Cloud Computing Security by DevSecOps

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

1 Associate Professor, Department of Computer Science and Applications

2 Assistant 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,

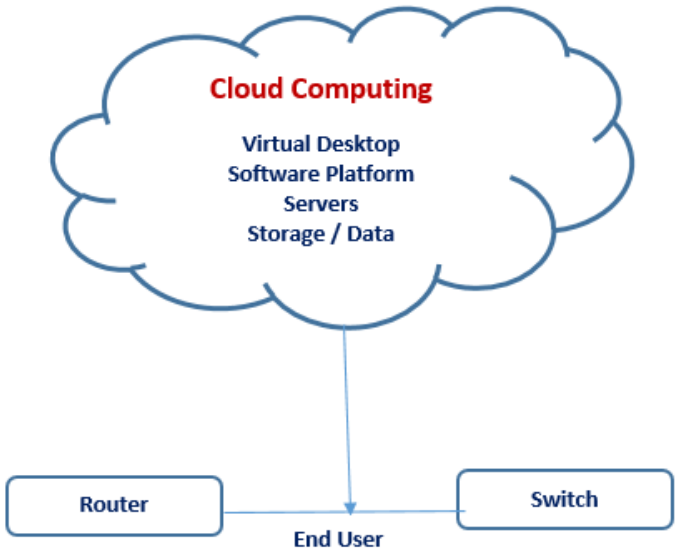
There are many emerging trends in Cloud computing, With this DevSecOps is an approach to software development that integrates security into the development process. Tools and services from cloud providers are available to assist enterprises in implementing DevSecOps procedures.

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?

Introduction:

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.

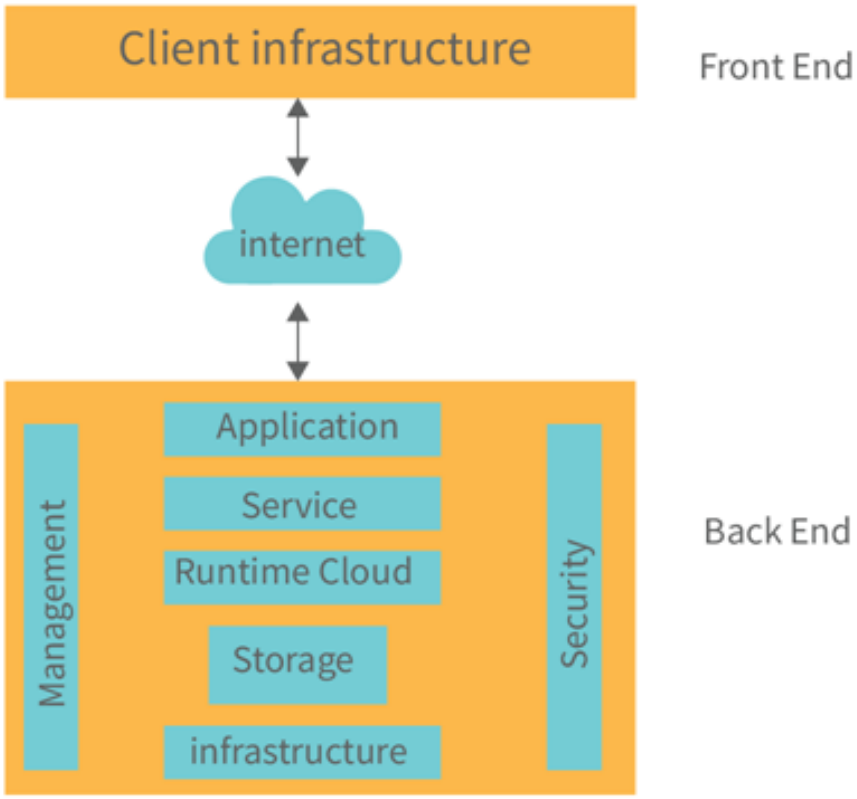
Cloud computing features three levels of connectivity such as cloud, network devices like routers and switches, and end-user. Resources including virtual desktops, software platforms, servers, apps, and data storage are included in the cloud. They process data through routers and switches. Any device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

DevOps's objective:

DevOps represents a change in how the IT culture thinks. As a way of extending Agile practises, DevOps prioritises rapid software delivery and incremental software development. A shared responsibility culture, improved cooperation, empathy, and accountability are necessary for success.

By using a DevOps strategy, businesses may increase operational effectiveness, provide better products faster, and reduce security and regulatory risks.

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**

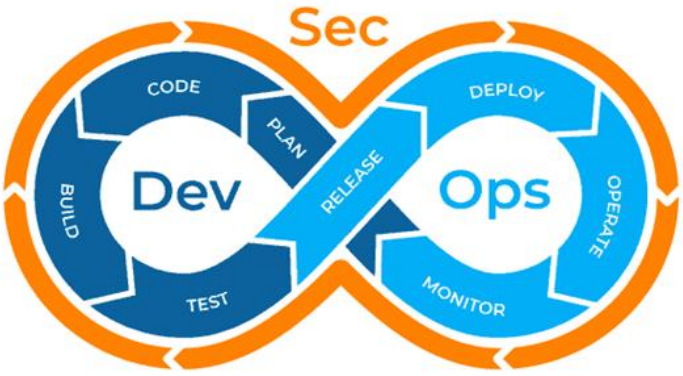
  You must manage and set up the infrastructure required to support your applications.

- **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.



Using DevSecOps to Secure the Cloud:

DevSecOps attempts to include security early in the software development life cycle (SDLC) by involving security teams in the collaboration between the development and operations teams. This collection of ideas, cultural tenets, customs, organisational frameworks for teams, and tools improves an organization's capacity to provide clients with high-speed applications and services. It helps in resolving issues with production as well as new demands quickly. This makes it possible for businesses to provide better customer service and engage in more profitable market competition.

DevSecOps aims to increase the predictability, efficiency, security, and maintainability of operational procedures. It makes it easier to incorporate security into every step of creating an application.

DevSecOps and cloud computing have a relationship that organisations need to grasp. DevSecOps focuses on enhancing the culture and practises of software development, whereas cloud computing is all about technology and services. In order for organisations to achieve their transformational goals, they must comprehend the value that both may provide when used together.

Tools for DevSecOps

DevSecOps is a tool that businesses in a variety of sectors may use to eliminate silos between development, security, and operations and produce more secure software more quickly.

- **Automotive:** DevSecOps can shorten drawn-out cycle times and aid in software compliance requirements.
- **Healthcare**: It can help with initiatives to digitally transform the sector while maintaining the security and privacy of sensitive patient data in accordance with regulations like HIPAA.
- **Financial, retail, and e-commerce:** DevSecOps can help with addressing the Open Web Application Security Project as well as maintaining data privacy and security compliance with PCI DSS payment card regulations for transactions involving customers, merchants, financial services, etc.

DevSecOps solutions on cloud platforms are anticipated to provide organisations with improved IT security, high performance, and expanded scalability while making code deployment in the production process simple.

Cloud and DevSecOps:

Adopting Cloud and DevSecOps can assist an organization's software processes become more agile, secure, quick, and high-quality. Apps can be created using any programming language, and they can be deployed and run quickly and reliably using any infrastructure. Adoption of these technologies also enables faster application development, improved application performance monitoring, and automation of software delivery procedures.

Components like omnichannel support, microservice adoption, API middleware, mobile apps, content management systems, etc. are included in applications created with next-generation technology. High availability and fault tolerance are required for these applications. With the help of the cloud, the infrastructure or platform resources may be immediately made available.

To reduce manual tasks in application installation and configuration, an organisation should adopt a cloud automation or Infrastructure as a Code culture.

DevSecOps Framework:

DevSecOps is a set or combination of tools that support the deployment, development, and management of applications throughout the system lifecycle. At the organizational level, development teams need to automate the entire lifecycle of creating, deploying, and deploying test environments, including tools, scripts, and test data, to ensure rapid deployment. These teams must collaborate on application architecture and monitor event-driven mechanisms to ensure smooth data flow across tool chains.

Below are some of the steps that any software or application must follow as part of the DevSecOps transformation process:

- Portfolio administration and collaboration
- Construction
- Source control
- Testing
- Continuous integration
- Implementation
- Configuration/Delivery
- Containerization tools
- Deposits
- Database management
- Surveillance

Following are the many stages of the DevSecOps life cycle and open-source products:

Portfolio administration: The application's present state and future plans are considered at this stage. The DevSecOps readiness assessment for the entire organisation is conducted, together with the requirements for DevSecOps implementation and the procedure for development and entry into operations. Along with defining the target stage, plans are made for transformation and execution. The business plan is created at this stage, and the ROI is determined. Additionally recognised are the original DevSecOps approach, the DevSecOps solution, and its connection to the cloud platform.

Build: DevSecOps helps a company produce software and IT services more quickly, with frequent revisions, by demonstrating the interconnectedness of software development and IT operations. Any language can be used for code development, but version control mechanisms are needed to keep it up to date. SonarQube, Maven, Ant, Git, SVN, and SonarQube are the software that are used the most frequently. Source code management: The most current versions are preserved in a single, trustworthy location. The 'latest committed' code facilitates collaboration among developers, and operations teams can access the same code when putting together a release. Ops can instantly roll back the released code and return to the prior stable state whenever a problem occurs during the deployment.

Git and GitLab are the most well-liked source control software. Working together on a distributed version control system is possible with Git. For developers, GitLab offers a centralised, integrated platform.

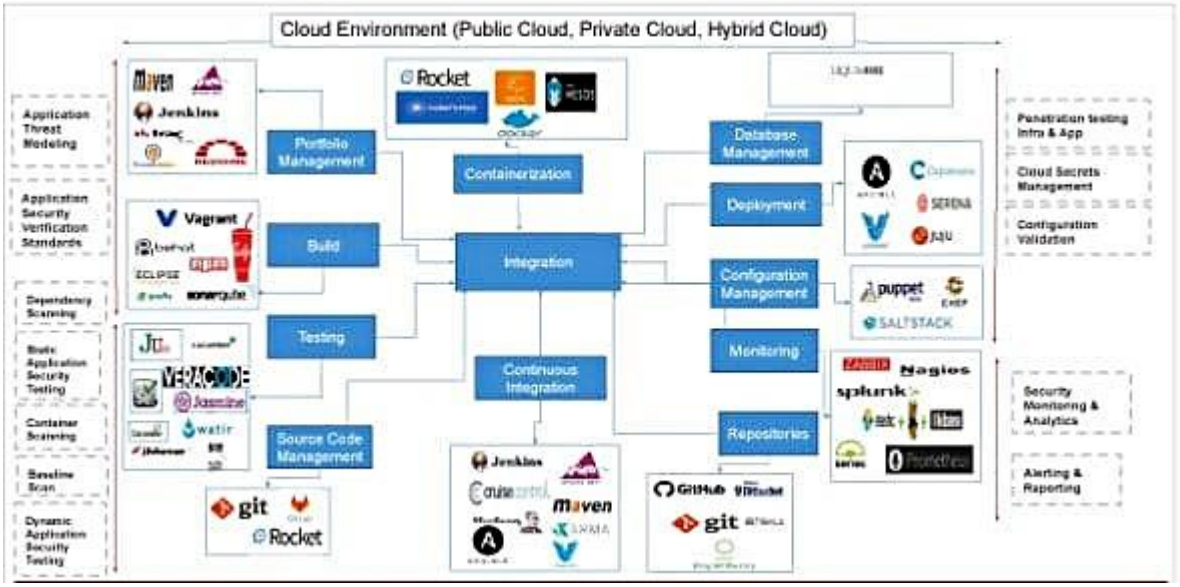


Fig1: DevSecOps lifecycle and open source tool mapping

Testing: Continuous testing encourages organizational-wide culture change to support skills like early, quick, and automated testing. In order to accomplish business and development objectives, Dev and Ops processes must be coordinated with testing and QA. This is done through continuous testing.

Tools used to automate test case execution include Tosca, Selenium, Veracode, SonarQube, Cucumber, and JUnit.

Continuous integration (CI): It enables developers to incorporate code into a common repository repeatedly throughout the day. 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.

Management of configuration and provision: It helps to establish and maintain consistency in the functional specifications and performance of an application. Configuration management tools use a master-slave architecture. In cloud environments, popular configuration management tools include Puppet, Chef, Ansible, and SaltStack.

Containerization: Containerization technologies support consistency across the environments used for the development, testing, and deployment of the application. By packaging and reproducing the same dependencies and packages that are used in the development, testing, and staging environments, containerization prevents failure in a production environment.

The most popular containerization tool is Docker.

Repositories: An artefact repository is a collection of binary software artefacts and metadata stored in a certain directory structure. Release repositories are for stable, static release artefacts, whereas snapshot repositories are often updated repositories that hold binary software artefacts from projects that are continually in development. The code is kept up to date in GitHub, a centralised source. Bitbucket and Nexus are two other repository tools.

Database management: This helps keep the script revisions for databases under control. Liquibase is a well-known open-source database solution that supports several databases.

Continuous monitoring: It is necessary for a DevSecOps implementation to be successful at all development, testing, and deployment stages of an application. Aggregating, storing, and analysing all logs in one location is a challenge that can be resolved by monitoring application performance and log management. Splunk, ELK Stack, Nagios, Sensu, and NewRelic are a few of the popular monitoring tools.

Open Source DevSecOps Tools:

To complete the DevSecOps toolchain capabilities, open-source DevSecOps solutions for the cloud are created and constructed utilising open-source technology. Which are:

- Portfolio management solutions, which give participants and stakeholders access to information.
- Teamwork facilitation tools for use at any time or place.
- Source control software, which is the only reliable source.
- Tools for tracking issues to improve visibility and responsiveness.
- Configuration management tools that enforce the desired condition.
- Tools for continuous integration.
- Binary repositories for managing releases, dependencies, and builds.
- Monitoring tools that ensure maximum performance and service availability.
- Tools for automated testing to improve quality.
- Deployment tools with a quick time to market.

Runtime application and self-protection, interactive application security and testing, and cloud security are examples of security solutions for the DevOps life cycle.

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: Infrastructure is converted into code using the open-source Chef automation framework. It functions in a hybrid, on-premises, or cloud environment. Before releasing changes into production, the Chef development kit offers the tools needed to write and test infrastructure automation code from a local workstation.

Docker: Software for OS-level virtualization is called Docker. Application packages are created, distributed, and run using containers. A developer can package a programme with all of its required parts, such as libraries and other dependencies, and ship it as a single file with the help of containers. Docker is compact, accessible, and safe.

Docker contains two components. 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.

GiftHub: About 200 different programming languages are supported by the website for group code reviews known as GitHub. Additionally, it supports every functionality of version control, such as check-in, commits, branching, merging, labels, task management, and wikis. It also supports push and pull to and from GitHub. Git is a popular distributed version control system that functions well for teams spread out throughout the world.

Hudson: This Java-based continuous integration system runs on VMware or the cloud. It can be used for managing, monitoring, continuous testing, and integration. It supports many frameworks for testing, build tools, code analysis tools, application servers, and source code management systems. There are straightforward installation and configuration processes, change set support, and real-time test failure signals all provided.

Jenkins: It is a cloud-based continuous integration platform that aids in automating the build, code analysis, and artefact archiving processes. Several processes start as soon as a developer or the team commits the code to the version control repository.

Jenkins is a CI tool that supports many different technologies, including C/C++, Java/J2EE,.NET, Angular JS, etc. It has a large number of plugins. As part of the automation process, it also offers plugins that may be integrated with SonarQube for code review, JFrog Artifactory for storing binary artefacts, and testing tools like Selenium, among others.

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

Kubernetes: Open source platform Kubernetes Kubernetes can be downloaded for free from its GitHub site. To a local system or cluster or to a system or cluster in a public cloud like AWS, Google Cloud Platform (GCP), or Microsoft Azure, administrators must build and deploy the Kubernetes release.

Puppet: A DevSecOps cloud tool called Puppet is used to manage and distribute software. The dependability and agility of Puppet are provided by the deployment automation. It provides ongoing automation and quicker delivery throughout the whole life cycle of software delivery. Along with increasing productivity and operational effectiveness, the technology also enhances infrastructure as code, configuration management, automated testing, and continuous delivery.

Veracode: This potent cloud-based software testing service suite can aid in the implementation of end-to-end security. It provides application security services and solutions to reduce risk in Web, mobile, and third-party apps. For DevSecOps, Veracode provides a variety of security services, such as:

- Static evaluation of security
- Analysis of software composition
- Testing for security and vendor analysis
- Scan web applications

Selenium: It is a tool for functional automated testing of web applications. When installed as a Firefox browser plugin, it makes it easier to record and replay test scenarios. In a DevSecOps situation, Selenium automated testing is started after the application has been set up in a test environment.

Supergiant: In a matter of minutes, Kubernetes may be deployed on a number of clouds using this open-source container management technology. The Supergiant API streamlines production deployment.

Apache Mesos: Apache Mesos abstracts CPU, memory, storage, and other computing resources away from machines, whether they are physical or virtual, making it easy to design and operate fault-tolerant and elastic distributed systems. The Mesos kernel runs on every computer and offers APIs for resource management and scheduling across the entire data centre and cloud environments to applications like Hadoop, Spark, Kafka, and Elastic search.

Synk: The open-source code and its dependencies' vulnerabilities are automatically found, given a priority, and fixed using this open-source security management tool. It facilitates the creation of cloud-native applications.

Adopting DevSecOps practises has several key advantages, including:

- It breaks down silos and fosters cooperation and teamwork.
- It identifies vulnerabilities and cuts down on the cost and time of software delivery.
- Setting up DevSecOps tools speeds up deployment by 80–90%. As an illustration, it reduces deployment time from 12 to 2 hours.
- Automated testing improves programme quality. It provides more efficient and reliable operations, lowers security issues, reduces rework, and increases service delivery dependability.
- It also cuts down on the expense and duration of testing and deployment-related downtime.
- The automatic and early detection of problems in the cycle increases development productivity and overall software quality by 20%.
- Enhances business value and increases consumer value by being flexible.

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.

Conclusion:

DevSecOps is a comprehensive and proactive approach to cloud computing security that is in line with DevOps' core values while making security a high priority. It promotes a culture of cooperation and ongoing development, making it a successful tactic for protecting cloud-based assets in a threat environment that is always shifting. DevSecOps integration will be essential for protecting the integrity, confidentiality, and availability of organisations' data and applications as they continue to use cloud computing.