Link for Hello World program on Elastic beanstalk using Visual studio AWS ToolKit:

<http://hellowordwebapp-env.ypsj2cpqgx.us-east-1.elasticbeanstalk.com/>

1:The Asp .net web application Hello World which i have built is through Visual studi0 2017.

Prerequisite for any .net web application.

1.Visual Studio 2013/2015/2017

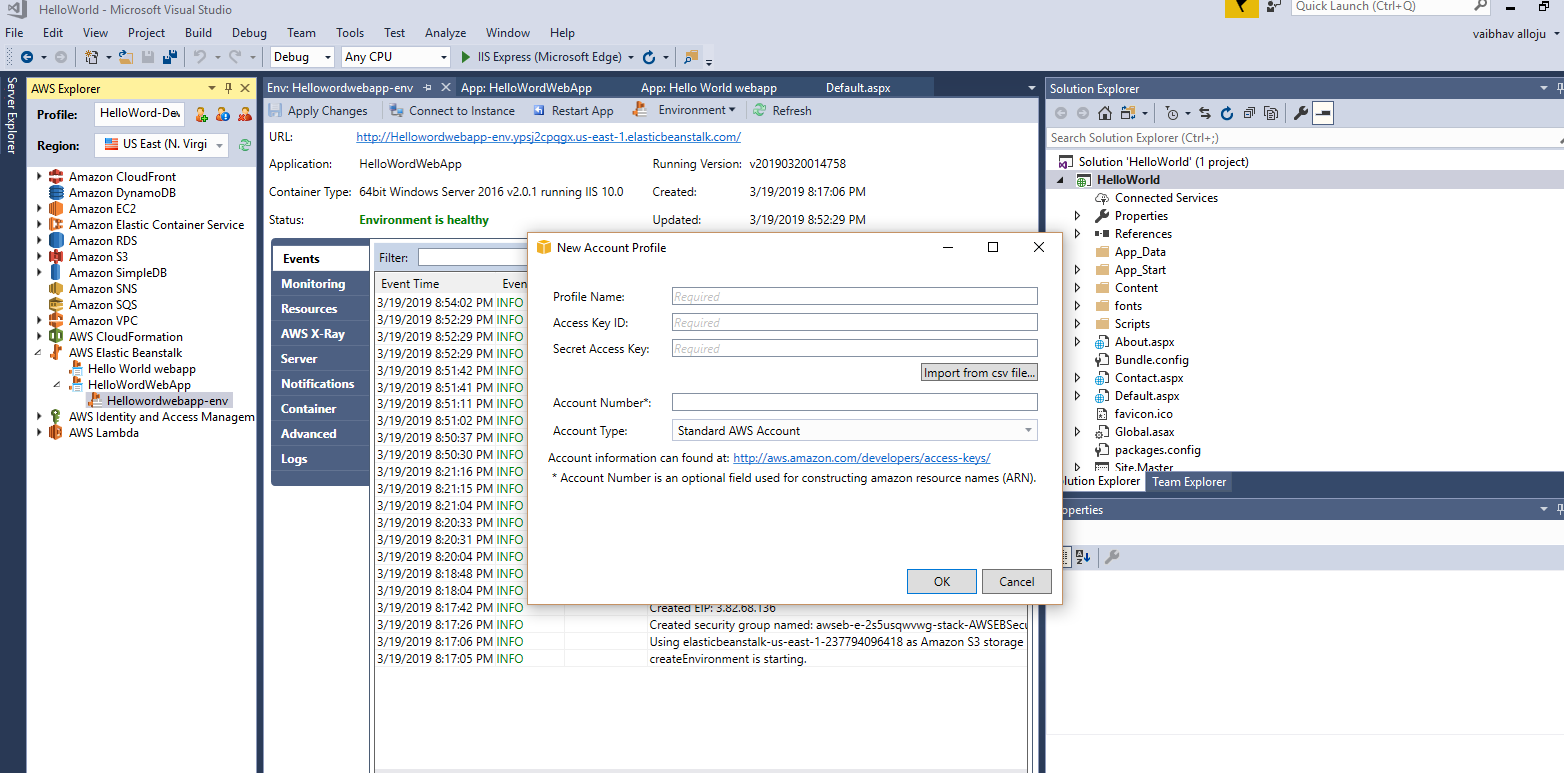
2.AWS subscription

3.AWS toolkit for Visual studio Install.

2.Once we install AWS toolkit for visual studio we get an option for publishing application on AWS Elastic Beanstalk.

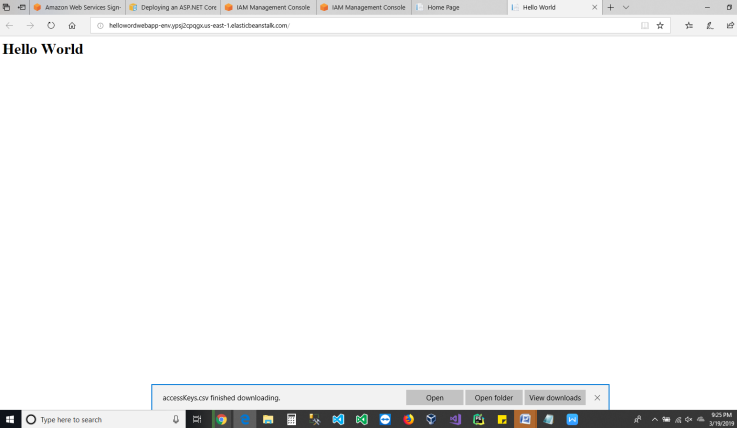
3.We get a window where we need to configure(Application,Environment,AWS options,VPC,Updates,Permissions,Options,Review)

* we need to create profile using Access Key ID,Secure access key .We get them after configuring IAM.



* Elastic beanstalk can be created directly in aws console or we can also configure in aws toolkit for visual studio.
* Create an IAM user. This will create an alias, essentially, for your account. You will need the access key and secret key, so download the file when prompted.  
  Second, while in the IAM management portal, click on Policies, on the left. You can check up to two options, and use them to create a Policy. Choose EC2 and Elastic Beanstalk, full access for each. You should do this with the IAM user you created,
* IF configuring through toolkit please specify Application Name, Environment Name, Url(Give the same name as Environment Name and check for availability)
* Provide Container Type, Instance Type, Keypair(if you have any),Check Single Instance Environment.
* Provide Project build configuration, App pool runtime, App Path
* Deployment version label is automatically generated.

4. Once everything is reviewed we can deploy the asp .net application to elastic beanstalk.



<https://www.hackerrank.com/challenges/validating-credit-card-number/problem>

**Validating Credit Card Numbers**

#condition 1, imports reference

import re

#condition 2, validates if input consists only number  
for k in range(int(raw\_input())):  
    S = raw\_input().strip()

#condition 3, validates if input consists only number using search function and must contain 16 digits and must only consists of digits from 0-9  
    pre\_match = re.search(r'^[456]\d{3}(-?)\d{4}\1\d{4}\1\d{4}$',S)  
    if pre\_match:

#condition 4, if you have – in between the numbers we can use split function to remove them  
        processed\_string = "".join(pre\_match.group(0).split('-'))

#condition 5,it checks it should not have 4 or more consecutives here  
        final\_match = re.search(r'(\d)\1{3,}',processed\_string)  
        print 'Invalid'

if final\_match else 'Valid'  
    else:  
        print 'Invalid'