Cloud Computing Security by DevSecOps

S.Brindha[1],J.A.Sophiya[2]

1 Assistant Professor, Department of Computer Science

2 Associate Professor, Department of Computer Science and Applications

1&2 St.Peter's Institute of Higher Education & Research, Chennai, Tamilnadu, India.

brindhas.mca@spiher.ac.in. sophiyaja.mca@spiher.ac.in,

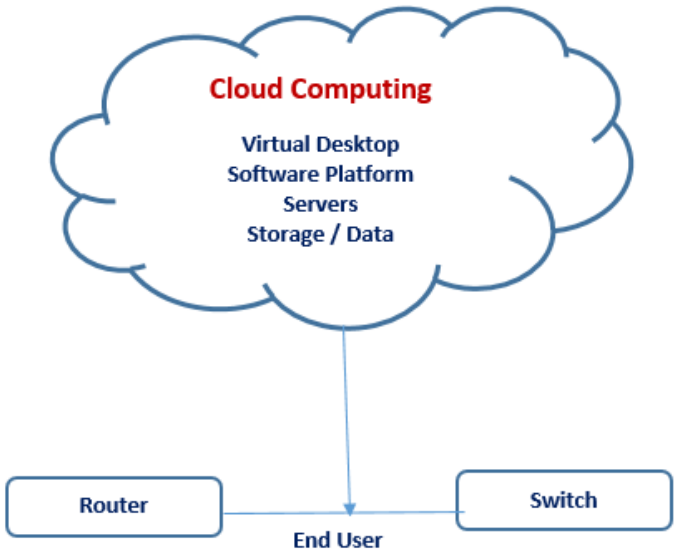
There are many emerging trends in Cloud computing, With this DevSecOps is an approach to software development that integrates security into the development process. Tools and services from cloud providers are available to assist enterprises in implementing DevSecOps procedures.

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:

As an alternative to local hardware and infrastructure, cloud computing offers computer resources (such as servers, storage, databases, and software applications) through the Internet. This indicates that users can use any internet-connected device at any time and from any location to access this content.

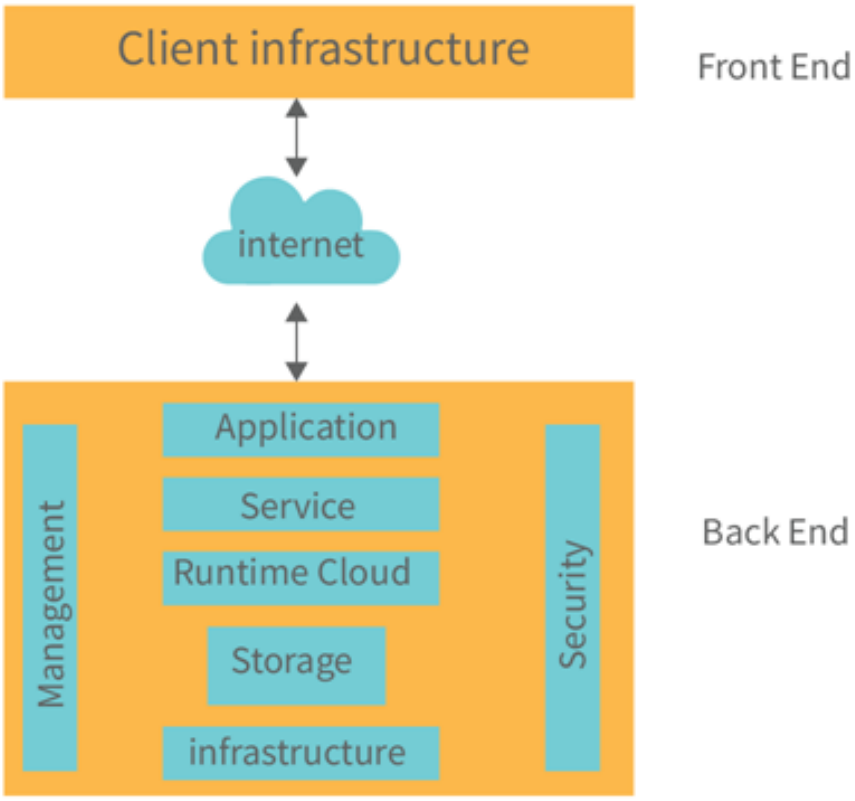
Cloud computing features three levels of connectivity such as cloud, network devices like routers and switches, and end-user. Resources including virtual desktops, software platforms, servers, apps, and data storage are included in the cloud. They process data through routers and switches. Any device can be used by the end user to access the data.



Cloud Architecture:

When discussing cloud computing settings, the term "cloud architecture" refers to how different cloud technology elements, such as hardware, virtual resources, software capabilities, and virtual network systems, interact and connect. It serves as a roadmap for the most effective approach to strategically integrate resources to create a cloud environment for a particular business purpose.

Components Of Cloud Computing Architecture



The following are crucial elements of the cloud computing architecture:

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

Any platform's face and brain are located at its front and back ends, respectively. Information can be transmitted via cloud-based application platforms thanks to cloud-based delivery. The terms Infrastructure-as-a-service (IaaS), Platforms-as-a-service (PaaS), and Software as a service (SaaS) refer to three prevalent types of infrastructure that can be utilised with cloud-based delivery services.

Benefits of cloud architecture

Cloud architecture has numerous advantages for businesses, including:

Cost-effective

You can decide to employ a cloud service provider's infrastructure in place of paying upfront charges for servers. By only paying for the computer resources you actually use, dynamic provisioning enables you to further reduce your cost.

Faster time to market

You don't have to wait to buy, install, and configure computing infrastructure anymore. You can quickly get up and running thanks to cloud architectures, which frees up more time for product development and delivery.

Scalability

Cloud architectures provide you more freedom to adjust the amount of processing power you have according to your infrastructure needs. Whether demand is increased as a result of growth or because of seasonal traffic surges, it is simple to scale.

Accelerated transformation

Utilising cloud services and automated environments to speed up modernization and promote digital transformation is possible with the help of cloud-native architectures like Kubernetes.

More innovation

Utilise the most recent technology for storage, security, analytics, and machine learning that resembles artificial intelligence.

High availability

High-performance computing resources enable continuous availability for applications managed and run on cloud architectures, independent of changing load.

Strong security

With the support of knowledgeable personnel and the newest technologies, cloud service providers continuously upgrade and improve their security measures to help protect your data, systems, and workloads.

DevOps:

In comparison to conventional procedures, DevOps increases the effectiveness, speed, and security of software development and delivery. The best way to describe it is as a team of people coming up with, creating, and delivering secure software quickly. Through automation, teamwork, quick feedback, and iterative improvement, DevOps practices allow software development (dev) and operations (ops) teams to expedite delivery.

The DevOps methodology's four core principles govern the efficacy and efficiency of application development and deployment. These recommendations, which are outlined below, focus on the best aspects of modern software development.

1. **Automation of the software development lifecycle.** This covers manual processes that could slow down the supply of software or involve human mistake, such as automated testing, builds, releases, the provisioning of development environments, and others.
2. **Collaboration and communication.** A competent DevOps team also has efficient communication and cooperation skills in addition to automation.
3. **Continuous improvement and minimization of waste.** High-performing DevOps teams are constantly searching for areas that could be improved, from automating repetitive operations to monitoring performance indicators for ways to decrease release delays or mean-time-to-recovery.
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 putting these ideas into practice, organizations can improve the quality of their code, shorten their time to market, and design their applications more effectively.

The goal of DevOps:

DevOps signifies a shift in the way the IT culture thinks. DevOps emphasises quick software delivery and incremental software development as a way of expanding on Agile practices. Success requires a culture of shared responsibility for corporate outcomes, improved cooperation, empathy, and accountability.

Businesses may improve operational efficiency, provide better products more quickly, and lower security and compliance concerns by using a DevOps strategy.

LifeCycle of DevOps:

The software lifecycle starts with software development and continues with delivery, maintenance, and security. The DevOps lifecycle's phases are:

- **Plan**

  Prioritise, organise, and keep track of the work that has to be done..

- **Create**

  Write, design, develop, and properly manage code and project data with your team..

- **Verify**

  Make sure your code functions well and complies with your quality requirements; preferably, use automated testing.

- **Package**

  Manage containers, build artefacts, and package your apps and dependencies.

- **Secure**

  Utilise static and dynamic testing, fuzz testing, and dependency scanning to look for vulnerabilities.

- **Release**

  Deploy the software to end users.

- **Configure**

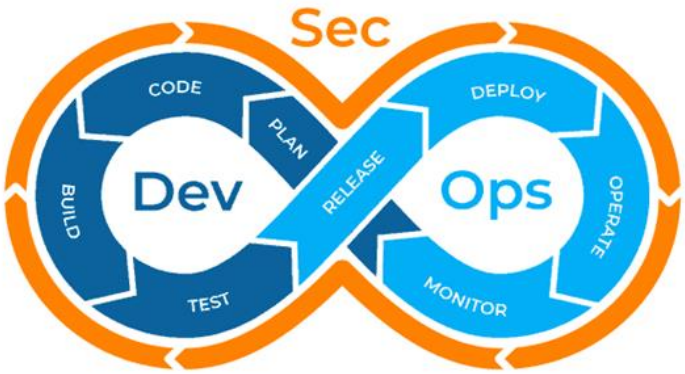
  You must manage and set up the infrastructure required to support your applications.

- **Monitor**

  Track performance metrics and mistakes to lessen incident severity and frequency.

- **Govern**

  Govern Manage compliance, regulations, and security vulnerabilities throughout the entire organization. Some businesses combine a number of technologies to acquire all of this capability, but doing so can be very expensive and difficult to implement, run, and maintain.



DevSecOps Approach to Cloud Security:

By include security teams in the collaboration between the development and operations teams, DevSecOps aims to introduce security early in the software development life cycle (SDLC). 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 assists in promptly responding to both production-related issues and new requirements. This enables organizations to serve their customers better and compete more effectively in the market.

The goal of DevSecOps is to improve operational processes' predictability, effectiveness, security, and maintainability. It aids in integrating security across the entire process of developing an application.

Organizations need to understand the relationship between DevSecOps and cloud computing. Cloud computing is about technology and services, while DevSecOps focuses on improving software development processes and culture. 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 support efforts to digitally change the industry while ensuring the security and privacy of sensitive patient data in accordance with laws like HIPAA.
- **Financial, retail, and e-commerce:** For transactions involving customers, merchants, financial services, etc., DevSecOps can assist with fixing the Open Web Application Security Project as well as preserving data privacy and security compliance with PCI DSS payment card requirements.

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 Any programming language may be used to create apps, and any infrastructure can be used to deploy and run such applications rapidly and consistently.. 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 require fault tolerance and high availability. The infrastructure or platform resources may be quickly made available with the aid of the cloud.

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.

The several phases that every software or application must go through as part of the DevSecOps transformation process are listed below.:

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

The many stages of the DevSecOps life cycle and open-source products are as follows:

Portfolio management: At this step, the application's current state and future planning are taken into account. The enterprise-wide DevSecOps readiness assessment is carried out, together with the specifications for DevSecOps implementation and the method for development and entry into operations. Plans for transformation and execution are created along with a definition of the target stage**.** In this stage, the business plan is developed and the ROI is calculated. The initial DevSecOps methodology, the DevSecOps solution, and its connection to the cloud platform are also identified.

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.
Code development can be done in any language, but version control systems are used to maintain it. The most often used programmes are SonarQube, Maven, Ant, Git, SVN, and SonarQube.

Source code management: Versions are kept up to date in a central location that serves as the only reliable source. The 'latest committed' code makes it easier for developers to work together, and operations teams may access the same code when preparing 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. The most popular source control systems are Git and GitLab. Git allows developers to collaborate on a distributed version control system. GitLab provides a centralized and integrated platform for developers.

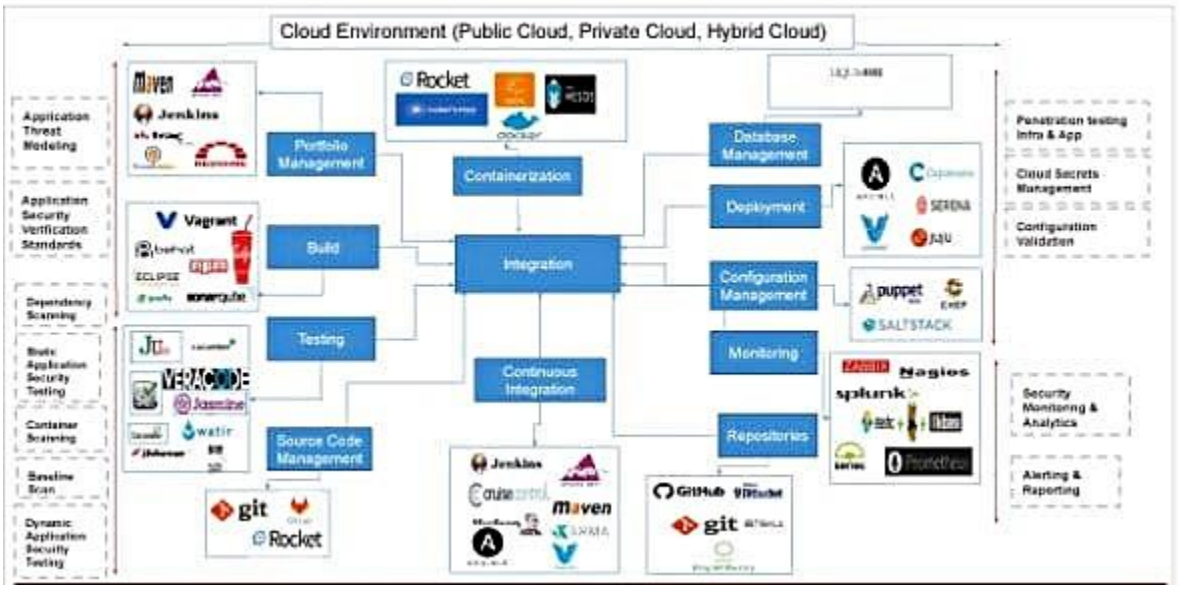


Figure 1: DevSecOps life cycle and mapping of open-source tools

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.
Tosca, Selenium, Veracode, SonarQube, Cucumber, and JUnit are a few examples of tools used to automate test case execution.

Continuous integration: Developers can integrate code into a shared repository numerous times per day with the use of continuous integration (CI). It checks each check-in and enables teams to identify issues early. It can identify faults more rapidly and locate them more easily by periodically integrating. Jenkins is the most widely used CI tool on the market. The CI tools Bamboo and Hudson are also widely used.

Continuous deployment: With continuous deployment, every change automatically enters production after passing through the pipeline, leading to a daily increase in the number of production deployments and the delivery speed and frequency of complicated applications. The best continuous deployment tools for use in a cloud environment are Ansible, Kamatera, and Vagrant.

Configuration/provision management: Management of configuration aids in establishing and preserving consistency in the functional requirements and performance of an application. Tools for configuration management operate on a master-slave design.

Puppet, Chef, Ansible, and SaltStack are common configuration management programmes used in cloud environments.

Containerization: Containerization 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: A collection of binary software artefacts and metadata kept in a specified directory structure is referred to as an artefact repository While snapshot repositories are often updated repositories that store binary software artefacts from projects that are always in progress, release repositories are for stable, static release artefacts. The code is maintained in a central repository called GitHub. There are also two other repository tools: Bitbucket and Nexus.

Database management: This aids in the control of script revisions for databases. A popular open-source database solution that supports different databases is called Liquibase.

Continuous monitoring: A successful DevSecOps implementation requires continuous monitoring during all stages of application development, testing, and deployment. Improving service quality by monitoring application performance and log management solves the problem of aggregating, storing, and analyzing all logs in one place. Some of the well-known monitoring tools include Splunk, ELK Stack, Nagios, Sensu, and NewRelic.

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 that facilitate teamwork anywhere and at any time.
- Source control tools, which are the single source of truth
- Tools for tracking issues to improve visibility and responsiveness
- Tools for configuration management that impose the intended state.
- Continuous integration tools
- Binary repositories that manage builds, releases, and dependencies
- Monitoring tools that guarantee service availability and peak performance,
- Automated test tools for higher quality
- Time-to-market tools for deployment

Security technologies for the DevOps life cycle, including runtime application and self-protection, interactive application security and testing, and cloud security.

Tools for cloud-based DevSecOps that are open source:

Ansible: Red Hat owns Ansible. This programme automates a number of routine IT operations processes, including cloud provisioning, configuration management, and application deployment. Jenkins, JIRA, Git, and many other DevOps technologies are just a few of the ones it integrates with. Ansible's open-source, free version is accessible on GitHub.

Chef: A framework for open-source automation called Chef turns 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. Containers are used to build, distribute, and operate application packages. With the aid of containers, a developer is able to bundle a programme with all of its necessary components, such as libraries and other dependencies, and ship it as a single file. Docker is lightweight, open, and secure.
Docker has two parts. The tool used to create and manage Docker containers is called Docker Engine. Application sharing and workflow automation are covered by the cloud-based service application known as Docker Hub.

GitHub: It is a platform for group code reviews that supports about 200 different programming languages. Additionally, it supports all of the version control features, including push and pull to and from GitHub and check-in, commits, branches, merging, labels, task management, and wikis. Git is a well-liked distributed version control system that works well for teams dispersed throughout the globe.

Hudson: This cloud-based or VMware-based continuous integration solution was created in Java. Managing, monitoring, continuous testing, and integration are all done with it. It supports various systems for the management of source code, application servers, code analysis tools, testing frameworks, and build tools. Change set support, real-time test failure messages, and simple installation and configuration procedures are all included.

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

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.

Through plugins, Jenkins assists in automating deployments to container platforms like Docker as well as app servers like Tomcat, JBoss, and Weblogic.

Kubernetes: Open source platform Kubernetes Kubernetes can be downloaded for free from its GitHub site. 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: Puppet is a DevSecOps cloud tool for managing and distributing software. By automating deployment, Puppet offers dependability and agility. Over the course of the whole software delivery life cycle, it offers continual automation and speedier delivery. The technology also improves infrastructure as code, configuration management, automated testing, and continuous delivery in addition to productivity and operational efficiency.

Veracode: This potent cloud-based software testing service suite can aid in the implementation of end-to-end security. In order to lower risk in Web, mobile, and third-party applications, it offers application security services and solutions. Veracode offers a range of security services for DevSecOps, including:

- 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. It facilitates the recording and replay of test situations when installed as a Firefox browser plugin. Selenium automated testing is launched in a DevSecOps scenario after the application has been installed in a test environment.

Supergiant: This open-source platform for container management can be utilized for Kubernetes deployment on multiple clouds in a matter of minutes. Production deployment is streamlined using the Supergiant API.

Apache Mesos: Apache Mesos makes it simple to create and efficiently operate fault-tolerant and elastic distributed systems by abstracting CPU, memory, storage, and other computing resources away from machines, whether they are physical or virtual.. 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 decreases the price and duration of software delivery while identifying vulnerabilities.
- A DevSecOps tools setup reduces the time of deployment by 80-90%. For instance, it cuts the deployment time in half, from 12 to 2 hours.
- It increases software quality with automated testing. It offers better and stable operations, lessens security concerns, minimizes rework, and improves the dependability of service delivery.
- It also decreases the cost and time needed for testing and deployment-related downtime.
- It improves development productivity and overall software quality by 20% with automated and early detection of defects in the cycle.
- Increases consumer value and enhances business value by being adaptable to change.

Application development and monitoring are automated and quick thanks to cloud computing and DevSecOps. This improves a company's capacity for rapid application and service delivery.