GitHub @ Penn State: What IT Pros Need to Know

Gretchen Kuwahara

First: A quick exercise

Let's signup for GitHub Enterprise!

If you haven't already signed up:
1. 🔍 Go to softwarerequest.psu.edu
2. Search for GitHub
3. Click the green "Order" button
4. Review the terms and click "Order"
5. ⏳ Wait for account provisioning

After that, follow the invite in your inbox to log in at github.com using your Penn State SSO!

Agenda

- What is Git and how to use it?

- About GitHub Enterprise Cloud

- Best Practices

- Automation

- Migrating to GitHub

- Resources for you and your customers

Sign up for
GitHub Enterprise

What is Git and how to use it?

What is Git?

Git is a version control system, that helps:

• Tracking code changes

• Tracking who made changes

• Coding collaboration

It's like a time machine & team organizer for your code

Git Terminology and Commands

Term	Command	Definition
Repository	`git init`	A folder where Git tracks your project and its history.
Clone	`git clone <URL>`	Make a copy of a remote repository on your computer.
Stage	`git add <filename>`	Tell Git which changes you want to save next.
Commit	`git commit –m "change"`	Save a snapshot of your staged changes.
Branch	`git branch my-branch`	Work on different versions or features at the same time.
Merge	`git merge <branch>`	Combine changes from different branches.
Pull	`git pull`	Get the latest changes from a remote repository.
Push	`git push`	Send your changes to a remote repository.
Log/History	`git log`	Detailed record of every change made to the project

Note: Most Git actions (like staging, committing, and viewing history) happen on your own computer.

About GitHub Enterprise Cloud

What is GitHub?

GitHub is a platform for hosting, sharing, and collaborating on code. It's built around **Git,** a version control system that tracks changes to files over time.

With GitHub you can:

- Store you code in repositories

- Collaborate with others via branches and pull requests

- Review and discuss code with comments

- Automate builds and testing

GitHub Uses

- 👨‍💻 Software Development

- 📚 Instruction & Coursework

- 🧪 Research Collaboration

- 🔄 Automation & Workflows

- 🌐 Publishing & Websites

- 👬 Team Collaboration

Penn State's GitHub Enterprise Cloud

GitHub Enterprise is GitHub tailored for organizations, offering advanced features for security, compliance, and scalability.

Key features:
- 🔐 Single Sign-On (SSO) with Penn State credentials
- 👩‍💼 Available to Faculty, Staff, and Students 🆕
- 🛡️ Enterprise-level security & compliance controls
- 🚀 Unlimited private repositories
- 🤝 Team and organization management for better collaboration

GitHub Enterprise Terminology

Term	Definition
Organization	A shared account for managing multiple repositories and users (e.g., a department or team).
Team	A group of users within an organization with defined repository access. Can be an UMG or Azure Group
Collaborator	A person with access to a repository. Can be internal (within the org) or external.
Repository (Repo)	A project space on GitHub that contains your code, issues, settings, and more.
Pull Request (PR)	A GitHub feature for reviewing and merging code from one branch into another.
Actions	GitHub's built-in automation platform for CI/CD and workflow automation.
GitHub Pages	A feature to host static websites directly from a repo.
GitHub Projects	A kanban-style board for managing issues, PRs, and tasks—now built on GitHub Issues.

🏢 GitHub Organizations

Shared Collaboration Spaces

- Enable team-based work across multiple repositories with enhanced security and admin tools.

Default Access

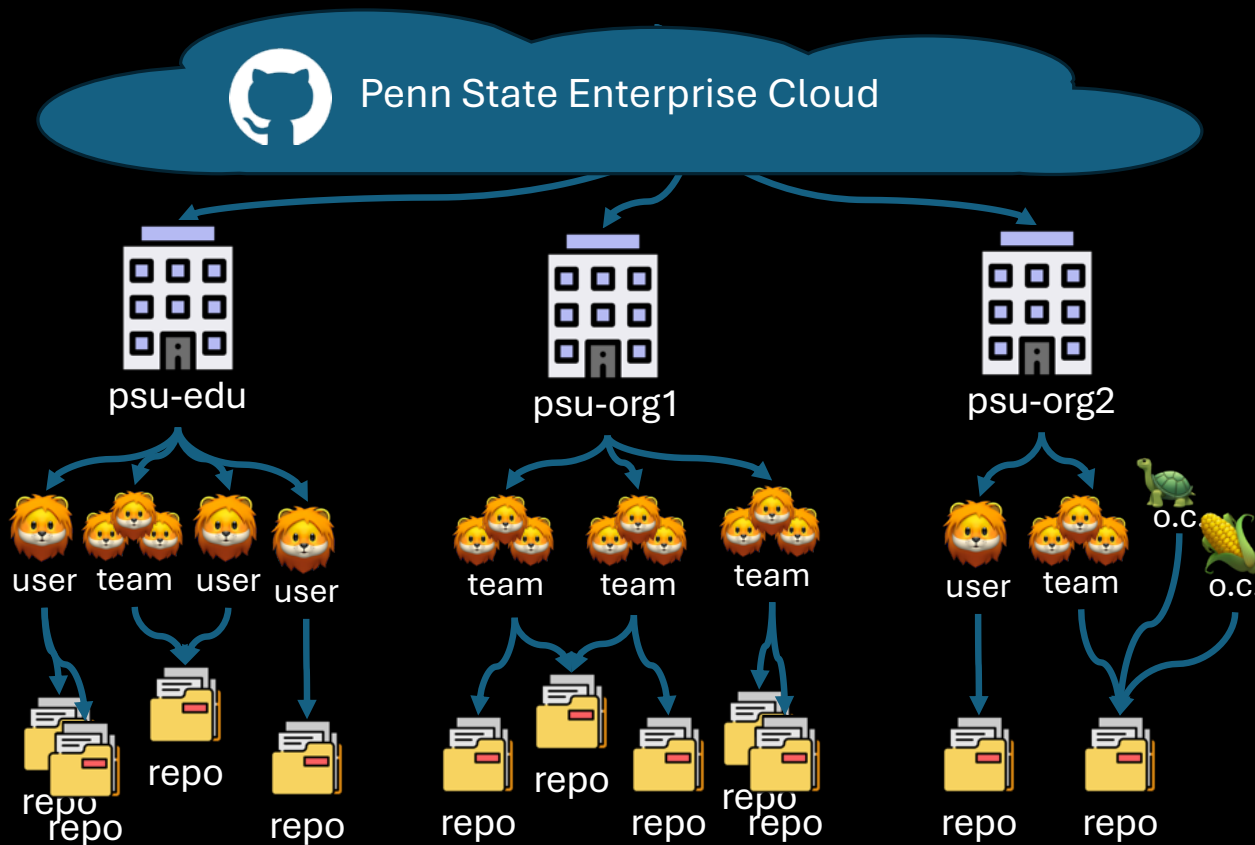- All Penn State users are auto-invited to psu-edu, our default GitHub Organization.

Need More Features?

- Departments, projects, or research groups can request their own Organization with added capabilities.

Need an Organization for your unit/group? Fill out a Request!

🔓 Security Features

Feature	Description	Availability
Dependabot Alerts & Updates	Scans for vulnerable dependencies and alerts users; Opens PR to update it	✅ Public & private repos
Branch Protection Rules	Require PR review, status checks, signed commits, etc.	✅ Public & private repos
Secret Scanning Alerts	Scans for exposed credentials (e.g., API keys)	✅ Public; ❌ Private
Security Overview Dashboard	Centralized dashboard of alerts and scanning status	✅ Public; ❌ Private
Audit Logs	Track activity like repo changes, access events, and user logins	✅ Public & private repos
IP Allow Lists	Restrict access to GitHub to specific IP ranges	✅ Public & private repos
Token & SSH Key Expiration Policies	Enforce expiration for credentials	✅ Public & private repos

Legend: ✅ Included in our service ❌ Requires GitHub Advanced Security

💰 Any costs for Penn State GitHub Service?

Feature	Public Repos	Internal/Private Repos	Notes
Basic Git Functionality	✅Unlimited	✅Unlimited	
GitHub Pages and Projects	✅	✅	
Actions	✅ 2,000/mo.	⚠️ Limited	
Dependabot Alerts	✅	✅	
Secret Scanning	✅	❌only with Advanced Security	Own 🏢 only
Large File Storage	💰 Add-on cost	💰 Add-on cost	Own 🏢 only
CodeSpaces	💰 Add-on cost	💰 Add-on cost	Own 🏢 only
GitHub CoPilot	💰 Personal subscription	💰 Business subscription	Own 🏢 only

Legend: ✅ Included ❌ Not available ⚠️ Limited 💰 Available for an **extra cost** 🏢 GH Organization

🤖 GitHub Copilot at Penn State

Important Notes
- Not the same as Microsoft Copilot or Copilot for Microsoft 365
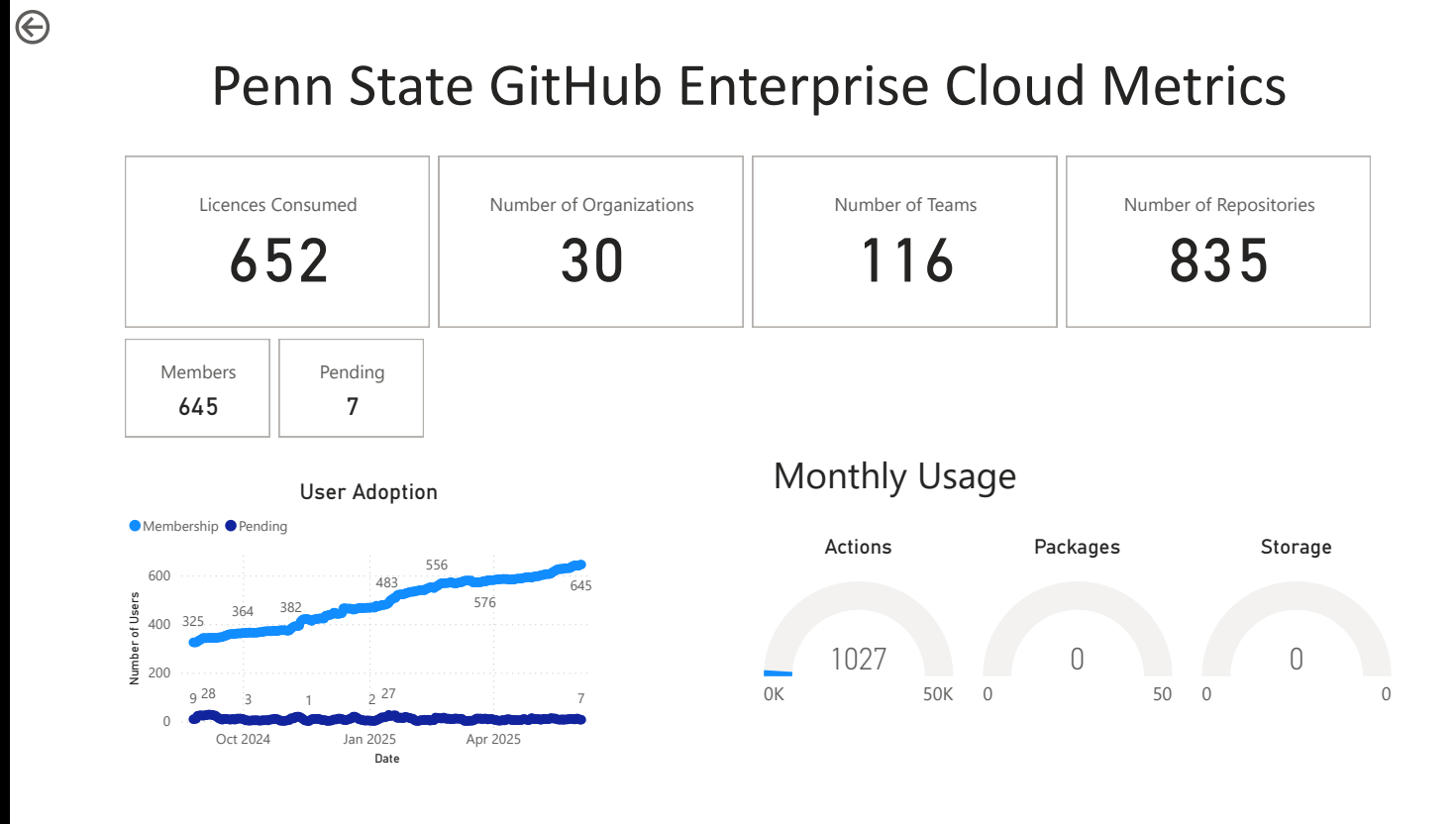- Only applies to GitHub.com Copilot features

Eligibility & Access
- Faculty & Students: Free access available for verified users
- Staff: 30-day free trial
- Paid license requires supervisor approval and departmental funding

Licensing Info
- Only Copilot Individual is supported
- Copilot Business & Enterprise not available currently for Penn State

Pro Tip: After activation, disable data sharing in Copilot Privacy Settings

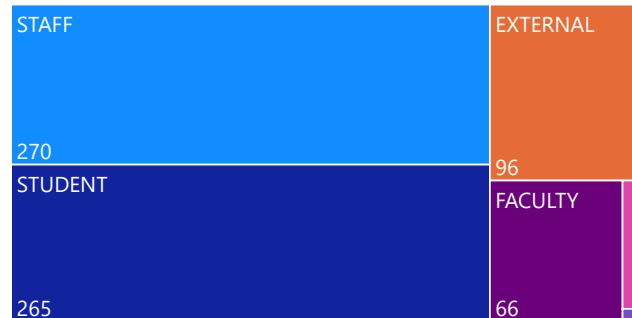# Who at Penn State is using it today?

As of 6/3/2025

Who at Penn State is using it today?

As of 6/3/2025

Best Practices

📦 Repositories

- **Types**: 🌍 Public | 🦁 Internal | 🔒 Private
- **Access**: Invite teams or users, set roles (Read–Admin)
- **Protect branches**: Use review rules on main
- **Use teams**: Easier than managing individual access
- **Name wisely**: Clear, consistent repo naming
- **Add basics**: README.md, .gitignore

👥 Collaboration

Within Penn State

- Ensure collaborators have an active Penn State GitHub account
- Invite them to the appropriate organization (e.g., psu-pennstateit)
- Add them to a team for scoped access and easier management

External Collaborators

(e.g., visiting faculty, vendors, sponsored accounts)

- Can only be invited at the individual repository level
- Cannot be added to teams or organization-wide roles
- Access should be reviewed regularly and limited to what's needed

Automation

⚙️ What is CI/CD?

CI/CD = Continuous Integration + Continuous Deployment/Delivery

- **CI**: Automatically build & test code when changes are made
- **CD**: Automatically deploy code to production/staging

Enables **faster feedback**, **fewer errors**, and **repeatable workflows**

🤖 GitHub Actions – Automation Engine

GitHub Actions lets you:

- Automate **builds, tests, deployments**, and more
- Trigger workflows on **push, pull request, schedule, or custom events**
- Use **prebuilt actions** or write your own

✅ Fully integrated into GitHub

✅ Supports matrix builds, containers, secrets, and self-hosted runners

🔁 Common CI/CD Use Cases

- **Run unit tests** on pull requests
- **Lint and format code** before merging
- **Build and publish** Docker containers
- **Deploy to cloud** (e.g., AWS, Azure, GCP, Netlify, Firebase)
- **Send alerts** to Teams/Slack/Discord
- **Schedule nightly jobs** or cleanup tasks

📦🚚 Migrating to GitHub

🔄 Typical Migration Workflow

1. Inventory existing repositories
2. Clean up old or inactive projects
3. Export & import repositories
4. Recreate workflows (CI/CD)
5. Map users & permissions
6. Test & validate
7. Decommission the original repository/project

🧰 Tools that Help

- Git Commands
  - Create an empty repo in GitHub
  - Get the GitHub clone URL
  - Add your local Git repo origin w/ URL
  - Git push local repo to new origin

- GitHub Importer
- GitHub CLI

```
# Update your local Git repo to point to the
new GitHub URL
git remote set-url new-origin
https://github.com/your-org/your-repo-name.git

# Push your local repository to GitHub
git push -u new-origin main
```

Resources for you
and your customers

🧠 Knowledge Base Articles

Over 30 articles in Penn State's Knowledge Base!

- Signing up for Service - KB0019812

- Getting Started - KB0020754

- Frequently Asked Questions - KB0019927

- For IT - KB0020773

Several articles on migrating from GitLab to GitHub too!
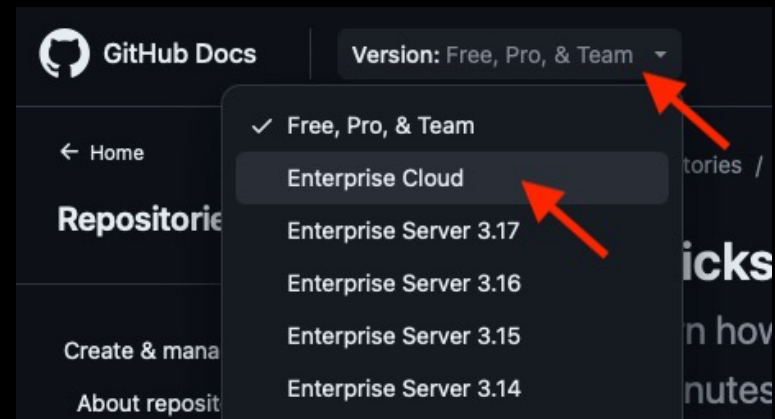
📚 Trainings

LinkedIn Learning – linkedinlearning.psu.edu

• Git Essential Training

• Learning Git and GitHub

GitHub Skills - https://skills.github.com

GitHub Documentation

https://docs.github.com

Pro Tip: Make sure you select "Enterprise Cloud" for version.

🙋 Getting Help

🧾 **Open a GitHub Support Ticket**

- Visit [GitHub Support](#) → Log in → *My Tickets* → *New Ticket*
  Choose personal or organizational context and submit your issue.

🏛️ **Access to a Penn State GitHub Organization**

- Visit http://github.psu.edu and email the org admins listed for access.

💬 **Contact the Penn State GitHub Team**

- Submit a request through the Code Repository Service help form.

Presentation Resources and Questions

Thanks for coming!

Questions?

Presentation slides and links available at:

https://github.com/psu-github/2025-GitHub-Presentation-Resources