OCI COMPUTE

3.1 Tujuan Praktikum

1. Memahami Arsitektur *Virtual Cloud Network* (VCN)
2. Menguasai *Deployment Compute Instance*
3. Mengimplementasikan Web Server Apache
4. Menganalisis Konfigurasi Keamanan Jaringan Cloud
5. Melakukan Validasi dan Pengujian Infrastruktur

3.2 Alat dan Bahan

- Laptop dengan koneksi internet stabil
- Akun *Oracle Cloud Infrastructure* (*Free Tier* atau berbayar)
- Peramban web terkini (direkomendasikan Chrome, Edge, atau Firefox)
- Terminal atau *SSH client* (PuTTY, Terminal, *Command Prompt*)

3.3 Dasar Teori

3.3.1 *Infrastructure as a Service* (IaaS)

Infrastructure as a Service (IaaS) merupakan model layanan komputasi awan yang menyediakan sumber daya komputasi fundamental melalui platform virtualisasi. *Oracle Cloud Infrastructure* (OCI) mengimplementasikan IaaS dengan arsitektur *regional deployment* yang terdiri dari *multiple availability domains* yang terisolasi secara fisik.

Karakteristik fundamental IaaS meliputi elastisitas dinamis, dimana pengguna dapat melakukan *scaling* sumber daya secara *real-time* berdasarkan kebutuhan *workload*, serta model biaya operasional (opex) yang hanya mengenakan biaya untuk sumber daya yang benar-benar dikonsumsi. OCI menerapkan prinsip *separation of concerns* dengan memisahkan infrastruktur fisik dari lapisan virtualisasi, memungkinkan organisasi untuk fokus pada pengembangan aplikasi tanpa perlu mengelola kompleksitas infrastruktur dasar.

3.3.2 *Virtual Cloud Network* (VCN)

Virtual Cloud Network (VCN) merupakan komponen jaringan terdefinisi perangkat lunak (*software-defined networking*) di OCI yang menyediakan lingkungan jaringan terisolasi dengan kontrol granular. Arsitektur VCN mengimplementasikan konsep *address space*

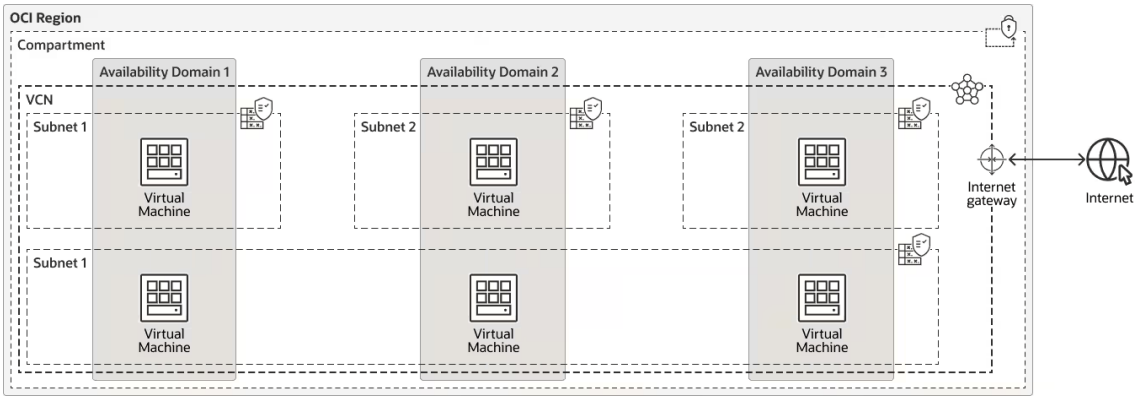
partitioning dengan blok CIDR yang dapat dikustomisasi dalam rentang RFC 1918 atau publik. Setiap VCN mendukung *hierarchical segmentation* melalui *subnetting* strategis yang dapat di-*deploy* di berbagai *availability domain* untuk mencapai *fault tolerance*.

Komponen kunci VCN meliputi *route tables* untuk mengatur *traffic flow*, *security lists* sebagai *firewall stateful*, dan *gateway services* (*Internet Gateway*, *NAT Gateway*, *Service Gateway*) yang memfasilitasi konektivitas *hybrid* dan internet.

Manfaat OCI VCN

1. **Pusat data yang aman di cloud**: Gunakan aturan keamanan yang dapat dikonfigurasi untuk mengontrol lalu lintas paket data yang masuk dan keluar *instance*. Tentukan subnet sebagai publik atau pribadi. Tempatkan VCN dalam zona keamanan untuk menerapkan kebijakan sesuai praktik terbaik.
2. **Biaya transfer data lebih rendah, bahkan terkadang gratis**: OCI tidak mengenakan biaya transfer data untuk perpindahan data dalam region yang sama, termasuk antara jaringan virtual atau *availability domain*.
3. **Atasi dan selesaikan masalah jaringan**: Tersedia alat-andal yang *powerful* untuk melihat, mendiagnosis, dan memeriksa jaringan Anda, termasuk tata letak jaringan visual, pengujian konektivitas, dan inspeksi tingkat paket data.

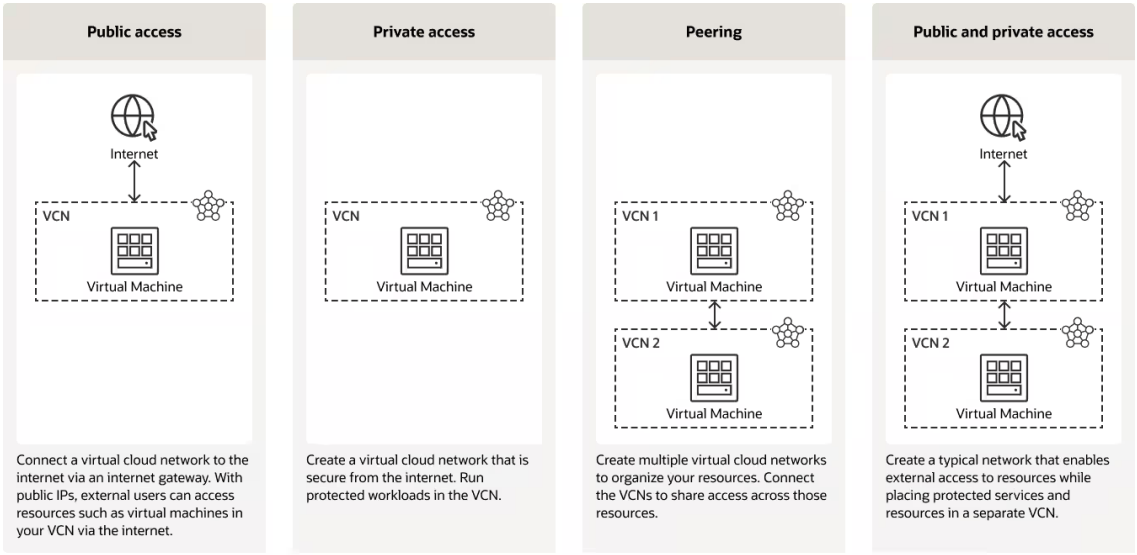
Cara Kerja OCI VCN



Virtual cloud networks (VCN) adalah jaringan pribadi virtual yang sangat mirip dengan jaringan tradisional, dilengkapi aturan *firewall* dan berbagai jenis *gateway* komunikasi yang dapat Anda pilih. VCN berada dalam satu region OCI dan mencakup satu atau beberapa blok CIDR (IPv4 dan IPv6, jika diaktifkan). Setiap subnet terdiri dari rentang alamat IP yang berurutan (untuk IPv4 dan IPv6, jika diaktifkan) yang tidak tumpang tindih dengan subnet lain dalam VCN.

Anda dapat menetapkan subnet untuk berada dalam satu domain ketersediaan tunggal atau menjangkau seluruh region (subnet regional disarankan). Semua antarmuka jaringan dalam subnet tertentu menggunakan *route tables*, aturan keamanan, dan opsi DHCP yang sama. Subnet dapat bersifat publik atau privat saat dibuat. Privat berarti antarmuka jaringan dalam subnet tidak dapat memiliki alamat IPv4 publik dan komunikasi internet dengan *endpoint* IPv6 akan diblokir. Publik berarti antarmuka jaringan dalam subnet dapat memiliki alamat IPv4 publik dan komunikasi internet dengan *endpoint* IPv6 diizinkan.

Kasus Penggunaan untuk OCI VCN



1. **Akses Publik (*Public Access*):**
    - **Konsep:** VCN dihubungkan ke internet menggunakan komponen bernama *Internet Gateway* (IGW).
    - **Fungsi:** Memungkinkan akses *inbound* dari internet ke sumber daya di dalam VCN (misalnya, ke sebuah web server pada mesin virtual) yang memiliki alamat IP publik. Ini cocok untuk *workload* yang perlu diakses oleh publik.
2. **Akses Privat (*Private Access*):**
    - **Konsep:** VCN sengaja diisolasi dan tidak memiliki rute langsung ke internet.
    - **Fungsi:** Menciptakan lingkungan yang sangat aman untuk menjalankan *workload* yang kritis atau sensitif, seperti *database*, server aplikasi *backend*, atau sistem yang mematuhi regulasi tertentu. Akses dari dan ke internet biasanya dilakukan melalui jalur yang lebih aman seperti *FastConnect* atau VPN.
3. ***Peering* VCN (VCN *Peering*):**

- **Konsep:** Menghubungkan dua VCN secara langsung dalam region yang sama (*Local Peering*) atau berbeda (*Remote Peering*) menggunakan komponen *Peering Gateway*.
- **Fungsi:** Memungkinkan komunikasi jaringan pribadi yang aman dan berlatensi rendah antar sumber daya di VCN yang berbeda. Pola ini digunakan untuk segmentasi jaringan (misalnya, memisahkan lingkungan *production* dan *development*), berbagi layanan, atau menghubungkan VCN dari departemen yang berbeda dalam satu organisasi.

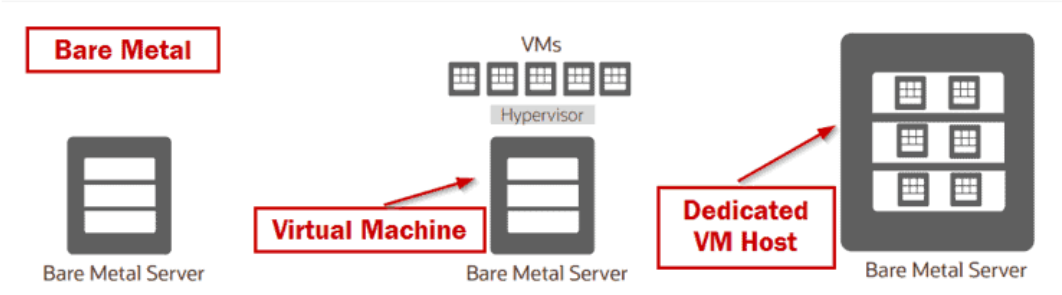
4. **Akses Publik dan Privat (*Hybrid/DMZ-like Architecture*):**

   - **Konsep:** Arsitektur yang menggabungkan konsep akses publik dan privat dengan menggunakan lebih dari satu VCN.
   - **Fungsi:** Menciptakan arsitektur berlapis keamanan (*defense in depth*). Biasanya, satu VCN (sering disebut subnet "DMZ") dikonfigurasi dengan akses publik untuk menerima lalu lintas dari internet (misalnya, *load balancer* atau web server). VCN ini kemudian di-*peer*-kan dengan VCN kedua yang bersifat privat, yang berisi sumber daya *backend* yang lebih sensitif (seperti *database* atau layer aplikasi). Ini memisahkan dan melindungi layer internal dari eksposur langsung ke internet.

3.3.3 *Compute Instance*

Compute Instance di OCI merepresentasikan *virtual machine* yang berjalan pada infrastruktur *bare metal* atau *virtualized host*. Arsitektur instance menggunakan *hardware-assisted virtualization* dengan Intel VT-x atau AMD-V untuk optimalisasi performa. Konsep *instance shapes* menentukan alokasi *resource* komputasi termasuk OCPU (Oracle CPU), memori, *storage* lokal, dan *bandwidth* jaringan. Setiap *instance* terintegrasi dengan VCN melalui *Virtual Network Interface Card (VNIC)* yang mendukung *multiple VNIC attachments* untuk arsitektur *multi-tier*. Mekanisme *instance lifecycle management* memungkinkan operasi *stop*, *start*, *reboot*, dan *terminate* dengan preservasi *boot* volume data.

Jenis-jenis *Instance*



1. *Bare Metal Server*

Bare Metal Server adalah sebuah server fisik yang sepenuhnya didedikasikan untuk satu pelanggan atau *tenant*. Karakteristik utamanya adalah beroperasi tanpa virtualisasi, yang berarti tidak ada lapisan *hypervisor* dan sistem operasi berjalan langsung di atas perangkat keras. Hal ini menghilangkan *overhead* atau beban tambahan dari virtualisasi, sehingga menghasilkan kinerja tinggi yang ideal untuk beban kerja yang membutuhkan kinerja maksimal, konsistensi, dan latensi rendah, seperti *database* berkinerja tinggi, komputasi HPC, atau aplikasi yang memerlukan akses langsung ke komponen *hardware*.

Keamanan dan isolasinya pun sangat tinggi karena tidak ada sumber daya yang dibagikan dengan pelanggan lain. Namun, kelemahan model ini adalah biasanya lebih mahal dan kurang fleksibel dibandingkan *Virtual Machine* (VM), karena proses penyediaan sumber dayanya membutuhkan waktu yang lebih lama.

2. *Virtual Machine* (VM)

Virtual Machine (VM) merupakan sebuah lingkungan komputasi virtual yang dibuat oleh perangkat lunak dan berjalan di atas sebuah server fisik. Karakteristik kuncinya adalah penggunaan *hypervisor*, yaitu sebuah lapisan perangkat lunak yang bertanggung jawab untuk membuat, menjalankan, dan mengelola beberapa VM di atas satu server fisik yang sama dengan mengabstraksikan sumber daya fisik seperti CPU, memori, dan penyimpanan lalu membagikannya ke berbagai VM.

Model ini mendorong efisiensi dan konsolidasi dengan memungkinkan beberapa sistem operasi dan aplikasi berjalan secara terisolasi pada satu mesin fisik, sehingga meningkatkan pemanfaatan sumber daya. VM juga menawarkan fleksibilitas dan kelincahan yang tinggi karena dapat dibuat, di-*clone*, dipindahkan, atau dihancurkan dengan sangat cepat, yang sangat mendukung kebutuhan *scaling* dan pengembangan. Kekurangannya terletak pada adanya sedikit *overhead* kinerja akibat lapisan virtualisasi, yang dapat berpengaruh pada aplikasi yang sangat sensitif terhadap kinerja.

3. *Dedicated VM Host*

Dedicated VM Host adalah sebuah model *hybrid* yang menggabungkan aspek dari *Bare Metal* dan VM. Dalam model ini, pelanggan mendapatkan akses eksklusif ke seluruh server fisik, atau disebut sebagai *Host* Fisik Dedikasi. Pelanggan kemudian memiliki kontrol penuh atas virtualisasi, di mana mereka dapat menginstal dan menjalankan *hypervisor* pilihannya sendiri untuk membuat dan mengelola banyak *Virtual Machine* di dalam server yang *dedicated* tersebut.

Keunggulan model ini adalah memberikan isolasi fisik dan keamanan layaknya *Bare Metal*, sekaligus fleksibilitas dari virtualisasi. Oleh karena itu, model ini sangat ideal bagi

organisasi yang memerlukan kontrol penuh atas lingkungan virtualisasi mereka tetapi tetap menginginkan isolasi pada level perangkat keras.

3.2 Keamanan di *Oracle Cloud Infrastructure*

Keamanan di OCI dibangun berdasarkan model *shared responsibility* yang membagi tanggung jawab keamanan antara Oracle dan *customer*.

3.2.1 Model *Shared Responsibility*

Tanggung Jawab Oracle:

- Keamanan fisik data *center*
- Keamanan infrastruktur *hypervisor*
- Keamanan jaringan dasar
- Platform dan layanan *foundation*

Tanggung Jawab *Customer*:

- Keamanan sistem operasi *guest*
- Konfigurasi *firewall* dan *security lists*
- Manajemen identitas dan akses (IAM)
- Enkripsi data dan manajemen kunci
- Keamanan aplikasi

3.2.2 Komponen Keamanan OCI

1. *Identity and Access Management* (IAM)

IAM adalah layanan fundamental untuk mengontrol akses ke sumber daya OCI.

Konsep Utama:

- **Kompartemen (*Compartments*):** *Logical container* untuk mengorganisir dan mengisolasi sumber daya
- **Kebijakan (*Policies*):** Pernyataan yang menentukan siapa yang dapat mengakses apa
- ***Dynamic Groups*:** Mengelompokkan instance berdasarkan karakteristik tertentu
- ***Instance Principal*:** Memungkinkan instance untuk berinteraksi dengan layanan OCI lainnya

Contoh *Policy*:

```
Allow group Developers to manage instances in compartment
Development
Allow group Admins to manage all-resources in tenancy
```

2. *Network Security*

***Security Lists* vs *Network Security Groups*:**

***Security Lists*:**

- *Firewall stateful* tradisional di level subnet
- Mengatur lalu lintas *ingress* dan *egress*
- Berlaku untuk semua VNIC dalam subnet

***Network Security Groups* (NSG):**

- *Firewall stateful* yang lebih granular
- Dapat diterapkan ke VNIC tertentu
- Mendukung *micro-segmentation*
- Lebih mudah dikelola untuk arsitektur kompleks

***Best Practice* Konfigurasi *Security Rules*:**

```
Ingress Rules:
  - Sumber: 0.0.0.0/0
    Protocol: TCP
    Port: 80,443   # HTTP/HTTPS untuk web server
    Deskripsi: "Akses web publik"

  - Sumber: 192.168.1.0/24
    Protocol: TCP
    Port: 22       # SSH hanya dari jaringan internal
    Deskripsi: "SSH internal only"

Egress Rules:
  - Tujuan: 0.0.0.0/0
    Protocol: TCP
    Port: 80,443   # Keluar untuk update dan external API
    Deskripsi: "Outbound web traffic"
```

3. *Data Encryption*

***Encryption at Rest*:**

- ***Block Volume*:** Dienkripsi secara *default* dengan kunci yang dikelola OCI
- ***Object Storage*:** Mendukung SSE-S3, SSE-KMS, dan SSE-C
- ***File Storage*:** Enkripsi *default* dengan AES-256
- **BYOK (*Bring Your Own Key*):** Menggunakan *OCI Vault* untuk manajemen kunci kustomer

***Encryption in Transit*:**

- TLS/SSL untuk komunikasi data
- VPN IPSec untuk koneksi *site-to-site*
- *FastConnect* untuk koneksi *dedicated private*

4. *Web Application Firewall* (WAF)

- Melindungi aplikasi web dari serangan umum (OWASP Top 10)
- *Protection* terhadap *DDoS attacks*
- *Bot management* dan *rate limiting*
- *Custom rules* berdasarkan IP, *geo-location*, atau *HTTP headers*

5. *Security Zones*

- Membuat kompartemen dengan kebijakan keamanan yang tidak dapat diubah
- Menerapkan *security controls* secara otomatis
- Mencegah konfigurasi yang tidak aman
- Cocok untuk *workload* dengan *compliance requirements* tinggi

3.2.3 *Best Practices* Keamanan OCI

1. ***Prinsip Least Privilege*:**
    - Berikan hanya akses yang diperlukan
    - Gunakan *groups* daripada *user* individual
    - *Review policies* secara berkala
2. ***Network Segmentation*:**
    - Pisahkan *tier* aplikasi dengan subnet berbeda
    - Gunakan NSG untuk *micro-segmentation*
    - Implementasi DMZ untuk layanan publik
3. ***Hardening Instance*:**

- *Update* sistem operasi secara berkala
- Hanya buka *port* yang diperlukan
- Gunakan *key pairs* untuk SSH *authentication*
- *Disable root login* via SSH

4. ***Monitoring* dan Audit:**
    - Aktifkan *OCI Audit service*
    - Gunakan *Cloud Guard* untuk *security monitoring*
    - *Setup notifications* untuk aktivitas mencurigakan
5. ***Backup* dan *Disaster Recovery*:**
    - *Regular backup* menggunakan *OCI Block Volume backups*
    - *Cross-region replication* untuk *critical* data
    - *Test recovery procedures* secara berkala

3.2.4 *Tools* Keamanan OCI

1. ***Cloud Guard*:** *Service security monitoring* yang terkelola
2. ***Security Advisor*:** Rekomendasi keamanan otomatis
3. ***Vault*:** Layanan manajemen kunci dan rahasia
4. ***Certificates*:** SSL/TLS *certificate management*
5. ***Threat Intelligence*:** Deteksi ancaman berbasis AI/ML

LANGKAH PRAKTIKUM

Ditulis oleh: Risnanda Candra Abdurrozaq

Pendahuluan

Oracle Cloud Infrastructure (OCI) menyediakan layanan komputasi elastis yang memungkinkan pengguna untuk menyebarkan dan mengelola server virtual. Panduan ini akan memandu Anda dalam membuat Virtual Cloud Network (VCN) dan compute instance, kemudian mengonfigurasi web server Apache di lingkungan OCI.

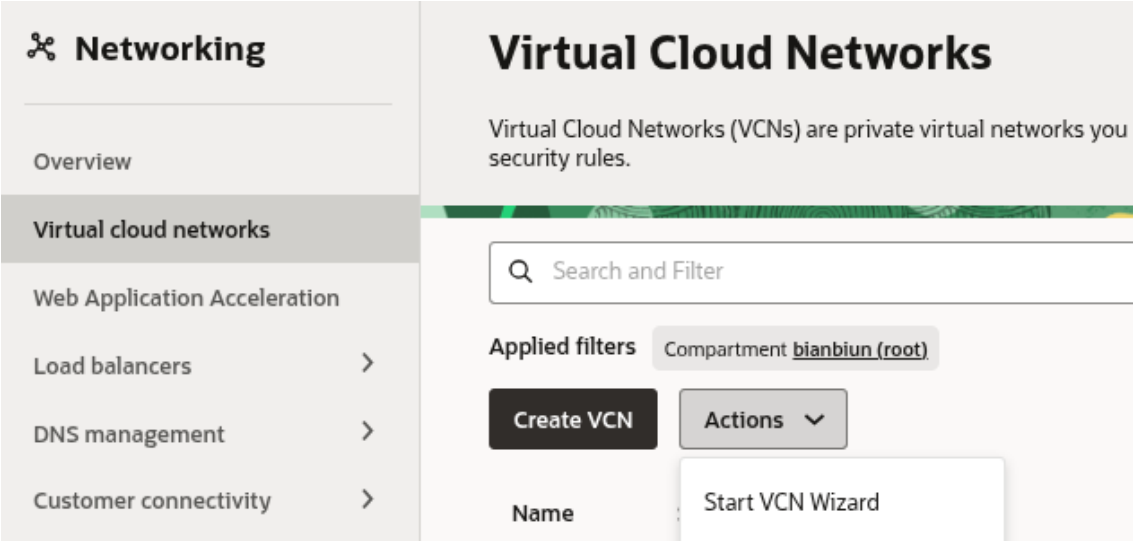
Prasyarat:

- Akun OCI (Free Tier atau berbayar)
- Akses ke Oracle Cloud Console
- Pengetahuan dasar tentang jaringan dan Linux

1. Membuat Virtual Cloud Network (VCN)

1.1 Mengakses Dashboard VCN

Navigasi ke layanan VCN di konsol OCI untuk memulai proses pembuatan jaringan virtual.

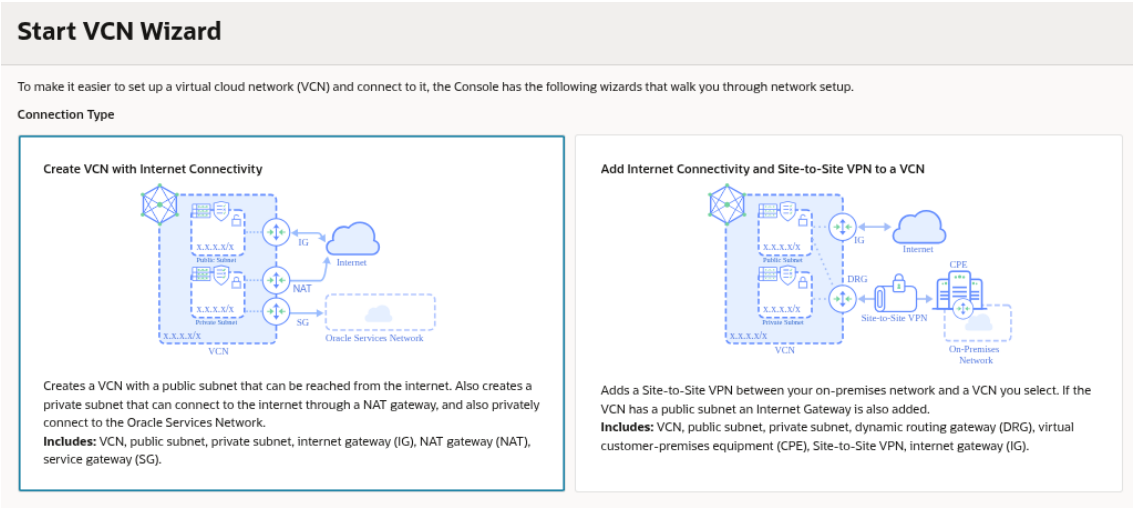


Langkah Akses:

1. Login ke Oracle Cloud Console
2. Buka menu navigation dan pilih "Networking"
3. Klik "Virtual cloud networks"
4. Pilih "Start VCN Wizard"

1.2 Memulai VCN Wizard

Pilih opsi "Create VCN with Internet Connectivity" untuk membuat VCN dengan konfigurasi jaringan yang sudah teroptimasi untuk konektivitas internet.



Opsi ini secara otomatis akan mengonfigurasi:

- Internet Gateway (IGW) untuk konektivitas internet outbound
- Route table dengan aturan routing ke internet
- Network Security Group (NSG) dengan aturan keamanan dasar
- Public dan private subnet dalam availability domain yang berbeda

Keuntungan: Menghemat waktu konfigurasi manual dan mengurangi risiko kesalahan konfigurasi jaringan.

1.3 Konfigurasi Informasi Dasar VCN

Isi informasi dasar VCN yang meliputi:

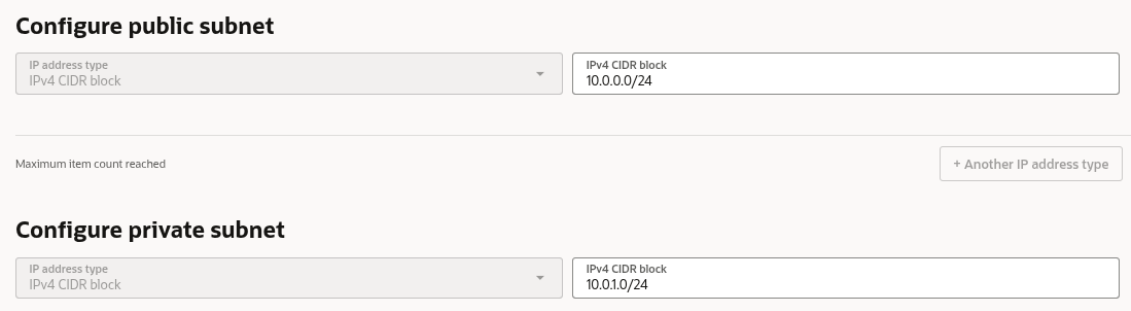
- **Nama VCN**: Identifikasi unik untuk VCN Anda
- **CIDR Block**: Rentang alamat IP untuk VCN (contoh: 10.0.0.0/16)

Catatan:

- **CIDR (Classless Inter-Domain Routing)** block menentukan rentang alamat IP privat yang akan digunakan dalam VCN
- Pilih blok CIDR yang tidak tumpang tindih dengan jaringan on-premise atau cloud lainnya
- Untuk skala kecil, /16 (65,536 alamat IP) biasanya cukup
- Hindari menggunakan blok 172.17.0.0/16 yang biasa digunakan Docker

1.4 Konfigurasi Public dan Private Subnet

Konfigurasikan subnet publik dan privat sesuai kebutuhan arsitektur jaringan Anda.



Public Subnet:

- Memungkinkan instance untuk memiliki alamat IP publik yang dapat diakses dari internet
- Cocok untuk load balancer, NAT gateway, bastion host
- Route table-nya termasuk route ke internet gateway

Private Subnet:

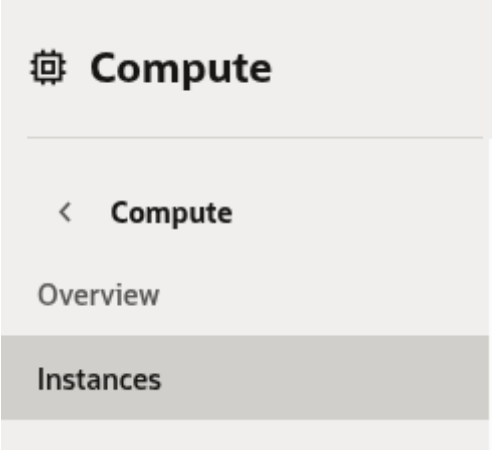
- Hanya dapat diakses dari within VCN atau melalui gateway NAT
- Ideal untuk database servers, application servers, backend services
- Lebih aman karena tidak terpapar langsung ke internet

> 💡 Gunakan arsitektur multi-tier dengan memisahkan layer application dan database di subnet yang berbeda.

2. Membuat Compute Instance

2.1 Mengakses Layanan Compute

Navigasi ke layanan Compute di konsol OCI untuk membuat instance virtual machine.

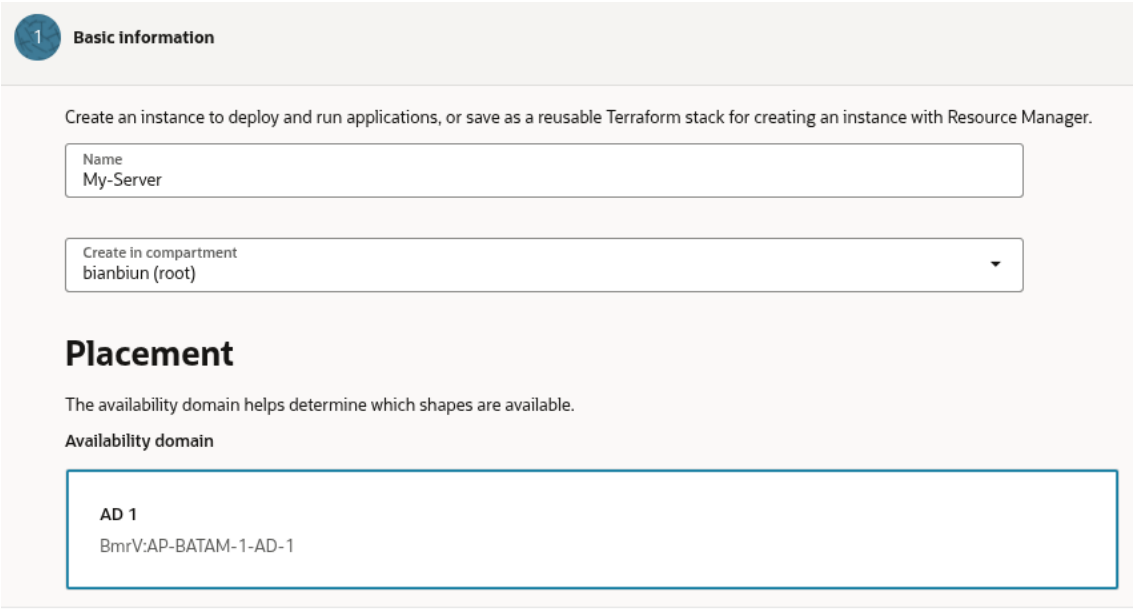


Langkah Akses:

1. Dari menu navigation, pilih "Compute"
2. Klik "Instances"
3. Pilih compartment yang diinginkan
4. Klik "Create Instance"

2.2 Konfigurasi Informasi Dasar Instance

Isi informasi dasar instance kompute termasuk nama instance dan availability domain.



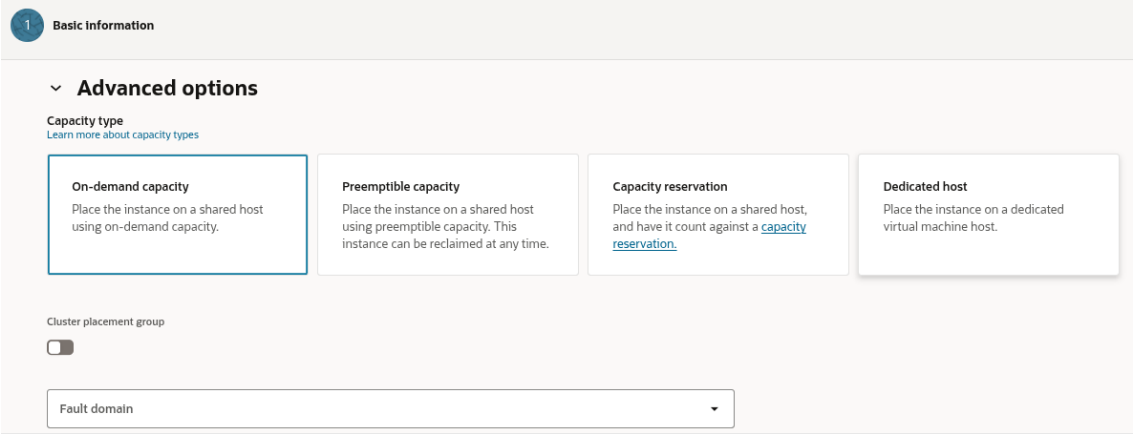
Keterangan:

- **Availability Domain**: Pada akun free tier, biasanya hanya tersedia satu availability domain
- Availability domain merupakan pusat data yang terisolasi dalam region OCI

- Setiap region biasanya memiliki 3 availability domain
- Pilihan availability domain penting untuk high availability

2.3 Konfigurasi Kapasitas On-Demand

Pada bagian advanced options, pilih opsi on-demand capacity untuk fleksibilitas alokasi resource.



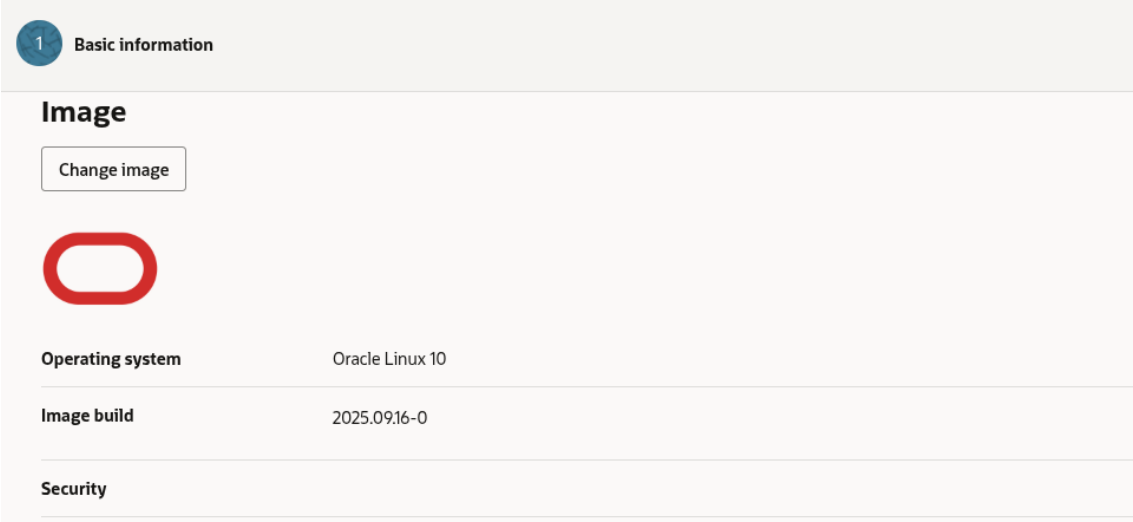
Keuntungan:

- Kapasitas on-demand memastikan ketersediaan resource tanpa perlu reservasi sebelumnya
- Fleksibel untuk workload yang berubah-ubah
- Cocok untuk development dan testing environment

Alternatif: Untuk production workload yang stabil, pertimbangkan reserved capacity untuk penghematan biaya.

2.4 Pemilihan Image Operating System

Pilih image Oracle Linux 10 yang berbasis RHEL (Red Hat Enterprise Linux).

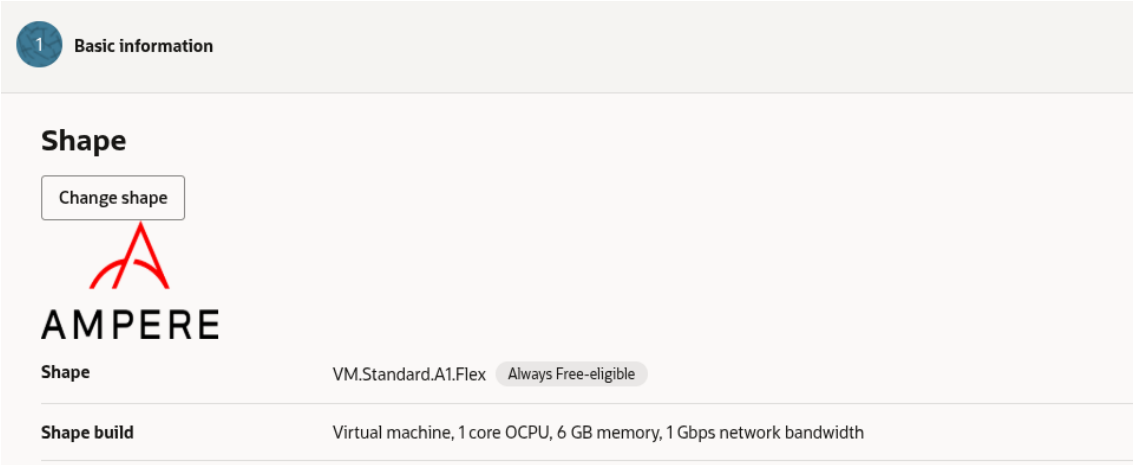


Alasan Pemilihan:

- Oracle Linux kompatibel dengan RHEL dan dioptimalkan untuk lingkungan OCI
- Menyediakan performa dan keamanan yang baik
- Gratis untuk digunakan di OCI (tidak ada biaya lisensi)
- Dukungan jangka panjang dan update security teratur
- Kompatibel dengan sebagian besar aplikasi enterprise

2.5 Konfigurasi Shape Instance

Pilih shape default yang eligible untuk free tier.

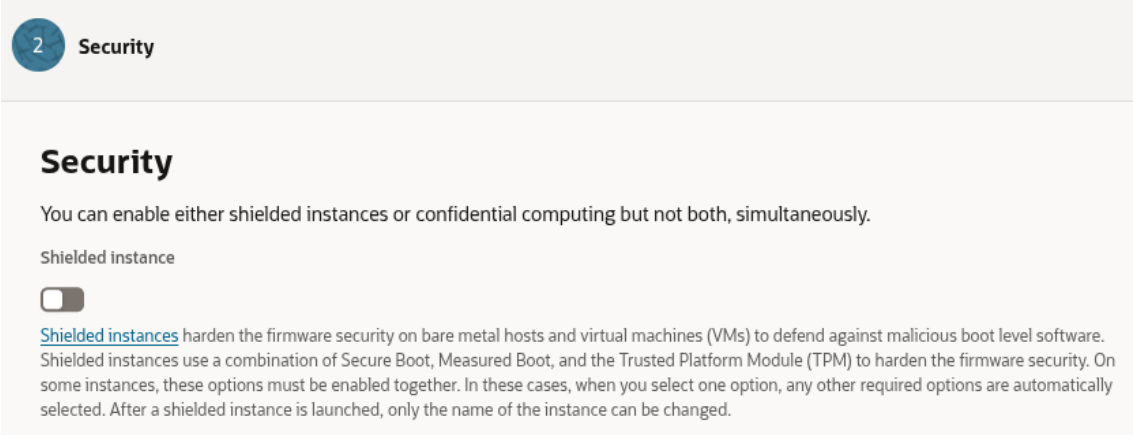


Informasi:

- Shape menentukan konfigurasi CPU dan memori instance
- Free tier menyediakan shape VM.Standard.A1.Flex (1 OCPU, 6GB memory)
- Untuk production, pertimbangkan shape yang sesuai dengan workload
- Perhatikan spesifikasi seperti OCPU (Oracle CPU), memory, network bandwidth

2.6 Konfigurasi Keamanan Instance

Pertahankan pengaturan keamanan default dengan menonaktifkan shielded instance.

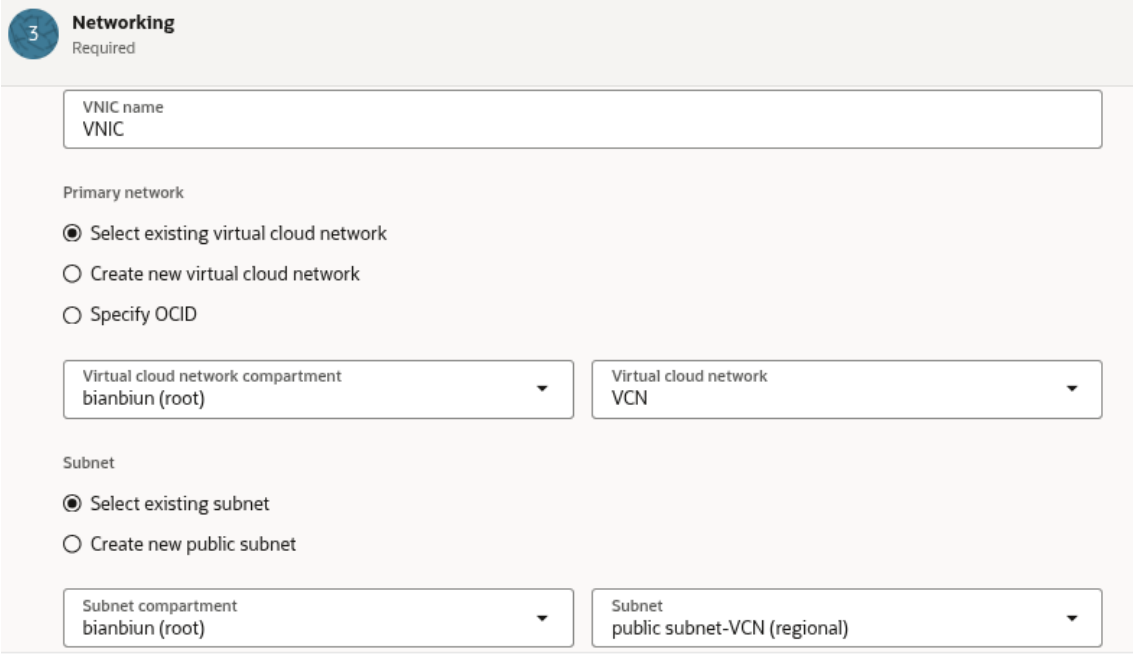


Pertimbangan:

- **Shielded instance** menawarkan keamanan tambahan (UEFI secure boot, virtual TPM)
- Untuk lingkungan production, enabledkan shielded instance
- Untuk development/testing, bisa dinonaktifkan untuk menghemat resource

2.7 Konfigurasi Jaringan Instance

Pilih VCN yang telah dibuat sebelumnya dan pilih public subnet untuk instance.

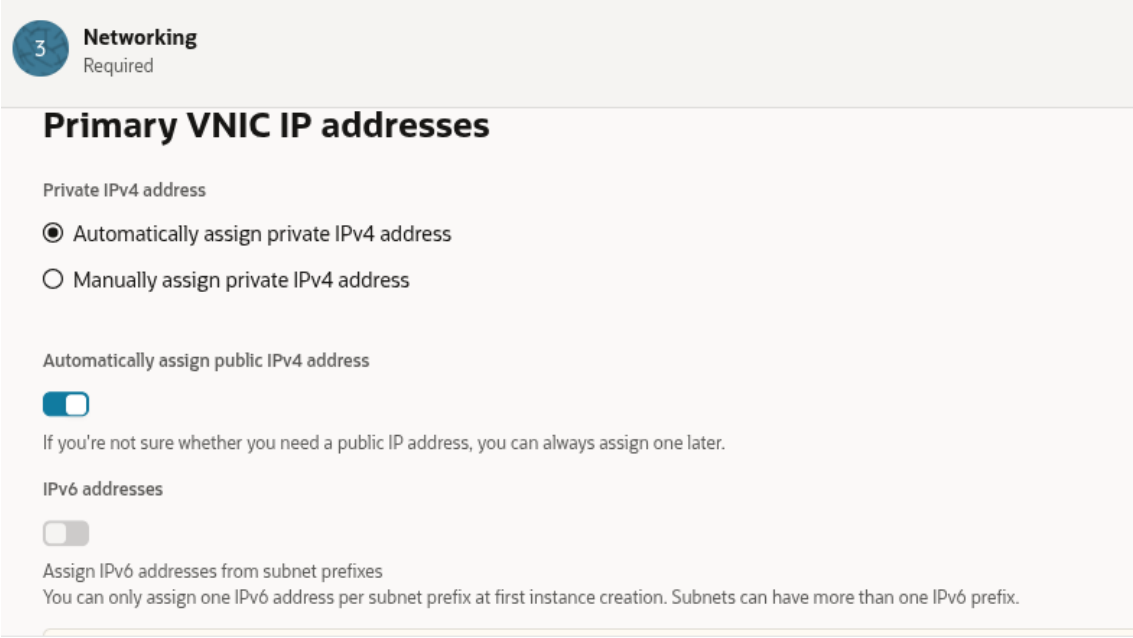


Strategi Jaringan:

- Pemilihan public subnet memungkinkan instance untuk memiliki alamat IP publik
- Pilih subnet yang sesuai dengan arsitektur aplikasi
- Pastikan VCN memiliki internet gateway untuk konektivitas outbound

2.8 Konfigurasi Alamat IP

Pastikan untuk memilih opsi "Automatically assign public IPv4 address".

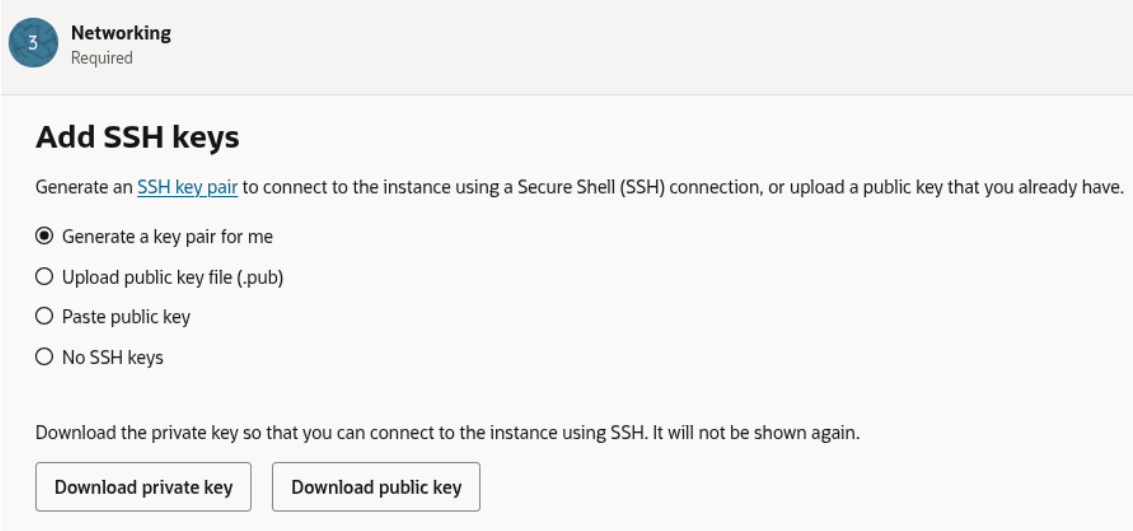


Penting:

- Alamat IP publik diperlukan untuk mengakses instance dari internet
- Alamat IP ini bersifat ephemeral (dapat berubah saat instance di-stop/start)
- Untuk alamat IP permanen, gunakan reserved public IP

2.9 Konfigurasi SSH Keys

Unduh private dan public key untuk autentikasi SSH.

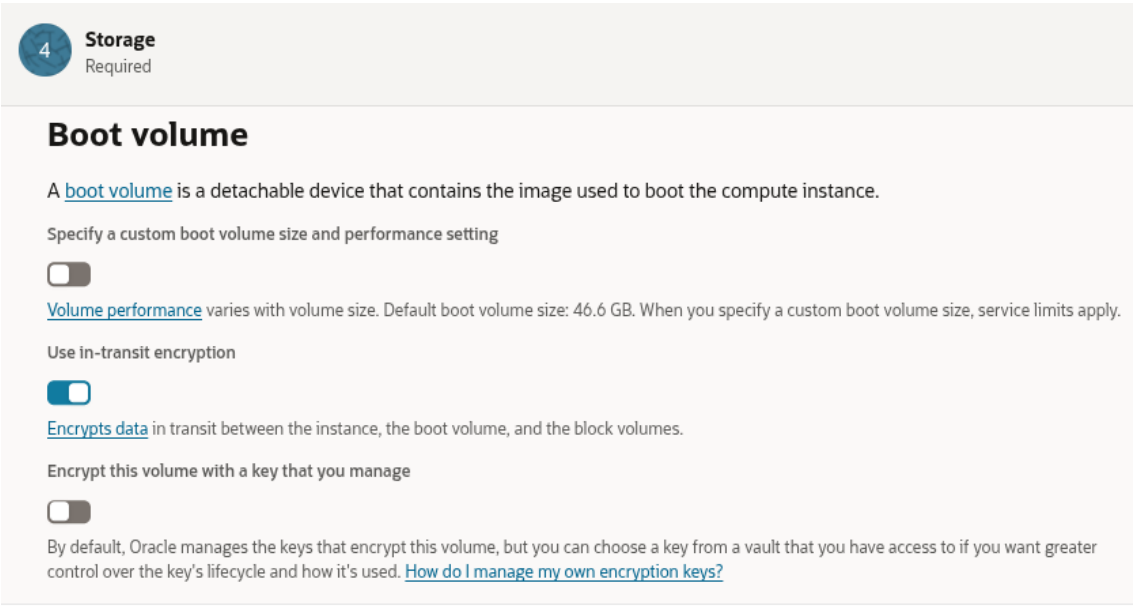


Keamanan Akses:

- Key pair SSH digunakan untuk autentikasi yang aman ke instance
- Public key akan diinstall di instance, private key disimpan di local
- Gunakan key strength minimal 2048-bit RSA
- Simpan private key di lokasi yang aman

2.10 Konfigurasi Boot Volume

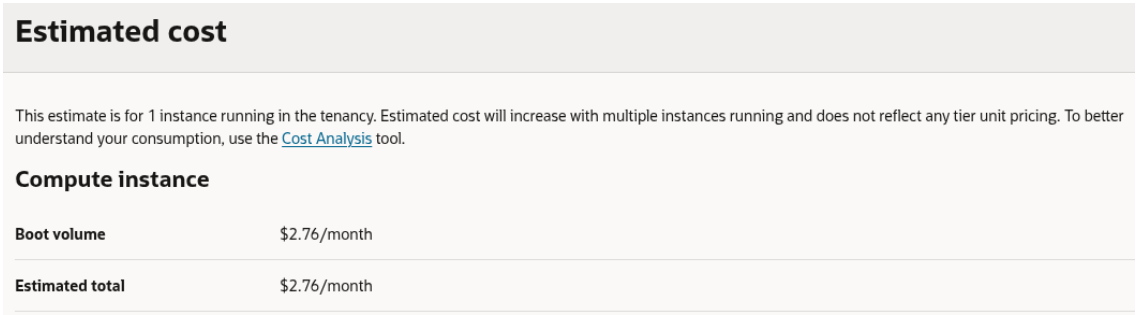
Pertahankan pengaturan default untuk boot volume.



Kustomisasi: Untuk workload khusus, bisa menyesuaikan size dan performance (VPU)

2.11 Review Estimasi Biaya

Sebelum membuat instance, periksa estimated cost untuk memastikan biaya sesuai dengan anggaran.



Best Practice:

- Selalu review estimasi biaya sebelum membuat resource cloud
- Manfaatkan cost estimator tools di OCI console
- Set up budget dan alert untuk monitoring biaya
- Free tier biasanya memiliki limit tertentu (contoh: 3000 OCPU hours per month)

2.12 Membuat Instance

Setelah semua konfigurasi selesai, buat instance dan pastikan status instance berubah menjadi "Running".

Provisining Process:

- Instance akan melalui status "Provisioning" → "Starting" → "Running"
- Waktu provisioning biasanya 1-5 menit
- Catat public IP address untuk koneksi SSH

3. Konfigurasi dan Instalasi Web Server

3.1 Koneksi SSH ke Instance

Buka terminal di sistem host dan gunakan private key yang telah diunduh untuk terhubung ke instance.

```
# Mengubah permission file private key
chmod 400 ssh-key.key
```

```
# Koneksi SSH ke instance
ssh -i ssh-key.key opc@168.110.208.247
```

Penjelasan:

- Perintah `chmod 400` mengatur permission file private key menjadi read-only untuk owner
- User default untuk Oracle Linux adalah `opc`
- Ganti IP address dengan public IP instance Anda
- Untuk Windows, gunakan Putty atau Windows SSH client

Troubleshooting Koneksi:

- Pastikan private key permission benar (600 atau 400)
- Periksa security list rules mengizinkan SSH (port 22)
- Verifikasi instance status "Running"

3.2 Instalasi Apache HTTP Server

Update repository dan instal paket Apache HTTP server.

```
sudo yum check-update && sudo yum install httpd -y
```

Fungsi:

- `yum check-update` - memperbarui repository metadata
- `yum install httpd -y` - menginstal Apache web server
- Apache HTTP Server adalah web server open-source yang powerful dan banyak digunakan

Alternatif Package Manager: Untuk Oracle Linux 8+, bisa menggunakan `dnf` sebagai pengganti `yum`

3.3 Menjalankan dan Mengonfigurasi Apache

Start service Apache dan konfigurasikan untuk berjalan otomatis saat boot.

```
sudo apachectl start && sudo systemctl enable httpd
```

Penjelasan:

- `apachectl start` - menjalankan Apache service
- `systemctl enable httpd` - mengonfigurasi service untuk berjalan otomatis saat sistem reboot
- `systemctl` adalah systemd service manager di Linux modern

Verifikasi Status:

```
sudo systemctl status httpd
```

3.4 Verifikasi Konfigurasi Apache

Jalankan tes konfigurasi untuk memastikan tidak ada error dalam konfigurasi Apache.

```
sudo apachectl configtest
```

Output yang Diharapkan:

- "Syntax OK" menandakan konfigurasi Apache valid
- Jika ada error, periksa file konfigurasi di `/etc/httpd/conf/httpd.conf`

3.5 Konfigurasi Firewall

Buka port HTTP di firewall untuk mengizinkan akses web.

```
sudo firewall-cmd --permanent --zone=public --add-
service=http
sudo firewall-cmd --reload
```

Fungsi:

- `firewall-cmd` adalah utilitas untuk mengelola firewalld
- `--permanent` - menyimpan aturan secara permanen
- `--add-service=http` - menambahkan rule untuk service HTTP (port 80)
- `--reload` - menerapkan perubahan tanpa restart service

Verifikasi Rules:

```
sudo firewall-cmd --list-all
```

3.6 Membuat Halaman Web Default

Buat file index.html sederhana untuk testing web server.

```
sudo bash -c 'echo "This is my Web-Server running on Oracle
Cloud Infrastructure" >> /var/www/html/index.html'
```

Struktur Directory:

- `/var/www/html/` - document root default Apache
- File `index.html` adalah default file yang dilayankan

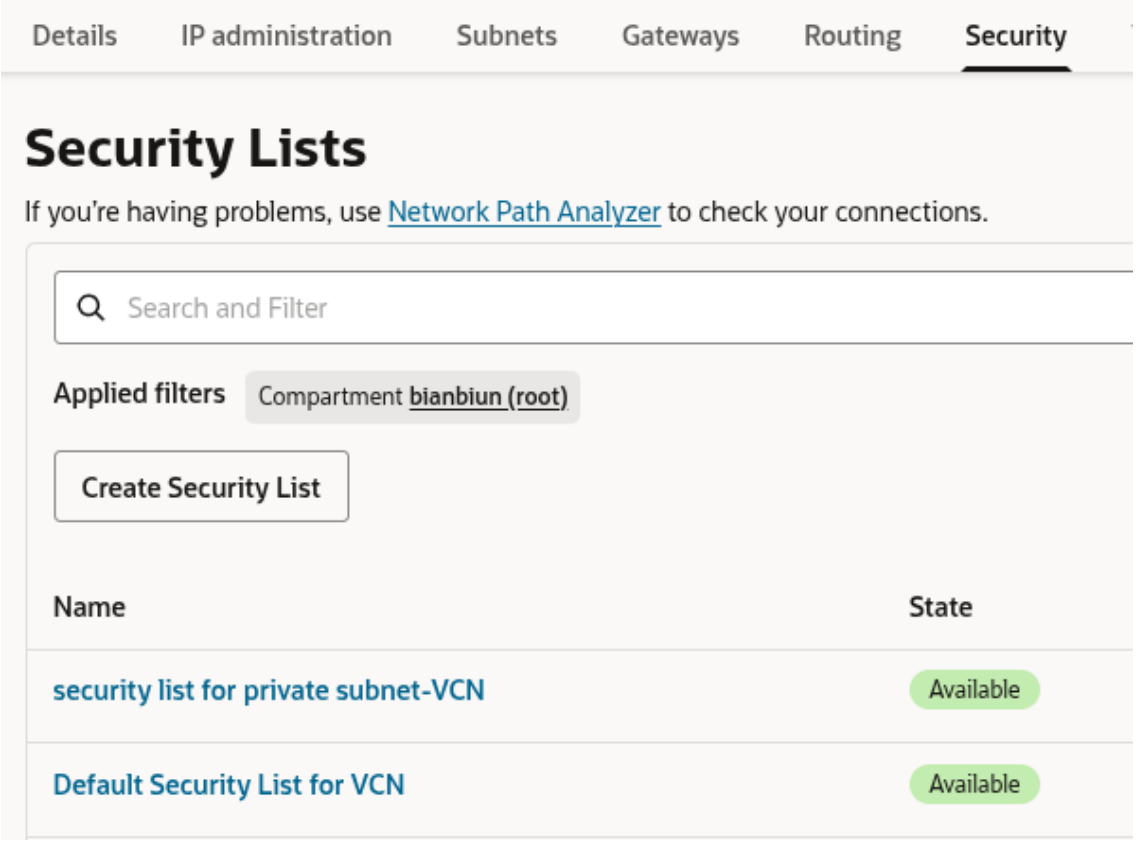
Opsional: Buat halaman HTML yang lebih kompleks:

```
sudo cat > /var/www/html/index.html << EOF
<html>
<head><title>OCI Web Server</title></head>
<body>
<h1>Welcome to OCI</h1>
<p>This web server is running on Oracle Cloud
Infrastructure</p>
<p>Instance IP: $(hostname -I)</p>
</body>
</html>
EOF
```

4. Konfigurasi Keamanan Jaringan

4.1 Mengakses Security List VCN

Buka VCN yang telah dibuat dan navigasi ke tab security. Pilih "Default Security List for VCN".

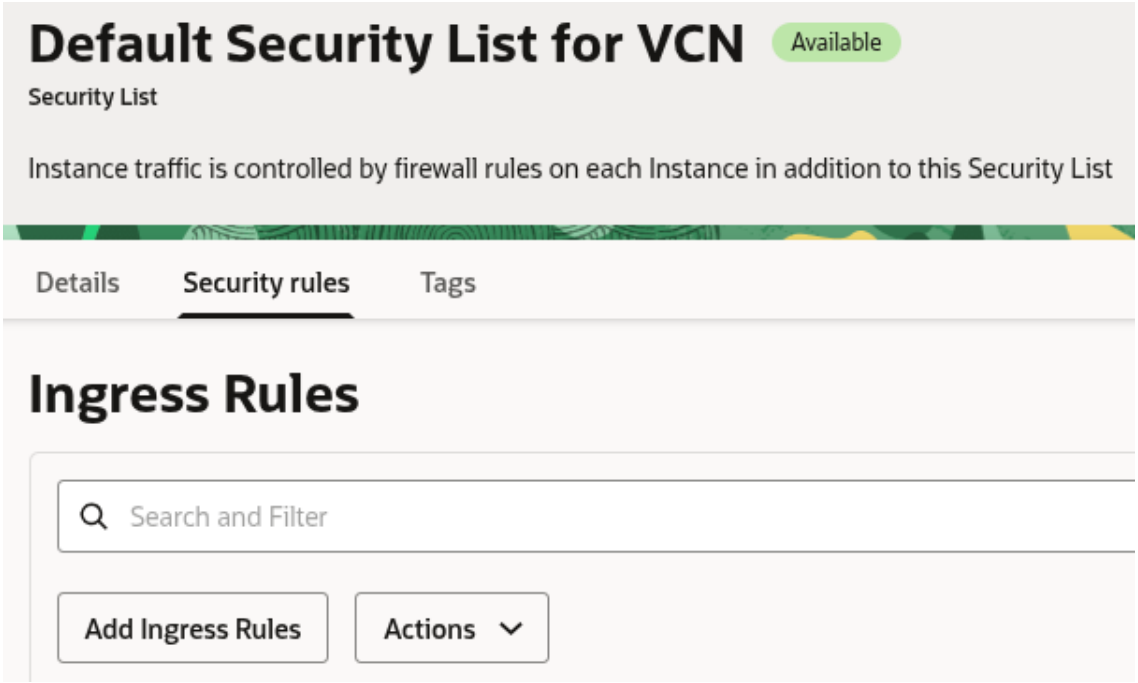


Konsep:

- Security List di OCI berfungsi sebagai firewall virtual
- Mengontrol traffic inbound dan outbound pada level subnet
- Setiap VCN memiliki default security list
- Bisa membuat custom security list untuk kontrol lebih granular

4.2 Menambahkan Ingress Rules

Tambahkan aturan inbound untuk mengizinkan traffic HTTP.



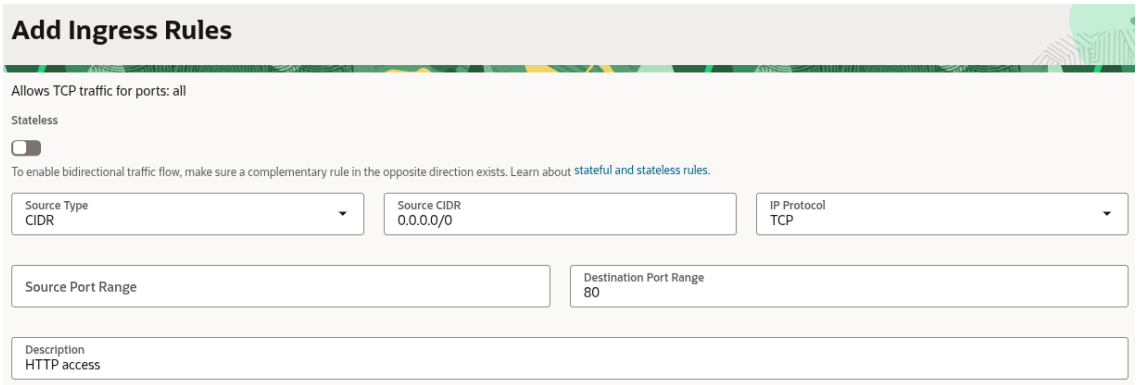
Default Rules:

- SSH (port 22) dari mana saja
- ICMP untuk ping
- Aturan stateful (return traffic diizinkan secara otomatis)

4.3 Konfigurasi Ingress Rule

Konfigurasikan aturan inbound dengan parameter berikut:

- **Source Type**: CIDR
- **Source CIDR**: 0.0.0.0/0
- **IP Protocol**: TCP
- **Source Port Range**: All
- **Destination Port Range**: 80



Penjelasan:

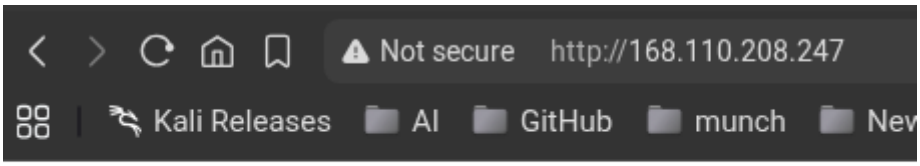
- **CIDR 0.0.0.0/0** mengizinkan akses dari semua alamat IP di internet
- **Port 80** adalah port default untuk HTTP
- Untuk lingkungan production, disarankan untuk membatasi source CIDR
- Pertimbangkan untuk menambahkan rule HTTPS (port 443) juga

Best Practice Security:

- Gunakan Network Security Groups (NSG) untuk kontrol yang lebih granular
- Implementasikan principle of least privilege
- Regular audit security rules

5. Verifikasi Web Server

Akses web server melalui browser dengan mengunjungi `http://Public-IPAddress` (ganti dengan alamat IP instance Anda).



Testing Steps:

1. Buka web browser
2. Ketik `http://[PUBLIC_IP]` (contoh: `http://168.110.208.247`)
3. Seharusnya menampilkan halaman "This is my Web-Server running on Oracle Cloud Infrastructure"

Verifikasi Tambahan:

```
# Test dari command line
curl http://localhost
```

Kesimpulan: Web server Apache sekarang berjalan dengan sukses di Oracle Cloud Infrastructure dan dapat diakses dari internet.

Troubleshooting Tips

Masalah Umum dan Solusi:

1. **Web server tidak dapat diakses:**
   - Pastikan instance dalam status "Running"
   - Verifikasi security rules telah dikonfigurasi dengan benar (port 80)
   - Cek apakah Apache service berjalan: `sudo systemctl status httpd`
   - Pastikan firewall di instance mengizinkan traffic port 80
2. **SSH connection failed:**
   - Periksa security list rules untuk port 22
   - Verifikasi private key dan permission
   - Pastikan menggunakan user `opc`
   - Cek route table untuk public subnet
3. **Apache tidak jalan:**
   - Start service: `sudo systemctl start httpd`
   - Cek error logs: `sudo tail -f /var/log/httpd/error_log`
   - Verifikasi konfigurasi: `sudo apachectl configtest`

Monitoring dan Maintenance:

```
# Monitor Apache access
sudo tail -f /var/log/httpd/access_log

# Check resource usage
top
htop

# Verify network connectivity
netstat -tulpn | grep :80
```

Konfigurasi Lanjutan:

Untuk konfigurasi lanjutan, pertimbangkan:

- Mengonfigurasi domain name dan SSL certificate dengan Let's Encrypt
- Mengatur load balancer untuk high availability
- Mengimplementasikan OCI Monitoring dan alerting
- Setup automatic backup menggunakan OCI Block Volume backup
- Konfigurasi instance pool dan auto scaling

Optimasi Performance:

1. **Apache Configuration:**
    - Edit `/etc/httpd/conf/httpd.conf` untuk optimasi
    - Adjust `KeepAlive`, `MaxKeepAliveRequests`, `KeepAliveTimeout`
    - Consider using `mod_cache` untuk caching
2. **System Optimization:**
    - Enable `httpd` service to start on boot
    - Configure swap space if needed
    - Regular system updates: `sudo yum update`

Dengan mengikuti panduan ini, Anda seharusnya dapat berhasil membuat dan mengonfigurasi web server Apache yang berjalan di Oracle Cloud Infrastructure.