Introducción a Infraestructura como Código (IaC)

Miguel Barajas

Version 1.0, 2021

Contenido

Dedicación

A Valeria, Roberto y Carlota. Mi mundo

Prólogo

TODO

Preface Sub-section

Preface sub-section body.

Infraestructura como Código

Para entender la infraestructura como código (IaC por sus siglas en inglés) primero debemos entender qué es infraestructura y qué es código. Infraestructura en el contexto de informática, es el hardware, físico, virtual o lógico que soporta una aplicación o plataforma. Del mismo modo, el código son las instrucciones que permiten crear una aplicación o software. Luego entonces, la infraestructura como código, es la descripción del estado deseado de la configuración de esta infraestructura en un documento, normalmente escrito en texto plano. Al estar escrito en texto plano, esta descripción puede tratarse de la misma manera que se trata un código de software, donde podemos versionarlo, probarlo, compartirlo, etc.

Para hacer esto realidad, debemos contar con un intermediario o intérprete que comprenda esta descripción y la convierta en una configuración que sea entendida por la infraestructura. Esto nos permite automatizar la infraestructura de tal manera que esa configuración la tratemos como software, esto nos da varias ventajas que estaremos discutiendo durante la presente obra, algunas de ellas son:

- Automatización declarativa en vez de imperativa
- Versionamiento de la configuración
- Replicación sencilla
- **ESCRIBIR MAS**

Automatización de Infraestructura

La automatización de la infraestructura no es un tema nuevo, desde hace décadas, está práctica existe, dado a la necesidad de hacer más eficiente el despliegue de esta infraestructura, esto se hace mediante ***scripts*** o desarrollos que permiten que esta infraestructura se despliegue automáticamente, el problema con este acercamiento, es que no solo le tenemos que planear qué se va a desplegar, si no tambien cómo se desplegará, por lo que es una automatización **imperativa**, esto nos acarrea un problema dado que muchas veces la infraestructura no puede ser cambiada una vez desplegada, puesto que en la automatización debemos tener en cuenta todas las excepciones que pudieran ocurrir durante el ciclo de vida de la infraestructura. Por ende, la automatización imperativa funciona muy bien solo para el despliegue de nueva infraestructura, pero no para mantener el ciclo de vida completa de ella. Es decir, despliegue, cambios y decomización.

Más allá de la automatización, existe el concepto de ***orquestación*** en el que normalmente existe un software llamado **orquestador** (***orchestrator*** en inglés) el cual cuenta con los conectores necesarios para automatizar el despliegue de las diferentes capas de la infraestructura: Almacenamiento, Red, Computo, Sistema Operativo, etc).

Esto permite "orquestrar" el despliegue de la pila completa de la infraestructura. Pero de la misma manera que se hace al automatizar de manera imperativa, capa por capa, la orquestación imperativa no permite manejar el ciclo de vida completo de la infraestructura.

Manejadores de configuración

Los manejadores de configuración, han existido, de igual manera, desde hace décadas, son piezas de software que nos permiten mantener la configuración de un sistema operativo o aplicación de manera declarativa. Es decir, en vez de indicar el qué y el cómo, se describe el qué, es decir se declara el estado deseado de la configuración del Sistema Operativo o aplicación y es el manejador de configuración quien hace la conciliación entre el estado declarado y el estado actual. Es decir, que solo modifica las partes de la configuración que no concuerdan con el estado deseado. Como se manifestó, los Manejadores de configuración (***Configuration Manager*** en inglés) fueron concebidos para mantener la configuración del Sistema Operativo y/o la aplicación. Con el avance de las tecnologías de virtualización y del concepto de ***Software Defined***, el despliegue y configuración de la infraestructura, también puede ser declarada y manejada como si de un sistema operativo o una aplicación se tratara.

El crecimiento de la popularidad de IaC

Sin duda, la adopción del ***Cloud Computing*** ha permitido que la Infraestructura como código se haya popularizado, la disponibilización de la interfaces de programabilidad (APIs), así como el cobro por tiempo utilizado de la infraestructura, ha hecho que IaC sea muy popular para este tipo de modelos de consumo. Sin embargo, también e influenciado por esto, los fabricantes de infraestructura física, han empezado a hacer

Chapters can contain sub-sections nested up to three deep. [1]

Chapters can have their own bibliography, glossary and index.

And now for something completely different: monkeys, lions and tigers (Bengal and Siberian) using the alternative syntax index entries. Note that multi-entry terms generate separate index entries.

Here are a couple of image examples: an [smallnew] example inline image followed by an example block image:

[Tiger image] | *images/tiger.png*

Figure 1. Tiger block image

Followed by an example table:

Table 1. An example table

Option	**Description**
-a 'USER GROUP'	Add 'USER' to 'GROUP'.
-R 'GROUP'	Disables access to 'GROUP'.

Example 1. An example example

> Lorum ipum...

Sub-section with Anchor

Sub-section at level 2.

Chapter Sub-section

Sub-section at level 3.

Chapter Sub-section

Sub-section at level 4.

This is the maximum sub-section depth supported by the distributed AsciiDoc configuration. [2]

[1] An example footnote.

[2] A second example footnote.

The Second Chapter

An example link to anchor at start of the first sub-section.

An example link to a bibliography entry [taoup].

The Third Chapter

Book chapters are at level 1 and can contain sub-sections.

Appendix A: Example Appendix

One or more optional appendixes go here at section level 1.

Appendix Sub-section

Sub-section body.

Example Bibliography

The bibliography list is a style of AsciiDoc bulleted list.

Books

- [taoup] Eric Steven Raymond. 'The Art of Unix Programming'. Addison-Wesley. ISBN 0-13-142901-9.
- [walsh-muellner] Norman Walsh & Leonard Muellner. 'DocBook - The Definitive Guide'. O'Reilly & Associates. 1999. ISBN 1-56592-580-7.

Articles

- [abc2003] Gall Anonim. 'An article', Whatever. 2003.

Example Glossary

Glossaries are optional. Glossaries entries are an example of a style of AsciiDoc labeled lists.

A glossary term

The corresponding (indented) definition.

A second glossary term

The corresponding (indented) definition.

Example Colophon

Text at the end of a book describing facts about its production.