Modern Cloud Engineer in 5 Months

A Project-Based Roadmap to Azure, DevOps & Kubernetes

Introduction

Most cloud roadmaps are broken. They give you a list of 100 technologies to memorize, you spend six months watching videos, and at the end, you still can't build a production environment.

This is the **Builder's Roadmap**.

What you'll achieve

- [ ] Five months of structured learning
- [ ] Twenty weeks of hands-on projects
- [ ] Fifteen production-grade portfolio pieces
- [ ] A GitHub portfolio that proves your skills

Core principle: BUILD. DOCUMENT. SHARE.

By the end of this roadmap, you won't just know Azure - you'll have a portfolio that proves it.

TL;DR - The 5-Month Journey

Month 1: Foundation - Master Azure fundamentals (VMs, Storage, Networking). Build the Cloud Resume Challenge with C# Functions and CosmosDB.

Month 2: Infrastructure as Code - Learn Terraform and automate everything. Build CI/CD pipelines with GitHub Actions and implement drift detection.

Month 3: Containers & Kubernetes - Dockerize applications and deploy to AKS. Build a microservices e-commerce platform with multiple services, Helm, and Ingress.

Month 4: Production Engineering - Implement security (Key Vault, Managed Identities), observability (Application Insights, Grafana), and reduce costs by 40%.

Month 5: Polish & Job Hunt - Document your portfolio, optimize your resume, pass AZ-104, and prepare for interviews with real project experience.

Result: 15 production-grade projects, a GitHub portfolio, and the skills to land Cloud Engineer, DevOps Engineer, or SRE roles.

Progress Tracker

- [ ] Month 1: Foundation
  - [ ] Week 1: Core Azure Services
  - [ ] Week 2: Networking Deep Dive
  - [ ] Week 3-4: Capstone — Cloud Resume Challenge

- [ ] Month 2: Infrastructure as Code
  - [ ] Week 5-6: Terraform Fundamentals
  - [ ] Week 7-8: CI/CD with GitHub Actions
- [ ] Month 3: Containers & Kubernetes
  - [ ] Week 9: Docker Fundamentals
  - [ ] Week 10-11: Azure Kubernetes Service
  - [ ] Week 12: Capstone — Microservices E-Commerce
- [ ] Month 4: Production Engineering
  - [ ] Week 13: Security Hardening
  - [ ] Week 14-15: Observability
  - [ ] Week 16: FinOps Challenge
- [ ] Month 5: Polish & Job Hunting
  - [ ] Week 17-18: Portfolio Documentation
  - [ ] Week 19: Resume & LinkedIn Optimization
  - [ ] Week 20: Certification & Interview Prep

Understanding the Role Landscape

Cloud Engineer vs. DevOps vs. SRE

In 2025, the lines between these roles have converged:

- **Cloud Engineers** need to know Terraform
- **DevOps Engineers** need to know Azure Networking
- **SREs** need to know Observability

This roadmap covers

- 60% Cloud Architecture
- 40% DevOps Practices

The foundation is identical across all three roles: **Automated Infrastructure**.

Month 1: The Foundation (Weeks 1-4)

Overview

Month 1 is about breaking things and understanding the "Big Three": Networking, Storage, and Compute.

Week 1: Core Azure Services

Focus: Deploying VMs and Static Sites

Key Tasks

- [ ] Create your first Azure Virtual Machine
- [ ] Deploy a static website to Azure Blob Storage
- [ ] Configure basic storage accounts
- [ ] Understand Azure Resource Groups

Deliverable: A running VM and a live static site

Week 2: Networking Deep Dive

Focus: VNets, Subnets, Network Security Groups

Key Concepts

- [ ] Virtual Networks (VNets)
- [ ] Subnets and IP addressing
- [ ] Network Security Groups (NSGs)
- [ ] Hub-and-Spoke topology

Critical: Do not skip networking. If you don't understand how a packet moves from a public IP to a private subnet, you cannot secure a cloud environment.

Deliverable: A documented Hub-and-Spoke network architecture

Week 3-4: Month 1 Capstone Project

Project: Cloud Resume Challenge

Components

1. [ ] **Frontend:** Static site hosted in Azure Blob Storage
2. [ ] **CDN:** Azure CDN fronting the static site
3. [ ] **Backend:** C# Azure Function API
4. [ ] **Database:** CosmosDB for visitor count
5. [ ] **Security:** Proper authentication and CORS configuration

Sample Azure Function Code

```
[Function("UpdateVisitorCount")]
public static async Task<HttpResponseData> Run(
    [HttpTrigger(AuthorizationLevel.Function, "get", "post")] HttpRequestD
    [CosmosDBInput(databaseName: "ResumeDb", containerName: "Counter",
                   Connection = "CosmosDbConnection")] Counter document)
{
    document.Count += 1;
    // Save logic here
    return response;
}

**Why this matters:** This project forces you to connect frontend, backend

---

## Month 2: Infrastructure as Code (Weeks 5-8)

### Overview

Stop clicking in the portal. Month 2 is about automating everything with T

#### Why Terraform?

- [ ] Cloud-agnostic
- [ ] Industry standard
- [ ] Version-controlled infrastructure

---

### Week 5-6: Terraform Fundamentals

#### Key Concepts

- [ ] Terraform providers and resources
- [ ] State management
- [ ] Variables and outputs
- [ ] Module structure
```

Sample Terraform Resource

```hcl
resource "azurerm_kubernetes_cluster" "aks" {
  name                = "aks-prod-01"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  dns_prefix          = "aks-prod"

  default_node_pool {
    name       = "default"
    node_count = 3
    vm_size    = "Standard_D2_v2"
  }
}
```

Deliverable: Build a module library for networking, compute, and stora

---

Week 7-8: CI/CD with GitHub Actions

Focus: Automated Infrastructure Deployment

Key Tasks

- [ ] Create GitHub Actions workflows for Terraform
- [ ] Implement terraform plan and apply automation
- [ ] Build drift detection pipelines
- [ ] Set up automated testing

Advanced Challenge: Build a pipeline that detects infrastructure drift

Deliverable: A fully automated infrastructure deployment pipeline with

---

Month 3: Containers & Kubernetes (Weeks 9-12)

Warning

Month 3 is the filter. This is where most people quit. We are tackling Doc

---

Week 9: Docker Fundamentals

Key Tasks

- [ ] Containerize a multi-tier application
- [ ] Create optimized Dockerfiles
- [ ] Use Docker Compose for local development
- [ ] Push images to Azure Container Registry

Week 10-11: Azure Kubernetes Service

Focus: AKS Networking, Ingress, and Helm

Key Concepts

- [ ] Kubernetes architecture and components
- [ ] AKS networking models
- [ ] Ingress controllers (nginx/traefik)
- [ ] Helm charts for deployment
- [ ] Service mesh basics

Critical Commands

```bash
kubectl get pods -n production
kubectl logs -f <pod-name>
kubectl describe service <service-name>
```

Week 12: Month 3 Capstone Project

Project: Microservices E-Commerce Platform

Architecture Components

1. [ ] **Auth Service** - User authentication and authorization
2. [ ] **Product Service** - Product catalog and inventory
3. [ ] **Cart Service** - Shopping cart management
4. [ ] **API Gateway** - Ingress routing

Technical Requirements

- [ ] All services in separate containers
- [ ] Communication via Service Mesh or Ingress
- [ ] Deployed via Helm charts
- [ ] Proper health checks and readiness probes

Success Criteria: If you can build this, document it, and explain how the Ingress routes traffic to the correct pod - you are hireable.

Month 4: Production Engineering (Weeks 13-16)

Philosophy

A Junior engineer builds it. A Senior engineer keeps it running.

Month 4 focuses on: **Security, Observability, and Cost (FinOps)**

Week 13: Security Hardening

Key Tasks

- [ ] Move all secrets to Azure Key Vault
- [ ] Implement Azure Policy for governance
- [ ] Configure Managed Identities
- [ ] Set up Azure AD authentication
- [ ] Implement network security rules

Deliverable: Zero hardcoded secrets in your codebase

Week 14-15: Observability

Focus: Application Insights and Log Analytics

Key Tasks

- [ ] Set up Application Insights for your services

- [ ] Configure Log Analytics workspace
- [ ] Write KQL queries to find failed requests
- [ ] Create dashboards in Grafana
- [ ] Set up alerts for critical metrics

Sample KQL Query

```
requests
| where success == false
| summarize count() by resultCode, bin(timestamp, 1h)
| order by timestamp desc
```

Week 16: FinOps Challenge

Project: Cost Optimization

Goal: Reduce your Month 3 project's running cost by 40%

Strategies

- [ ] Implement spot instances for non-critical workloads
- [ ] Rightsize VMs and node pools
- [ ] Use Azure Cost Management tools
- [ ] Implement auto-scaling policies
- [ ] Clean up unused resources

Why this matters: "Reduced infrastructure costs by 40%" is a resume bullet point that recruiters love.

Month 5: Polish & Job Hunting (Weeks 17-20)

Week 17-18: Portfolio Documentation

Key Deliverables

- [ ] README files for all projects with architecture diagrams
- [ ] Blog posts explaining your technical decisions
- [ ] Video walkthroughs (optional)
- [ ] Clean, commented code

Week 19: Resume & LinkedIn Optimization

Bad Resume Bullet

- [ ] "Watched a course on Terraform"

Good Resume Bullet

- [ ] "Built and deployed a microservices platform on AKS handling simulated traffic, reduced infrastructure costs by 40% using spot instances and rightsizing"

Profile Optimization

- [ ] Update LinkedIn with project links
- [ ] Share your GitHub portfolio

- [ ] Write technical posts about your journey

Week 20: Certification & Interview Prep

Recommended Certification: AZ-104 (Azure Administrator)

Strategy: Take the certification AFTER you've done the projects. The exam is easy when you've actually built the resources.

Interview Preparation

- [ ] Practice explaining your projects
- [ ] Prepare for architecture whiteboard sessions
- [ ] Review common Azure interview questions
- [ ] Mock interviews with peers

Time Commitment

Expected: 20-30 hours per week

Reality Check

- This will be hard
- You will get stuck on YAML indentation errors
- You will struggle with networking concepts
- You will want to quit during Month 3

But: If you commit to the Project-First methodology, in 5 months you won't be looking for tutorials - you'll be looking for contracts to sign.

Tech Stack Summary

Core Technologies

- **Cloud Platform:** Microsoft Azure
- **IaC:** Terraform
- **Containers:** Docker, Kubernetes (AKS)
- **CI/CD:** GitHub Actions
- **Languages:** C#, Bash, YAML
- **Observability:** Application Insights, Log Analytics, Grafana
- **Security:** Azure Key Vault, Azure AD, Managed Identities

Tools You'll Master

- Azure CLI
- kubectl
- Terraform CLI
- Docker CLI
- Git & GitHub
- VS Code
- Helm

Getting Started

Action Steps

1. [ ] Create your Azure free account
2. [ ] Set up your development environment (VS Code, Azure CLI, Docker)
3. [ ] Fork/clone the project templates (link your GitHub)
4. [ ] Start Week 1 today

> Don't save this for "later." Open the terminal. Start building.

Additional Resources List

- Official Azure Documentation
- Terraform Azure Provider Docs
- Kubernetes Documentation
- Your project GitHub repositories
- Community support channels

Remember: BUILD. DOCUMENT. SHARE.

Your portfolio is your proof. Make it count.