**Name:** Rohit Harawade

**Id:** TU4F2122009

**Batch:** A1

**Roll Number:** A 9

**Experiment - 1**

**Aim**: To understand the concept of DevOps with related technology.

**Tools required**: PC with an internet connection.

**Lab Outcomes**:

To understand DevOps practices which aims to simplify Software Development Life Cycle.

**Theory**:

1. What is DevOps?

(Ans)

* DevOps is 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.
* DevOps is all about the unification and automation of processes, and DevOps engineers are instrumental in combining code, application maintenance, and application management. All of these tasks rely on understanding not only development life cycles, but DevOps culture, and its philosophy, practices, and tools.

1. Why is DevOps needed?

(Ans)

* DevOps—the amalgamation of development (Dev) and operations (Ops) teams—is an organizational approach that enables faster development of applications and easier maintenance of existing deployments. By enabling organizations to create stronger bonds between Dev, Ops and other stakeholders in the company, DevOps promotes shorter, more controllable iterations through the adoption of best practices, automation and new tools. DevOps is not a technology per se, but it covers everything from the organisation to culture, processes and tooling. Initial steps usually include [Continuous integration and continuous delivery (CI/CD)](http://www.appdynamics.com/topics/ci-cd-tools), real-time monitoring, incident response systems and collaboration platforms.
* DevOps adoption is growing rapidly. [IDC forecasts](https://www.idc.com/getdoc.jsp?containerId=US43722718) the worldwide DevOps software market to reach $6.6 billion in 2022, up from $2.9 billion in 2017. The forces driving DevOps adoption include enterprise investments in software-driven innovation, adoption of microservices-based architectures and associated development methodologies, and increased investment by CTOs and CEOs in collaborative and automated application development and operational processes, says IDC analyst Stephen Elliot.

1. How is DevOps different from traditional IT systems?

(Ans)

* DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high quality.
* Traditional IT systems, on the other hand, are siloed, with separate teams for development, testing, and operations. This can lead to delays, errors, and a lack of visibility into the entire process.
* Here are some of the key differences between DevOps and traditional IT systems:

1. **Culture**: DevOps is a cultural shift that emphasizes collaboration and communication between development and operations teams. Traditional IT systems are often more siloed, with teams working independently.
2. **Process**: DevOps uses a continuous delivery model, where code is released frequently and automatically. Traditional IT systems often use a waterfall model, where code is released in large batches.
3. **Tools**: DevOps uses a variety of tools to automate the development and deployment process. Traditional IT systems may use fewer tools, or may rely on manual processes.
4. **Metrics**: DevOps teams track a variety of metrics to measure the success of their deployments. Traditional IT teams may track fewer metrics, or may not track any metrics at all.
5. Why is DevOps used?

(Ans)

DevOps proponents describe several business and technical benefits, many of which can result in happier customers. Some benefits of DevOps include:

* Faster, better product delivery
* Faster issue resolution and reduced complexity
* Greater scalability and availability
* More stable operating environments
* Better resource utilization
* Greater automation
* Greater visibility into system outcomes
* Greater innovation

1. When to adopt DevOps and when not to?

(Ans)

* Some factors to consider when deciding whether or not to adopt DevOps:

1. Your company's culture: DevOps requires a culture of collaboration and communication between development and operations teams. If your company's culture is not conducive to this, DevOps may not be a good fit.
2. Your company's size: DevOps can be a complex and challenging undertaking. If your company is small or has a limited IT budget, DevOps may not be the right choice.
3. Your company's goals: DevOps can help you achieve a number of goals, such as faster time to market, higher quality software, and reduced costs. However, if your company's goals do not align with the benefits of DevOps, it may not be worth the effort.

* Here are some signs that your company may be ready for DevOps:

1. You have a culture of collaboration and communication between development and operations teams.
2. You are committed to continuous improvement and innovation.
3. You are willing to invest in the resources necessary to implement DevOps.

* If you think your company is ready for DevOps, here are some steps you can take to get started:

1. Assess your current state. What are your current processes for development, testing, and deployment? What are your goals for DevOps?
2. Form a DevOps team. This team should include representatives from development, operations, and other relevant teams.
3. Set clear goals and objectives. What do you want to achieve with DevOps?
4. Choose the right tools and technologies. There are a number of tools and technologies that can help you implement DevOps.
5. Start small and scale up. Don't try to implement everything at once. Start with a few key areas and then scale up as you gain experience.

* DevOps is not a silver bullet, but it can be a valuable tool for improving the speed, quality, and reliability of your software delivery. If you are considering adopting DevOps, be sure to do your research and make sure it is the right fit for your company.
* Here are some situations when you might not want to adopt DevOps:

1. Your company is not ready for a cultural change. DevOps requires a culture of collaboration and communication, which may not be present in your company.
2. Your company is not committed to continuous improvement. DevOps is an ongoing process, and you need to be committed to making continuous improvements in order to be successful.
3. Your company does not have the resources to implement DevOps. DevOps can be a complex and expensive undertaking, and you need to make sure you have the resources in place before you start.
4. Explain the DevOps lifecycle.

(Ans)

* The [DevOps lifecycle](https://about.gitlab.com/devops-tools/) stretches from the beginning of software development through to delivery, maintenance, and security. The stages of the DevOps lifecycle are:

1. **Plan**Organize the work that needs to be done, prioritize it, and track its completion.
2. **Create**Write, design, develop and securely manage code and project data with your team.
3. **Verify**Ensure that your code works correctly and adheres to your quality standards — ideally with automated testing.
4. **Package**Packages your applications and dependencies, manage containers, and build artifacts.
5. **Secure**Check for vulnerabilities through static and dynamic tests, fuzz testing, and dependency scanning.
6. **Release**Deploy the software to end users.
7. **Configure**Manage and configure the infrastructure required to support your applications.
8. **Monitor**Track performance metrics and errors to help reduce the severity and frequency of incidents.
9. **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.

1. Explain the DevOps principles.

(Ans)

* The DevOps methodology comprises [four key principles](https://about.gitlab.com/blog/2022/02/11/4-must-know-devops-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 automating 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 really 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.

1. Explain in terms of the roles, skills and responsibilities of a DevOps Engineer.

(Ans)

* A [DevOps engineer](https://about.gitlab.com/blog/2022/04/25/career-spotlight-sre-vs-devops-engineer-vs-devops-platform-engineer/) is responsible for all aspects of the software development lifecycle, including communicating critical information to the business and customers.
* Adhering to DevOps methodologies and principles, they efficiently integrate development processes into workflows, introduce automation where possible, and test and analyze code.
* They build, evaluate, deploy, and update tools and platforms (including IT infrastructure if necessary).
* DevOps engineers manage releases, as well as identify and help resolve technical issues for software users.
* DevOps engineers require knowledge of a range of programming languages and a strong set of communication skills to be able to collaborate among engineering and business groups.

**Lab Outcome**:

To understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements.

**Conclusion**:

The most important thing to remember about DevOps is that it is a journey, not a destination. It takes time and effort to implement DevOps successfully, but the rewards can be significant. If you are willing to put in the work, DevOps can help you improve your software delivery and achieve your business goals.