**How to start dev ops from scratch**

Starting from scratch can be a daunting task, but with the right mindset and approach, anyone can start learning DevOps. Here are some steps you can take to start learning DevOps from scratch:

1. Understand the Key Concepts: Before you start working with DevOps tools and technologies, it's important to understand the key concepts behind DevOps. This includes understanding the principles of continuous integration, continuous delivery, and continuous deployment, as well as the importance of collaboration and automation.
2. Learn the Tools and Technologies: Once you have a good understanding of the key concepts behind DevOps, you can start learning the tools and technologies used in DevOps. Start by learning some basic scripting languages like Bash, Python, and PowerShell. Then, start exploring popular DevOps tools like Jenkins, Git, Docker, and Kubernetes.
3. Practice, Practice, Practice: DevOps is a hands-on practice, so it's important to practice what you've learned. Set up a small project, such as a website, and use DevOps tools to automate the deployment process. Try to automate as many tasks as possible, from building and testing to deployment and monitoring.
4. Join DevOps Communities: Joining DevOps communities, such as online forums, meetups, and conferences, can be a great way to learn from others and get support as you learn. Joining a community will help you stay up-to-date on the latest trends and technologies in DevOps.
5. Keep Learning: DevOps is constantly evolving, so it's important to keep learning and expanding your knowledge. Follow blogs and online resources, take online courses, and attend conferences to stay on top of the latest trends and technologies.

Starting from scratch with DevOps may seem overwhelming at first, but by following these steps and staying persistent, you can build a solid foundation in DevOps and start working towards your goals. Remember, DevOps is a journey, not a destination, so enjoy the process and keep learning along the way.

Python scripting for dev ops

Python is a popular scripting language for DevOps due to its simplicity, readability, and flexibility. Here are some examples of how Python can be used in DevOps:

1. Infrastructure as Code: Python can be used to automate the provisioning and management of infrastructure as code (IaC). You can use Python frameworks like Ansible, SaltStack, or Terraform to define infrastructure resources and configurations in a code format.
2. Continuous Integration and Deployment: Python can be used to automate the CI/CD pipeline. For example, you can use Jenkins or GitLab CI to automate the building, testing, and deployment of your code. Python can be used to write custom scripts or plugins to integrate with these CI/CD tools.
3. Monitoring and Alerting: Python can be used to automate the monitoring and alerting of your applications and infrastructure. You can use Python libraries like Prometheus or Nagios to write custom monitoring scripts or plugins.
4. Cloud Automation: Python can be used to automate the management of cloud infrastructure. For example, you can use Python libraries like Boto3 to automate the management of AWS resources, or use the Google Cloud SDK to automate the management of Google Cloud Platform resources.
5. Data Analysis: Python can be used for data analysis and visualization in DevOps. You can use Python libraries like Pandas and Matplotlib to analyze and visualize data from various sources, such as monitoring tools or log files.

Overall, Python is a versatile language that can be used in many different ways in DevOps. Its flexibility and readability make it a great choice for scripting tasks, automation, and data analysis