Implantació d'un sistema de monitorització i gestió de les incidències

Manuel Cuesta Criado

Resum— Actualment, cada vegada les xarxes informàtiques són més grans i gestionar-les és més complicat. Per aquesta raó, hi ha eines que s'encarreguen d'aquesta tasca com Nagios. Aquesta eina disposa d'un versió gratuïta i permet monitoritzar un gran ventall de dispositius (Linux, Windows, OS X) a més de routers, impressores, etcètera. Els avantatges són clars: gestió eficient dels recursos, cost reduït i no hi ha limitació pel tipus de màquina. Unit a la monitorització, es presenta una comparativa entre dues eines per la gestió d'incidències molt conegudes en el seu àmbit com són JIRA i OTRS. L'objectiu és clar: gestionar les incidències amb una mateixa eina i que sigui configurable a les nostres necessitats. També s'implantarà Teampass, una eina per gestionar les credencials de forma centralitzada i que aporta molts avantatges en comparació a d'altres eines dedicades a la mateixa tasca com són: actualització en un únic punt, no depèn de cap aplicació i és accesible via web. Per últim, com tot l'escenari de treball és virtual, es veurà la potència que aquesta tècnica aporta com ara la portabilitat, la gestió eficient dels recursos disponibles i la facilitat per fer actualitzacions ja que permet fer un ràpid rollback si és necessari.

Paraules clau — Nagios, OTRS, JIRA, Teampass, monitorització, incidències, virtualització

Abstract— Nowadays, computer networks are bigger and managing them is getting harder. For that reason, there are some tools available in order to do that, like Nagios which has a free version and allows us to monitor not only a wide range of devices (Linux, Windows, OS X) but also routers, printers, etcetera. The advantadges are evident: efficient resources management, low cost and there is no limitation in terms of device. Joint to monitoring, a comparison between JIRA and OTRS, which are two tools made to manage incidents and very well-know in its topic, is shown. The purpose is obvious: managing all incidents in only one tool which also allows us to configure it fulfilling our needs. Teampass, a tool that will help us to manage passwords, will be also deployed because it offers some aspects that other similitar tools do not do like: update once, no application need to be installed and is web-based. Finally, as all the working scenario has been virtualized, the power of that technique will be shown in aspects such as portability, efficient resources management and low risk of failure when updating because a quick rollback is possible.

Index Terms — Nagios, OTRS, JIRA, Teampass, monitoring, incidents, virtualization

1 Introducció

Actualment, cada dia és més freqüent trobar dispositius electrònics connectats a servidors que poden estar ubicats a kilòmetres de distància i, per tant, els serveis oferts han d'estar accessibles i operatius les 24 hores del dia. Per aquest motiu, és important disposar d'una eina que informi quan hi ha alguna màquina o servei que no està funcionant com s'espera per, d'aquesta manera, poder avançar-se als usuaris i alhora poder centralitzar els recursos disponibles allà on realment són necessaris. D'altra banda, és també interessant tenir un control de les incidències que hi ha actualment obertes per poder fer un seguiment centralitzat d'aquestes. Tampoc s'ha d'oblidar que la part de la seguretat s'ha de tenir sempre en compte i, per tant, les credencials d'accés a cada màquina haurien de ser diferents i suficientment robustes. Per últim, disposar d'un entorn virtualitzat és un punt a favor tenint en compte els grans avantatges que ofereix com poden ser l'ús òptim dels recursos de la màquina, la facilitat de migració a una altra o el poder fer canvis sense por a no poder tornar enrere en cas que no s'obtingui el resultat esperat.

Per tant, els objectius a complir són els següents:

- Instal·lar un sistema de monitorització per poder ser informats ràpidament en cas de detectar algun problema
- Instal·lar una eina per gestionar les incidències per poder fer un seguiment i gestió eficaç dels casos oberts
- Instal·lar una eina per gestionar les credencials per fer-les diferents i complexes per cada màquina i no haver-les de recordar
- Virtualitzar l'escenari per aprofitar la potència d'aquesta tècnica

A continuació es trobarà l'estat de l'art; l'escenari de treball; l'eina de monitorització; la comparació entre les eines de gestió d'incidències; l'eina de credencials; els resultats; les conclusions; les futures línies de treball; els agraïments; la bibliografia i els apèndixs.

- *E-mail de contacte: manuel.cuesta@e-campus.uab.es*
- *Menció realitzada: Tecnologies de la Informació.*
- *Treball tutoritzat per: Ramon Musach i Pi (dEIC)*
- *Curs 2013/14*

2 ESTAT DE L'ART

Anys enrere, quan la quantitat de dispositius que es podien trobar a les empreses eren reduïts i amb pocs serveis associats, el disposar d'un eina que monitoritzés la xarxa era un aspecte que podia no ser de gran necessitat. No obstant, a mesura que aquests dispositius han anat creixent i millorant, el disposar d'aquesta eina ha passat a ser un factor clau ja que, com s'ha mencionat anteriorment, permet focalitzar els recursos disponibles allà on hi ha un problema. Eines que realitzin aquesta tasca n'hi ha de diverses i alguns exemples són Centreon, Nagios, Xymon o Zabbix.

Unit a aquest punt, amb la crescuda dels dispositius, les incidències associades a aquests són majors i, per tant, s'ha passat d'haver de gestionar un número petit de problemes a un nombre molt major i amb diferents proveïdors i/o clients, fet que ha motivat l'aparició d'eines per la gestió dels incidents com poden ser HP Open View, JIRA o OTRS.

Per últim, un aspecte que també ha canviat respecte del passat és el canvi de mentalitat referent a l'ús d'aquests dispositius. Fa 20 anys, els ordinadors es feien servir per fer càlculs i les tasques que es volien dur a terme en ells estaven prèviament estudiades per tal de que la CPU d'aquests estigués sempre a valors alts; no obstant, actualment, el factor de l'ús de CPU intensiu no és el que més es persegueix sinó el de fer un ús eficient dels recursos disponibles i evitar en la mesura del possible els talls en el servei. És en aquest punt on entra la virtualització dels escenaris de treball ja que ofereix grans avantatges com els prèviament esmentats.

3 ESCENARI DE TREBALL

En aquesta secció es definirà, per una banda, l'escenari de treball amb el què es treballarà i, tot seguit, es presentaran els avantatges de la virtualització de l'entorn.

3.1 Descripció de l'escenari

L'escenari amb el què es treballarà està format pels següents elements:

- Firewall: connectarà la xarxa real (en mode NAT) amb la xarxa virtual (en mode Host Only)
- DNS: resoldrà els noms de la xarxa virtual i farà també de DHCP
- Base de dades: allotjarà les bases de dades
- 3 servidors webs: per les eines de monitorització, gestió de les incidències i de credencials
- Host Windows: per monitoritzar un màquina que no estigui corrent sobre un sistema operatiu Linux.

A la *Figura 1* es mostra gràficament aquest escenari del qual cal destacar dues parts: la primera és que l'únic element no virtualitzat és el router i, la segona, que la funció del Firewall és només la de connectar les xarxes real i virtual[1].

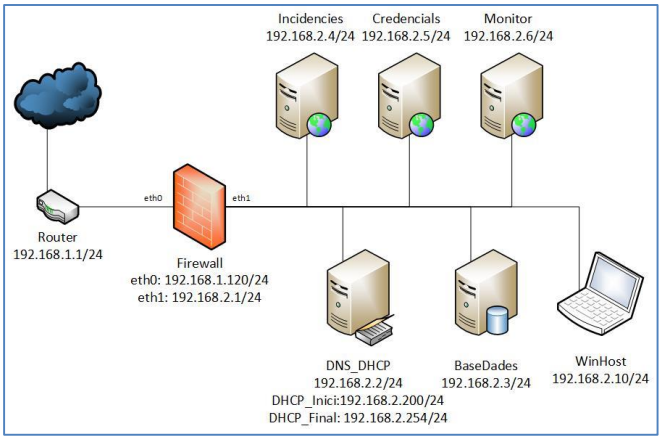

Figura 1. Escenari

3.2 Avantatges i inconvenients de la virtualització

A continuació es mostrarà la potència que aporta la virtualització:

- Permet fer un ús òptim dels recursos ja que la màquina física que conté les màquines virtuals pot dedicar més o menys recursos a una màquina virtual o una altra en funció de les seves necessitats
- Estalvi d'elements físics com ara ratolins, pantalles, teclats, fonts d'alimentació, discos durs… i es guanya espai físic.
- Tasques de migració i portabilitat molt més senzilles ja que només cal fer una exportació de la màquina virtual i una importació a la màquina física on es vulgui allotjar. Pot ser útil també per replicar serveis i disposar d'entorns de producció i desenvolupament.
- Actualitzacions i desplegaments amb un perill molt menor ja que es poden fer instantànies de la màquina virtual (snapshot) i en cas que la modificació no proporcioni el resultat esperat es pot fer marxa enrere (roll-back) amb un cost molt reduït.
- Facilita l'alta disponibilitat (High Availability) tenint en compte que permet la migració automàtica de màquines en cas que es detecti que una de les màquina físiques presenta problemes.

D'altra banda, també hi ha alguns inconvenients com ara:

- La màquina física ha de disposar de hardware d'altes prestacions; és a dir, no val qualsevol màquina.
- Moltes màquines virtuals depenen d'una màquina física.

Tot i els inconvenients, els avantatges són molt superiors i aporten solucions que sense virtualitzar tindrien un cost bastant superior.

[1] S'ha fet ús d'iptables per dur a terme aquesta tasca. A l'Apèndix A1 es pot trobar l'script que s'encarrega d'activar aquesta funció.

4 EINA PER LA MONITORITZACIÓ

L'eina que s'ha fet servir és Nagios. És Open Source i té diferents versions, una de les quals és gratuïta i és la que s'ha utilitzat: Nagios Core DIY (Do It Yourself).

La diferència entre les versions radica en aspectes com ara si disposa –o no– de suport fora dels fòrums i manuals, funcionalitats extres com poden ser la visualització de gràfiques de l'estat dels serveis o el poder afegir màquines i/o serveis via interfície web i no modificant fitxers de configuració a mà, entre d'altres.

4.1 Introducció

Nagios permet monitoritzar dispositius que corrin sobre Linux, Windows o OS X i en cas que no corrin sobre aquests, es poden monitoritzar si fan servir el protocol SNMP. Aquest protocol és especialment útil si es tracta de dispositius com ara impressores o routers. Val a dir que tot i que Nagios permet obtenir l'estat de màquines que facin servir diferents tipus de sistemes operatius, el host on estigui Nagios corrent ha de ser basat en Linux.

Nagios diposa d'un paquet de serveis bàsics que s'anomena Nagios Plugins el qual conté chequejos (checks) a serveis referents a la càrrega del sistema, l'espai ocupat en disc, als processos que estan corrent, usuaris connectats, accions sobre webs… i en funció dels llindars que es defineixin retornarà un estat o un altre. Perquè Nagios i les diferents màquines que es volen monitoritzar s'entenguin, cal instal·lar un agent a les màquines per tal que s'encarregui de recopilar la informació i enviar-se-la al host Nagios de forma que la pugui entendre i tractar. Dit això, per les màquines Linux s'ha fet servir l'agent NRPE (Nagios Remote Plugin Executor) i per la Windows, l'agent NSClient++.

4.2 Instal·lació

La instal·lació de Nagios es pot fer via repositoris (`apt-get install nagios`) o compilant el codi [1]. La principal diferència entre fer una instal·lació fent servir repositoris o compilant el codi és que a la primera via no es pot escollir la versió (és la que hi hagi al repositoris i pot no ser la més actual) mentre que compil·lant el codi es pot escollir la versió i la instal·lació és més personalitzada.

Referent a la instal·lació de l'agent NRPE és com l'anterior cas: a través de repositoris (`apt-get install nagios-nrpe-plugin`) o compil·lant-lo [2].

Per últim, la instal·lació de l'agent NSClient++ es troba a la seva web [3].

4.3 Configuració

Nagios disposa del fitxer `nagios.cfg` en el qual es troba la informació del procés, els diferents fitxers que ha de llegir per carregar informació (com ara el de les màquines i serveis a processar), els valors de timeout, la ubicació del fitxer de log i d'altres dades més. Com aquest fitxer és de vital importància pel correcte funcionament de l'eina, cada cop que es fa una modificació –si conté algun error– en aquest o un dels fitxers que estan associats es corre el perill de perdre la monitorització fins que no es solucioni. Nagios diposa de la següent instrucció:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Aquesta, permet fer una comprovació prèvia la qual informa si els fitxers són correctes i, així, evitar problemes.

Per poder afegir màquines i serveis a la monitorització cal definir un fitxer i lincar-lo al `nagios.cfg`. Això permet una gran modularitat ja que es poden crear tants fitxers com es desitgi i així poder separar les màquines, els serveis, els contactes… fent que sigui molt més fàcil tractar amb fitxers d'extensions petites a un amb tota la informació. A l'hora de definir qualsevol objecte (ja sigui màquina, servei, contacte…) s'han de definir moltes propietats associades a cadascun i, per evitar aquesta repetició, Nagios permet definir plantilles i poder fer ús d'elles amb la propietat `use $plantilla` [4]. Sabent aquesta informació, es pot definir una plantilla (`linux-server`) perquè demani només les dades que són pròpies de cada màquina com podrien ser el `host_name`, `alias` (si es vol), `address` i si té alguna dependència amb la propietat `parent` (p. e. si una màquina està darrere d'un switch, el switch seria el seu `parent`). Un exemple seria el següent:

```
define host{
        use             linux-server
        host_name       monitor
        alias           monitor
        address         127.0.0.1
        parents         firewall
    }
```

Un cop definida una màquina, podem associar-li comprovacions (checks) a monitoritzar. Abans, però, cal destacar que Nagios treballa amb els següents tipus d'estat:

- Ok: la comprovació està dintre dels llindars
- Warning: la comprovació està per sobre del llindar però no és crític
- Critical: la comprovació està per sobre del llindar i és crític.
- Unknown: la informació que s'ha obtingut no pot ser interpretada per Nagios.
- Pending: s'ha afegit una comprovació a la monitorització i encara no s'ha realitzat cap check.

I els següents estats afecten només a hosts i no a serveis:

- Up: s'arriba al host
- Down: no s'arriba al host
- Unreachable: no es pot arribar al host però perquè un element del que depèn està caigut.

Un cop definits els estats, podem començar a afegir comprovacions. Per dur a terme aquesta tasca farem ús d'una altra plantilla (`local-service`) i afegirem les propietats `host_name`, `service_description`, `com-`

`mand_check` i `contacts`. Un exemple seria el següent:

```
define service{
        use                     local-service
        host_name               monitor
        service_description     PING
        check_command           check_ping!1.0,20%!5.0,60%
        contacts                admin
}
```

Opcionalment es poden afegir altres propietats com ara `notes_url` (per associar algun tipus de documentació) o `icon_image` (per associar-li alguna icona al servei). A més, si es volgués que aquesta comprovació es fes per tots els hosts, el camp `host_name` podria canviar-se per * o, en cas que fos una comprovació que engloba un conjunt de hosts concrets, es podria definir un hostgroup i substituir la propietat `host_name` per `hostgroup_name` de tal manera que ens estalviaríem haver d'escriure cada cop tots els hosts que formessin part d'aquest grup.

Les propietats que pot tenir associades cada objecte les podem trobar a la documentació oficial de Nagios a la secció Object Types [5]. Cal destacar que la propietat `check_command` s'ha definit de la següent manera:

```
check_ping!1.0,20%!5.0,60%
```

Això, és degut a que les instruccions que ha de processar Nagios han d'estar prèviament especificades perquè pugui entendre què volen dir els paràmetres i quin acció fer amb ells. Fent ús de l'exemple del check de ping, a continuació, es mostra com ha estat definit:

```
define command{
        command_name    check_ping
        command_line    $USER1$/check_ping -H $HOS
TADDRESS$ -w $ARG1$ -c $ARG2$ -p 5
        }
```

La propietat `command_name` indica amb quin nom ens referirem a l'hora de cridar aquest check i la propietat `command_line` indica quins paràmetres ha de rebre. `USER1` i `HOSTADDRESS` són dues macros que agafen la informació automàticament de l'usuari i host respectivament [6] i les restants indiquen el llindar perquè es consideri alerta Warning (`-w`), Critical (`-c`) i el nombre de paquets a enviar (`-p`) [7].

Una altra propietat que s'ha fet servir és la contacts:

```
contacts admin
```

Perquè Nagios sàpigui qui és admin, se li ha d'especificar. Per a fer-ho, farem ús novament d'una plantilla (`generic-contact`) que ens facilita l'eina i només haurem de definir les propietats `contact_name, alias` i `email`, com es pot veure a continuació:

```
define contact{
        contact_name    admin
        use             generic-contact
        alias           Nagios Admin
        email           usuari@domini.tld
        }
```

Per qüestions relacionades amb l'enviament dels mails i quina informació afegir-hi, veure l'Apèndix A2.

Per últim, val a dir que, tot i que Nagios ofereixi el paquet de checks que s'ha mencionat anteriorment (Nagios Plugins), sempre es poden definir checks creats per nosaltres o algú altre fent que es pugui arribar a monitoritzar qualsevol tipus de servei. A l'Apèndix A3 es pot trobar un check propi.

4.4 Gestió de les alertes

Un cop s'han definit les diferents comprovacions, podem veure quin és l'aspecte gràfic que presenta Nagios com es mostra a la *Figura 2*.

Figura 2. Vista Nagios

Aquí podem observar que es tracta d'un host basat en Debian i que té diferents serveis, dos dels quals estan en Critical. La icona que té aspecte de document/paper és un enllaç a la documentació que té aquest servei associada (recordar la propietat `notes_url`) i la icona del gràfic és per veure informació gràfica d'aquest servei com ara quan supera els llindar definits.

Com aquestes dues alertes crítiques sabem de què tracten (el Firewall està aturat expressament perquè saltin), Nagios permet habilitar un mecanisme anomenat Acknowledge el qual permet indicar que aquesta alerta està "sota control" i que, per tant, no l'ha de reportar. També disposa d'un altre mecanisme anomenat Downtime el qual durant el temps que es marqui les alertes que apareguin no seran notificades ja que com el mateix nom indica estan, a efectes de monitorització, en "període de ignorar". A la *Figura 3* es mostra gràficament com es veu aquesta informació aplicant els mecanismes de Acknowledge (icona "correcte") i Downtime (icona del rellotge).

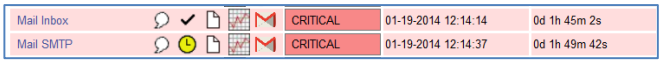

Figura 3. Acknowledge i Downtime

La icona del globus indica el comentari que s'ha afegit a l'hora de definir l'Acknowledge i el Downtime.

5 EINA PER LA GESTIÓ DE LES INCIDÈNCIES

Per poder gestionar les incidències farem ús de dues aplicacions que funcionen via interfície web: JIRA i OTRS. La funció de les dues eines és clara: poder disposar d'una eina centralitzada on puguem notificar, actualitzar, tancar… les incidències, alhora que puguem obtenir informes sobre les incidències mensuals i els dispositius dels quals més problemes es reporten.

5.1 JIRA

5.1.1 Introducció

JIRA és una eina de l'empresa Atlassian orientada a la gestió d'incidències de projectes (de Software, p. e.) i té una interfície web molt amigable i senzilla. No obstant, l'eina és de pagament encara que ofereix descomptes de fins al 50% per ONGs, entitats educatives, entre d'altres.

Com a punts d'interés, buscant a Google pel nombre d'entrades de JIRA en surten prop de 4.5 milions; a més, l'empresa Atlassian ofereix altres eines que es poden integrar a JIRA com són Confluence (per la gestió documental) i BitBucket i Stash (cloud i repositoris de Software).

5.1.2 Instal·lació

L'eina es pot instal·lar en sistemes operatius Linux, OS X i Windows i les bases de dades que permet són PostgreSQL, MySQL, Oracle, SQL Server i HSQLDB; un dels motius pels quals és tan extès. El procés d'instal·lació es pot fer via interfície gràfica o compil·lant el codi via terminal [8].

5.1.3 Gestió i Configuració

Per poder començar a fer ús de l'eina, el primer que cal fer és definir un projecte ja que les incidències estan lligades a ell i el diagrama de flux del mateix. Acte seguit, es poden definir diverses propietats; algunes d'elles són:

- les components del projecte: això seria els camps genèrics on poder associar les incidències (Bases de dades, Xarxa, Correu…)
- els tipus d'incidències que es poden crear: per defecte són error, petició, millora i tasca; encara que l'eina permet crear-ne de noves.
- les prioritats/impacte: bloquejant, crítica, alta, normal i baixa; encara que també se'n poden crear de noves

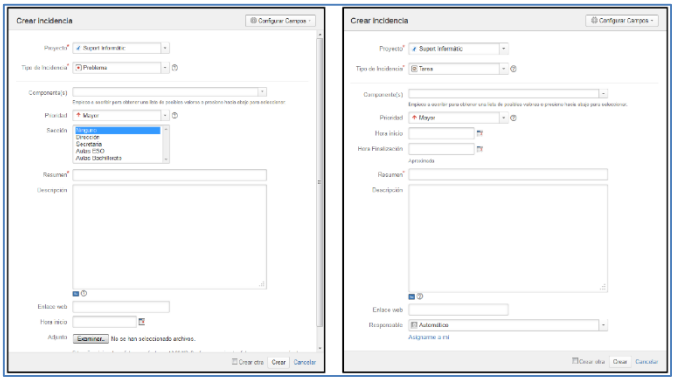

Figura 4. Incidències JIRA

Un altre aspecte que fa que JIRA sigui tan extès és la seva interfície gràfica ja que és molt intuïtiva i fàcil de fer servir. Els camps que s'han mencionat abans venen per defecte però si es volguessin afegir de nous com ara `ubicació` i que aparegués un llistat amb `<CPD, RRHH, Administració, …>` és molt senzill; a més, si es volgués mostrar unes propietats o unes altres en funció del tipus d'incidència també és possible; és a dir, si per una incidència d'`error` es volgués mostrar les propietats `A` i `B` i per una incidència de tipus `millora` les propietats `A` i `C`, l'eina ho permet. A la *Figura 4* es mostra un exemple.

Un cop s'ha creat la incidència, tindrem una vista com la que es motra a la *Figura 5*.

Figura 5. Vista incidència JIRA (I)

Aquesta incidència seguirà el diagrama de flux que s'hagi prèviament definit. En aquest cas, no s'ha assignat a cap tècnic com es mostra a la propietat `Responsable` de la dreta. En el moment que s'assigni, la propietat `estat` canviarà al que s'hagi definit al diagrama de flux (en aquest cas, a `En progreso`) com es motra a la *Figura 6*.

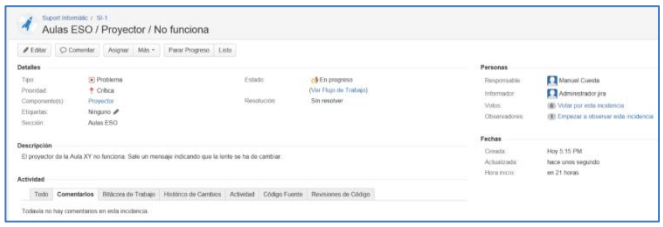

Figura 6. Vista incidència JIRA (II)

Per últim, JIRA permet extreure diferents informes de les incidències de forma gràfica; a la *Figura 7* es llisten quins són aquests.

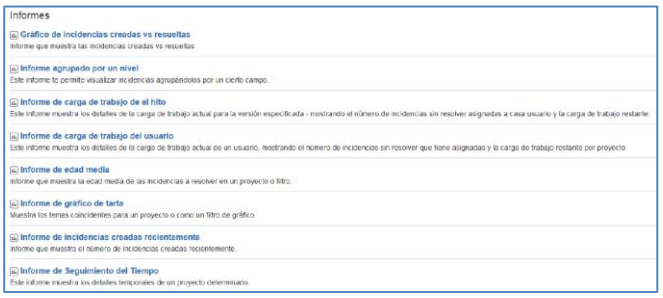

Figura 7. Informes JIRA

A la secció següent, es parlarà sobre OTRS i un cop acabada aquesta part, es compararan JIRA i OTRS.

5.2 OTRS

5.2.1 Introducció

OTRS és una eina gratuïta per a la gestió d'incidències i està orientada al tracte amb els clients i els proveïdors; motiu pel qual disposa de dues interfícies webs ben diferenciades: una pels usuaris interns (coneguts com a agents) i una altre pels usuaris externs (clients i proveïdors).

Com a dada d'interés, buscant a Google per OTRS apareixen prop d'1 milió d'entrades.

5.2.2 Instal·lació

L'eina es pot instal·lar en sistemes operatius Linux i Windows i les bases de dades que permet són PostgreSQL, MySQL, Oracle i SQL Server. El procés d'instal·lació es pot fer via interfície gràfica o compil·lant el codi via terminal [9].

5.2.3 Configuració

A l'hora de configurar OTRS cal tenir molt clars els conceptes següents:

- Ticket: ID de la incidència o petició i conté tot l'històric d'actuacions.
- Cua: categoria on s'assigna el ticket
- Agent: correspon als usuaris interns (els tècnics, per exemple) els quals hauran de tractar la incidència
- Client: correspon a aquells els quals reporten les incidències o peticions als agents.

Com es pot veure a la *Figura 8* aquesta distinció prèvia és totalment evident tal i com es mostra en els 4 primers quadres; els 2 quadres restants fan referència a qüestions de missatgeria i a dades del sistema.

Figura 8. Configuració OTRS

La creació d'agents i clients és molt senzilla i l'assignació dels permisos i grups als agents es fa via una taula on s'indica l'agent (en fila) i els permisos i grups possibles (en columna) i per assignar-ne o treure'n només cal marcar o desmarcar la casella corresponent.

Un cop creats els agents i el grups, es poden crear les cues que aniran associades a cada grup. La creació és molt simple i moltes opcions es presenten via llistes per reduir errors humans; opcionalment, a cada cua se li poden assignar respostes automàtiques i firmes per presentar una visió més uniforme i estàndard al client.

Referent als tickets, es poden definir les prioritats, impactes, estats i d'altres més opcions possibles perquè només calgui escollir la més adient d'una llista desplegable. A més, també permet crear camps en cas que els disponibles no siguin suficients o els destitjats.

En la part del correu, cal fer especialment menció a l'opció PostMaster Mail Accounts perquè les adreces que s'afegeixen seran parsejades per OTRS per, en cas de rebre algun mail, crear el ticket o actualitzar la incidència si ja existeix. La resta d'opcions fan referència a certificats i filtres que es poden crear.

Per últim, a les opcions que resten, podem fer enviaments de correus als agents informant d'alguna tasca (p. e. de manteniment), gestionar les sessions obertes, gestionar paquets/mòduls i configurar aspectes més interns del sistema com ara els camps que es mostren al crear un ticket o a l'afegir una nota.

5.2.4 Gestió

Com s'ha comentat anteriorment, OTRS diferencia entre agents i clients i, per tant, mostra dues interfícies ben diferenciades. A la *Figura 9* es mostra la vista pels agents i a la *Figura 10* la vista pels clients.

5.2.4.1 Gestió clients

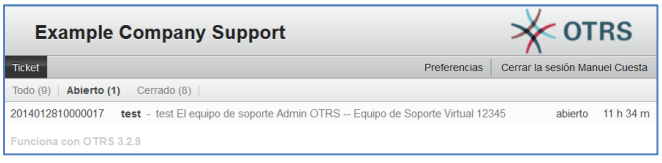

Figura 9. Vista gestió client

Com és d'esperar, la vista dels clients és molt simple: permet obrir incidències i veure les que s'estan cursant/tancades de forma que sigui de molt fàcil tractament inclús per a gent amb pocs coneixements tècnics.

A més, pel client és totalment transparent com funciona l'aplicació, és a dir, per exemple, no ha d'assignar una incidència directament a un tècnic, ni ha d'escollir cap cua; només ha d'omplir els camps que li són d'interès com el tipus d'incident, la prioritat i la descripció.

5.2.4.2 Gestió agents

Figura 10. Vista gestió agents

D'altra banda, la vista de l'agent és molt diferent; la vista inicial presenta un llistat de les incidències en curs i gràfiques indicant la relació entre casos oberts i tancats; a més, el panell superior permet crear incidències, generar estadístiques/informes, gestionar clients i administrar l'eina.

Mentre que en la vista del client, pel tractament d'una incidència s'havien de marcar pocs camps, en el cas dels agents és diferent; l'agent ha d'indicar el propietari de la incidència, l'estat en què es troba, la cua, pot afegir notes (internes; no visibles pel client) i modificar els camps que hagi introduït el client si s'escau.

En cas de voler extreure estadístiques n'hi ha de diverses i l'eina permet crear-ne de noves si és necessari.

Per últim, la pestanya clients mostra un llistat d'aquests i l'opció d'administració fa referència al que s'ha descrit al punt *5.2.3 Configuració*.

5.3 Comparativa JIRA i OTRS

A continuació, a la *Taula 1*, es mostra una comparativa les dues eines per la gestió d'incidències: JIRA i OTRS.

	JIRA	OTRS
Tipus	Pagament (*)	Gratuït
Complements i Millores	Gratuïtes i de pagament	
Orientació	Projectes	Clients
Accés	Interfície web	
Interfície	Molt fàcil	Normal
Usabilitat	Molt fàcil	Normal
Configuració	Molt fàcil	Normal
Formularis segons tipus d'incidència	Sí	No
Crear nous camps	Sí	
SO	Windows, Linux, Mac	
BD	MySQL, Oracle, Postgres, SQL Server i HSQLDB	MySQL, Oracle, Postgres i SQL Server
Notificació per correu	Sí	
Permet SLA	No	Sí
Programat	Java	Perl
Google	4.5 milions	1 milió
Clients	Adobe , Apache, Audi, Cisco, Citrix, eBay, EMC	Fujitsu, Intel, KFC, Lufthansa, NASA, Nokia, Porsche

Taula 1. Comparativa JIRA i OTRS

(*) Amb descomptes per a ONGs i entitats educatives, entre d'altres.

6 Eina per la gestió de les credencials

Per últim, queda presentar Teampass, l'eina on gestionarem les credencials i que és totalment gratuïta.

6.1 Introducció

Teampass és una eina la qual permet la gestió de credencials fent ús d'una interfície web simple. Punts a favor d'aquesta eina són els següents:

- Centralització: totes les credencials es troben en un punt
- No cal cap aplicació: funciona a través de web
- Administració de l'aplicació: la informació que conté és prou sensible com per confiar-la a una 3ª persona.
- Actualització transparent: a l'estar centralitzat, les credencials només cal actualitzar-les en un únic punt i no informar a tothom cada cop que es fa un canvi perquè siguin conscients. KeePass, per exemple, necessita carregar una BD i pot ser que no sigui la més recent (Teampass no presenta aquest problema)
- Grups i rols: permet definir grups i rols per donar/treure visibilitat a les credencials i/o permisos concrets com ara lectura/escriptura.
- Validació LDAP: en cas que estigui definida.
- Permet fer còpies de seguretat

Tot i que també presenta un punt negatiu a tenir molt en compte i és que totes les credencials depenen d'un únic punt; és a dir, en cas que algú tingui accés a Teampass pot obtenir tota la informació de les credencials. Per aquesta raó, s'ha de protegir molt l'accés a aquest servei web (p. e. permeten només l'accés dintre de la xarxa privada i fent ús de VPN en cas d'estar a l'exterior).

6.2 Instal·lació

Els requeriments previs són disposar d'una màquina LAMP; és a dir, sistema operatiu Linux, un servidor web Apache, una base de dades MySQL i PHP [10]. També és possible instal·lar-lo sobre Windows.

6.3 Configuració

Un cop s'ha instal·lat, Teampass es troba en mode manteniment i amb les opcions del menú superior podem configurar l'eina al nostre gust. Encara que el primer de tot i més recomanable és canviar les credencials per defecte de l'usuari administrador. A la *Figura 11* es mostra el panell superior amb les opcions disponibles.

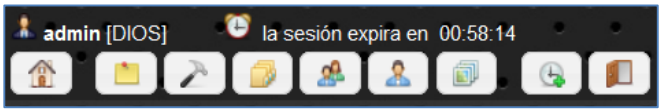

Figura 11. Menú Teampass

Un cop s'ha fet el canvi de credencials, es poden crear carpetes i definir usuaris i rols. La interfície és molt intuïtiva i fàcil de fer servir pel que no calen uns grans coneixements un cop s'ha fet la instal·lació. A més, sempre es poden fer còpies de seguretat de l'eina i, en aquest cas, com la màquina és virtual sempre es pot fer una snapshot per evitar problemes.

6.4 Gestió

Un cop s'han definit les carpetes, els usuaris i els rols, podem accedir amb un dels usuaris creats –l'usuari administrador no pot afegir credencials, només s'encarrega de l'administració– i veurem que el panell superior ha canviat lleugerament com es mostra a la *Figura 12*.

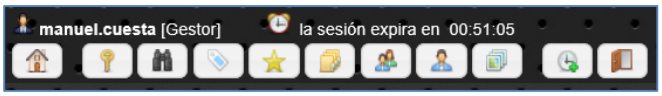

Figura 12. Gestió Teampass

Per exemple, com a Gestor apareix la icona de la clau per accedir a les credencials. A la *Figura 13* es mostra la vista que presenta Teampass a l'hora de tractar amb les credencials.

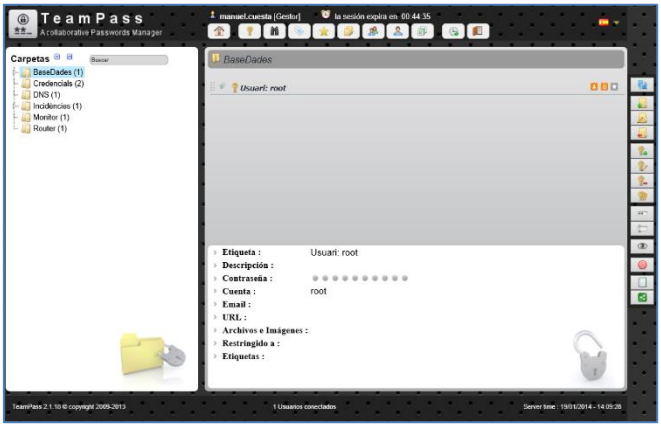

Figura 13. Vista Gestor

El panell esquerra mostra l'arbre de directoris; el central, la informació que penja del directori escollit; i el dreta, les opcions que es poden realitzar amb la credencial marcada o el directori actual. A mode d'exemple, si es volgués afegir una credencial s'hauria de clicar a la icona de la dreta que té dibuixada una clau i un signe de "+" mentre que si es volgués veure la credencial de l'usuari root s'hauria de clicar sobre la icona de l'ull.

7 RESULTATS

Abans d'entrar en les conclusions, cal destacar que els resultats del treball han complert amb els objectius inicials ja que s'ha implantat Nagios per poder dur a terme la monitorització de la xarxa informàtica; s'han instal·lat i configurat les dues eines de gestió d'incidències JIRA i OTRS a més de comparar-les entre elles; s'ha instal·lat també Teampass per poder disposar d'una eina per gestionar les credencials i, finalment, tot l'escenari de treball (tret del router) ha estat virtualitzat.

8 CONCLUSIONS

Les conclusions del treball estan dividides segons la funció; és a dir, el sistema de monitorització, el de gestió de les incidències i el de gestió de credencials. També es farà una menció a la utilitat d'haver virtualitzat l'entorn.

- **Sistema de monitorització**

Pel que fa referència al sistema de monitorització m'agradaria destacar que, per una banda, tot el que s'ha implementat ha estat completament gratuït i que, per tant, disposar d'una eina que informi de l'estat de la nostra empresa/organització es pot fer sense cap cost de software ni de llicències. D'altra banda, un dels problemes principals que té la versió gratuïta és que la interfície web no disposa d'una secció per tal de poder afegir hosts, serveis, contactes… a la monitorització, fent que sigui necessari tractar amb fitxers de configuració i amb els problemes inherents a aquests. A més, un altre punt a tenir en compte és que la possibilitat d'afegir plugins/add-ons externs (gratuïts o no) fa que l'eina passi a ser molt més robusta i aporti més valor a l'empresa/organització.

És per això que amb totes aquestes raons i les comentades al llarg del treball, considero que disposar actualment d'una eina com Nagios és quelcom molt important a tenir en compte ja que té una funció que és molt necessària en el dia d'avui com és poder focalitzar els recursos allà on es troba el problema i poder estalviar temps, diners i esforços. A més, és útil per poder detectar problemes i saber on caldria invertir en cas de destinar alguna partida econòmica ja que ens pot indicar els punts que han estat més conflictius.

- **Sistema de gestió de les incidències**

En relació al sistema de gestió de incidències, m'agradaria destacar que JIRA i OTRS són dos sistemes molt emprats arreu però amb algunes diferències. Depenent d'on s'hagués d'instal·lar optaria per una opció o una altra, és a dir, si fos en una empresa que disposa de diversos clients escolliria OTRS ja que està més orientat a aquesta funció mentre que si s'hagués d'instal·lar en un empresa que està més orientada a desenvolupar projectes (de software, per exemple) o un col·legi/universitat recomanaria més emprar JIRA ja que està pensat per resoldre problemes internament.

El preu és, evidentment, un factor a tenir també en compte. JIRA és de pagament encara que per segons quins tipus d'institucions o entitats (per exemple, relacionades amb l'educació) disposa de descomptes de fins al 50%; pel que fa a OTRS és gratuït i només s'ha de pagar en cas de necessitar suport o funcionalitats afegides.

Pel que fa a la usabilitat i configuració, JIRA és la meva primera opció ja que permet modificar diagrames de flux, mostrar uns camps o uns altres en funció del tipus d'incidència i és molt intuïtiu i editable. Pel que fa a OTRS la configuració no és tan user-friendly i no permet mostrar diferents camps en funció del tipus d'incidència. Tot i això, cal recordar que OTRS és gratuït.

- **Servidor de gestió de credencials**

Tot i disposar de diferents aplicacions que podien oferir el mateix resultat que Teampass, s'ha optat per aquest ja que té diferents aspectes que són molt interessants com

ara:

- centralització
- actualització transparent
- funciona via web
- administrar l'aplicació
- no cal aplicació
- grups i rols
- validació LDAP
- còpies de seguretat

D'altres aplicacions com ara KeePass no ofereixen l'opció de l'actualització única ja que cal carregar una base de dades per disposar de les credencials i aquesta tasca no és transparent per l'usuari (pot fer servir una base de dades desactualitzada) o de serveis webs (com ara https://www.passpack.com/) que ofereixen aquesta possibilitat –que generalment són de pagament– sobre els quals no es té cap (o molt poca) capacitat d'administració del servidor.

- **Virtualització de l'entorn**

Tot i que aquest objectiu no era el principal del treball i es va afegir ja que últimament s'està optant per virtualitzar els entorns a tot arreu on conec, he pogut comprovar que és una opció molt potent per diverses raons com ara l'optimització de recursos, la portabilitat i la facilitat per fer millores sense risc a que el canvi no funcioni com s'esperava. De fet, durant el projecte, el disc dur que feia servir es va espatllar i vaig haver de comprar-ne un altre per poder seguir treballant; pel fet de virtualitzar, vaig perdre només les dades fins l'última exportació; en comptes de la màquina sencera i començar des de zero novament.

9 FUTURES LÍNIES DE TREBALL

- **Implementar el treball en un entorn real**

Aquest treball s'ha realitzat sobre un entorn fictici i, per tant, les màquines que hi trobem són poques. Per tant, de cara al futur, pot ser interessant realitzar aquest treball o quelcom semblant en un entorn real com ara una empresa, col·legi, universitat (potser en algun departament de l'UAB, si fos d'interès).

- **Comparativa entre diferents sistemes de monitorització**

Per qüestions de temps i calendari, no ha estat possible fer una comparativa amb d'altres sistemes de monitorització com podrien ser Centreon i/o Zabbix –els quals també són Open Source–; per tant, podria ser interessant comparar-los amb Nagios. Seria quelcom semblant al que s'ha fet amb JIRA i OTRS.

- **Afegir nous plugins/add-ons a Nagios**

Si es volgués fer que Nagios fos encara més complet, es podria tractar d'afegir més plugins/add-ons per incloure més funcionalitats. Alguns d'ells podrien ser NagVis o NDOUtils. Destacar que si s'optés per fer servir Centreon, el plugin NagVis seria també compatible.

- **Afegir un sistema de gestió de documentació**

Nagios permet afegir procediments quan salta una alerta. Aquests procediments podrien estar documentats en algun sistema de gestió de la documentació com pot ser una Wiki on es pogués disposar d'informació referent a quina actuació dur a terme, a qui avisar en cas de que no es resolgui, o si hi ha quelcom més a tenir en compte…

- **Enviament de les alertes per altres vies**

JIRA i OTRS poden ser configurats per enviar alertes via SMS (fent ús de complements); pot ser interessant explorar aquesta via.

AGRAÏMENTS

M'agradaria agrair a en Ramon haver estat el meu tutor per haver-me orientat i comentat els punts que em feien dubtar; per indicar-me el calendari d'entregues de documents i per resoldre'm dubtes fora del seu horari laboral.

També, a l'Albert i en Carlos, per haver estat els companys de pràctiques i d'estudi amb els quals hem estat hores durant tota la carrera.

I, per últim, a la meva família per haver fet possible que anés a l'universitat per estudiar el que volia desde petit.

BIBLIOGRAFIA

[1] COMUNITAT NAGIOS. "Install Nagios Core From Source". [en línia]. Enllaç web: http://assets.nagios.com/downloads/nagioscore/docs/Installing_Nagios_Core_From_Source.pdf [última consulta: 30/gener/2014]

[2] COMUNITAT NAGIOS. "Nagios: NRPE Documentation". [en línia]. Enllaç web: http://nagios.sourceforge.net/docs/nrpe/NRPE.pdf [última consulta: 30/gener/2014]

[3] COMUNITAT NSCLIENT++; "Installation Documentation" [en línia]. Enllaç web: http://www.nsclient.org/nscp/wiki/doc/installation [última consulta: 30/gener/2014]

[4] COMUNITAT NAGIOS. "Object Inheritance". [en línia]. Enllaç web: http://nagios.sourceforge.net/docs/3_0/objectinheritance.html [última consulta: 30/gener/2014]

[5] COMUNITAT NAGIOS. "Object Definitions". [en línia]. Enllaç web: http://nagios.sourceforge.net/docs/3_0/objectdefinitions.html [última consulta: 30/gener/2014]

[6] COMUNITAT NAGIOS. "Standard Macros in Nagios". [en línia]. Enllaç web: http://nagios.sourceforge.net/docs/3_0/macrolist.html [última consulta: 30/gener/2014]

[7] NAGIOS PLUGINS DEVELOPMENT TEAM. The check_ping Plugin. [en línia]. Enllaç web: http://www.nagios-plugins.org/doc/man/check_ping.html [última consulta: 30/gener/2014]

[8] ATLASSIAN. JIRA Installation and Upgrade Guide [en línia]. Enllaç web: https://confluence.atlassian.com/display/JIRA061/JIRA+Installation+and+Upgrade+Guide [última con-

sulta: 30/gener/2014]
[9] OTRS Open Technology Real Services. Documentation OTRS 3.3 – Admin Manual. [en línia]. http://doc.otrs.org/3.3/en/html/installation.html [última consulta: 30/gener/2014]
[10] LAUMAILLÉ, Nils. "Installation & Update | Teampass" [en línia]. Enllaç web: http://www.teampass.net/category/installation-and-update/ [última consulta: 30/gener/2014]

APÈNDIX

A1. SECCIÓ D'APÈNDIX

Per permetre que el Firewall permeti l'enrutament dels datagrames que li arriben, cal habilitar el Forwarding i l'ús NAT via iptables.

A l'script següent s'executen aquestes instruccions de forma automàtica a l'iniciar el Firewall:

```
if [ $(cat /proc/sys/net/ipv4/ip_forward) = '1' ]; then
   echo "- Forwarding: OK"
else
   echo -n "- Forwarding: KO"
   echo "1" > /proc/sys/net/ipv4/ip_forward;
   if [ $(cat /proc/sys/net/ipv4/ip_forward) = '1' ]; then
        echo -e "\n- Forwarding: OK"
   else
         echo -e "\n- Forwarding: KO"
    fi
fi

iptables -t nat -F
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to
$(ip -4 addr show eth0 | tail -1 | cut -d "t" -f 2 |
cut -d "/" -f 1);

if [ $(iptables -L -t nat | grep $(ip -4 addr show eth0
| tail -1 | cut -d "t" -f 2 | cut -d "/" -f 1) | grep
SNAT | wc -l) -eq 1 ]; then
   echo -e "- SNAT: OK"
else
   echo -e "- SNAT: KO"
fi
```

A2. ENVIAMENT DE MAILS

Perquè Nagios pugui enviar els mails, cal definir un servidor SMTP; en el nostre cas farem ús de smtp.gmail.com. A més, caldrà també un usuari i contrasenya perquè l'enviament es pugui fer efectiu. Amb aquestes dades editarem el fitxer de configuració ubicat a `/usr/local/nagios/etc/resource.cfg` afegint la següent informació:

```
$USER10$=usuari@domini.tld
$USER11$=smtp.gmail.com
$USER12$=usuari
$USER13$=password
```

Aquesta informació servirà perquè Nagios la faci servir com a macro i no calgui escriure-la a tots els punts on sigui necessari; només haurem de posar `$USERX$` i Nagios entendrà que és el contingut que hi hagi en aquest usuari fins a un límit de 32 usuaris.

Un cop definits aquests camps, modificarem el fitxer `/usr/local/nagios/etc/objects/commands.cfg` per tal que rebi la informació anterior i així pugui enviar les notificacions. Existeixen dos tipus de notificacions: una que informa dels hosts (`notify-host-by-email`) i una altra dels serveis (`notify-service-by-email`).

El fitxer `commands.cfg` hauria de contenir la següent informació:

```
#'notify-host-by-email' command definition
define command{
  command_name notify-host-by-email
  command_line /usr/bin/printf "%b" "***** Nagios
*****\n\nNotification Type: $NOTIFICATIONTYPE$\nHost:
$HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOS-
TADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONG-
DATETIME$\n" | /usr/local/bin/sendEmail -s $USER11$ -
xu $USER12$ -xp $USER13$ -t $CONTACTEMAIL$ -f $US-
ER10$ -l /var/log/sendEmail -u "** $NOTIFICATIONTYPE$
Host Alert: $HOSTNAME$ is $HOSTSTATE$ **" -m "*****
Nagios *****\n\nNotification Type: $NOTIFICATION-
TYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress:
$HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time:
$LONGDATETIME$\n" }

#'notify-service-by-email' command definition
define command{
  command_name notify-service-by-email
  command_line /usr/bin/printf "%b" "***** Nagios
*****\n\nNotification Type: $NOTIFICATION-
TYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALI-
AS$\nAddress: $HOSTADDRESS$\nState: $SER-
VICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional
Info:\n\n$SERVICEOUTPUT$" | /usr/local/bin/sendEmail
-s $USER11$ -xu $USER12$ -xp $USER13$ -t $CON-
TACTEMAIL$ -f $USER10$ -l /var/log/sendEmail -u "**
$NOTIFICATIONTYPE$ Service Alert: $HOSTALI-
AS$/$SERVICEDESC$ is $SERVICESTATE$ **" -m "*****
Nagios *****\n\nNotification Type: $NOTIFICATION-
TYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALI-
AS$\nAddress: $HOSTADDRESS$\nState: $SER-
VICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional
Info:\n\n$SERVICEOUTPUT$" }
```

Com es pot veure, hi ha moltes dades que estan entre símbos de $. Això indica que són macros que Nagios entén com per exemple, `HOSTADDRESS` que indica la IP del host o `HOSTNAME` pel nom del host [6].

El contingut de la notificació és completament editable i es pot incloure la informació que es consideri necessària.

A3. CHECKS PROPIS

A continuació es mostra un check propi creat per obtenir la memòria RAM ocupada en Debian. Els llindars estan escrits dintre del check, motiu pel qual no accepta paràmetres.

- [0, 79] = Ok
- [80, 89] = Warning
- [90, 100] = Critical

```
#!/bin/bash

space=`free -m | grep Mem: | awk '{print $3/$2*100}' |
cut -d "." -f 1`
if ((0<=$space && $space<=79))
then
        echo "OK - Ocupat: $space%"
        exit 0
elif ((80<=$space && $space<=89))
then
        echo "Warning - Ocupat: $space%"
        exit 1
elif ((90<=$space && $space<=100))
then
        echo "CRITICAL - Ocupat: $space%"
        exit 2
else
        echo "UNKNOWN - Revisa la configuracio"
        exit 3
fi
```