

NobleProg

Introducción al Curso

The World's Local Training Provider

NobleProg® Limited 2018
All Rights Reserved

NobleProg - El Proveedor Local de Capacitación del Mundo



13+

years on the market



13+

branches all over the world



550+

trainers



900+

training courses



2200+

companies that trusted us



46000+

satisfied participants

Acerca de tu Instructor

- Mi nombre
- He trabajado con NobleProg por ... años como experto en ...
- Experiencia
- Algunos proyectos previos

Acerca de ti

- Tu nombre
- Tu experiencia
- ¿Cuál es el propósito del curso?
- ¿Ya completaste nuestro Cuestionario Previo?
- ¿Dónde y cómo usarás el conocimiento adquirido?
- Actualmente ¿qué sabes acerca del tema a revisar?
- Cual es, exactamente, la aplicación con la que estás trabajando? (versión, vendor, os, etc...)

Organización del curso

- Lunch
- Horas: 9:00am a 4:00pm, breaks:



- Registro Curso

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
 - ¿Por qué la integración continua?
 - Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- **Probando con Jenkins**
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- **Despliegue con Jenkins**
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Programa del Curso

- Introducción a Jenkins
- ¿Por qué la integración continua?
- Distribución de construcciones en diferentes sistemas.
- Configurando Jenkins
- Instalación de complementos de Jenkins
- Construyendo con Jenkins
- Probando con Jenkins
- Despliegue con Jenkins
- Automatizando con Jenkins
- Asegurando a Jenkins
- Las mejores prácticas para Jenkins
- Informes en Jenkins
- Observaciones finales

Al Final del curso



Training
Evaluation Form



Certificate of
Attendance

Introducción a Jenkins

Software de integración continua

Principales características

- Escrito en Java
- Open Source
- Extensible
- Fácil de usar
- Maduro
- Multiplataforma
- Distribuido



Jenkins

Principales tareas

- Monitoreo constante de los cambios en servidor de control de versiones.
- Compilar el código.
- Ejecutar las pruebas.
- Notificación de errores que se hayan detectado tras las ejecuciones de pruebas, por ejemplo, vía mail, twitter, chat, etc.

Un poco de Historia...

Anteriormente llamado Hudson, el proyecto fue iniciado en 2005 por Sun Microsystems. Con la adquisición de Sun Microsystems por parte de Oracle la marca “Hudson” fue reclamada.

A principios de 2011 la comunidad de Hudson decide crear un “fork” del proyecto con el nombre Jenkins.

NobleProg

¿Por qué integración continua?

Hay un hecho diferencial que ha hecho posible que esta nueva forma de trabajo sea aplicable de forma sencilla: la aparición de Herramientas para Integración Continua.

Definición

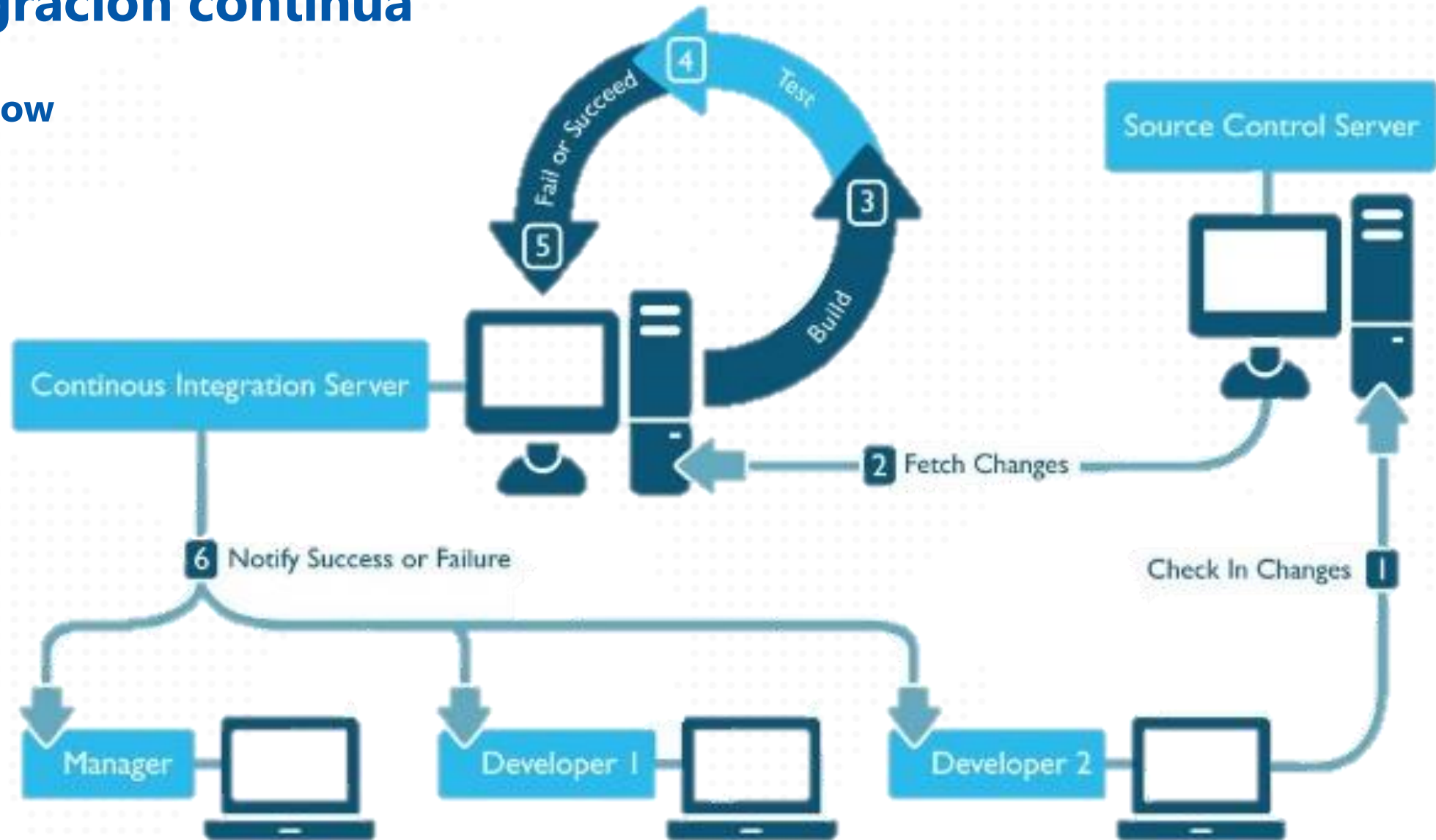
La Integración continua es una práctica de desarrollo descrita por **Martín Fowler** según la cual, en lugar de trabajar de forma independiente, y cada componente del equipo por separado para, y, finalmente, integrar el resultado (con los consiguientes problemas de incompatibilidades que hay que resolver) los equipos de trabajo pueden estar realizando integraciones de forma continua

Principales Ventajas

- La mayor ventaja es evitar el caos de última hora. Un fallo en un sistema acabado no evaluado supone analizar todas las piezas que lo forman. Testeando periódicamente conseguimos señalar fallos en el momento de cometerlos, asegurando el resultado final.
- Disponemos en todo momento de una visión clara de la fase en la que nos encontramos, ya que revisamos cada tarea realizada.
- Posibilidad de testar tanto piezas aisladas, como la integración de varios componentes.
- Disponemos de métricas y datos relativos al proyecto desde el momento en el que comenzamos a evaluar.

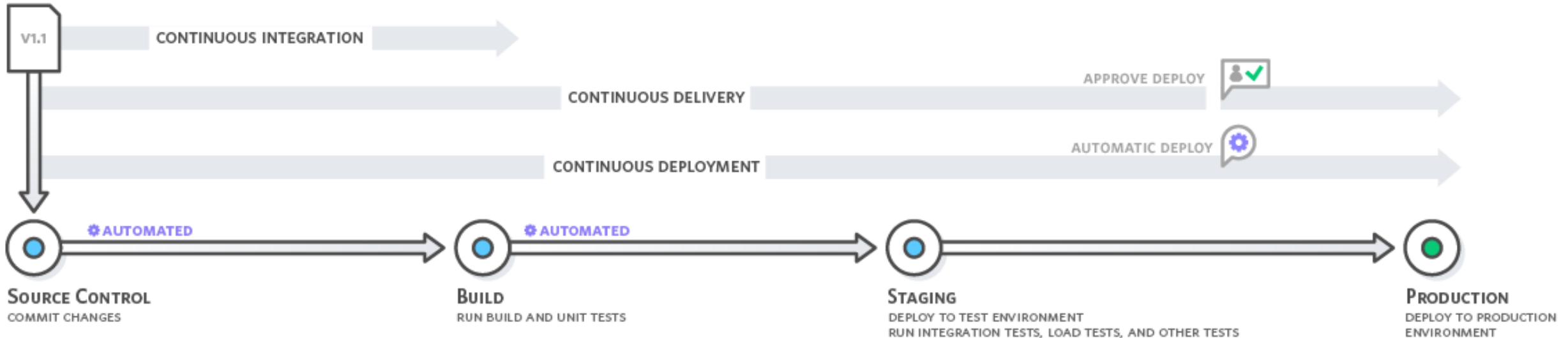
Integración continua

Workflow



Integración continua – Entrega Continua – Despliegue Continuo

Diferencias



Distribución de construcciones en diferentes sistemas

Conexión a través de SSH

Ventajas:

- Conexión segura.
- Gestión del nodo esclavo desde el maestro.
- Actualización automática a nuevas versiones.

Inconvenientes:

- Uso de usuario/contraseña para la conexión.
- Requiere la instalación de un servidor ssh en Windows.

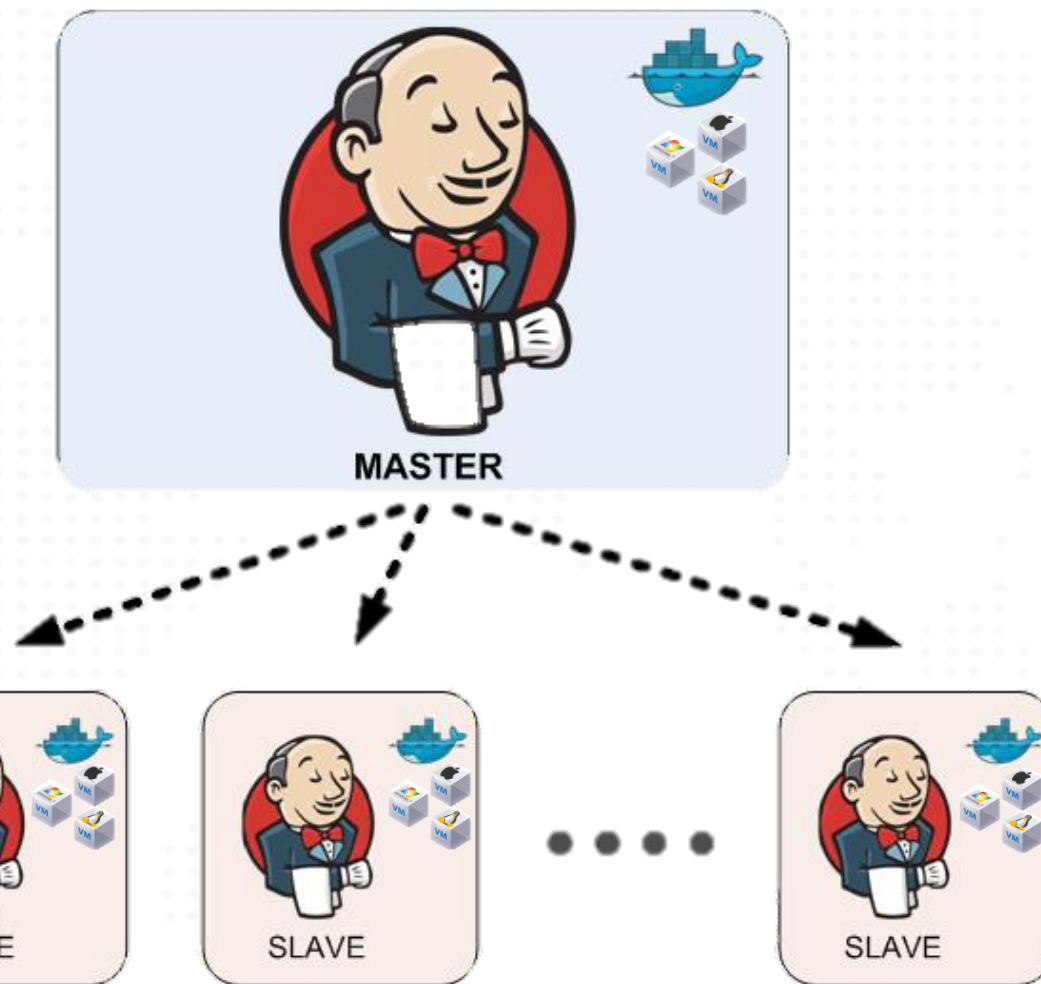
Conexión vía Java Web Start (JNLP)

Ventajas:

- Solución más sencilla de implementar.
- Elimina el tráfico de red necesario para crear/gestionar el servicio desde el nodo maestro al esclavo.

Inconvenientes:

- Requiere que el programa Java Web Start esté arrancado en el nodo esclavo, pero una vez instalado puede ser configurado como servicio automático en el nodo esclavo.
- Cuando se realiza una actualización en el master se debe actualizar manualmente los slaves.



Configurando Jenkins

Instalando plugins

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:

```
/Users/Shared/Jenkins/Home/secrets/initialAdminPassword
```

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

Administrator password

Continue

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 2.32.1

Configurando Jenkins

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	⌛ Build Timeout	⌛ Credentials Binding
⌛ Timestamper	⌛ Workspace Cleanup	⌛ Ant	⌛ Gradle
⌛ Pipeline	⌛ GitHub Branch Source	⌛ Pipeline: GitHub Groovy Libraries	⌛ Pipeline: Stage View
⌛ Git	⌛ Subversion	⌛ SSH Slaves	⌛ Matrix Authorization Strategy
⌛ PAM Authentication	⌛ LDAP	⌛ Email Extension	⌛ Mailer

** Script Security
** Command Agent Launcher

Folders

** bouncycastle API
** Struts
** Pipeline: Step API

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

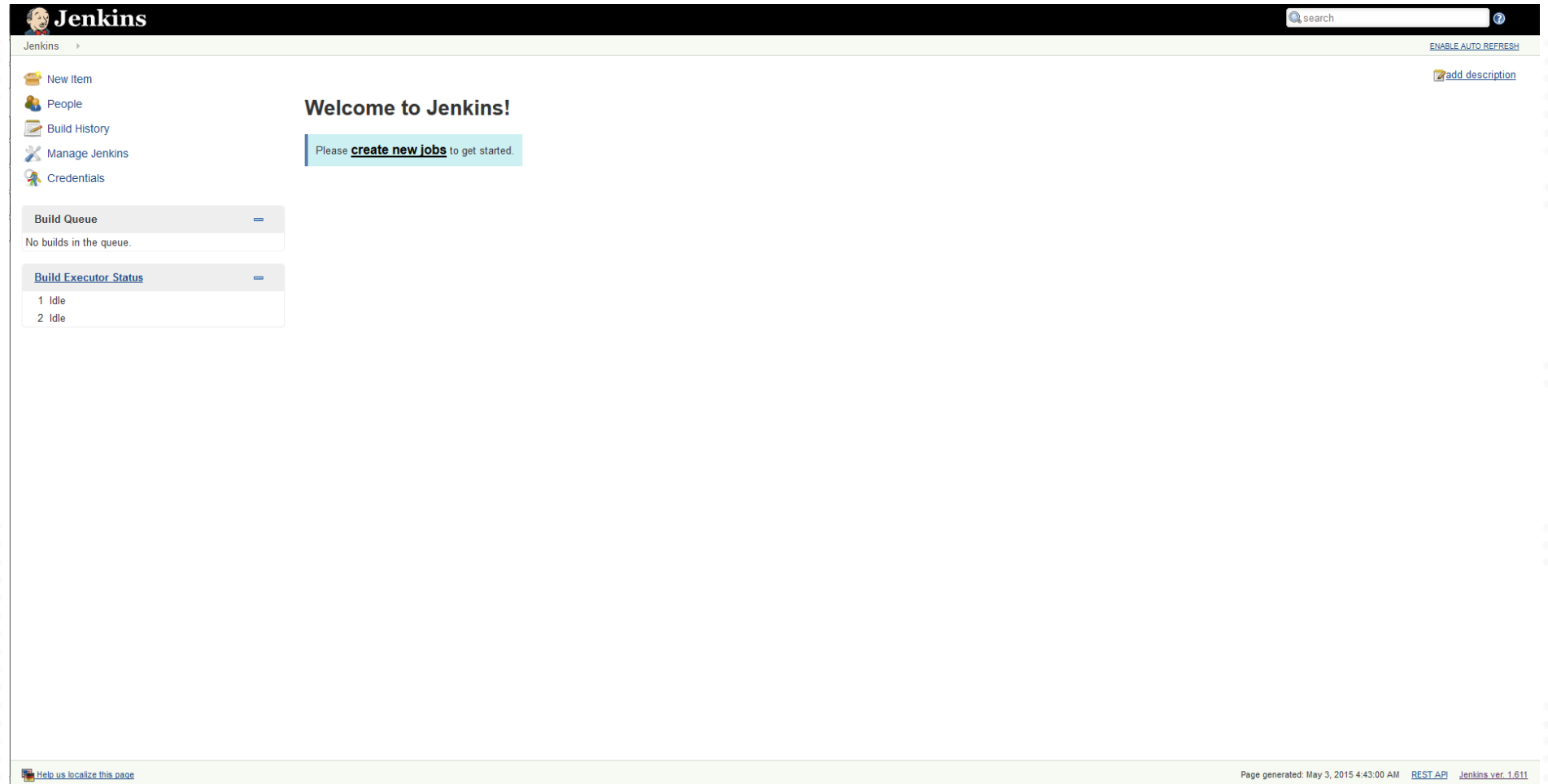
Jenkins 2.32.2

Continue as admin

Save and Finish

Configurando Jenkins

Bienvenido a Jenkins



The screenshot displays the Jenkins web interface. At the top, there is a black header with the Jenkins logo and name. Below the header, a navigation sidebar on the left contains links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials'. The main content area features a 'Welcome to Jenkins!' message with a prompt to 'create new jobs'. Below this, there are two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing two 'Idle' executors). The footer includes a 'Help us localize this page' link, a page generation timestamp, and links to the 'REST API' and 'Jenkins ver. 1.611'.

Jenkins

search

ENABLE AUTO REFRESH

add description

New Item

People

Build History

Manage Jenkins

Credentials

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

Please [create new jobs](#) to get started.

[Help us localize this page](#)

Page generated: May 3, 2015 4:43:00 AM [REST API](#) [Jenkins ver. 1.611](#)


Jenkins - Tareas

Nueva Tarea


- Freestyle project
- Pipeline
- Multi-configuration Project
- Folder
- Multibranch Pipeline

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.




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.

Jenkins Pipelines

Aportan

- Mantenibilidad
- Flexibilidad
- Visibilidad

Ventajas

- Control de versiones
- Evitamos el riesgo de pérdida de código y configuración.
- Reutilización de código, parámetros, visualización, plugins...

Desventajas

- Necesidad de mayores conocimientos técnicos.

SCRIPTED PIPELINE	DESCRIPTED PIPELINE
Modelo de programación imperativo	Modelo de programación declarativo
Entorno de programación con todas las funciones	Sintaxis más simple y pragmática para la creación de Jenkins pipeline
Gran flexibilidad y extensibilidad	Limitado a lo que puede hacer un usuario
Se declara y ejecuta mediante nodos	Se declara y ejecuta mediante el comando pipeline y los diferentes stages, stage, steps y step
No hay muchas limitaciones para usuarios expertos y requerimientos complejos	La mejor opción para el desarrollo de estrategias de CI y CD con Pipelines
En común: <ul style="list-style-type: none">– Usan Groovy– Pueden utilizar librerías compartidas– Pueden utilizar script y shell– Se guardan en un fichero .jenkinsfile	

Jenkins Pipelines

Sintaxis Básica

- **Pipeline {}** Identificamos dónde empieza y termina el pipeline así como los pasos que tiene
- **Agent.** Especificamos cuando se ejecuta el pipeline. Uno de los comandos más utilizados es any, para ejecutar el pipeline siempre y cuando haya un ejecutor libre en Jenkins.
- **Stages.** Bloque donde se definen una serie de estados a realizar dentro del pipeline.
- **Stage.** Bloque que define una serie de tareas realizadas dentro del pipeline, por ejemplo: Build, test, deploy, etc. Podemos utilizar varios plugins en Jenkins para visualizar el estado o el progreso de estos estados.
- **Steps.** Son todos los pasos a realizar dentro de un stage. Podemos definir uno o varios pasos.
- **Step.** Es una tarea simple dentro del pipeline. Fundamentalmente es un paso donde se le dice a Jenkins qué hacer en un momento específico o paso del proceso. Por ejemplo, para ejecutar un comando en shell podemos tener un paso en el que tengamos la línea sh "ls" para mostrar el listado de ficheros de una carpeta.

```
pipeline {  
    agent any  
    stages {  
        stage('Build') {  
            steps {  
                sh 'make'  
            }  
        }  
        stage('Test'){  
            steps {  
                sh 'make check'  
                junit 'reports/**/*.xml'  
            }  
        }  
        stage('Deploy') {  
            steps {  
                sh 'make publish'  
            }  
        }  
    }  
}
```