

Homework Sesi 20 Bootcamp DevOps Engineer Digital Skola

1. Install Jenkins
2. Buat 1 job untuk pengecekan disk pada server

Langkah-langkah instalasi Jenkins

Langkah 1: Instalasi Java

Jenkins membutuhkan Java Runtime Environment (JRE) untuk berjalan. Versi Java yang direkomendasikan untuk Jenkins adalah Java 17 atau yang lebih baru. Kita akan menginstal OpenJDK 17.

1. **Perbarui daftar paket:** Buka terminal dan jalankan perintah berikut untuk memperbarui daftar paket yang tersedia:

sudo apt update

```
ubuntu_admin@jenkins:~$ sudo apt update
```

Perintah ini akan mengambil informasi terbaru tentang paket dan versinya dari repositori yang dikonfigurasi di sistem. Ini penting untuk memastikan instalasi versi terbaru dari paket yang dibutuhkan.

2. **Instal OpenJDK 17:** Jalankan perintah berikut untuk menginstal OpenJDK 17:

sudo apt-get install fontconfig openjdk-17-jre

```
ubuntu_admin@jenkins:~$ sudo apt-get install fontconfig openjdk-17-jre
```

Perintah ini akan mengunduh dan menginstal paket openjdk-17-jre beserta dependensinya. JRE (Java Runtime Environment) adalah lingkungan yang dibutuhkan untuk menjalankan aplikasi Java seperti Jenkins.

3. **Verifikasi instalasi Java:** Setelah instalasi selesai, dapat diverifikasi apakah Java telah terinstal dengan benar menggunakan perintah berikut:

java -version

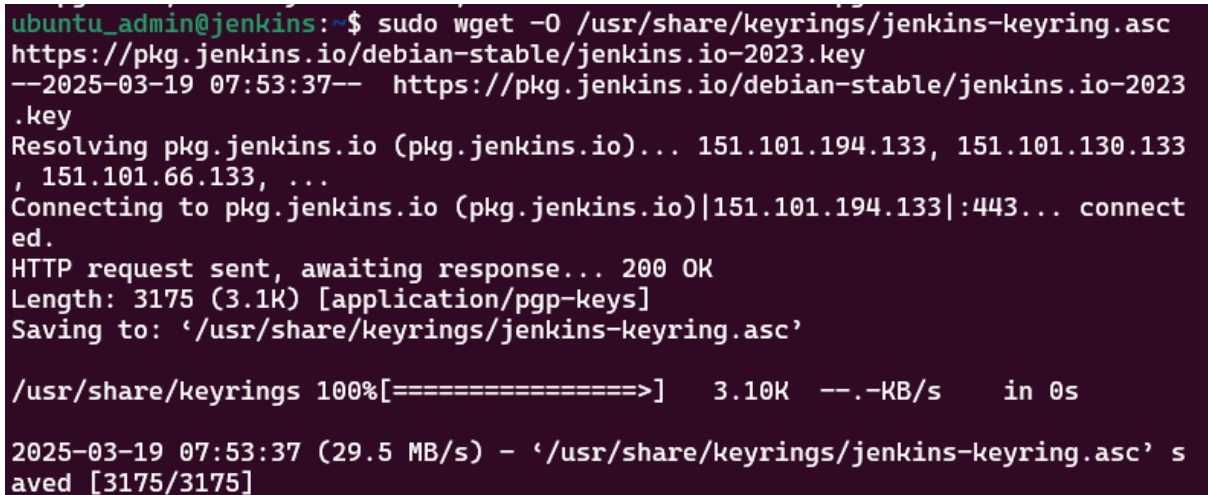
```
ubuntu_admin@jenkins:~$ java -version
openjdk version "17.0.14" 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-120.04)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-120.04, mixed mode, sharing)
```

Perintah ini akan menampilkan informasi versi Java yang terinstal di sistem. Pastikan outputnya menunjukkan bahwa OpenJDK 17 telah terinstal.

Langkah 2: Tambahkan Kunci Repositori Jenkins

Untuk menginstal Jenkins dari repositori resminya, perlu ditambahkan kunci GPG (GNU Privacy Guard) repositori tersebut ke sistem. Ini memastikan bahwa paket yang diunduh berasal dari sumber yang terpercaya. Jalankan perintah berikut untuk mengunduh dan menambahkan kunci repositori Jenkins:

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```



```
ubuntu_admin@jenkins:~$ sudo wget -O /usr/share/keyrings/jenkins-keyring.asc
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2025-03-19 07:53:37-- https://pkg.jenkins.io/debian-stable/jenkins.io-2023
.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.194.133, 151.101.130.133
, 151.101.66.133, ...
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.194.133|:443... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings 100%[=====] 3.10K --.-KB/s in 0s

2025-03-19 07:53:37 (29.5 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' s
aved [3175/3175]
```

Penjelasan:

- **wget:** Ini adalah utilitas baris perintah yang digunakan untuk mengunduh file dari internet.
- **-O /usr/share/keyrings/jenkins-keyring.asc:** Opsi -O (huruf besar O) menentukan nama dan lokasi file untuk menyimpan konten yang diunduh. Dalam hal ini, file akan disimpan sebagai jenkins-keyring.asc di direktori /usr/share/keyrings/. Direktori ini adalah lokasi standar untuk menyimpan kunci APT (Advanced Package Tool) yang digunakan untuk memverifikasi integritas paket.
- **https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key:** Ini adalah URL dari kunci publik Jenkins. Kunci ini digunakan oleh sistem manajemen paket APT untuk memverifikasi bahwa paket Jenkins yang akan diunduh berasal dari sumber yang terpercaya dan tidak dimodifikasi selama proses pengunduhan. Ini adalah langkah keamanan penting untuk memastikan integritas perangkat lunak.

Langkah 3: Tambahkan Repositori Jenkins ke Sumber APT

Setelah menambahkan kunci, perlu diberitahu sistem APT di mana menemukan paket Jenkins. Akan ditambahkan alamat repositori Jenkins ke daftar sumber paket sistem. Jalankan perintah berikut untuk menambahkan repositori Jenkins ke daftar sumber APT:

```
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]"
https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list
```

```
ubuntu_admin@jenkins:~$ echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list
deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" https://pkg.jenkins.io/debian-stable binary/
```

Penjelasan:

- **echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" https://pkg.jenkins.io/debian-stable binary/**: Perintah echo akan mencetak teks yang berada di dalam tanda kutip ke output standar. Teks ini adalah baris yang akan ditambahkan ke file sumber paket.
 - **deb**: Menunjukkan bahwa ini adalah repositori paket Debian (yang kompatibel dengan Ubuntu).
 - **[signed-by=/usr/share/keyrings/jenkins-keyring.asc]**: Opsi ini memberi tahu sistem APT bahwa paket dari repositori ini harus ditandatangani oleh kunci yang disimpan di /usr/share/keyrings/jenkins-keyring.asc. Ini memastikan bahwa hanya paket yang ditandatangani oleh Jenkins yang akan dianggap valid.
 - **https://pkg.jenkins.io/debian-stable**: Ini adalah URL dari repositori paket Jenkins yang stabil untuk Debian (dan Ubuntu).
 - **binary/**: Menentukan bahwa repositori ini berisi paket biner (program yang sudah dikompilasi).
- **|**: Ini adalah operator pipe. Output dari perintah echo akan dialirkan (diteruskan) sebagai input ke perintah berikutnya.
- **sudo tee /etc/apt/sources.list.d/jenkins.list**: Perintah tee membaca input dari standar input dan menuliskannya ke standar output (yang dalam kasus ini dibuang oleh > /dev/null) dan juga ke satu atau lebih file. Di sini, input dari perintah echo akan ditulis ke file /etc/apt/sources.list.d/jenkins.list. File ini adalah salah satu file konfigurasi yang digunakan oleh APT untuk menentukan lokasi dari mana paket-paket dapat diunduh dan diinstal. Membuat file terpisah di direktori /etc/apt/sources.list.d/ adalah praktik yang baik untuk mengelola repositori pihak ketiga.

Langkah 4: Perbarui Indeks Paket APT

Setelah menambahkan repositori Jenkins, perlu diperbarui indeks paket APT agar sistem mengenali paket Jenkins yang tersedia. Jalankan perintah berikut:

sudo apt update

```
ubuntu_admin@jenkins:~$ sudo apt update
Hit:1 http://asia-southeast1.gce.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://asia-southeast1.gce.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://asia-southeast1.gce.archive.ubuntu.com/ubuntu focal-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:6 https://packages.cloud.google.com/apt google-cloud-ops-agent-focal-2 InRelease
Hit:7 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
```

Penjelasan: Perintah ini digunakan untuk memperbarui indeks paket lokal dari semua repositori yang telah dikonfigurasi dalam sistem, termasuk repositori Jenkins yang baru ditambahkan. Proses ini mengunduh informasi terbaru tentang paket yang tersedia, versi mereka, dan dependensinya dari setiap repositori. Penting untuk menjalankan perintah ini setelah menambahkan repositori baru agar sistem mengetahui keberadaan paket Jenkins. Perintah ini tidak mengunduh atau menginstal paket apa pun; hanya memperbarui informasi tentang paket yang tersedia.

Langkah 5: Menginstal Jenkins

Sekarang siap untuk menginstal paket Jenkins. Jalankan perintah berikut:

sudo apt install jenkins

```
ubuntu_admin@jenkins:~$ sudo apt install jenkins
```

Penjelasan: Perintah ini akan mengunduh dan menginstal paket jenkins beserta semua dependensinya. Proses ini mungkin memakan waktu beberapa menit tergantung pada kecepatan koneksi internet.

Langkah 6: Kelola Layanan Jenkins

Setelah instalasi selesai, Jenkins akan berjalan sebagai layanan (service) di sistem. Dapat digunakan perintah systemctl untuk mengelola layanan ini.

1. **Mulai layanan Jenkins:** Jika layanan Jenkins belum dimulai secara otomatis, dapat dimulai dengan perintah:

sudo systemctl start jenkins

```
ubuntu_admin@jenkins:~$ sudo systemctl start jenkins
```

Penjelasan: Perintah ini akan memulai proses Jenkins

2. **Aktifkan layanan Jenkins untuk memulai saat boot:** Untuk memastikan Jenkins dimulai secara otomatis setiap kali sistem dihidupkan, jalankan perintah:

sudo systemctl enable jenkins

```
ubuntu_admin@jenkins:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
```

Penjelasan: Perintah ini akan mengkonfigurasi layanan Jenkins untuk secara otomatis dimulai saat sistem boot.

3. **Periksa status layanan Jenkins:** Dapat diperiksa status layanan Jenkins untuk memastikan bahwa layanan tersebut berjalan dengan benar menggunakan perintah:

sudo systemctl status jenkins

```
ubuntu_admin@jenkins:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-03-20 08:05:58 UTC; 1min 9s ago
     Main PID: 519 (java)
       Tasks: 44 (limit: 1134)
      Memory: 314.4M
    CGroup: /system.slice/jenkins.service
            └─519 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins
```

Penjelasan: Perintah ini akan menampilkan informasi tentang status layanan Jenkins, termasuk apakah layanan tersebut aktif (berjalan) atau tidak, serta log terbaru.

Langkah 7: Akses Antarmuka Web Jenkins

Secara default, Jenkins berjalan pada port 8080. Antarmuka web Jenkins dapat diakses melalui browser web dengan membuka alamat IP server Ubuntu diikuti dengan port 8080.

`http://alamat_ip_server:8080`

Ganti `alamat_ip_server` dengan alamat IP aktual server Ubuntu.

Langkah 8: Konfigurasi Awal Jenkins

Saat pertama kali mengakses antarmuka web Jenkins, akan terlihat halaman "Unlock Jenkins".

1. Jenkins secara otomatis membuat kata sandi administrator awal yang disimpan dalam file di server. Perlu diambil kata sandi ini untuk melanjutkan konfigurasi. Jalankan perintah berikut di terminal server:

`sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

```
ubuntu_admin@jenkins:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
f937f313a0e04150a1fa0dabc9035550
```

2. Salin kata sandi yang ditampilkan di terminal dan tempelkan ke kolom "Administrator password" di halaman web Jenkins. Kemudian klik tombol "Continue".
3. Instal Plugin yang Direkomendasikan atau Pilih Plugin: Akan diberikan dua opsi:
 - Install suggested plugins: Opsi ini akan menginstal sekumpulan plugin yang umum digunakan dan direkomendasikan untuk sebagian besar penggunaan Jenkins. Ini adalah opsi yang baik untuk pemula.
 - Select plugins to install: Opsi ini memungkinkan untuk memilih plugin secara manual dari daftar yang tersedia. Opsi ini dapat dipilih jika memiliki kebutuhan plugin spesifik.

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.492.2

Pilih salah satu opsi dan tunggu hingga proses instalasi plugin selesai. Proses ini mungkin memakan waktu beberapa menit tergantung pada jumlah plugin yang dipilih dan kecepatan koneksi internet.

- Setelah instalasi plugin selesai, akan diminta untuk membuat pengguna administrator pertama. Isi formulir dengan informasi yang diminta (Nama Pengguna, Kata Sandi, Konfirmasi Kata Sandi, Nama Lengkap, Alamat Email) dan klik tombol "Save and Continue".

Create First Admin User

Username

Avi

Password

Confirm password

Full name

Avi Majid

E-mail address

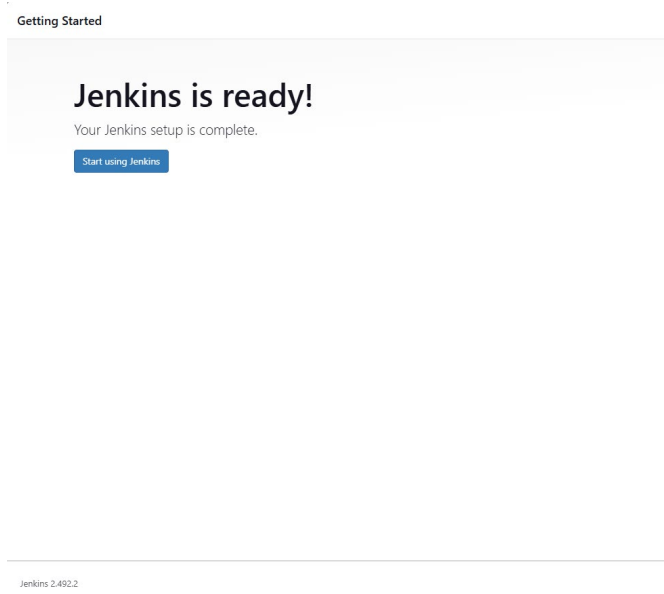
avimdjd65@gmail.com

Jenkins 2.492.2

[Skip and continue as admin](#)

[Save and Continue](#)

5. Akan diminta untuk mengkonfigurasi URL untuk instans Jenkins. Secara default, URL akan diisi dengan alamat yang digunakan untuk mengakses Jenkins. Dapat dibiarkan seperti itu atau diubah jika perlu. Klik tombol "Save and Finish".
6. Akan terlihat pesan yang menunjukkan bahwa Jenkins siap digunakan. Klik tombol "Start using Jenkins" untuk mengakses dasbor Jenkins.



Langkah-langkah Membuat Job Pengecekan Disk:

Langkah 1: Buat Job Baru

Pada halaman utama Jenkins, klik tombol "Create a new job".

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job



Set up a distributed build

Set up an agent



Configure a cloud



Learn more about distributed builds



Langkah 2: Masukkan Nama Job

- Di kolom "Enter an item name", berikan nama yang deskriptif untuk job. Contoh: Cek_Disk_Server.

- Pilih jenis job. Untuk kasus sederhana ini, pilih "Pipeline".
- Klik tombol "OK".

New Item

Enter an item name

Cek_Disk_Server

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.



Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

OK

Langkah 4: Definisikan pipeline

Definisikan pipeline dengan format yang telah ditentukan untuk membuatnya yaitu Jenkins Pipeline. Pipeline yang dibuat kali ini sangat sederhana guna mendemonstrasi cara pembuatan pipeline, yaitu hanya pengecekan disk. Pipeline sebagai berikut:

```
pipeline {
  agent any

  stages {
    stage('Cek Disk') {
      steps {
        sh "df -h"
      }
    }
  }
}
```

Penjelasan:

- **pipeline { ... }:** Blok ini adalah inti dari definisi Jenkins Pipeline. Semua langkah dan konfigurasi pipeline akan berada di dalam blok ini.

- **agent any:** Bagian ini menentukan di mana pipeline ini akan dijalankan. agent any berarti pipeline ini dapat dijalankan di agen Jenkins mana pun yang tersedia.
- **stages { ... }:** Blok ini mendefinisikan serangkaian tahapan (stages) yang akan dijalankan dalam pipeline. Tahapan ini membantu mengatur alur kerja pipeline menjadi bagian-bagian yang lebih logis dan mudah dipantau.
- **stage('Cek Disk') { ... }:** Ini adalah definisi dari sebuah tahapan yang diberi nama "Cek Disk". Nama tahapan ini akan muncul di antarmuka Jenkins, memudahkan untuk melihat progres dan status dari setiap tahapan.
- **steps { ... }:** Di dalam setiap tahapan, terdapat blok steps yang berisi daftar tindakan (steps) yang akan dieksekusi dalam tahapan tersebut.
- **sh "df -h":** Ini adalah salah satu jenis langkah (step) yang tersedia di Jenkins. sh digunakan untuk menjalankan perintah shell. Dalam kasus ini, perintah shell yang dijalankan adalah df -h.
 - df adalah perintah Linux/Unix yang digunakan untuk menampilkan informasi tentang penggunaan ruang disk pada sistem file.
 - -h adalah opsi yang digunakan dengan perintah df untuk menampilkan ukuran dalam format yang lebih mudah dibaca oleh manusia (misalnya, menggunakan satuan seperti K, M, G).

Klik save setelah selesai didefinisikan.

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

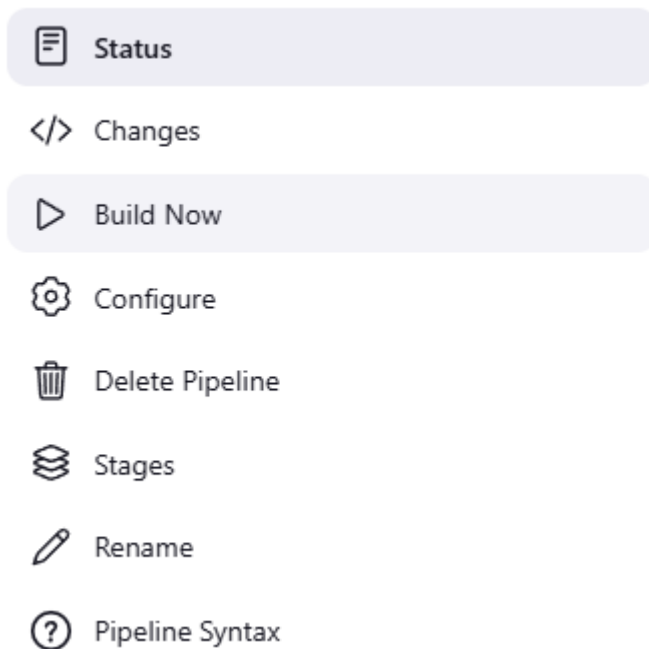
```

1 pipeline {
2     agent any
3
4     stages {
5         stage('Cek Disk') {
6             steps {
7                 sh "df -h"
8             }
9         }
10    }
11 }
12

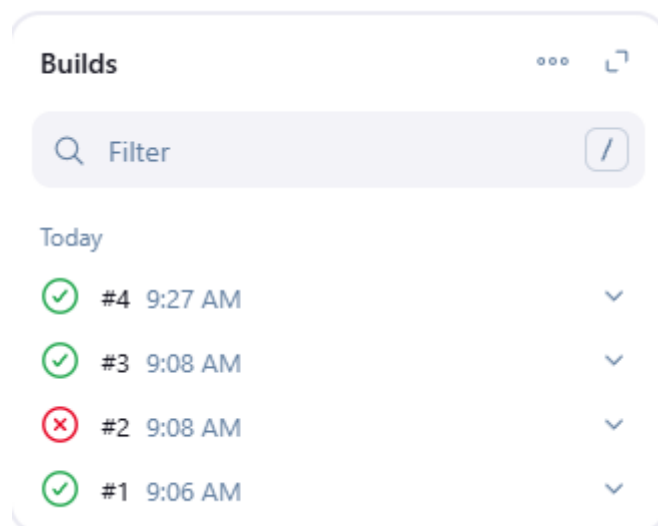
```

Langkah 5: Build

Setelah pipeline didefinisikan, klik “Build Now” pada panel bagian kiri untuk mem-build pipeline yang telah didefinisikan.



Pipeline yang telah di-build akan muncul pada bagian “Builds” dan dapat diketahui apakah pipeline berhasil di-build atau tidak.



Jika diklik pada build, dapat dilihat output dari pipeline yang telah dibuild yang menunjukkan Sukses atau Gagal. Contoh sebagai berikut:



Console Output

```
Started by user Avi Majid
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Cek_Disk_Server
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cek Disk)
[Pipeline] sh
+ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        9.6G  4.5G  5.1G  47% /
devtmpfs         473M    0  473M   0% /dev
tmpfs            477M    0  477M   0% /dev/shm
tmpfs            96M  1012K   95M   2% /run
tmpfs            5.0M    0   5.0M   0% /run/lock
tmpfs            477M    0  477M   0% /sys/fs/cgroup
/dev/loop0       64M   64M    0 100% /snap/core20/2434
/dev/loop1       64M   64M    0 100% /snap/core20/2496
/dev/loop2      401M  401M    0 100% /snap/google-cloud-cli/307
/dev/loop3       92M   92M    0 100% /snap/lxd/29619
/dev/loop4       92M   92M    0 100% /snap/lxd/32662
/dev/loop5       45M   45M    0 100% /snap/snapd/23545
/dev/sda15       105M   6.1M   99M   6% /boot/efi
tmpfs            96M    0   96M   0% /run/user/1001
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```