

---

## Bab 6

# AWS Basic Server Production Part 1



### Detail Materi

Indikator :  
Mengenal Cloud Platform AWS, dapat mengoperasikan layanan Amazon VPC dan Amazon EC2 sesuai kebutuhan. Melakukan Deployment Aplikasi pada Server Production



Cloud Computing dan  
Platform Cloud

Kenapa AWS Menjadi Pilihan  
Platform Cloud



Pengenalan & Praktek Amazon VPC

Pengenalan & Praktek Amazon EC2

Deployment Aplikasi pada  
Server Production



Deployment Aplikasi pada  
Server Production

## Modul Sekolah DevOps Cilsy

Hak Cipta © 2020 PT. Cilsy Fiolution Indonesia

Hak Cipta dilindungi Undang-Undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronis maupun mekanis, termasuk mecopy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis dan Penerbit.

Penulis : Muhammad Fakhri A., Adi Saputra, Irfan Herfiandana & Tresna Widiyaman

Editor: Taufik Maulana, Muhammad Fakhri Abdillah, Rizal Rahman & Tresna Widiyaman

**Revisi Batch 9**

Penerbit : **PT. Cilsy Fiolution Indonesia**

Web Site : <https://cilsyfiolution.com> , <https://devops.cilsy.id>

Sanksi Pelanggaran Pasal 113  
Undang-undang Nomor 28 Tahun 2014  
tentang Hak Cipta

1. Setiap orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam pasal 9 ayat (1) huruf i untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan atau pidana denda paling banyak Rp100.000.000,00 (seratus juta rupiah).
2. Setiap orang yang dengan tanpa hak dan atau tanpa izin pencipta atau pemegang hak cipta melakukan pelanggaran hak ekonomi pencipta sebagaimana dimaksud dalam pasal 9 ayat (1) huruf c, huruf d, huruf f, dan atau huruf h, untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah)
3. Setiap orang yang dengan tanpa hak dan atau tanpa izin pencipta atau pemegang hak melakukan pelanggaran hak ekonomi pencipta sebagaimana dimaksud dalam pasal 9 ayat (1) huruf a, huruf b, huruf e, dan atau huruf g, untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah)
4. Setiap orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah)

## Daftar Isi

CoveModul Sekolah DevOps Cilsy.....	2
Daftar Isi.....	3
6. AWS Basic Server Production Part 1.....	6
Learning Outcomes.....	6
Outline Materi.....	6
6.1. Amazon Web Service (AWS).....	7
6.1.1. Apa itu Cloud Computig?.....	7
6.1.2. Apa itu AWS ?.....	9
6.1.3. Sistem Pembayaran AWS.....	10
6.1.4. Membuat Akun AWS.....	11
6.1.5. Login Akun AWS.....	14
6.1.6. Perbedaan AWS dengan Cloud Sejenisnya.....	16
6.1.6.1. Harga.....	17
6.1.6.2. Layanan.....	18
6.1.6.3. Pilihan Stack.....	19
6.1.7. Exercise.....	19
6.2. Roadmap Pembelajaran.....	20
6.3. Amazon VPC.....	21
6.3.1. Apa itu Amazon VPC ?.....	21
6.3.2. Service AWS yang dapat deploy di VPC.....	22
6.3.3. Fitur Amazon VPC.....	22
6.3.3.1. Ketersediaan high availability dan Availability Zone pada Region AWS.....	23
6.3.3.2. Subnet.....	23
6.3.3.3. Route Tables.....	23
6.3.3.4. Internet Gateways (IGW).....	23
6.3.3.5. NAT Gateways.....	24

6.3.3.6. Network Access Control Lists (NACL).....	24
6.3.4. Contoh Skema Amazon VPC.....	24
6.3.5. Exercise.....	25
6.4. Praktik VPC – Virtual Private Cloud.....	25
6.4.1. Membuat VPC.....	25
6.4.2. Membuat Subnet.....	29
6.4.3. Internet Gateways.....	32
6.4.4. NAT Gateway.....	35
6.4.5. Menghapus VPC.....	39
6.4.6. Exercise.....	41
6.5. Amazon EC2.....	42
6.5.1. Apa itu Amazon EC2 ?.....	42
6.5.2. AMI.....	42
6.5.3. Instance.....	43
6.5.4. Instance Type.....	44
6.5.4.1. Instance Generasi Saat Ini.....	45
6.5.4.2. Instance Generasi Sebelumnya.....	45
6.5.5. KeyPairs.....	46
6.5.6. Request Limit Instance AWS.....	46
6.5.7. Exercise.....	49
6.6. Praktik EC2 – Elastic Cloud Computing.....	49
6.6.1. Membuat Linux EC2 Instance.....	49
6.6.1.1. Remote Instance via Linux.....	58
6.6.1.2. Remote EC2 via Putty Windows.....	59
6.6.2. Membuat Windows Server EC2 Instance.....	63
6.6.2.1. Remote Desktop Windows Server melalui Linux.....	69
6.6.2.2. Remote Desktop Windows Server melalui Windows.....	74
6.6.3. Elastic IP.....	78
6.6.4. Mematikan dan Menghapus Instance.....	80
6.6.4.1. Mematikan Instance.....	80
6.6.4.2. Menghapus Instance.....	81



6.6.5. Excercise.....	82
6.7. Deploy Aplikasi Web di EC2.....	83
6.7.1. Setup Layanan Web Server.....	83
6.7.2. Setup Web Aplikasi.....	83
6.8. Summary.....	85

## 6.

# AWS Basic Server Production Part 1

## Learning Outcomes

Setelah selesai mempelajari bab ini, peserta mampu :

1. Memahami Cloud Computing dan Platform didalamnya.
2. Memahami Kenapa AWS Menjadi Pilihan Platfrom Cloud.
3. Mampu Menggunakan Layanan Amazon VPC di AWS.
4. Dapat Membuat Instance Sebagai VM Server di Amazon EC2.
5. Dapat Melakukan Deploy Aplikasi Pada Server Production.

## Outline Materi

1. Pengenalan Cloud Computing
2. Pengenalan Amazon Web Services
3. Macam-macam Platform Cloud
4. Pengenalan & Praktek Amazon VPC
5. Pengenalan & Praktek Amazon EC2
6. Setup WebServer Pada Server Production
7. Setup Web Aplikasi pada Server Production

## 6.1. Amazon Web Service (AWS)

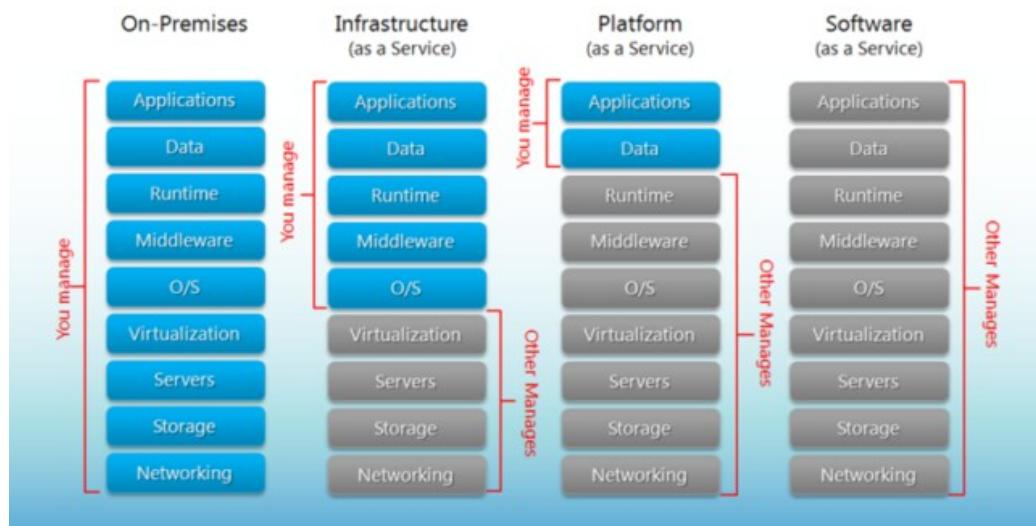
### 6.1.1. Apa itu Cloud Computing?

Cloud Computing dapat diibaratkan seperti menyewa sumberdaya IT. Bentuknya dapat berupa infrastuktur (virtual machine / network), storage, aplikasi, database, dll. Disediakan siap pakai oleh Cloud Provider, dapat kita akses dan gunakan secara fleksibel sesuai kebutuhan. Nantinya, kita akan dikenakan biaya oleh Cloud Provider hanya sebesar resource yang kita gunakan (*Pay-as-You-go*). Dengan Cloud Computing, kita tidak perlu memiliki server dan perangkat sendiri.

Ada 3 (tiga) model pengiriman dalam Cloud Computing :

1. Software as a Service (SaaS) merupakan layanan untuk menggunakan aplikasi yang telah disediakan penyedia layanan mengelola platform dan infrastruktur yang menjalankan aplikasi tersebut.
2. Platform as a Service (PaaS) merupakan layanan untuk menggunakan platform yang telah disediakan pengembang fokus pada aplikasi yang dibuat tanpa memikirkan tentang pemeliharaan platform. Jika kita analogikan seperti halnya kita menyewa hotel, kita tinggal tidur dikamar yang kita sewa tanpa peduli bagaimana perawaan dari kamar dan lingkungannya. Salah satu contoh layanan PaaS adalah AWS, Azure, DLL.
3. Infrastructure as a Service (IaaS) merupakan layanan untuk menggunakan infrastruktur yang telah disediakan. Jika kita analogikan seperti kita menyewa kamar kontrakan kosong yang bisa kita isi dalamnya sesuai dengan kebutuhan (sistem operasi). Sebagai contoh layanan IaaS adalah Amazon EC2, GCE, BizNetCloud, DLL.

## Separation of Responsibilities

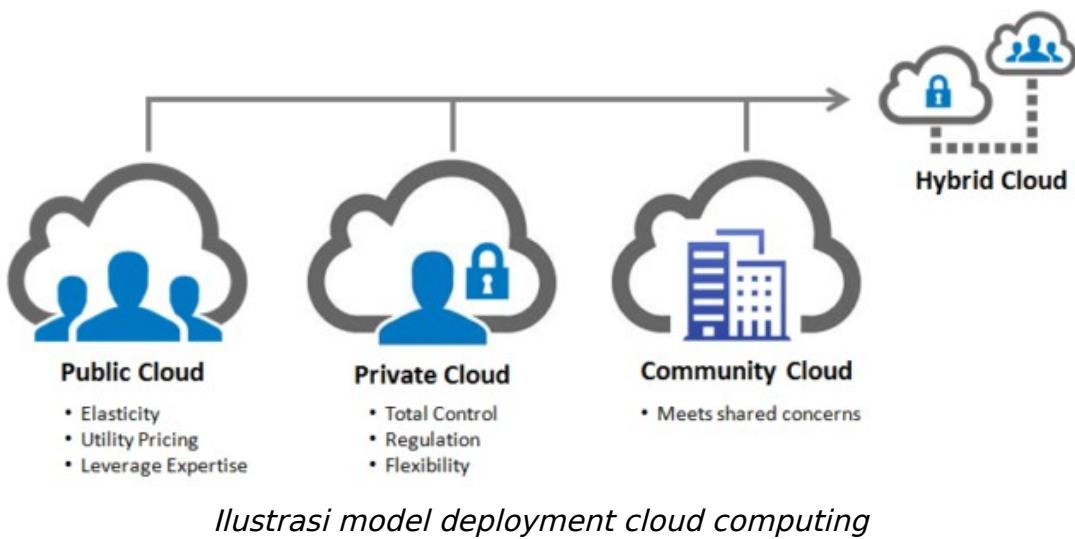


*Ilustrasi perbandingan model pengiriman cloud*

Berdasarkan model penyebaran (deployment) terapat 4 model deployment dalam Cloud Computing yang diantaranya adalah :

1. Public Cloud merupakan layanan Cloud Computing yang disediakan untuk masyarakat umum. Kita sebagai user tinggal mendaftar ataupun bisa langsung memakai layanan yang ada. Banyak layanan Public Cloud yang gratis, dan ada juga yang perlu membayar untuk bisa menikmati layanan-nya.
2. Private Cloud merupakan layanan Cloud Computing, yang disediakan untuk memenuhi kebutuhan internal dari organisasi/perusahaan. Biasanya departemen IT akan berperan sebagai Service Provider (penyedia layanan) dan departemen lain menjadi user (pemakai).
3. Hybrid Cloud merupakan gabungan dari layanan Public Cloud dan Private Cloud yang di-implementasikan oleh suatu organisasi/perusahaan. Dalam Hybrid Cloud ini, kita bisa memilih proses bisnis mana yang bisa dipindahkan ke Public Cloud dan proses bisnis mana yang harus tetap berjalan di Private Cloud.

4. Community Cloud dapat digunakan bersama-sama oleh beberapa perusahaan yang memiliki kesamaan kepentingan



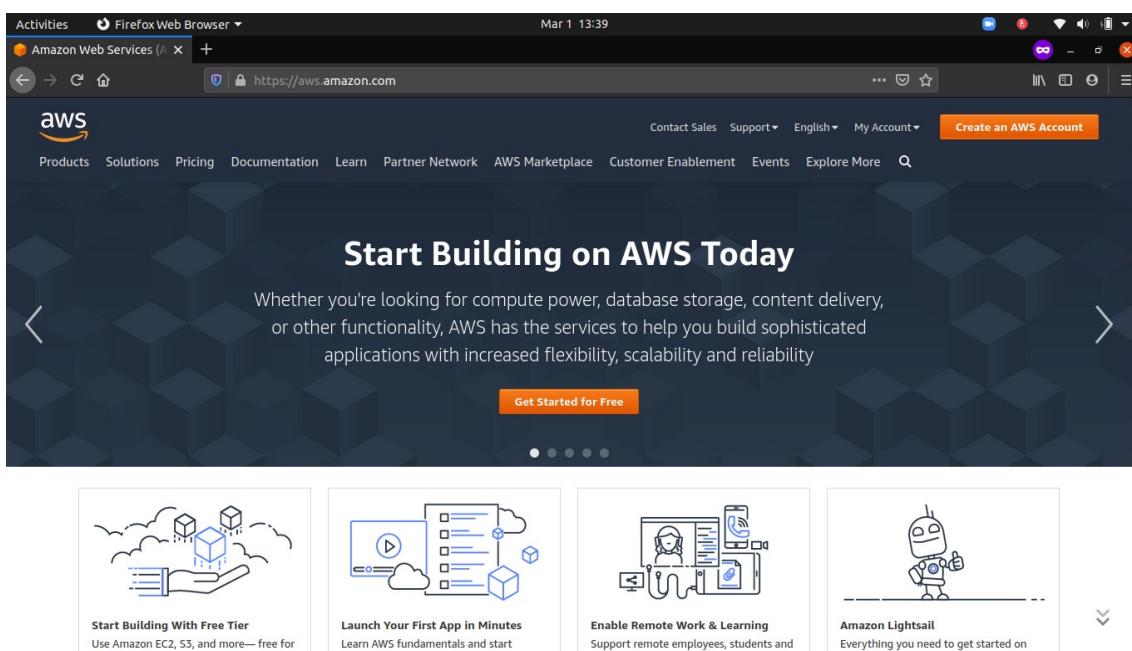
### 6.1.2. Apa itu AWS ?

Amazon Web Service (AWS) adalah platform layanan cloud yang menawarkan komputasi canggih, menyimpan database, mengirimkan konten, dan fungsionalitas lain untuk membantu dan mengembangkan bisnis. AWS ini merupakan penyedia layanan cloud paling awal dan sangat populer karena Amazon menyediakan layanan-layanan yang saling integrasi dan mudah untuk kustomisasi.

Amazon memperkenalkan layanan Amazon Elastic Cloud Compute (EC2) pada tahun 2006 sebagai commercial service yang menyediakan akses cloud kepada perusahaan dan individu untuk menyewa komputer yang bisa digunakan sebagai platform pengembangan aplikasi secara online dan inilah awal dari infrastruktur cloud.

AWS dapat diakses dimana saja dan kapan saja, inilah yang membuat AWS menjadi infrastruktur yang sangat fleksibel. Selain itu perusahaan-perusahaan akan lebih dimudahkan dalam melakukan jual beli dengan menggunakan AWS.

Untuk dapat menikmatinya kalian hanya perlu mengakses situs resminya yaitu <https://aws.amazon.com/>



Situs resmi AWS

AWS menawarkan pembayaran sesuai yang kita gunakan untuk pemberian harga layanan cloud. Dengan AWS kita hanya membayar layanan individual yang kita perlukan, selama kita menggunakannya, dan tanpa kontrak jangka panjang atau lisensi rumit. Pemberian harga AWS serupa dengan bagaimana kita membayar keperluan seperti air atau listrik. kita hanya membayar layanan yang digunakan, dan setelah kita berhenti menggunakan, tidak ada biaya tambahan atau biaya penghentian.

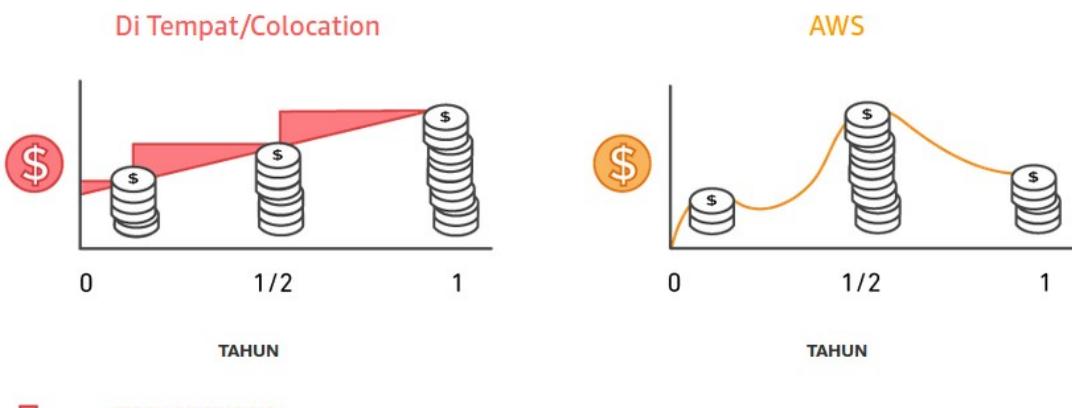
### 6.1.3. Sistem Pembayaran AWS

Dengan AWS Anda tidak perlu lagi mendedikasikan sumber daya berharga untuk membangun infrastruktur berbiaya mahal, termasuk pembelian server, lisensi perangkat lunak, atau menyewa fasilitas. Dengan AWS Anda dapat menggantikan pengeluaran di muka yang tinggi dengan biaya variabel lebih rendah dan hanya membayar untuk apa yang anda gunakan dan selama anda memerlukannya. Semua layanan AWS tersedia sesuai permintaan, dan tidak memerlukan kontrak jangka panjang serta ketergantungan lisensi yang rumit.

Pemberian harga bayar berdasarkan pemakaian memungkinkan Anda untuk beradaptasi dengan mudah dalam mengubah kebutuhan bisnis tanpa terlalu



banyak anggaran dan meningkatkan respons Anda terhadap perubahan. Dengan model bayar berdasarkan pemakaian, Anda dapat menyesuaikan bisnis tergantung pada keperluan dan bukan prediksi, mengurangi risiko, berlebihan ketentuan, atau kehilangan kapasitas.



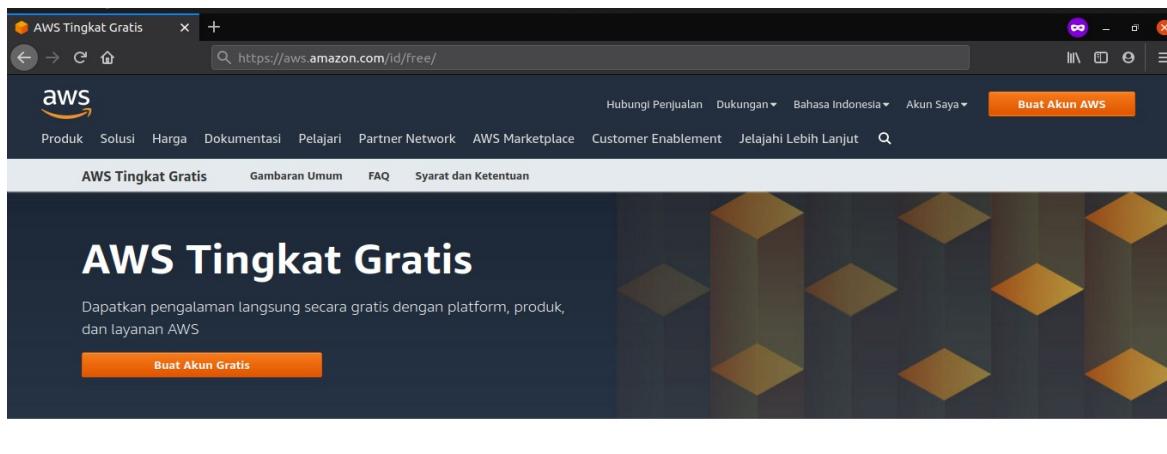
*Perbandingan sistem pembayaran AWS*

Dengan membayar untuk layanan berbasis keperluan, Anda dapat mengalihkan fokus pada inovasi dan penemuan, mengurangi kerumitan pengadaan, dan memungkinkan bisnis Anda menjadi elastis.

#### 6.1.4. Membuat Akun AWS

Membuat akun AWS pada dasarnya sama saja dengan membuat akun di situs-situs lainnya, hanya saja kita perlu memasukan kartu kredit atau debit untuk jaminan pembayaran layanan yang akan kita gunakan. AWS juga menyediakan sebuah layanan Free Tier atau layanan gratis selama 1 tahun yang dapat kita nikmati dengan syarat dan ketentuan berlaku.

Untuk informasi lebih lengkap mengenai layanan Free Tier dari AWS, kalian dapat mengakses situs resminya langsung di <https://aws.amazon.com/id/free/>



### Jenis penawaran

Jelajahi lebih dari 85 produk dan mulai membangun di AWS menggunakan tingkat gratis. Tersedia tiga jenis penawaran gratis yang berbeda tergantung produk yang digunakan. Lihat detail di bawah ini pada setiap produk.



*Situs utama AWS*



Sedangkan untuk membuat akun AWS baru, kita hanya perlu masuk ke situs <https://portal.aws.amazon.com/billing/signup> dan mengisi form registrasi yang sudah disediakan.

*Form registrasi akun baru AWS*

Setelah selesai melakukan pendaftaran, nanti akan muncul konfirmasi ke email dan coba untuk melakukan konfirmasi pada link yang sudah diberikan pada email tersebut.

Untuk melakukan pendaftaran akun AWS, kita memerlukan kartu Debit/Kredit internasional. Penulis menyarankan untuk menggunakan kartu debit Jenius. Selain tidak ada bunga dan biaya administrasi bulanan, kartu debit Jenius juga memiliki banyak fitur canggih dan mudah dibuat. Pembuatan kartu debit Jenius tidak sesulit kartu kredit yang susahnya minta ampun.



Kartu Debit Jenius BTPN

Setelah membuat akun jenius dan telah teraktivasi, langkah selanjutnya kita bisa langsung mendaftar akun AWS menggunakan kartu Jenius kita. Saran dari kami, **gunakan virtual card** yang bisa kita atur saldoanya, karena ditakutkan apabila Anda menggunakan kartu utama yang bisa jadi Anda gunakan untuk keperluan yang lain juga, nantinya saldo tabungan Anda akan tersedot secara tidak sadar.

### 6.1.5. Login Akun AWS

Setelah selesai membuat akun aws, sekarang kita coba untuk login pada akun AWS di alamat <https://aws.amazon.com/> lalu klik **Create an AWS Account** atau klik **My Account** lalu klik **AWS Management Console**.



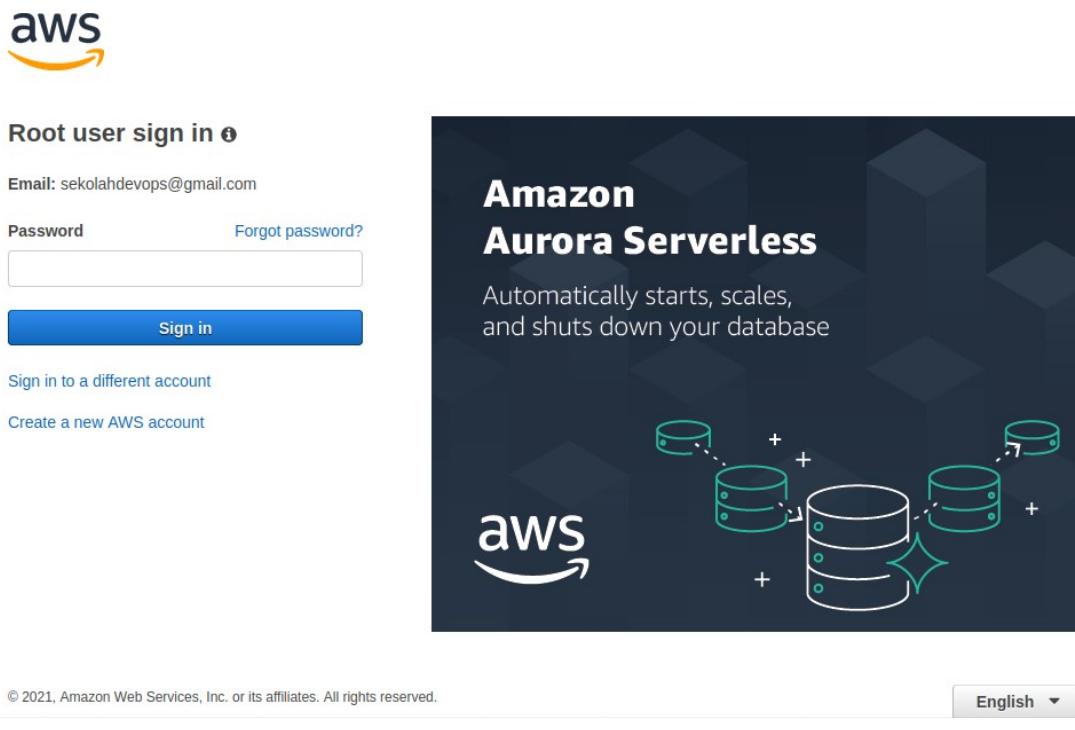
*Sign up menu aws*

Setelah itu masukan alamat email dari akun yang kalian buat.

The image shows the AWS sign-in page. It features the AWS logo at the top left. Below it, there are two radio button options: "Root user" (selected) and "IAM user". The "Root user" option is described as an "Account owner that performs tasks requiring unrestricted access". The "IAM user" option is described as "User within an account that performs daily tasks". Below these options is a field labeled "Root user email address" containing the placeholder "username@example.com". A blue "Next" button is positioned below this field. At the bottom of the sign-in form, there is a link to the AWS Customer Agreement and Privacy Notice, followed by a "New to AWS?" link and a "Create a new AWS account" button. To the right of the sign-in form, there is an advertisement for "Amazon Aurora Serverless". The ad features the Amazon logo and the text "Amazon Aurora Serverless" with the subtext "Automatically starts, scales, and shuts down your database". It includes a diagram of a database cluster with multiple cylinders and plus signs indicating scaling.

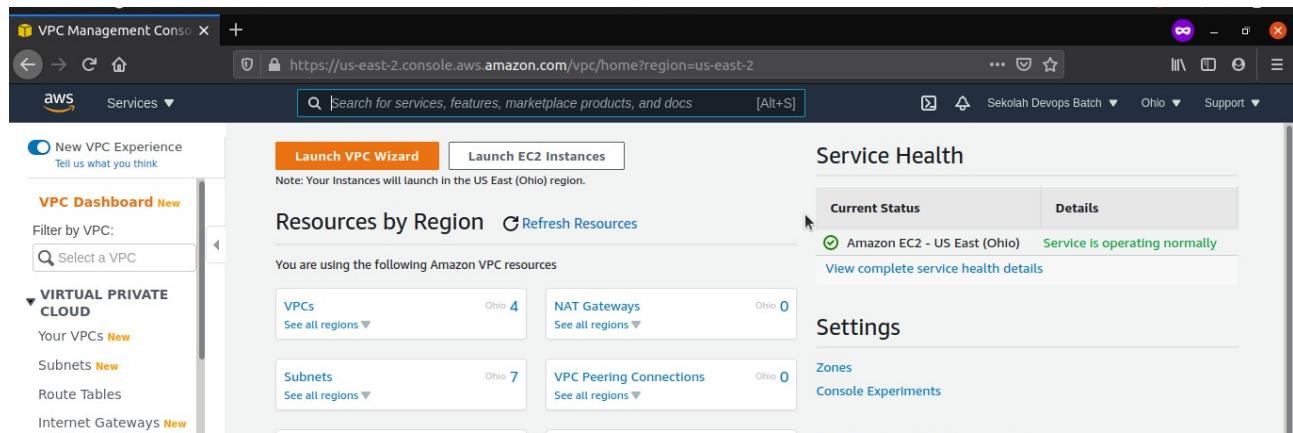
*Login form AWS*

Lalu masukan password



*Login form AWS*

Setelah login kita akan diarahkan ke dasboard console milik AWS.



*Dashboard Management Console AWS*

Dengan begini kalian sudah dapat menikmati layanan AWS ditambah dengan Free Tier selama satu tahun. Akan tetapi jika melebihi batas persyaratan free tier kita harus membayar sendiri kelebihan tersebut.

## 6.1.6. Perbedaan AWS dengan Cloud Sejenisnya

Infrastruktur cloud merupakan bagian yang sangat populer untuk saat ini, tidak hanya Amazon yang menyediakan layanan tersebut, banyak dari vendor lain yang menyediakan pula infrastruktur cloud yang diantaranya adalah Google Cloud Platform (GCP) dan Microsoft Azure. Ketiga provider tersebut merupakan pemain besar diantara penyedia layanan cloud lainnya dan ketiganya juga menyediakan layanan trial/percobaan bagi pengguna baru, sehingga kita bisa mencoba terlebih dahulu ketiga layanan tersebut.

Meskipun begitu, ketiganya memiliki beberapa perbedaan dan keunggulannya masing masing yang bisa kita uraikan sebagai berikut.



*Logo Google Cloud Platform*



*Logo Microsoft Azure*

### 6.1.6.1. Harga

Harga merupakan sebuah permasalahan yang paling krusial, mengingat hal ini menjadi salah satu pertimbangan yang dijadikan pedoman dalam membeli sebuah infrastruktur cloud. Dari ketiga platform tersebut kita akan coba membahas mengenai skema trial yang diberikan.

- AWS menyediakan percobaan gratis pada beberapa layanannya selama satu tahun. Tidak semua layanan bisa kita gunakan seperti Amazon EC2 yang hanya bisa menggunakan type t2.micro dengan spesifikasi server 1

vCPU, 1GB RAM, dan Elastic Storage. Layanan ini cukup memumpuni untuk menghandle aplikasi skala kecil.

- GCP menyediakan percobaan gratis dengan skema memberikan credits sebanyak \$300 yang dapat digunakan pada layanan apapun pada GCP tersebut. Credits tersebut dapat digunakan dalam jangka 12 Bulan. Dengan credit tersebut kita bisa membuat sebuah Google Compute Enginer (Seperti EC2 AWS) seharga \$30/bulan untuk server 1 vCPU,n 4GB RAM, dan 10GB Storage.
- Azure menyediakan percobaan gratis dengan skema yang menyerupai GCP, yaitu memberikan kredit sebesar \$200 yang dapat digunakan pada semua layanan azure. Credit ini hanya bisa digunakan dalam jangka waktu 30 Hari. Dengan credit tersebut kita bisa membuat VM dengan harga \$60 untuk server 1 vCPU, 4GB RAM, dan 16GB Storage.

Jika kita bandingkan ketiga layanan VM tersebut pada setiap platfrom dengan spesifikasi server 1 vCPU, 4GB RAM, dan Storage >10GB. Maka dapat kita lihat perbandingannya seperti berikut.

Layanan	AWS EC2	GCE	Azure VM
1 vCPU, 4GB RAM, dan Storage > 10GB	\$33,4/month (t2.medium)	\$24,3/month (n1-standard)	\$50/month (B2S)

### ***6.1.6.2. Layanan***

Hal lain yang menjadi perhatian adalah layanan, karena semakin baik layanan yang diberikan oleh platfrom tersebut maka akan semakin nyaman juga customer menggunakannya. Kita akan coba bahas beberapa layanan yang ada pada ketiganya menurut sumber yang kami kumpulkan.

- Salah satu layanan yang menarik dari AWS adalah penyelenggaraan Web Training Tutorial yang dapat kita akses secara gratis di



<https://aws.amazon.com/training/>. Kita dapat mempelajari AWS dari sana sesuai dengan Sertifikasinya, Selain itu ketersediaan dokumentasi pengguna AWS yang lengkap dan apik memudahkan kita untuk mengatasi masalah-masalah yang kita alami pada saat menggunakan.

- GCP sendiri sering mengadakan event-event, seperti sebelumnya yang pernah diselenggarakan adalah Google Cloud Summit Jakarta. GCP juga menyediakan dokumentasi yang mudah untuk diperoleh, akan tetapi menurut sumber yang kami dapat, beberapa dokumentasi ini tidak cukup baik seperti halnya AWS.
- Azure sendiri menurut sumber yang didapat belum menemukan layanan yang menarik baik telpon maupun email. Secara dokumentasi azure memiliki kualitas dokumentasi yang tidak jauh dengan GCP.

Maka jika kita simpulkan layanan dan dokumentasi dari ketiga platform tersebut yang paling unggul adalah AWS dengan layanan terbaiknya.

#### **6.1.6.3. Pilihan Stack**

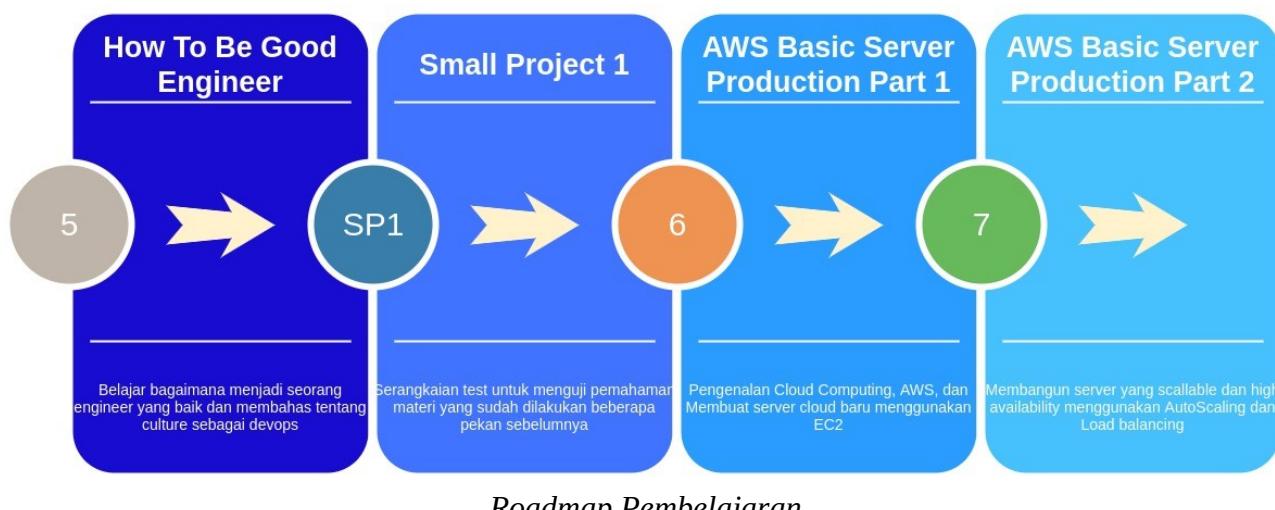
Pada bagian ini yang akan kita bahas berkaitan dengan region server. Karena semakin banyak region semakin banyak pula pilihan yang bisa digunakan.

- AWS memiliki banyak pilihan region diantara yang lainnya yaitu 14 region dengan region yang paling dekat dengan indonesia adalah region singapura.
- GCP sendiri memiliki pilihan region sebanyak 10 region dengan region yang paling terdekat dengan indonesia adalah region singapura.
- Azure memiliki jumlah region sebanyak 11 region dengan region paling terdekat yaitu masih di region singapura.

### 6.1.7. Exercise

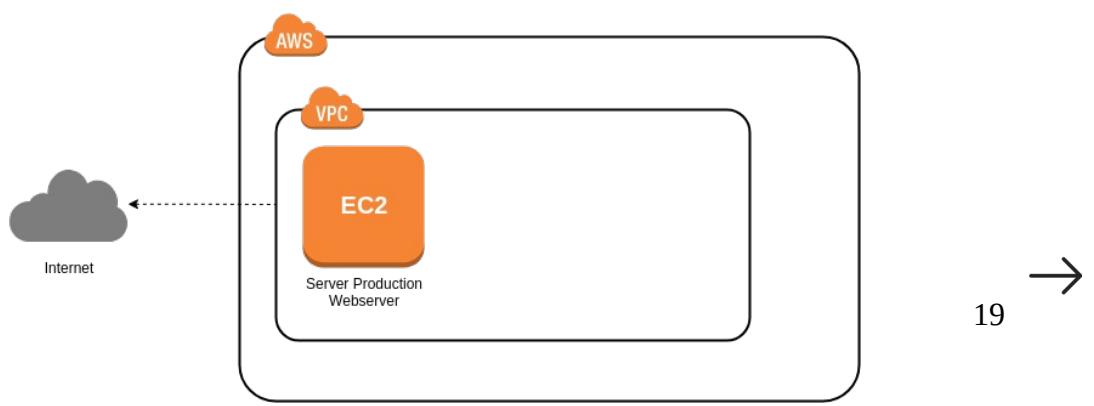
1. Jelaskan perbedaan dari ketiga model pengiriman dalam cloud computing !
2. Apa perbedaan antara Public Cloud dan Private Cloud ?

## 6.2. Roadmap Pembelajaran



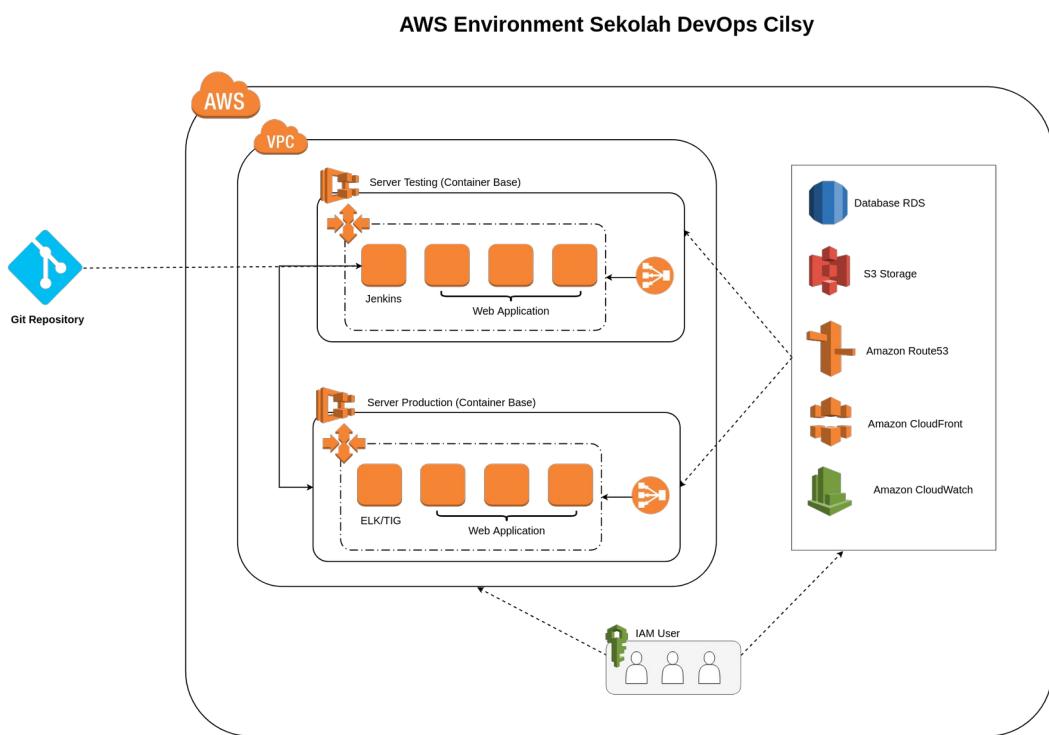
Jika kita lihat pada roadmap yang sudah kita bahas sebelumnya, kita kemarin sudah melewati **tahap pertama sampai dengan tahap ketiga** yaitu **pengenalan Git**. Sekarang kita masuk **tahap keempat** yaitu **Membuat Webserver di Production Server EC2**

Berdasarkan topologinya sekarang, kita akan membangun infrastructure didalam layanan AWS menggunakan Service Amazon VPC dan EC2 seperti berikut.



*Roadmap Topologi VPC dan EC2*

Topologi diatas merupakan sebagian dari topologi total yang akan kita buat pada Platfrom AWS. Dimana nanti sedikit demi sedikit yang sudah kita buat akan saling berintegrasi dan membuat infrastructure yang sempurna seperti hasil akhir dibawah ini.



## 6.3. Amazon VPC

### 6.3.1. Apa itu Amazon VPC ?

Konsep dari virtual network yang diberikan tidak berbeda jauh dengan jaringan **on premise** pada biasanya. Kita diberikan kendali penuh atas konfigurasi jaringan didalamnya termasuk pemilihan range alamat IP, pembuatan subnet, dan konfigurasi tabel routing serta gateway jaringan. Kita dapat menggunakan seluruh sumber daya yang ada pada Amazon VPC.

Amazon VPC menawarkan sistem keamanan yang berlapis yang diantaranya adalah kemampuan untuk mengijinkan dan menolak trafik dari internet dan jaringan internal AWS secara spesifik. Selain itu Amazon VPC memungkinkan service AWS lainnya untuk melakukan deploy didalam VPC tersebut karena setiap service di amazon sendiri dibangun dalam sebuah jaringan dengan sistem keamanan yang sudah disertakan dalam pembuatannya.



*Logo Virtual Private Cloud*

### 6.3.2. Service AWS yang dapat deploy di VPC

Dari sekian service yang ada di Amazon Web Service, berikut merupakan beberapa diantaranya yang dapat deploy pada service Amazon VPC diantaranya sebagai berikut.



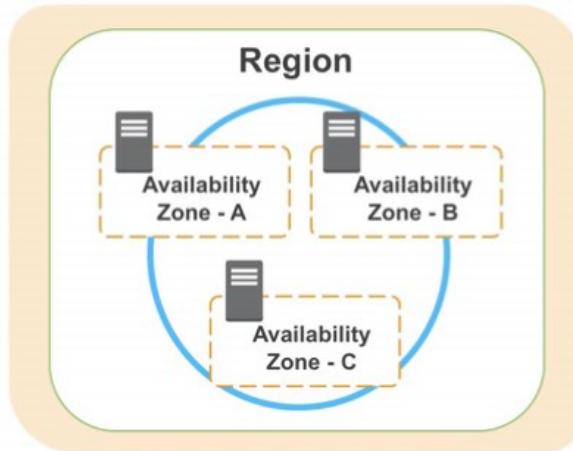
*Ilustrasi layanan yang dapat deploy di AWS*

### 6.3.3. Fitur Amazon VPC

Dalam penggunaannya, layanan ini memiliki beberapa fitur yang bisa kita nikmati dan gunakan, berikut beberapa fitur yang ada pada amazon VPC.

#### 6.3.3.1. Ketersediaan *high availability* dan Availability Zone pada Region AWS

Amazon VPC sendiri dapat diakses di semua region, dalam satu region kita dapat membuat beberapa availability zone dan kita juga dapat membuat multiple VPC per akun, dimana dalam satu akun kita bisa membuat lebih dari satu VPC.



*Ilustrasi Availability Zone dalam satu buah region*

#### 6.3.3.2. Subnet

Subnet yang ada pada Amazon VPC digunakan untuk membagi IP address pada VPC, subnet juga mengijinkan amazon VPC untuk melakukan multiple span Availability Zone.

#### 6.3.3.3. Route Tables

Routing tabel disini berfungsi untuk mengontrol trafik yang pergi dan keluar melalui subnets.

#### **6.3.3.4. Internet Gateways (IGW)**

Internet Gateway berfungsi untuk mengijinkan amazon VPC mengakses internet, selain itu kita juga bisa mematikan internet pada VPC tertentu.

#### **6.3.3.5. NAT Gateways**

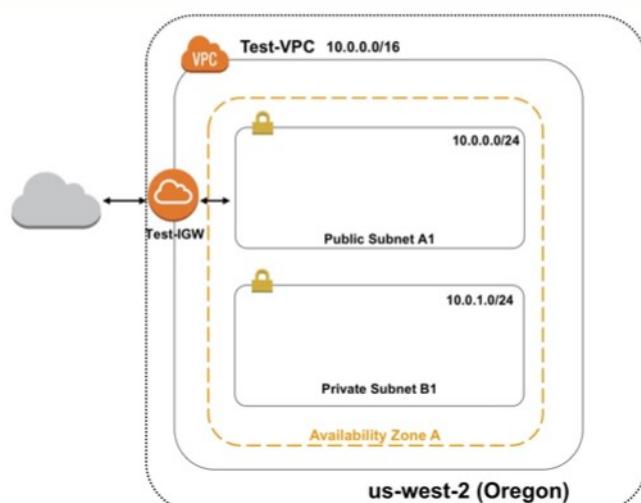
NAT berfungsi untuk mengijinkan private subnet untuk melakukan akses ke jaringan internet, dengan begini ip private tersebut akan dianggap sebagai ip address.

#### **6.3.3.6. Network Access Control Lists (NACL)**

Bagian access control list ini berfungsi untuk mengontrol akses ke subnet yang bukan dari bagian VPC.

### **6.3.4. Contoh Skema Amazon VPC**

Pada bagian ini kita akan melihat salah satu skema penggunaan VPC pada implementasinya, disini kita bisa lihat skema dari sebuah kerangka VPC dibawah ini.



*Skema Penggunaan Amazon VPC*

Pada gambar skema diatas bisa kita lihat bahwa kita sudah membuat sebuah VPC baru dengan nama Test-VPC yang memiliki address 10.0.0.0/26. Selanjutnya kita membuat dua buah subnet yang diantaranya satu public

subnet dan satunya lagi adalah private subnet. Keduanya masuk kedalam satu wilayah availability Zone.

Public Subnet sendiri diberikan akses dengan menggunakan fitur Internet Gateway dengan nama konfigurasi Test-IGW sehingga bagian subnet public bisa mengakses ke jaringan internet. Sedangkan priaate subnet sendiri tidak diberikan akses ke internet.

### 6.3.5. Exercise

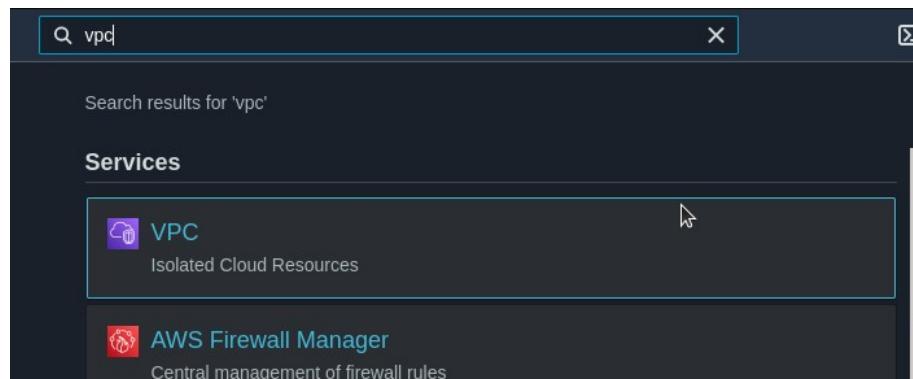
1. Jelaskan kembali apa yang dimaksud dengan VPC !
2. Buat sebuah skema VPC dengan beberapa subnet dan ceritakan skemanya !

## 6.4. Praktik VPC - Virtual Private Cloud

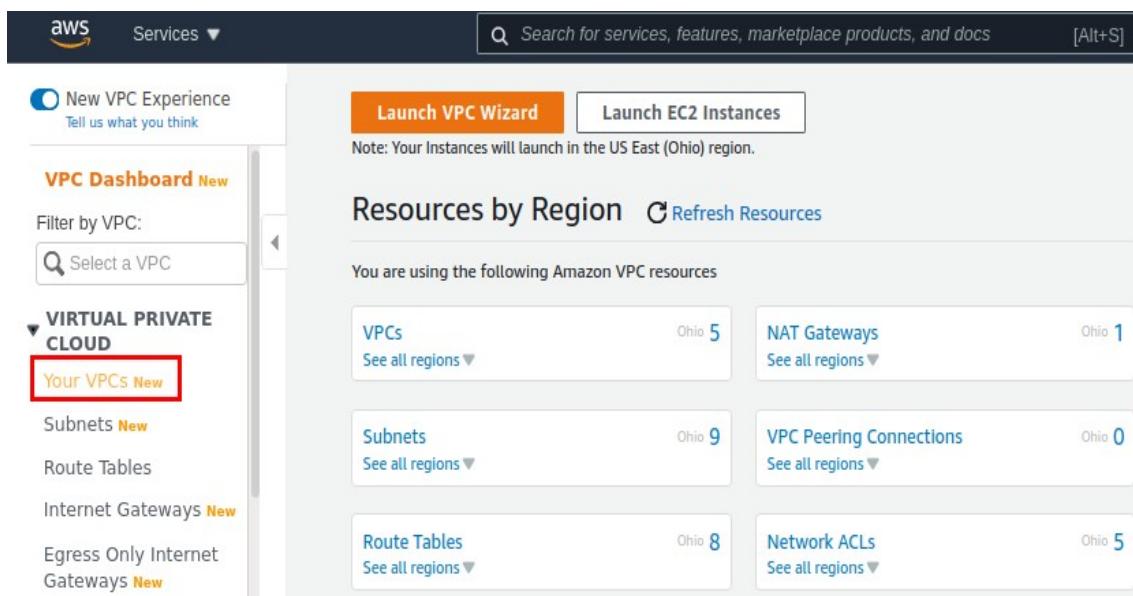
Di materi sebelumnya kita sudah mempelajari VPC dan beberapa komponen yang ada di dalamnya. Selanjutnya kita akan coba untuk mempraktekan bagaimana membuat sebuah VPC dan komponen-komponennya.

### 6.4.1. Membuat VPC

Untuk memulai membuat sebuah **VPC**, kita harus login terlebih dahulu ke akun AWS. Setelah itu buka menu service lalu ketikkan VPC untuk menampilkan menu nya. Setelah itu klik pada menu VPC yang tampil



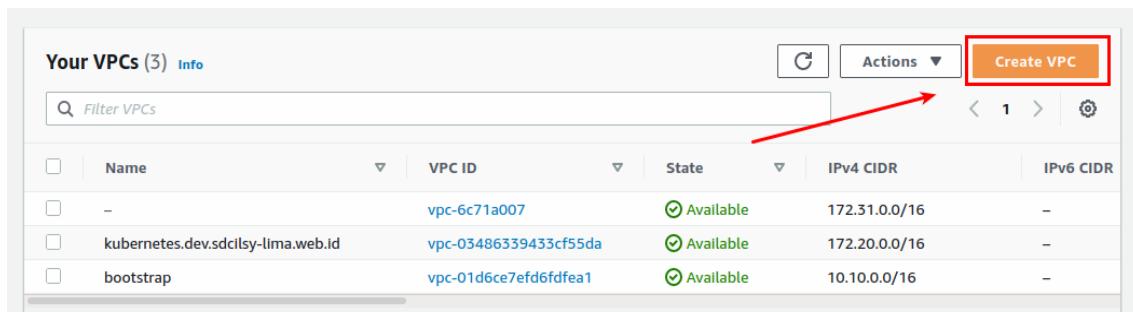
Setelah itu kita akan ditampilkan **dashboard VPC**, kita masuk dulu ke menu **Your VPCs**.



The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with options like 'New VPC Experience' (radio button selected), 'VPC Dashboard New', 'Filter by VPC' (with a dropdown for 'Select a VPC'), and a 'VIRTUAL PRIVATE CLOUD' section with 'Your VPCs New' (highlighted with a red box). Other items in this section include 'Subnets New', 'Route Tables', 'Internet Gateways New', and 'Egress Only Internet Gateways New'. At the top right, there are two buttons: 'Launch VPC Wizard' (orange) and 'Launch EC2 Instances'. Below these buttons, a note says 'Note: Your Instances will launch in the US East (Ohio) region.' The main area is titled 'Resources by Region' with a 'Refresh Resources' button. It lists Amazon VPC resources across the Ohio region:

Resource Type	Count	Region
VPCs	5	Ohio
NAT Gateways	1	Ohio
Subnets	9	Ohio
VPC Peering Connections	0	Ohio
Route Tables	8	Ohio
Network ACLs	5	Ohio

Selanjutnya akan tampil menu seperti di bawah ini, lalu kita klik **Create VPC** untuk membuat sebuah **Virtual Private Cloud** baru.



The screenshot shows the 'Your VPCs' list page. At the top, it says 'Your VPCs (3) Info' and has a 'Actions' dropdown and a 'Create VPC' button (highlighted with a red box). Below is a table with columns: Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR. The table contains three rows of data:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-6c71a007	Available	172.31.0.0/16	-
kubernetes.dev.sdclsy-lima.web.id	vpc-03486339433cf55da	Available	172.20.0.0/16	-
bootstrap	vpc-01d6ce7efd6fdf1	Available	10.10.0.0/16	-

Setelah itu kita tentukan **subnet** yang akan kita buat, disini saya menggunakan **subnet 10.0.0.0/16** dengan Name tag **Cilsy VPC**. Lalu klik **Create** untuk membuat.



**Create VPC** Info

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances.

**VPC settings**

Name tag - *optional*  
Creates a tag with a key of 'Name' and a value that you specify.

Cilsy VPC

IPv4 CIDR block Info

10.0.0.0/16

IPv6 CIDR block Info

No IPv6 CIDR block  
 Amazon-provided IPv6 CIDR block  
 IPv6 CIDR owned by me

Tenancy Info

Default

**Tags**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - <i>optional</i>
<input type="text"/> Name	<input type="text"/> Cilsy VPC <span style="border: 1px solid black; padding: 2px;">X</span> <span style="border: 1px solid black; padding: 2px;">Remove</span>

Add new tag

You can add 49 more tags.

Cancel Create VPC

Setelah itu akan muncul **VPC** yang baru. Seperti dibawah ini.

Search for services, features, marketplace products, and docs [Alt+S]

You successfully created vpc-0cc476711abe17ac8 / Cilsy VPC

VPC > Your VPCs > vpc-0cc476711abe17ac8

vpc-0cc476711abe17ac8 / Cilsy VPC Actions ▾

Details		Actions	
VPC ID	vpc-0cc476711abe17ac8	State	Available
Tenancy	Default	DNS hostnames	Disabled
Default VPC	No	DNS resolution	Enabled
Owner ID	898130718046	IPv4 CIDR	10.0.0.0/16
		IPv6 pool	-
		Main route table	rtb-07a34a47b16f02b81
		Main network ACL	acl-0c1f86990816719fc

Your VPCs (4) [Info](#)

Actions ▾ [Create VPC](#)

Filter VPCs

< 1 >

<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	-	vpc-6c71a007	Available	172.31.0.0/16	-
<input checked="" type="checkbox"/>	Cilsy VPC	vpc-0cc476711abe17ac8	Available	10.0.0.0/16	-
<input type="checkbox"/>	kubernetes.dev.sdcilsy-lima.web.id	vpc-03486339433cf55da	Available	172.20.0.0/16	-
<input type="checkbox"/>	bootstrap	vpc-01d6ce7efd6fdf1ea1	Available	10.10.0.0/16	-

Selanjutnya kita akan aktifkan hostname pada VPC tersebut, agar ketika kita membuat sebuah instance baru maka akan otomatis dibuatkan DNS oleh si AWS langsung. Caranya klik kanan pada VPC yang sudah kita buat lalu pilih **Edit DNS hostnames**

Your VPCs (1/4) [Info](#)

Actions ▾ [Create VPC](#)

Filter VPCs

<input type="checkbox"/>	Name	VPC ID	State	IP	Actions
<input type="checkbox"/>	-	vpc-6c71a007	Available	10.0.0.0/16	
<input checked="" type="checkbox"/>	Cilsy VPC	vpc-0cc476711abe17ac8	Available	10.0.0.0/16	
<input type="checkbox"/>	kubernetes.dev.sdcilsy-lima.web.id	vpc-03486339433cf55da	Available	172.20.0.0/16	
<input type="checkbox"/>	bootstrap	vpc-01d6ce7efd6fdf1ea1	Available	10.10.0.0/16	

Selanjutnya centang bagian enable, lalu save perubahan yang sudah dilakukan.

Edit DNS hostnames [Info](#)

**DNS hostnames**  
Indicates whether instances with public IP addresses get corresponding public DNS hostnames.

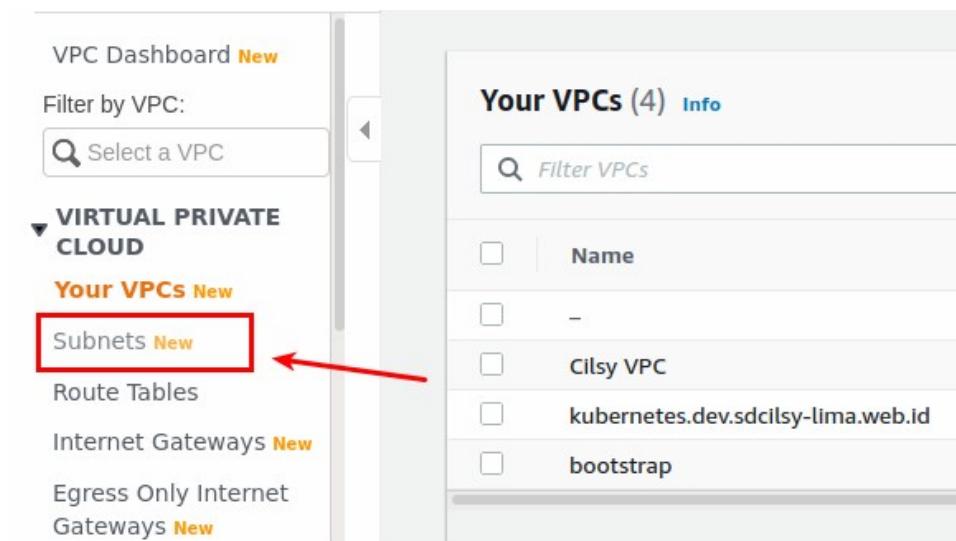
VPC ID <input type="text"/> vpc-0cc476711abe17ac8	DNS hostnames <input checked="" type="checkbox"/> Enable
--	---

[Cancel](#) [Save changes](#)

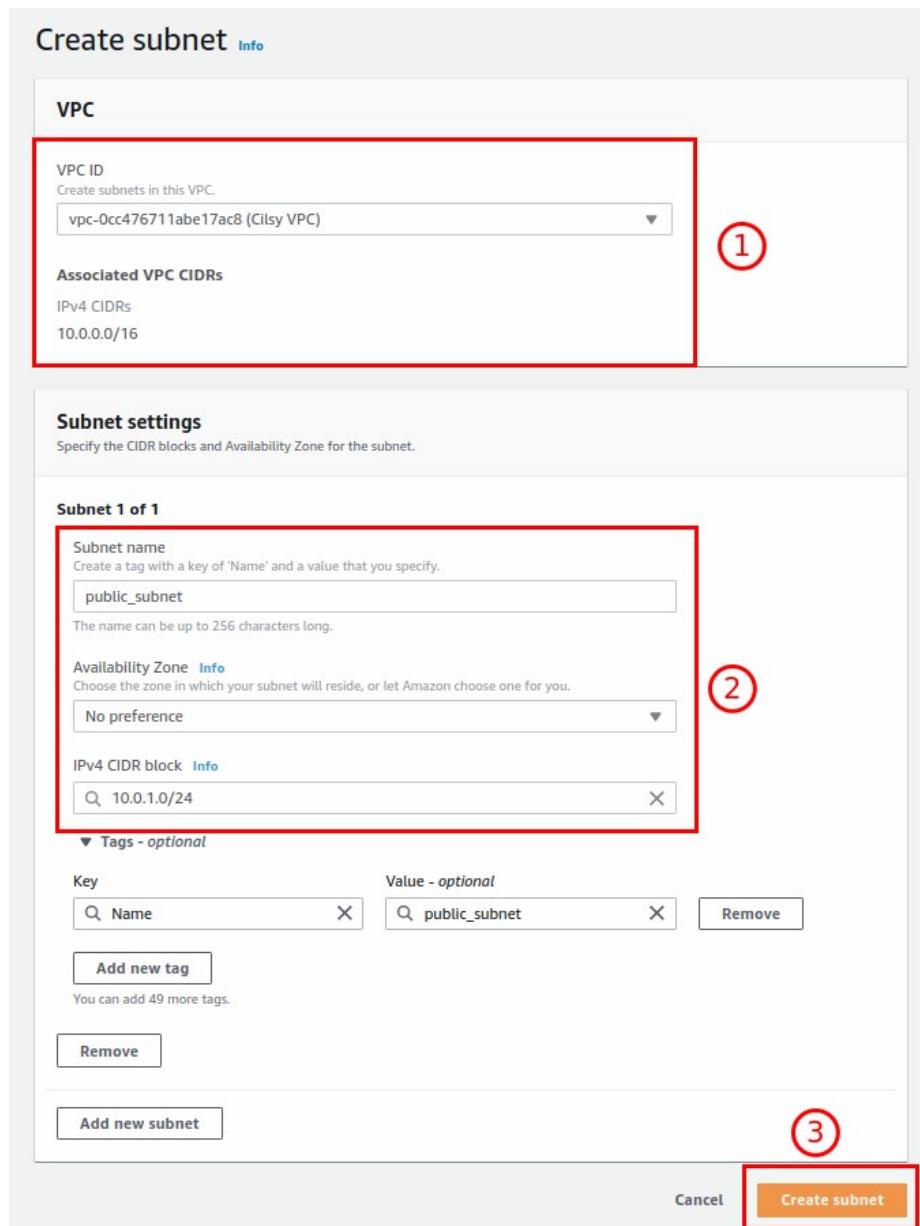
Begitulah cara untuk membuat **VPC** atau isolated network untuk **EC2** kita. Selanjutnya akan dibahas mengenai komponen VPC lainnya untuk memecah network kita menjadi lebih kecil, yaitu Subnet.

### 6.4.2. Membuat Subnet

Setelah selesai membuat **VPC**, kita beralih ke **subnet**, klik **Subnet** di bawah menu **VPC**



Kita lihat sebetulnya sudah ada **subnet default** yang menempel pada **VPC default**. Untuk praktek kali ini kita akan membuat **subnet** sendiri yang akan kita tempelkan pada **Cilsy VPC**. Klik **Create subnet** maka kita akan diberi setting seperti berikut.



Setelah itu kita click **Create**, maka **subnet** akan dibuat oleh AWS dan akan muncul di **dashboard Subnet** kita. Ada pilihan lebih lanjut untuk setting : **auto assign IP address**, bisa dipilih dengan mengklik **Actions** lalu '**Modify auto-assign IP settings**'.

Subnets (6) <a href="#">Info</a>					
	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	bootstrap-sub	subnet-022af30360ea94f3b	<span>Available</span>	vpc-01d6ce7efd6fdfea1   boot...	10.10.10.0/24
<input type="checkbox"/>	-	subnet-d0aa52bb	<span>Available</span>	vpc-6c71a007	172.31.0.0/20
<input type="checkbox"/>	public_subnet	subnet-07f03a6b5e4ae8d	<span>Available</span>	vpc-0cc476711abe17ac8   Cils...	10.0.1.0/24
<input type="checkbox"/>	-	subnet-ca4ed086	<span>Available</span>	vpc-6c71a007	172.31.32.0/20
<input type="checkbox"/>	us-east-2a.kuberne...	subnet-0c01bd1d54706da75	<span>Available</span>	vpc-03486339433cf55da   ku...	172.20.32.0/19
<input type="checkbox"/>	-	subnet-753d0b0f	<span>Available</span>	vpc-6c71a007	172.31.16.0/20

Setelah **public subnet** berhasil, kita akan membuat **private subnet**, langkah-langkahnya sama seperti sebelumnya, tapi kita harus menghitung *network address* selanjutnya yang akan dipakai.

### Create subnet [Info](#)

**VPC**

VPC ID  
Create subnets in this VPC.

**Associated VPC CIDRs**

IPv4 CIDRs  
10.0.0.0/16

**Subnet settings**

Specify the CIDR blocks and Availability Zone for the subnet.

**Subnet 1 of 1**

**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 CIDR block [Info](#)**

**Tags - optional**

Key	Value - optional	Remove
<input style="width: 100%;" type="text" value="Name"/>	<input style="width: 100%;" type="text" value="private_subnet"/>	<input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>		
You can add 49 more tags.		

(1)
(2)
(3)

30

→

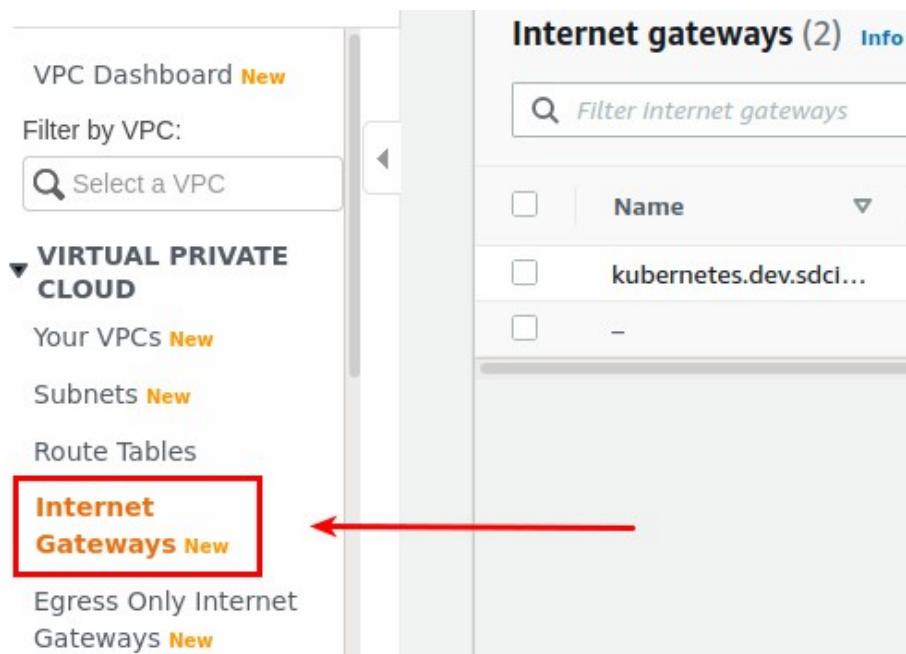
Bab 6

kita lihat bahwa **public dan private subnet** sudah terbentuk.

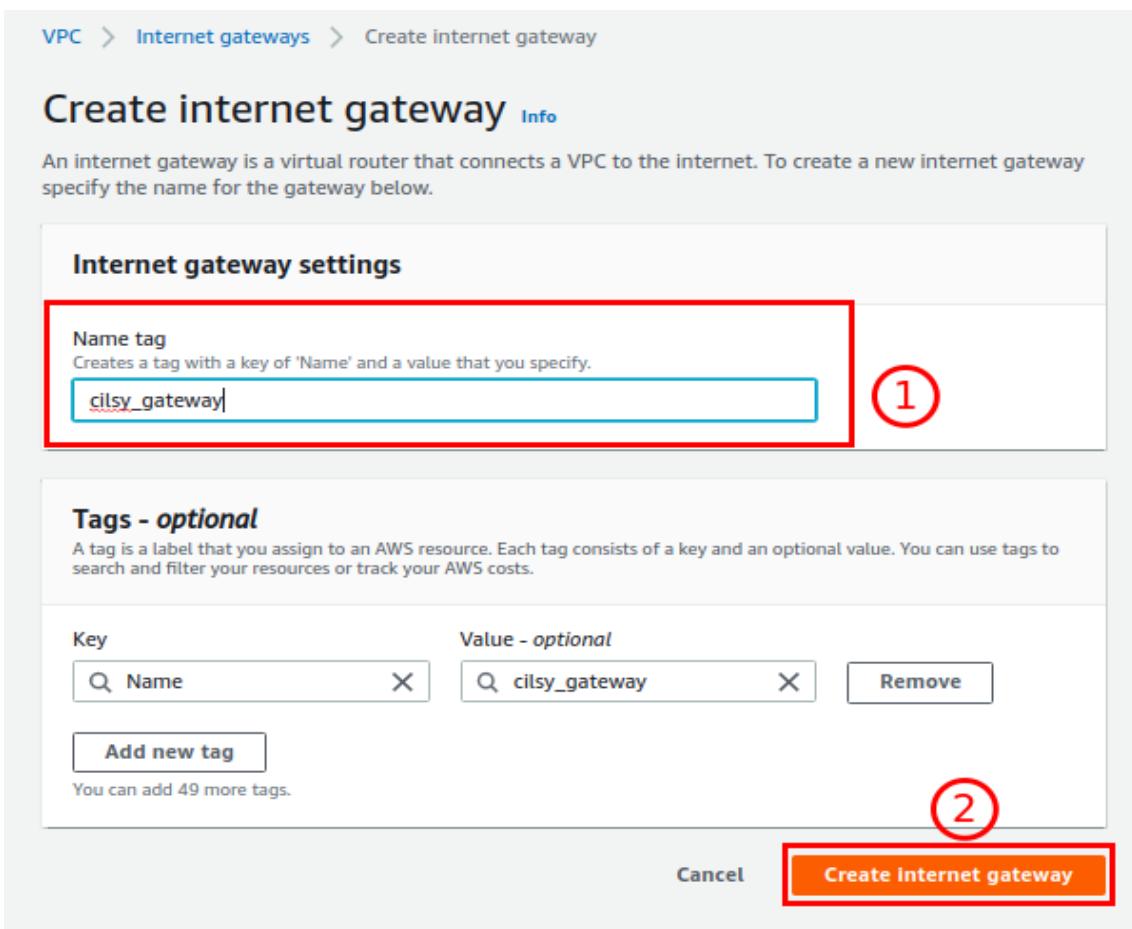
Subnets (7) <a href="#">Info</a>						
	Name	Subnet ID	State	VPC	IPv4 CIDR	
<input type="checkbox"/>	bootstrap-sub	subnet-022af30360ea94f3b	<span>Available</span>	vpc-01d6ce7efd6fdfea1   boot...	10.10.10.0/24	
<input type="checkbox"/>	-	subnet-d0aa52bb	<span>Available</span>	vpc-6c71a007	172.31.0.0/20	
<input type="checkbox"/>	public_subnet	subnet-07f03a6b5e4aede8d	<span>Available</span>	vpc-0cc476711abe17ac8   Cils...	10.0.1.0/24	
<input type="checkbox"/>	-	subnet-ca4ed086	<span>Available</span>	vpc-6c71a007	172.31.32.0/20	
<input type="checkbox"/>	us-east-2a.kuberne...	subnet-0c01bd1d54706da75	<span>Available</span>	vpc-03486339433cf55da   ku...	172.20.32.0/19	
<input type="checkbox"/>	-	subnet-753d0b0f	<span>Available</span>	vpc-6c71a007	172.31.16.0/20	
<input type="checkbox"/>	private_subnet	subnet-04a3a8786cccd51dc1	<span>Available</span>	vpc-0cc476711abe17ac8   Cils...	10.0.2.0/24	

### 6.4.3. Internet Gateways

Internet Gateway berfungsi untuk memberikan koneksi internet kepada **VPC**. **Default internet gateway** yang tersedia, menempel pada **VPC default**. Jadi VPC yang kita buat tadi, belum bisa mengakses **internet** karena kita belum membuat **gateway** ini. Untuk membuatnya, klik **Internet Gateways** di bawah **Subnet**.



Untuk **gateway** ini, kita tinggal memberikan nama saja, yang akan mengatur keseluruhannya adalah AWS, jadi kita sama sekali tidak memanage **Internet Gateways**. Klik **Create internet gateway** dan isi seperti dibawah ini.



The screenshot shows the 'Create internet gateway' wizard. The path is VPC > Internet gateways > Create internet gateway. The first step, 'Internet gateway settings', has a red box around the 'Name tag' input field containing 'cilsy\_gateway'. A red circle with the number 1 is to the right of the input field. The second step, 'Tags - optional', shows a single tag 'Name: cilsy\_gateway' with a red circle with the number 2 below it. The final step shows the 'Create Internet gateway' button highlighted with a red box.

VPC > Internet gateways > Create internet gateway

## Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

### Internet gateway settings

**Name tag**  
 Creates a tag with a key of 'Name' and a value that you specify.

cilsy\_gateway 1

### Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="cilsy_gateway"/> <span style="border: 1px solid red; border-radius: 50%; padding: 2px;">2</span>

Add new tag 2  
 You can add 49 more tags.

2 2

2 2

Cancel Create Internet gateway

Setelah itu klik **Create**. Tapi tidak berhenti sampai disitu, kita harus menempelkan **gateway** tersebut ke **VPC** yang tadi kita buat. Untuk melakukannya, **klik di gateway** yang kita buat, lalu **klik Actions**, dan pilih **Attach to a VPC**.



The screenshot shows a confirmation message: "The following internet gateway was created: igw-0795b1b25f798ec46. You can now attach to a VPC to enable the VPC to communicate with the internet." A red box highlights the "Attach to a VPC" button.

**igw-0795b1b25f798ec46 / cilsy\_gateway**

**Actions ▾**

**Details** Info

Internet gateway ID  
 Igw-0795b1b25f798ec46

**Internet gateways (1/3)** Info

Name	Internet gateway ID	State	Action
<input checked="" type="checkbox"/> cilsy_gateway	Igw-0795b1b25f798ec46	Attached	<span style="color: red;">②</span> <b>Actions ▾</b> <span style="color: red;">③</span> <b>Attach to VPC</b> <small>View details</small> <small>Detach from VPC</small> <small>Manage tags</small> <small>Delete Internet gateway</small>
<input type="checkbox"/>	Igw-ad1c6ec5	Attached	<small>vpc-6c71a007</small>

Lalu kita pilih **Cilsy VPC**.



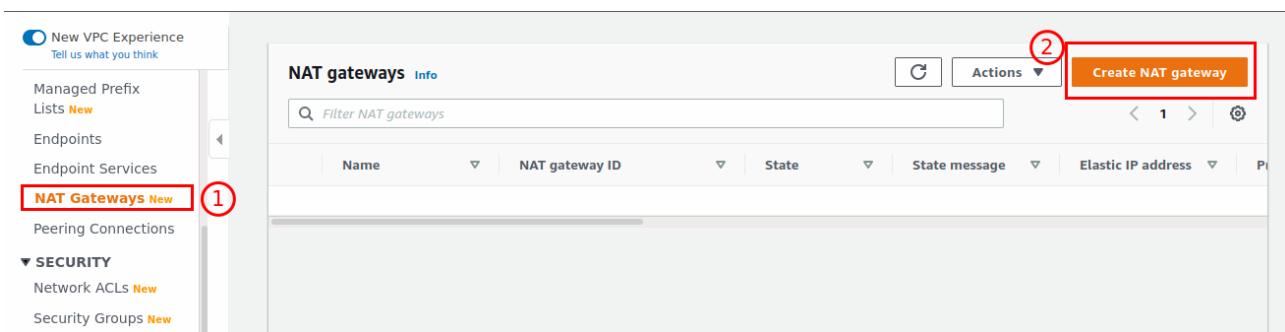
Setelah di **attach**, maka kita akan melihat perubahan status pada bagian **Internet Gateways** milik **Cilsy\_gateway** seperti berikut.

Internet gateways (3) <a href="#">Info</a>					
<input type="checkbox"/>	Name	Internet gateway ID	State	VPC ID	Owner
<input type="checkbox"/>	kubernetes.dev.sdcl...	igw-0144f0a8d07833d3b	Attached	vpc-03486339433cf55da   kubernetes...	898130718046
<input type="checkbox"/>	cilsy_gateway	igw-0795b1b25f798ec46	Attached	vpc-0cc476711abe17ac8   Cilsy VPC	898130718046
<input type="checkbox"/>	-	igw-ad1c6ec5	Attached	vpc-6c71a007	898130718046

#### 6.4.4. NAT Gateway

Jika kita memiliki sebuah **private subnet**, tapi masih menginginkan instance di dalam subnet bisa mengakses internet, maka salah satu caranya adalah menggunakan **NAT Gateway**. Selain **internet Gateway** yang sudah kita bahas sebelumnya, bagian ini membuat **instance** di dalam **private subnet** bisa mengakses **internet**, tapi mencegah internet melakukan koneksi ke dalam instance yang ada di belakang NAT Gateway, sehingga membuat instances di belakang NAT Gateway lebih aman. Untuk membuat NAT Gateway, kita membutuhkan 1 Elastic IP Caranya sebagai berikut.

Masuk kedalam menu **NAT Gateway**, lalu klik **Create NAT Gateway**.



Lalu isi kolom **Subnet** dengan subnet yang dikehendaki, dan isi **elastic IP** untuk di-**assign** ke **NAT Gateway**.

VPC > NAT gateways > Create NAT gateway

### Create NAT gateway Info

Create a NAT gateway and assign it an Elastic IP address.

**NAT gateway settings**

Name - *optional*  
Create a tag with a key of 'Name' and a value that you specify.  
my-nat-gateway-01  
The name can be up to 256 characters long.

**Subnet**  
Select a public subnet in which to create the NAT gateway.  
subnet-07f03a6b5e4aede8d (public\_subnet) 1

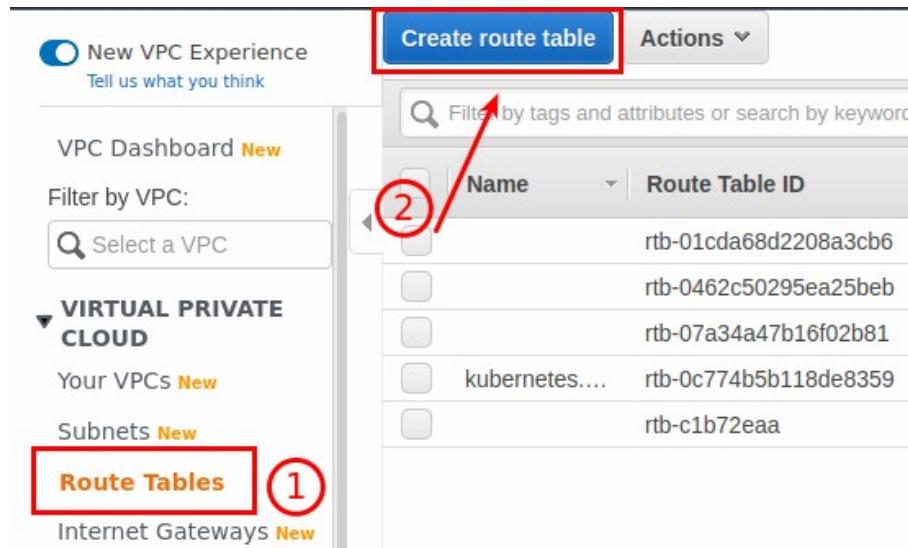
Elastic IP allocation ID Info  
Assign an Elastic IP address to the NAT gateway.  
eipalloc-0b37de1561ac9dbd2 2

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.  
No tags associated with the resource.  
Add new tag 3  
You can add 50 more tags.

Cancel Create NAT gateway

Setelah selesai, NAT tidak otomatis langsung terpakai, tetapi kita harus edit atau create Route Table terlebih dahulu, untuk menentukan Subnet mana yang harus di pasang NAT Gateway.

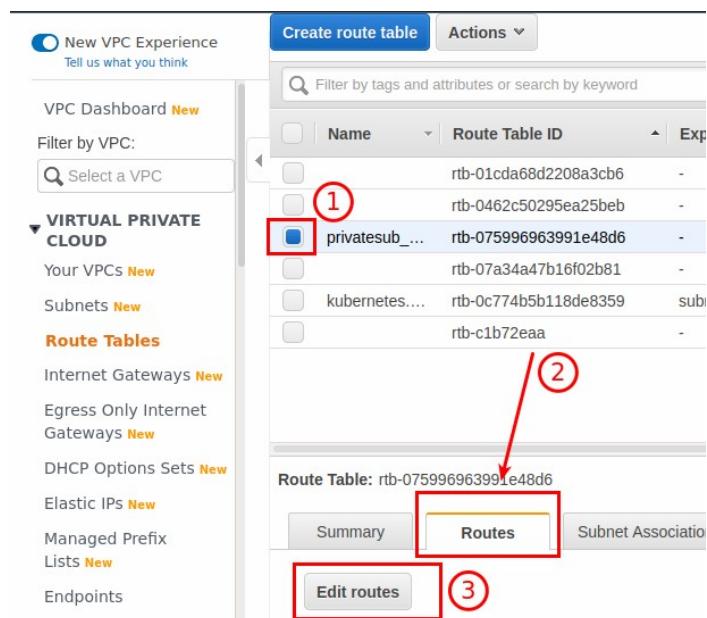
Dengan mengeklik menu **Route Tables**, kita akan dihadapkan pada **Route Tables** dashboard. Klik **Create route table**.



lalu isi nama route table dan pilih VPC.

The screenshot shows the 'Create route table' wizard. At the top, there's a navigation bar with 'aws', 'Services ▾', a search bar, and account information. Below it, the path 'Route Tables > Create route table' is shown. The main form has a title 'Create route table' and a note: 'A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.' It contains fields for 'Name tag' (set to 'privatesub\_route') and 'VPC\*' (set to 'vpc-0cc476711abe17ac8'). Below these are 'Key' and 'Value' input fields. A note below says 'This resource currently has no tags'. At the bottom, there are 'Add Tag' and 'Create' buttons, with a note '\* Required' next to the 'Add Tag' button.

Klik Create lalu pilih route table yang baru saja dibuat, di bagian bawah, kita masuk ke tab Routes dan klik edit. Disini kita akan menentukan subnet mana yang akan kita tempelkan NAT Gateway.



New VPC Experience  Tell us what you think

VPC Dashboard [New](#)

Filter by VPC:  Select a VPC

**VIRTUAL PRIVATE CLOUD**

- Your VPCs [New](#)
- Subnets [New](#)
- Route Tables**
- Internet Gateways [New](#)
- Egress Only Internet Gateways [New](#)
- DHCP Options Sets [New](#)
- Elastic IPs [New](#)
- Managed Prefix Lists [New](#)
- Endpoints

Create route table Actions ▾

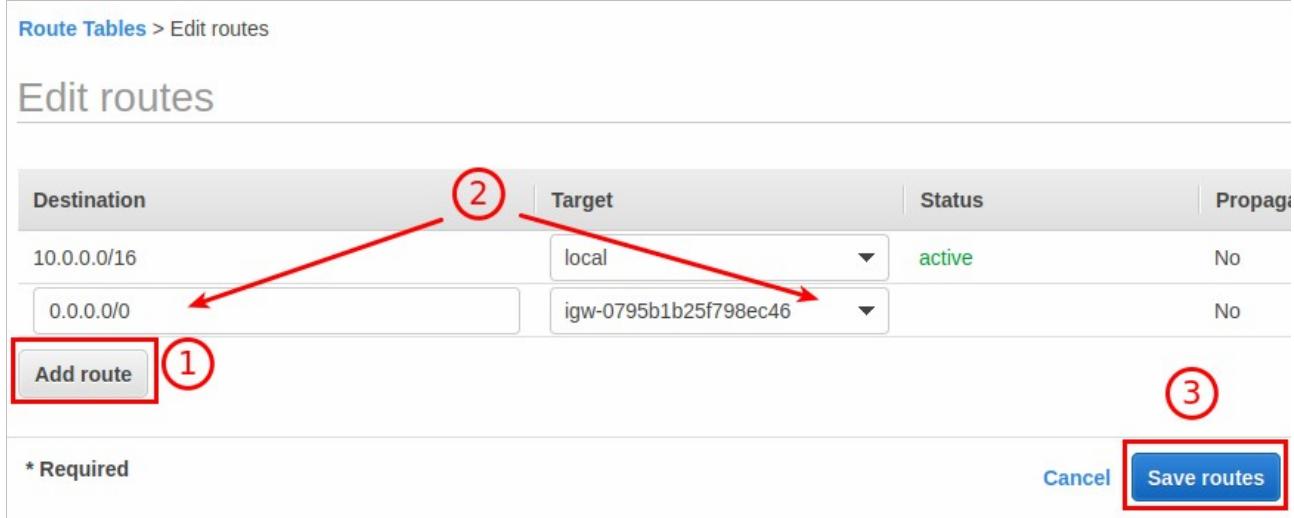
Name	Route Table ID	Exp
rtb-01cda68d2208a3cb6		-
rtb-0462c50295ea25beb		-
<b>privatesub...</b>	<b>rtb-075996963991e48d6</b>	
rtb-07a34a47b16f02b81		-
kubernetes....	rtb-0c774b5b118de8359	sub
rtb-c1b72eaa		-

Route Table: rtb-075996963991e48d6

Summary Routes Subnet Association

Edit routes

Setelah Klik edit, kita isi Destination dengan 0.0.0.0/0 (semua tujuan, dalam hal ini internet), lalu kita isi Target dengan **Internet gateway ID** yang kita buat sebelumnya. Setelah itu klik Save. Bila kita perhatikan, di route sebelumnya sudah terisi 172.31.0.0/16. Itu adalah route untuk mengakses IP lokal di dalam VPC kita, jadi hanya koneksi ke internet yang akan dihandle oleh internet gateway.



Route Tables > Edit routes

### Edit routes

Destination	Target	Status	Propagation Status
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-0795b1b25f798ec46	No	No

Add route 1

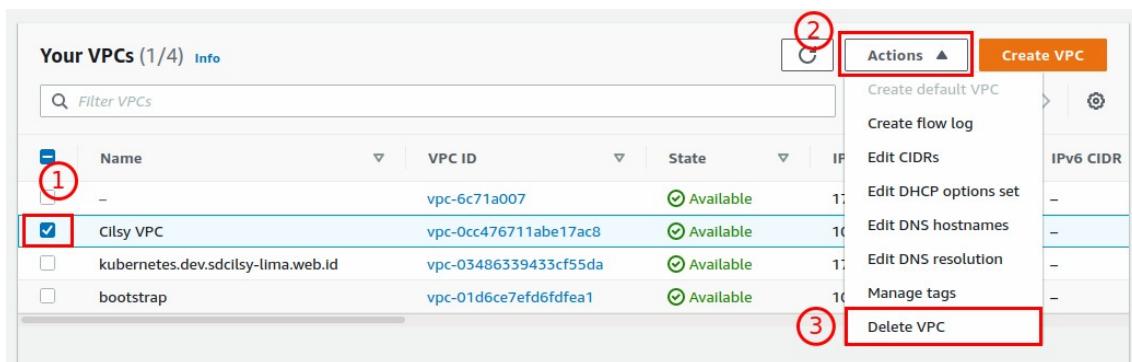
\* Required 2

Cancel 3 Save routes

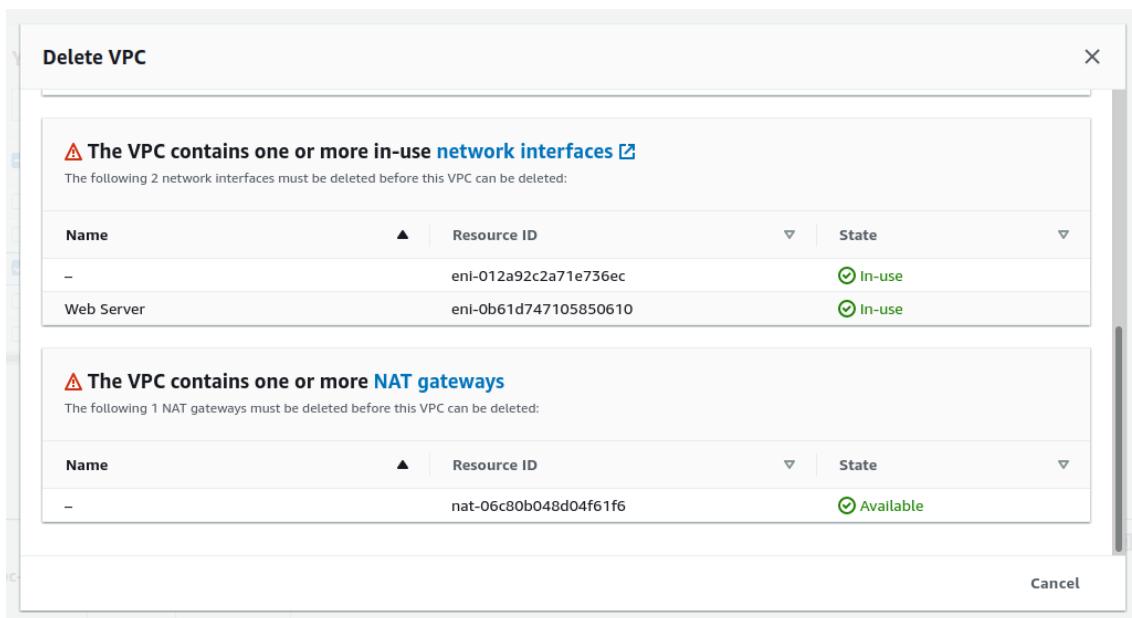
Setelah itu silahkan anda mencoba untuk tes koneksi internet di mesin – mesin di dalam private subnet anda.

#### 6.4.5. Menghapus VPC

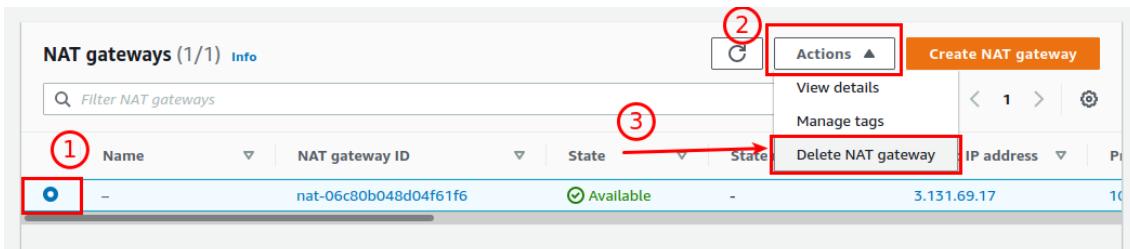
Pada bagian ini kita akan mencoba cara **menghapus VPC** yang sudah kita buat sebelumnya, cara menghapusnya sangat mudah sekali. Kita hanya perlu masuk ke **dashboard VPC** lalu pilih **VPC** yang akan kita hapus.



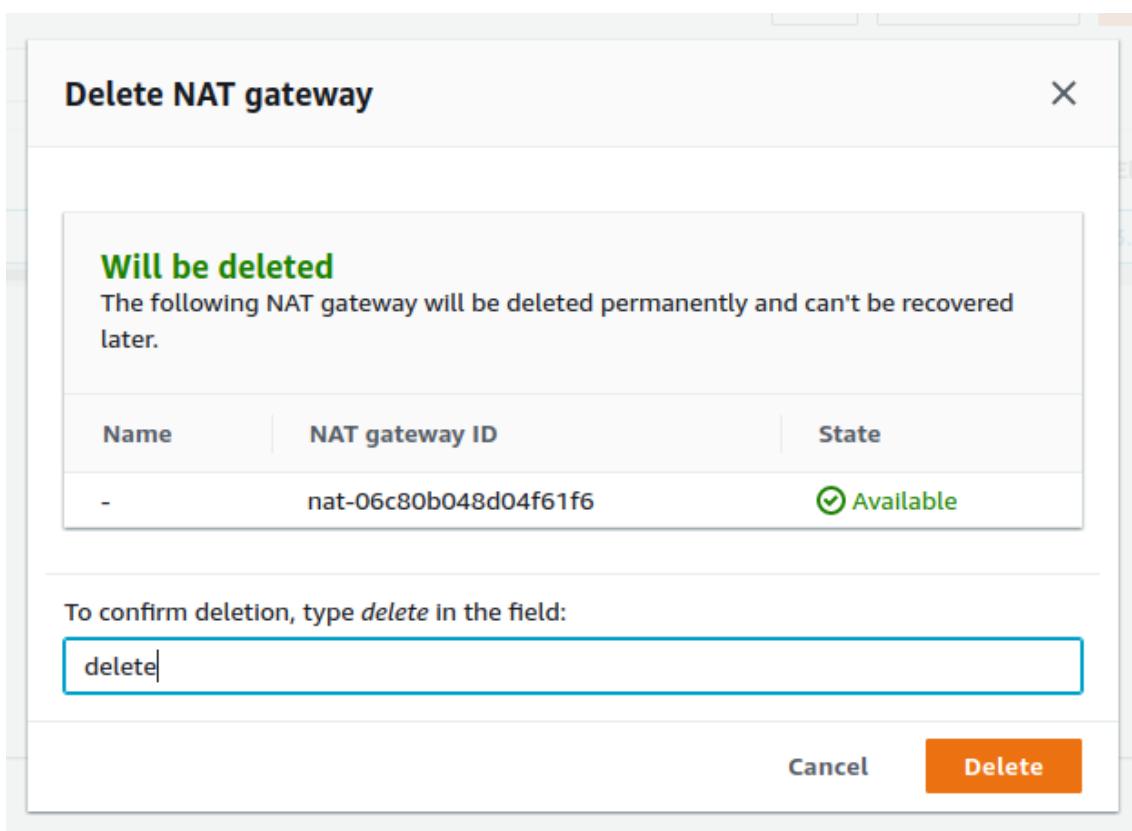
Jika muncul popout bahwa **VPC** tidak dapat di delete secara langsung, berarti ada yang perlu kita hapus terlebih dahulu. Misalnya pada kasus ini adalah Network Interface dan **NAT Gateway** yang terpasang pada **VPC**.



Berikut adalah cara menghapus **NAT Gateway**. Dengan memilih **NAT Gateway** yang akan dihapus, lalu mengeklik **Action** dan **Delete NAT Gateway**

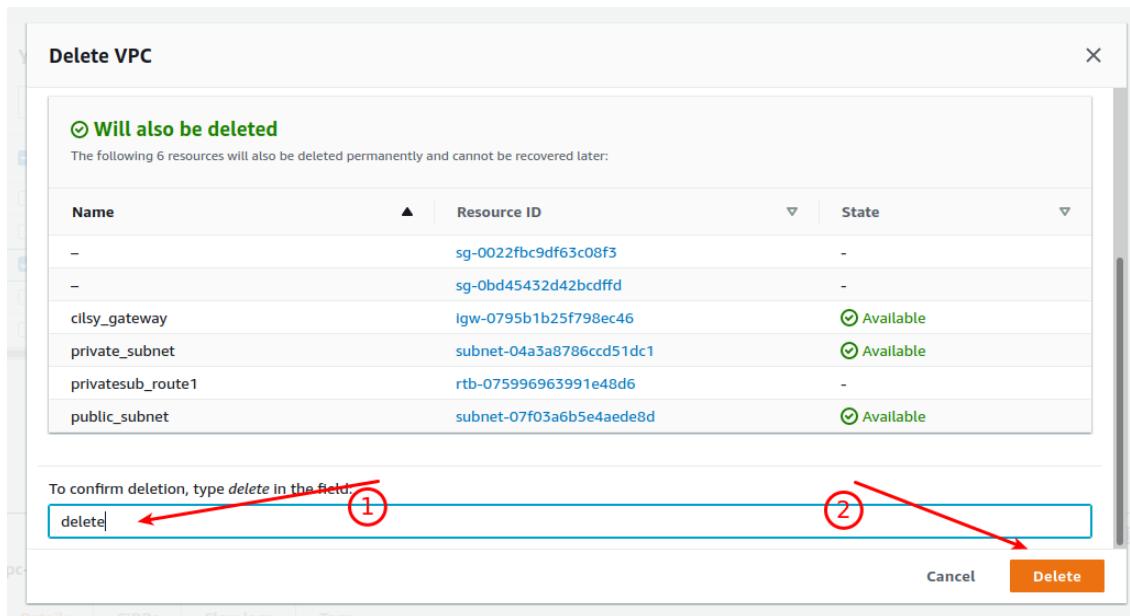


Setelah itu akan muncul popout untuk mengkonfirmasi penghapusan **NAT Gateway**. Kita diharuskan untuk mengetikan “*delete*” pada kolom yang tersedia

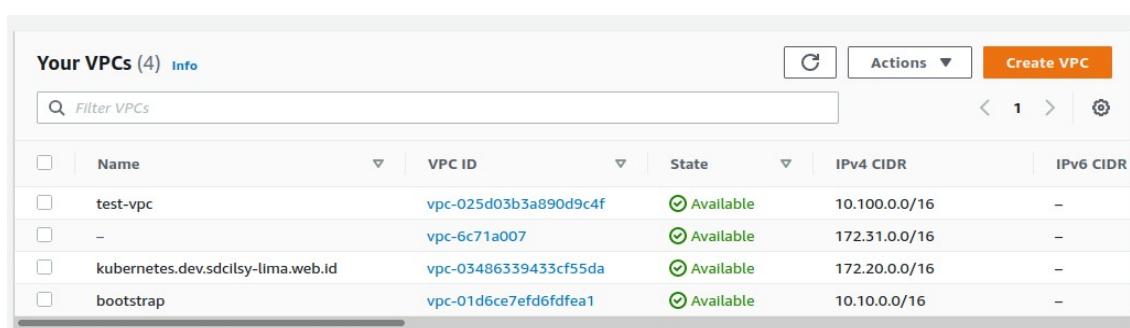


Setelah **NAT Gateway** berhasil dihapus, kita lanjutkan langkah untuk menghapus VPC yakni dengan mengulang langkah seperti yang sudah dilakukan.

Setelah itu akan muncul pop out untuk mengkonfirmasi penghapusan VPC. Kita diharuskan untuk mengetikan “*delete*” pada kolom yang tersedia



Apabila sudah berhasil maka akan muncul seperti dibawah ini.



## 6.4.6. Exercise

1. Buat sebuah VPC dengan nama kalian sendiri !
2. Setting Konfigurasi Subnet, Gateway, dan NAT pada VPC yang sudah dibuat !

## 6.5. Amazon EC2

### 6.5.1. Apa itu Amazon EC2 ?

Amazon Elastic Cloud Compute (Amazon EC2) merupakan salah satu layanan yang ada pada AWS yang memungkinkan kita untuk membuat sebuah Virtual Machine (komputer virtual) didalam ekosistem AWS. Kita dapat menentukan sendiri banyaknya CPU, ukuran RAM, Storage dan juga OS untuk Machine yang akan kita buat.



Amazon EC2

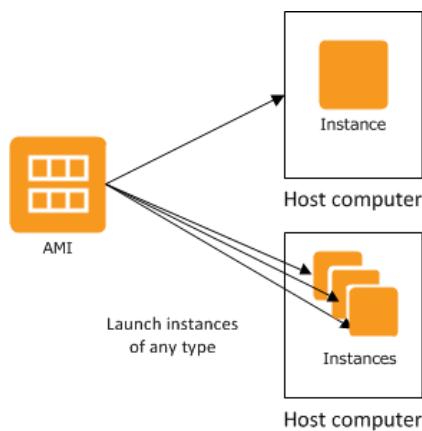
*Logo Amazon EC2*

Untuk kalian yang pernah menggunakan aplikasi Proxmox, VirtualBox, VMWare pasti sudah tidak akan asing lagi menggunakan layanan ini, karena pada dasarnya sistem pembuatan dan penggunaannya cukup sama.

Pada Amazon EC2 terdapat beberapa komponen yang menunjang penggunaannya serta memiliki fungsinya tersendiri seperti Instance, IAM, Keypair dll.

### 6.5.2. AMI

Amazon Machine Image (AMI) adalah template yang berisi konfigurasi perangkat lunak (misalnya, sistem operasi, server aplikasi, dan aplikasi). Dari AMI, kita dapat meluncurkan instance, yang merupakan salinan AMI yang berjalan sebagai server virtual di cloud. Kita dapat meluncurkan beberapa contoh AMI, seperti yang ditunjukkan pada gambar berikut.



*Ilustrasi penggunaan AMI*

Amazon Web Services (AWS) menerbitkan banyak AMI yang berisi konfigurasi perangkat lunak umum untuk penggunaan publik. Selain itu, anggota komunitas pengembang AWS telah menerbitkan AMI kustom mereka sendiri. Kita juga dapat membuat AMI atau kustom AMI sendiri, dengan ini kita bisa lebih cepat membuat instance baru sesuai dengan kebutuhan yang kita inginkan.

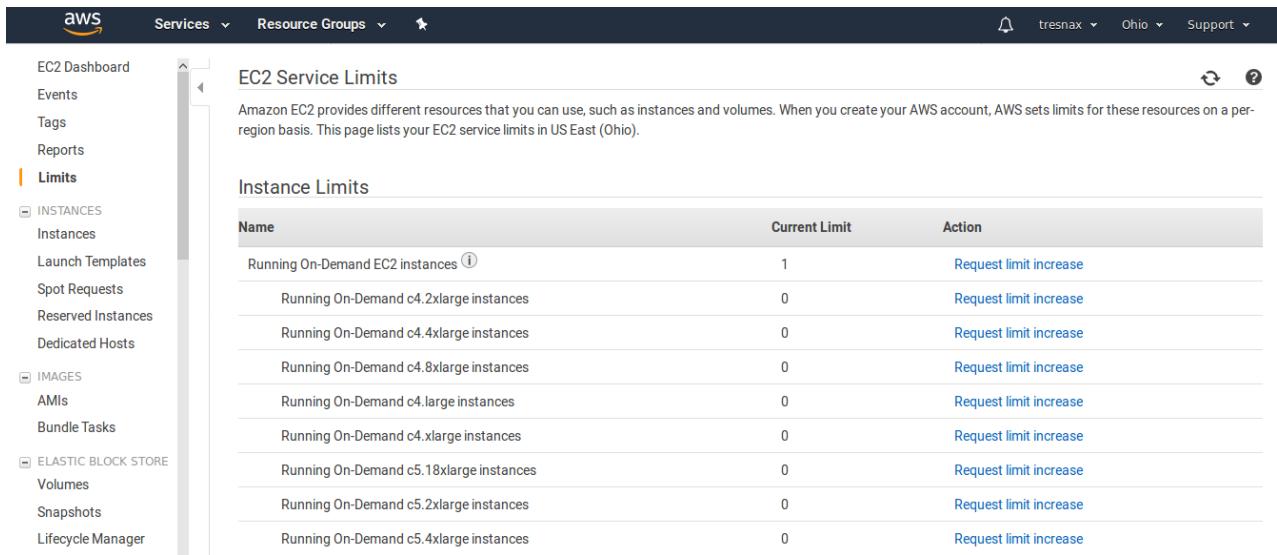
Semua AMI didukung oleh Amazon EBS, yang berarti bahwa hardisk untuk instance yang diluncurkan dari AMI adalah volume dari Amazon EBS, atau didukung oleh toko instance, yang berarti bahwa hardisk untuk instance yang diluncurkan dari AMI adalah volume dari toko, contoh yang dibuat dari template yang disimpan di Amazon S3.

### 6.5.3. Instance

Kita dapat meluncurkan berbagai jenis instance dari satu AMI. Jenis instance pada dasarnya menentukan perangkat keras komputer host yang digunakan untuk instance kita. Setiap jenis instance menawarkan kemampuan komputasi dan memori yang berbeda. Pilih jenis instance berdasarkan jumlah memori dan daya komputasi yang kita butuhkan untuk aplikasi atau perangkat lunak yang kita rencanakan untuk dijalankan pada instance.

Setelah kita meluncurkan sebuah instance, ia terlihat seperti host tradisional, dan kita dapat berinteraksi dengannya seperti halnya komputer mana pun. kita

memiliki kontrol penuh atas instance kita, untuk meng akses root kita bisa menggunakan perintah sudo.



The screenshot shows the AWS EC2 Service Limits page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, and Limits (which is currently selected). The main content area is titled "EC2 Service Limits" and contains a sub-section titled "Instance Limits". A table lists the current limits for various instance types:

Name	Current Limit	Action
Running On-Demand EC2 instances	1	Request limit increase
Running On-Demand c4.2xlarge instances	0	Request limit increase
Running On-Demand c4.4xlarge instances	0	Request limit increase
Running On-Demand c4.8xlarge instances	0	Request limit increase
Running On-Demand c4.large instances	0	Request limit increase
Running On-Demand c4.xlarge instances	0	Request limit increase
Running On-Demand c5.18xlarge instances	0	Request limit increase
Running On-Demand c5.2xlarge instances	0	Request limit increase
Running On-Demand c5.4xlarge instances	0	Request limit increase

*Limit EC2 yang hanya 1 instance*

Akun AWS kita memiliki batasan jumlah instance yang dapat kita jalankan dan region yang dapat kita gunakan. Pada saat awal menggunakan AWS kita hanya akan diberikan batasan 1 instance/region, dan region yang dapat kita gunakan hanya tiga yaitu Ohio, Oregon, dan Virginia. Untuk mendapatkan limit instance dan region yang lebih banyak, kita harus melakukan request terlebih dahulu pada pihak AWS, dengan begitu kita bisa mendapatkan limit lebih banyak dari defaultnya.

#### 6.5.4. Instance Type

Saat kita meluncurkan instance baru, jenis instance yang kita pilih akan menentukan hardware yang kita gunakan pada komputer virtual kita. Setiap jenis instance menawarkan kemampuan komputasi, memori, dan penyimpanan yang berbeda dan dikelompokkan berdasarkan kemampuan dan kebutuhan yang kita inginkan.

Amazon EC2 menyediakan setiap instance dengan jumlah kapasitas CPU yang konsisten dan dapat diprediksi, terlepas dari perangkat keras yang mendasarinya.



#### **6.5.4.1. Instance Generasi Saat Ini**

Untuk kinerja yang terbaik, kita disarankan untuk menggunakan jenis instance generasi saat ini ketika meluncurkan sebuah instance baru. Dibawah ini merupakan tabel instance generasi saat ini yang banyak digunakan.

Instance Family	Current Generation Instance Types
General purpose	t2.nano   t2.micro   t2.small   t2.medium   t2.large   t2.xlarge   t2.2xlarge   t3.nano   t3.micro   t3.small   t3.medium   t3.large   t3.xlarge   t3.2xlarge   m4.large   m4.xlarge   m4.2xlarge   m4.4xlarge   m4.10xlarge   m4.16xlarge   m5.large   m5.xlarge   m5.2xlarge   m5.4xlarge   m5.12xlarge   m5.24xlarge   m5d.large   m5d.xlarge   m5d.2xlarge   m5d.4xlarge   m5d.12xlarge   m5d.24xlarge
Compute optimized	c4.large   c4.xlarge   c4.2xlarge   c4.4xlarge   c4.8xlarge   c5.large   c5.xlarge   c5.2xlarge   c5.4xlarge   c5.9xlarge   c5.18xlarge   c5d.xlarge   c5d.2xlarge   c5d.4xlarge   c5d.9xlarge   c5d.18xlarge
Memory optimized	r4.large   r4.xlarge   r4.2xlarge   r4.4xlarge   r4.8xlarge   r4.16xlarge   r5.large   r5.xlarge   r5.2xlarge   r5.4xlarge   r5.12xlarge   r5.24xlarge   r5d.large   r5d.xlarge   r5d.2xlarge   r5d.4xlarge   r5d.12xlarge   r5d.24xlarge   x1.16xlarge   x1.32xlarge   xle.xlarge   xle.2xlarge   xle.4xlarge   xle.8xlarge   xle.16xlarge   xle.32xlarge   zld.large   zld.xlarge   zld.2xlarge   zld.3xlarge   zld.6xlarge   zld.12xlarge
Storage optimized	d2.xlarge   d2.2xlarge   d2.4xlarge   d2.8xlarge   h1.2xlarge   h1.4xlarge   h1.8xlarge   h1.16xlarge   i3.large   i3.xlarge   i3.2xlarge   i3.4xlarge   i3.8xlarge   i3.16xlarge   i3.metal
Accelerated computing	f1.2xlarge   f1.16xlarge   g3.4xlarge   g3.8xlarge   g3.16xlarge   p2.xlarge   p2.8xlarge   p2.16xlarge   p3.2xlarge   p3.8xlarge   p3.16xlarge

*Tabel Instance generasi saat ini*

#### **6.5.4.2. Instance Generasi Sebelumnya**

Amazon Web Services menawarkan instance generasi sebelumnya untuk pengguna yang telah mengoptimalkan aplikasi mereka di sekitar instance ini dan belum melakukan peningkatan. AWS mendorong kita untuk menggunakan instance generasi terbaru untuk mendapatkan kinerja terbaik, tetapi AWS masih terus mendukung generasi sebelumnya. Jika saat ini kita menggunakan instance generasi sebelumnya, Kita dapat melihat instance generasi saat ini yang bisa digunakan untuk peningkatan sesuai dengan model instance generasi sebelumnya. Berikut tabel instance generasi sebelumnya.

Instance Family	Previous Generation Instance Types
General purpose	m1.small   m1.medium   m1.large   m1.xlarge   m3.medium   m3.large   m3.xlarge   m3.2xlarge
Compute optimized	c1.medium   c1.xlarge   cc2.8xlarge   c3.large   c3.xlarge   c3.2xlarge   c3.4xlarge   c3.8xlarge
Memory optimized	m2.xlarge   m2.2xlarge   m2.4xlarge   cr1.8xlarge   r3.large   r3.xlarge   r3.2xlarge   r3.4xlarge   r3.8xlarge
Storage optimized	hs1.8xlarge   i2.xlarge   i2.2xlarge   i2.4xlarge   i2.8xlarge
GPU optimized	g2.2xlarge   g2.8xlarge
Micro	t1.micro

*Tabel Instance generasi sebelumnya*

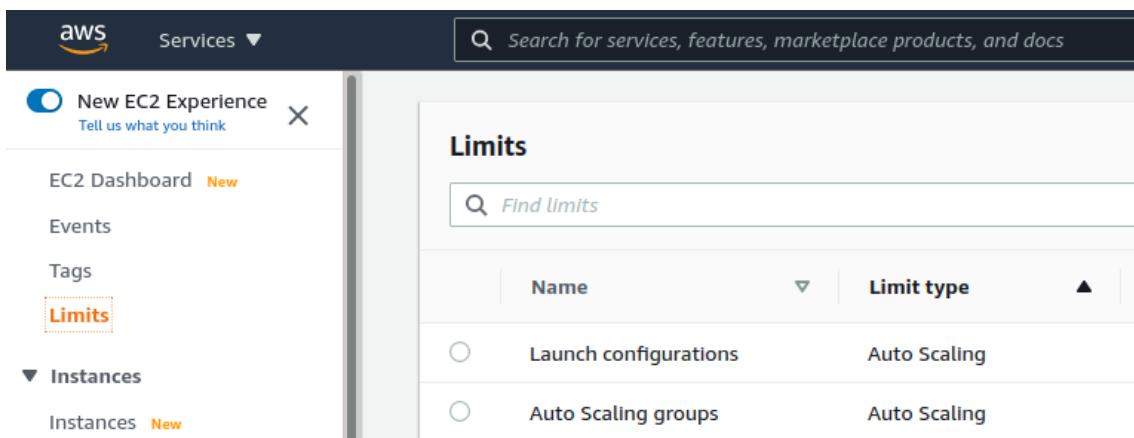
### 6.5.5. KeyPairs

Amazon EC2 menggunakan kriptografi public key untuk mengenkripsi dan mendekripsi informasi login. Kriptografi public key digunakan untuk mengenkripsi sepotong data, seperti kata sandi, kemudian penerima menggunakan private key untuk mendekripsi data. public key dan privat key dikenal sebagai keypair.

Untuk masuk kedalam sebuah instance, kita harus membuat keypair, tentukan nama keypair saat kita meluncurkan instance, dan berikan private key ketika kita terhubung ke instance. Pada contoh Linux, konten public key ditempatkan di entri dalam `~/.ssh/authorized_keys`. Ini dilakukan saat boot dan memungkinkan kita mengakses instance dengan aman menggunakan private key, bukan kata sandi.

### 6.5.6. Request Limit Instance AWS

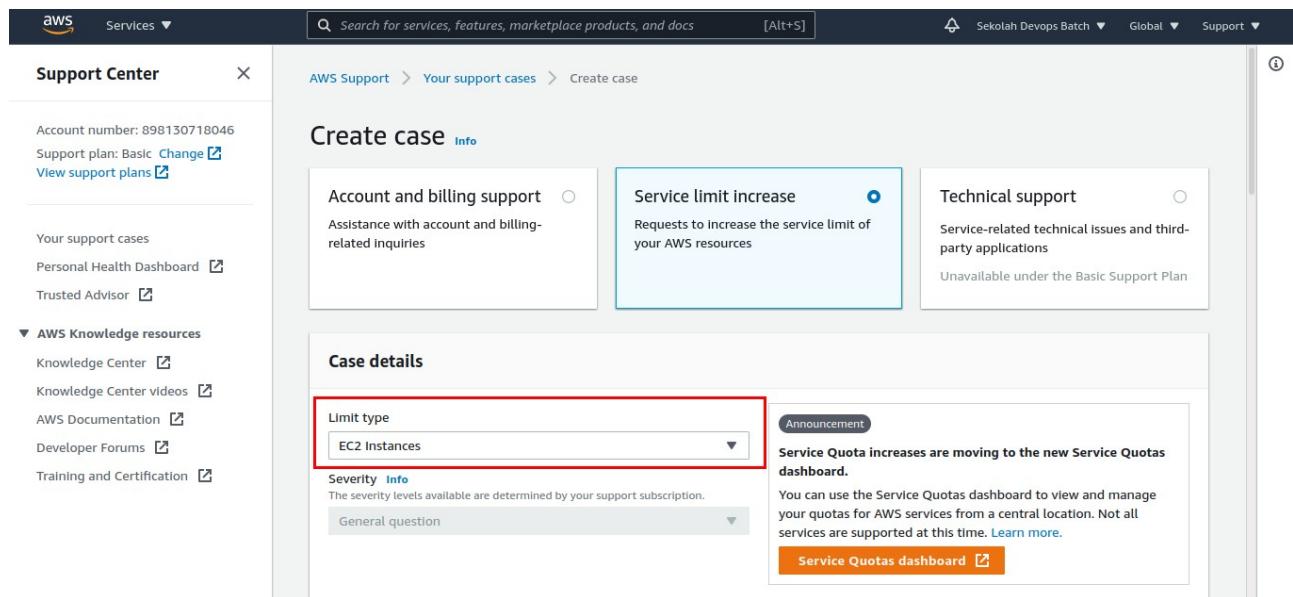
Untuk menambahkan limitasi baru bagi instance yang akan kita buat, kita perlu melakukan request pada CS AWS. Caranya cukup mudah, kita hanya perlu masuk ke **EC2 Dashboard > Limits**. Klik Request Limit Inscrase pada sambing jumlah limit instace.



Name	Limit type
Launch configurations	Auto Scaling
Auto Scaling groups	Auto Scaling

*Tampilan menu limit pada EC2 Dashboard*

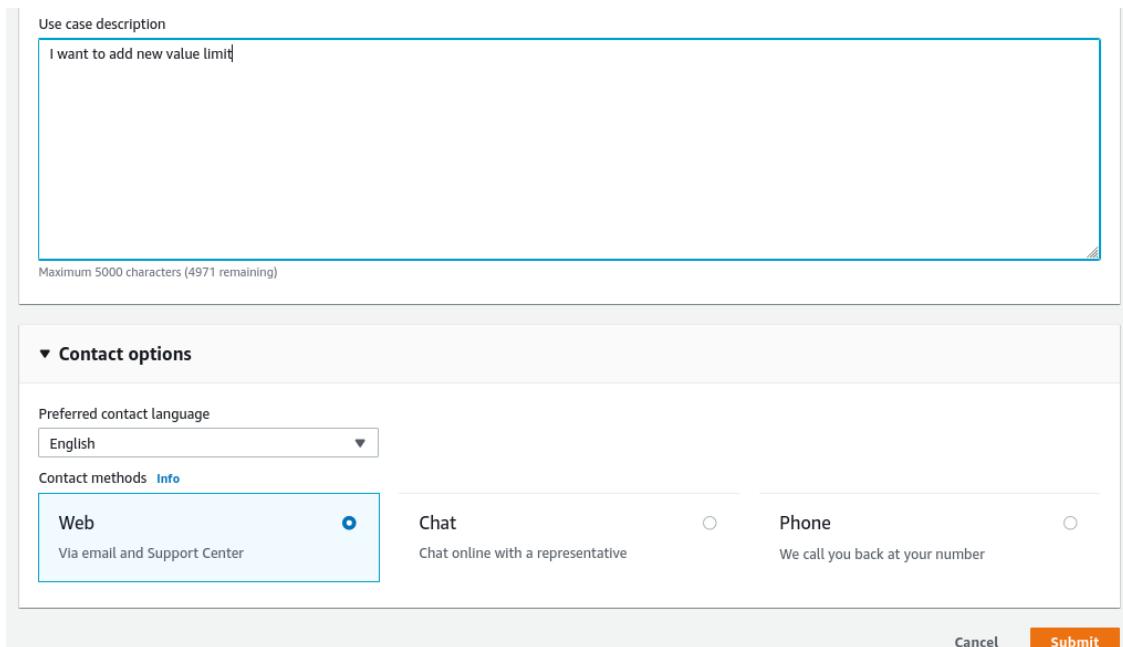
Setelah itu kita akan diarahkan ke tab baru untuk mengisi informasi mengenai instance yang akan kita request limitnya.



The screenshot shows the AWS Support Center interface. In the top navigation bar, 'Services' is selected. The main area is titled 'Create case' with a sub-section 'Service limit increase' highlighted with a blue border. Below this, there are three categories: 'Account and billing support', 'Service limit increase' (selected), and 'Technical support'. Under 'Case details', the 'Limit type' dropdown is set to 'EC2 Instances', which is also highlighted with a red box. To the right of this section, there is an announcement about Service Quotas moving to a new dashboard.

*Form request limit AWS*

Jangan lupa juga isikan descriptionnya agar informasinya bisa jadi lebih jelas, tetapi disini saya mengertakan deskripsi yang sedikit. Setelah itu submit.

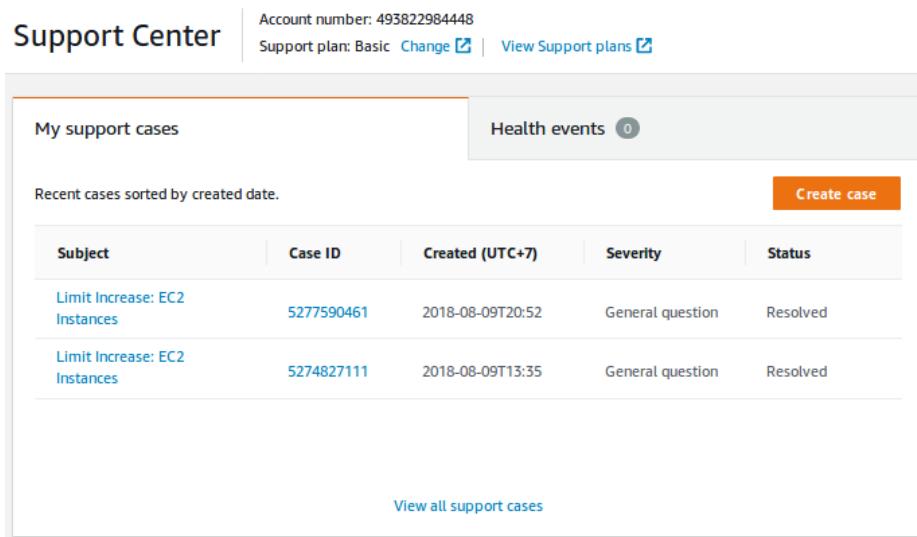


The screenshot shows the 'Create case' form. The 'Use case description' field contains the text 'I want to add new value limit'. Below this, a note says 'Maximum 5000 characters (4971 remaining)'. The 'Contact options' section includes a 'Preferred contact language' dropdown set to 'English' and a 'Contact methods' section with three options: 'Web' (selected), 'Chat', and 'Phone'. At the bottom right of the form are 'Cancel' and 'Submit' buttons.

*Form request limit AWS*



Tunggu sekitar 5-7 hari sampai AWS meng approve limit value yang kita inginkan. Kita dapat mengeceknya dengan masuk dashboard AWS Support Center yang ada di menu service.



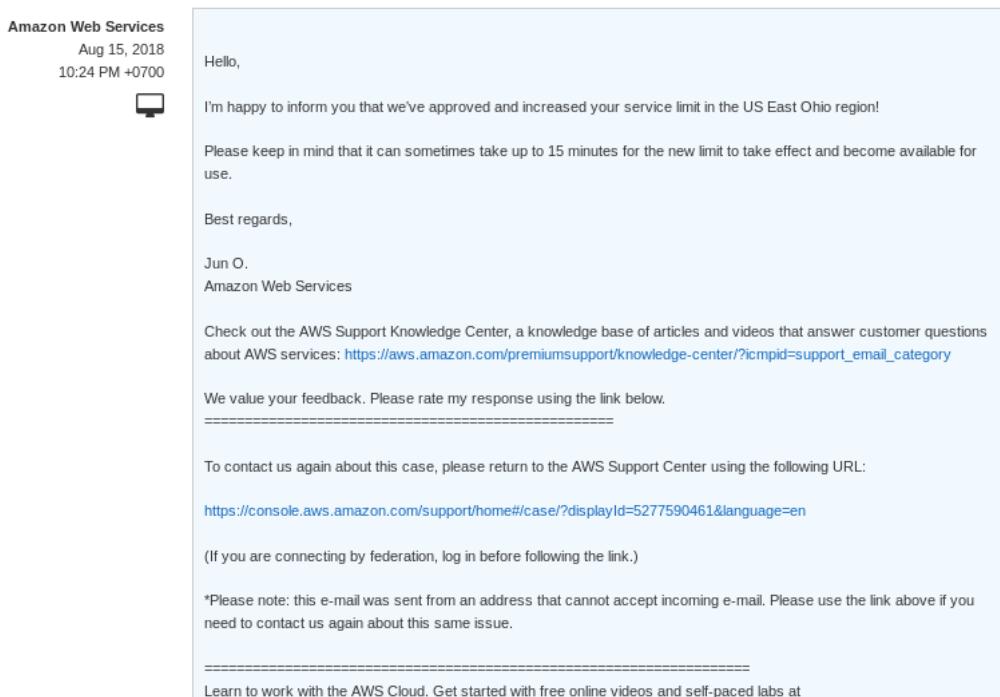
The screenshot shows the AWS Support Center interface. At the top, it displays the account number (493822984448) and support plan (Basic). Below this, there are two tabs: "My support cases" and "Health events (0)". A "Create case" button is located in the top right corner. The "My support cases" section lists two recent cases sorted by creation date:

Subject	Case ID	Created (UTC+7)	Severity	Status
Limit Increase: EC2 Instances	<a href="#">5277590461</a>	2018-08-09T20:52	General question	Resolved
Limit Increase: EC2 Instances	<a href="#">5274827111</a>	2018-08-09T13:35	General question	Resolved

At the bottom of the "My support cases" section, there is a link "View all support cases".

*Tampilan Support Ticker di AWS Suport Center.*

Apabila sudah berhasil, nanti akan ada pemberitahuan melalui email dan juga subject kita akan memiliki status resolved dan dibalas seperti berikut ini.



The screenshot shows an email from Amazon Web Services. The header information includes:

- Amazon Web Services
- Aug 15, 2018
- 10:24 PM +0700
- 

The body of the email contains the following text:

Hello,

I'm happy to inform you that we've approved and increased your service limit in the US East Ohio region!

Please keep in mind that it can sometimes take up to 15 minutes for the new limit to take effect and become available for use.

Best regards,

Jun O.  
Amazon Web Services

Check out the AWS Support Knowledge Center, a knowledge base of articles and videos that answer customer questions about AWS services: [https://aws.amazon.com/premiumsupport/knowledge-center/?icmpid=support\\_email\\_category](https://aws.amazon.com/premiumsupport/knowledge-center/?icmpid=support_email_category)

We value your feedback. Please rate my response using the link below.

To contact us again about this case, please return to the AWS Support Center using the following URL:

<https://console.aws.amazon.com/support/home#/case/?displayId=5277590461&language=en>

(If you are connecting by federation, log in before following the link.)

\*Please note: this e-mail was sent from an address that cannot accept incoming e-mail. Please use the link above if you need to contact us again about this same issue.

Learn to work with the AWS Cloud. Get started with free online videos and self-paced labs at

*Pesan dari AWS bahwa limit instance sudah di ACC*



## 6.5.7. Exercise

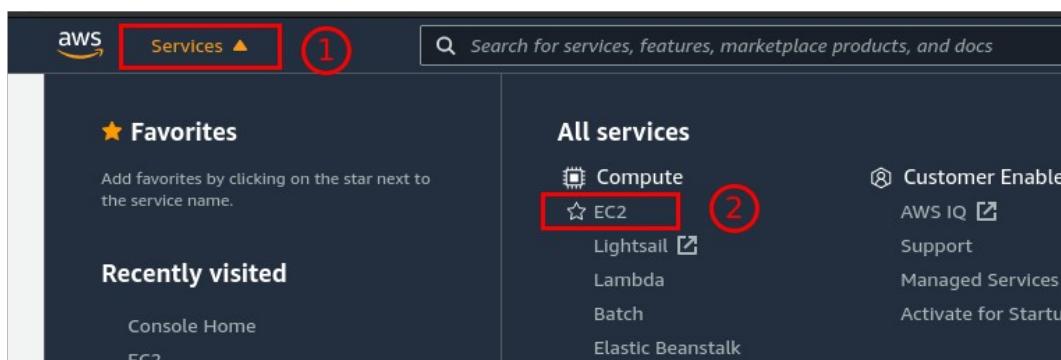
1. Apa peran AMI dan Instance pada Amazon EC2 ?
2. Jelaskan kembali fungsi dari keypair !

## 6.6. Praktik EC2 - Elastic Cloud Computing

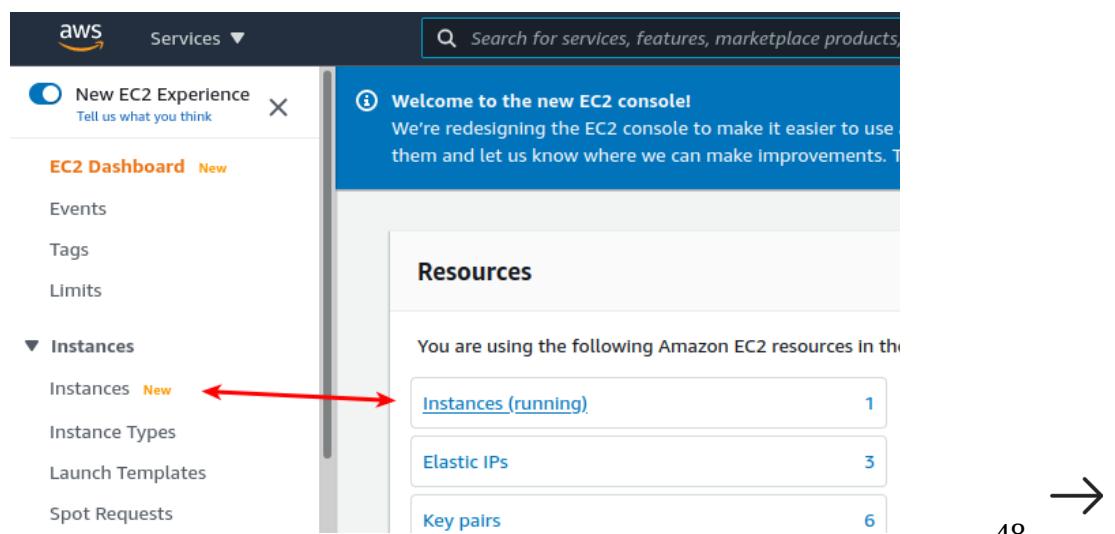
### 6.6.1. Membuat Linux EC2 Instance

Setelah kita bermain-main dengan network infrastruktur, kita akan mencoba untuk spin-up EC2 instance dan konfigurasi firewall. Selain itu kita akan melakukan deploymen aplikasi pada server yang sudah kita buat. Tapi sebelumnya, kita akan membuat EC2 terlebih dahulu.

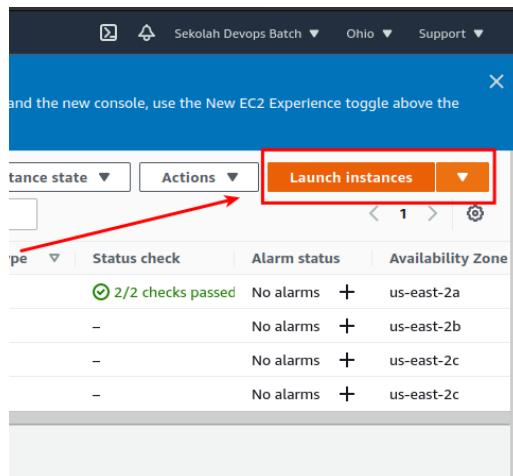
Klik **services** di kiri atas lalu klik **EC2**. Akan tampil **EC2 Dashboard**.



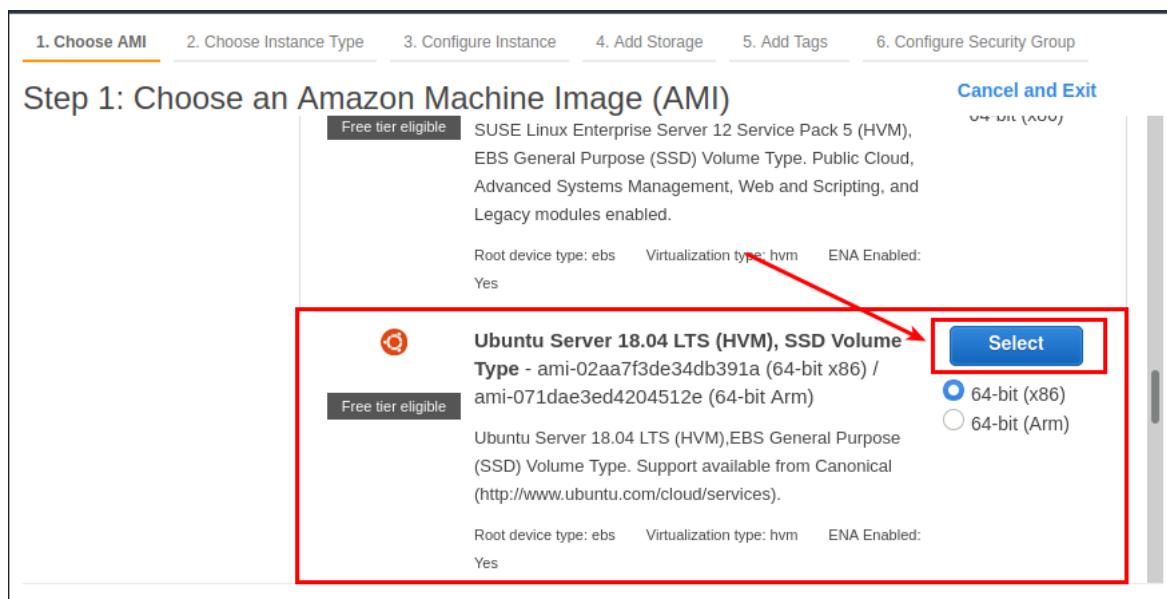
Setelah EC2 dashboard tampil, klik **Instances**, atau **Running Instances**



Untuk membuat cloud server, klik **Launch Instance**.



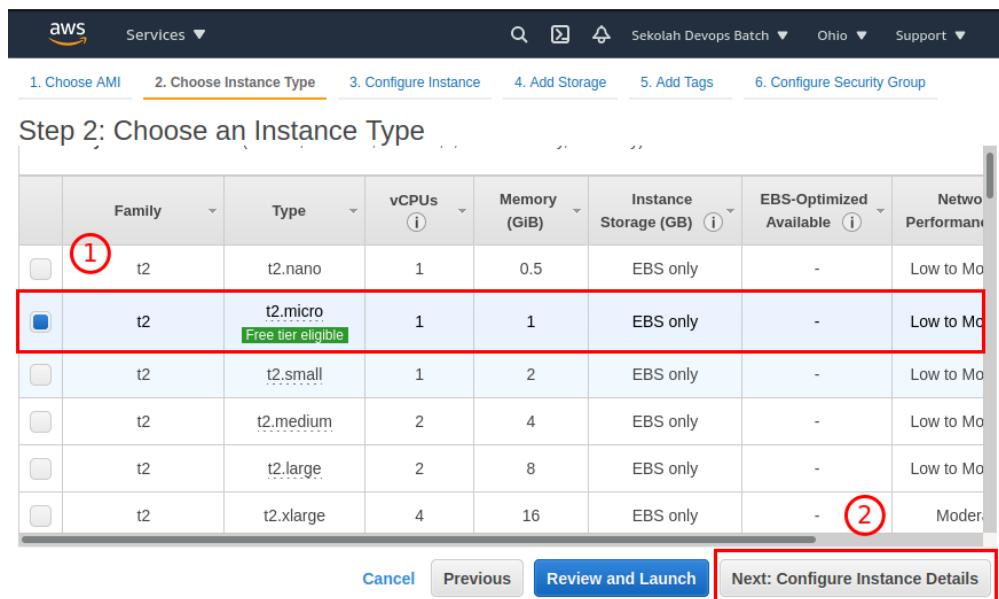
Lalu pilih **base image** atau **OS** yang diinginkan, setelah itu klik **select**. Untuk kali ini, kita akan membuat sebuah **Ubuntu Server**.



Selanjutnya kita akan memilih spesifikasi dari instance yang akan kita buat, pilihannya sangat beragam. Jenis-jenisnya bisa dilihat lebih detail [disini](#). Kita akan memilih **t2.micro** pada praktek ini.

Selanjutnya klik **Configure instance detail**. Kita bisa saja menetapkan opsi default untuk membuat instance, dengan langsung klik **Review and Launch**.

Tetapi kita lebih baik melihat lebih dalam opsi-opsi yang diberikan oleh AWS agar ke depannya kita bisa menentukan spesifikasi yang tepat.



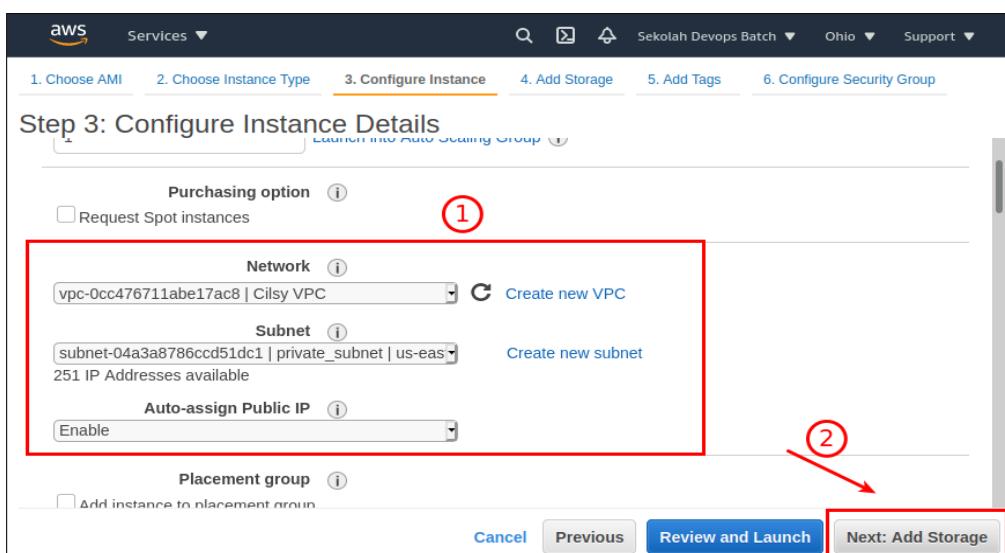
**Step 2: Choose an Instance Type**

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2	t2.nano	1	0.5	EBS only	-	Low to Moderate
t2	t2.micro	1	1	EBS only	-	Low to Moderate
t2	t2.small	1	2	EBS only	-	Low to Moderate
t2	t2.medium	2	4	EBS only	-	Low to Moderate
t2	t2.large	2	8	EBS only	-	Low to Moderate
t2	t2.xlarge	4	16	EBS only	-	Moderate

Cancel Previous Review and Launch **Next: Configure Instance Details**

Selanjutnya akan ada opsi yang lebih banyak. **Number of instance** yang diinginkan, ada juga **spot instance** yang memiliki harga yang bisa jauh lebih murah. **VPC** bisa dipilih dengan menggunakan **VPC** yang sudah kita buat tadi dengan memilih di bagian **network**.

Bagian lainpun seperti **subnet** dan **Public IP** pun bisa kita tentukan *assignment* nya disini, tapi kita tidak bisa memilih IP sendiri karena itu ditentukan oleh AWS. Setelah selesai, kita lanjut ke **storage configuration** dengan klik tombol **Next : Add Storage**.



**Step 3: Configure Instance Details**

Purchasing option (i)  
 Request Spot instances **1**

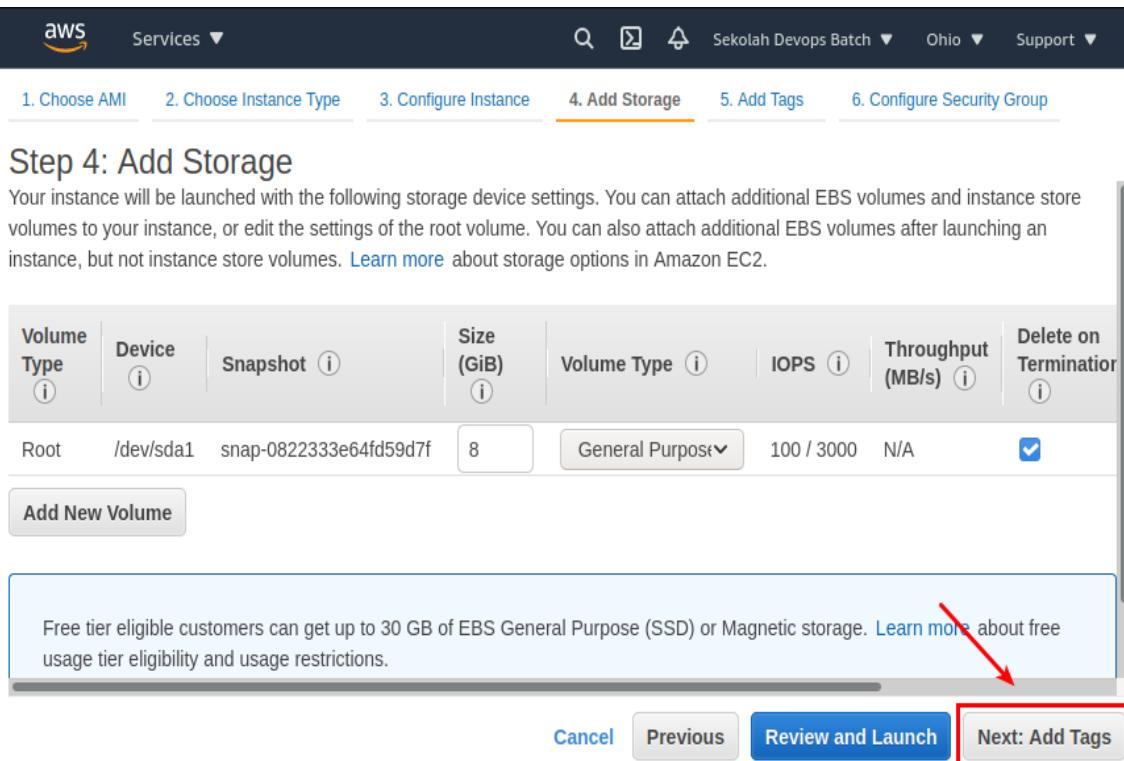
Network (i)  
 vpc-0cc476711abe17ac8 | Cilsy VPC   
 Subnet (i)  
 subnet-04a3a8786cccd51dc1 | private\_subnet | us-eas   
 251 IP Addresses available

Auto-assign Public IP (i)  
 Enable

Placement group (i)  
 Add instance to placement group

Cancel Previous **Review and Launch** **Next: Add Storage** **2**

Di bagian **storage**, kita bisa memilih seberapa besar storage kita dalam satuan gigabyte, ada juga pilihan **tipe storage dan enkripsinya**. Setelah semuanya sesuai, kita klik **Next: Add Tags**.



**Step 4: Add Storage**

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

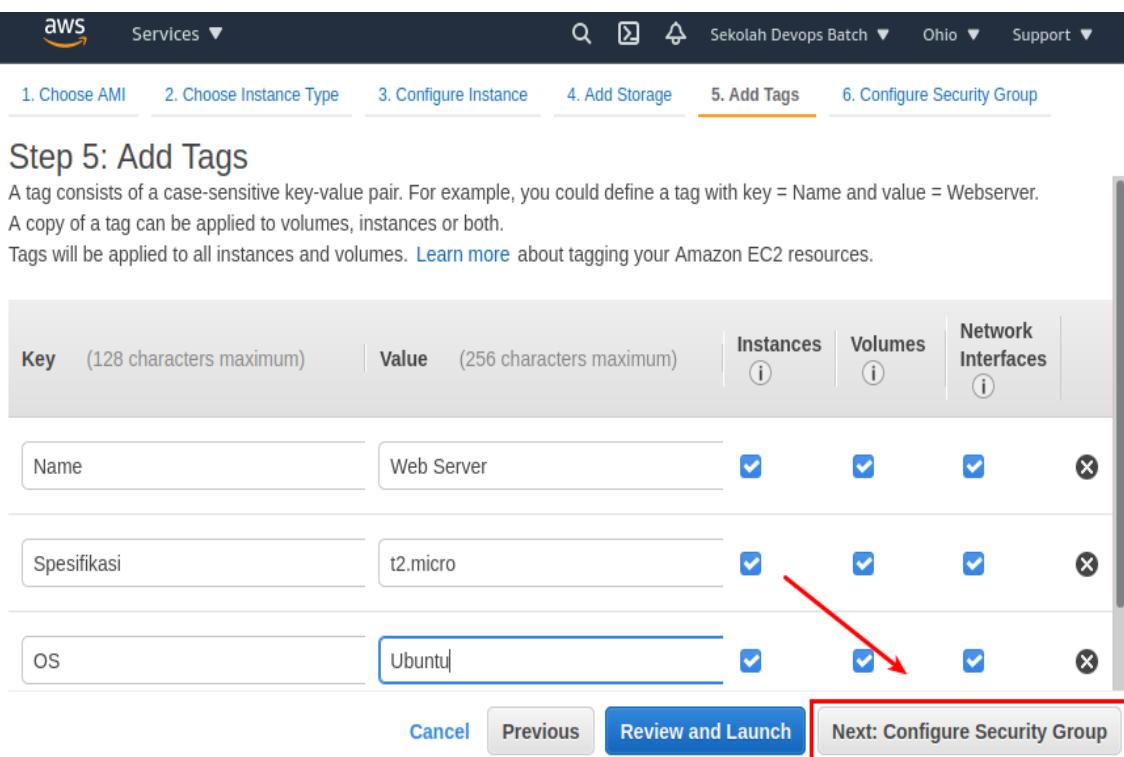
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination
Root	/dev/sda1	snap-0822333e64fd59d7f	8	General Purpose	100 / 3000	N/A	<input checked="" type="checkbox"/>

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) **Next: Add Tags**

**Tag** berfungsi untuk menandai atau memberi label pada instance. Tag bisa diisi lebih dari 1 macam agar pencarian bisa lebih akurat ketika kita memanage banyak instance.



**Step 5: Add Tags**

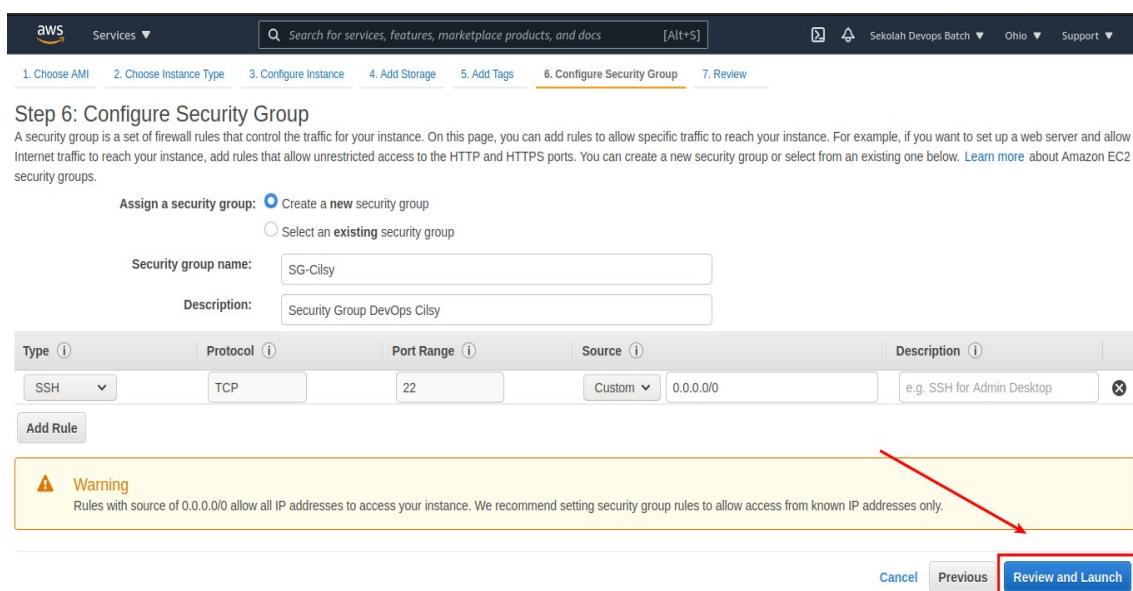
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes	Network Interfaces
Name	Web Server	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Spesifikasi	t2.micro	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
OS	Ubuntu	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Cancel](#) [Previous](#) [Review and Launch](#) **Next: Configure Security Group**

Selanjutnya adalah **Security Group**. Ini adalah layer security di level **port TCP** dan **UDP**. **Security Group** bekerja layaknya **firewall**. Kita bisa menentukan port mana yang bisa diakses, dan dari IP mana port tersebut bisa diakses.

Untuk praktik ini, kita akan membuka **port SSH (22)** dan memberi permission pada IP kita. Kita juga bisa mengisi banyak rule untuk Security Group ini dengan memilih **Add Rule** dan kita bisa memilih untuk membuat yang baru atau memakai yang sudah ada.



Setelah konfigurasi sudah lengkap, di halaman selanjutnya akan tampil seluruh konfigurasi yang kita buat untuk di **review** dan di **edit** jika masih ada yang belum sesuai. Jika sudah sesuai, kita bisa klik **Launch**.

Setelah kita klik **Launch**, AWS akan membaca semua konfigurasi dan membuat server sesuai dengan apa yang kita pilih.

1. Choose AMI    2. Choose Instance Type    3. Configure Instance    4. Add Storage    5. Add Tags    6. Configure Security Group    **7. Review**

### Step 7: Review Instance Launch

**AMI Details** [Edit AMI](#)

 **Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-01e7ca2ef94a0ae86**  
Free tier eligible  
Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
Root Device Type: ebs Virtualization type: hvm

**Instance Type** [Edit instance type](#)

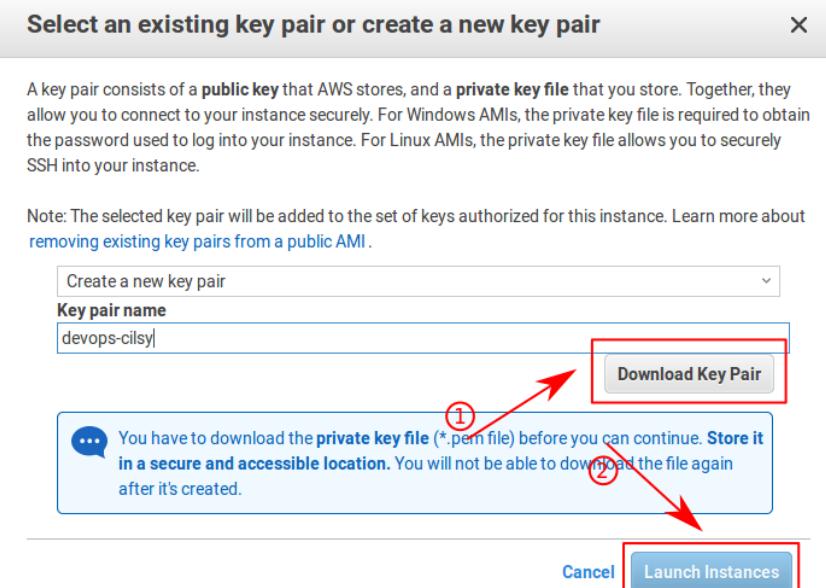
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

**Security Groups** [Edit security groups](#)

Cancel
Launch

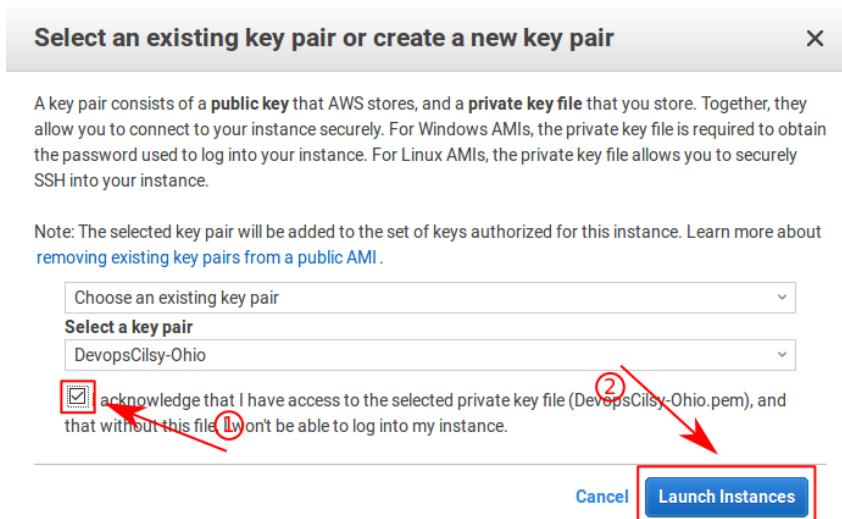
Selanjutnya kita hanya perlu klik **Launch instance** dan nanti akan muncul popout menu **new key pair**. Key pair ini berfungsi sebagai pengganti password dari instance yang sudah kita buat, jadi ketika kita **mengakses instance EC2** tersebut dengan hanya memasukan hostname dan ip public tanpa memasukan password.

Pilih **Create New Key Pair** lalu isikan nama keypair yang akan kita gunakan, selanjutnya **Download Key Pair** untuk menyimpannya keypair tersebut. Lalu klik **Launch Instances**.

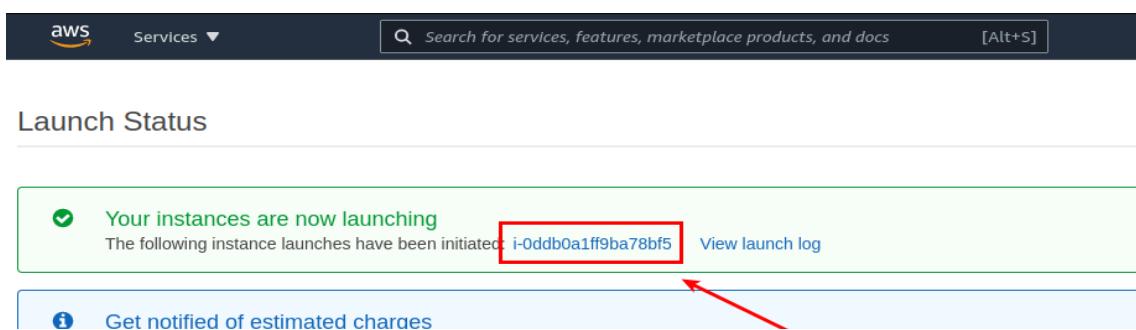


Apabila kita sudah pernah membuat key pair sebelumnya, kita hanya perlu memilih menu **Choose an exiting key pair** dan pilih nama keypair yang sudah pernah kita buat. Jangan lupa ceklis bagian **I acknowledge** lalu klik **Launch Instances**.

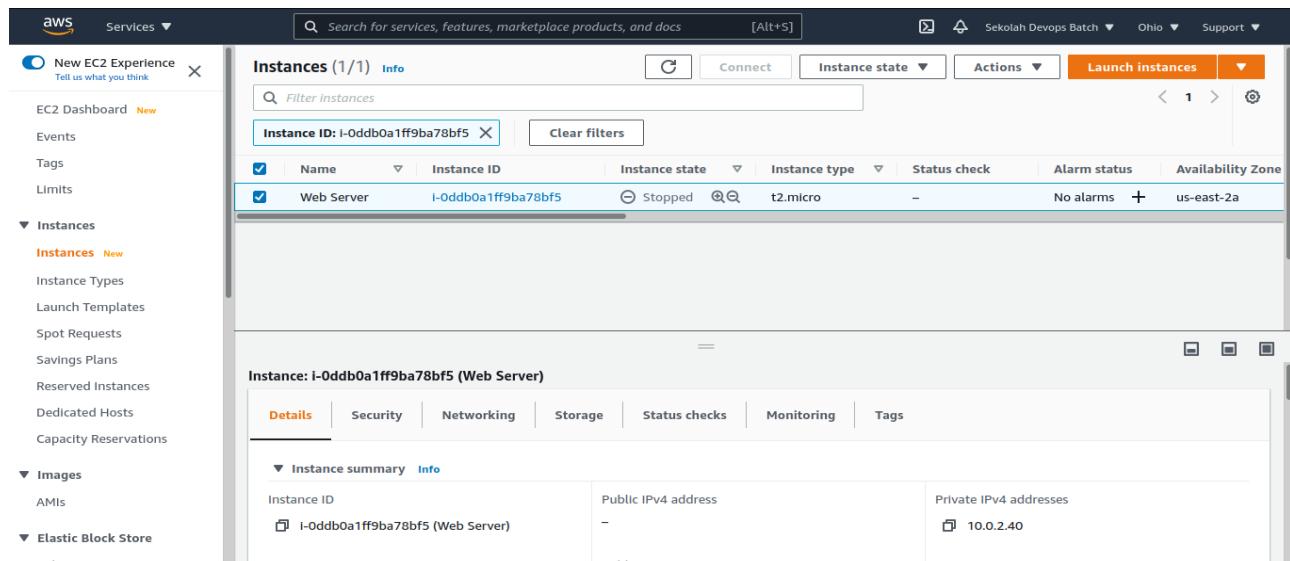
**Catatan :** Pastikan bahwa kita memiliki keypair, karena kita tidak akan bisa login ke instance tanpa keypair.



Setelah kita klik **Launch Instance**, kita akan diarahkan ke **Launch Status**.



Klik **id dari instance** yang sudah kita buat, dengan begitu kita akan diarahkan ke instance yang sudah kita buat seperti dibawah ini.



The screenshot shows the AWS EC2 Instances page. On the left sidebar, under 'Instances', 'Instances' is selected. In the main content area, the 'Instances (1/1) Info' section displays one instance. The instance details are as follows:

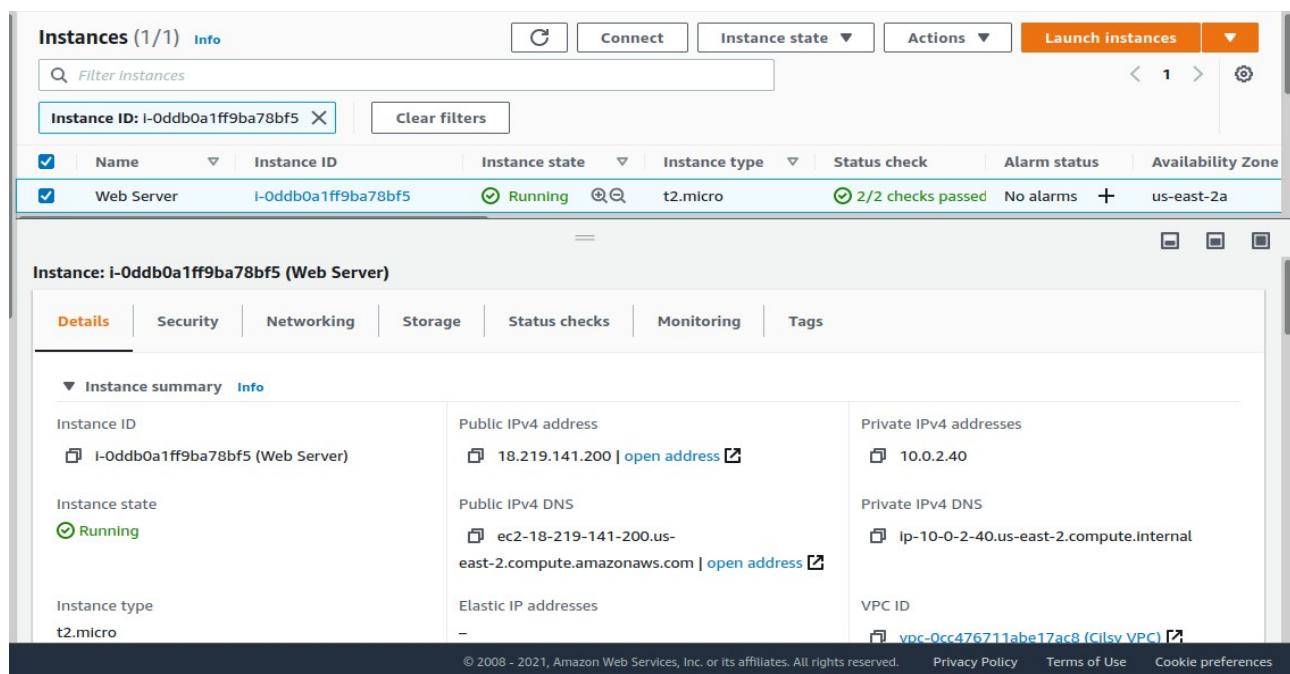
- Instance ID:** i-0ddb0a1ff9ba78bf5
- Name:** Web Server
- Instance state:** Stopped
- Instance type:** t2.micro
- Status check:** -
- Alarm status:** No alarms
- Availability Zone:** us-east-2a

Below the table, the 'Instance: i-0ddb0a1ff9ba78bf5 (Web Server)' details are shown. The 'Details' tab is selected, displaying the following information:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0ddb0a1ff9ba78bf5 (Web Server)	-	10.0.2.40

Bagian **dashboard** hanya akan menampilkan instance sesuai id instance yang sudah kita buat tadi, untuk melihat semua kita hanya perlu menekan tombol close di bagian search.

Untuk melihat detail dari instance yang sudah kita buat kita hanya perlu menekan tombol **tab Description** yang ada dibagian bawah. Disana kita bisa lihat informasi mengenai IP Public, DNS, Storage, Private IP dll. Jika kita lihat maka akan muncul seperti dibawah ini.



The screenshot shows the AWS EC2 Instances page with the same instance now in a **Running** state. The 'Instances (1/1) Info' section shows the following details:

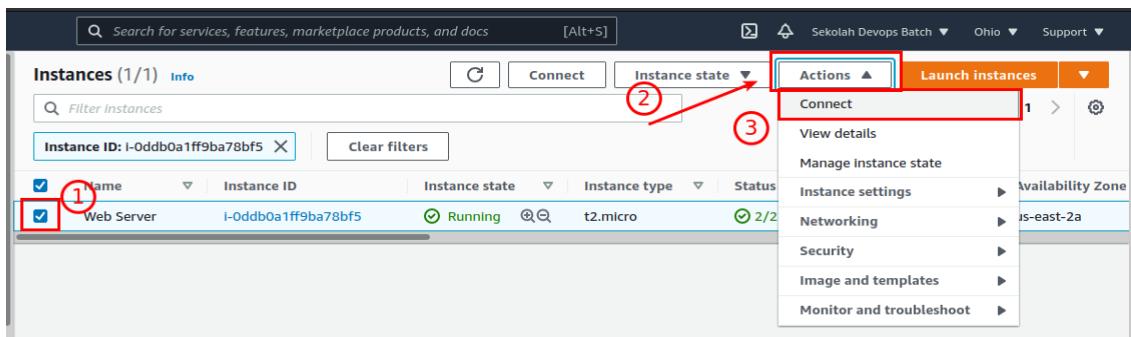
- Instance ID:** i-0ddb0a1ff9ba78bf5
- Name:** Web Server
- Instance state:** Running
- Instance type:** t2.micro
- Status check:** 2/2 checks passed
- Alarm status:** No alarms
- Availability Zone:** us-east-2a

Below the table, the 'Instance: i-0ddb0a1ff9ba78bf5 (Web Server)' details are shown. The 'Details' tab is selected, displaying the following information:

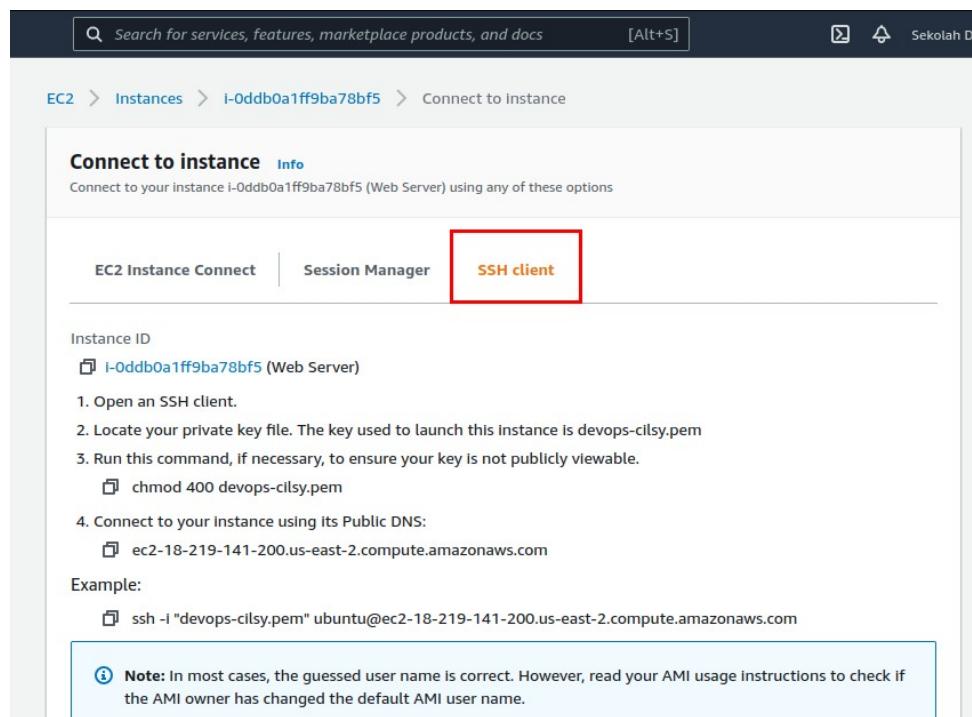
Instance ID	Public IPv4 address	Private IPv4 addresses
i-0ddb0a1ff9ba78bf5 (Web Server)	18.219.141.200   open address	10.0.2.40

Other tabs available include **Security**, **Networking**, **Storage**, **Status checks**, **Monitoring**, and **Tags**.

Untuk mengetahui informasi mengenai bagaimana cara kita **mengakses instance** tersebut, kita hanya perlu memilih tombol **Connect** disamping **Launch Instance** diatas. Maka akan muncul informasi mengenai cara koneksi ke instance yang sudah kita buat.



Lalu akan ada beberapa opsi untuk Connect ke instance yang sudah dibuat yaitu **EC2 Instance Connect**, **Session Manager** dan **SSH Client**. Kali ini kita praktikan mengkoneksikan ke instance dengan metode **SSH Client**



### 6.6.1.1. Remote Instance via Linux

Agar kita dapat mengakses instance yang sudah kita buat, kita dapat melakukannya dengan menggunakan remote ssh. Di linux sendiri kita tidak perlu menginstall software pihak ketiga untuk melakukan remote karena pada terminal linux kita sudah bisa langsung melakukan remote ssh. Tidak seperti windows yang harus menginstallkan putty terlebih dahulu.

Untuk melakukan remote sendiri kita membutuhkan **keypair dengan format \*.pem** yang sudah kita download sebelumnya di instance EC2 AWS. Berikut tahap remote via terminal.

Pertama buka software terminal di linux yang kalian gunakan, lalu masuk ke direktori dimana file keypair tersebut berasal.

Selanjutnya berikan hak akses pada keypair agar dapat kita execute/digunakan.

```
chmod 400 devops-cilsy.pem
```

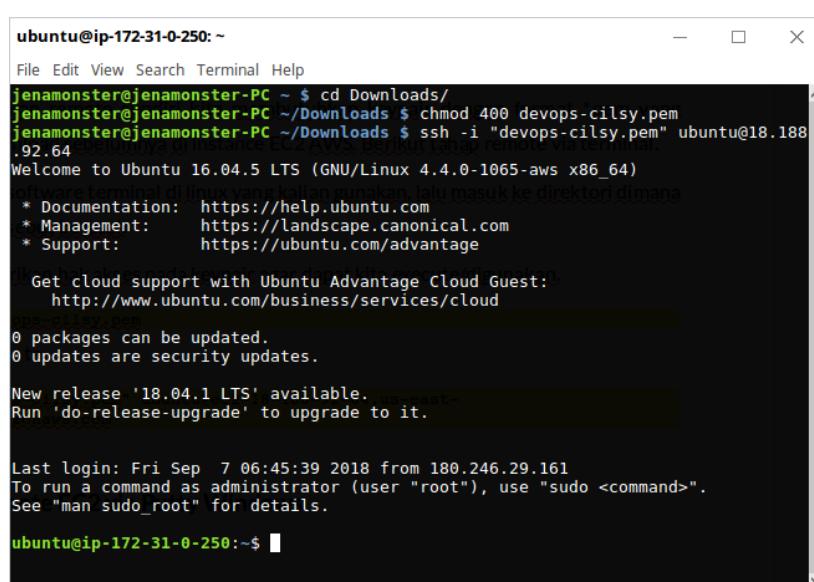
Setelah itu remote instance menggunakan alamat ip public atau dns yang sudah diberikan.

```
ssh -i "devops-cilsy.pem" ubuntu@ec2-3-17-187-181.us-east-2.compute.amazonaws.com
```

atau

```
ssh -i "devops-cilsy.pem" ubuntu@18.188.92.64
```

Maka hasilnya akan muncul seperti gambar dibawah ini.



```
ubuntu@ip-172-31-0-250: ~
File Edit View Search Terminal Help
jenamonster@jenamonster-PC ~ $ cd Downloads/
jenamonster@jenamonster-PC ~/Downloads $ chmod 400 devops-cilsy.pem
jenamonster@jenamonster-PC ~/Downloads $ ssh -i "devops-cilsy.pem" ubuntu@18.188
.92.64
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1065-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Sep  7 06:45:39 2018 from 180.246.29.161
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-0-250:~$
```



Maka dengan ini kita sudah bisa melakukan akses dan modifikasi pada linux instance yang sudah kita buat di Amazon EC2.

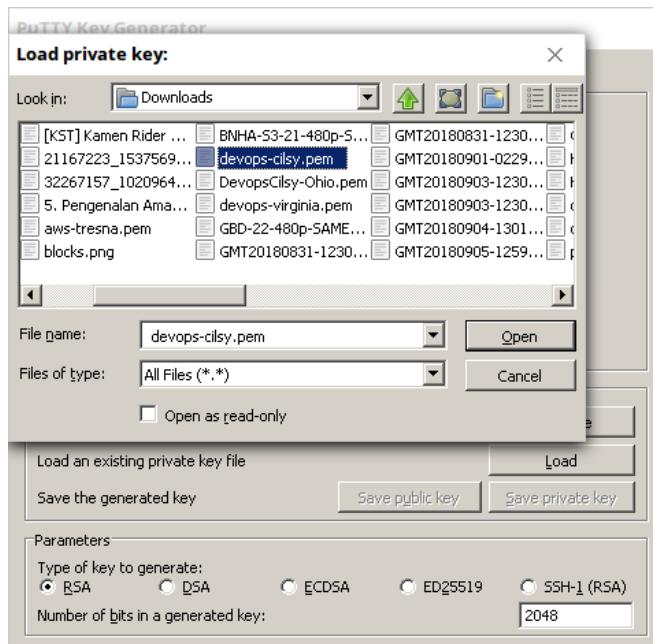
### **6.6.1.2. Remote EC2 via Putty Windows**

Untuk melakukan akses remote ke instance melalui windows kita membutuhkan software ketiga yang bernama putty, kita bisa download putty tersebut di website resminya <https://www.putty.org/>. Setelah itu install sampai selesai.

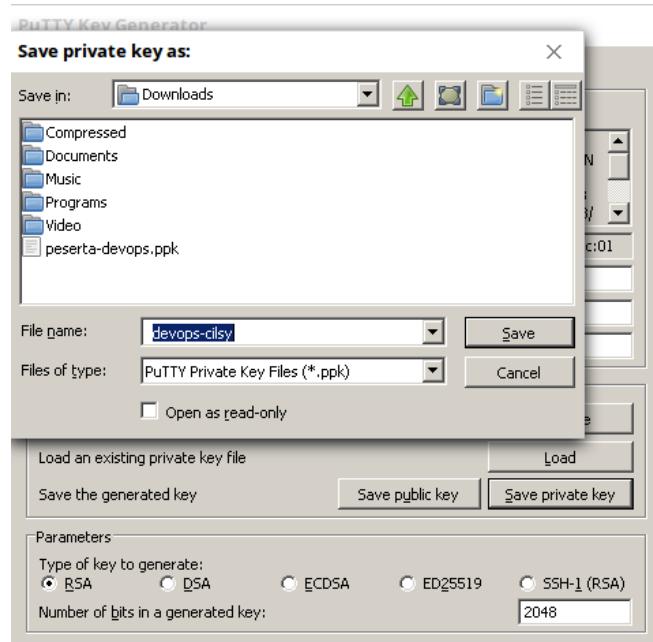
Setelah terinstall kita buka puttygen untuk melakukan generate pada keypair yang sudah kita download sebelumnya, karena putty sendiri tidak support untuk menggunakan keypair format \*.pem maka kita harus ubah menjadi file \*.pkk.



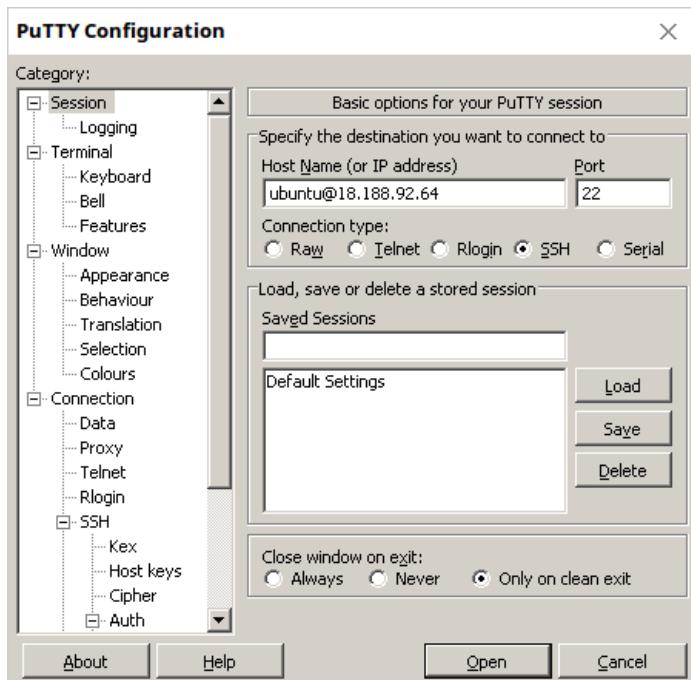
Klik load pada bagian action untuk mencari file keypair, pada bagian type pilih All File (\*.\*) agar bisa menampilkan semua file dan pilih keypair tersebut.



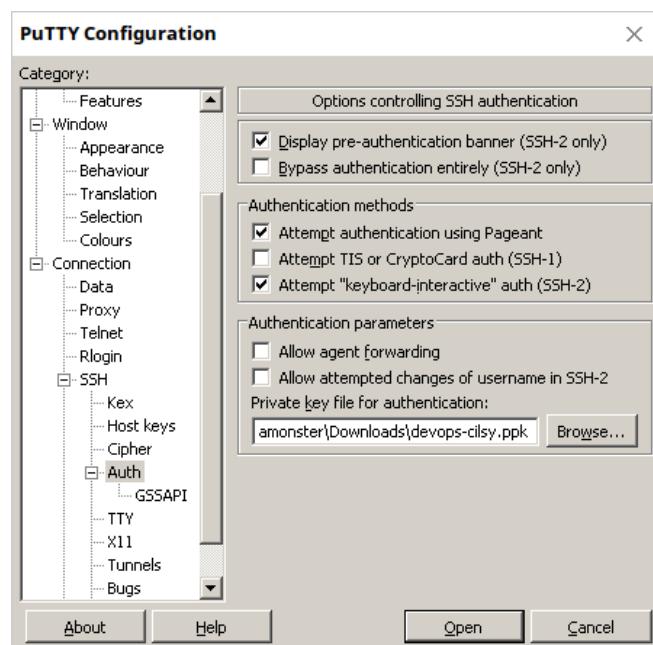
Klik save private key untuk menyimpan keypair menjadi file \*.ppk, setelah itu close dari aplikasi putty generator.



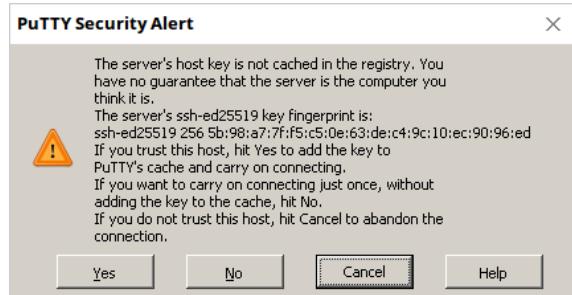
Buka aplikasi Putty untuk melakukan akses ke instance, setelah itu isikan username dan ip public dari instance yang akan kita remote.



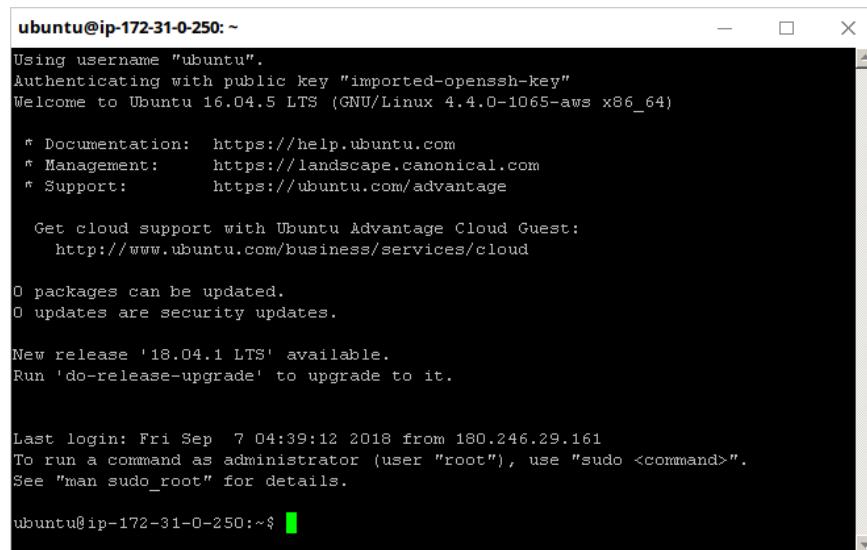
Sebelum klik open, pilih terlebih dahulu menu Auth yang ada di categori menu connection > SSH > Auth. Klik tombol browse dan cari file keypair.ppk yang sudah kita buat tadi.



Klik open lalu kita akan diarahkan ke pop out security alert, pilih yes untuk melanjutkan.



Dengan begitu kita sudah selesai login ke instance EC2 yang sudah kita buat, dan hasilnya adalah sebagai berikut ini.



```
ubuntu@ip-172-31-0-250: ~
Using username "ubuntu".
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1065-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 Get cloud support with Ubuntu Advantage Cloud Guest:
   http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

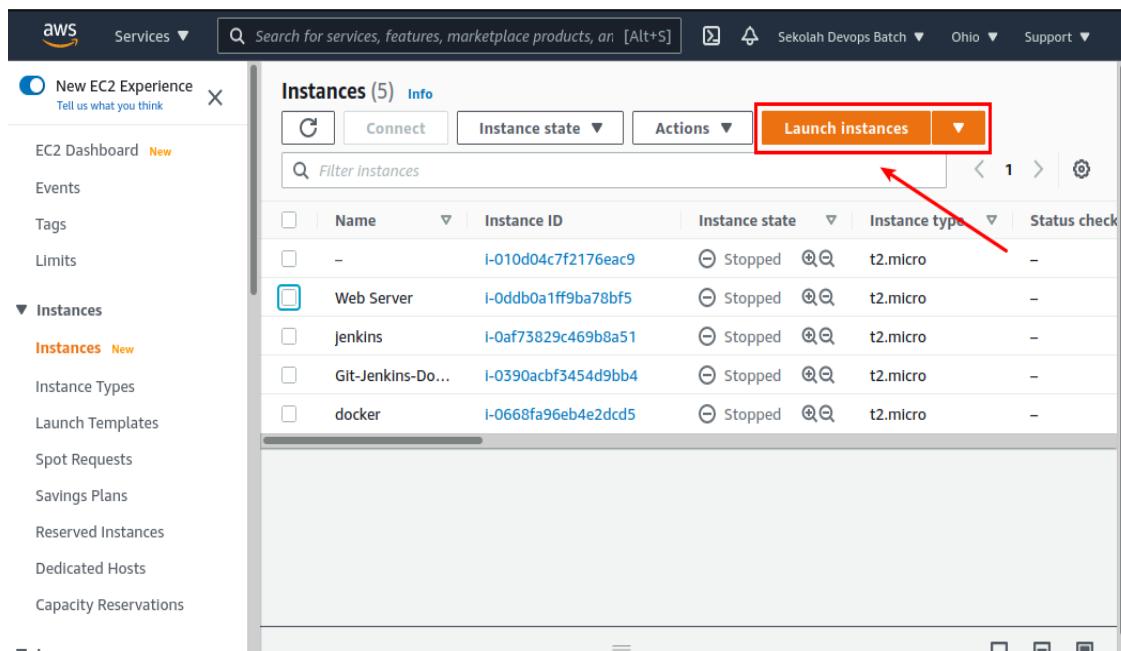
New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Sep  7 04:39:12 2018 from 180.246.29.161
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

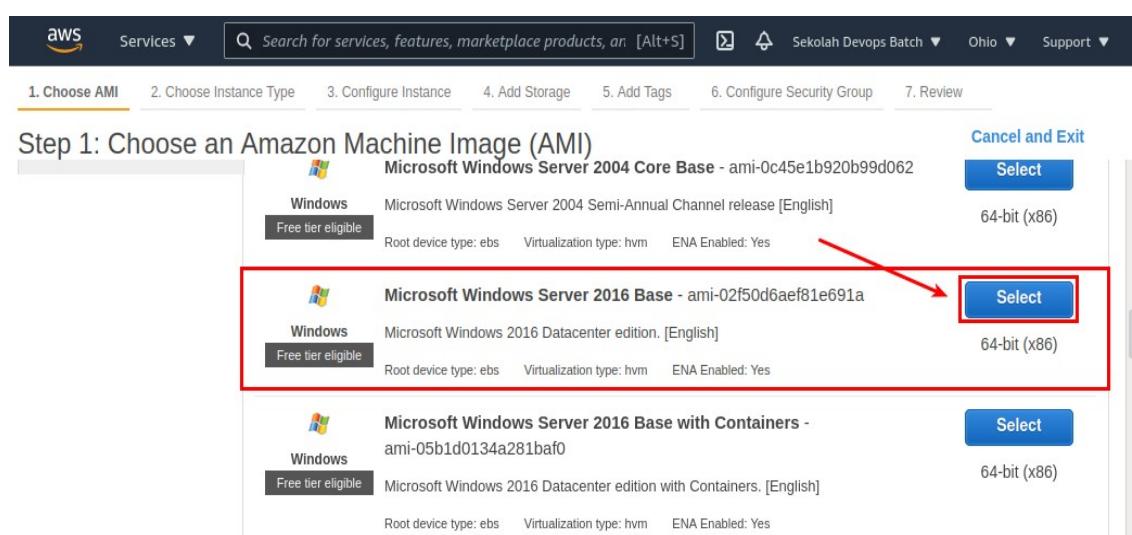
ubuntu@ip-172-31-0-250:~$
```

## 6.6.2. Membuat Windows Server EC2 Instance

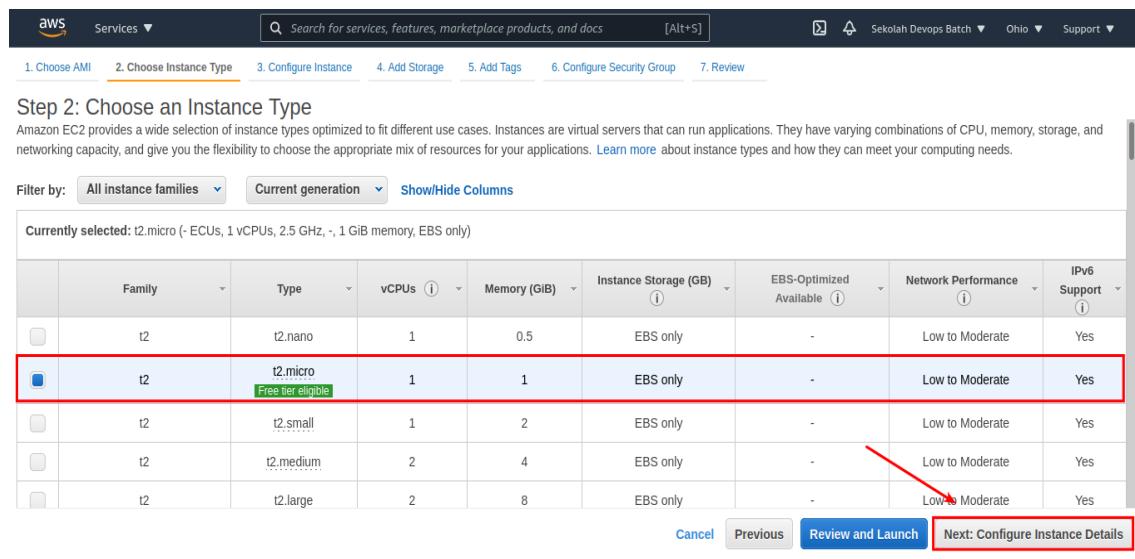
Setelah kita membuat sebuah **EC2 Instance** menggunakan **OS GNU/linux Ubuntu Server**, sekarang kita akan coba untuk membuat EC2 menggunakan **OS Windows server**. Tata cara yang digunakan tidak jauh berbeda, hanya pada pemilihan OS nya saja. Untuk memulainya, klik **Launch Instance**.



Lalu pilih base image atau **OS Windows Server 2016 Base**, setelah itu klik select.



Selanjutnya kita akan memilih spesifikasi dari instance yang akan kita buat. Kita akan memilih **t2.micro**. Selanjutnya klik **Configure instance detail**.



**Step 2: Choose an Instance Type**

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

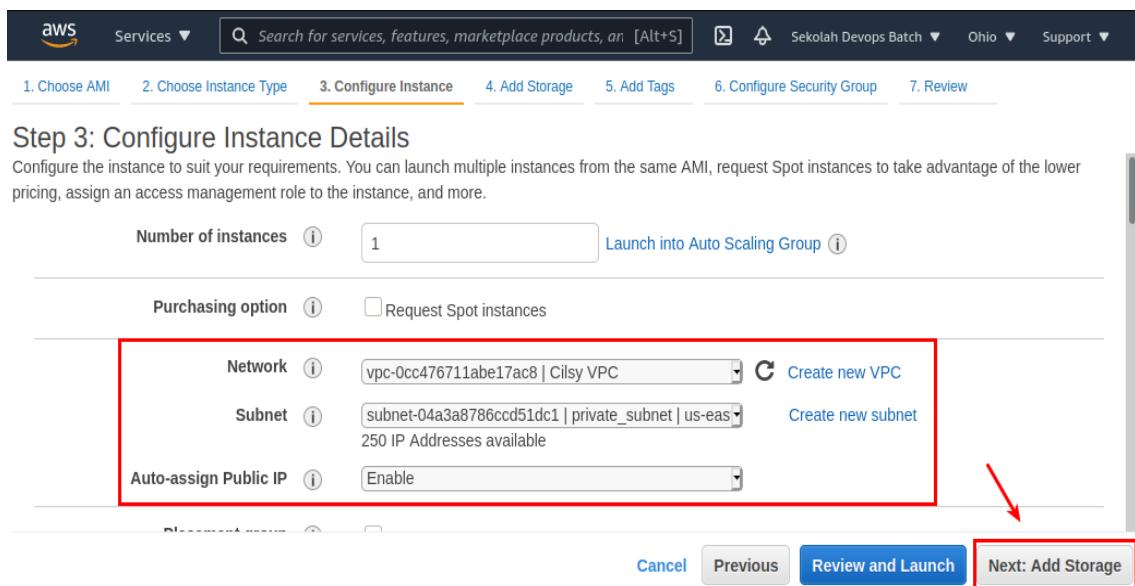
Filter by: All instance families ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	<b>t2.micro</b> <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch **Next: Configure Instance Details**

Selanjutnya adalah bagian **Configure Instance Detail**, atur sesuai dengan konfigurasi sebelumnya pada saat membuat linux. Tapi ada yang bebeda disini karena akan ada tambahan Elastic GPU di detail bawah, ini dikarenakan windows server berbasis GUI dengan menggunakan grafis. kita lanjut ke storage configuration.



**Step 3: Configure Instance Details**

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: Request Spot instances

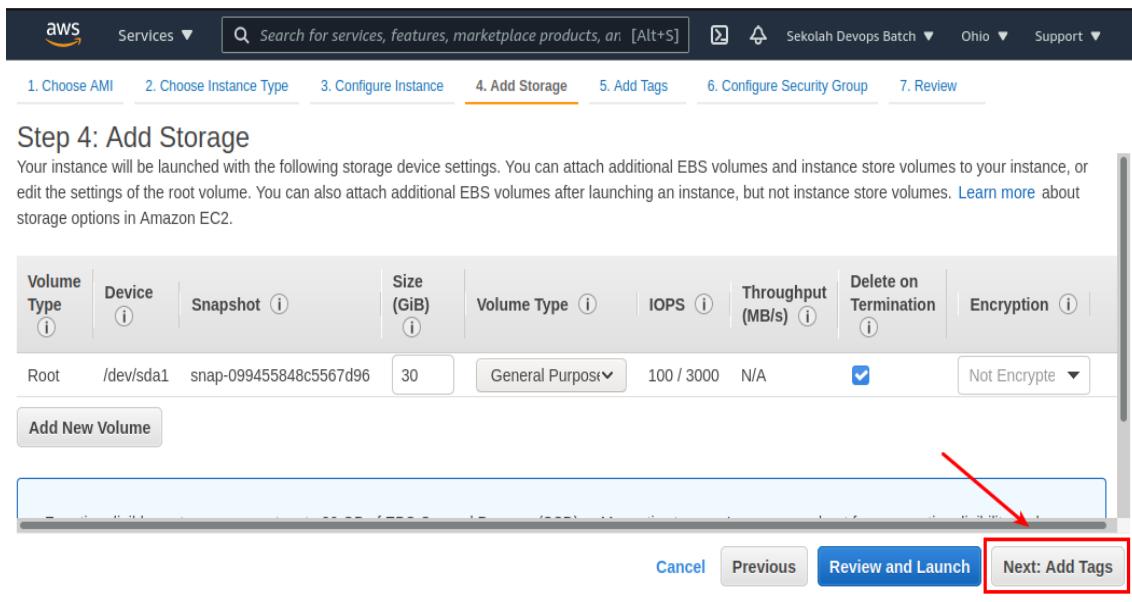
Network: vpc-0cc476711abe17ac8 | Cilsy VPC

Subnet: subnet-04a3a8786cccd51dc1 | private\_subnet | us-east-1

Auto-assign Public IP: Enable

Buttons: Cancel, Previous, Review and Launch, Next: Add Storage

Dikarenakan menggunakan sistem operasi windows, disini kita isikan sizenya dengan **ukuran 30GB**, setelah itu **Next Configuration**.



**Step 4: Add Storage**

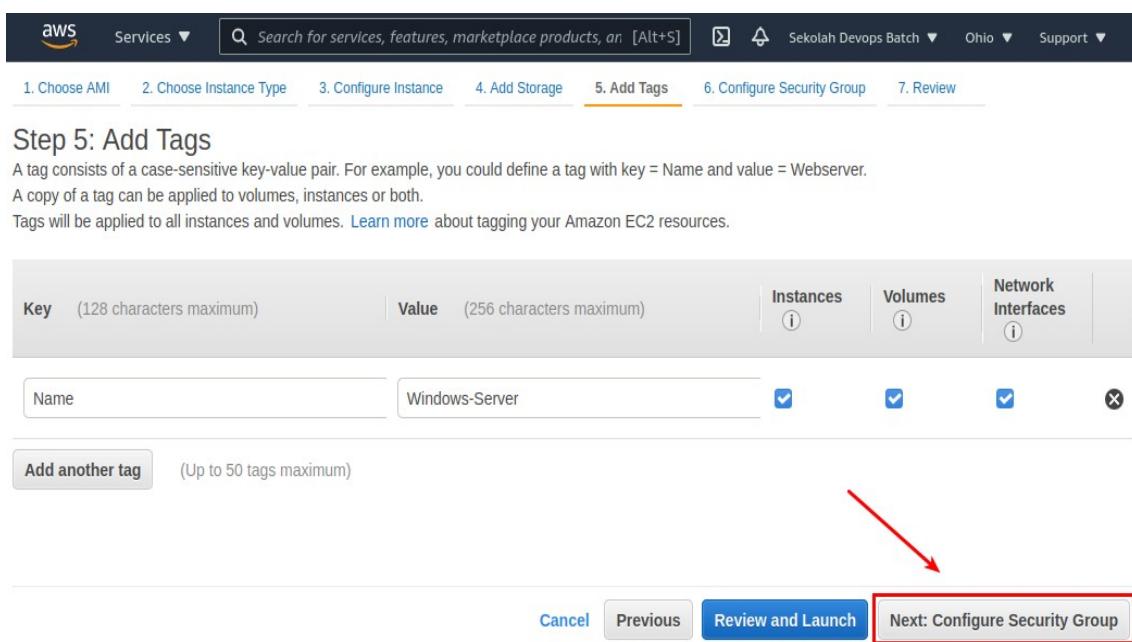
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-099455848c5567d96	30	General Purpose	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

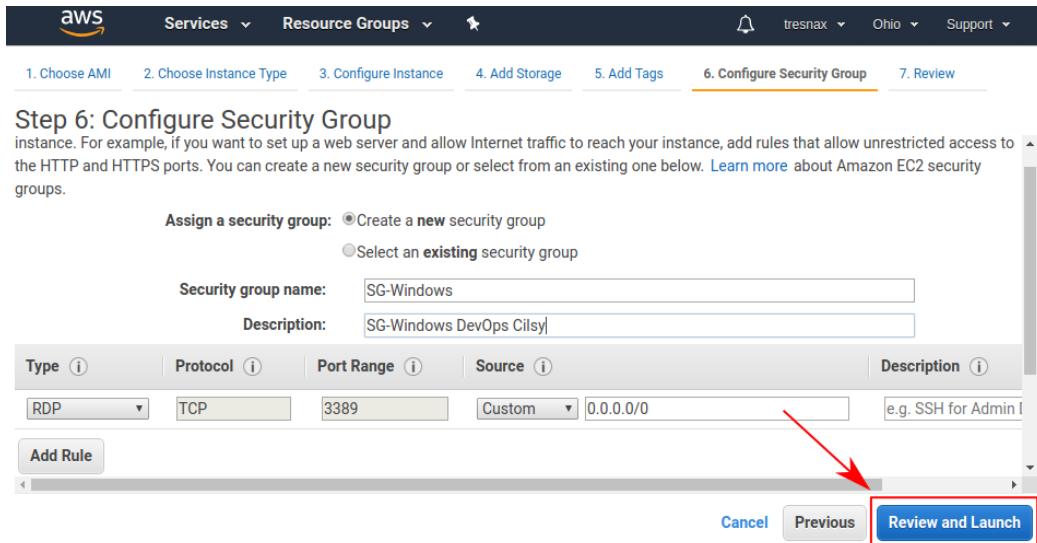
Add New Volume

Buttons: Cancel, Previous, Review and Launch, Next: Add Tags

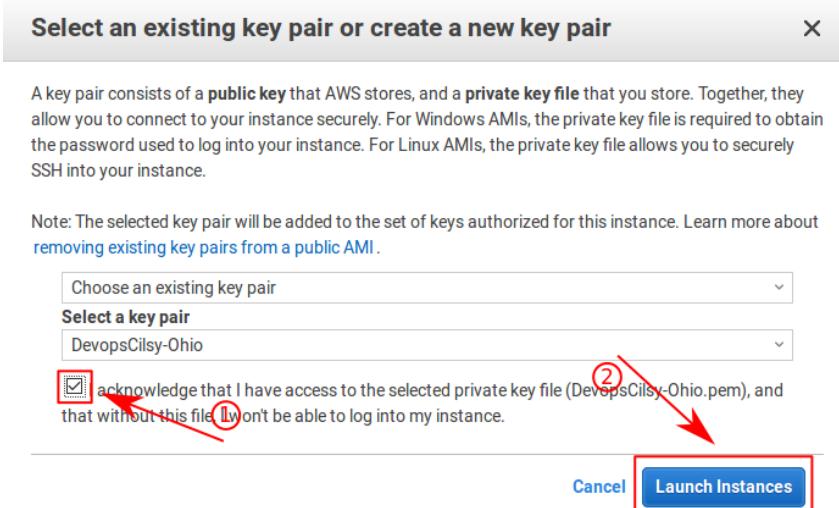
Isikan **tag** dengan key berupa name dan value dengan **Windows Server**. Selanjutnya **Next**.



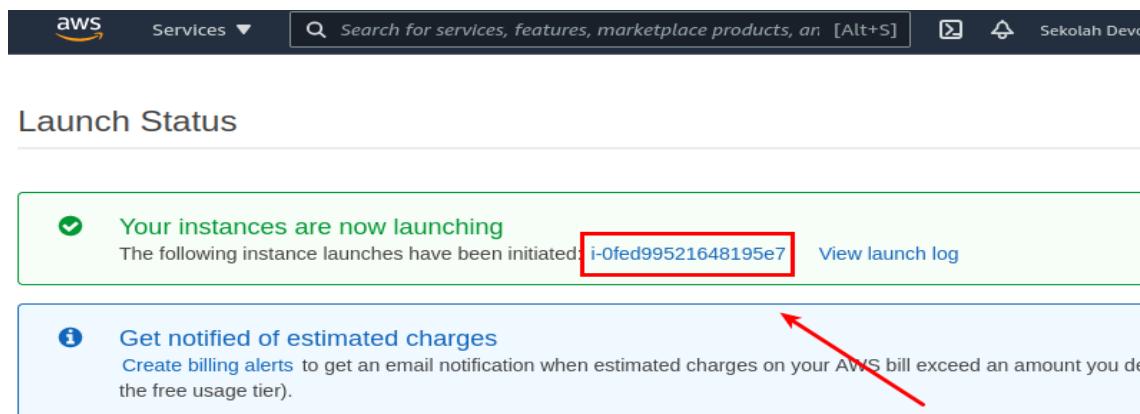
Selanjutnya adalah **Security Group**, disini akan berisi **RDP** karena kita menggunakan remote desktop setelah itu klik **Next**.



Selanjutnya kita hanya perlu klik Launch instance dan nanti akan muncul popout menu new key pair. Gunakan keypair yang sudah ada, kita hanya perlu memilih menu **choose an exiting key pair** dan pilih nama keypair yang sudah pernah kita buat. Jangan lupa ceklis bagian **I acknowledge** lalu klik **Launch Instances**.

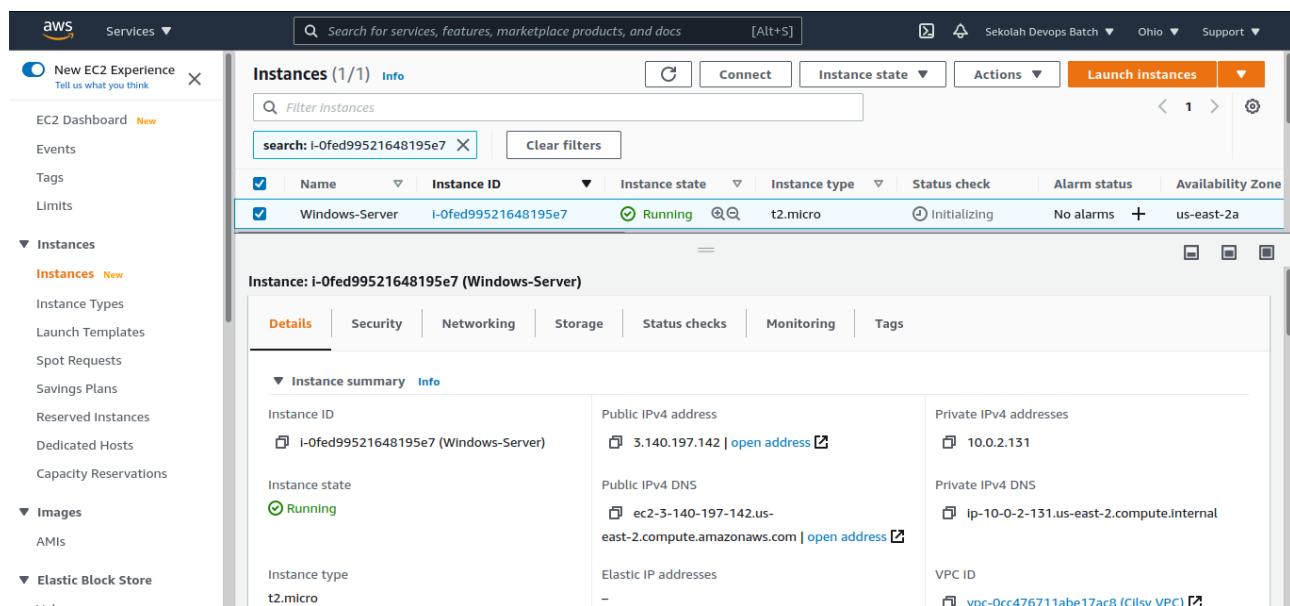


Setelah itu kita akan diarahkan ke **Launch Status**.



The screenshot shows the AWS Launch Status page. At the top, there is a green success message: "Your instances are now launching". Below it, a red box highlights the instance ID "i-0fed99521648195e7". To the right of the ID is a "View launch log" link. Below this, there is a blue info message: "Get notified of estimated charges". A red arrow points from the text "Create billing alerts" to this message. The page also includes a section titled "How to connect to your instances" with a note about instance launching times.

Klik id dari instance yang sudah kita buat, dengan begitu kita akan diarahkan ke instance yang sudah kita buat seperti dibawah ini.



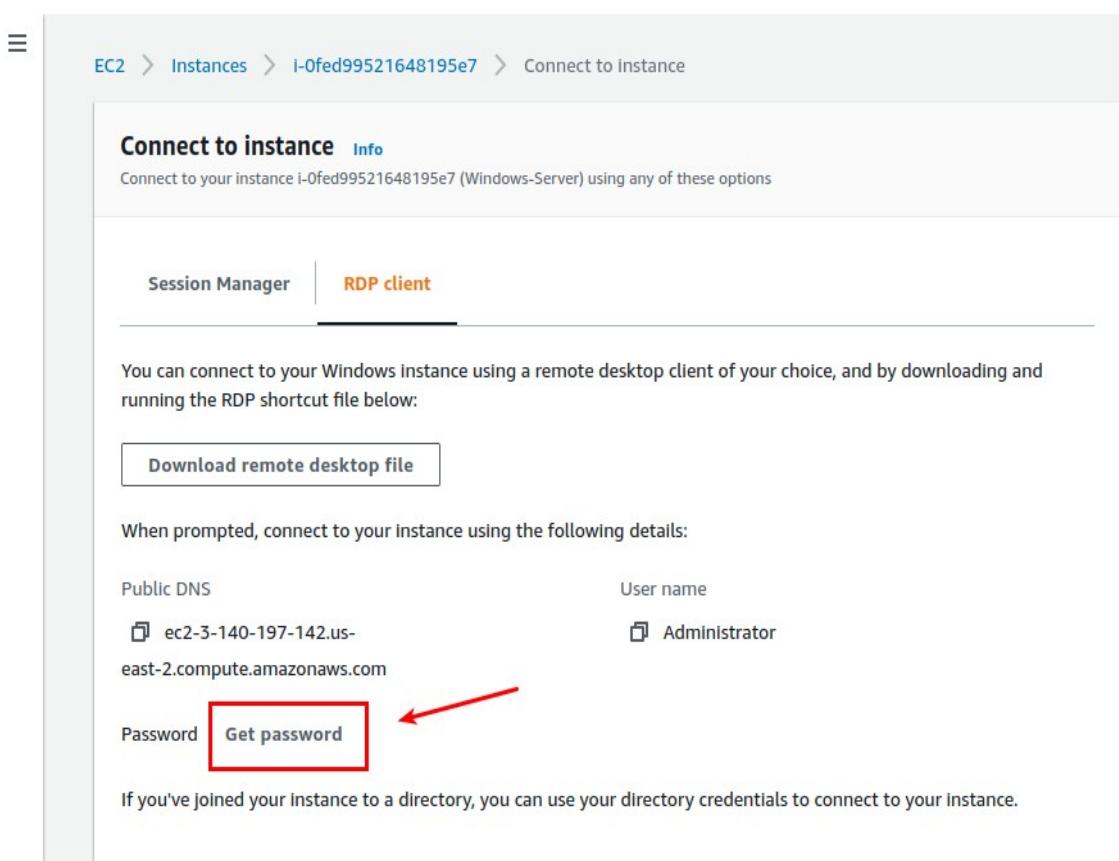
The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the "Instances" section, "Instances" is selected. In the main content area, the "Instances (1/1)" table shows one instance named "Windows-Server" with the ID "i-0fed99521648195e7", which is currently "Running". The "Actions" dropdown menu is open, with "Launch instances" highlighted. Below the table, the "Instance: i-0fed99521648195e7 (Windows-Server)" details page is displayed, showing the instance's summary and various configuration tabs like Details, Security, Networking, Storage, Status checks, Monitoring, and Tags.

Bagian dashboard hanya akan menampilkan instance sesuai id instance yang sudah kita buat tadi, untuk melihat semua kita hanya perlu menekan tombol close di bagian search.

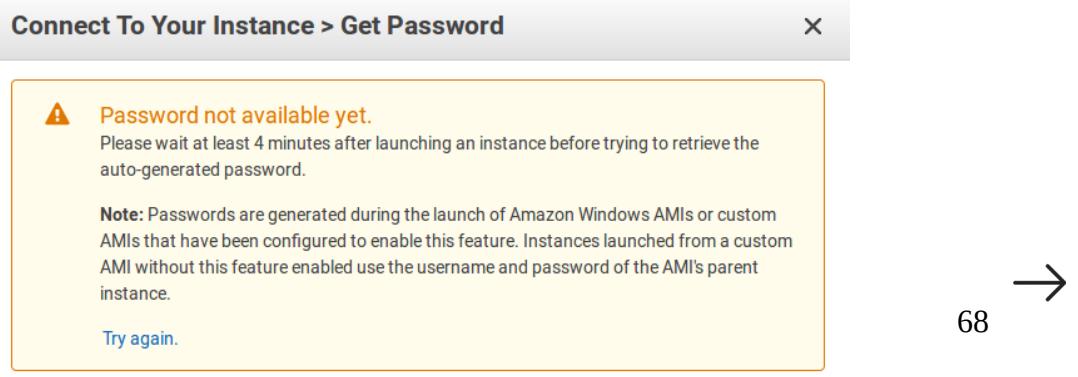
### **6.6.2.1. Remote Desktop Windows Server melalui Linux**

Untuk mengakses windows server mungkin akan sedikit berbeda dengan mengakses linux server, karena pada windows server kita diharuskan menggunakan remote desktop. Untuk melakukan remote kita klik bagian **Connect** di **dashboard instance EC2 AWS** hingga muncul seperti dibawah ini.

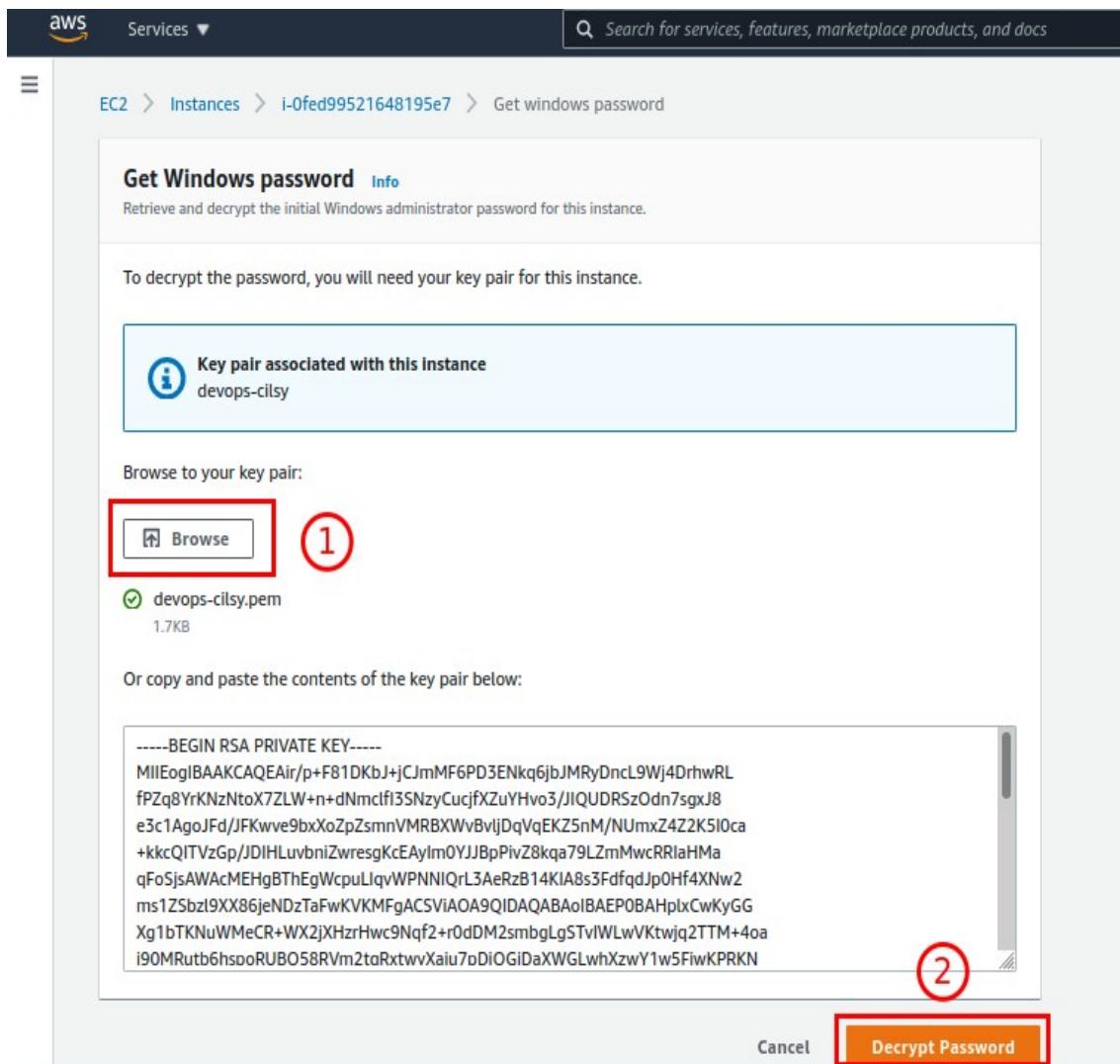
Klik bagian **Get Password** untuk mendapatkan password dari windows yang akan kita akses.



Apabila muncul seperti dibawah ini, berarti kita harus menunggu selama 4 menit setelah pembuatan agar kita dapat melakukan generate password.

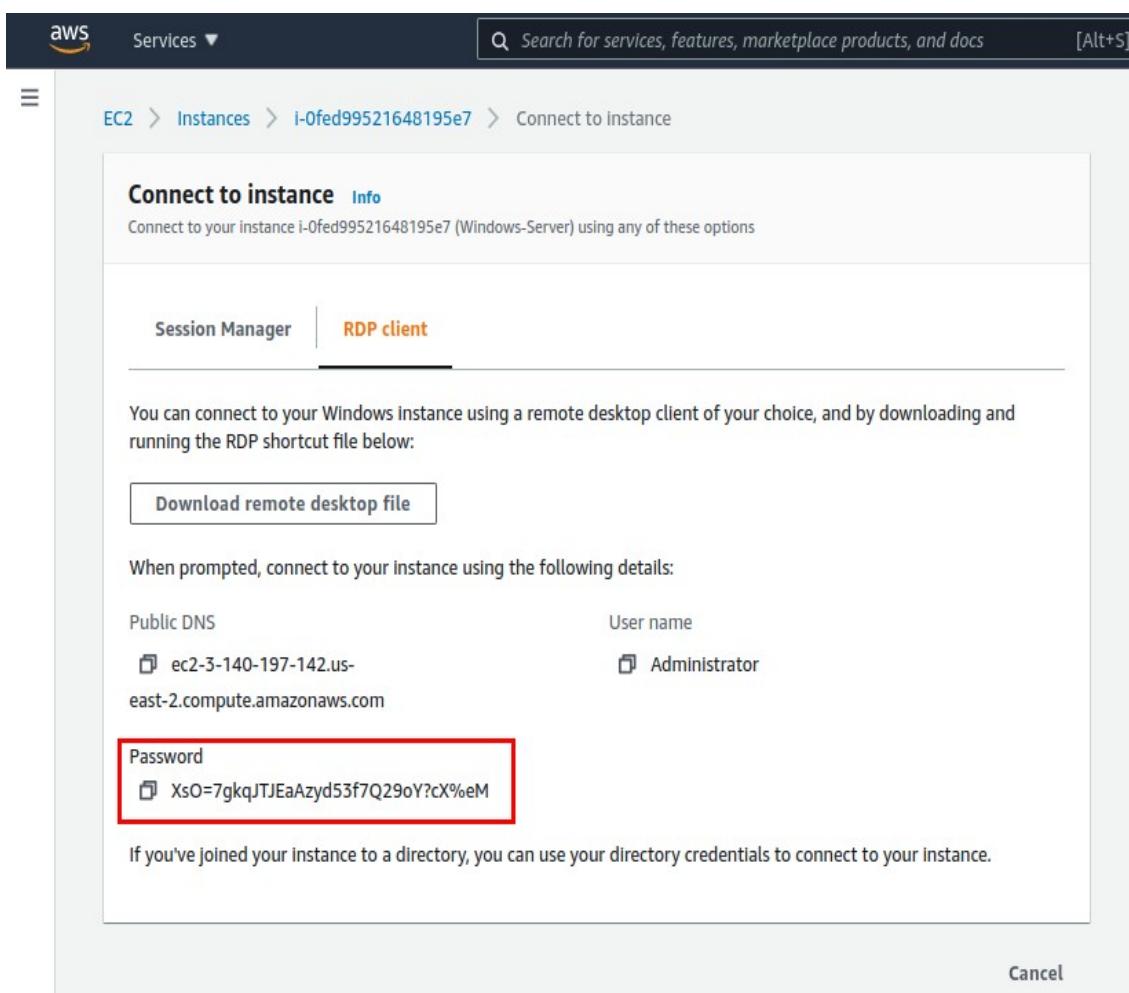


Selanjutnya akan muncul menu seperti dibawah ini, klik **browse** dan cari **keypair** yang sebelumnya sudah kita set untuk men enkripsi password dari windows server tersebut. Setelah itu klik **decrypted password**.



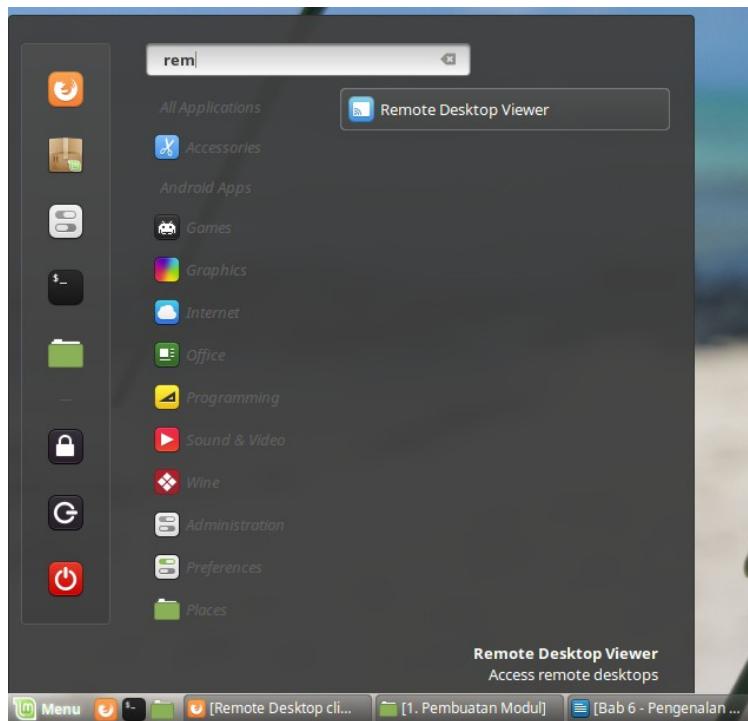
The screenshot shows the AWS EC2 Instances interface for getting a Windows password. It displays a key pair associated with the instance: 'devops-cilsy'. A 'Browse' button is highlighted with a red box and circled with a red number '1'. Below it, a checked checkbox for 'devops-cilsy.pem' is shown. A large text area contains a long RSA private key, with a red circle labeled '2' highlighting its right edge. At the bottom right, a red box highlights the 'Decrypt Password' button.

Setelah berhasil nanti kita akan diarahkan ke menu awal connect dan akan mendapatkan password seperti dibawah ini, kita hanya perlu **mencopy password** tersebut.

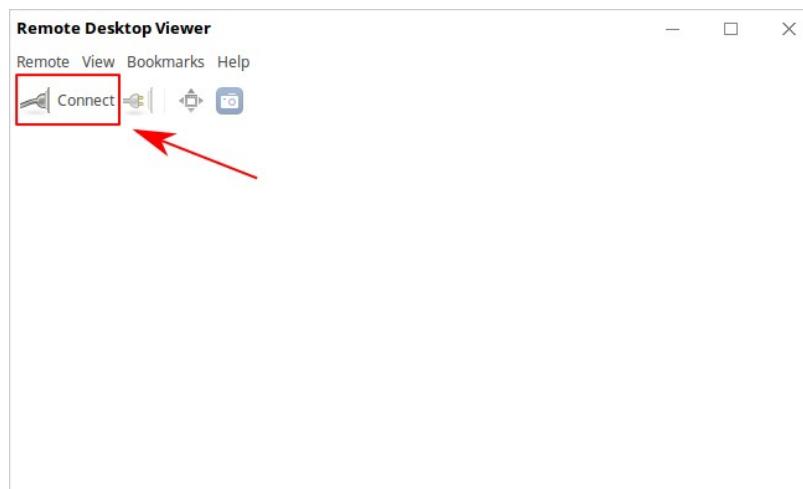


The screenshot shows the AWS EC2 Instances page for an instance named i-0fed99521648195e7. The URL in the browser bar is [EC2 > Instances > i-0fed99521648195e7 > Connect to instance](#). The main section is titled "Connect to instance" with an "Info" link. It instructs the user to connect to the Windows instance using any of the provided options. Two tabs are visible: "Session Manager" (disabled) and "RDP client" (selected). A note below states: "You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:". A button labeled "Download remote desktop file" is present. Below this, connection details are listed: "Public DNS" (ec2-3-140-197-142.us-east-2.compute.amazonaws.com) and "User name" (Administrator). A red box highlights the "Password" field, which contains the value "XsO=7gkqJTJEaAzyd53f7Q29oY?cX%eM". A note at the bottom says: "If you've joined your instance to a directory, you can use your directory credentials to connect to your instance." A "Cancel" button is located at the bottom right.

Untuk mengaksesnya, pada linux kita sudah disediakan software default yaitu **Remote Desktop Viewer** yang sudah bisa kita langsung akses di menu dash linux tersebut.

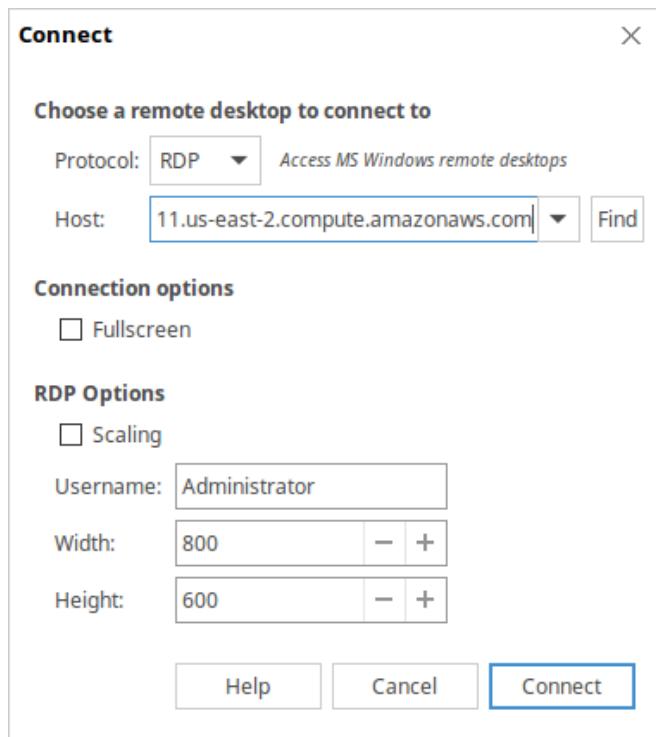


Selanjutnya akan muncul aplikasi remote desktop tersebut. Untuk melakukan remote kita perlu men-klik menu **Connect** yang ada di samping atas menu.

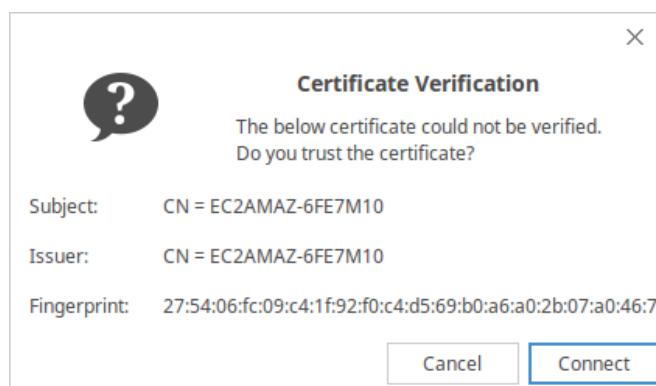


Selanjutkan akan muncul menu pop out baru, kita isikan alamat host yang akan kita remote. Bisa kita isikan dns atau ip public dari instance yang kita gunakan.

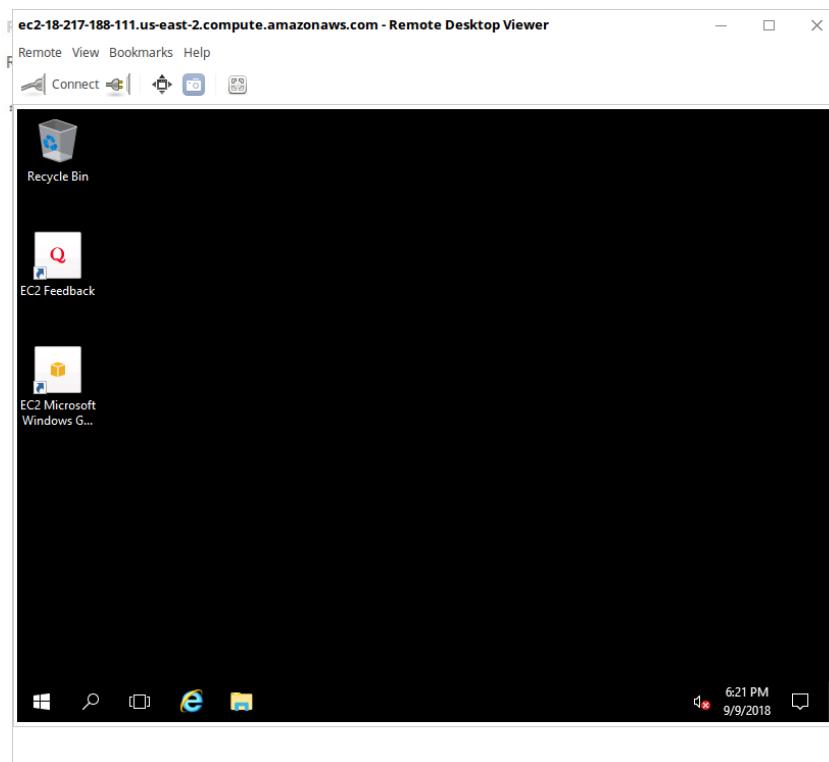
Isikan username dengan user yang tertera pada instance juga, lalu klik **Connect**.



Akan muncul pop out sertificate, klik **Connect** untuk meneruskan koneksi.



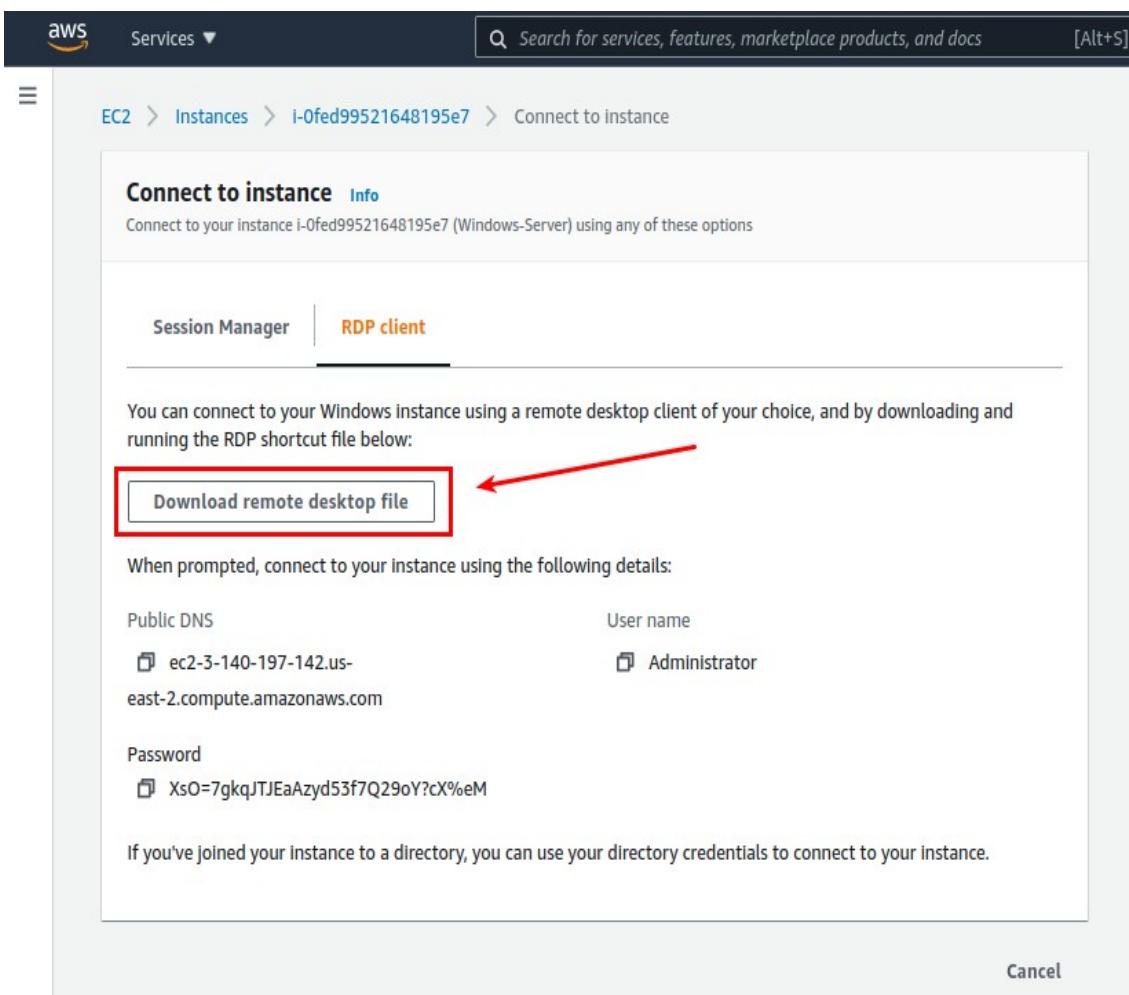
Setelah berhasil terkoneksi, remote desktop akan menampilkan isi layar dari desktop windows sever yang sudah kita buat sebelumnya. Dengan ini kita sudah dapat menggunakan windows server tersebut.



### **6.6.2.2. Remote Desktop Windows Server melalui Windows**

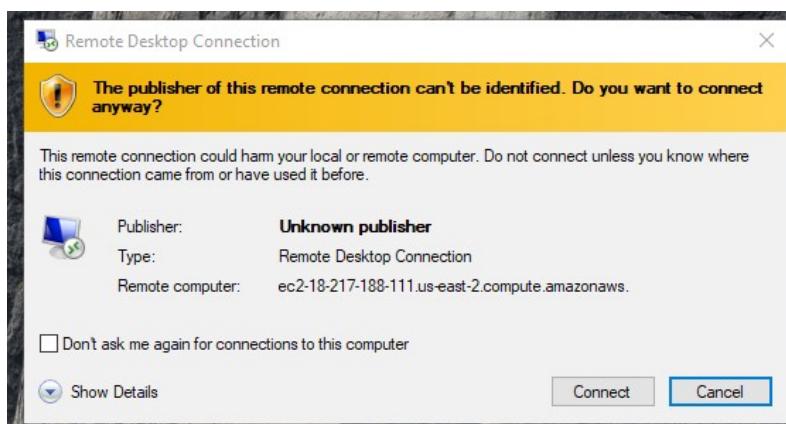
Pada bagian ini kita akan melakukan akses pada windows server melalui OS windows dengan menggunakan **Microsoft Remote Desktop**. Cara menggunakannya tidak berbeda jauh dengan yang sebelumnya pada OS linux.

Apabila menggunakan **Microsoft Remote Desktop**, AWS menyediakan save file yang bisa kita load sehingga kita tidak perlu memasukan host dan username pada koneksi yang akan kita tuju. Kita hanya perlu mendownload file tersebut di menu connect seperti dibawah ini.



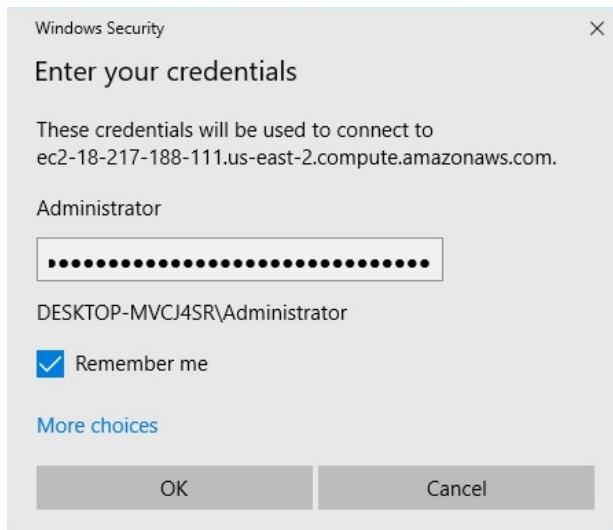
The screenshot shows the AWS EC2 Instances page for an instance named i-0fed99521648195e7. The 'Connect to instance' section is displayed, with the 'RDP client' tab selected. A red box highlights the 'Download remote desktop file' button, and a red arrow points to the download link below it. The page also displays connection details: Public DNS (ec2-3-140-197-142.us-east-2.compute.amazonaws.com), User name (Administrator), and Password (XsO=7gkqJTEaAzyd53f7Q29oY?cX%eM). A note at the bottom states: 'If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.'

Setelah itu buka file Remote Desktop yang sudah didownload tadi, nanti akan muncul popout seperti gambar dibawah ini, pilih connect untuk melanjutkan.



Tunggu sampai aplikasi tersebut mencoba untuk menyambungkan ke server yang kita tuju dan muncul form login, disini kita masukan password yang sudah

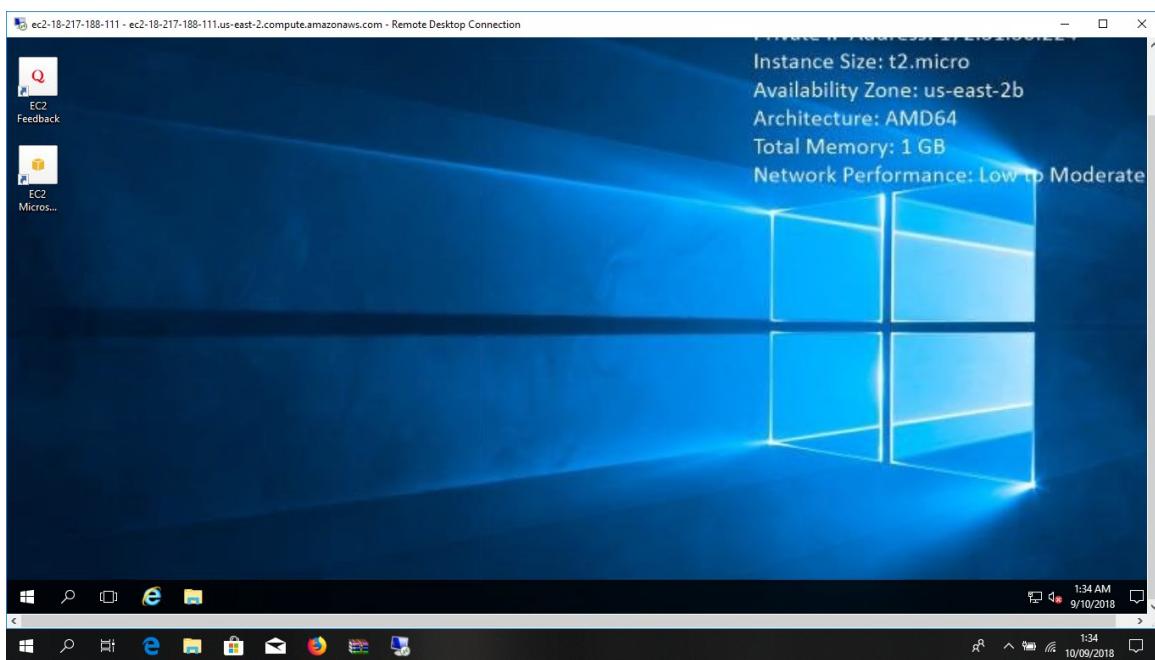
kita dapatkan di AWS Dashboard sebelumnya dengan menggunakan metode yang sama seperti pada saat menggunakan di OS Linux.



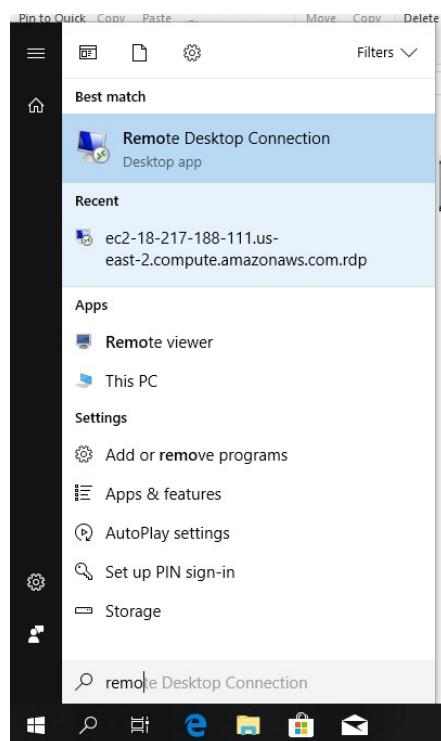
Selanjutnya pilih **Yes** untuk menyetujui certificate koneksi yang akan dihubungkan ke windows server tersebut.



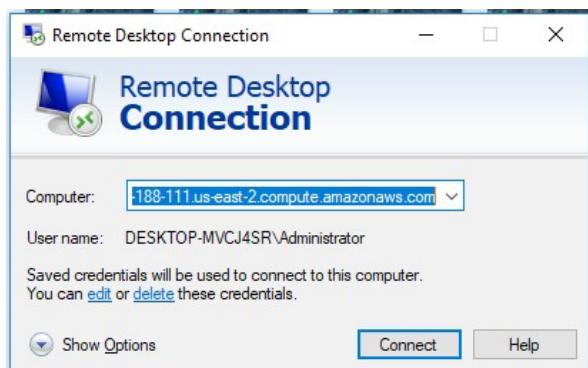
Tunggu sebentar hingga aplikasi menampilkan desktop dari windows server yang ada di AWS. Dengan begini kita sudah dapat mengakses windows server melalui microsoft remote desktop.



Selain itu kita juga dapat menggunakan cara manual, yaitu membuka aplikasi remote desktop terlebih dahulu tanpa menggunakan file remote desktop. Pertama kita buka aplikasi remote desktop.



Setelah muncul pop out aplikasi tersebut, masukan alamat dns ataupun public ip dari server windows yang kita miliki lalu klik **Connect**.

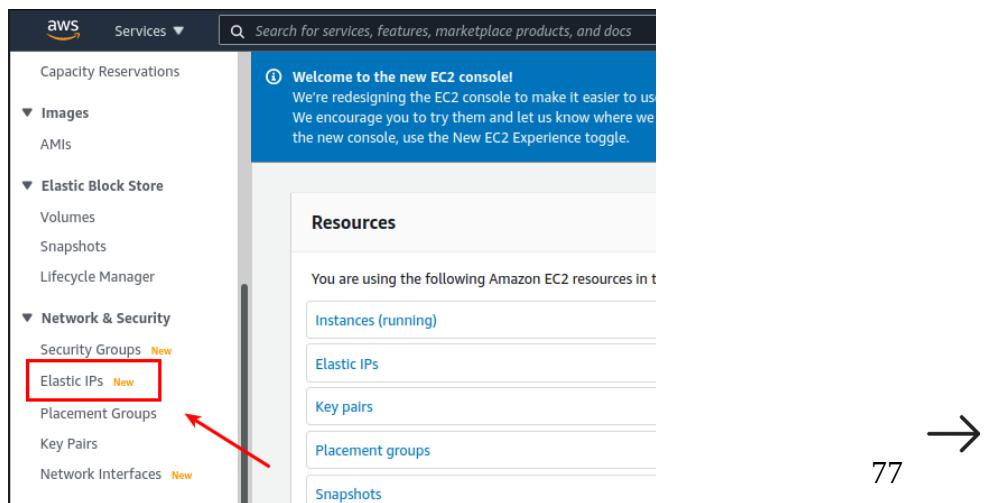


Setelah itu kita akan diminta memasukan password dari server tersebut, setelahnya kita akan langsung diarahkan ke desktop servernya. Dengan begini kita sudah dapat mengakses windows server melalui OS Linux dan OS Windows.

### 6.6.3. Elastic IP

Salah satu komponen untuk menghubungkan EC2 instance ke internet, adalah Elastic IP. Kelebihan Elastic IP dibandingkan dengan auto-assigned IP oleh subnet, yaitu : Elastic IP tidak akan berubah meskipun kita mematikan dan menyalakan kembali sebuah Instance. Dalam artian, Elastic IP adalah IP static, sedangkan auto-assigned IP adalah IP Dynamic. Untuk meminta Elastic IP dari AWS, ikut langkah – langkah berikut:

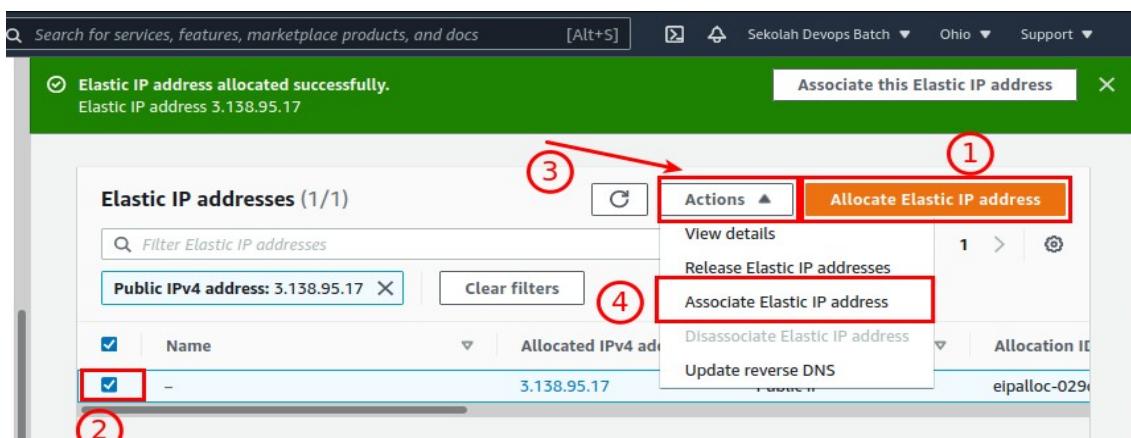
Masuk ke **EC2 Dashboard** lalu scroll ke bawah dan pilih **Elastic IP**.



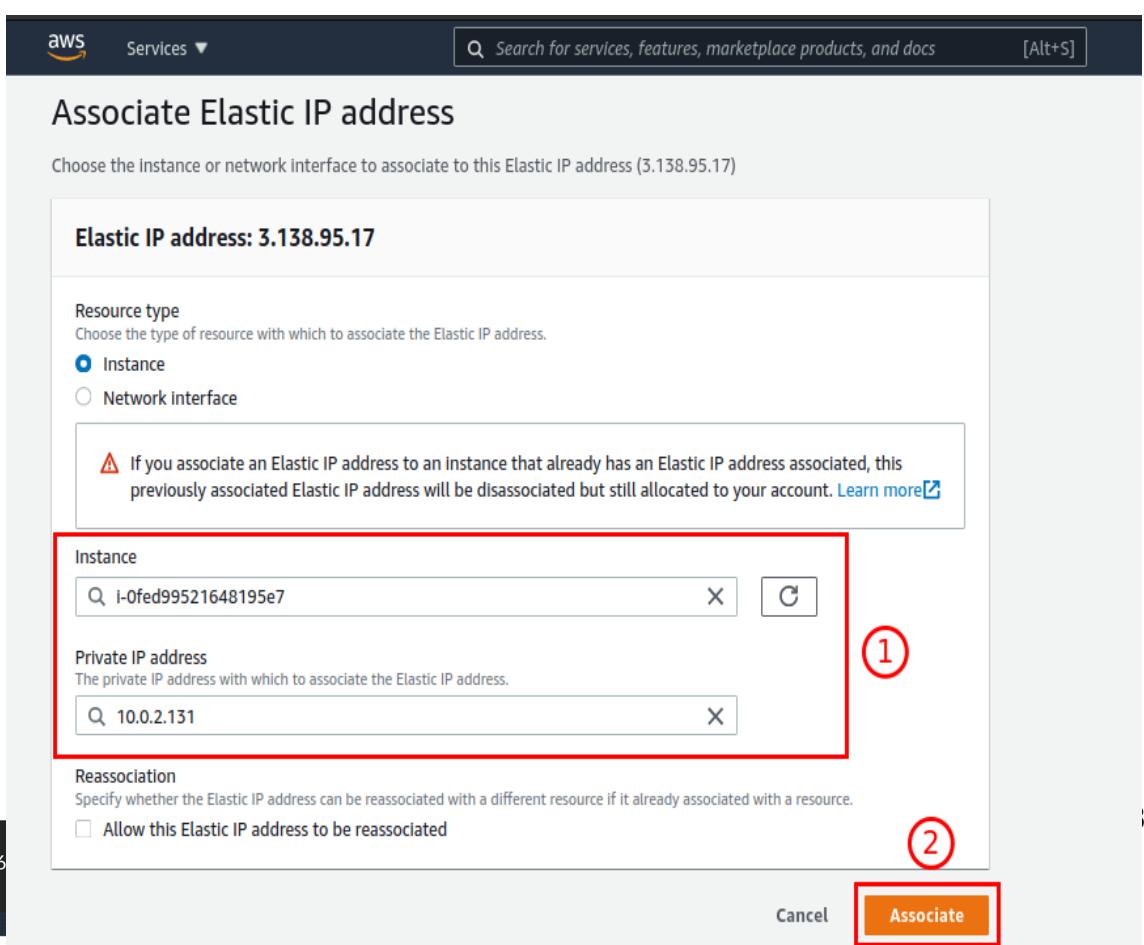
IP Address	Status	Action
18.207.111.111	Allocated	Release

Klik **Allocate new address** untuk mendapatkan IP Static kita, lalu klik **Allocate**. Jika sudah, maka akan muncul IP address baru. Tapi IP yang muncul belum terpakai, maka, kita harus menempelkan IP tersebut ke sebuah instances.

Klik kanan Elastic IP dan pilih *associate address*. Atau bisa juga dengan klik **Action** dan pilih *associate address*.



Lalu isi pilihan yang ada. Kita bisa memilih instance yang ingin kita assign IP baru, atau kita bisa atur IP baru untuk ditempelkan pada network interface yang ada di AWS. Setelah semuanya OK, klik **associate**.



**Associate Elastic IP address**

Choose the instance or network interface to associate to this Elastic IP address (3.138.95.17)

**Elastic IP address: 3.138.95.17**

**Resource type**  
Choose the type of resource with which to associate the Elastic IP address.

Instance

Network interface

**Instance**  
The instance with which to associate the Elastic IP address.

**Private IP address**  
The private IP address with which to associate the Elastic IP address.

**Reassociation**  
Specify whether the Elastic IP address can be reassigned with a different resource if it already associated with a resource.

Allow this Elastic IP address to be reassigned

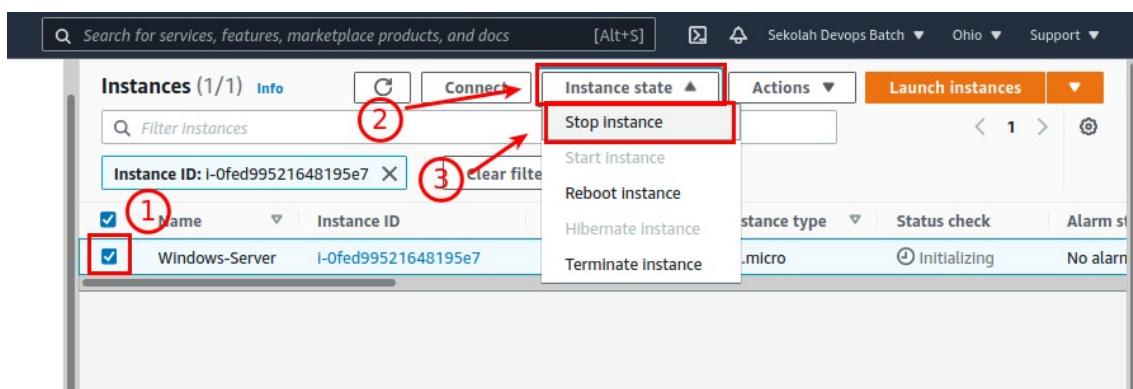
Cancel **Associate**

## 6.6.4. Mematikan dan Menghapus Instance

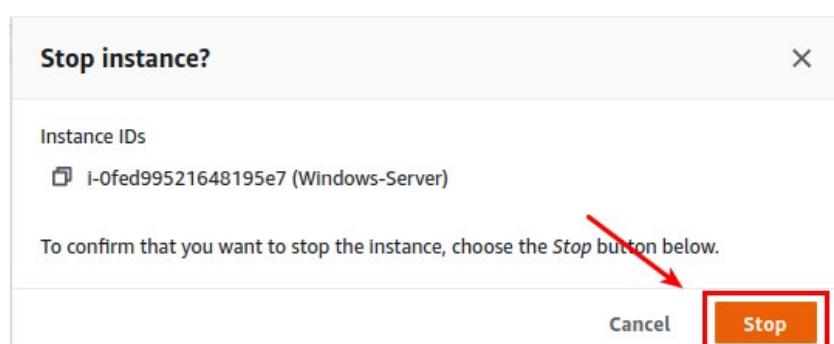
Pada bagian ini kita akan membahas mengenai mematikan dan menghapus instance pada EC2, kedua cara ini sangatlah penting karena ada suatu keadaan dimana kita hanya akan menghentikan instance dan akan menjalankannya kembali, ataupun menghapusnya secara permanen.

### 6.6.4.1. Mematikan Instance

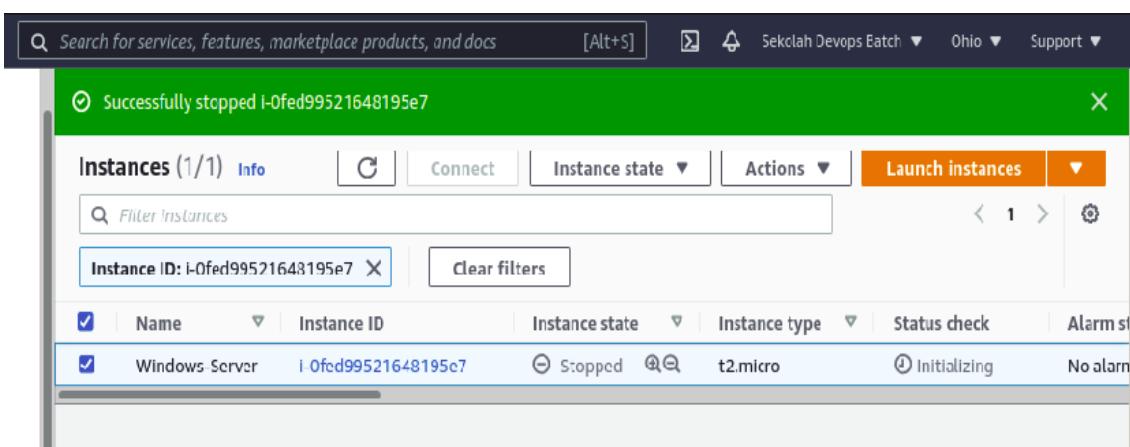
Cara mematikan instance sangat mudah sekali, kita hanya perlu memilih instance mana yang akan kita matikan, selanjutnya pilih menu **action** yang ada diatas lalu **instance state > stop**.



Selanjutnya akan muncul pop out notifikasi, pilih yes untuk melakukan stop instance.



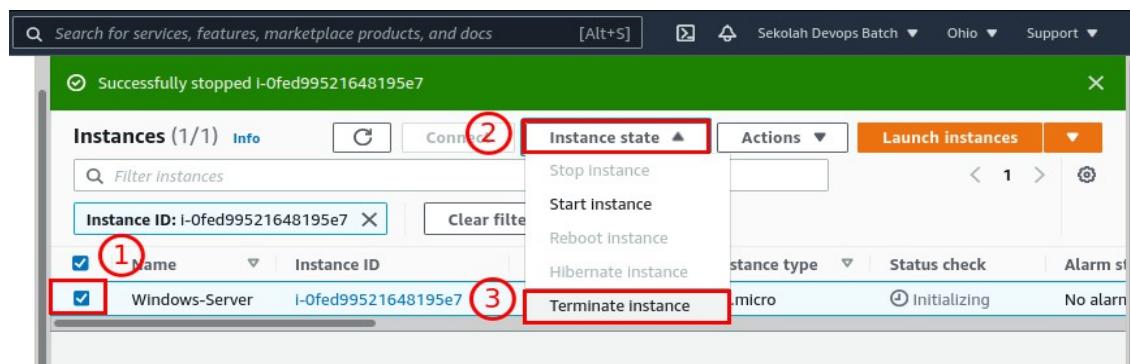
Setelah berhasil, tunggu beberapa detik dan refresh page hingga muncul seperti dibawah ini. Ini menunjukan bahwa instance sudah berhenti berjalan.



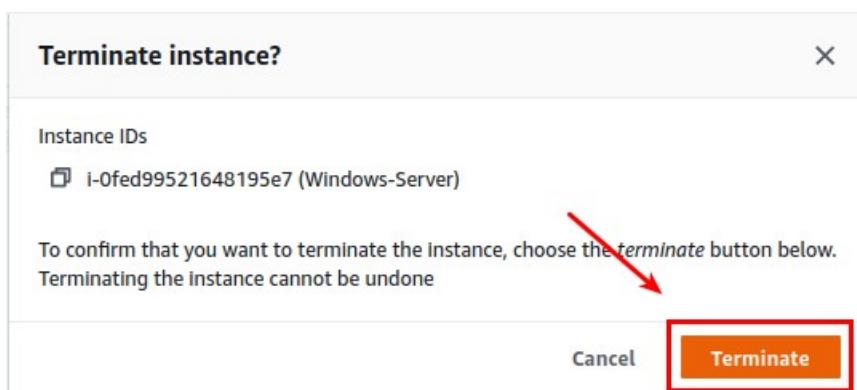
#### **6.6.4.2. Menghapus Instance**

Pada bagian ini kita akan menghapus instance atau terminate instance, jika kita sudah menghapus instance kita tidak bisa mengembalikannya kembali ke semua, maka harus berhati hati pada tahap ini.

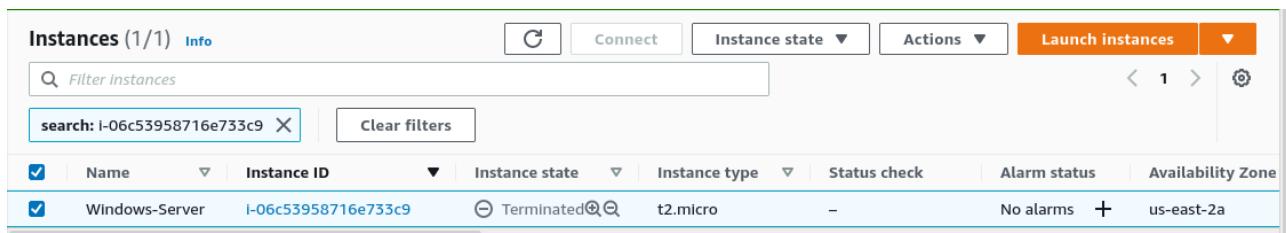
Untuk melakukan terminate, kita tinggal pilih instance tersebut selanjutnya pilih menu **action** yang ada diatas lalu **instance state > Terminate**.



Akan muncul pop out informasi untuk konfirmasi penghapusan, pilih yes untuk menghapus.



Tunggu beberapa detik hingga status pada instance tersebut berubah menjadi terminate seperti gambar dibawah ini.



The screenshot shows the AWS EC2 Instances dashboard with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Windows-Server	I-06c53958716e733c9	Terminated	t2.micro	-	No alarms	us-east-2a

Dengan begini instance sudah kita hapus atau terminate, untuk beberapa saat history instance ini akan tetap muncul di dashboard. Tapi beberapa jam kemudian history instance ini akan menghilang dengan sendirinya.

### 6.6.5. Excercise

1. Buat sebuah EC2 dengan menggunakan spesifikasi berikut
  - Menggunakan Amazon Linux
  - Menggunakan VPC yang sudah pernah dibuat
  - Diberikan akses HTTP, HTTPS, dan SSH
2. Buat sebuah EC2 dengan menggunakan spesifikasi berikut
  - Menggunakan Windows Server 2016
  - Menggunakan VPC yang sudah pernah dibuat
  - Diberikan akses HTTP dan RDC

## 6.7. Deploy Aplikasi Web di EC2

1. Pada bagian ini kita masuk kedalam tahap deployment Aplikasi di EC2, bagian ini tidak jauh berbeda dengan apa yang kita setup di BAB 3 sebelumnya yaitu Setup Webserver dan Setup Web Aplikasi.

### 6.7.1. Setup Layanan Web Server

Pada bagian ini kita akan coba untuk melakukan setup pada webserver yang akan kita gunakan, pada bagian ini hanya akan diterangkan bagian apa saja yang harus di installkan. Kalian dapat menggunakan cara yang sama seperti di bab 3 ataupun menggunakan script yang sudah dibuat sebelumnya untuk installasi.

Berikut adalah beberapa bagian yang harus di installkan pada webserver tersebut.

1. Apache2
2. php dan php-mysql
3. mysql-server

### 6.7.2. Setup Web Aplikasi

Pada bagian ini kita akan coba melakukan setup web aplikasi, masih tidak berbeda dengan tahap yang sudah kita kerjakan di bab 3 sebelumnya. Tahap pertama yang harus kita lakukan adalah membuat database dengan kriteria sebagai berikut :

1. Buat user mysql dengan nama **devopscilsy** dan password **1234567890**
2. Berikan **privileges** akses pada user tersebut
3. login pada user **devopscilsy** dan buat database **dbsosmed**

Selanjutnya setelah database siap, kita akan coba mengambil web aplikasi yang sudah kita simpan di **repository github/gitlab** yang masing-masing kalian miliki.

Sebagai contoh kita akan coba melakukan clone dari repo git cilsy.

```
sudo git clone https://github.com/sdcilsy/sosial-media.git /var/www/html
```

Clone repo tersebut langsung diarahkan pada direktori `/var/www/html` sehingga kita tidak perlu susah untuk memindahkan kembali isi konten yang ada pada repo tersebut.

Selanjutnya kita lakukan import file database ke database yang sudah kita buat seperti biasanya dengan menggunakan perintah berikut :

```
cd /var/www/html/  
sudo mysql -u devopscilsy -p dbsosmed < dump.sql
```

Setelah selesai, sekarang kita bisa ujicoba web aplikasi tersebut dengan memanggil kembali alamat server local yang kita miliki.



*Halaman utama Aplikasi Web*

Selamat kamu sudah menyelesaikan tahap deploy aplikasi ke EC2 Instance dengan mengambil aplikasi yang kamu miliki di repo yang ada di github/gitlab. Sekarang kamu bisa dengan mudah melakukan deployment aplikasi di AWS.

## 6.8. Summary

1. Kita sudah mengenal Cloud Computing, mulai dari model pengiriman dan juga jenis-jenisnya. Begitupun dengan AWS Platform Cloud yang kita gunakan pada sekolah devops ini.
2. Kita juga membahas mengenai Amazon VPC, bagaimana kita merancang infrastruktur yang akan kita gunakan untuk stack yang kita buat nanti, mulai dari mengatur subnet, NAT, hingga gateway.
3. Pada bagian Amazon EC2, kita sudah membahas mengenai type instance yang ada di AWS, penjelasan keypair hingga membuat sebuah instance baru dengan menggunakan Ubuntu Server dan Windows Server.
4. Yang terakhir, kita sudah membahas mengenai deployment aplikasi pada stack yang sudah kita buat. Dengan menggunakan aplikasi pesbus dan layanan yang diinstall secara automasi.