**INSFRAKTRUKTUR AS CODE (IAC)**

Merupakan proses penyediaan infrastruktur yang mana sistem dibangun dan dikelola melalui kode secara otomatis. Dengan menggunakan kode dan mengotomatisasi, proses setting dan konfigurasi baremetal, virtual mesin, cloud computing baik itu instalasi baru, perubahan kofnigurasi dapat dilakukan secara cepat, mudah dan berulang. Selain itu juga memiliki manfaat yang lain yaitu dokumentasi (siapapun akan tahu konfigurasi server, kebutuhan aplikasi server dsb).

Untuk menerapkan IAC tools/alat yang digunakan adalah Packer (build image), Terraform (build vm dan privisioning instance/vm/server), Ansible, Chef, Puppet & Salt Stack (Configurations Management).

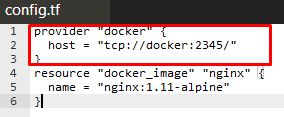
Berikut adalah contoh penerapan tool yang dapat membantu dalam mengintegrasikan IaC, yaitu Terraform. Terraform adalah tool pneyedia insfraktruktur yang dibuat oleh Hashicorp. Terraform memiliki sifat cloud-agnostic (kemampuan untuk berfungsi tanpa “mengetahui” rincian yang mendasari sistem yang bekerja di dalam) yang mana mampu untuk mengotomatisasi tumpukan insfrastruktur dari beberapa penyedia layanan cloud secara bersamaan dan mengintegrasikan layanan pihak ketiga lainnya.

Deploy kontainer NGINX menggunakan Terraform

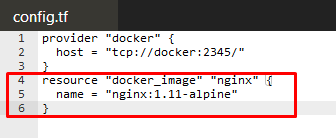
<https://www.katacoda.com/courses/terraform>

1. Membuat konfigurasi Terraform

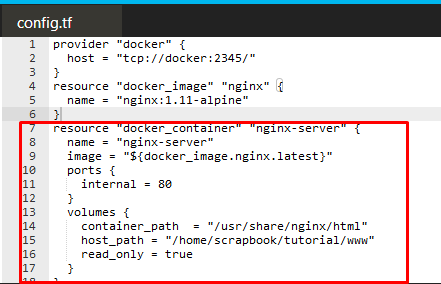
* Membuat file config.tf pada bagian editor. Yang mana di bagian pertama mendefinisikan host docker mana yang akan digunakan untuk menerapakan konfigurasi.



* Mendefinisikan resources infrastruktur yang akan dibuat. Disini resource pertama adalah Image Docker. Resources memiliki dua parameter yaitu type dan name.type/jenisnya adalah docker\_image dan namanya adalah nginx.

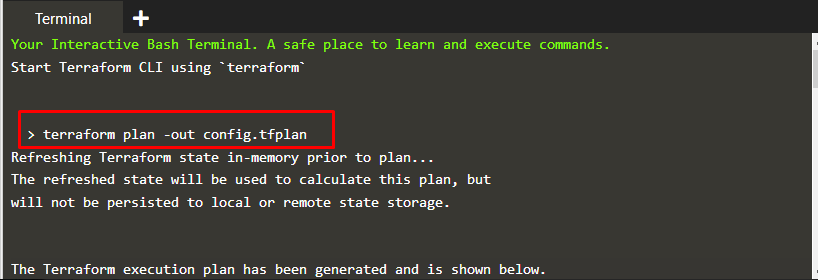


* Disini kita dapat menentukan resource kontainernya. Jenis resource nya adalah docker\_container dan namanya adalah nginx-server. Di dalam blok diatur parameter resourcenya.

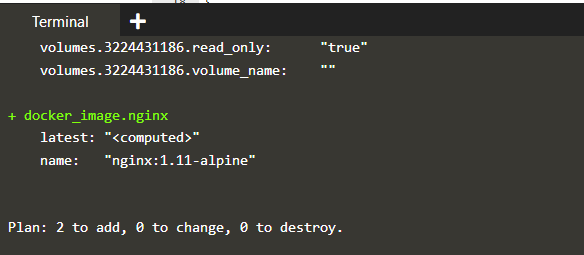


1. Plan Terraform Actions

Setelah dibuat konfigurasinya, selanjutnya perlu dibuat sebuah rencana ekseskusi. Terraform mengambarkan tentang tindakan yang diperlukan untuk mencapai sebuah keadaan yang diinginkan. Plan nya dapat di simpan menggunakan –out. Selanjutnya perapannya adalah di langkah berikutnya. Untuk membuat sebuah plan / rencana menggunakan CLI bisa menggunakan perintah sebagai berikut



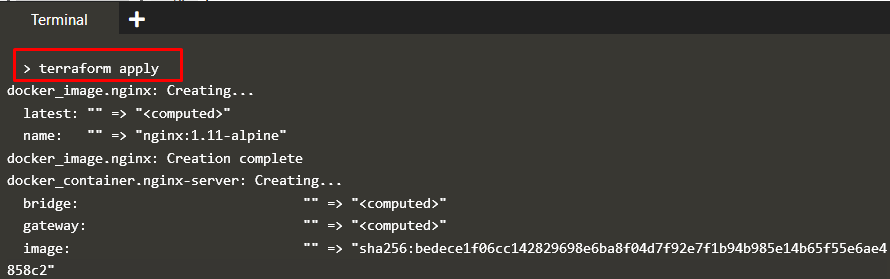


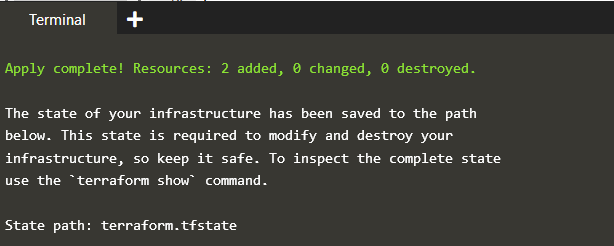


Output dari perintah diatas menunjukkan sebuah perubahan. Dalam hal ini kita dapat melihat \_+ dockercontainer.nginx-server dan \_+ dockerimage.nginx yang merupakan resource baru. Finally a summary of plan: 2 to add, 0 to change, 0 to destroy.

1. Apply Terraform Action

Setelah rencana sudah dibuat, kita perlu menerapkannya agar sesuai dengan keadaan yang kita inginkan. Dalam perintah di dalam CLI ini, terraform akan mem pull images apa pun yang diperlukan kan menampilkan kontainer baru.

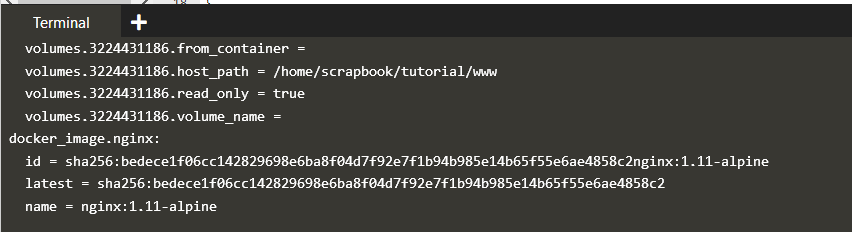




1. Inspecting Infrastructure

Memeriksa infrastruktur dapat menggunakan CLI dari Docker agar bisa melihat perubahan dan melihat kontainer yang baru ditampilkan. Bisa menggunakan perintah dibawah ini , selain itu kita juga dapat memerika di masa yang akan datang dengan menggunakan CLI dari Terraform.





1. Updating Infrastructure

Ketika infrastruktur sedang berkembang dan berubah, terraform akan mengelola dan memastikan bahwa terraform masih menyediakan area yang kita inginkan. Disini juga dapat mengubah kontainer untuk menampilkan dua instance masing – masing dengan nama yang berbeda. Untuk menyesuaikan infrastruktur agar dapat sesuai dengan konfigurasi yang telah dibuat bisa menggunakan perintah : *terraform plan –out config.tfplan* . disini perubahan akan di urai, karena pada perintah ini terdapat perubahan nama dan penambahan resource. Dengan mengubah nama kontainer dapat memaksa resource untuk membuat ulang nama nginx-server" => "nginx-server-1" (memaksa membuat resource baru) dengan menambahkan kontainer baru \_+ dockercontainer.nginx-server.1. selanjutnya bisa menerapkan sebuah rencana / plan seperti di langkah sebelumnya.

