Q1. What are the fundamental differences between DevOps & Agile?

The differences between the two are listed down in the table below.

Features	**DevOps**	**Agile**
Agility	Agility in both Development & Operations	Agility in only Development
Processes/ Practices	Involves processes such as CI, CD, CT, etc.	Involves practices such as Agile Scrum, Agile Kanban, etc.
Key Focus Area	Timeliness & quality have equal priority	Timeliness is the main priority
Release Cycles/ Development Sprints	Smaller release cycles with immediate feedback	Smaller release cycles
Source of Feedback	Feedback is from self (Monitoring tools)	Feedback is from customers
Scope of Work	Agility & need for Automation	Agility only

Q2. What is the need for DevOps?

According to me, this answer should start by explaining the general market trend. Instead of releasing big sets of features, companies are trying to see if small features can be transported to their customers through a series of release trains. This has many advantages like quick feedback from customers, better quality of software etc. which in turn leads to high customer satisfaction. To achieve this, companies are required to:

1. Increase deployment frequency
2. Lower failure rate of new releases
3. Shortened lead time between fixes
4. Faster mean time to recovery in the event of new release crashing

DevOps fulfills all these requirements and helps in achieving seamless software delivery. You can give examples of companies like Etsy, Google and Amazon which have adopted DevOps to achieve levels of performance that were unthinkable even five years ago. They are doing tens, hundreds or even thousands of code deployments per day while delivering world-class stability, reliability and security.

If I have to test your knowledge on DevOps, you should know the difference between Agile and DevOps. The next question is directed towards that.

Q3. How is DevOps different from Agile / SDLC?

I would advise you to go with the below explanation:

Agile is a set of values and principles about how to produce i.e. develop software. Example: if you have some ideas and you want to turn those ideas into working software, you can use the Agile values and principles as a way to do that. But, that software might only be working on a developer’s laptop or in a test environment. You want a way to quickly, easily and repeatably move that software into production infrastructure, in a safe and simple way. To do that you need DevOps tools and techniques.

You can summarize by saying Agile software development methodology focuses on the development of software but DevOps on the other hand is responsible for development as well as deployment of the software in the safest and most reliable way possible. Here’s a blog that will give you more information on the evolution of DevOps.

Now remember, you have included DevOps tools in your previous answer so be prepared to answer some questions related to that.

Q4. Which are the top DevOps tools? Which tools have you worked on?

The most popular DevOps tools are mentioned below:

- Git : Version Control System tool
- Jenkins : Continuous Integration tool
- Selenium : Continuous Testing tool

- Puppet, Chef, Ansible : Configuration Management and Deployment tools
- Nagios : Continuous Monitoring tool
- Docker : Containerization tool

You can also mention any other tool if you want, but make sure you include the above tools in your answer.

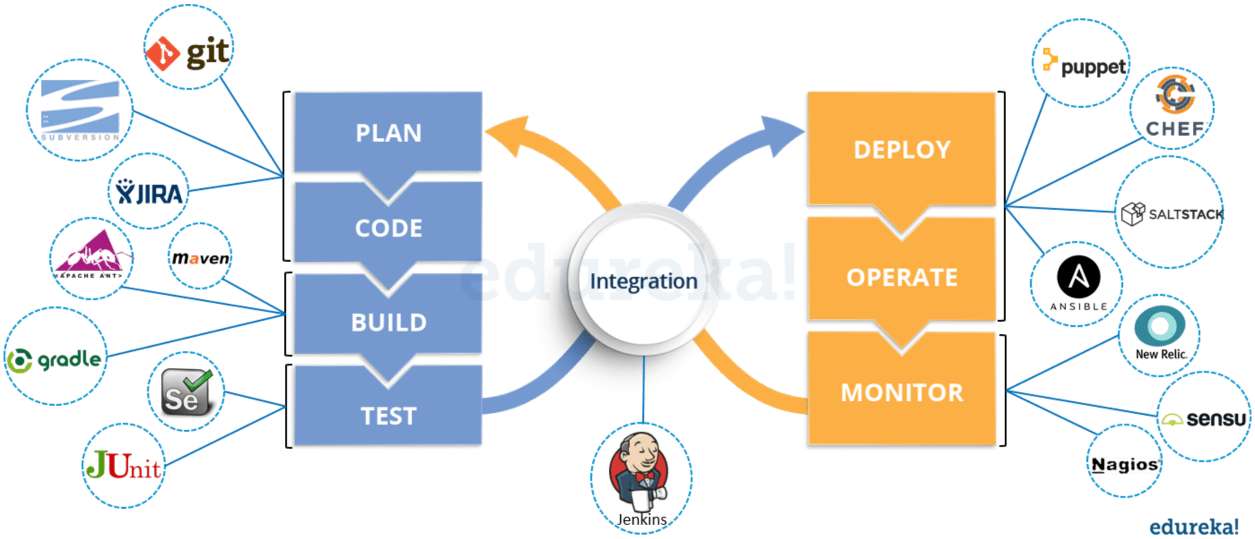
The second part of the answer has two possibilities:

1. If you have experience with all the above tools then you can say that I have worked on all these tools for developing good quality software and deploying those softwares easily, frequently, and reliably.
2. If you have experience only with some of the above tools then mention those tools and say that I have specialization in these tools and have an overview about the rest of the tools.

Q5. How do all these tools work together?

Given below is a generic logical flow where everything gets automated for seamless delivery. However, this flow may vary from organization to organization as per the requirement.

1. Developers develop the code and this source code is managed by Version Control System tools like Git etc.
2. Developers send this code to the Git repository and any changes made in the code is committed to this Repository.
3. Jenkins pulls this code from the repository using the Git plugin and build it using tools like Ant or Maven.
4. Configuration management tools like puppet deploys & provisions testing environment and then Jenkins releases this code on the test environment on which testing is done using tools like selenium.
5. Once the code is tested, Jenkins send it for deployment on the production server (even production server is provisioned & maintained by tools like puppet).
6. After deployment It is continuously monitored by tools like Nagios.
7. Docker containers provides testing environment to test the build features.



Q6. What are the advantages of DevOps?

For this answer, you can use your past experience and explain how DevOps helped you in your previous job. If you don't have any such experience, then you can mention the below advantages.

Technical benefits:

- Continuous software delivery
- Less complex problems to fix
- Faster resolution of problems

Business benefits:

- Faster delivery of features
- More stable operating environments
- More time available to add value (rather than fix/maintain)

Q7. Mention some of the core benefits of DevOps?

- Faster development of software and quick deliveries.
- DevOps methodology is flexible and adaptable to changes easily.
- Compared to the previous software development models, confusion about the project is decreased due to increased product quality.
- The gap between the development team and operation team is bridged. i.e, the communication between the teams has been increased.

- Efficiency is increased by the addition of automation of continuous integration and continuous deployment.
- Customer satisfaction is enhanced.

Q8. What is the most important thing DevOps helps us achieve?

According to me, the most important thing that DevOps helps us achieve is to get the changes into production as quickly as possible while minimizing risks in software quality assurance and compliance. This is the primary objective of DevOps. Learn more in this DevOps tutorial blog.

However, you can add many other positive effects of DevOps. For example, clearer communication and better working relationships between teams i.e. both the Ops team and Dev team collaborate together to deliver good quality software which in turn leads to higher customer satisfaction.

Q9. Explain with a use case where DevOps can be used in industry/ real-life.

There are many industries that are using DevOps so you can mention any of those use cases, you can also refer the below example:

Etsy is a peer-to-peer e-commerce website focused on handmade or vintage items and supplies, as well as unique factory-manufactured items. Etsy struggled with slow, painful site updates that frequently caused the site to go down. It affected sales for millions of Etsy's users who sold goods through online market place and risked driving them to the competitor.

With the help of a new technical management team, Etsy transitioned from its waterfall model, which produced four-hour full-site deployments twice weekly, to a more agile approach. Today, it has a fully automated deployment pipeline, and its continuous delivery practices have reportedly resulted in more than 50 deployments a day with fewer disruptions.

Q10. Explain your understanding and expertise on both the software development side and the technical operations side of an organization you have worked with in the past.

For this answer, share your past experience and try to explain how flexible you were in your previous job. You can refer the below example:

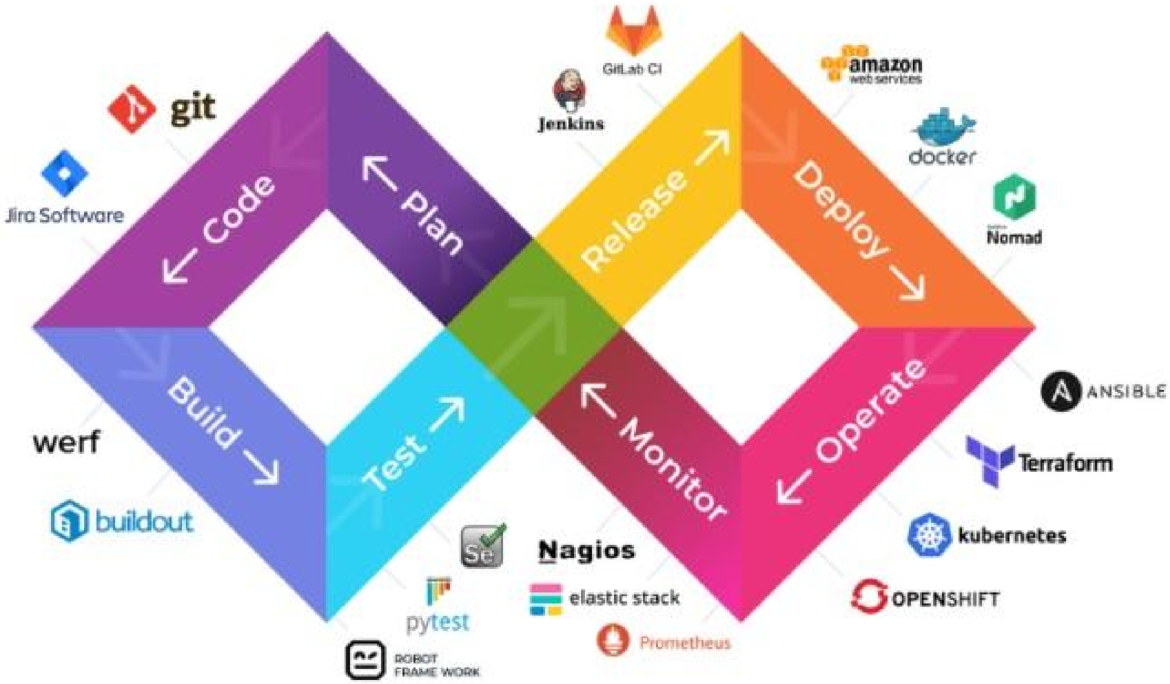
DevOps engineers almost always work in a 24/7 business-critical online environment. I was adaptable to on-call duties and was available to take up real-time, live-system responsibility. I successfully automated processes to support continuous software deployments. I have experience with public/private clouds, tools like Chef or Puppet, scripting and automation with tools like Python and PHP, and a background in Agile.

Q11. What are the anti-patterns of DevOps?

A pattern is common usage usually followed. If a pattern commonly adopted by others does not work for your organization and you continue to blindly follow it, you are essentially adopting an anti-pattern. There are myths about DevOps. Some of them include:

- DevOps is a process
- Agile equals DevOps?
- We need a separate DevOps group
- Devops will solve all our problems
- DevOps means Developers Managing Production
- DevOps is Development-driven release management
    1. DevOps is not development driven.
    2. DevOps is not IT Operations driven.
- We can’t do DevOps – We’re Unique
- We can’t do DevOps – We’ve got the wrong people

Q12. Explain the different phases in DevOps methodology?



The various phases of the DevOps lifecycle are as follows:

- Plan – In this stage, all the requirements of the project and everything regarding the project like time for each stage, cost, etc are discussed. This will help everyone in the team to get a brief idea about the project.
- Code – The code is written over here according to the client's requirements. Here codes are written in the form of small codes called units.
- Build – Building of the units is done in this step.
- Test – Testing is done in this stage and if there are mistakes found it is returned for re-build.
- Integrate – All the units of the codes are integrated into this step.
- Deploy – codeDevOpsNow is deployed in this step on the client's environment.
- Operate – Operations are performed on the code if required.
- Monitor – Monitoring of the application is done over here in the client's environment.

Q13. Explain your understanding and expertise on both the software development side and the technical operations side of an organization you have worked with in the past?

- Deployment frequency: This measures how frequently a new feature is deployed.
- Change failure rate: This is used to measure the number of failures in deployment.
- Mean Time to Recovery (MTTR): The time is taken to recover from a failed deployment.

Q14. What are the KPIs that are used for gauging the success of a DevOps team?

KPI Means Key Performance Indicators are used to measure the performance of a DevOps team, identify mistakes and rectify them. This helps the DevOps team to increase productivity and which directly impacts revenue.

There are many KPIs which one can track in a DevOps team. Following are some of them:

- Change Failure rates: This is used to measure the number of failures in deployments.
- Meantime to recovery (MTTR): The time is taken to recover from a failed deployment.
- Lead time: This helps to measure the time taken to deploy on the production environment.
- Deployment frequency: This measures how frequently a new feature is deployed.
- Change volume: This is used to measure how much code is changed from the existing code.
- Cycle time: This is used to measure total application development time.
- Customer Ticket: This helps us to measure the number of errors detected by the end-user.
- Availability: This is used to determine the downtime of the application.
- Defect escape rate: This helps us to measure the number of issues that are needed to be detected as early as possible.
- Time of detection: This helps you understand whether your response time and application monitoring processes are functioning correctly.

Q15. Why has DevOps become famous?

As we know before DevOps there are two other software development models:

- Waterfall model
- Agile model

In the waterfall model, we have limitations of one-way working and lack of communication with customers. This was overcome in Agile by including the communication between the customer and the company by taking feedback. But in this model, another issue is faced regarding communication between the Development team and operations team due to which there is a delay in the speed of production. This is where DevOps is introduced. It bridges the gap between the development team and the operation team by including the automation feature. Due to this, the speed of production is increased. By including automation, testing is integrated into the development stage. Which resulted in finding the bugs at the very initial stage which increased the speed and efficiency.