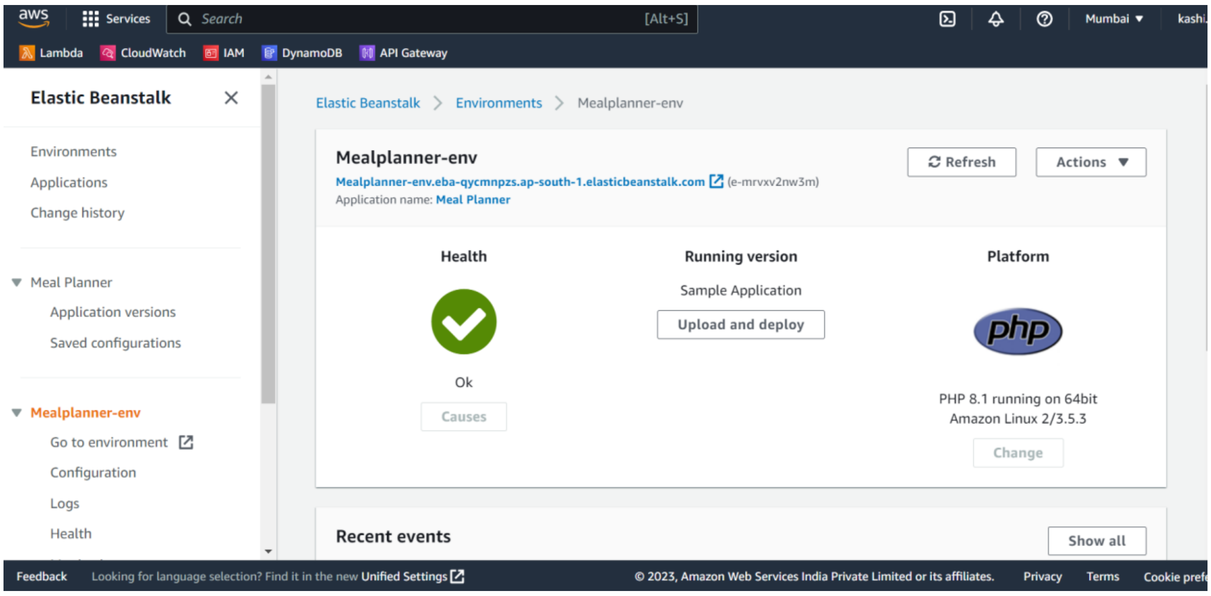
Module 9 case study

Problem Statement:

You work for xyz organization. Your organization uses AWS cloud to host their web applications.

You have been asked to:

1. Create a Beanstalk environment to host a Custom PHP application to host our application on it.
2. Upload a new PHP file which reads “Updated Hello World” to the Beanstalk environment
3. Create a Lambda Function which uses S3 bucket Object creation as a trigger.
4. Launch an OpsWork stack and host a Hello World html page in the instance of that stack



aws Services Search [Alt+S] Mumbai kashi
Lambda CloudWatch IAM DynamoDB API Gateway
Elastic Beanstalk
Environments
Applications
Change history
Meal Planner
Application versions
Saved configurations
Mealplanner-env
Go to environment
Configuration
Logs
Health
Elastic Beanstalk > Environments > Mealplanner-env
Refresh
Actions
Health
Running version
Platform
Upload and deploy
Causes
Change
Recent events
Show all
Feedback Looking for language selection? Find it in the new Unified Settings
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie pref

Mealplanner-env

Mealplanner-env.eba-qycmnpzs.ap-south-1.elasticbeanstalk.com (e-mrvxv2nw3m)

Application name: Meal Planner

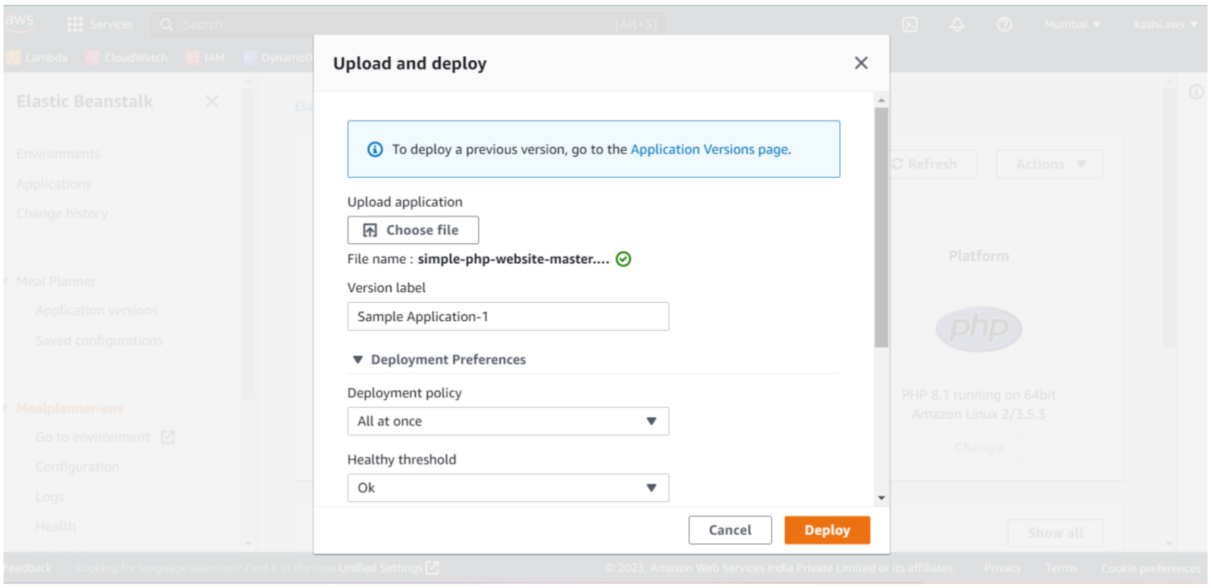
Health: Ok

Running version: Sample Application

Platform: PHP 8.1 running on 64bit Amazon Linux 2/3.5.3

1.

➔ Created EB Environment with PHP as runtime



To deploy a previous version, go to the Application Versions page.

Upload application

File name : simple-php-website-master....

Version label

Sample Application-1

Deployment policy

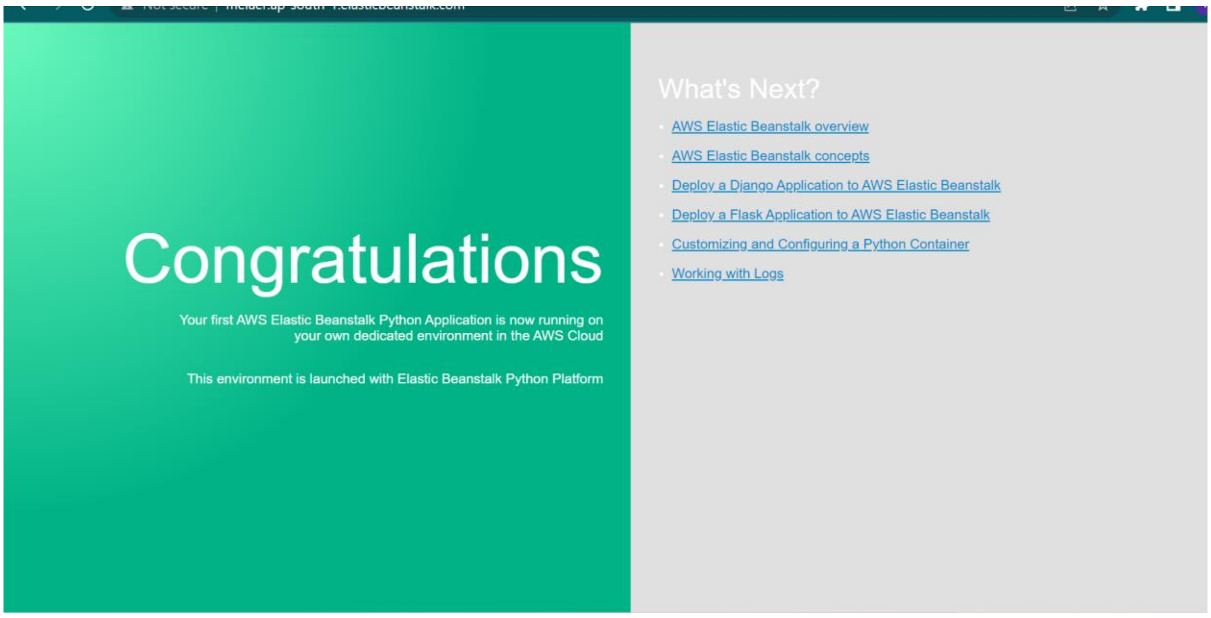
All at once

Healthy threshold

Ok

➔

➔ PHP Application file added

2. 

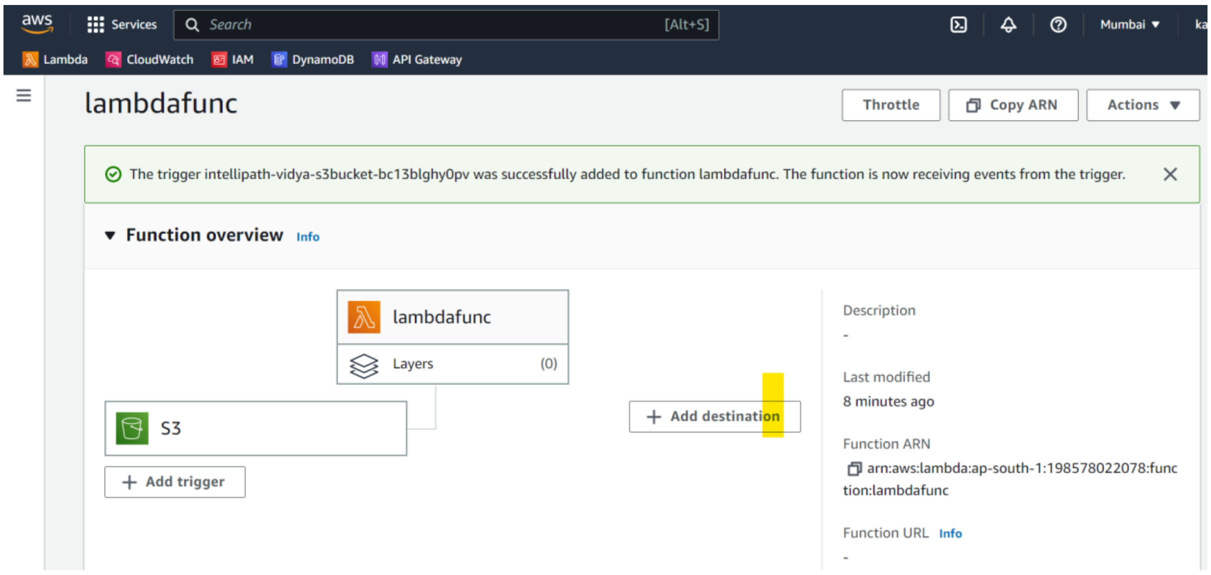
Congratulations

Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Python Platform

What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy a Django Application to AWS Elastic Beanstalk
- Deploy a Flask Application to AWS Elastic Beanstalk
- Customizing and Configuring a Python Container
- Working with Logs

3. 

lambdafunc

The trigger intellipath-vidya-s3bucket-bc13blghy0pv was successfully added to function lambdafunc. The function is now receiving events from the trigger.

Function overview Info

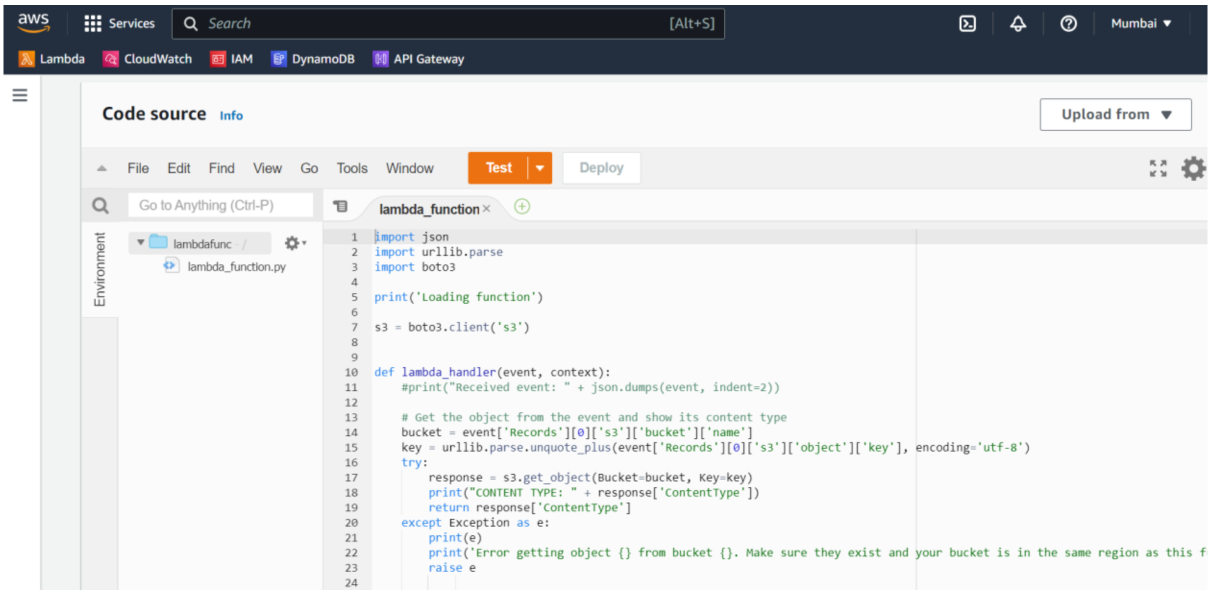
Description
-

Last modified
8 minutes ago

Function ARN
arn:aws:lambda:ap-south-1:198578022078:function:lambdafunc

Function URL Info
-

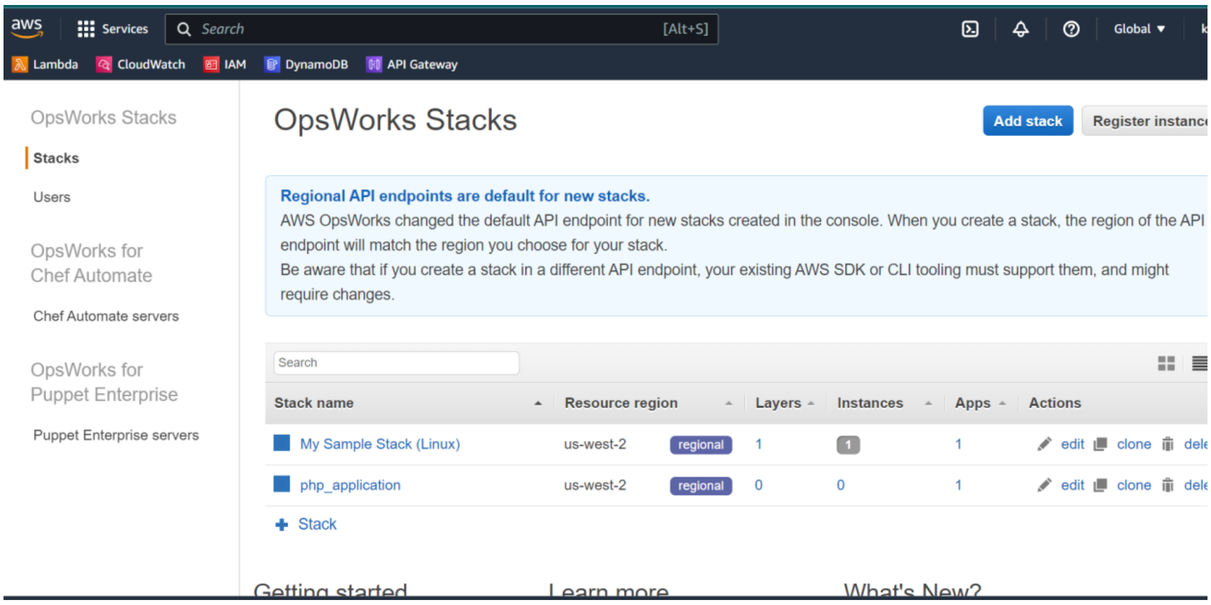
➔ Lambda function created with python for s3 trigger



```python
import json
import urllib.parse
import boto3

print('Loading function')

s3 = boto3.client('s3')


def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event, indent=2))

    # Get the object from the event and show its content type
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
    try:
        response = s3.get_object(Bucket=bucket, Key=key)
        print("CONTENT TYPE: " + response['ContentType'])
        return response['ContentType']
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this f
        raise e
```

-> This is the lambda function

4.

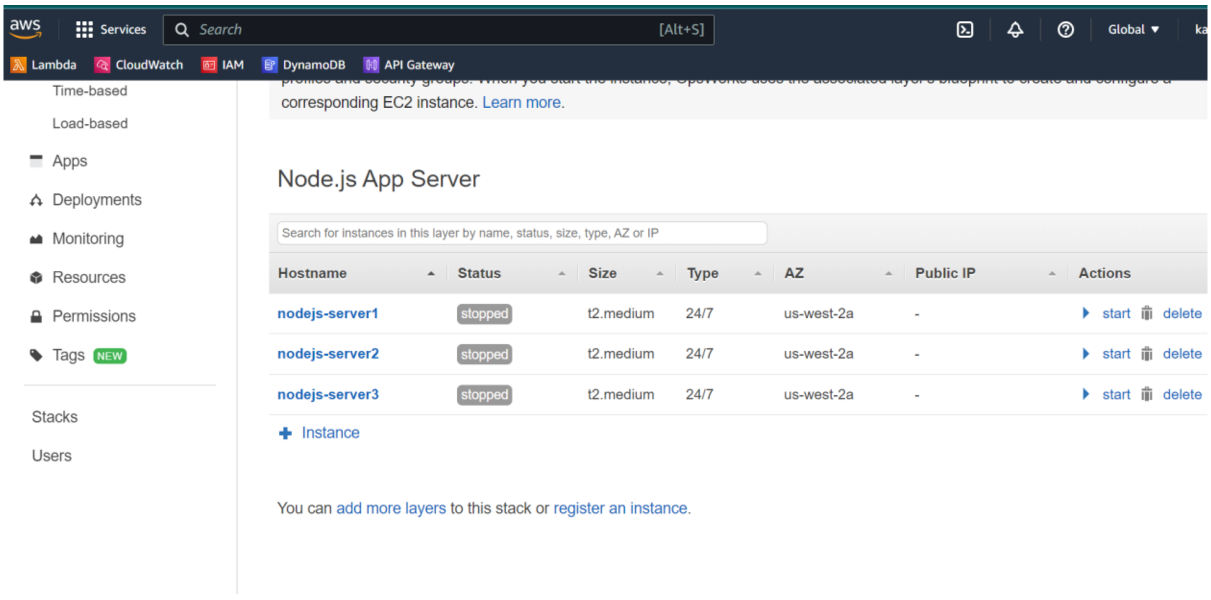


OpsWorks Stacks

Regional API endpoints are default for new stacks.
AWS OpsWorks changed the default API endpoint for new stacks created in the console. When you create a stack, the region of the API endpoint will match the region you choose for your stack.
Be aware that if you create a stack in a different API endpoint, your existing AWS SDK or CLI tooling must support them, and might require changes.

Stack name	Resource region		Layers	Instances	Apps	Actions
My Sample Stack (Linux)	us-west-2	regional	1	1	1	edit clone dele
php_application	us-west-2	regional	0	0	1	edit clone dele

-> stack with html hello world page



corresponding EC2 instance. Learn more.

Node.js App Server

Search for instances in this layer by name, status, size, type, AZ or IP

Hostname	Status	Size	Type	AZ	Public IP	Actions
nodejs-server1	stopped	t2.medium	24/7	us-west-2a	-	start delete
nodejs-server2	stopped	t2.medium	24/7	us-west-2a	-	start delete
nodejs-server3	stopped	t2.medium	24/7	us-west-2a	-	start delete

Instance

You can add more layers to this stack or register an instance.