

PROCEDIMIENTO SVN

|  |  |
| --- | --- |
|  | Proyecto: [Optimización] |
| CITI |

Confidencialidad del documento.

Todos los derechos reservados. El contenido de este documento es propiedad de CITI. Estrictamente prohibido cualquier reproducción total o parcial sin el permiso escrito de CITI.

Este documento está sujeto a cambios. Comentarios, correcciones o preguntas acerca del mismo, debe de ser dirigidas al autor.

HISTORIAL DE REVISIONES

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FECHA | | VER. | | DESCRIPCIÓN | | AUTOR | | REQ. | | Aprobado por: |  |
| 9/06/15 | | v1.0 | | Aprobación de Documento | | Cristhian Méndez | | CMMI | | NA |  |
| 10/06/15 | | v1.1 | | Aprobación de Documento | | Cristhian Méndez | | CMMI | | Carlos Zuñiga |  |
| 22/02/2016 | | V1.2 | | Se agregó los pasos para creación de TAG en SVN | | Cristhian Méndez | | CMMi | | NA |  |
| 16/03/2016 | | V3.9 | | Se documenta mejora Integración de BitBucket como repositorio a los procesos operativos Procedimientos de repositorio. | | Cristhian Méndez | | Mejora Continua | | N/A |  |
|  |  | |  | |  | |  | |  | |  |
|  |  | |  | |  | |  | |  | |  |

TABLA DE CONTENIDO

HISTORIAL DE REVISIONES 2

TABLA DE CONTENIDO 3

1. Procedimientos SVN 4

1.1. Procedimiento de creación de nueva rama de código. 4

1.2. Mezclar de nuevos cambios del trunk al branch 4

1.3. Pasar cambios de un branch al trunk. 4

1.4. Crear repositorio 5

1.5. Crear estructura de carpetas 6

1.6. Permisos 7

1.7. Proceso de Commit 7

1.8. Agregar usuarios a Subversion 7

1.9. Crear Respaldos y recuperación de archivos 8

1.10. Creación de TAG SVN 9

1.11. Procedimientos BitBucket 10

1. Procedimientos SVN
2. Procedimiento de creación de nueva rama de código.

Todas las ramas (branchs) se deberían crear a partir de una Rama principal (trunk)

* Ir a la ventana "SVN Repositories" > Trunk > New > Branche (con el botón derecho)
* Configuráis el nombre de la rama, por defecto lo pondrá en el directorio Branches.

1. Mezclar de nuevos cambios del trunk al branch

* Actualiza tu copia de trabajo a la última versión del branch
* Averigua el número de revisión del último merge (si todavía no se hizo ninguno, el número de revisión de creación, que, si sigues las indicaciones deberías tener en alguno de los comentarios)
* Utiliza la herramienta de merge de subclipse para mezclar el trunk en tu copia de trabajo (Team -> Merge)
* Las URL‘s que aparecen en “From” y en “To” deberían ser las del trunk
* La revisión del “From” debería ser el número de revisión del último merge
* La revisión del “To” debería ser “Head“
* Pulsa en el botón Merge y espera a que termine
* En la consola se muestra un log del proceso realizado. Si aparece algún conflicto (marcado con una C roja, lo que hay que hacer es localizar dichos ficheros en el navegador de archivos, botón derecho y pulsar en Team -> Edit Conflicts, lo que abrirá un nuevo editor con dos paneles con una versión en cada lado. Se resolverán los conflictos (como se realiza normalmente al subir cambios al SVN) y, una vez corregidos, se vuelve al navegador de archivos, sobre el mismo y se pulsa Team -> Mark as Resolved.
* Probar todo una vez mezclado :)
* Subir los cambios de la mezcla a la rama.

1. Pasar cambios de un branch al trunk.

* Actualiza tu copia de trabajo a la última versión del trunk
* Averigua el número de revisión del último merge (si todavía no se hizo ninguno, el número de revisión de creación del branch, debería aparecer en los comentarios)
* Utiliza la herramienta de merge de subclipse para mezclar el branch en tu copia de trabajo (Team -> Merge)
* Las URL‘s que aparecen en “From” y en “To” deberían ser las del branch
* La revisión del “From” debería ser el número de revisión del último merge
* La revisión del “To” debería ser “Head“
* Pulsa en el botón Merge y espera a que termine
* En la consola se muestra un log del proceso realizado. No debería aparecer ningún conflicto ya que se supone que el branch contiene lo último deltrunk mas sus propios cambios.
* Probar todo una vez mezclado :)
* Subir los cambios de la mezcla al trunk.

1. Crear repositorio

* Se localiza el fichero dav\_svn.conf. Lo vamos a editar para hacer algunas modificaciones para crear un repositorio llamado pruebas.

Localizamos las lines donde se define el repositorio por defecto SVN.

# <Location URL> ... </Location>

# URL controls how the repository appears to the outside world.

# In this example clients access the repository as <http://hostname/svn/>

#<Location /svn> %modificamos la linea y cambiamos el nombre de repositorio pruebas.

# <Location URL> ... </Location>

# URL controls how the repository appears to the outside world.

# In this example clients access the repository as <http://hostname/svn/>

<Location /pruebas>

* Tal y como indica, des comentamos para activar el repositorio:

# Uncomment this to enable the repository,

# DAV svn

* Asi

# Uncomment this to enable the repository,

#DAV svn

* Modificamos también las líneas:

# Set this to the path to your repository

# SVNPath /var/lib/svn

* Para establecer el path del repositorio:

# Set this to the path to your repository

SVNPath /var/lib/svn/pruebas

* Inicialmente comentaremos las siguientes líneas para desactivar la seguridad:

<LimitExcept GET PROPFIND OPTIONS REPORT>

Require valid-user

</LimitExcept>

* No olvidemos descomentar la última línea. Ya que es donde termina el repositorio.

</Location>

* Como somos muy ordenados, queremos crear un repositorio por cada proyecto. Así será más sencillo gestionar las políticas de seguridad. Todos estos repositorios se van a crear dentro del directorio /var/lib/svn por lo que tenemos que crearlo previamente.

$ sudo mkdir /var/lib/svn

* Ahora tenemos que crear el repositorio en sí:

$ sudo svnadmin create /var/lib/svn/pruebas

* Sólo nos queda reiniciar el servidor Apache para que vuelva a leer la configuración actualizada:

$ sudo /etc/init.d/apache2 restart.

1. Crear estructura de carpetas

* Para crear el primer usuario, ejecute (cambie <usuario> por el nombre de usuario a crear) :sudo htpasswd -c /etc/subversion/passwd <usuario>
* A continuacion suministre la clave asignada al usuario
* Para crear el segundo utilice el mismo comando pero sin la opcion -c  
  sudo htpasswd /etc/subversion/passwd <usuario>.

1. Permisos

* Para añadir los usuarios con permisos de acceso a nuestro repositorio los añadimos al grupo svn de la siguiente forma:

usermod -G svn juan

* Para que el subversion pueda gestionar las credenciales de los usuarios creamos el fichero /etc/apache2/dav\_svn.passwd que anteriormente habilitamos en el fichero de configuración del repositorio dav\_svn.conf (nos pedirá introducir la contraseña para el usuario):

1. htpasswd -c /etc/apache2/dav\_svn.passwd juan
2. New password:
3. Re-type new password:
4. Adding password for user juan.
5. Proceso de Commit

Cuando se crean archivos o carpetas que no existían en la copia bajada del repositorio, en necesario agregarlas al control de versiones. Si corremos el subcomando *status* después de haber creado archivos nuevos, aparecerán precedidos por un carácter?, indicando que son archivos o carpetas desconocidas. Para agregar un archivo o una carpeta y todo su contenido al control de versiones se usa el subcomando *add*

~/svn/svn$ svn add archivo.src carpeta/

Cuando se agregan directorios al control de versiones, se ignorarán los archivos terminados en *~*, ya que se asume que son backups. Los archivos serán agregados al repositorio en el siguiente commit.

1. Agregar usuarios a Subversion

Actualmente existen varios esquemas de autentificación para los repositorios de Subversion, desde el uso de un archivo de passwords (a la manera del clásico /etc/passwd) a comunicación con servidores LDAP.

En el caso más simple de autentificación utilizamos el modelo de seguridad del servidor Apache, considerando que Subversion en si mismo funciona directamente como un módulo del servidor Web, facilitando el acceso. Por lo tanto, en la configuración del directorio virtual de svn encontramos algo como lo siguiente:

<Location /svn/Pruebas>  
DAV svn  
SVNParentPath /var/svn/Pruebas  
AuthType Basic  
AuthName "Repositorio Subversion de pruebas"  
AuthUserFile /etc/svn-auth  
Require valid-user  
</Location>

En este extracto del archivo subversion.conf vemos que estamos utilizando autentificación básica, y que el archivo sobre el cual debemos autentificar es /etc/svn-auth.

Para crear un usuario o modificar su contraseña, ejecutamos el siguiente comando:

htpasswd /etc/svn-auth usuario

Donde usuario es el login que queremos crear o cambiar. A continuación htpasswd preguntará dos veces la contraseña, y con eso el usuario ya tiene acceso al repositorio. Este último está configurado de modo que el usuario apache puede entrar y modificar contenidos (por supuesto autentificando antes), por lo que no necesitamos más cambios para terminar esta configuración.

1. Crear Respaldos y recuperación de archivos

Los respaldos en la empresa se realizan en automático los días lunes de cada semana mientras que la recuperación se realiza a través de un disco duro en el cual se almacenan estos.

Los respaldos se realizan utilizando Cron en Linux Ubuntu es un administrador regular de proceso en segundo plano que ejecuta procesos o guiones a intervalos regulares por ejemplo cada minuto, semana, mes o en nuestro caso los días lunes de todas las semanas.

El funcionamiento de Cron inicia desde /etc/rc.d o /etc/init.d dependiendo de la distribución de Linux de cada servidor en la empresa, este cron revisa cada minuto de la tabla tareas de cron que se encuentra en /etc/crontab o /var/spool/cron y deberá cumplir las tareas ahí descritas, crontab es un simple archivo de texto que guarda una lista de comandos a ejecutar en un tiempo especificado el archivo de texto o script que se utiliza para respaldar los archivos es el siguiente.

#!/bin/sh

#Backup Repositorio

#CITI2015

#Version 1.0

TIME=`date +%d-%m-%y`

FILENAME=rep-backup-$TIME.tar.gz

SRCHDIC=/usr/local/svn

RESPDIC=/home/citiadmin/Backup2015

tar -cpzf $RESPDIC/$FILENAME $SRCHDIC

Este ya es un archivo definido por sistemas internos de CITI y tiene como extencion archivo.sh

Una vez teniendo esto se agregaran las tareas a Crontab.

Ejecutamos la edición del crontab con el siguiente comando. Crontab –e, apareciéndoles lo siguiente:

# m h dom mon dow user command

Donde:

* m corresponde al minuto en que se va a ejecutar el script, el valor va de 0 a 59
* h la hora exacta, se maneja el formato de 24 horas, los valores van de 0 a 23, siendo 0 las 12:00 de la medianoche.
* dom hace referencia al día del mes, por ejemplo se puede especificar 15 si se quiere ejecutar cada dia 15
* dow significa el día de la semana, puede ser numérico (0 a 7, donde 0 y 7 son domingo) o las 3 primeras letras del día en inglés: mon, tue, wed, thu, fri, sat, sun.
* user define el usuario que va a ejecutar el comando, puede ser root, u otro usuario diferente siempre y cuando tenga permisos de ejecución del script.
* command refiere al comando o a la ruta absoluta del script a ejecutar, ejemplo:/home/usuario/scripts/actualizar.sh, si acaso llama a un script este debe ser ejecutable

El cron tab de la empresa esta configurado de la siguiente manera.

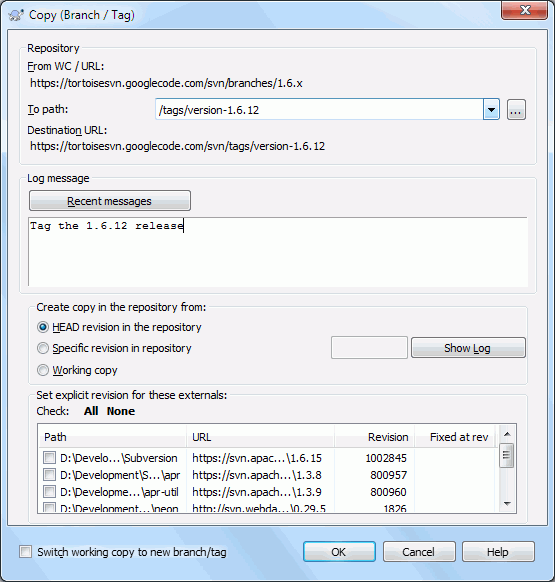
00 00 \* \* 1 root apt-get –y update Usuario root

Recuperación de archivos.

En caso de alguna falla en el servidor se tomará el ultimo archivo comprimido que se respaldó, esta recuperación se realizará conectando el disco de almacenamiento al nuevo servidor, la copia de los nuevos archivos al nuevo servidor donde serán alojados los datos/información se realiza de la siguiente manera.

1. Primero que nada tendremos que ver el disco de almacenamiento en la consola del server y esto se realiza con el siguiente comando df- h aquí podrás ver todos los duros y particiones en tu servidor.
2. Una vez tengamos identificado el disco duro en el cual está alojado nuestro respaldo se tendrá que realizar la copia de este en el nuevo servidor cp [fichero a copiar] [ruta donde se copiara] (Sin los []).
3. Creación de TAG SVN

Seleccione la carpeta en su copia de trabajo de la que desea hacer una rama o una etiqueta, y luego seleccione el comando TortoiseSVN → Rama/Etiqueta.... posteriormente selecciones “Ok”

Eje.

1. Procedimientos BitBucket

***Dar clic para ver tutorial***