## Steps to execute PS script for Application Configuration

#### **Application Server**

#### Application server Name:

TNP

CoreIssue

CoreAuth

WorkFlow

Services Appserver

**Table of Contents**

Initial Steps [2](#_Toc28575)

[Prerequisites 2](#_Toc8904)

[Steps to perform unit testing 2](#_Toc17010)

[output of application configuration automation  6](#_Toc24442)

[Application Server Script Details 6](#_Toc24061)

#### Initial steps:

* User must have access to AWS account.

#### Prerequisites: (we are installing it through powershell scripts)

**Following software should be installed:**

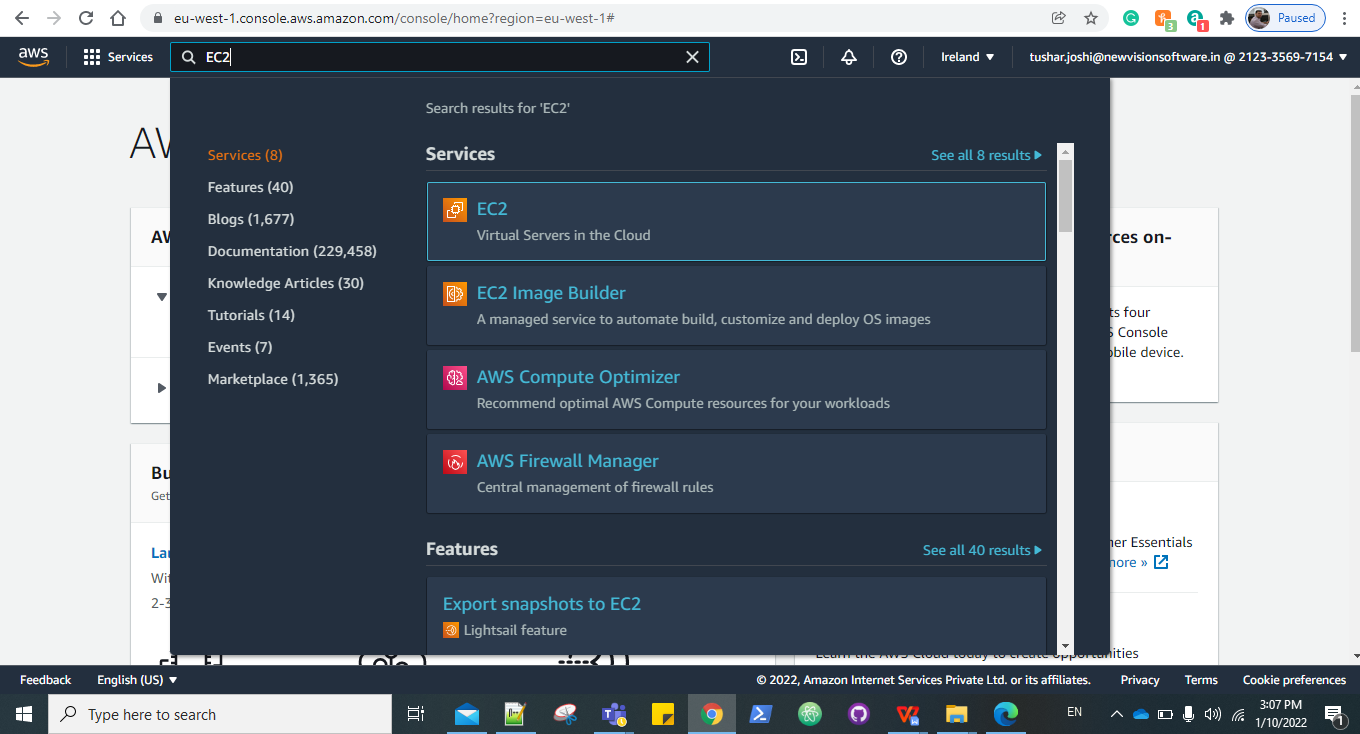
1. Redistributable
2. Native client
3. Mysqlodbc(32-bit)
4. AWS CLI
5. SSM Agent
6. Cloudwatch Agent

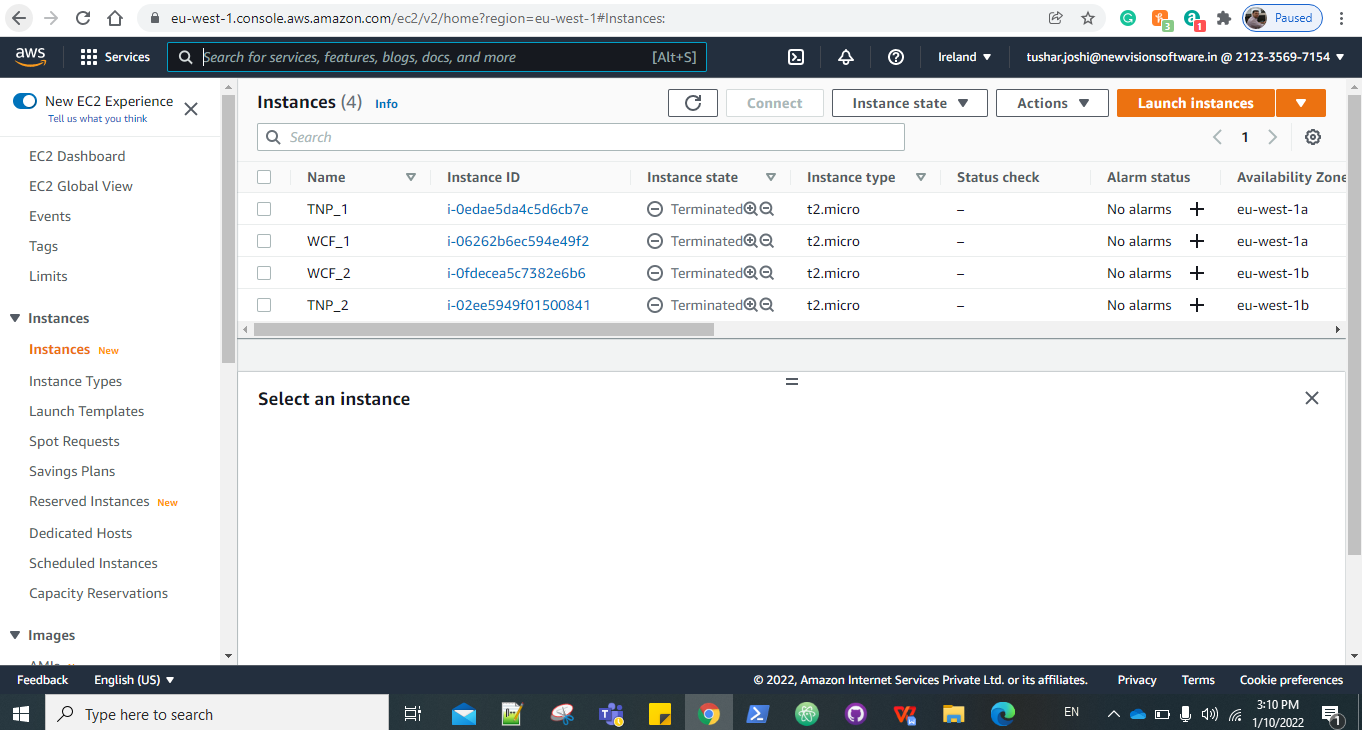
**Two buckets should be created in S3:**

1. corecard-setup-files: where all the data of core card will be uploaded.
2. application-configuration-scripts: where all the PS scripts will be uploaded under Application\_Configuration\_server folder.
3. In **corecard-setup-files** there will be a folder of name **Prerequisites** which will contains all the prerequisites EXE & MSI files that needs to be installed.

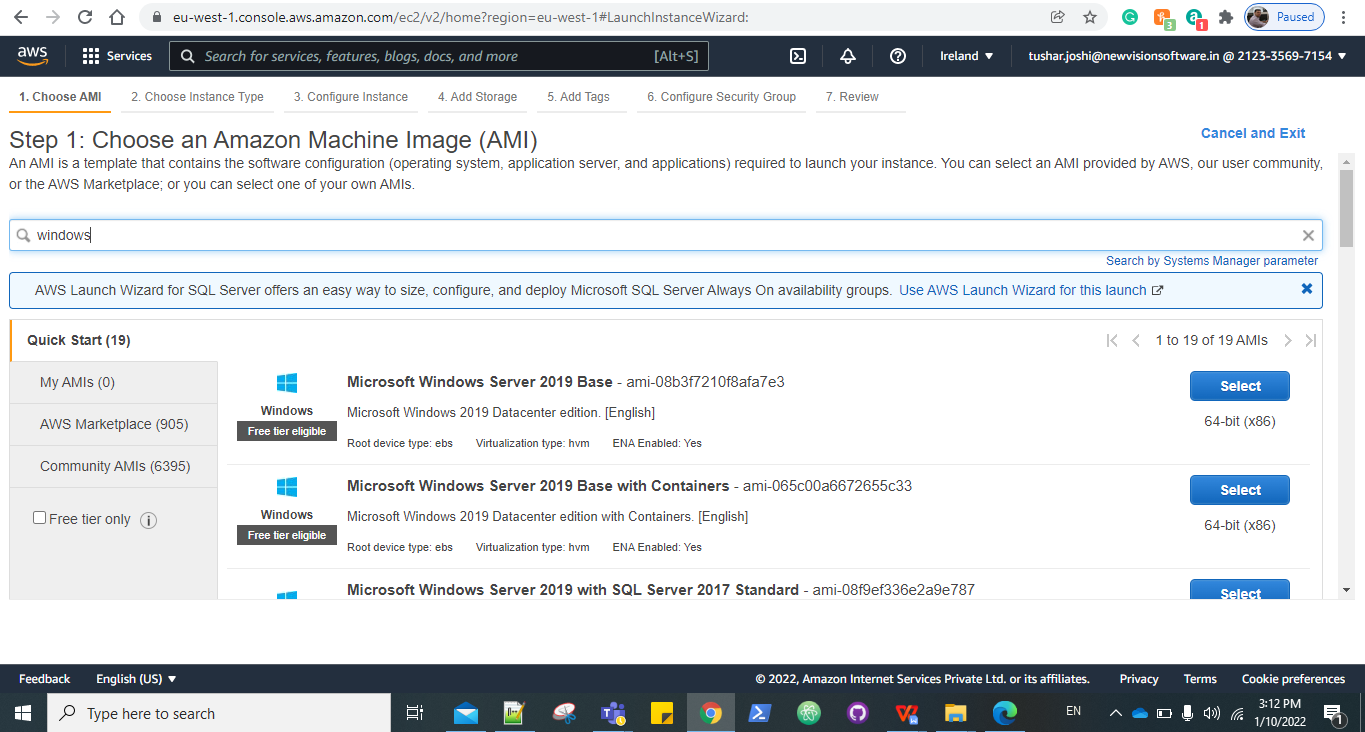
**Steps to perform unit testing: (These steps should be performed for manual implementation)**

Step1. Search EC2 & click on EC2 and then click on launch instance

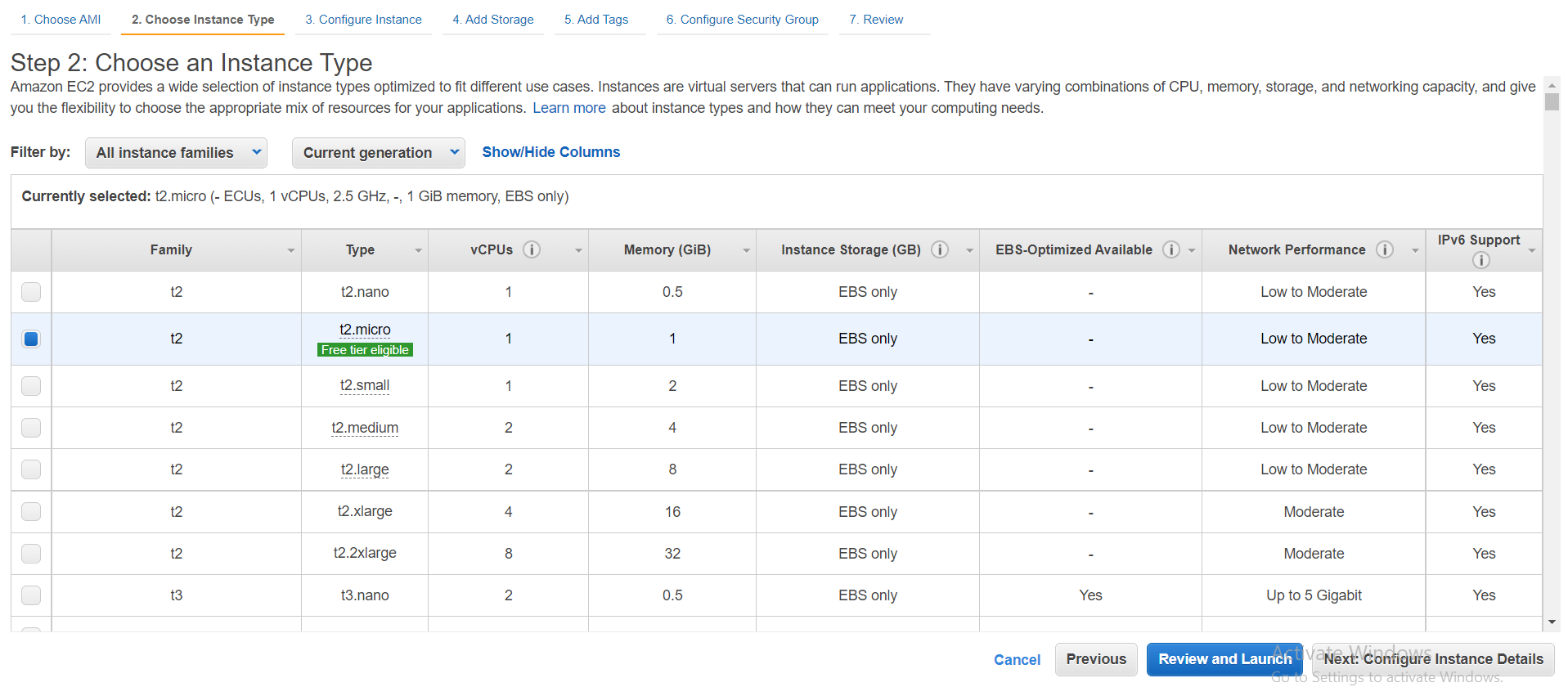




Step2.Choose Amazon machine image



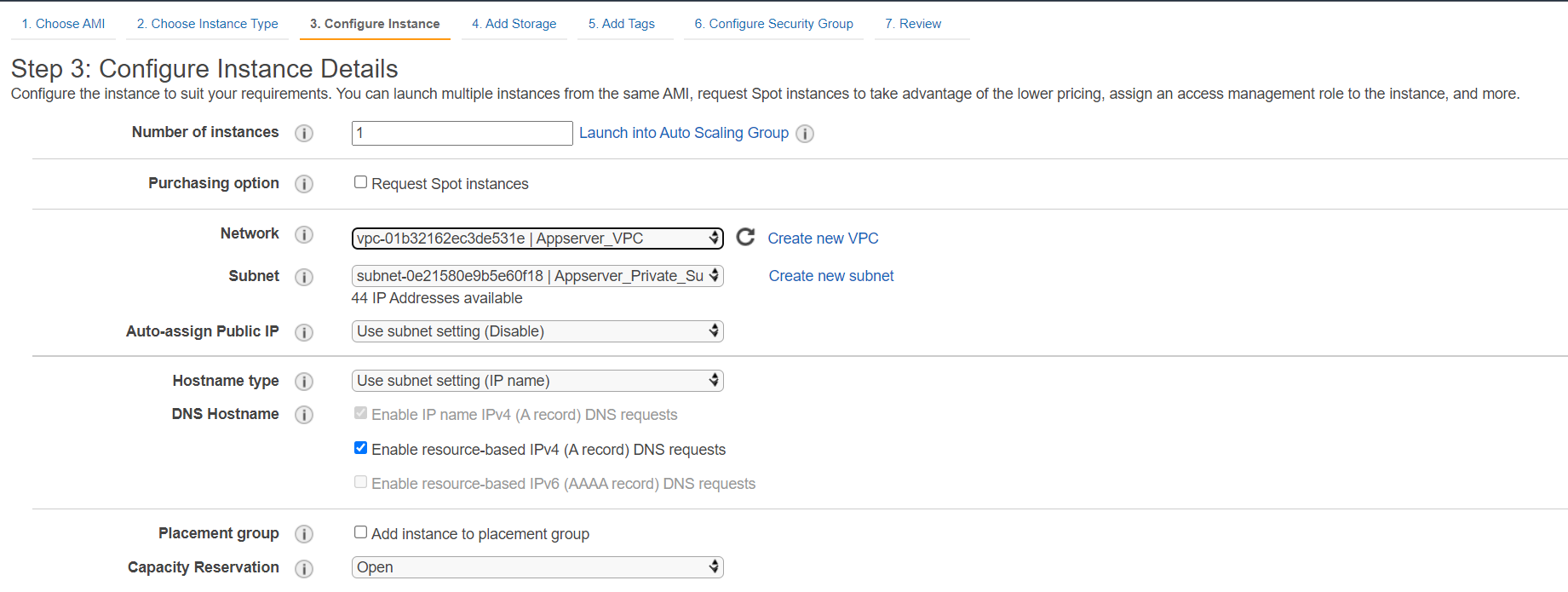
Step 3: Choose an Instance Type.

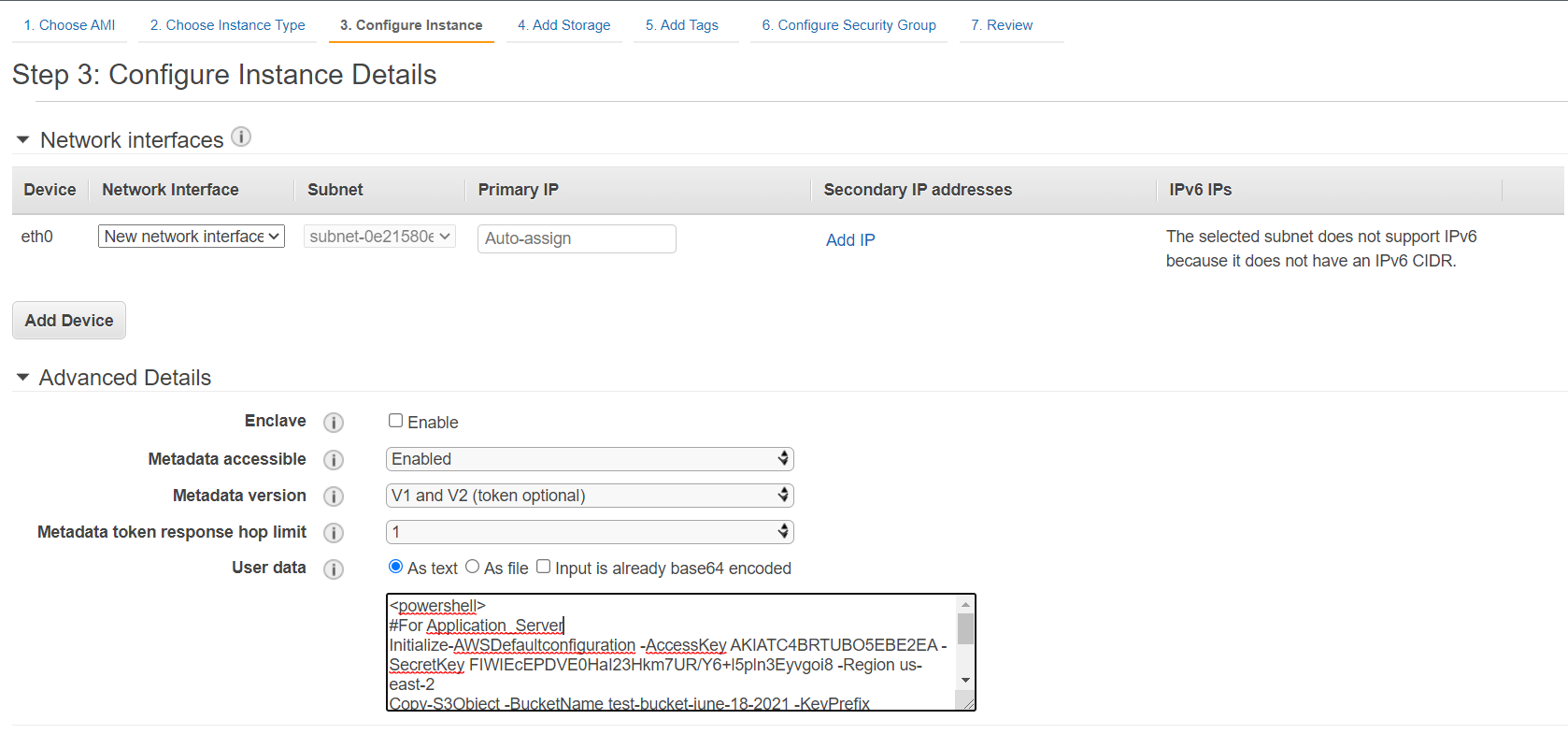


Step 4: In Configure Instance details select the no. of instances, Application\_Configuration\_VPC, and in user data paste the following command:

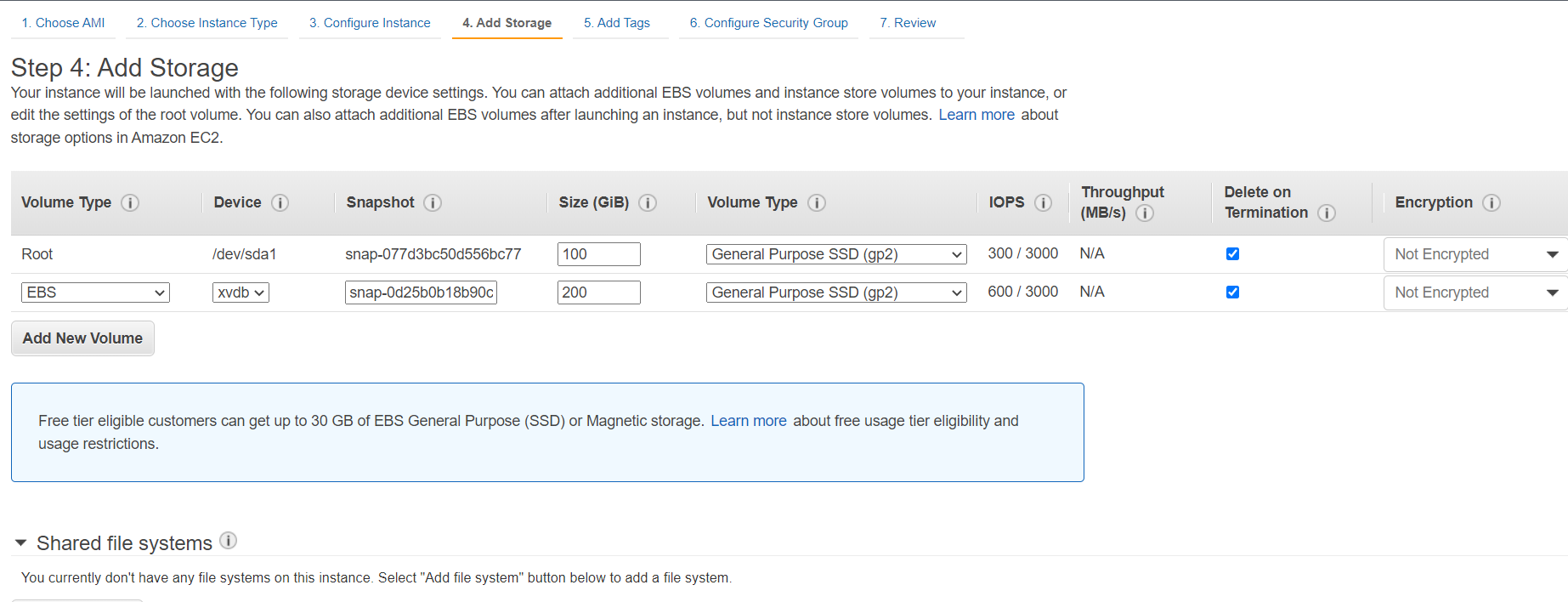
|  |
| --- |
| **<powershell>**  **#For Application\_Server**  **Initialize-AWSDefaultconfiguration -AccessKey <AccessKey> -SecretKey <SecretKey> -Region us-east-2 Copy-S3Object -BucketName application-configuration-scripts -KeyPrefix Application\_Configuration\_server -LocalFolder C:\ -Force**  **#Hit following command after successfully copying data from s3 bucket**  **Powershell.exe -File C:\Master.ps1 </powershell>** |

Note: Enter you AWS access key and secret key ID.

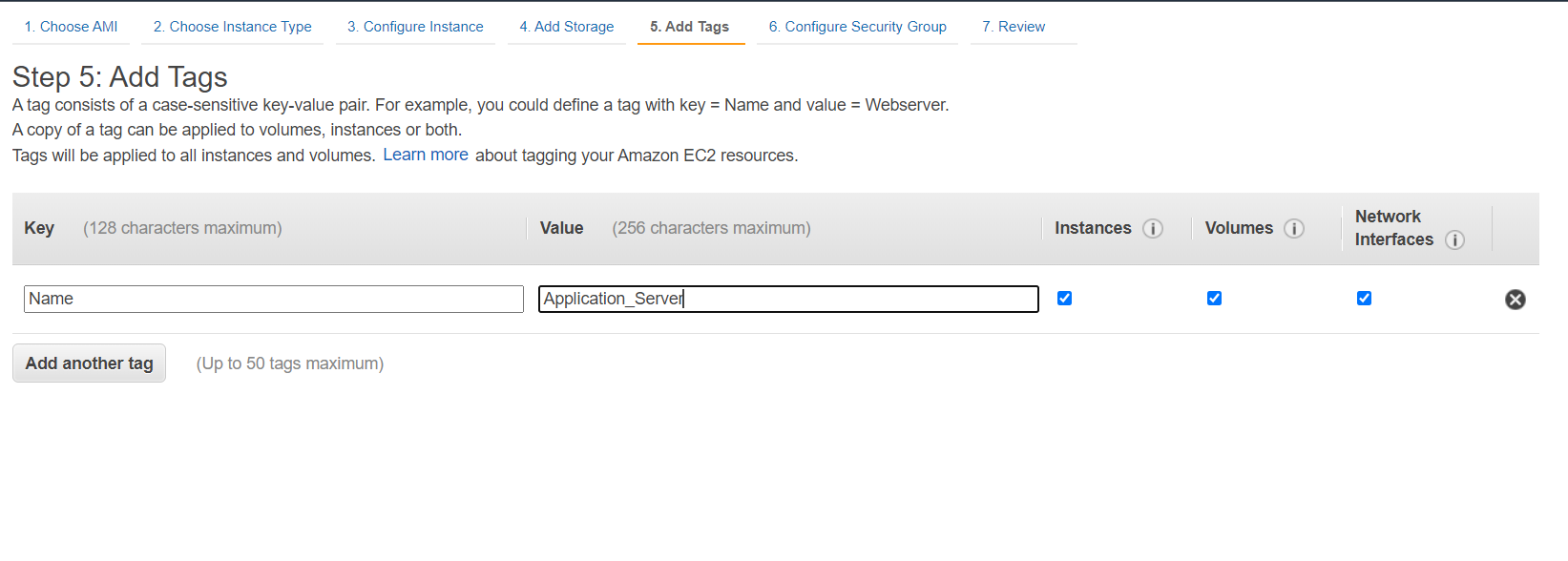




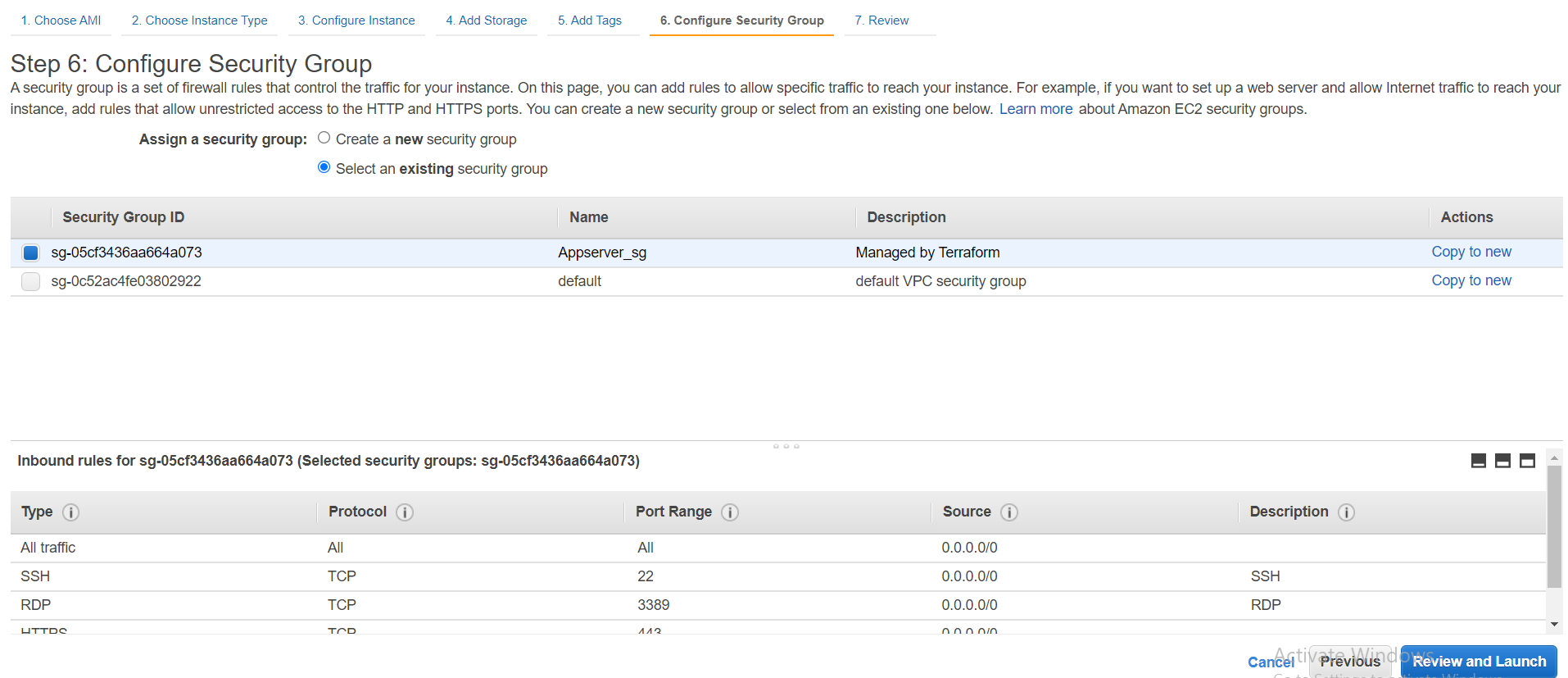
Step 5: Add storage.



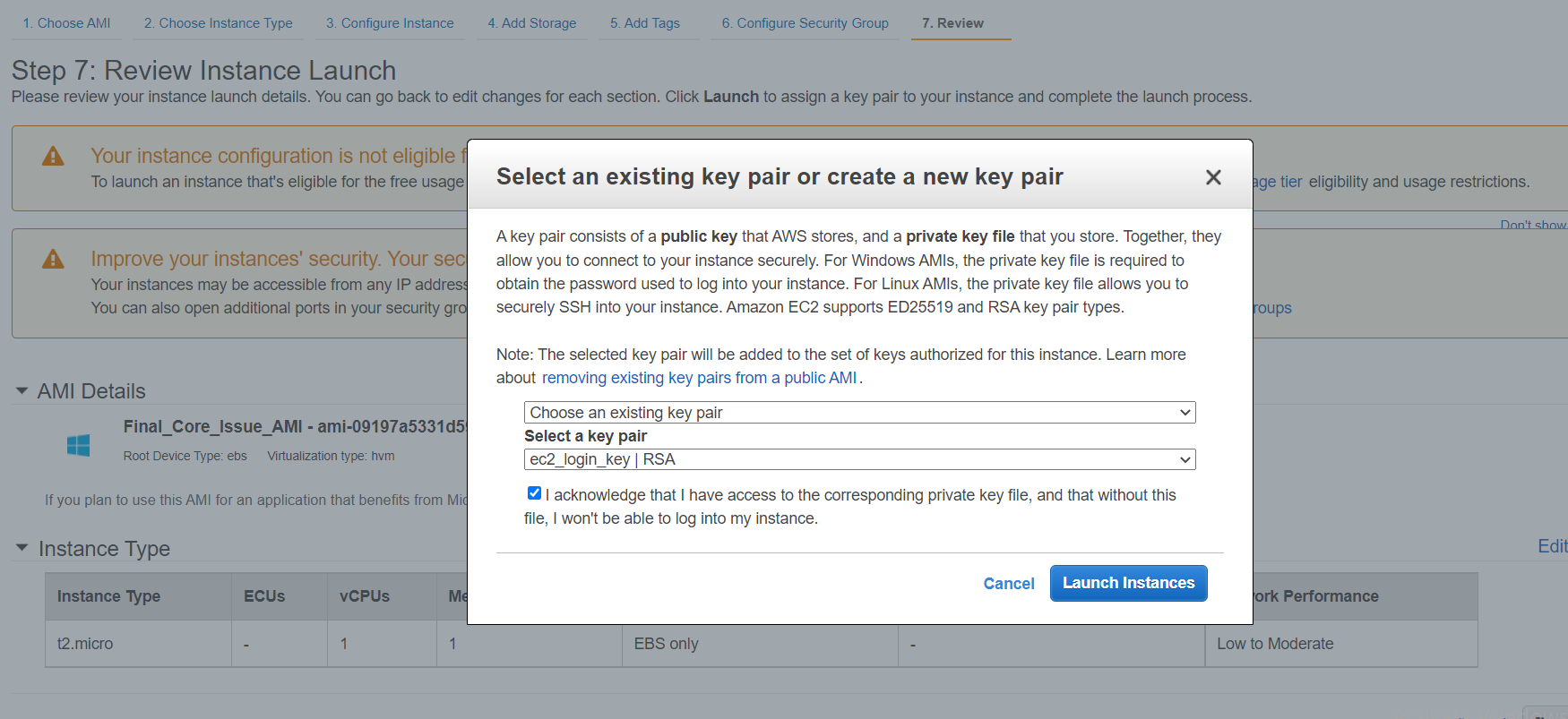
Step 6: Give the Tags if needed.



Step 7: Select the security group with name **Appserver\_sg**.



Step 8: Review and launch, Select the existing key pair or create new and launch.



Step 9: It will take some time to configure the whole server and the scripts will be running in background.

#### **To see the output of application configuration automation:**

1. User should connect to bastion host.
2. Then, from bastion host user can connect to server through RDP.

#### **Application Server Script Details:**

In the S3 bucket all the files listed below are present:

**Note**: Please refer S3 bucket for updated files

1. **Administrator\_password**: This script will set the password for user “Administrator”
2. **CI\_Client\_DSl :** This script file is used to import variables for each object of CI\_client\_var from path “C:\testvariable.xml “ and it will define the variables in lookuptable array.

Also it will check if the contents like $\_.Key matching with $original\_file.If it's matches then it will define the source path in $destination\_file for those contents.

1. **Cleanup:** This script file is used to cleanup all related files from C & D drive. Once cleanup completed server will ask to permission to restart the machine.
2. **Core\_Auth\_Configuration:** This script file is used to import variables for each object of Core\_Auth from path “C:\testvariable.xml “ and it will define the variables in lookuptable array.

Also it will check if the contents like $\_.Key matching with $original\_file.If it's matches then it will define the source path in $destination\_file for those contents.

1. **Core\_Issue\_Config:** This script file is used to import variables for each object of Core\_Issue from path “C:\testvariable.xml “ and it will define the variables in lookuptable array.

And then it will check if the contents like $\_.Key matching with $original\_file.If it's matches then it will define the source in $destination\_file for those contents.

1. **CoreAuth\_Client\_DSL:** This script file is used to import variables for each object of Core Auth Client from path “C:\testvariable.xml “ and it will define the variables in lookuptable array.

And then it will check if the contents like $\_.Key matching with $original\_file.If it's matches then it will define the source path in $destination\_file for those contents.

1. **Drive\_Configuration:** This script is used to get the drive details where objects are raw, then it will initialize the disk and will create new partition with volume of NTFS filesystem
2. e**xtract\_msi\_from\_dbbide:** This script is used to extract MSI by executing dbbide4.2.42.12.exe file & it will store the MSI file into C drive
3. **Extractzip:** This script is used to extract the contents from zip file D:\DBBSETUP\MonitoringScript.zip and store at DestinationPath D:\DBBSETUP. Then it will remove all zip file contents from LiteralPath.
4. **install\_prerequisites**: This script is used to install all prerequisites as below:
5. INSTALL SQL NATIVE CLIENT 11
6. INSTALL REDISTRIBUTABLE FILE
7. INSTALL ZIP.EXE
8. AWS CLI
9. odbc DRIVER 17
10. CLOUD WATCH AGENT
11. SSM AGENT
12. Redistributable
13. **install-bddie:** This script is used to install DBBIDE4.2.42.12.msi application into path D:\CC\_Runtime

12**. master:** This is master script used to execute all power shell scripts to configure application servers

13**. ODBC Setup:** This script is used to setup ODBC Driver 17 for SQL Server

1. **powershell\_aws\_commands:** This script will copy all the core card data from S3 bucket “corecard-setup-files”
2. **Powershell\_commands:** This script is used to copy items from source to destination path
3. **powershell\_new\_directories:** This script is used to create directory “CC\_Runtime“ under D drive:
4. **tls\_script:** This script is used for Configuring IIS with SSL/TLS Deployments
5. **wfCommonAuth\_DSL:** This script file is used to import variables for each object of Core Auth Client from “C:\testvariable.xml “ and it will define the variables in lookuptable array and then it will check if the contents like $\_.Key matching with $original\_file.If it's matches then it will define the source in $destination\_file for those contents.
6. **Windows\_Firewall\_config:** This script is used to setup firewall configuration AppServer-Ports by allowing Inbound port numbers '4421','4422','4427'

1. **testvariable.xml:** In this xml file contains all the variables for all the servers, In this file the tag name will fetch all the variables for that specific environment Eg. <ENV>DEV</ENV>

It will consist all environment tag which will include all server tags.

1. **scale\_file\_creation:** Following script will create scale file in D:DBBSETUP/Scalefiles, Script will get all ec2 instance based on the naming parameter.

App server follows following parameter,

1. It will fetch all the IP from ec2 instance which names are CC\_AUTH, CC\_ISSUE, CC\_SVC, CC\_WEB

2. In source section all webservers Ip will be added.

3. In sink section all CC\_AUTH, CC\_ISSUE, CC\_SVC Ip will be added.

Original file contains all the values which will be needed to set in destination file. for example base\_var is declared in Core\_Auth\_Configuration.ps1 and it will set the value in destination file.

Original files are as below:

1. **CI\_Client.DSL**
2. **CoreAuth\_Client.DSL**
3. **wfCommonAuth.DSL**

**4.- CoreAuthSetup**

**5. SetupCI**