



Práctica 4 - exploratoria

Dado que los ejercicios que se presentaron en las unidades poseen una característica incremental, lo visto en esta práctica tomará como base lo anterior instalado y configurado. Para ello se instalará una aplicación para automatizar el proceso de Integración Continua y otra para llevar adelante la gestión de configuración de ambientes.

Se instalará el software de aprovisionamiento Puppet el cual está escrito en el lenguaje de programación Ruby. Puppet nos permitirá gestionar las configuraciones y las distintas dependencias de software que tenga nuestro ambiente de una manera explícita, declarando el estado en que queremos que ese encuentre nuestra máquina. Una vez declarado el estado deseado, Puppet se encargará de hacer lo necesario para hacer que esa máquina se encuentre en dicho estado. La principal ventaja de esta herramienta es la simplificación de la administración de un gran volumen de servidores, los cuales pueden estar bajo sistemas operativos Linux o Windows y poseer un detalle de dichos cambios.

Anteriormente hemos instalado el software Vagrant para la gestión de nuestro ambiente virtual. Vagrant soporta el aprovisionamiento de software a través de Puppet en su configuración pero no será nuestro objetivo, sino que lo que queremos es que puedan obtener el conocimiento para comprender el funcionamiento básico de esta herramienta sin la presencia de Vagrant, por lo cual el ejemplo se centrará en la configuración de Puppet sin la integración con Vagrant.

Puppet utiliza archivos de configuración llamados manifiestos (la extensión del archivo es **.pp**). Es bueno aclarar que el contenido de los manifiestos no es ejecutado en un orden secuencial sino que Puppet armará su propia lista de ejecución a partir de las dependencias establecidas.

La arquitectura de Puppet es de cliente/servidor, es decir un equipo actuará como maestro y uno o más agentes estarán instalados en los nodos que requieren el aprovisionamiento. No es lo recomendable pero a fines de realizar un ejemplo práctico, se instalará Puppet como maestro y agente en un mismo entorno. La comunicación que se realiza entre cliente-servidor o agente-maestro se realiza mediante encriptación a través de certificados SSL (el nodo maestro será la autoridad certificante). El tipo de comunicación será push hacia el maestro, es decir que los agentes se comunicarán automáticamente al servidor para buscar novedades de



actualización (por defecto esto ocurrirá cada 30 minutos), de esta manera al tener un nodo central las configuraciones que se realicen serán impactadas en los nodos agentes. Al establecerse la comunicación cada cierto periodo de tiempo es fundamental que el nodo maestro tenga un servicio de configuración automático de su huso horario para evitar errores en las sincronizaciones. Cuando el agente se comunica con el maestro le envía una serie de datos (facts) que se verifican con la configuración establecida como estado deseado para dicho nodo, en el caso de que no coincida el agente se encargará de alcanzar dicho estado, por ejemplo una actualización de un servicio, de manera automática a través de lo declarado en el manifiesto.

La manera de asignar los nodos agentes que poseerán conexión al nodo maestro se realiza mediante nombres de dominio. El ejemplo de este curso se remite a la configuración manual de los nombres de dominio, sin tener que configurar un servidor DNS pero recomiendo que en un ambiente en donde se utilizará este tipo de herramientas se realice mediante un servidor DNS para facilitar la administración de los nodos.

Esta herramienta se provee de manera gratuita y de código abierto y también se puede encontrar una versión paga, Puppet Enterprise, la cual posee más funcionalidades.

Jenkins, es un software de automatización de compilación, de pruebas y de despliegue en ambientes, es decir que automatiza el pipeline de un producto de software. Es una herramienta gratuita y de código abierto, muy utilizada para la automatización de la práctica de Integración Continua y Deployment. Esta herramienta está basada en la gestión de tareas (Jobs) en los cuales se pueden especificar distintas características tales como su orden de ejecución, las condiciones necesarias para que ejecute y la tarea en sí misma. Por otro lado, Jenkins provee una interfaz gráfica que permite visualizar métricas de la salud del código del software que se está automatizando.

Importante: en esta práctica sólo se solicita la instalación de Jenkins y algunos plugins, no habrá ningún tipo de configuración adicional y ejecución de tareas.



Importante

Para cada unidad encontrarán una guía detallada describiendo cada paso necesario para completar las prácticas. Su objetivo es guiar con ejemplos a los alumnos que se encuentren bloqueados con alguna consigna. Así mismo, cada alumno puede tener distintas o ninguna experiencia en el conjunto tecnológico que se brinda en este curso, por lo cual alentamos a que las prácticas se realicen en grupo y que al menos haya algún integrante que cuente con experiencia en desarrollo u operaciones.

Por otro lado, no es necesario entregar ambas prácticas. Pueden optar por la entrega de una de las dos: la guía detallada o la exploratoria. Aún así, si lo desean pueden hacer entrega de ambas.

Prerequisitos:

Conocimientos en:

- Linux
- Git
- Puppet
- Pipeline

Consigna

El objetivo de esta práctica consistirá en configurar Jenkins pipeline en la mayor medida de los posible mediante un Jenkinsfile alojado en un repositorio de código.

Se deberá completar lo siguiente:

- Los archivos de infraestructura deben estar versionados en un branch con el nombre “unidad-4-jenkins”, se debe crear a partir de los últimos cambios del branch “unidad-3-puppet”. Proyecto “utn-devops”.

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**

- Se deberá utilizar Jenkins Pipelines
- Se deberá versionar el archivo Jenkinsfile en el repositorio unidad-4-jenkins
- Se deberán obtener los cambios en un repositorio de aplicación y que jenkins ejecute al menos una prueba unitaria de dicha aplicación. Puede ser al realizar el merge o por pull-request.

Entregables:

- URLs de los 2 repositorios git.
 - Indicar directorio en donde se almacenan los archivos que se utilizarán, manifiestos de Puppet.
- Capturas de pantalla indicadas para mostrar las evidencias, además se debe incluir la visualización del pipeline y la ejecución de un build
- Presentar los entregables en único archivo con el nombre: NOMBRE EQUIPO - Practica 4. **Sólo una entrega por equipo.**

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning