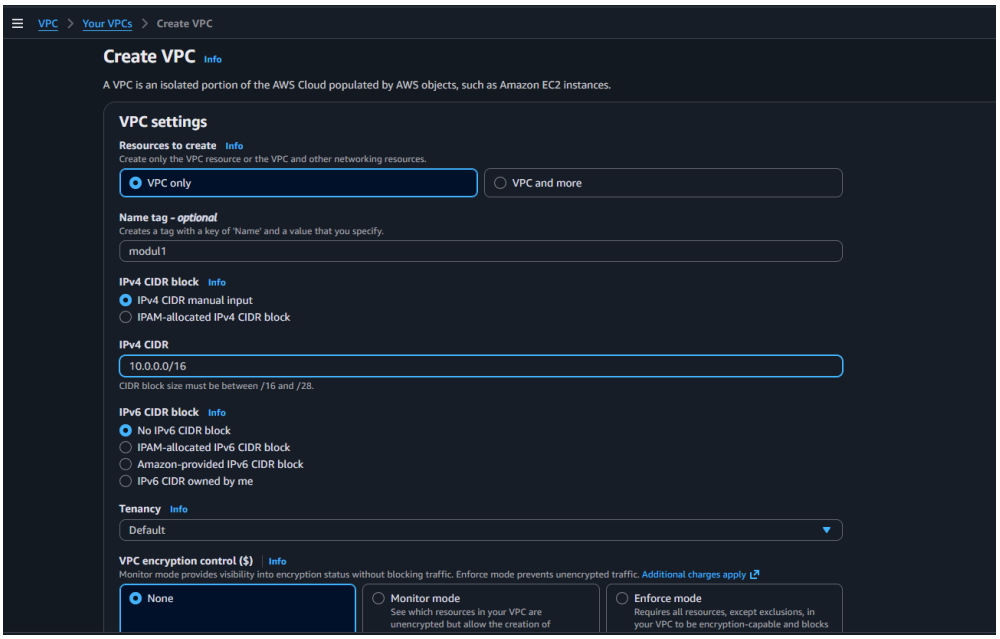
UJI COBA LKS CLOUD COMPUTING

Apa itu VPC : https://www.ibm.com/id-id/think/topics/vpc

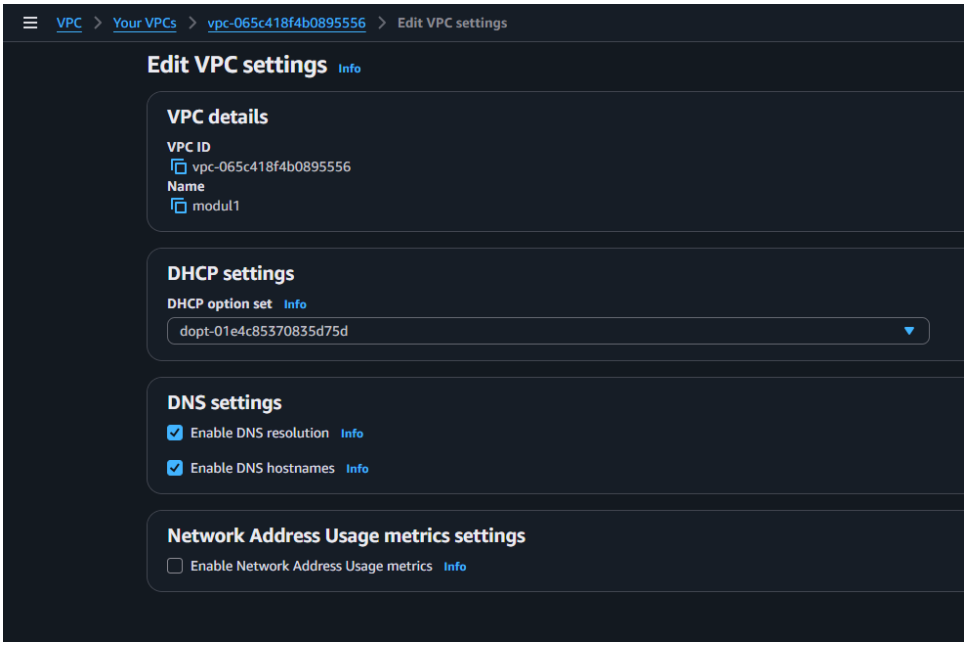
Apa itu EC2 : https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

Latihan kali ini kita akan membuat aplikasi static berbasis ec2 dan vpc kamu bisa ikuti Langkah Langkah berikut :

1. Buatlah vpc sebagai berikut kamu bisa kasih nama bebas contoh di sini saya membuat dengan nama modul1 dan untuk ipv4 nya 10.0.0.0/16

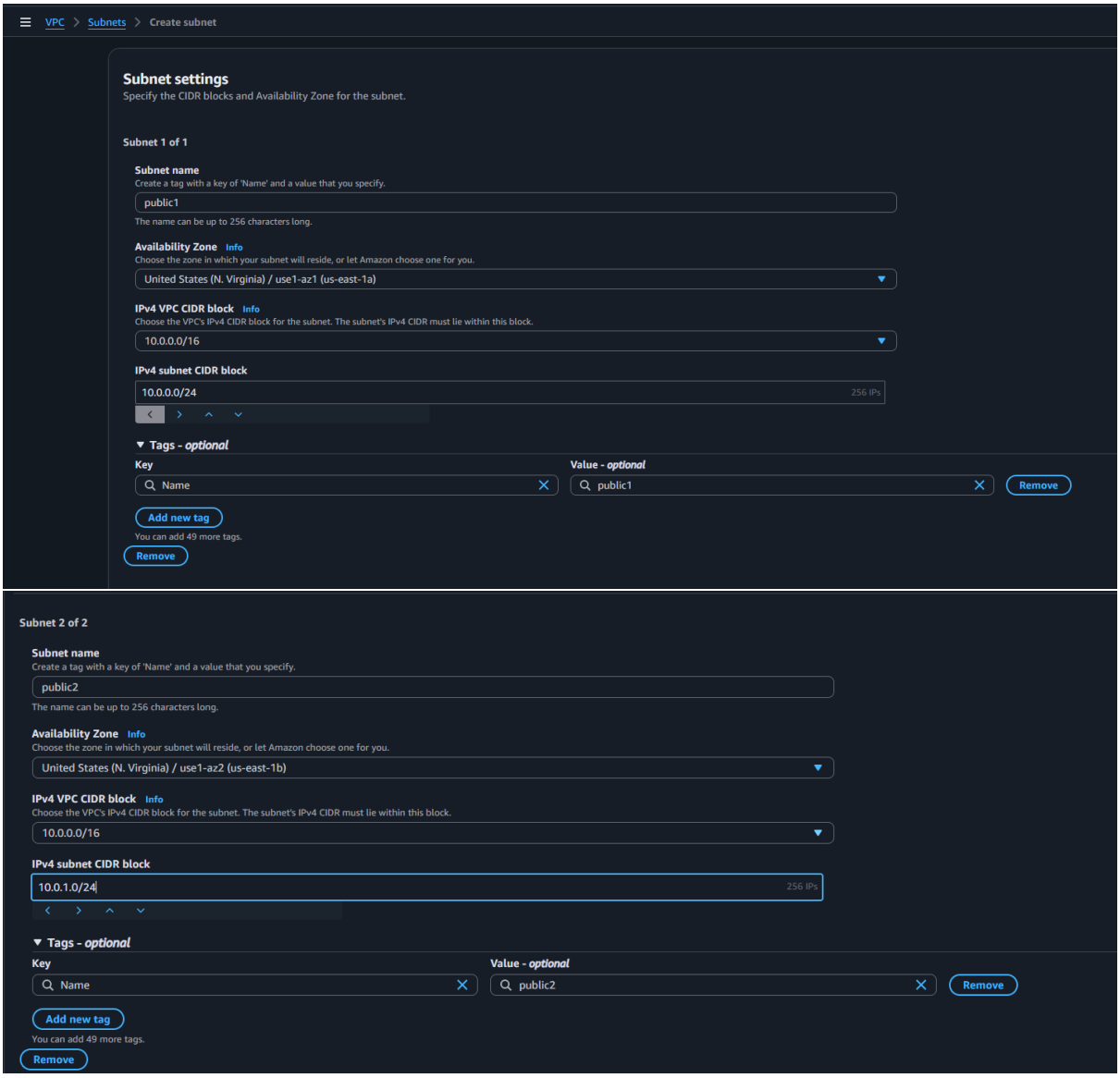


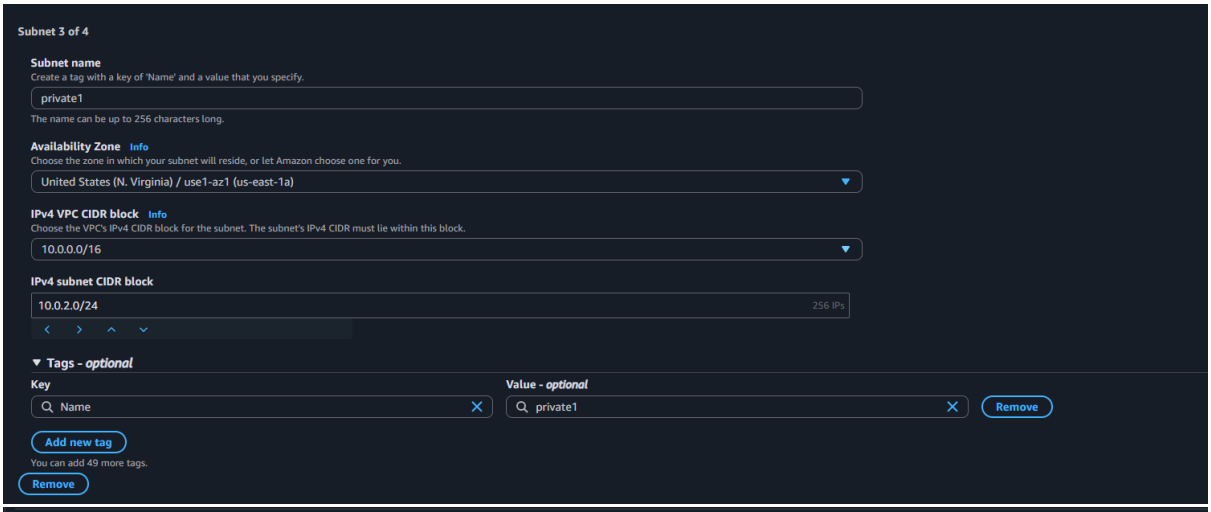
2. Jika sudah setting vpc untuk enable dns hostname seperti berikut :

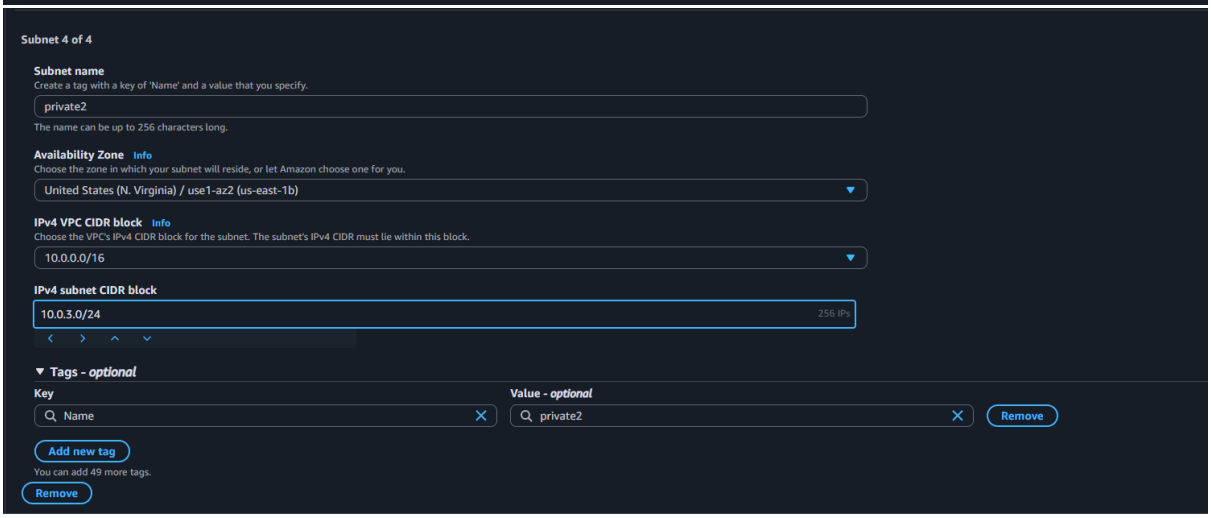


3. Buat 2 subnet public dan 2 subnet private dengan ketentuan ip sebagai berikut
   Public1 : 10.0.0.0/24, az : us-east-1a
   Public2 : 10.0.1.0/24, az : us-east-1b
   Private1 : 10.0.2.0/24, az : us-east-1a
   Private2 : 10.0.3.0/24, az : us-east-1b
   Kurang lebih jika di interface nya sebagai berikut :

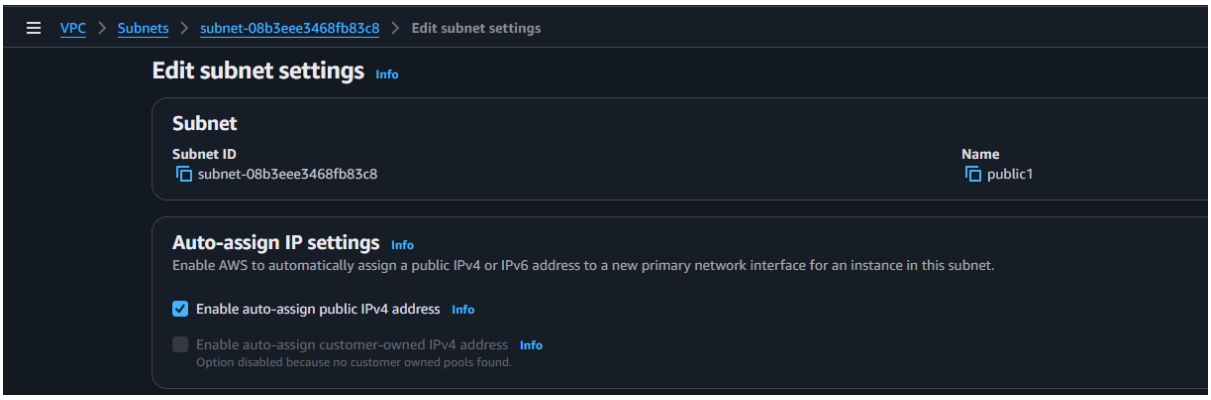
   Catatan : pastikan semua subnet berada di vpc yang tadi dibuat sebelumnya



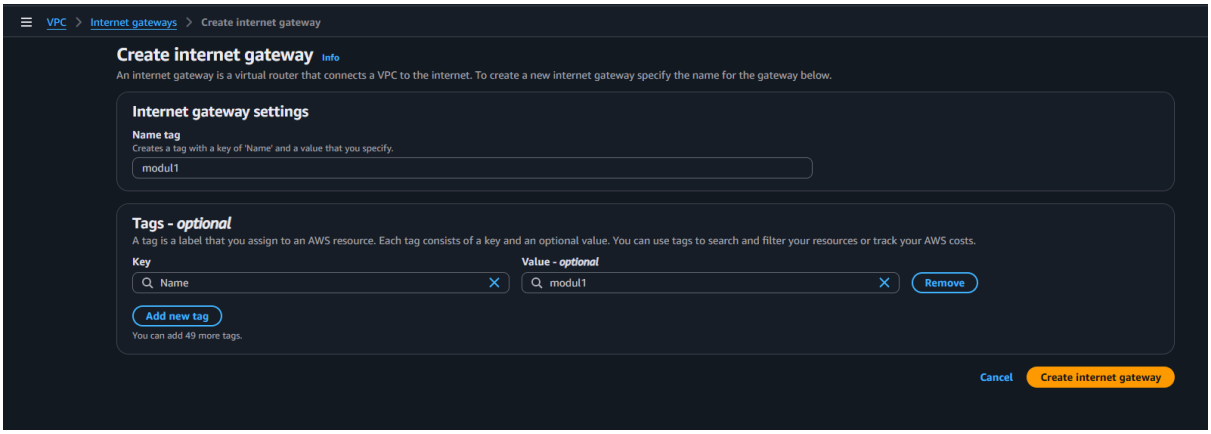




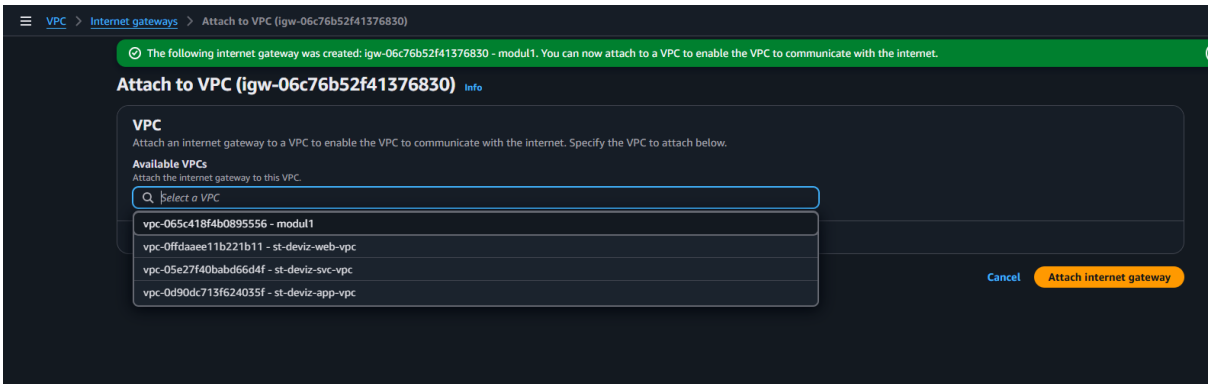
4. Jika sudah setting kedua subnet public yang tadi dibuat agar enable auto assign ipv4 seperti berikut :



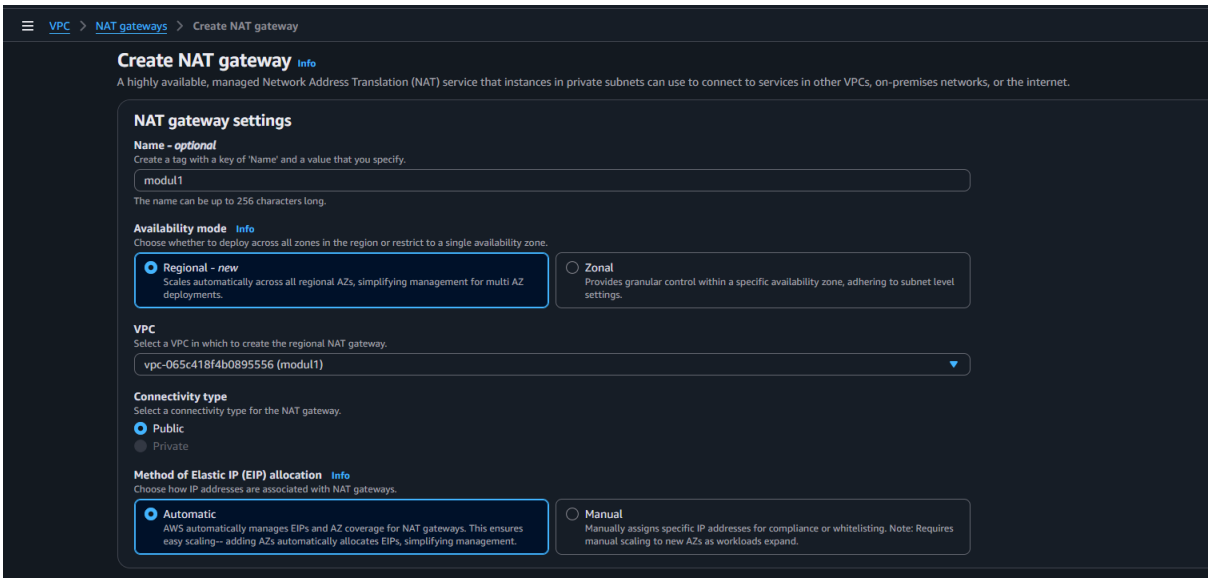
5. Lanjut step ketiga yaitu internet gateway atau gerbang menuju internet buatlah seperti berikut :



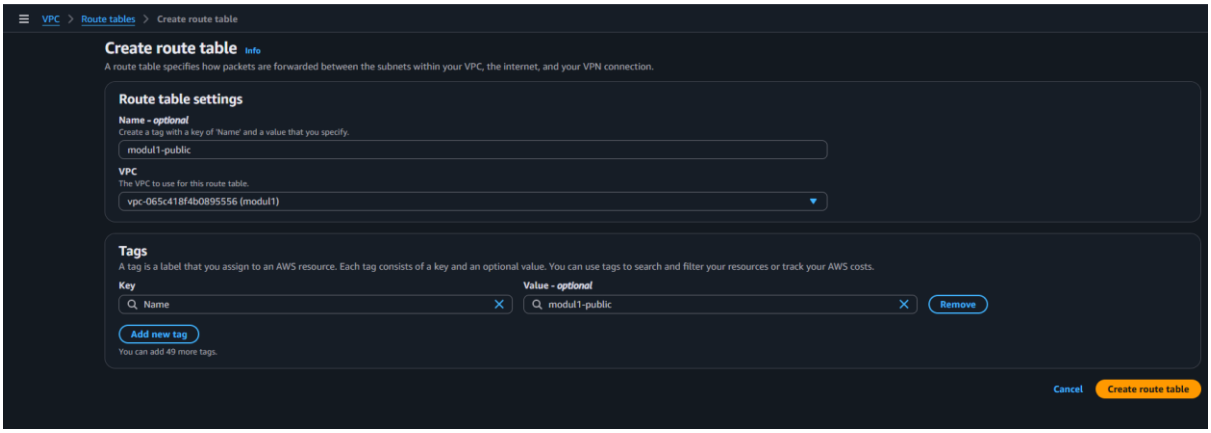
6. Atach igw ke vpc yang kita buat seperti berikut :



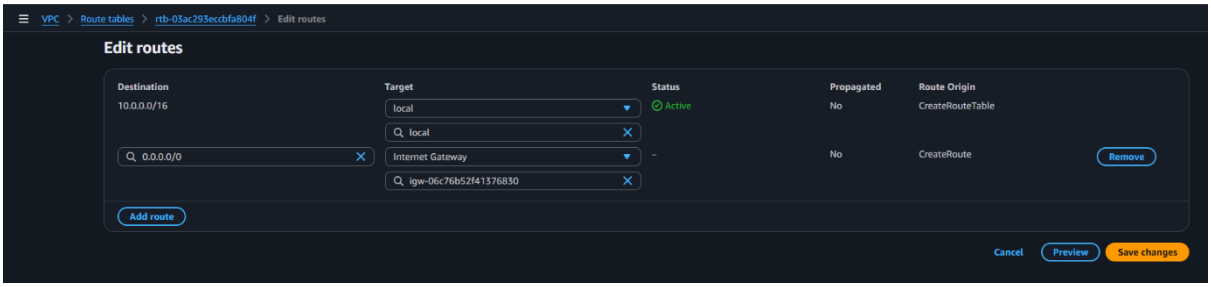
7. Step ke 4 kita buat nat gateway agar subnet private dapat terhubung ke internet secara private dari vpc sebelumnya dan dengan ketentuan seperti di gambar :



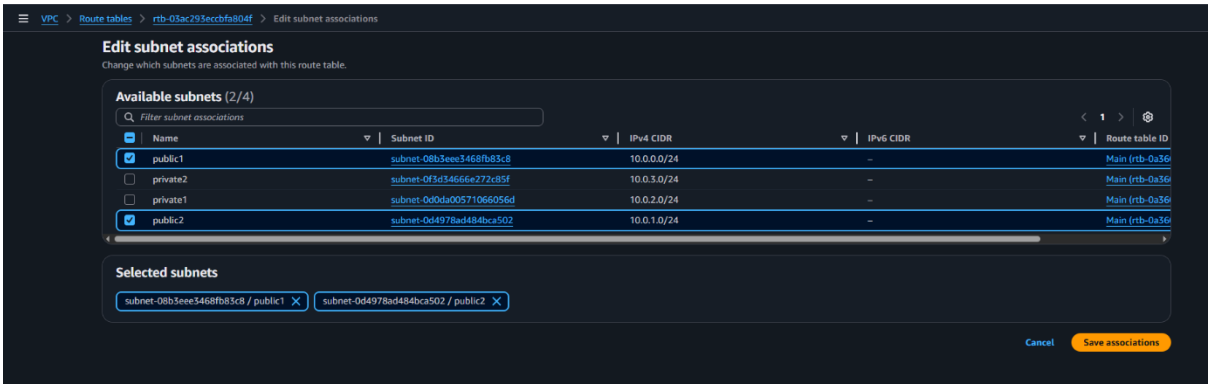
8. Step ke 5 kita buat route table agar jaringan yang kita buat tadi bisa terhubung ke internet
9. Pertama buat route table untuk public seperti berikut dan pastikan vpc nya yang tadi dibuat



10. Edit bagian route agar terhubung je internet gateway seperti berikut :



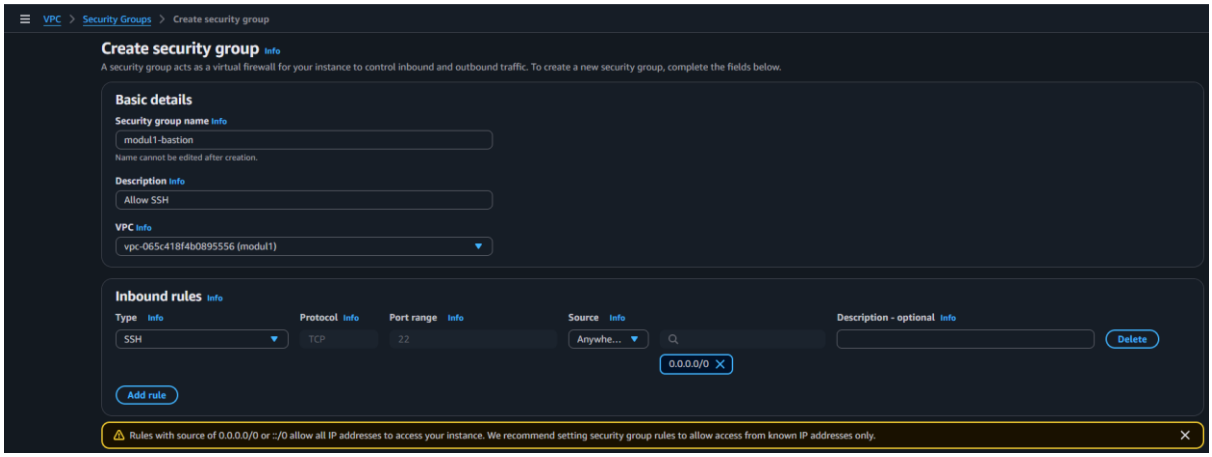
11. Tambahkan subnet association seperti berikut :



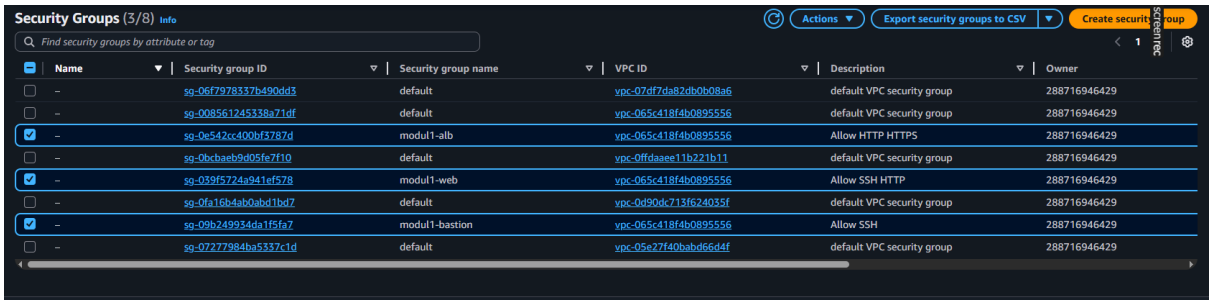
12. Jika sudah lakukan hal tadi secara berulang (bagian step ke 5). untuk bagian yang private cuman perbedaanya di bagian routes dan subnet association.
13. Untuk route table private routes nya arahin ke nat gateway, dan subnet association nya centang bagian yang private saja
14. Jika kedua route table sudah terbuat kita lanjut ke step ke 6 yaitu security group atau bisa disebut sebuah aturan dalam jaringan agar tidak sembarangan port dapat mengakses jaringan kita untuk penjelasan langsungnya kalian bis acari di google agar lebih paham
15. Buatlah security group dengan ketentuan berikut

Security Group	Rule	Protocol.	Port	Source
modul1-bastion	Allow SSH	SSH	22	0.0.0.0/0
modul1-alb	Allow HTTP HTTPS	HTTP	80	0.0.0.0/0
		HTTPS	443	0.0.0.0/0
modul1-web	Allow SSH HTTP	SSH	22	modul1-bastion
		HTTP	80	modul1-alb

Aku kasih contoh 1 implementasi dalam pembuatan security group agar kalian tidak bingung. Berikut contohnya

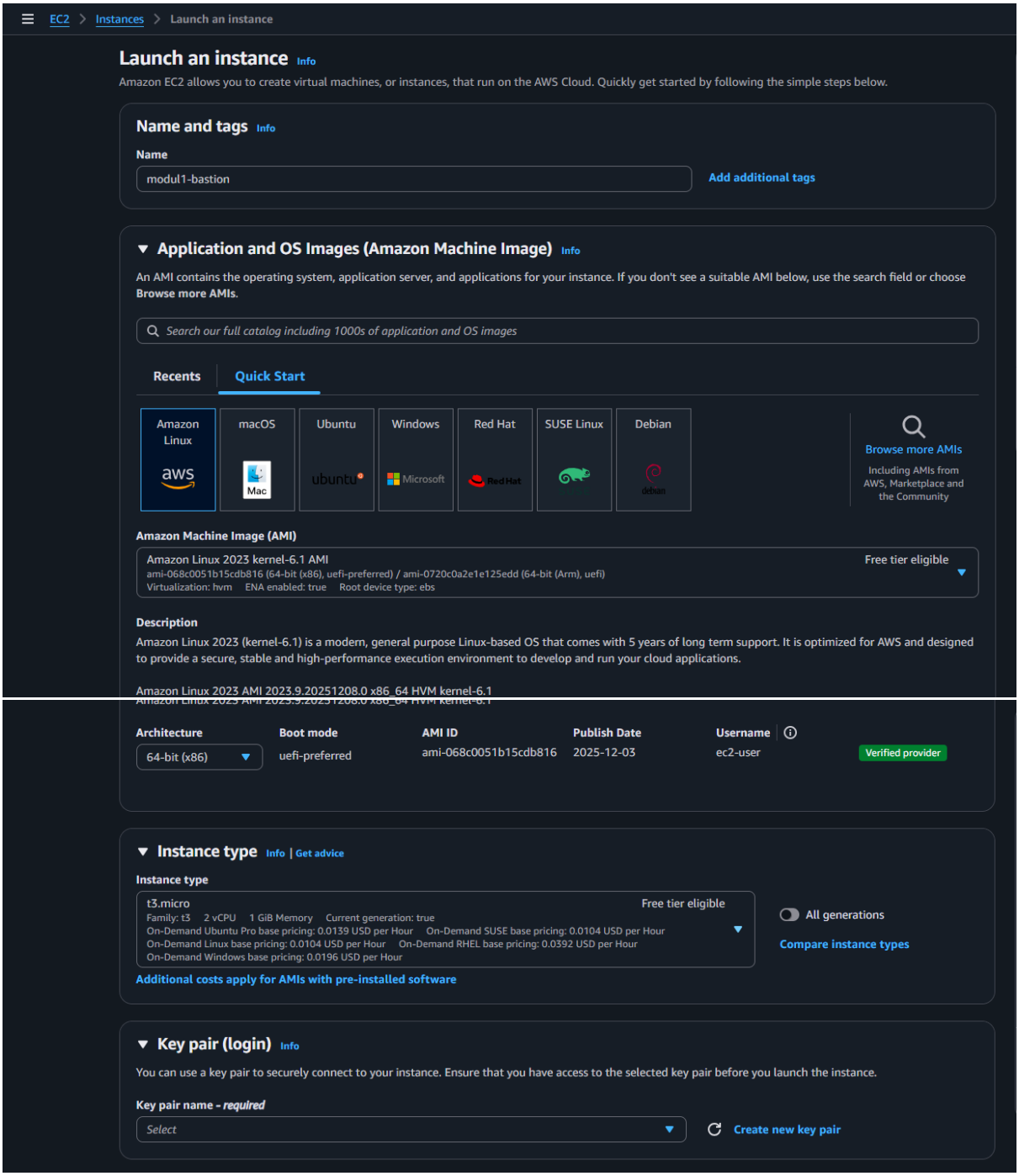


16. Disana ada beberapa segmen pertama untuk nama kamu sesuaikan di table
17. Untuk description sesuaikannya dengan Rule pada table
18. Untuk bagian vpc sudah jelas pake vpc yang kita buat tadi
19. Dan tambahkan inbound sesuai ketentuan di table contoh untuk sg modul1-bastion dia inbound nya SSH Port 22 dan source nya 0.0.0.0/0 (Anywhere)
20. Lakukan hal yang sama pada bagian security group yang lain dengan ketentuan seperti di table
21. Nantinya kita akan punya 3 sg seperti berikut

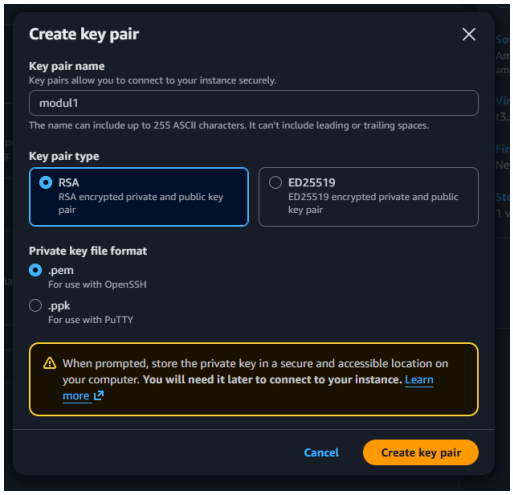


22. Jika sudah semua security group dibuat artinya untuk bagian vpc kita sudah selesai

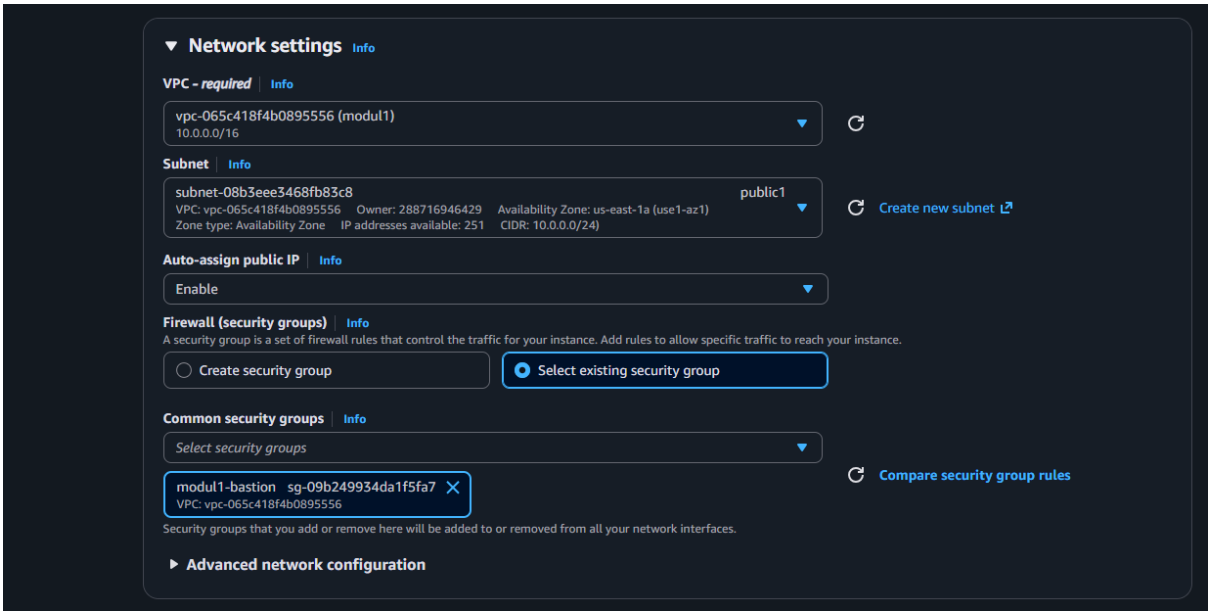
23. Lanjut ke bagian dua yaitu EC2
24. Buat ec2 bastion dengan ketentuan berikut :



Pada bagian keypai create new key pair seperti berikut :



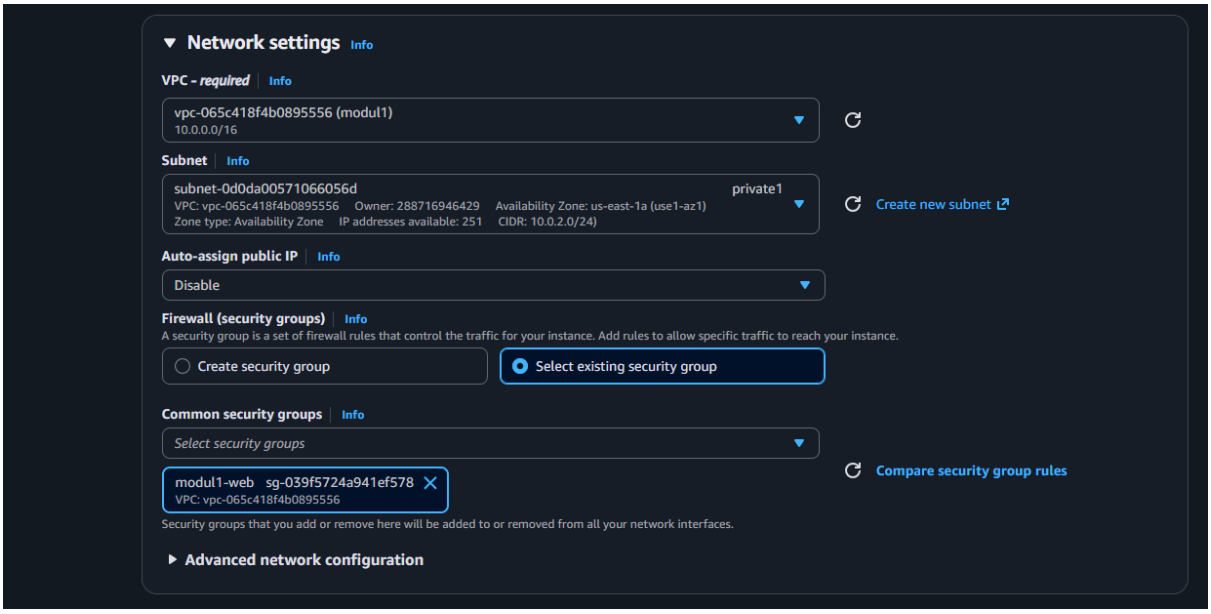
25. Lanjut bagian network setting sebagai berikut :



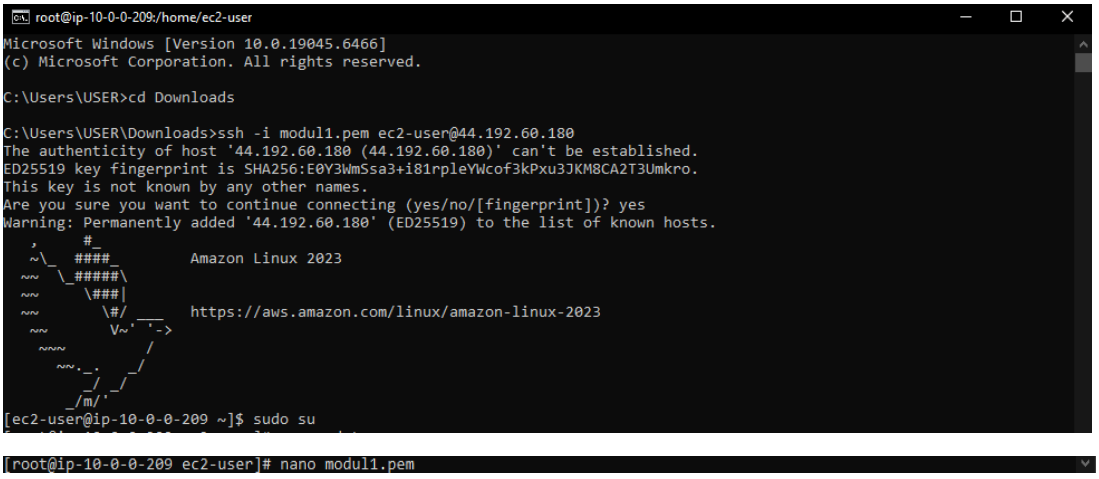
Sesuaikan saja setting nya seperti di gambar kemudian launch

26. Buat 1 ec2 lagi dengan nama modul1 untuk pembuatannya masih sama seperti sebelumnya hanya beda di bagian network setting berikut settingan network untuk ec2 ke 2

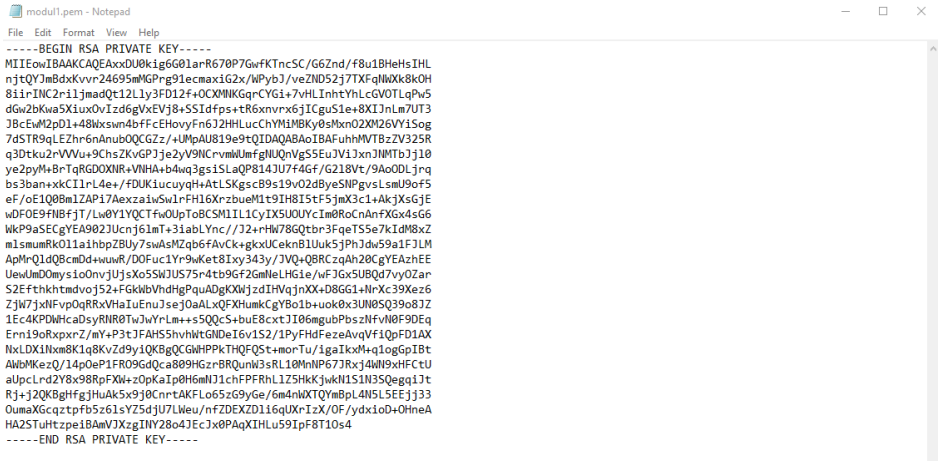
Catatan : pastikan menggunakan keypair yang sama seperti sebelumnya yang di buat tadi



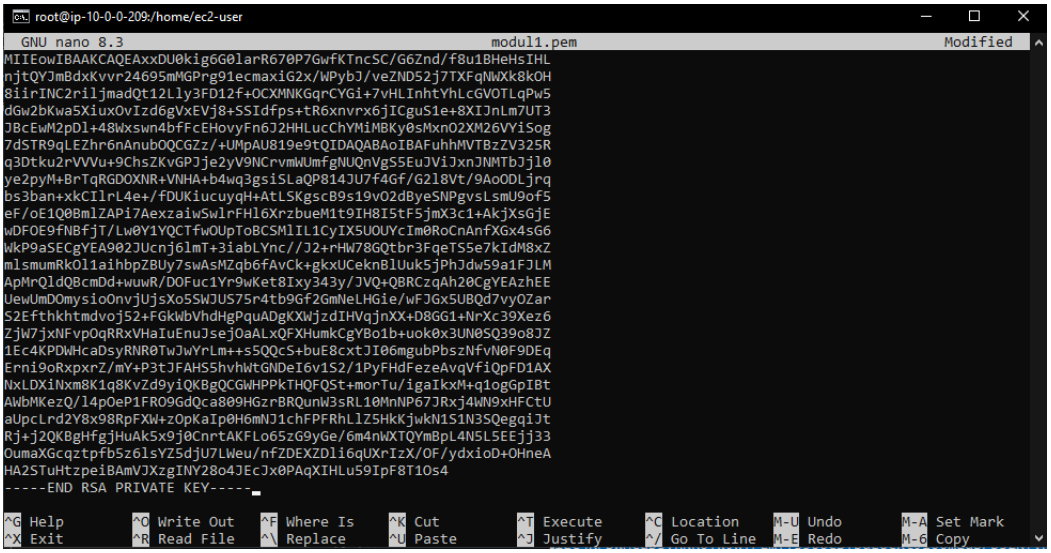
27. Oke next ke step selanjutnya dan inilah step yang paling utama yaitu instalasi server nya pastikan pahami dengan baik
28. Jika sudah ke 2 ec2 dibuat login di cmd sebagai berikut :
29. Buka ec2 yang modul1-bastion kemudia copy ip public nya
30. Jika sudah buka cmd masing masing lakukan instalasi seperti berikut :



31. Buka keypair yang tadi di download menggunakan notepad seperti berikut



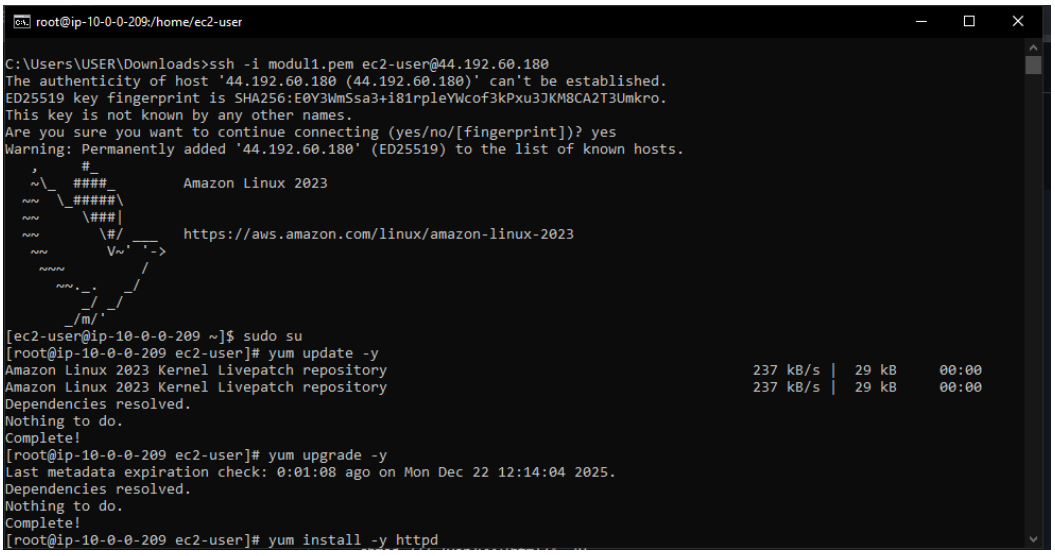
Lalu paste di nano tadi seperti ini :

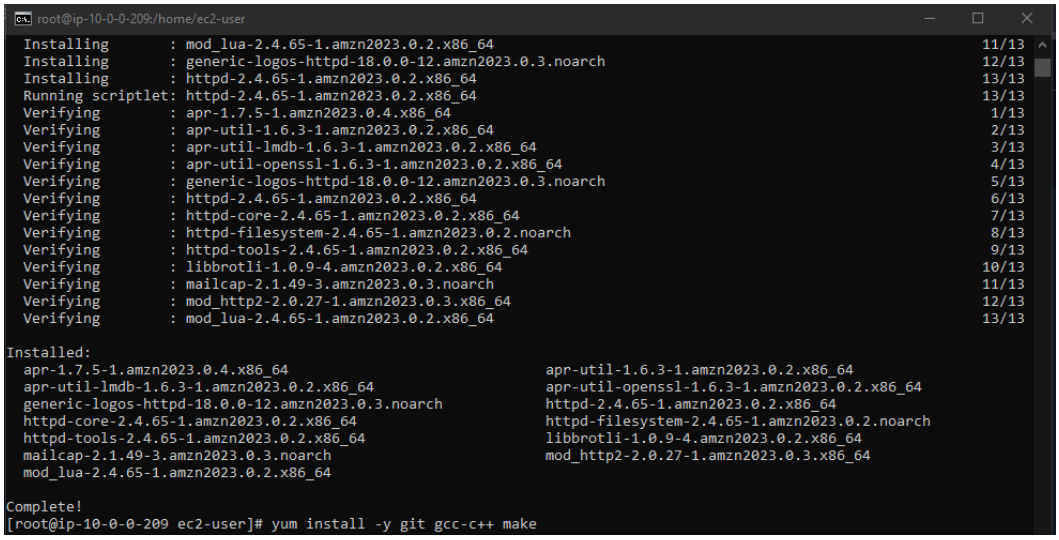


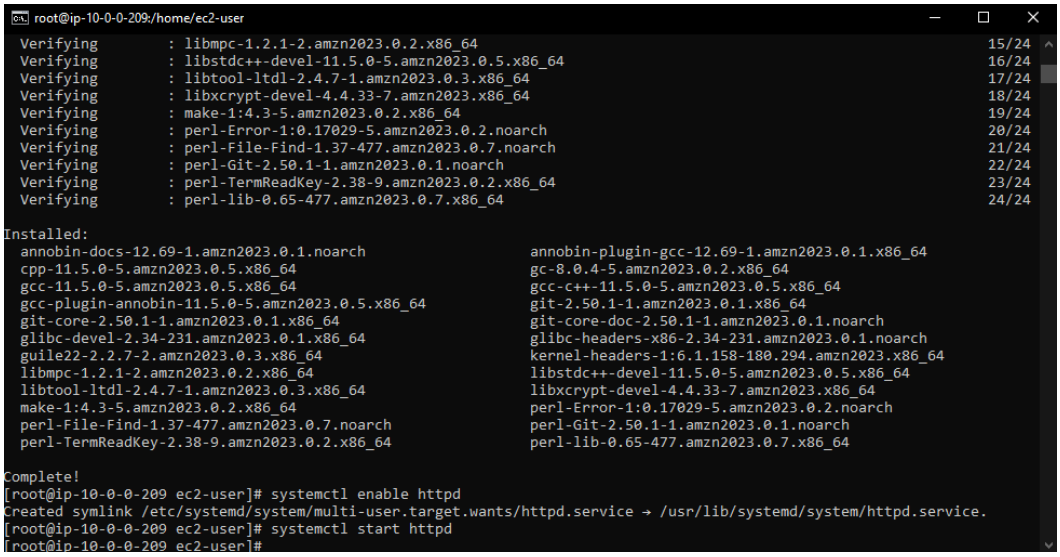
Jika sudah CTRL + X, Y, ENTER

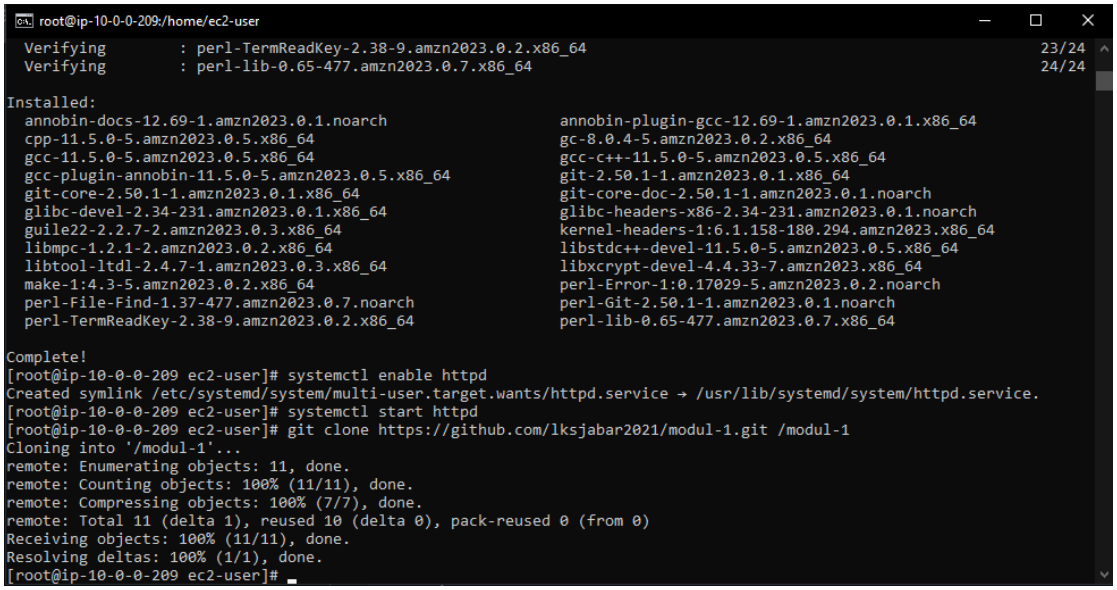
32. Lakukan chmod seperti berikut

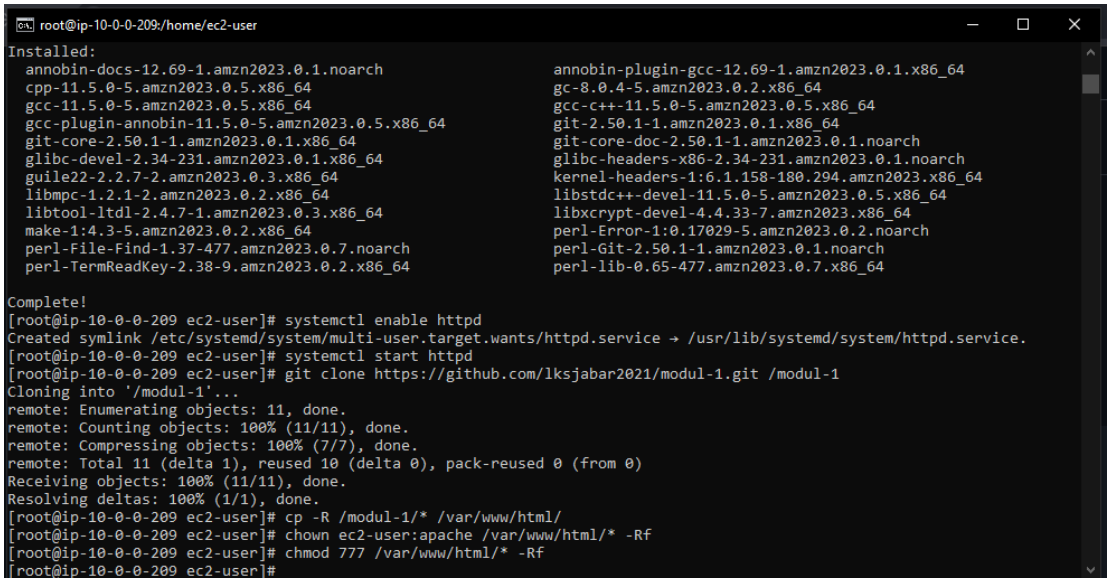
33. Jika sudah lakukan ssh lagi tetapi pakai ip private yang ec2 kedua seperti berikut

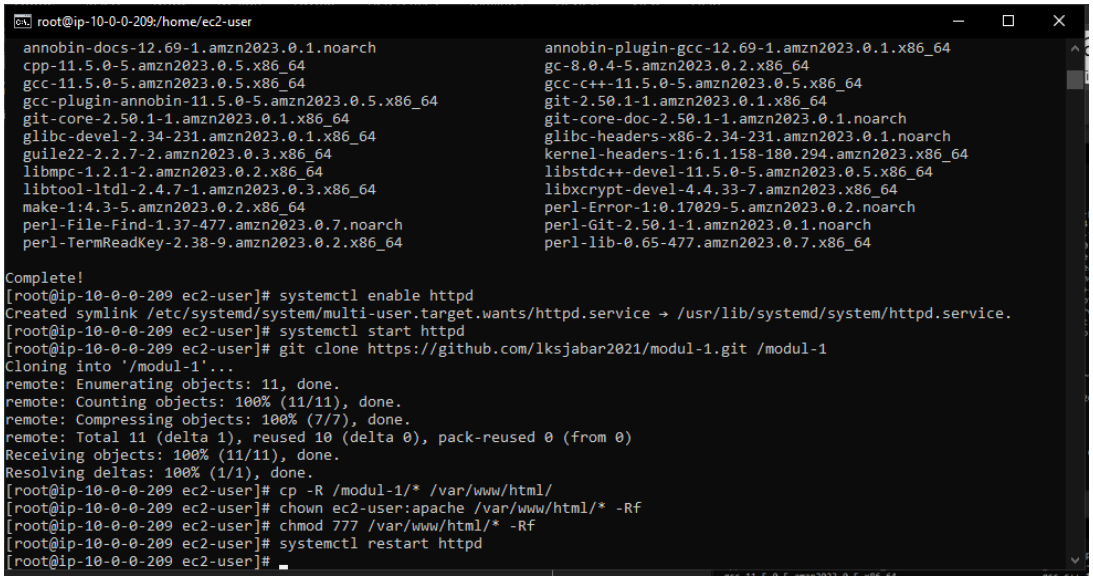



root@ip-10-0-0-209:/home/ec2-user
Installing : mod_lua-2.4.65-1.amzn2023.0.2.x86_64 11/13
Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 12/13
Installing : httpd-2.4.65-1.amzn2023.0.2.x86_64 13/13
Running scriptlet: httpd-2.4.65-1.amzn2023.0.2.x86_64 13/13
Verifying : apr-1.7.5-1.amzn2023.0.4.x86_64 1/13
Verifying : apr-util-1.6.3-1.amzn2023.0.2.x86_64 2/13
Verifying : apr-util-lmdb-1.6.3-1.amzn2023.0.2.x86_64 3/13
Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.2.x86_64 4/13
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 5/13
Verifying : httpd-2.4.65-1.amzn2023.0.2.x86_64 6/13
Verifying : httpd-core-2.4.65-1.amzn2023.0.2.x86_64 7/13
Verifying : httpd-filesystem-2.4.65-1.amzn2023.0.2.noarch 8/13
Verifying : httpd-tools-2.4.65-1.amzn2023.0.2.x86_64 9/13
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 10/13
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 11/13
Verifying : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 12/13
Verifying : mod_lua-2.4.65-1.amzn2023.0.2.x86_64 13/13
Installed:
apr-1.7.5-1.amzn2023.0.4.x86_64
apr-util-1.6.3-1.amzn2023.0.2.x86_64
apr-util-lmdb-1.6.3-1.amzn2023.0.2.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.2.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.65-1.amzn2023.0.2.x86_64
httpd-core-2.4.65-1.amzn2023.0.2.x86_64
httpd-filesystem-2.4.65-1.amzn2023.0.2.noarch
httpd-tools-2.4.65-1.amzn2023.0.2.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_http2-2.0.27-1.amzn2023.0.3.x86_64
mod_lua-2.4.65-1.amzn2023.0.2.x86_64
Complete!
[root@ip-10-0-0-209 ec2-user]# yum install -y git gcc-c++ make


root@ip-10-0-0-209:/home/ec2-user
Verifying : libmpc-1.2.1-2.amzn2023.0.2.x86_64 15/24
Verifying : libstdc++-devel-11.5.0-5.amzn2023.0.5.x86_64 16/24
Verifying : libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64 17/24
Verifying : libxcrypt-devel-4.4.33-7.amzn2023.x86_64 18/24
Verifying : make-1:4.3-5.amzn2023.0.2.x86_64 19/24
Verifying : perl-Error-1:0.17029-5.amzn2023.0.2.noarch 20/24
Verifying : perl-File-Find-1.37-477.amzn2023.0.7.noarch 21/24
Verifying : perl-Git-2.50.1-1.amzn2023.0.1.noarch 22/24
Verifying : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 23/24
Verifying : perl-lib-0.65-477.amzn2023.0.7.x86_64 24/24
Installed:
annobin-docs-12.69-1.amzn2023.0.1.noarch
annobin-plugin-gcc-12.69-1.amzn2023.0.1.x86_64
cpp-11.5.0-5.amzn2023.0.5.x86_64
gc-8.0.4-5.amzn2023.0.2.x86_64
gcc-11.5.0-5.amzn2023.0.5.x86_64
gcc-c++-11.5.0-5.amzn2023.0.5.x86_64
gcc-plugin-annobin-11.5.0-5.amzn2023.0.5.x86_64
git-2.50.1-1.amzn2023.0.1.x86_64
git-core-2.50.1-1.amzn2023.0.1.x86_64
git-core-doc-2.50.1-1.amzn2023.0.1.noarch
glibc-devel-2.34-231.amzn2023.0.1.x86_64
glibc-headers-x86-2.34-231.amzn2023.0.1.noarch
guile22-2.2.7-2.amzn2023.0.3.x86_64
kernel-headers-1:6.1.158-180.294.amzn2023.x86_64
libmpc-1.2.1-2.amzn2023.0.2.x86_64
libstdc++-devel-11.5.0-5.amzn2023.0.5.x86_64
libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64
libxcrypt-devel-4.4.33-7.amzn2023.x86_64
make-1:4.3-5.amzn2023.0.2.x86_64
perl-Error-1:0.17029-5.amzn2023.0.2.noarch
perl-File-Find-1.37-477.amzn2023.0.7.noarch
perl-Git-2.50.1-1.amzn2023.0.1.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64
perl-lib-0.65-477.amzn2023.0.7.x86_64
Complete!
[root@ip-10-0-0-209 ec2-user]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-10-0-0-209 ec2-user]# systemctl start httpd
[root@ip-10-0-0-209 ec2-user]#


root@ip-10-0-0-209:/home/ec2-user
Verifying : perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64 23/24
Verifying : perl-lib-0.65-477.amzn2023.0.7.x86_64 24/24
Installed:
annobin-docs-12.69-1.amzn2023.0.1.noarch
annobin-plugin-gcc-12.69-1.amzn2023.0.1.x86_64
cpp-11.5.0-5.amzn2023.0.5.x86_64
gc-8.0.4-5.amzn2023.0.2.x86_64
gcc-11.5.0-5.amzn2023.0.5.x86_64
gcc-c++-11.5.0-5.amzn2023.0.5.x86_64
gcc-plugin-annobin-11.5.0-5.amzn2023.0.5.x86_64
git-2.50.1-1.amzn2023.0.1.x86_64
git-core-2.50.1-1.amzn2023.0.1.x86_64
git-core-doc-2.50.1-1.amzn2023.0.1.noarch
glibc-devel-2.34-231.amzn2023.0.1.x86_64
glibc-headers-x86-2.34-231.amzn2023.0.1.noarch
guile22-2.2.7-2.amzn2023.0.3.x86_64
kernel-headers-1:6.1.158-180.294.amzn2023.x86_64
libmpc-1.2.1-2.amzn2023.0.2.x86_64
libstdc++-devel-11.5.0-5.amzn2023.0.5.x86_64
libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64
libxcrypt-devel-4.4.33-7.amzn2023.x86_64
make-1:4.3-5.amzn2023.0.2.x86_64
perl-Error-1:0.17029-5.amzn2023.0.2.noarch
perl-File-Find-1.37-477.amzn2023.0.7.noarch
perl-Git-2.50.1-1.amzn2023.0.1.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64
perl-lib-0.65-477.amzn2023.0.7.x86_64
Complete!
[root@ip-10-0-0-209 ec2-user]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-10-0-0-209 ec2-user]# systemctl start httpd
[root@ip-10-0-0-209 ec2-user]# git clone https://github.com/lksjabar2021/modul-1.git /modul-1
Cloning into '/modul-1'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 11 (delta 1), reused 10 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (1/1), done.
[root@ip-10-0-0-209 ec2-user]#





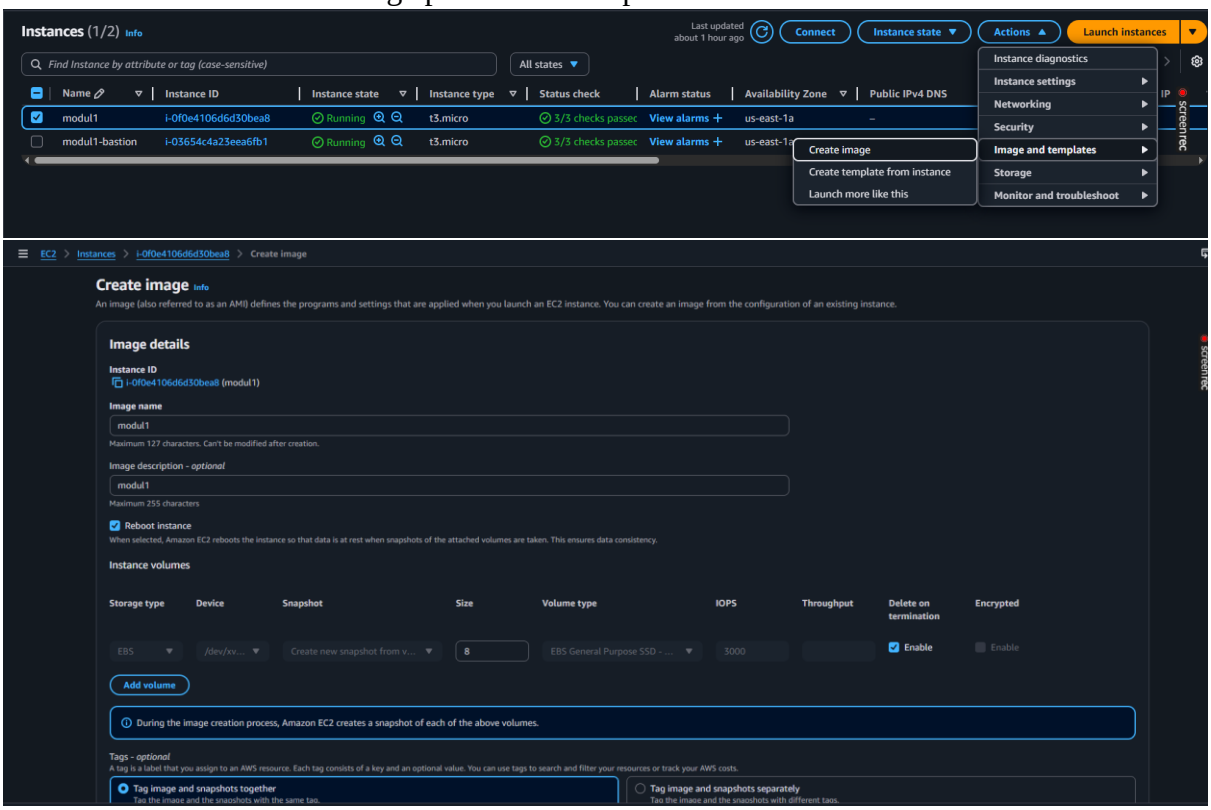
Jika kalian sudah menyelsaikannya seperti di gambar berarti instalasi berhasil

34. Jika masih belum menangkap syntaknya susunannya seperti berikut
35. Lakukan ssh ke ec2 bastion pake ip public
36. Copy keypair yang tadi di download ke ec2 bastion
37. Lakukan chmod kemudian lakukan ssh ke ec2 yang ke 2 (yang inti)
38. Dan lakukan susunan syntak berikut di ec2 inti

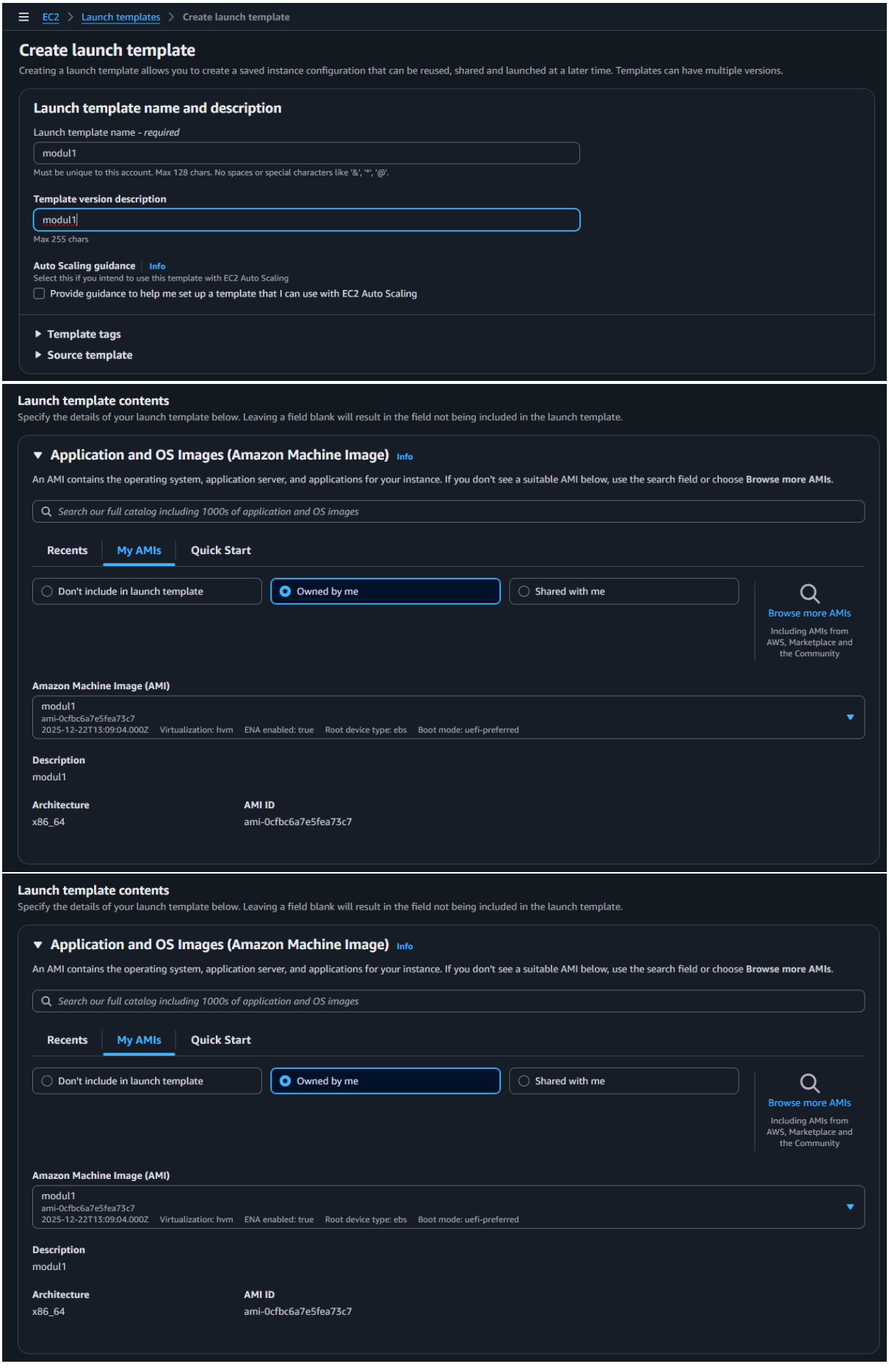
```
#!/bin/bash

# Update and Upgrade
yum update -y
yum upgrade -y

# Httpd
yum install -y httpd
```

```
yum install -y git gcc-c++ make
systemctl enable httpd
systemctl start httpd

# Clone project
git clone https://github.com/lksjabar2021/modul-1.git /modul-1
cp -R /modul-1/* /var/www/html/
chown ec2-user:apache /var/www/html/* -Rf
chmod 777 /var/www/html/* -Rf

# Restart Httpd
systemctl restart httpd
```

39. Jika sudah lakukan create image pada ec2 inti seperti berikut :

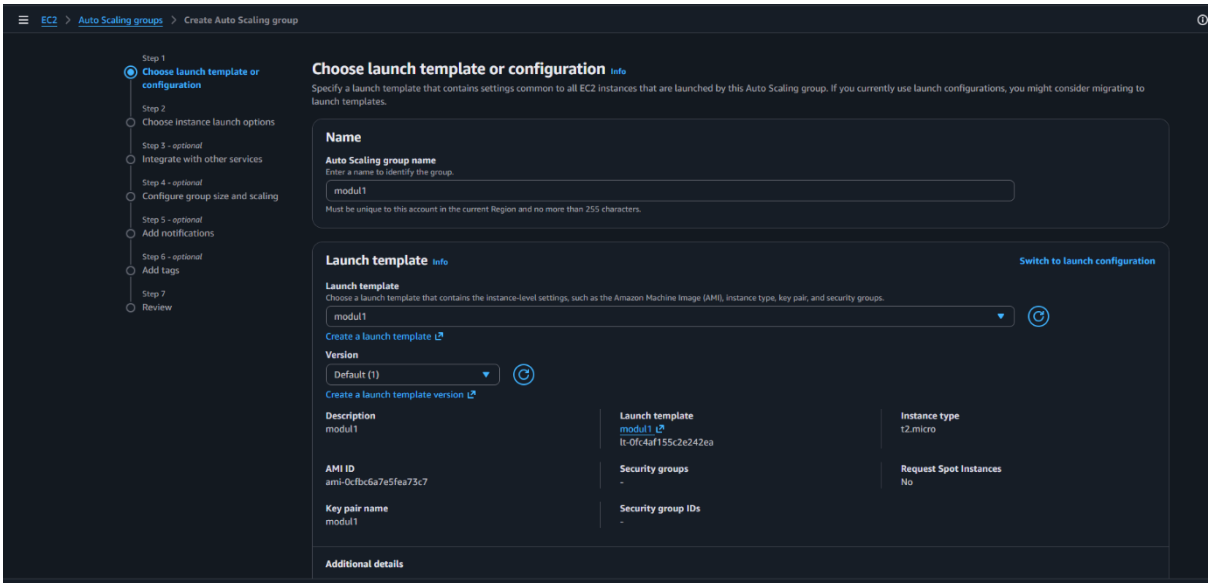


40. Lanjut step ke 2 yaitu membuat launch template dari ami yang di create tadi

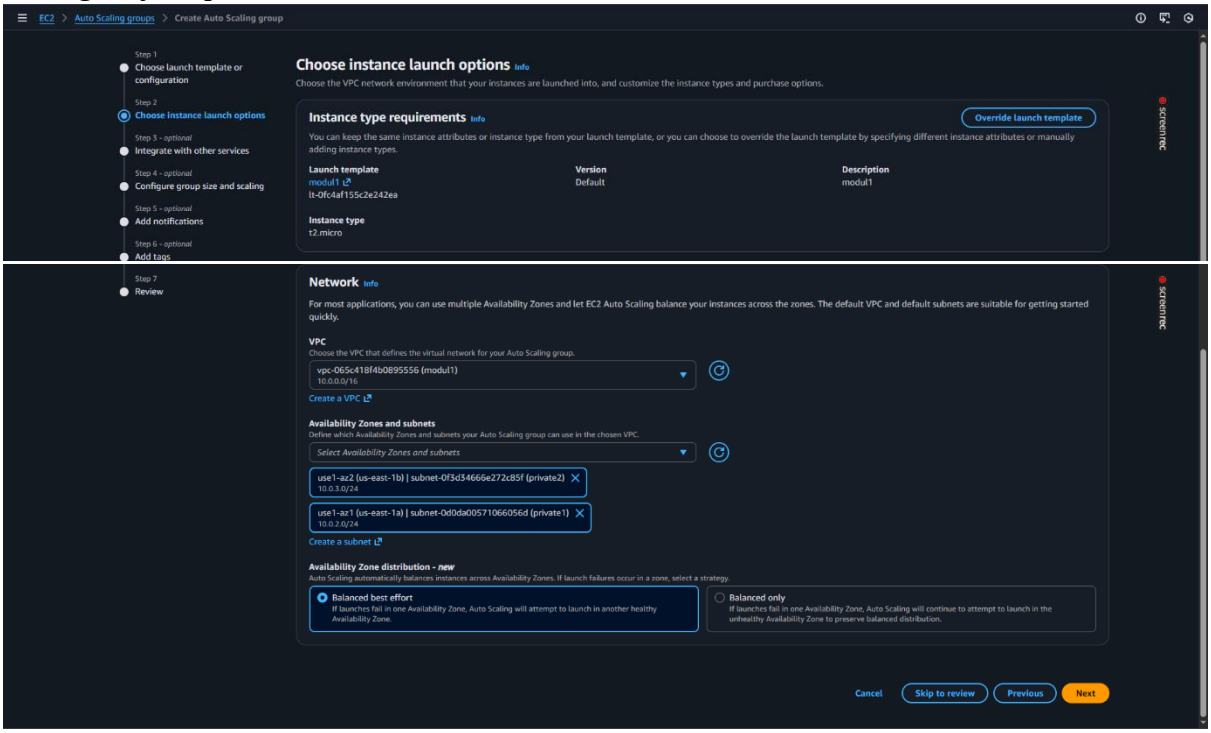

EC2 > Launch templates > Create launch template
Create launch template
Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.
Launch template name and description
Launch template name - required
modul1
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.
Template version description
modul1
Max 255 chars
Auto Scaling guidance | Info
Select this if you intend to use this template with EC2 Auto Scaling
Provide guidance to help me set up a template that I can use with EC2 Auto Scaling
Template tags
Source template
Launch template contents
Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.
Application and OS Images (Amazon Machine Image) Info
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose Browse more AMIs.
Search our full catalog including 1000s of application and OS images
Recents
My AMIs
Quick Start
Don't include in launch template
Owned by me
Shared with me
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
modul1
ami-0cfbc6a7e5fea73c7
2025-12-22T13:09:04.000Z Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred
Description
modul1
Architecture
x86_64
AMI ID
ami-0cfbc6a7e5fea73c7
Launch template contents
Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.
Application and OS Images (Amazon Machine Image) Info
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose Browse more AMIs.
Search our full catalog including 1000s of application and OS images
Recents
My AMIs
Quick Start
Don't include in launch template
Owned by me
Shared with me
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
modul1
ami-0cfbc6a7e5fea73c7
2025-12-22T13:09:04.000Z Virtualization: hvm ENA enabled: true Root device type: ebs Boot mode: uefi-preferred
Description
modul1
Architecture
x86_64
AMI ID
ami-0cfbc6a7e5fea73c7

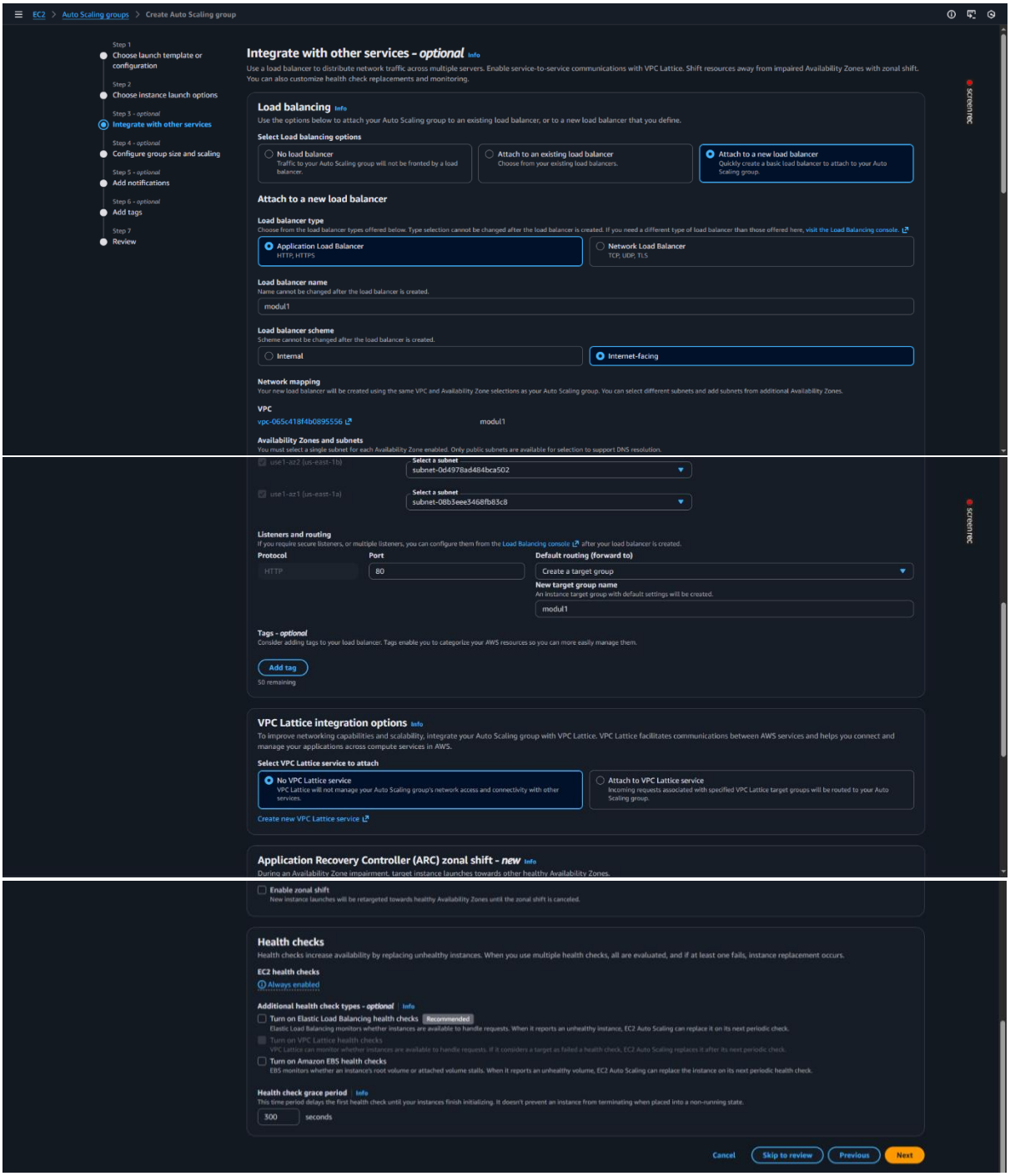
Untuk network setting biarkan default, hanya pada bagian security group pastikan menggunakan sg yang modul1-web jika sudah create

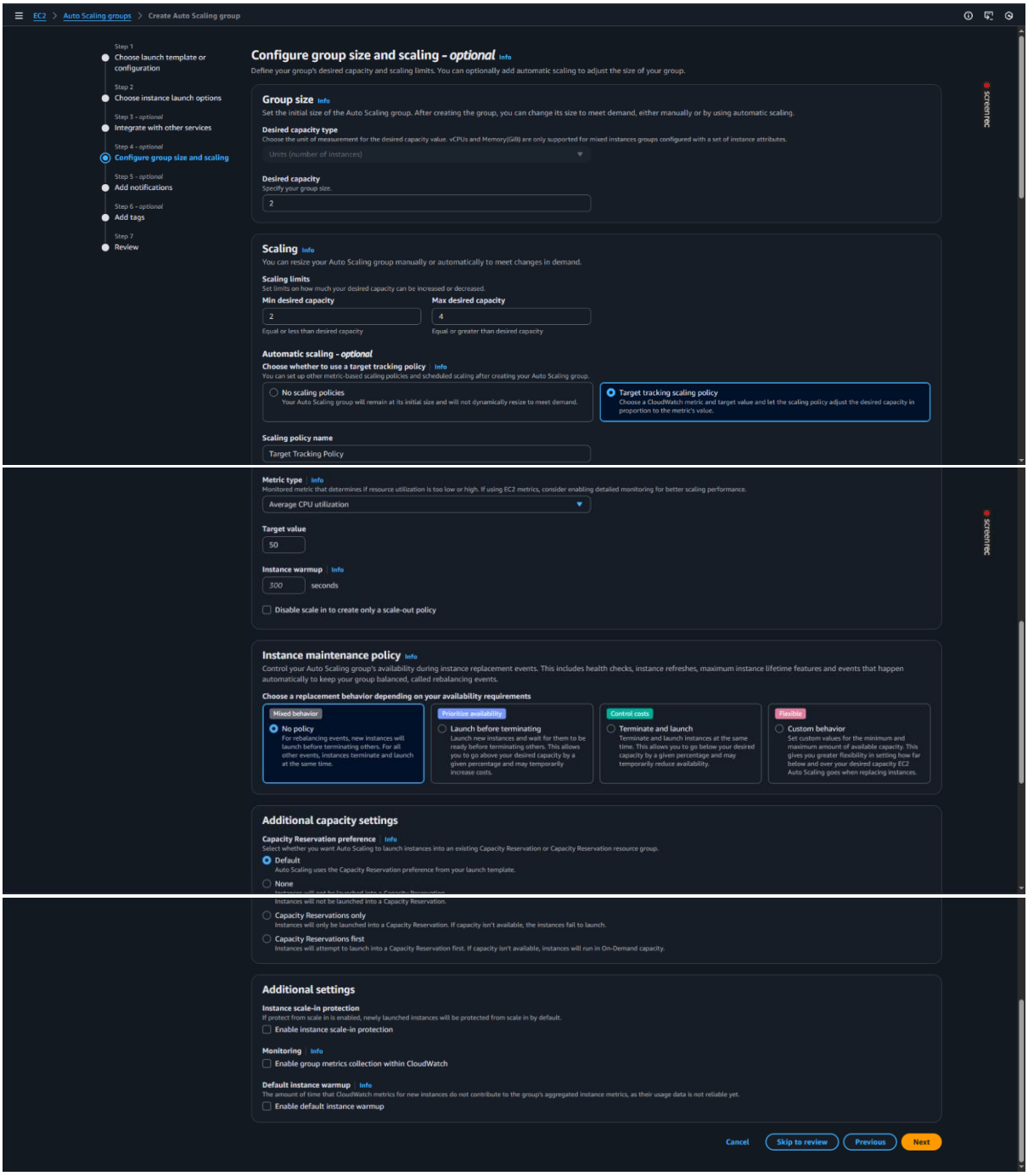
41. Lanjut step ke 3 yaitu membuat Auto Scalling Group seperti berikut :



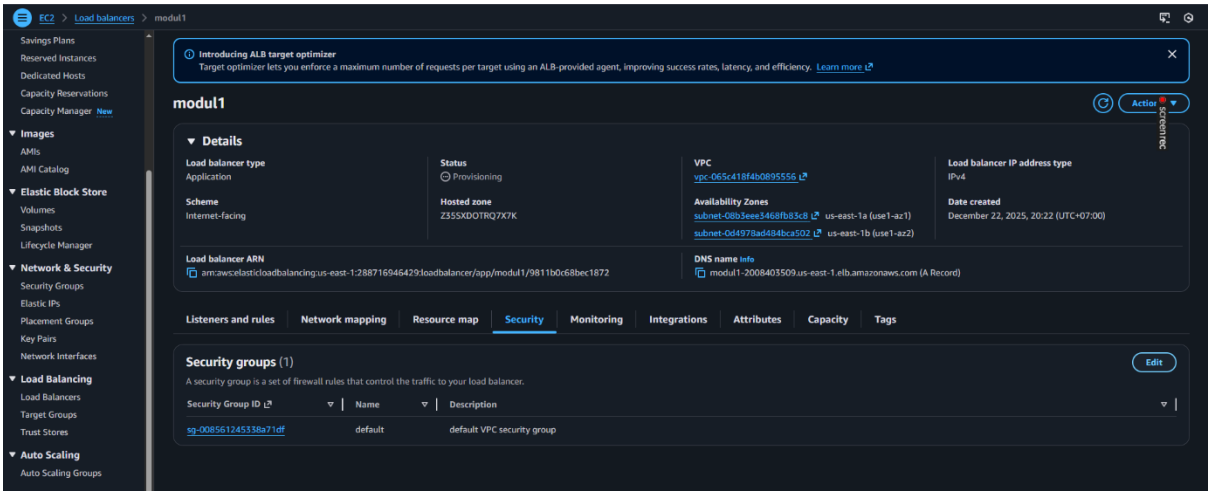
42. Settingannya seperti berikut :



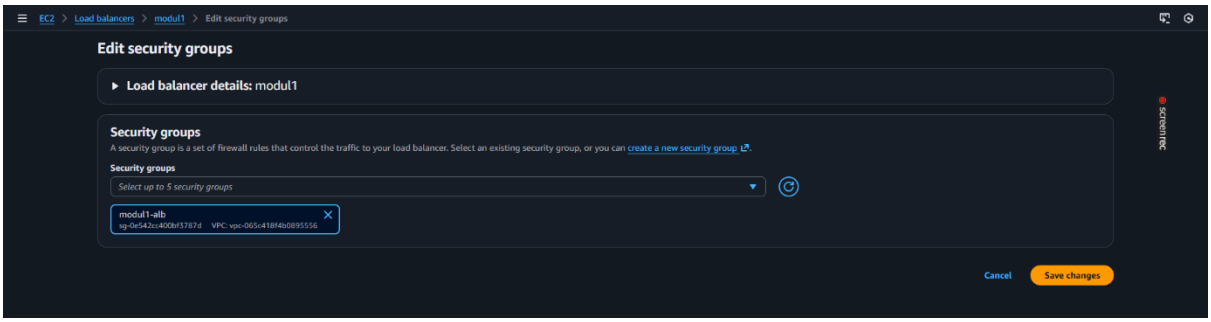

EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template or configuration
Step 2
Choose instance launch options
Step 3 - optional
Integrate with other services
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review
Integrate with other services - optional Info
Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.
Load balancing Info
Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.
Select Load balancing options
No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.
Attach to an existing load balancer
Choose from your existing load balancers.
Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.
Attach to a new load balancer
Load balancer type
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the Load Balancing console.
Application Load Balancer
HTTP, HTTPS
Network Load Balancer
TCP, UDP, TLS
Load balancer name
Name cannot be changed after the load balancer is created.
modul1
Load balancer scheme
Scheme cannot be changed after the load balancer is created.
Internal
Internet-facing
Network mapping
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.
VPC
vpc-065c418f4b0895556
modul1
Availability Zones and subnets
You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.
use1-az2 (us-east-1b)
Select a subnet
subnet-0d4978ad484bca502
use1-az1 (us-east-1a)
Select a subnet
subnet-08b3eee3468fb83c8
Listeners and routing
If you require secure listeners, or multiple listeners, you can configure them from the Load Balancing console after your load balancer is created.
Protocol
HTTP
Port
80
Default routing (forward to)
Create a target group
New target group name
An instance target group with default settings will be created.
modul1
Tags - optional
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.
Add tag
50 remaining
VPC Lattice integration options Info
To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.
Select VPC Lattice service to attach
No VPC Lattice service
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.
Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.
Create new VPC Lattice service
Application Recovery Controller (ARC) zonal shift - new Info
During an Availability Zone impairment, target instance launches towards other healthy Availability Zones.
Enable zonal shift
New instance launches will be retargeted towards healthy Availability Zones until the zonal shift is canceled.
Health checks
Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.
EC2 health checks
Always enabled
Additional health check types - optional Info
Turn on Elastic Load Balancing health checks
Recommended
Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.
Turn on VPC Lattice health checks
VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.
Turn on Amazon EBS health checks
EBS monitors whether an instance's root volume or attached volume stalls. When it reports an unhealthy volume, EC2 Auto Scaling can replace the instance on its next periodic health check.
Health check grace period Info
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.
300
seconds
Cancel
Skip to review
Previous
Next



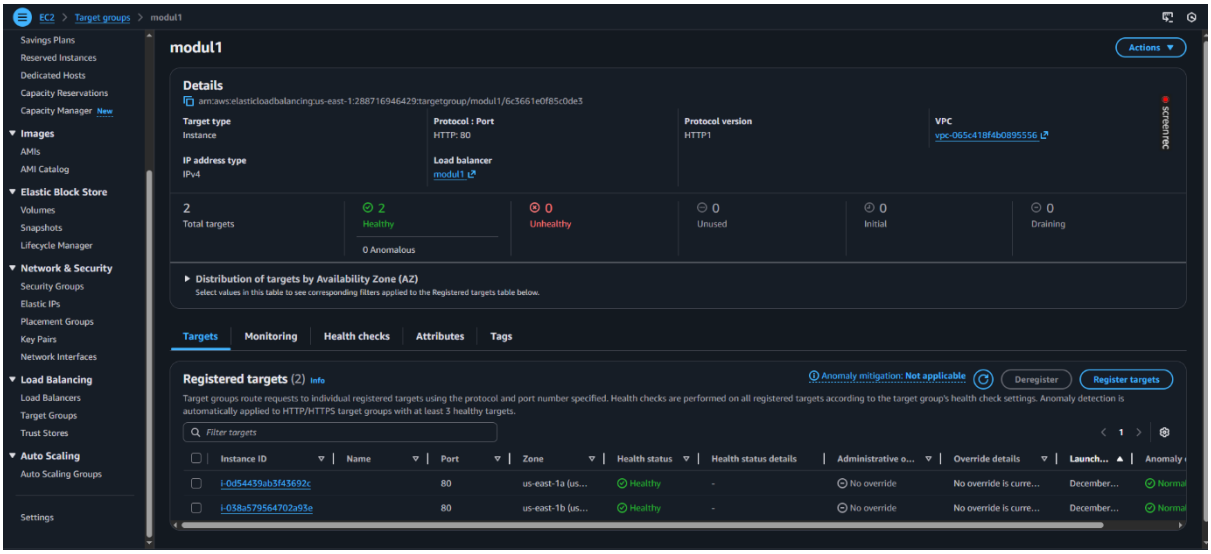
43. Pada step ke 3 tadi kalua kamu notice kita secara tidak langsung sudah membuat load balancer dan target group yang seharusnya dia masuk step 4. Itu tergantung opsi sebenarnya kamu bisa membuat load balancer dan target group secara manula tetapi cara yang tadi juga bisa
44. Jika sudah lanjut buka load balancer dan ubah bagian ecurity group nya agar menggunakan sg yang modul1-alb



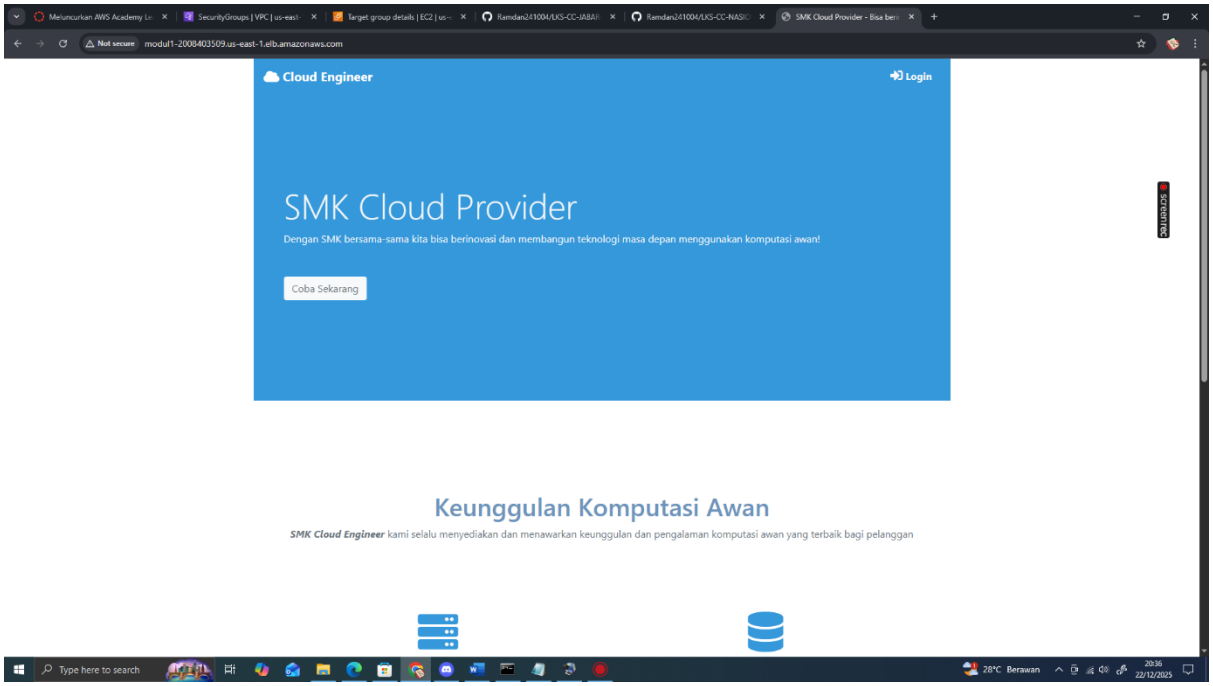
Disini sg nya masih default kamu edit bagian sg tersebut dan ubah ke sg modul1-alb seperti berikut :



45. Lanjut buka target group dan pastikan instance nya healthy semua seperti berikut :



46. Jika sudah healthy kamu bisa cek server di browser dengan menyalin dns dari load balancer dan halamannya seperti ini :



Selamat Mencoba !!!!