Análisis, diseño, implementación de un sistema para la automatización de despliegues de clientes de gestión de backups IBM Tivoli Storage Manager

Javier Méndez Leiva

Resumen—En este proyecto nos centraremos en las instalaciones y actualizaciones de software en máquinas de clientes del Software IBM TSM. Este software es un aplicativo desarrollado desde IBM para centralizar y coordinar las acciones pertinentes a la ejecución, administración y realización de backups.

La criticalidad del proyecto viene dado del enorme entorno al que afecta (1000 máquinas) y del hecho de ser llevado a cabo en una empresa significativa del sector como es NTT Data. Por tal de encajar en los marcos universitarios el proyecto mostrado en este artículo será una porción del mismo (60 máquinas) y tomaremos en cuenta la parte inicial de desarrollo del automatismo necesario para efectuar estos despliegues.

Para llevar a cabo estos despliegues haremos uso de otra tecnología llamada Ansible Tower que se encuentra actualmente integrada en la red del sistema y nos permitirá, con la debida autorización, efectuar los despliegues en volúmenes de múltiples máquinas simultáneamente. Haciendo uso de una template de Python y YAML como código de ejecución.

Palabras clave—Despliegue, NTT Data, Backup, Ansible Tower, IBM TSM, Versión, template.

1 Introducción - Contexto del trabajo

El trabajo de final de grado que se describe se desarrolla en un entorno muy exigente, donde el cliente es un importante miembro del sector bancario. Debido a esta situación, el proyecto presenta numerosas complicaciones en su desarrollo y avance.

El sector bancario es uno de los más regulados y complejos del mundo empresarial, lo cual implica que las soluciones tecnológicas que se implementen deben cumplir con una serie de requisitos y estándares muy específicos. Además, se espera que se cumplan altas expectativas en cuanto a la calidad, la seguridad y la eficiencia de los procesos que se implementen.

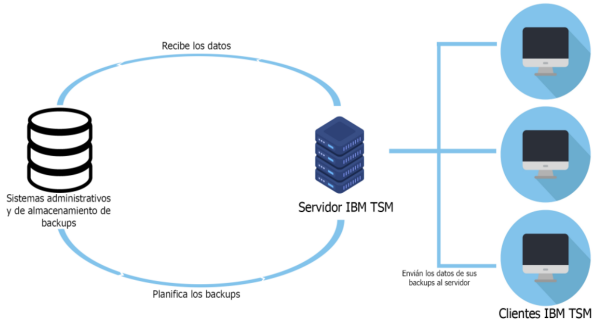
En este contexto, es fundamental que la persona encargada tenga una sólida comprensión de las necesidades y requerimientos, así como de las particularidades del sector bancario. Es necesario que se diseñen y desarrollen soluciones eficientes que cumplan con los estándares requeridos y que se adapten a las necesidades específicas.

1.1 Estado del arte

Para ubicarnos debemos entender que este proyecto ha sido llevado a cabo para la empresa NTT Data dentro del grupo tecnológico de Storage y Backups. Desde este grupo, trabajamos para asegurar que las copias de seguridad de diferentes clientes se hagan de forma efectiva mientras trabajamos en mejorar su almacenamiento.

Dentro de este departamento contamos con una gran variedad de tecnologías, en este caso una de ellas actuando como intermediario de los backups. Esta tecnología es el IBM Spectrum Protect, el cual centraliza la información de los backups y permite dirigir los flujos de datos de los mismos desde una máquina cliente a una máquina servidor para su almacenaje..

Figura 1.1 Diagrama que detalla el funcionamiento de IBM TSM



La situación de este producto, requiere que su otro producto, IBM Tivoli Storage Manager, se encuentre instalado en la máquina que gestiona la planificación y ejecución del backup y adicionalmente, en la máquina donde se realiza este mismo.

En este proyecto nos centraremos en las instalaciones de cliente, es decir las pertenecientes a las máquinas donde se realizan estos mismos.

Para dar más situación, debemos entender que hay un gran volúmen de proyectos actualmente que tienen máquinas con este software ya instalado, pero a más antiguo es el proyecto, más posible es que tengamos una versión muy antigua de este programa instalado, la finalidad en esta parte es conseguir que estas máquinas tengan un versionado soportado por parte del distribuidor o reciente.

Por otro lado, otro foco de atención es conseguir que las nuevas máquinas, que generamos se establezcan con una versión reciente y actual de este software.

A su mismo tiempo, también hay que asegurar que los backups se hagan correctamente dentro de las máquinas sobre las que actuemos. Puesto que, en el primero de los casos mencionados anteriormente, tenemos una gran cantidad de backups fallados no tratados por la volumetría interna de trabajo. De modo que regularizar estos equipos es también parte de los objetivos.

Finalmente, también se quiere investigar la mejora de rendimiento, puesto que en algunos casos estamos hablando de backups de un gran volúmen (30-50 Terabytes diarios) y puede ser un estímulo mucho más grande por tal de promover la actualización del software de cara a la mejora de rendimiento de estos backups.

1.2 Objetivos

En vista de desglosar los diferentes objetivos del proyecto, utilizaremos la siguiente tabla:

Tabla 1.1 Objetivos del proyecto

NÚMERO DE OBJETIVO	PRIORIDAD
Generar un automatismo para el despliegue del *software TSM IBM*.	Crítico
Detectar potenciales problemáticas para la implementación del *software*.	Crítico
Garantizar el correcto funcionamiento del *software* desplegado.	Prioritario
Desplegar el software en 4 sistemas diferentes. (60 máquinas)	Prioritario
Simplificar el despliegue para usuarios con conocimientos superficiales de estas tecnologías.	Segundario

De los objetivos previamente mostrados cabe puntualizar una serie de ideas sobre los mismos.

Cabe entender, que al garantizar el funcionamiento del software desplegado, no solo interpretamos asegurarnos de que este funcione según la verificación al instalar sino también incluímos asegurar que los backups funcionen en los días consecutivos.

Finalmente, el último objetivo, realmente pasa por proveer este automatismo al grupo de aprovisionamiento, el cual no muestra un gran grado de conocimiento sobre este programa y realmente solo gestionan acciones de instalación, permitiendo acciones más ágiles que las nuestras a la hora de realizar despliegues.

2 Metodología

En lo que refiere a metodologías, cabe remarcar que lo llevamos a cabo dentro del Kanban integrado en el proyecto general del departamento y teniendo en cuenta siempre los indicadores establecidos por el modelo de gestión administrativa ITIL.

Este modelo permite mantener el flujo asiduo de trabajo y delimitar las tareas para conocer el progreso de las mismas y el alcance de las que se deberán realizar en determinadas iteraciones.

Para gestionar el Kanban se utilizó la herramienta adjunta al propio Microsoft Teams y se efectuaron las reuniones de seguimiento semanal y de definición de backlog, mientras que para aligerar carga técnica se omitirá el seguimiento diario y las reuniones diarias de seguimiento del mismo.

Adicionalmente y de forma paralela se hizo uso también del calendario integrado en Microsoft Teams por tal de facilitar el seguimiento de reuniones y acciones en fechas concretas.

De forma paralela y debido a que estos datos no pueden ser compartidos por cuestiones obvias de confidencialidad en la empresa, se predispuso un Trello por tal de facilitar el seguimiento externo del proyecto y de acceso público, el cual puede ser encontrado en la bibliografía del proyecto.

Cabe definir que después de una breve propuesta y análisis inicial, se definió que el proyecto seguiría un ciclo iterativo, de modo que las fases se llevarían a cabo con el siguiente patrón, implicando que aunque una fase se concluya es posible que se vuelva en un futuro para trabajar de nuevo en nuevas casuísticas detectadas.

Figura 2.1 Diagrama exponiendo el ciclo iterativo del proyecto

Es en parte esta naturaleza iterativa, la que en muchas ocasiones dificulta dentro del proyecto estimar tiempos para algunas tareas, puesto que en base a los problemas que se encontraran, supondría un severo incremento en el tiempo y las acciones a efectuar.

2.2. Descomposición de tareas

Debido a la estructura del proyecto por tal de definir las tareas en los Sprints nos vismo obligados a definir un diagrama de Gantt que estimaría las horas de las tareas y a su vez nos serviría para trabajar la volumetría de los Sprints.

Inicialmente las tareas especificadas fueron los siguientes y con los siguientes plazos:

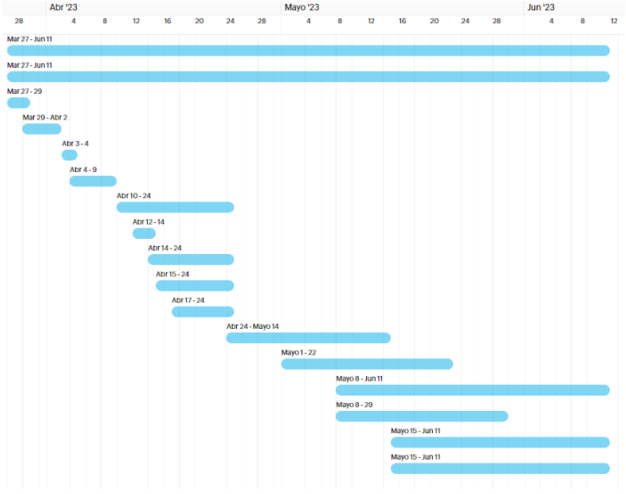
Tabla 2.1 Tareas del proyecto

Nombre de la tarea	Fecha de inicio	Fecha de vencimiento
Diseño funcional del código	04-04-2023	04-09-2023
Implementación del código	04-10-2023	04-17-2023
Creación del entorno de programación	04-12-2023	04-14-2023
Integración en entorno de Ansible Tower	04-17-2023	04-24-2023
Analizar preexistencias	03-29-2023	04-02-2023
Contactar con el proveedor para obtener información	03-27-2023	03-29-2023
Detectar máquinas afectadas	04-03-2023	04-04-2023
Análisis de problemas encontrados durante despliegues	03-27-2023	06-11-2023
Pruebas funcionales del automatismo	04-14-2023	04-17-2023
Pruebas de integración del automatismo	04-15-2023	04-17-2023
Mantenimiento continuado y control del servicio	03-27-2023	06-11-2023
Entablar comunicación con departamento	04-24-2023	04-30-2023
Establecer fecha para el despliegue	05-01-2023	05-14-2023
Desplegar el software sobre las máquinas	05-08-2023	05-22-2023
Integrar código dentro del ciclo de aprovisionamiento	05-08-2023	05-22-2023
Dejar que el equipo de aprovisionamiento haga uso del automatismo	05-15-2023	06-04-2023
Documentación interna del automatismo	05-15-2023	06-04-2023

Esta planificación inicial se vio afectada propiciando un retraso significativo de unas aproximadamente 45 horas de trabajo debido a un problema encontrado durante el tercer Sprint que posteriormente será mencionado.

Esto puede verse reflejado en el diagrama previamente mencionado:

Figura 2.2 Diagrama de Gantt representando las tareas ordenadas como en la Tabla 1.2



3 ANÁLISIS INICIAL

IBM Tivoli Storage Manager, es un software de centralización de envío y recuperación de datos que permite centralizar los datos, controlar la organización de los mismos, planificar y coordinar las tareas de realización de copias de seguridad.

A nivel de acciones este software permite definir un servidor local en el cliente, este establece una conexión a un servidor mediante la definición de un fichero llamado dsm.sys, donde situamos los parámetros de conectividad con una máquina virtual o instancia que actúe como servido.

El nombre del servidor cliente que se genere se definirá en este fichero y nos referiremos a este como nodo TSM.

Figura 3.1 Detalle del fichero dsm.sys

```
[PRE][          ].      :~ # cat /opt/tivoli/tsm/client/ba/bin/dsm.sys
************************************
*       NODO DE SISTEMA OPERATIVO      *
************************************

SErvername           SO
   COMMmethod        TCPip
   TCPPort           [    ]
   TCPServeraddress  [          ]
   Nodename          [          ]
   Passwordaccess    Generate
   Virtualmountpoint /opt/tivoli/tsm
   Virtualmountpoint /dev
   Virtualmountpoint /
   Virtualmountpoint /var
   Virtualmountpoint /usr
   Virtualmountpoint /boot
   Virtualmountpoint /root
   Passworddir       [          ]
```

Al mismo tiempo desde este servidor se pueden observar las acciones realizadas por los backups, acceder a logs sobre lo que se ha realizado, enviar los datos a otro servidor que centralice y almacene la información de los mismos y recuperar los datos remotamente desde el cliente con facilidad.

Por otro lado desde el servidor centralizado se pueden gestionar todos los datos y los almacenamientos de los mismos, coordinando las expiraciones de los datos y las recuperaciones de las mismas con los clientes y dirigiendo al mismo tiempo los backups que se realicen sobre los sistemas de clientes.

Este producto cuenta con un soporte dirigido a determinadas versiones del propio software y de diferentes sistemas operativos que nos asegura que recibamos asesoramiento por parte del proveedor a la hora de detectar un problema. Por ende, el objetivo central pasa por obtener una versión que tenga soporte de sistema operativo.

Las compatibilidades por sistema operativo son extensas y relevantes para este proyecto y se pueden encontrar en la bibliografía[]**sadsadasd**

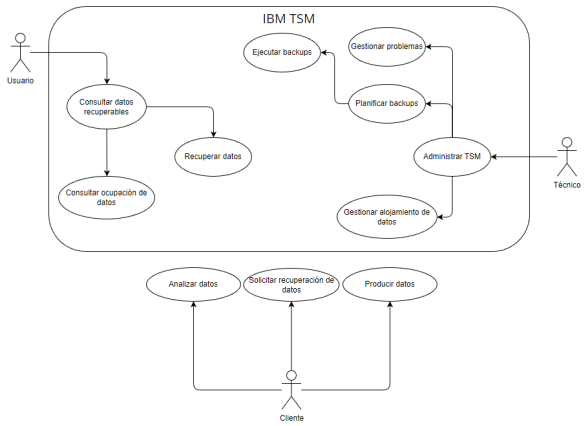
Por otro lado, hace falta entender que es Ansible tower. Ansible Tower es un sistema respaldado por la arquitectura RedHat para hacer distribuciones a los diferentes componentes de un inventario de máquinas de modo automático y simultáneo.

3.1. Perspectiva usuario cliente

A nivel de usuario encontrarán mejoras en cuanto a la velocidad y agilidad del soporte que podamos ofrecerles y también a la hora de recuperar datos.

También del mismo modo deberíamos poder minimizar la volumetría de los errores encontrados en los backups de estas máquinas al contar con un software más reciente.

Figura 3.2 Diagrama de casos de uso del software a desplegar



Para entender cómo se administran los errores debo explicar otra de las herramientas de IBM, llamada IBM Spectrum Protect, esta herramienta permite conocer y verificar el estado de las ejecuciones de los scripts de backups llevados a cabo por la herramienta IBM TSM y nos permite contabilizar la cantidad de errores encontrados de forma diaria, del mismo modo también permite administrar nuevas ejecuciones o incluso filtrar las ejecuciones de determinadas máquinas.

Desde esta herramienta podemos monitorizar que el aplicativo desde el cliente funcione correctamente y conocer qué acciones debe realizar el mismo en caso de detectar una anomalía por tal de corregir el error.

Esto nos lleva a que el cliente se encontrará con un mayor grado de monitorización y seguridad sobre sus backups asegurando que en mayor medida podamos garantizar el correcto funcionamiento de estos. Al mismo tiempo, simplificará el proceso de solución de otros errores debido a versionados obsoletos.

3.2. Diagnóstico

Finalmente, por tal de determinar nuestros equipos objetivos efectuamos un cruce entre dos inventarios existentes durante el tercer sprint, uno siendo el inventario propio de máquinas y versionados de TSM y de los backups que deberíamos tener o tenemos y, por otro lado, el inventario general de la casa.

Esto nos permitió generar un inventario propio interno que nos indica si una máquina es antigua y sigue activa y la última fecha de conectividad de la máquina a nuestro sistema de backups.

Por tal de cuantificar puedo indicar que este inventario cuenta con 22774 elementos actualmente debido a que selecciona todos los elementos existentes, incluyendo problemas ajenos al proyecto. Al filtrar por los datos que presentan las características que estamos buscando detectamos 814 máquinas afectadas por la plenitud del proyecto.

Tomando en cuenta estos datos habría falta reflejar la cuantía de máquinas que tenemos con estos sistemas operativos para tomar la referencia de la importancia de cada uno:

Figura 3.3 Gráfica de los sistemas afectados por sistema operativo

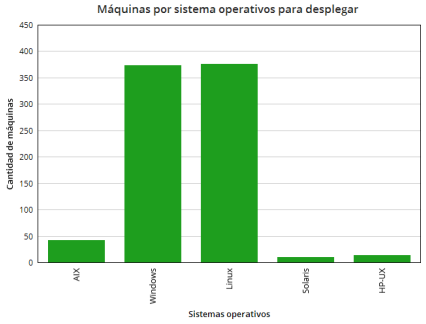


Figura 3.4 Gráfica de los sistemas operativos por versiones de tipo Windows

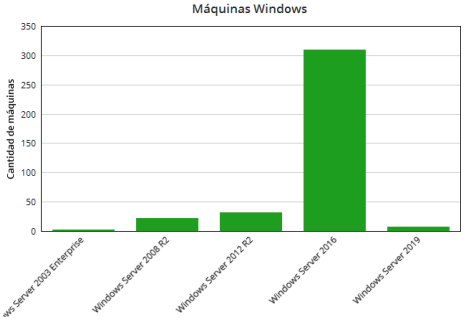
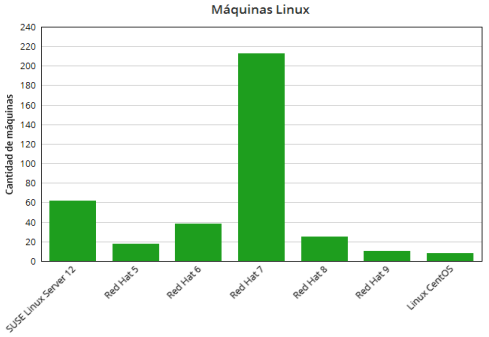


Figura 3.5 Gráfica de los sistemas operativos por versiones de tipo Linux



De las indicadiones en las anteriores gráfica son muy a tener en cuenta tanto las Windows Server 2008 las cuales no son soportadas en la versión más actual de TSM y deberán usar la versión 7.1 como, por otro lado, la existencia de máquinas con Windows Server 2003 que no tiene soporte en ningún versionado actual.

Del mismo modo en las gráficas de datos de máquinas linux encontramos un pequeño número de máquinas CentOS no soportadas que deberán ser reportadas al departamento propietario por tal de que lo solucionen y también tenemos la misma problemática con unas pocas máquinas de sistemas operativos Red Hat relativamente antiguos.

4 PRIMEROS PASOS

Para poder empezar a desarrollar el proyecto nuestro primer objetivo era no solo establecer los prerrequisitos previos ya mencionados sino también obtener un entorno desde el cual poder realizar pruebas dentro de una empresa significativamente grande y contando nosotros con recursos insuficientes inicialmente para poder establecerlo. Donde nuestro departamento no está centralizado en el desarrollo a pesar de contar con un pequeño grupo de soluciones TIC.

Nuestro primer acercamiento a este conflicto fue establecer contacto con los responsables de la tecnología Ansible Tower durante el segundo Sprint y tratar de definirles no solo el alcance del proyecto sino presentar que en nuestro caso teníamos a un integrante del grupo dispuesto a realizar la implementación necesaria a nivel de código para así poder obtener acceso a las herramientas de desarrollo a lo largo del siguiente Sprint.

Una vez efectuamos la primera reunión con ellos obtuvimos un feedback muy positivo de cara a que se hagan proyectos usando sus herramientas, puesto que son un departamento muy sobrecargado de solicitudes. Por ello, nos dieron conocimiento de una reciente iniciativa de múltiples departamentos presentando sus proyectos en una reunión semanal por tal de conseguir herramientas y proporcionar accesos a máquinas de test.

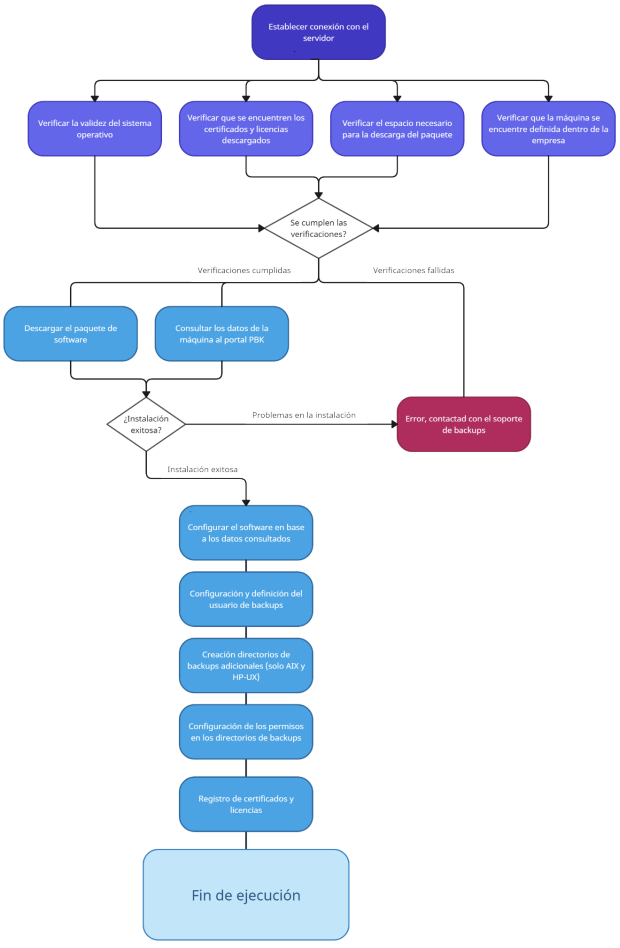
A raíz de esto conseguimos acceso a los paquetes de desarrollo y las pautas de mejores prácticas por tal de mantener el estilo de código de la empresa y tener acceso a preexistencias de automatismos y funciones ya desarrolladas en la plataforma de Ansible que posteriormente aliviaría en gran medida la carga del desarrollo

5 DISEÑO E IMPLEMENTACIÓN

Para poder efectuar un despliegue en ansible tower se requiere de una plantilla en YAML con funciones definibles en Python.

A continuación y basándonos en los datos previos presentó el esquema lógico del código que se ha desarrollado sobre este proyecto:

Figura 5.1 Diagrama de flujo de la plantilla de código



Descomponiendo el anterior esquema lógico podemos ver como el primer paso es tomar cuenta en una serie de verificaciones.

La primera es validar el sistema operativo definido en la máquina para saber qué versión debemos implementar o si este no está soportado.

En caso de que el sistema no esté soportado lanzaremos directamente un mensaje de error, y en estos casos hablaremos con las personas responsables de la máquina para que coordinen la actualización del sistema operativo de la máquina.

Acto seguido, deberemos verificar que el sistema se encuentre definido dentro de la empresa, esto es un prerrequisito interno para garantizar la integridad de la ejecución dentro de las redes de la empresa.

Siguiendo, deberemos verificar que los certificados se encuentren descargados en la máquina.

Finalmente, deberemos comprobar que el sistema cuente con suficiente espacio para gestionar el paquete de instalación descomprimido y en caso de contar con el espacio suficiente descargamos el paquete que tenemos ubicado en un repositorio de nuestro departamento.

Una vez realizado pasamos a descomprimir el paquete de instalación ejecutando el fichero binario adjunto, este paquete es de un tamaño significativo y lleva un tiempo.

Se ha de remarcar que se hace uso de una instalación por defecto y es posible que en versiones posteriores en caso de requerir configuraciones especiales se deba modificar algunos de los ficheros de configuración en base a alguna parametrización del usuario.

De forma paralela procederemos con la obtención de los datos del nodo del cliente para establecer la conectividad con el servidor, para esto haremos uso de una herramienta de desarrollo interno. Esta herramienta recibe y permite visualizar solicitudes y también gestiona múltiples de ellas de forma automatizada, una de estas solicitudes en concreto se creó con tareas de aprovisionamiento en mente y devuelve los datos correspondientes al fichero dsm.sys tomando como valores de entrada la máquina aprovisionada.

Una vez se completa la instalación, debemos verificar el correcto funcionamiento del agente de instalación, para esto se verifica la existencia de los directorios de ficheros de instalación y se prueba el comando dsmc para comprobar la interacción del software y que esté se haya definido correctamente como una variable de entorno.

Prosiguiendo en caso de que sea exitosa, deberemos efectuar la configuración del usuario para la gestión de los backups.
A nivel de gestión y administración de backups, es necesario un usuario específico dentro de los sistemas. Para evitar problemas de filtrado de información llamaremos al usuario “adminbkp”.

Antes de acabar, tendremos también que contemplar un caso adicional, en el caso de ser la máquina un sistema AIX o HP-UX se deberán generar una serie de directorios en las carpetas de la instalación para el volcado de ficheros temporales.

Finalmente, registramos cada una de las diferentes licencias en el directorio por defecto de licencias.

Figura 5.2 Prueba de conectividad de un cliente a un servidor TSM

```
[PRE][          ].       :~ # dsmadmc -se=SO
IBM Spectrum Protect
Command Line Administrative Interface - Version 8, Release 1, Level 9.0
(c) Copyright by IBM Corporation and other(s) 1990, 2019. All Rights Reserved.

Enter your user id:

Enter your password:

Session established with server       : Linux/x86_64
  Server Version 8, Release 1, Level 9.000
  Server date/time: 05/22/2023 21:08:28  Last access: 05/22/2023 19:12:56


Protect:       >
```

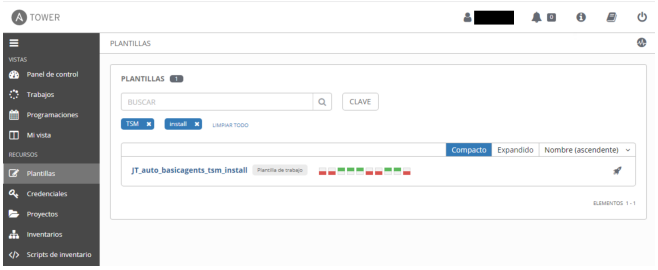
5.1. Pruebas funcionales

Para dar lugar a las pruebas funcionales del proyecto se llevaron a cabo pruebas en cada uno de los diferentes sistemas operativos diagnosticados previamente.

Se generó en un sistema preproductivo, un entorno virtual donde lanzar las pruebas y se fue modificando el sistema operativo para lanzar una ejecución completa.

El objetivo final era poder efectuar una conexión desde nuestro servidor y poder verificar que teníamos una instalación completa de la versión que buscamos sobre el servidor en cuestión.

Figura 5.3 Muestra de la plantilla desarrollada en el portal de Ansible Tower

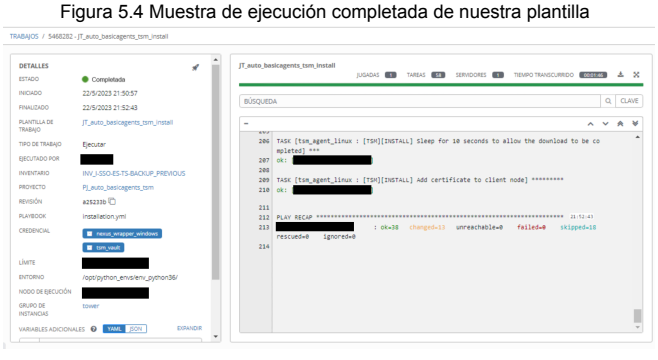


Para empezar accederemos al aplicativo de Ansible Tower y accederemos al apartado de plantillas, buscando la nuestra e indicando la ip de conexión para la máquina donde lo vamos a ejecutar. En nuestra prueba usaremos una máquina virtual de un entorno de linux.

Una vez introducidos los datos de inventario de la máquina y la DNS o IP de la misma llegamos a la ventana de ejecución donde esperaremos el feedback de la ejecución.

Una vez obtengamos el siguiente resultado, podemos verificar el correcto funcionamiento de la plantilla no solo mediante el indicador de errores de la ventana de ejecución sino contactando con un técnico de backups para que se conecte desde el servidor al nodo y verifique la versión, como muestro a continuación.

Figura 5.4 Muestra de ejecución completada de nuestra plantilla

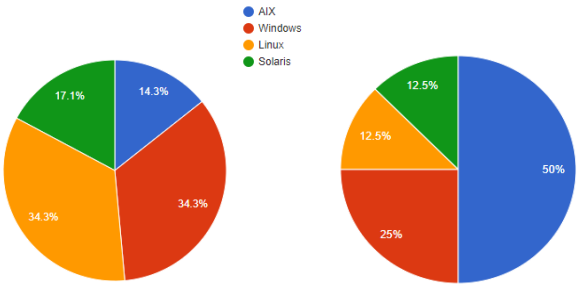


Se hicieron 43 de estas pruebas sobre exactamente 5 máquinas, cada máquina originalmente con un sistema operativo diferente, tomando como tal un Windows Enterprise Server 2016, una Windows Enterprise Server 2019, una Linux Red Hat 7.6, una máquina Solaris y una máquina AIX 7.2.

De estas 43 pruebas 7 de ellas fallaron previas a la primera revisión del código del cuarto sprint y 13 fueron

exitosas. Se detectó una problemática con los sistemas AIX que se corregiría durante la segunda iteración al rediseñar el código, el problema detectado era causado por el licenciado de los aplicativos AIX y con la configuración de los usuarios. Una vez corregidos, de las 23 pruebas restantes 22 fueron exitosas y encontramos una prueba errónea en Solaris al encontrar un problema con el directorio de incompatibilidad de software.

Gráfica 5.1 A la izquierda, gráfica de porcentaje de tests por sistemas operativos, a la derecha tasa de errores por cada sistema operativo contemplado.



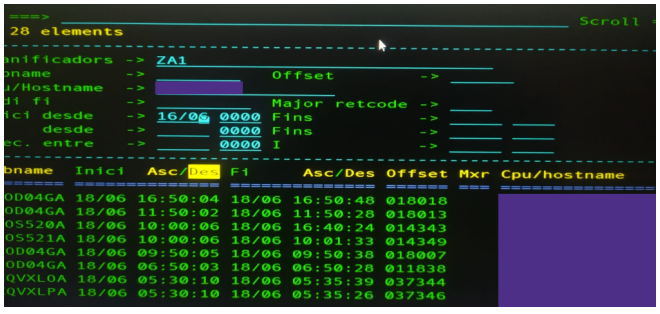
6 RESULTADOS

Por tal de garantizar control realista sobre los resultados haremos uso de la herramienta IBM Spectrum Protect para monitorizar las ejecuciones de los backups de forma diaria, cada máquina tiene una cantidad variable de backups diarios o semanales y que también se pueden lanzar backups a mano en fechas concretas, debido a esto, las ejecuciones espontáneas, también serán contempladas entre los indicadores que tomaremos.

De forma paralela también se monitorizará los tiempos de ejecución, pero en este campo seremos bastante más concretos, puesto que una gran mayoría de los backups afectados no son de un tamaño significativo para analizar una volumetría significativa de tiempo, nos centraremos en 3 casos especialmente significativos dentro de los afectados.

Es importante especificar que todos estos datos fueron extraídos del aplicativo IBM Spectrum Protect, el cual como muestro a continuación, nos muestra todos los detalles de la ejecución de los backups.

Figura 6.1 Muestra de ejecución completada de nuestra plantilla, adicionalmente se puede observar uno de los backups contemplados para la medición de tiempo con un tiempo de ejecución de aproximadamente 7 horas.



6.1. Mejora sobre el impacto de backups

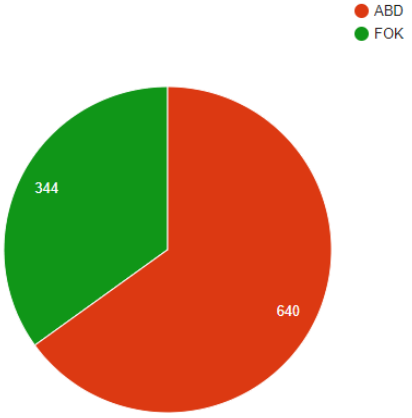
Por tal de evaluar el impacto hemos de inicialmente entender que un 40% de las máquinas que se han tratado durante este proyecto inicialmente no estaban teniendo backups funcionales precisamente debido a la versión obsoleta del cliente frente a la versión del servidor, esto obviamente afectará a los indicadores de forma significativa pues el hecho de conseguir un backup sobre estos mismos ya es una mejoría absoluta a los rendimientos iniciales de los mismos.

Por otro lado, de las 60 máquinas, obtenemos una media de 78 backups diarios, los cuales se han monitorizado por un periodo hábil de 2 semanas para obtener datos fiables en cuanto a éxito en las ejecuciones, del 29 de Mayo, al 11 de Junio. Esto debería dar un total de 1092 backups, sin embargo, y debido a ejecuciones semanales o puntuales el total monitorizado es de 1142 backups. De este modo podemos clarificar estos como los datos obtenidos en los Sprints 10 y 11

Tomando un plano visible de hace 1 mes, para ser exactos del periodo del 1 de Mayo al 14 (Sprints 7 y 6) del mismo de estos mismos sistemas, vemos 984 backups, los cuales utilizaremos para obtener una muestra de la mejora en finalizaciones exitosas obtenidas para los backups.

Hay un obvio problema viendo esta cifra y es que vemos que es significativamente menor a pesar de contemplar las mismas máquinas y el mismo tiempo, esto se debe a un volúmen pequeño de máquinas que no tansolo no estaba finalizando correctamente sus backups, sino que no estaba registrando backups en absoluto.

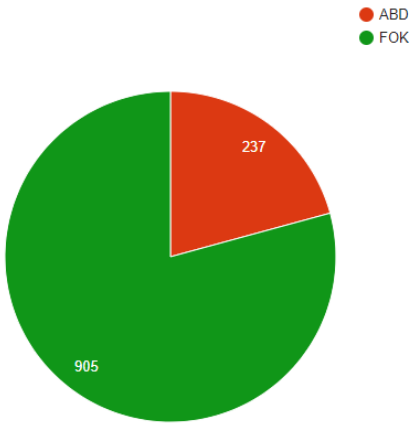
Gráfica 6.1 Gráfica mostrando el índice de backups exitosos en el periodo del 1 de Mayo al 14 de Mayo



Estos datos muestran un índice de éxito del 28.6% y muestran el alarmante estado de algunas de estas máquinas.

Por otro lado, tomando como muestra el total de backups contemplados en el periodo de Junio después de nuestras acciones, podemos encontrar el siguiente rendimiento

Gráfica 6.2 Gráfica mostrando el índice de backups exitosos en el periodo del 29 de Mayo al 11 de Junio



Como vemos la mejora es más que significativa, hemos mejorado a una tasa de un 79.24% y esto contando que muchos de los errores mostrados corresponden a fases tempranas del despliegue donde es mucho más frecuente localizar errores a corregir que posteriormente solventen el comportamiento inadecuado del sistema, por lo que es posible que este número sea mayor inclusive del actualmente obtenido.

Por otro lado hay que contemplar que se calcula una mejora de un 50.64% respecto al estado inicial de los sistemas, cabe recalcar que esta mejora solo se contempla sobre los sistemas involucrados en este proyecto y que en el balance total del departamento esta afectación es bastante significativa, pero obviamente, no tan exagerada, como se establece en la introducción, estamos hablando del impacto en 60 máquinas en un entorno de unas 1000 potencialmente afectadas y a corregir de cara a futuro, pero establece un precedente significativo del impacto que podría tener.

6.2. Mejora sobre el impacto de backups

De cara a evaluar los rendimientos de backups se decidió tomar un espacio muestral muy reducido pero que reflejaba el impacto de nuestras acciones de una forma mucho más significativa.

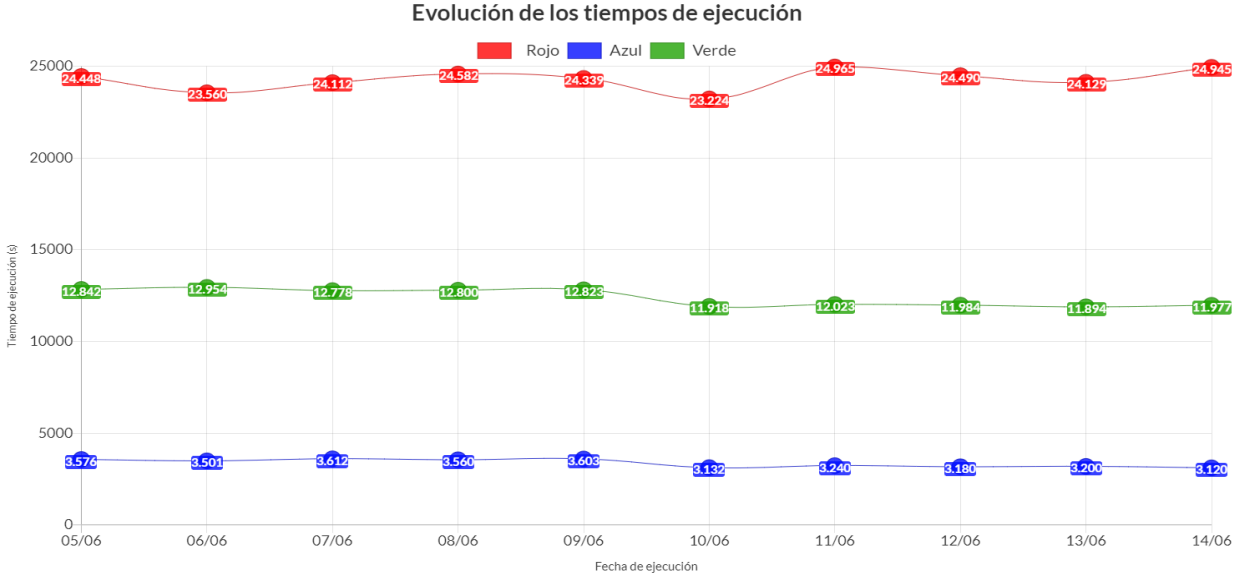
Debe entenderse que los 3 backups que se han considerado para esta evaluación, son backups de entornos críticos y que por ende no presentan errores de forma habitual, en caso de presentarlos, la acción sobre ellos es tan urgente que se deben solventar en el mismo día e incluso se relanzarán durante el transcurso del mismo.

Se han seleccionado 3 backups que almacenan unos espacios suficientemente grandes para implicar un tiempo de backup significativo, esto se debe a que en caso de contemplar todas las ejecuciones como previamente se hizo en el apartado anterior, obtendríamos muchos resultados con tiempos de ejecución ínfimos.

Para ello tomaremos como muestra los backups que llamaremos rojo, azul y verde, siendo estos nombres tentativos debido a que algunos nombres de máquinas muestran parte de las tecnologías involucradas en los mismos nombres.

El backup rojo cuenta con una exportación diaria de unos 18Tbs por cada backup de tipo incremental y un tiempo de ejecución medio de 6 horas y 37 minutos. Este es un backup sobre una base de datos y aunque no es el más crítico, es un backup del cual el tiempo nos es significativo, pues es posible que se mueva a otro aplicativo para su gestión futura en caso de no presenciar un impacto relevante en los tiempos de ejecución.

Gráfica 6.3 Gráfica mostrando la evolución de los tiempos de ejecución y el efecto



Por otro lado, el backup azul, es un backup que exporta una media diaria de 20Tbs por cada backup de tipo incremental y cuenta con un tiempo de ejecución de media de unos 59 minutos y 20 segundos, cabe clarificar, que este es un backup hecho mediante cables de fibra óptica y eso tiene un rendimiento muy significativo en la velocidad de transferencia de datos.

Finalmente, el backup verde es un backup que exporta una media diaria de 54Tbs, estos datos no son incrementales puesto que su retención es diaria y se eliminan al final del día al efectuarse la siguiente copia de seguridad, este backup tiene una duración promedio de 3 horas y 34 minutos, este backup se realiza también mediante un cable de fibra óptica y por ello a pesar de ser una volumetría tan significativa el tiempo no es tan elevado como en el primer caso.

El gráfico 6.3 muestra el detalle de los backups previamente mencionados.

Para entender el detalle de este gráfico debo indicar que los despliegues para estas máquinas se efectuaron el día 10/06, durante el Sprint 11, y por ende el efecto que tratamos de observar debería darse en ese punto exacto puesto que los backups se realizan durante las noches de los días indicados.

A efectos prácticos podemos ver que los tiempos de ejecución no han variado en gran medida, pero si prestamos más atención, vemos que parece tener un escalón en el backup que más tiempo consume pero del mismo modo vemos que no parece ser algo que se mantenga como en los otros dos casos.

Es muy posible que este fenómeno se deba a que el primer backup es gestionado a nivel de base de datos y por ende el encargado de gestionar este backup es otro producto intern nosotros hemos desplegado con este proyecto. Descartando este resultado la mejora que obtenemos es algo más interesante de lo que aparenta puesto que los backups de fibra son generalmente backups muy cuantiosos y de una criticalidad elevada y cualquier mejora de rendimiento.

En este caso podemos ver que los valores han reducido los tiempos de ejecución en torno a un 0,92% de su tiempo de ejecución, lo cual puede parecer poco pero en backups de estos tamaños y siendo directo en la medida del backup verde es una mejora de unos 5 minutos en tiempo de ejecución.

Por otro lado, y desgraciadamente como hemos contemplado, no parece tener una afectación directa a los backups en bases de datos, como hemos indicado anteriormente probablemente se debe a que el efecto se vería reflejado de actualizar de forma paralela el administrador de backups de bases de datos integrado en el cliente, una opción que podría incorporarse por tal de mejorar el actual elemento ya desarrollado.

7 CONCLUSIONES

7.1.Problemáticas encontradas durante el proyecto

El mayor problema que se ha encontrado de momento durante el proyecto ha sido el conseguir habilitar el proyecto, puesto que esto es un proyecto de empresa para un cliente significativo con grandes capas de burocracia, el proyecto no podía iniciar hasta que se dieran una serie de condiciones y se cumplieran una serie de compromisos y requisitos previos.

Durante las primeras pruebas que efectuamos, que no fueron llevadas a cabo en máquinas nuevas sino en máquinas de un entorno propio antiguo que contaba con una instalación previa de TSM, nos llevó a detectar problemas con la definición del usuario que en aquel momento estábamos dando por sentado que estaba previamente configurado de acorde a las necesidades que teníamos.

Para ser más precisos detectamos que los usuarios “adminbkp” no se encontraban configurados, elemento que no se tuvo en cuenta pues se presupuso que la configuración de usuarios se hacía de forma previa a la instalación de la paquetería. Esto nos llevó a profundizar más en la definición del usuario y también a verificar los correctos permisos de los mismos los cuales no se encontraban definidos correctamente.

Finalmente, y atado a los problemas burocráticos, otro de los problemas que se tuvo fue conseguir entornos de los sistemas operativos deseados, puesto que nosotros desde nuestro departamento no contamos con muchos de estos sistema. Entonces múltiples de las pruebas iniciales no se hicieron en entornos reales sino en virtuales intentando emular las condiciones base de los sistemas, hasta que se consiguió acceso a máquinas de otros departamentos de tipo test.

Adicionalmente, estas pruebas estaban atadas a unos tiempos de confirmación por los comités internos de cambios y por la dirección de la empresa.

7.1. Contribución general al proyecto

El estudiante es autor de gran parte de la documentación y del modelo del código. También ha coordinado la comunicación necesaria para obtener algunas de las herramientas y recursos que hemos necesitado.

Para los temas burocráticos e internos debido a la inexperiencia con gran parte de la estructura se recibió ayuda de perfiles de carácter senior y también de cara a poder formalizar parte de las reuniones con los departamentos.

Una vez ya efectuado el inicio, el estudiante ha efectuado las comprobaciones y validaciones de que el código cumpliera su función al mismo tiempo que adaptó el primer diseño a estructuras y diseños de código propios de la empresa. En varios casos reutilizando algunas funciones ya proporcionadas por elementos preexistentes como ya he mencionado anteriormente. Algunos de estos siendo el licenciado del aplicativo y las llamadas sobre la API interna. Y ante todo destacar que en todo momento se ha podido contar con la asistencia del perfil senior para cualquier duda.

De cara al inventario presentado, es un inventario que se solicitó de forma interna en el departamento y se desarrolló con la ayuda del equipo de desarrollo de soluciones TIC dentro de su ya existente, y mencionado anteriormente, portal PBK.

Analizando los objetivos, vemos que están asumidos pero si es cierto que el carácter iterativo del proyecto no se piensa que se haya manifestado todavía en su totalidad. Esto se debe a que en gran medida inicialmente se planteaba que el proyecto fuese a facilitar de forma más auspiciosa el despliegue en otros departamentos, lo que daría lugar a pruebas más directas, en entornos más reales y por ende a errores más directos, y también inicialmente la volumetría de máquinas sobre las cuales pretendemos efectuar el despliegue de cara a este proyecto de carácter universitario también era mayor.

Por otro lado y como parte de la mejora y de futuros subproyectos procedentes de este, cade destacar la posibilidad de automatizar otros softwares sucedáneos del cliente ya presentado que permitan la acción de backups sobre tecnologías de tipo Oracle, SQL o similares, los cuales constan como de uso habitual para el departamento y también agilizaría enormemente y aliviaría la carga de trabajo.

Del mismo modo y debido a que hemos obtenido una herramienta fuerte como puede ser también la autonomía a desarrollar automatismos mediante Ansible Tower mediante la aprobación de un comité de su equipo se podrían proceder a atacar otros posibles automatismos debido a también la documentación interna y los manuales de uso obtenidos sobre el aplicativo y su funcionamiento.

7.2. Reflexiones finales y generales

Para resumir las experiencias obtenidas durante el proyecto, se debe decir que la presión del proyecto ha sido significativa. Siempre ha habido una constante de problemáticas intrínsecas a factores internos de la empresa que es difícil de obviar y que del mismo modo se refleja en muchos elementos mencionados y nombrados en este informe, ha habido una presión constante a obtener ciertos resultados en ciertos tiempos y el proyecto y su naturaleza no facilitaban este concepto en los tiempos establecidos.

Dando un punto y final, se debe decir que se ha obtenido un gran grado de nuevo conocimiento de este proyecto donde no solo se han entendido estructuras internas de la empresa sino también a nivel tecnológico y de aplicación, también dando a conocer una gran cantidad de flujos de procesos y tareas intermedias necesarias para poder alcanzar los objetivos y efectuar las tareas que planteamos en el proyecto de forma inicial.

Agradecimientos

Agradecimientos en especial a Yolanda Benítez Fernández por su apoyo, ayuda y orientación de cara al enfoque empresarial del proyecto, a Marc Rodriguez Quintela y a todo el resto del departamento de Storage & Backups por su apoyo y asistencia.

Bibliografía

[1] «IBM Documentation», 28 de marzo de 2021.», https://www.ibm.com/docs/en/tsm/7.1.0?topic=servers-tivoli-storage-manager-overview
[2] A. Hat Red, «Red Hat Ansible - Automation controller». https://www.ansible.com/products/controller «Red Hat Ansible Tower», Red Hat Customer Portal. https://access.redhat.com/products/ansible-tower-red-hat
[3] «Acceso a Trello», Invitación a Trello. https://trello.com/invite/b/7VhCaeud/ATTI1f1fdc17d278aaf4f63420bd104929368A66B3C8/tfg
[4] «IBM Documentation», 3 de Marzo de 2021 https://www.ibm.com/docs/es/sgklm/3.0?topic=manager-errors-during-installation
[5] «IBM Documentation», 5 de Enero de 2022 https://www.ibm.com/docs/es/noi/1.6.2?topic=automations-creating-ansible-tower
[6] «Red Hat Workshop», https://aap2.demoredhat.com/exercises/ansible_rhel/2.3-projects/README.es.html
[7] «IBM Documentation», 17 de Marzo de 2023», https://www.ibm.com/support/pages/ibm-aix-client-requirements
[8] «IBM Documentation», 17 de Marzo de 2023», https://www.ibm.com/support/pages/ibm-aix-client-requirements
[9] «IBM Documentation», 3 de Marzo de 2021», https://www.ibm.com/docs/es/content-manager/8.5.0?topic=manager-requirements-tivoli-storage
[10] «IBM Documentation», 12 de Mayo de 2021», https://www.ibm.com/docs/es/spp/10.1.12?topic=platform-verifying-installation-spectrum-protect-plus-instance
[11] «Documentación interna sobre uso de IBM Spectrum protect», creada 10 de Abril de 2020
[12] «Documentación interna sobre uso de IBM TSM», creada 23 de Mayo de 2020
[13] «VisualParadigm»,https://online.visual-paradigm.com/es/diagrams/solutions/free-use-case-diagram-tool/

E-mail de contacte: javi.mendez.leiva@gmail.com
Menció realitzada: Enginyeria del Software
Treball tutoritzat per: Joan Protasio Ramírez
Curs 2022/23

Juny de 2023, Escola d'Enginyeria (UAB)