myHomelab

K19G

2026-02-28

Table of contents

Preface

Welcome to this book!

Structure

This book is organized into categories and subcategories to help you navigate the content effectively.

How to Use This Book

You can read this book in various formats:

- Online HTML version
- Downloadable PDF
- EPUB for e-readers

Prerequisites

List any prerequisites or requirements here.

Acknowledgments

Add acknowledgments here.

Foundations

Chapter 01: Vision, Scope, and Success Criteria

Word target: 2,500
Primary deliverable: Homelab charter and measurable outcomes
Key diagrams: Mission pyramid, constraints map

Learning Goals

- Define scope boundaries for a learning lab versus production-like workloads.
- Convert goals into measurable KPIs (availability, recovery time, cost, time-to-deploy).
- Create a risk register for hardware failure, misconfiguration, and internet dependency.

MVP Lab Worksheet

- Objective: Create a one-page homelab charter.
- Starting state: Existing Proxmox laptop and Azure Ubuntu VPS.
- Steps:
  1. List top 5 services you want to run.
  2. Set SLO-lite target for each service.
  3. Define acceptable monthly operating cost.
- Evidence: Saved charter doc and KPI table.
- Exit criteria: Charter reviewed and versioned in repo.
- Rollback: N/A (documentation-only).

Advanced Lab Worksheet

- Objective: Build a 12-month capability roadmap.
- Starting state: MVP charter complete.
- Steps:
  1. Define quarterly milestones.
  2. Add risk score and mitigation for each milestone.
  3. Add “kill criteria” for initiatives that add complexity without value.
- Evidence: Roadmap + risk matrix.
- Exit criteria: Signed ADR for roadmap priorities.
- Rollback: Revert roadmap to previous version tag.

ℹ Author Prompt

Add a short personal story: why this homelab exists and what failure taught you most.

Chapter 02: Ubuntu-First Reference Architectures

Word target: 3,000
Primary deliverable: Baseline architecture decision record (ADR-001)
Key diagrams: Single-node, hybrid edge-cloud, future multi-node topology

Learning Goals

- Compare architecture patterns by resilience, cost, and operational overhead.
- Select a baseline architecture aligned to current hardware limits.
- Define an evolution path that avoids vendor and hardware lock-in.

MVP Lab Worksheet

- Objective: Select baseline architecture.
- Starting state: Charter complete.
- Steps:
  1. Draw three candidate topologies.
  2. Score each for cost, complexity, and uptime.
  3. Choose one and publish ADR-001.
- Evidence: Diagram set + ADR file.
- Exit criteria: One approved baseline architecture.
- Rollback: Reopen ADR and select runner-up design.

Advanced Lab Worksheet

- Objective: Define migration plan to multi-node future.
- Starting state: Baseline architecture approved.
- Steps:
  1. Identify bottleneck components.
  2. Define triggers for scaling out.
  3. Create staged migration timeline.
- Evidence: Migration roadmap with prerequisites.
- Exit criteria: Executable transition plan.
- Rollback: Freeze to baseline and defer scale-out.

ℹ Portability Placeholder

Map “compute”, “storage”, and “network” abstractions to interchangeable implementations.

Chapter 03: Ubuntu Base Image and OS Standards

Word target: 3,500
Primary deliverable: Ubuntu baseline profile and cloud-init template
Key diagrams: Image lifecycle pipeline

Learning Goals

- Standardize Ubuntu 24.04 server baseline configuration.
- Use cloud-init for reproducible first-boot provisioning.
- Establish OS hardening and patching defaults.

MVP Lab Worksheet

- Objective: Build Ubuntu baseline checklist.
- Starting state: Fresh Ubuntu VM.
- Steps:
    1. Configure users, SSH keys, timezone, and NTP.
    2. Enable unattended upgrades and essential packages.
    3. Capture baseline cloud-init file.
- Evidence: Baseline checklist + cloud-init YAML.
- Exit criteria: Reproducible VM bootstrap in under 10 minutes.
- Rollback: Revert to clean snapshot and rerun bootstrap.

Advanced Lab Worksheet

- Objective: Versioned image pipeline.
- Starting state: MVP bootstrap validated.
- Steps:
    1. Build golden image version tags.
    2. Add automated validation tests.
    3. Add deprecation policy for old images.
- Evidence: Image catalog with validation status.
- Exit criteria: Repeatable image release workflow.
- Rollback: Promote prior image version.

💡 Ubuntu Focus

Explicitly document: `cloud-init`, `systemd`, `netplan`, `ufw`, `apt`, `unattended-upgrades`.

Chapter 04: Networking Fundamentals (`netplan`-First)

Word target: 3,500
Primary deliverable: IP plan, DNS plan, and routing policy
Key diagrams: Subnet map, DNS flow, ingress/egress boundaries

Learning Goals

- Build a clean internal addressing strategy.
- Apply Ubuntu `netplan` patterns for static and DHCP deployments.
- Validate routing, DNS, and firewall behavior with repeatable tests.

MVP Lab Worksheet

- Objective: Implement base network plan.
- Starting state: One Proxmox VM + Azure VPS.
- Steps:
  1. Define subnet/IP/hostname conventions.
  2. Configure netplan on VM and VPS.
  3. Validate DNS resolution and connectivity paths.
- Evidence: Netplan configs + test logs.
- Exit criteria: Stable L3 and name resolution.
- Rollback: Restore previous netplan file and apply.

Advanced Lab Worksheet

- Objective: Add segmented network zones.
- Starting state: Base network operational.
- Steps:
  1. Define trusted, app, and management segments.
  2. Add policy restrictions between zones.
  3. Run connectivity and denied-path tests.
- Evidence: Policy matrix + test evidence.
- Exit criteria: Segmentation enforced and documented.
- Rollback: Return to flat network policy.

⚠ Author Gap Check

Add a section explaining common netplan failure recovery from console access.

Chapter 05: Identity, Authentication, and Secrets

Word target: 3,500
Primary deliverable: Access control baseline and secret rotation playbook
Key diagrams: Trust boundary and credential flow

Learning Goals

- Enforce key-based access and least privilege.
- Design identity approach that can evolve from local-only to centralized auth.
- Implement secret storage and rotation procedures.

MVP Lab Worksheet

- Objective: Enforce secure SSH baseline.
- Starting state: Ubuntu nodes reachable.
- Steps:
  1. Disable password SSH auth.
  2. Configure least-privilege sudo roles.
  3. Store service credentials in a secrets workflow.
- Evidence: SSH config diff + access tests.
- Exit criteria: No password-based remote login paths remain.
- Rollback: Emergency break-glass console procedure documented.

Advanced Lab Worksheet

- Objective: Credential rotation without downtime.
- Starting state: Services with managed secrets.
- Steps:
  1. Rotate one DB/app credential pair.
  2. Validate service continuity.
  3. Record rotation cadence and ownership.
- Evidence: Rotation log + uptime proof.
- Exit criteria: Rotation runbook validated.
- Rollback: Revert to previous secret version.

ℹ Portability Placeholder

Document identity provider-neutral patterns before naming specific tooling.

Platform And Hybrid

Chapter 06: Runtime Choices (VM, Container, `systemd` Service)

Word target: 3,500
Primary deliverable: Workload placement policy
Key diagrams: Runtime decision tree

Learning Goals

- Select runtime model by workload characteristics and risk profile.
- Keep host operations and application operations cleanly separated.
- Define reproducible deployment contracts across runtimes.

MVP Lab Worksheet

- Objective: Deploy one service in chosen runtime.
- Starting state: Ubuntu node prepared.
- Steps:

  1. Select service and runtime based on policy.
  2. Deploy with health check.
  3. Verify restart/recovery behavior.

- Evidence: Service unit/compose file + health output.
- Exit criteria: Service survives restart and reboot.
- Rollback: Restore prior service method.

Advanced Lab Worksheet

- Objective: Runtime comparison.
- Starting state: MVP deployment complete.
- Steps:

  1. Run same app via two runtime models.
  2. Compare performance, operability, and failure handling.
  3. Update placement policy.

- Evidence: Benchmark + operational comparison matrix.
- Exit criteria: Policy includes measurable thresholds.
- Rollback: Standardize on baseline runtime.

ℹ Author Prompt

Include one anti-pattern case where containers were the wrong choice.

Chapter 07: Storage Design and Data Durability

Word target: 3,200
Primary deliverable: Data classification and durability strategy
Key diagrams: Data criticality matrix, backup flow

Learning Goals

- Classify service data by criticality and retention requirements.
- Map storage classes to backup and restore expectations.
- Avoid hidden state in unmanaged paths.

MVP Lab Worksheet

- Objective: Classify all service data.
- Starting state: Core services listed.
- Steps:
  1. Mark data as ephemeral, important, or critical.
  2. Assign storage path and backup policy per class.
  3. Snapshot one critical workload.
- Evidence: Data inventory + snapshot logs.
- Exit criteria: Every service mapped to a data policy.
- Rollback: Restore original storage mapping doc from git.

Advanced Lab Worksheet

- Objective: Restore validation by class.
- Starting state: Data policy defined.
- Steps:
  1. Restore one item from each data class.
  2. Measure restore time vs targets.
  3. Tune retention and replication.
- Evidence: Restore timing sheet.
- Exit criteria: RPO/RTO targets verified.
- Rollback: Revert retention changes.

Ubuntu Focus

Document filesystem and mount conventions used on Ubuntu nodes.

Chapter 08: Proxmox Lifecycle for Ubuntu VMs

Word target: 3,600
Primary deliverable: Ubuntu template lifecycle on Proxmox
Key diagrams: Template patch and release workflow

Learning Goals

- Build and maintain Ubuntu VM templates in Proxmox.
- Use cloud-init integration for fast provisioning.
- Version and retire templates safely.

MVP Lab Worksheet

- Objective: Create first golden Ubuntu template.
- Starting state: Proxmox host available.
- Steps:
    1. Install Ubuntu and apply baseline.
    2. Convert VM to template with cloud-init support.
    3. Clone and validate first consumer VM.
- Evidence: Template ID + clone validation checklist.
- Exit criteria: New VM from template in <10 minutes.
- Rollback: Revert to previous template snapshot.

Advanced Lab Worksheet

- Objective: Template patch pipeline.
- Starting state: Existing template lifecycle.
- Steps:
    1. Patch template and run smoke tests.
    2. Roll out to staging VMs.
    3. Promote only after validation gates pass.
- Evidence: Release notes + test outcomes.
- Exit criteria: Controlled template promotion workflow.
- Rollback: Reassign deployments to previous template version.

Portability Placeholder

Map Proxmox template mechanics to cloud image workflows.

Chapter 09: Azure Ubuntu VPS Patterns

Word target: 3,400
Primary deliverable: Reproducible Azure Ubuntu baseline deployment
Key diagrams: Cloud boundary and dependency map

Learning Goals

- Apply same Ubuntu baseline policy in cloud environment.
- Isolate cloud-provider specifics from operating model.
- Validate parity between local and cloud nodes.

MVP Lab Worksheet

- Objective: Standardize one Azure Ubuntu VPS.
- Starting state: Existing Azure Ubuntu host.
- Steps:

  1. Apply Ubuntu baseline profile.
  2. Add host inventory and role tags.
  3. Validate parity checklist against Proxmox VM.

- Evidence: Parity report.
- Exit criteria: Baseline parity >90% across environments.
- Rollback: Revert config using saved baseline snapshot.

Advanced Lab Worksheet

- Objective: IaC-driven cloud provisioning.
- Starting state: Manually configured VPS.
- Steps:

  1. Codify VPS provisioning.
  2. Integrate post-provision configuration.
  3. Run destroy/recreate test safely.

- Evidence: IaC repo and apply logs.
- Exit criteria: Reprovisioning from code only.
- Rollback: Restore known-good VM image.

⚠ Author Gap Check

Add cloud cost and quota pitfalls unique to VPS onboarding.

Chapter 10: Hybrid Connectivity (Edge + Azure)

Word target: 3,800
Primary deliverable: Secure private interconnect
Key diagrams: Overlay network topology and failover paths

Learning Goals

- Connect edge and cloud workloads securely.
- Define traffic classes and trust boundaries.
- Test failure behavior and recovery workflows.

MVP Lab Worksheet

- Objective: Build private link between Proxmox side and Azure side.
- Starting state: Nodes independently reachable.
- Steps:

    1. Deploy encrypted mesh/VPN.
    2. Expose one private service across sites.
    3. Verify path and latency baseline.

- Evidence: Connectivity test and route table output.
- Exit criteria: Private access works without public exposure.
- Rollback: Disable tunnel and restore direct local access.

Advanced Lab Worksheet

- Objective: Validate failover and recovery.
- Starting state: Hybrid link active.
- Steps:

    1. Simulate primary path outage.
    2. Confirm degraded-mode behavior.
    3. Restore and measure recovery time.

- Evidence: Failure drill timeline.
- Exit criteria: Recovery within defined threshold.
- Rollback: Return to baseline single-path routing.

ℹ Author Prompt

Document one real outage scenario and what changed after it.

Automation And Operations

Chapter 11: DNS, PKI, and Ingress

Word target: 3,500
Primary deliverable: Naming and TLS operating model
Key diagrams: Split DNS and certificate lifecycle

Learning Goals

- Design internal/external DNS strategy.
- Automate TLS issuance and renewal.
- Standardize ingress/reverse proxy patterns.

MVP Lab Worksheet

- Objective: Publish one internal TLS service.
- Starting state: DNS authority selected.
- Steps:

  1. Configure internal domain and records.
  2. Issue certificate for service endpoint.
  3. Validate HTTPS and cert renewal path.

- Evidence: DNS records + TLS test output.
- Exit criteria: Service reachable with valid cert chain.
- Rollback: Revert to previous ingress config.

Advanced Lab Worksheet

- Objective: Dual ingress model.
- Starting state: Internal ingress running.
- Steps:

  1. Add external endpoint with restricted exposure.
  2. Enforce authentication and rate controls.
  3. Test cert renewal and fail-safe behavior.

- Evidence: Security test checklist.
- Exit criteria: Internal and external ingress policies verified.
- Rollback: Disable external path and retain internal access.

Ubuntu Focus

Note where certificates and service configs live on Ubuntu filesystem.

Chapter 12: Infrastructure as Code and Configuration Management

Word target: 3,600
Primary deliverable: Git-managed infra and config workflows
Key diagrams: GitOps reconciliation loop

Learning Goals

- Separate provisioning from configuration responsibilities.
- Build idempotent, reviewable automation.
- Detect and remediate configuration drift.

MVP Lab Worksheet

- Objective: Provision one Ubuntu node from code.
- Starting state: Manual workflows in place.
- Steps:

  1. Create repo structure for IaC + config.
  2. Provision target node via declarative workflow.
  3. Run idempotency check.

- Evidence: Pipeline/apply logs.
- Exit criteria: Repeatable no-drift apply.
- Rollback: Reapply previous stable commit.

Advanced Lab Worksheet

- Objective: Drift detection pipeline.
- Starting state: IaC baseline active.
- Steps:

  1. Introduce controlled drift.
  2. Detect via automated checks.
  3. Reconcile and document incident.

- Evidence: Drift alert + reconciliation evidence.
- Exit criteria: Drift closed without manual snowflake edits.
- Rollback: Freeze automation and restore from stable tag.

ℹ Portability Placeholder

Ensure provider-specific modules are isolated behind common interfaces.

Chapter 13: Observability and SLO-lite Operations

Word target: 3,200
Primary deliverable: Baseline telemetry stack and alert runbook
Key diagrams: Metrics/logs/alerts architecture

Learning Goals

- Capture essential metrics, logs, and uptime signals.
- Define practical service objectives for homelab scale.
- Create alerts that are actionable, not noisy.

MVP Lab Worksheet

- Objective: Deploy minimum observability stack.
- Starting state: Services running without centralized telemetry.
- Steps:

  1. Add node and service metrics.
  2. Configure centralized log collection.
  3. Create three core alerts.

- Evidence: Dashboards and alert test screenshots.
- Exit criteria: Alerts fire and resolve as expected.
- Rollback: Remove alert rules causing noise.

Advanced Lab Worksheet

- Objective: Add SLO error budget view.
- Starting state: Basic telemetry active.
- Steps:

  1. Define SLI/SLO for two critical services.
  2. Add burn-rate alerting.
  3. Tie alerts to incident runbook actions.

- Evidence: SLO dashboards + incident drill logs.
- Exit criteria: Error-budget visibility in weekly ops review.
- Rollback: Revert to baseline alert set.

⚠ Author Gap Check

Include “what not to monitor yet” to keep MVP scope realistic.

Chapter 14: Backup, Restore, and Disaster Recovery

Word target: 3,600
Primary deliverable: Tested backup and DR runbooks
Key diagrams: 3-2-1 flow and DR timeline

Learning Goals

- Implement 3-2-1 backup strategy across edge and cloud.
- Define service-specific RPO/RTO targets.
- Validate restore paths through regular drills.

MVP Lab Worksheet

- Objective: Restore one VM and one app dataset.
- Starting state: Backup jobs configured.
- Steps:

  1. Trigger backup and verify completion.
  2. Restore VM snapshot to isolated target.
  3. Restore application data and verify integrity.

- Evidence: Restore logs + functional checks.
- Exit criteria: Meets defined RPO/RTO for MVP services.
- Rollback: Destroy restored test targets after validation.

Advanced Lab Worksheet

- Objective: Full DR tabletop and execution drill.
- Starting state: MVP restore verified.
- Steps:

  1. Simulate primary-node loss.
  2. Execute staged recovery in runbook order.
  3. Capture timeline and corrective actions.

- Evidence: DR report with measured recovery times.
- Exit criteria: Recovery procedure reproducible by another operator.
- Rollback: Return services to normal topology.

💡 Ubuntu Focus

Document backup coverage for `/etc`, service configs, and persistent data paths.

Chapter 15: Ubuntu Hardening and Patch Operations

Word target: 3,400
Primary deliverable: Practical Ubuntu hardening profile
Key diagrams: Layered defense model

Learning Goals

- Apply host-level hardening suited to homelab operations.
- Build predictable patch windows and emergency patch policy.
- Measure and reduce exposed attack surface.

MVP Lab Worksheet

- Objective: Implement baseline hardening controls.
- Starting state: Ubuntu baseline active.
- Steps:

  1. Configure firewall and service exposure.
  2. Enforce update policy and package hygiene.
  3. Validate hardening with checklist.

- Evidence: Hardening checklist with pass/fail results.
- Exit criteria: Critical controls enabled and verified.
- Rollback: Restore previous firewall/service policy snapshot.

Advanced Lab Worksheet

- Objective: Continuous security remediation loop.
- Starting state: Baseline hardened.
- Steps:

  1. Run periodic vulnerability scans.
  2. Prioritize findings by risk and exploitability.
  3. Remediate and retest.

- Evidence: Scan diff report and closure log.
- Exit criteria: High-risk findings closed within policy window.
- Rollback: Revert risky remediations with change control record.

ℹ Portability Placeholder

Keep distro-specific hardening commands in sidebars, not core narrative.

Reliability And Growth

Chapter 16: Reliability Engineering and Incident Response

Word target: 3,200
Primary deliverable: Incident model and response toolkit
Key diagrams: Incident lifecycle swimlane

Learning Goals

- Establish incident response workflow for small teams/solo operators.
- Reduce mean time to recovery with runbook discipline.
- Convert incidents into operational improvements.

MVP Lab Worksheet

- Objective: Run first game-day incident.
- Starting state: Monitoring and runbooks available.
- Steps:

  1. Simulate one controlled outage.
  2. Execute incident response template.
  3. Perform postmortem with action items.

- Evidence: Incident timeline + postmortem doc.
- Exit criteria: Action items assigned with due dates.
- Rollback: Restore affected service to normal operations.

Advanced Lab Worksheet

- Objective: Multi-service cascading failure drill.
- Starting state: Single-service game day complete.
- Steps:

  1. Simulate dependency failure chain.
  2. Practice communication and triage priorities.
  3. Refine alerts and runbooks.

- Evidence: Updated runbooks and reduced MTTR metrics.
- Exit criteria: Improved drill score against prior baseline.
- Rollback: Disable failure injection mechanisms.

ℹ Author Prompt

Include one postmortem excerpt showing a concrete change that prevented repeat failure.

Chapter 17: CI/CD and GitOps for Infrastructure

Word target: 3,400
Primary deliverable: Change pipeline with rollback controls
Key diagrams: Pipeline stages and promotion flow

Learning Goals

- Gate infrastructure changes with automated checks.
- Build safe promotion path from dev to live environments.
- Implement deterministic rollback for failed changes.

MVP Lab Worksheet

- Objective: Add CI checks to infra repo.
- Starting state: IaC/config repos exist.
- Steps:

  1. Add lint/validation jobs.
  2. Enforce PR-based change approvals.
  3. Verify failed check blocks merge.

- Evidence: CI run logs and policy config.
- Exit criteria: Unvalidated changes cannot reach main branch.
- Rollback: Temporarily disable strict gate via emergency policy.

Advanced Lab Worksheet

- Objective: Test end-to-end rollback path.
- Starting state: CI gates active.
- Steps:

  1. Introduce a controlled bad deploy.
  2. Trigger rollback workflow.
  3. Validate service and infra state recovery.

- Evidence: Deployment and rollback timeline.
- Exit criteria: Rollback completes inside target window.
- Rollback: Reapply last known good release tag.

⚠ Author Gap Check

Document when manual intervention is allowed and who approves it.

Chapter 18: Service Catalog Blueprints

Word target: 3,200
Primary deliverable: Reusable service blueprint template
Key diagrams: Service blueprint canvas

Learning Goals

- Define a standard onboarding blueprint for new services.
- Enforce minimum requirements for security, backup, and observability.
- Reduce variance between service deployments.

MVP Lab Worksheet

- Objective: Deploy two services with one blueprint.
- Starting state: Blueprint template drafted.
- Steps:
  1. Apply blueprint to documentation/wiki service.
  2. Apply blueprint to automation/tooling service.
  3. Validate both with readiness checklist.
- Evidence: Two completed service scorecards.
- Exit criteria: Both services pass baseline quality gate.
- Rollback: Remove non-compliant service deployment.

Advanced Lab Worksheet

- Objective: Add policy-as-code checks.
- Starting state: Service blueprint operational.
- Steps:
  1. Encode mandatory controls as checks.
  2. Run checks in CI before deployment.
  3. Block deployments that fail controls.
- Evidence: Policy check outputs.
- Exit criteria: Enforced compliance before runtime.
- Rollback: Switch to manual approval while fixing checks.

Portability Placeholder

Add “equivalent implementations” table for local VM, cloud VM, and container targets.

Chapter 19: Growth Path (Kubernetes, HA, Multi-Node)

Word target: 3,200
Primary deliverable: Expansion decision framework
Key diagrams: Complexity trigger decision tree

Learning Goals

- Decide when scaling complexity is justified.
- Build migration strategy from single-node to distributed setups.
- Keep architecture portable as hardware and providers change.

MVP Lab Worksheet

- Objective: Define objective scale triggers.
- Starting state: Stable baseline operations.
- Steps:

  1. Set CPU/RAM/storage and incident-rate thresholds.
  2. Define migration trigger matrix.
  3. Publish ADR for next-stage architecture.

- Evidence: Trigger matrix + ADR.
- Exit criteria: Scale decisions become criteria-driven.
- Rollback: Keep single-node baseline and defer migration.

Advanced Lab Worksheet

- Objective: Run one small multi-node prototype.
- Starting state: Trigger matrix approved.
- Steps:

  1. Prototype clustered workload.
  2. Validate operational overhead and resilience gain.
  3. Compare gains against complexity cost.

- Evidence: Prototype report with go/no-go outcome.
- Exit criteria: Data-backed expansion decision.
- Rollback: Decommission prototype and retain baseline.

Ubuntu Focus

Document Ubuntu-specific operational changes required when moving to clustered systems.

Chapter 20: Capstone and Living Documentation

Word target: 2,800
Primary deliverable: End-to-end validated reference implementation
Key diagrams: Final architecture and operations map

Learning Goals

- Integrate all prior chapters into one coherent Ubuntu-first system.
- Demonstrate operational readiness with evidence.
- Create continuous documentation and release workflow for the book itself.

MVP Lab Worksheet

- Objective: Build full reference MVP stack.
- Starting state: Prior chapter artifacts complete.
- Steps:

  1. Deploy canonical services using documented standards.
  2. Run security, backup, and observability validation.
  3. Record outcomes in final scorecard.

- Evidence: Capstone validation dossier.
- Exit criteria: MVP meets defined SLO-lite and recovery goals.
- Rollback: Revert to previous stable snapshot bundle.

Advanced Lab Worksheet

- Objective: Operational audit and v1.0 publication.
- Starting state: MVP capstone validated.
- Steps:

  1. Conduct full operational audit.
  2. Close critical gaps.
  3. Publish v1.0 handbook and next-cycle backlog.

- Evidence: Audit report + release notes.
- Exit criteria: Book and lab both release-ready.
- Rollback: Publish release candidate with known issues list.

❗ Final Author Checklist

Ensure each chapter has: prerequisites, command outputs, failure tests, rollback, and “what changed” reflection.

Appendices

Appendix A: Ubuntu-to-Other Platforms Mapping

- Package management equivalents.
- Network configuration equivalents.
- Service management equivalents.
- Firewall and logging equivalents.

Appendix B: Reusable Templates

- ADR template.
- Incident report template.
- Runbook template.
- Service blueprint template.
- DR checklist template.

Appendix C: Validation Checklists

- Security baseline.
- Backup/restore validation.
- Observability baseline.
- Deployment readiness.

Appendix D: Diagram and Screenshot Shotlist

- Required diagrams per chapter.
- Screenshot checklist for command evidence.
- Consistent naming convention for assets.