

STUDI LITERATUR TERKAIT TECH & TOOLS DEVOPS
MATA KULIAH DEVOPS



ANGGOTA TIM:

ISMAIL ABDURRAHIM (0110221173)

IZZUDIN AL QASAM (0110220077)

MAULANA HISYAM (0110220104)

MUHAMMAD AZKA RIZQI (0110221051)

MUHAMMAD DZAKY ZAHRAN (0110221044)

MUHAMMAD ZAIN RIZQULLOH (0110221017)

RIZWAN NUR SAPUTRA (0110221312)

STT TERPADU NURUL FIKRI
PROGRAM STUDI TEKNIK INFORMATIKA
DEPOK
2023

Penjelasan Terkait *Tech* dan Aplikasi pada DevOps

Aplikasi atau alat untuk DevOps harus memiliki beberapa syarat atau aspek tertentu. Pertama, aplikasi atau alat tersebut harus mampu terintegrasi dengan alat DevOps lainnya, seperti manajemen konfigurasi, alat orkestrasi, alat pengujian, dan alat manajemen siklus hidup perangkat lunak (SDLC). Selain itu, aplikasi atau alat tersebut harus mudah digunakan dan dipahami oleh tim pengembang dan DevOps.

Kemudian, aplikasi atau alat tersebut harus memiliki kemampuan otomatisasi yang kuat untuk mempercepat waktu rilis perangkat lunak. Aplikasi atau alat DevOps yang baik juga harus dapat mendukung lingkungan cloud computing, seperti Amazon Web Services (AWS) atau Microsoft Azure, sehingga dapat memudahkan tim untuk melakukan deploy perangkat lunak ke lingkungan cloud.

Terakhir, aplikasi atau alat tersebut harus memiliki dukungan dan fitur untuk sistem manajemen versi, seperti Git atau SVN, untuk memudahkan tim dalam mengelola dan mengontrol versi perangkat lunak. Dengan memenuhi syarat-syarat tersebut, aplikasi atau alat DevOps dapat membantu tim pengembang dalam meningkatkan efisiensi dalam proses pengembangan dan pengiriman perangkat lunak.

1. Plan and Track

1.1 Azure Boards

Azure boards merupakan sebuah aplikasi yang berfungsi untuk membantu merencanakan dan melihat progres dari hal yang sedang dikerjakan. Penjelasan sederhananya azure boards menyediakan kartu visual untuk diisi sesuai dengan tujuan yang sedang dikejar lalu melihat progres hal-hal yang dikerjakan dan juga melihat kinerja tim.

Azure boards membantuk tim untuk melihat progres mana yang belum selesai, pekerjaan mana yang belum selesai, pekerjaan yang sudah selesai, pekerjaan yang tertunda serta kinerja tim. Visual yang dihadirkan menampilkan bar progres yang membantu melihat seberapa jauh perkembangan dari pekerjaan yang sedang dilakukan.

Azure boards membantu melacak pekerjaan yang sedang dikerjakan, ditambah lagi dapat diintegrasikan dengan aplilkasi lain seperti text editor yang mengelola code aplikasi proyek yang sedang digunakan sehingga bisa terlacak progresnya secara langsung tanpa perlu di cek secara manual. Azure boards juga menyediakan fitur-fitur yang sangat membantu seperti pelacakan bug sehingga memungkinkan tim untuk memperbaiki kesalahan atau meminimalisir resiko dari bug tersebut.

Dengan semua fitur dan layanan yang diberikan semua menjadi mudah karena semua dapat dilihat dalam satu dashboard sehingga meningkatkan kolaborasi antar anggota tim dalam mengerjakan tugasnya.

(Ismail Abdurrahim)

1.2 Jira

JIRA adalah alat pengembangan perangkat lunak yang digunakan insinyur untuk melacak dan mengelola tugas. Anda dapat menggunakannya untuk manajemen proyek Agile dan waterfall untuk melacak bug, fitur, dan item pekerjaan lainnya. Anda juga dapat mengonfigurasi JIRA untuk bekerja dengan banyak alat manajemen layanan, menjadikannya alat serbaguna untuk mengelola tugas.

(Maulana Hisyam)

2. Code

2.1 Github

Github adalah sebuah alat penting yang mempunyai VCS atau version control system. VCS ini menyediakan suatu layanan yang fungsinya adalah untuk menyimpan berbagai jenis repository atau file yang berbentuk source code. Github memungkinkan terwujudnya suatu kolaborasi saat mengerjakan suatu proyek tertentu sehingga engineer bisa menguji coba pembaruan secara rutin, menyimpan berbagai perubahan dalam proyek tersebut secara online, dan memperbaiki bug.

Sederhananya, Github membuatmu bisa menguji coba update secara rutin, memperbaiki bug, hingga melacak dan menyimpan perubahan-perubahan yang terjadi dalam proyek tersebut secara online. Selain itu, GitHub juga memungkinkan kamu berkolaborasi dengan orang lain untuk mengerjakan satu proyek yang sama.

(Rizwan Nur Saputra)

2.2 Bitbucket

BitBucket adalah platform hosting Git yang populer digunakan oleh pengembang perangkat lunak dan tim pengembangan untuk menyimpan dan mengelola kode mereka secara terpusat. Fasilitas dan fitur-fitur yang ada di BitBucket sangat bervariasi dan dirancang untuk membantu tim pengembang dalam mengembangkan perangkat lunak dengan lebih mudah dan efisien.

Beberapa fasilitas dan fitur-fitur yang ada di BitBucket antara lain adalah:

Version Control: BitBucket menyediakan sistem kontrol versi yang memungkinkan pengembang untuk mengelola kode sumber mereka dan melakukan revisi pada kode sumber.

Collaborative coding: Fitur ini memungkinkan anggota tim untuk bekerja bersama dan berkolaborasi pada kode sumber yang sama.

Code review: BitBucket menyediakan fitur untuk meninjau kode sumber sebelum digabungkan ke cabang utama. Fitur ini membantu memastikan kualitas kode sumber dan mencegah masalah yang tidak diinginkan.

Issue tracking: BitBucket memiliki fitur pelacakan masalah (issue tracking) yang memungkinkan pengembang untuk mengelola dan melacak masalah di proyek mereka.

Continuous Integration (CI) and Continuous Deployment (CD): BitBucket dapat diintegrasikan dengan alat CI/CD untuk mengotomatisasi proses pengujian dan distribusi

(Izzuddin Al Qasam)

3. Build and Test

3.1 Github Action

GitHub Actions adalah sebuah fitur yang disediakan oleh GitHub untuk otomatisasi dalam pengembangan perangkat lunak. Dengan menggunakan GitHub Actions, pengguna dapat membuat dan menjalankan serangkaian tindakan otomatis yang terdiri dari beberapa langkah atau tugas, yang disebut dengan workflow. Workflow ini dapat digunakan untuk mempermudah proses pengembangan perangkat lunak, pengujian, dan penerapan secara efisien dan efektif.

Untuk membuat workflow, kita menggunakan file YAML yang terdapat di dalam repository. File ini berisi konfigurasi untuk tindakan apa yang akan dilakukan, seperti pengujian atau penerapan kode, serta kapan tindakan tersebut akan dijalankan, seperti saat kode di-push ke repository. Dengan demikian, kita dapat mengatur alur kerja dan mengotomatiskan tugas-tugas yang berulang, sehingga dapat menghemat waktu dan meningkatkan produktivitas.

Salah satu keunggulan dari GitHub Actions adalah kemudahannya dalam penggunaan dan integrasi dengan platform GitHub. GitHub Actions sudah terintegrasi dengan platform GitHub, sehingga tidak perlu melakukan konfigurasi yang rumit. Selain itu, GitHub Actions menyediakan berbagai macam tindakan yang dapat digunakan secara langsung, seperti tindakan untuk pengujian, penerapan, notifikasi, dan sebagainya, yang memudahkan untuk membangun workflow yang tepat untuk kebutuhan.

GitHub Actions juga mendukung banyak bahasa pemrograman dan lingkungan pengembangan, seperti JavaScript, Python, Ruby, Docker, dan lain sebagainya. Hal ini memudahkan untuk membuat workflow sesuai dengan lingkungan pengembangan. Selain itu, pengguna juga dapat memanfaatkan

repository publik di GitHub untuk mengambil contoh workflow yang sudah tersedia dan mengadaptasinya sesuai dengan kebutuhan kita.

(Ismail Abdurrahim)

3.2 Azure Pipelines

Azure Pipelines merupakan salah satu layanan yang disediakan oleh Azure DevOps perusahaan yang sama yang memberikan layanan Azure Boards. Layanan Azure Pipelines merupakan sebuah layanan yang membantu dalam otomatisasi pengujian, pengembangan dan pengiriman aplikasi dengan lebih cepat.

Konsep pada Azure Pipelines adalah menguji kode sumbernya lalu menguji kode dan jika sudah diuji maka aplikasi bisa dikirim untuk deployment. Layanan ini akan menguji sesuai dengan ketentuan standar yang telah ditentukan oleh tim sehingga kualitasnya akan terjaga, pastinya kode sumber akan dihubungkan ke layanan ini agar setiap perubahan yang dilakukan maka akan diuji untuk melihat kualitasnya.

Layanan Azure Pipelines dapat diintegrasikan dengan layanan lain yang disediakan oleh Azure DevOps yaitu Azure Boards dan Azure Repos yang menjadikan ekosistem lebih baik. Dengan pendukungan berbagai macam bahasa pemrograman menjadikan Azure Pipelines membantu dalam menyajikan aplikasi yang berkualitas dalam waktu yang cepat dan efisien.

(Ismail Abdurrahim)

4. Deploy

4.1 Arm Templates

Azure Resource Manager Templates (ARM Templates) adalah fitur yang disediakan oleh Microsoft Azure untuk melakukan deployment atau pengaturan infrastruktur secara otomatis pada lingkungan Azure. ARM Templates merupakan file JSON yang digunakan untuk mendefinisikan sumber daya yang akan dibuat dan diatur pada lingkungan Azure. ARM Templates memungkinkan pengguna untuk melakukan deployment atau pengaturan infrastruktur secara konsisten, berulang, dan dapat diulang dengan mudah.

Setiap ARM template terdiri dari beberapa bagian, yaitu:

- a. Parameters: bagian ini berisi nilai-nilai yang diperlukan oleh template dan dapat diubah sesuai kebutuhan pengguna. Contoh

parameter yang umum digunakan adalah nama sumber daya, ukuran virtual machine, atau lokasi geografis.

- b. Variables: bagian ini berisi variabel-variabel yang digunakan dalam template, seperti nilai default atau kombinasi dari nilai parameter.
- c. Resources: bagian ini berisi definisi sumber daya Azure yang akan dibuat dan diatur. Setiap sumber daya didefinisikan dalam format JSON dan harus memiliki jenis (type) yang sesuai dengan jenis sumber daya yang dibuat.
- d. Outputs: bagian ini berisi nilai-nilai yang dihasilkan dari template dan dapat digunakan oleh pengguna sebagai input untuk template lain atau sebagai informasi hasil deployment.

ARM Templates dapat diatur dan dijalankan menggunakan Azure Portal, Azure PowerShell, atau Azure CLI. Dalam pengaturannya, pengguna hanya perlu mengunggah file JSON template ke Azure dan memilih konfigurasi yang diperlukan. Setelah deployment selesai, sumber daya akan langsung tersedia dan dapat digunakan sesuai kebutuhan.

Berikut ini adalah contoh kecil ARM Template yang hanya mendefinisikan sebuah virtual network (VNet) pada lingkungan Azure:

```
{
"$schema":
"https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.js
on#",
  "contentVersion": "1.0.0.0",
  "resources": [
    {
      "type": "Microsoft.Network/virtualNetworks",
      "apiVersion": "2020-06-01",
      "name": "myVNet",
      "location": "eastus",
      "properties": {
        "addressSpace": {
          "addressPrefixes": [
```

```

        "10.0.0.0/16"
    ]
},
"subnets": [
    {
        "name": "mySubnet",
        "properties": {
            "addressPrefix": "10.0.0.0/24"
        }
    }
]
}
}
]
}

```

JSON di atas hanya berisi definisi sebuah virtual network dengan nama "myVNet" dan satu subnet dengan nama "mySubnet". Template ini dapat dijalankan pada Azure Resource Manager untuk membuat sebuah virtual network sesuai dengan definisi yang tertera pada JSON tersebut.

(Muhammad Dzaky Zahran)

4.2 Terraform

Terraform adalah tools open-source untuk mengelola infrastruktur sebagai kode. Dengan Terraform, infrastruktur dapat didefinisikan sebagai kode dan diotomatiskan, sehingga memungkinkan pengguna untuk mengontrol pengaturan, pengujian, dan infrastruktur deployment secara konsisten. Terraform mendukung berbagai penyedia layanan infrastruktur seperti AWS, GCP (Google Cloud Platform), dan Azure.

Konfigurasi pada Terraform dibuat dalam file dengan ekstensi .tf. Konfigurasi file tersebut mengandung definisi infrastruktur yang akan dibangun, termasuk penyedia layanan infrastruktur yang akan digunakan.

Beberapa elemen penting dalam konfigurasi Terraform adalah:

- a. Provider: Menentukan penyedia layanan infrastruktur yang digunakan, seperti AWS, GCP, atau Azure.

- b. Resource: Mendefinisikan jenis infrastruktur yang akan dibuat, seperti instance EC2 di AWS, virtual machine di Azure, atau bucket S3 di AWS.
- c. Variable: Mendefinisikan variabel yang digunakan dalam konfigurasi, seperti nama instance atau jumlah instance.
- d. Output: Mendefinisikan output yang dihasilkan oleh konfigurasi, seperti alamat IP atau nama host.

Berikut ini adalah contoh konfigurasi Terraform untuk membuat sebuah instance EC2 pada AWS

```
provider "aws" {  
    region = "us-west-2"  
}  
  
resource "aws_instance" "example" {  
    ami      = "ami-0c55b159cbfafa1f0"  
    instance_type = "t2.micro"  
}
```

Konfigurasi tersebut menentukan bahwa infrastruktur akan dibuat pada penyedia layanan AWS di region us-west-2, dan akan dibuat sebuah instance EC2 dengan jenis t2.micro dan menggunakan AMI (Amazon Machine Image) dengan ID ami-0c55b159cbfafa1f0.

(Muhammad Azka Rizqi)

4.3 Chef

Chef adalah salah satu tools yang memungkinkan pengguna dapat melakukan konfigurasi dan otomasi infrastruktur. Chef biasanya digunakan oleh profesional IT dalam mengelola dan mengotomatisasi sistem, dengan tujuan agar lebih efisien.

Berikut merupakan fasilitas-fasilitas Chef:

- a. Chef infra Server

Chef Infra Server adalah komponen penting dari platform manajemen konfigurasi Chef Infra. Chef Infra Server bertindak sebagai

pusat kontrol untuk pengaturan dan manajemen konfigurasi pada server dan mesin virtual dalam sebuah infrastruktur.

Dalam konteks Chef Infra, konfigurasi berarti menentukan bagaimana perangkat lunak dan layanan harus diatur dan dijalankan pada setiap mesin. Chef Infra Server menyimpan konfigurasi dan aturan Chef dalam sebuah repositori sentral, dan menyediakan antarmuka web yang memungkinkan administrator untuk menentukan konfigurasi dan aturan secara terpusat.

Chef Infra Server juga bertanggung jawab untuk menyediakan manajemen identitas, manajemen akses, dan manajemen layanan. Hal ini memungkinkan administrator untuk mengontrol akses ke konfigurasi dan aturan Chef, serta memantau dan melacak perubahan pada infrastruktur.

Chef Infra Server berjalan pada platform perangkat lunak open-source yang dapat diinstal di mesin fisik atau virtual. Chef Infra Server juga dapat diintegrasikan dengan platform cloud seperti AWS, Azure, dan Google Cloud Platform, sehingga memudahkan administrator untuk mengatur konfigurasi pada infrastruktur cloud mereka.

b. Chef infra Client

Chef Infra Client adalah agen yang diinstal pada setiap mesin yang ingin dikonfigurasi dan dikelola oleh Chef Infra Server. Chef Infra Client terhubung ke Chef Infra Server untuk mendapatkan konfigurasi dan aturan terbaru, dan mengeksekusinya pada mesin target. Chef Infra Client juga memberikan informasi tentang status mesin target ke Chef Infra Server.

Chef Infra Client menggunakan bahasa konfigurasi Chef DSL untuk menjalankan konfigurasi dan aturan yang fleksibel dan mudah dipelajari.

c. Chef Workstation

Chef Workstation adalah perangkat lunak desktop yang digunakan oleh para pengembang untuk membuat, mengelola, dan memelihara konfigurasi infrastruktur mereka secara otomatis. Perangkat lunak ini berfungsi sebagai platform pengembangan dan distribusi kode otomatis untuk mempercepat pengembangan dan penerapan aplikasi serta mengelola infrastruktur IT. Chef Workstation dapat diinstal pada berbagai sistem operasi, termasuk Windows, MacOS, dan Linux, dan dilengkapi

dengan berbagai fitur dan alat untuk membantu pengguna dalam mengelola infrastruktur mereka.

d. Chef inSpec

Chef InSpec adalah sebuah tool open-source yang digunakan untuk melakukan testing, compliance, dan auditing pada infrastruktur dan aplikasi yang dikelola dengan menggunakan Chef. Tool ini dapat digunakan untuk melakukan pengujian pada berbagai platform, termasuk cloud, container, dan juga on-premise.

Chef InSpec menyediakan banyak fitur, diantaranya adalah kemampuan untuk memeriksa konfigurasi pada sistem operasi, memeriksa keamanan, melakukan pengujian fungsional, serta melakukan audit pada konfigurasi aplikasi. Selain itu, Chef InSpec juga mendukung integrasi dengan tool lain seperti Chef Automate dan Jenkins.

e. Chef Habitat

Chef Habitat merupakan Sebuah platform yang berfungsi sebagai deployment dan manajemen aplikasi pada berbagai lingkungan seperti cloud, on-premise, dan baremetal.

Chef Habitat merupakan solusi otomasi platform-agnostik untuk menerapkan dan mengelola aplikasi di semua infrastruktur, termasuk pusat data tradisional, cloud pribadi atau publik, dan container. Chef Habitat juga terintegrasi dengan produk Chef lainnya, seperti Chef Infra dan Chef inSpec, demi menyediakan solusi otomasi komprehensif untuk pengembangan dan aplikasi modern.

(Muhammad Zain Rizqullah)

5. Operate

5.1 Policy

Policy merujuk pada serangkaian aturan dan standar yang diterapkan pada lingkungan IT. Ini dapat mencakup kebijakan keamanan, peraturan privasi, dan persyaratan regulasi yang harus dipatuhi oleh organisasi. Dalam konteks DevOps, policy digunakan untuk memastikan bahwa aplikasi dan sistem yang dihasilkan oleh tim DevOps memenuhi standar kualitas, keamanan, dan kinerja yang diharapkan.

Dalam DevOps, policy juga digunakan untuk memastikan keseragaman dan konsistensi dalam pengembangan dan pengiriman perangkat lunak.

Aplikasi policy yang biasa digunakan untuk DevOps adalah alat untuk menerapkan dan memastikan bahwa kebijakan organisasi atau standar keamanan diikuti dalam seluruh siklus hidup aplikasi dan infrastruktur. contoh aplikasinya adalah Chef, HashiCorp, Open Policy Agent, dan Sysdig Secure.

(Muhammad Zain Rizqulloh)

5.2 Ansible Tower

Ansible Tower adalah sebuah platform manajemen otomatisasi IT yang menyediakan antarmuka grafis dan pengaturan manajemen sentral untuk memudahkan pengguna dalam penggunaan Ansible. Ansible Tower dirancang untuk mengatur dan memantau berbagai aktivitas otomatisasi di seluruh lingkungan infrastruktur IT. Ansible Tower menyediakan dasbor interaktif yang memungkinkan pengguna mengelola host inventaris, mengatur tugas dan jadwal, melihat aktivitas log, dan mengelola hak akses pengguna. Pengguna dapat menggunakan antarmuka grafis Tower untuk membuat playbook Ansible dan menjalankannya pada host atau grup host yang dipilih dengan mudah.

(Muhammad Azka Rizqi)

6. Monitor

6.1 ELK Stack

ELK Stack merupakan produk open source dari Elastic yang dirancang untuk membantu pengguna dalam mengambil data dari semua jenis sumber dan dalam format apapun. Selain itu, ELK Stack juga mampu mencari, menganalisis, dan memvisualisasikan data secara real time. ELK terdiri dari tiga tools open source, diantaranya Elasticsearch, Logstash, dan Kibana. Alat ketiga ini memilikinya sendiri-sendiri.

(Muhammad Azka Rizqi)

6.2 Splunk

Splunk adalah sebuah platform perangkat lunak analisis data dan pencatatan log (manajemen log) yang berfokus pada operasional data. Splunk digunakan untuk mengumpulkan, mengindeks, mencari, dan menganalisis data dari berbagai sumber seperti server, aplikasi, perangkat jaringan, dan sistem IoT (Internet of Things).

Dengan Splunk, pengguna dapat memantau dan menganalisis data secara real-time atau near real-time, serta mengekstraksi informasi yang berguna dari data yang terkumpul. Splunk juga dapat menggabungkan data dari berbagai sumber dan memungkinkan pengguna untuk membuat dashboard, grafik, dan laporan interaktif untuk memvisualisasikan data dan mencari informasi yang penting.

(Muhammad Azka Rizqi)