**Q. Name some of the leading DevOps tools used today.**

Some of today’s most popular tools for DevOps are:

* Puppet
* Selenium
* Docker
* Ansible
* Git
* Chef
* Jenkins

**Q. Describe the various phases of DevOps.**

* Planning: In the beginning, enterprises need a plan that covers which type of application they are trying to develop. Getting at least a rough outline of the process always helps.
* Coding: Once planning is done, applications are coded according to the needs of the end-user.
* Building: The next step is to build the app with code integration.
* Testing: Before you can roll out an app to the world, make sure you test the app. Based on the results and its functions, modify or rebuild the app as needed.
* Integration: Code written by multiple programmers gets integrated into one.
* Deployment: Next, the code gets deployed in an existing cloud environment for future use. Developers must ensure that new changes don’t affect a high-traffic website’s functions.
* Operation: Any necessary operations are carried out on the integrated code.
* Monitoring: Finally, developers must monitor app performance and make any changes necessary to meet the needs of the end-user.

**Q. What are some core benefits of DevOps?**

The benefits that DevOps provides can be clubbed into technical and business benefits.

**Technical benefits:**

* Continuous software delivery
* Efficient error detection and defect correction
* Fewer complexities and issues

**Business benefits:**

* Stable environments for operation
* Faster feature delivery
* Efficient inter-team collaboration and collaboration

**Q. How does continuous monitoring help in maintaining system architecture?**

Continuous DevOps monitoring requires consistent identification, detection and reporting of any threats or faults within a system’s entire infrastructure. This provides several benefits such as:

* It ensures applications, resources and services are running properly on servers.
* It helps monitor server status and determine whether apps are working or not.
* It also allows continuous auditing, inspecting transactions as well as controlled monitoring.

Q. What role does [AWS](https://www.koenig-solutions.com/aws-training) play in DevOps?

AWS provides the following functionalities in DevOps:

* **Flexible services:** It provides flexible, ready-to-use services that don’t require extensive software setup or installation.
* **Built for scalability:** Everything from one instance to multiple thousands can be managed with AWS.
* **Automation:** AWS helps automate business processes and operations, providing more time for innovation and productivity.
* **Security:** AWS IAM (Identity and Access Management) helps set up user policies and permissions.
* **Large partner ecosystem:** It supports an extensive partner ecosystem to extend and integrate with AWS services.

**Q. What does IaC (Infrastructure as Code) mean in relation to configuration management?**

IaC plays the following roles in the configuration management context:

* Writes code to manage deployment, automatic provisioning and configuration
* Manages data centres using machine-readable defining files instead of physical hardware configuration
* Ensures servers and all other infrastructural components are effortlessly and consistently provisioned
* Administers cloud computing environments in the form of an IaaS (Infrastructure as a Service) offering

**Q. How do you implement IaC using AWS?**

Before IaC entered the picture, commands were written onto script files and tested in separate environments before they were deployed. This legacy method is slowly getting replaced by IaC. Like the code written for multiple services using AWS, IaC helps developers with writing, testing and maintaining infrastructure entities descriptively using formats like YAML or JSON. This ensures infrastructure changes are deployed faster and development is simplified.

**Q. What are the benefits of using version control?**

There are several benefits of using Version Control. These are:

* Every team member is free to operate any file at any time using the VCS (Version Control System). Going forward, VCS enables teams to integrate each modified version into a single file.
* VCS requires a brief summary of all the changes made each time a new project version is saved. It also allows developers to examine exactly what is modified in a file’s contents. Consequently, you can see which changes were made to the given project and who made those changes.
* Within the VCS, every previous version and variant is stored meticulously. You can request any version instantly and retrieve snapshots of complete projects in a matter of seconds.
* A distributed VCS like Git allows all the members of a team to retrieve a project’s complete history. This enables various stakeholders and developers to access local repositories should the central server crash or stop working at any time.

**Q. What is the blue/green deployment pattern?**

The Blue/Green deployment pattern is a continuous deployment method used to bring down downtime. It enables the transfer of traffic from instance to instance. For a fresh version of code to be included the new code version must replace the older one.

The new code version is present in a green environment while the old is in a blue environment. When changes are made to previous versions, you need new instances from the old code to execute the new version of the instance.

**Q. What is continuous testing?**

Continuous testing is made up of automated test runs as a part of existing software delivery pipelines. This helps in providing instant feedback on business risks in most recent releases. To prevent any issues with step-switching in the lifecycle of software delivery and allow developers to get feedback immediately, each build is tested this way continually. As a result, the speed of the developer’s productivity increases significantly by eliminating the need to rerun every test after every update and project re-building.

**Q. What is automation testing?**

Manual testing automation or test automation refers to the process that developers use to automate manual procedures so that applications or systems can be tested. Automation testing includes using independent testing tools that help develop test scripts. Developers can run these scripts repeatedly without human intervention.

**Q. List the benefits of automation testing:**

There are several benefits of automation testing to employees as well as organisations.

* It helps save time and money.
* It helps perform unattended and automated execution.
* Developers can easily test large test matrices.
* Human error reduces, resulting in higher accuracy and consistent results.
* Repetitive tasks in testing and execution are handled and supported easily.

**Q. How can you automate testing in the DevOps lifecycle?**

A developer is bound to add every change made to the source code to the DevOps shared repository. Each time changes are made to the code, continuous integration tools like Jenkins take these changes from the repository. This change is then deployed for continuous testing which is carried out using Selenium and other similar tools.

**Q. What are the differences between a centralised and distributed VCS (Version Control System)?**

**Centralised VCS:**

* All existing file versions get stored on one centralised server.
* No developer or local system can have a copy of every file.
* All the data from a project is lost should the central server crash.

**Distributed VCS:**

* In this system, one developer has copies of every version of the code on local systems.
* Distributed VCS allows team members to function offline instead of depending on a centralised location for code backups.
* Developers and enterprises are at no risk of losing data even if the central server crashes.

**Q. Can you elaborate on the core differences between DevOps and Agile?**

There are several fundamental differences between the two technologies.

|  |  |  |
| --- | --- | --- |
| Factor | DevOps | Agile |
| Agility | Agility in development as well as operations | Agility in development only |
| Processes and workflows | Key processes involved are CI, CD, CT, etc. | Key practices include Agile Scrum, Agile Kanban and more. |
| Key focus areas | It prioritises quality and timeliness in all deliverables. | Its priority is always timeliness. |
| Development sprint or release cycle | Small release cycles, immediate feedback | Small release cycles, no immediate feedback |
| Feedback source | Self, using monitoring tools | Customers |
| Scope of work | Agility and automation requirements | Only business agility |

**Q. How do the DevOps tools all work together?**

The functioning of DevOps tools together can be understood through the following logical flow. Everything is automated to ensure seamless delivery, although the flow varies from one company to another depending on the business objective.

1. The developer creates the code, which is managed by Git or any other Version Control System tool.
2. The developer sends the code to the Git repository. Any change to the source code gets updated within the repository.
3. The Jenkins tool takes this code from the VCS repository and builds on it using Ant, Maven or similar tools.
4. Configuration management tools such as Puppet deploy and provision testing environments. Once this is done, Jenkins releases the source code into the test environment for testing with resources like Selenium.
5. Once the code gets tested and approved, Jenkins shares it on the Production server for deployment. The Production server is maintained and provisioned using tools such as Puppet.
6. Tools like Nagios monitor the code continuously after it is deployed.
7. Docker containers provide robust testing environments to test build features.

**Q. What is the most critical thing you achieve using DevOps?**

According to experts, among the many things DevOps helps enterprises achieve, the most important one is getting the changes into the production stage as fast as possible, with minimal risk in software compliance and quality assurance. Other benefits include constructive working relationships and better communication between operations and development teams. These teams collaborate to deliver high-quality software, leading to increased customer satisfaction.

**Q5. What do you understand about software development and technical operations within an organisation? Can you share your expertise based on your experience?**

This answer is highly subjective, which means you cannot explain more than you have done. However, how you frame your answer makes all the difference and can accentuate your stature before your hiring manager.

Typical DevOps engineers work in business-critical environments only around the clock. Mention that you adapted quickly to on-call duties, and you have always been available for live-system real-time responsibilities. Add statements like, ‘I helped to automate business processes successfully, supporting continuous software deployments. I also have experience using tools like Puppet and Chef, public and private cloud systems, a background in Agility and automation and scripting experience with tools such as PHP and Python.’

**Q6. What are the anti-patterns of DevOps?**

A pattern usually follows common usage. If a pattern that businesses typically use isn’t working for you, but you continue following it blindly, it becomes an anti-pattern. There are also several myths existing around DevOps. The most prominent ones are:

* DevOps is a process
* Agile and DevOps are the same things.
* Businesses need separate DevOps groups.
* DevOps can single-handedly solve all business problems.
* DevOps means Developers Managing Production.
* DevOps is development-driven release management.
* My business cannot use DevOps as we are unique.
* My business cannot use DevOps as we don’t have the right people.

**Q7. Explain the various phases of DevOps methodology.**

There are several phases to the DevOps methodology. These are:

* **Plan:** This is the first stage where all the project requirements are discussed, along with other details such as timelines for every stage, combined costing and post-delivery services. This stage helps everyone to stay on the same page regarding the project deliverables and timelines.
* **Code:** In this stage, the developer writes the code in line with the client’s needs. The code is written as small codes known as units.
* **Build:** The completed units are built in this step.
* **Test:** This stage tests the code and units made so far. If any mistakes or errors come up, the units go back for rebuilding.
* **Integrate:** Once the tests succeed, all units get integrated.
* **Deploy:** The client’s environment deploys codeDevOpsNow at this stage.
* **Operate:** Any operations required on the deployed code are performed at this stage.
* **Monitor:** Developers and operations experts monitor the app and its performance within the intended environment.

**Q. What are the KPIs that measure a DevOps team’s success?**

Having clear Key Performance Indicators or KPIs to measure performance is a great way to ensure teams stay on track with their requirements, adhere to established processes and maximise productivity. Some of the most common and effective KPIs used are:

* **Change Failure Rate:** This KPI measures how many deployment failures occur in a single project.
* **MTTR (Meantime to Recovery):** This measures the time you can recover from failed deployments.
* **Lead Time:** This helps measure the time taken for deployment to the production environment.
* **Deployment Frequency:** This KPI measures how frequently new features get deployed.
* **Change Volume:** This helps measure how drastically the source code differs from the existing code.
* **Cycle Time:** This KPI measures the total time taken for application development
* **Customer Ticket:** This measures the total errors that an end-user detects in an application.
* **Availability:** Availability determines an application’s total downtime.
* **Defect Escape Rate:** This step helps measure the total number of issues that must be discovered as soon as possible.
* **Time of detection:** This KPI determines if your app monitoring response time processes function correctly.

**Q9. What has led to the popularity of DevOps?**

DevOps follows two models of software development.

1. The Waterfall Model
2. The Agile Model

This development model is limited due to limited communication with the end customer and one-way operations. Agile helped overcome this limitation with the inclusion of communication between the enterprise and company for the purpose of feedback. However, another limitation is limited communication between development and operations teams which negatively impacts project productivity. At this point, DevOps steps in. DevOps bridges any gaps between development and operations teams with the help of automation. With the integration of automation, production time reduces and productivity increases. DevOps and automation also help integrate the testing and development stages. This means developers can find bugs in the initial stages, increasing speed and efficiency.