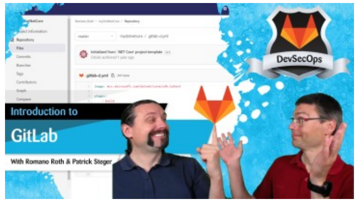
The Epic Comparison

GitHub vs. GitLab

With Romano Roth & Patrick Steger

What is GitLab?
https://youtu.be/sHK8uN5fBhs



Introduction to GitLab
https://youtu.be/GQ3x9bkCK90

SCA
https://youtu.be/l69W5Ym_M5o

License Compliance
https://youtu.be/Kmbj_PCiHyk

SAST
https://youtu.be/owwIMUamdDc

Container Scanning
https://youtu.be/1AUKQ32K6D4

SECR
DevSecOps
What is
Secret Detection in GitLab?
With Romano Roth & Patrick Steger

Secret Detection
https://youtu.be/Qs28ONnj00s

DAST
https://youtu.be/Jy1OiuPZrKs

Vulnerability Management
https://youtu.be/XSrlVyv0H1c

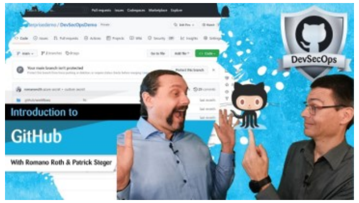
Merge Request
https://youtu.be/h4AN7S2gwug

Schedule Pipeline
https://youtu.be/PqPW3zQeP94

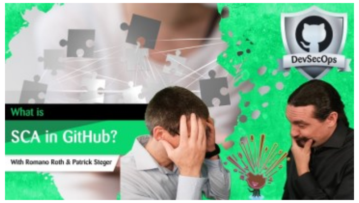
Recommendati
SecOps
How to build a DevSecOps pipeline with Gitlab
Our Recommendations
With Romano Roth & Patrick Steger

Recommendations
https://youtu.be/dphgw9xxjuw

What is GitHub?
https://youtu.be/_m5KYEi1ThA



Introduction to GitHub
https://youtu.be/6ZdxXDu8ZDA



SCA
https://youtu.be/xM3elerxjYo

License Compliance
https://youtu.be/l7IBh2xkDcQ

SAST
https://youtu.be/p4xS2X5KsNk

DevSecOps
How to do
Container Scanning in GitHub?
With Romano Roth & Patrick Steger

Container Scanning
https://youtu.be/_ZeKh3GcbgU

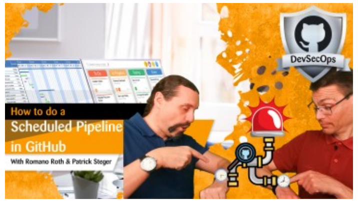
SECRET
DevSecOps
What is
Secret Detection in GitHub?
With Romano Roth & Patrick Steger

Secret Detection
https://youtu.be/k-uuPTLNXGM

DAST
https://youtu.be/v_xo1kgNYsE

Vulnerability Management
https://youtu.be/cDf-U-wMgfc

Pull Requsts
https://youtu.be/Yy3KAloE5e0



Schedule Pipeline
https://youtu.be/xsLCR7b4u9k

Recommendati
evSecOps
How to build a DevSecOps pipeline with GitHub
Our Recommendations
With Romano Roth & Patrick Steger

Recommendations
https://youtu.be/zCxZhVTUpNE

Background

- We have implemented an enterprise-ready DevSecOps pipeline on both, GitLab and GitHub. Focus is the **Sec** part of DevSecOps.
  -> You can find the 24 videos on our YouTube channel
- We strive for a balanced solution providing good enough security controls for acceptable effort – this may vary in your case!
- We use the enterprise / ultimate (paid) version of the platforms for the comparison.
- Disclaimer:
  - Platforms are changing, our comparison is based on what we experienced during our journey to implement the pipelines.
  - This is our opinion and experience only. You may disagree.

Feature comparison

Feature	GitLab	GitHub
Number of users	> 30 million	> 100 million
Deployment options	☑ SaaS, Self-Managed	☑ SaaS, Self-Managed
Price	☑ More expensive Free: 0$ Premium: $24/user/month Ultimate: $99/user/month	☑ ☑ More affordable Free: 0 $ Team: $3.67/user/month Enterprise: $19.25/user/month
Open/Closed Source	☑ Open Source (MIT)	● Closed Source
SLA	☑ 99.5% uptime guarantee	☑ ☑ 99.99% uptime guarantee
Personal Use	☑ 5GB of storage 400 CI/CD minutes/month 5 users per repository.	☑ ☑ Unlimited storage 2'000 CI/CD minutes per month, unlimited number of contributors.
Enterprise Use	☑ Ultimate: 250 GB of storage 50'000 CI/CD minutes/month Protected Branches Code Owners Merge Requests with Approval Rules Security Dashboards Vulnerability Management Dependency Scanning Container Scanning Static Application Security Testing Dynamic Application Security Testing	☑ Enterprise: Unlimited storage 50'000 CI/CD minutes/month Protected Branches Code Owners Pull request with Approval Rules GitHub Advanced Security

Feature comparison

Feature	GitLab	GitHub
Ease to learn	☑ ☑	◉ More complex harder orientation/navigation
Documentation	☑ ☑ Good documentation	☑ But hard to figure out how to do something
On platform code edit	☑ ☑ Fully fletched Web-based IDE	☑ ☑ Fully fletched Web-based IDE
Power of Pipeline Description Language	☑ Harder to keep an overview	☑ ☑
Out-of-the-box security tooling	☑ Defaults for everything you need, some tools are weak	✕ No defaults, hard to find
Vulnerability Management	☑ limited, but you can potentially survive with it	◉ Missing core capability e.g., add custom vulnerabilities
Secret Management	✕ Built-in solution is not secure, Vault integration complicated and no real default	☑ ☑ Built-in solution is ok, Azure Key Vault integration
Supply chain risk	☑ GitLab curated code/tools	✕ ✕ Mostly community curated code/tools -> Review Overhead or way more risk
Custom tool integration	✕ Complicated, GitLab specific	☑ Better standardized, easier to use
Merge/Pull Request support	☑ ☑	☑ Resolved issues not visible, not all tools considered, no dynamic security approval

GitLab: Our wishes for improvement – in order of importance

- Enable proper and easy secret management
- Vulnerability Management:
    - Allow to add description why we dismissed something
    - Allow “dismiss until” functionality
    - Allow to change severity (with comment)
    - Improve company view of vulnerability management
    - Provide capability to “alarm” or break pipeline when new vulnerability of a configured severity appears
    - Extended reporting functions

GitLab: Our wishes for improvement – in order of importance

- Allow more flexibility regarding branches that are used in vulnerability management and scheduling (too much is restricted to default branch)
- Improve the DAST tool default configuration
- Make it easier to re-use artifacts from previous pipeline steps
- Make it (way) easier to integrate custom (security) tools into GitLab and the vulnerability management.

GitHub: Our wishes for improvement – in order of importance

- Provide default, company-curated security tooling
- Create a "trusted" marketplace for tools, where Microsoft conducts ongoing reviews and verifies that the code poses no security risks.
- Vulnerability Management:
  - Allow to add/manage external security issues/vulnerabilities
  - Allow “dismiss until” functionality
  - Allow changing severity (with comment)
  - Provide company aggregated view (multiple projects) and Dashboard to see improvement over time

GitHub: Our wishes for improvement – in order of importance

- Vulnerability management (cont.):
  - Provide capability to “alarm” or break pipeline when new vulnerability of a configured severity appears
  - Extended reporting functions
- Schedulers for all branches
- Make sure there are security tools available for all areas (i.e., proper License Compliance is missing today)
- Enable possibility to trigger approval requirement when new vulnerabilities (security tool findings) of a given criticality are found on pull request
- Make it easier to re-use artifacts from previous pipeline steps

Summary GitLab and GitHub

- Enterprise ready DevSecOps pipelines are possible on both platforms.
- Focus on tools that provide the most value for effort: SCA, Container-Scanning, SAST, and Secret Detection + License Compliance.
- Remember to regularly do at least SCA/container-scanning for code that is in production.
- Try to use tools that integrate into the vulnerability management view of the platform – saves you the cost for extra security management tools. Target: One platform for developers – today only with GitLab achievable.
- Define and establish processes how to handle findings – both new and existing.

Summary GitLab and GitHub

- Protect the release/production branches. Require review/approval, no overriding there.
- Consider managing resources you “include” from somewhere else on your own. (I.e. copy them to your own repositories)
  - Marketplace tools
  - Provided pipeline-jobs (e.g. templates)

Summary of the summary

- **GitLab** is faster to deliver results and has out-of-the-box tooling for everything but lacks proper secret management
  -> This would be our recommendation when you want to get there fast and are ok to stick to the defaults

- **GitHub** offers more flexibility, supports great secret management and has a living community but comes with high supply chain risk, has no reasonable security tool defaults and is missing a critical vulnerability management feature (add external vulnerability)
  -> This would be our recommendation when you have complex applications/pipelines, or you must integrate with a few external (security) tools

GitLab vs. GitHub: DevSecOps Pipeline

https://www.romanoroth.com/post/gitlab-vs-github-devsecops

GitLab vs. GitHub: DevSecOps Pipeline

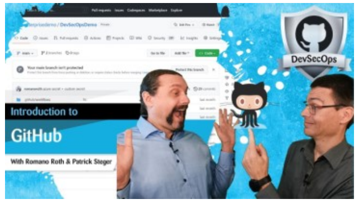
by Romano Roth and Patrick Steger

This blog post will show you how to build up an enterprise-ready DevSecOps Pipeline with GitLab and GitHub and compare the two platforms.

Romanos and Paddis Videos on DevSecOps with GitHub

What is GitHub?

https://youtu.be/_m5KYEi1ThA



Introduction to GitHub

https://youtu.be/6ZdxXDu8ZDA

SCA

https://youtu.be/xM3elerxjYo

License Compliance

https://youtu.be/l7IBh2xkDcQ

SAST

https://youtu.be/p4xS2X5KsNk

DevSecOps
How to do
Container Scanning in GitHub?
With Romano Roth & Patrick Steger

Container Scanning

https://youtu.be/_ZeKh3GcbgU

SECRET
DevSecOps
What is
Secret Detection in GitHub?
With Romano Roth & Patrick Steger

Secret Detection

https://youtu.be/k-uuPTLNXGM

DAST

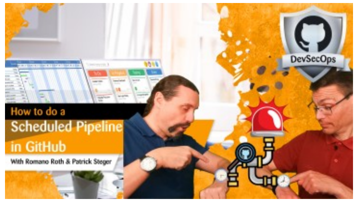
https://youtu.be/v_xo1kgNYsE

Vulnerability Management

https://youtu.be/cDf-U-wMgfc

Merge Request

https://youtu.be/Yy3KAloE5e0



Schedule Pipeline

https://youtu.be/xsLCR7b4u9k

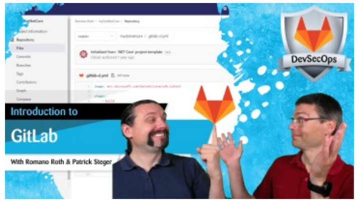
Recommendatic
DevSecOps
How to build a DevSecOps pipeline with GitHub
Our Recommendations
With Romano Roth & Patrick Steger

Recommendations

https://youtu.be/zCxZhVTUpNE

Romanos and Paddis Videos on DevSecOps with GitLab

What is GitLab?
https://youtu.be/sHK8uN5fBhs



Introduction to GitLab
https://youtu.be/GQ3x9bkCK90

SCA
https://youtu.be/l69W5Ym_M5o

License Compliance
https://youtu.be/Kmbj_PCiHyk

SAST
https://youtu.be/owwIMUamdDc

Container Scanning
https://youtu.be/1AUKQ32K6D4

DevSecOps
What is
Secret Detection in GitLab?
With Romano Roth & Patrick Steger

Secret Detection
https://youtu.be/Qs28ONnj00s

DAST
https://youtu.be/Jy1OiuPZrKs

Vulnerability Management
https://youtu.be/XSrlVyv0H1c

Merge Request
https://youtu.be/h4AN7S2gwug

Schedule Pipeline
https://youtu.be/PqPW3zQeP94

Recommendati
DevSecOps
How to build a DevSecOps pipeline with Gitlab
Our Recommendations
With Romano Roth & Patrick Steger

Recommendations
https://youtu.be/dphgw9xxjuw