

Hello Application

Overview:

Hello Application is test service, which allows users to perform GET operation and response with the JSON format value, for identifying which instances processed the request, you can validate the header information "Origin-Instance"

Load-balance URL: <http://hello-lb-606423950.us-east-2.elb.amazonaws.com:8080/>

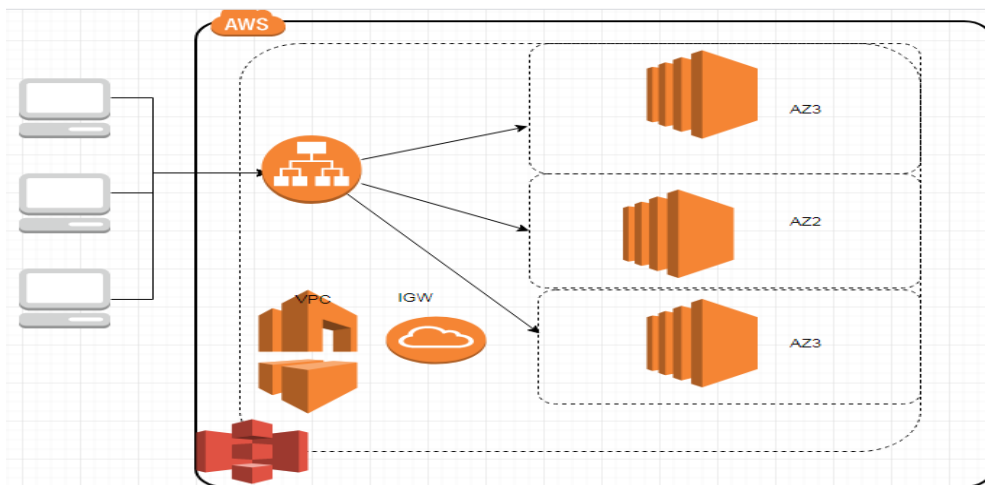
Repository: <https://github.com/sksindian/hello-service/tree/hello-dev>

Design:

Components:

- AWS Cluster in region US-EAST-2 and its 3 AZ
- AutoScalling group across the all 3 availability zone.
- Application Load-balance
- 1 VPC for US-EAST-2 region
- 3 Subnet across each Availability zone.
- 3 EC2 instances in each Availability zone
- S3 Bucket for uploading and downloading the code for running the instances
- terraform supporting multiple workspace for dev and prod
- Python3 with Flask is running as system service
- Terraform codes in GITHUB

Diagram:



Build Strategy:

- terraform created with workspace and it will allow to configure the DEV and PROD environment
- S3 created for keeping the files, and during the terraform build it will update the AWS credential and push the userdata to copy the script from s3 to local server
- So if you edited the files it will automatically re-upload the files in to S3
- the FrontEnd code written in python with Flask and we are updating the URL Header with

- instance ID, and return the json format
- In GIT Hub created the dev branch and once testing completed we can merge with master code
- Here is the evidence for header information changing in the every query and python requests is providing the input as json
- terraform apply output also uploaded into the github

```
[root@ip-172-31-15-94 ~]# curl -i http://hello-lb-606423950.us-east-2.elb.amazonaws.com:8080/
HTTP/1.1 200 OK
Date: Thu, 11 Jul 2019 20:20:22 GMT
Content-Type: application/json
Content-Length: 27
Connection: keep-alive
Origin-Instance: i-0f8cb6ef528d99eda
Server: Werkzeug/0.15.4 Python/3.7.3

{"response":"hello-world"}
[root@ip-172-31-15-94 ~]# curl -i http://hello-lb-606423950.us-east-2.elb.amazonaws.com:8080/
HTTP/1.1 200 OK
Date: Thu, 11 Jul 2019 20:20:32 GMT
Content-Type: application/json
Content-Length: 27
Connection: keep-alive
Origin-Instance: i-0c873ff660bb34382
Server: Werkzeug/0.15.4 Python/3.7.3

{"response":"hello-world"}
[root@ip-172-31-15-94 ~]# curl -X GET http://hello-lb-606423950.us-east-2.elb.amazonaws.com:8080/
{"response":"hello-world"}
[root@ip-172-31-15-94 ~]# python
Python 2.7.16 (default, Jun 19 2019, 17:20:54)
[GCC 7.3.1 20180303 (Red Hat 7.3.1-5)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import requests
>>> abc = requests.get("http://hello-lb-606423950.us-east-2.elb.amazonaws.com:8080/")
>>> abc.json()
{'response': 'hello-world'}
>>>
```

Name	Headers	Preview	Response	Timing
hello-lb-606423950.us-east-2....	Connection: keep-alive Content-Length: 27 Content-Type: application/json Date: Thu, 11 Jul 2019 20:03:38 GMT Origin-Instance: i-0b0e9f3772a02a922 Server: Werkzeug/0.15.4 Python/3.7.3			

AWS Console Output:

Instances are Running in different availability zone

Filter:

1 to 1 of 1 Auto Scaling Groups

<input type="checkbox"/>	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Gr
<input type="checkbox"/>	helloAG	terraform-20190717133...	3	3	3	3	us-east-2a, us-east-2b, us-e...	60	300

Auto Scaling Group: helloAG

Details

Activity History

Scaling Policies

Instances

Monitoring

Notifications

Tags

Scheduled Actions

Lifecycle Hooks

Filter: Any Status

1 to 3 of 3 History Items

	Status	Description	Start Time	End Time
▶	Successful	Launching a new EC2 instance: i-0bb3e922806fff598	2019 July 17 19:03:25 UTC+5:30	2019 July 17 19:03:59 UTC+5:30
▶	Successful	Launching a new EC2 instance: i-0fea8fbe2929a2f73	2019 July 17 19:03:25 UTC+5:30	2019 July 17 19:03:58 UTC+5:30
▶	Successful	Launching a new EC2 instance: i-0a4dc0f0b2181f712	2019 July 17 19:03:25 UTC+5:30	2019 July 17 19:03:58 UTC+5:30

Target group with all active state

<input type="checkbox"/>	Name	Port	Protocol	Target type	Load Balanc	VPC ID	Monitoring
<input type="checkbox"/>	hello-tg	8080	HTTP	instance	hello-lb	vpc-02f09abf84d40e90b	<input type="checkbox"/>

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health check. As demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Edit

Registered targets

Instance ID	Name	Port	Availability Zone	Status
i-0bb3e922806fff598		8080	us-east-2b	healthy ⓘ
i-0fea8fbe2929a2f73		8080	us-east-2a	healthy ⓘ
i-0a4dc0f0b2181f712		8080	us-east-2c	healthy ⓘ