**Here’s a structured roadmap 👇**

**🛠 DevOps Roadmap for a System Administrator**

1. Strengthen Your Foundations (you already have partial skills)

✅ You already know: OS, servers, networking, troubleshooting.  
🔹 Add/Improve on:

* Linux basics → shell scripting (bash), package management, systemd, monitoring.
* Windows basics → PowerShell scripting, IIS, Active Directory (you probably know this).
* Networking → DNS, Load Balancers, Reverse Proxies (Nginx/HAProxy).

**2. Learn Version Control (Git + GitHub/Azure Repos)**

* Understand git clone, commit, push, branch, merge, rebase.
* Practice with small projects (you’re already doing this ✅).
* Learn Git branching strategies (GitFlow, Trunk-based).

**3. Build CI/CD Skills (Azure DevOps, GitHub Actions, Jenkins)**

* Learn YAML pipelines (you already started 💯).
* Write simple pipelines: build → test → publish artifacts.
* Learn multi-stage pipelines (Build → Test → Deploy).
* Get hands-on with Azure DevOps Pipelines (since you’re already inside Azure).

**4. Configuration Management**

* Learn Infrastructure as Code (IaC) tools:
  + ARM/Bicep templates → Azure-specific infra-automation.
  + Terraform → Provision cloud infra (Azure/AWS).
  + Ansible → Automate server configurations.

**5. Containers & Orchestration**

* Learn Docker: build images, run containers.
* Learn Kubernetes basics: pods, services, deployments.
* Use AKS (Azure Kubernetes Service) for practice.

**6. Cloud Skills (Azure recommended for you)**

* Learn Azure basics:
  + Azure VMs, Storage, Networking.
  + Azure DevOps, Azure CLI, Resource Groups.
* Learn Azure PaaS:
  + App Service, Functions, Azure SQL.
* Practice deploying apps from your pipeline → Azure App Service.

**7. Monitoring & Logging**

* Learn Prometheus + Grafana or Azure Monitor + Log Analytics.
* Understand how to collect logs, create dashboards, set alerts.

**8. Security & Best Practices**

* Secrets management → Azure Key Vault.
* Secure CI/CD → no passwords in code.
* Role-based access (least privilege).

**🎯 How You Can Progress Step by Step**

1. **Short-term (1–3 months):**
   * Master Git + Azure DevOps pipelines (CI/CD).
   * Learn basic scripting (bash + PowerShell).
   * Deploy small apps to Azure App Service with pipelines.
2. **Medium-term (3–6 months):**
   * Learn Terraform/Ansible (infra automation).
   * Build pipelines that provision infra + deploy apps.
   * Learn Docker basics.
3. **Long-term (6–12 months):**
   * Learn Kubernetes (AKS).
   * Integrate monitoring + alerting into pipelines.
   * Work on real-world DevOps projects (infra as code + CI/CD + containers).

**📚 Recommended Resources**

* Git → [learngitbranching](https://learngitbranching.js.org/) (interactive).
* Azure DevOps → Microsoft Learn modules (free).
* Terraform → HashiCorp Learn.
* Docker → Play with Docker labs.
* Kubernetes → Kubernetes by Example.

**DevOps** is a combination of "Development" and "Operations"—a culture, set of practices, and tools that bring together software development and IT operations teams to:

* Shorten the software development lifecycle, and
* Deliver high-quality software continuously (known as CI/CD – Continuous Integration/Continuous Delivery).

🔑 **Key Concepts in DevOps**

| **Concept** | **Description** |
| --- | --- |
| Collaboration | Developers and operations teams work together instead of in silos. |
| Automation | Repetitive tasks (e.g., testing, deployment, monitoring) are automated. |
| CI/CD | Continuous Integration & Continuous Delivery – ensures faster and reliable releases. |
| Monitoring | Systems are continuously monitored to detect and respond to issues quickly. |
| Infrastructure as Code (IaC) | Infrastructure is managed using code (e.g., Terraform, Ansible). |

🧰 Common DevOps Tools

| **Function** | **Tools** |
| --- | --- |
| Version Control | Git, GitHub, GitLab |
| CI/CD | Jenkins, GitHub Actions, Azure DevOps, GitLab CI |
| Configuration Management | Ansible, Puppet, Chef |
| Containerization | Docker |
| Orchestration | Kubernetes |
| Monitoring | Prometheus, Grafana, ELK Stack |
| Cloud Platforms | AWS, Azure, GCP |

**📈 Benefits of DevOps**

* Faster delivery of features and fixes
* Improved collaboration and efficiency
* Better product quality and reliability
* Reduced deployment failures and rollback time
* Continuous feedback and monitoring

**🌍 Real-Life Example**

Instead of a developer writing code and "throwing it over the wall" to the operations team to deploy, DevOps ensures:

* Code is automatically tested and deployed via pipelines.
* Systems are monitored in real-time.
* Any issues are quickly fixed and deployed again with minimal downtime.