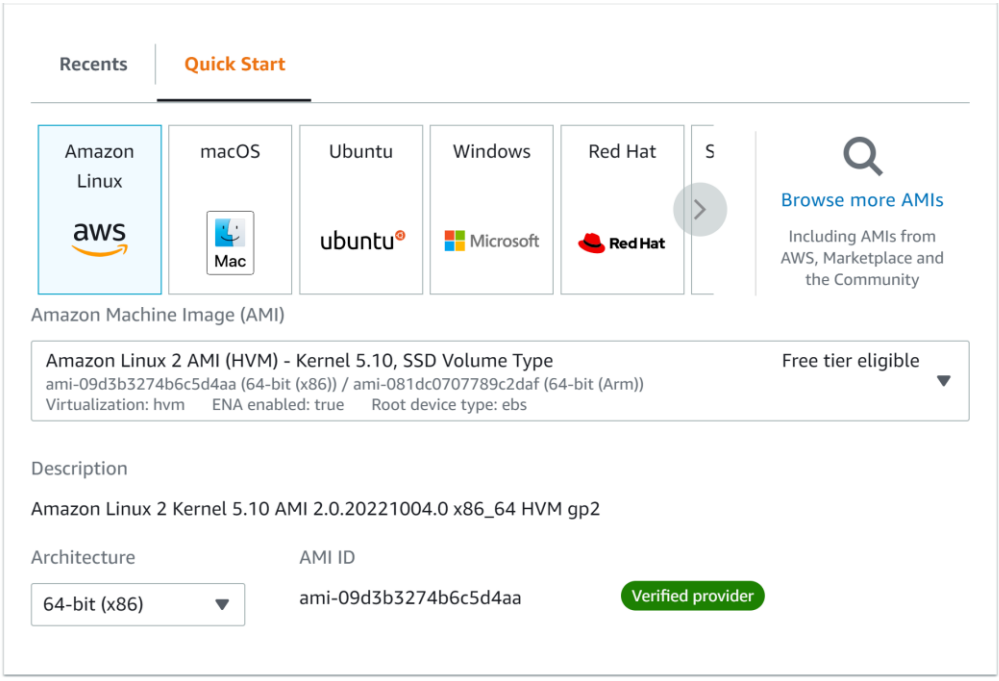
Create EC2-Instance

I. Login to aws console
II. Go to EC2 under the services section
III. Click on Launch instances

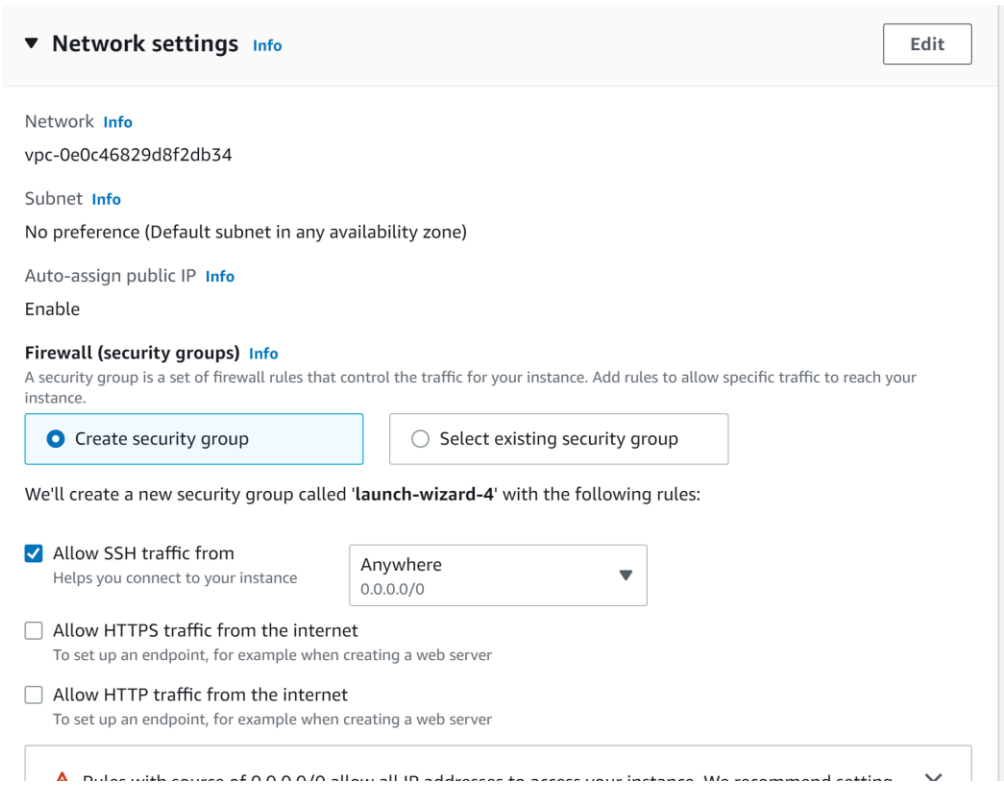
Name: <Machine-1>

Application and OS Images (Amazon Machine Image)

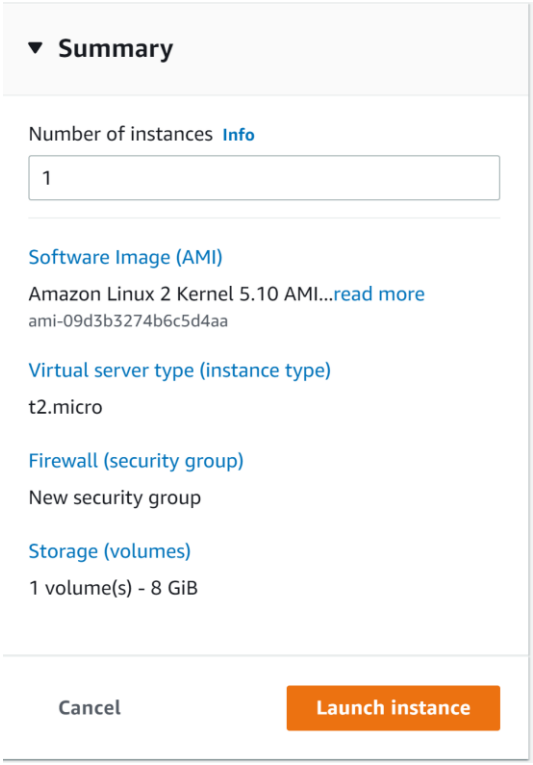


Instance type : t2.micro

Key pair (login): already created key pair



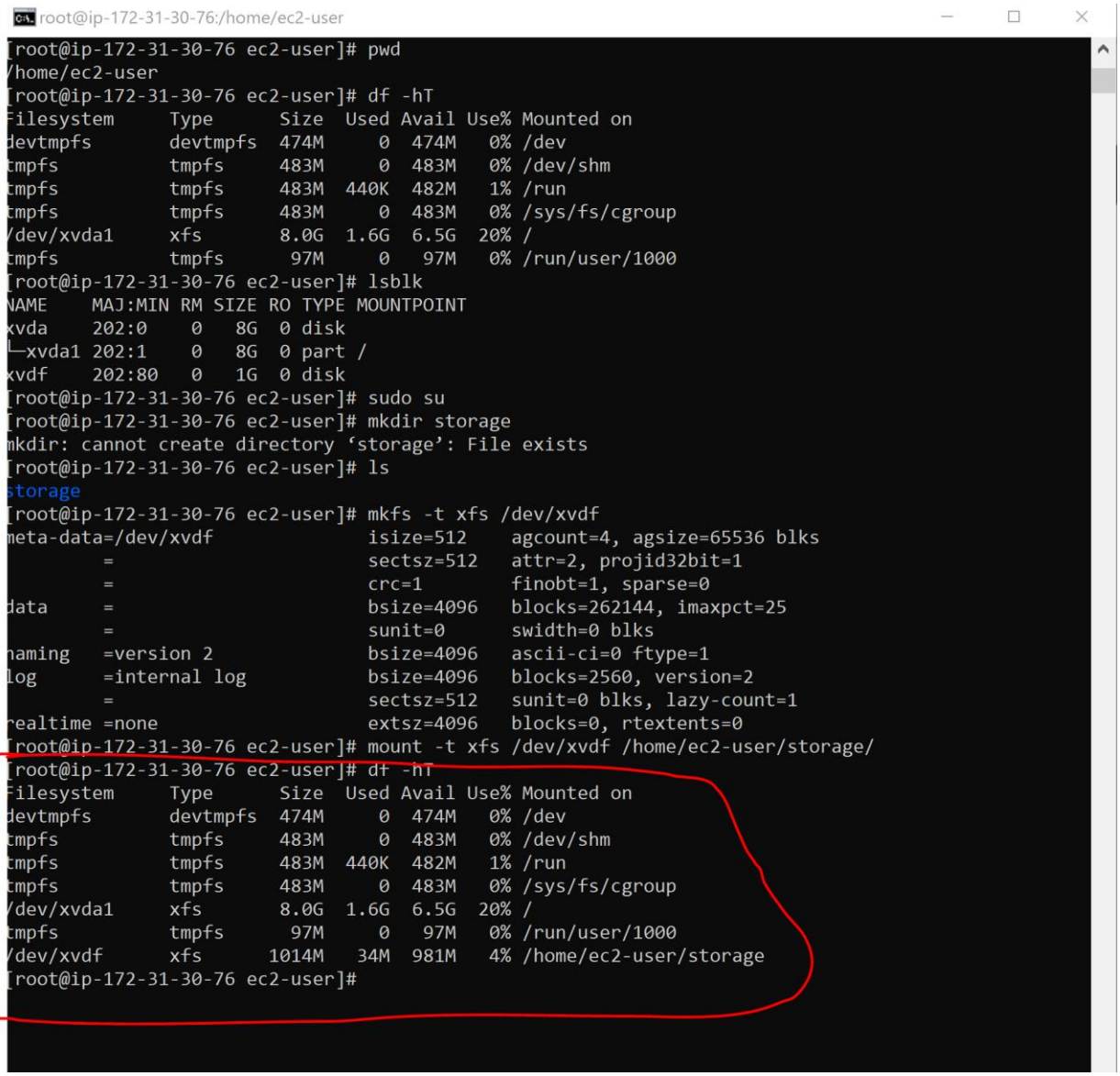
Check Summary

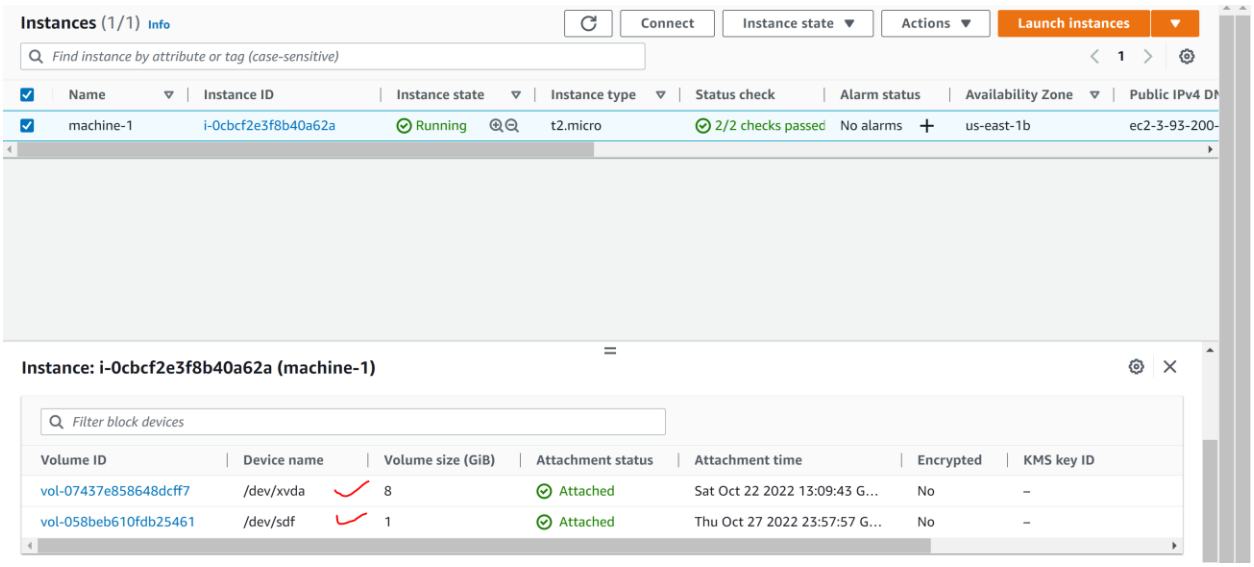


Launch Instance

2. Create Elastic Block Store:

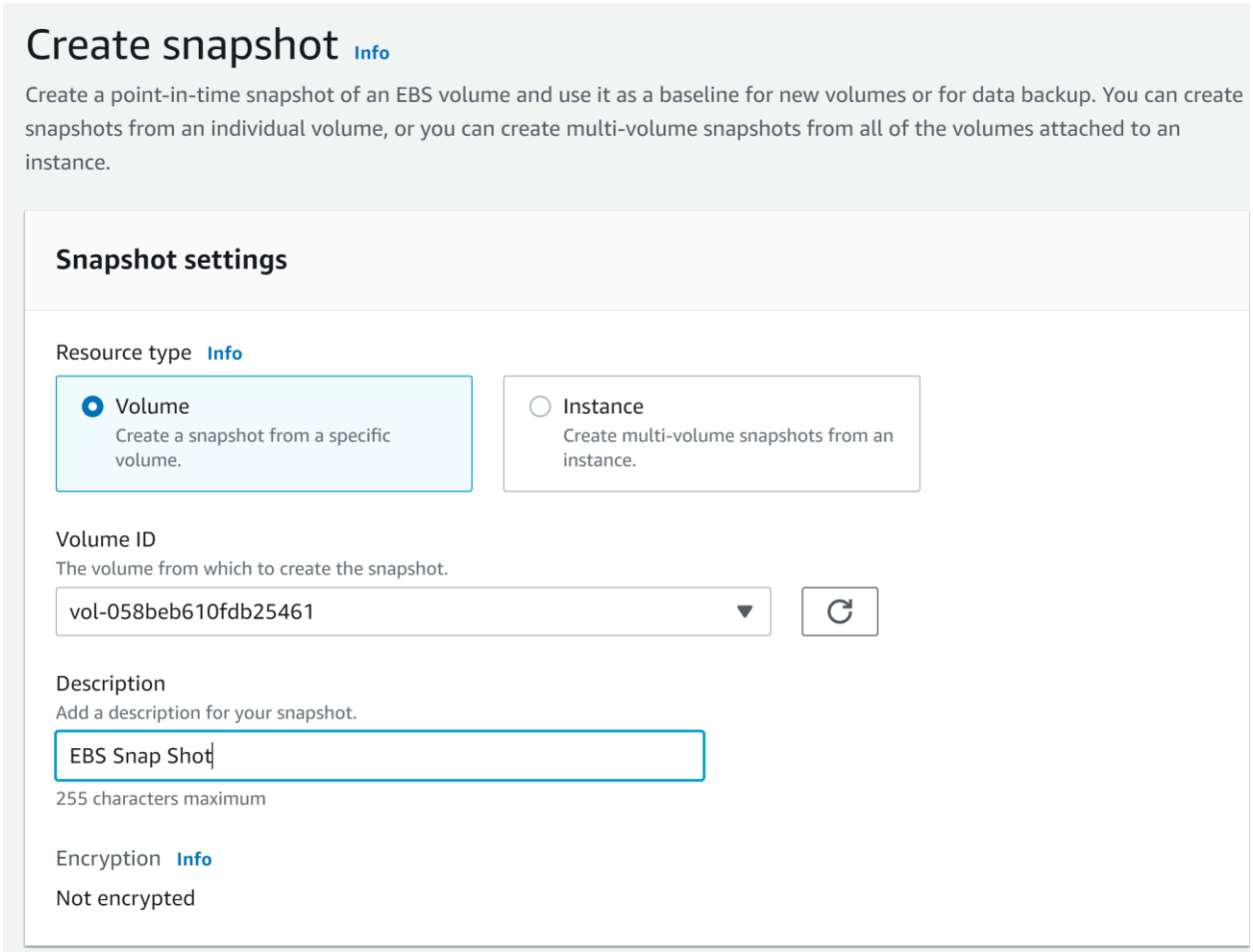
- Go to volumes
- Create volume
- Select availability zone which is ec2 instance is available zone
- Go to action → attach volume

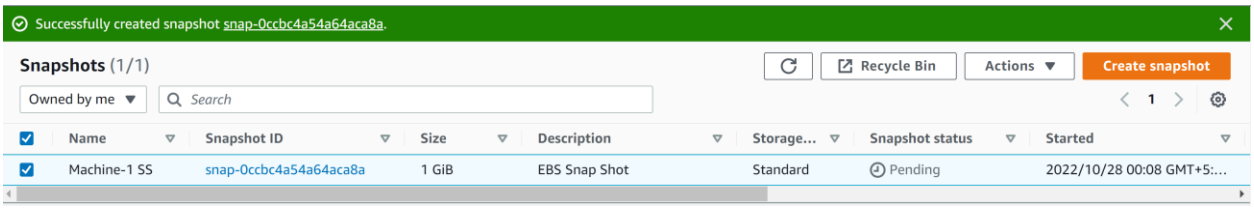




3. Snapshot Screenshot creation:

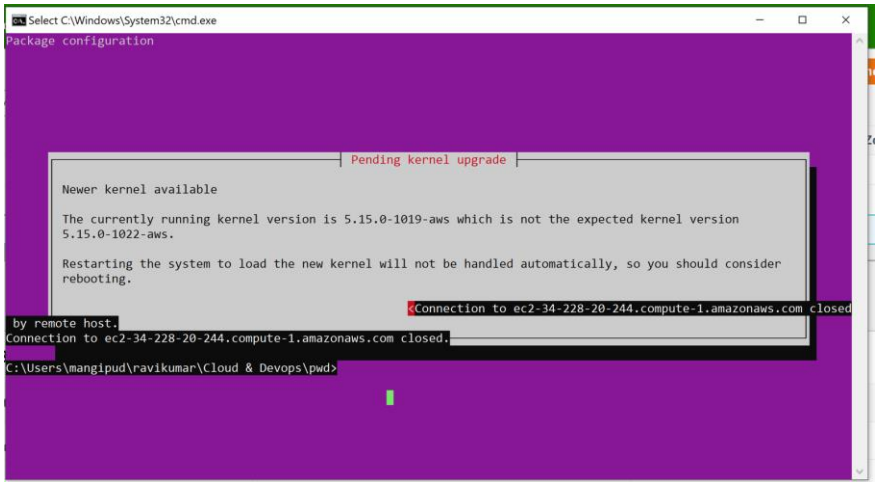
Example taken only for volume snapshot





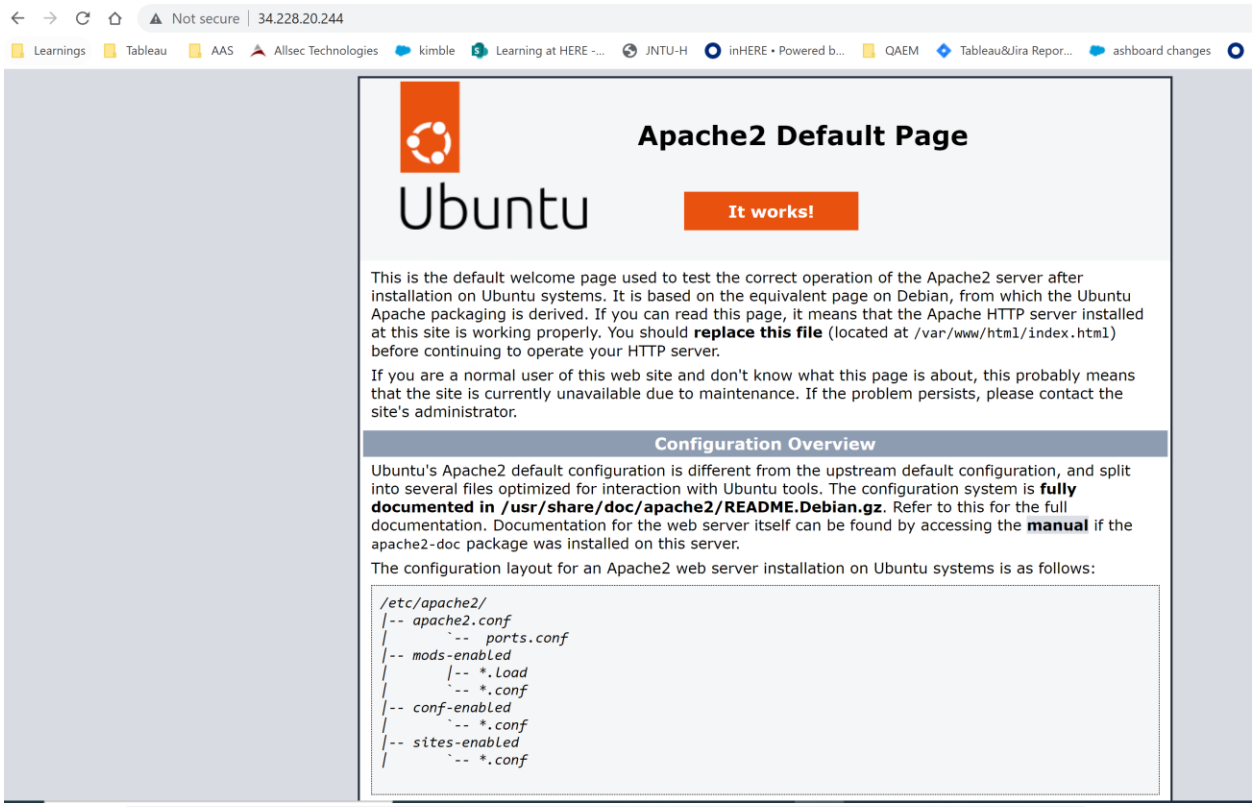
4. AMI Creation:

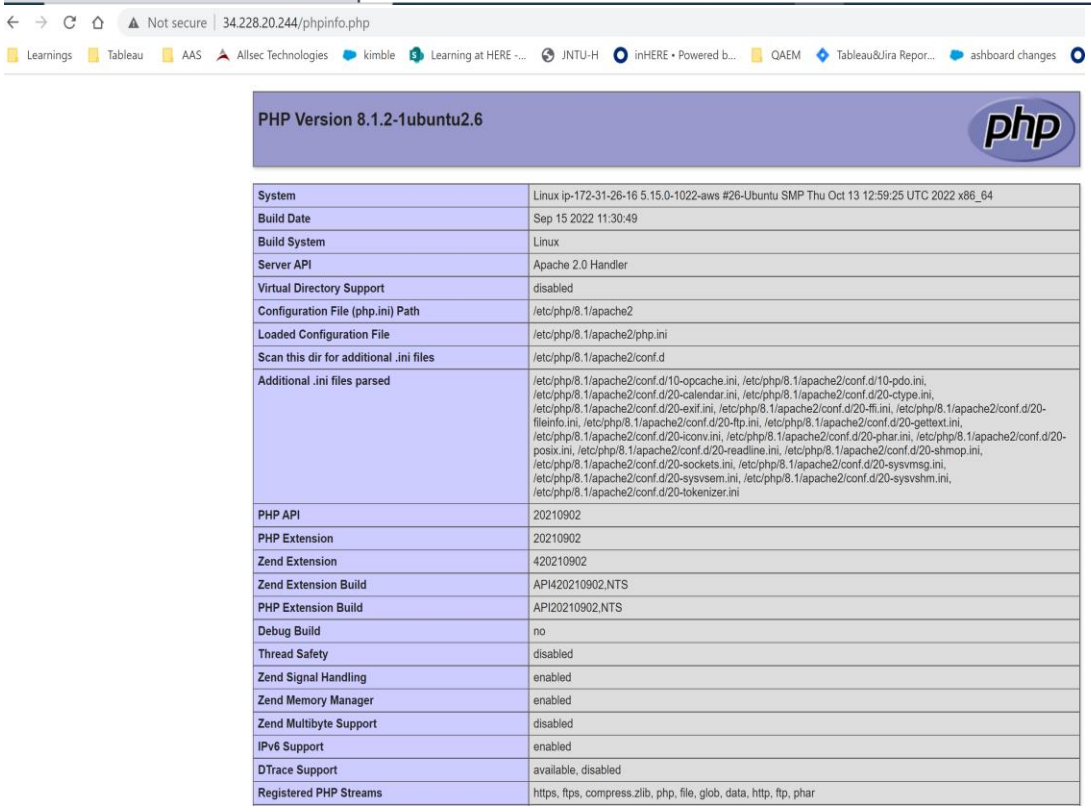
- Created ubuntu instance with t2.micro
- Installed lamp for testing on ubuntu instance
- My observation while running the command **“sudo apt upgrade”**



- Rebooted instance manually on aws console
- Confirmation of apache 2

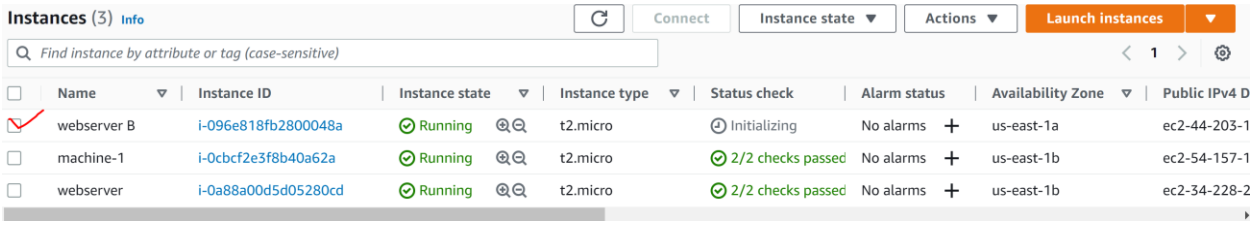
```
root@ip-172-31-26-16:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2022-10-28 07:38:27 UTC; 6s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1614 (apache2)
      Tasks: 55 (limit: 1143)
     Memory: 4.8M
        CPU: 29ms
     CGroup: /system.slice/apache2.service
             ├─1614 /usr/sbin/apache2 -k start
             ├─1616 /usr/sbin/apache2 -k start
             └─1617 /usr/sbin/apache2 -k start
```





System	Linux ip-172-31-26-16 5.15.0-1022-aws #26-Ubuntu SMP Thu Oct 13 12:59:25 UTC 2022 x86_64
Build Date	Sep 15 2022 11:30:49
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar

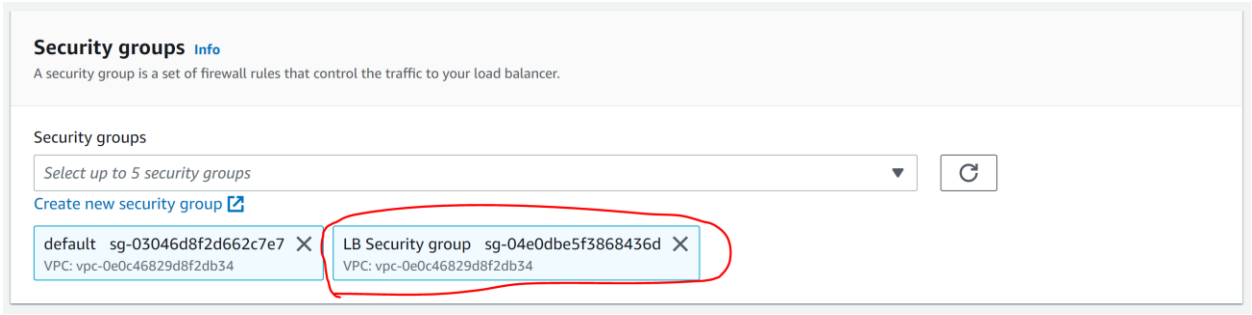
- Right click in instance →Image and template→create image
- In AMI section right click on image→ Launch instance from AMI



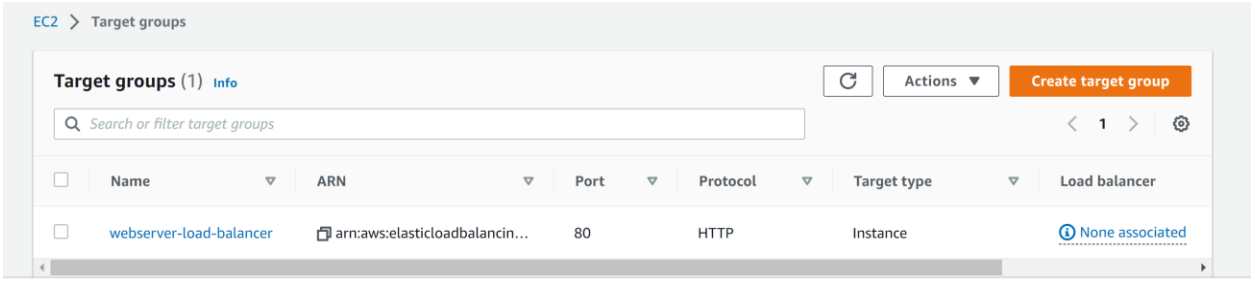
**Observation: however instance created from AMI, inbound rule is not covered as reference ec2- instance security group I manually allotted inbound rule for newer instance on http

5. Load Balancer creation:

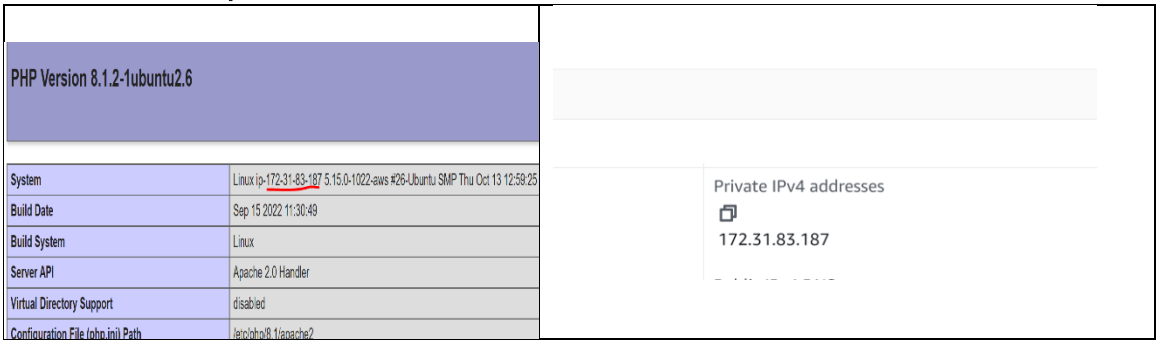
Under EC2 section click on Load balancer
Created custmise security group



Created target group on instances



Load balancer output



System	Linux ip-172-31-26-16 5.1
Build Date	Sep 15 2022 11:30:49
Build System	Linux
Server API	Apache 2.0 Handler