**Navigating DevOps Learning Process as a Newbie**

Transitioning into a DevOps career can be an exciting yet daunting prospect. People consider this career switch more as there is a higher demand for DevOps professionals. In this blog post, we will explore how to navigate the learning process as a newbie and provide compelling reasons for choosing DevOps as a career.

**DevOps and its key Concept**

DevOps is a combination of practices, tools, and cultural philosophies that automate and integrate the processes between software development and IT teams. It emphasizes collaboration, continuous integration, delivery, and deployment to improve the speed and quality of software development.

**Some Key Concepts in DevOps:**

* **Continuous Integration (CI):** Regularly merging code changes into a shared repository.
* **Continuous Delivery (CD):** Ensuring the software can be reliably released at any time.
* **Infrastructure as Code (IaC):** Managing and provisioning computing infrastructure through machine-readable configuration files.

Crafting a meticulously structured learning plan can enhance your skills, accelerate your progress, focus your efforts, and provide a clear roadmap through the often overwhelming world of new technologies. A well-crafted learning plan can help transform you from a complete beginner to a confident DevOps professional, setting you on a promising career trajectory in today's rapidly evolving economy.

**Six-month Solid Learning Plan for a newbie:**

### **Month 1: Foundations**

**Week 1-2: Introduction to DevOps**

* Study up on the concepts and background of DevOps.
* Examine the advantages and difficulties of DevOps.
* To gain an overview of DevOps concepts, read Gene Kim's "The Phoenix Project".

**Week 3-4: Linux Basics**

* Recognise shell programming, the Linux file system, and fundamental commands.
* Try out an operating system that runs on Linux (Ubuntu, CentOS, etc.).
* Resources: Linux courses at freeCodeCamp and Linux Academy.

### **Month 2: Version Control & CI/CD Basics**

**Week 1-2: Git and Version Control**

* Learn Git basics: cloning, committing, branching, merging.
* Explore advanced Git concepts: rebasing, stashing, tagging.
* Resources: Pro Git book, GitHub Learning Lab.

**Week 3-4: Continuous Integration**

* Understand the principles of Continuous Integration (CI).
* Set up a simple CI pipeline using tools like Jenkins or GitHub Actions.
* Resources: Jenkins documentation, GitHub Actions guides.

### **Month 3: Configuration Management & IaC**

**Week 1-2: Configuration Management**

* Learn about configuration management tools like Ansible, Puppet, and Chef.
* Practice writing Ansible playbooks or Puppet manifests.
* Resources: Ansible documentation, Puppet Learning VM.

**Week 3-4: Infrastructure as Code (IaC)**

* Understand the concept of Infrastructure as Code.
* Learn Terraform basics: writing and applying Terraform configurations.
* Resources: Terraform documentation, HashiCorp Learn platform.

### **Month 4: Containerization & Orchestration**

**Week 1-2: Docker**

* Learn Docker basics: images, containers, Dockerfile, Docker Compose.
* Practice building and running Docker containers.
* Resources: Docker documentation, Docker for Developers course.

**Week 3-4: Kubernetes**

* Understand Kubernetes architecture and key concepts.
* Set up a local Kubernetes cluster using Minikube or Kind.
* Deploy applications to Kubernetes and practice scaling and managing them.
* Resources: Kubernetes documentation, Kubernetes by Example.

### **Month 5: Monitoring & Logging**

**Week 1-2: Monitoring**

* Learn about monitoring principles and tools (Prometheus, Grafana).
* Set up Prometheus and Grafana to monitor a sample application.
* Resources: Prometheus documentation, Grafana tutorials.

**Week 3-4: Logging**

* Understand the importance of centralized logging.
* Learn about ELK stack (Elasticsearch, Logstash, Kibana) or EFK stack (Fluentd instead of Logstash).
* Set up centralized logging for your applications.
* Resources: Elastic documentation, Fluentd docs.

### **Month 6: Security & Advanced Topics**

**Week 1-2: DevSecOps**

* Learn about integrating security into the DevOps pipeline.
* Study security best practices for CI/CD and cloud environments.
* Resources: DevSecOps blogs, OWASP DevSecOps guidelines.

**Week 3-4: Advanced Topics**

* Explore advanced topics like GitOps, service mesh (Istio), and serverless.
* Choose a project to implement what you've learned, such as setting up a full CI/CD pipeline with security and monitoring.
* Resources: Official documentation, community tutorials.

### **Continuous Learning & Practice**

* Join DevOps communities (Slack, Reddit, etc.) to stay updated and seek help.
* Contribute to open-source projects to gain real-world experience.
* Consider certifications like AWS Certified DevOps Engineer, Docker Certified Associate, or Certified Kubernetes Administrator (CKA).

#### 

#### **Reasons to Switch to a DevOps Career**

**1. High Demand and Competitive Salaries:**

* The demand for DevOps professionals is on the rise, leading to lucrative job opportunities and competitive salaries.

**2. Diverse Career Opportunities:**

* DevOps skills are applicable across various industries, from tech startups to large enterprises.

**3. Continuous Learning and Growth:**

* The DevOps field is dynamic, offering continuous learning opportunities and keeping professionals engaged.

**4. Improved Collaboration and Efficiency:**

* DevOps fosters a collaborative work environment, breaking down silos between development and operations teams, leading to more efficient and effective workflows.

**5. Enhanced Job Satisfaction:**

* The ability to automate tasks and streamline processes can lead to a more satisfying and less stressful work environment.

**6. Contribution to Innovation:**

* DevOps practices enable faster delivery of software updates and innovations, making a tangible impact on the business and user experience.

(Try and get one or two pictures/diagrams into the post. For the learning plan, what is your source? Or did you curate that yourself? If not then you need to reference the source. You should also add a short conclusion or short summary stating that we do this and can help a newbie transition to devops, linking our website for them to get started)