



Network Automation Pipelines – From Zero to Hero

Jon Howe: Principal Solution Architect



Agenda

Why people-centric automation matters

Pipeline at a glance

Core stages deep dive

Live Demo



The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency.

- Bill Gates

Who is your audience?

- Understand how your consumers do their job today?
 - Network Equipment
 - Ecosystem (Secrets, IPAM, CMDB)
- Humans need trust + visibility
- No tickets? No proof. No audit. No compliance.

Pain

- Multiple handoffs, messy approvals
- Drift between “intent” and “reality”
- No traceability -> no trust

Promise

- Centralized requests and governance (Jira)
- Inventory validation (Nautobot)
- Automated audit trail (GH Actions)
- Faster, safer changes

The background is a dark blue gradient. It features several concentric circles of a lighter blue shade, centered around the text. There are also some abstract, curved, light blue shapes that resemble stylized waves or brushstrokes scattered across the background.

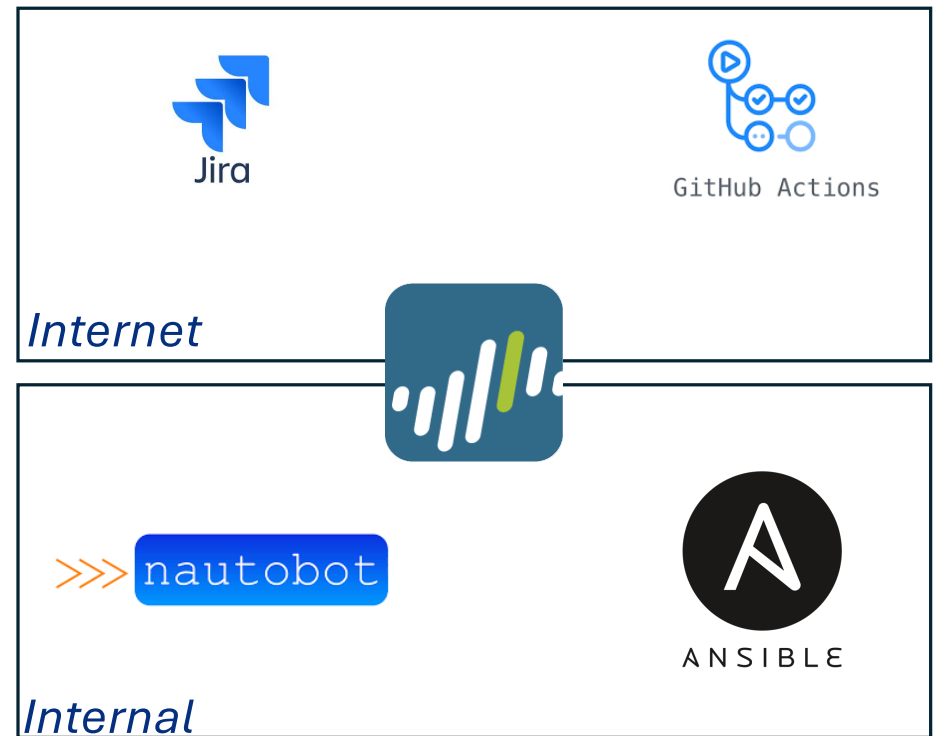
Pipeline At A Glance

End to End Architecture

1. **User Needs a change**
2. Jira: Request Details and approval
3. GitHub Actions: orchestrates
4. Ansible: executes
5. Nautobot: inventory
6. Palo Alto: enforces
7. GitHub Actions Updates Ticket

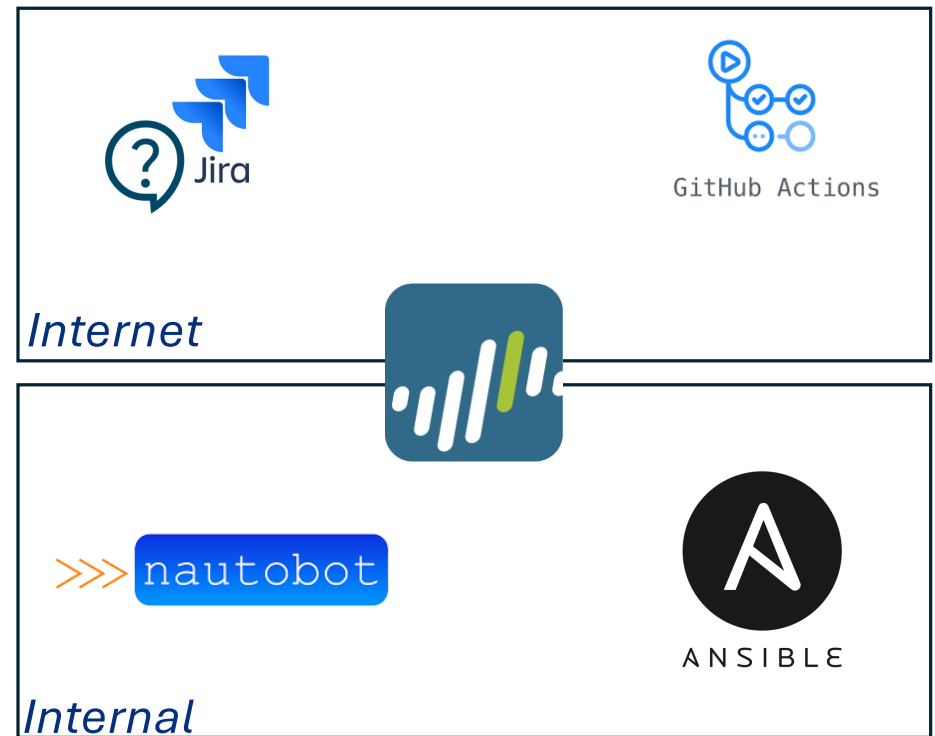


I need to
access an
app, help!



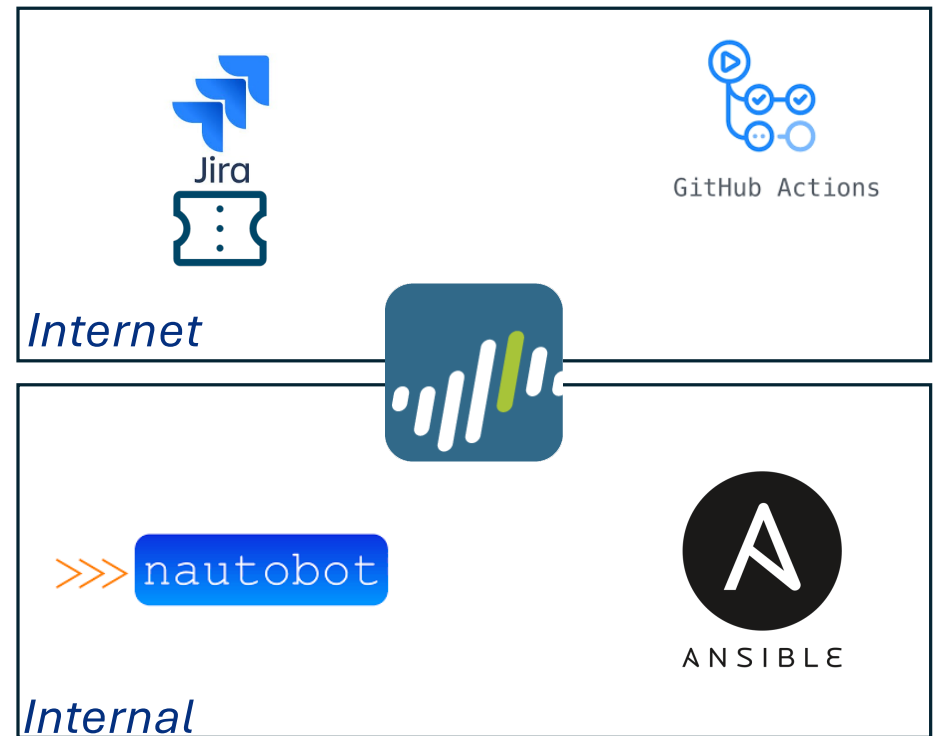
End to End Architecture

1. User Needs a change
2. **Jira: Request Details and approval**
3. GitHub Actions: orchestrates
4. Ansible: executes
5. Nautobot: inventory
6. Palo Alto: enforces
7. GitHub Actions Updates Ticket



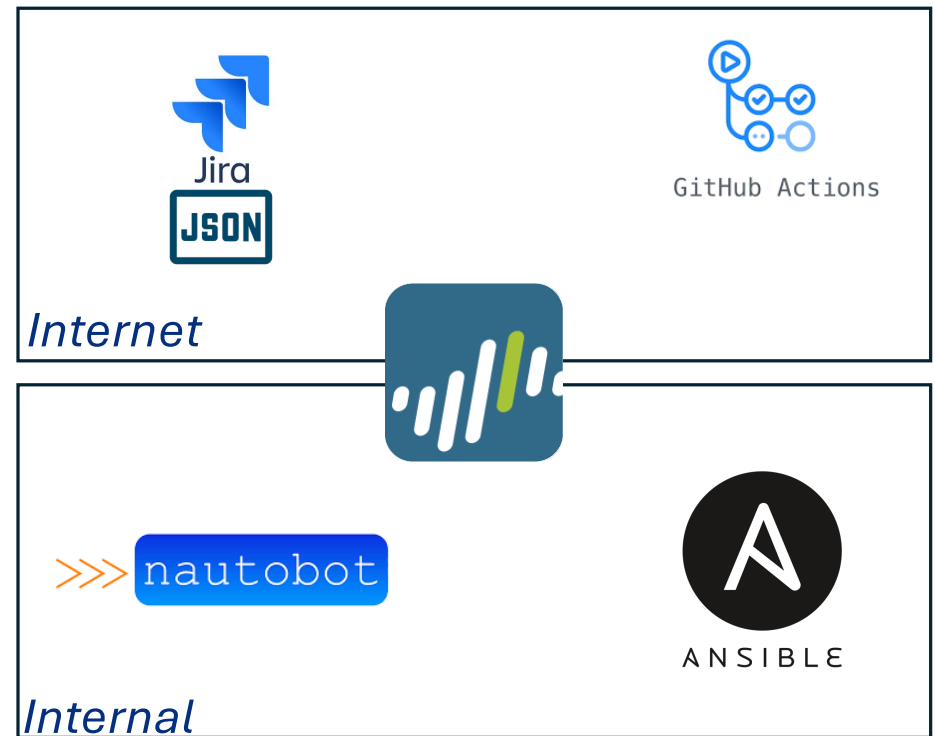
End to End Architecture

1. User Needs a change
2. **Jira: Request Details and approval**
3. GitHub Actions: orchestrates
4. Ansible: executes
5. Nautobot: inventory
6. Palo Alto: enforces
7. GitHub Actions Updates Ticket



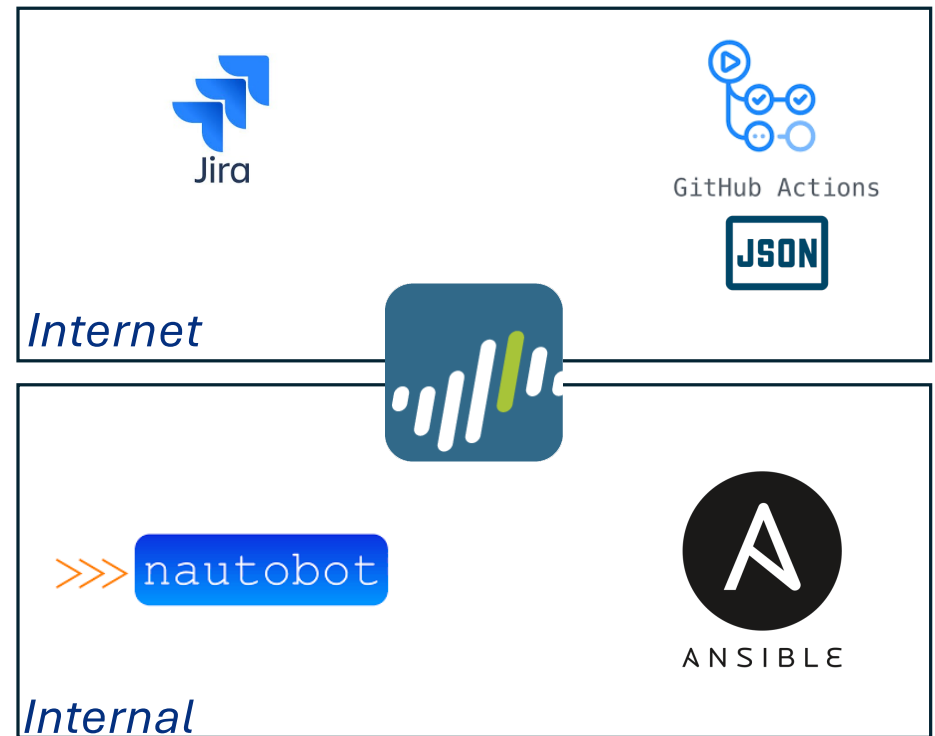
End to End Architecture

1. User Needs a change
2. **Jira: Request Details and approval**
3. GitHub Actions: orchestrates
4. Ansible: executes
5. Nautobot: inventory
6. Palo Alto: enforces
7. GitHub Actions Updates Ticket



End to End Architecture

1. User Needs a change
2. Jira: Request Details and approval
3. **GitHub Actions: orchestrates**
4. Ansible: executes
5. Nautobot: inventory
6. Palo Alto: enforces
7. GitHub Actions Updates Ticket

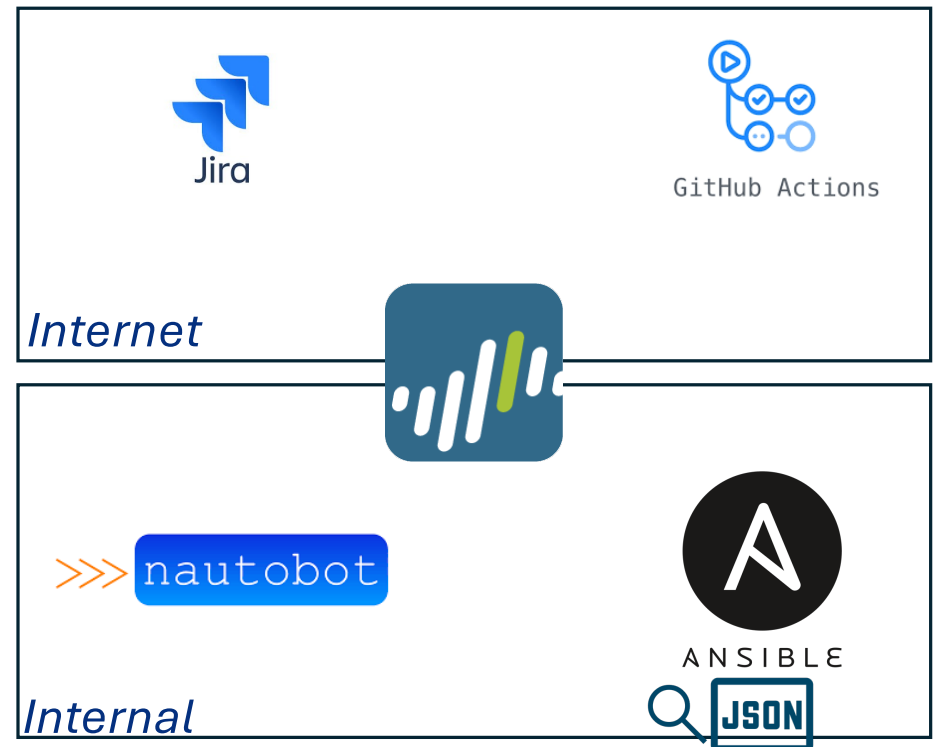


End to End Architecture

1. User Needs a change
2. Jira: Request Details and approval
3. GitHub Actions: orchestrates
4. **Ansible: executes**
5. Nautobot: inventory
6. Palo Alto: enforces
7. GitHub Actions Updates Ticket



I need to
access an
app, help!

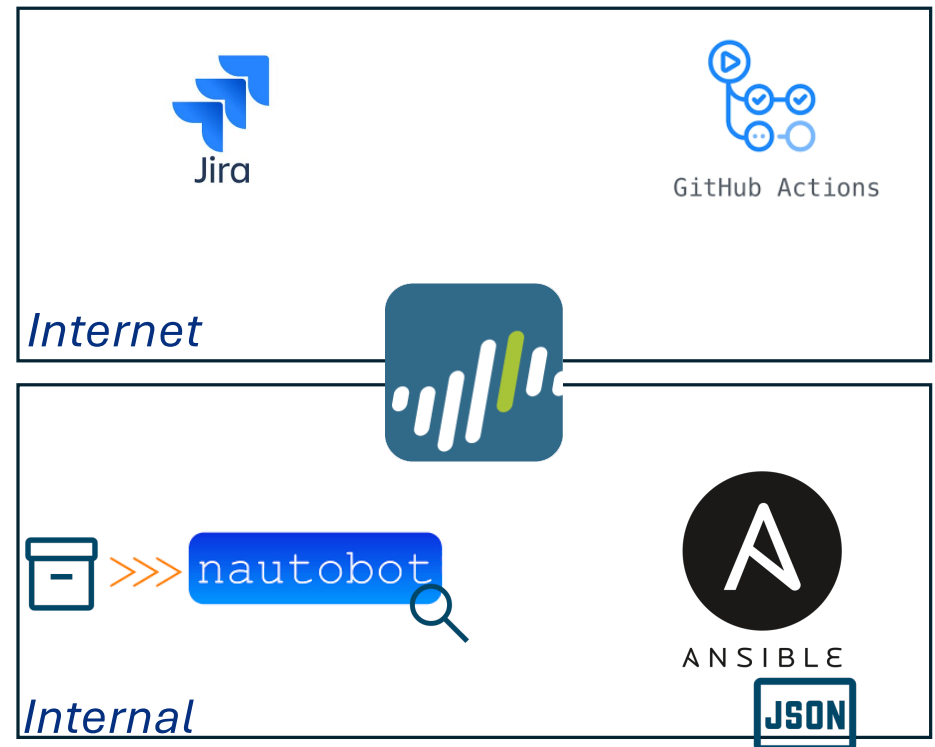


End to End Architecture

1. User Needs a change
2. Jira: Request Details and approval
3. GitHub Actions: orchestrates
4. **Ansible: executes**
5. **Nautobot: inventory**
6. Palo Alto: enforces
7. GitHub Actions Updates Ticket



I need to
access an
app, help!

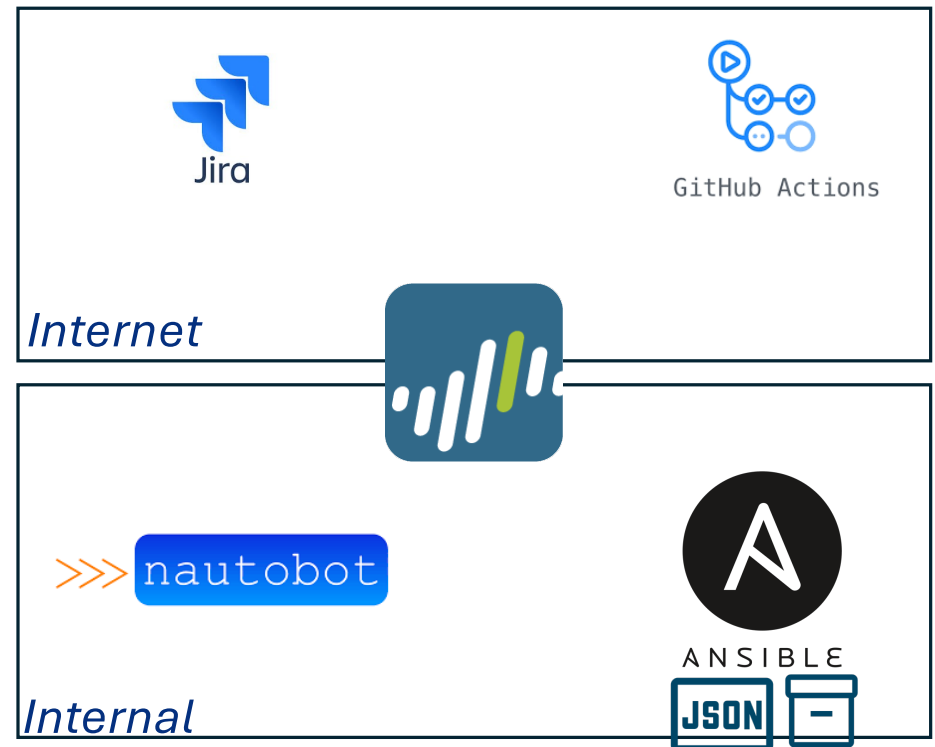


End to End Architecture

1. User Needs a change
2. Jira: Request Details and approval
3. GitHub Actions: orchestrates
4. **Ansible: executes**
5. **Nautobot: inventory**
6. Palo Alto: enforces
7. GitHub Actions Updates Ticket



I need to
access an
app, help!

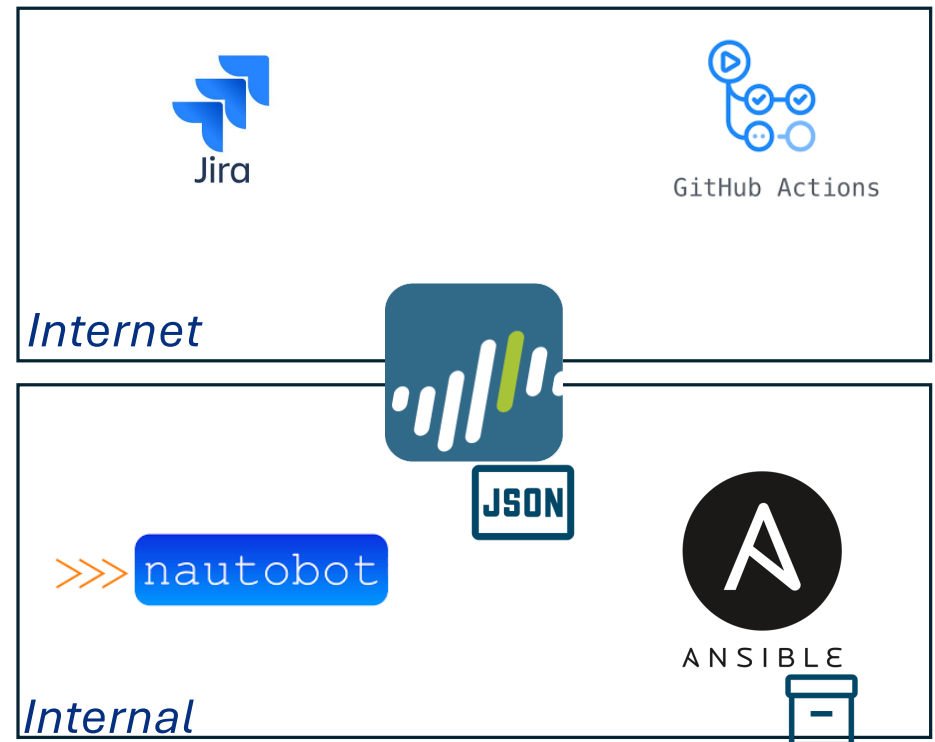


End to End Architecture

1. User Needs a change
2. Jira: Request Details and approval
3. GitHub Actions: orchestrates
4. **Ansible: executes**
5. **Nautobot: inventory**
6. **Palo Alto: enforces**
7. GitHub Actions Updates Ticket

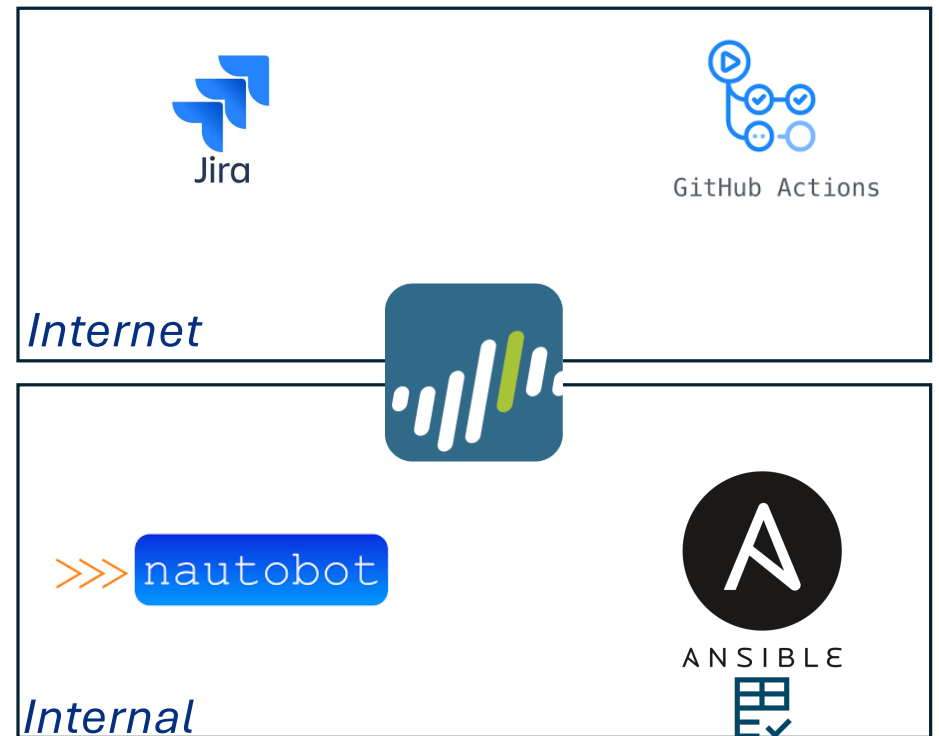


I need to
access an
app, help!



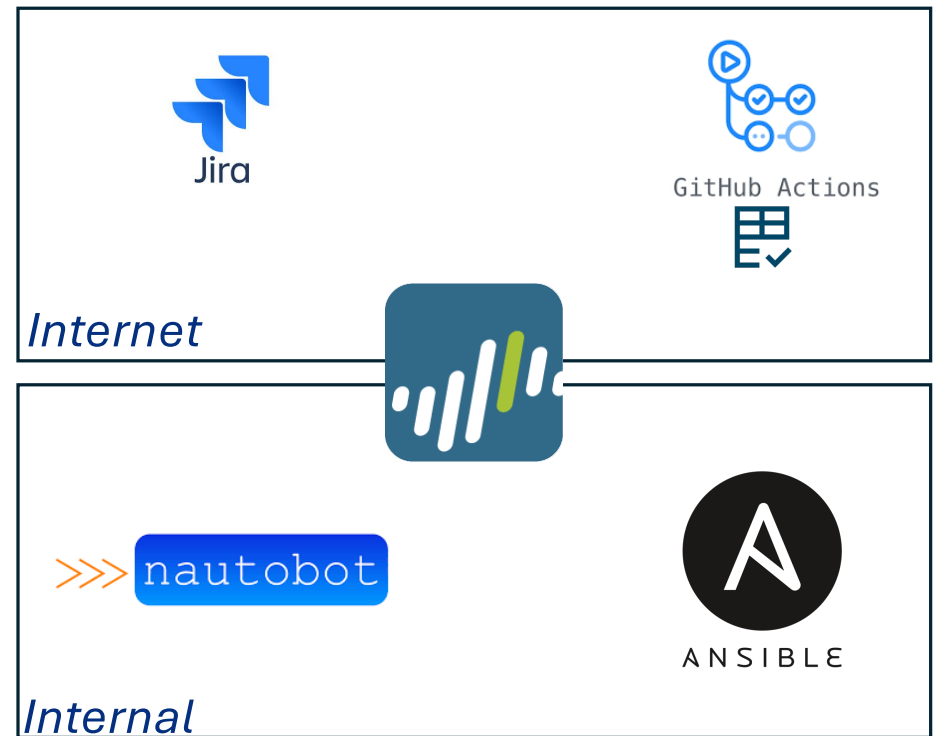
End to End Architecture

1. User Needs a change
2. Jira: Request Details and approval
3. GitHub Actions: orchestrates
4. Ansible: executes
5. Nautobot: inventory
6. Palo Alto: enforces
7. **GitHub Actions Updates Ticket**



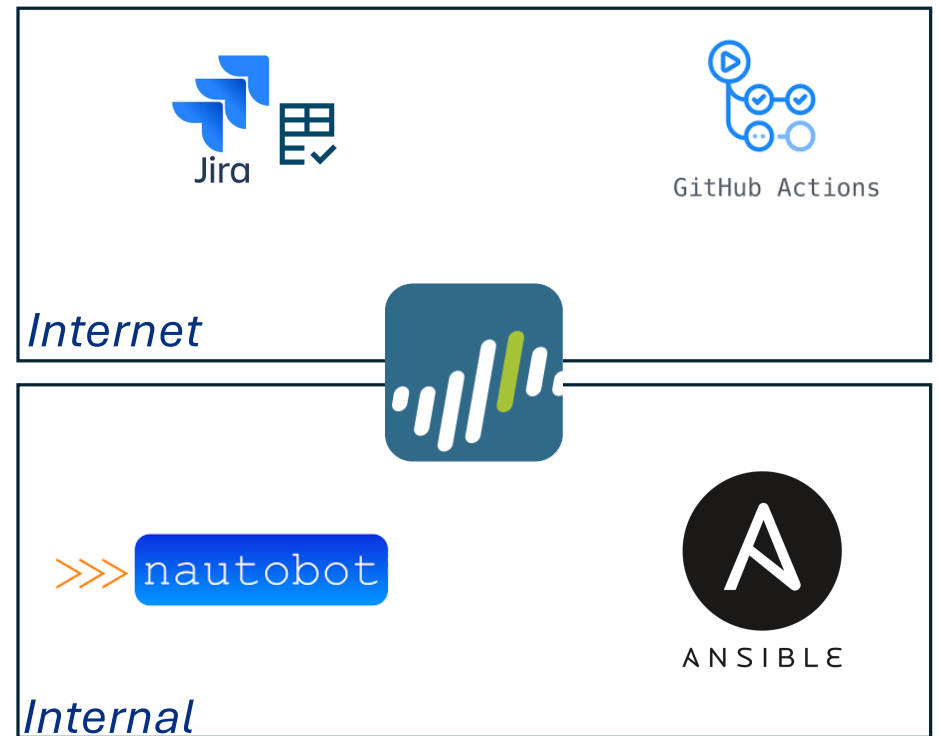
End to End Architecture

1. User Needs a change
2. Jira: Request Details and approval
3. GitHub Actions: orchestrates
4. Ansible: executes
5. Nautobot: inventory
6. Palo Alto: enforces
7. **GitHub Actions Updates Ticket**



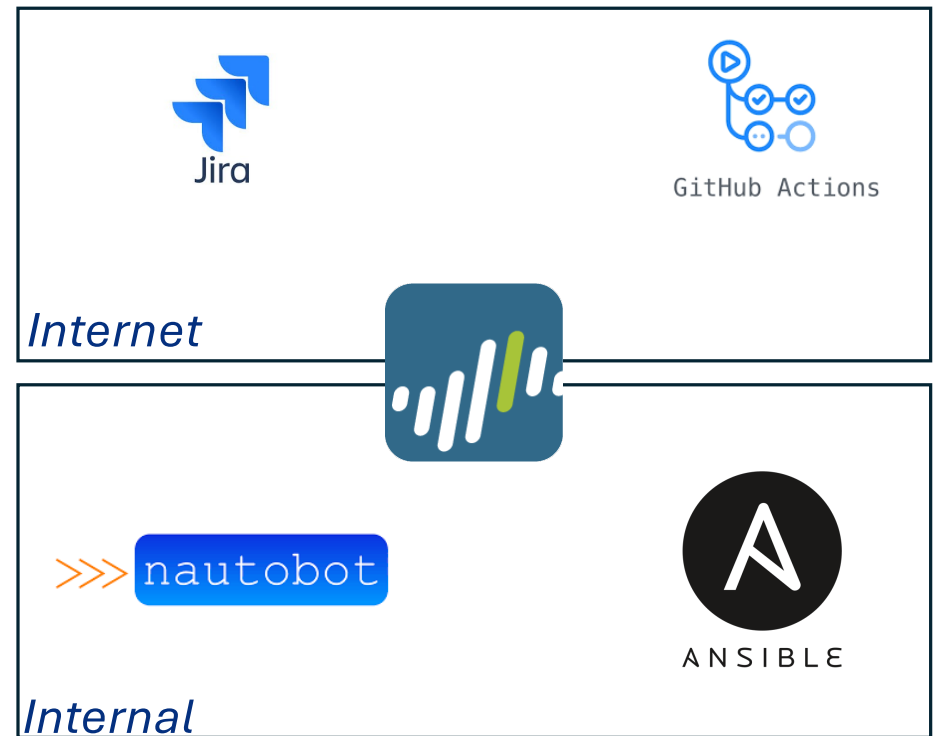
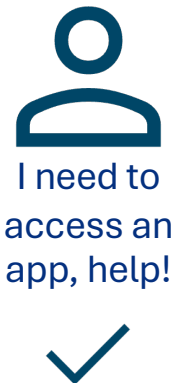
End to End Architecture

1. User Needs a change
2. Jira: Request Details and approval
3. GitHub Actions: orchestrates
4. Ansible: executes
5. Nautobot: inventory
6. Palo Alto: enforces
7. **GitHub Actions Updates Ticket**



End to End Architecture

1. User Needs a change
2. Jira: Request Details and approval
3. GitHub Actions: orchestrates
4. Ansible: executes
5. Nautobot: inventory
6. Palo Alto: enforces
7. GitHub Actions Updates Ticket





Core Stages

Stage 1: Jira Service Management

- Custom form: devices, type, rollback
- Approvals: auto or manual
- Ticket moves to “Ready for Change” status
- Metadata = playbook variables
- Full ticket history for audits

Stage 2: GitHub Actions Orchestration

- Scheduled job polls Jira
- Find “Ready for Change” tickets
- Parse fields and trigger workflows
- Execute Ansible Code
- Update status of Jira ticket

Stage 3: Nautobot Validation + Ansible Execution

- Pull inventory from Nautobot
- Dynamic inventory for Ansible
- Dry-run playbook with diffs
- Make changes to the environment

Stage 4: Configuration Enforcement & Traceability

Enforcement:

- Firewall objects and NAT rules created dynamically using Ansible's Palo Alto modules
- Playbooks executed directly on EC2 instance
- Post-deployment Palo Alto firewall configuration commit with validation

Traceability:

- GitHub Actions logs available for troubleshooting and audit
- Ansible execution results (success/failure, detailed logging) captured clearly
- Automatic Jira ticket state transitions to “Confirmed Deployed” after successful changes

The background is a solid dark blue color. It features several thin, light blue concentric circles centered around the word "Demo". Additionally, there are several organic, wavy, light blue lines that sweep across the background, creating a sense of movement and depth. These lines are more prominent on the left and right sides, framing the central text.

Demo



This site can't be reached

3.135.53.23 took too long to respond.

Try:

- Checking the connection
- [Checking the proxy and the firewall](#)

ERR_CONNECTION_TIMED_OUT

[Details](#)[Reload](#)

The background is a dark blue gradient. It features several concentric circles of varying radii, centered around the text. There are also several abstract, elongated, teardrop-like shapes scattered across the background, some of which are lighter blue than the background, creating a subtle pattern.

Takeaways & Next Steps

Making it Real

What we did and where we're going

Best Practices & Outcomes

- Store secrets in GitHub Secrets
- Linting / Syntax Checks and Dry Run before deployment
- Enforce PR Reviews on Main Branches

What we Gained

- Faster MTTR via Automation
- Better audit & config compliance
- Full lifecycle visibility = user trust

What's Next

- Nautobot: firewall rule DB & intent validation
- Rollback and self-healing workflows
- Analytics for drift and policy insight
- Jira: Include logs

Resources & Getting Started

- **Sample repo – QR Code**
 - GitHub repo (Actions workflows, Ansible playbooks)
 - Nautobot inventory & Jira integration scripts
- **Quick checklist:**
 - Jira project configured & statuses ready
 - GitHub Actions workflow deployed
 - Terraform-built EC2 + Nautobot reachable
 - EC2 connectivity to Palo Alto firewall confirmed
 - Credentials stored securely



The background is a solid dark blue. It features several concentric circles of a lighter blue shade, centered around the text. Additionally, there are several abstract, elongated, teardrop-like shapes in the same lighter blue color, scattered around the circles.

Thank You!