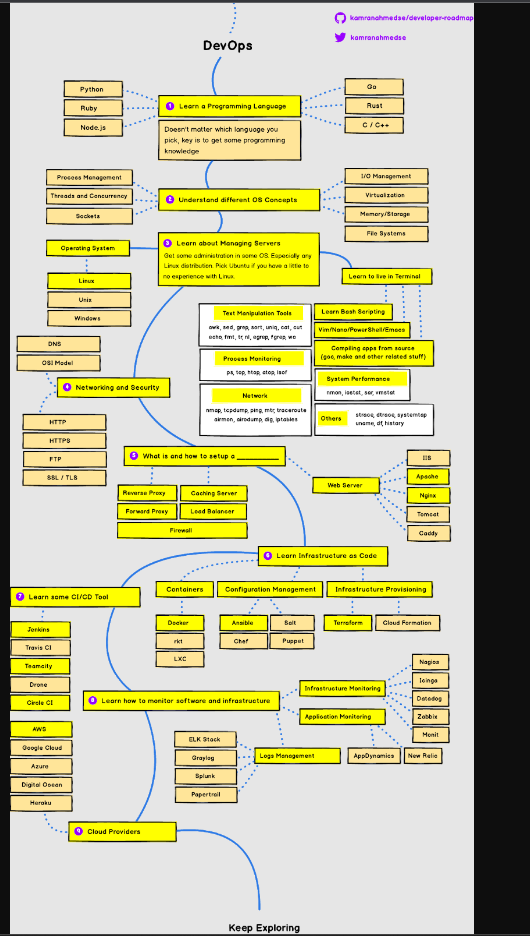
**[DevOps Engineer RoadMap [UPDATED]](https://javarevisited.blogspot.com/2018/09/the-2018-devops-roadmap-your-guide-to-become-DevOps-Engineer.html" \o "The 2023 DevOps Engineer RoadMap [UPDATED])**

**DevOps is really hot at the moment, and most of my friends, colleagues, and senior developers I know are working hard to become a DevOps engineers and project themselves as DevOps champions in their organization. While I truly understand the benefit of**[**DevOps**](https://javarevisited.blogspot.com/2018/09/10-devops-courses-for-experienced-java-developers.html)**, which is directly linked to improved software development and deployment, from my limited experience, I can say that it's not an easy job. It's tough to choose the right path in mind of so many tools and practices. Many of my readers also ask me this question is how to become a DevOps engineer, which means should I learn? Which practices should I follow? Does learning Maven and Jenkins is a must for DevOps Engineers?**

**How about**[**Docker**](https://javarevisited.blogspot.com/2019/05/top-5-courses-to-learn-docker-and-kubernetes-for-devops.html#axzz6A3SoXHZX)**and**[**Kubernetes**](https://dev.to/javinpaul/top-10-courses-to-learn-docker-and-kubernetes-for-programmers-4lg0)**? Does the infrastructure automation part of DevOps? Should I learn Chef, Puppet, or Ansible are just some of those questions which keep coming to me.  
  
I have tried hard to answer those with minimal experience. Still, I couldn't jot down in the manner which is simply excellent and reusable but not to worry, today I am going to share with you an awesome resource that will help you to become the DevOps Engineer you always wanted to be, the complete DevOps RoadMap.  
  
I was casually surfing through the internet yesterday when I come across**[**this**](https://github.com/kamranahmedse/developer-roadmap)**excellent GitHub page by**[**Kamranahmedse**](https://github.com/kamranahmedse)**, which shows a couple of useful roadmaps to become a front-end developer, back-end developer, a full-stack web developer, and last but not the least, the**[**DevOps Engineer**](https://medium.com/javarevisited/top-5-online-courses-to-become-a-devops-engineer-in-2020-764f5e60c2b)**.  
  
   
This RoadMap is fantastic in any sense as it not only highly what is the role of a DevOps engineer but also tells which tool you need to learn to cover that area. On top of that, it's really visually appealing with beautiful colors, so you can just take a printout and stick it on your desk for easier reference.  
  
In order to complete the roadmap, I have also shared some useful online courses so that you can learn and improve the tool or area you want.**

**The 2023 DevOps Engineer RoadMap for Developers**

**Anyway here is the complete DevOps RoadMap I am talking about:  
  
 **

**Now, let's go through the RoadMap step by step and find out how can we learn the essential skills required to become a DevOps guru in 2023:**

**1. Learn a Programming Language**

**Obviously, I assume you guys definitely know one of the three main programming languages, i.e., Java, Python, or JavaScript. If you didn't, don't worry, you can take a look at the courses below to learn your choice of language, though I strongly suggest you learn at least one of these three major general-purpose programming languages.**

**If you want to learn Java, then**[**The Complete Java**](https://click.linksynergy.com/fs-bin/click?id=JVFxdTr9V80&subid=0&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https%3A%2F%2Fwww.udemy.com%2Fjava-the-complete-java-developer-course%2F)**Masterclass is a great course, which is also recently updated for Java 10.**

**[](https://javarevisited.blogspot.sg/2017/11/top-5-free-java-courses-for-beginners.html)**

**If you want to learn Python, then**[**The Complete Python BootCamp**](https://click.linksynergy.com/fs-bin/click?id=JVFxdTr9V80&subid=0&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https%3A%2F%2Fwww.udemy.com%2Fcomplete-python-bootcamp%2F)**is my favorite resource, which will teach you Python 3, the most popular version of Python.**

**[](http://javarevisited.blogspot.sg/2018/03/top-5-courses-to-learn-python-in-2018.html)**

**And, if you want to learn JavaScript, then you should not look beyond Mosh Hamdani's**[**JavaScript Basics for Beginners**](https://click.linksynergy.com/fs-bin/click?id=JVFxdTr9V80&subid=0&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https%3A%2F%2Fwww.udemy.com%2Fjavascript-basics-for-beginners%2F)**course on Udemy.**

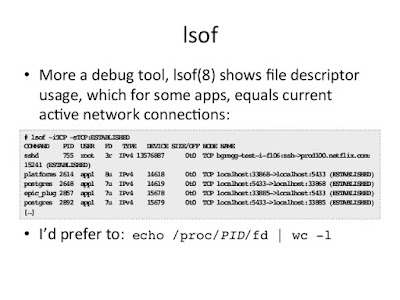
**[](https://javarevisited.blogspot.com/2018/06/top-10-courses-to-learn-javascript-in.html)**

**2. nderstand different OS concepts**

**This is where the Ops part came in, earlier it was solely supported by guys and sysadmin people who were responsible for knowing about OS and hardware, but with**[**DevOps**](https://medium.com/hackernoon/the-2018-devops-roadmap-31588d8670cb)**, now the developer also needs to know them.  
  
You at least need to know about Process Management, Threads and Concurrency, Sockets, I/O Management, Virtualization, Memory storage, and File systems, as suggested in the roadmap.  
  
Since most of us work in Linux, I suggest you go through the**[**Linux Administration BootCamp**](https://click.linksynergy.com/fs-bin/click?id=JVFxdTr9V80&subid=0&offerid=323058.1&type=10&tmpid=14538&RD_PARM1=https%3A%2F%2Fwww.udemy.com%2Flinux-administration-bootcamp%2F)**course on Udemy to learn and understand Linux OS better.**

**3. Learn to Live in terminal**

**For a DevOps guy, it's important to have good command in the command line, particularly if he is working in Linux. Knowing some Linux shells like**[**Bash**](https://medium.com/javarevisited/6-free-courses-to-learn-bash-shell-scripting-in-linux-and-unix-a50461ecd4fe)**, or Ksh and tools like**[**find**](http://javarevisited.blogspot.sg/2011/03/10-find-command-in-unix-examples-basic.html#axzz5E2uHdG3w)**,**[**grep**](http://www.java67.com/2017/07/grep-command-example-list-only-file-names-matching-string.html)**, awk, [sed](http://javarevisited.blogspot.sg/2013/05/sed-command-examples-in-unix-and-linux.html" \l "axzz56KhwFZ4z" \t "_blank), [lsof](http://javarevisited.blogspot.sg/2016/06/10-example-of-lsof-command-in-unix-linux.html" \l "axzz5CkWP96Nb" \t "_blank), and networking commands like nslookup and netstat is mandatory.  
  
If you feel you need to refresh these commands and tools, then you should join the**[**Linux Command Line Interface (CLI) Fundamentals**](https://pluralsight.pxf.io/c/1193463/424552/7490?u=https%3A%2F%2Fwww.pluralsight.com%2Fcourses%2Flinux-cli-fundamentals)**course on Pluralsight.**

**[](https://javarevisited.blogspot.com/2018/02/5-courses-to-learn-shell-scripting-in-linux.html)**

**4. Networking and Security**

**Gone are the days of isolation; in today's world, everything is connected to everything, which makes networking and security very important. In order to become a good DevOps engineer, you must know about basic networking and security concepts like DNS, OSI Model,**[**HTTP**](https://javarevisited.blogspot.com/2013/07/how-ssl-https-and-certificates-works-in-java-web-application.html)**, HTTPS, FTP, SSL, TLS, etc. In order to refresh this concept, you can take a look at this course on Pluralsight.**

**5. What is and how to setup**

**As a DevOps champion, you should know what is set up in your machine and how you can set that up, only that you can think about automating it. In general, a DevOps engineer should know how to set up a Web Server like IIS,**[**Apache**](https://javarevisited.blogspot.com/2017/01/12-essential-apache-web-server-interview-questions-answers-java-linux.html)**, and**[**Tomcat**](https://javarevisited.blogspot.com/2018/07/how-to-setup-jndi-database-connection-pool-tomcat-spring-example-tutorial.html)**. He should also know about Caching Server, Load balancer, Reverse Proxy, Firewall, etc.  
This is probably the most important thing for a DevOps engineer, and this is a very vast area as well. As a DevOps engineer, you should know about containers like**[**Docker**](https://javarevisited.blogspot.com/2018/02/10-free-docker-container-courses-for-Java-Developers.html)**and**[**Kubernetes**](https://javarevisited.blogspot.com/2018/09/10-devops-courses-for-experienced-java-developers.html)**, Configuration management tools like**[**Ansible**](https://javarevisited.blogspot.com/2019/11/top-5-course-to-learn-ansible-for-devops.html#axzz6OglQbBaX)**, Chef, Salt, and**[**Puppet**](https://javarevisited.blogspot.com/2020/05/top-5-puppet-courses-for-programmers-and-devops-engineers.html)**, Infrastructure Provisionings like Terraform and Cloud formation. Here are some of my recommended courses to learn these tools.**

**7. Learn some Continuous Integration and Delivery (CI/CD) tools**

**This is another very important thing for DevOps gurus and champions, i.e., to set up a pipeline for continuous integration and delivery. There are a lot of tools in the CI/CD area, like**[**Jenkins**](https://medium.com/javarevisited/7-best-courses-to-learn-jenkins-and-ci-cd-for-devops-engineers-and-software-developers-df2de8fe38f3)**, TeamCity, Drone, etc.  
  
But, I strongly recommend learning at least Jenkins, as it's the most widely used and probably the most sophisticated CI/CD tool in the market. If you don't know Jenkins, then this course is best to start with.**

**[](https://javarevisited.blogspot.com/2018/09/top-5-jenkins-courses-for-java-and-DevOps-Programmers.html)**

**8. Learn to monitor software and infrastructure**

**Apart from setup and deployment, monitoring is another important aspect of**[**DevOps**](https://medium.com/javarevisited/top-5-online-courses-to-become-a-devops-engineer-in-2020-764f5e60c2b)**, and that's why it's important for a**[**DevOps engineer**](https://dev.to/javinpaul/top-10-devops-training-courses-for-programmers-and-software-developers-195n)**to learn about Infrastructure and application monitoring.  
  
There are a lot of tools in this space, like Nagios, Icing,**[**Datadog**](https://www.datadoghq.com/)**, Zabbix, Monit, AppDynanic, New Relic, etc. You can choose some of them depending upon which one is used in your company like AppDynamic and Nagios.**

**9. Learn about Cloud Provides**

**Cloud is the next big thing, and sooner or later, you have to move your application to the cloud; hence it's important for a DevOps engineer to at least know about some of the popular Cloud Providers and their basics.  
  
While**[**AWS**](http://www.java67.com/2018/05/top-5-amazon-web-services-or-aws-courses-to-learn-online.html)**is clearly the leader in the cloud it's not alone, Google Cloud and Azure are slowly catching up, and then we have some other players like Heroku, Cloud Foundry, and Digital Ocean.**

**Closing Notes**

**Thanks for reading this article so far … Good luck on your DevOps journey! It’s certainly not going to be easy, but by following this roadmap and guide, you are one step closer to becoming a DevOps engineer. Btw, don't get overwhelmed by the size of this RoadMap; there is a good chance that you already know most of the stuff; just focus on what you don't know and go deep on essential tools and technologies like Docker, Jenkins, and Kubernetes.  
  
   
Biswaraj Sahoo**

**Click here to check the Linktree Post…**

**https://linktr.ee/biswaraj333**