



TECHWORLD  
WITH NANA

# Study Guide

*To Become  
Proficient in DevOps  
in 6 months*





## Copyright Protected

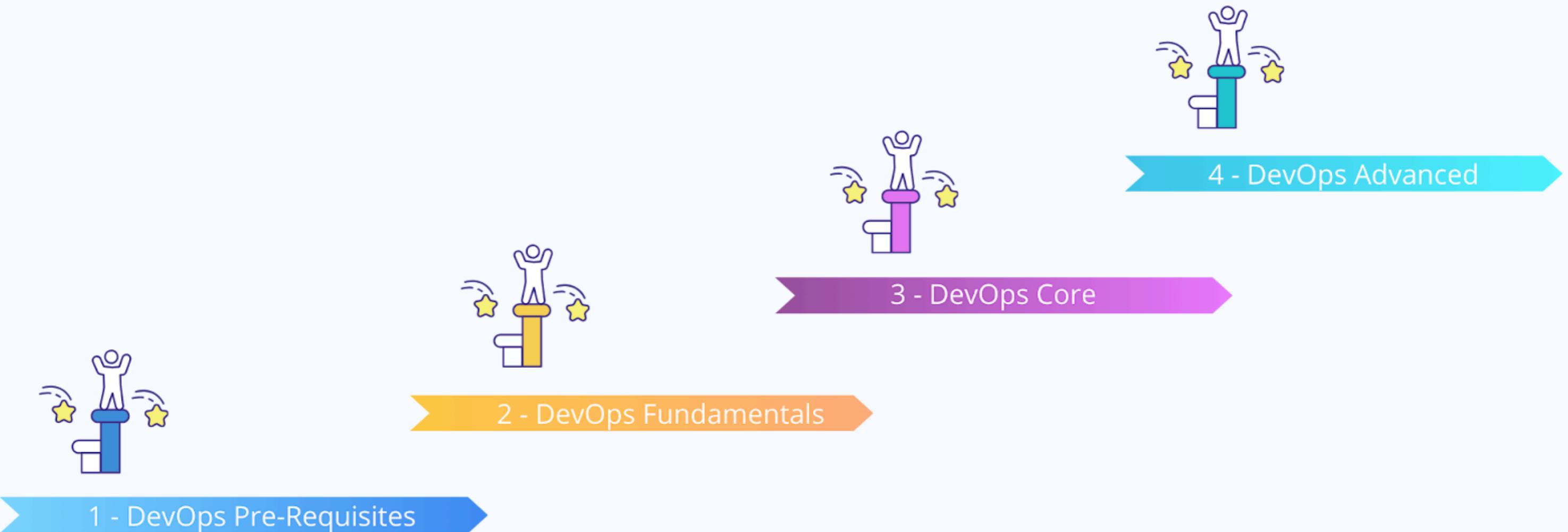
### Terms of Use:

*Any materials provided by TechWorld with Nana are copyright protected and license is given for personal usage.*

*Under this license you may not: modify or copy the materials; use the materials for any public display; transfer the materials to another person.*

# Master the DevOps Game

Go through these **4 LEVELS** - from Zero to DevOps Mastery:



**3 WEEKS**



## Level 1 - DevOps Prerequisites:

Many newcomers to Cloud and DevOps struggle due to a lack of foundational knowledge. These modules bridge that gap by providing essential foundational knowledge of Linux, Git, and build tools, which are critical for any DevOps engineer.

**4 WEEKS**



## Level 2 - DevOps Fundamentals:

Now, your DevOps journey really starts. These are really the non-negotiable skills for a DevOps engineer. You will be learning Cloud basics and gaining hands-on experience working with remote VMs in the Cloud, as well as artifact repositories with Nexus and containerization with Docker.

**9 WEEKS**



## Level 3 - DevOps Core:

This is where we take your skills to an advanced level and speed up your career growth. You will be learning CI/CD pipelines, container orchestration with K8s, complex cloud infrastructure with AWS, K8s on AWS with EKS and building multiple real life capstone projects with all of the tools you have learned in each of the previous modules.

**10 WEEKS**



## Level 4 - DevOps Advanced:

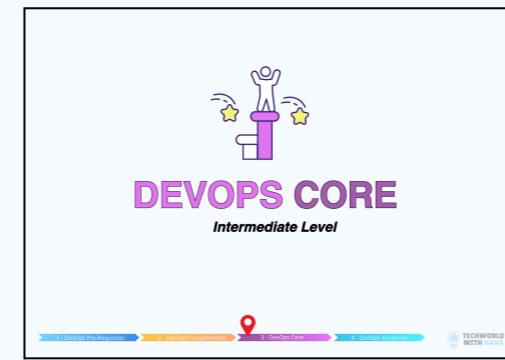
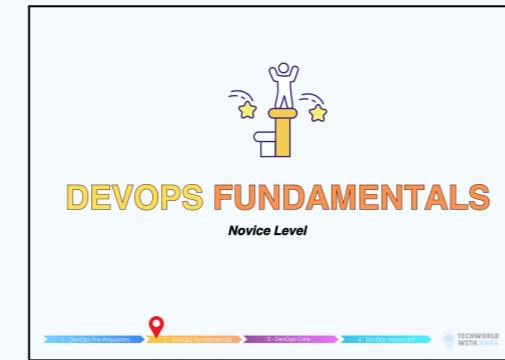
Elevate to an expert level by mastering 4 different tools of automating every part of your systems! Terraform for infrastructure, Ansible for configuration management, Python for any kind of smaller flexible task automation and finally monitoring with Prometheus for detecting and alerting any problems or issues in your systems automatically.

# Learning Schedule - Overview

This is your learning schedule to **complete each level** successfully in a total of 6 months with an estimated of 15 hours per week:



We prepared separate study guides for each learning phase.



# Need some Guidance?



We know learning DevOps can be extremely challenging. That's why we created the DevOps Bootcamp to help engineers from zero DevOps know-how to become confident Senior DevOps and Cloud Engineers.

If you want to **save yourself time**, have a **guaranteed successful career upgrade** and become a “Certified DevOps Practitioner” with our certification, then check out our intensive program [HERE](#) or copy link: <https://www.techworld-with-nana.com/devops-bootcamp>

# Or Need some Motivation?

Two side-by-side cards. The left card is titled "Mastering DevOps... A software developer's path to upskill" featuring Kevin Pirner, a Software Developer, with a diagram showing a path from "Mid-Level" to "Senior". The right card is titled "From Sys Admin to DevOps Engineer: Jose's..." featuring Jose Maria Briones, a DevOps Engineer, with a diagram showing a path from "Systems Administrator" to "Software Developer" to "DevOps Engineer".

We successfully educated 10,000s+ of engineers. Read about their inspiring DevOps journeys from different backgrounds, like developer, sys admin etc. Read [HERE](#) or copy link: <https://www.techworld-with-nana.com/success-stories>

If they can do it, YOU can too! 🤓



# 1 - DevOps Pre-Requisites

## Week 1

### Weekly Progress Tracker

Focus: Linux Operating System

#### Key Topics for the Week

- **OS and VM Basics:** Overview of OS types, RAM, storage, and virtual machines (Type 1 vs Type 2)
- **CLI and Commands:** Key Linux commands (`pwd`, `ls`, `rm`), GUI vs CLI
- **User and File Management:** User creation, setting permissions, manage file ownership (`chown`)
- **Shell Scripting:** Writing bash scripts with conditionals, functions, and best practices

#### Checklist for This Week

- Complete Ubuntu VM setup on VirtualBox
- Practise Linux commands

## Week 2

### Weekly Progress Tracker

Focus: Linux Operating System & Git Version Control

#### Key Topics for the Week

- **Environment Variables:** Define and print environment variables, and understand their use cases
- **Networking Fundamentals:** IP addressing, Subnets, NAT, Firewalls and DNS
- **Version Control Overview:** Understand version control, merges, conflicts, and best practices
- **Git Commands:** Practise essential Git commands - `git add`, `git commit`, `git log`, `git pull`
- **Branching and Conflicts:** Manage branches, create and resolve merge conflicts, and practice stashing files

#### Checklist for This Week

- Configure environment variables and practice using them in scripts
- Create a DigitalOcean Droplet and connect via SSH
- Practice using basic Git commands
- Work on multiple branches and resolve a merge conflict

## Week 3

### Weekly Progress Tracker

Focus: Git Version Control & Build Tools

#### Key Topics for the Week

- **Branch Merging:** Practise merging bugfix and master branches
- **Databases in Software Development:** Overview of local vs. remote databases, how to configure database connections and best practices for managing databases in prod. environments
- **Types of Databases:** Explanation of different database types, including key-value, wide column, document, relational, graph, and search databases, and their use cases in modern applications
- **Build Tools:** Understand the purpose and use cases of build tools like Maven and Gradle
- **Windows vs Linux CLI:** Learn key differences between Windows and Linux CLI
- **Package Management:** Explore pom.xml, build.gradle, and package.json files for dependency management in Maven and Gradle
- **Build Tools for DevOps:** Understand how Docker simplifies build processes with an example Dockerfile and how to deploy build tools as part of CI/CD pipelines

#### Checklist for This Week

- Merge bugfix and master branches and resolve conflicts
- Install IntelliJ IDEA, Java SDK, Node.js, and Git on your local machine
- Build Maven and Gradle applications
- Review the pom.xml, build.gradle, and package.json files



## 2 - DevOps Fundamentals

### Week 4

#### Weekly Progress Tracker

**Focus:** Cloud Basics & Nexus Artifact Repository Manager

#### Key Topics for the Week

- **Cloud and IaaS Concepts:** High-level overview of cloud computing, focusing on Infrastructure as a Service (IaaS) and its key components
- **Deploying Applications:** How to build a JAR file and run an application on cloud infrastructure
- **Repository Management:** Learn about repository managers, like Nexus, their purpose, and key features
- **Nexus Installation and Configuration:** How to Install Nexus on a DigitalOcean droplet, configure its service, and access it via a browser
- **Managing Repositories:** How to create a proxy repository in Nexus, configure roles, and publish an application by updating the build.gradle file

#### Checklist for This Week

- Create and configure a DigitalOcean droplet with the correct firewall settings and Java installed
- Build a JAR file locally and deploy it to the droplet
- Add a new user to the droplet and configure user permissions
- Install and configure Nexus
- Set up a Docker proxy repository and publish a build by updating your build.gradle file

### Week 5

#### Weekly Progress Tracker

**Focus:** Nexus Artifact Repository Manager & Containers with Docker

#### Key Topics for the Week

- **Nexus Repository Manager:** Understand GET request responses from a Nexus server, blob stores, and storage considerations
- **Repository and Cleanup Policies:** Deep dive into creating repositories, setting cleanup policies, and scheduling tasks to manage storage efficiently
- **Containers and Docker Basics:** Introduction to containers and Docker, covering installation and essential commands such as `- docker run, docker pull, docker ps, docker stop, docker logs, and docker exec`
- **Docker vs VMs:** How Docker differs from virtual machines
- **Docker Architecture:** Docker Server, Docker API, Docker CLI

#### Checklist for This Week

- Demo a GET request from the Nexus server and verify the response
- Create a blob store in Nexus and configure storage
- Set up a repository and define a cleanup policy in Nexus
- Install Docker and run basic commands (`docker run, docker pull, docker ps`)
- Run more advanced Docker commands like `docker stop` and `docker logs` to debug containers

### Week 6

#### Weekly Progress Tracker

**Focus:** Containers with Docker

#### Key Topics for the Week

- **Docker Networking:** How Docker networks allow containerized services to communicate
- **Docker Compose:** Understand how Docker Compose can be used to build and deploy an application
- **Dockerfile & Image Building:** Learn how to write a Dockerfile and use it to build a custom image
- **AWS ECR:** Learn how to push and pull images from Amazon Elastic Container Registry (ECR) and manage Docker image tagging
- **Volumes in Docker:** Understand how Docker volumes manage persistent storage for containers

#### Checklist for This Week

- Set up Mongo Express with Docker and create a Docker network for communication between containers
- Use Docker Compose to deploy a multi-container application
- Write a Dockerfile and build a custom image
- Push and pull Docker images to/from AWS ECR, including image tagging
- Implement Docker volumes to manage persistent data in your containers



## 2 - DevOps Fundamentals

### Week 7

#### Weekly Progress Tracker

**Focus:** Containers with Docker

#### Key Topics for the Week

- **Adding Volumes to Docker Applications:** Learn how to attach Docker volumes to a Node.js application for persistent data storage
- **Nexus for Docker Image Hosting:** Understand how Nexus can be used as a private Docker registry to host Docker images
- **Running Nexus in Docker:** Set up and run Nexus as a Docker container using the official Nexus Docker image
- **Best Practices for Docker:** Explore essential best practices for managing Docker images, containers, and volumes

#### Checklist for This Week

- Add Docker volumes to your Node.js application for data persistence
- Configure Nexus to host Docker images and manage a private Docker registry
- Run Nexus as a Docker container using the official Nexus image



## 3 - DevOps Core - Part 1

### Week 8

#### Weekly Progress Tracker

Focus: CI/CD with Jenkins

#### Key Topics for the Week

- **Jenkins in Docker:** Deploy Jenkins on a DigitalOcean droplet & run it in a Docker container
- **Jenkins UI Tutorial:** Navigating and using the Jenkins user interface
- **Gradle & Maven in Jenkins:** See how Gradle and Maven are used in Jenkins for building and managing Java projects
- **Jenkins Freestyle Jobs:** Learn how to create a freestyle job, connect it to a Git project, and execute shell commands within the job
- **Docker Integration:** Learn how to install Docker within Jenkins and how to run Docker commands in Jenkins
- **Jenkins Pipeline Jobs:** Understand the concept of Pipeline jobs as well as their benefits and use cases

#### Checklist for This Week

- Set up a DigitalOcean droplet, install Docker, and run a Jenkins container
- Configure Jenkins
- Install Gradle and Maven in Jenkins and verify setup
- Create a freestyle job connected to a Git project and execute shell commands
- Configure Jenkins volumes, install Docker, and integrate Docker build/login/push in Jenkins

#### Recommendations & Tips

- **Tip:** Ensure you pay close attention to indentation when creating and modifying your Jenkinsfile
- **Tip:** If you notice performance issues with your Jenkins server, and your allocated resources are sufficient - try updating any additional plugins to see if it improves performance

### Week 9

#### Weekly Progress Tracker

Focus: CI/CD with Jenkins

#### Key Topics for the Week

- **Creating a Jenkinsfile:** How a Jenkinsfile can be used to automate build, test, and deployment stages in a CI/CD pipeline
- **Advanced Jenkinsfile Configuration:** Using multiple stages in a Jenkinsfile, adding shell commands, credentials, and Groovy scripts to enhance pipeline functionality
- **Jenkins Credentials:** Explore the different types of credentials supported in Jenkins and when to use each (e.g., secret text, SSH keys, username/password)
- **Multibranch Pipeline:** How Jenkins can be configured to automatically build different branches from your Git repository
- **Groovy Scripts for Shared Libraries:** See how script.groovy files for Jenkins shared libraries can be used to centralize common pipeline code across multiple projects
- **GitLab Integration:** How to use the GitLab plugin to configure webhooks for automatic pipeline triggers when changes are pushed to the GitLab repository

#### Checklist for This Week

- Write a Jenkinsfile to automate build, test, and deployment processes
- Implement multiple stages in the Jenkinsfile, including shell commands and Groovy scripting
- Create a Jenkins pipeline with defined credentials
- Set up a multibranch pipeline to handle multiple branches of your Git project
- Write and test a script.groovy for a shared library in Jenkins
- Install and configure the GitLab plugin in Jenkins, and add a webhook to trigger pipelines automatically

### Week 10

#### Weekly Progress Tracker

Focus: CI/CD with Jenkins & AWS Cloud

#### Key Topics for the Week

- **Pom.xml file:** How to use a pom.xml file to automate version increments during builds
- **Jenkinsfile Version Management:** How to use a Jenkins pipeline to increment the version, commit, and push changes
- **AWS Introduction:** Understanding AWS' UI, key services such as IAM and VPC and the Free Tier
- **AWS Global Infrastructure:** AWS regions, availability zones

#### Checklist for This Week

- Configure pom.xml to increment the java-app project version automatically during Jenkins builds
- Update Jenkinsfile to handle version increments, commit, and push to Git
- Explore the Jenkins UI and understand key services in pipeline management
- Set up an IAM user, configure roles and groups

#### Recommendations & Tips

- **Reminder:** Keep an eye on your free-tier usage in AWS to prevent accidental charges



## 3 - DevOps Core - Part 1

### Week 11

#### Weekly Progress Tracker

Focus: AWS Cloud

#### Key Topics for the Week

- **CIDR Blocks:** Understand CIDR notation, subnetting, and how to use subnet calculators for network planning
- **EC2 Instance management:** Instances, key pairs & security groups
- **Jenkins & Docker Integration with EC2:** How to increment your app's version, build and push an application image and deploy to EC2 using a Jenkins pipeline
- **AWS CLI:** How AWS CLI commands can be used to manage your infrastructure
- **Introduction to IaC and Microservices:** Understanding Infrastructure as Code (IaC) principles and the concept of microservices, and Amazon EKS for container orchestration

#### Checklist for This Week

- Use a subnet calculator to understand CIDR and subnetting
- Create an EC2 instance, configure SSH, install Docker, pull an image, and run the app
- Add the EC2 SSH key to Jenkins and update the Jenkinsfile with Docker and EC2 commands
- Install Docker Compose on EC2, create a docker-compose.yml file, and copy it using Jenkins
- Install and configure AWS CLI, create a security group, and launch an EC2 instance using CLI commands



## 3 - DevOps Core - Part 2

### Week 12

#### Weekly Progress Tracker

**Focus:** Container Orchestration with Kubernetes

#### Key Topics for the Week

- **Introduction to Containers and Orchestration:** Understand the need for containers and explore features of container orchestration tools like Kubernetes.
- **Kubernetes Components:** Learn about nodes, pods, services (internal and external), volumes, deployments, statefulsets, replicas, and daemonsets
- **Kubernetes Architecture:** Understand key Kubernetes elements: worker nodes, container runtime, kubelet and control plane processes
- **Kubectl & Minikube:** Introduction to kubectl and Minikube
- **Using Kubectl:** Essential kubectl commands like get nodes, create/edit deployments, get logs, describe resources
- **YAML Configuration in Kubernetes:** Service and deployment structure, specifications, statuses, indentation rules, and port configuration
- **Kubernetes Namespaces and Services:** Kubernetes namespaces and service types like ClusterIP, NodePort, and LoadBalancer

#### Checklist for This Week

- Install Minikube and Docker Desktop, and run basic kubectl commands for managing clusters
- Write and deploy a Kubernetes YAML file, ensuring correct indentation and syntax
- Deploy MongoDB with secrets, services, and a Mongo Express deployment using a LoadBalancer

#### Recommendations & Tips

- **Reminder:** Pay close attention to YAML indentation when writing Kubernetes configurations as errors can lead to failed deployments.

### Week 13

#### Weekly Progress Tracker

**Focus:** Container Orchestration with Kubernetes

#### Key Topics for the Week

- **Ingress & Routing Rules:** Learn about Ingress, routing rules, and Ingress controllers.
- **Minikube:** Explore Minikube Dashboard and Minikube tunnel, and understand TLS certificates for secure communication.
- **Persistent Storage in Kubernetes:** Explanation of Persistent Volumes (PVs), Persistent Volume Claims (PVCs), ConfigMaps, Secrets, volume types, and storage classes.
- **Stateful Sets and Scaling:** Explore StatefulSets in Kubernetes, focusing on scaling database applications and managing stateful workloads.
- **Managed vs Unmanaged Clusters:** Understand the differences between managed and unmanaged Kubernetes clusters, data persistence, and load balancing considerations
- **Helm and Package Management:** Learn about Helm, charts, the Artifact Hub, templating and values, and how to manage releases using Helm
- **Kubernetes Operator:** What it is, why and when it is used

#### Checklist for This Week

- Configure Minikube dashboard and tunnel
- Create a Mosquitto deployment using volumes and a secret file for configuration.
- Compare managed and unmanaged Kubernetes clusters, focusing on load balancing and data persistence
- Install a Bitnami Helm chart in an LKE cluster, including MongoDB, Nginx Ingress Controller, and Mongo Express
- Deploy images from a private docker repository - create a secret, authenticate with AWS ECR, and deploy from ECR to your cluster

### Week 14

#### Weekly Progress Tracker

**Focus:** Container Orchestration with Kubernetes

#### Key Topics for the Week

- **Kubernetes RBAC and Role Configuration:** Roles, RoleBindings, and RBAC (Role-Based Access Control) in Kubernetes. Learn about users, groups, external authentication, API server communication, and application authentication
- **Microservices:** Introduction to microservices, their benefits, and how services communicate with each other within a distributed system
- **Linode Kubernetes Engine:** How to deploy microservices on Linode Kubernetes Engine (LKE).
- **Kubernetes Best Practices:** Learn about version management, health checks, liveness probes, resource limits, and load balancers for microservices, as well as vulnerability scanning
- **Helm Charts and Deployments:** How to use YAML configuration files to standardize deployments, a Helmfile to manage multiple charts, and where to host and manage Helm charts for future use.

#### Checklist for This Week

- Create a YAML file for microservices, including variables and ports, and deploy them on LKE
- Set up health checks, liveness probes, and resource limits for microservices, and ensure proper load balancing
- Create a Helm chart with appropriate values and configuration, and deploy it to LKE
- Create a Helmfile to manage multiple deployments and understand where to host Helm charts for reuse

#### Recommendations & Tips

- **Tip:** When configuring RBAC, start with least-privileged access to enhance security within your cluster.



# 3 - DevOps Core - Part 3

## Week 15

### Weekly Progress Tracker

Focus: Kubernetes on AWS (EKS)

### Key Topics for the Week

- **Containers on AWS:** Overview of container services on AWS—ECS, Fargate, and EKS. Understand the pros and cons of each service and compare their use cases.
- **EKS Setup:** Learn how to use IAM roles and VPCs worker nodes using CloudFormation to create an EKS cluster. Understand how EC2 roles can be used to set up a node group to manage compute resources.
- **Autoscaling in EKS:** Understand autoscaling in an EKS cluster, how to review autoscaling logs, and ensure resource efficiency, and how a load balancer can be used to manage traffic distribution.
- **Fargate:** See how to use Fargate for running serverless containers

### Checklist for This Week

- Compare ECS, Fargate, and EKS, understanding their pros, cons, and use cases
- Create an EKS IAM role, VPC, EC2 role, and node group using CloudFormation
- Configure autoscaling for EKS, monitor logs, and set up a load balancer for traffic management
- Create a Fargate role and profile, and run serverless containers using Fargate
- Safely terminate the EKS cluster, node groups, and Fargate infrastructure to avoid ongoing costs

### Recommendations & Tips

- **Tip:** Ensure you are creating the EKS cluster role with the correct AWS account to avoid any permissions issues

## Week 16

### Weekly Progress Tracker

Focus: Kubernetes on AWS (EKS)

### Key Topics for the Week

- **EKSCTL Setup and Infrastructure:** Learn how the eksctl command line tool can be used to provision infrastructure in AWS
- **Deployment to EKS and LKE from Jenkins:** How to deploy to EKS and LKE using Jenkins pipeline
- **Jenkins Credential Management Best Practices:** Explore best practices for managing credentials securely in Jenkins and Kubernetes
- **Complete CI/CD Pipeline:** How to configure your environment to run a complete CI/CD pipeline using EKS, DockerHub and ECR

### Checklist for This Week

- Install eksctl and recreate the EKS infrastructure from previous lectures using the command line
- Configure AWS IAM Authenticator, kubectl, and kubeconfig in Jenkins, and update the Jenkinsfile for EKS deployment
- Create an LKE cluster, configure Jenkins for LKE authentication, and update the Jenkinsfile
- Update the deployment YAML and Jenkinsfile to deploy from DockerHub using Jenkins
- Create an ECR repository, set up credentials in Jenkins, and deploy the application from ECR



## 4 - DevOps Advanced - Part 1

### Week 17

#### Weekly Progress Tracker

**Focus:** Infrastructure as Code with Terraform

#### Key Topics for the Week

- **What is Terraform:** Understand what Terraform is and how it works. Compare Terraform with Ansible, focusing on differences in architecture and the declarative vs. imperative approach.
- **Providers in Terraform:** Explanation of Terraform providers and how to create an AWS provider.
- **Terraform Commands:** Useful commands such as - `terraform init`, `terraform plan`, `terraform apply`
- **Resources & Data Sources:** Learn about Terraform resources and data sources and how they can be used to create infrastructure like VPCs and subnets
- **Terraform State and Outputs:** Explore the state file and state-related commands
- **Terraform Variables:** How variables and environment variables can be added to projects

#### Checklist for This Week

- Install Terraform and the VS Code plugin for better code management
- Create an AWS provider using Terraform and initialize it with `terraform init`
- Write Terraform code to create a VPC and subnet, and explore data sources
- Add tags to Terraform resources, use `terraform plan`, `apply` with auto-approve, and `destroy` to manage infrastructure
- Learn about Terraform state files, commands, and outputs, and apply them in a demo project

#### Recommendations & Tips

- **Tip:** Use the `terraform plan` command to preview changes before applying them to avoid unexpected resource modifications

### Week 18

#### Weekly Progress Tracker

**Focus:** Infrastructure as Code with Terraform

#### Key Topics for the Week

- **Terraform Network Configuration:** Learn how to define your network infrastructure (e.g. VPCs, subnets) in a Terraform file.
- **Terraform - AMIs and SSH Keys:** Use AMIs to generate a key pair and create an EC2 instance
- **User Data:** See how user data can be used to automate provisioning steps
- **Provisioners in Terraform:** Understand and implement remote-exec provisioner to execute commands on the EC2 instance and the file provisioner to copy files to instances during provisioning

#### Checklist for This Week

- Extend your Terraform file to include a VPC, subnet, route table, and security group
- Add an AMI and create an EC2 instance, including setting up a key pair
- Use user data to automate Docker installation, start Docker, and run an NGINX container on the instance
- Implement the remote-exec provisioner to replace the entry script on the instance
- Use the file provisioner to copy necessary files to the EC2 instance during deployment

#### Recommendations & Tips

- **Reminder:** Provisioners are helpful but should only be used as a last resort in specific circumstances

### Week 19

#### Weekly Progress Tracker

**Focus:** Infrastructure as Code with Terraform

#### Key Topics for the Week

- **Terraform Modules:** Introduction to Terraform modules, explaining their purpose, structure, and how they help organize infrastructure as code
- **Module Structure:** Set up a module structure within your Terraform project, including modules for a web server and a subnet to improve code reusability and clarity.
- **Module Variables:** Define variables within the module structure to dynamically configure infrastructure components
- **EKS Module & Cluster Creation:** Add an EKS (Elastic Kubernetes Service) module to the project, configure the cluster, and create the necessary infrastructure.
- **NGINX Deployment:** Learn how to deploy NGINX into an EKS cluster

#### Checklist for This Week

- Modularize your Terraform project
- Define and implement module variables for creating instances and other components
- Add VPC and subnet resources to your Terraform module file, including tags
- Configure and deploy an EKS cluster using an EKS module in Terraform
- View your newly created infrastructure in the AWS UI
- Deploy an NGINX server into your EKS cluster



## 4 - DevOps Advanced - Part 1

### Week 20

#### Weekly Progress Tracker

**Focus:** Infrastructure as Code with Terraform

#### Key Topics for the Week

- **Jenkins Configuration for Terraform CI/CD:** Set up Jenkins to manage Terraform infrastructure as part of a continuous integration/continuous deployment (CI/CD) pipeline
- **Deploying Infrastructure via Jenkins:** Deploy the Terraform infrastructure using Jenkins, ensuring smooth integration and automation of the provisioning process.
- **Best Practices for Terraform & State Management:** Learn and apply best practices for working with Terraform, focusing on managing Terraform state files, locking state, versioning, and securing sensitive data.

#### Checklist for This Week

- Add a provision server stage in your `Jenkinsfile`
- Create a key pair, install Terraform, and update the entry script for Docker Compose installation on the Jenkins server
- Add a sleep stage to the Jenkins pipeline to control deployment timing and configure Docker credentials for the deployment stage
- Deploy the infrastructure via Jenkins
- Configure S3 as the backend for Terraform state and verify the state is stored correctly
- Review best practices for managing Terraform and its state

#### Recommendations & Tips

- **Reminder:** Remember to ensure the defined S3 bucket name is globally unique when configuring the remote state



## 4 - DevOps Advanced - Part 2

### Week 21

#### Weekly Progress Tracker

**Focus:** Programming Basics with Python

#### Key Topics for the Week

- **Introduction to Python:** Definition, benefits, use cases, and its importance in DevOps
- **Basic Syntax and Variables:** Use the print() command, perform text and number calculations, and add variables to main.py.
- **Functions and User Input:** How to create a function, add user input, and enhance with if/elif statements and return calculations
- **Error Handling and Loops:** Try/except for error handling and while True loops for continuous input
- **Lists, Sets, and Dictionaries:** Explain lists and sets, compare them, and demonstrate dictionaries for storing key-value pair

#### Checklist for This Week

- Install Python and configure PyCharm for project setup
- Explore print() command, text/number calculations, and variable creation in main.py
- Create a days\_to\_units function and integrate user input with control flow using if/elif
- Add try/except for error handling and implement a while True loop for continuous input
- Explain lists and sets, add a for loop, and compare lists vs. sets
- Demonstrate dictionaries and basic usage of Python data structures in your project

#### Recommendations & Tips

- **Tip:** Use PyCharm's built-in tools to simplify debugging and code management

### Week 22

#### Weekly Progress Tracker

**Focus:** Programming Basics with Python & Automation with Python

#### Key Topics for the Week

- **Object-Oriented Programming (OOP) Overview:** Explain key OOP concepts, such as creating a class for managing user lists, the significance of the self parameter, and how to add post functionality within a class.
- **Working with GitLab API:** Understand how the requests library can be used to complete API requests
- **AWS Automation with Python:** Overview of tasks that can be automated on AWS using Python, such as managing EC2 instances, subnets, snapshots, and more.
- **Boto3 Setup:** Boto3 installation and a deep-dive into the official documentation
- **Snapshot and Volume Management:** See how Python Create a snapshot via Python, remove snapshots using Python, and create an EC2 instance with a backed-up volume attached.
- **Website Monitoring:** How to use python to automate the monitoring of a website

#### Checklist for This Week

- Install and explore boto3, connect to EC2, and automate subnet creation and listing
- Use boto3 to manage EC2 instances, print their states, and add environment tags
- Launch an EC2 instance via AWS UI, create snapshots, and manage snapshots using Python
- Schedule Python commands and integrate Terraform to create infrastructure
- Create EC2 instances with backed-up volumes and manage snapshots programmatically
- Set up a Linode server, install Docker and NGINX, and configure Python to send email alerts and restart the server when it goes down

#### Recommendations & Tips

- **Reminder:** Be careful when scheduling automated commands to avoid unintended state changes in your cloud infrastructure



## 4 - DevOps Advanced - Part 3

### Week 23

#### Weekly Progress Tracker

**Focus:** Configuration Management with Ansible

#### Key Topics for the Week

- **Ansible Overview:** Introduction to Ansible, covering use cases, benefits, modules, playbooks, and the hosts file
- **Control Node Setup:** Explanation of the control node, and demo of its installation and setup for managing remote servers
- **Ansible Modules and Collections:** Understand how Modules and Collections are used in Ansible and a look at Ansible Galaxy
- **Node.js Deployment:** Using Ansible to deploy a Nodejs application
- **Variables:** How to make Ansible Playbooks customizable using variables

#### Checklist for This Week

- Set up a control node and demo its installation for remote server management
- Create a DO droplet, add it to the hosts file with variables, and test connectivity with ansible ping
- Set up an EC2 instance, configure the private key, and update the hosts file; configure authorized\_keys for secure SSH access
- Write an Ansible playbook to configure and start an NGINX server, and explore Ansible documentation, plugins, and collections
- Create a deploy-node.yaml playbook to install Node.js, npm, and deploy a Node.js app, ensuring proper execution with debugging steps
- Add variables to your Ansible Playbook

#### Recommendations & Tips

- **Tip:** Use Ansible Galaxy to find and use reusable playbooks and roles to speed up your deployments

### Week 24

#### Weekly Progress Tracker

**Focus:** Configuration Management with Ansible

#### Key Topics for the Week

- **Nexus Deployment with Ansible:** How to recreate the deployment of Nexus using Ansible - including directory structure and permissions
- **Ansible Default Inventory file:** How to use a Git repository for configuration management of your infrastructure
- **Infrastructure Deployment with Terraform:** Use Terraform to create the infrastructure, install Docker, add the ec2-user to the Docker group, and test the docker pull command
- **Docker and Ansible:** How to run your Docker applications using Ansible
- **Dynamic Inventory for EC2:** How to use Ansible to check the inventory of EC2 instances

#### Checklist for This Week

- Recreate the Nexus deployment steps using Ansible, including directory setup, dependencies, and installation
- Set permissions on the Nexus directory, start the service, and verify the deployment using netstat and ps
- Set up a Git project for inventory file management
- Deploy infrastructure using Terraform, install Docker, add the ec2-user to the Docker group, and test docker pull
- Start Docker containers, log into Docker, and copy Docker Compose files using Terraform's local exec provisioner
- Update the Ansible config with the AWS EC2 plugin and use keyed\_groups to manage EC2 instance inventory



## 4 - DevOps Advanced - Part 4

### Week 25

#### Weekly Progress Tracker

**Focus:** Configuration Management with Ansible & Monitoring with Prometheus

#### Key Topics for the Week

- **Deploying to Kubernetes:** How to use the kubeconfig file in Ansible to deploy a NGINX app to K8s
- **Use Jenkins pipeline to run Ansible:** How to build a Jenkins pipeline that successfully executes an Ansible playbook - including steps for copying files, managing credentials, running scp commands
- **Prometheus and Monitoring:** Explanation of Prometheus use cases and components such as targets, metrics, endpoints, exporters, configuration, alert manager, storage, and querying

#### Checklist for This Week

- Create EKS infrastructure using Terraform and deploy a NGINX app via Ansible using kubeconfig and Python dependencies
- Install Ansible, boto3, and botocore on Jenkins and create EC2 instances
- Build and execute a Jenkinsfile that handles file copying, credentials, scp, and running Ansible playbooks
- Finalize the Jenkins pipeline and deploy services, including Docker using roles
- Learn about Prometheus, its components, and use cases, and set up monitoring for your EKS cluster
- Use eksctl to create a cluster, deploy Prometheus, and a microservices app

### Week 26

#### <sup>17</sup> Weekly Progress Tracker

**Focus:** Monitoring with Prometheus

#### Key Topics for the Week

- **Prometheus Queries and Filters:** Configure queries with filters, and adjust Prometheus configurations
- **Intro to Grafana:** Understand the dashboard and queries using PromQL
- **Prometheus Custom Alerts and Alert Manager Configuration:** How to use custom alerts in Prometheus. Explain the role of receivers and routes in managing alerts and how to use these when creating alerts
- **Third-Party Applications and Exporters:** Explain the use of third-party applications and exporters with Prometheus. How to add additional dashboards to Grafana.
- **Expose Metrics and Configure Custom Monitoring:** How to view custom app metrics in Prometheus and build custom Grafana dashboards for detailed monitoring

#### Checklist for This Week

- Port forward and access the Prometheus and Grafana UIs, configure queries, and observe metrics such as CPU usage
- Create and run alerts in Prometheus, including custom alerts for CPU load and Kubernetes restarts
- Validate custom alerts by running CPU stress tests and configure receivers and routes in the alert manager
- Set up an alert manager email configuration, trigger alerts, and verify functionality
- Deploy Redis exporter with Prometheus, configure Redis alerts, and add Redis dashboards to Grafana
- Deploy a Node.js app, create a ServiceMonitor, and build custom Grafana dashboards for application monitoring

# Need some Guidance?



We know learning DevOps can be extremely challenging. That's why we created the DevOps Bootcamp to help engineers from zero DevOps know-how to become confident Senior DevOps and Cloud Engineers.

If you want to **save yourself time**, have a **guaranteed successful career upgrade** and become a “Certified DevOps Practitioner” with our certification, then check out our intensive program [HERE](#) or copy link: <https://www.techworld-with-nana.com/devops-bootcamp>

# Or Need some Motivation?

Two side-by-side cards. The left card is titled "Mastering DevOps... A software developer's path to upskill" featuring a photo of Kevin Pirner and a diagram showing a path from "Mid-Level" to "Senior" developer. The right card is titled "From Sys Admin to DevOps Engineer: Jose's..." featuring a photo of Jose Maria Briones and a diagram showing a path from "Software Developer" to "Systems Administrator" to "DevOps Engineer".

We successfully educated 10,000s+ of engineers. Read about their inspiring DevOps journeys from different backgrounds, like developer, sys admin etc. Read [HERE](#) or copy link: <https://www.techworld-with-nana.com/success-stories>

If they can do it, YOU can too! 🤓