Universidad Nacional Autónoma de México

Facultad de Contaduría y Administración

Monitoreo y administración centralizada en clústeres de HPC.

Diseño de un Sistema o Proyecto

Arlette Karina Guerrero López

México, D.F. **2015**

Universidad Nacional Autónoma de México

Facultad de Contaduría y Administración

Monitoreo y administración centralizada en clústeres de HPC.

Diseño de un Sistema o Proyecto

Que para obtener el título de:
Licenciada en Informática

Presenta:
Arlette Karina Guerrero López

Asesor:
M.I. Lourdes Yolanda Flores Salgado

México, D.F. **2015**

Índice de contenido

Introducción....5
Objetivo del proyecto....6
Objetivos particulares....6
Alcances y limitaciones....7
Capítulo I....9
1.1 Historia....9
1.1 Coordinación de Supercómputo....13
Capítulo II....16
2.1 ¿Qué es Informática?....16
2.2 TICS....16
2.3 ¿Qué es el cómputo científico y cómo se relaciona con la informática?....16
2.4 Arquitectura de Computadoras....17
2.4.1 MIMD....18
2.5 ¿Qué es Supercómputo?....19
2.6 ¿Qué es un clúster?....19
2.6.1 Alto rendimiento....20
2.6.2 Alta disponibilidad....20
2.6.3 Discos....21
2.6.4 Tarjetas de red....22
2.6.5 Fuentes de alimentación....22
2.6.6 Reemplazo en caliente (Hot Swap)....22
2.7 Administración de Sistemas....23
2.8 Administración OSI....24
2.8.1 Administración de fallas....24
2.8.2 Administración de configuración....25
2.8.3 Administración de contabilidad (administración de usuarios)....25
2.8.4 Administración del desempeño....25
2.8.5 Administración de seguridad....26
2.9 Extensión geográfica....28
2.10 SNMP....28
2.10.1 La base de datos (MIB)....29
2.11 CMIP....30
2.12 Gestión de red....31
2.13 Jerárquico....32
Capítulo III....33
3.1 LDAP....33
3.2 Puppet....33
3.2.1 Módulos....35
3.2.2 Manifests....35
3.2.3 Site.pp....36
3.2.4 init.pp....36
3.2.5 Catálogo....36

3.3 Instalación y configuración....38
3.4 Instalación de repositorios....38
3.5 Instalación de Puppet en el nodo maestro....39
3.5.1 Archivo de configuración....40
3.5.2 Host....41
3.5.3 Firewall....42
3.6 Configuración del nodo maestro (Master)....42
3.7 Instalación del Agente....44
3.9 Creación de autoridad certificadora (CA)....44
3.10 Creación de un manifests....47
3.11 Creación de un módulo....48
3.12 Puppet Forge....50
Capítulo IV....55
Conclusiones....55
Comentarios personales....56
Glosario....57
Referencias....59

Dedicatorias

Este libro va dedicado a mi abuelita, que aunque ya no se encuentra con nosotros siempre estuvo a mi lado regañándome, sacando la fuerza que necesitas para seguir adelante.

A mi madre y a mi padre que han estado conmigo y me han apoyado en todo lo que hago.

A mi maestra Yolanda Flores que para mí es como mi mami académica.

A mejor amigo Eduardo que es como un hermano, que siempre ha estado ahí para apoyarme.

Agradecimientos

Agradezco a la Universidad Nacional Autónoma de México UNAM por permitirme ser parte de su alumnado, por enseñarme a ser critica y regalarme los mejores años de mi vida académica.
Agradezco a la Facultad de Contaduría y Administración.
Agradezco a la Coordinación de Supercómputo y a José Luis Gordillo Ruíz.

Introducción

En el siglo XX el crecimiento en la tecnología cambió significativamente, lo hemos visto durante los últimos años, ahora se cuenta con grandes máquinas de cómputo que hacen billones de operaciones por segundo.

Hoy en día se necesita de una gran capacidad de procesamiento para encontrar soluciones a diferentes problemas que en la actualidad son muy comunes, ya sea en el ámbito científico o empresarial. Un ejemplo es el almacenamiento y procesamientos de datos para la toma de decisiones, para esto se necesita de una administración óptima y segura para que no haya pérdida de datos o fuga de información.

En la dirección General de Cómputo y de Tecnologías de Información y Comunicación de la UNAM (DGTIC), la Coordinación de Supercómputo, administra la infraestructura central de la supercomputadora para que científicos utilicen los recursos para el desarrollo y la investigación.

Las supercomputadoras tienen una gran capacidad de cómputo, cuentan con varios microprocesadores que paralelamente trabajan y, su capacidad de almacenamiento, es mayor que una computadora personal.

Las macrocomputadoras o también conocidas como mainframes, son de gran capacidad de procesamiento, grandes y caros.

Un clúster es una estructura a partir de un conjunto de computadoras muy similares entre sí, estas computadoras o nodos, se encuentran conectadas de forma coordinada, con una red de alta velocidad.

Al administrar alguna de estas poderosas máquinas, ya sea supercomputadora, clúster o

mainframes se pueden encontrar varios problemas.

De los problemas más comunes que un clúster puede tener es la replicación de datos, evitando que las configuraciones se encuentren diferentes en cada uno de los nodos. Para evitar estar actualizando cada nodo por separado, es necesario que el nodo maestro tenga la última versión de la configuración y este se replique hacia los demás nodos. Para eso, es necesario utilizar herramientas que nos ayuden a mejorar la administración, optimizando tiempos y automatizando configuraciones.

En la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) actualmente se encuentra la supercomputadora Miztli. Esta supercomputadora se administra de manera manual, replicando los archivos de configuración por medio de utilerías. Un *script* se encarga de replicar los archivos a los nodos, pero si el *script* por alguna razón no se puede conectar al nodo, ésta copia ya no se lleva a cabo y el nodo se queda sin los cambios en la configuración.

Objetivo del proyecto

Desarrollar un procedimiento que permita monitorear y mantener coherencia de archivos en un clúster mediante una administración centralizada utilizando la herramienta puppet.

Objetivos particulares

- Conocer y analizar el funcionamiento de la herramienta Puppet.
- Configurar la herramienta Puppet.
- Desarrollo del procedimiento y pruebas.

Alcances y limitaciones.

La Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC), en la Coordinación de Supercómputo, cuenta con un sistema de cómputo que necesita de una administración óptima, uno de los problemas con los que nos encontramos es la replicación de archivos hacia los demás equipos. Se necesita de una herramienta que automatice esta tarea. Para encontrar la mejor opción, se llevarán a cabo pruebas en un clúster prototipo. El prototipo será un clúster virtual donde se simularán 3 nodos, siendo uno el nodo maestro.

Con el desarrollo de este proyecto se busca, entre otras cosas, aportar documentación en idioma español, ya que la mayoría de la documentación sobre Puppet se encuentra en inglés. Se intenta documentar la investigación que se realizó sobre algunos temas importantes para la administración de clúster, ¿qué es Puppet?, y algunas configuraciones para administrar un clúster.

Capítulo 1

Este capítulo nos habla de la antecedentes de la organización, así como su misión y visión a la cual se le realizará este proyecto.

Capítulo 2

Habla sobre los temas en que la solución se basa y se hace un diagnóstico del problema que se presenta en la organización.

Capítulo 3

En este capítulo se presenta una propuesta de solución al problema, que se necesita y que la organización presenta.

Capítulo 4

En este apartado se mostrará los resultados esperados que se presentan con esta propuesta y las conclusiones.

Capítulo I

Dirección General de Cómputo y de Tecnologías de la Información y Comunicación (DGTIC) .

1.1 Historia.

En 1958 la primera computadora en México se instala en la UNAM, era un sistema IBM 650 la cual se instaló en la planta baja de la Facultad de Ciencias. Las necesidades de varias áreas de la universidad aumentaron y se vieron obligados a crear una entidad que se encargará de administrar los recursos de la computadora.
Esta dependencia de la UNAM comenzó en el año de 1981 creando el Programa Universitario de cómputo (PUC) se inauguro el edificio que actualmente se ocupa, contando con cuatro áreas principales:

- Cómputo para la Docencia.
- Cómputo para la Administración Académica.
- Cómputo para la Investigación .
- Cómputo para la Administración.

Tiempo después, en 1985 se le cambio el nombre a Dirección General de Servicios de Computo Académico (DGSCA), actualmente su nombre es Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC). En 1989 la UNAM comenzó a incorporarse a Internet, la red telefónica con la que se contaba se cambio por conmutadores digitales[1].

En 1991 la UNAM tiene a disposición la supercomputadora para universitarios e investigadores.

1 50 años del Cómputo en México, *El cómputo en México*, Cronología y Aplicaciones, Recuperado de http://caomputo50.unam.mx/computoenmexico2.html

1992 Se inaugura en el área de Telecomunicaciones de la Red Integral.
1999 Como miembro fundador, la UNAM instaura la CUDI, (Corporación Universitaria para el desarrollo de Internet).
2004 Se Inaugura el Observatorio de Visualización Inmersiva (IXTLI).
2006 Inauguración de la Red Inalámbrica Universitaria (RIU)

Misión: La Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC), contribuye al logro de los objetivos de la UNAM como punto de unión de la comunidad universitaria para aprovechar los beneficios que las tecnologías de la información y las comunicaciones pueden aportar a la docencia, la investigación, la difusión de la cultura y la administración universitaria.[2]

Objetivos:

- Proporcionar nuevos programas de capacitación y actualización permanente, considerando la convergencia digital.
- Transformar RedUNAM en una red multimedia integral que integre más a los universitarios, con servicios tales como telefonía, voz sobre IP, videoconferencia y audioconferencia.
- Consolidar la información histórica, presente y futura de la institución, en diversos formatos, dentro de una red de acervos digitales (RedUNAM), sustentada por la más avanzada tecnología de seguridad y resguardo de la información, de tal forma que todo contenido educativo, todo resultado de alguna investigación o todo recurso cultural esté al alcance de los universitarios desde cualquier dispositivo de información digital.
- Posicionar a la UNAM a la vanguardia como una institución digital, lo que requerirá desde la formación de su comunidad en todo este abanico de tecnologías hasta la constante innovación en cómo esos recursos se integrarán a la misión y objetivos de

2 Dirección General de Cómputo y de Tecnologías de Información y Comunicación, *¿Quienes somos?* Recuperado de http://www.tic.unam.mx/mision.html.

la Universidad.

- Apoyar a la institución en la mejora permanente de sus procesos administrativos con la ayuda de tecnologías de información y comunicación que den certidumbre, pertinencia y eficiencia a tales procesos a un costo más reducido.
- Impulsar la participación de la UNAM en la protección al medio ambiente, reduciendo el consumo de papel, impresos, mensajería y transportes, siendo sustituida esta información por recursos digitales.[3]

3 Dirección General de Cómputo y de Tecnologías de Información y Comunicación, Acerca de clústererado de http://www.tic.unam.mx/pdfs/AcuerdoTIC.pdf

Organigrama general de La Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC).

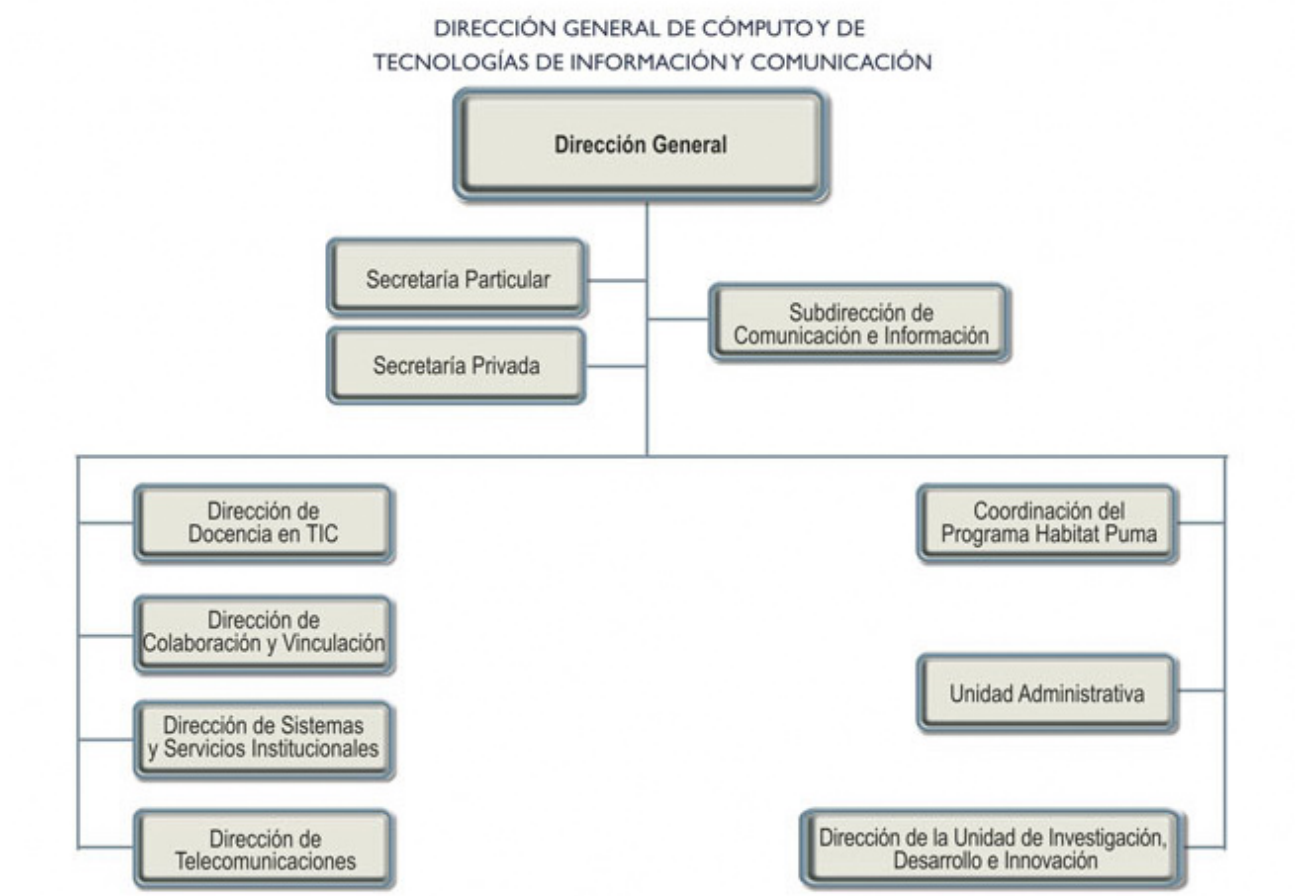


Ilustración 1: Organigrama obtenido de http://www.tic.unam.mx/organigrama.html

1.1 Coordinación de Supercómputo

Anteriormente la Coordinación de Supercómputo comenzó como Departamento de Supercómputo.

En la UNAM el supercómputo comenzó en 1991, en este año se puso en funcionamiento la Cray Y/MP/464, primera supercomputadora de América Latina. Conforme ha pasando el tiempo la tecnología a avanzando, también lo han hecho las supercomputadoras.
En 1997 la Cray YMP, se cambió por la Cray SGI Origin 2000 (Berenice), esta supercomputadora contaba con 40 procesadores y 10 GB de memoria.
En 2003 llegó HP AlphaServer SC45 (Bakliz) con 32 procesadores, 32 GB de memoria y 1,000 GB de memoria.
En 2007 llegó la HP Cluster Platform 4000, conocida como KanBalam con 1,368 núcleos de procesamiento, 3,016 GB de memoria y 160,000 GB de almacenamiento.

Actualmente se cuenta con Miztli, una HP Cluster Platform 3000 SL, llegó en el 2013 con 5,312 núcleos, 23,000 GB de memoria y 750,00 GB de almacenamiento.

Tabla 1: Supercomputadoras en la UNAM

	Sirio	Berenice	Bakliz	KanBalam	Miztli
marca	CRAY	SGI	HP	HP	HP
procesador	Vectorial	R10000	Alpha EV67	Opteron Dual Core	Intel E2670 8 cores
número de procesadores	4	40	32	1,368	5,312
rendimiento numérico (GFlops)	1.02	15.6	80	7,113	118,000
memoria (Gigabytes)	0.512	10	32	3,016	23,000
almacenamiento (Gigabytes)	19	170	1,000	160,000	750,000

La definición que en DGTIC se le da al supercómputo es:

“El supercómputo se refiere a la utilización de computadoras con capacidades excepcionales, para la realización de investigaciones. La UNAM pone a disposición de la comunidad universitaria equipos de supercómputo, así como servicios de asesoría y ayuda para su mejor aprovechamiento.”[4]

4Supercómputo. (2009 - 2014). *¿Qué es Supercómputo?* . Febrero-12-2014, de Universidad Nacional Autónoma de México DGTIC Sitio web: http://www.super.unam.mx/index.php/home/enlace/quesc

Organigrama de la Coordinación de Supercómputo

Coordinación de Supercómputo

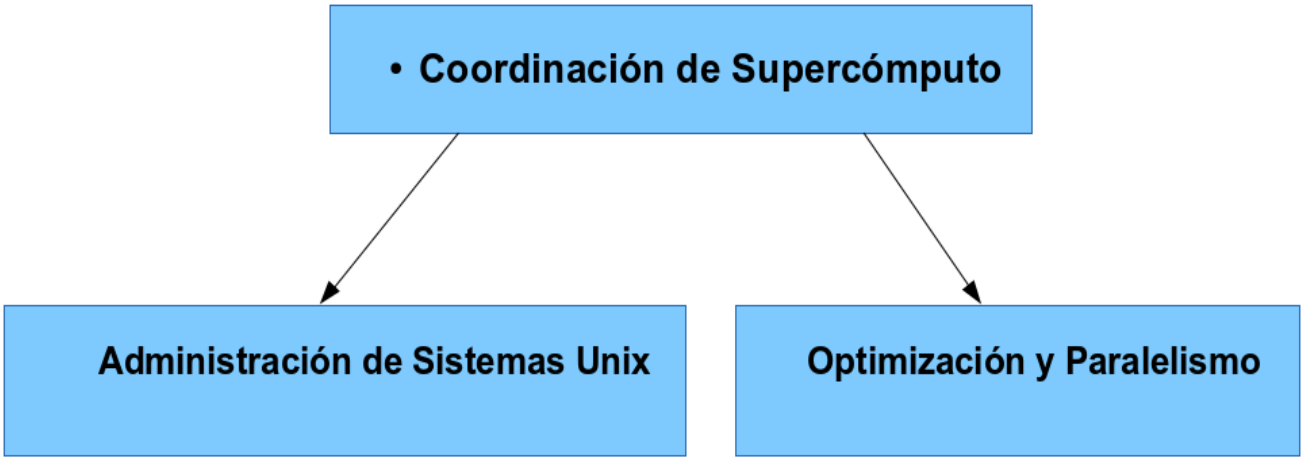


Ilustración 2: Organigrama de la Coordinación de Supercómputo, autoría propia.

Capítulo II

2.1 ¿Qué es Informática?

Es la ciencia donde se utiliza la tecnología para automatizar la información con el fin de implementar soluciones o toma de decisiones, a partir del procesamiento de un conjunto de datos.

El término es acrónimo de "Información automática", que significa: todo aquello que tiene relación con el procesamiento de datos, utilizando las computadoras o los equipos de procesamiento automático de información.[5]

Cada dato (número, letras, etc) que se recopila, por si sólo no tiene ningún valor, se necesita que los datos sean procesados por un sistema de cómputo para que tengan sentido, y con base a la información que se obtuvo se podrá tomar decisiones para el beneficio de la organización u empresa.

2.2 TICS

Las TICS que sus siglas significan Tecnologías de la Información y la Comunicación son herramientas y programas que nos ayudan a administrar y compartir información mediante la tecnología informática como las telecomunicaciones, Internet etc.

2.3 ¿Qué es el cómputo científico y cómo se relaciona con la informática?

Hoy en día el uso de las supercomputadoras se destinan para varias aplicaciones, las cuales, nos ayudan a solucionar problemas de tipo científico implementando modelos matemáticos y de cómputo, haciendo simulaciones para conocer los resultados de esos modelos. Por ejemplo para análisis de sismos, modelos económicos, la cura de alguna enfermedad, etc.

5 Gonzalo Ferreyra Cortés. (2006). *Informática paso a paso*. México: Alfaomega

2.4 Arquitectura de Computadoras

Desde que comenzaron a fabricar computadoras con válvulas, la tecnología ha ido evolucionando, el primer microprocesador fue de 4 bits, se introdujo en el mercado en 1971, después de algunos años IBM lanzó su computadora personal (PC).

Por los años ochenta llegaron los microprocesadores de 16 bits, en 1985 son incorporados los microprocesadores de 32 bits. Después, la microelectrónica comenzó con microprocesadores de 32 bits con proceso en paralelo.

La palabra arquitectura, puede tener varios significados, sobre todo en el área de la informática.

En los años sesenta, definieron la arquitectura de computadoras como:

Cualquier otra arquitectura, es el arte de determinar las necesidades del usuario de una estructura y de diseñar para satisfacerlas lo más eficazmente posible, dentro de las restricciones económicas y tecnológicas[6]

Los años pasan y los diseños de las arquitecturas de las computadoras van mejorando, depende mucho de las aplicaciones y las herramientas necesarias para que estas puedan desempeñarse de la mejor manera, actualmente podemos definir arquitectura como:

La arquitectura de computadoras contempla las técnicas de interconexión y compatibilidad de equipos, así como portabilidad de software, y estudia el diseño modular en previsión de ampliación de un Sistema Informático, así como su rentabilidad y costes de explotación[7]

Existen arquitecturas avanzadas que se centran en procesos paralelos, su rendimiento es más elevado, su hardware y software son específicos para ser usados en la solución de problemas que necesitan de un mayor procesamiento,

En 1972 Michael J. Flynn propone una clasificación de las computadoras. Se basa en el número de instrucciones y la secuencia de datos que la computadora necesita para procesar.

La clasificación consta de 4 tipos de computadoras

---

6 E. Alcalde/F. Ormaechea J.Portillo/F. Garcia Merayo. (1991). *Arquitectura de computadoras*. España: MCGraw-Hill.
7 E. Alcalde/F. Ormaechea J.Portillo/F. Garcia Merayo. (1991). *Arquitectura de computadoras*. España: MCGraw-Hill.

- SISD Single Instruction Single Data
- SIMD: Single Instruction Multiple Data
- MIMD : Multiple Instruction Multiple Data
- MISD : Multiple Instruction Single Data

De acuerdo a la temática de este trabajo, sólo se hablará a detalle de MIMD.

2.4.1 MIMD

Este sistema tiene varios procesadores, la misma secuencia de instrucciones la ejecutan pero teniendo diferentes datos, es decir, cada procesador puede ejecutar un programa con datos diferentes.

Sistema de memoria compartida

En este tipo de sistema, cada procesador tiene comunicación y puede entrar a la memoria principal, pueden tener acceso de manera simultánea porque acceden por un sólo canal. Esto puede ocasionar problemas porque pueden tener acceso al mismo tiempo lo cual desencadena errores.

Sistema de memoria distribuida

A diferencia de el sistema de memoria compartida, en este sistema sus procesadores comparten información enviando mensajes. Si el procesador 1 necesita datos del procesador 2, el procesador 1 le envía un mensaje al procesador 2 para solicitarlos. A este envío de mensajes se le conoce como paso de mensajes.[8]

8 José Castillo C.& Ricardo Benjamín O. (2006). *Implementación de un clúster OpenMosix para cómputo científico en el Instituto de Ingeniería*. Abril-22-14, de Licencia: Creative Commons Sitio web: http://www.josecc.net/archivos/tesis/tesis_html1/node6.html

2.5 ¿Qué es Supercómputo?

Una supercomputadora es una computadora que tiene una capacidad de procesamiento mucho mayor a lo que comúnmente conocemos, tiene varios procesadores que trabajan en paralelo. También tiene una gran capacidad de almacenamiento, memoria y velocidad de comunicación, esto es para que se puedan hacer billones de instrucciones por segundo, mejorando los tiempos en que una investigación puede llevarse con una máquina menos poderosa.

El uso de una supercomputadora es para fines científicos y de investigación, donde se procesa una gran cantidad de datos o se necesita realizar cálculos matemáticos en el menor tiempo posible. Por esto, es necesario que las velocidades de estas computadoras sean mayores.

Para conseguir esto se necesita de técnicas especiales, en sus circuitos, sus algoritmos y lenguajes de programación. Otra forma de conseguir que la velocidad sea mayor, es usando la ejecución concurrente o la ejecución simultánea.

2.6 ¿Qué es un clúster?

Un clúster está conformado por un grupo de computadoras que se encuentran interconectados. A cada uno de las computadoras se les llama nodos y se ubican en un espacio exclusivo montados en un *rack*. Cuentan con uno o varios procesadores, los cuales son configurados de forma coordinada para simular que es un sólo recurso. Este tipo de computadoras se encuentran en un cuarto grande, con equipos de enfriamiento para evitar que se sobre calienten.

En un clúster se puede encontrar:

Nodo Maestro: Este nodo planifica las tareas, el espacio de almacenamiento y al usuario le da acceso a los recursos de cómputo.

Nodos de Cómputo: Nodos asignados para realizar cálculos.

Nodo de Servicio: Provee servicios básicos para el clúster, como autenticación, balanceo de carga, planificación de tareas, etc.
Nodo de I/O o servidor de archivos: Da acceso a los recursos de almacenamiento de las aplicaciones o usuarios.

- Alto rendimiento (High Perfomance)
- Alta disponibilidad (High Availability)

2.6.1 Alto rendimiento.

El objetivo de esta clasificación, como su nombre lo dice, es mejorar el rendimiento para la obtención de un resultado sobre un problema optimizando tiempos. Suele darse para cálculos matemáticos, compilación de programas y ejecución de los mismos.
Para tener un alto rendimiento es necesario que se cree un clúster con varios nodos, cada nodo se encarga de realizar tareas distribuyendo la carga de trabajo según sus características, configuración, equipos, etc.
Una característica importante es que evita que la máquina se sature, por ejemplo, en un servicio web a veces se llega a saturar porque varios usuarios acceden al servicio, si los usuarios que acceden llega a aumentar, el servicio podría fallar. Una solución es que la carga se trabajo sea distribuido en varias máquinas (servidores). A esto se le conoce como balance de carga.

2.6.2 Alta disponibilidad.

Hoy en día con los servicios que se ofrecen en la red, es necesario que tengan una disponibilidad alta para evitar pérdidas de datos o de dinero teniendo en cuenta que puede haber una falla, ya sea de hardware o software, en caso de que llegará a suceder, el usuario no debe de percatarse de que existe o existió algún error en el funcionamiento.

Para que exista una disponibilidad alta es necesario que se configure con base al hardware o software del sistema, un ejemplo es el detectar fallas de hardware, las fuentes de alimentación proporcionan electricidad y existe la probabilidad de que deje de proporcionar corriente, si esto sucediera automáticamente debería entrar otra fuente de alimentación para que el sistema no deje de dar servicio.

Actualmente para que un sistema funcione de la mejor manera se necesita de varios componentes, pero, entre más componentes las posibilidades de que falle es mayor.
La redundancia depende de la importancia del servicio que se está brindando y las pérdidas que pueda ocasionar al no tener disponible los datos almacenados.
Los componentes redundantes son los discos, fuentes de alimentación, tarjetas de red etc.

2.6.3 Discos

Dispositivo donde se graban datos. Es común que en un servidor algún disco duro falle, si el servidor tuviera sólo un disco duro todo lo que se encuentra almacenado se perdería.
Existen técnicas para minimizar éste tipo de problemas y para que el servidor siga funcionando sin dejar de dar servicio.
Una técnica común es la llamada RAID (Redundant Array of Independent Disks). Un arreglo (*array*) de RAID es un grupo de discos que actúan como almacenamiento simulando ser uno sólo soportando el fallo de algún disco sin perder información.
RAID utiliza *striping* que es el dividir la información antes de distribuirla en bloques que almacena en diferentes discos para evitar la pérdida de datos. Existen diferentes niveles RAID que tienen características diferentes en rendimiento y en replicar los datos. Por ejemplo el RAID 10 es una combinación de RAID 0 y 1, nos brinda velocidad y tolerancia a fallos creando una copia idéntica de datos en 1 o varios discos.

2.6.4 Tarjetas de red.

Permite que los servidores se encuentren conectados a la red para que tengan comunicación con el exterior. Por eso es común que cuenten mínimo con 2 tarjetas de red por si alguno llegaría a fallar el otro entraría en función.
Una técnica común es el *bonding*. Consiste en que varias tarjetas de red funcionen simulando ser una, asignándoles la misma dirección ip.

2.6.5 Fuentes de alimentación.

Sen encargan de proporcionarle electricidad al servidor. Generalmente cuentan con 2 o más fuentes de alimentación que se encuentran conectados a diferentes sistemas eléctricos garantizando el suministro de electricidad tanto al equipo como al sistema de enfriamiento para evitar que los equipos se calienten demasiado.

2.6.6 Reemplazo en caliente (*Hot Swap*).

El reemplazo en caliente se utiliza para cambiar configuraciones, reparar o sustituir componentes del sistema sin llegar a interrumpir el servicio.
Permite que se intercambie o eliminen componentes que tengan alguna falla mientras sigue activo y operando.

Para evitar pérdidas los equipos deben de estar diseñados con redundancia, en caso de que exista algún fallo, otras partes del sistema harían su trabajo, mientras el componente dañado se retira y se sustituye por otro en buen estado. Por ejemplo un arreglo de discos RAID permite que algún disco que tenga alguna falla se intercambie por uno en buen estado.
Uno de los inconvenientes es que se necesita de personal especializado para insertar o eliminar componentes del sistema para minimizar daños.

2.7 Administración de Sistemas.

Las organizaciones actualmente necesitan de los sistemas de información para mejorar la eficiencia de sus operaciones. Con el paso del tiempo, se han convertido en algo esencial para la competitividad de la organización.

Para mantener esta área de las organizaciones, es necesario un administrador de sistemas responsable de que el sistema funcione de manera correcta, llevando a cabo los cambios necesarios como configurar, actualizar, asegurar de que personas externas a la organización no tenga acceso, entre otras cosas.

Para ello el administrador debe tener conocimientos en diversos aspectos como seguridad informática, telecomunicaciones, programación, conocimientos en hardware hasta conocimientos en aspectos administrativos.

Actualmente existen varios sistemas operativos, por la temática de este trabajo nos enfocaremos al sistema UNIX.

El administrador de sistemas UNIX será el “superusuario” del sistema, esto es, el usuario que tiene privilegios y permisos para hacer modificaciones en el sistema, así podrá configurar , mantener y actualizar el sistema.

Un administrador de sistemas debe considerar varias actividades:

- Planear las actividades: Definir el procedimiento que debe realizar en cada una, considerando las consecuencias.
- Registrar sus actividades: Las actividades que el administrador lleve a cabo debe registrarlas para hacer constancia de los cambios que realizó en la configuración.
- Realizar copias de seguridad: Antes de hacer alguna modificación al sistema, se debe de hacer un respaldo para evitar pérdidas en caso de que la nueva configuración no funcione como se espera.
- Entendimiento de la documentación del sistema: En los sistemas UNIX, no existe una estandarización, dependiendo de las necesidades de cada empresa es como se llega

a tener una configuración. Un ejemplo es la instalación de programas en el sistema operativo UNIX, depende de las características físicas del equipo. Por eso a veces es necesario realizar consultas a libros, manuales y páginas web especializados en el tema.

- Conocimiento del hardware del sistema: Se debe conocer como se encuentra el equipo físicamente, que tiene conectado, su capacidad de almacenamiento, memoria, velocidad, localización, etc.

2.8 Administración OSI.

La Organización Internacional de Normalización (ISO) creó un marco de gestión de red FCAPS.

Los objetivos de trabajo tienen 5 categorías, donde se define la gestión de redes.

2.8.1 Administración de fallas.

La función de esta administración es la detección de fallas asegurando la disponibilidad de un sistema distribuido y de los servicios que presta.

Identificar las fallas y llevar un seguimiento es un problema donde se presenta en los sistemas de procesamiento, por eso es importante la detección de fallas y la solución

Las tareas que son necesarias para este tipo de administración son:

- Monitoreo de la red y del sistema.
- Atención y respuesta ante una falla .
- Diagnosticar la causa de la falla.
- Evaluar las medidas para recuperarse de los errores.
- Darle asistencia a usuarios.

2.8.2 Administración de configuración.

Configurar equipos implica varias cosas como instalar o actualizar software, hacer cambios en la topología de red etc. Las configuraciones vienen con procedimientos para hacer cambios dependiendo de las necesidades. Se incluyen procesos controlados, ajustes de parámetros (configuración) como seleccionar ciertas funciones de autorización y de conexión, etc.

En esta administración los cambios son necesarios porque se definen valores y se establecen filtros y parámetros.

2.8.3 Administración de contabilidad (administración de usuarios).

La administración de contabilidad se encarga de recopilar datos de uso como asignación de cuentas, bitácoras de contabilidad, estadísticas de uso, recursos a usar, servicios y políticas de contabilidad, basándose en el monitoreo y medición para que se generen cargos a los usuarios.

Las políticas de la organización influye para decidir como se establecerá el sistema de contabilidad, cómo serán distribuidos los costos y que enfoque tendrán para recopilar los parámetros de contabilidad.

2.8.4 Administración del desempeño.

La administración del desempeño se puede ver como una continuación de la administración de fallas, busca que el sistema tenga un buen comportamiento, definiendo la calidad de servicio.

Las medidas que son necesarias para que la calidad del servicio sea de un nivel alto incluye:

- Establecer parámetros y métricas de calidad de servicio.
- Monitorear los recursos para detectar problemas como cuellos de botella y evitar que

el desempeño sea menor.

- Realizar medidas para prevenir fallas.
- Evaluar bitácoras.
- Elaborar reportes de desempeño.
- Planificar el desempeño y capacidad. Nos ayuda a conocer como se pueden comportar con nuevas configuraciones o aplicaciones proporcionando modelos para verificar resultados con estos nuevos cambios.

2.8.5 Administración de seguridad.

Este tipo de administración tiene que ver con la administración de la seguridad en los sistemas distribuidos. Los servicios, información e infraestructura se encuentran expuestos a amenazas o, a que lo usen de una manera inadecuada. Por eso es necesario prevenir pérdidas de datos estableciendo medidas que prevengan o enfrenten las amenazas que un sistema de red puede llegar a tener.

La mayoría de las amenazas son:

- Ataques pasivos: A través de un análisis al tráfico de la red (*sniffers*), se puede robar información.
- Ataques activos: Acceso no autorizado, suplantación de usuarios, modificación de mensajes, reconfiguración no autorizada, *spoofing*, etc.

Se debe definir qué políticas de seguridad son necesarias para diminuir la probabilidad de que alguien no autorizado tenga acceso. Para mejorar la seguridad, también se debe de tener en cuenta:

- Analizar las amenazas y disminuir la probabilidad de que alguien no autorizado tenga acceso a la información .
- Definir e implementar políticas de seguridad.
- Garantizar confidencialidad.

- Verificar identidad.
- Integridad en los datos.
- Monitoreo de los sistemas.

2.8.6 Administración de Redes.

Con el paso del tiempo y el avance tecnológico , Internet se ha convertido en una herramienta esencial para el mundo como para las organizaciones y para la sociedad.

Muchos equipos se encuentran bajo el sistema operativo UNIX o UNIX Like, ofreciendo un servicio que desde cualquier parte del mundo se podría tener acceso con tan sólo una computadora con Internet. Para esto se necesita de una persona que se encargue de administrar los recursos que son necesarios para el buen funcionamiento de la infraestructura.

La infraestructura que se tiene para administrar el servicio de red es específica, depende de las necesidades de la empresa, como cuántos equipos necesita que tengan servicio de Internet o el número de equipos que sólo deben tener servicio de red interna.

Una red es un conjunto de dispositivos que se encuentran interconectados entre ellos a través de un medio que permite que intercambien información y/o compartan recursos.

Para poder determinar que topología es la mejor opción, se necesita de conocer el número y tipo de equipos que se encontrarán interconectados, tipo de red a utilizar, tipo de cableado y conectores. El administrador debe conocer como se encuentran interconectados los equipos para asegurar el buen funcionamiento de la red, su seguridad y su mantenimiento.

Algunas topologías son:

Bus: Esta topología utiliza un sólo segmento de red (*backbone*), los *hosts* quedan conectados, si alguno llegará a fallar, la red seguiría funcionando. Si el bus tuviera algún problema, en este caso, si dejaría de trabajar.

Estrella: En esta topología todos los nodos se encuentran conectadas a un central, por lo

regular un *switch*. Si algún nodo llegará a fallar, los demás nodos seguirían funcionando, pero si el nodo central llegará a fallar, la red dejaría de funcionar.

2.9 Extensión geográfica.

Considerando el tamaño de el lugar donde se encontrarán los equipos y cuántos se encontrarán interconectados se debe tomar en cuenta ésta clasificación:

LAN (*Local Area Network*): Es una red de alta velocidad que cubre una área pequeña. Un edificio, un avión o habitación etc.

MAN (*Metropolitan Area Network*): Área que agrupa las áreas LAN's, provee velocidades altas para la comunicación de una ciudad o campus.

WAN (*Wide Area Network*): Este tipo de red abarca grandes áreas empleando medios de comunicación que no son habituales, como fibra óptica, satélites, etc.

2.10 SNMP.

Sus siglas en ingles *Simple Network Management Protocol*, este protocolo se encuentra en la capa de aplicación, nos facilita el intercambio de información entre los dispositivos que se encuentren conectados a la red.

Cada equipo que implemente el SNMP de manera predeterminada utilizará el puerto 161 (UDP) para envío y recepción de solicitudes, el puerto 162 para la recepción (*trap*) de los dispositivos administrados. Algunos proveedores permiten que esta configuración se pueda cambiar.

Los equipos que se encuentran conectados a la red ejecutan procesos (agentes) para que se administre tanto local como remotamente.

Los procesos van actualizando la base de batos (MIB) conforme los resultados del análisis.

Define la comunicación, el formato y el significado de los mensajes de un administrador con un agente. Lo único que no se define son las variables, el MIB tiene diferentes grupos de variables para cada dispositivo de hardware o protocolo. El MIB contiene el grupo al cual el SNMP puede acceder.

Se compone por:

- Estación de gestión (administración).
- Agente (*agent*).
- Base de datos de gestión .
- Protocolo de gestión de red .

La estación de gestión es la interfaz del administrador de red. Mantiene la base de datos con formato SMI. Contiene el software de gestión.

El agente utiliza el protocolo SNMP envía, recibe y analiza (pasando a través de un *parser*) mensajes. Interactúa con el dispositivo obteniendo información para responderle al NMS y enviar mensajes de notificación (*trap*).

El Protocolo de gestión de red (NMS) es un *host* que envía peticiones y recibe respuestas (SNMP) y mensajes de notificación (*trap*) desde los nodos gestionados.

2.10.1 La base de datos (MIB).

Se encuentra los datos y el histórico de toda la información que se a obtenido de la MIB.

Los MIB se encuentra organizado en niveles, a su vez se lo hace en módulos que donde contiene los grupos de variables.

Existen 5 tipos de mensajes que se intercambian entre el agente y los administradores:

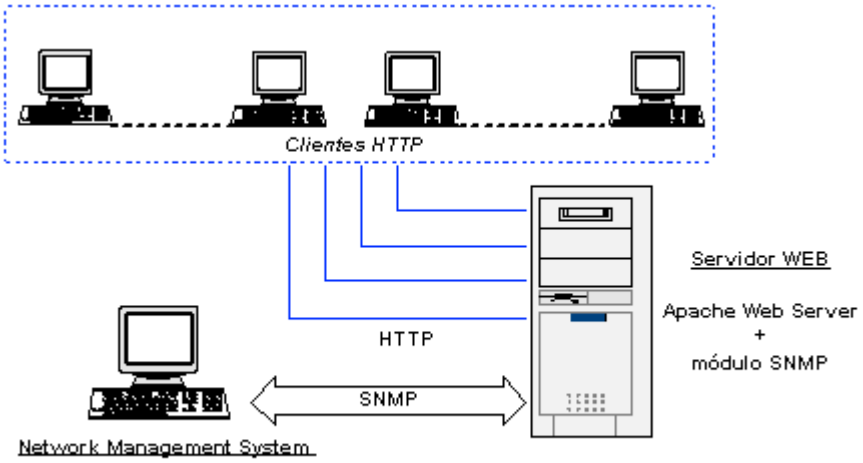
- `GetRequest`: Es una petición que se le hace al Agente para que envíe los valores que contiene el MIB (base de datos).

- `Get Next Request`: Es una petición que se le hace al Agente para que envíe los valores que contiene el MIB referente al objeto siguiente especificado anteriormente.
- `Get Response`: Respuesta del Agente a las peticiones que se hicieron anteriormente.
- `Set Request`: Petición que se le envía a Agente para que cambie el valor en el MIB referente a un determinado objeto.
- `Trap`: Mensaje que es enviado del Agente al Administrador cuando se detecta un suceso inusual. Por ejemplo una conexión o desconexión.

La segunda versión del protocolo `snmp` tiene mejoras en seguridad y transferencia de datos.

- `GetBulk` Es una mejora de `Get Next`
- `Inform`: Es similar al `Trap`, envía un mensaje de que se detectó algo inusual.

Protocolo SNMP



2.11 CMIP.

Sus siglas en ingles *Common Management Information Protocol*, este protocolo proporciona mecanismos de intercambio de información, su sistema de manejo de redes esta muy bien diseñado mejorando los defectos del protocolo SNMP ya que su funcionamiento esta basado en el protocolo CMIS.

Este protocolo se encuentra orientado a conexión, aportando mayor confiabilidad. Su arquitectura se basa en administrador-agente.

Integra dos grupos de servicio:

- Servicios de notificación: Permite a los agentes informar a los gestores de sucesos especiales.
- Servicios de operación: Existen 6 servicios, M-GET, M-SET, M-ACTION, M-CREATE, M-DELETE, M-CANCELE-GET. Se usan por el gestor para invocar operaciones a los agentes y los resultados sean devueltos a los gestores.

2.12 Gestión de red.

Un sistema de gestión de red es una herramienta que permite planificar, organizar, controlar y monitorear la red. Esta diseñado para que se vea la red entera con etiquetas y sus direcciones. Las estaciones de gestión y los equipos se enlazan por el protocolo `snmp`.

Es un entorno de ejecución y por lo regularmente también de desarrollo, que ofrece servicios a las aplicaciones de gestión.

Se garantiza que el nivel del servicio sea alto y adecuado para las necesidades de acuerdo al giro de la organización.

Una plataforma consta de 4 elementos:

- Una API DE Gestión, utiliza las aplicaciones de gestión no importando el protocolo de gestión del recurso que se gestiona.
- Un sistema de gestión de Información: Una base de datos donde accedan las aplicaciones.
- Una interfaz de usuario: Para las utilidades y aplicaciones.
- Conjunto de aplicaciones propias: Ofrezcan una funcionalidad básica de gestión de red.

2.13 Jerárquico.

El *backbone* también llamado núcleo de la red, se encarga de transportar los datos de forma fiable y rápida.

Un fallo en el núcleo de la red afectaría a todos los usuarios que se encuentran conectados en esta red.

Cuando se diseña una red, se debe de tener en cuenta que elementos nos ayudan a minimizar los fallos o errores que un núcleo puede tener.

La red de distribución se asocia con el filtrado y comunicación con el núcleo de la red.

Por lo regular en este tipo de de redes se utilizan 2 switches.

Capítulo III

3.1 LDAP.

Sus siglas en inglés *Lightweight Directory Access Protocol* (Protocolo Ligero de Acceso a Directorios) es un conjunto de protocolos que se encuentran abiertos para que se pueda tener acceso a información que se tiene guardada. La información la ordena en modo jerárquico usando directorios.

En varios casos LDAP es usado como directorio, permitiendo que los usuarios tengan acceso a información de otros usuarios. LDAP Es capaz de propagar su consulta a otros servidores LDAP

LDAP es un sistema cliente/servidor. Cuando un cliente LDAP se conecta a un servidor LDAP puede consultar o modificar un directorio. En una consulta, el servidor puede puede contestar o dirigir la consulta a un servidor LDAP que tenga la respuesta. Antes de que se realice alguna modificación, primero verifica que el usuario tenga permiso, si tiene permiso, se efectúa el cambio.

La principal utilidad de un directorio LDAP es autentificar a las personas para que tengan servicios de un sistema.

3.2 Puppet.

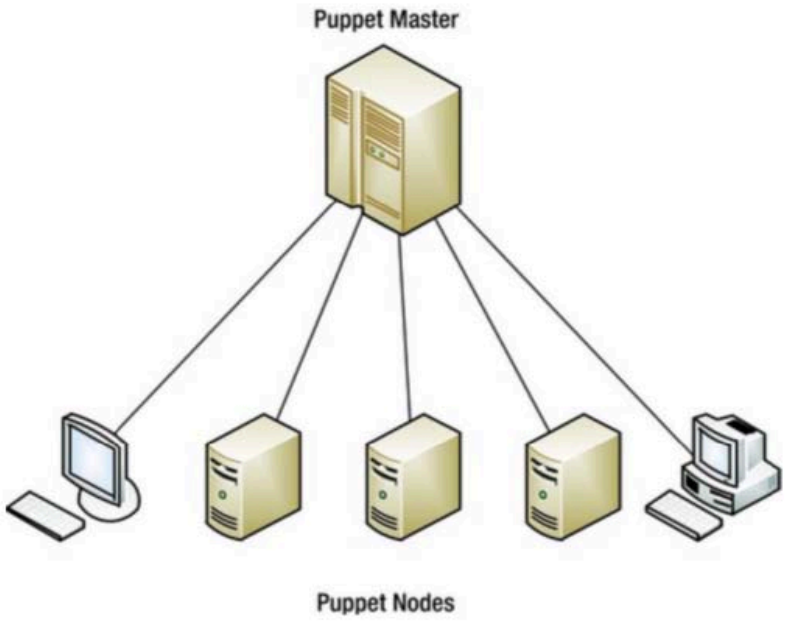
Un administrador de sistemas necesita automatizar tareas que son rutinarias y necesarias para su buen funcionamiento.

Puppet es una *framework open source* para gestionar la configuración y automatización de los sistemas. Permite construir y compartir herramientas que eviten que se duplique soluciones para el mismo problema.

Existen dos versiones; una libre y la Enterprise. La versión Enterprise es una versión de pago.

Puppet esta basado en Ruby, con un lenguaje declarativo, su arquitectura es de cliente/servidor donde el servidor (*puppetmaster*) centraliza las configuraciones. Cada cliente contacta al servidor en un tiempo determinado, descargando la última configuración y si hubo cambios la actualiza. Esta herramienta es idempotente, lo que significa que Puppet sólo hará cambios si así se le pide no importando el número de veces que una regla se ejecuta. El cliente envía un informe al servidor indicando si realizó algún cambio o no.

El servidor maestro (puppetmaster) ejecuta un demonio en el *host* que contiene información necesaria para el entorno específico. Los agentes se conectan al servidor (puppetmaster) a través de una conexión cifrada y autenticación mediante SSL estándar recuperando o sustituyendo cualquier configuración anterior. Hace uso de certificados, puppetmaster es una pequeña entidad certificadora.



Un grupo de `recursos` puede organizarse en clases que son las unidades más grandes de configuración. Un `recurso` puede describir un `archivo` o `paquetes`.

Una clase puede describir lo necesario para que se configure un servicio o una aplicación.(archivos de configuración, demonios de servicios, etc).

Una clase pequeña pueden combinar clases mas grandes que describen las funciones del sistema.

Esta herramienta aunque tenga una serie de pasos no necesariamente las aplica al sistema en el orden en el que se encuentran. Esto es porque puppet asume que los recursos no están relacionados entre sí.

Existen directorios que son importantes para los archivos de configuración y de clases.

3.2.1 Módulos

Los módulos son directorios que contienen archivos como recursos, clases, definiciones y archivos.

Un módulo contiene lo necesario para que una aplicación se configure.

Por ejemplo, almacena el recurso de instalación del paquete de Apache, el archivo de configuración, el servicio de Apache, formando el módulo Apache el cual se puede asignar a los nodos.

Cada módulo tiene una estructura de directorio específica. Con esta estructura Puppet carga automáticamente los módulos.

La ruta de un módulo es una serie de directorios que Puppet comprueba para que la carga sea de manera automática. La ruta que Puppet tiene que seguir, se encuentra en el archivo de configuración, sección maestro (*Master*), en el ModulePath.

3.2.2 Manifests

Un manifests se encuentra dentro de un Módulo. Este directorio contiene archivos con terminación .pp

3.2.3 Site.pp

El archivo `site.pp` se encuentra dentro del directorio `manifests`. El nodo maestro siempre lee el directorio `manifest`, llamado "*site manifest*" o `site.pp`

Este archivo contiene información sobre la configuración que van a tener los nodos, te dice que nodo y que configuración se va a cargar. La encontramos bajo el directorio `/etc/puppet`.

3.2.4 init.pp

Archivo que contiene la definición de clases.

3.2.5 Catálogo

Los `manifiestos (manifest)` pueden contener sentencias, funciones etc, para llegar a un estado deseado.
Los `manifests` consiguen compilar en un documento llamado `catálogo`.
Un `catálogo` contiene `recursos`, describiendo un estado deseado para el sistema.
Cada `agente` ve su propia información, por seguridad ningún nodo puede ver el `catálogo` de otro.

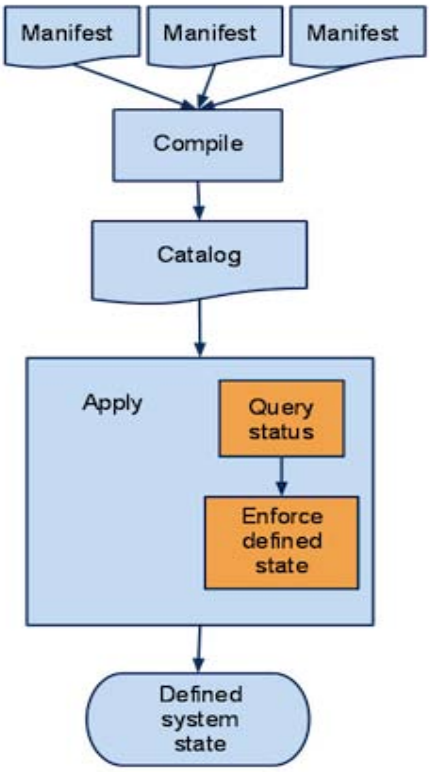


Ilustración 3: Diagrama de cómo funciona un catálogo.

En este trabajo Puppet se usará en un `clúster` prototipo, con 2 `nodos` que serán los `agentes` y un `nodo` que será el maestro. Puppet tiene la capacidad para trabajar con infraestructuras más grandes, con `nodos` que pasen los 100, es importante aclarar porque algunos programas no se instalan de la misma manera con un grupo menor a 100 `nodos`. Sin embargo hay entornos donde, aun cuando sean muchos `nodos`, se puede usar el mismo método ya sea para instalar programas o configuraciones.

3.3 Instalación y configuración.

Antes de instalar Puppet, se debe de tomar en cuenta que Puppet necesita de Ruby, ya que su código está basado en este lenguaje.
Se puede instalar desde binario pero no es recomendable porque se tiene que actualizar y desinstalar, llevarlo a cabo en varios equipos puede ser tardado.

El sistema operativo donde Puppet "trabaja" mejor es GNU/Linux, en este sistema, los archivos de configuración son almacenados en /etc/, Puppet hace uso de esta ruta para almacenar sus archivos propios de configuración. En un sistema Windows, Puppet puede ser instalado pero no podrá trabajar como Maestro.

3.4 Instalación de repositorios.

Antes de instalar Puppet, es necesario instalar repositorios, se recomiendan de dos tipos. El primero se llama `EPEL`, a sido creado por la comunidad de Fedora para crear, mantener y gestionar un conjunto de paquetes de alta calidad para Enterprise Linux, incluyendo a Red Hat Enterprise linux (RHEL), CentOS y Scientific Linux. El segundo es de los repositorios de Puppet Labs.

Para este diseño se usará CentOS, Puppet Open source y los repositorios de PuppetLabs. Los repositorios de PuppetLabs se mantienen en constante actualización por los desarrolladores.

Para instalarlos se usará el comando rpm, donde se especifica la ruta de descarga del .rpm o archivo de paquete. El RPM es un administrador de paquetes utilizado principalmente por las distribuciones linux basadas en Red Hat.
El comando indica qué vamos a descargar y necesitamos para nuestro PuppetLabs.

```
#rpm -ivh http://yum.puppetlabs.com/el/6/products/i386/puppetlabs-release-6-7.noarch.rpm
```

El Maestro (Master) es el único que debe de tener instalado un repositorio, ya sea EPEL o PuppetLabs.

3.5 Instalación de Puppet en el nodo maestro

El paquete Facter contiene un sistema de herramientas de inventario que recopila información sobre los nodos como nombre del host, dirección IP, sistema operativo, versión, etc. .
Para instalarlo se usará:

```
# yum install puppet puppet-server facter
```

El paquete Puppet contiene el agente, el paquete puppet-server contiene el maestro y el paquete facter contiene la herramienta Facter.
Con este comando también se instala la herramienta Hiera.
Hiera es una herramienta de búsqueda para los datos de configuración que se construyen para Puppet. Esta herramienta permite establecer datos de un nodo específico sin necesidad de repetir lo mismo.

Otra forma de instalar Puppet y Facter es a tráves de RubyGems. La diferencia a la anterior es que aquí se instala Ruby y RubyGems.

El Puppet Master contiene los datos de configuración que encontramos en la siguiente ruta:

```
/etc/puppet/
```

El servidor (Puppet Master) ejecuta un demonio en el *host* que contiene información necesaria para el entorno específico. Los agentes se conectan al nodo maestro (Puppet Master) a través de una conexión cifrada y autenticación mediante SSL estándar, recuperando o sustituyendo cualquier configuración anterior.

3.5.1 Archivo de configuración

El archivo de configuración se encuentra dividido en 4 secciones, cada sección configura un elemento en particular.

Las secciones son:

Sección	Función
Agent	Es utilizado por el servicio del agente.
Master	Configura el binario de Puppet.
Main	Esta parte es para la configuración global utilizado por todos los comandos y servicios.

Los datos de configuración se encuentran en un directorio llamado manifests. Dentro de este directorio encontraremos:

Archivo	Función
Recursos:	Elementos de configuración individuales.
Archivos:	Archivos físicos que los agentes pueden utilizar.
Plantillas:	Archivos de plantilla que pueden utilizar.
Nodos:	Especifica la configuración de cada agente.
Clases:	Colecciones de recursos.
Definiciones:	Colecciones compuestas de recursos.

Al instalar puppet, se crea automáticamente el archivo de configuración, en caso de que no se tuviera el archivo, se crea con el comando:

```
$ cd /etc/puppet/
$ puppet master --genconfig > puppet.conf
```

En caso de que no exista la sección principal en el archivo `puppet.conf`, se crea:

```
[main]
server = nombre.servidor.com
server = puppet.puppet.com
```

En la parte server, sustituye por el nombre del dominio del anfitrión.

Se recomienda crear un DNS CNAME para el puppet master.

Un DNS Sistema de nombres de dominio (Domain Name System) identifica el nombre del dominio con su dirección IP. A veces varios nombres de dominios resuelven la misma IP, aquí entra el CNAME, es una base de datos donde se registra sus sus alias enlazando con su nombre de dominio auténtico.

3.5.2 Host

Para instalar Puppet, tendremos que hacer varias cosas:
Se agrega en `/etc/host` o DNS.

```
# /etc/hosts
192.168.1.1 puppet puppetmaster
```

3.5.3 Firewall

EL Puppet Master hace uso del puerto TCP 8140, para el *firewall*, se necesitará tener este puerto abierto y tener una regla en el maestro. El cliente deberá ser capaz de conectar con el maestro.

Para definir esta configuración, se usará `iptables`.

Este comando, `iptables`, es un programa que viene en un módulo que Linux utiliza (`netfilter`). Este comando nos ayuda a definir reglas de filtrado de paquetes, estableciendo qué paquetes acepta y qué paquetes rechaza.

```
$ iptables -A INPUT -p tcp -m state --state NEW --dport 8140 -j
ACCEPT save
```

Se debe de tener en cuenta que desde cualquier lugar se puede tener acceso desde el puerto TCP 8140 al Master

Para restringir el acceso desde la subred 192.168.1.0/2 al puerto 8140.

```
$ iptables -A INPUT -p tcp -m state --state NEW -s 192.168.1.1/24
--dport 8140 -j ACCEPT
```

Para guardar las reglas que se definieron en iptables se ejecuta:

```
#service iptables save
```

3.6 Configuración del nodo maestro (Master).

Editamos el archivo de configuración que se encuentra en la siguiente ruta

```
/etc/puppet/puppet.conf
```

Agregamos las siguientes líneas:

```
server=puppet
```

El agente debe de ser capaz de resolver el nombre del maestro (Master) para conectarse, para eso agregamos esta línea donde le decimos que nombre llevará el maestro con el que tendrá contacto.

En el archivo de configuración en la sección [master] se agrega:
Nota: Sí la sección [master] no existe en el archivo de configuración, se agrega.

```
dns_alt_names = puppetmaster,puppet
```

En esta línea se pone los nombres por el cual se puede conocer al maestro.

Otra línea que se debe agregar

```
certname=puppet
```

Estamos indicando como se llama la máquina del nodo maestro.

También se agrega:

```
autosign=false
```

Con esta línea indicamos que el maestro no debe firmar las peticiones de certificados que le lleguen, así evitamos que firme peticiones que puedan ser agentes externos (intrusos).
Agregamos también:

```
strict_hostname_checking=true
```

Esta línea hará que Puppet use el nombre completo de los nodos, si coinciden, llevará a cabo los cambios.
En Puppet se puede hacer uso de expresiones regulares como nombres de nodos, con este método se declararía una vez el nombre del nodo que coincide con varios nodos.

Activar el servicio de Puppet

Para que el servicio de Puppet se active en CentOS sería:

```
# service puppetmaster start
```

3.7 Instalación del Agente.

Para instalar el agente necesitaremos de Facter. Se ejecutará este comando:

```
# yum install facter puppet
```

3.8 Configuración del nodo Agente.

Se edita el archivo de configuración `/etc/puppet/puppet.conf` del agente y se añade en la sección [main] lo siguiente:

```
server=puppet
```

Esta línea se agrega el nombre del servidor con el que tendrá contacto.

Iniciamos el agente

```
# service puppet start
```

3.9 Creación de autoridad certificadora (CA).

Una autoridad certificadora (CA) se encarga de expedir certificados de identidad cuando un usuario lo solicita.

La autoridad certificadora verifica la identidad del usuario que esta solicitando, generando el certificado correspondiente firmando con su clave privada.

Se necesitará de una autoridad certificadora local (CA) para crear los certificados y claves para el maestro. La información SSL de Puppet y los certificados se encuentran en el directorio:

```
/var/lib/puppet/ssl
```

En el nodo maestro se encuentra la autoridad certificadora(CA), las solicitudes de certificados de sus clientes, un certificado por su maestro y certificados para todos su clientes.
El agente crea una solicitud de firma de certificado y una clave privada. Se envía la solicitud de certificado al maestro esperando que sea firmado y le devuelvan el certificado. Ya que tenga el certificado firmado, volverá a ejecutar el registro de entrada y ejecutar Puppet.

Se recomienda iniciar puppet desde línea de comandos para depurar y ver si hubo algún problema.
Iniciamos el demonio:

```
# puppet master --verbose –no-daemonize
```

Podemos conectar el agente con el maestro con el siguiente comando:

```
#puppet agent --test --server=nombre_servidor
#puppet agent --test --server=192.168.1.1
```

La opción `server` especifica el nombre o dirección del maestro al cual se debe conectar.

Para completar la conexión se tiene que firmar el certificado que el agente envió al maestro.
Para ver los certificados pendientes que se tienen que firmar se ejecuta:

```
# puppet cert list
```

Para firmar los certificados se ejecuta:

```
# puppet cert sign (nombre del nodo)
# puppet cert sign 192.168.1.2
```

Existe la opción `autosign` donde todos los certificados que vengan de conexiones de direcciones de IP específicas o un rango de direcciones se firman automáticamente `autosign (autosing=true)`.

Con el siguiente comando se firmará todos los certificados que se encuentran en espera para ser firmados.

```
# puppet cert sign --all
```

Para verificar los certificados que se han firmado se ejecuta:

```
# puppet cert --list –all
```

Esta línea nos mostrará toda la lista de certificados, los que se encuentran firmados tienen al principio un + (signo más).

Después de firmar el certificado, se debe de esperar 2 minutos, y el agente estaría autentificado con el maestro.

Se mostrara lo siguiente:

```
#puppet agent --test
Info: Retrieving plugin
Info: Caching catalog for nodo1.super.unam.mx
Info: Applying configuration version '1365655737'
Notice: Finished catalog run in 0.13 seconds
```

Se puede mostrar un error, esto es porque todavía no se tiene configuraciones para el nodo.

```
Info: Retrieving plugin
Error: Could not retrieve catalog from remote server: Error 400 on
SERVER: Could not find default
node or by name with 'nodo1.super.unam.mx, nodo1' on node
nodo1.super.unam.mx
```

```
Warning: Not using cache on failed catalog
Error: Could not retrieve catalog; skipping run
```

Antes de configurar el maestro se debe sincronizar el reloj, las conexiones SSL se basan en el tiempo, si los relojes no se encuentran sincronizados, la conexión puede marcar error.

3.10 Creación de un manifests

El directorio `manifests` y el archivo `site.pp`, al igual que el archivo de configuración, se crean en la instalación, en caso de que no existan se crean:

```
# mkdir /etc/puppet/manifests
# touch /etc/puppet/manifests/site.pp
```

Se puede cambiar la ruta de el directorio y el archivo pero para facilidad lo dejaremos donde Puppet lo crea o lo debería haber creado.

Para definir los nodos se necesita del directorio `manifests`, dentro encontramos los archivos de configuración con la terminación `.pp`.

Entramos al archivo `manifests`

```
# cd  /etc/puppet/manifests
```

Para definir los 2 agentes, en el archivo `site.pp` agregamos la siguientes líneas:

```
node 'nodo1.super.unam.mx' {
```

```
}
node 'nodo2.super.unam.mx' {
}
```

Dependiendo en que nodo queremos que se agregue la `clase` , se incluye el nombre de la misma:

```
node 'nodo1.super.unam.mx {
include nombre-de-la-clase
}
```

3.11 Creación de un módulo

Se crea primero el directorio y se nombra al módulo sudo con el siguiente comando:

```
    # mkdir -p /etc/puppet/modules/nombre_de_la_clase/ manifests
           # mkdir -p /etc/puppet/modules/sudo/ manifests

  # touch /etc/puppet/modules/nombre_de_la_clase/manifests/init.pp
         # touch /etc/puppet/modules/sudo/manifests/init.pp
```

Los archivos `init.pp` son el núcleo de los `módulos`, cada `módulo` debe de tener uno.
Creamos un archivo `init.pp` y agregaremos la siguiente configuración:

```
class sudo {
package { 'sudo':
ensure => present,
}

file { '/etc/sudoers':
```

```
owner => 'root',
group => 'root',
mode => '0440',
source => "puppet:///modules/sudo/etc/sudoers",
require => Package['sudo'],
}
}
```

El código contiene un recursos de la `clase`, un `paquete` y un `recurso` de archivo. El primer recurso se asegura que se encuentre instalado el paquete sudo `ensure => present`.

El recurso nos indica donde se encontrará el archivo `sudoers`, el propietario, el grupo a que pertenecerá y los permisos que tendrá el archivo.

El atributo `source` nos indica donde va a recuperar el archivo, en este caso se le indica que lo recupere del servidor del cliente a cual se encuentre conectado.

Mandamos a llamar este archivo.

En el archivo `site.pp` agregamos la siguiente línea, considerando en que nodos queremos que se agregue la `clase`:

```
Node 'node2.super.unam.mx' {
Include sudo
}
```

Crearemos un `módulo` que nos avisará cuando un archivo de configuración sea modificado.

```
       # mkdir -p /etc/puppet/modules/avisacambios/ manifests
     # touch /etc/puppet/modules/avisacambios/manifests/init.pp
```

Una parte del código es:

```
exec { 'Verificando archivos de Configuración':
```

```
    path => "/usr/sbin",.
    unless => "find Verifica.sh"
   command => "./Verifica.sh"
  }
```

Esta parte se encarga de ejecutar el *script* que se hizo en *shell* para verificar los archivos de configuración, si hubo algún cambio, se avisará con un mensaje.

3.12 Puppet Forge

Puppet Forge es un repositorio donde podemos encontrar módulos preexistentes, que son escritos por usuarios.
Para encontrar módulos en Forge, se utiliza el comando:

```
# puppet module search nombre_del_módulo
```

Con este comando buscaremos los módulos que se encuentran preescritos para monitoreo de los nodos.

```
# puppet module search monitoring
```

Se presenta una tabla que contiene:

Columna	Descripción
Name	Nombre que el autor le pone a su módulo
Description	Describe lo que el módulo hace.
Autor	Nombre del autor del módulo
Keywords	Palabras clave.

Nombre del módulo.

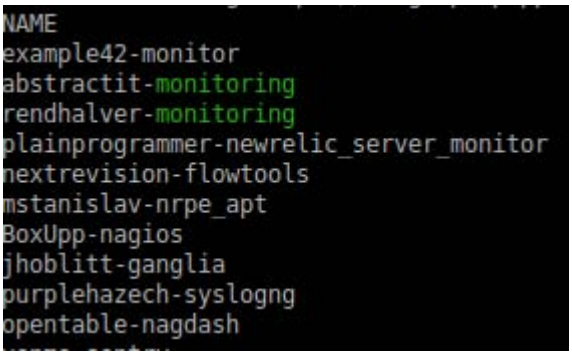


Ilustración 4: Ejemplo del nombre del módulo.

Descripción del módulo.

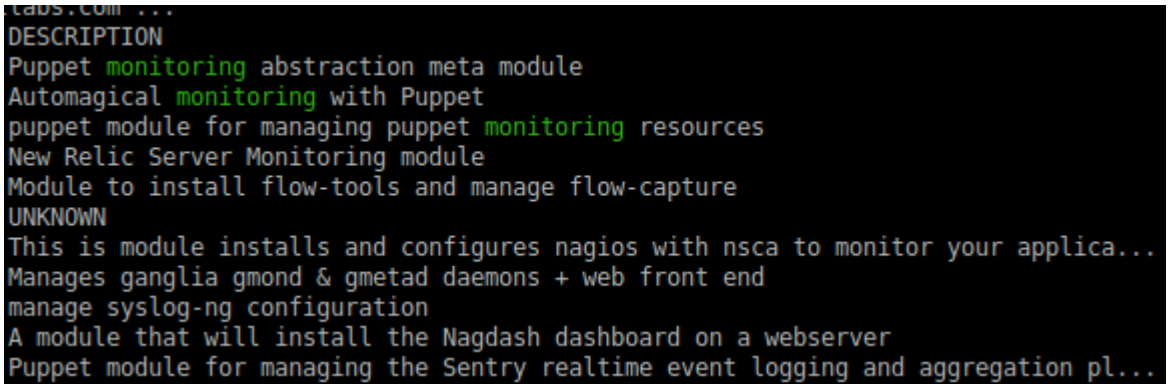


Ilustración 5: Ejemplo de descripción del módulo.

Nombre del autor .

Ilustración 6: Ejemplos de autores del módulo.

Palabras clave sobre el módulo.

```
KEYWORDS
monitor example42 abstraction
monitoring
nagios monitoring nrpe icinga
monitoring newrelic nrsysmond
debian ubuntu monitoring centos flow
nagios monitoring nrpe monitor
nagios monitoring nsca log4jxml
monitoring rhel report centos gmond
monitoring logging syslog syslog-ng log
nagios monitoring nagdash dashboards
monitoring logging alerting event sentry
debian ubuntu monitoring rhel centos
```

Ilustración 7: Ejemplos de palabras clave del módulo.

Como no se sabe que es lo que realmente hace el módulo, se recomienda ir a la página de Puppet sección forge y buscar, tomando en cuenta el número de descargas para saber qué módulos son los más usados y por lo tanto, más confiables.

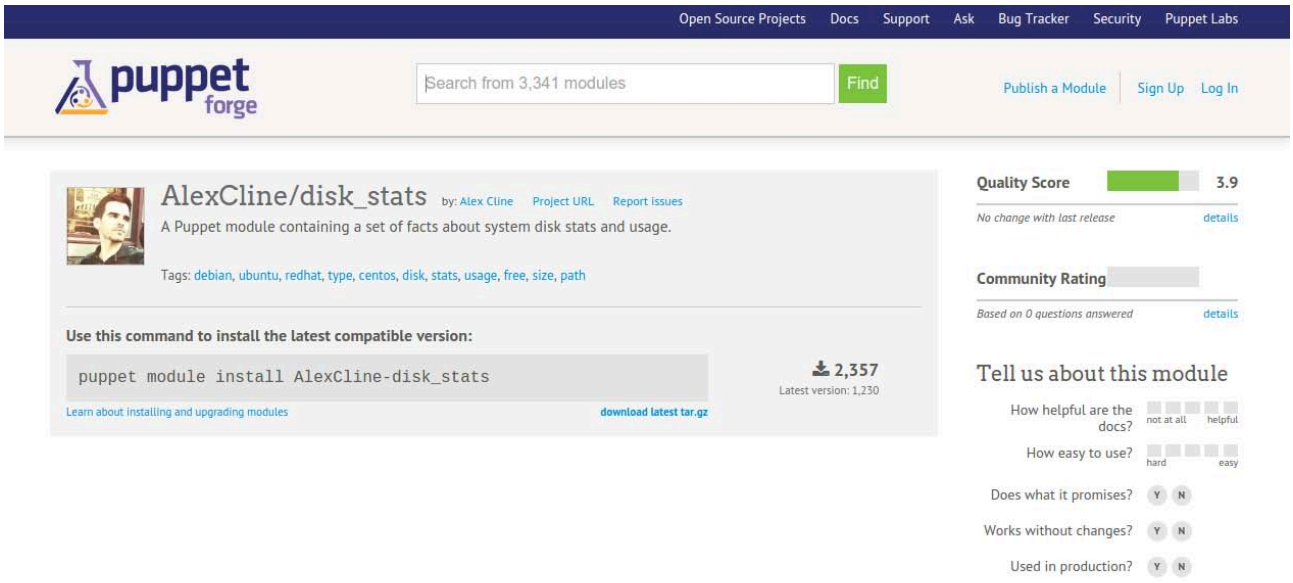


Ilustración 8: Sitio de módulos.

Los `módulos` tienen un `README` donde el autor especifica cómo se debe de instalar el `módulo` y sus características, o en la misma página explican como instalar el `módulo`.

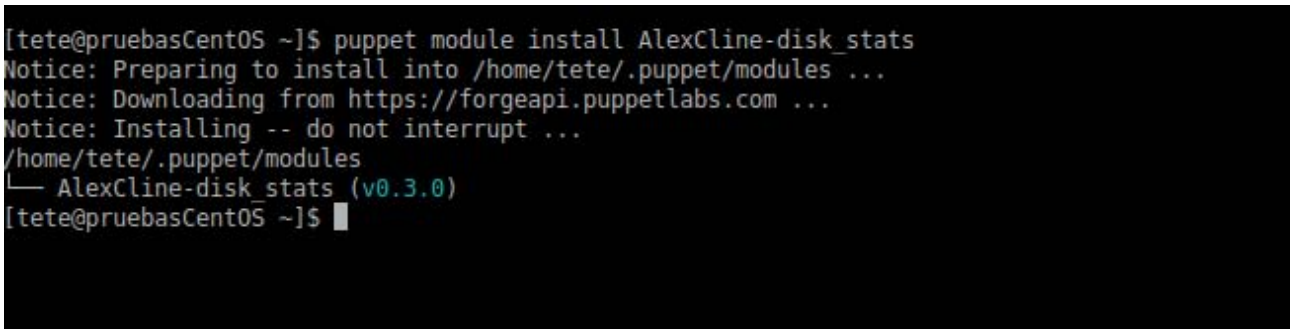


Ilustración 9: Instalación de un módulo.

Una parte del código del módulo descargado es:

```
  exec { 'Install sys-filesystem rubygem dependency':
    command => "${env_path} gem install sys-filesystem",
      unless   => "${env_path} gem list --local | /bin/grep sys-
filesystem",
    require => $require_list,
  }
}
```

Donde se requiere que los paquetes `rubygems`, `ruby-devel`, `make`, `gcc-c++` y `sys-filesystem` se encuentren instalados .

Se instala `sys-filesystem`, si el resultado del comando donde se encuentra `unless` es 1, pide que se muestre:

```
                disk_stats_root_free => 18 GB
          disk_stats_root_options => rw,errors=remount-ro
                  disk_stats_root_path => /
                disk_stats_root_size => 20 GB
                 disk_stats_root_type => ext4
                  disk_stats_root_used => 2 GB
```

Puede salir un error al instalar algún módulo. Si el error es:

```
Error: Request to Puppet Forge failed.
```

Es porque el módulo que se trato de descargar sólo está disponible para Puppet Enterprise

Para verificar los módulos que tenemos instalados:

```
#puppet module list
```

Se mostrará una lista de los módulos instalados.

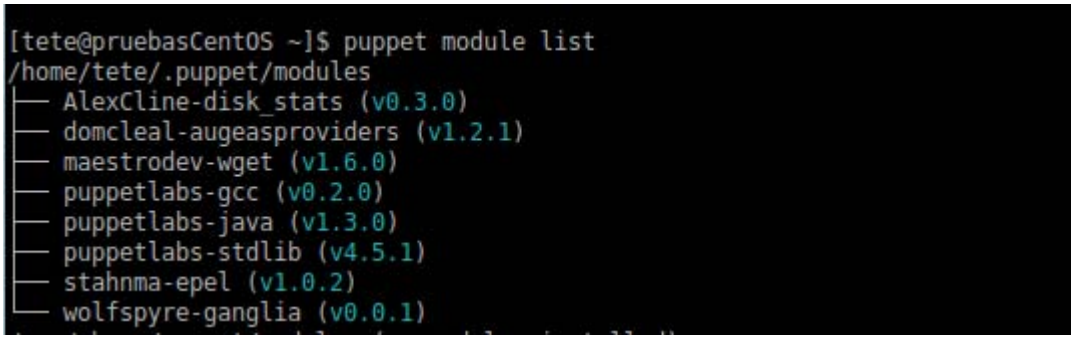


Ilustración 10: Lista de módulos instalados.

Capítulo IV

Como se mencionó anteriormente, en el departamento de Supercómputo de la DGTIC, se necesita de un sistema capaz de administrar la supercomputadora Miztli. Puppet es un sistema para automatizar tareas para administrar un clúster. Su funcionamiento es de cliente/servidor, donde el cliente se encarga de conectarse al nodo maestro(Master) para sincronizar los archivos de configuración y si se realizó algún cambio, se actualicen.
Con este sistema Puppet, se busca monitorear y controlar el acceso a los archivos de configuración del sistema que se encuentran en la supercomputadora Miztli.

Conclusiones

Como parte de los objetivos de este trabajo se investigó cómo funciona la herramienta y con base en eso, llegamos a determinar una primera configuración básica que permita tener comunicación con los nodos para verificar los archivos, en caso de que alguno hubiese sido modificado, se manda reporte.

Se necesitó de un sistema GNU/Linux CentOS para hacer pruebas con Puppet. Se llegó a la instalación del sistema operativo y del sistema de monitoreo Puppet, en los agentes y en el maestro (Master). Las configuraciones que se pudieron llevar a cabo en los nodos fueron de seguridad y algunas de red.

Durante esta investigación se realizó asimismo, la documentación lo que se hizo, cómo se hizo y el porqué se hizo así, es decir del procedimiento, con lo cual se cumplió otro de los objetivos de este proyecto. En algunos casos existían varias formas de realizar lo mismo, sobre todo en la parte de configurar Puppet, pero tomando en cuenta las características del sistema y las necesidades de la organización, se decidió hacerlo como se encuentra en este trabajo.

Comentarios personales

Para conocer Puppet, se debe de tener conocimientos básicos sobre algunas áreas de la informática, ya que sin esto, el entender esta herramienta de monitoreo se volvería más complicado.

La administración en Unix/Linux es la base para aprender a utilizar Puppet sobre todo porque sólo permite que el nodo Master tenga un sistema operativo GNU/Linux. También nos ayuda a entender como Puppet organiza sus directorios y cómo invoca algunas funciones.

También se debe tener bases sobre programación, sin esto no se podrían crear los módulos que son necesarios para que Puppet administre los nodos y, aunque el lenguaje que utiliza Puppet no es complicado por ser interpretado, el tener conocimientos sobre programación nos facilita el entender y aprender su lenguaje.

Glosario

Autenticación: Procedimiento que asegura que un usuario es quien dice ser.

Autentificación: Proceso para verificar la identidad de una persona.

Backbone: Enlace principal de una red.

Base de datos: Colección de datos relacionados, organizados y almacenados.

Bit: *Binary Digit*, Unidad mínima de almacenamiento. Sólo puede tomar dos valores**:** 1 ó 0

Byte: Unión de 8 bits. La capacidad de almacenamiento de información de una computadora es medida en bytes.

Cifrado: método para transcribir un mensaje o clave que se quiere ocultar en símbolos o letras.

Datos: Unidad mínima de almacenamiento

Demonio: Proceso que se encuentra en ejecución en segundo plano.

Gateway: Conecta o comunica a dos redes de diferente arquitectura.

Mainframe: También conocidas como macrocomputadoras, son computadoras de gran capacidad de procesamiento como las supercomputadoras y sus costos son elevados. A diferencia de una supercomputadora, las macrocomputadoras o *mainframe* su tamaño es mayor y el poder de procesamiento es mucho más grande.

Modelo OSI: Interconexión de Sistemas Abiertos, desarrollado por la Organización Internacional de est[andares (ISO). Las funciones que se necesitan para transferir información entre computadoras se dividen en 7 capas.

Capa 7 – Capa de Aplicación.

Capa 6 Capa de presentación.

Capa 5- Capa de sesión.

Capa 4 – Capa de transporte.

Capa 3- Capa de red.

Capa 2– Capa de enlace de datos.

Capa 1- Capa física.

Protocolo: Es un conjunto de normas que permiten que dos equipos establezcan comunicación para transmitir información.

Superusuario: Usuario privilegiado del sistema que tiene acceso a todos los recursos del sistema y permisos permitiendo que mantenga configure y actualice el sistema.

SSL: *Secure Socket Layer* Protocolo para realizar conexiones seguras. Utiliza un cifrado simétrico.

Switch: Dispositivo que enlaza segmentos de red muy similar a un puente pero con más puertos.

Referencias

1. Carlos M.. (2005). Base de datos. Marzo-12-2015, de UNAM FCA Sitio web: http://fcasua.contad.unam.mx/apuntes/interiores/docs/2005/informatica/3/1365.pdf
2. D-Link Building Networks for People. (No disponible). RAID Características, ventajas y aplicaciones . 15-octubre-2014, de D-Link Sitio web: http://www.dlink.com/-/media/Files/B2B%20Briefs/ES/dlinkraid.pdf
3. Dirección General de Cómputo y de Tecnologías de Información y Comunicación, *Acerca de clústererado de* http://www.tic.unam.mx/pdfs/AcuerdoTIC.pdf
4. Dirección General de Cómputo y de Tecnologías de Información y Comunicación *¿Quienes somos?* Recuperado de http://www.tic.unam.mx/mision.html.
5. Dr. Victor J.. (No aplica). *SNMP*. septiembre-17-2014, de cinvestav Tamaulipas Sitio web: http://www.tamps.cinvestav.mx/~vjsosa/clases/redes/SNMP.pdf
6. E. Alcalde/F. Ormaechea J.Portillo/F. Garcia Merayo. (1991). *Arquitectura de computadoras*. España: MCGraw-Hill.
7. Eugenia Ma.. (2012). *CMIP*. Octubre-3-2014, de Ma. Eugenia Macías Ríos Sitio web: http://redyseguridad.fi-p.unam.mx/pp/maru/labpracticas/CMIP.pdf
8. Gonzalo Ferreyra Cortés. (2006). *Informática paso a paso*. México: Alfaomega.
9. INTEF Instituto Nacional de Tecnologias Educativas y de Formación del Profesorado. (No dice). OpenLDAP. Octubre-15-2014, de INTEF Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado Sitio web: http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m6/openldap.html
10. José Castillo C.& Ricardo Benjamín O. (2006). *Implementación de un clúster OpenMosix para cómputo científico en el Instituto de Ingeniería*. Abril-22-14, de Licencia: Creative Commons Sitio web: http://www.josecc.net/archivos/tesis/tesis_html1/node6.html
11. JoseCC & SuperBenja. (2006-06-08). *Implementación de un clúster Open Mosix para cómputo científico en el Instituto de Ingeniería*. Febrero-12-2014, de josecc.net Sitio web: http://www.josecc.net/archivos/tesis/tesis_html1/node6.html

12. Nate Campi/Kirk Bauer. (2009). *Administración Linux/Unix Automatización de tareas y procesos*. España: Grupo anaya .
13. Oscar A. (2000). *Arquitectura de administración OSI*. septiembre-3-2014, de Oscar Agudelo R. Sitio web: http://www.arcesio.net/osinm/osinmfuncion.html
14. Puppet Labs . (2015). *Puppet Labs Documentation*. diciembre-8-2014, de Puppet Labs Sitio web: http://docs.puppetlabs.com/
15. Project Hosting Help. (2014). *Overview.* marzo 17 2014, de Google Project Hosting Sitio web: https://code.google.com/p/pdsh/
16. R. Hernando. (2002.). Gestión de redes. Octubre-2-2014, de R. Hernando Sitio web: http://www2.rhernando.net/modules/tutorials/doc/redes/Gredes.html
17. Red Hat Enterprise Linux. (2005). *Red Hat Enterprise Linux 4*: Manual de referencia. Octubre-2-2014, de Red Hat Enterprise Linux 4 Sitio web: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ldap.html
18. Spencer K., William V., Ben K. , James T. y Jeffrey M.. (2013). *Pro Puppet.* New York: apress.
19. Universidad Nacional Autónoma de México (UNAM). (2014). *Supercómputo*. Mayo, 5, 2014, de Dirección de Sistemas y Servicios Institucionales Sitio web: http://sistemas.tic.unam.mx/?q=supercomputo
20. Supercómputo. (2009 - 2014). *¿Qué es Supercómputo?* . Febrero-12-2014, de Universidad Nacional Autónoma de México DGTIC Sitio web: http://www.super.unam.mx/index.php/home/enlace/quesc
21. Yolanda F. & Reyna C.. (2005). *Administración de UNIX*. Abril-10-2014, de UNAM FCA Sitio web: http://fcasua.contad.unam.mx/apuntes/interiores/docs/2005/informatica/6/2048.pdf
22. 50 años del Cómputo en México, *El cómputo en México*, Cronología y Aplicaciones, Sitio web: http://caomputo50.unam.mx/computoenmexico2.html