

DevOps Topics
Practice 4
INGENIERÍA DE SOFTWARE II



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Johan Sebastian Salamanca Gonzalez
Juan Camilo Monterrosa Sanchez

Universidad Nacional de Colombia
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas e Industrial
Docente Yury Niño Roa
2019

Procesos para instalar docker.

Se ejecutan los comandos respectivos para instalar Docker

```
jukcs@jukcs-ubuntu: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service →  
/lib/systemd/system/containerd.service.  
Configurando bridge-utils (1.5-15ubuntu1) ...  
Configurando ubuntu-fan (0.12.10) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service →  
/lib/systemd/system/ubuntu-fan.service.  
Configurando pigz (2.4-1) ...  
Configurando docker.io (18.09.7-0ubuntu1~18.04.4) ...  
Añadiendo el grupo 'docker' (GID 128) ...  
Hecho.  
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/sy  
stemd/system/docker.socket.  
Procesando disparadores para systemd (237-3ubuntu10.33) ...  
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...  
Procesando disparadores para ureadahead (0.100.0-21) ...  
ureadahead will be reprofiled on next reboot  
jukcs@jukcs-ubuntu:~$ sudo systemctl start docker  
jukcs@jukcs-ubuntu:~$ sudo systemctl enable docker  
Synchronizing state of docker.service with SysV service script with /lib/systemd  
/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable docker  
jukcs@jukcs-ubuntu:~$ docker --version  
Docker version 18.09.7, build 2d0083d  
jukcs@jukcs-ubuntu:~$
```

Una vez concluida la instalación se realiza una prueba de que todo quedara instalado de forma correcta.

```
jukcs@jukcs-ubuntu: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)  
   Active: active (running) since Thu 2019-12-12 20:50:12 -05; 18min ago  
     Docs: https://docs.docker.com  
    Main PID: 12410 (dockerd)  
      Tasks: 15  
     CGroup: /system.slice/docker.service  
             └─12410 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock  
  
dic 12 20:50:09 jukcs-ubuntu dockerd[12410]: time="2019-12-12T20:50:09.025401107-05:00" level=warning msg="Your kernel does not support cgroup rt period"  
dic 12 20:50:09 jukcs-ubuntu dockerd[12410]: time="2019-12-12T20:50:09.025422532-05:00" level=warning msg="Your kernel does not support cgroup rt runtime"  
dic 12 20:50:09 jukcs-ubuntu dockerd[12410]: time="2019-12-12T20:50:09.026632120-05:00" level=info msg="Loading containers: start."  
dic 12 20:50:09 jukcs-ubuntu dockerd[12410]: time="2019-12-12T20:50:09.938805406-05:00" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0  
dic 12 20:50:10 jukcs-ubuntu dockerd[12410]: time="2019-12-12T20:50:10.413998624-05:00" level=info msg="Loading containers: done."  
dic 12 20:50:10 jukcs-ubuntu dockerd[12410]: time="2019-12-12T20:50:10.440145353-05:00" level=warning msg="failed to retrieve runc version: unknown output format: runc  
dic 12 20:50:11 jukcs-ubuntu dockerd[12410]: time="2019-12-12T20:50:11.583063388-05:00" level=info msg="Docker daemon" commit=2d0083d graphdriver(s)=overlay2 version=18  
dic 12 20:50:11 jukcs-ubuntu dockerd[12410]: time="2019-12-12T20:50:11.583240195-05:00" level=info msg="Daemon has completed initialization"  
dic 12 20:50:12 jukcs-ubuntu dockerd[12410]: time="2019-12-12T20:50:12.752821181-05:00" level=info msg="API listen on /var/run/docker.sock"  
dic 12 20:50:12 jukcs-ubuntu systemd[1]: Started Docker Application Container Engine.  
~  
[lines 1-19/19 (END)]
```

Se realiza la descarga de Jenkins

```
jukcs@jukcs-ubuntu: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
jukcs@jukcs-ubuntu:~$ sudo docker pull jenkins/jenkins  
[sudo] contraseña para jukcs:  
Using default tag: latest  
latest: Pulling from jenkins/jenkins  
844c33c7e6ea: Pull complete  
ada5d61ae65d: Pull complete  
f8427fdf4292: Pull complete  
f025bafc4ab8: Pull complete  
67b8714e1225: Pull complete  
64b12da521a3: Pull complete  
2e38df533772: Pull complete  
86e76854be12: Pull complete  
6687f6493b04: Pull complete  
4057472b0863: Pull complete  
affe86263608: Pull complete  
ad7b8002bbc6: Pull complete  
0d91d7c87f18: Pull complete  
4f2e135dd0db: Pull complete  
41977bb3b6d2: Pull complete  
ad1f8c66d85f: Pull complete  
81ca1b6d5f6c: Pull complete  
cb349469b8e4: Pull complete  
Digest: sha256:48384d6b22c2a544fb04251c0776e2071ca0c2c0c98f6c4189877b808509fc34  
Status: Downloaded newer image for jenkins/jenkins:latest  
jukcs@jukcs-ubuntu:~$
```

En el primer parte se realiza el acceso a jenkins en localhost iniciando el container. En el segundo comando se realiza la configuración de la contraseña

```
jukcs@jukcs-ubuntu:~$ sudo docker run -d --name=Jenkins-Master -p 8080:8080 -p 50000:50000 -v jenkins-data:/var jenkins/jenkins  
76efbe881297a6afe2636e92911d9f5730035b84b56417b5ef23df6143db669b  
jukcs@jukcs-ubuntu:~$ sudo docker ps  
CONTAINER ID        IMAGE               NAMES               COMMAND               CREATED             STATUS              PORTS  
76efbe881297       jenkins/jenkins    Jenkins-Master      "/sbin/tini -- /usr/..." 24 seconds ago     Up 16 seconds      0.0.0.0:8080->8080/tcp,  
0.0.0.0:50000->50000/tcp  
jukcs@jukcs-ubuntu:~$ sudo docker exec Jenkins-Master cat /var/jenkins_home/secrets/initialAdminPassword  
78e61285902a4443889efa904ed7d098  
jukcs@jukcs-ubuntu:~$
```

Aquí tenemos el acceso a Jenkins desde el navegador.

Getting Started


Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/jenkins_home/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password



Continue

Se configura la URL sobre la que se correr Jenkins

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.208

Not now

Save and Finish

Se crea el usuario administrador

Getting Started

Create First Admin User

Usuario:

Contraseña:

Confirma la contraseña:

Nombre completo:

Dirección de email:

Jenkins 2.208

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

Se agregan la configuraciones iniciales

Getting Started

Bienvenido a Jenkins

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

Install suggested plugins

Install plugins the Jenkins community finds most useful.

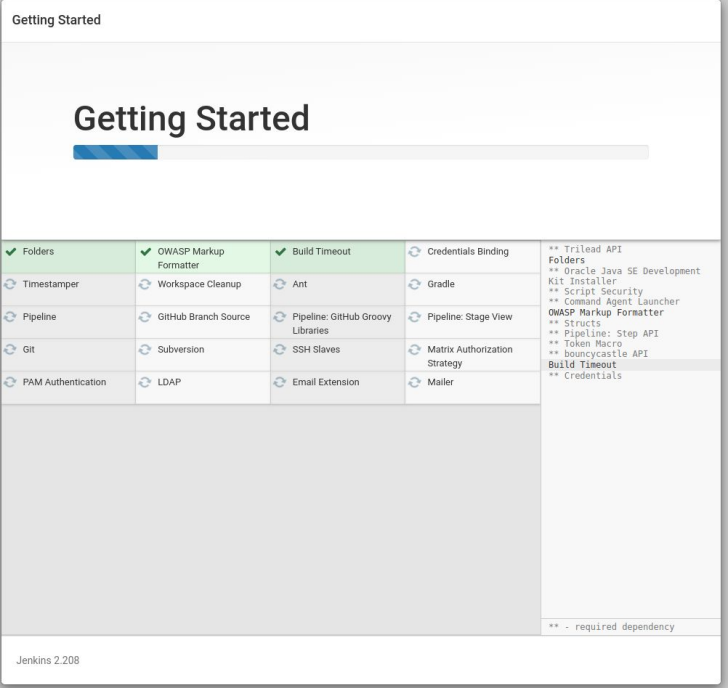
Select plugins to install

Select and install plugins most suitable for your needs.

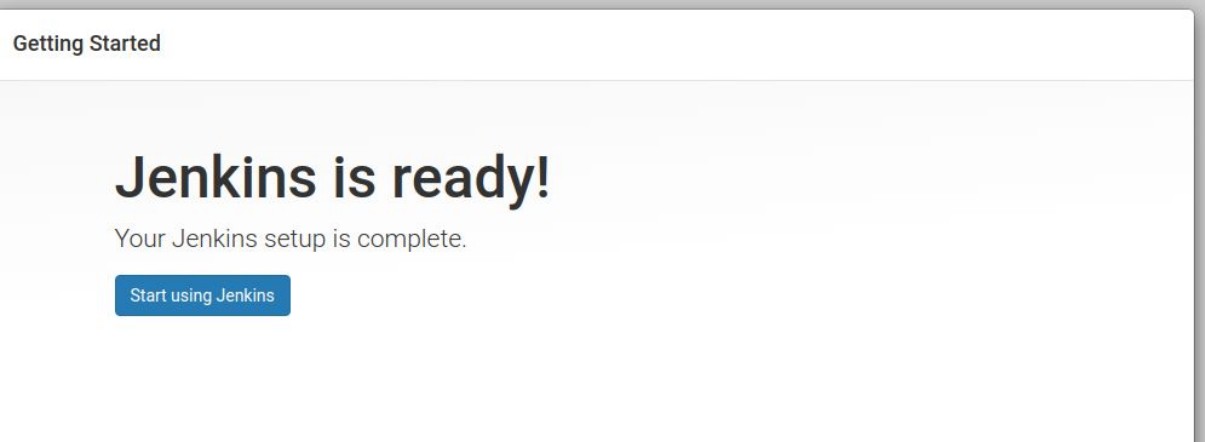


Jenkins 2.208

Se realiza automáticamente el proceso de configuración del entorno.

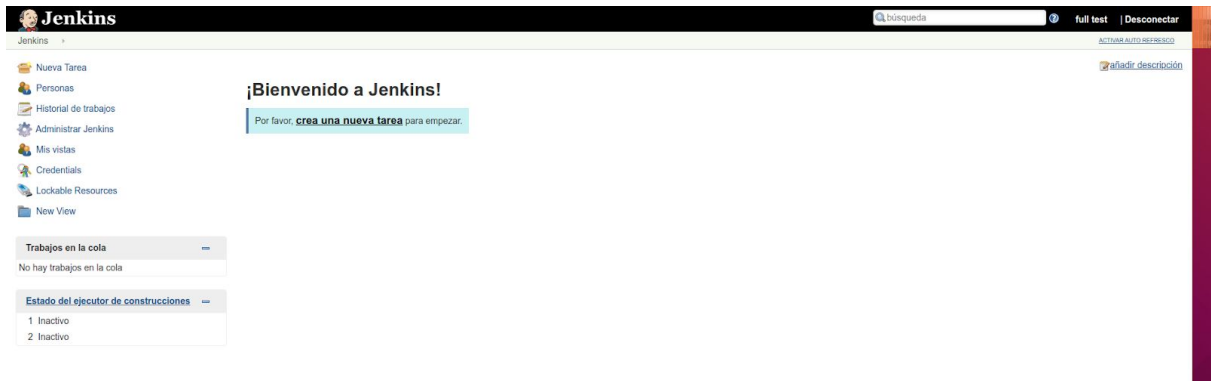


Resultado Obtenido al finalizar

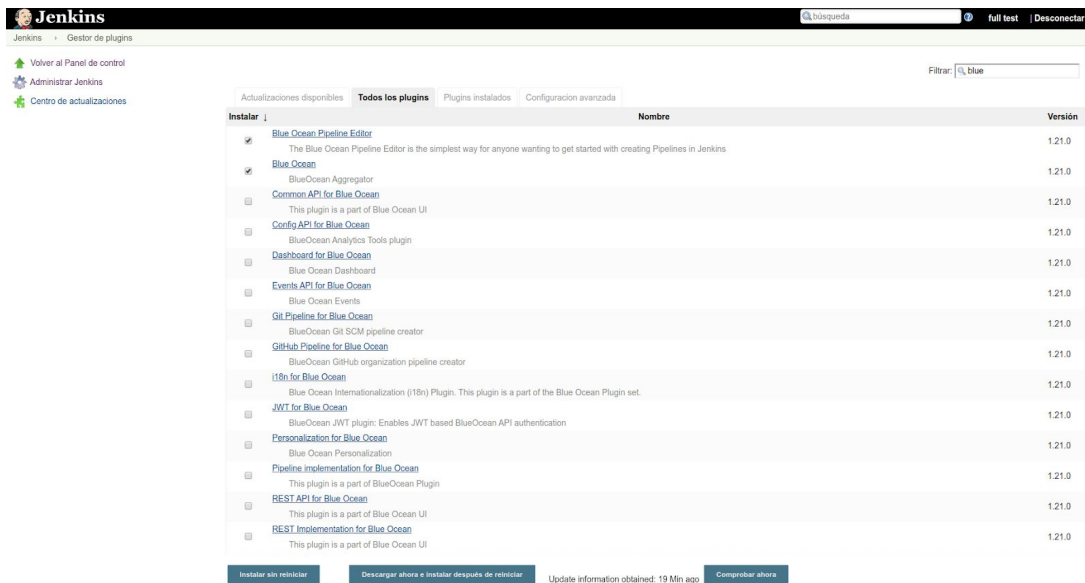


1. Instalar Blue Ocean

Panel principal al ingresar a Jenkins, aquí nos dirigimos a Administrar Jenkins para instalar Blue Ocean



Aquí nos dirigimos a Todos los Plugins y buscamos el plugin




Aquí el proceso de instalación actualizará e instalará los módulos que sean necesarios

Jenkins	Update Center
Design Language	Actualizado
Blue Ocean Core JS	Actualizado
Common API for Blue Ocean	Actualizado
REST API for Blue Ocean	Actualizado
Pub-Sub "light" Bus	Actualizado
Pipeline SCM API for Blue Ocean	Actualizado
HTML Publisher	Actualizado
Variant	Actualizado
Web for Blue Ocean	Actualizado
JWT for Blue Ocean	Actualizado
Favorite	Actualizado
REST Implementation for Blue Ocean	Actualizado
Pipeline implementation for Blue Ocean	Actualizado
GitHub Pipeline for Blue Ocean	Actualizado
Git Pipeline for Blue Ocean	Actualizado
Config API for Blue Ocean	Actualizado
Mercurial	Actualizado
Handy Uri Templates 2.x API	Actualizado
Bitbucket Branch Source	Pendiente
Bitbucket Pipeline for Blue Ocean	Pendiente
Dashboard for Blue Ocean	Pendiente
Personalization for Blue Ocean	Pendiente
JIRA	Pendiente
JIRA Integration for Blue Ocean	Pendiente
Display URL for Blue Ocean	Pendiente
Server Sent Events (SSE) Gateway	Pendiente
Events API for Blue Ocean	Pendiente
Blue Ocean Pipeline Editor	Pendiente
i18n for Blue Ocean	Pendiente
Autofavorite for Blue Ocean	Pendiente
Blue Ocean	Pendiente
Blue Ocean Pipeline Editor	Pendiente
Loading plugin extensions	Pending

2. Creación del Pipeline.

Ingresamos a Jenkins con el usuario creado anteriormente



Welcome to Jenkins!

test

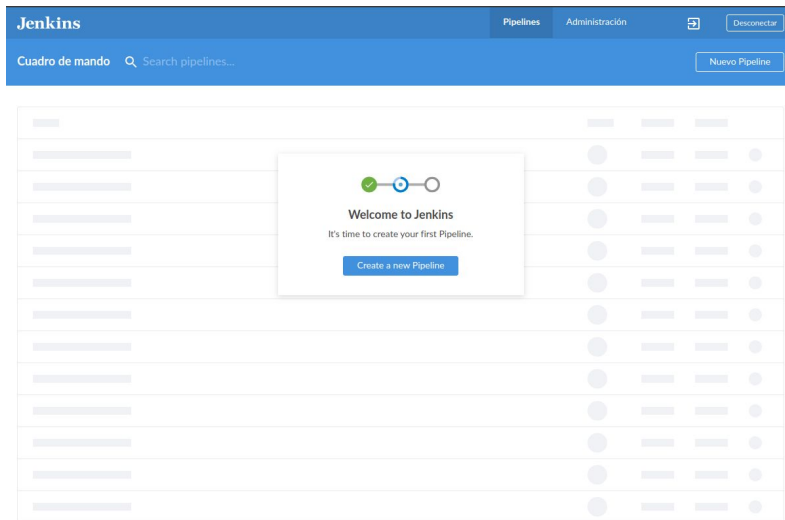
.....

Sign in

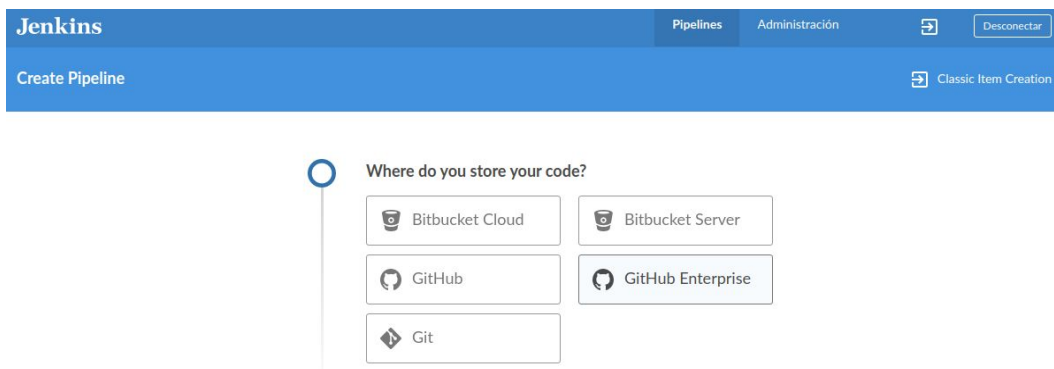
☒ Keep me signed in

Luego de ingresar en el menú lateral, En la opción de Blue Ocean damos click lo cual nos lleva a la siguiente ventana.


Aquí seleccionamos crear un nuevo Pipeline





Seguimos los pasos y seleccionamos GitHub.




Luego de generar el token de acceso en GitHub, se elige la organización y posteriormente el repositorio en el que se almacenará nuestro Pipeline

 GitHub

 Git


 GitHub Enterprise




Connect to GitHub


Jenkins needs an access token to authorize itself with GitHub.
[Create an access token here.](#)


.....





Which organization does the repository belong to?

 jcmonterrosas



Choose a repository

Loaded 10 repositories


Algoritmos


DevOps-Topics

Functional-Patterns

JavaScript

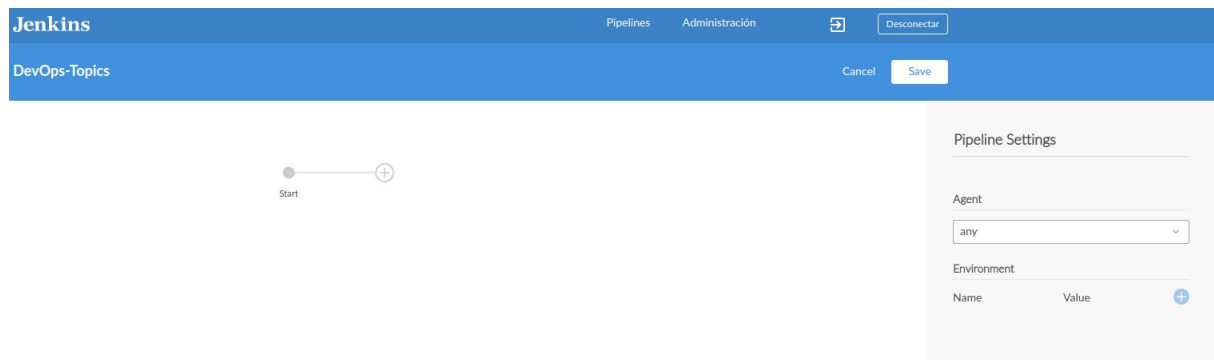
Lenguajes



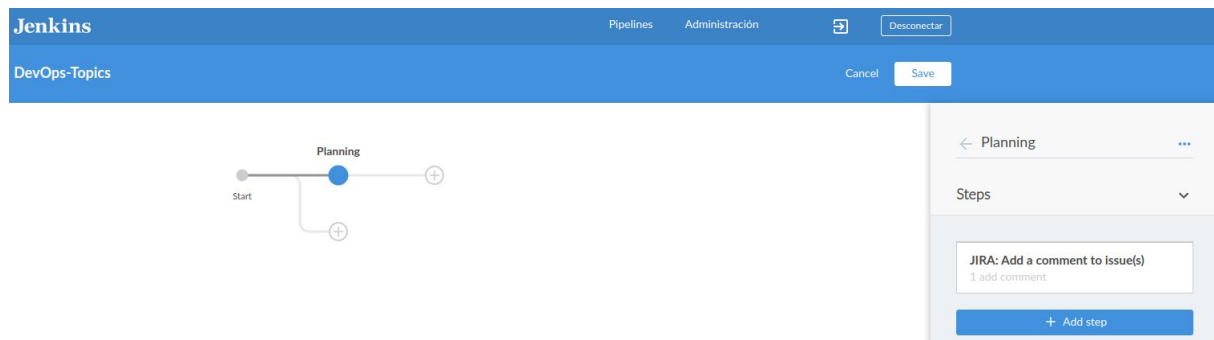


Complete

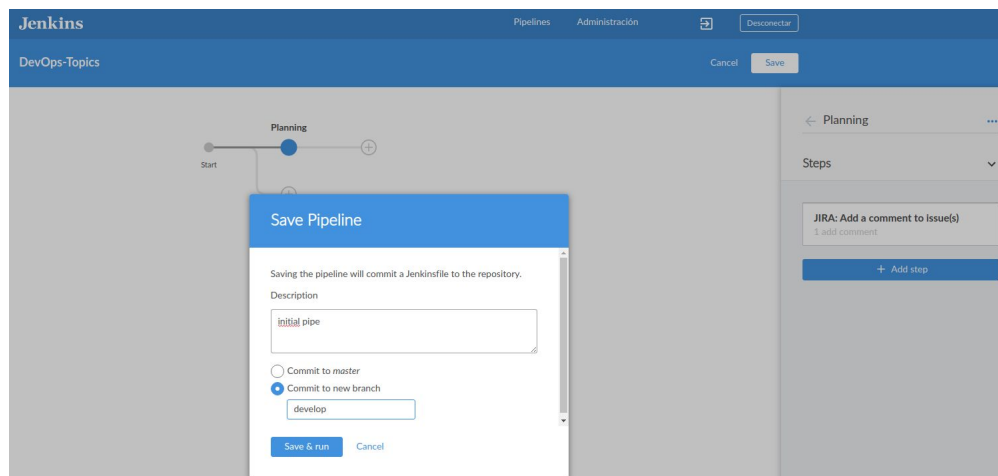
Una vez completado este proceso llegaremos a la siguiente ventana, donde empezaremos con el pipeline



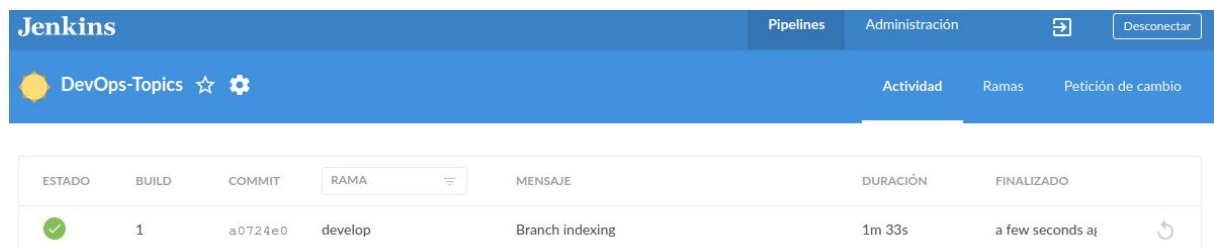
Aquí se crean las etapas



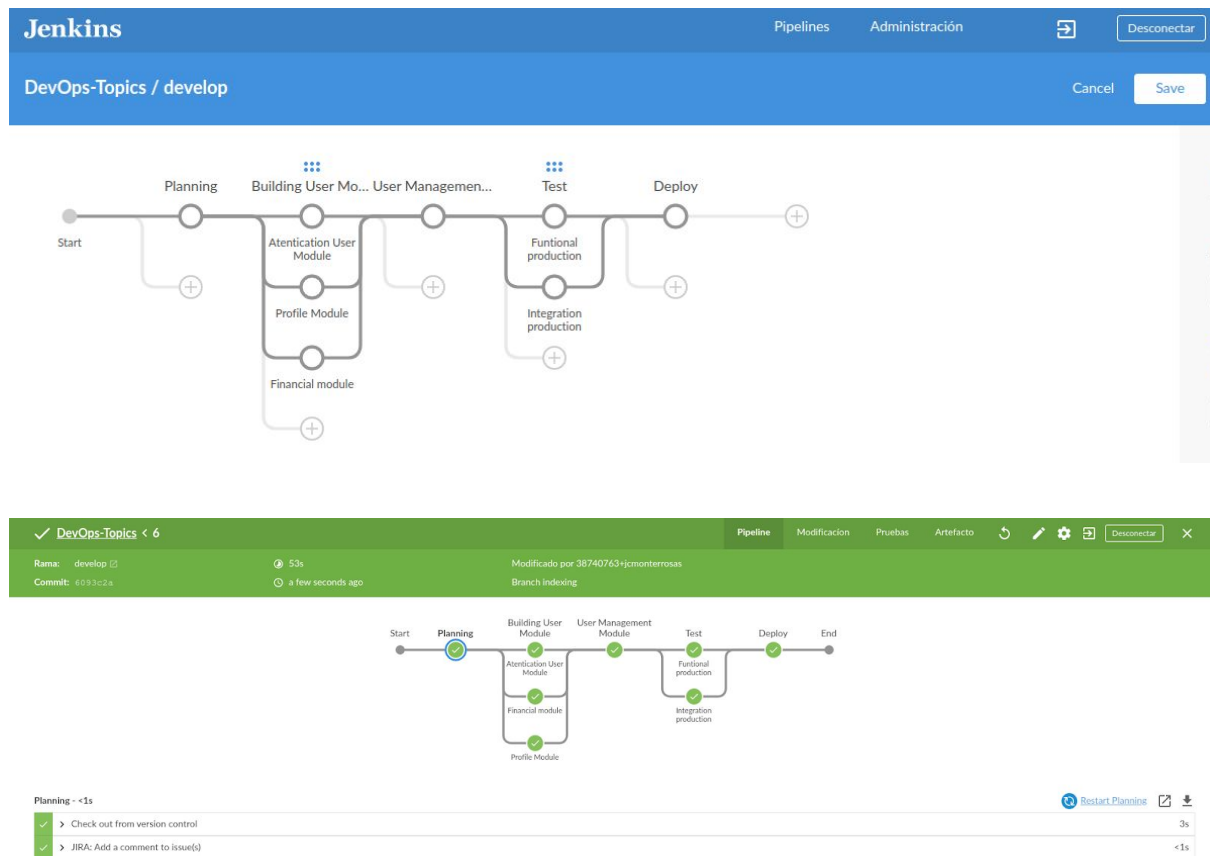
Luego se crean los steps dependiendo de las elecciones.



Una vez terminado el trabajo se realiza el commit.



Nuestro Pipeline después de terminado el trabajo.

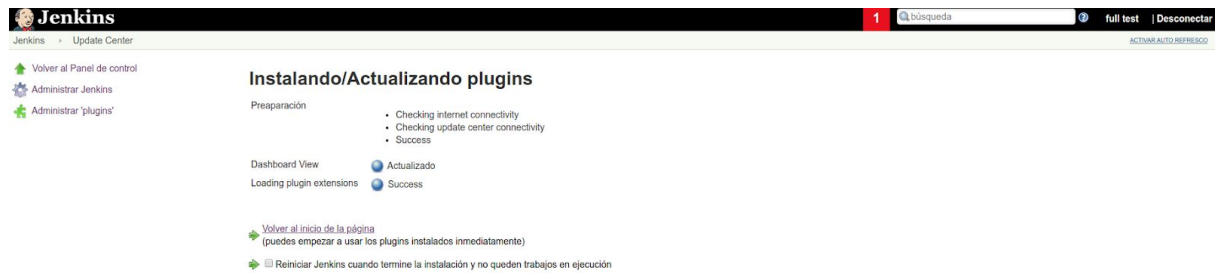


3. Instalación del plugin Dashboard view plugin

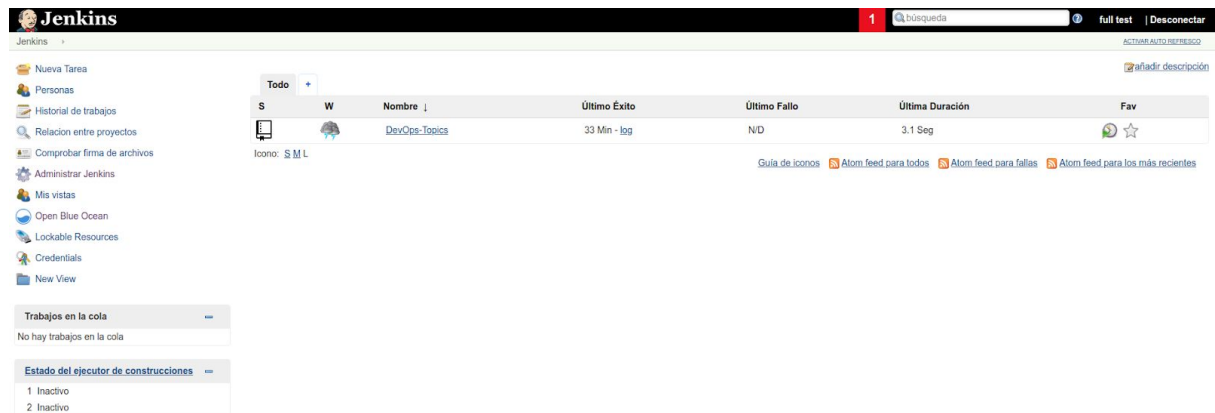
Ingresamos a todos los plugins y buscamos el plugin

The screenshot shows the Jenkins 'Todos los plugins' (All Plugins) page. The page has a search bar at the top with the text 'dash'. Below the search bar, there are tabs for 'Actualizaciones disponibles', 'Todos los plugins', 'Plugins instalados', and 'Configuración avanzada'. The 'Todos los plugins' tab is selected. A list of plugins is displayed, including 'Project statistics', 'Semantic Versioning', 'AppDynamics Dashboard', 'Dashup Portlets', 'Dashboard View', 'Deployment Dashboard', 'Mission Control', 'Pipeline Aggregator', 'New Relic Deployment Notifier', 'New Relic Deployment Notifier', 'Dynatrace Application Monitoring', and 'Environment Dashboard'. The 'Dashboard View' plugin is highlighted with a green checkmark, indicating it is installed. A warning message is visible at the bottom of the list, stating 'Warning: This plugin version may not be safe to use. Please review the following security notices: Missing permission check'.

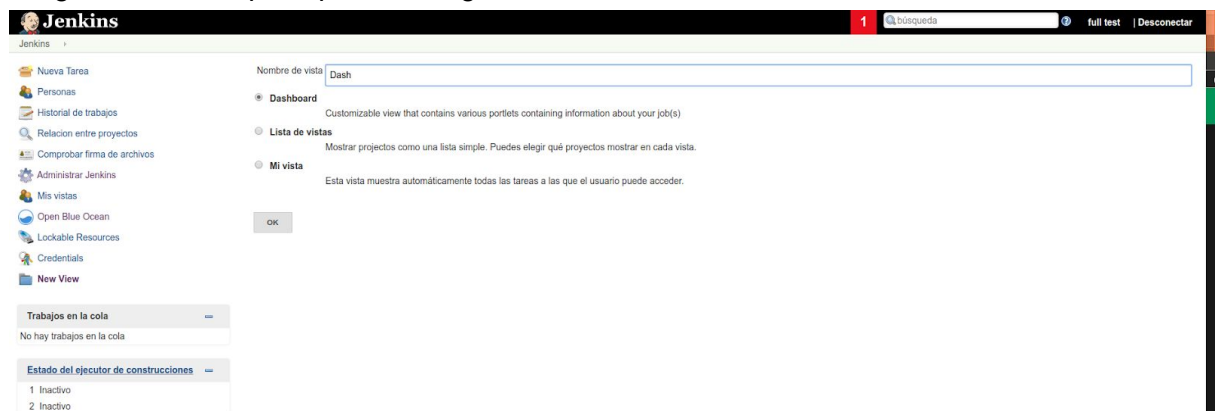
Se realiza el proceso de instalación.



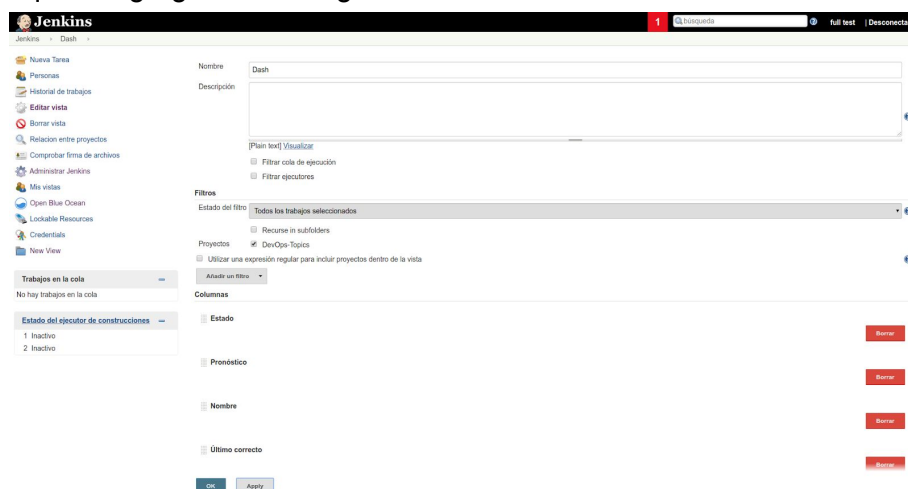
Resultado una vez terminado el proceso



Luego en el menú principal nos dirigimos a new View, seleccionamos dash board



Aquí se agregan las configuraciones básicas de nuestro nuevo tablero.



Aquí seleccionamos la información que mostrará nuestro dash board

Una vez terminado todo esto obtenemos como resultado nuestro tablero.

Job health	Description	Number of jobs
No recent builds failed		0
20-40% of recent builds failed		0
40-60% of recent builds failed		0
60-80% of recent builds failed		0
All recent builds failed		0
Unknown status		0
Total jobs	All jobs	0

Status of the build	Description	Number of builds	Percentage of total builds
Fallido		0	
Inestable		0	
Correcto		0	
Desactivado		0	
Desactivado		0	
Abortado		0	
Sin ejecutar.		0	
Total builds	All builds	0	

Job health	Description	Number of jobs
No recent builds failed		0
20-40% of recent builds failed		0
40-60% of recent builds failed		0
60-80% of recent builds failed		0
All recent builds failed		0
Unknown status		0
Total jobs	All jobs	0

4. Jenkins and Auto DevOps

A primera vista tanto el uso de Jenkins como de la Auto DevOps de gitlab brindan soluciones para el manejo en todas las etapas de un proyecto y las diferencias entre estos parecen ser más en la facilidad de uso dependiendo de la metodología de trabajo que se está usando ya que Gitlab maneja un gran número de herramientas propias que facilitan las etapas desde la planeación, creación y verificación así como los aspectos de seguridad. Sin embargo con Jenkins tenemos una gran variedad de módulos que nos permiten realizar estas tareas.

Aquí tendríamos un beneficio en el uso de Jenkins por la modularidad que nos permite al momento de decidir los módulos que realmente necesita nuestro proyecto.

Por otro lado uno de los aspectos en los que se destaca GitLab es en la seguridad ya que maneja gran variedad en funcionalidades, incluyendo algunas que o es posible trabajar en

Jenkins como lo son soluciones para “Interactive application security testing” o IAST y pruebas de Fuzzing testing. Esta información se resume en el siguiente gráfico.

