01)

Zappa - a framework for running "serverless" Python web applications utilizing AWS Lambda and AWS API Gateway. Zappa handles the greater part of the arrangement and organization naturally - now, you can send an endlessly adaptable application to the cloud with a solitary order - for a moment division of the cost of a conventional web server. The significant distinction is that the server's whole life expectancy exists inside the lifecycle of a given HTTP request. Where typical web servers like Apache and Nginx need to sit out of gear day in and day out, sitting tight for new demands to come in, with Zappa, the server is made after the HTTP request which comes in through API Gateway. At that point, it transforms the API Gateway request into typical Python WSGI, processes the request, and returns it back through the API Gateway to the client. And after that, poof - the server is no more.

Versatility

This accompanies some significant focus points over conventional web servers. The first is adaptability. Since AWS Lambda handles the greater part of the solicitations, you can have the same number of reactions prepared in parallel as you require. With AWS Lambda, you get 100 executions for every second ideal out of the case, yet the cutoff is subjective and on the off chance that you scale past that you just need to request that Amazon raise your number of requests.

Cost

The following significant preferred standpoint is taken a toll. With AWS Lambda, you pay by the millisecond. So as opposed to paying to have a meaty EC2 machine running all day, every day for your site, you just pay in light of the measure of solicitations, your server - which regularly implies you'll just be paying pennies every month for a conventional site. Also the cost saving on not spending time on organization, operations, and support!

Viability and Ease of Use

Zappa is additionally amazingly simple to convey. It's actually a solitary summon - python manage.py send generation - to arrange and convey your code, and from that point onward, you never need to stress over it again. No provisioning machines, no setting up web servers, no DevOps, no working frameworks, no security overhauls, no fixing, no downtime. It just works!

Hacks

AWS Lambda and API Gateway are new advances, so there are so many hacks that make this all conceivable. Those incorporate, however, aren't constrained to:

- Using VTL to outline, headers, technique, params and question strings into JSON, and after that transforming that into substantial WSGI.

- Attaching reaction codes to response bodies, Base64 encoding the entire thing, utilizing that as a regex to course the reaction code, interpreting the body in VTL, and mapping the response body to that.

- Packing and Base58 encoding various treats into a solitary treat since we can just guide exceptional.

- Turning treat setting 301/302 reactions into 200 responses with HTML redirects, in light of the fact that we have no real way to set headers on redirects.

In the event that you need to take in more, investigate the hood!

Future Work

In spite of the fact that Zappa is currently include finish enough for an underlying discharge, there is as yet a considerable lot of work to do. For example, there is just a single customer library up until this point, Django-Zappa, yet it ought to be genuinely simple to include bolster for Flask, Pylons, and some other WSGI python web system. A similar rule that makes Zappa conceivable ought to likewise work for NodeJS applications, yet I think that it is significantly more agreeable to create in Python.

02)

Cohesion in software engineering, as in real life, is how much the elements consisting a whole(in our case let's say a class) can be said that they actually belong together. Thus, it is a measure of how strongly related each piece of functionality expressed by the source code of a software module is. One way of looking at cohesion in terms of OO is if the methods in the class are using any of the private attributes. Now the discussion is bigger than this but High Cohesion (or the cohesion's best type - the functional cohesion) is when parts of a module are grouped because they all contribute to a single well-defined task of the module.

Coupling in simple words, is how much one component (again, imagine a class, although not necessarily) knows about the inner workings or inner elements of another one, i.e. how much knowledge it has of the other component.Loose coupling is a method of interconnecting the components in a system or network so that those components, depend on each other to the least extent practically possible