

PROVISIONING GOOGLE KUBERNETES ENGINE CLUSTER DENGAN MENGGUNAKAN TERRAFORM DAN JENKINS PADA DUA ENVIRONMENT

Odi Pramadika*¹⁾, Dian Widiyanto Chandra²⁾

1. Universitas Kristen Satya Wacana, Indonesia
2. Universitas Kristen Satya Wacana, Indonesia

Article Info

Kata Kunci: Provisioning, Google Kubernetes Engine, Jenkins, Terraform

Keywords: Provisioning, Google Kubernetes Engine, Jenkins, Terraform

Article history:

Received 22 January 2023

Revised 29 January 2023

Accepted 24 February 2023

Available online 1 June 2023

DOI :

<https://doi.org/10.29100/jipi.v8i2.3630>

* Corresponding author.

Corresponding Author

E-mail address:

672015071@student.uksw.edu

ABSTRAK

Pada sistem dengan cakupan besar membutuhkan ratusan bahkan ribuan server. Setiap server yang ada juga harus saling terhubung satu sama lain agar dapat berkomunikasi secara dua arah dan setiap server memiliki kapasitasnya masing-masing. Pada tahap development selalu membutuhkan minimal dua environment, yaitu development environment dan production environment. Provisioning GKE Cluster menggunakan Terraform dan Jenkins dapat memudahkan tim DevOps dalam menyiapkan Server sesuai kebutuhan secara otomatis. Hanya memerlukan waktu 15 menit 18,412 detik untuk menyiapkan satu cluster dengan tiga node pada environment Development dan 13 menit 26,407 detik untuk menyiapkan satu cluster dengan satu node pada environment Production. Dengan adanya sistem ini membantu tim DevOps untuk menyiapkan server dengan lebih cepat dan minim kendala karena semakin kompleksnya kebutuhan suatu perusahaan.

ABSTRACT

On large-scale systems, hundreds or even thousands of servers are required. Each existing server must also be connected to each other in order to communicate in two directions and each server has its own capacity. At the development stage, it always requires at least two environments, namely the development environment and the production environment. Provisioning GKE Clusters using Terraform and Jenkins can make it easier for the devops team to set up Servers as needed automatically. It only takes 15 minutes 18.412 seconds to set up a cluster with three nodes in the Development environment and 13 minutes 26.407 seconds to set up a cluster with one node in the Production environment. With this system, it helps the Devops team to set up servers faster and with minimal constraints due to the increasingly complex needs of a company.

I. PENDAHULUAN

Server pada suatu perusahaan yang bergerak dibidang teknologi merupakan faktor yang sangat penting dalam proses berjalannya suatu sistem. Sehingga sangat merepotkan DevOps Engineer dalam menyiapkan setiap server secara manual satu persatu ketika sistem tersebut dibangun. Pada tahap development selalu membutuhkan minimal dua environment, yaitu development environment dan production environment. Development environment diperuntukan untuk tahap pengembangan aplikasi maupun program. Sedang Production Environment berisikan data-data riil yang sudah berjalan dan sudah dapat digunakan oleh pengguna. Namun kebutuhan server environment dapat terus berkembang seiring dengan kompleksitas suatu perusahaan. Sehingga DevOps Engineer harus mengatur secara manual provisioning server setiap step dengan menyesuaikan kondisi server saat ini dan jika membutuhkan beberapa server secara bersamaan harus di setting satu persatu. Hal tersebut sangat memakan waktu dan kurang dinamis dalam pembangunan server.

Pipeline adalah sekelompok peristiwa yang saling terkait satu sama lain secara berurutan. Pipeline menyediakan seperangkat alat yang dapat diperluas untuk memodelkan pipeline pengiriman sederhana hingga kompleks sebagai code melalui Bahasa Groovy sehingga lebih dinamis. [1]

Terraform adalah satu layanan infrastruktur open-source dari HashiCorp. Terraform memiliki ekstensi berkas khusus dengan akhiran *.tf, *.tfvars, dan *.tfstate. Terraform merupakan tool Infrastructure as Code (IaC) yang digunakan untuk membuat, mengubah, dan juga menggabungkan sebuah infrastruktur dengan aman dan efisien menggunakan code. Dengan IaC, file konfigurasi berisikan spesifikasi infrastruktur, sehingga memudahkan untuk

mengedit dan mendistribusikan konfigurasi sesuai kebutuhan dalam skala besar. [2]

Seiring perkembangannya suatu perusahaan maka semakin kompleks kebutuhan yang harus disiapkan, dengan adanya provisioning GKE menggunakan Terraform dan Jenkins dapat memudahkan tim devops dalam menyiapkan Server sesuai kebutuhan pada kedua environment. Namun terdapat hal yang tidak dapat terimplementasi dengan sempurna dan perlu dilakukan pengembangan di penelitian kedepannya yaitu mengembangkan sistem penambahan Terraform Backend untuk listening dan sinkronasi file stage agar lebih dinamis. Selain itu karena keterbatasan dalam akun yang digunakan dalam penelitian sehingga hanya dapat melakukan pengukuran terbatas, maka dapat dilakukan beberapa kali deployment untuk melihat selisih pada masing masing environment..

Pada penelitian terdahulu yaitu implementasi infrastruktur server berbasis cloud computing untuk web service berbasis teknologi Google Cloud Platform (GCP), infrastruktur server dibangun secara manual pada penelitian ini menggunakan produk Google Cloud Platform (GCP) dan menerapkan fitur autohealing dan autoscaling sebagai metode untuk menjaga tingkat reliabilitas dan availability dari infrastruktur server yang akan dirancang dan dibangun. [3]

Selain itu berdasarkan penelitian analisis perbandingan performa web server Docker Swarm dengan Kubernetes Cluster, dapat disimpulkan bahwa dalam mengorkestrasi cluster container, Kubernetes Cluster memberikan performa yang bagus dengan response time yang lebih cepat dibanding Docker Swarm.[4]

Kemudian pada penelitian implementasi web server menggunakan infrastructure as code Terraform berbasis cloud computing, disimpulkan bahwa hasil pengujian quality of service web server instance menggunakan infrastructure as code Terraform dan dibuat menggunakan Openstack sesuai dengan standar TIPHON dan infrastructure as code Terraform dapat digunakan sebagai alat otomatisasi sebagai bekal untuk pembangunan infrastruktur server web.[5]

Selanjutnya pada penelitian implementasi CI/CD Pipeline pada Framework Androbase menggunakan Jenkins dengan studi kasus PT. Andromedia. Penelitian tersebut diangkat berawal dari masalah proses deployment dalam pembangunan sebuah aplikasi kerap kali bermasalah dikarenakan cara deploy yang masih manual yaitu dengan cara pull aplikasi dari server produksi, cara ini dirasa kurang praktis dikarenakan developer harus berulang kali melakukan pull dari server apabila ada update pada aplikasi, masalah serupa terjadi pada sebuah perusahaan jasa system informasi PT.Andromedia dimana proses deploy yang dilakukan masih secara manual. Oleh karena itu Farid sebagai penulis penelitian ini, penulis menawarkan solusi dengan mengintegrasikan proses deployment aplikasi dengan menggunakan konsep continuous integration and deployment atau yang biasa disingkat CI/CD. [1]

Berdasarkan penelitian yang sudah ada sebelumnya, maka penulis melakukan provisioning pada Google Kubernetes Engine (GKE) Cluster menggunakan Terraform sebagai Infrastructure as Code (IaC). Infrastructure as Code (IaC) digunakan untuk mengelola komponen infrastruktur salah satunya server dan layanan cloud. Kemudian menggunakan Jenkins Pipeline untuk mempercepat alur kerja CI/CD. Pipeline menggunakan Bahasa pemrograman Apache Groovy yang cukup powerful dan dinamis. Hal ini bertujuan untuk memudahkan tim devops dalam menyediakan Server sesuai kebutuhan.

II. METODOLOGI PENELITIAN

Cloud Computing adalah metode penyampaian berbagai layanan melalui internet. Cloud Computing memiliki banyak keuntungan, seperti menghemat biaya, meningkatkan produktivitas, kecepatan, efisiensi, performa, dan keamanan. Cloud Computing menyediakan infrastruktur yang fleksibel, efisien dan berbagai aplikasi untuk bisnis.[3] Pengguna hanya membutuhkan koneksi internet untuk mengakses secara nyaman, sigap, dan sesuai dengan kebutuhan kepada suatu kumpulan sumber daya Cloud Computing. Terdapat banyak service provider yang menyediakan Cloud Computing, salah satunya adalah Google Cloud Platform (GCP). [9]

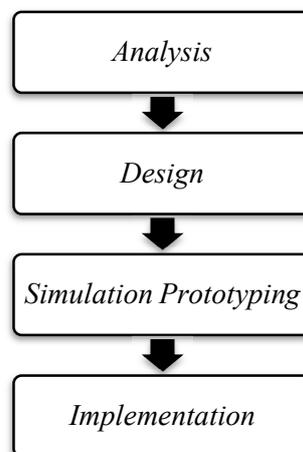
GCP adalah layanan cloud yang ditawarkan langsung oleh Google. GCP merupakan sistem Cloud Computing yang menawarkan banyak manfaat. Salah satu hal yang cukup menjadikan GCP sebagai pertimbangan beberapa perusahaan adalah harga yang cenderung lebih murah dari kompetitor, keamanan sistem yang mumpuni, fitur yang memumpuni. Salah satu fitur dalam GCP adalah Google Kubernetes Engine (GKE). [10] GKE adalah manajemen cluster dan sistem orkestrasi yang handal untuk menjalankan container Docker yang dibangun di atas sistem Kubernetes open source. Mode operasi standar GKE memberikan kontrol penuh atas node dengan kemampuan untuk menyesuaikan konfigurasi yang diperlukan. [6]

Jenkins adalah alat DevOps yang membantu pengguna menghabiskan lebih sedikit waktu untuk mengonfigurasi infrastruktur dasar seperti instalasi dan konfigurasi, dan lebih banyak waktu untuk melakukan pengembangan aktual dan mempercepat alur kerja CI/CD. Jenkins mendukung semua platform cloud termasuk AWS, Google Cloud, IBM Cloud, Microsoft Azure, Red Hat OpenShift, dan Pivotal. [1]

Apache Groovy adalah bahasa yang powerful, bersifat dinamis, dan berorientasi pada objek. Bahasa ini termasuk cukup dinamis yang artinya mampu berjalan dengan lancar di library Java. Apache Groovy termasuk bahasa pemrograman yang bersifat open source. Di dalam pembuatannya, bahasa pemrograman Groovy sudah mendapat beberapa pengaruh dari bahasa pemrograman lain seperti Python, Ruby, Smalltalk, Perl, dan tentu saja Java. Groovy dapat digunakan untuk mengatur pipeline di Jenkins. [7]

GitHub digunakan untuk menyimpan source code pada suatu proyek dan melacak riwayat lengkap semua perubahan code tersebut. GitHub memiliki versi induknya, yaitu “git”. Git sendiri adalah sebuah software VCS (Version Controlled System) yang bertugas untuk mencatat perubahan seluruh file atau repository suatu project yang diakses secara offline. Selain itu GitHub memberikan layanan cloud untuk menyimpan dan mengelola project atau repository git. Karena bersifat online, pengguna dapat mengubah sebuah repository atau project secara bersamaan dengan orang lain di tempat yang berbeda. Oleh karena itu, platform ini sangat membantu tim project dalam menyusun suatu folder yang berisikan files terkait pemrograman. [8]

Pada Gambar 1 merupakan tahapan penelitian provisioning Google Kubernetes Engine (GKE) Cluster dengan menggunakan Terraform dan Jenkins pada dua environment yang bertujuan untuk mempermudah seorang devops engineer dalam menyiapkan Server sesuai kebutuhan.

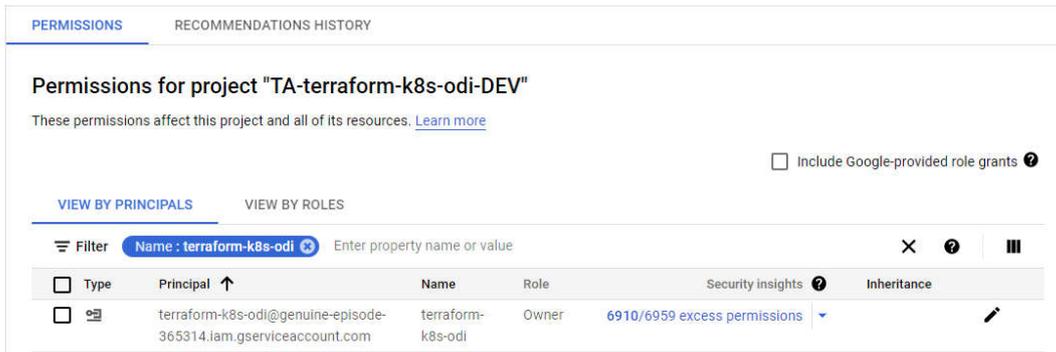


Gambar 1 Tahapan Penelitian

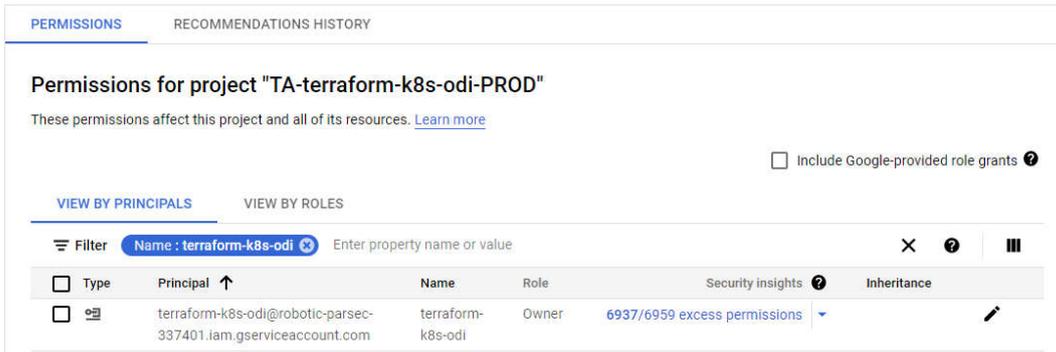
Tahapan penelitian pada Gambar 1 terdiri dari empat tahapan yaitu Analysis, Design, Simulation Prototyping, dan Implementation. Pada tahap Analysis dilakukan pengumpulan data berupa infrastruktur, cloud computing, produk Google Cloud Platform, IaC, Jenkins, dan pipeline yang sesuai dengan kebutuhan pembangunan penelitian ini. Tahap kedua ialah tahap Design merupakan rancangan sistem yang dibangun pada penelitian ini. Tahap ketiga merupakan Simulation Prototyping ialah perancangan lebih detail mengenai tools yang digunakan dan melakukan uji coba verifikasi konfigurasi dan skenario. Tahap keempat ialah Implementation merupakan tahap eksekusi konfigurasi membangun sistem yang dirancang sebelumnya.

A. Analisis Kebutuhan

Berdasarkan hasil pengumpulan data terkait infrastruktur, cloud computing, dan produk Google Cloud Platform (GCP). Produk GCP yang digunakan pada penelitian ini adalah Google Kubernetes Engine (GKE) versi 1.23.6-gke.2200. Terraform yang digunakan adalah versi lebih dari 0.13. Terraform membutuhkan penyedia provider yaitu google sebagai GCP bersumber dari hashicorp/google dengan versi 4.10.0 sampai dengan versi 5 dan Kubernetes bersumber dari hashicorp/Kubernetes dengan versi 2.10. Jenkins yang digunakan adalah Jenkins versi 2.361.2. Selain itu dibutuhkan key service account pada kedua cluster yaitu prod dan dev. Service account tersebut digunakan sebagai autentikasi agar dapat mengakses GKE. Service account berbentuk email address yang unik seperti pada Gambar 2 dan Gambar 3.



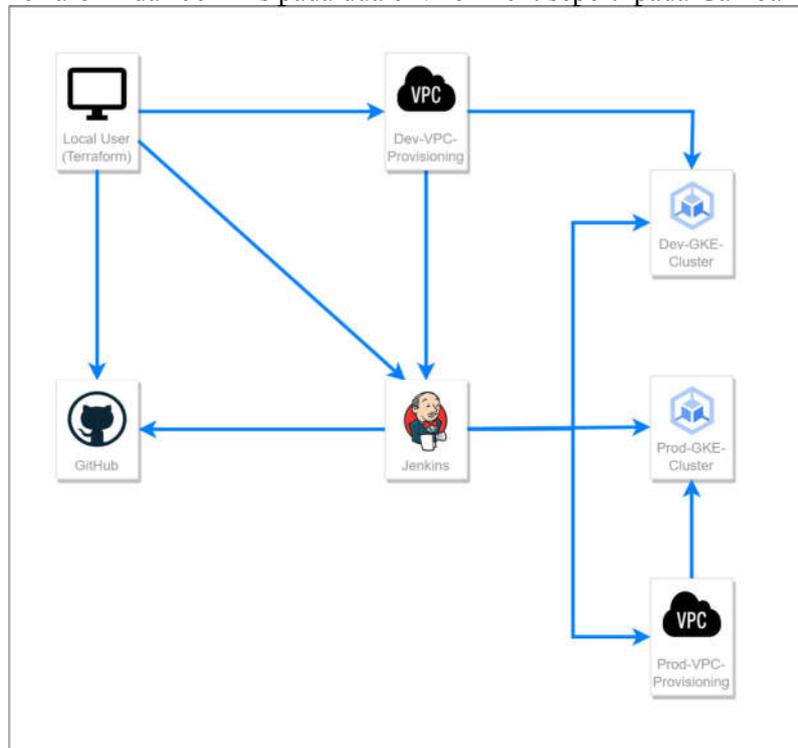
Gambar 2 Service Account Dev



Gambar 3 Service Account Prod

B. Desain Sistem

Pada penelitian kali ini penulis merancang alur system provisioning Google Kubernetes Engine (GKE) Cluster dengan menggunakan Terraform dan Jenkins pada dua environment seperti pada Gambar 4.



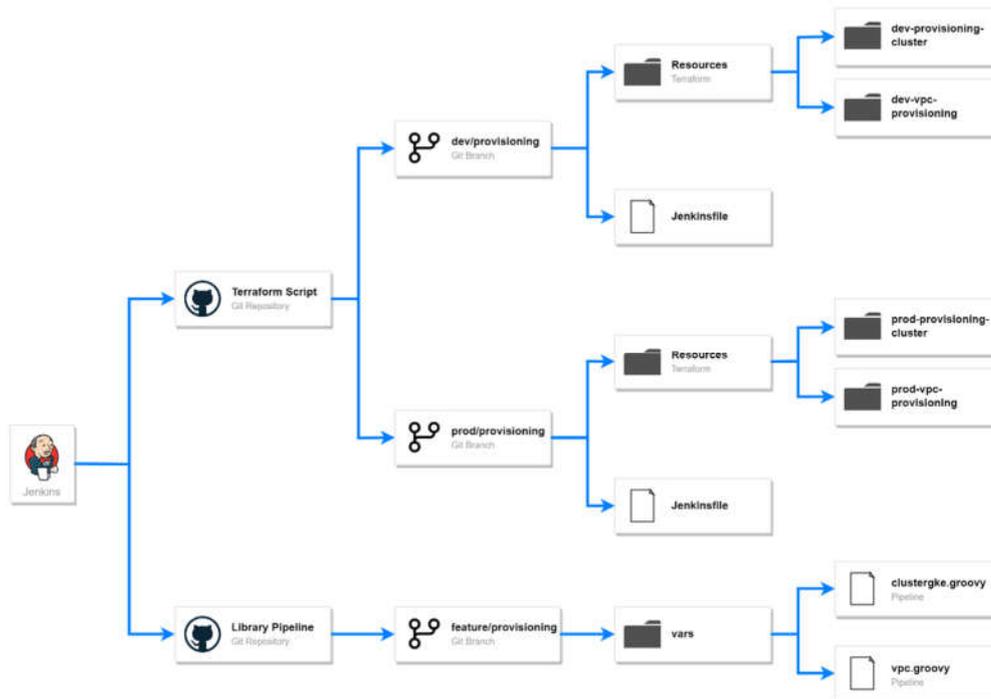
Gambar 4 Alur Sistem

Proses dimulai dengan tim Devops sebagai Local User, melakukan tiga hal secara parallel yaitu, mengatur repository dan branch pada GitHub, build Jenkins Server, dan build Virtual Private Cloud (VPC) atau dalam penelitian ini dinamakan dengan Dev-VPC-Provisioning. Dev-VPC-Provisioning memiliki peran mengatur IP subnet yang digunakan pada Jenkins Server. Jika pengaturan pada Jenkins sudah sesuai maka Jenkins melakukan

scanning ke Github dan Jenkins mem-build hasil scanning tersebut sehingga menghasilkan Dev-GKE-Cluster, Prod-GKE-Cluster, dan Prod-VPC-Provisioning. Selain digunakan di Jenkins Server, Dev-VPC-Provisioning juga berperan dalam mengatur IP Subnet di Dev-GKE-Cluster. Kemudian Prod-VPC-Provisioning digunakan sebagai IP Subnet pada Prod-GKE-Cluster.

C. Simulasi Prototype

Pada Gambar 5 sebelum Jenkins server melakukan scanning ke GitHub, tim Devops perlu menyiapkan apa saja dalam membangun arsitektur Github?



Gambar 5 Arsitektur GitHub

Pada Arsitektur GitHub terdapat dua repository dengan perannya masing-masing. Repository Terraform Script berperan untuk menyimpan file untuk masing masing environment yaitu, branch development dan branch production. Setiap branch memiliki file JenkinsFile dan folder Resources. Jenkinsfile memiliki peran yaitu memanggil library pada repository Library Pipeline melalui Jenkins Server. Sedangkan Resources memiliki folder provisioning cluster yang berperan untuk scrip terraform yang akan di deploy oleh pipeline dan VPC Provisioning berperan dalam pengaturan IP Subnet pada masing masing cluster. Selain repository Terraform Script terdapat juga repository Library Pipeline yang hanya memiliki satu branch yaitu feature provisioning. Di dalam branch tersebut memiliki 1 folder Vars guna menyimpan dua file pipeline cluster gke, dan VPC dalam bentuk .groovy.

III. HASIL DAN PEMBAHASAN

Hasil Dev-VPC-Provisioning yang telah dibuild dapat terlihat di GKE seperti pada Gambar 6, IP subnet yang dihasilkan yaitu 10.226.0.0/16. IP subnet tersebut digunakan di Jenkins Server dan Dev-Provisioning-Cluster.

Name ↑	Region	Subnets	MTU ?	Mode	Internal IP ranges	External IP ranges
▶ default		36	1460	Auto	None	
▼ dev-vpc-k8s-odi		1	1460	Custom	None	
	asia-southeast1	dev-k8s-subnet-odi			10.226.0.0/16	None

Secondary IPv4 ranges	Gateways	Firewall Rules	Global dynamic routing	Flow logs
		4	Off	
		9	Off	
10.227.0.0/16, 10.2...	10.226.0.1			Off

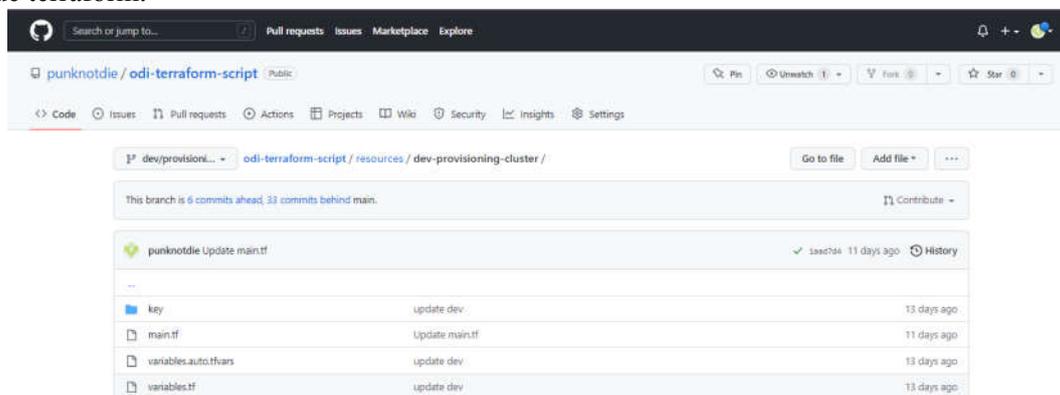
Gambar 6 Dev-VPC-Provisioning

Pada Gambar 7 merupakan hasil Jenkins Server yang telah di build dapat terlihat di GKE, bernama jenkins-ubuntu-odi memiliki IP Subnet 10.226.0.0/16 yang di dapat dari VPC dev.

<input type="checkbox"/>	<input checked="" type="checkbox"/>	jenkins-ubuntu-odi	asia-southeast1-a	10.226.0.39 (nic0)	34.124.178.145 (nic0)	SSH	⋮
--------------------------	-------------------------------------	--------------------	-------------------	--------------------	-----------------------	-----	---

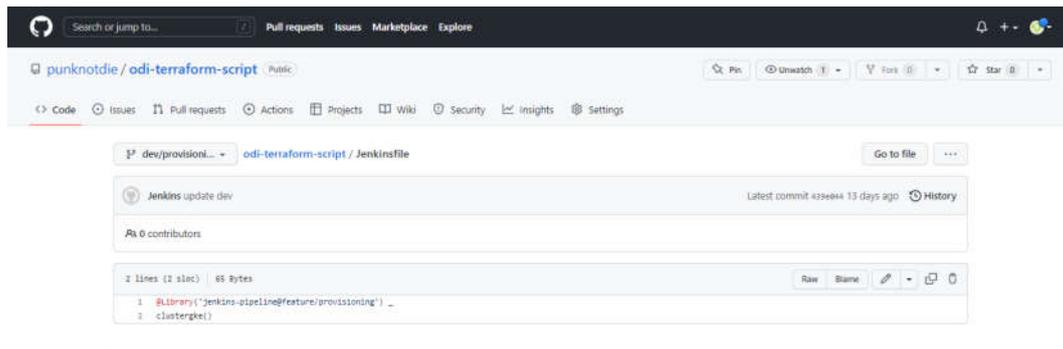
Gambar 7 Jenkins Server

Hasil setingan Repository Odi-Terraform-Script terdapat dua branch, yaitu dev/provisioning dan prod/provisioning. Di setiap branch tersebut memiliki folder resources dan file Jenkinsfile. Seperti pada Gambar 8 folder Resources pada kedua branch tersebut menyimpan file main.tf, variable.auto.tfvars, dan variable.tf. Masing masing file tersebut memiliki peranannya masing-masing, main.tf merupakan script utama terraform. Variable.auto.tfvars merupakan file value dari variable terraform. Variable.tf merupakan define variable type dan default value terraform.



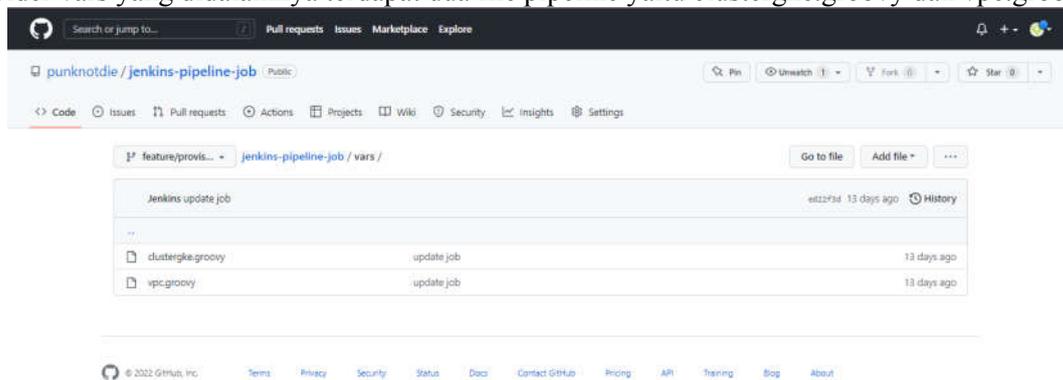
Gambar 8 Provisioning Cluster

Selain itu pada kedua branch tersebut memiliki file Jenkinsfile yang berisikan perintah untuk memanggil library pada repository Library Pipeline melalui Jenkins Server seperti pada Gambar 9.



Gambar 9 Jenkins File

Selain Repository Odi-Terraform-Script terdapat pula repository Jenkins-Pipeline-Job yang mana didalam repository tersebut hanya terdapat satu branch, yaitu feature/provisioning seperti pada Gambar 10. Branch tersebut memiliki folder vars yang didalamnya terdapat dua file pipeline yaitu clustergke.groovy dan vpc.groovy.



Gambar 10 Jenkins Pipeline

Pipeline ditulis menggunakan Bahasa Apache Groovy dan disimpan di repository Jenkins-Pipeline-Job. Pipeline tersebut merupakan perintah untuk menjalankan. Terdapat dua pipeline yaitu vpc.groovy dan clustergke.groovy. Masing –masing Pipeline memiliki tiga stage yaitu Git Clone, Deploy Terraform, dan Check State. Stage Git Clone berisikan script yang digunakan untuk mengambil file GitHub dari Repository yang dituju. Stage Deploy Terraform berisikan script jika branch name bernama dev/provisioning maka menjalankan script terraform yang berada di dev/provisioning. Jika branch name bernama prod/provisioning maka menjalankan script terraform yang berada di prod/provisioning. Selanjutnya terdapat stage Check State yang berisikan script untuk menampilkan file terraform.tfstate yang merupakan hasil deployment terraform.

Seperti pada Gambar 11 Hasil deployment Terraform pada Branch dev/Provisioning memerlukan waktu total 15 menit 18,412 detik untuk menjalankan ketiga stage dari script pipeline sebelumnya dan menghasilkan cluster dengan tiga node. Dalam melakukan suatu penelitian diperlukan uji coba beberapa kali dan mengalami kegagalan sebelum akhirnya berhasil. Pada proses build dev/provisioning telah dilakukan dua belas kali percobaan hingga akhirnya berhasil dan rata rata waktu keseluruhan dalam dua belas kali percobaan tertera pada Average full run time yaitu 6 menit 6 detik.

Branch dev/provisioning

Full project name: TERRAFORM-DEPLOYMENT/dev%2Fprovisioning

Stage View



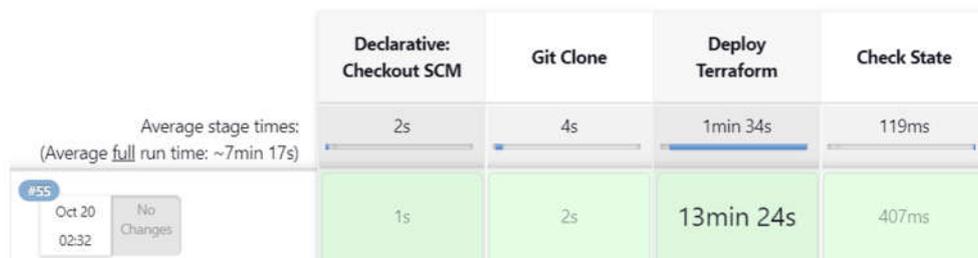
Gambar 11 dev/provisioning

Seperti pada Gambar 12 Hasil deployment Terraform pada Branch prod/Provisioning memerlukan waktu total 13 menit 26,407 detik untuk menjalankan ketiga stage dari script pipeline sebelumnya dan menghasilkan cluster dengan satu node. Dalam melakukan suatu penelitian diperlukan uji coba beberapa kali dan mengalami kegagalan sebelum akhirnya berhasil. Pada proses build prod/provisioning telah dilakukan lima puluh lima kali percobaan hingga akhirnya berhasil dan rata rata waktu keseluruhan dalam lima puluh lima kali percobaan tertera pada Average full run time yaitu 7 menit 17 detik.

Branch prod/provisioning

Full project name: TERRAFORM-DEPLOYMENT/prod%2Fprovisioning

Stage View



Gambar 12 prod/provisioning

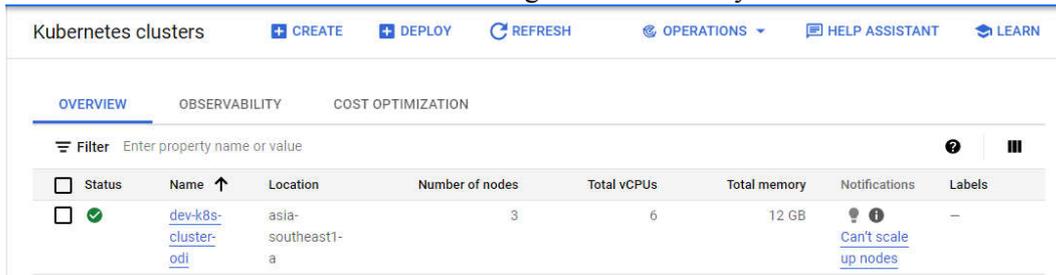
Seperti pada Gambar 13 merupakan Hasil Prod-VPC-Provisioning yang telah dibuild dapat terlihat di GKE, IP subnet yang dihasilkan yaitu 10.116.0.0/16. IP subnet tersebut digunakan di Prod-Provisioning-Cluster.

Name ↓	Region	Subnets	MTU ?	Mode	Internal IP ranges	External IP ranges
prod-vpc-k8s-odi		1	1460	Custom	None	
	asia-southeast1	prod-k8s-subnet-odi			10.116.0.0/16	None
Secondary IPv4 ranges	Gateways	Firewall Rules	Global dynamic routing	Flow logs		
		5	Off			
10.117.0.0/16, 10.1...	10.116.0.1			Off		▼

Gambar 13 Prod-VPC-Provisioning

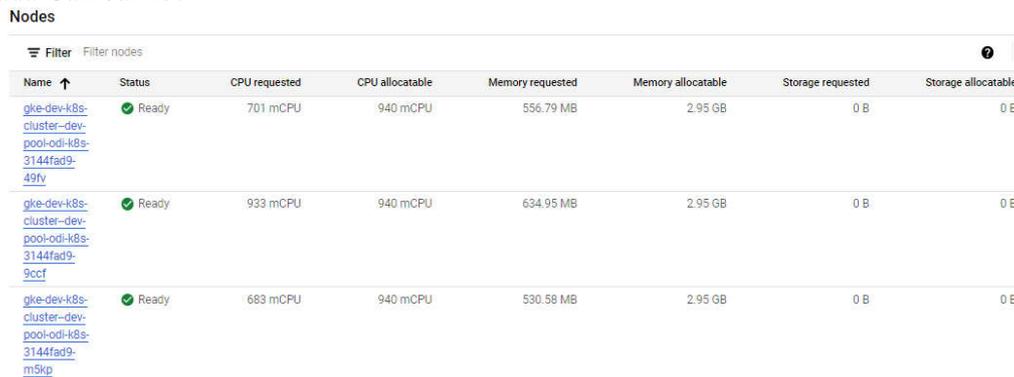
Pada Gambar 14 merupakan hasil provisioning GKE dev cluster yang sudah berhasil terbentuk. GKE dev cluster

yang terbentuk bernama dev-k8s-cluster-odi memiliki tiga node didalamnya.



Gambar 14 GKE Cluster Dev

Pada dev cluster terdapat tiga node yang terbentuk sesuai dengan script yang tertulis di folder main.tf resources dev seperti pada Gambar 15.



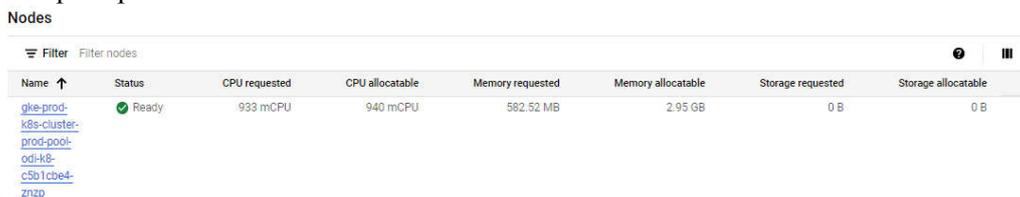
Gambar 15 GKE Node Dev

Pada Gambar 16 merupakan hasil provisioning GKE prod cluster yang sudah berhasil terbentuk. GKE dev cluster yang terbentuk bernama prod-k8s-cluster-odi memiliki satu node didalamnya.



Gambar 16 Node Cluster GKE

Didalam prod cluster terdapat satu node yang terbentuk sesuai dengan script yang tertulis di folder main.tf resources prod seperti pada Gambar 17.



Gambar 17 Node Cluster GKE

IV. KESIMPULAN

Dari penelitian ini dapat disimpulkan bahwa dengan adanya provisioning GKE Cluster menggunakan Terraform dan Jenkins dapat memudahkan tim devops dalam menyiapkan Server sesuai kebutuhan. Terbukti hanya memerlukan waktu 15 menit 18,412 detik untuk menyiapkan satu cluster dengan tiga node pada environment Development dan 13 menit 26,407 detik untuk menyiapkan satu cluster dengan satu node pada environment Production. Dengan adanya sistem ini membantu tim devops untuk menyiapkan server dengan lebih cepat dan minim kendala karena semakin kompleksnya kebutuhan suatu perusahaan.

Adapun saran dari penelitian ini adalah dapat mengembangkan sistem penambahan Terraform Backend untuk listening dan sinkronasi file stage agar lebih dinamis. Selain itu karena keterbatasan dalam akun yang digunakan

dalam penelitian sehingga hanya dapat melakukan pengukuran terbatas, maka dapat dilakukan beberapa kali deployment untuk melihat selisih pada masing masing environment.

DAFTAR PUSTAKA

- [1] A. Farid, "Implementasi CI/CD Pipeline Pada Framework Androbase Menggunakan Jenkins (Studi Kasus: PT. Andromedia)," UMG Repository, 2022.
- [2] N. Ramsari and A. Ginanjar, "Implementasi Infrastruktur Server Berbasis Cloud Computing Untuk Web Service Berbasis Teknologi Google Cloud Platform," SENATIK, vol. VII, 24 Februari 2021.
- [3] Y. Hidayat and B. Arifwido, "Implementasi Web Server Menggunakan Infrastructure As Code Terraform Berbasis Cloud Computing," Jurnal Format, vol. 10, pp. 192 - 201, 2021.
- [4] S. E. Prasetyo and Y. Salimin, "Analisis Perbandingan Performa Web Server Docker Swarm dengan Kubernetes Cluster," Conference on Management, Business, Innovation, Education and Social Science , vol. 1 (1), 2021.
- [5] H. Hata, N. Novielli, S. Baltes, R. G. Kula and C. Treude, "GitHub Discussions: An exploratory study of early adoption," Empirical Software Engineering, vol. 27 (1), 2022.
- [6] Terraform, HashiCorp, "What is Terraform?," HashiCorp Terraform, 25 Oktober 2022. [Online]. Available: <https://developer.hashicorp.com/terraform/intro>. [Accessed 30 Oktober 2022].
- [7] Apache Software Foundation, "Apache Groovy," Apache Software Foundation, 2022. [Online]. Available: <https://groovy-lang.org/>. [Accessed 30 10 2022].
- [8] Google Cloud, "Google Kubernetes Engine (GKE)," Google Cloud, 2007. [Online]. Available: <https://cloud.google.com/kubernetes-engine#section-1>. [Accessed 30 10 2022].
- [9] N.M Agung and S. Cuk, "Analisi Cluster Container Pada Kubernetes Dengan Infrastruktur Google Cloud Platform," Jurnal Ilmiah Penelitian dan Pembelajaran Informatika, vol. 03 (2), 84 - 93, 2021.
- [10] I. Simon. (2021). Get Started with Google Cloud Platform (GCP). In: Getting Started with Containers in Google Cloud Platform . Apress, Berkeley, CA.