

AZURE DEVOPS TO GITHUB MIGRATION PLAYBOOK

Unlocking Agentic DevOps



vighnesh-
notes

vighneshnaik.com

[linkedin.com/in/
vighneshnaik6](https://linkedin.com/in/vighneshnaik6)



WHY MIGRATE?



Unified developer platform — code, CI/CD, security, and collaboration in one place.



AI-native & agentic — work with GitHub Copilot, coding agents, models, and GitHub Agents.



Open ecosystem — best-of-breed tools and a vibrant marketplace.



Faster innovation — ship secure software faster at scale.



Cost & simplicity — consolidate, optimize, and modernize.

FROM



Azure DevOps



TO



GitHub

PLAYBOOK OVERVIEW



1. ASSESS



2. PLAN



3. PREPARE



4. MIGRATE



5. VALIDATE



6. OPTIMIZE
& ADOPT



SUCCESS OUTCOME

A modern, secure, AI-powered DevOps platform that accelerates delivery and empowers teams!



Keep It Simple. Design for Scale. Secure by Design. Cost Optimized.





2. ASSESS YOUR CURRENT STATE

WHAT TO ASSESS



Projects & Repos —
Git repos, branches,
policies, wiki, size



Pipelines & Releases —
build/release pipelines,
templates, variables,
approvals



Artifacts & Packages —
feeds, artifacts,
container registries



Boards & Work Items —
projects, boards,
process, queries



Test Plans & Test Cases —
manual, automated,
test data



Service Connections —
environments, agents,
variable groups



Extensions & Integrations —
third-party tools,
marketplace extensions

ASSESSMENT ACTIVITIES

- Inventory everything
- Identify dependencies & critical paths
- Classify: Must Have / Nice to Have
- Evaluate usage & value
- Identify risks & gaps
- Define migration priority



OUTPUT

- ✓ Current state report
- ✓ Migration scope & boundaries
- ✓ Tool parity analysis
- ✓ Gap & risk register
- ✓ High-level migration roadmap



TIP

Start with an iterative approach.
Migrate in waves, not big-bang!





3. PLAN YOUR MIGRATION

DESIGN YOUR FUTURE STATE



Choose GitHub Cloud
or GitHub Enterprise Cloud



Define org structure,
teams & repositories



Standardize branching
strategy & policies



Select CI/CD strategy
(GitHub Actions +
Azure Pipelines where
needed)



Plan security, identity
& access model

MIGRATION STRATEGY



Phased / Wave –
migrate projects
in phases



Parallel Run –
run both platforms
until confidence



Big Bang –
migrate all at once
(high risk)



Pilot First –
prove with a
limited scope

BUILD YOUR MIGRATION ROADMAP

SHORT TERM (0-30 days)

- Pilot project
- Core repos
- CI/CD pipelines

MID TERM (31-90 days)

- More projects
- Integrations
- Knowledge transfer

LONG TERM (90+ days)

- Full adoption
- Optimize & scale
- Decommission ADO

**TIP**

| Start with a pilot team/project. Learn, adapt, and scale.



Keep It Simple. Design for Scale. Secure by Design. Cost Optimized.





4. PREPARE FOR MIGRATION

SET UP GITHUB FOUNDATION

- ✓ Create organization & teams
- ✓ Configure SSO, SCIM, MFA
- ✓ Set up repositories & permissions
- ✓ Define branch protection & required reviews
- ✓ Configure environments & deployment protection rules
- ✓ Set up secrets management (GitHub Secrets / OIDC)
- ✓ Set up GitHub Actions runners (hosted or self-hosted)

MAPPING GUIDE

Azure DevOps	→	GitHub
Projects	→	Organizations / Teams
Repos	→	Repositories
Pipelines (Build/Release)	→	GitHub Actions or Azure Pipelines
Artifacts	→	GitHub Packages
Service Connections	→	Environments + Secrets
Variable Groups	→	Secrets / Variables
Boards / Work Items	→	GitHub Issues / Projects or retain Azure Boards
Wiki	→	GitHub Wiki

TOOLS & ACCELERATORS



GitHub Importer
for Azure DevOps



REST / GraphQL
APIs



CLI &
automation scripts



Migration spreadsheets /
trackers



TIP

Automate the boring stuff.
Repeatability reduces risk!



Keep It Simple. Design for Scale. Secure by Design. Cost Optimized.





5. MIGRATE WORKLOADS

1.

CODE
REPOSITORIES

- Use GitHub Importer for Azure DevOps
- Migrate repos, branches, tags
- Preserve commit history
- Validate & clean up

2.

PIPELINES
& CI/CD

- Convert YAML pipelines
- Map build/release to GitHub Actions or retain Azure Pipelines
- Migrate templates & variables
- Recreate approvals & gates
- Test workflows

3.

ARTIFACTS
& PACKAGES

- Migrate feeds to GitHub Packages
- Container images to GHCR
- Migrate pipeline artifacts
- Verify integrity

4.

WORK ITEMS
& BOARDS

- Export & migrate work items
- Map fields & relationships
- Configure GitHub Projects
- Validate workflows

5.

TESTS
& QUALITY

- Migrate test cases & plans
- Integrate test results into Actions
- Set up code scanning & quality gates
- Validate results



DOCUMENT EVERYTHING

Runbooks, how-tos, mappings, troubleshooting guides.



Keep It Simple. Design for Scale. Secure by Design. Cost Optimized.





6. VALIDATE & ENSURE QUALITY

VALIDATION CHECKLIST

- ☒ Repository content & history
- ☒ Builds & workflows
- ☒ Artifacts & packages
- ☒ Environments & deployments
- ☒ Approvals & gates
- ☒ Security & access
- ☒ Work items & workflows
- ☒ Integrations & webhooks
- ☒ Monitoring & alerts

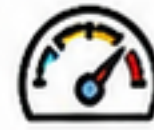
TEST & VERIFY



Run end-to-end scenarios



Compare results



Performance testing



Security validation



User acceptance testing



Sign-off from stakeholders

SUCCESS CRITERIA



- All key workflows working
- Teams are productive
- No critical issues
- Go / No-Go decision



TIP — Don't skip validation. Confidence comes from testing!

7. OPTIMIZE & ADOPT AGENTIC DEVOPS



vighnesh-
notes

vighneshnaik.com

linkedin.com/in/vighneshnaik6

OPTIMIZE



Refactor & simplify workflows



Leverage reusable actions & templates



Enable caching & parallelism



Cost & performance optimization



Decommission unused resources

EMBRACE AGENTIC DEVOPS WITH GITHUB



GitHub Copilot
AI pair programmer
in your IDE & PRs



Security & Quality
advanced code scanning,
secret scanning, Dependabot



GitHub Agents
automate tasks across
issues, PRs & workflows



Automation
create bots & workflows
that act on events



Code Review
summaries and
AI-powered insights
for faster reviews



Metrics & Insights
DORA metrics, deployment
frequency, lead time

ADOPTION BEST PRACTICES



Train teams



Share knowledge



Build champions



Iterate & improve



Celebrate wins



Keep It Simple. Design for Scale. Secure by Design. Cost Optimized.



8. MIGRATION SUCCESS CHECKLIST

VALIDATION CHECKLIST

- ☒ Assessment completed & roadmap approved
- ☒ GitHub foundation ready
- ☒ Pilot migration completed
- ☒ Repositories migrated
- ☒ Pipelines migrated & validated
- ☒ Artifacts & packages migrated
- ☒ Work items & boards migrated
- ☒ Security, access & compliance validated
- ☒ Teams trained & onboarded
- ☒ Decommission Azure DevOps resources
- ☒ Celebrate & iterate

KEY PRINCIPLES



People First —
empower teams, communicate
clearly, support change



Secure by Design —
zero trust, least privilege,
continuous verification



Automate Everything —
automate build, test, deploy
and repetitive tasks



Measure & Improve —
data-driven decisions,
continuous improvement



MIGRATE. MODERNIZE. INNOVATE.
Build the future with GitHub!



☆ ☆ ☆
Think architecture.
Think Azure!
You can do it!