Cloud Computing Security by DevSecOps

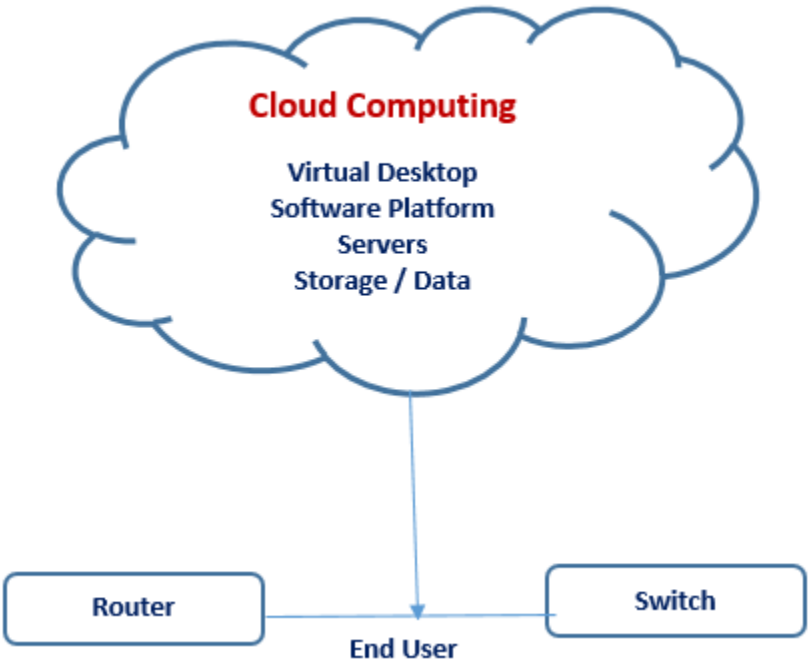
There are many emerging trends in Cloud computing, With this DevSecOps is an approach to software development that integrates security into the development process. Cloud providers are offering tools and services to help businesses implement DevSecOps practices.

This article contains What is cloud computing?, Cloud computing Architecture, What is DevOps and How it works?, How DevSecOps can be used for cloud computing security?

Cloud Computing:

Cloud computing is a framework that provides computing resources (such as servers, storage, databases, and software applications) over the Internet alternative to local hardware and infrastructure. This means that users can access these resources from anywhere, at any time on any device with an internet connection.

Cloud computing features three levels of connectivity such as cloud, network devices like routers and switches, and end-user. The cloud comprises resources like virtual desktops, software platforms, servers, applications, and data storage. They process data through routers and switches. The end-user can access the information from any device.

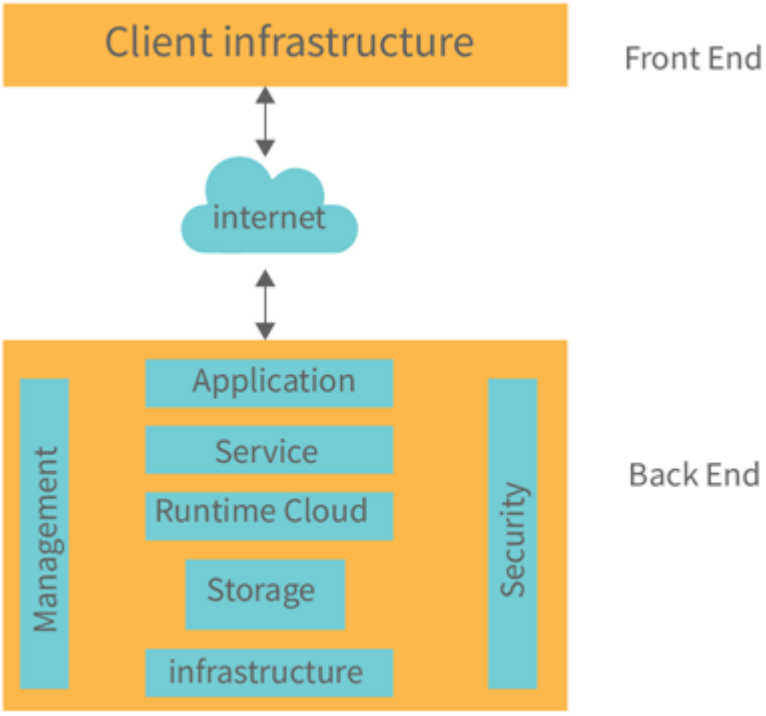


Cloud Architecture:

Cloud architecture refers to how various cloud technology components, such as hardware, virtual resources, software capabilities, and virtual network systems interact and connect to create

cloud computing environments. It acts as a blueprint that defines the best way to strategically combine resources to build a cloud environment for a specific business need.

Components Of Cloud Computing Architecture



The important components of the cloud computing architecture are as follows:

- Front-end platform
- Back-end platform
- Cloud-based delivery

The front-end and back-end of any platform are the face and brains, respectively. Cloud-based delivery allows one to transmit information via application platforms that exist on the cloud. Three popular types of infrastructure that can be used with cloud-based delivery services include Infrastructure-as-a-service (IaaS), Platforms-as-a-service (PaaS), and Software as a service (SaaS).

Benefits of cloud architecture

There are many benefits of cloud architecture for organizations, including:

Cost-effective

Instead of investing upfront costs for servers, you can opt to use the infrastructure of a cloud service provider. Dynamic provisioning allows you to further optimize spending by paying only for the computing resources you use.

Faster time to market

You no longer need to wait to procure, set up, and configure computing infrastructure. Cloud architectures enable you to get up and running fast, so you spend more time focusing on developing and delivering new products.

Scalability

Cloud architectures give you more flexibility to scale computing resources up (or down) based on your infrastructure requirements. You can easily scale to meet higher demand, whether from growth or seasonal spikes in traffic.

Accelerated transformation

Cloud-native architectures like Kubernetes let you make the most of cloud services and automated environments to speed up modernization and drive digital transformation.

More innovation

Cloud architectures allow you to leverage the latest technologies for storage, security, analytics, and AI-like machine learning.

High availability

Applications run and managed on cloud architectures benefit from high-performance computing resources that ensure continuous availability, regardless of fluctuating loads.

Strong security

Cloud service providers consistently upgrade and improve their security mechanisms with expert professionals and the latest technologies to help secure your data, systems, and workloads.

DevOps:

DevOps combines development and operations to increase the efficiency, speed, and security of software development and delivery compared to traditional processes. It can be best explained as people working together to conceive, build and deliver secure software at top speed. DevOps practices enable software development (dev) and operations (ops) teams to accelerate delivery through automation, collaboration, fast feedback, and iterative improvement.

The DevOps methodology comprises four key principles that guide the effectiveness and efficiency of application development and deployment. These principles, listed below, center on the best aspects of modern software development.

1. **Automation of the software development lifecycle.** This includes automated testing, builds, releases, the provisioning of development environments, and other manual tasks that can slow down or introduce human error into the software delivery process.

2. **Collaboration and communication.** A good DevOps team has automation, but a great DevOps team also has effective collaboration and communication.
3. **Continuous improvement and minimization of waste.** From automating repetitive tasks to watching performance metrics for ways to reduce release times or mean-time-to-recovery, high-performing DevOps teams are regularly looking for areas that could be improved.
4. **Hyperfocus on user needs with short feedback loops.** Through automation, improved communication and collaboration, and continuous improvement, DevOps teams can take a moment and focus on what real users want, and how to give it to them.

By adopting these principles, organizations can improve code quality, achieve a faster time to market, and engage in better application planning.

The goal of DevOps:

DevOps represents a change in mindset for IT culture. In building on top of Agile practices, DevOps focuses on incremental development and rapid delivery of software. Success relies on the ability to create a culture of accountability, improved collaboration, empathy, and joint responsibility for business outcomes.

Adopting a DevOps strategy enables businesses to increase operational efficiencies, deliver better products faster, and reduce security and compliance risks.

LifeCycle of DevOps:

The lifecycle stretches from the beginning of software development through to delivery, maintenance, and security. The stages of the DevOps lifecycle are:

- **Plan**

  Organize the work that needs to be done, prioritize it, and track its completion.
- **Create**

  Write, design, develop and securely manage code and project data with your team.
- **Verify**

  Ensure that your code works correctly and adheres to your quality standards — ideally with automated testing.
- **Package**

  Package your applications and dependencies, manage containers, and build artifacts.

- **Secure**

  Check for vulnerabilities through static and dynamic tests, fuzz testing, and dependency scanning.
- **Release**

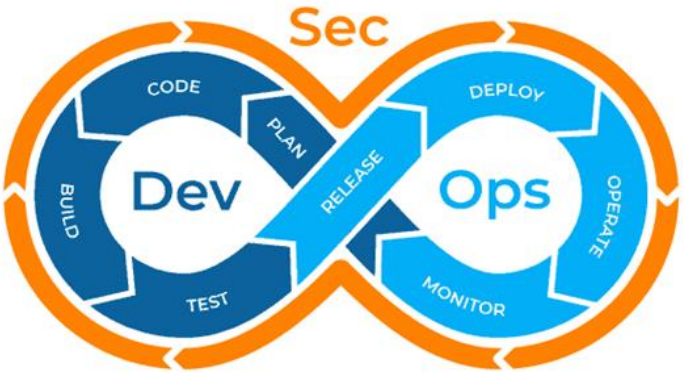
  Deploy the software to end users.
- **Configure**

  Manage and configure the infrastructure required to support your applications.
- **Monitor**

  Track performance metrics and errors to help reduce the severity and frequency of incidents.
- **Govern**

  Manage security vulnerabilities, policies, and compliance across your organization.

  Some organizations string together a series of tools to gain all of this functionality, but that can be incredibly costly and complex to deploy, manage, and maintain.



DevSecOps Approach to Cloud Security:

DevSecOps is about introducing security early in the software development life cycle (SDLC) by expanding the collaboration between the development and operations teams to include security teams. This set of concepts, cultural philosophies, practices, team organization structures, and tools increases an organization's ability to deliver applications and services at high velocity to its clients. It helps in responding to new requirements quickly, as well as to problems that occur in production. This enables organizations to serve their customers better and compete more effectively in the market.

DevSecOps aims to maximize the predictability, efficiency, security, and maintainability of operational processes. It helps in building security into application development from end to end.

Organizations need to understand the relationship between DevSecOps and cloud computing. DevSecOps is about improvement in software development processes and culture, while cloud computing is about technology and services. Organizations need to understand the value that both can bring, when combined, to achieve their transformation objectives.

DevSecOps tools

Organizations in multiple industries can implement DevSecOps to break down silos between development, security, and operations so that they can release more secure software faster.

- **Automotive:** DevSecOps can reduce lengthy cycle times and help meet software compliance standards.
- **Healthcare:** It can enable digital transformation efforts while maintaining the privacy and security of sensitive patient data as per regulations such as HIPAA.
- **Financial, retail, and e-commerce:** Here, DevSecOps can help to fix the Open Web Application Security Project, and also maintain data privacy and security compliance with PCI DSS payment card standards for transactions among consumers, retailers, financial services, etc.

DevSecOps solutions on cloud platforms are expected to help organizations deploy codes easily in the production process, along with enhanced IT security, high performance, and increased scalability.

Cloud and DevSecOps:

Cloud and DevSecOps adoption by an organization helps in providing agility, security, speed, and quality to software processes. Companies can build applications in any programming language, as well as deploy and run them quickly and reliably on any infrastructure. The adoption of these technologies also supports the automation of software release processes, faster application development, and better monitoring of applications and infrastructure performance.

Applications built on next-generation technologies include components such as omnichannel enablement, microservices adoption, API middleware, mobile apps, content management systems, etc. These applications need high availability and fault tolerance. The cloud can help to make the infrastructure or platform resources available within no time.

Cloud automation or Infrastructure as a Code should become a part of the culture of an organization to eliminate manual activities in application installation and configuration.

DevSecOps reference framework

DevSecOps is a set or combination of tools that help in the delivery, development, and management of applications throughout a system's life cycle. At the organization level, the software teams need to automate the entire cycle of build, provisioning, and deployment of test environments, including the tools, scripts, and test data to ensure rapid delivery. These teams need to collaborate around the application architecture and monitor event-based mechanisms for seamless data flow across the toolchains.

Listed below are the different stages any software or application has to pass through as part of the DevSecOps transformation journey:

- Portfolio management and collaboration
- Build
- Source code management
- Testing
- Continuous integration
- Deployment
- Configuration/provision
- Containerisation tools
- Repositories
- Database management
- Monitoring

Different phases of the DevSecOps life cycle and the open-source products:

Portfolio management: The current state of the application and future planning is considered at this stage. DevSecOps readiness assessment across the enterprise is done, along with requirements for DevSecOps implementation, and the approach for the development and transition into operations. The target stage is defined, and the transformation and execution plans are made. The ROI is calculated and the business strategy is made in this phase. In addition, identification of the initial DevSecOps process, the DevSecOps solution, and its linkage to the cloud platform is done.

Build: DevSecOps establishes the interdependence of software development and IT operations and helps an organization produce software and IT services more rapidly, with frequent iterations.

Development of code may be done in any language but is maintained by using version control tools. The most popular tools used are Git, SVN, SonarQube, Maven, and Ant.

Source code management: Versions are maintained in a central repository that acts like a single source of truth. It helps developers to collaborate on the 'latest committed' code, and operations teams can access the same code when they plan to make a release. Whenever there is a fault during the release, Ops can quickly roll back the deployed code and revert to the previous stable state.

Git and GitLab are the leading source control systems. Git allows developers to collaborate on a distributed version control system. GitLab provides a centralized and integrated platform for developers.

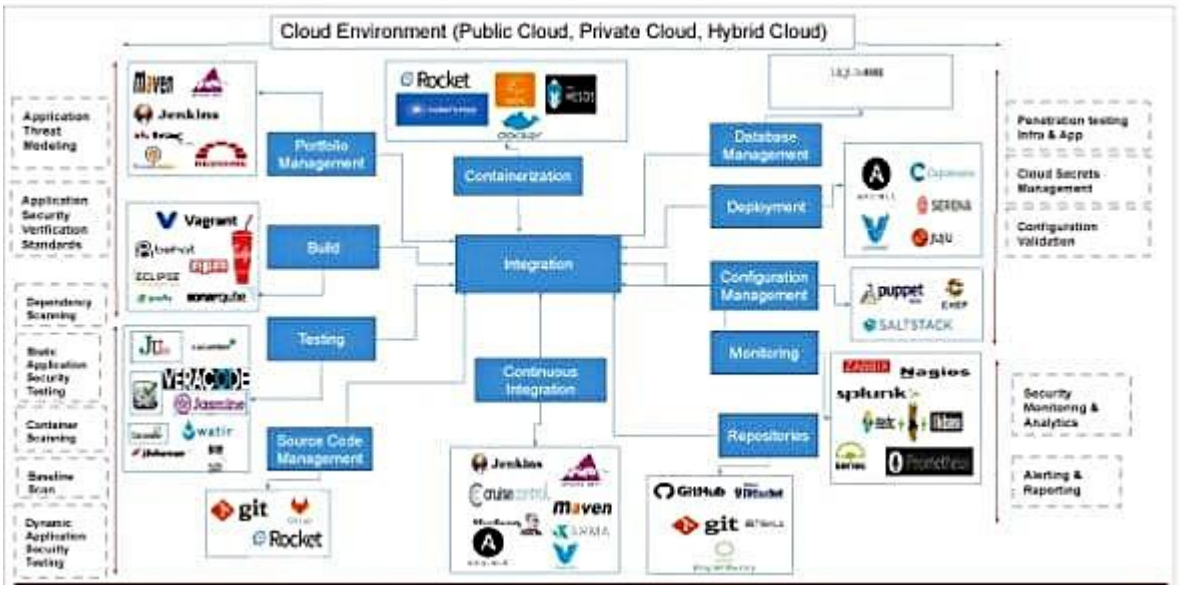


Figure 1: DevSecOps life cycle and mapping of open-source tools

Testing: Continuous testing promotes organization-wide cultural change to promote capabilities like testing early, testing faster, and automating. Continuous testing synchronizes testing and QA with Dev and Ops processes that are optimized to achieve business and development goals.

Tools like Tosca, Selenium, Veracode, SonarQube, Cucumber, and JUnit are used to automate the execution of test cases.

Continuous integration: Continuous integration (CI) helps developers to integrate code into a shared repository several times a day. It allows teams to detect problems early and verifies each check-in. By integrating regularly, it can detect errors quickly and locate them more easily. The most popular CI tool in the market is Jenkins. Other popular CI tools are Bamboo and Hudson.

Continuous deployment: In continuous deployment, every change goes through the pipeline and is automatically put into production, resulting in many production deployments every day with greater delivery speed and frequency for complex applications.

Ansible, Kamatera, and Vagrant are the most useful continuous deployment tools used in the cloud environment.

Configuration/provision management: Configuration management helps in establishing and maintaining consistency in an application's functional requirements and performance. Configuration management tools work based on the master-slave architecture.

Popular configuration management tools used in the cloud environment are Puppet, Chef, Ansible, and SaltStack.

Containerization: Containerisation tools help in maintaining consistency across the environments where the application is developed, tested, and deployed. Containerization eliminates the failure in a production environment by packaging and replicating the same dependencies and packages that are used in the development, testing, and staging environments.

Docker is the most used containerization tool.

Repositories: An artifact repository is a collection of binary software artifacts and metadata stored in a defined directory structure. A repository stores two types of artifacts — releases and snapshots. Release repositories are for stable, static release artifacts, and snapshot repositories are frequently updated repositories that store binary software artifacts from projects that are under constant development. GitHub is the central repository where the code is maintained. Bitbucket and Nexus are the other repository tools.

Database management: This helps in managing revisions of database schema scripts. Liquibase is a widely used open-source database solution that supports various databases.

Continuous monitoring: Continuous monitoring across all phases of application development, testing, and deployment is crucial for a successful DevSecOps implementation. Improving service quality by monitoring application performance and log management solves the problem of aggregating, storing, and analyzing all logs in one place. Splunk, ELK Stack, Nagios, Sensu, and NewRelic are some of the popular tools for monitoring.

Open Source DevSecOps Tools:

Open-source DevSecOps tools for the cloud are designed and developed using open-source technologies to fulfill the DevSecOps toolchain capabilities. These are:

- Portfolio management tools, which provide transparency to stakeholders and participants
- Collaboration tools to help teams work together, anywhere and anytime
- Source control tools, which are the single source of truth
- Issue tracking tools to increase responsiveness and visibility

- Configuration management tools that enforce the desired state
- Continuous integration tools
- Binary repositories that manage builds, releases, and dependencies
- Monitoring tools, which ensure service uptime and optimal performance
- Automated test tools for higher quality
- Deployment tools to improve time to market

Security tools to provide security in the DevOps life cycle, covering interactive application security and testing, runtime application and self-protection, and cloud security

Open source DevSecOps tools that can be used in the cloud:

Ansible: Red Hat owns Ansible. This tool automates various common tasks related to IT operations such as application deployment, configuration management, and cloud provisioning. It integrates with numerous DevOps tools including Jenkins, JIRA, Git, and many others. The free open-source version of Ansible is available on GitHub.

Chef: Chef is an open-source automation platform that transforms infrastructure into code. It operates in the cloud, on-premises, or a hybrid environment. The Chef development kit provides the tools to develop and test infrastructure automation code from a local workstation before deploying changes into production.

Docker: Docker is software used for OS-level virtualization. It is used to create, deploy and run application packages called containers. Containers allow the developer to package an application with all the parts it needs, such as libraries and other dependencies, and ship it as one package. Docker is lightweight, open, and secure.

Docker has two parts. Docker Engine is a tool responsible for creating and running Docker containers. Docker Hub is a service application based on the cloud, which covers the concept of application sharing and workflow automation.

GitHub: This is a collaborative code review tool supporting around 200 software languages. It also supports all the version control features of check-in, commits, branches, merging, labels, task management, wikis, push and pull to/from GitHub, etc. Git fits in very well as a popular and distributed version control system for teams located at different geographical locations.

Hudson: This continuous integration tool is developed in Java and runs on a VMware host or cloud. It is used for managing, monitoring, continuous testing, and integration. It supports various systems for the management of source code, application servers, code analysis tools, testing frameworks, and

build tools. It has real-time notifications of test failures, change set support, and an easy installation and configuration process.

Jenkins: Jenkins is a cloud-based continuous integration tool that helps to automate the activities of build, code analysis, and storing of artifacts. These activities are triggered once a developer or the team commits the code to the version control repository.

Jenkins has many plugins and works as a CI tool for various technologies like C/C++, Java/J2EE, .NET, Angular JS, etc. It also provides plugins to integrate with SonarQube for code review, JFrog Artifactory for storing binary artifacts, and testing tools like Selenium, etc, as a part of the automation process.

Jenkins helps to automate deployments to app servers like Tomcat, JBoss, and Weblogic through plugins, and also to container platforms like Docker.

Kubernetes: Open source Kubernetes is free and downloaded from its repository on GitHub. Administrators must build and deploy the Kubernetes release to a local system or cluster, or a system or cluster in a public cloud such as AWS, Google Cloud Platform (GCP), or Microsoft Azure.

Puppet: This is a cloud DevSecOps tool for operating and delivering software. Puppet automates deployment to provide reliability and agility. It provides continuous automation and delivery faster across the complete software delivery life cycle. Also, the tool increases productivity and operational efficiency, infrastructure as code, configuration management, automated testing, and continuous delivery.

Veracode: This powerful cloud-based service suite for software testing can help to implement end-to-end security. It provides application security services and solutions to reduce risk in Web, mobile, and third-party applications. The various security services provided by Veracode for DevSecOps are:

- Static analysis security testing
- Software composition analysis
- Vendor analysis security testing
- Web application scanning

Selenium: This is an automated functional testing tool to test Web applications. Installed as a Firefox browser plugin, it helps to record and playback test scenarios. In a DevSecOps scenario, once the application is deployed in a test environment, Selenium automated testing is invoked.

Supergiant: This open-source platform for container management can be utilized for Kubernetes deployment on multiple clouds in a matter of minutes. The Supergiant API is used for streamlining production deployment.

Apache Mesos: Apache Mesos abstracts CPU, memory, storage, and other computer resources away from machines, whether they are physical or virtual, and enables fault-tolerant and elastic distributed systems to be easily built and run effectively. The Mesos kernel runs on every machine and provides applications like Hadoop, Spark, Kafka, and Elastic search, with APIs for resource management and scheduling across the entire data center and cloud environments.

Synk: This open-source security management tool is used to automatically find, prioritize and fix vulnerabilities in the open-source code and its dependencies. It helps in developing cloud-native applications.

Key Benefits of adopting DevSecOps processes are:

- It eliminates silos and promotes collaboration and teamwork.
- It identifies the vulnerabilities and reduces the cost and time to deliver software.
- A DevSecOps tools setup reduces the time of deployment by 80-90%. As an example, it reduces deployment time from 12 hours to 2 hours.
- It increases software quality with automated testing. It also reduces the cost and time needed to test and deployment-related downtime.
- It provides improved and stable operations, diminishes security threats, reduces rework, and increases the reliability of service delivery.
- It improves development productivity and overall software quality by 20% with automated and early detection of defects in the cycle.
- Improves business value and provides increased customer value by being responsive to change.

The combination of cloud computing and DevSecOps provides automated and fast application development and monitoring. This increases an organization’s ability to deliver applications and services at high velocity.