Timeline

Description automatically generated

Popular DevOps tools review.

I am listing some of the DevOps technologies and tools used these days. Don’t worry if you are new to DevOps and containerization, because I will be using simple and as plain language as possible. I shall be mentioning some of the popular ones, and the ones I mentioned may not be ‘the most popular’ furthermore I will mention many of them only only by name and short definition due to article’s volume restriction.

We are witnessing a rapid advancement of DevOps technologies these days therefore today’s popular and powerful tool may become next years’ not so popular one and may even be quickly replaced by another. View foregoing, we are aiming with this short article not to judge which ones are most popular or functional, but to give readers an idea how the DevOps tools work and how the tools are used in the process.

**What is DevOps?**

DevOps is the practise of development and IT operations engineers participate together in entire agile software development lifecycle from design through the development and to production. Agile is software development lifecycle principals declared by 17 independent software developers in 2001. Agile manifesto is very short and straight to the point as follows:

# “Manifesto for Agile Software Development

# We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools  
Working software over comprehensive documentation  
Customer collaboration over contract negotiation  
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

Please fer to <https://agilemanifesto.org/> for more information.

Over the last 20 years since Agile manifesto declared, agile principals agreed by vast number of organisations and some adopted Scrum framework for their project management. Sprint concept of scrum framework and increasingly product centric culture as well as time pressure imposed by fast moving competitive markets forced many software developers to deliver product incrementally and quickly. Please refer to <https://www.scrum.org/resources/what-is-scrum> for more info regarding scrum. Continuous delivery required continuous integration and CI/CD forced producers to adopt new tools and technologies so that many tasks could be automated and more than one processes can be carried out simultaneously and quickly. Emerging newly adopted software development lifecycle culture resulted developers, quality assurance, IT operations, product designers, and product owners to work ever closer. And that is where DevOps concept became inevitable.

On top of traditional version control, collaboration and deployment tools, rapid production and delivery forced software companies to use more tools to meet agility expectation such as project management, automation, containerisation, orchestration, monitoring, analytics and visualisation technologies.

**DevOps tools:**

1. **Version Control Systems:**

Git; Git is an Open Source Distributed Version Control System. Please refer to <https://git-scm.com/about> for more information.

GitHub; GitHub is a Git repository hosting service.

Bitbucket; Git-based source code repository hosting service by Atlassian.

Apache Subversion (SVN); Apache Subversion is a software versioning and revision control system distributed as open source under the Apache License.

Azure DevOps Server; Azure DevOps Server is a Microsoft product that provides version control.

There are more version control technologies such as And the other are Gitlab, Mercurial, Bazaar, AWS CodeCommit, Perforce Helix Core and SourceForge.

1. **Containerisation:**

2a - **Docker:** Docker is a containerization software that performs operating-system-level-virtualization. The developer of this software is Docker, Inc. The initial release of this software happened in the year 2013. It is written in ‘Go’ programming language.

Container: A container is a standard unit of software that packages the code and all its dependencies, so the application can run quickly and reliably from one environment to another.

Container images: A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for Linux as well as Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Docker Engine: Docker container technology was launched in 2013 as an open-source Docker Engine. Docker's technology is unique because it focuses on the requirements of developers and systems operators to separate application dependencies from infrastructure.

Technology available from Docker and its open source. Integrated into cloud technologies by all major data centre vendors and cloud providers. Many of these providers are leveraging Docker for their container-native IaaS offerings. Additionally, the leading open source serverless frameworks utilize Docker container technology.

2b – **Amazon ECS**: Amazon Elastic Container Service is an orchestration service that supports Docker containers run to execute containerised applications on Amazon AWS. Eradicates installation and management of separate container orchestration software. Although ESC is free of charger Amazon still charges for the resources used.

2c - **AWS Fargate**: AWS Fargate lets you execute containers without needing manage servers or clusters. Amazon charges for memory and virtual CPU resources used to run Fargate.

2d- **Google Cloud platform**: Google cloud provides you with different options to choose from for running the containers. These are Google Kubernetes Engine (for container cluster management), Google Compute Engine (for Virtual Machines and CI/CD pipeline) and Google App Engine Flexible Environment (for containers on fully managed PaaS). Google provides production ready infrastructure for implementing containerised applications. Google charges for the resources used.

2e – **LXC**: LXC is the acronym for Linux Containers which is a type of OS-level virtualization method for executing numerous isolated Linux systems (containers) sitting on a control host employing a single Linux Kernel. This is an open source and free of charge tool under the GNU LGPL License. It is available on the GitHub Repository.

2f – **Microsoft Azure**: Similar to AWS ECS, no charge for container management but charges for what you use.

2g – **Apache Mesos**: Developed by Apache Software Foundation, Apache Mesos is an open-source project to handle computer clusters. First released in 2016 and it is written in C++ programming language. Free to use.

2h – **Docker compose:** Docker Compose is used for running multi container applications. Docker-compose describes groups of interconnected services that share software dependencies and are orchestrated and scaled together. You can use a single YAML to configure your application’s services. Then, with a docker-compose up command, you create and start all the services from your configuration.

1. **Container orchestration**:

With big corporations containerizing their applications and moving them to the cloud, there is a growing demand for container orchestration solutions. There are many solutions available, although not exhaustive, the list below provides a few different container orchestration tools and services available.

3a – **Kubernetes**: Kubernetes is an open-source platform supports automation of deployment, scaling and management of containerised services. Kubernetes originally developed by google and maintained by the Cloud Native Computing Foundation. There are number of paid or free to use products available, most popular of them is Kubernetes.

Deploying Kubernetes gives us a cluster. A Kubernetes cluster consists of a set of worker machines, called nodes, that run containerized applications. Every cluster has at least one worker node. Each node is managed by the control plane and contains the services necessary to run pods. Each pod is a logical host for a container. The worker node(s) host the pods that are the components of the application workload. The control node manages the worker nodes and the Pods in the cluster. Control plane and nodes can be physical or virtual machines. It is not unusual to have both control plane and nodes run across multiple computers.

3b – **Docker swarm**: With Docker Swarm mode we can define the desired state of the service, and Docker will maintain that state. Docker Enterprise Edition has integrated Kubernetes with Swarm. Docker is now providing flexibility in the choice of orchestration engine. Docker engine CLI is used to create a swarm of docker engines where application services can be deployed.

3c – **Minikube:** Minikube allows you to use Kubernetes on your local machine and allows you to use single node Kubernetes cluster on your computer.

3d – **Rancher**: Rancher is an open-source platform that uses container orchestration known as cattle. It allows you to leverage orchestration services like Kubernetes, Swarm, Mesos. Rancher provides the software required to manage containers so that organizations don’t need to build container services platforms from scratch using a distinct set of open-source technologies. Rancher allows the management of Kubernetes clusters running on the customer-specified providers.

3e – **AWS EKS** (Amazon’s fully managed Kubernetes service).

3f - **AKS** (Microsoft Azure’s Fully managed Kubernetes service). Azure Container Instance (ACI) is a basic container orchestration service provided by Microsoft Azure. Azure Service Fabric is an open-source container orchestrator provided by Microsoft Azure.

3g – **Marathon**: Marathon is a framework to run containers at scale on Apache Mesos.

3h – **Nomad**: Nomad is the container and workload orchestrator provided by HashiCorp.

1. **IT automation:**

4a – **Jenkins**: Jenkins is the open-source, free to use, platform agnostic automation server. Jenkins offers several plugins to support building, deploying, and automating any project. It is a self-contained Java-based program. It supports Windows, Mac, and other UNIX like operating systems. Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed.

4b – **Chef**: Chef INFRA is a platform for automating the infrastructure configuration. It will ensure every system is configured correctly and consistently. With this platform, infrastructure is defined as a code. Chef INFRA will make sure that the configuration policy is flexible, version able, testable, and human-readable. Chef is a contract based paid platform.

4c- **Puppet**: It can connect the cloud providers, DevOps tools, and other APIs. According to the signals from your existing DevOps tools, Relay triggers the workflow to orchestrate actions on downstream services. It will let you customize and extend workflows through YAML-based configuration. It has an integration library which makes it easier to add steps. With Puppet Enterprise you can deliver and manage any cloud, infrastructure, or service. Puppet is a contract based paid platform.

4d - **Ansible**: Ansible is an open-source IT automation tool by RedHat. Ansible can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployment. Ansible uses playbooks for configuration management and multi-machine deployment system.

4e - **Terraform**: Terraform is an open-source infrastructure as code software tool by Hashi Corp. Terraform is a tool to build, change and manage infrastructure. Terraform codifies cloud APIs into declarative configuration files. You can use **Terraform** to manage environments with a configuration language called the **HashiCorp Configuration Language (HCL)** for human-readable, automated deployments.

1. **Collaboration**:

5a – **Jira**: Jira is a tool developed to help teams for project management, bug tracking, and issue tracking. In simple terms, it is an **issue tracker**. Jira is widely used by big companies in software development and software testing. It is web-based and licensed product created by Australian Company Atlassian, first released in 2002, widely popular and free version available for limited usage. Jira software used by agile teams, designed for methodologies such as scrum and Kanban. Jira also widely used by financial services, retail, software, automotive, non-profit organisations and governments.

5b – **Slack**: Slack is a proprietary business communication platform developed by Slack Technologies. Slack offers many IRC-style features, including persistent chat rooms organized by topic, private groups, direct messaging keeping files in one place. Slack claims that it's faster, better organised and more secure than email.

5c- **Monday**: Monday.com is project management tool that used by companies for task management, project collaboration, process tracking and communication.

5d- **Google workspace**: Google workspace is Google’s app-based productivity platforms combined. Messaging, calendar, file sharing, chat, video meeting, team organisation tools put together and offered free to use basis by Google to all its users.

1. **Monitoring and visualisation**:

6a – **Datalog**: Datadog is primarily a monitoring tool for cloud applications. It provides detailed metrics for all your cloud applications, servers, and networks. It can be easily integrated with other automation tools such as Ansible, Chef, Puppet, etc. Datalog is per host paid service.

6b- **Prometheus**: Prometheus is a free software application used for event monitoring and alerting. Prometheus pulls metrics (key/value) and stores the data as time-series, allowing users to query data and alert in a real-time fashion. It communicates to targets at given intervals to collect metrics, aggregate data, show data, or even alert if some thresholds are met.

6c – **Grafana:** **Grafana** is open-source visualization and analytics software. It allows you to query, visualize, alert on, and explore your metrics no matter where they are stored. In plain English, it provides you with tools to turn your time-series database (TSDB) data into beautiful graphs and visualizations. Grafana helps us track the user behaviour, application behaviour, frequency of errors popping up in production or a pre-prod environment, type of errors popping up & the contextual scenarios by providing relative data. **Grafana Cloud** is a highly available, fast, fully managed OpenSaaS metrics platform operated by Grafana Labs. **Grafana Enterprise** is a commercial edition of Grafana that includes additional features not found in the open-source version.

6d – **Loki**: **Loki** is a horizontally-scalable, highly-available, multi-tenant log aggregation system inspired by Prometheus. It is designed to be very cost-effective and easy to operate. Unlike other logging systems, it does not index the contents of the logs, but rather a set of labels for each log stream.

**Conclusion:**

We are spoiled by availability of so many great technologies around us and most of DevOps tools offered free to use basis and furthermore most of them work together seamlessly allowing us to manage increasing demand for automation, continuous integration, continuous delivery and continuous improvement. I am very glad to be part of the community.

Author:

M. Altun

19Feb2021, London

DevOps Engineer @ Finspire Technology