**10 Haftalık DevOps Başlangıç Programı (okul temposuna uygun)**

Haftalık ~4–6 saat ayırman yeter. “Zor bir hafta” gelirse görevleri ikiye böl ve sırayı bozma.

**Hafta 1 — AWS temelleri + bulut kavramları**

**Hedef:** Bulut servis modelleri (IaaS/PaaS), bölgeler/AZ’ler, IAM, EC2, S3, VPC’nin ne olduğunu kavramak.  
**Görev:** AWS’de ücretsiz katmanla hesap aç, IAM’de *least privilege* bir kullanıcı oluştur.  
**Kaynaklar:**

* AWS Cloud Practitioner Essentials (ücretsiz dijital eğitim). [Amazon Web Services, Inc.+1](https://aws.amazon.com/training/digital/?utm_source=chatgpt.com)[skillbuilder.aws](https://skillbuilder.aws/learn/94T2BEN85A/aws-cloud-practitioner-essentials/8D79F3AVR7?utm_source=chatgpt.com)
* AWS Well-Architected Framework (giriş ve 6 sütun). [docs.aws.amazon.com](https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html?utm_source=chatgpt.com)

**Hafta 2 — Docker’a giriş**

**Hedef:** Container nedir, image–container–registry ilişkisi.  
**Görev:** Docker Desktop’ı kur; bir “Hello, World” web servisini container olarak çalıştır, Dockerfile yaz, image’i local registry’ye it.  
**Kaynaklar:** Docker “Get Started” ve resmi dokümanlar. [Docker Documentation+1](https://docs.docker.com/get-started/?utm_source=chatgpt.com)[Docker](https://www.docker.com/get-started/?utm_source=chatgpt.com)

**Hafta 3 — Docker ile çok-container’lı geliştirme**

**Hedef:** docker-compose, volume, network, healthcheck.  
**Görev:** Basit bir “web + db (PostgreSQL)” stack’i compose ile ayağa kaldır, restart policy ve healthcheck ekle.  
**Kaynaklar:** Docker “Get Started” + topluluk ders notu. [Docker Documentation](https://docs.docker.com/get-started/?utm_source=chatgpt.com)[A Docker Tutorial for Beginners](https://docker-curriculum.com/?utm_source=chatgpt.com)

**Hafta 4 — GitHub Actions ile CI**

**Hedef:** “Kod push → test/build” pipeline’ı.  
**Görev:** Bir repo aç; PR açılınca testleri çalıştıran workflow yaz (.github/workflows/ci.yml).  
**Kaynaklar:** GitHub Actions Quickstart & Docs. [GitHub Docs+2GitHub Docs+2](https://docs.github.com/actions/quickstart?utm_source=chatgpt.com)

**Hafta 5 — CD ve temel AWS deploy**

**Hedef:** CI çıktısını bir ortama otomatik dağıtmak.  
**Görev:** Docker image’i GitHub Actions ile build edip registry’e pushla; EC2’ye SSH ile basit deploy (veya bir küçük PaaS’a).  
**Kaynaklar:** GitHub Actions genel başlangıç. [GitHub Docs](https://docs.github.com/actions?utm_source=chatgpt.com)[GitHub](https://github.com/features/actions/getting-started?utm_source=chatgpt.com)

**Hafta 6 — Kubernetes’e giriş**

**Hedef:** Pod, Deployment, Service kavramları.  
**Görev:** Minikube kur; örnek web uygulamasını Deployment + Service ile expose et, scale et (replica sayısını değiştir).  
**Kaynaklar:** Kubernetes “Basics” interaktif eğitim ve resmi tutorial’lar. [Kubernetes+1](https://kubernetes.io/docs/tutorials/kubernetes-basics/?utm_source=chatgpt.com)

**Hafta 7 — K8s pratikleri**

**Hedef:** ConfigMap, Secret, rolling update / rollback.  
**Görev:** Uygulamanın konfigürasyonunu ConfigMap/Secret’la ayır; rolling update yap ve gerekirse rollback et.  
**Kaynaklar:** Kubernetes tutorials sayfası. [Kubernetes](https://kubernetes.io/docs/tutorials/?utm_source=chatgpt.com)

**Hafta 8 — IaC: Terraform’a giriş**

**Hedef:** Altyapıyı kodla tanımlamak.  
**Görev:** Terraform ile AWS’de küçük bir kaynak (ör. S3 bucket veya EC2) oluştur, değiştir, sil; state mantığını anla.  
**Kaynaklar:** HashiCorp Terraform resmi “Get Started”/Tutorials. [HashiCorp Developer+1](https://developer.hashicorp.com/terraform?utm_source=chatgpt.com)

**Hafta 9 — Uygulamalı Terraform + 12-Factor**

**Hedef:** Modüler Terraform, değişkenler/output’lar; 12-Factor App prensipleriyle uygulamayı buluta uygun hale getirme.  
**Görev:** Terraform modülleriyle basit bir VPC + EC2 tanımla; uygulama konfigürasyonunu environment değişkenlerine taşı.  
**Kaynaklar:** HashiCorp tutorials + 12-Factor App. [HashiCorp Developer](https://developer.hashicorp.com/terraform/tutorials?utm_source=chatgpt.com)[12factor.net](https://www.12factor.net/?utm_source=chatgpt.com)

**Hafta 10 — Mini Capstone (uçtan uca)**

**Hedef/Çıktı:**

* Dockerized web app → GitHub Actions ile CI
* Image → registry
* Terraform ile altyapı (ör. S3 + EC2 ya da küçük EKS/Minikube)
* Otomatik deploy (CD) ve temel gözlemleme (logs).  
  **Kalite rehberi:** AWS Well-Architected kontrol listesi (özellikle güvenlik ve maliyet sütunları). [docs.aws.amazon.com](https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html?utm_source=chatgpt.com)

**Ek Okuma ve “mindset”**

* **12‑Factor App** (buluta uygun uygulama geliştirme prensipleri). [12factor.net](https://www.12factor.net/?utm_source=chatgpt.com)
* **Google SRE kitapları** (gözlemlenebilirlik, SLO/SLA, incident kültürü). [sre.google+1](https://sre.google/?utm_source=chatgpt.com)

**Cloud Security’yi unutmadan (siber güvenlik köprüsü)**

* AWS’de IAM, şifre yönetimi, şifreli depolama/S3, network segmentation ve **Well‑Architected Security Pillar** notlarıyla ilerle. [docs.aws.amazon.com](https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html?utm_source=chatgpt.com)
* GitHub Actions “secrets”, branch koruması, minimum izinli PAT/oidc. [GitHub Docs](https://docs.github.com/actions?utm_source=chatgpt.com)

**Program bitince elinde olacaklar**

* **Portföy:** Dockerfile + Compose, bir **CI** workflow’u, basit **CD**, Terraform IaC kodu, kısa bir **runbook/README**.
* **Pratik:** K8s’te scale/update, Terraform ile plan/apply, CI’de test/build, temel AWS kaynak yönetimi.
* **Bakış:** 12‑Factor ve Well‑Architected ile “doğru” mimari refleksi.