

1 End-to-End Development Workflow — Combined User Stories

Below are all user stories from the chapters, reorganized into a single, execution-ready workflow.

1.1 Additional Stories

CI-1 — Getting Started

Epic / Feature	Continuous Integration
Business Value	Establish shared understanding of CI, fast feedback, and “keep main green” to reduce integration risk
Priority / Estimate	Priority: Must SP: 3
Persona	developer on a new repo
Dependencies	Build tooling, unit test framework
Assumptions / Risks	If build toolchain differs locally vs CI, setup time may expand; risk of flaky initial test

Story *As a developer on a new repo, I want to Getting Started so that Establish shared understanding of CI, fast feedback, and “keep main green” to reduce integration risk.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Initialize repo with build tool (`npm/mvn/gradle`) and a `hello_world` unit test.
- Create `ci.yml` with triggers on `push/pull_request` to `main`.
- Add steps: checkout, setup toolchain, install deps, build, run tests, upload test report.
- Enable required status checks on `main`; protect branch with fast-forward or merge queue.
- Add CI badge to `README.md`; document “keep main green” policy.

CI-2 — Introducing CI

Epic / Feature	Continuous Integration
Business Value	Codify team habits (frequent commits, fix red builds) to increase flow efficiency
Priority / Estimate	Priority: Must SP: 2
Persona	team contributor
Dependencies	Branch protection supported in VCS
Assumptions / Risks	Cultural adoption risk; enforce via required checks and stop-the-line policy

Story *As a team contributor, I want to Introducing CI so that Codify team habits (frequent commits, fix red builds) to increase flow efficiency.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Define commit/PR guidelines (small PRs, issue links, naming).
- Configure required reviews and required CI checks for `main`.
- Add CODEOWNERS for critical paths; auto-request reviewers.
- Create an on-call rotation to “stop the line” on red builds; document SLA.
- Add PR template with checklist (tests, docs, security notes).

CI-3 — Reducing Risks Using CI

Epic / Feature	Continuous Integration
Business Value	Artifacting and visible quality metrics reduce late defects and deployment surprises
Priority / Estimate	Priority: Must SP: 3
Persona	release engineer
Dependencies	Artifact storage and coverage tooling
Assumptions / Risks	Coverage thresholds may fail initially; iterate thresholds upward

Story *As a release engineer, I want to Reducing Risks Using CI so that Artifacting and visible quality metrics reduce late defects and deployment surprises.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Publish build artifacts (packages/bundles) to artifact store; retain for 30 days.
- Generate coverage report; upload as artifact and comment summary on PR.
- Add static analysis (lint/type-check) and fail on error.
- Gate merges with a minimum coverage threshold; start low, ratchet weekly.
- Produce a build manifest (commit, version, artifact SHA) and attach to job summary.

CI-4 — Build at Every Change

Epic / Feature	Continuous Integration
Business Value	Fast, repeatable builds shorten feedback loops and improve developer throughput
Priority / Estimate	Priority: Must SP: 5
Persona	build engineer
Dependencies	Cache support, job matrix
Assumptions / Risks	SLA breach risk if dependencies uncached; add caching and stage split

Story *As a build engineer, I want to Build at Every Change so that Fast, repeatable builds shorten feedback loops and improve developer throughput.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Add dependency cache with robust keys and restore-keys.
- Split jobs (lint, unit, build) to run in parallel.
- Add language/runtime matrix (e.g., Node LTS-1, LTS, latest).
- Set max commit-stage duration target (e.g., \leq 10 minutes); alert on regressions.
- Capture build scan/timing metrics; post to job summary.

CI-5 — Continuous Database Integration

Epic / Feature	Continuous Integration
Business Value	Versioned migrations prevent schema drift and enable safe evolution
Priority / Estimate	Priority: Must SP: 5
Persona	backend developer
Dependencies	DB service in CI, migration tool
Assumptions / Risks	Migration ordering conflicts; use sandbox DB and rollback scripts

Story As a backend developer, I want to Continuous Database Integration so that Versioned migrations prevent schema drift and enable safe evolution.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed flagged.

Tasks

- Add migration tool (Flyway/Liquibase/Prisma) to repo.
- Provision ephemeral CI database service; apply clean schema on each run.
- Implement forward migration and matching rollback script.
- Seed minimal test data; run DB tests after migrations.
- Upload migration logs and DB schema diff as artifacts.

CI-6 — Continuous Testing

Epic / Feature	Continuous Integration
Business Value	Layered tests (unit→component→system) increase confidence with speed
Priority / Estimate	Priority: Must SP: 5
Persona	QA engineer
Dependencies	Test categorization, runners
Assumptions / Risks	Flaky tests create noise; quarantine and deflake policy

Story As a QA engineer, I want to Continuous Testing so that Layered tests (unit→component→system) increase confidence with speed.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Tag tests by layer (@unit, @component, @system); wire selective runners.
- Run unit tests on every push; run slower suites on PR or schedule.
- Fail fast on test flakiness; auto-quarantine and create issue.
- Collect JUnit/HTML reports and screenshots/videos for failures.
- Track pass rate and flake rate trends in job summaries.

CI-7 — Continuous Inspection

Epic / Feature	Continuous Integration
Business Value	Automated inspection (lint, SAST, coverage) raises baseline quality
Priority / Estimate	Priority: Should SP: 3
Persona	security champion
Dependencies	Linters, SAST scanner
Assumptions / Risks	Initial findings may be high; add waivers and remediation backlog

Story *As a security champion, I want to Continuous Inspection so that Automated inspection (lint, SAST, coverage) raises baseline quality.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Add linters/formatters to CI (`eslint`, `black`, `golangci-lint`, etc.).
- Enable SAST/SCA scans; upload SARIF to code scanning.
- Establish allowlist/waiver mechanism with expiry dates.
- Fail on new high/critical issues; summarize in PR.
- Schedule weekly full scan job; export trend report.

CI-8 — Continuous Deployment (Intro)

Epic / Feature	Continuous Integration
Business Value	Versioned packages and rollback scripts de-risk promotions
Priority / Estimate	Priority: Should SP: 3
Persona	release manager
Dependencies	Registry access; signing keys
Assumptions / Risks	Rollback untested; include simulated rollback in staging

Story *As a release manager, I want to Continuous Deployment (Intro) so that Versioned packages and rollback scripts de-risk promotions.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Implement semantic versioning and changelog generation.
- Sign artifacts/images; push to registry with provenance.
- Create staging deploy script; add `-rollback` path.
- Run canary or blue/green simulation in staging; record outcome.
- Document release/rollback steps in `RELEASE.md`.

CI-9 — Continuous Feedback

Epic / Feature	Continuous Integration
Business Value	Visible signals (badges, PR summaries, alerts) accelerate fixing time
Priority / Estimate	Priority: Should SP: 2
Persona	team lead
Dependencies	Chat/webhook integration
Assumptions / Risks	Alert fatigue risk; tune thresholds and channels

Story *As a team lead, I want to Continuous Feedback so that Visible signals (badges, PR summaries, alerts) accelerate fixing time.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Add CI status/coverage badges to README.md.
- Emit concise job summaries (key metrics, links to artifacts).
- Integrate notifications to chat with failure-only or noisy-channel rules.
- Create “First failure owner” routing; page on red builds during business hours.
- Add post-merge dashboard (lead time, pass rate).

2 Continuous Delivery (Humble & Farley) — Part I: Foundations

CD-1 — The Problem of Delivering Software

Epic / Feature	Continuous Delivery
Business Value	Baseline DORA metrics (lead time, CFR, MTTR) to focus improvement
Priority / Estimate	Priority: Must SP: 3
Persona	product owner
Dependencies	Metrics capture in pipeline
Assumptions / Risks	Metric definitions inconsistent; document glossary and method

Story *As a product owner, I want to The Problem of Delivering Software so that Baseline DORA metrics (lead time, CFR, MTTR) to focus improvement.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Define metric glossary (lead time, CFR, MTTR, deploy frequency).
- Capture deployment events and commit timestamps in CI/CD.
- Compute metrics in a scheduled job; publish to dashboard.
- Add goals and alerts for regressions; include in retro.
- Document data sources and caveats in `docs/metrics.md`.

CD-2 — Configuration Management

Epic / Feature	Continuous Delivery
Business Value	Immutable builds & SBOM improve traceability and compliance
Priority / Estimate	Priority: Must SP: 5
Persona	release engineer
Dependencies	SBOM tool, artifact store
Assumptions / Risks	Dependency metadata gaps; pin versions and generate SBOM

Story *As a release engineer, I want to Configuration Management so that Immutable builds & SBOM improve traceability and compliance.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed flagged.

Tasks

- Pin base images and dependencies; lockfiles committed.
- Generate SBOM (e.g., CycloneDX/SPDX) during build; attach to artifact.
- Store artifacts immutably with content address (SHA).
- Verify signatures and SBOM presence in release gate.
- Record build provenance (builder, inputs) in job summary.

CD-3 — Continuous Integration (Bridge)

Epic / Feature	Continuous Delivery
Business Value	Align CI policies with CD (small PRs, reviews) to sustain flow
Priority / Estimate	Priority: Must SP: 2
Persona	tech lead
Dependencies	Merge rules in VCS
Assumptions / Risks	Merge queue learning curve; document examples

Story As a tech lead, I want to Continuous Integration (Bridge) so that Align CI policies with CD (small PRs, reviews) to sustain flow.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repository and pipeline configuration are available

When the user completes the *Hands-on Objective* for this chapter

Then the stated *Outcome* for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- ☐Enforce PR size checks (warn on > 400 LOC changed).
- ☐Require review from codeowners on critical areas.
- ☐Enable merge queue/linear history; document usage.
- ☐Add examples of good PRs and review checklist.

CD-4 — Testing Strategy

Epic / Feature	Continuous Delivery
Business Value	Acceptance tests as specification reduce rework
Priority / Estimate	Priority: Must SP: 5
Persona	QA lead
Dependencies	AAT framework, test data
Assumptions / Risks	Environment instability; use ephemeral envs

Story As a QA lead, I want to Testing Strategy so that Acceptance tests as specification reduce rework.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Stand up acceptance test framework (e.g., Playwright/Cypress/Behave).
- Define acceptance criteria as executable specs; tag `@acceptance`.
- Provision ephemeral test env per PR with seeded data.
- Capture screenshots/video and HAR on failure.
- Publish acceptance report and link from job summary.

3 Continuous Delivery — Part II: The Deployment Pipeline

CD-5 — Anatomy of the Deployment Pipeline

Epic / Feature	Continuous Delivery
Business Value	Explicit stages and gates clarify artifact flow and ownership
Priority / Estimate	Priority: Must SP: 3
Persona	platform engineer
Dependencies	Pipeline-as-code
Assumptions / Risks	Diagram-policy drift; keep diagram in-repo

Story *As a platform engineer, I want to Anatomy of the Deployment Pipeline so that Explicit stages and gates clarify artifact flow and ownership.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Model stages (commit → acceptance → NFR → staging).
- Define promotion gates and required evidence per stage.
- Generate pipeline diagram and commit to `docs/pipeline.png`.
- Add ownership and on-call mapping to each stage.
- Validate artifact handoffs with a dry-run.

CD-6 — Build & Deployment Scripting

Epic / Feature	Continuous Delivery
Business Value	Idempotent, one-command deploys remove manual error
Priority / Estimate	Priority: Must SP: 5
Persona	devops engineer
Dependencies	IaC module, deploy script
Assumptions / Risks	Hidden manual steps; remove or automate

Story As a devops engineer, I want to Build & Deployment Scripting so that Idempotent, one-command deploys remove manual error.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed flagged.

Tasks

- Create idempotent deploy script (`deploy.sh`/Make target) with `-dry-run`.
- Externalize config/env via parameters or env files; no secrets in code.
- Integrate infra provisioning via IaC module; validate plan before apply.
- Add health checks and post-deploy verification step.
- Document rollback procedure and test it in staging.

CD-7 — The Commit Stage

Epic / Feature	Continuous Delivery
Business Value	Sub-10-minute commit stage preserves rapid feedback
Priority / Estimate	Priority: Must SP: 3
Persona	build engineer
Dependencies	Caching, parallelization
Assumptions / Risks	SLA risk; monitor and enforce guardrails

Story *As a build engineer, I want to The Commit Stage so that Sub-10-minute commit stage preserves rapid feedback.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Profile commit stage; identify slowest steps.
- Parallelize independent tasks; add caching for deps/builds.
- Defer integration/system tests to later stages.
- Enforce time budget; fail builds exceeding threshold with guidance.
- Track median/p95 duration; surface trend.

CD-8 — Automated Acceptance Testing

Epic / Feature	Continuous Delivery
Business Value	Reliable AAT improves release confidence
Priority / Estimate	Priority: Should SP: 5
Persona	QA engineer
Dependencies	Ephemeral env; seed data
Assumptions / Risks	Flaky env; capture videos/screenshots for triage

Story *As a QA engineer, I want to Automated Acceptance Testing so that Reliable AAT improves release confidence.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- ☐ Provision ephemeral env per PR with deterministic seed data.
- ☐ Run AAT in parallel shards; retry once on failure with quarantine tag.
- ☐ Collect artifacts (videos, traces); upload for triage.
- ☐ Block promotion on failing critical AAT scenarios.
- ☐ Track AAT pass rate by suite/component.

CD-9 — Testing Nonfunctional Requirements

Epic / Feature	Continuous Delivery
Business Value	Perf/security smoke gates catch regressions early
Priority / Estimate	Priority: Should SP: 5
Persona	SRE / Sec Eng
Dependencies	k6, SCA/SAST
Assumptions / Risks	False positives; thresholds and baselines required

Story As a SRE / Sec Eng, I want to Testing Nonfunctional Requirements so that Perf/security smoke gates catch regressions early.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Add quick perf smoke (e.g., k6 1–3 min) against staging; alert on % change vs baseline.
- Run SCA/SAST; fail on new high/critical.
- Capture latency/throughput/error-rate; publish trend.
- Define thresholds and suppression policy with expiry.
- Include dependency vulnerability diff in job summary.

CD-10 — Deploying and Releasing Applications

Epic / Feature	Continuous Delivery
Business Value	Feature flags and canaries decouple deploy from release
Priority / Estimate	Priority: Should SP: 5
Persona	release manager
Dependencies	Flag service
Assumptions / Risks	Flag debt; add cleanup policy

Story *As a release manager, I want to Deploying and Releasing Applications so that Feature flags and canaries decouple deploy from release.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Integrate feature flag SDK; default flags off on deploy.
- Create canary strategy (small % traffic); monitor key metrics.
- Add progressive rollout workflow with manual approval step.
- Record flag change events alongside deploy events.
- Establish flag cleanup SLA; add linter to detect stale flags.

4 Continuous Delivery — Part III: The Delivery Ecosystem

CD-11 — Managing Infrastructure and Environments

Epic / Feature	Continuous Delivery
Business Value	IaC with parity reduces “works-on-my-machine” failures
Priority / Estimate	Priority: Must SP: 5
Persona	platform engineer
Dependencies	IaC repo, drift detection
Assumptions / Risks	Version drift; pin images and modules

Story *As a platform engineer, I want to Managing Infrastructure and Environments so that IaC with parity reduces “works-on-my-machine” failures.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Definition of Done: All ACs pass; Tests green; Security/all checks; Docs updated; Deployed flagged.

Tasks

- Create environment definitions (dev/stage/prod) via IaC modules.
- Pin base images and module versions; document upgrade path.
- Enable drift detection; alert on config divergence.
- Bake golden images; rotate regularly.
- Add smoke checks per environment; publish status page.

CD-12 — Managing Data

Epic / Feature	Continuous Delivery
Business Value	Expand/contract enables zero-downtime DB changes
Priority / Estimate	Priority: Must SP: 5
Persona	database engineer
Dependencies	Migration plan
Assumptions / Risks	Backward-compat assumptions; test dual-write/dual-read

Story *As a database engineer, I want to Managing Data so that Expand/contract enables zero-downtime DB changes.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Design expand/contract plan; add nullable columns/dual paths.
- Implement dual-write; monitor divergence.
- Backfill data safely (batch/ throttled).
- Switch reads; verify; then contract and remove legacy.
- Add rollback guardrails and validation queries.

CD-13 — Managing Components & Dependencies

Epic / Feature	Continuous Delivery
Business Value	Contracts + pinning stabilize integrations
Priority / Estimate	Priority: Should SP: 5
Persona	service owner
Dependencies	CDC tests, dep scan
Assumptions / Risks	API breakage; run CDC in PRs

Story *As a service owner, I want to Managing Components & Dependencies so that Contracts + pinning stabilize integrations.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Define API contracts (OpenAPI/AsyncAPI) and publish.
- Implement consumer-driven contract tests in CI.
- Pin external service versions/clients; use compat ranges.
- Enable dep update bot; require CI green + contract pass.
- Add chaos test for dependency outage fallback.

CD-14 — Advanced Version Control

Epic / Feature	Continuous Delivery
Business Value	Trunk or short-lived branches scale delivery cadence
Priority / Estimate	Priority: Should SP: 3
Persona	tech lead
Dependencies	Policy doc
Assumptions / Risks	Policy drift; include examples & bots

Story *As a tech lead, I want to Advanced Version Control so that Trunk or short-lived branches scale delivery cadence.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- ☐Adopt trunk-based policy; enforce short-lived branches (< 2 days).
- ☐Enable pre-commit hooks (format/lint/test).
- ☐Configure merge queue and auto-rebase.
- ☐Provide exemplars of small PRs and review heuristics.

CD-15 — Managing Continuous Delivery

Epic / Feature	Continuous Delivery
Business Value	Automated audit trails satisfy governance/compliance
Priority / Estimate	Priority: Must SP: 3
Persona	compliance owner
Dependencies	Change log export
Assumptions / Risks	Gaps in metadata; enrich job summaries

Story *As a compliance owner, I want to Managing Continuous Delivery so that Automated audit trails satisfy governance/compliance.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed flagged.

Tasks

- Capture change metadata (who/what/when/why) in job outputs.
- Export audit log to central store; retain per policy.
- Generate release notes automatically from commits/PRs.
- Add compliance checklist to PR template.
- Schedule periodic audit report generation.

5 Learning GitHub Actions (Brent Laster) — Part I: Foundations

GA-1 — The Basics

Epic / Feature	GitHub Actions Platform
Business Value	Working CI workflow standardizes contribution checks
Priority / Estimate	Priority: Must SP: 2
Persona	repo maintainer
Dependencies	ci.yml
Assumptions / Risks	Badge visibility; README update needed

Story *As a repo maintainer, I want to The Basics so that Working CI workflow standardizes contribution checks.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Create `.github/workflows/ci.yml` with checkout, setup, build, test.
- Add matrix for OS/runtime if relevant.
- Upload test reports and coverage artifacts.
- Add build badge to `README.md`; link to Actions tab.

GA-2 — How Actions Work

Epic / Feature	GitHub Actions Platform
Business Value	Multi-job workflows with conditionals enable clear stage flow
Priority / Estimate	Priority: Must SP: 3
Persona	automation engineer
Dependencies	needs/if usage
Assumptions / Risks	Over-complex logic; keep expressions simple

Story As a automation engineer, I want to How Actions Work so that Multi-job workflows with conditionals enable clear stage flow.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repository and pipeline configuration are available

When the user completes the *Hands-on Objective* for this chapter

Then the stated *Outcome* for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Definition of Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Create multi-job workflow (lint → test → build).
- Use `needs`: to express dependencies; add `if`: conditionals for paths/labels.
- Share artifacts between jobs; consume in downstream steps.
- Demonstrate manual re-run and `workflow_dispatch`.

GA-3 — What's in an Action?

Epic / Feature	GitHub Actions Platform
Business Value	Local composite action reduces duplication across repos
Priority / Estimate	Priority: Should SP: 3
Persona	tooling engineer
Dependencies	.github/actions/lint
Assumptions / Risks	Versioning; tag action revisions

Story *As a tooling engineer, I want to What's in an Action? so that Local composite action reduces duplication across repos.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repository and pipeline configuration are available

When the user completes the *Hands-on Objective* for this chapter

Then the stated *Outcome* for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Definition of Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Create .github/actions/lint/action.yml composite action.
- Parameterize inputs (paths, config file).
- Replace duplicated steps in workflows with the composite action.
- Tag versions and document usage in README.

GA-4 — Working with Workflows

Epic / Feature	GitHub Actions Platform
Business Value	Manual dispatch with parameters supports ops tasks
Priority / Estimate	Priority: Should SP: 2
Persona	ops engineer
Dependencies	workflow_dispatch
Assumptions / Risks	Undocumented inputs; README examples required

Story *As a ops engineer, I want to Working with Workflows so that Manual dispatch with parameters supports ops tasks.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Add workflow_dispatch with typed inputs (env, version).
- Validate inputs; fail with helpful messages.
- Document runbook and examples in README.
- Restrict to maintainers with environment protections.

GA-5 — Runners

Epic / Feature	GitHub Actions Platform
Business Value	Self-hosted runners unlock custom environments and scale
Priority / Estimate	Priority: Could SP: 5
Persona	platform engineer
Dependencies	Runner labels
Assumptions / Risks	Security posture; isolate and rotate tokens

Story *As a platform engineer, I want to Runners so that Self-hosted runners unlock custom environments and scale.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Provision self-hosted runner VM/container; apply labels.
- Configure ephemeral runners; auto-scale pool.
- Lock down network and credentials; rotate tokens.
- Route heavy jobs to self-hosted via `runs-on: [self-hosted, label]`.
- Monitor utilization and queue times; auto-scale thresholds.

6 GitHub Actions — Part II: Building Blocks

GA-6 — Managing Workflow Environments

Epic / Feature	GitHub Actions Platform
Business Value	Protected environments and OIDC harden deployments
Priority / Estimate	Priority: Must SP: 3
Persona	security champion
Dependencies	Env rules, OIDC
Assumptions / Risks	Reviewer fatigue; configure sensible wait timers

Story *As a security champion, I want to Managing Workflow Environments so that Protected environments and OIDC harden deployments.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Create environments (staging, prod) with required reviewers and wait timers.
- Set permissions: `read-all` at top; elevate per job minimally.
- Configure OIDC trust with cloud provider; remove long-lived secrets.
- Use `environment:` and `protection_rules` in deploy jobs.

GA-7 — Managing Data Within Workflows

Epic / Feature	GitHub Actions Platform
Business Value	Caching & artifacts speed builds and preserve outputs
Priority / Estimate	Priority: Should SP: 3
Persona	build engineer
Dependencies	cache/action, artifacts
Assumptions / Risks	Stale cache; use keys and restore-keys wisely

Story *As a build engineer, I want to Managing Data Within Workflows so that Caching & artifacts speed builds and preserve outputs.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed flagged.

Tasks

- Add actions/cache with precise keys and restore-keys.
- Upload key build outputs as artifacts; set retention days.
- Use actions/upload-artifact for logs/reports.
- Periodically bust caches on lockfile change.

GA-8 — Managing Workflow Execution

Epic / Feature	GitHub Actions Platform
Business Value	Matrices & concurrency optimize throughput
Priority / Estimate	Priority: Should SP: 3
Persona	automation engineer
Dependencies	matrix, concurrency.group
Assumptions / Risks	Overlapping runs; group naming convention

Story As a automation engineer, I want to Managing Workflow Execution so that Matrices & concurrency optimize throughput.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed flagged.

Tasks

- Add job matrix (os/runtime/db) with fail-fast disabled when appropriate.
- Configure concurrency: group with cancel-in-progress: true.
- Use paths(paths-ignore) to scope triggers.
- Surface matrix results in summary table.

7 GitHub Actions — Part III: Security and Monitoring

GA-9 — Actions and Security

Epic / Feature	GitHub Actions Platform
Business Value	Least-privilege and fork hardening reduce supply-chain risk
Priority / Estimate	Priority: Must SP: 3
Persona	security champion
Dependencies	permissions: read-all, policy
Assumptions / Risks	Breakages due to perms; document elevation steps

Story *As a security champion, I want to Actions and Security so that Least-privilege and fork hardening reduce supply-chain risk.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed flagged.

Tasks

- Set default permissions: contents: read; elevate per job.
- Restrict pull_request from forks from accessing secrets; use pull_request_target safely if needed.
- Pin third-party actions by commit SHA; audit monthly.
- Enable Dependabot security updates for actions and packages.
- Add policy checks for disallowed actions and secret scanning.

GA-10 — Monitoring, Logging, and Debugging

Epic / Feature	GitHub Actions Platform
Business Value	Job summaries improve triage speed and observability
Priority / Estimate	Priority: Should SP: 2
Persona	dev lead
Dependencies	summary markdown
Assumptions / Risks	Noisy logs; link to artifacts and coverage

Story *As a dev lead, I want to Monitoring, Logging, and Debugging so that Job summaries improve triage speed and observability.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Write concise {{ github.step_summary }} markdown (key metrics, links).
- Enable step-level if: always() logs on failure; upload debug bundle.
- Add ACTIONS_STEP_DEBUG toggle via secrets for deep dives.
- Include permalinks to artifacts, coverage, and failing tests.

8 GitHub Actions — Part IV: Advanced Topics

GA-11 — Creating Custom Actions

Epic / Feature	GitHub Actions Platform
Business Value	Versioned public action promotes reuse org-wide
Priority / Estimate	Priority: Could SP: 5
Persona	tooling maintainer
Dependencies	JS action repo
Assumptions / Risks	API changes; semantic versioning policy

Story *As a tooling maintainer, I want to Creating Custom Actions so that Versioned public action promotes reuse org-wide.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Scaffold JS action (`action.yml`, `src/`, `dist/`); commit built assets.
- Add inputs/outputs with validation and error handling.
- Write unit tests and an example workflow.
- Publish v1 tag and release notes; maintain v1 major tag.
- Set up automated release via `release-please`/`semantic-release`.

GA-12 — Advanced Workflows

Epic / Feature	GitHub Actions Platform
Business Value	Reusable workflows enforce governance and DRY
Priority / Estimate	Priority: Must SP: 3
Persona	org admin
Dependencies	caller workflows
Assumptions / Risks	Adoption lag; publish examples and starter kits

Story As a org admin, I want to Advanced Workflows so that Reusable workflows enforce governance and DRY.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repository and pipeline configuration are available

When the user completes the *Hands-on Objective* for this chapter

Then the stated *Outcome* for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Definition of Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Create reusable workflow triggered by `workflow_call` with inputs/secrets.
- Migrate two repos to use the reusable workflow.
- Add org-level examples and starter templates.
- Version reusable workflows; document breaking changes.

GA-13 — Advanced Workflow Techniques

Epic / Feature	GitHub Actions Platform
Business Value	Containerized jobs and services enable realistic CI envs
Priority / Estimate	Priority: Should SP: 5
Persona	integration engineer
Dependencies	services:, gh CLI
Assumptions / Risks	Resource limits; right-size service containers

Story *As a integration engineer, I want to Advanced Workflow Techniques so that Containerized jobs and services enable realistic CI envs.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Run job in a container (`container:`) with required tools.
- Add `services:` (db/cache); wait-for-health before tests.
- Use gh CLI for release/promo tasks; store token via OIDC.
- Tune resource limits and cleanup steps to avoid leaks.

GA-14 — Migrating to GitHub Actions

Epic / Feature	GitHub Actions Platform
Business Value	Parity migration de-risks CI platform change
Priority / Estimate	Priority: Must SP: 5
Persona	platform owner
Dependencies	Parity checklist
Assumptions / Risks	Hidden gaps; track transformers and exceptions

Story *As a platform owner, I want to Migrating to GitHub Actions so that Parity migration de-risks CI platform change.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Build a parity checklist (triggers, envs, secrets, artifacts, approvals).
- Port one pipeline end-to-end; compare outputs/timings.
- Implement transformers for env variables and workspace semantics.
- Run dual CI for a sprint; fix gaps; cut over with rollback plan.
- Document exceptions and future optimizations.

Capstone & Milestones (Reference)

CI Milestone: fast/complete tiers, layered tests, inspections, DB migrations, artifacts, manual promote + rollback.

CD Milestone: commit → acceptance → NFR → staging, scripted deploys, flags, audit trail.

GHA Milestone: reusable workflows, protected envs, least-privilege defaults, custom action, migration playbook.

Appendix — Original Chapters

User Stories by Chapter: CI → CD → GitHub Actions

Compiled for Jordan Suber

Contents

1	End-to-End Development Workflow — Combined User Stories	1
1.1	Additional Stories	1
2	Continuous Delivery (Humble & Farley) — Part I: Foundations	10
3	Continuous Delivery — Part II: The Deployment Pipeline	14
4	Continuous Delivery — Part III: The Delivery Ecosystem	20
5	Learning GitHub Actions (Brent Laster) — Part I: Foundations	25
6	GitHub Actions — Part II: Building Blocks	30
7	GitHub Actions — Part III: Security and Monitoring	33
8	GitHub Actions — Part IV: Advanced Topics	35
9	Continuous Integration (Duvall, Matyas, Glover)	42

How This Set Was Built

Each story maps one chapter's *Learning Goals* to a concise user story, uses the chapter's *Hands-on Objective* as the operative behavior, and verifies the published *Outcome*. Stories are intentionally compact (per INVEST) and ready for backlog import. (Source study plan and template referenced externally.)

9 Continuous Integration (Duvall, Matyas, Glover)

CI-1 — Getting Started

Epic / Feature	Continuous Integration
Business Value	Establish shared understanding of CI, fast feedback, and “keep main green” to reduce integration risk
Priority / Estimate	Priority: Must SP: 3
Persona	developer on a new repo
Dependencies	Build tooling, unit test framework
Assumptions / Risks	If build toolchain differs locally vs CI, setup time may expand; risk of flaky initial test

Story *As a developer on a new repo, I want to Getting Started so that Establish shared understanding of CI, fast feedback, and “keep main green” to reduce integration risk.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Initialize repo with build tool (`npm/mvn/gradle`) and a `hello_world` unit test.
- Create `ci.yml` with triggers on `push/pull_request` to `main`.
- Add steps: checkout, setup toolchain, install deps, build, run tests, upload test report.
- Enable required status checks on `main`; protect branch with fast-forward or merge queue.
- Add CI badge to `README.md`; document “keep main green” policy.

CI-2 — Introducing CI

Epic / Feature	Continuous Integration
Business Value	Codify team habits (frequent commits, fix red builds) to increase flow efficiency
Priority / Estimate	Priority: Must SP: 2
Persona	team contributor
Dependencies	Branch protection supported in VCS
Assumptions / Risks	Cultural adoption risk; enforce via required checks and stop-the-line policy

Story *As a team contributor, I want to Introducing CI so that Codify team habits (frequent commits, fix red builds) to increase flow efficiency.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Define commit/PR guidelines (small PRs, issue links, naming).
- Configure required reviews and required CI checks for `main`.
- Add CODEOWNERS for critical paths; auto-request reviewers.
- Create an on-call rotation to “stop the line” on red builds; document SLA.
- Add PR template with checklist (tests, docs, security notes).

CI-3 — Reducing Risks Using CI

Epic / Feature	Continuous Integration
Business Value	Artifacting and visible quality metrics reduce late defects and deployment surprises
Priority / Estimate	Priority: Must SP: 3
Persona	release engineer
Dependencies	Artifact storage and coverage tooling
Assumptions / Risks	Coverage thresholds may fail initially; iterate thresholds upward

Story *As a release engineer, I want to Reducing Risks Using CI so that Artifacting and visible quality metrics reduce late defects and deployment surprises.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Publish build artifacts (packages/bundles) to artifact store; retain for 30 days.
- Generate coverage report; upload as artifact and comment summary on PR.
- Add static analysis (lint/type-check) and fail on error.
- Gate merges with a minimum coverage threshold; start low, ratchet weekly.
- Produce a build manifest (commit, version, artifact SHA) and attach to job summary.

CI-4 — Build at Every Change

Epic / Feature	Continuous Integration
Business Value	Fast, repeatable builds shorten feedback loops and improve developer throughput
Priority / Estimate	Priority: Must SP: 5
Persona	build engineer
Dependencies	Cache support, job matrix
Assumptions / Risks	SLA breach risk if dependencies uncached; add caching and stage split

Story *As a build engineer, I want to Build at Every Change so that Fast, repeatable builds shorten feedback loops and improve developer throughput.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Add dependency cache with robust keys and restore-keys.
- Split jobs (lint, unit, build) to run in parallel.
- Add language/runtime matrix (e.g., Node LTS-1, LTS, latest).
- Set max commit-stage duration target (e.g., \leq 10 minutes); alert on regressions.
- Capture build scan/timing metrics; post to job summary.

CI-5 — Continuous Database Integration

Epic / Feature	Continuous Integration
Business Value	Versioned migrations prevent schema drift and enable safe evolution
Priority / Estimate	Priority: Must SP: 5
Persona	backend developer
Dependencies	DB service in CI, migration tool
Assumptions / Risks	Migration ordering conflicts; use sandbox DB and rollback scripts

Story As a backend developer, I want to Continuous Database Integration so that Versioned migrations prevent schema drift and enable safe evolution.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed flagged.

Tasks

- Add migration tool (Flyway/Liquibase/Prisma) to repo.
- Provision ephemeral CI database service; apply clean schema on each run.
- Implement forward migration and matching rollback script.
- Seed minimal test data; run DB tests after migrations.
- Upload migration logs and DB schema diff as artifacts.

CI-6 — Continuous Testing

Epic / Feature	Continuous Integration
Business Value	Layered tests (unit→component→system) increase confidence with speed
Priority / Estimate	Priority: Must SP: 5
Persona	QA engineer
Dependencies	Test categorization, runners
Assumptions / Risks	Flaky tests create noise; quarantine and deflake policy

Story As a QA engineer, I want to Continuous Testing so that Layered tests (unit→component→system) increase confidence with speed.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Tag tests by layer (@unit, @component, @system); wire selective runners.
- Run unit tests on every push; run slower suites on PR or schedule.
- Fail fast on test flakiness; auto-quarantine and create issue.
- Collect JUnit/HTML reports and screenshots/videos for failures.
- Track pass rate and flake rate trends in job summaries.

CI-7 — Continuous Inspection

Epic / Feature	Continuous Integration
Business Value	Automated inspection (lint, SAST, coverage) raises baseline quality
Priority / Estimate	Priority: Should SP: 3
Persona	security champion
Dependencies	Linters, SAST scanner
Assumptions / Risks	Initial findings may be high; add waivers and remediation backlog

Story *As a security champion, I want to Continuous Inspection so that Automated inspection (lint, SAST, coverage) raises baseline quality.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Add linters/formatters to CI (`eslint`, `black`, `golangci-lint`, etc.).
- Enable SAST/SCA scans; upload SARIF to code scanning.
- Establish allowlist/waiver mechanism with expiry dates.
- Fail on new high/critical issues; summarize in PR.
- Schedule weekly full scan job; export trend report.

CI-8 — Continuous Deployment (Intro)

Epic / Feature	Continuous Integration
Business Value	Versioned packages and rollback scripts de-risk promotions
Priority / Estimate	Priority: Should SP: 3
Persona	release manager
Dependencies	Registry access; signing keys
Assumptions / Risks	Rollback untested; include simulated rollback in staging

Story *As a release manager, I want to Continuous Deployment (Intro) so that Versioned packages and rollback scripts de-risk promotions.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Implement semantic versioning and changelog generation.
- Sign artifacts/images; push to registry with provenance.
- Create staging deploy script; add `-rollback` path.
- Run canary or blue/green simulation in staging; record outcome.
- Document release/rollback steps in `RELEASE.md`.

CI-9 — Continuous Feedback

Epic / Feature	Continuous Integration
Business Value	Visible signals (badges, PR summaries, alerts) accelerate fixing time
Priority / Estimate	Priority: Should SP: 2
Persona	team lead
Dependencies	Chat/webhook integration
Assumptions / Risks	Alert fatigue risk; tune thresholds and channels

Story As a team lead, I want to Continuous Feedback so that Visible signals (badges, PR summaries, alerts) accelerate fixing time.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Add CI status/coverage badges to README.md.
- Emit concise job summaries (key metrics, links to artifacts).
- Integrate notifications to chat with failure-only or noisy-channel rules.
- Create “First failure owner” routing; page on red builds during business hours.
- Add post-merge dashboard (lead time, pass rate).

Capstone & Milestones (Reference)

CI Milestone: fast/complete tiers, layered tests, inspections, DB migrations, artifacts, manual promote + rollback.

CD Milestone: commit → acceptance → NFR → staging, scripted deploys, flags, audit trail.

GHA Milestone: reusable workflows, protected envs, least-privilege defaults, custom action, migration playbook.

Tailored User Stories: Emscripten + WebGL2 + WASM

A1 — Set up Emscripten SDK & CMake

Epic / Feature	Build Toolchain & Project Scaffolding
Business Value	Compile C/C++ to WebAssembly reproducibly with one command
Priority / Estimate	Priority: Must SP: 3
Persona	developer
Dependencies	emsdk; CMake; shell
Assumptions / Risks	Version drift across dev/CI; pin SDK

Story As a developer, I want to Set up Emscripten SDK & CMake so that Compile C/C++ to WebAssembly reproducibly with one command.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Install and activate `emsdk`; pin a known-good version in `README` and CI.
- Provide `toolchain.cmake`; document `emcmake`/`emmake` usage.
- Verify `emcc` and `emar` on PATH; emit to `build-wasm/`.
- AC:** Running `emcmake cmake` then `emmake make -j` emits `.html/.js/.wasm` without errors.

A2 — Build Profiles (Debug/Release)

Epic / Feature	Build Toolchain & Project Scaffolding
Business Value	Switch quickly between fast iteration and optimized output
Priority / Estimate	Priority: Must SP: 2
Persona	developer
Dependencies	CMakePresets.json
Assumptions / Risks	Wrong flags in Release can hurt perf

Story As a developer, I want to Build Profiles (Debug/Release) so that Switch quickly between fast iteration and optimized output.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Add CMakePresets.json: Debug (-O0 -g -sASSERTIONS=1 -sSAFE_HEAP=1), Release (-O3 -flto -sASSERTIONS=0).
- Document preset usage in README.
- AC:** -DCMAKE_BUILD_TYPE=Debug|Release toggles symbols/optimizations; Release bundle is smaller.

B1 — WebGL2 Context Defaults

Epic / Feature	WebGL2 Context & Canvas
Business Value	Consistent rendering across browsers
Priority / Estimate	Priority: Must SP: 3
Persona	player
Dependencies	WebGL2-enabled browser
Assumptions / Risks	Context loss without recovery

Story *As a player, I want to WebGL2 Context Defaults so that Consistent rendering across browsers.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Request WebGL2 with sensible attributes (alpha/depth/stencil).
- Log capabilities; show friendly error UI on failure.
- Handle context lost/restored events.
- AC:** Context is WebGL2; capabilities logged; fallback path documented.

B2 — Hi-DPI Canvas Resizing

Epic / Feature	WebGL2 Context & Canvas
Business Value	Crisp visuals on retina/high-DPR displays
Priority / Estimate	Priority: Must SP: 3
Persona	player
Dependencies	Resize listeners
Assumptions / Risks	Performance drop on resize

Story As a player, I want to Hi-DPI Canvas Resizing so that Crisp visuals on retina/high-DPR displays.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Listen for resize; set canvas `width/height` using device pixel ratio.
- Update viewport and projection; preserve aspect ratio.
- AC:** Resizing updates viewport/projection; no stretching; FPS stable.

C1 — Fixed-Step Update, Variable Render

Epic / Feature	Game Loop & Timing
Business Value	Deterministic simulation with smooth rendering
Priority / Estimate	Priority: Must SP: 3
Persona	developer
Dependencies	rAF/emscripten_set_main_loop
Assumptions / Risks	Unbounded delta causes instability

Story *As a developer, I want to Fixed-Step Update, Variable Render so that Deterministic simulation with smooth rendering.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Implement 60 Hz accumulator loop with clamped delta and interpolation.
- Pause on blur; resume on focus; bound time drift.
- AC:** Fixed tick drives simulation; rendering interpolates; pause/resume behaves.

D1 — Unified Input Layer

Epic / Feature	Input & Focus
Business Value	Consistent controls across desktop and mobile
Priority / Estimate	Priority: Must SP: 3
Persona	player
Dependencies	DOM events; Emscripten input helpers
Assumptions / Risks	Key repeat noise; gesture ambiguity

Story As a player, I want to Unified Input Layer so that Consistent controls across desktop and mobile.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Map keyboard, mouse, and touch to the same action set.
- Debounce key repeats; map common gestures; pause input on blur.
- AC:** Repeats debounced; gestures mapped; escape/menu works across devices.

E1 — Preload Assets

Epic / Feature	Assets & FS
Business Value	No “first-frame” misses; deterministic startup
Priority / Estimate	Priority: Must SP: 3
Persona	developer
Dependencies	Emscripten <code>-preload-file</code>
Assumptions / Risks	Silent asset failures

Story *As a developer, I want to Preload Assets so that No “first-frame” misses; deterministic startup.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Package textures/shaders/levels with `-preload-file` (or packager).
- Show loading screen until files are ready; log missing assets clearly.
- AC:** Build emits a `.data` (or preloaded files); loading screen hides only when ready.

E2 — Persist Settings with IDBFS

Epic / Feature	Assets & FS
Business Value	Player settings and progress survive reloads
Priority / Estimate	Priority: Should SP: 2
Persona	player
Dependencies	IndexedDB available
Assumptions / Risks	Quota errors; partial writes

Story As a player, I want to Persist Settings with IDBFS so that Player settings and progress survive reloads.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- ☐ Persist settings/state; FS.syncfs to IndexedDB.
- ☐ Provide reset-to-defaults; show quota errors.
- ☐ **AC:** Data appears in IndexedDB; reload restores state; reset clears storage.

F1 — Gesture-Gated Audio

Epic / Feature	Audio
Business Value	Low-latency audio that complies with browser policies
Priority / Estimate	Priority: Must SP: 2
Persona	player
Dependencies	WebAudio/SDL
Assumptions / Risks	Autoplay blocked

Story As a player, I want to Gesture-Gated Audio so that Low-latency audio that complies with browser policies.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repository and pipeline configuration are available

When the user completes the *Hands-on Objective* for this chapter

Then the stated *Outcome* for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Resume WebAudio context on first click/tap.
- Add mute/unmute; persist volume via IDBFS.
- AC:** First gesture enables audio; mute/volume persist.

G1 — Optimized Release Profile

Epic / Feature	Performance
Business Value	Higher FPS and smaller downloads
Priority / Estimate	Priority: Should SP: 3
Persona	developer
Dependencies	wasm-opt; LTO; SIMD where safe
Assumptions / Risks	Aggressive opts reduce compat

Story As a developer, I want to Optimized Release Profile so that Higher FPS and smaller downloads.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/all checks; Docs updated; Deployed flagged.

Tasks

- Use -O3 -flto -sASSERTIONS=0 and apply wasm-opt.
- Record FPS and bundle size in CI artifacts.
- AC:** CI shows size & FPS metrics; profile build has source maps.

G2 — In-Game FPS Overlay

Epic / Feature	Performance
Business Value	Quick perf checks during gameplay
Priority / Estimate	Priority: Could SP: 2
Persona	developer
Dependencies	HUD toggle
Assumptions / Risks	Overlay overhead

Story As a developer, I want to In-Game FPS Overlay so that Quick perf checks during gameplay.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repository and pipeline configuration are available

When the user completes the *Hands-on Objective* for this chapter

Then the stated *Outcome* for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Definition of Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- ☐ F3 toggles overlay; show FPS, frame time, draw calls, memory.
- ☐ Keep overhead minimal; update each frame.
- ☐ **AC:** Overlay is accurate and light-weight.

H1 — PThreads + OffscreenCanvas (Optional)

Epic / Feature	Multithreading
Business Value	Move heavy work off the main thread
Priority / Estimate	Priority: Could SP: 5
Persona	developer
Dependencies	Cross-origin isolation; SAB
Assumptions / Risks	Fallback path often forgotten

Story As a developer, I want to PThreads + OffscreenCanvas (Optional) so that Move heavy work off the main thread.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Provide -sUSE_PTHREADS=1 build with OffscreenCanvas path.
- Detect crossOriginIsolated; fallback to main thread if false.
- AC:** Worker mode runs under isolation; documented fallback otherwise.

H2 — COOP/COEP Headers

Epic / Feature	Multithreading
Business Value	Enable SharedArrayBuffer safely
Priority / Estimate	Priority: Must SP: 2
Persona	DevOps engineer
Dependencies	Server config
Assumptions / Risks	Prod headers differ from local

Story As a DevOps engineer, I want to COOP/COEP Headers so that Enable SharedArrayBuffer safely.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repository and pipeline configuration are available

When the user completes the *Hands-on Objective* for this chapter

Then the stated *Outcome* for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Configure COOP/COEP for local dev and production hosting.
- Add runtime isolation check and warning UI.
- AC:** SAB works under isolation; warning shown if missing.

I1 — Scores API (Fetch/WebSockets)

Epic / Feature	Networking
Business Value	Leaderboards & telemetry without UI lockups
Priority / Estimate	Priority: Should SP: 3
Persona	player
Dependencies	HTTP endpoint
Assumptions / Risks	Offline gaps; retry storms

Story As a player, I want to Scores API (Fetch/WebSockets) so that Leaderboards & telemetry without UI lockups.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Implement fetch wrapper with timeout/retry/backoff; offline queue.
- AC:** POST/GET succeed; errors handled; retries/backoff non-blocking.

J1 — Headless Smoke Tests

Epic / Feature	Testing & CI/CD
Business Value	Catch regressions automatically
Priority / Estimate	Priority: Must SP: 3
Persona	maintainer
Dependencies	Playwright/Puppeteer
Assumptions / Risks	Flaky CI env

Story As a maintainer, I want to Headless Smoke Tests so that Catch regressions automatically.

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario Happy path

Given the target repository and pipeline configuration are available

When the user completes the *Hands-on Objective* for this chapter

Then the stated *Outcome* for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Launch page headlessly; assert WebGL2 context; step frames; assert FPS threshold in Release.
- Attach screenshots/videos on failure.
- AC:** CI run fails on missing WebGL2/low FPS; artifacts uploaded.

J2 — Build & Deploy to GitHub Pages

Epic / Feature	Testing & CI/CD
Business Value	Frictionless publishing from CI
Priority / Estimate	Priority: Must SP: 3
Persona	maintainer
Dependencies	GitHub Actions; Pages
Assumptions / Risks	Cache misses slow builds

Story *As a maintainer, I want to Build & Deploy to GitHub Pages so that Frictionless publishing from CI.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Cache emsdk/CMake; build Release; upload artifact; deploy on tag; correct public path.
- AC:** Tag push publishes Pages; caches used on subsequent builds.

K1 — CSP & Strict MIME

Epic / Feature	Security
Business Value	Mitigate injection and loading risks
Priority / Estimate	Priority: Must SP: 3
Persona	security engineer
Dependencies	HTTP headers
Assumptions / Risks	Over-strict CSP blocks assets

Story *As a security engineer, I want to CSP & Strict MIME so that Mitigate injection and loading risks.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set.

Done: All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Define CSP (report-only first); then enforce; ensure `application/wasm` and script MIME; add SRI if applicable.
- Enable `crossOriginEmbedderPolicy` when threads are used.
- AC:** CSP enforced without breaking; correct MIME types served.

L1 — Onboarding & Troubleshooting Guide

Epic / Feature	Docs & DX
Business Value	New contributor builds in \leq 10 minutes
Priority / Estimate	Priority: Must SP: 2
Persona	new contributor
Dependencies	Screenshots; links
Assumptions / Risks	Setup friction stalls adoption

Story *As a new contributor, I want to Onboarding & Troubleshooting Guide so that New contributor builds in \leq 10 minutes.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repository and pipeline configuration are available
When	the user completes the <i>Hands-on Objective</i> for this chapter
Then	the stated <i>Outcome</i> for this chapter is observable and recorded in the pipeline/job summary

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed flagged.

Tasks

- Write step-by-step: emsdk install, emcmake flow, common flags.
- Include common errors/fixes; link to Emscripten/WebGL2 references.
- AC:** Following the guide yields a successful build and run locally.