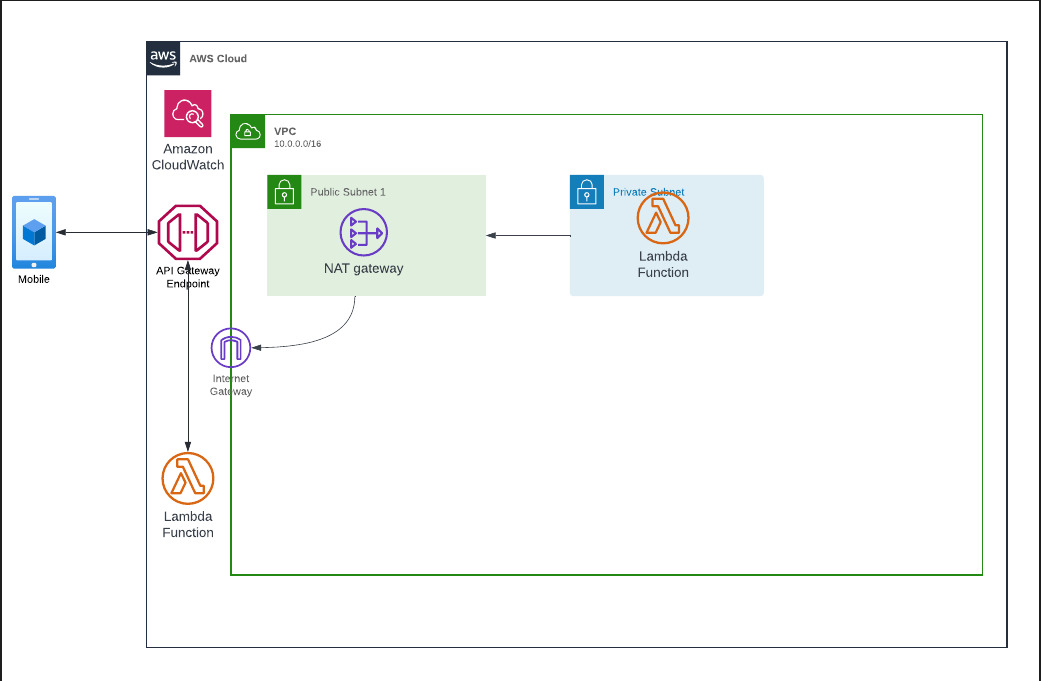
## Setup infrastructure for Events API endpoint using AWS Web Console.



**Steps for creating the stack from AWS Web Console:**

1. Under Lambda service, create Lambda Function with default execution role. Update code on Lambda function to handle Authorization from request headers and return statuscode and response accordingly. See index.js file in Lambda Code folder of this repo.
2. Under CloudWatch Log groups, create Log group with name "EventsAPIRequestLogs", this will be used for API Gateway requests logging.
3. Setup AWS API Gateway with a name "EventAPI" and change $default stage to prod (and allow auto deploy)
4. Under Routes, add Routes for /events with POST method
5. Configure Integration for the Lambda function within the route under POST method
6. Enable Logging with above created Log Group arn
7. To add external access to Lambda Function use attached CloudFormation template to spin up VPC which will complete step 1-6, then continue from step 7.
8. Create VPC
9. Create Public & Private Subnet
10. Create and attach internet gateway to VPC
11. Allocate Elastic IP
12. Create NAT Gateway within public subnet and associate Elastic IP
13. Add routes for public and private subnets: Under route tables, add 0.0.0.0/0 targeting to internet gateway in public subnet and 0.0.0.0/0 targeting to Nat Gateway in private subnet.
14. Add AWSLambdaVPCAccessExecutionRole permission in existing Lambda IAM role configured on the Lambda created above
15. Configure Lambda function to connect to VPC
16. Under VPC configuration of Lambda Function, choose Edit. Then, do the following:
17. For Virtual Private Cloud (VPC), choose the VPC created above.
18. For Subnets, select the private subnet created above.

**Steps to tear down the above setup from AWS Web Console:**

1. Delete Lambda Function and delete its associated IAM Role along with its customer managed permission policy.
2. Delete API Gateway and delete the associated IAM Role setup on creation.
3. Delete Log group created for API Gateway request logging from CloudWatch
4. Delete stack from CloudFormation Stack created with attached cf.template file.

**Explanation for using above approach to this project:**

I have chosen to use Lambda Function and API Gateway for this stack as that was the first thing that came to my mind when reading the requirements and also, I am more familiar with the AWS services to setup such API. Traditionally, I have setup API on Ec2 instance such as WAMP or NodeJS with middleware such as Express to listen to web request, which is unsecure and hard to manage and scale, as well as not so much cost effective provided we have to manage the underlying server OS and Runtime environments, Lambda together with API Gateway provides robust, scalable and future proof approach to spin up API backends.

Application Load Balancer is on the other hand used for distributing the load across different computer units, so in this case using API Gateway is more suitable option.