The Combined Power of GitOps and Policy as Code

HOW TO DRIVE SPEED, SECURITY, AND EFFICIENCY IN CLOUD NATIVE ENVIRONMENTS

Brought to you by Akuity and Nirmata

nirmata

Table of Contents

INTRODUCTION

Today's dynamic cloud landscape demands continuous innovation while upholding robust security practices. This white paper explores how GitOps, coupled with Policy as Code (PaC), empowers cloud-native organizations to achieve these goals. By establishing Git as the single source of truth for infrastructure and application configurations, GitOps streamlines operations and accelerates delivery. Policy as Code introduces an automated enforcement layer, ensuring security and compliance policies are ingrained throughout the development lifecycle.

Together, these technologies unlock a path to:

- **Faster Time-to-Market:** Streamlined deployments and automated processes significantly reduce release cycles.
- **Enhanced Developer Experience:** Self-service infrastructure provisioning and familiar Git workflows empower development teams.
- **Operational Efficiency:** Automated reconciliation minimizes configuration drift and ensures consistent environments.
- **Unparalleled Security:** Policy-driven deployments and continuous compliance monitoring drastically reduce risks.

This white paper dives into the technical aspects of GitOps and PaC, explores their combined value proposition, and showcases how they address critical challenges faced by CISOs, CTOs, and cloud-native security experts.

THE IMPERATIVE FOR CLOUD-NATIVE EXCELLENCE

Cloud-native applications are revolutionizing the way organizations build and deliver software. These applications provide a competitive advantage through scalability, agility, and efficiency. However, maintaining this edge requires a fundamentally different approach to operations, security, and innovation. Traditional approaches designed for infrastructure managed solely by operations teams often need help in the dynamic and rapidly evolving world of self-service cloud-native applications.

This can lead to several challenges, including:

- **Slow Release Cycles:** Manual configuration management and approvals can introduce significant delays in the deployment process. This can slow innovation and make it difficult to respond quickly to changing business needs.
- **Configuration Drift:** Inconsistencies between intended and actual configurations can create security vulnerabilities and operational issues. Many factors can cause this, such as human error, inadequate automation, or changes made outside the typical deployment process.
- **Security Vulnerabilities:** Lack of policy enforcement throughout the development lifecycle can expose applications to security risks. This can include vulnerabilities in the code itself, as well as misconfigurations that allow unauthorized access to sensitive data.

To address these challenges, organizations must adopt a cloud-native approach to operations, security, and innovation, which includes:

- **Automating Everything:** Automating all aspects of the development and deployment process can reduce errors, improve efficiency, and ensure consistency.
- **Using Policy as Code:** Policy as code is a declarative approach to enforcing security and compliance requirements. It can help ensure that all applications are configured consistently and securely.
- **Adopting a Continuous Delivery Culture:** Continuous delivery is a software development practice emphasizing frequent, automated deployments. This can help to accelerate innovation and reduce the risk of security vulnerabilities.

GITOPS: THE FOUNDATION FOR AUTOMATED INFRASTRUCTURE AND APPLICATION MANAGEMENT

GitOps is a unified approach emphasizing automation, declarative configuration, continuous infrastructure, and application management deployment. It treats infrastructure as code and commonly uses Git as the source of truth. GitOps leverages a centralized repository for version control, collaboration, immutable references, and automated deployments. It focuses on declarative configuration, simplifying management. It also excels in continuous deployment, enabling frequent, low-risk changes. Organizations use GitOps with tools like Argo CD or Flux for automated deployments and compliance. GitOps provides improved automation, consistency, and faster delivery, helping organizations optimize operations and deliver high-quality software.

Declarative Configuration

- GitOps utilizes declarative languages such as YAML or HCL to define infrastructure and application configurations.
- This methodology enables a “state-driven” model, in which the desired system state is explicitly declared, and any alterations are made through declarative updates to the configuration files.
- Using declarative languages enhances transparency and streamlines change management, eliminating the necessity for convoluted scripts or imperative commands.

Version Control

- Git, a well-recognized distributed version control system, is the central repository for infrastructure and application configurations in GitOps.
- Each alteration made to the configuration is recorded as a Git commit, providing a comprehensive history of changes.
- This facilitates granular audibility, enabling teams to trace who made changes, when they were affected and the precise modifications that occurred.
- Version control allows for seamless rollbacks in the event of unintended changes or issues, ensuring swift and facile recovery.

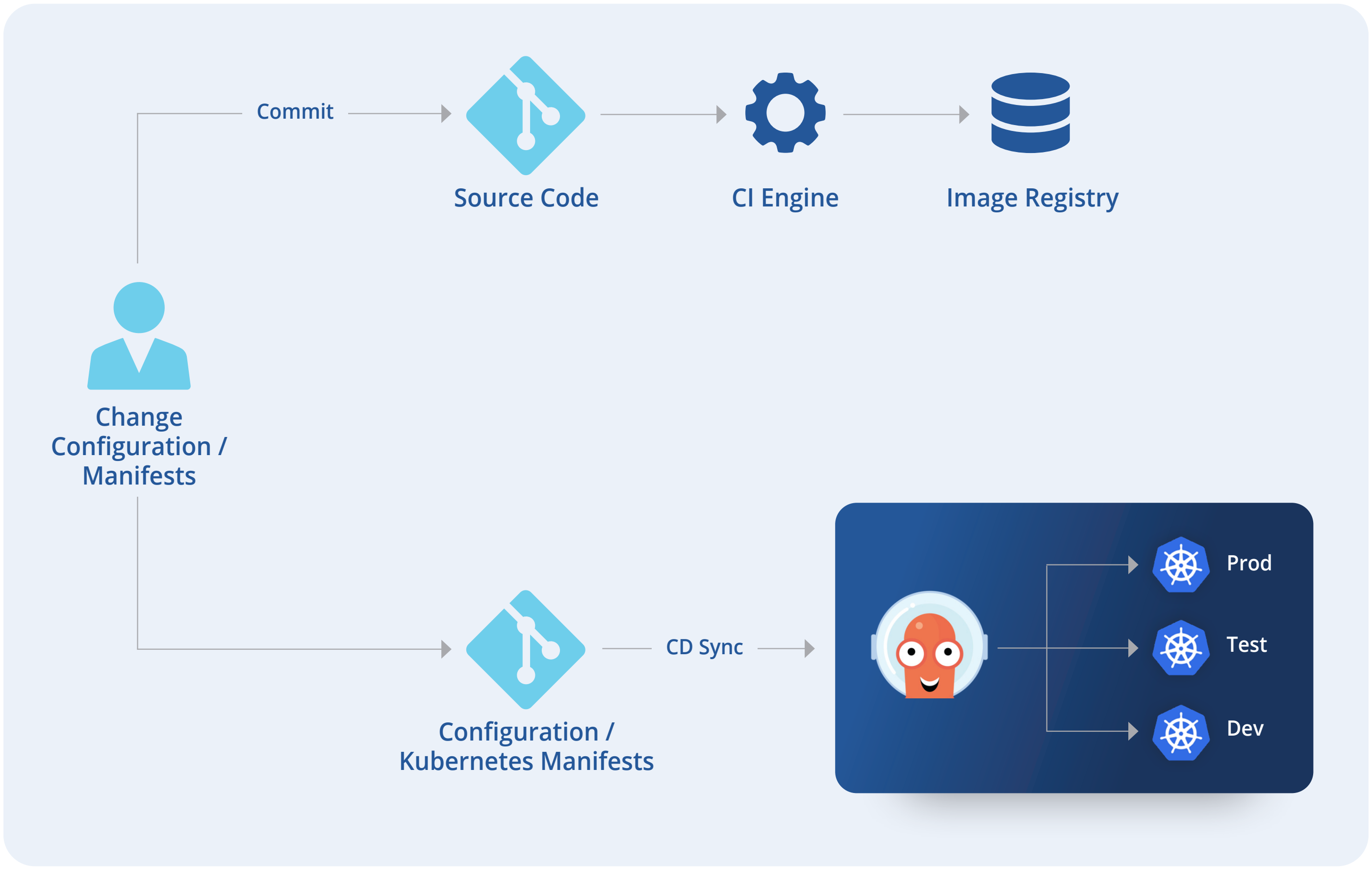
Automated Reconciliation

- GitOps leveraging controllers to continuously monitor and reconcile the live state of environments with the desired state defined in the Git repository.
- These agents periodically compare the actual state of the system against the desired state specified in the configuration files.
- Automated reconciliation guarantees that the system’s live state remains synchronized with the intended configuration, reducing manual intervention and minimizing the likelihood of configuration drift.

BENEFITS OF IMPLEMENTING GITOPS

Looking at the characteristics mentioned above of the GitOps approach, we instantly see that implementing it in your organization will result in the following:

- Heightened Efficiency in Development
- Increased Speed and Frequency of Deployment
- Assured Infrastructure Stability


Commit
Source Code
CI Engine
Image Registry
Change Configuration / Manifests
Configuration / Kubernetes Manifests
CD Sync
Prod
Test
Dev

POLICY AS CODE: AUTOMATING SECURITY AND COMPLIANCE ENFORCEMENT

You'll find that the Policy as Code approach draws directly from GitOps principles, which become a perfect foundation for implementing PaC, thus transforming security, compliance, and operational policies into structured, machine-readable code. Codifying policies offers numerous benefits, including centralized policy management, automated enforcement, compliance auditing, scalability, cost reduction, enhanced security, DevOps integration, regulatory compliance, and open-source support. PaC revolutionizes policy management in cloud-native environments by automating enforcement, streamlining processes, reducing costs, and enhancing security and compliance.

Policy as Code Definition

- PaC leverages structured, human-readable languages like YAML to express policies. This promotes clarity, version control, and collaboration among different stakeholders.
- Policies can be organized in modular and reusable components, making managing complex and evolving policy requirements easier.

Integrated with Infrastructure as Code (IaC)

- PaC aligns with the principles of Infrastructure as Code (IaC). Policies are defined alongside infrastructure definitions, ensuring that security and compliance are integral parts of the DevOps workflow.
- This integration enables the enforcement of policies during infrastructure provisioning, ensuring consistency and reducing manual errors.

Shift-Down Security

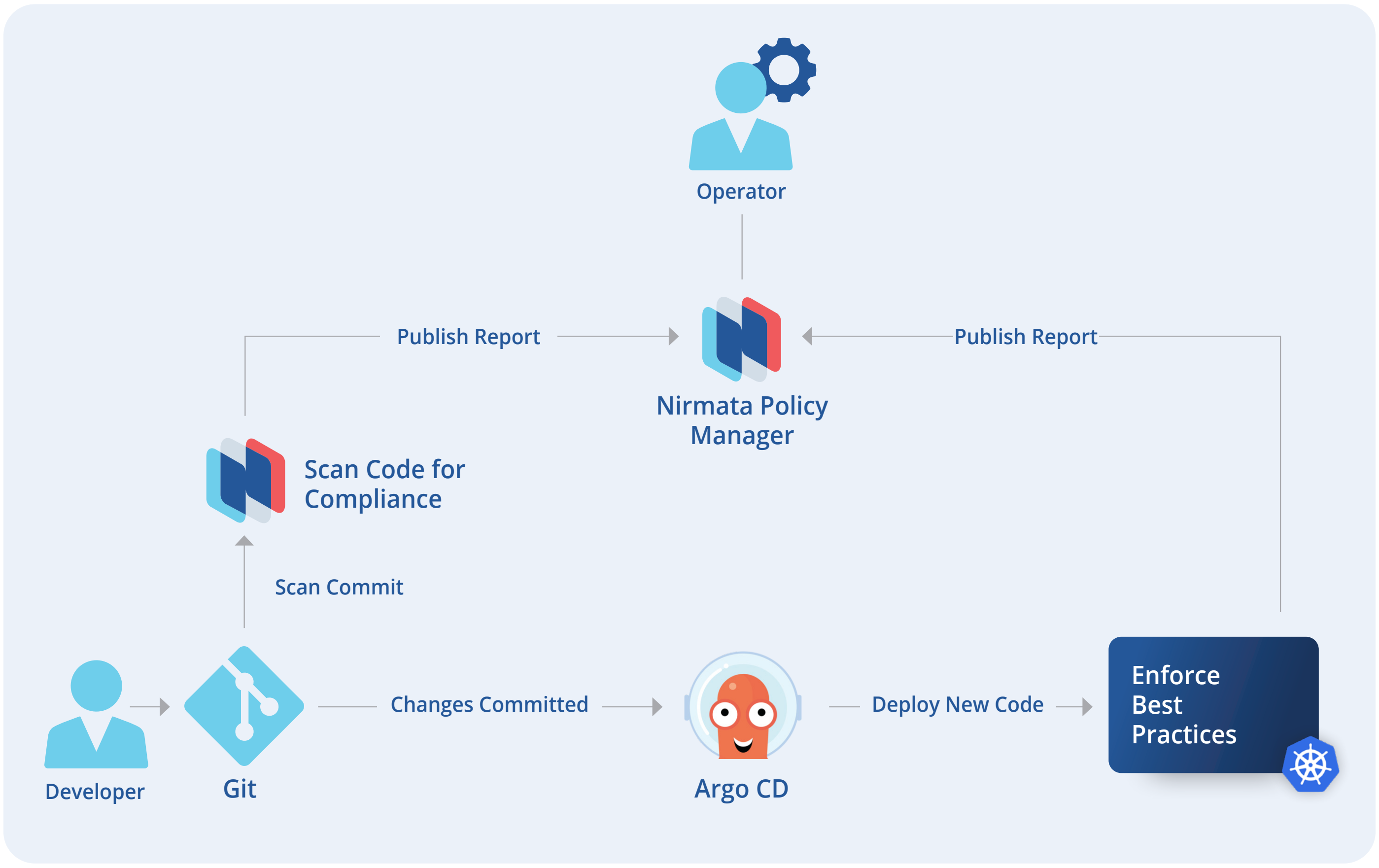
- PaC engines validate changes against policies directly within the Git workflow. This approach, known as "shift-down security," catches policy violations early in the development lifecycle.
- PaC prevents security vulnerabilities and compliance issues from materializing by identifying non-compliant configurations before they reach production.

Continuous Compliance

- PaC tools continuously monitor live environments, detecting and addressing policy violations post-deployment. This ongoing evaluation ensures adherence to established standards and regulations.
- Automated remediation capabilities can be implemented to correct policy violations, reducing the risk of security breaches, common misconfigurations, and compliance failures.

BENEFITS OF IMPLEMENTING POLICY AS CODE

- Improved Security
- Enhanced ComplianceOperational Efficiency
- Collaboration and Transparency



CRITICAL USE CASES FOR GITOPS AND PAC FOR ADDRESSING ENTERPRISE CHALLENGES

Combining the joint experience of Akuity and Nirmata - the creators of leading opensource tools for GitOps and PaC (Argo CD and Kyverno), we see the following enterprise level use cases where combining both of these code-centric approaches results in cloudnative operational excellence:

- **Secure Kubernetes Configuration:** The Akuity Platform manages Kubernetes manifests (deployments, services) across thousands of clusters from a central instance, while the Nirmata Policy Manager enforces policies like container privilege restrictions, image source verification, and mandatory resource limits.
- **Cloud Resource Governance:** IaC code in Git defines cloud infrastructure, while PaC ensures compliance with cost controls, naming conventions, encryption requirements, and security best practices.
- **Application Deployment Safeguards:** The Akuity Platform enforces where resources can be sourced from and where they can be deployed, while the Nirmata Policy Manager mandates network isolation policies, enforces pod resource requests/limits, and ensures images originate from approved registries.
- **Scalability and Multi-Cloud Support:** Solutions like Akuity Platform offer scalable, agent-based architectures for easy GitOps deployments across large, multi-cloud environments.
- **Increased Security Posture:** Nirmata Policy Manager automates security checks and enforces best practices throughout the development lifecycle, significantly reducing risk and minimizing the attack surface. Akuity Platform's agent-based architecture provides secure and isolated GitOps controllers, audit logs, and CVE detections in your cluster. Together, your security posture grows exponentially.
- **Compliance Assurance:** Nirmata Policy Manager streamlines compliance with industry regulations by providing auditable logs and automated enforcement mechanisms. Akuity Platform provides audits for every action Argo CD makes in your environments.

BEST PRACTICES AND CONSIDERATIONS

- **Policy Scope:** Strike a balance between granular policies and avoiding overly broad restrictions that might hinder development.
- **Integrated Testing:** Incorporate policy checks throughout the testing and staging pipeline to validate changes pre-production.
- **Change Management:** Establish transparent processes for policy modifications, ensuring security and operations teams are involved in maintaining alignment with evolving requirements.
- **Application Security:** How to shift from reactive to proactive application security with policy as code.
- **GitOps Best Practices:** Learn from insights and lessons learned from deploying Argo CD at scale in production in Akuity's GitOps Best Practice whitepaper.
- **Solution Selection:** Evaluate enterprise-grade tools like Akuity and Nirmata for their scalability, advanced policy management capabilities, and support for multicloud and complex environments.
- **Hands-On:** To get hands-on with the practices in this paper, check out the Securing GitOps with Policy as Code webinar with Nicholas Morey, Boris Kurktchiev, and Carlos Santana.

CRITICAL USE CASES FOR GITOPS AND PAC FOR ADDRESSING ENTERPRISE CHALLENGES

GitOps and Policy as Code offer a transformational approach to cloud-native development and operations, driving speed, security, and operational efficiency. By recognizing these technologies' value proposition and technical depth, CTOs, CISOs, and cloud-native experts can empower their organizations with a clear competitive advantage.

To find out more about these velocity and security-increasing approaches to shipping software, feel free to reach out to us and schedule a demo:

- **Akuity Platform:** from the creators of The Argo Project
- **Nirmata:** from the creators of Kyverno

Brought to you by Akuity and Nirmata