



BEST PRACTICE CLOUD FORMATION

AWS CloudFormation optimiza la implementación de cargas de trabajo clave en la nube de AWS. Con AWS CloudFormation, modela y aprovisiona todos los recursos necesarios para sus aplicaciones en varias regiones y cuentas de forma automatizada y segura. Las plantillas de AWS CloudFormation son archivos de texto, escritos en formato YAML o JSON, que definen los recursos de AWS que se implementarán.

Con las herramientas de automatización (AWS Lambda y AWS Systems Manager), puede automatizar el aprovisionamiento de cargas de trabajo personalizadas sobre la infraestructura de AWS definida.

1. COMIENCE CON LAS PLANTILLAS DE AWS CLOUDFORMATION EXISTENTES

Cuando utilice AWS CloudFormation para desarrollar infraestructura como código, primero verifique si lo que planea implementar ya se ha desarrollado y compartido con la comunidad.

Si está creando una implementación automatizada en la infraestructura de AWS que incluye recursos comunes de AWS (EC2, S3, RDS), comenzar con una plantilla existente puede ahorrarle horas o días de esfuerzo.

2. CREA PLANTILLAS MODULARES

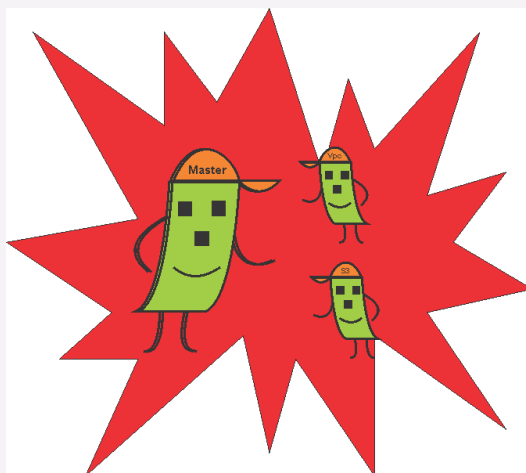
Si bien puede usar una sola plantilla de AWS CloudFormation para automatizar una implementación compleja, es más fácil trabajar con varias plantillas más pequeñas.

"Utilice un enfoque modular", Si tuviera que definir cada parte de esta implementación en una sola plantilla de AWS CloudFormation, la plantilla sería grande y difícil de solucionar. La solución de problemas y la iteración serían más fáciles si, en cambio, creara una plantilla para cada uno de estos componentes. Tendría una plantilla secundaria para la base de datos, una plantilla secundaria para la capa de aplicación, una plantilla secundaria para la capa web y una plantilla principal que implementa las plantillas secundarias en secuencia.

3. UTILICE REPOSITORIOS EXISTENTES COMO SUBMÓDULOS

Utilice submódulos para aprovechar el trabajo que ya se ha realizado. Los submódulos no solo le permiten ahorrar tiempo durante el desarrollo, sino que también reducen la necesidad de mantener recursos redundantes.

Ejm: Su pila principal puede llamar a la plantilla de submódulo como pila secundaria, pasándole parámetros para definir la VPC. Los parámetros incluyen la cantidad de zonas de disponibilidad que se utilizarán, si se implementarán puertas de enlace de traducción de direcciones de red (NAT), etc.



4. UTILICE UN ENTORNO DE DESARROLLO INTEGRADO CON LINTING

La cantidad de tiempo que se tarda en lanzar una plantilla de AWS CloudFormation varía desde unos minutos hasta varias horas. Con implementaciones complejas, un desarrollador puede cometer un simple error y no darse cuenta hasta que una pila falla tarde durante la prueba. Descubrir errores en el último minuto puede generar una frustración innecesaria y una pérdida de tiempo.

AWS CloudFormation Linter (cfn-lint) es una herramienta de código abierto mantenida por el equipo de AWS CloudFormation. Cfn-lint analiza las plantillas de AWS CloudFormation y busca errores sintácticos.

5. UTILICE PARÁMETROS PARA DEFINIR RUTAS A SUS ACTIVOS EXTERNOS.

Prepare sus plantillas de AWS CloudFormation para el futuro evitando rutas codificadas a activos externos. Por ejemplo, imagine que ha desarrollado una plantilla que implementa una instancia EC2 que descarga medios de instalación desde un bucket de S3. Puede codificar la ruta al medio de instalación como algo como esto:

`https://mibucket.s3.us-east-1.amazonaws.com/install.tar.gz`

Sin embargo, si lo hiciera, los cambios futuros en el depósito de S3, como eliminar el acceso público o eliminar el depósito por completo, provocarían un error en la implementación. En lugar de codificar la ruta al medio de instalación, defínalo como un parámetro y establezca la URL como la predeterminada.

Otra forma de acceder a sus activos externos a prueba de futuro es almacenarlos en el mismo repositorio que sus plantillas de AWS CloudFormation. Puede almacenar activos externos a los que llama la plantilla de AWS CloudFormation en el mismo repositorio que la plantilla, lo que reduce las dependencias externas.

6. UTILICE LOS MISMOS NOMBRES PARA LOS PARÁMETROS COMUNES

Cuando utiliza un enfoque modular con implementaciones complejas, puede encontrarse trabajando con una gran cantidad de plantillas de AWS CloudFormation. Los resultados de una plantilla se pasan como valores de parámetro a otras plantillas. Mantener la coherencia de los nombres de los parámetros en todas las plantillas de AWS CloudFormation facilita la resolución de problemas y la iteración.

7. AUTOMATICE LAS PRUEBAS DE AWS CLOUDFORMATION CON TASKCAT

Implementar manualmente plantillas de AWS CloudFormation en varias regiones de AWS lleva mucho tiempo y luego limpiar los activos que implementó durante las pruebas. Además, puede perder el rastro de los activos que implementó. Por lo tanto, cuando limpia el entorno de prueba manualmente, puede pasar por alto los activos que permanecen activos en su cuenta. Estos activos pasados por alto pueden generar cargos inesperados en su próximo estado de cuenta de AWS.



Para abordar estos problemas, el equipo de AWS ha desarrollado un proceso para automatizar las pruebas de AWS CloudFormation mediante una herramienta de código abierto llamada TaskCat. Ingresa las regiones y los valores de los parámetros en un archivo con formato YAML llamado `taskcat.yml` en la raíz de su repositorio. TaskCat utiliza estos valores para implementar automáticamente sus plantillas de AWS CloudFormation. Luego, registra los resultados de la implementación, ya sea que haya tenido éxito o no (y, si falla, el motivo), y elimina los activos que se implementaron.

También puede evitar que los recursos que se implementaron durante las pruebas permanezcan en sus cuentas de AWS.

8. MANTENGA SUS PLANTILLAS

Dado que las plantillas de AWS CloudFormation se utilizan para automatizar las implementaciones, a menudo se reutilizan durante meses o años. Como resultado, pueden quedar desactualizados. Por ejemplo, los tipos de instancias EC2 pueden quedar obsoletos, las imágenes de máquinas de Amazon (AMI) pueden eliminarse de la disponibilidad general y pueden agregarse nuevas funciones a un servicio de base de datos.

Pruebe sus plantillas de AWS CloudFormation periódicamente para asegurarse de que sigan funcionando como se esperaba. Incorpore nuevas funciones de servicio a medida que estén disponibles.