DEVOPS

1. Definisi

DevOps adalah pekerjaan mengotomatiasasi pengembangan software dengan tim pengembang agar dapat menjalankan siklus pengembangan perangkat lunak mulai dari build, test dan release secara lebih efektif efisien.

1. Proses

DevOps Proses Flow mengutamakan otomatisasi dan agile. Continuous Development adalah istilah umum yang menggambarkan proses berulang untuk mengembangkan perangkat lunak yang akan dikembangkan. Ini meliputi Continuous Integration, Continuous Delivery, Continuous Deployment, Continuous Management, Infrastructure as a Code, Monitoring, Logging dan Communication & Collaboration.

1. Continuous Integration

Continuous Integration adalah praktek software development dimana para pengembang dengan regular melakukan perubahan kode dan melakukan merge pada sebuah repository, dan ketika terdapat push commit baru, automated build & test dijalankan, dan hasil akan segera diterima. Jika terjadi kesalahan/bug dalan build/test, tim pengembang dapat dengan cepat melakukan update code sehingga kualitas produk bisa ditingkatkan, dan mengurangi waktu untuk melakukan validasi sebuah update.

1. Continuous Delivery

Continuous Delivery adalah praktek software development dimana para pengembang yang melakukan perubahan pada code, sudah melakukan build & test yang dijalankan otomatis oleh Continuous Integration dan siap untuk deploy ke environment production namun memilih untuk tidak deploy yang umumnya karena masalah bisnis. Ketika Continuous Delivery dijalankan dengan baik, tim pengembang akan selalu memiliki build artifact yang siap untuk di deploy ke environment production.

1. Continuous Deployment

Continuous Deployment merupakan salah satu rangkaian setelah Continuous Integration dan Continuous Delivery selesai dijalankan. Umumnya organisasi/perusahaan memiliki environment test / development, dan disinilah fungsi utama continuous deployment, yaitu ketika hasil dari Continuous Integration sudah dinyatakan baik, tim pengembang dapat segera melihat perubahan pada environment test / development / production.

Continuous Deployment merupakan cara yang sangat baik untuk meningkatkan jumlah feedback yang diterima oleh pengguna aplikasi dan mengurangi tekanan karena tidak ada lagi "Hari Rilis". Tim pengembang dapat fokus dalam membangun produk dan dapat melihat perubahan code mereka beberapa menit setelah mereka selesai mengerjakannya.

Aplikasi populer yang mendukung dalam Continuous Integration/Continuous Delivery/Continuous Deployment diantaranya [Jenkins](https://jenkins.io/), [Gitlab CI/CD](https://about.gitlab.com/features/gitlab-ci-cd/) [Bitbucket Pipeline](https://bitbucket.org/product/features/pipelines), [Semaphore CI](https://semaphoreci.com/), [Travis CI](https://travis-ci.org/), [Circle CI](https://circleci.com/), [AWS Code Build](https://aws.amazon.com/codebuild/) dan [AWS Code Pipeline](https://aws.amazon.com/codepipeline/).

1. Configuration Management

Configuration Management adalah praktek dalam proses System Engineering yang memiliki tujuan untuk me-maintain konfigurasi sebuah produk, dan memastikan konsitensinya dalam seluruh environment. Dengan menggunakan Configuration Management, proses konfigurasi produk dapat diotomatisasi, distandardisasi dan mengurangi proses konfigurasi yang manual. Tahap selanjutnya, Configuration Management akan mempermudah dalam konfigurasi banyak server dan dapat meminimalisir kesalahan, karena konfigurasi ditulis dalam code, tidak lagi menjalankan perintah manual.

Aplikasi populer yang mendukung dalam Configuration Management diantaranya [Ansible](https://www.ansible.com/), [Chef](https://www.chef.io/), [Puppet](https://puppet.com/products/open-source-projects), [Capistrano](http://capistranorb.com/), [Fabric](http://www.fabfile.org/) dan [SaltStack](https://saltstack.com/salt-open-source/).

1. Infrastructure as a Code

Infrastructure as a Code adalah sebuah praktek dalam System Architecture yang mana infrastruktur sebuah produk didefinisikan dalam code yang dapat diprogram, distandardiasikan dan mudah untuk di duplikasi. Produk skala menengah, mungkin membutuhkan lebih dari satu mesin. Dengan IAAC, tim pengembang dapat dengan mudah menambah mesin melalui satu baris kode.

Produk dalam skala besar, umumnya bergantung pada beberapa produk yang ditawarkan olehprovider cloud, seperti provider aws yang menawarkan produk seperti [AWS EC2](https://aws.amazon.com/ec2/) dan [AWS Kinesis Data Stream](https://aws.amazon.com/kinesis/data-streams/). Dengan IAAC, tim pengembang tidak lagi melakukan operasi manual pada console provider cloud, cukup dengan merubah beberapa baris kode, hal yang lebih penting yaitu dapat meminimalisir kesalahan.

Aplikasi populer yang mendukung IAAC diantaranya [Terraform](https://www.terraform.io/), [Ansible Tower](https://www.ansible.com/products/tower), [Chef](https://www.chef.io/), [Puppet](https://puppet.com/products/open-source-projects), [SaltStack](https://saltstack.com/salt-open-source/) dan [AWS CloudFormation](https://aws.amazon.com/cloudformation/).

1. Monitoring

Sebuah produk haruslah di-monitoring untuk mengetahui bagaimana produk digunakan oleh pengguna. Dalam praktek DevOps, monitoring merupakan hal yang sangat penting. Tim pengembang arus mengetahui bagaimana perubahan kodenya berdampak ada produk juga penggunanya melalui monitoring tools.

Aplikasi monitoring yang populer diantaranya [Nagios](https://www.nagios.org/), [Zabbix](https://www.zabbix.com/), [Sensu](https://sensuapp.org/), [Cacti](https://www.cacti.net/), [OpenNMS](https://opennms.org/), [Icinga](https://www.icinga.com/), [Prometheus](https://prometheus.io/), [Grafana](https://grafana.com/) dan [AWS CloudWatch](https://aws.amazon.com/cloudwatch/).

1. Logging

Log aplikasi adalah salah satu cara untuk mengetahui apakah produk kita berjalan dengan baik atau tidak. Namun seiring dengan tingkat kompleksitas sebuah produk, ada banyak log komponen yang harus diterima dan dianalisis. Dan log tersebut haruslah terpusat, tidak terpisah-pisah. Maka istilah Centralized Logging merupakan hal yang tidak dapat dipisahkan dalam praktek DevOps.

Aplikasi logging yang populer diantaranya [ElasticSearch + Logstash + Kibana](https://www.elastic.co/elk-stack), [Splunk](https://www.splunk.com/), [Graylog](https://www.graylog.org/), [AWS Cloudwatch](https://aws.amazon.com/cloudwatch/) dan [AWS Kinesis Data Firehose](https://aws.amazon.com/kinesis/data-firehose/).

1. Communication & Collaboration

Salah satu aspek utama dalam praktek DevOps yaitu meningkatnya komunikasi dan kolaborasi dalam sebuah organisasi/perusahaan, baik dalam bentuk fisik maupun non fisik. Praktek DevOps yang berjalan dengan baik, akan meningkatkan aspek komunikasi dan kolaborasi tidak hanya pada tim pengembang, namun juga tim marketing, sales, operations, dan tim lain yang ada didalam organisasi/perusahaan.

[Slack](https://slack.com/) adalah aplikasi yang banyak digunakan sebagai sebuah wadah untuk ber-komunikasi dalam bentuk chat. Dengan banyaknya jumlah [integrasi](https://slack.com/apps) yang bisa dilakukan oleh slack, tidak heran slack menjadi favorit. Aplikasi seperti [Skype](https://web.skype.com/), [Google Hangouts](https://hangouts.google.com/) dan [Google Meets](https://meet.google.com/) masih menjadi favorit untuk komunikasi dalam bentuk video/voice call. Tools kolaborasi yang banyak menjadi favorit diantaranya [Trello](https://trello.com/), [Jira](https://www.atlassian.com/software/jira), [Pivotal Tracker](https://www.pivotaltracker.com/) dan [Meistertask](https://www.meistertask.com/).