

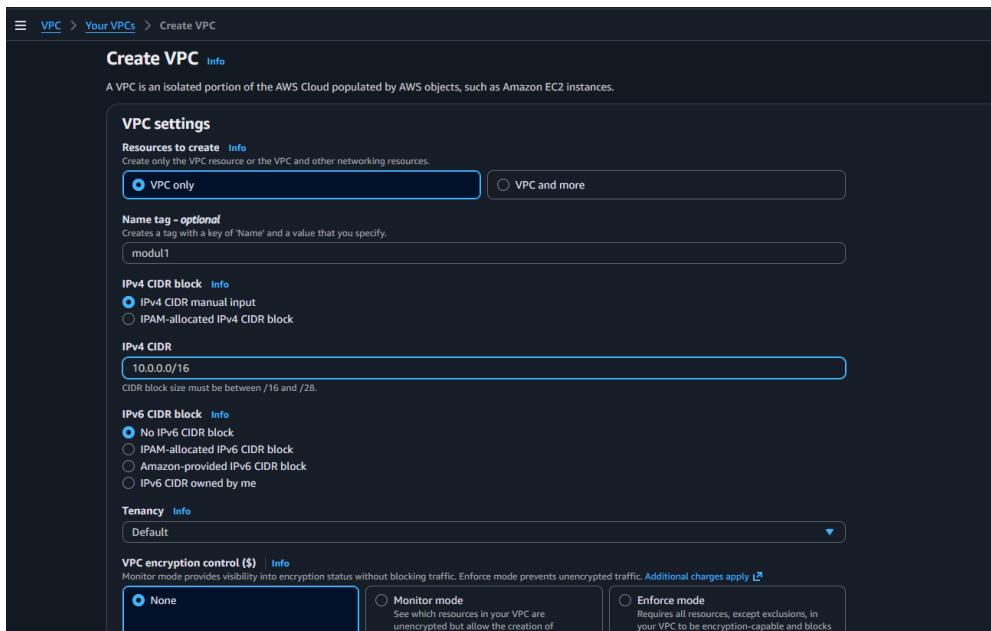
UJI COBA LKS CLOUD COMPUTING

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

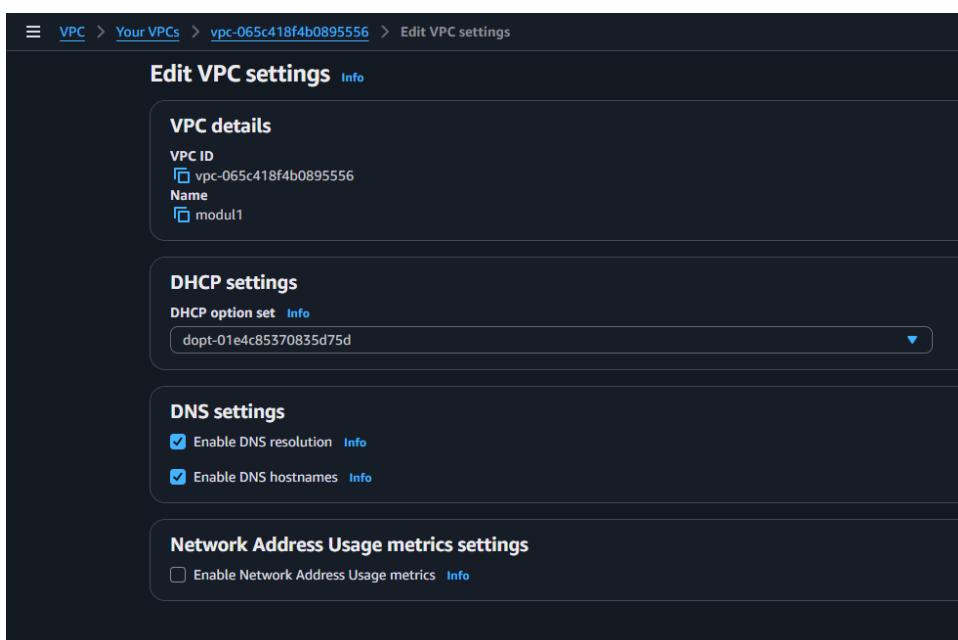
Apa itu EC2 : <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

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 :



- 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

The screenshot shows the AWS VPC Subnets creation interface. It displays two sections: "Subnet 1 of 1" and "Subnet 2 of 2".

Subnet 1 of 1 Settings:

- Subnet name:** public1
- Availability Zone:** United States (N. Virginia) / use1-az1 (us-east-1a)
- IPv4 VPC CIDR block:** 10.0.0.0/16
- IPv4 subnet CIDR block:** 10.0.0.0/24
- Tags - optional:** Name: public1

Subnet 2 of 2 Settings:

- Subnet name:** public2
- Availability Zone:** United States (N. Virginia) / use1-az2 (us-east-1b)
- IPv4 VPC CIDR block:** 10.0.0.0/16
- IPv4 subnet CIDR block:** 10.0.1.0/24
- Tags - optional:** Name: public2

Subnet 3 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs
< > ^ v

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="private1"/> Remove

[Add new tag](#)
You can add 49 more tags.
[Remove](#)

Subnet 4 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs
< > ^ v

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="private2"/> Remove

[Add new tag](#)
You can add 49 more tags.
[Remove](#)

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

[VPC](#) > [Subnets](#) > [subnet-08b3eee3468fb83c8](#) > [Edit subnet settings](#)

Edit subnet settings [Info](#)

Subnet

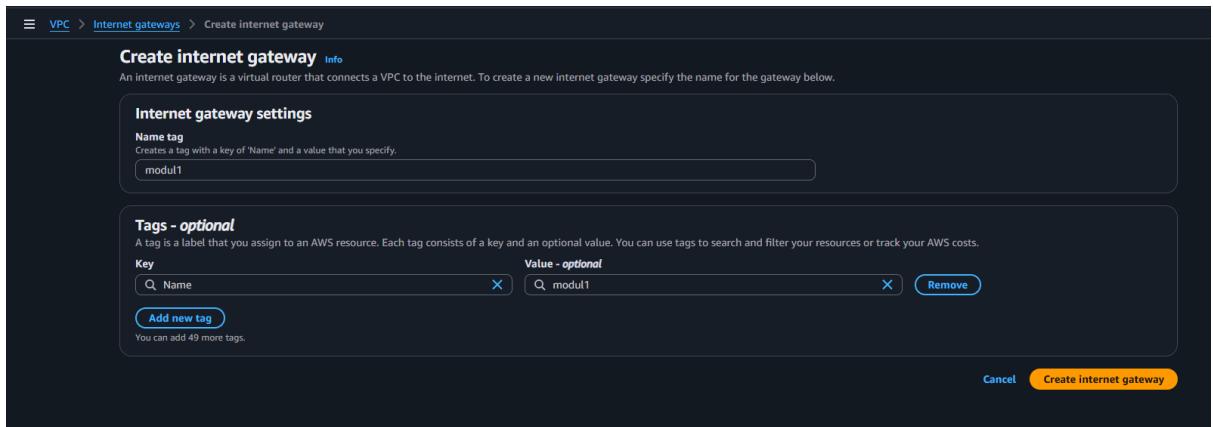
Subnet ID <input type="text" value="subnet-08b3eee3468fb83c8"/>	Name <input type="text" value="public1"/>
---	---

Auto-assign IP settings [Info](#)
Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

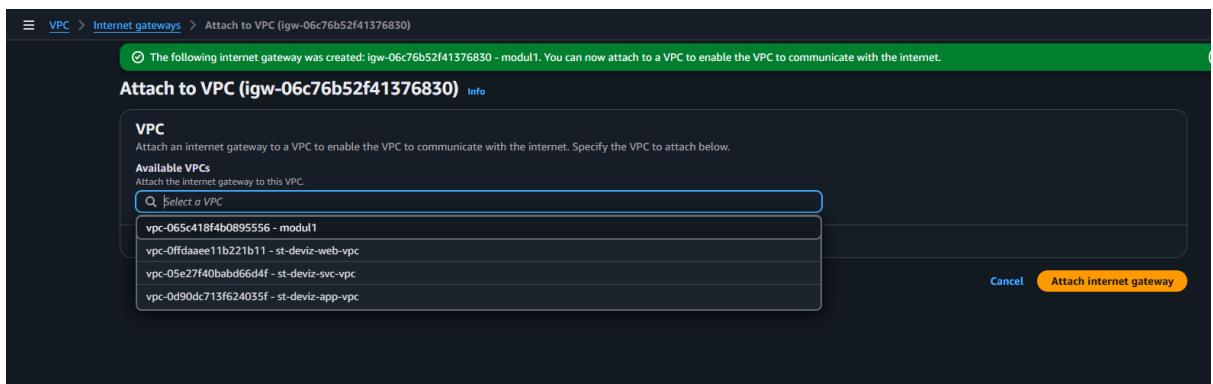
<input checked="" type="checkbox"/> Enable auto-assign public IPv4 address	<input type="checkbox"/> Enable auto-assign customer-owned IPv4 address
--	---

Option disabled because no customer owned pools found.

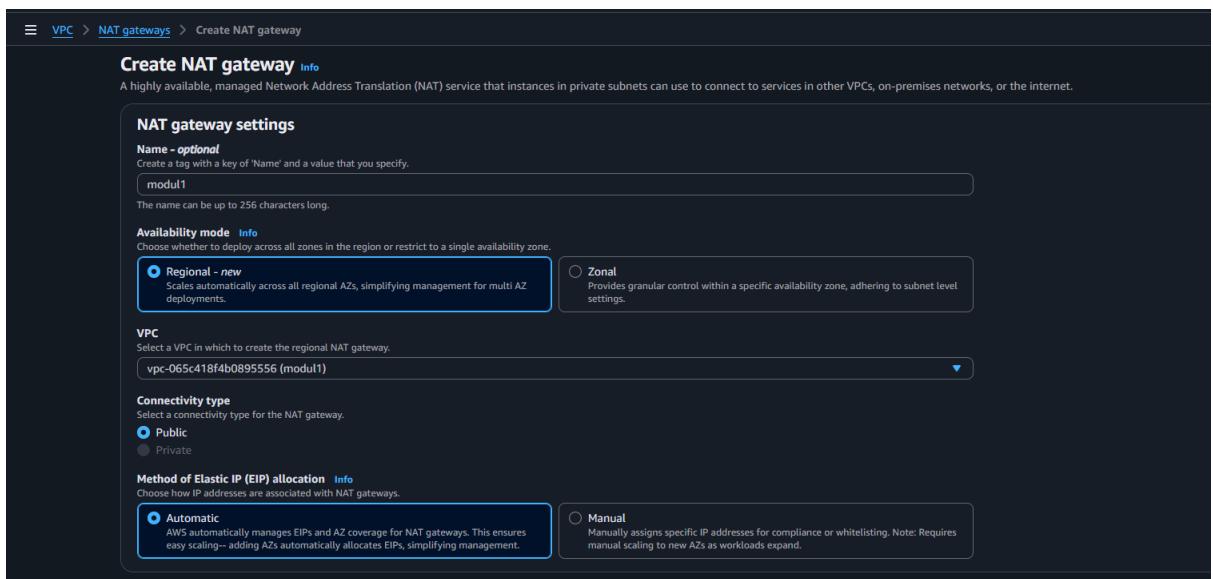
5. Lanjut step ketiga yaitu internet gateway atau gerbang menuju internet buatlah seperti berikut :



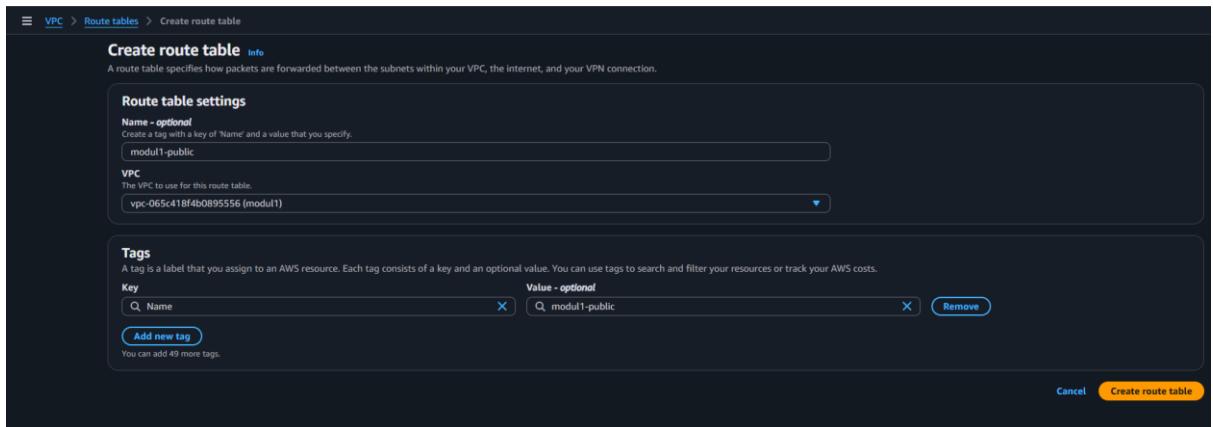
6. Attach igw ke vpc yang kita buat seperti berikut :



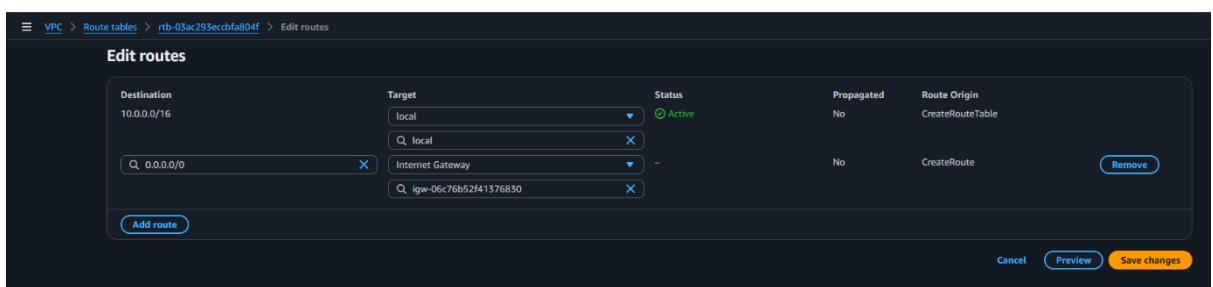
7. Step ke 4 kita buat nat gateway agar subnet private dapat terhubung ke internet secara private dari vpc sebelumnya dan dengan ketentuan seperti di gambar :



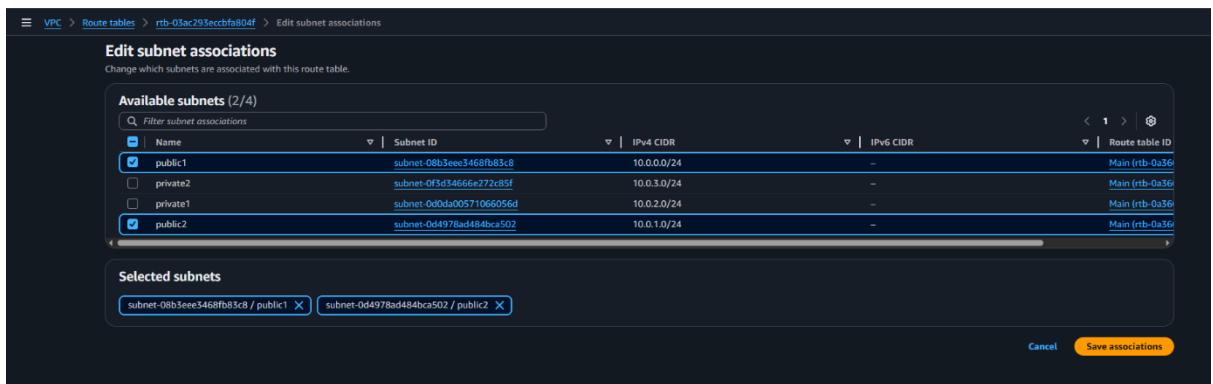
8. Step ke 5 kita buat route table agar jaringan yang kita buat tadi bisa terhubung ke internet
9. Pertama buat route table untuk public seperti berikut dan pastikan vpc nya yang tadi dibuat



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

Security Groups (3/8) <small>Info</small>						
	Name	Security group ID	Security group name	VPC ID	Description	Owner
<input type="checkbox"/>	-	sg-06f7978337b490d43	default	vpc-076f77da82db0b08a6	default VPC security group	288716946429
<input type="checkbox"/>	-	sg-008561245358a71df	default	vpc-065c418f4b0895556	default VPC security group	288716946429
<input checked="" type="checkbox"/>	-	sg-0e542cc400bf5787d	modul1-alb	vpc-065c418f4b0895556	Allow HTTP HTTPS	288716946429
<input type="checkbox"/>	-	sg-0bchaeb9d05fe7f10	default	vpc-0ffdaae11b221b11	default VPC security group	288716946429
<input checked="" type="checkbox"/>	-	sg-039f5724a941ef578	modul1-web	vpc-065c418f4b0895556	Allow SSH HTTP	288716946429
<input type="checkbox"/>	-	sg-0fa16b4ab0abd1bd7	default	vpc-0d90dc713f624035f	default VPC security group	288716946429
<input checked="" type="checkbox"/>	-	sg-09b249934da1f5fa7	modul1-bastion	vpc-065c418f4b0895556	Allow SSH	288716946429
<input type="checkbox"/>	-	sg-07277984ba5337c1d	default	vpc-05e27f40babd66d4f	default VPC security group	288716946429

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

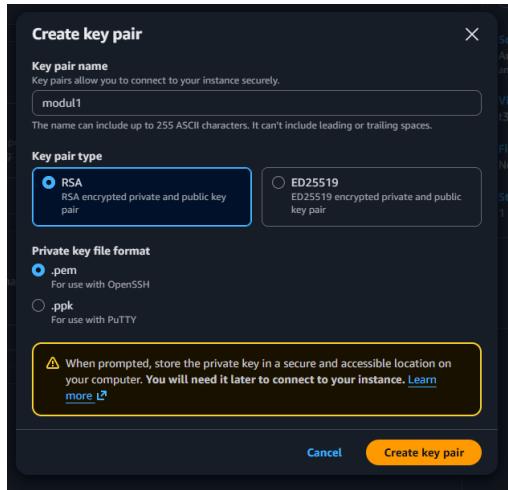
23. Lanjut ke bagian dua yaitu EC2

24. Buat ec2 bastion dengan ketentuan berikut :

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The steps are as follows:

- Name and tags**: A name 'modul1-bastion' is entered, and a link to 'Add additional tags' is available.
- Application and OS Images (Amazon Machine Image)**: The 'Quick Start' tab is selected. It lists several AMI categories: Amazon Linux (selected), macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. A search bar is present, and a link to 'Browse more AMIs' is shown.
- Amazon Machine Image (AMI)**: The 'Amazon Linux 2023 kernel-6.1 AMI' is selected. Details include: AMI ID: ami-068c0051b15cdb816, Virtualization: hvm, ENA enabled: true, Root device type: ebs. It is marked as 'Free tier eligible'. A description notes it's a modern, general purpose Linux-based OS optimized for AWS.
- Description**: The same information is repeated from the previous step.
- Instance type**: The 't3.micro' instance type is selected. It is marked as 'Free tier eligible'. Details: Family: t3, 2 vCPU, 1 GiB Memory, Current generation: true. Pricing: On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour, On-Demand SUSE base pricing: 0.0104 USD per Hour, On-Demand Linux base pricing: 0.0104 USD per Hour, On-Demand RHEL base pricing: 0.0392 USD per Hour, On-Demand Windows base pricing: 0.0196 USD per Hour. A note states 'Additional costs apply for AMIs with pre-installed software'.
- Key pair (login)**: A key pair named 'Select' is chosen. A link to 'Create new key pair' is available.

Pada bagian keypair create new key pair seperti berikut :

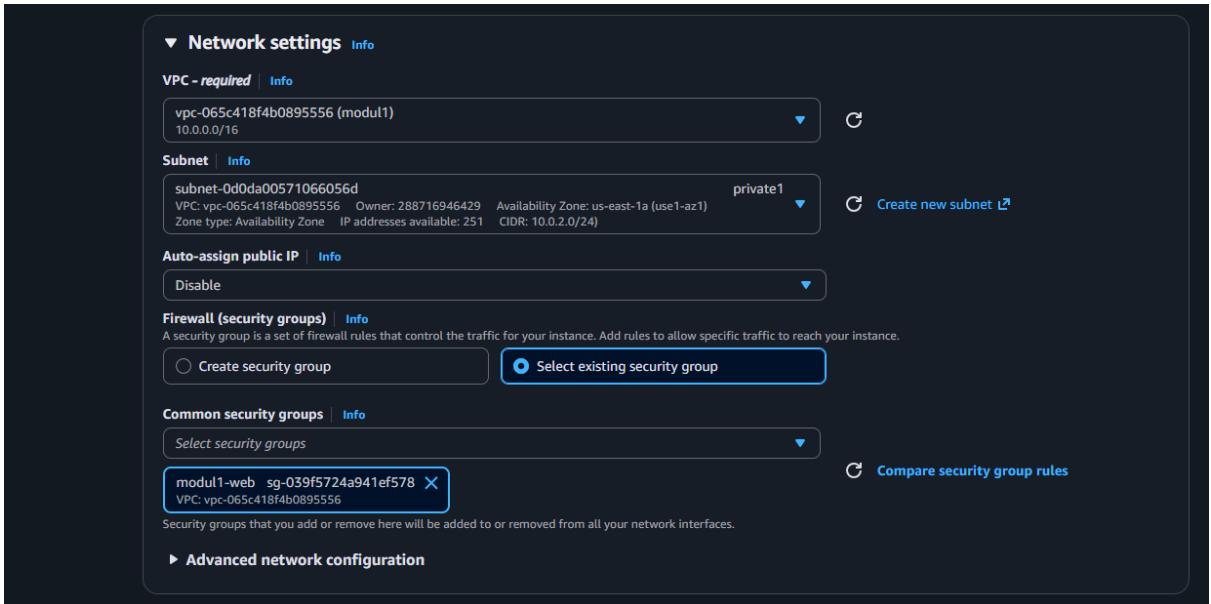


25. Lanjut bagian network setting sebagai berikut :

Sesuaikan saja setting nya seperti di gambar kemudian launch

26. Buat 1 ec2 lagi dengan nama modull1 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 :

```
root@ip-10-0-0-209:/home/ec2-user
Microsoft Windows [Version 10.0.19045.6466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>cd Downloads

C:\Users\USER\Downloads>ssh -i modul1.pem ec2-user@44.192.60.180
The authenticity of host '44.192.60.180' ('44.192.60.180') can't be established.
ED25519 key fingerprint is SHA256:E0Y3VmSsa3+i81rpleYwcof3kPxu3JKM8CA2T3Umkro.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.192.60.180' (ED25519) to the list of known hosts.

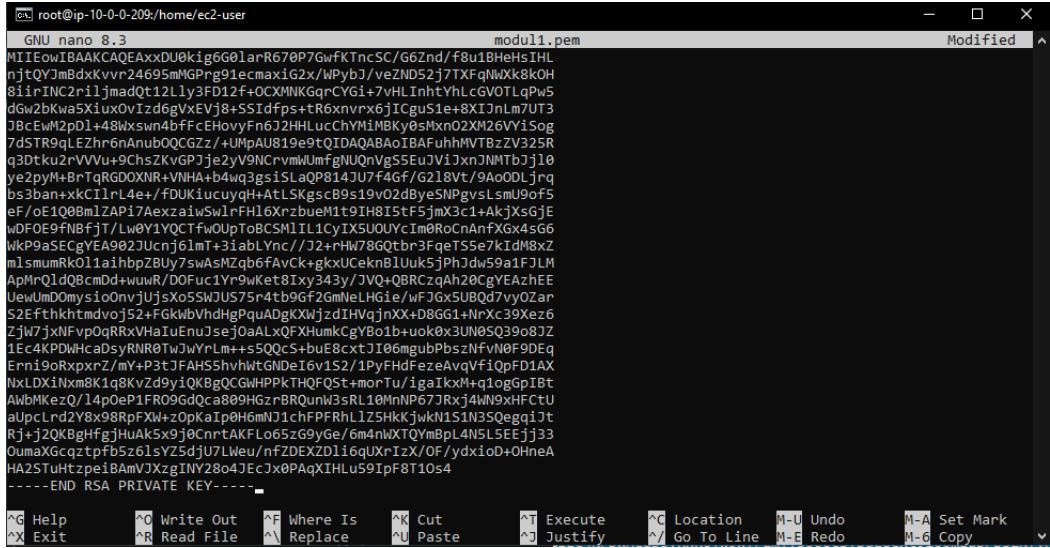
      _#
     ~\_\_###_      Amazon Linux 2023
     ~~ \####\_
     ~~ \###|
     ~~ \#/   https://aws.amazon.com/linux/amazon-linux-2023
     ~~ V~-'>
     ~~ /
     ~~ .-/
     ~~ /_/
     _m/_/`_/
[ec2-user@ip-10-0-0-209 ~]$ sudo su

[root@ip-10-0-0-209 ec2-user]# nano modul1.pem
```

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

```
modul1.pem - Notepad
File Edit Format View Help
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAx0Ukig6G01arR670P7GwfKTncSC/G6Znd/f8u18HeHsIHL
njtQJmBdxvrv+24695mGPg91ecmax162x/WPybJ+veZNDs2j7TFXqJlWxk8kOH
8iirINC2r1l1jmad0t12l1y3lFd12+0CMKIGqrCYG1+7vHLInhYhLcGVOTLgPw5
dgw2Bkwa5Xlxx0vIz6dgVxeVj8+S1Idp5+r6xnxrx61CguS1e+8XJ1nLm/UT3
3BeEm2Mp148Kxsw40ffCHeOvyFn612HHLucChY1M8k/05Mn02M26V1ySog
7dSTR9oLEzhr6nAnub0QC GZ2-/+UpAUB19e9tQDIAQABAoIBAfuhMWfBzzV25R
q30tku2rVVv+9chsZKvPjje2yV9ICrvmlnfghUQnvgs5EeJvJ1xJNMTbj10
ye2pyM+B-TqRGDXIN+vVNHa+b4wq3gs1SLaQP814JU7f4GfG228Vt/9AoODLjrq
bs3ban-xKC11L4e+/FDUK1ucucyH+AtLSKgscB9s19v02ByeSMFgvslsmU9of5
eF/oE1080m1ZAP1jAexza1Sw1r/H16XzbeuM191H815tF5jnxKc1+AkJxsGjE
wDf94Nbfj1/Lw0Y1Y1f+u0j1oBcM111C1y1X5UUUY1mehKo.nXg4x4s6
WkP9aScCgYEAg092JUcnj61mT+3iab1Yn /72+rW78GQtbr3Fe1S5e1K1dM8xZ
m1smumRk11a1hbzB7Uy7swAsMzb6fAvC+gkxUeKnB1Uuk5jPhJd9a1fJLM
ApMrQ1Q0BcmDd+wuR/DOfU+1Y9wKet81xy343y /JVQ+Q8RCzAh2B6jYEAzhEE
UewUmDMyms1oNyUjxsOSSNJUS75+4tb9fG26mLeLHg1e/wfJgsxUBQd7v0Zar
S2eftkhckmdvoJ52+FGkbDvndigpAuDgKXwjd1Hvq1xN+D8G1+Nx39xe2
ZjW7jxFvpdqRxVh1eUnujs0aAxQFХумкcгBоl+uоkхJUN08039o8zJ
1Ec4KPDMcDaSyNR0tWjvYlN++s500c5+bu8cxtJ10ngubPbszlfvNf0Eq
Ern19oRpxprZ/m+P+3tFAHS5nhvltkGMDt6v1S2/1PvFHdFzezAvqF1q0pFD1AX
NxLDK1Nx8k81q8Kv2d9y1QK8gQCGhIPPktTHQFSt+morTu/igaIkM+q1ogpIBt
AbWkMe0/14q0e1FR096Qca899HGr-BRQunW3sRL10MnP673Rxj4Wl9xHFcCU
aUpclrd2Y8x98RpFXW+oPaIp0H6mNj1chPFRhL175HkKjwkH151N3Sqeq1jt
R+j2QkBgfHgHuAk5x98Cn+AKFL0652G9yge/6m4n1XTQfmBlpL4H515Ej33
QunaXGcqztpFb5261sYz5dJ7UWeu/nfZDEXZD116qUx1zX/Of/dxi0+0hneA
HA25Tuhtzpe1AmV3Xzg1NY284jEcJx0pAqXHULu591pf8T104
-----END RSA PRIVATE KEY-----
```

Lalu paste di nano tadi seperti ini :



```
root@ip-10-0-0-209:/home/ec2-user
GNU nano 8.3                         moduli.pem
MIIEowIBAAKCAQEaxxDU0kig6G0larR670P7GwFKTncSC/G6Znd/f8u1BHeHsIHL
njtQYmBdxKvr24695mMGPr91ecmaxiG2x/WPybJ/vezND52j7TXFgNWxkbkOH
8iinINC2riljmadQt12lly3FD12f+OCXMKNGqrCYGi+7vHLnhLYLcVOtLqPw5
dGw2bkWa5xiu0vIzd6gVxEVj8-SS1dfps+tR6xnvrx6jICgu51e+8XIJnLm7UT3
JBCewM2p0l+48Vxswm4bfccEhovyFn632HHlucChYMiMBKyosMxn02XM26VViSog
7dSTR9qLEZhr6nAnubQCGGZ/+UMpAU819e9tQIDAQABoIAFuHmVTBzV325R
q3Dtku2rVVVu+9chsZkVGPJje2yV9NcrvmlUmfQNUrnVg5EuJvijxnJNMTbj10
ye2py+Br-TqRGDXMR+NHA+b4wq3gs1laopR814JU7f4Gf/G2l8Vt/9Ao0DLjrq
bs3ban+xKCllrl4e+/fDUKicuqyqH+AtLSkgscB9s19v02dByeSNPgvslsmu9of5
eF/oEI00bm1zAPi7exzalwSw1rFH16Xrzblue1t9IH815t5jmX3c1+AkjXsGjE
wDOFO9fNBfjt/Lw0Y1YCTfw0UpToBcSm1L1cyIX5UOUYc1m0R0CnAnfxGx4sG6
Wkp9aSEcgYE902JUcnj61mt+3iab1Yrc//J2+rHv78GQtbn3FqeTS5e7kldM8xZ
m1smumrk011aihbz8Uy7swAsMZqb6fAvCk+gkxUceknb1Uuk5jPhJdw59a1fJLM
ApMrQldQbcmDd+wuW/DOfuciYr9wKet8Ix3y3/JVQ+QBRCzqAh20CgYEazhEE
UewUmDomy5io0nvjujsx05SWJUS7r4tb9gf2gmneLHG1e/wFJGx5UBqd7vyOzar
S2Efthkhtmdvoj52+FgKwbVhdHgPquADgKXkjzdiHvqjnX+D8G1+NrXc39XeZ6
Zjw7xhFvpQgRRxVhaIuEnuJejoaLxqFXHumkCgyBo1b+uokbx3UN0S03908JZ
1Ec4KPDWkcaDsyRNROTwJyYr1m++s5Q0Cs+buE8cxtJI06mgubPbszNFvN0F9DEq
Erni90rpxpxrZ/m+p3tFAHS5hvhwGhde16v1S2/1pYFhdFezeAvqVfiopFD1AX
NxLDXInxm8K1q8kvzdy9iQK8gQCgWHPPkTHQFOSt+morTu/iga1kxM+1ogGpIBt
AwbIKez0/14p0ep1FR09Gdoca899Hgz+BR0unW3rL10mNP673Rxj4wN9xHFcTu
aUpcLrd2Y8x98RpFXW+OpKaIp0H6mN1chFPFRHllz5HkKjwKn1S1N3S0eqgi0t
Rj+j2QkBghfgjhUAK5x9j0CnrtAKFLo65zG9yGe/6m4nWTQyMbpL4N5L5EEjj33
OumaXGcqztpfb5z6lsYz5djU7LWeu/nfZDEXZDl16qUxIxZ/Of/ydxioD+OhneA
HA2STUhzpeIBmVJXzgINY28o4JEcJx0PAqXHlu59IpF8T1Os4
-----END RSA PRIVATE KEY-----
```

File menu: G Help, A Write Out, F Where Is, K Cut, T Execute, C Location, M-U Undo, M-A Set Mark
Edit menu: X Exit, R Read File, R Replace, U Paste, J Justify, ^/ Go To Line, M-E Redo, M-C Copy

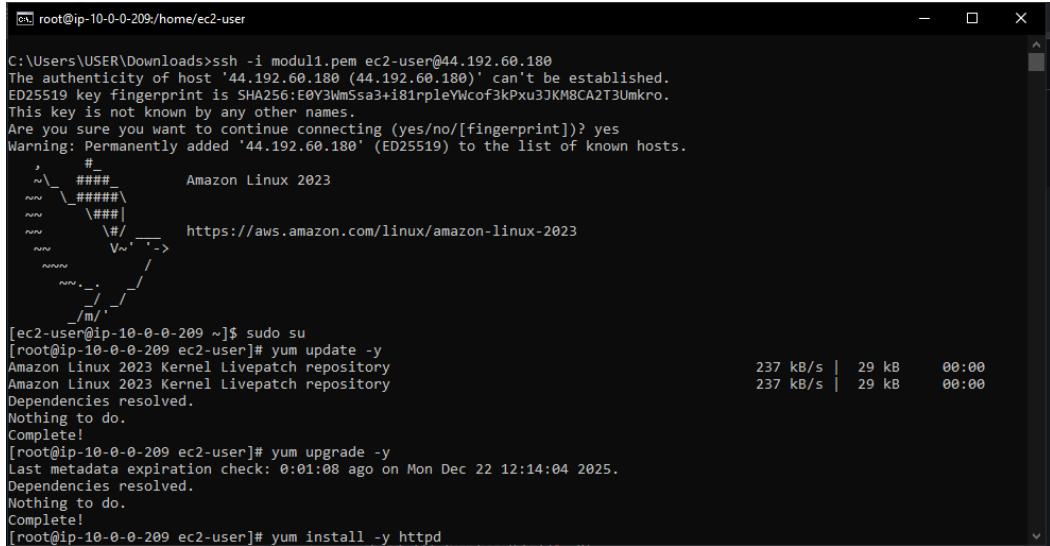
Jika sudah CTRL + X, Y, ENTER

32. Lakukan chmod seperti berikut



```
[root@ip-10-0-0-209 ec2-user]# nano moduli.pem
[root@ip-10-0-0-209 ec2-user]# chmod 400 moduli.pem
```

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



```
root@ip-10-0-0-209:/home/ec2-user
C:\Users\USER\Downloads>ssh -i moduli.pem ec2-user@44.192.60.180
The authenticity of host '44.192.60.180' ('44.192.60.180') can't be established.
ED25519 key fingerprint is SHA256:E0Y3WmSsa3+i81rpleYwcof3kPxu3JKM8CA2T3Umkro.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.192.60.180' (ED25519) to the list of known hosts.

[ec2-user@ip-10-0-0-209 ~]$ sudo su
[ec2-user@ip-10-0-0-209 ~]$ yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-0-209 ~]$ sudo su
[ec2-user@ip-10-0-0-209 ~]$ yum upgrade -y
Last metadata expiration check: 0:01:08 ago on Mon Dec 22 12:14:04 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-0-209 ~]$ yum install -y httpd
```

```

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-htpd-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-htpd-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-lmdb-1.6.3-1.amzn2023.0.2.x86_64
generic-logos-htpd-18.0.0-12.amzn2023.0.3.noarch
httpd-core-2.4.65-1.amzn2023.0.2.x86_64
httpd-tools-2.4.65-1.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 : libxcbrypt-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
cpp-11.5.0-5.amzn2023.0.5.x86_64
gcc-11.5.0-5.amzn2023.0.5.x86_64
gcc-plugin-annobin-11.5.0-5.amzn2023.0.5.x86_64
git-core-2.50.1-1.amzn2023.0.1.x86_64
glIBC-devel-2.34-231.amzn2023.0.1.x86_64
guile22-2.2.7-2.amzn2023.0.3.x86_64
libmpc-1.2.1-2.amzn2023.0.2.x86_64
libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64
make-1:4.3-5.amzn2023.0.2.x86_64
perl-File-Find-1.37-477.amzn2023.0.7.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64

annobin-plugin-gcc-12.69-1.amzn2023.0.1.x86_64
gc-8.0.4-5.amzn2023.0.2.x86_64
gcc-c++-11.5.0-5.amzn2023.0.5.x86_64
git-2.50.1-1.amzn2023.0.1.x86_64
git-core-doc-2.50.1-1.amzn2023.0.1.noarch
glIBC-headers-x86-2.34-231.amzn2023.0.1.noarch
kernel-headers-1:6.1.158-180.294.amzn2023.x86_64
libstdc++-devel-11.5.0-5.amzn2023.0.5.x86_64
libxcbrypt-devel-4.4.33-7.amzn2023.x86_64
perl-Error-1:0.17029-5.amzn2023.0.2.noarch
perl-Git-2.50.1-1.amzn2023.0.1.noarch
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
cpp-11.5.0-5.amzn2023.0.5.x86_64
gcc-11.5.0-5.amzn2023.0.5.x86_64
gcc-plugin-annobin-11.5.0-5.amzn2023.0.5.x86_64
git-core-2.50.1-1.amzn2023.0.1.x86_64
glIBC-devel-2.34-231.amzn2023.0.1.x86_64
guile22-2.2.7-2.amzn2023.0.3.x86_64
libmpc-1.2.1-2.amzn2023.0.2.x86_64
libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64
make-1:4.3-5.amzn2023.0.2.x86_64
perl-File-Find-1.37-477.amzn2023.0.7.noarch
perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64

annobin-plugin-gcc-12.69-1.amzn2023.0.1.x86_64
gc-8.0.4-5.amzn2023.0.2.x86_64
gcc-c++-11.5.0-5.amzn2023.0.5.x86_64
git-2.50.1-1.amzn2023.0.1.x86_64
git-core-doc-2.50.1-1.amzn2023.0.1.noarch
glIBC-headers-x86-2.34-231.amzn2023.0.1.noarch
kernel-headers-1:6.1.158-180.294.amzn2023.x86_64
libstdc++-devel-11.5.0-5.amzn2023.0.5.x86_64
libxcbrypt-devel-4.4.33-7.amzn2023.x86_64
perl-Error-1:0.17029-5.amzn2023.0.2.noarch
perl-Git-2.50.1-1.amzn2023.0.1.noarch
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]#

```

```

root@ip-10-0-0-209:/home/ec2-user
Installed:
  annobin-docs-12.69-1.amzn2023.0.1.noarch
  cpp-11.5.0-5.amzn2023.0.5.x86_64
  gcc-11.5.0-5.amzn2023.0.5.x86_64
  gcc-plugin-annobin-11.5.0-5.amzn2023.0.5.x86_64
  git-core-2.50.1-1.amzn2023.0.1.x86_64
  glibc-devel-2.34-231.amzn2023.0.1.x86_64
  guile22-2.2.7-2.amzn2023.0.3.x86_64
  libmpc-1.2.1-2.amzn2023.0.2.x86_64
  libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64
  make-1:4.3-5.amzn2023.0.2.x86_64
  perl-File-Find-1.37-477.amzn2023.0.7.noarch
  perl-TermReadKey-2.38-9.amzn2023.0.2.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]# cp -R /modul-1/* /var/www/html/
[root@ip-10-0-0-209 ec2-user]# chown ec2-user:apache /var/www/html/* -Rf
[root@ip-10-0-0-209 ec2-user]# chmod 777 /var/www/html/* -Rf
[root@ip-10-0-0-209 ec2-user]#

```

```

root@ip-10-0-0-209:/home/ec2-user
Installed:
  annobin-docs-12.69-1.amzn2023.0.1.noarch
  cpp-11.5.0-5.amzn2023.0.5.x86_64
  gcc-11.5.0-5.amzn2023.0.5.x86_64
  gcc-plugin-annobin-11.5.0-5.amzn2023.0.5.x86_64
  git-core-2.50.1-1.amzn2023.0.1.x86_64
  glibc-devel-2.34-231.amzn2023.0.1.x86_64
  guile22-2.2.7-2.amzn2023.0.3.x86_64
  libmpc-1.2.1-2.amzn2023.0.2.x86_64
  libtool-ltdl-2.4.7-1.amzn2023.0.3.x86_64
  make-1:4.3-5.amzn2023.0.2.x86_64
  perl-File-Find-1.37-477.amzn2023.0.7.noarch
  perl-TermReadKey-2.38-9.amzn2023.0.2.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]# cp -R /modul-1/* /var/www/html/
[root@ip-10-0-0-209 ec2-user]# chown ec2-user:apache /var/www/html/* -Rf
[root@ip-10-0-0-209 ec2-user]# chmod 777 /var/www/html/* -Rf
[root@ip-10-0-0-209 ec2-user]# systemctl restart httpd
[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
```

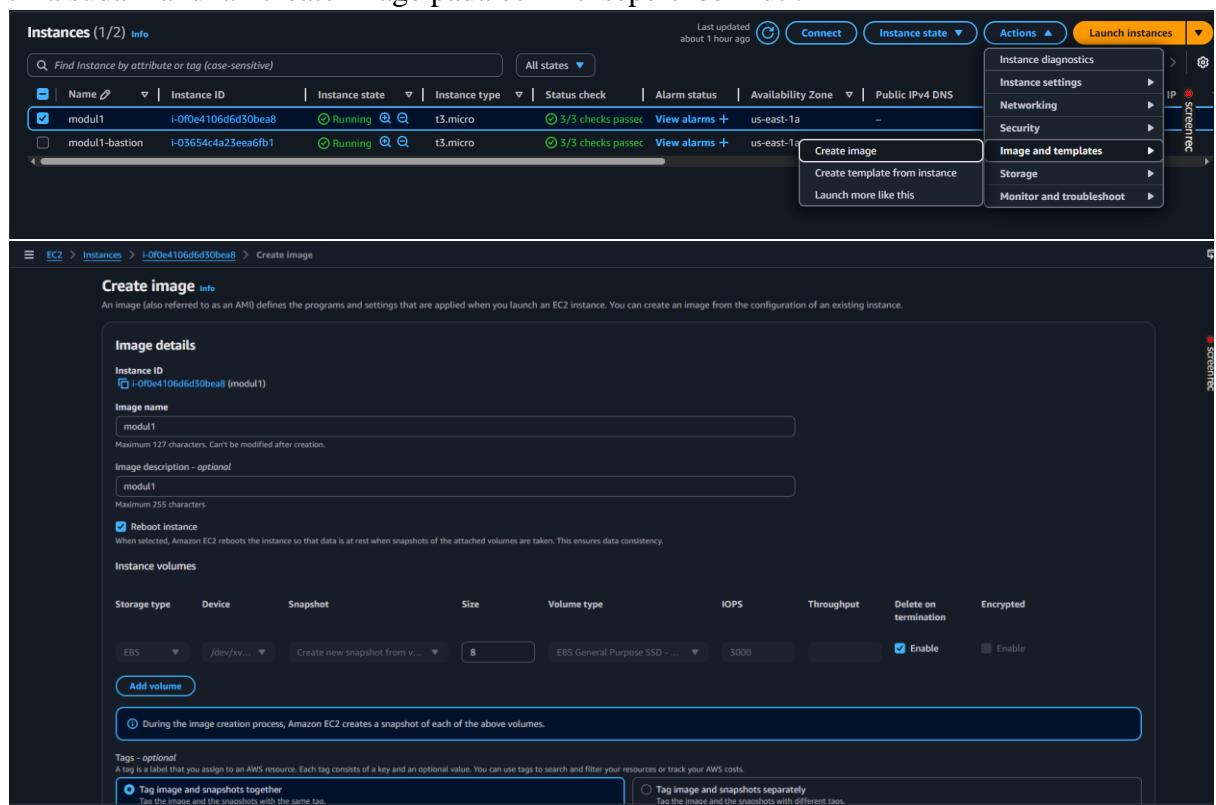
```
# Httpd
yum install -y httpd
```

```
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	ENI 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	ENI 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 modull-web jika sudah create

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

The screenshot shows the AWS EC2 Auto Scaling Groups wizard. Step 1 is selected: "Choose launch template or configuration". The sidebar on the left lists steps 1 through 7. The main form is titled "Choose launch template or configuration" and contains the following fields:

- Name:** Auto Scaling group name (modul1)
- Launch template:** modul1 (selected)
- Description:** modul1
- AMI ID:** ami-0fcfbc8a7e5fea73c7
- Key pair name:** modul1
- Launch template:** modul1 (lt-0fc4af1f155c2e242ea)
- Instance type:** t2.micro
- Security groups:** -
- Request Spot Instances:** No
- Security group IDs:** -

42. Settingannya seperti berikut :

The screenshot shows the AWS EC2 Auto Scaling Groups wizard. Step 2 is selected: "Choose instance launch options". The sidebar on the left lists steps 1 through 7. The main form is titled "Choose instance launch options" and contains the following fields:

- Instance type requirements:** Launch template: modul1 (selected), Version: Default, Instance type: t2.micro
- VPC:** vpc-065c1814b0895556 (selected)
- Network:** Availability Zones and subnets:
 - us-east-1a2 (us-east-1b) | subnet-0f5d54666e272c85f (private2)
 - use-1a1 (us-east-1a) | subnet-0d0da00571066056d (private1)
- Availability Zone distribution - new:** Balanced best effort (selected)

At the bottom right, there are buttons for "Cancel", "Skip to review", "Previous", and "Next".

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
 Choose launch template or configuration
 Step 2
 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-065c418f4b089556 Info

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.

<input type="checkbox"/> us-east-1a (us-east-1a)	Select a subnet subnet-0d4978ad484bca502
<input type="checkbox"/> us-east-1a (us-east-1a)	Select a subnet subnet-0bb5eee5468fb83c8

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 Info

Application Recovery Controller (ARC) zonal shift - new Info
During an Availability Zone replacement, 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 monitors 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**

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
 Choose launch template or configuration
 Step 2
 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

Configure group size and scaling - optional Info

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size Info

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity
Specify your group size.

Scaling Info

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits
Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity Equal or less than desired capacity

Max desired capacity Equal or greater than desired capacity

Automatic scaling - optional
Choose whether to use a target tracking policy. You can set up other metric-based scaling policies and scaling after creating your Auto Scaling group.

No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy
Choose a CloudWatch metric as target value and let the scaling policy adjust the desired capacity in response to the metric's value.

Scaling policy name

Metric type Info
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

Target value

Instance warmup Info
 seconds

Disable scale-in to create only a scale-out policy

Instance maintenance policy Info

Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.

Choose a replacement behavior depending on your availability requirements

Mixed behavior <input checked="" type="radio"/> No policy <small>For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.</small>	Prioritize availability <input type="radio"/> Launch before terminating <small>Launch new instances and wait for them to be ready before terminating others. This allows you to go below your desired capacity at a given percentage and may temporarily increase costs.</small>	Control costs <input type="radio"/> Terminate and launch <small>Terminate and launch instances at the same time. This allows you to go below your desired capacity at a given percentage and may temporarily reduce availability.</small>	Flexible <input type="radio"/> Custom behavior <small>Set custom values for the minimum and maximum number of available instances. This gives you greater flexibility in setting EC2 Auto Scaling goes when replacing instances.</small>
---	---	--	---

Additional capacity settings

Capacity Reservation preference Info
Select whether you want Auto Scaling to launch instances into an existing Capacity Reservation or Capacity Reservation resource group.

Default
Auto Scaling uses the Capacity Reservation preference from your launch template.

None

Instances
Instances will not be launched into a Capacity Reservation.

Capacity Reservations only
Instances will only be launched into a Capacity Reservation. If capacity isn't available, the instances fail to launch.

Capacity Reservations first
Instances will attempt to launch into a Capacity Reservation first. If capacity isn't available, instances will run on On-Demand capacity.

Additional settings

Instance scale-in protection
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

Enable instance scale-in protection

Monitoring Info

Enable group metrics collection within CloudWatch

Default instance warmup Info
The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.

Enable default instance warmup

Cancel **Skip to review** **Previous** **Next**

43. Pada step ke 3 tadi kalau 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 modull1-alb

The screenshot shows the AWS EC2 Load Balancers console. On the left sidebar, under 'Network & Security', 'Load Balancing' is selected. In the main area, a load balancer named 'modul1' is displayed. The 'Security' tab is active, showing a single security group assigned: 'sg-008561245338a71df' (default). Other tabs include 'Listeners and rules', 'Network mapping', 'Resource map', 'Monitoring', 'Integrations', 'Attributes', 'Capacity', and 'Tags'.

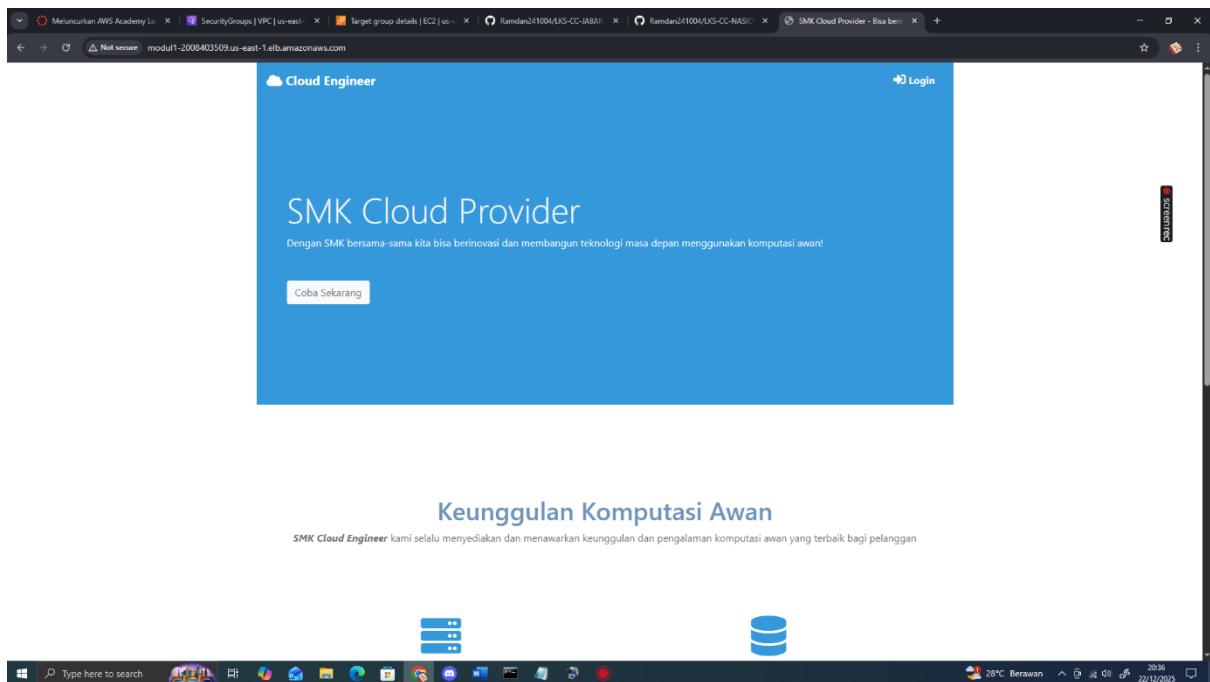
Disini sg nya masih default kamu edit bagian sg tersebut dan ubah ke sg modul1-alb seperti berikut :

The screenshot shows the 'Edit security groups' dialog for the load balancer 'modul1'. Under the 'Security groups' section, 'modul1-alb' is selected from a dropdown menu. The 'Save changes' button is visible at the bottom right.

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

The screenshot shows the AWS EC2 Target Groups console. On the left sidebar, under 'Network & Security', 'Load Balancing' is selected. In the main area, a target group named 'modul1' is displayed. The 'Targets' tab is active, showing 2 healthy targets. The 'Registered targets' table lists two instances: 'i-0d54439ab3f43692c' and 'i-058a579564702a93e', both marked as 'Healthy'.

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



Selamat Mencoba !!!!