



# devops

Technology and Tools Overview



devops

# Table of contents

## DevOps

|                                     |    |
|-------------------------------------|----|
| Technology and Tools Overview ..... | 01 |
| Table of contents .....             | 02 |
| Introduction .....                  | 03 |

## DevOps - Technology and Tools overview

|  |    |
|--|----|
| What is DevOps .....                                     | 04 |
| DevOps Practices .....                                   | 05 |
| DevOps Benefits .....                                    | 07 |
| Technology and Tools overview .....                      | 08 |
| Cloud Services .....                                     | 09 |
| Source Control Management (SCM) .....                    | 11 |
| Web/App Servers .....                                    | 13 |
| Continous Integration / Continous Delivery (CI/CD) ..... | 15 |
| Automation Tools .....                                   | 16 |
| Monitoring .....   | 17 |
| Networking .....   | 20 |
| Virtualization .....                                     | 21 |
| Microsoft Environment .....                              | 22 |

# Introduction

This document describes DevOps process and technology stack that is in use in Gecko Solutions as well as the benefits from using our DevOps services.

Having in mind that time and people are two most valuable resources, our wish is to reduce both time, and number of engineers needed for configuration and administration tasks, that often knows to be not only tedious, but time-consuming as well. This policy is deeply incorporated in every project and enables both Gecko and its clients and partners to stay competitive on the market.

We are achieving the common goal by engaging top-of-the line DevOps experts, and by using proven technologies, that are accepted as industry standards.

Our secret recipe for successful, yet affordable DevOps services lays in correct application of our business and development processes and practices, that were developed as a result of more than 10 years of experience in DevOps, and Software delivery in general.

# DevOps – Technology and Tools Overview

## What is DevOps

We at Gecko Solutions strongly believe that DevOps is not only the collection of technical skills and procedures, but rather the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market. Our DevOps believes in continuous collaboration, deployments, testing, monitoring and feedback that can be achieved by involving OPS team in the early stage of development and with their active participation until the production releases.

The basic fundamental of DevOps is to implement the automation in all the stages delivery, right from the code verification to deployment, which includes code integration, builds, testing, deploying, verifying the deployed builds.

This automation accelerates all the stages of software delivery so that our developers get feedback and impact of their changes fast which help to speed up an overall time to market. In other words, using automated software build, test and release tools a Gecko team has more control over the entire end-to-end process and eliminates a lot of the friction between the functional silos.

# DevOps Practices

There are a few key practices that help Gecko Solutions innovate faster through automating and streamlining the software development and infrastructure management processes. Most of these practices are accomplished with proper tooling.

## **Continuous Integration**

Continuous integration in Gecko Solutions is a software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run. The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.

## **Continuous Delivery**

Here at Gecko Solutions, Continuous delivery is a software development practice where code changes are automatically built, tested, and prepared for a release to production. It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage. When continuous delivery is implemented properly, developers will always have a deployment-ready build artifact that has passed through a standardized Gecko Test process.

## **Infrastructure as Code**

Infrastructure as code is a practice in which infrastructure is provisioned and managed using code and software development techniques, such as version control and continuous integration. The cloud's API-driven model enables developers and system administrators to interact with infrastructure programmatically, and at scale, instead of needing to manually set up and configure resources. Thus, our engineers can interface with infrastructure using

code-based tools and treat infrastructure in a manner similar to how they treat application code. Because they are defined by code, infrastructure and servers can quickly be deployed using standardized patterns, updated with the latest patches and versions, or duplicated in repeatable ways.

### **Monitoring and Logging**

We, at Gecko Solutions, monitor metrics and logs to see how application and infrastructure performance impacts the experience of their product's end user. By capturing, categorizing, and then analyzing data and logs generated by applications and infrastructure, our engineers understand how changes or updates impact users, shedding insights into the root causes of problems or unexpected changes. Active monitoring is increasingly important as services, as well, and must be available 24/7 as application and infrastructure update frequency increases. Creating alerts or performing real-time analysis of this data also helps us to more proactively monitor our services.

### **Communication and Collaboration**

Increased communication and collaboration in Gecko Solutions is one of the key cultural aspects of DevOps. The use of DevOps tooling and automation of the software delivery process establishes collaboration by physically bringing together the workflows and responsibilities of development and operations. Building on top of that, our teams set strong cultural norms around information sharing and facilitating communication through the use of chat applications, issue or project tracking systems, and wikis. This helps speed up communication across our developers and operations, and even other teams like marketing or sales, allowing all parts of our organization to align more closely on projects and common goals.

## DevOps Benefits

Our DevOps practices improve IT performance. Because of the use of tools to perform work and automate processes our clients get faster turn-around times, better quality at a reduced cost. As a bonus, they can also get metrics on the development process.

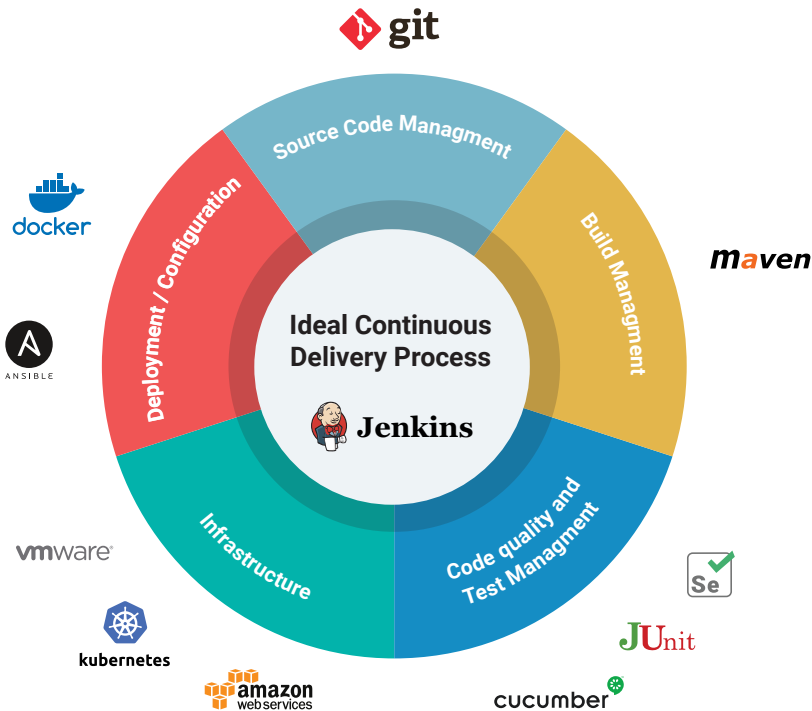
Finally, in addition to speeding up our software development process, to delivery business value quickly while improving quality – Gecko DevOps with the right tools also provides great traceability as well. Traceability is essential for meeting compliance requirements to show how work is initiated, approved, tested and deployed as approved. With traceability it is much easier for our software development process to pass an internal or external audit. That way, it is much easier and efficient to present evidence of your work products along with the check points.

Once the Gecko DevOps team is running properly, the ability to release software will dramatically improve well past monthly releases to weekly, daily, even multiple times per day if the business requires it.

To get the full benefit of DevOps we use automation tools for build, test and deploy. This automation provides a rinse and repeat action to building software that cannot be achieved through legacy process or manual steps.

# Technology and Tools overview

Following image shows some of the technologies and tools that is in use in Gecko DevOps.



tools



# Cloud Services

Below are enlisted some of most noticeable technologies, that are in use, here in Gecko Solutions.

## **Amazon Web Services (AWS)**

Amazon Web Services (AWS) is a subsidiary of Amazon.com that provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis. The technology allows subscribers to have at their disposal a virtual cluster of computers, available all the time, through the Internet. AWS's version of virtual computers emulate most of the attributes of a real computer including hardware (CPU(s) & GPU(s) for processing, local/RAM memory, hard-disk/SSD storage); a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, CRM, etc.

Each AWS system also virtualizes its console I/O (keyboard, display, and mouse), allowing AWS subscribers to connect to their AWS system using a modern browser. The browser acts as a window into the virtual computer, letting subscribers log-in, configure and use their virtual systems just as they would a real physical computer. They can choose to deploy their AWS systems to provide internet-based services for themselves and their customers.

## **Linode**

Linode, LLC is an American privately owned virtual private server provider company. Linode offers multiple products and services for its clients. Its flagship products are cloud-hosting services with multiple packages at different price points. Linode Backup allows customers to backup their servers on a daily, weekly, or monthly basis. Linode Manager and NodeBalancer both allow users to manage multiple server instances across a single system.

## **Rackspace**

Rackspace Inc. is a managed cloud computing company. The Rackspace Cloud is a set of cloud computing products and services billed on a utility computing basis from the US-based company Rackspace. Offerings include web application hosting or platform as a service ("Cloud Sites"), Cloud Storage ("Cloud Files"), virtual private server ("Cloud Servers"), load balancers, databases, backup, and monitoring.

## **Hetzner**

Hetzner Online GmbH is an Internet hosting company and data center operator based in Gunzenhausen, Germany. Hetzner Online provides dedicated hosting, shared web hosting, virtual private servers, managed servers, domain names, SSL certificates, storage boxes, and cloud solutions. At the data center parks located in Nuremberg and Falkenstein, customers can also connect their hardware to Hetzner Online's energy-efficient and state of the art infrastructure and network with the company's colocation services.



# Source Control Management (SCM)

## **Mercurial**

Mercurial is a distributed revision-control tool for software developers. It is supported on Microsoft Windows and Unix-like systems, such as FreeBSD, macOS and Linux. Mercurial's major design goals include high performance and scalability, decentralized, fully distributed collaborative development, robust handling of both plain text and binary files, and advanced branching and merging capabilities, while remaining conceptually simple. It includes an integrated web-interface. Mercurial has also taken steps to ease the transition for users of other version control systems, particularly Subversion. Mercurial is primarily a command-line driven program, but graphical user interface extensions are available, e.g. TortoiseHg, and several IDEs offer support for version control with Mercurial.

## **GitLab**

GitLab is a web-based Git-repository manager with wiki and issue-tracking features, using an open-source license, developed by GitLab Inc. GitLab is the first single application for all stages of the DevOps lifecycle. Only GitLab enables Concurrent DevOps, unlocking organizations from the constraints of the toolchain. GitLab provides unmatched visibility, higher levels of efficiency, and comprehensive governance. This makes the software lifecycle 3 times faster, radically improving the speed of business.

## **BitBucket**

Bitbucket is a web-based version control repository hosting service owned by Atlassian, for source code and development projects that use either Mercurial (since launch) or Git (since October 2011) revision control systems. Bitbucket offers both commercial plans and free accounts. It offers free accounts with an unlimited number of private repositories (which can have up to five users in the case of free accounts) as of September 2010. Bitbucket integrates with other Atlassian software like Jira, HipChat, Confluence and Bamboo.

## Git

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision control system, it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

## Apache Subversion (SVN)

Apache Subversion (often abbreviated SVN, after its command name svn) is a software versioning and revision control system distributed as open source under the Apache License. Software developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly compatible successor to the widely used Concurrent Versions System (CVS).



GitLab

 Bitbucket



# Web/App Services

## **Nginx**

NGINX is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. NGINX is known for its high performance, stability, rich feature set, simple configuration, and low resource consumption.

## **WildFly (JBoss)**

WildFly, formerly known as JBoss AS, or simply JBoss, is an application server authored by JBoss, now developed by Red Hat. WildFly is written in Java and implements the Java Platform, Enterprise Edition (Java EE) specification. It runs on multiple platforms.

## **Jetty**

Eclipse Jetty is a Java HTTP (Web) server and Java Servlet container. While Web Servers are usually associated with serving documents to people, Jetty is now often used for machine to machine communications, usually within larger software frameworks. Jetty is developed as a free and open source project as part of the Eclipse Foundation.

## **Apache Tomcat**

Apache Tomcat, often referred to as Tomcat Server, is an open-source Java Servlet Container developed by the Apache Software Foundation (ASF). Tomcat implements several Java EE specifications including Java Servlet, JavaServer Pages (JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment in which Java code can run.

## Apache

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

## Liberty Core

WebSphere Application Server (WAS) is a software product that performs the role of a web application server. More specifically, it is a software framework and middleware that hosts Java based web applications. It is the flagship product within IBM's WebSphere software suite.



# Continuous Integration / Continuous Delivery (CI/CD)

## **Jenkins**

Jenkins is the number one open-source project for automating your projects. With thousands of plugins to choose from, Jenkins help our teams to automate any task that would otherwise put a time-consuming strain on software team. Common uses include building projects, running tests, bug detection, code analysis, and project deployment.

## **GitLab CI**

GitLab is a rapidly growing code management platform for the modern developer. It provides tools for issue management, code views, continuous integration and deployment, all within a single dashboard. From an idea to production stages, with GitLab you get to put developer in a bird's-eye view of how his project is growing and maturing. GitLab ships pre-built packages for popular Linux distributions, it installs in minutes, has a friendly UI, and offers detailed documentation on every feature.

## **Puppet**

Puppet's platform is built to manage the configs of Unix and Windows systems. As software, it's an Open-Source config management tool. Puppet gives our developers a way to deliver and operate their software regardless of its origin.

# Automation Tools

## **Ansible**

Ansible is software that automates software provisioning, configuration management, and application deployment. As with most configuration management software, Ansible has two types of servers: controlling machines and nodes. First, there is a single controlling machine which is where orchestration begins. Nodes are managed by a controlling machine over SSH. The controlling machine describes the location of nodes through its inventory.

## **Terraform**

Terraform is an infrastructure as code software by HashiCorp. It allows users to define a datacenter infrastructure in a high-level configuration language, from which it can create an execution plan to build the infrastructure such as OpenStack or in a service provider such as IBM Cloud (formerly Bluemix), AWS, Microsoft Azure or Google Cloud Platform Infrastructure is defined in a HCL Terraform syntax or JSON format.

## **Packer**

HashiCorp Packer is easy to use and automates the creation of any type of machine image. It embraces modern configuration management by encouraging you to use automated scripts to install and configure the software within your Packer-made images. Packer brings machine images into the modern age, unlocking untapped potential and opening new opportunities.

## **Docker**

Docker is a computer program that performs operating-system-level virtualization also known as containerization. It is developed by Docker, Inc. Docker is primarily developed for Linux, where it uses the resource isolation features of the Linux kernel such as cgroups and kernel namespaces, and a union-capable file system such as OverlayFS and others to allow independent "containers" to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines (VMs).



# Monitoring

## Icinga

Icinga is an open source computer system and network monitoring application. Icinga has following features:

### Monitoring

- Monitoring of network services (SMTP, POP3, HTTP, NNTP, ping, etc.)
- Monitoring of host resources (CPU load, disk usage, etc.)
- Monitoring of server components (switches, routers, temperature and humidity sensors, etc.)
- Simple plug-in design that allows users to easily develop their own service checks
- Parallelized service checks
- Ability to define network host hierarchy using “parent” hosts, allowing detection of and distinction between hosts that are down and those that are unreachable
- Ability to define event handlers to be run during service or host events for proactive problem resolution

### Notification

- Notification of contact persons when service or host problems occur and get resolved (via email, pager, instant message, or user-defined method)
- Escalation of alerts to other users or communication channels

## Visualization and Reporting

- Two optional user interfaces (Icinga Classic UI and Icinga Web) for visualization of host and service status, network maps, reports, logs, etc.
- Icinga Reporting module based on open source Jasper Reports for both Icinga Classic and Icinga Web user interfaces
- Template based reports (e.g. Top 10 problematic hosts or services, synopsis of complete monitoring environment, availability reports, etc.)
- Report repository with varying access levels and automated report generation and distribution
- Optional extension for SLA reporting that distinguishes between critical events from planned and unplanned downtimes and acknowledgement periods
- Capacity utilization reporting
- Performance graphing via add-ons such as PNP4Nagios, NagiosGrapher and InGraph

## Cacti

Cacti is an open-source, web-based network monitoring and graphing tool designed as a front-end application for the open-source, industry-standard data logging tool RRDtool. Cacti allows a user to poll services at predetermined intervals and graph the resulting data. It is generally used to graph time-series data of metrics such as CPU load and network bandwidth utilization.

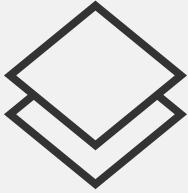
A common usage is to monitor network traffic by polling a network switch or router interface via Simple Network Management Protocol (SNMP).

The primary features of Cacti include: unlimited graph items, auto-padding support for graphs, graph data manipulation, flexible data sources, data gathering on a non-standard timespan, custom data-gathering scripts, built-in SNMP support, graph templates, data source templates, device templates, tree, list, and preview views of graph data, user and user group-based management and security, remote data collection, graph aggregation etc.

### **Netdata**

Netdata is a scalable, distributed, real-time, performance and health monitoring open-source solution for Linux, FreeBSD and MacOS. Out of the box, it collects 1k to 5k metrics per server per second. It is the corresponding of: top, vmstat, iostat, iotop, sar, systemd-cgtop and a dozen more console tools running in parallel. netdata is very efficient in this: the daemon needs just 1% to 3% cpu of a single core, even when it runs on IoT.

Netdata also supports real-time alarms. Netdata alarms can be setup on any metric or combination of metrics and can send notifications.



osi

## Networking

A network connects computers, mobile phones, peripherals, and even IoT devices. Switches, routers, and wireless access points are the essential networking basics. Through them, devices connected to your network can communicate with one another and with other networks, like the Internet. The Open System Interconnection (OSI) model defines a networking framework to implement protocols in seven layers. Layers 1-4 are considered the lower layers, and mostly concern themselves with moving data around. Layers 5-7, the upper layers, contain application-level data. Networks operate on one basic principle: "pass it on." Each layer takes care of a very specific job, and then passes the data onto the next layer.

# Virtualization

Virtualization remains one of the hottest trends in business IT. Whether your organization has already invested heavily in the cloud or is considering a first-time migration, it can be critical to consider the role of a hypervisor in your overall experience. A hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time. The guest OS shares the hardware of the host computer, such that each OS appears to have its own processor, memory and other hardware resources. A hypervisor is also known as a virtual machine manager (VMM).

## **Microsoft Hyper-V**

Hyper-V is Microsoft's hardware virtualization product. Hyper-V is built into Windows Server, or can be installed as a standalone server, known as Hyper-V Server. It offers a unified set of integrated management tools, regardless of whether organizations are striving to migrate to physical servers, a private cloud, a public cloud, or a "hybrid" mixture of these three options.

## **VMware vSphere**

vSphere provides a powerful, flexible, and secure foundation for business agility that accelerates the digital transformation to hybrid cloud and success in the digital economy. It helps you run, manage, connect and secure your applications in a common operating environment across the hybrid cloud. With vSphere, you can support new workloads and use cases while keeping pace with the growing needs and complexity of your infrastructure. vSphere Standard, Enterprise Plus, and Operations Management Enterprise Plus offer varying features and degrees of fault tolerance, allowing organizations to select the best coverage for their needs and growth goals.

# Microsoft Environment

## Active Directory

Active Directory (AD) is a Microsoft technology used to manage computers and other devices on a network. It is a primary feature of Windows Server, an operating system that runs both local and Internet-based servers.

In Active Directory, you can organize objects in classes, which are logical groupings of objects. For example, an object class might be user accounts, groups, computers, domains, or organizational units (OUs).

Some of AD benefits:

**Group Policy** – allows you to centralize the management of computers on your network without having to physically go to and configure each computer individually

**Single Sign-On (SSO)** – once we log on to domain controller it can be used to gain access to other servers without having a separate username and password (Microsoft Exchange, Microsoft SQL, etc.)

**Windows Server Update Services (WSUS)** – centralized and automated update management system which adds SHA256 hash capability for additional security

**Password policies** – An Active Directory account will conform to a central password policy. This allows the business to enforce password complexity and frequent changes across the whole team, something which greatly tightens security.

## **Microsoft Exchange Server**

Microsoft Exchange Server is Microsoft's email, calendaring, contact, scheduling and collaboration platform deployed on the Windows Server operating system for use within a business or larger enterprise. Microsoft designed Exchange Server to give users access to the messaging platform on smartphones, tablets, desktops and web-based systems. Telephony capabilities in Exchange Server support voice messages. Exchange users collaborate through calendar and document sharing. Storage and security features in the platform let organizations archive content, perform searches and execute compliance tasks.

To enable encryption for one or more Exchange services, the Exchange server needs to use a certificate. SMTP communication between internal Exchange servers is encrypted by the default self-signed certificate that's installed on the Exchange server.

To encrypt communication with internal or external clients, servers, or services, you'll likely want to use a certificate that's automatically trusted by all clients, services and servers that connect to your Exchange organization.



Đorđa Stanojevića 12  
11070 Belgrade, Republic of Serbia

+381 11 21 29 724  
+381 11 21 29 726  
office@gecko.rs

[www.gecko.rs](http://www.gecko.rs)



If there are any questions or comments,  
please feel free to contact us.