



# Certified Tech Developer

The Ultimate Degree

Infraestructura I

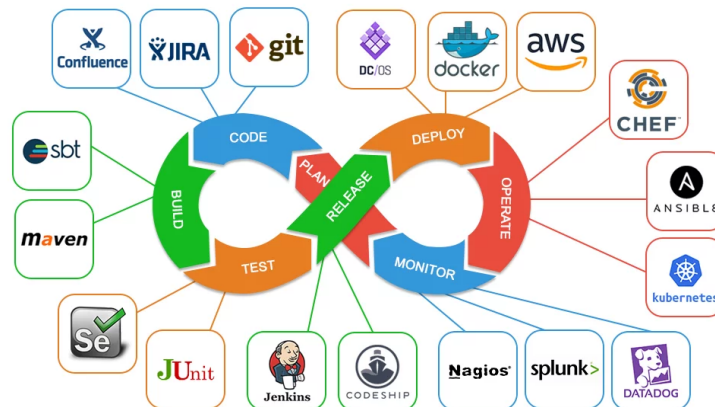
## Configuración y mantenimiento del sistema

Retomando el concepto que vimos:

***“La automatización de la infraestructura consiste en el uso de sistemas de software para crear instrucciones y procesos repetibles a fin de reemplazar o reducir la interacción humana con los sistemas de TI”.***

Este plantea una visión muy amplia, ya que tiene que abarcar los diversos rubros de las empresas que poseen su área de sistema. Según el tipo de organización en la que trabajamos, es el grado de automatización con el que nos vamos a encontrar en la infraestructura de TI.

Uno de los roles del equipo de trabajo del área de sistemas es el **DevOps** engineer, encargado de **automatizar** los procesos del área de **Development** y **Operations**. Su función es hacer más amigable y eficiente la relación entre estos dos equipos de trabajo.



## Infraestructura y servicios

Primeramente tenemos que analizar los dispositivos tenemos en nuestra red de IT. Si contamos con servidores **legacy** o servicios contratados por un **cloud providers**. De las dos opciones anteriores, lo que tenemos que tener en cuenta es el sistema operativo con el que trabajan (MAC, Linux, Windows).

De los cloud providers más conocidos contamos con:

- AWS
- Google Cloud
- Microsoft Azure



Para la elección se evalúan los costos y la oferta de servicios que disponen. Esta última es cada vez más robusta y completa (load balancing, backup, clustering, security, etc.).

## Manejo del código

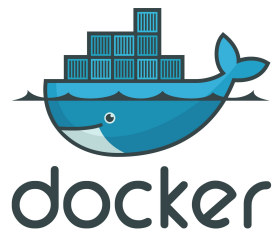
Partimos desde donde alojamos nuestro código de trabajo si somos programadores. Por excelencia el gestor de versionado e integrador de código más utilizado es Git. Los alojamientos de estos repositorios más conocidos en la nube son GitLab, GitHub, Bitbucket, Gitea, entre otros. En estos se almacena nuestro código de las aplicaciones que pueden estar en lenguajes de programación diversos según lo que necesitemos –por ejemplo: Python, JavaScript, Java, .NET, etc.–.

También podemos automatizar el deploy de nuestro código con CI/CD (refiere a las prácticas combinadas de integración continua y entrega continua). Algunas herramientas para hacerlo son:

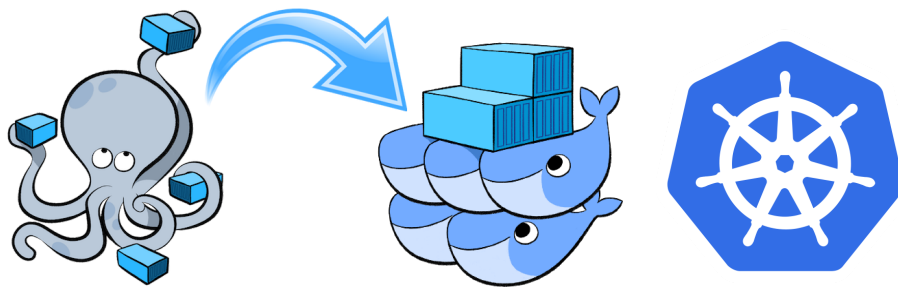
- Jenkins
- GitActions
- JetBrains

## Contenedores

Los contenedores se están convirtiendo en el modelo de empaquetado de software del producto que desarrollamos. Permiten la virtualización de ambientes de trabajo compatibles transportables, totalmente configurados para que nuestro código funcione en todos los equipos. El más popular es Docker.

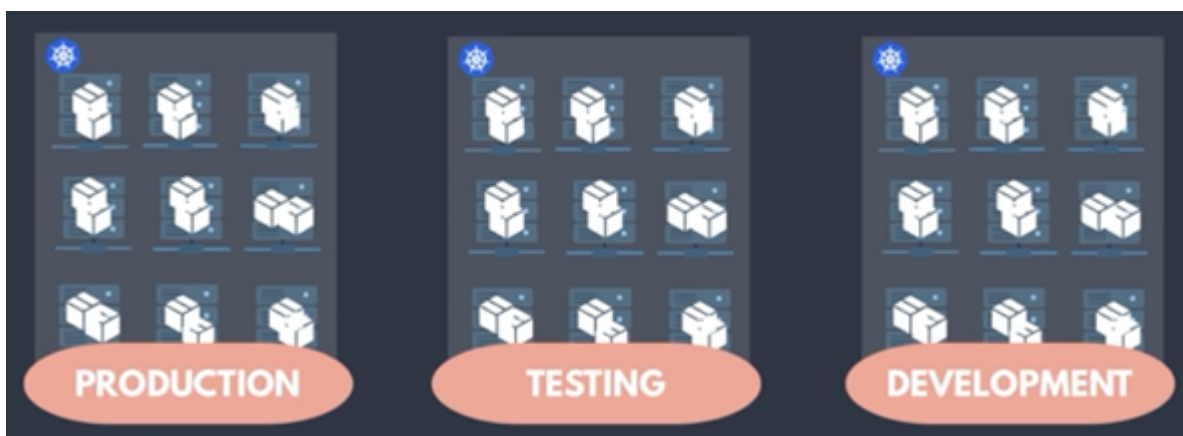


Cuando manejamos muchos contenedores, tenemos que migrar a un clúster de contenedores, dirigidos por los orquestadores de contenedores. Por ejemplo: Kubernetes o Docker Swarm.



## Ambientes de trabajo

Luego, hay que tener en cuenta en qué ambiente está nuestro clúster.



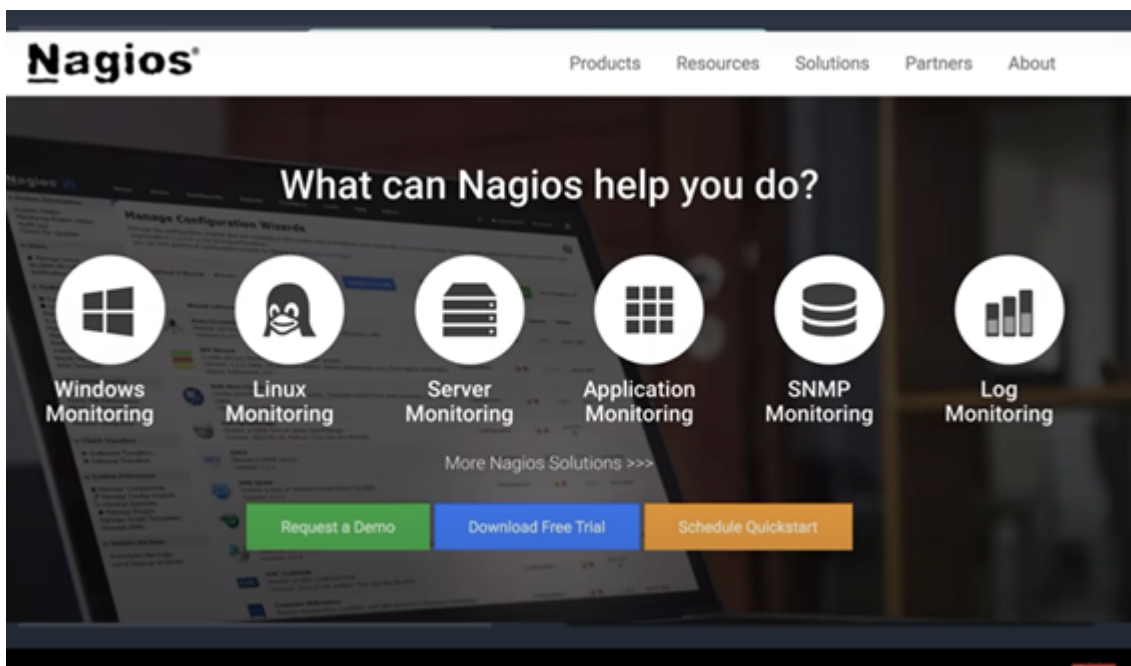
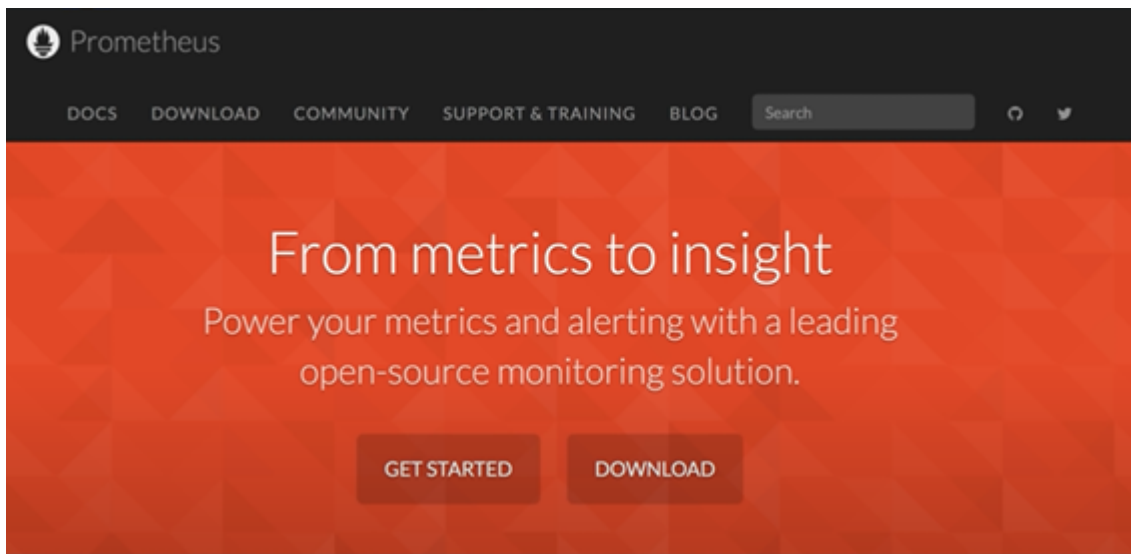
En cada ambiente se trabaja con distintas tecnologías, según el grado de exposición del producto. Automatizando estos procesos vamos a tener más eficiencia, transparencia, facilidad de replicación y recuperación. Los ambientes más populares son:

- Ansible
- Chef
- Puppet
- Terraform



## Monitores de red

Es muy importante tener la supervisión de los dispositivos que hay en nuestra red y de los servicios, y programar alertas en caso de que suceda algún cambio. Para esto podemos utilizar: Nagios, Prometheus, Icinga2 o DataDog.





The image shows a banner for the Icinga website. At the top left is the Icinga logo. To its right is a navigation menu with links: 'Productos', 'Apoyo', 'Comunidad', 'Aprender', 'Blog', and 'Empezar'. The main heading is 'Supervise toda su infraestructura'. Below it is a paragraph of text: 'Encuentre respuestas, tome medidas y conviértase en un solucionador de problemas. Sea flexible y siga sus propios caminos. Mantén la curiosidad, mantente apasionado, mantente al tanto. Afronte su desafío de monitoreo.' At the bottom left are two buttons: 'Hacer un recorrido' and 'Empezar'. On the right side of the banner is a large, abstract, glowing blue circular graphic.

icinga

Productos ▾ Apoyo Comunidad Aprender Blog Empezar

# Supervise toda su infraestructura

Encuentre respuestas, tome medidas y conviértase en un solucionador de problemas. Sea flexible y siga sus propios caminos. Mantén la curiosidad, mantente apasionado, mantente al tanto. Afronte su desafío de monitoreo.

Hacer un recorrido Empezar

## Lenguajes de scripting

Necesitamos trabajar en estrecha colaboración con los desarrolladores y el administrador del sistema para automatizar tareas de los operadores y desarrolladores (como pueden ser backups, cron jobs, system monitoring).

Según el sistema operativo, podemos usar: Bash o PowerShell.



Pero también existen lenguajes de scripting independientes del SO, como Python, Ruby o Go. El más popular es Python, ya que posee muchas librerías y es de fácil lectura y aprendizaje.



## Conclusión

Estas tecnologías expuestas para automatizar la infraestructura TI no son todas las que están en el mercado, sino una visión general de las que hoy son las más populares. Hay que tener en cuenta que cada día se están desarrollando nuevas herramientas y metodologías de trabajo para cada empresa, como también empresas que brindan el servicio de automatización outsourcing de la organización. Siempre hay que tomar la decisión basándose en la necesidad, recursos disponibles y experiencia que cuenta la empresa.