End-to-End DevOps POC Plan (Structured from Basic to Advanced)

Phase 1: Container Image Preparation

Task 1: Write Optimal Dockerfiles for Jenkins, Nexus, and SonarQube

Objective:
Create secure, minimal, and production-ready Docker images, then push them to Azure Container Registry (ACR).

Requirements:

- Use official base images
- Apply Docker best practices:
    - Non-root user
    - Lightweight layers
    - Remove unnecessary files
- Add:
    - Jenkins plugins
    - Nexus default repositories
    - SonarQube tuning

Deliverables:

- Dockerfile for each service
- `README.md` with:

- Build instructions
- ACR push steps
- Security & optimization notes

Phase 2: Infrastructure Setup Using Terraform

Task 2: Prepare Terraform Modules for Multi-AZ AKS Setup

Objective:
Create reusable Terraform modules to provision a scalable AKS environment in Azure.

Modules:

- VNET, Subnets, NSGs
- Route Tables
- NAT Gateway
- Application Gateway
- Internal/External Load Balancer
- VMSS
- AKS (Multi-AZ, 3 nodes)

Deliverables:

- Modular Terraform structure
- `README.md` with:
  - Module usage
  - Inputs/Outputs

- Root module example for full stack deployment

Task 3: Setup a 3-Node Multi-Zone AKS Cluster Using Terraform

Objective:
Use the Terraform modules to provision a high availability AKS cluster.

Deliverables:

- Terraform configuration for AKS deployment
- Verified 3-node, multi-zone AKS cluster

Phase 3: Application Deployment via Helm or YAML

Task 4: Develop Helm Charts for Argo CD, Jenkins, Nexus, SonarQube

Objective:
Build modular, parameterized Helm charts for core DevOps tools.

Requirements:

- Use `values.yaml` for dynamic config
- Include:
    - ConfigMaps/Secrets
    - RBAC
    - Ingress with TLS
- Support external DBs or PVCs

Deliverables:

- Helm charts for each application

- `README.md` with:
    - Install & upgrade steps
    - Configuration guide

Task 5: Deploy Jenkins and Nexus Using Helm/Manifests

Objective:
Use Kubernetes manifests or Helm charts to deploy apps with custom Docker images.

Deliverables:

- YAML or Helm configurations
- `README.md` for deployment steps

Phase 4: CI/CD Automation

Task 6: Setup Jenkins Pipeline for Hello World Spring Boot App

Objective:
Automate CI for a Spring Boot app: build, test, and publish artifacts to Nexus.

Deliverables:

- Jenkinsfile
- Sample Spring Boot repo or snippet
- Nexus repository configuration

Phase 5: Image Automation with Packer

Task 7: Write Packer Template for Windows 10 Image

Objective:
Build a Windows 10 golden image with apps and updates using Packer and PowerShell.

Configuration:

- Install updates
- Add sample apps
- PowerShell-based provisioning

Deliverables:

- `windows.pkr.hcl` or JSON
- PowerShell scripts
- `README.md` with:
    - Packer usage
    - WinRM setup
    - Image verification

Phase 6: Observability & Logging

Task 8: Script to Upload Jenkins Build Logs to Azure Blob

Objective:
Backup Jenkins logs to Azure for long-term storage and analysis.

Script Features:

- Python or Shell-based
- Jenkins API authentication

- Azure Blob upload

Deliverables:

- Script file
- `requirements.txt` (if Python)
- `README.md` with usage and setup instructions

Final Structure Overview

Phase	Task	Tools	Output
1	Dockerfiles for Jenkins/Nexus/SonarQube	Docker, ACR	Secure custom images
2	Terraform Infra Modules	Terraform	Reusable cloud infra
3	AKS Cluster Deployment	Terraform + kubectl	3-node AKS
4	Helm Charts for DevOps Tools	Helm	Deployable charts
5	App Deployments with Helm/YAML	kubectl + Helm	Jenkins/Nexus running
6	CI Pipeline for Spring Boot	Jenkins + Maven + Nexus	CI/CD pipeline
7	Windows Image via Packer	Packer + PowerShell	Golden image
8	Jenkins Logs to Azure Blob	Python, Azure SDK	Archived logs
