below is the end-to-end plan to introduce an Azure DevOps CICD pipeline for every project the dev team maintains.  
The same template will be applied to each repository; only environment names and IIS site names change.

1. STANDARD PIPELINE STAGES (applies to every project)
2. Build  
   • Restore, compile, unit-test, security-scan, package  
   • Artifact produced: drop-{SemVer}-build.{BuildId}
3. Deploy → ****IIS-Test**** (non-production)  
   • Automatic deployment after every merge to main  
   • Smoke tests run
4. Manual Approval Gate → ****IIS-Staging****  
   • CAB group notified by e-mail / mobile app  
   • Approval window = 4 hours; timeout = reject
5. Deploy → ****IIS-Production**** (identical package)  
   • Zero-downtime blue-green switch via “Take App Offline”  
   • Post-deployment health-check
6. APPROVAL WORKFLOW

• Environment “IIS-Staging” has an ****Approvals & Checks**** rule:  
CAB members = “[DevOps-CAB@company.com](mailto:DevOps-CAB@company.com)”  
Required votes = 1 out of 3  
Escalation after 4 h = automatic reject & alert manager

• Emergency hot-fix path (break-glass):  
Tag branch “hotfix/\*” → pipeline bypasses staging approval and goes straight to production after manager single-approval via Teams bot.

1. ROLLBACK PLAN

A. Rollback Trigger  
• Health-check fails OR  
• > 5 % error rate in 5 min window after prod deploy OR  
• Manual rollback request via Teams channel “#prod-rollback”

B. Rollback Steps (≤ 5 min target)

1. Pipeline “Rollback-{Project}” is triggered automatically or manually
2. Swaps IIS site back to previous deployment slot (kept warm)
3. Database: run “Down” migration script stored in repo /scripts/rollback
4. Notifies CAB & manager via e-mail & Teams

C. Rollback Verification  
• Same smoke tests executed on rolled-back version  
• Ticket created in backlog to fix forward

1. TIMELINE (per project)

Week 0 – Kick-off & repo inventory (½ day)  
Week 1 – Create build stage + unit-test gate (2 days)  
Week 1-2 – Set up IIS-Test environment + auto-deploy (3 days)  
Week 2 – Create IIS-Staging environment + CAB approvals (2 days incl. CAB training)  
Week 3 – Create IIS-Production environment + rollback job (2 days)  
Week 3 – Dry-run a dummy release + rollback drill (½ day)  
Week 4 – Go-live, monitor first 3 prod releases (buffer 2 days)

Total ****7-9 working days per project**** (parallel tracks possible).

1. PARALLEL EXECUTION PLAN (across projects)

Week 0 – Identify 6 active projects  
Week 1 – Build stage for all 6 (shared template)  
Week 2 – IIS-Test for 3 projects, build stage for the other 3  
Week 3 – IIS-Staging for first 3, IIS-Test for second 3  
Week 4 – IIS-Production for first 3, IIS-Staging for second 3  
Week 5 – Complete second 3 → all 6 live

1. RISKS & MITIGATIONS

• CAB overload → stagger go-lives, rotate reviewers  
• Rollback script missing → each team must supply before Week 3  
• Secret leaks → migrate to Azure Key Vault + variable groups

Please let me know if you’d like a Gantt chart or budget estimate.

Best regards,