

CI/CD Jenkins

Detail Materi



Konsep Dasar Jenkins.

Indikator :
Dapat menjalankan konsep CI/CD
menggunakan Jenkins pada server testing dan
Server Production.



Instalasi Jenkins pada Host
dan juga Container



CI/CD Menggunakan Jenkins



Management Node Untuk
Membagi Beban.



Deployment Aplikasi pada Node

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 elektronik maupun mekanis, termasuk mecropy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis dan Penerbit.

Penulis : Estu Fardani

Editor: Muhammad Fakhri Abdillah, Iqbal Ilman Firdaus

Revisi Batch 4

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

Cover.....	1
9. CI/CD Jenkins.....	4
Learning Outcomes.....	4
Outline Materi.....	4
9.1. Pengertian CI/CD.....	5
9.2. Pengembangan dan Hubungan dengan Git Flow.....	6
9.3. Tool CI/CD.....	7
9.4. Jenkins.....	8
9.4.1. Apa itu Jenkins.....	8
9.4.2. Keuntungan Menggunakan Jenkins.....	8
9.5. Instalasi Jenkins.....	9
9.6. Setup Jenkins.....	11
9.6.1. Setup Awal Jenkins.....	11
9.6.2. Jenkins Menu.....	15
9.6.2.1. New Item.....	16
9.6.2.2. Manage Jenkins.....	17
9.6.3. Jenkins Job.....	18
9.6.4. Exercise.....	21
9.7. Mengatur Domain Jenkins.....	22
9.8. Summary.....	24

9.

CI/CD Jenkins

Learning Outcomes

Setelah selesai mempelajari bab ini, peserta mampu :

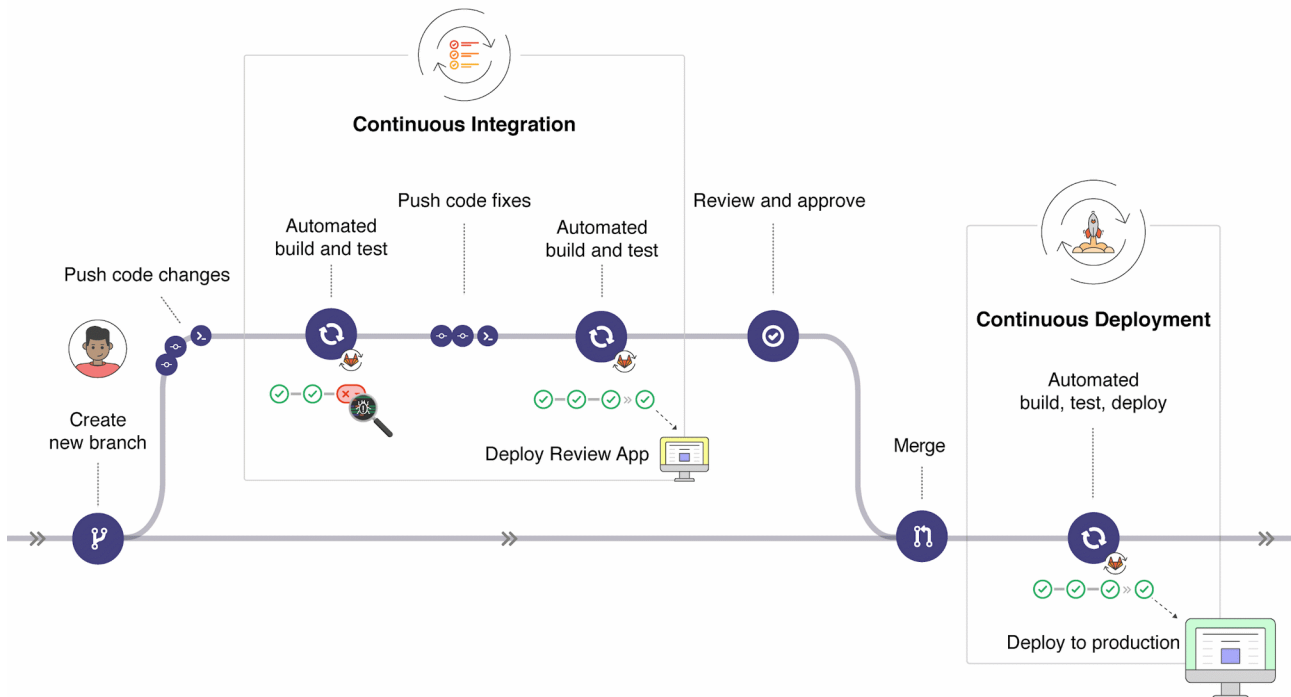
1. Memahami Konsep Dasar Jenkins.
2. Menginstall Jenkins pada Host.
3. Melakukan CI/CD Menggunakan Jenkins.

Outline Materi

1. Konsep dasar Jenkins
2. Installasi Jenkins di host
3. Setup awal jenkins
4. Jenkins menu dan Job
5. Tambah, hapus, dan jenkins node
6. Mengatur Domain Jenkins

9.1. Pengertian CI/CD

CI/CD adalah istilah penting dalam dunia DevOps. Dalam perjalanannya ada 3 istilah yang akan selalu kita gunakan.



1. Continuous Integration (CI)

Pertimbangkan aplikasi yang kodenya disimpan dalam repositori Git di GitLab. *Developer* mendorong perubahan kode setiap hari, beberapa kali sehari. Untuk setiap *push* ke repositori, Kita dapat membuat satu set skrip untuk membuat dan menguji aplikasi secara otomatis, mengurangi kemungkinan memperkenalkan kesalahan pada aplikasi yang ada.

2. Continuous Delivery (CD)

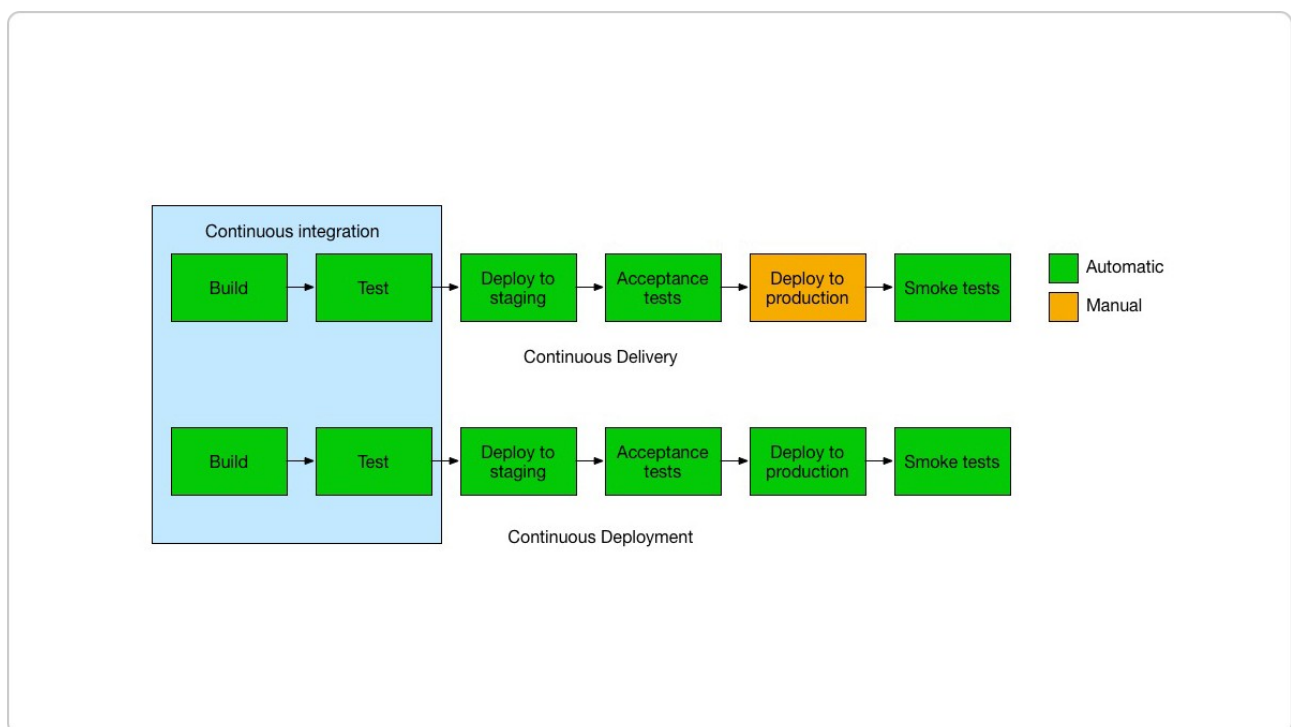
Continuous Delivery adalah langkah di luar Integrasi Berkelanjutan. Aplikasi tidak hanya dibangun dan diuji pada setiap perubahan kode yang di-*push* ke basis kode, tetapi, sebagai langkah tambahan, aplikasi ini juga digunakan secara terus menerus, meskipun penerapannya dipicu secara manual. Metode ini memastikan kode diperiksa secara otomatis tetapi

membutuhkan campur tangan manusia untuk secara manual dan strategis memicu penyebaran perubahan.

3. Continuous Deployment (CDP)

Continuous Deployment juga merupakan langkah lebih jauh dari *Continuous Integration*, mirip dengan *Continuous Delivery*. Perbedaannya adalah bahwa alih-alih menggunakan aplikasi secara manual, kita mengaturnya untuk digunakan secara otomatis. Sama sekali tidak memerlukan campur tangan manusia untuk memiliki aplikasi kita untuk di *deploy*.

Perbedaan Ketiganya dapat dilihat pada gambar berikut:



9.2. Pengembangan dan Hubungan dengan Git Flow

Dalam kenyataannya di dunia kerja setidaknya ada 3 lingkungan pengembangan wajib. Dan beberapa lingkungan pendukung. Step-step ini bisa saja diimplementasikan berbeda, sesuai kebutuhan tiap-tiap perusahaan.

Berikut lingkungan tersebut dan beberapa penjelasannya:

- *Development*. Dalam fase ini, kode akan diuji coba oleh pembuat kode. Biasanya dalam lumbung kode sumber menggunakan nama cabang (*branch*) *feature/namafitur*.
- *Staging*. Dalam fase ini, kode akan diuji coba oleh Tim QA (Jaminan Kualitas). Biasanya dalam lumbung kode sumber menggunakan nama cabang *staging*
- *Production*. Dalam fase ini, kode akan diuji coba oleh *User*. Biasanya menggunakan cabang *master*.

9.3. Tool CI/CD

Dalam implementasinya di dunia kerja, butuh beberapa tool untuk menjalankan proses CI/CD maupun CDP. Beberapa tools ini biasanya merupakan produk open source. Beberapa bisa menggunakan versi gratis, beberapa memilih menggunakan versi berbayarnya.

Berikut adalah beberapa tools CI/CD yang biasa digunakan di dunia industri.

1. **Travis-CI**. Travis merupakan CI/CD kawakan yg sangat powerfull dan sudah menghasilkan aplikasi CI/CD turunan yang disesuaikan dengan kebutuhan masing-masing. Biasanya digunakan pada proyek-proyek besar di github. Proses konfigurasi CI/CD disimpan dalam repo dengan nama `".travis-ci.yml"`.
2. **Gitlab-CI**. Gitlab Runner merupakan CI/CD dari gitlab. Mengadopsi konfigurasi dari Travis. Konfigurasi CI/CD ini disimpan dalam repo dengan nama `".gitlab-ci.yml"`
3. **Jenkins**. Ini merupakan tool CI/CD hasil pengembangan Bamboo dari Hudson. Jenkins berjalan diatas JVM. Konfigurasi CI/CD bisa dibuat dalam format `".groovy"` dan ada beberapa yang menggunakan format `"Jenkinsfile"`.

9.4. Jenkins

9.4.1. Apa itu Jenkins

Jenkins adalah salah satu tools yang sangat populer di dunia DevOps. Bukan hanya karena merupakan produk open source, tapi karena kemampuan serta segudang plugins yang tersedia. Hal ini tidak terlepas dari besarnya kontribusi dari komunitas Jenkins. Dalam modul ini kita akan belajar bagaimana membuat otomasi dan memanfaatkan plugins yang ada.



Logo Jenkins

Jenkins sendiri merupakan software menunjang proses Continuous Integration, nantinya jenkins akan dipasang pada server central yang sedang kita gunakan. Berikut merupakan diagram sederhana bagaimana jenkins dapat bekerja.

9.4.2. Keuntungan Menggunakan Jenkins

Ada beberapa keuntungan yang bisa kita gunakan apabila kita menggunakan *software* jenkins di server kita, berikut keuntungan tersebut.

1. Ini adalah *open source* dan mudah digunakan, mudah dipasang dan tidak memerlukan instalasi atau komponen tambahan.
2. Mudah dikonfigurasi. Jenkins dapat dengan mudah dimodifikasi dan diperluas. Ini menyebarkan kode secara instan, menghasilkan laporan pengujian. Jenkins dapat dikonfigurasi sesuai dengan persyaratan untuk integrasi berkelanjutan dan pengiriman berkelanjutan.
3. *Platform* Independen. Jenkins tersedia untuk semua platform dan sistem operasi yang berbeda, baik OS X, Windows atau Linux.

4. Banyak Plugin yang tersedia. Kumpulan plugin yang luas membuat Jenkins fleksibel dan memungkinkan pembangunan, penggelaran dan mengotomatisasi di berbagai platform.
5. Sebagian besar pekerjaan integrasi dilakukan secara otomatis. Oleh karena itu lebih sedikit masalah integrasi. Ini menghemat waktu dan uang selama jangka waktu proyek.

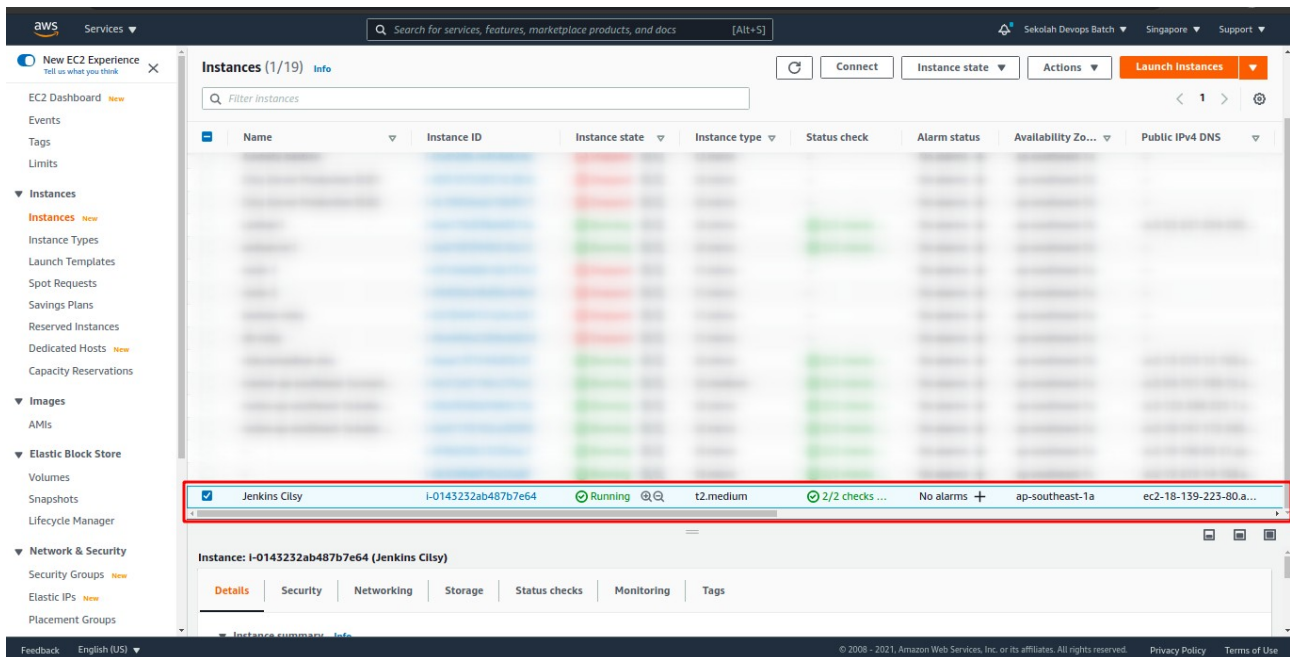
9.5. Instalasi Jenkins

Jenkins dapat diinstall dalam banyak konfigurasi. Namun pada praktik di Kelas. Kita akan memasang Jenkins di EC2 dengan Sistem Operasi Ubuntu.

Hal pertama yang harus kita lakukan adalah menyiapkan EC2 Instance terlebih dahulu, kalian bisa membuat EC2 Instance baru dengan menggunakan spesifikasi sebagai berikut :

- AMI **Ubuntu Server 18.04 LTS**
- Instance type size **t2.medium**
- Isikan Tag bila perlu, bisa menggunakan nama kalian misal : **Jenkins-Cilsy**
- Pilih security group yang ada, bisa juga buat baru. Tambahkan **port 8080 dan 50000** pada security group karena jenkins akan menggunakan port tersebut.
- Gunakan Keypair yang sudah ada dan kalian miliki.

Setelah itu kalian bisa menunggu hingga instance yang kalian buat sudah siap untuk digunakan dan di installkan Jenkins.



Instance Jenkins

Selanjutnya kita remote instance yang sudah kita buat tadi, lalu kita tambahkan repository jenkins karena repo bawaan ubuntu biasanya hanya menyediakan versi yang lawas.

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > \
/etc/apt/sources.list.d/jenkins.list'
```

Setelah itu, kita akan memasang openjdk sebagai dependensi jenkins.

```
sudo apt-get update
sudo apt-get install openjdk-8-jdk
sudo apt-get install jenkins
```

Setelah jenkins selesai kita install, sekarang kita check apakah service-nya sudah berjalan.

```
sudo systemctl status jenkins
```

Jika status sudah aktif, kita bisa mengakses **Jenkins** melalui **port 8080**. Sebelum lanjut ke *first time setup*, kita akan membahas instalasi Jenkins dengan docker.



9.6. Setup Jenkins

9.6.1. Setup Awal Jenkins

Pada bagian ini kita akan melakukan setup pada jenkins yang sudah terpasang. Jenkins dapat diakses melalui web browser dengan memanggil IP_ADDRESS:8080



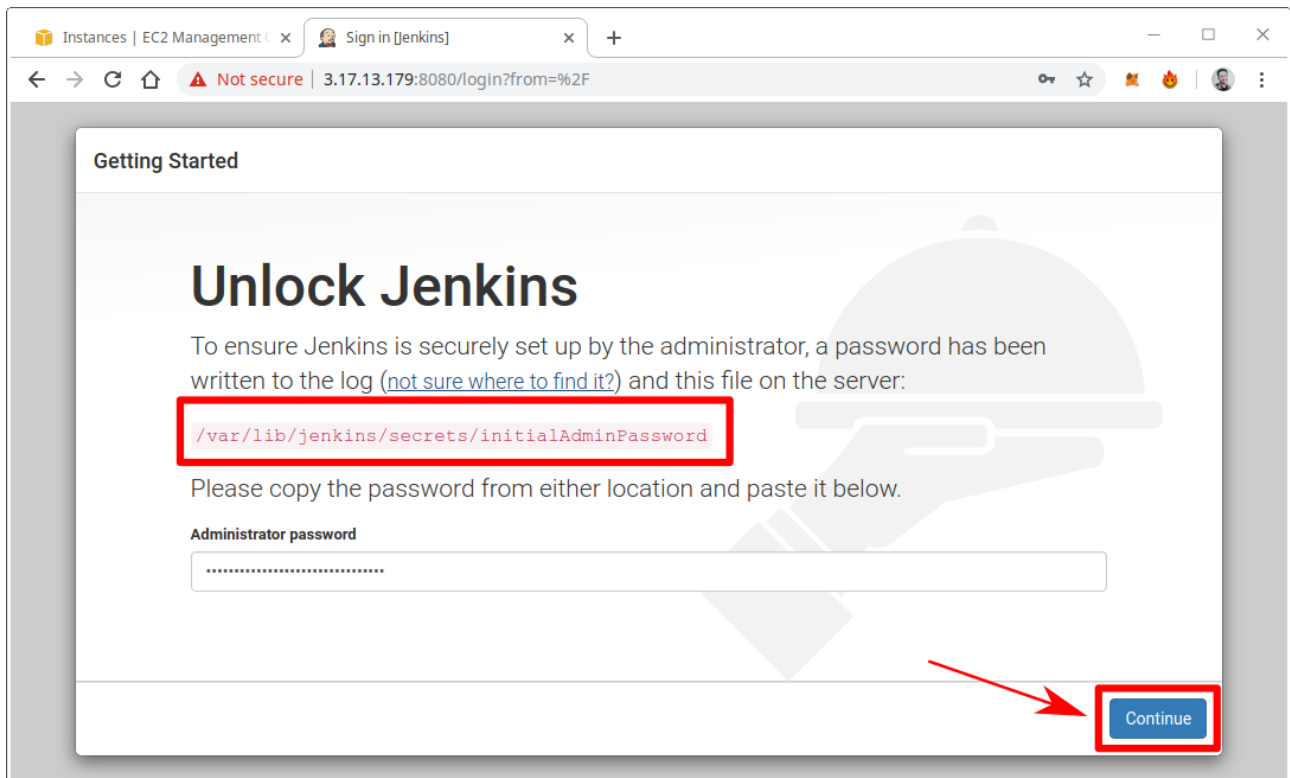
Pada step pertama kita akan memasukkan initial password yang bisa ditemukan di `/var/jenkins_home/secrets/initialAdminPassword`. File tersebut hanya bisa diakses oleh user root, maka kita bisa harus masuk sebagai root user atau menggunakan sudo. Jika sudah salin dan tempel password lalu klik tombol *Continue*.

```
ubuntu@ip-172-31-25-183:~$ sudo systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Tue 2021-02-09 08:02:42 UTC; 1min 22s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 4686)
   CGroup: /system.slice/jenkins.service

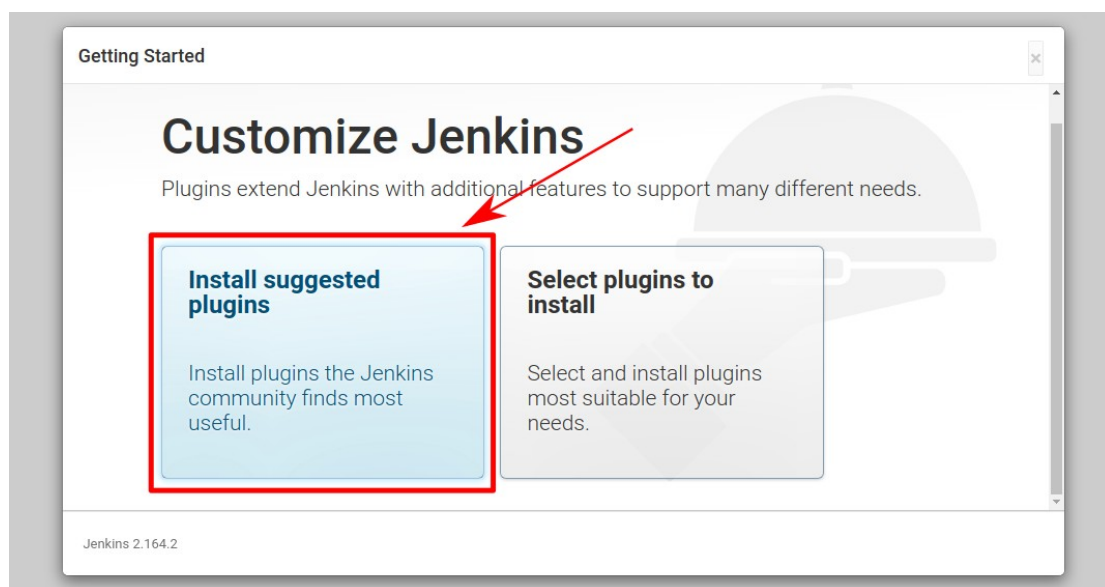
Feb 09 08:02:41 ip-172-31-25-183 systemd[1]: Starting LSB: Start Jenkins at boot
Feb 09 08:02:41 ip-172-31-25-183 jenkins[7783]: Correct java version found
Feb 09 08:02:41 ip-172-31-25-183 jenkins[7783]: * Starting Jenkins Automation S
Feb 09 08:02:41 ip-172-31-25-183 su[7838]: Successful su for jenkins by root
Feb 09 08:02:41 ip-172-31-25-183 su[7838]: + ??? root:jenkins
Feb 09 08:02:41 ip-172-31-25-183 su[7838]: pam_unix(su:session): session opened
Feb 09 08:02:41 ip-172-31-25-183 su[7838]: pam_unix(su:session): session closed
Feb 09 08:02:42 ip-172-31-25-183 jenkins[7783]: ...done.
Feb 09 08:02:42 ip-172-31-25-183 systemd[1]: Started LSB: Start Jenkins at boot

ubuntu@ip-172-31-25-183:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
5a3fc681b5264d...
```

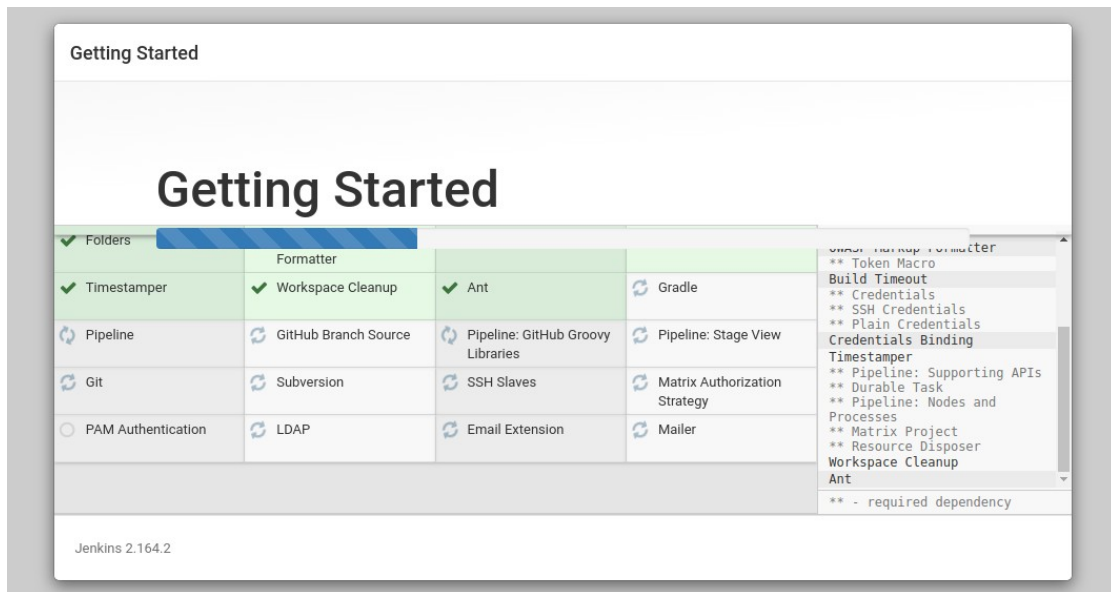




Setelah mengisi unlock jenkins, selanjutnya akan ada pilihan apakah kita akan menginstall plugin yang direkomendasikan atau memilih sendiri. Untuk kemudahan, kita pilih "Install Suggested Plugin". Akan ada progress instalasi plugin. Tunggu hingga selesai.



Pemilihan Instalasi Plugin



Proses instalasi plugin

Selanjutnya akan muncul halaman untuk membuat user baru, silahkan diisi. Di bagian paling bawah ada pilihan “Skip and Continue as admin”. Ini maksudnya, kita akan melanjutkan dengan user yang sedang dipakai yaitu : admin, dengan password yang kita dapat dari file initialAdminPassword. Tapi alangkah baiknya kita langsung membuat user admin baru.

Getting Started

Create First Admin User

Username:

taufik

Password:

.....

Confirm password:

.....

Full name:

taufik maulana

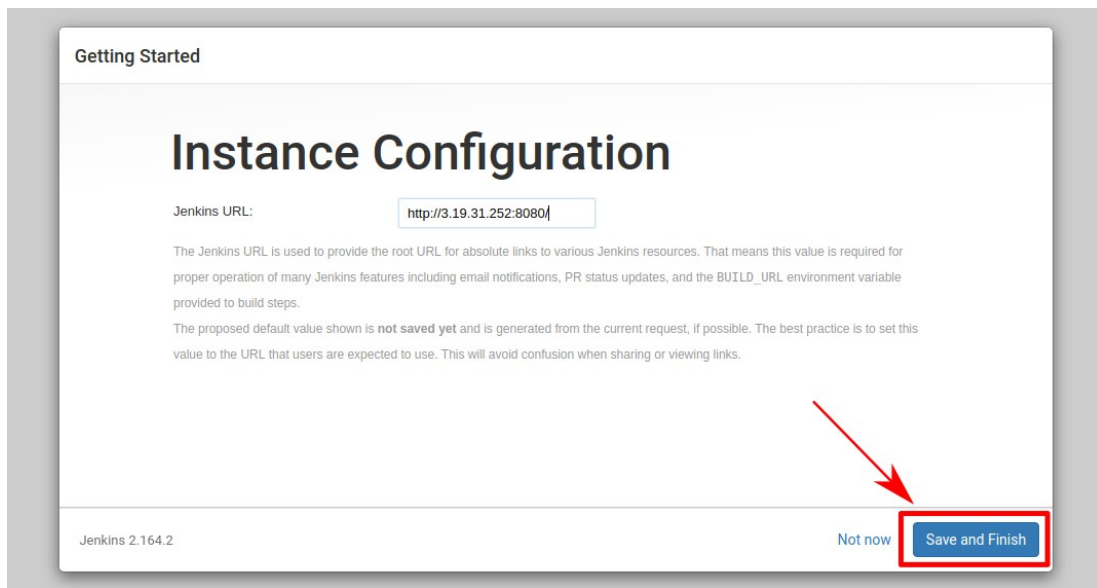
E-mail address:

taufik@mail.com

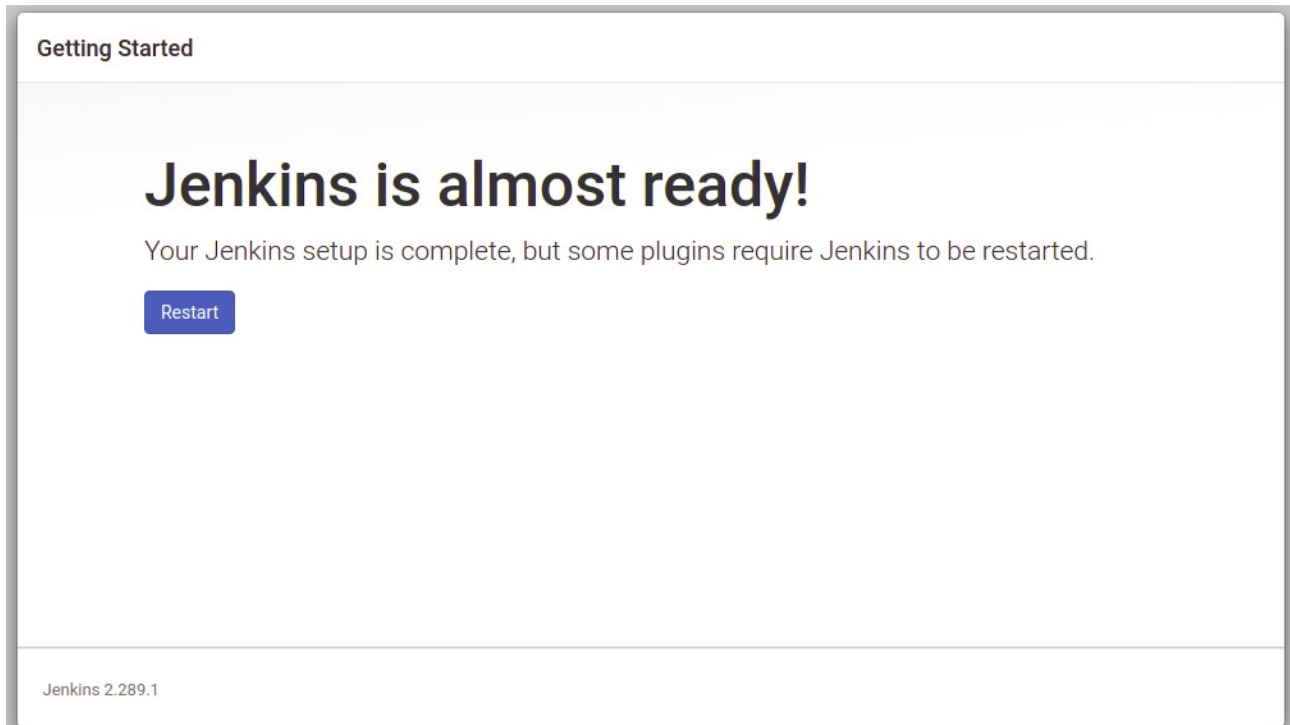
Jenkins 2.289.1

[Skip and continue as admin](#)
[Save and Continue](#)





Setelah itu akan ada setting untuk jenkins URL. Karena kita belum menggunakan domain, kita biarkan apa adanya, lalu **“Save and Finish”** dan klik **“Restart”**



Dengan begini setup dan installasi jenkins sudah selesai. Dilanjutkan dengan login menggunakan user dan password baru.



Welcome to Jenkins!

taufik

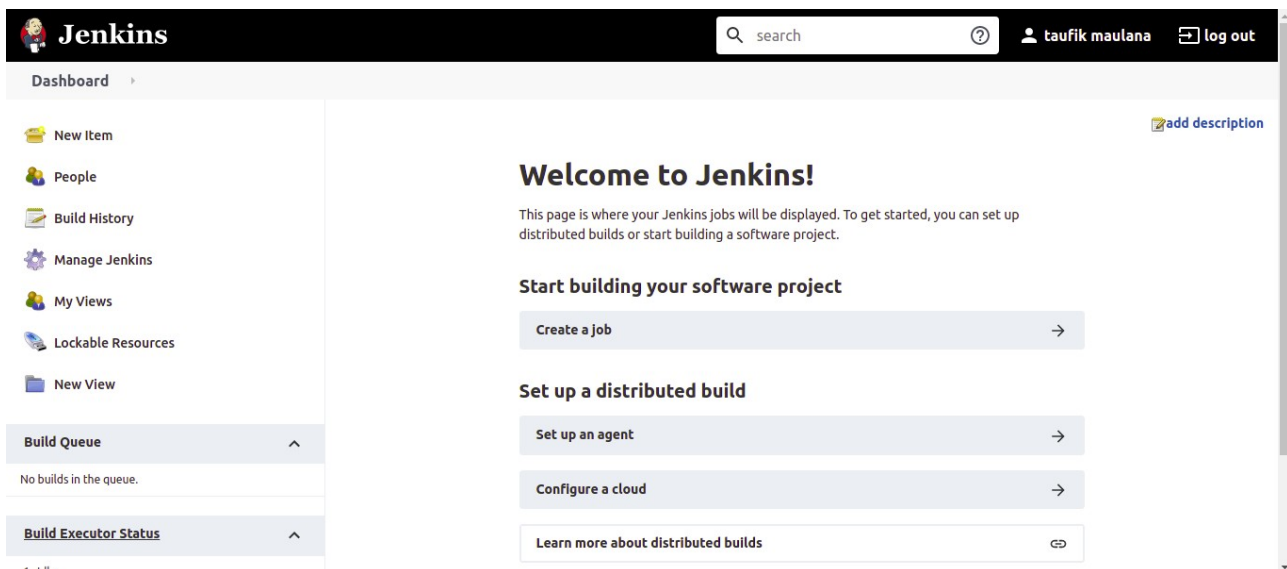
Sign in

☐ Keep me signed in

Login page Jenkins

9.6.2. Jenkins Menu

Setelah kita melakukan setup, kita akan langsung diarahkan ke dashboard jenkins. Di dashboard Jenkins terdapat beragam pilihan menu dan submenu seperti terlihat pada gambar di bawah.



The screenshot shows the Jenkins Dashboard. At the top, there's a header with the Jenkins logo, a search bar, and user information (taufik maulana) with a log out button. The main content area is divided into a left sidebar and a main panel. The sidebar contains a 'Dashboard' link and a list of menu items: New Item, People, Build History, Manage Jenkins, My Views, Lockable Resources, and New View. Below these are expandable sections for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing '1 Idle'). The main panel displays a 'Welcome to Jenkins!' message, followed by instructions on how to get started. It includes a section 'Start building your software project' with a 'Create a Job' button, and a section 'Set up a distributed build' with buttons for 'Set up an agent', 'Configure a cloud', and a link to 'Learn more about distributed builds'.



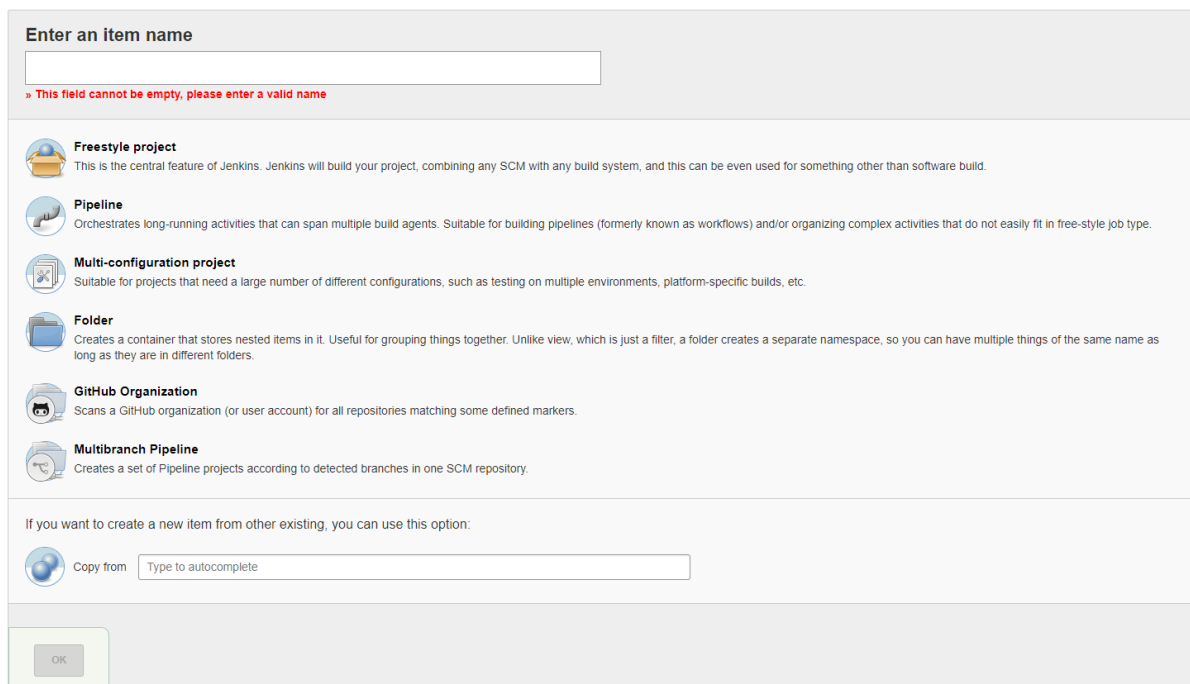
Disini kita akan coba jelaskan beberapa menu yang ada pada jenkins tersebut bersama dengan fungsi dan penggunaannya. Berikut merupakan menu-menu tersebut.

9.6.2.1. New Item

New Item adalah menu untuk membuat folder ataupun project (biasa disebut jenkins job)

Ada beberapa macam item yang paling populer dipakai, yaitu:

1. **Freestyle Project** : Ada fitur pembuatan jenkins jobs dengan konfigurasi kosong dan membebaskan user untuk mengaturnya sesuai keinginan.
2. **Pipeline** : Membuangun Jenkins Jobs dalam urutan pipeline dengan hasil akhir sebuah fiile Jenkinsfile yang disimpat pada folder root repository.
3. **Folder** : Dibutuhkan untuk mengorganisir proyek sehingga mudah diakses.
4. **Multibranch Pipeline** : Memungkinkan menjalankan CI/CD dalam beberapa branch sekaligus



Enter an item name

» This field cannot be empty, please enter a valid name

- Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- 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.
- GitHub Organization**
Scans a GitHub organization (or user account) for all repositories matching some defined markers.
- Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

If you want to create a new item from other existing, you can use this option:

Copy from

OK

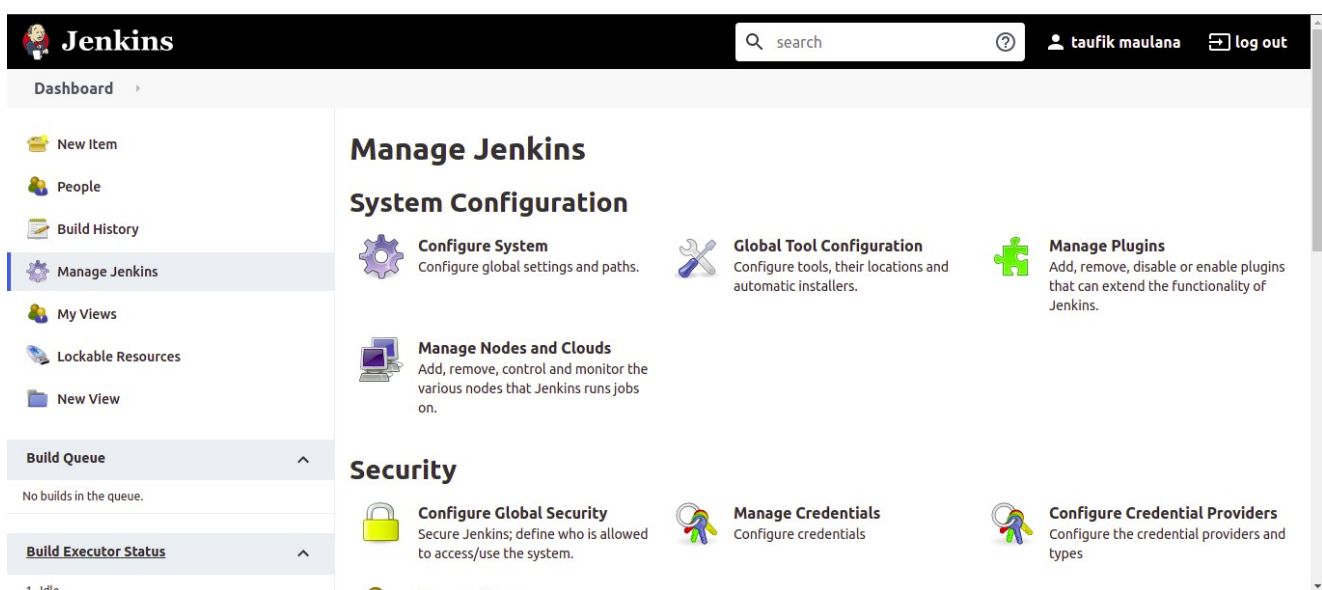
Submenu dari new item



9.6.2.2. Manage Jenkins

Manage Jenkins adalah menu untuk melakukan berbagai macam konfigurasi yang terinstall di Jenkins, Ada beberapa submenu yang terapat dalam manage jenkins seperti berikut ini.

1. **Configure System** : Sebagian besar konfigurasi ada disini, mulai dari kofigurasi jenkins sampai konfigurasi Plugin
2. **Configure Credentials** : Sesuai namanya, kita bisa mengatur credential untuk repository, docker registry, SSH, dll.
3. **Manage Plugins** : Menginstall, mengupdate ataupun menghapus plugins.
4. **System Information** : Menampilkan informasi terkait server Jenkins
5. **Jenkins Log** : Untuk mengecek log file jenkins
6. **Load Statistics** : Statistik tentang executor dan beban kerjanya
7. **Manage Nodes** : Mengatur, menambah atau menghapus Nodes/worker/executor
8. **Manage Users** : Mengatur, menambah atau menghapus user.

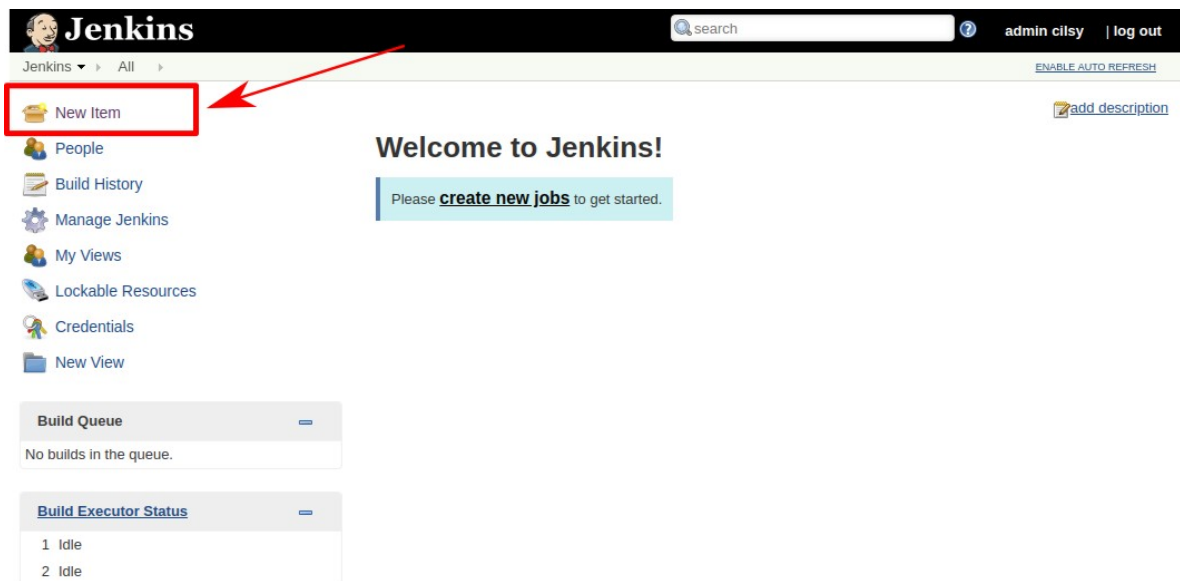


Submenu Manage Jenkins



9.6.3. Jenkins Job

Kali ini kita akan coba membahas **freestyle jobs**. Disini kita akan coba membuatnya dengan Klik pada **New item**.



lalu isi nama project dan klik **Freestyle Project**. Setelah itu klik **OK**.

Enter an item name

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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.

Kita akan masuk ke dashboard Jenkins Job kita. Untuk **Description**, tulislah keterangan untuk job yang sedang kita buat.

General

Source Code Management

Build Triggers

Build Environment

Build

Post-build Actions

Description

Job ini adalah percobaan dari sekolah devops cilsy

[Plain text] [Preview](#)

☐ Discard old builds

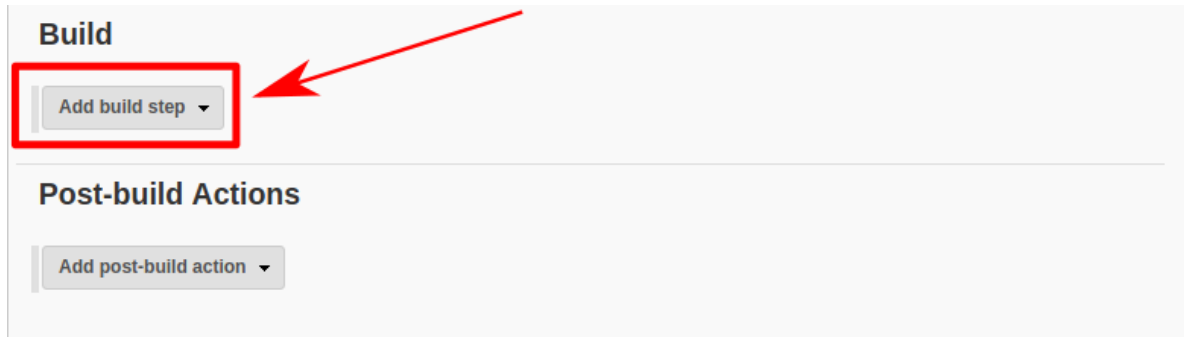
☐ GitHub project

Di bagian **Source Code Management** pilih **Git**, lalu masukkan URL git project anda (**misalkan [landing page Tekkadan](#)**) serta credential dan branchnya.

Jika Credential belum ada, kembali ke Dashboard **Jenkins>Credential>Jenkins>Global Credentials>Add Credential**. Pilih username and password untuk type, ketikkan username dan password dari github yang kita miliki, lalu klik **OK**.

Kembali ke **konfigurasi Freestyle job**. Kita juga bisa memilih **Poll SCM** untuk automatic trigger ketika ada push ke origin repository.

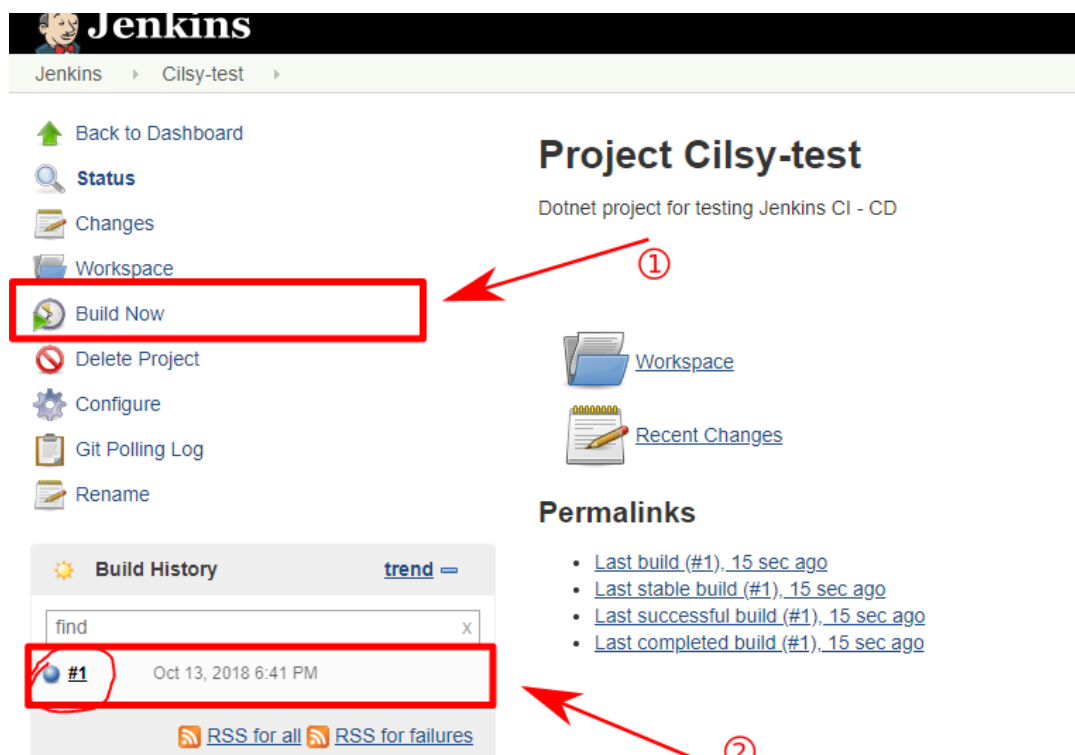
Di bagian **Build**, klik **add build step**, lalu pilih **execute shell**. Kita akan mencoba meminta jenkins untuk melakukan beberapa command.



Add build Step



Klik **Save** di paling bawah lalu klik **Build Now**. Lihat di bagian build history akan muncul icon dan job number seperti di bawah.



Klik tepat di bulatan berwarna biru, maka kita akan dilempar ke halaman **Log build** tersebut.

Console Output

```
Started by user admin
Building remotely on Jenkins-Agent (agent) in workspace /var/jenkins/workspace/Cilsy-test
Cloning the remote Git repository
Cloning repository https://gitlab.com/Irfangit/jenkins_cilsy.git
> git init /var/jenkins/workspace/Cilsy-test # timeout=10
Fetching upstream changes from https://gitlab.com/Irfangit/jenkins_cilsy.git
> git --version # timeout=10
using GIT_ASKPASS to set credentials gitlab access
> git fetch --tags --progress https://gitlab.com/Irfangit/jenkins_cilsy.git +refs/heads/*:refs/remotes/origin/*
> git config remote.origin.url https://gitlab.com/Irfangit/jenkins_cilsy.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://gitlab.com/Irfangit/jenkins_cilsy.git # timeout=10
Fetching upstream changes from https://gitlab.com/Irfangit/jenkins_cilsy.git
using GIT_ASKPASS to set credentials gitlab access
> git fetch --tags --progress https://gitlab.com/Irfangit/jenkins_cilsy.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 607ac1f8c09fb5813790ba7a579c8a0302e37b50 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 607ac1f8c09fb5813790ba7a579c8a0302e37b50
Commit message: "update command"
First time build. Skipping changelog.
[Cilsy-test] $ /bin/sh -xe /tmp/jenkins4340416244174819160.sh
+ ls
Dockerfile
Program.cs
Properties
Startup.cs
bin
dotnet.csproj
obj
+ git --version
git version 2.17.1
Finished: SUCCESS
```

Hasil dari Build Jobs

Bisa kita lihat dari hasil **job freestyle** yang sudah kita konfigurasi tadi, kita telah berhasil menjalankan beberapa perintah yang sudah kita buat sebelumnya.

9.6.4. Exercise

Soal Teori :

1. Apa yang membedakan antara Item **Freestyle Project** dengan item lainnya ?

Soal Praktek :

1. Buat sebuah Job baru dengan type **Freestyle**.
2. Build aplikasi pesbuk kedalam job baru tersebut dan lihat hasil yang terjadi !



9.7. Mengatur Domain Jenkins

Untuk mempermudah dalam menggunakan jenkins, sebaiknya memasang domain yang benar di mesin jenkins. Berikut langkah-langkah yang dilakukan untuk mengatur domain.

1. Atur subdomain sebagai A record (Address) ke ip publik Jenkins di route53. Misal jenkins.domain.id > xx.xxx.xxx.xx
2. Lakukan install nginx, letsencrypt
3. Buat ssl dengan letsencrypt
4. Pasang konfigurasi di NGINX
5. Atur URL Jenkins dari menu admin
6. Testing Domain

Masukan perintah berikut untuk melakukan instalasi nginx berserta certbot

```
sudo apt install nginx software-properties-common
sudo rm -rf /etc/nginx/sites-enabled/default
sudo add-apt-repository ppa:certbot/certbot
sudo apt update
sudo apt install certbot
sudo certbot certonly --standalone -d jenkins.domain.id
sudo mkdir /etc/nginx/ssl/
sudo openssl dhparam -out /etc/nginx/ssl/dhparam.pem 2048
sudo service nginx stop
sudo vim /etc/nginx/sites-available/jenkins.domain.id
sudo ln -s /etc/nginx/sites-available/jenkins.domain.id /etc/nginx/sites-enabled/jenkins.domain.id
sudo nginx -t
sudo systemctl restart nginx
```

Setelah instalasi selesai, masukan konfigurasi nginx berikut

```
server {
    listen 80;
    server_name jenkins.domain.id;
```



```

    return 301 https://$host$request_uri;
}
server {
    listen 443 ssl default deferred;
    server_name jenkins.domain.id;

    ssl on;

    ssl_certificate /etc/letsencrypt/live/jenkins.domain.id/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/jenkins.domain.id/privkey.pem;

    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 1440m;

    ssl_prefer_server_ciphers on;
    ssl_ciphers ECDH+AESGCM:ECDH+AES256:ECDH+AES128:DH+3DES:!ADH:!AECDH:!MD5;

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;

    ssl_dhparam /etc/nginx/ssl/dhparam.pem;
    add_header Strict-Transport-Security "max-age=63072000; includeSubdomains";

    ssl_stapling on;
    ssl_stapling_verify on;
    ssl_trusted_certificate /etc/letsencrypt/live/jenkins.domain.id/fullchain.pem;
    resolver 8.8.8.8 8.8.4.4 valid=300s;
    resolver_timeout 5s;

    access_log          /var/log/nginx/jenkins.domain.id.access.log;
    error_log           /var/log/nginx/jenkins.domain.id.error.log;
    proxy_buffers 16 64k;
    proxy_buffer_size 128k;
    location / {
        proxy_pass http://localhost:8080;
    }
}

```



```

        proxy_next_upstream error timeout invalid_header http_500 http_502
http_503 http_504;
        proxy_redirect off;

        proxy_set_header    Host            $host;
        proxy_set_header    X-Real-IP       $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header    X-Forwarded-Proto https;
    }
}

```

9.8. Summary

1. Jenkins merupakan tool berbasis *opensource* yang berfungsi untuk memberikan otomasi pada continous integration dan continous deployment.
2. Jenkins dapat di installkan baik pada host instance maupun pada sistem container sekalipun.
3. Untuk melakukan automasi pada deployment, kita memerlukan sebuah credetial dari platfrom GIT yang akan kita gunakan.
4. Kita dapat melakukan automation deployment.