**From where I have to start learning aws devops:**

To start learning AWS DevOps, you should have a good understanding of basic programming concepts and experience in using Linux operating systems. Here are the steps you can follow to begin your AWS DevOps journey:

1. Learn basic programming: You should start by learning a programming language such as Python, Java or Ruby. You can learn from online tutorials, books or courses.
2. Get familiar with Linux: Linux is widely used in DevOps, so you should become familiar with basic Linux commands, shell scripting, and Linux administration.
3. Understand DevOps principles: You should understand the principles of DevOps, including continuous integration, continuous delivery, and continuous deployment.
4. Learn about AWS services: You should learn about various AWS services such as EC2, S3, IAM, and CloudFormation. AWS provides a lot of documentation, tutorials, and online courses to learn their services.
5. Learn about DevOps tools: There are many DevOps tools available, such as Jenkins, Git, Docker, and Kubernetes. You should learn how to use these tools to automate your development and deployment processes.
6. Practice: Practice is the key to mastering any skill. You should practice what you have learned by building your own projects, working on open-source projects, or participating in hackathons.
7. Certifications: You can also pursue certifications such as AWS Certified DevOps Engineer or AWS Certified Solutions Architect to validate your knowledge and skills.

Remember, AWS DevOps is a vast and continuously evolving field, and you need to keep yourself updated with the latest technologies and best practices

from where i have to start learning aws devops

To start learning AWS DevOps, here are some recommended steps you can follow:

1. Learn the basics of AWS: It's important to understand the fundamentals of Amazon Web Services (AWS) before diving into DevOps. You can start with AWS's free digital training courses, which cover topics such as computing, storage, networking, security, and database services.
2. Get familiar with DevOps concepts: DevOps is an approach to software development that emphasizes collaboration, communication, and automation. You can start by learning about the key principles of DevOps, including continuous integration (CI), continuous delivery (CD), and infrastructure as code (IaC).
3. Choose a DevOps tool: AWS offers a range of DevOps tools and services, including AWS CodePipeline, AWS CodeBuild, and AWS CodeDeploy. Choose a tool that aligns with your requirements and start learning how to use it.
4. Learn scripting languages: DevOps involves a lot of scripting, and you will need to learn scripting languages such as Python, Bash, and PowerShell.
5. Practice and experiment: DevOps is a hands-on field, and the best way to learn is to practice and experiment with different tools and technologies. AWS offers a free tier that allows you to experiment with its services without incurring any costs.
6. Join AWS communities: Join online communities and discussion forums where you can ask questions, share your experiences, and learn from others. AWS offers its own online community called AWS DevOps, and there are also many other online communities such as Reddit and Stack Overflow.

Remember, DevOps is a constantly evolving field, so it's important to keep learning and adapting as new technologies and practices emerge. Good luck on your learning journey!

From where i have to start learning aws devops

3 / 3

Learning AWS DevOps can be a challenging task, but with the right resources and approach, you can quickly get started. Here are the steps you can follow to start learning AWS DevOps:

1. Learn the basics of AWS: If you are new to AWS, start by learning the basics of AWS services, such as EC2, S3, and VPC, as they form the building blocks of AWS DevOps.
2. Understand DevOps principles: DevOps is a cultural and professional movement that emphasizes collaboration, communication, and automation. Understanding the principles of DevOps is essential to learning AWS DevOps.
3. Learn about AWS DevOps tools: AWS offers a range of DevOps tools, including CodePipeline, CodeCommit, CodeBuild, CodeDeploy, and CloudFormation, among others. Learn about these tools and their features.
4. Practice with AWS DevOps tools: To gain hands-on experience, practice using AWS DevOps tools by setting up pipelines, deploying applications, and automating tasks.
5. Join AWS DevOps communities: Join online communities such as AWS DevOps forums, blogs, and meetups to stay up-to-date with the latest trends and best practices.
6. Get certified: AWS offers a range of DevOps certifications, such as AWS Certified DevOps Engineer, which validate your skills and expertise in AWS DevOps.

Remember, learning AWS DevOps is an ongoing process that requires dedication and continuous learning. Start with the basics, gain hands-on experience, and stay updated with the latest trends and best practices.

From where i have to start learning aws devops

Learning AWS DevOps can be a complex journey, but there are a few key areas that you can focus on to get started:

1. Understanding the fundamentals of DevOps: Before diving into AWS DevOps, it's essential to understand the principles and practices of DevOps, including continuous integration, continuous delivery, and automation.
2. Familiarizing yourself with AWS: Learn the basics of AWS services such as EC2, S3, RDS, and Lambda, and how to use them to build scalable and reliable systems.
3. Learning Infrastructure as Code: AWS DevOps heavily relies on Infrastructure as Code (IaC) tools such as CloudFormation and Terraform. It is essential to learn the concepts and principles behind IaC, and how to use these tools to create, manage and provision infrastructure.
4. Familiarizing yourself with automation tools: Automation tools such as AWS CodePipeline, CodeBuild, and CodeDeploy are critical for automating the development and deployment process. You should learn how these tools work, how to configure them, and how to integrate them into your workflow.
5. Practice: As with any new skill, the best way to learn AWS DevOps is to practice. Create your own projects and experiment with different AWS services and DevOps tools. Join online communities and forums to get help from other professionals and share your experiences.

There are many resources available to learn AWS DevOps, including online courses, tutorials, and certifications. Some popular online platforms include AWS Training and Certification, Udemy, Coursera, and edX

from where i have to start learning aws devops

5 / 5

To start learning AWS DevOps, here are some steps you can follow:

1. Learn the basics of AWS: Before diving into DevOps, it's essential to understand the basic concepts of Amazon Web Services (AWS). Start with the AWS documentation and familiarize yourself with the various AWS services, including EC2, S3, IAM, and more.
2. Learn the principles of DevOps: DevOps is a methodology that combines software development and operations to increase the speed and efficiency of software delivery. Understand the principles of DevOps, such as continuous integration, continuous delivery, and automation.
3. Familiarize yourself with DevOps tools: AWS provides various tools for DevOps, such as AWS CodePipeline, AWS CodeDeploy, AWS CodeBuild, AWS CloudFormation, and more. Start learning these tools to understand how they work and how they fit into the DevOps process.
4. Learn a programming language: As a DevOps engineer, you need to have knowledge of at least one programming language. Python is a popular choice for DevOps, but other languages such as Ruby, JavaScript, and Go are also useful.
5. Practice with hands-on projects: Once you have a good understanding of the basics, start practicing with hands-on projects. AWS provides many resources for learning and practicing, such as AWS Free Tier, AWS Training and Certification, AWS documentation, and more.
6. Join the community: Join online communities such as AWS forums, DevOps forums, and Slack channels to connect with other DevOps professionals, ask questions, and share knowledge.

Remember that learning AWS DevOps is an ongoing process. Keep yourself updated with the latest trends and best practices in the industry, and continuously improve your skills.

Linux architecture

Linux is an open-source operating system that is built on a modular architecture that is composed of several layers. Here are the key components of the Linux architecture:

1. Kernel: The kernel is the core of the Linux operating system, and it is responsible for managing the system's resources, such as memory, input/output (I/O) devices, and CPU. It provides the necessary interfaces for user-level applications to access these resources.
2. Shell: The shell is the user interface to the Linux operating system. It is a command-line interface that allows users to interact with the system by typing commands. The shell also provides scripting capabilities to automate repetitive tasks.
3. Filesystem: The filesystem is a hierarchical organization of files and directories that is used to store data on the disk. In Linux, the filesystem is organized as a tree, with the root directory at the top and all other directories and files branching out from it.
4. Libraries: Libraries are collections of pre-written code that can be used by applications to perform common tasks. Linux provides a wide range of libraries for various purposes, such as graphics, networking, and encryption.
5. Applications: Applications are user-level programs that run on top of the Linux operating system. Linux supports a vast range of applications, from text editors and web browsers to database management systems and scientific computing tools.
6. Device drivers: Device drivers are software components that allow the operating system to communicate with hardware devices, such as printers, network cards, and graphics cards. Linux supports a wide range of hardware devices, and its device drivers are open source, which means that they can be modified and customized by users and developers.

Overall, the Linux architecture is designed to be modular, flexible, and customizable, which makes it a popular choice for a wide range of applications, from servers and supercomputers to mobile devices and embedded systems.