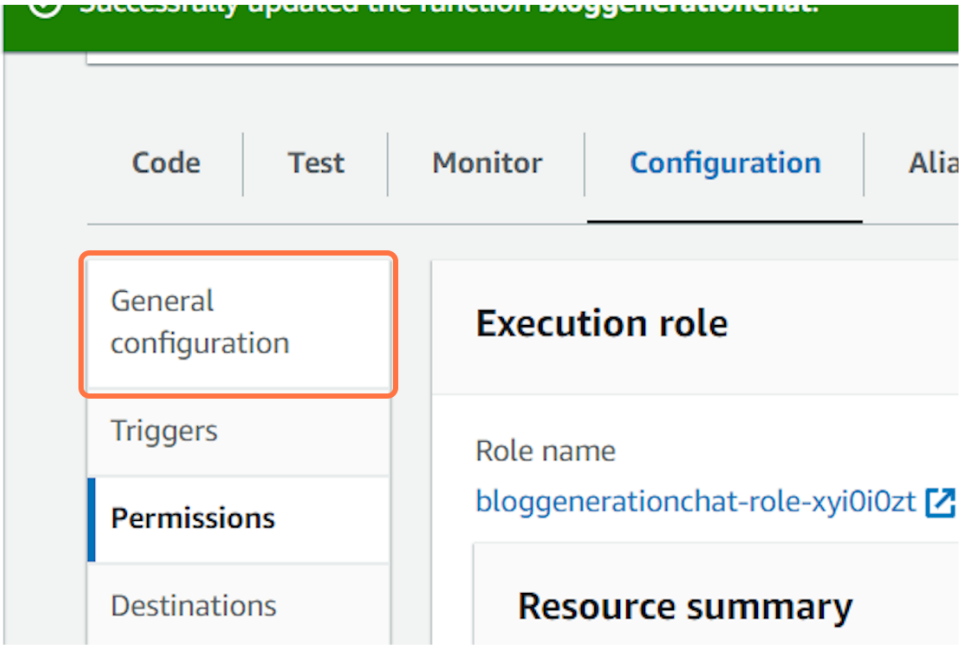
STEP 38

Click on Configuration



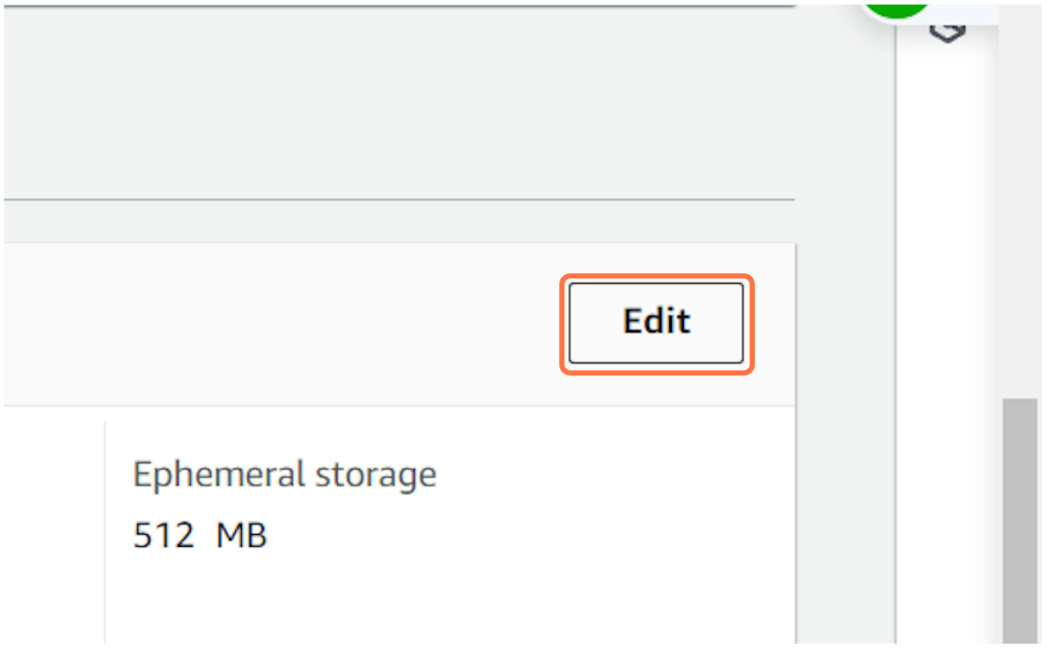
STEP 39

Click on General configuration



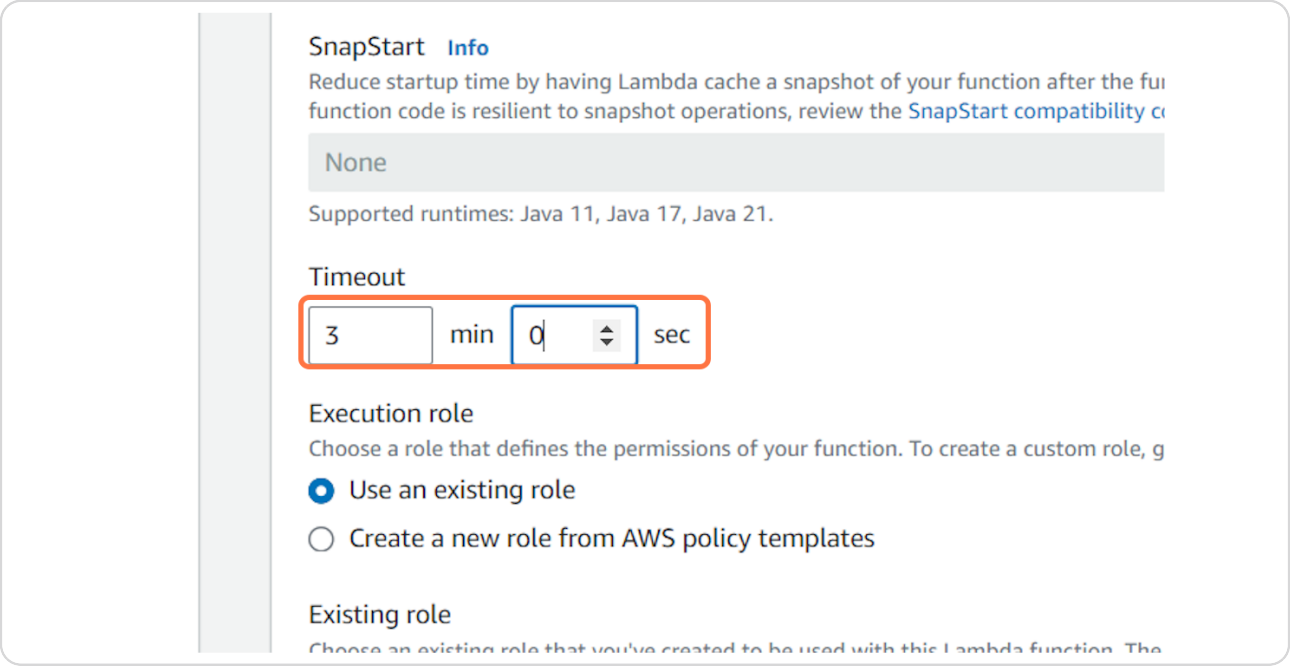
STEP 40

Click on Edit



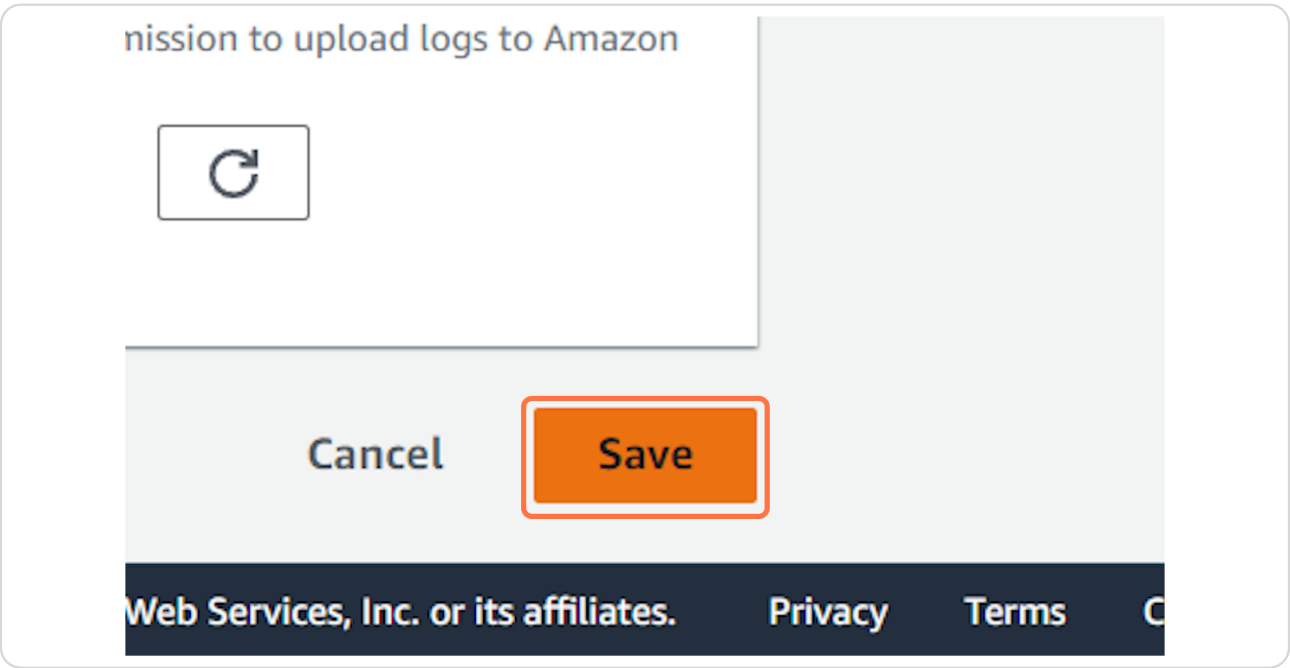
STEP 41

Set timeout to 3 mins



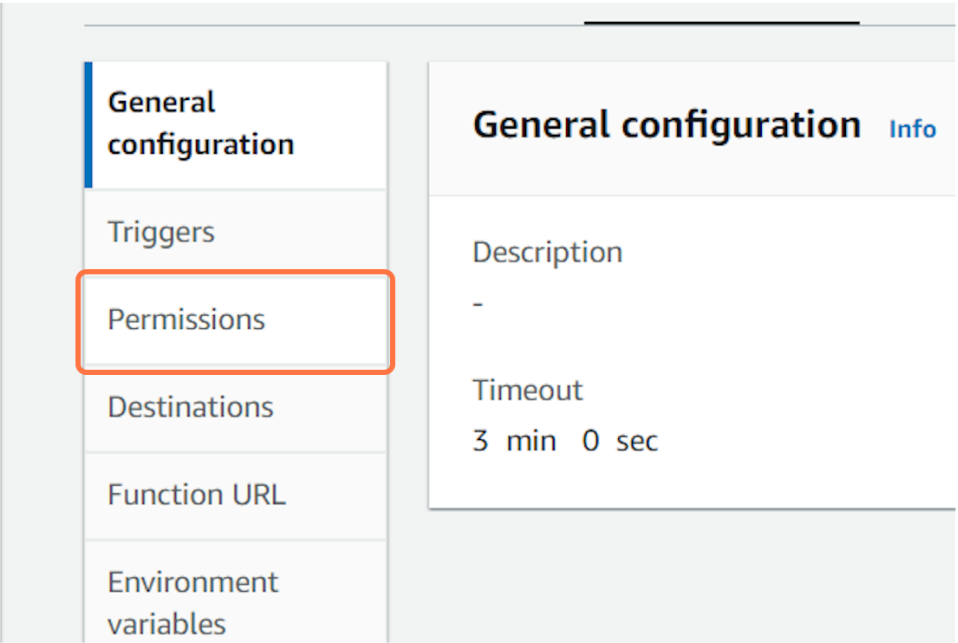
STEP 42

Click on Save



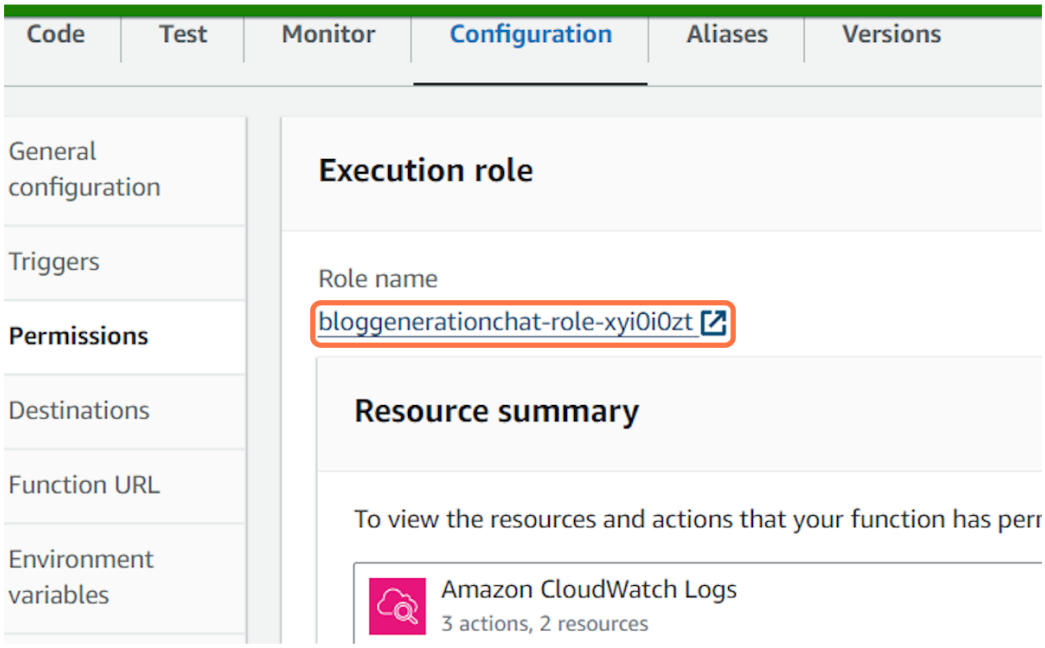
STEP 43

Click on Permissions



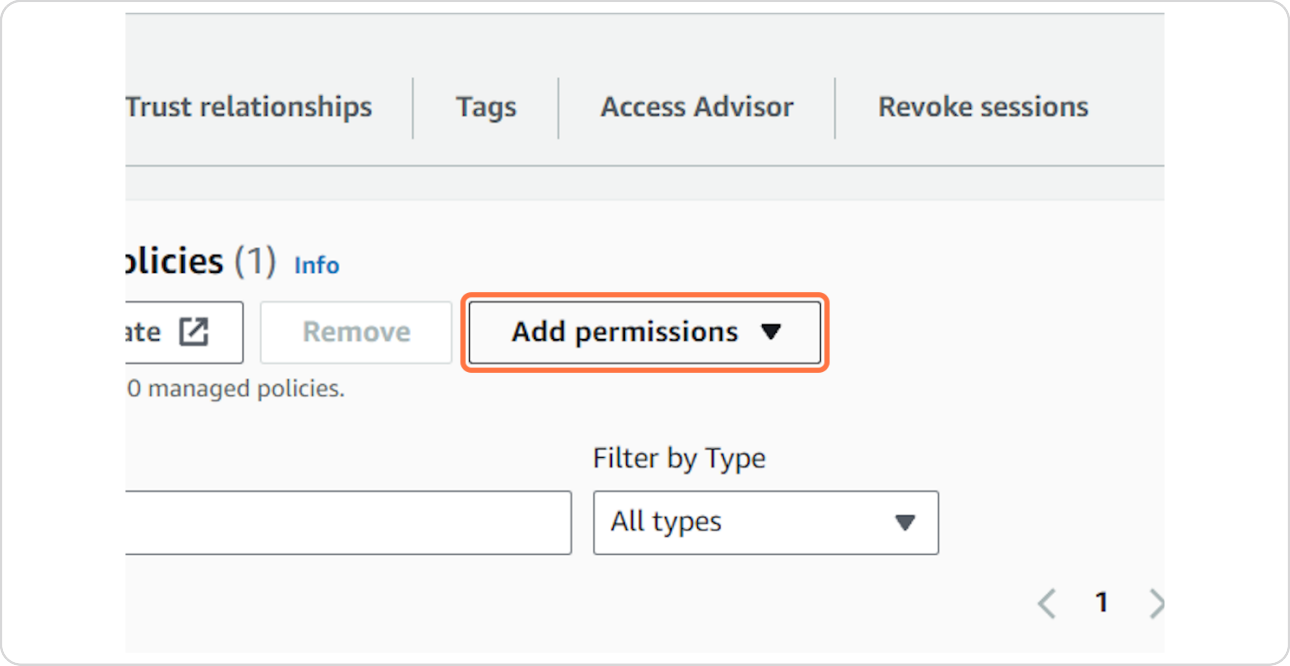
STEP 44

Click on role name



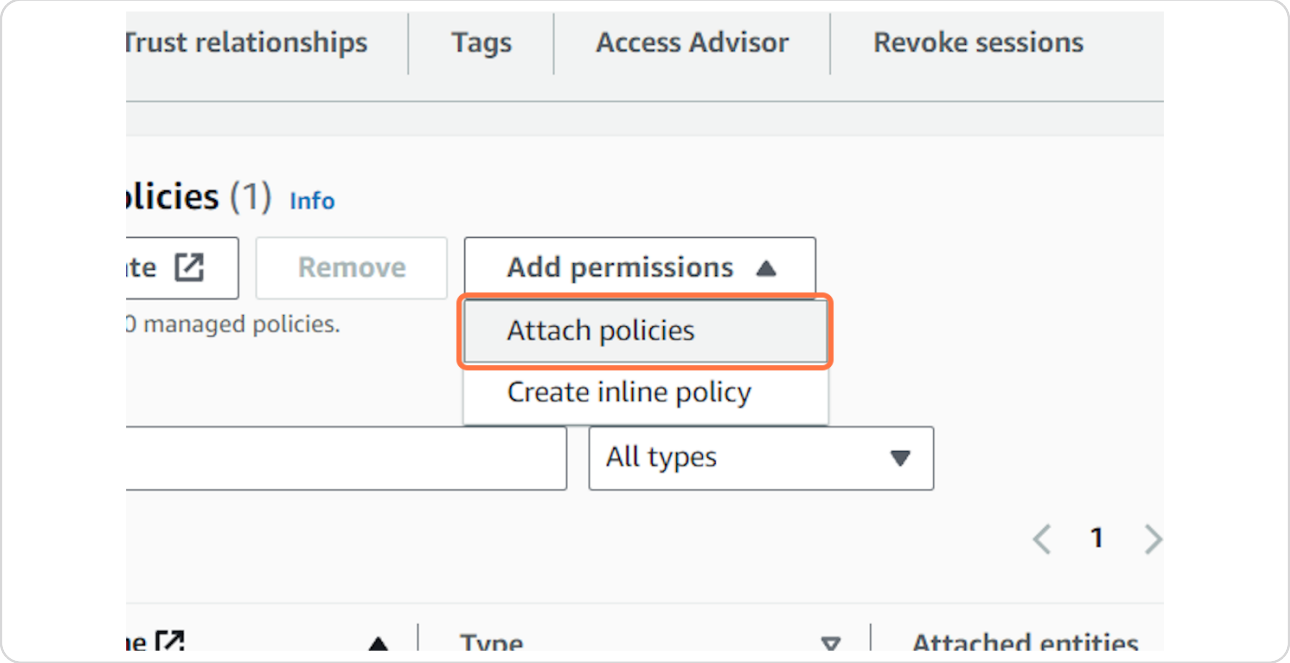
STEP 45

Click on Add permissions



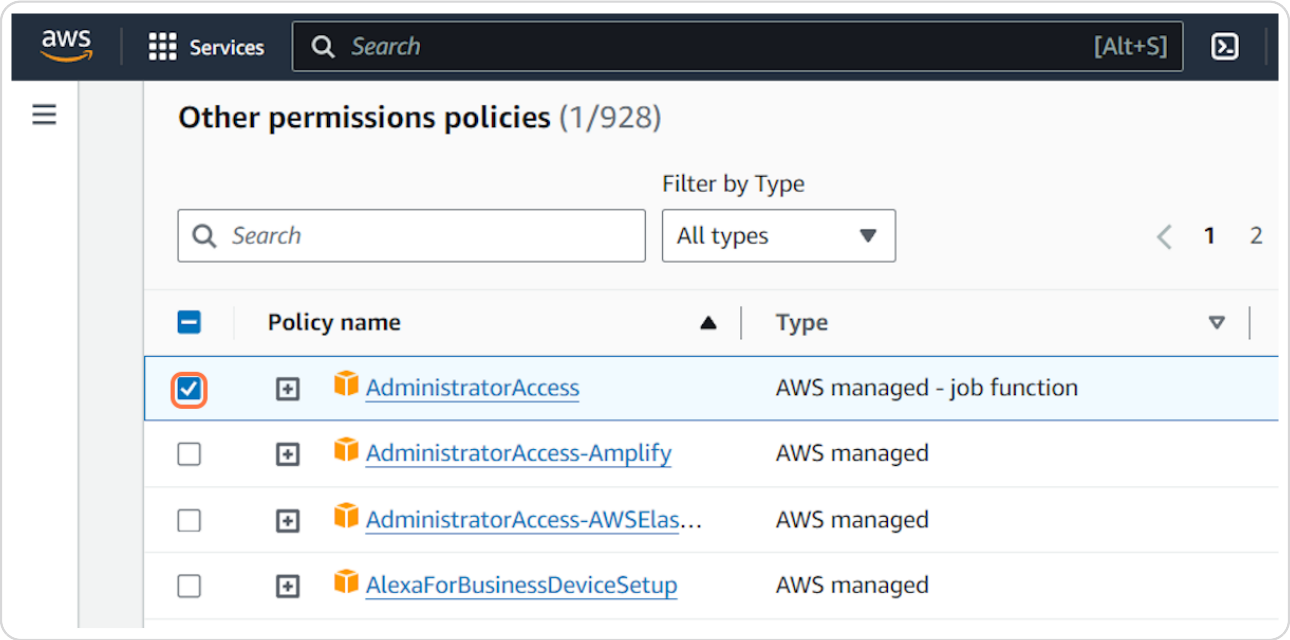
STEP 46

Click on Attach policies



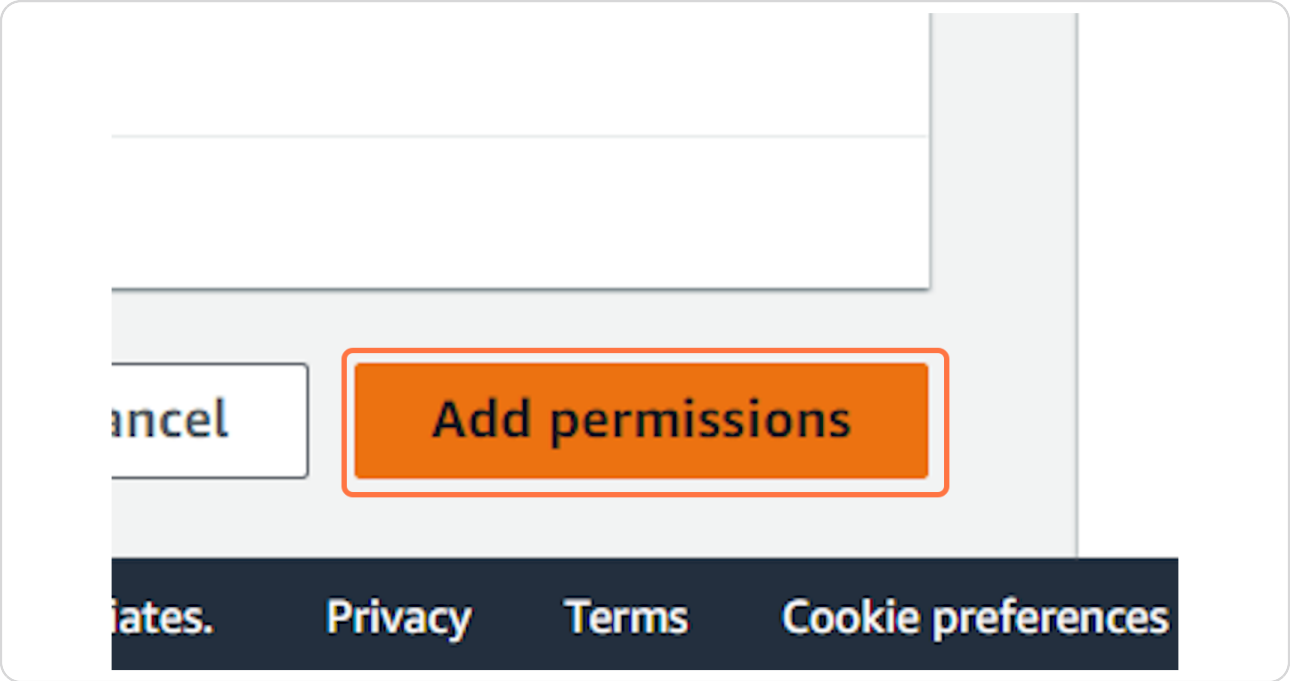
STEP 47

Check on AdministratorAcess



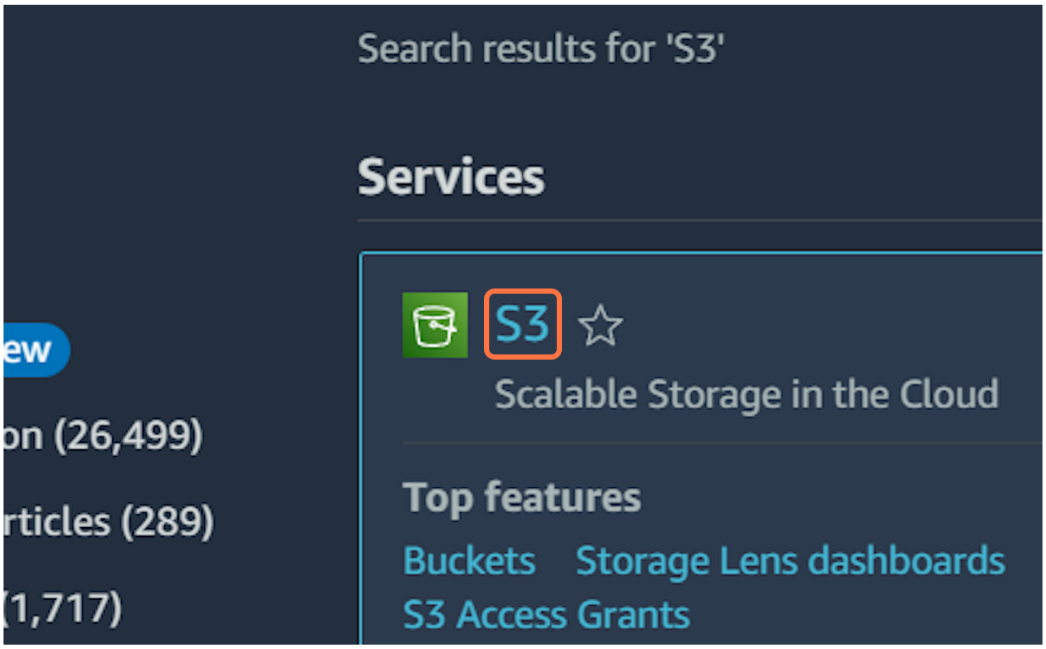
STEP 48

Click on Add permissions



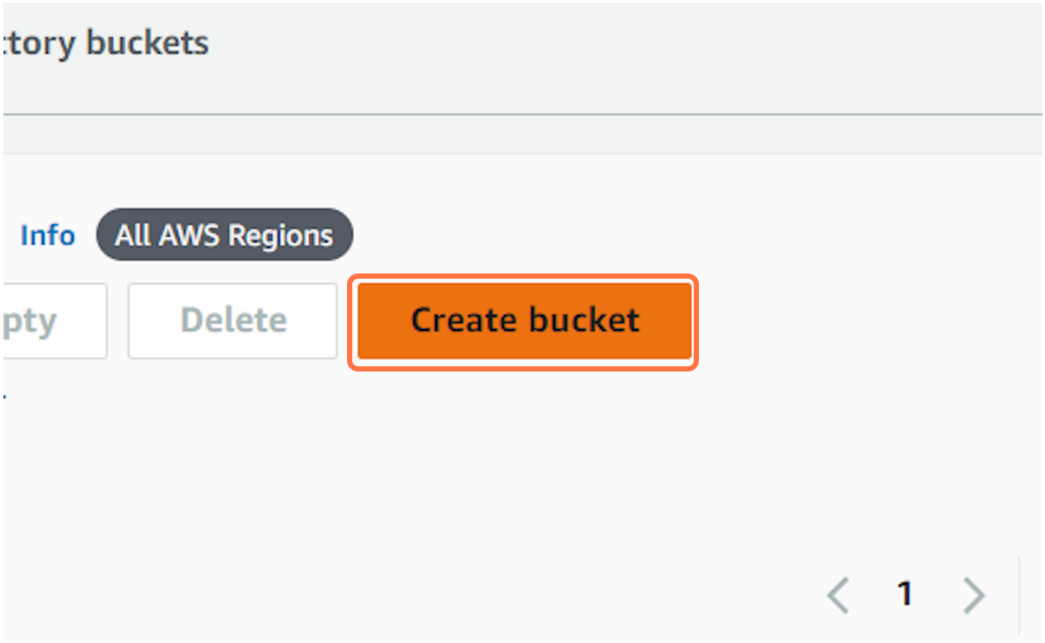
STEP 49

Go on to S3 to create a bucket



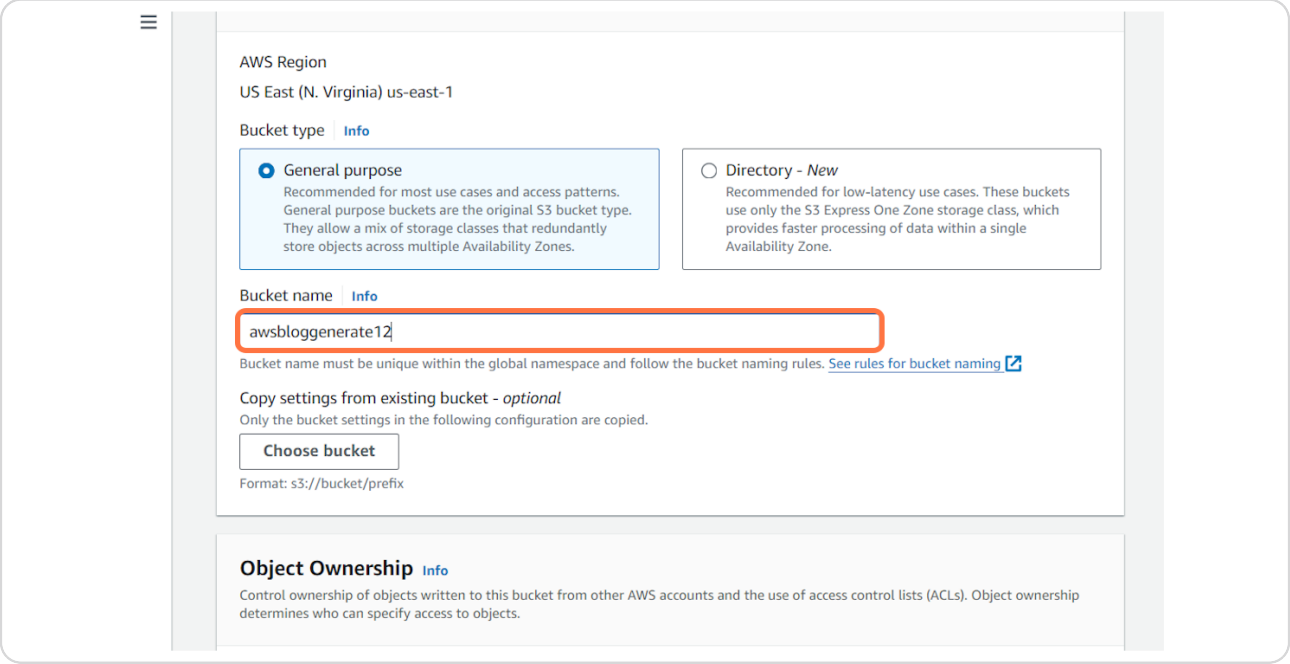
STEP 50

Click on Create bucket



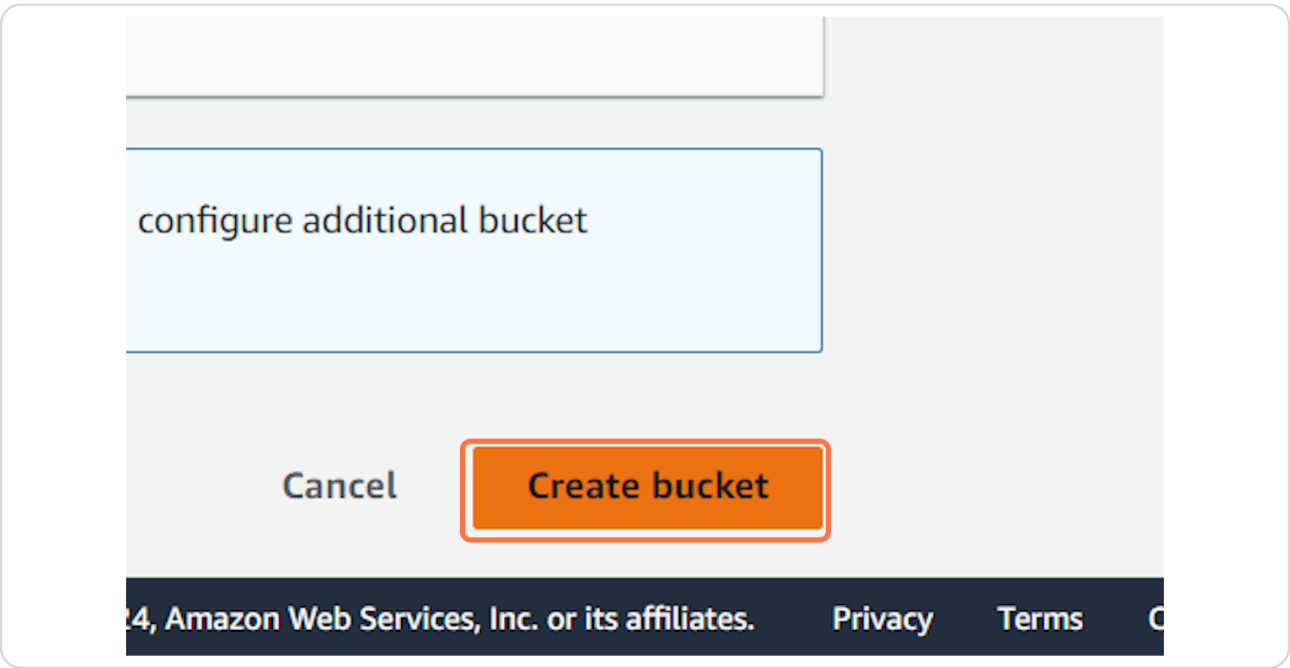
STEP 51

Type "awsbloggenerate12"



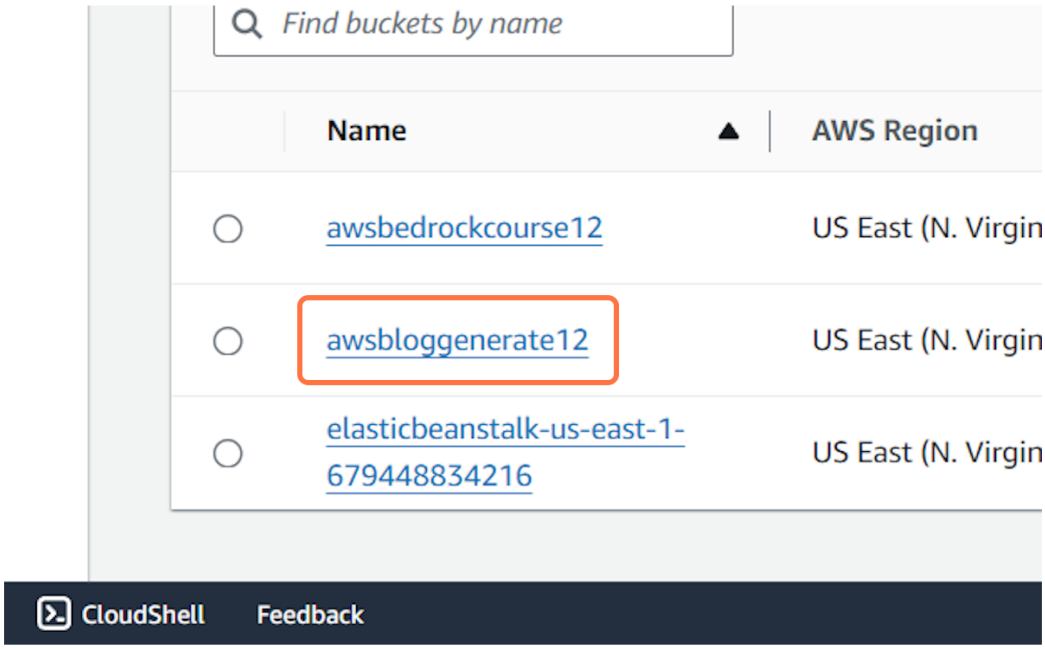
STEP 52

Click on Create bucket



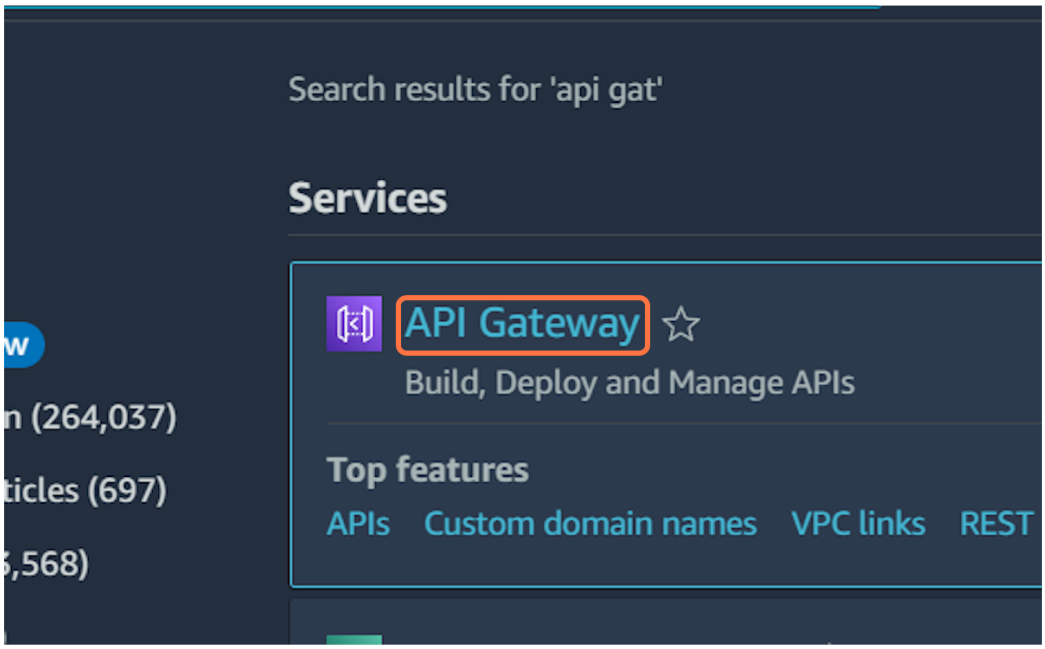
STEP 53

Verify whether bucket creation is successful



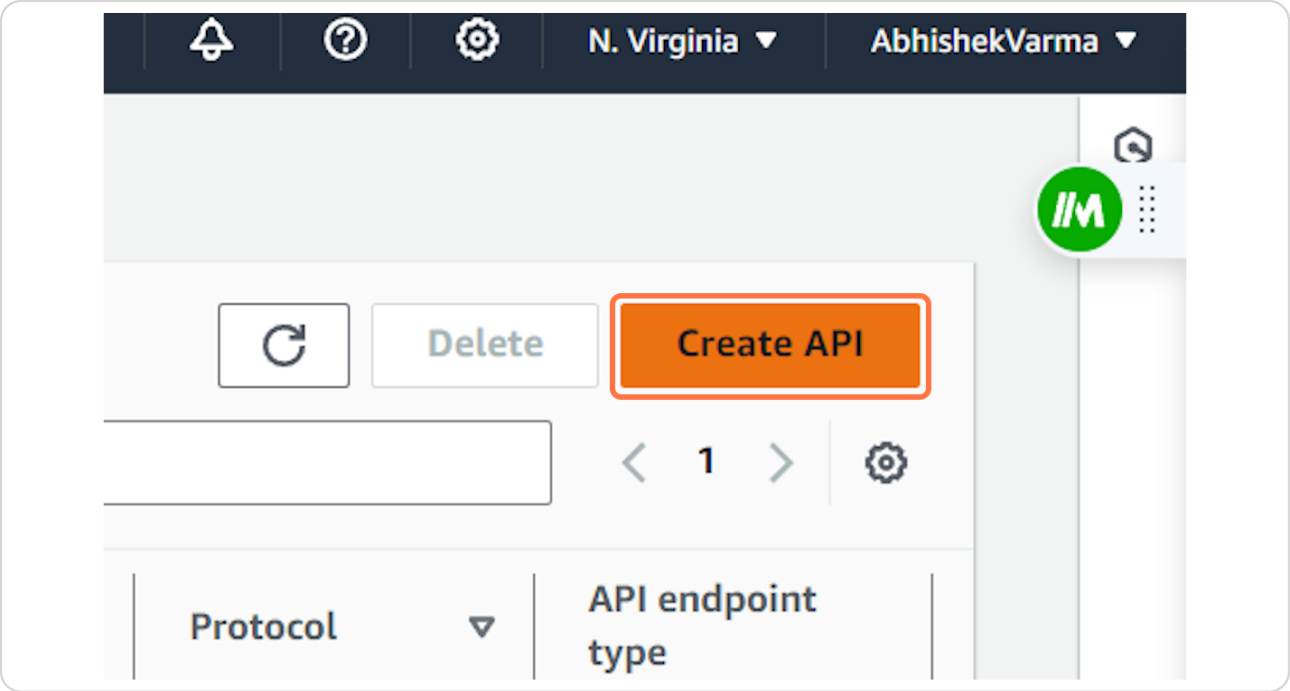
STEP 54

Go on to API Gateway



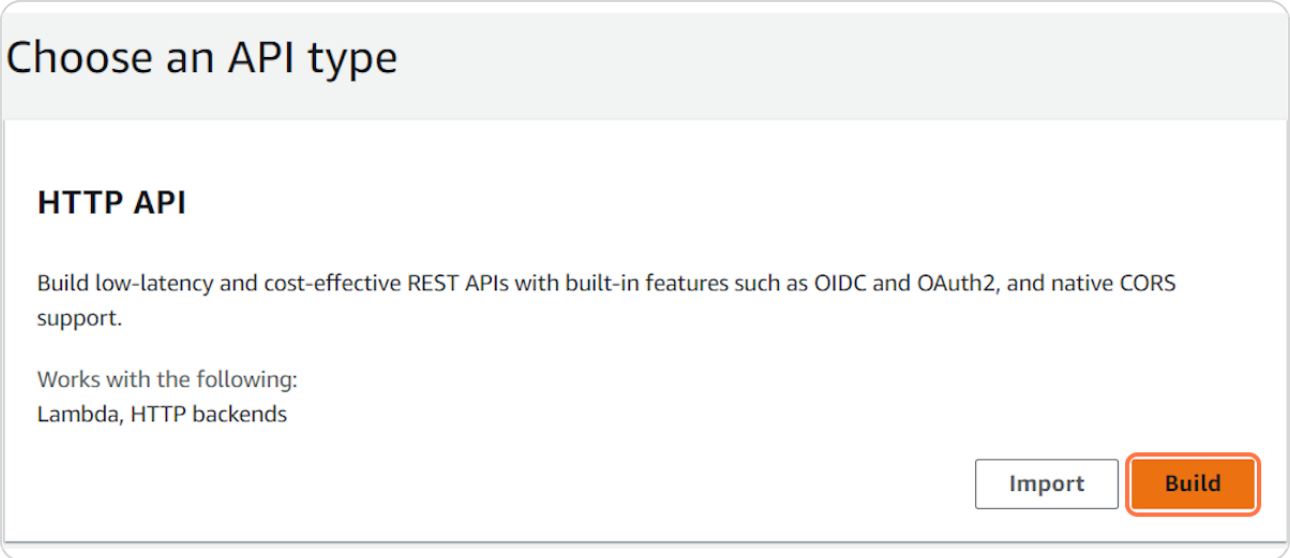
STEP 55

Click on Create API



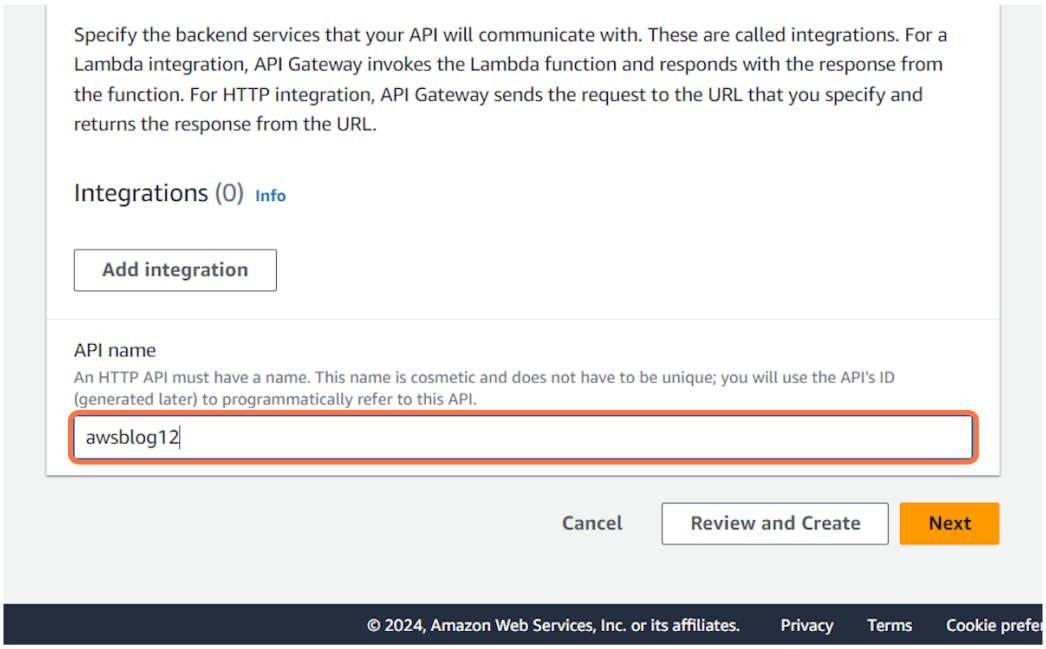
STEP 56

Click on Build HTTP API



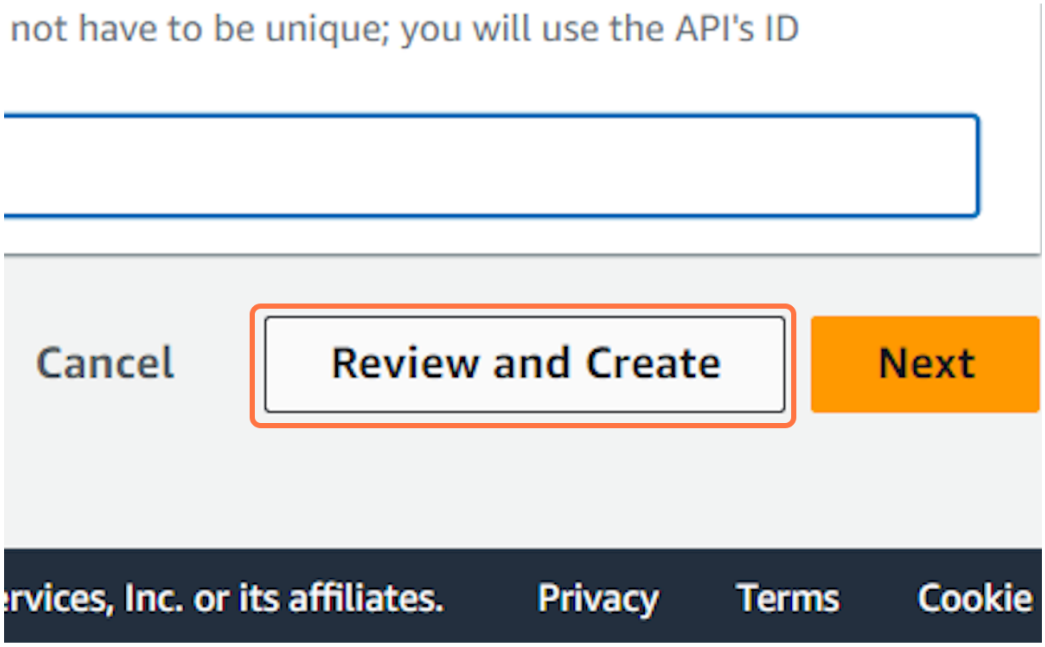
STEP 57

Type "awsblog12"



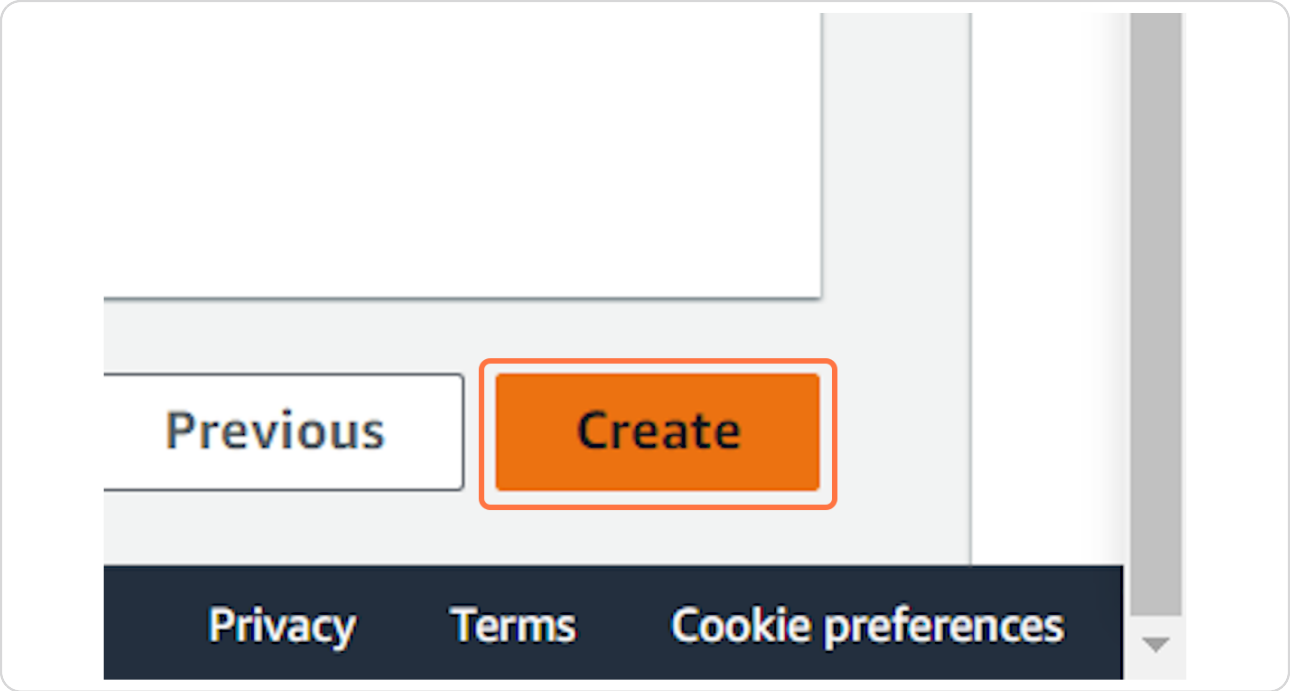
STEP 58

Click on Review and Create



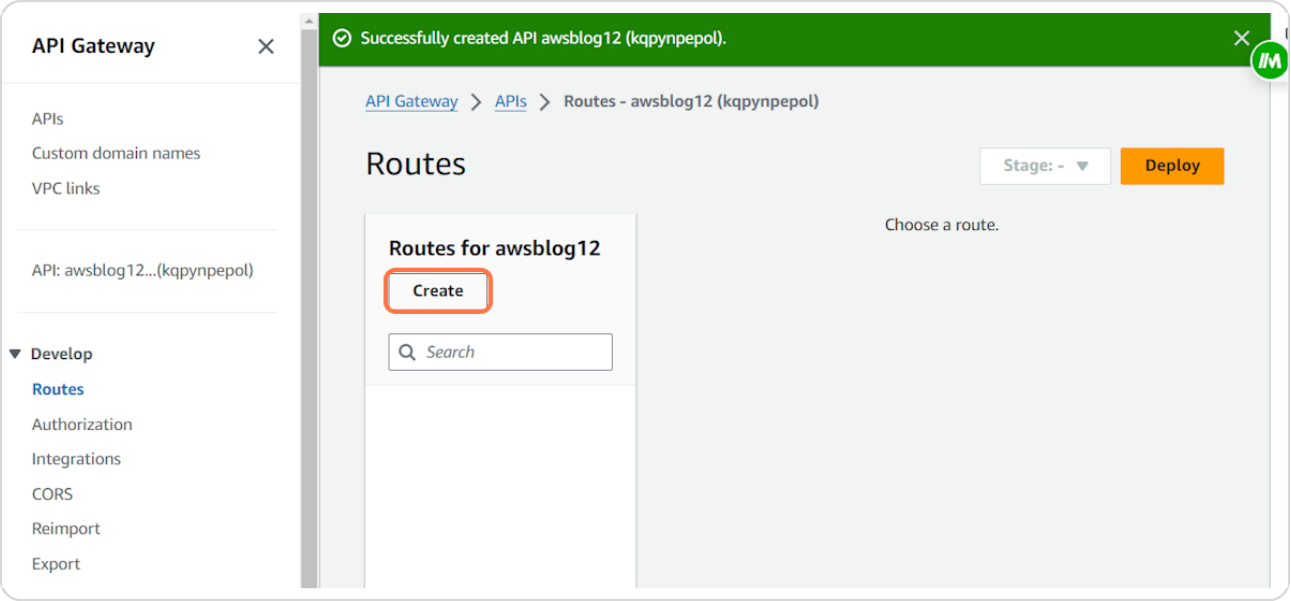
STEP 59

Click on Create



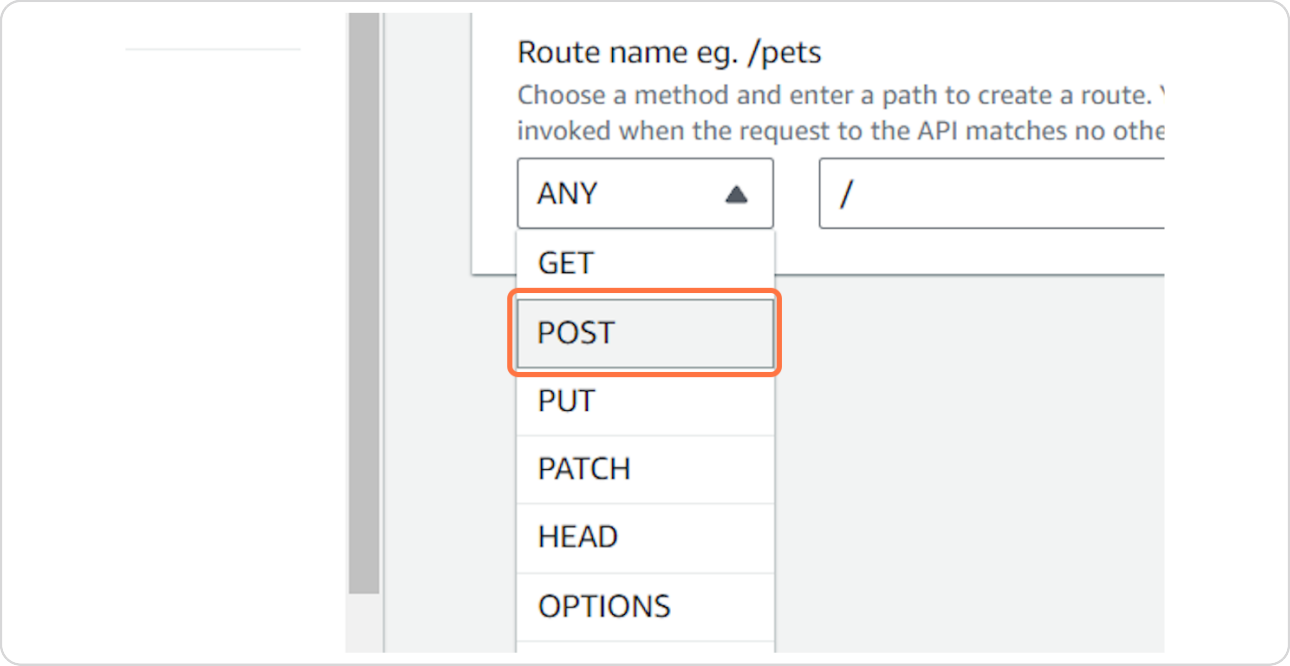
STEP 60

Click on Create



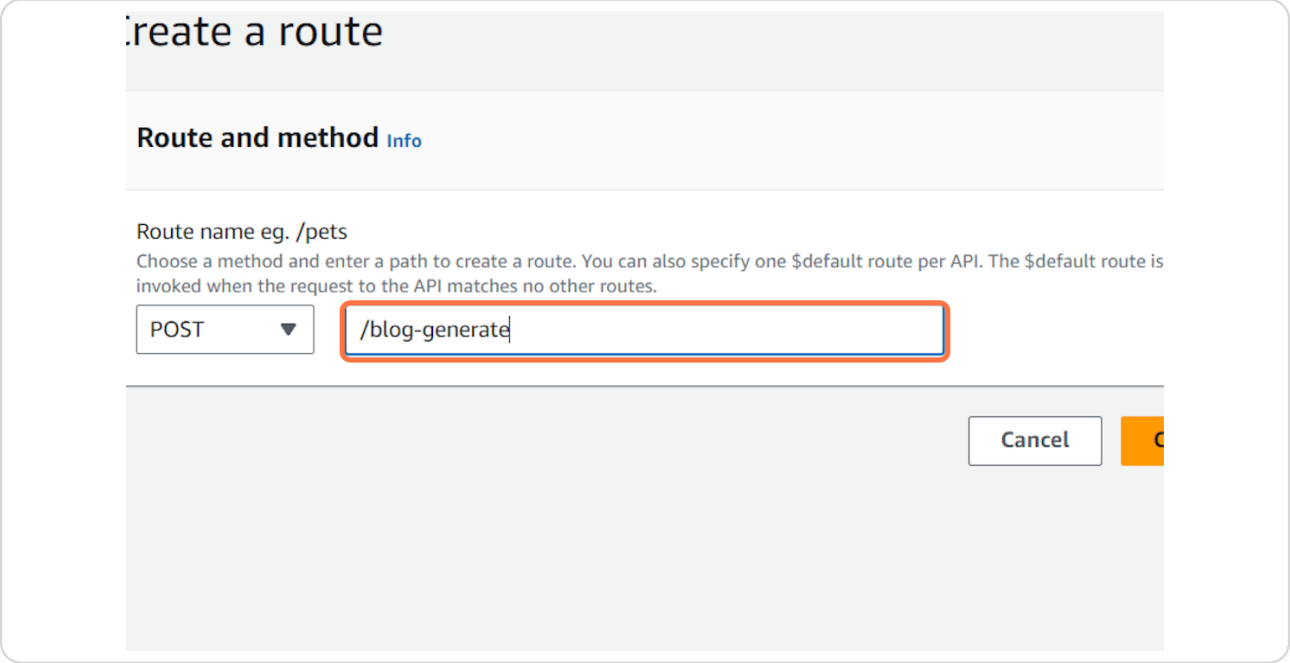
STEP 61

Click on POST...



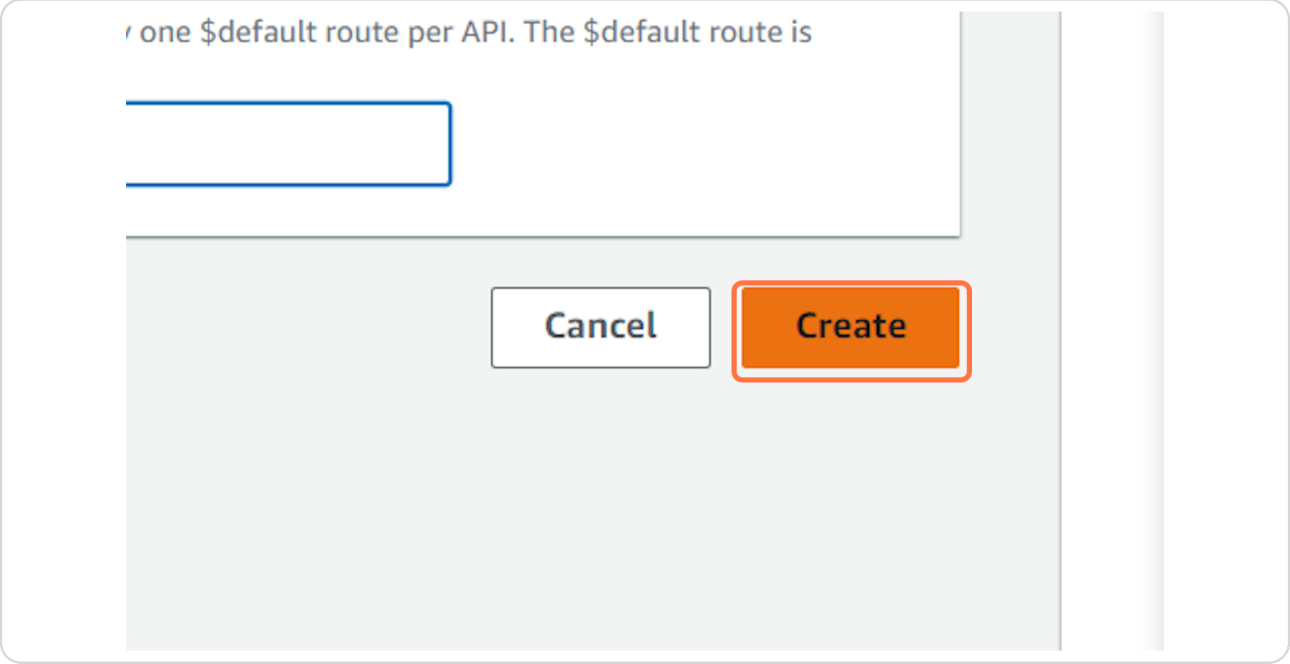
STEP 62

Type "/blog-generate"



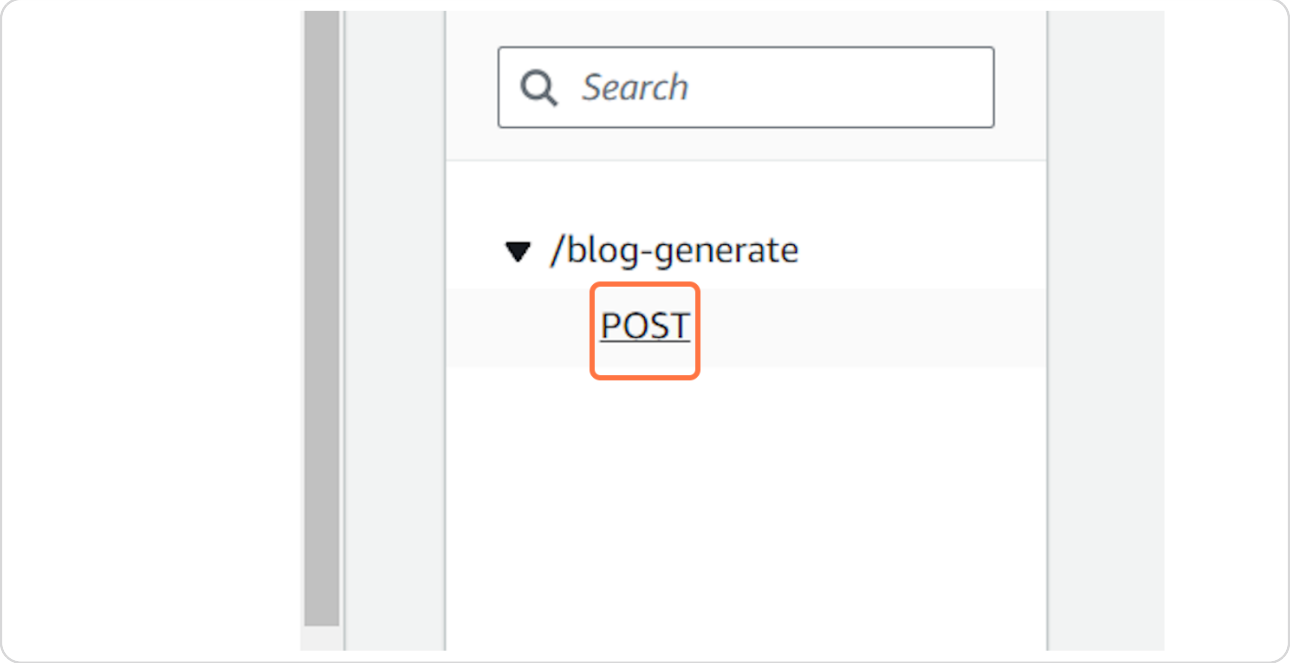
STEP 63

Click on Create



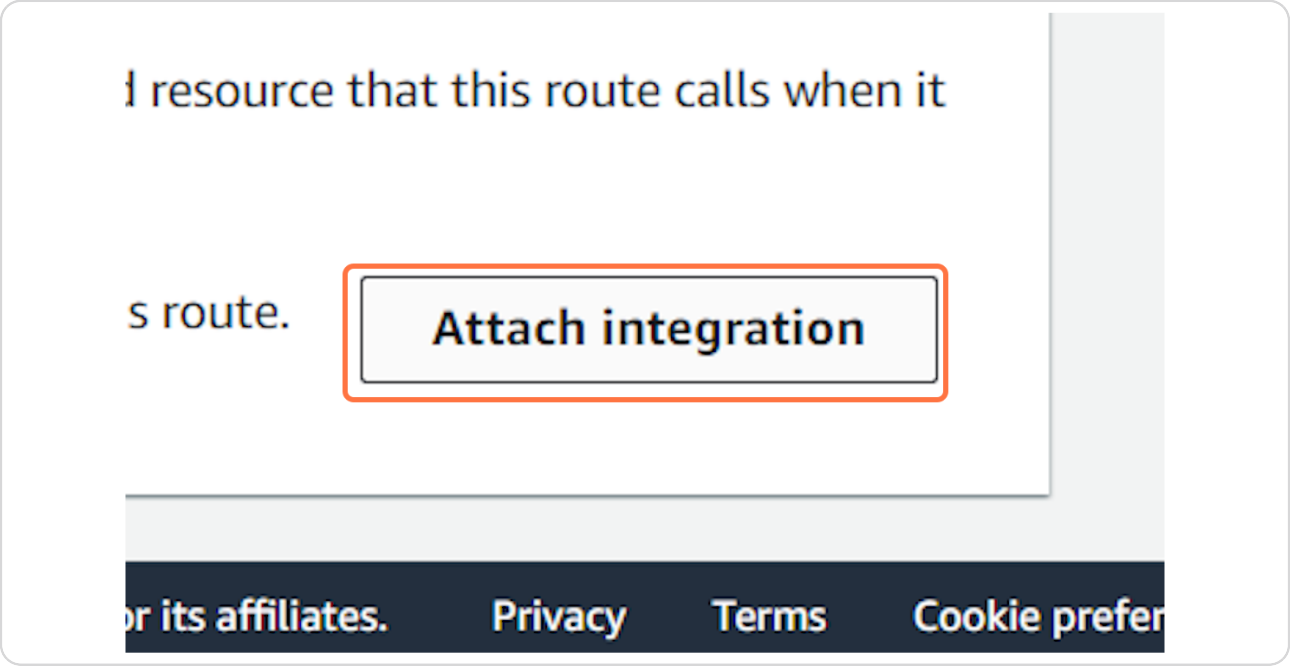
STEP 64

Click on POST



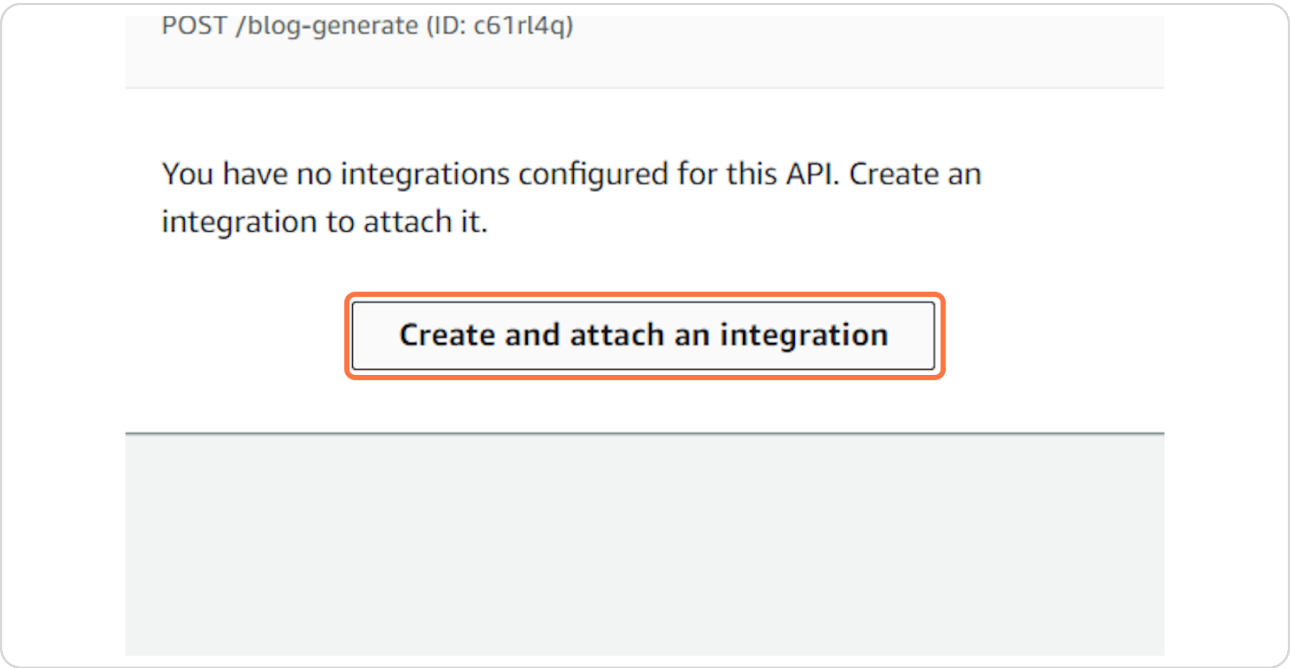
STEP 65

Click on Attach integration



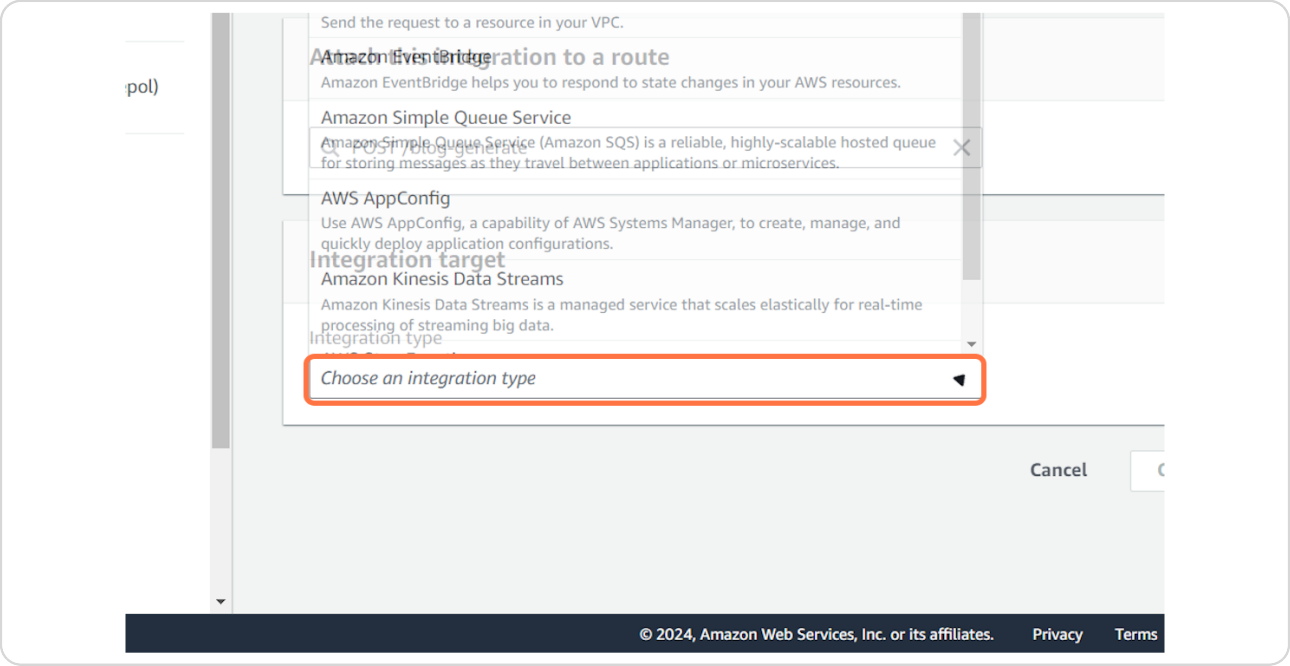
STEP 66

Click on Create and attach an integration



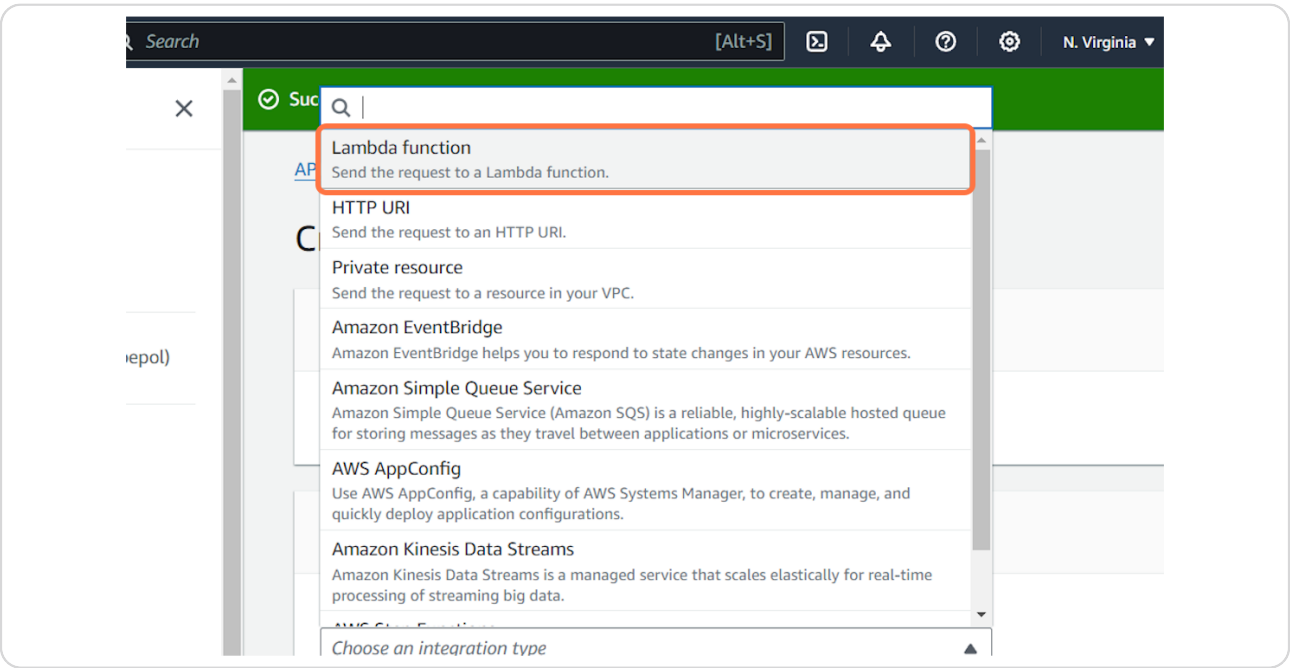
STEP 67

Click on Choose an integration type



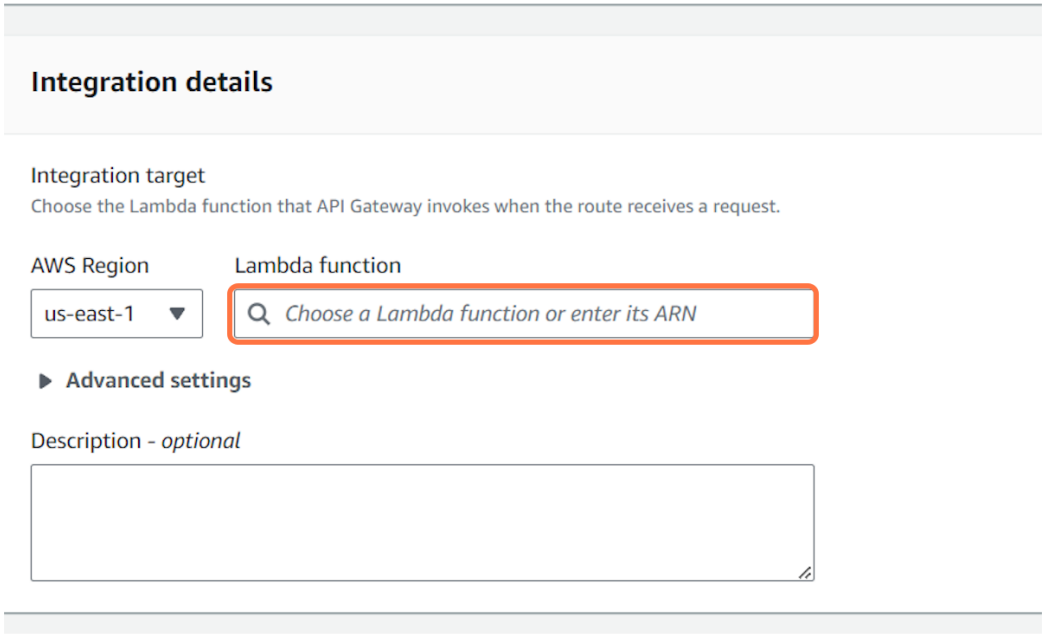
STEP 68

Click on Lambda function...



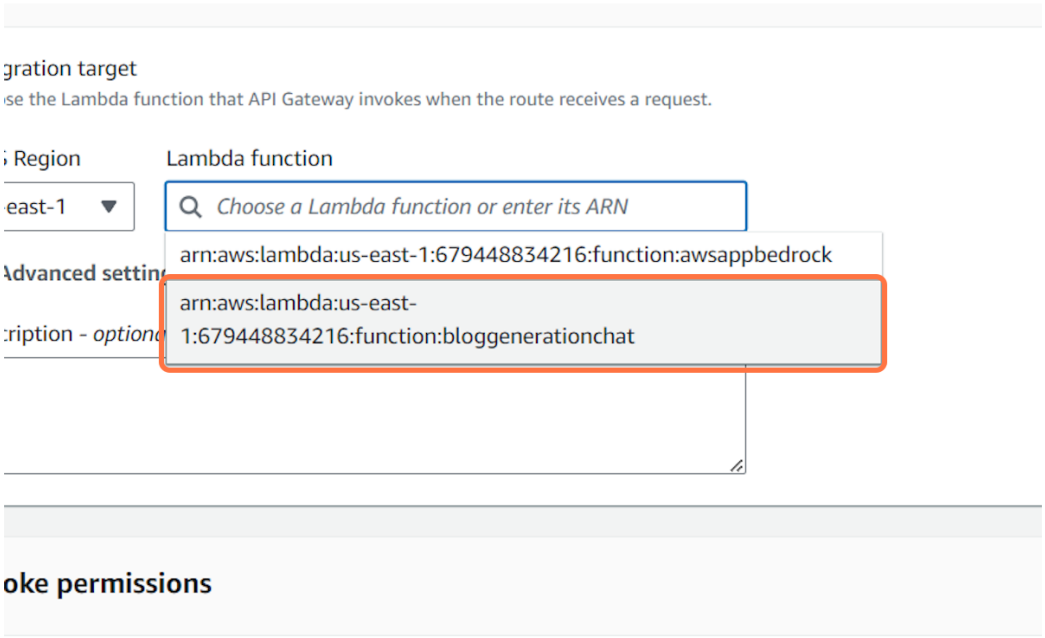
STEP 69

Click on Lambda function



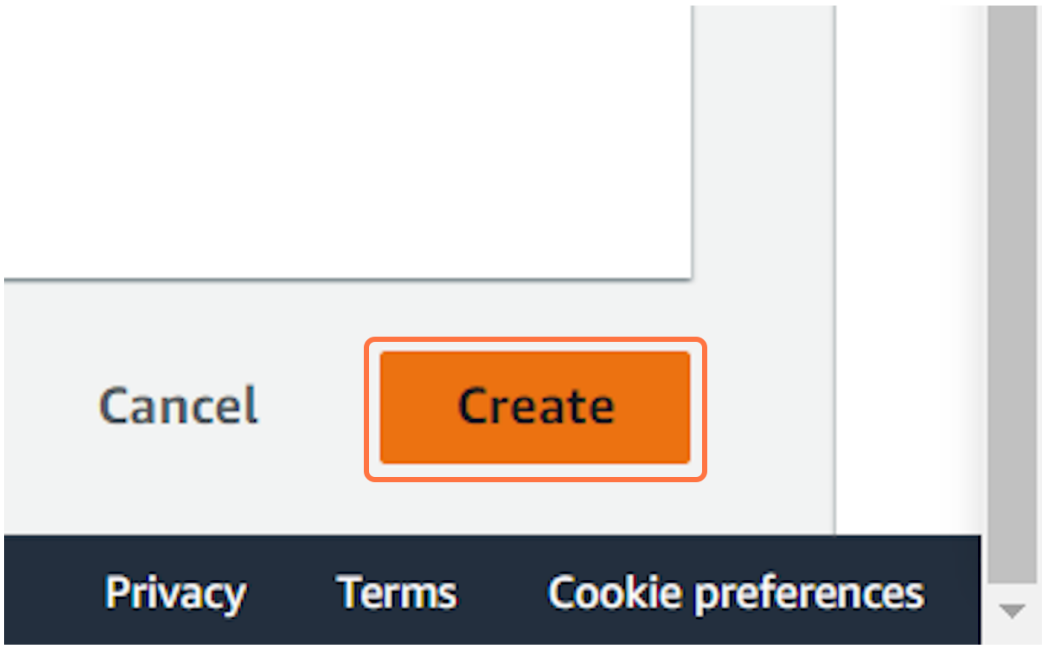
STEP 70

Click on bloggenerationchat lambda fu



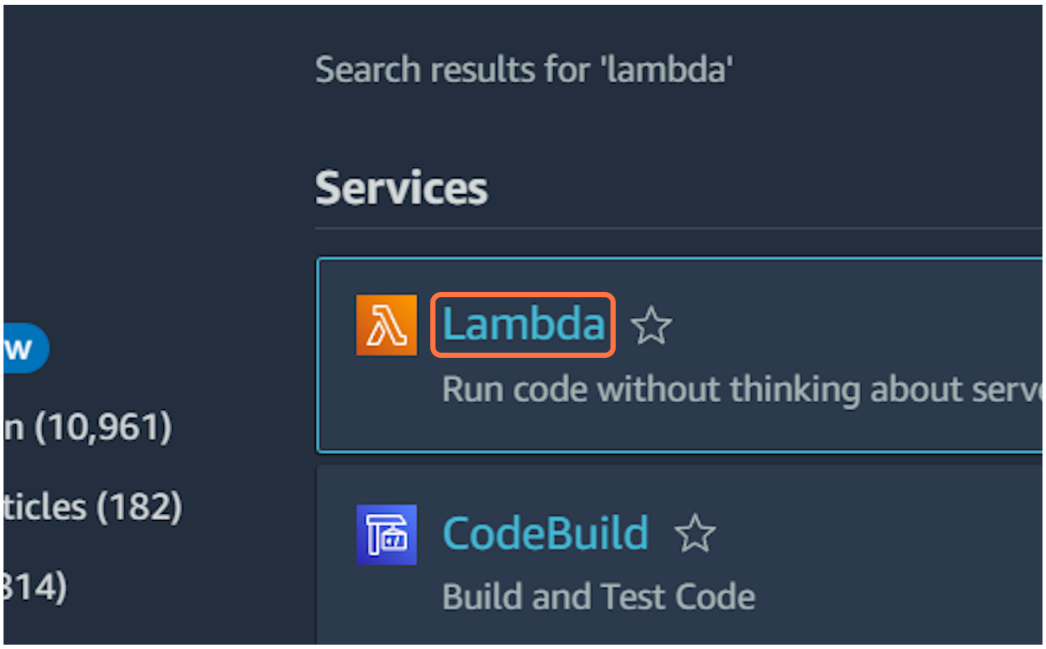
STEP 71

Click on Create



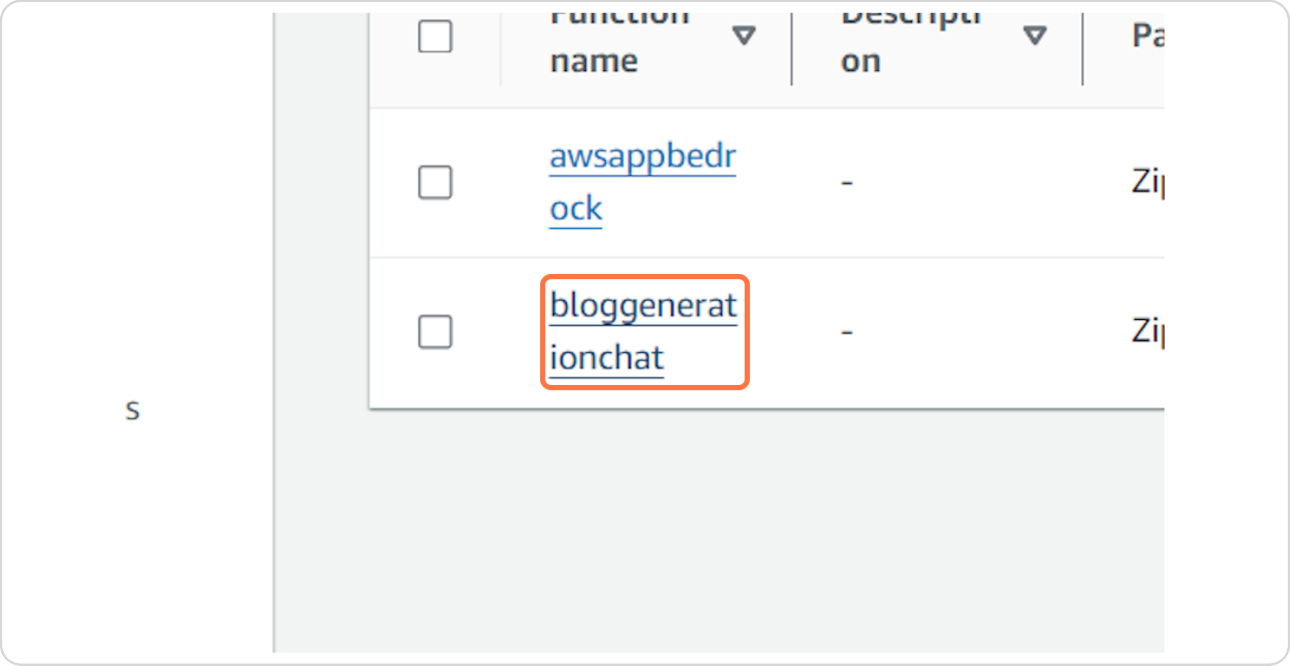
STEP 72

Go on to Lambda to refactor s3 bucket name



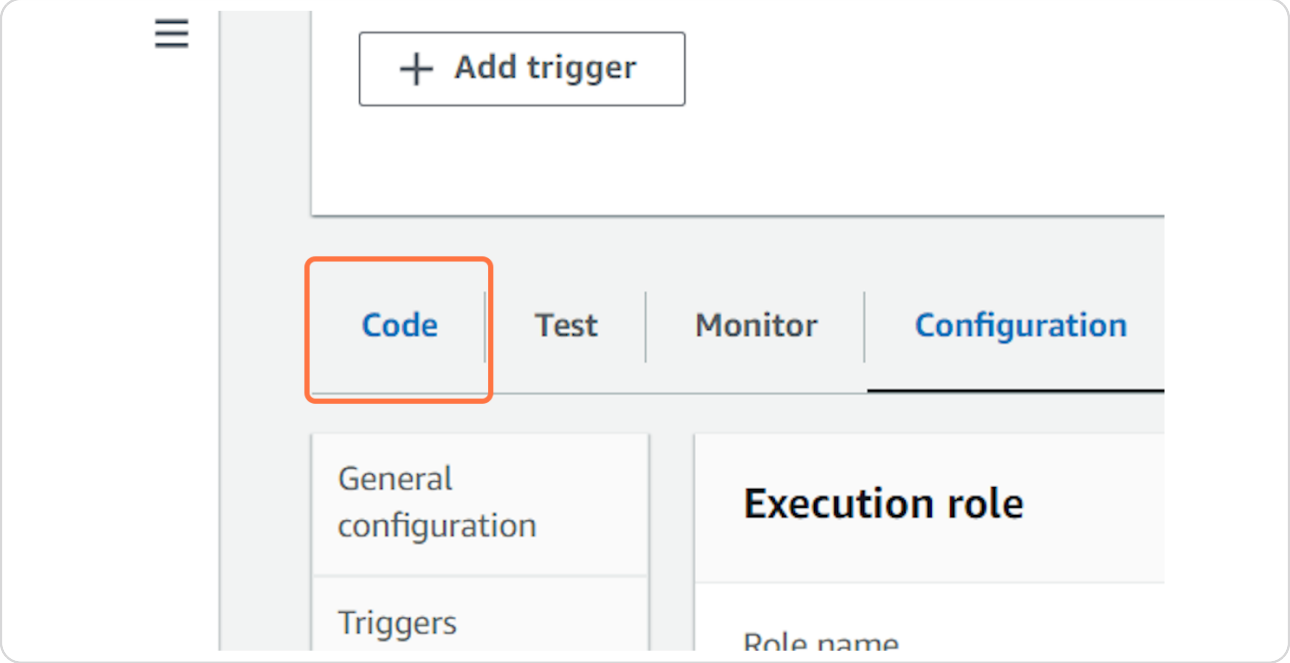
STEP 73

Click on bloggenerationchat



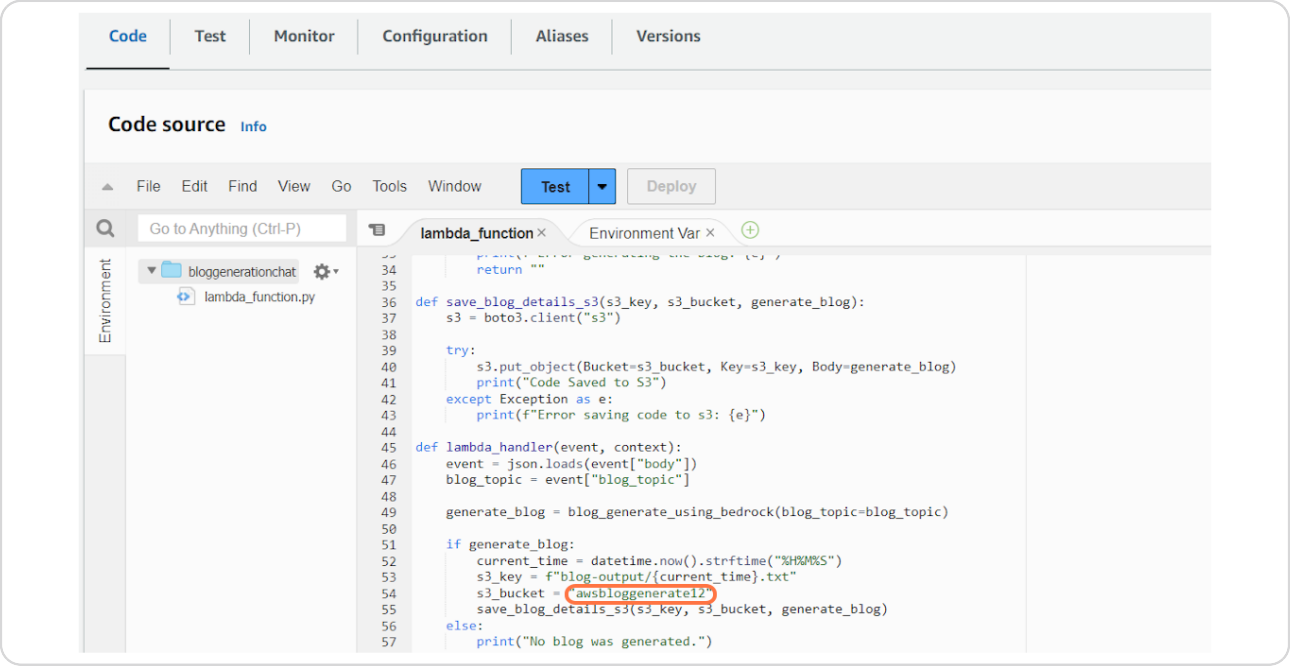
STEP 74

Click on Code



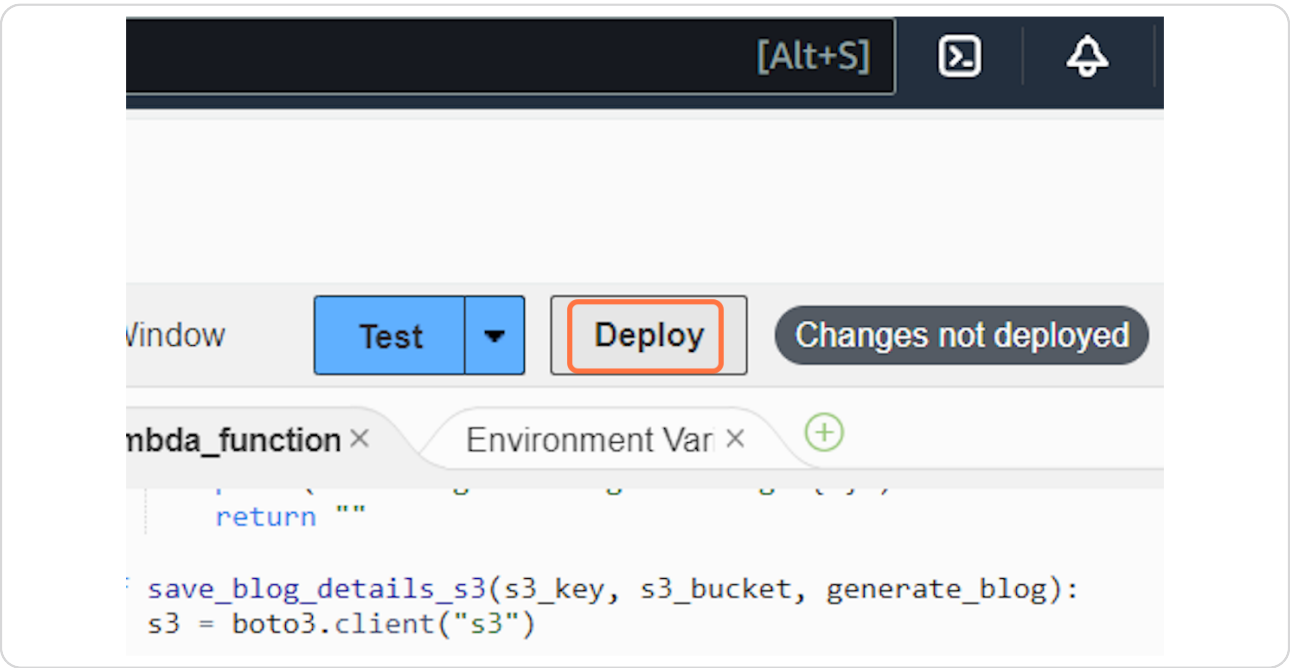
STEP 75

replace s3_bucket to "awsbloggenerate12"



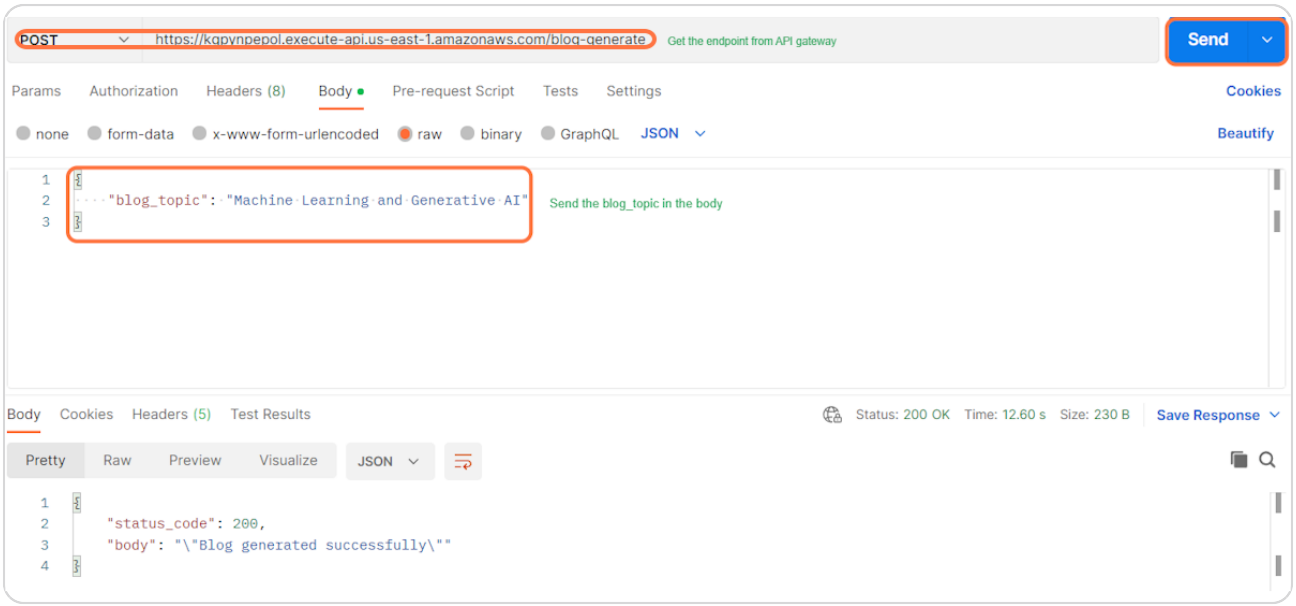
STEP 76

Click on Deploy



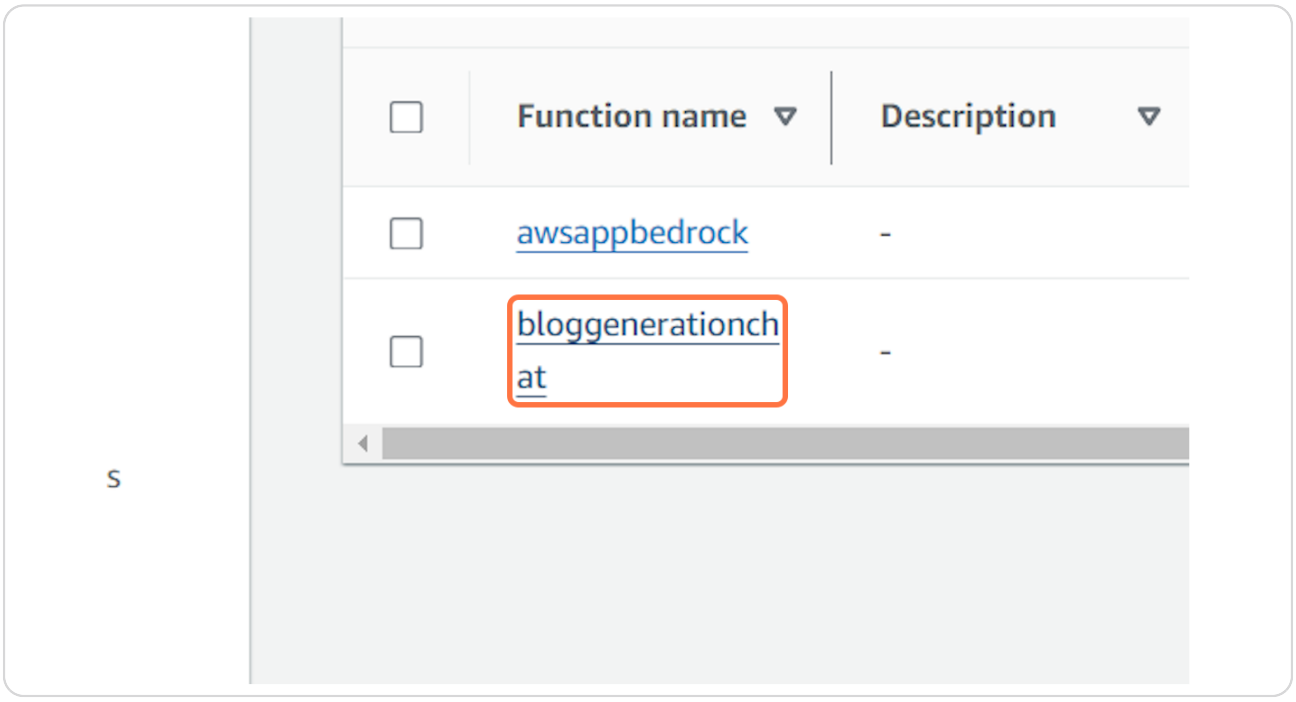
STEP 77

Head over to Postman to hit the API gateway with the route "blog-generate"



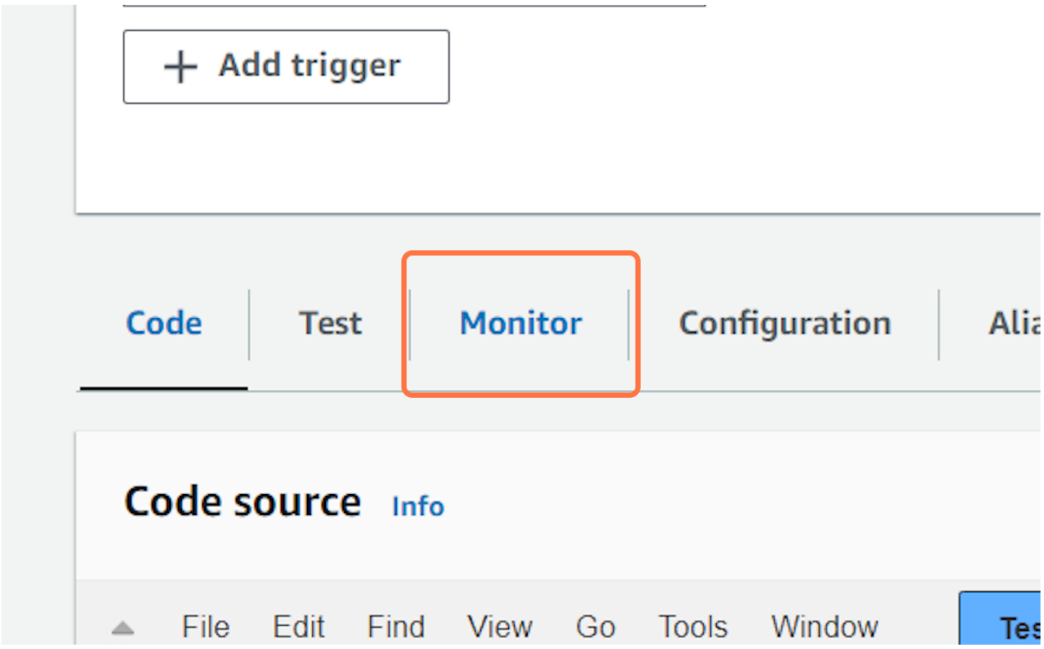
STEP 78

Click on bloggenerationchat



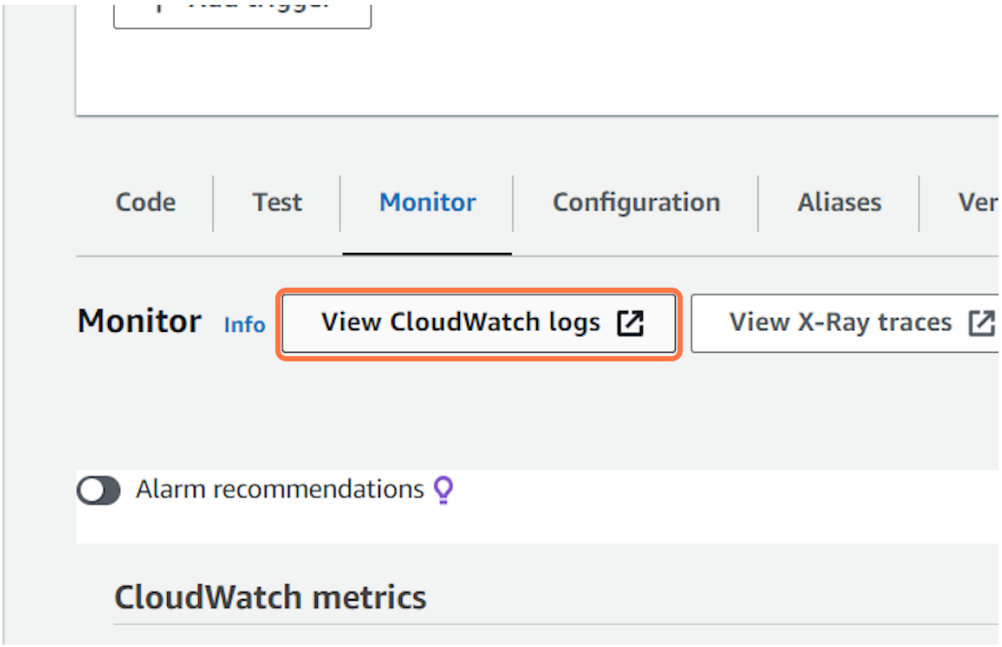
STEP 79

Click on Monitor



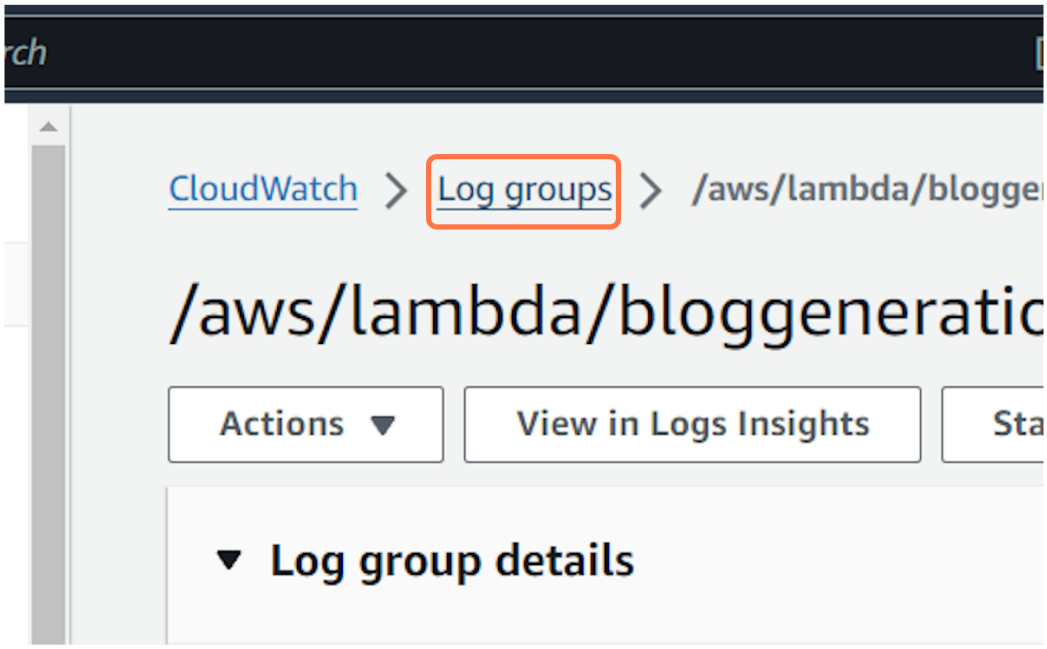
STEP 80

Click on View CloudWatch logs



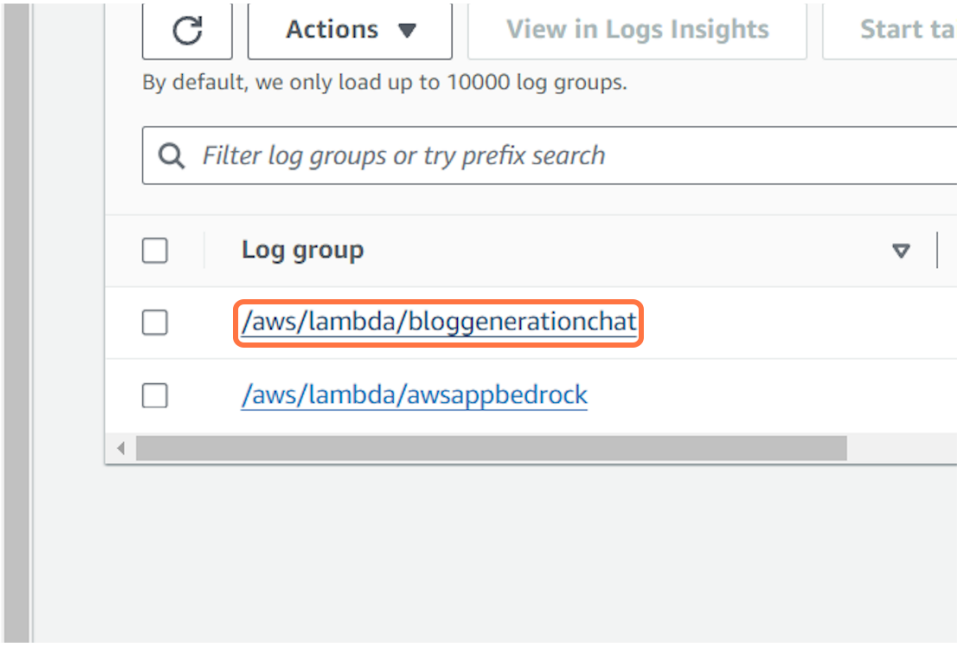
STEP 81

Click on Log groups



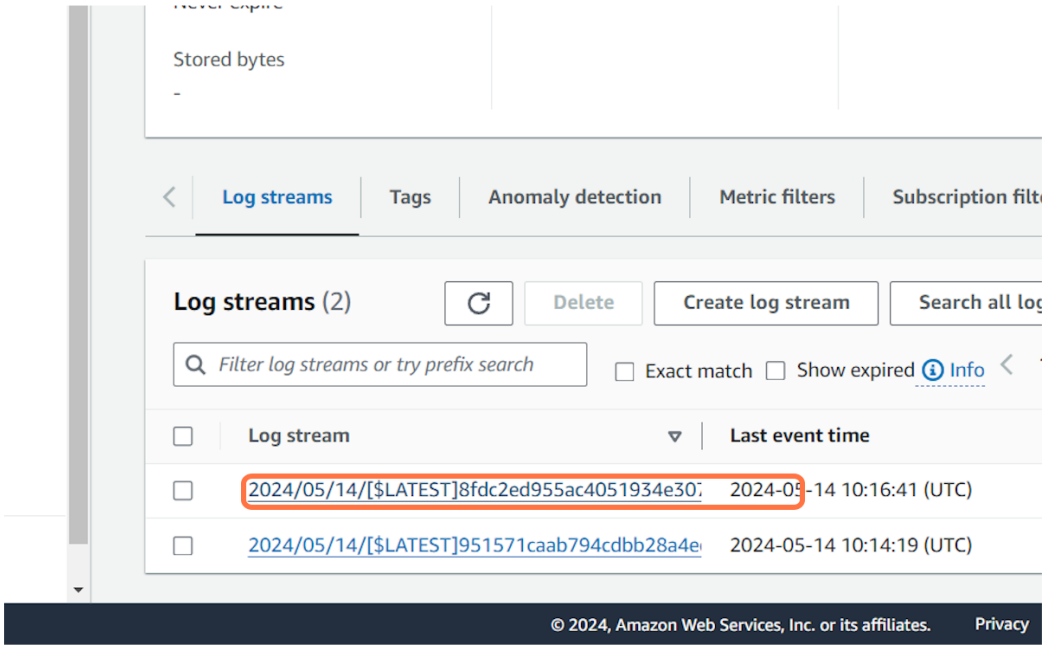
STEP 82

Click on /aws/lambda/bloggenerationchat



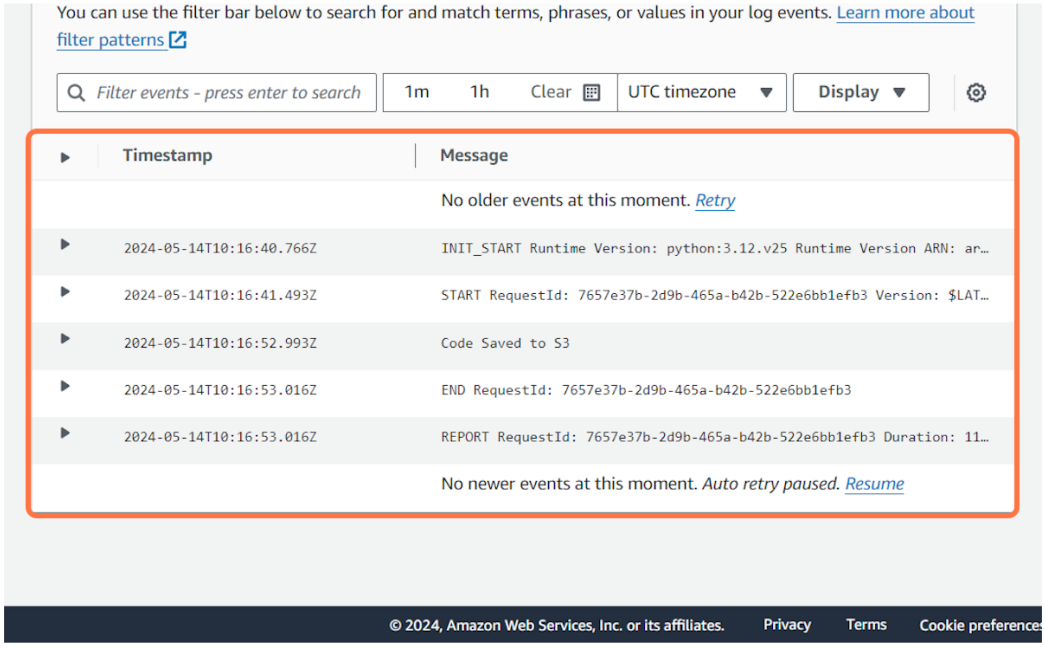
STEP 83

Click on recent call log generated



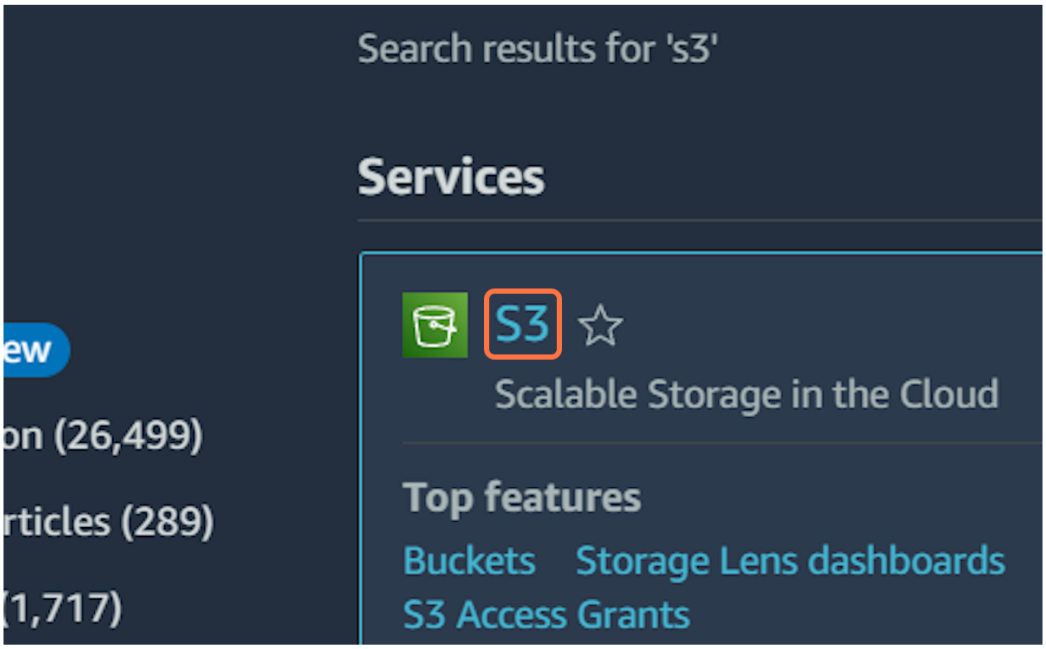
STEP 84

See if any errors occurred or code saved to S3



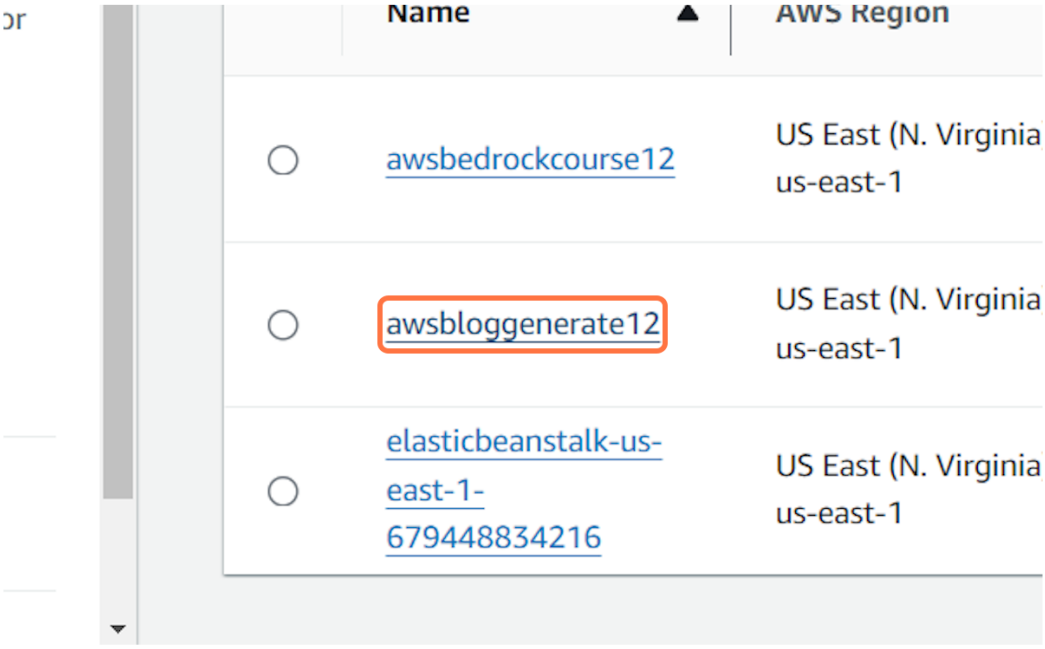
STEP 85

Head over to S3



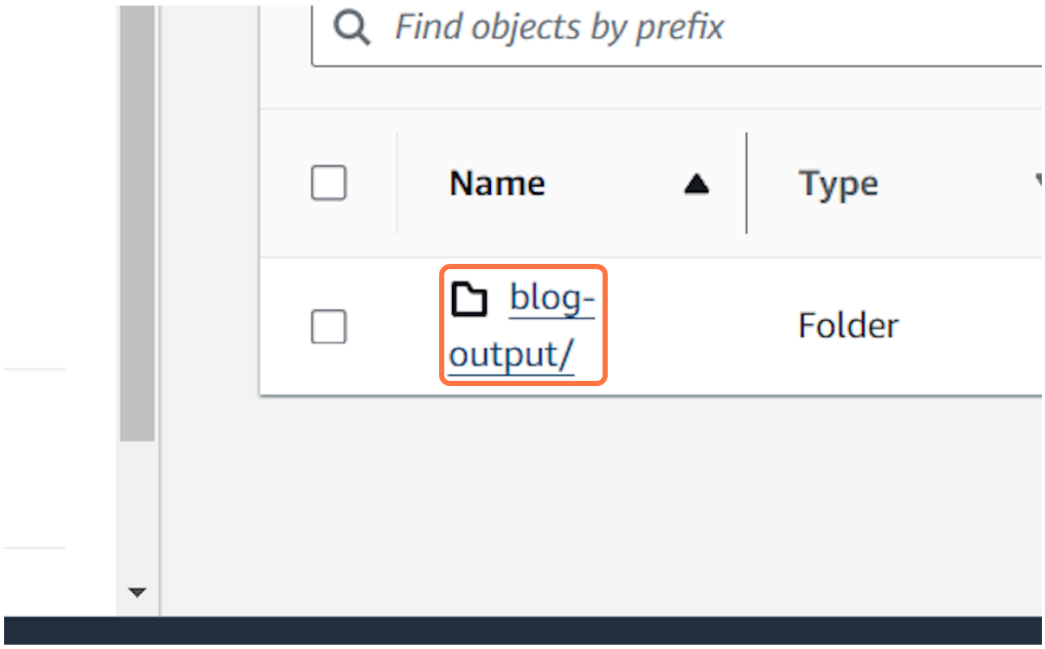
STEP 86

Click on awsbloggenerate12



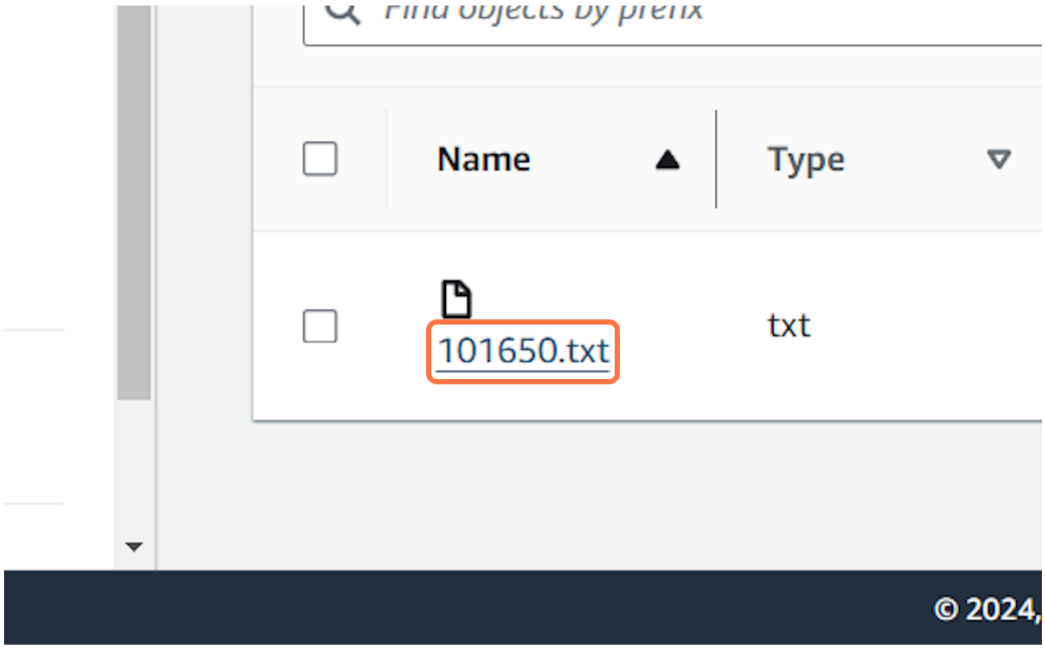
STEP 87

Click on blog-output/



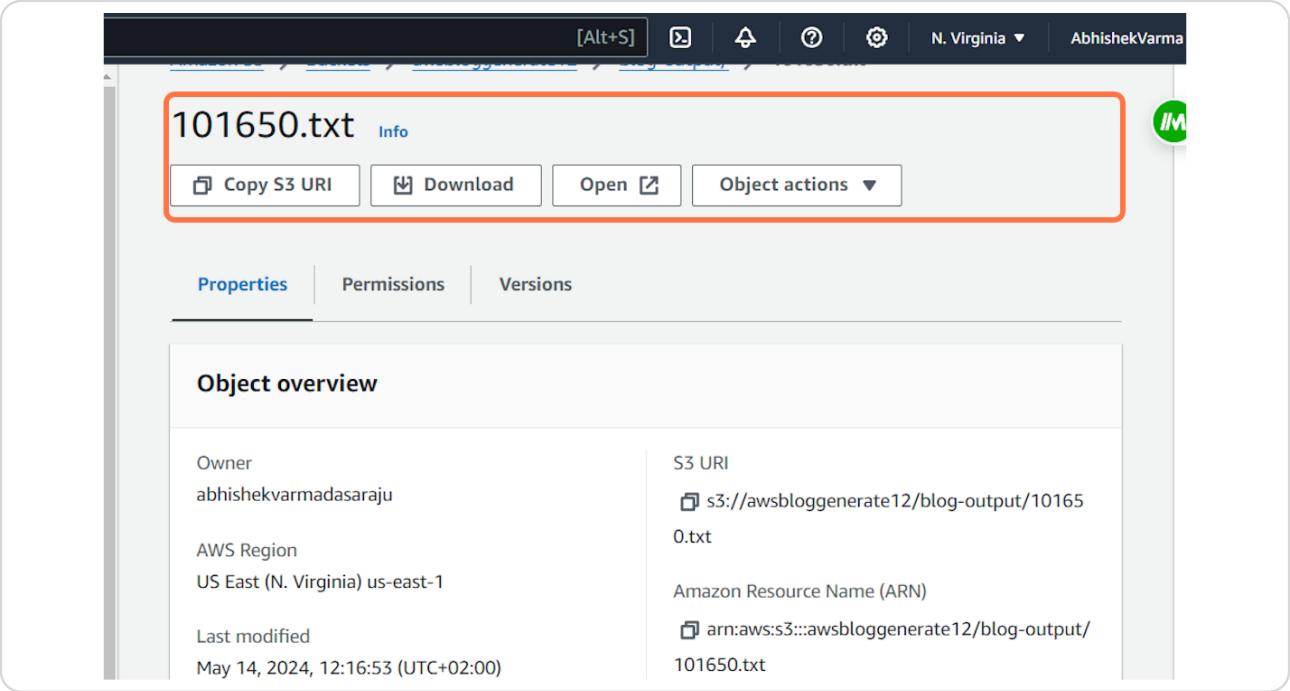
STEP 88

Click on 101650.txt



STEP 89

Download and see the generated blog output.



STEP 90

That's it! You learned successfully how to utilize Bedrock foundational models. Now build your ideas with the knowledge gained through this workflow.

Tango

Never miss a step again. Visit Tango.us