



GitLab

DevOps Metrics with GitLab



- What are DevOps Research and Assessment (DORA) metrics
- Why are they important
- Where can you find them in GitLab
- Other metrics available in GitLab

“I have great respect for the past. If you don't know where you've come from, you don't know where you're going. I have respect for the past, but I'm a person of the moment. I'm here, and I do my best to be completely centered at the place I'm at, then I go forward to the next place.”

- Maya Angelou

Is it true, is it true?



People, Process, Technology

Ask yourselves... Do you have:

Great Developers?

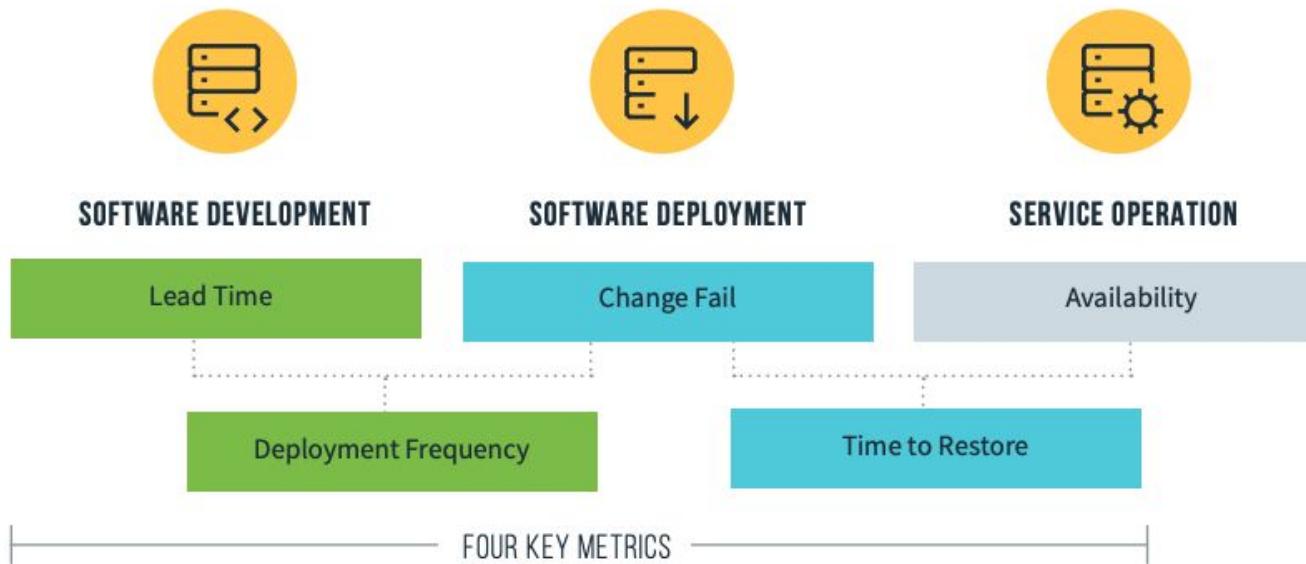
Simple Process?

Technology that works and doesn't need much to keep it that way?



DORA's research into what makes high performing technology teams is considered foundational in the DevOps space

PERFORMANCE METRICS



Where do we start?



Optimize for Performance

Psychological safety
Change management
Cloud
Technical practices
Disaster Recovery Testing

Optimize for Productivity

Psychological safety
Useful easy-to-use tools
Internal search
External search
Technical debt

Start by identifying your goal...



Psychological Safety



🔑 Tool: Foster psychological safety

YouTube Building a psychologically safe workplace | Amy Edmondson | TEDxHGSE

re:Work

THE WORK ISSUE

What Google Learned From Its Quest to Build the Perfect Team

New research reveals surprising truths about why some work groups thrive and others falter.

 **DORA**
DEVOPS RESEARCH & ASSESSMENT

 GitLab



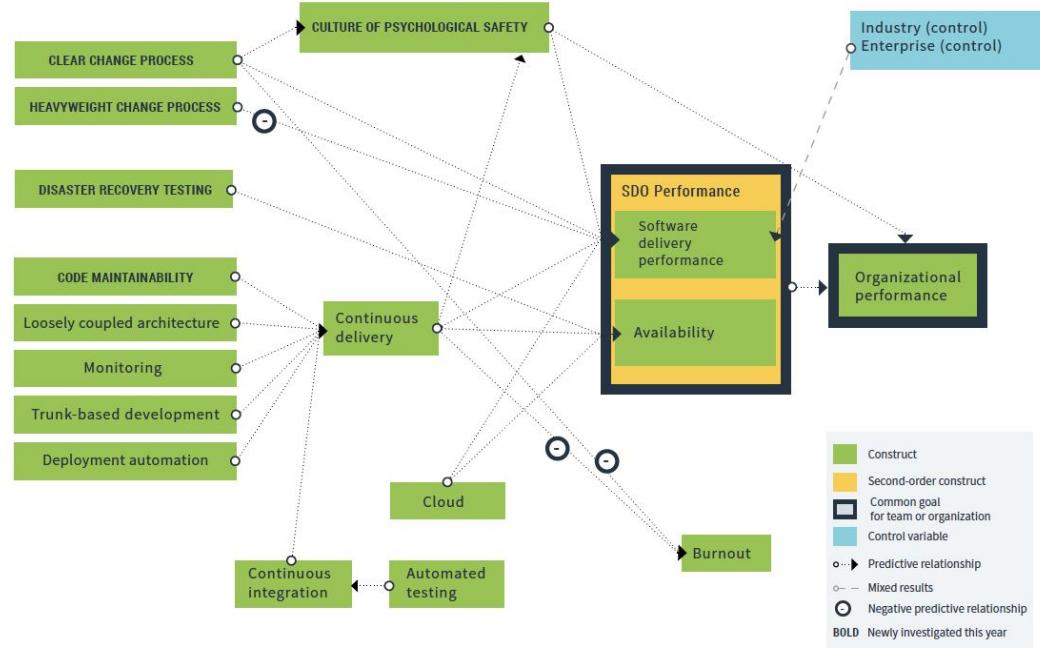
re:Work

Performance



SOFTWARE DELIVERY & OPERATIONAL PERFORMANCE

- Culture
- Clear change process
- Continuous delivery
- Cloud
- Then focus on those that are your biggest constraints.



Productivity



Productivity is the ability to get complex, time-consuming tasks completed with minimal distractions and interruptions. Many of us describe this as getting into a good work flow or rhythm.

- Easy to use tools
- Manage technical debt
- Work recovery



Identify what to measure



DORA recommends:

- **Lead Time** - how long does it take to go from code committed to code successfully running in production)
- **Deployment Frequency** - how often does your organization deploy code to production or release it to end users?
- **Change Failure Rate** - what percentage of changes to production or released to users result in degraded service and subsequently require remediation
- **Time to Restore Service** - how long does it generally take to restore service when a service incident or a defect that impacts users occurs

GitLab additionally recommends:

- **Security** - how much security risk is taken on for each of these projects



—

Where to find these things in GitLab

Customizable Value Stream Analytics - “Lead Time”



What's in it?

- Identifies bottlenecks in development process
- Customize stages aligned with your KPIs

Helps customers answer

- How long are we taking from Idea to Production (Cycle Time)
- Where are the bottlenecks in my process?

How to navigate?

- Project: Analytics -> Value Stream Analytics

The screenshot shows the Jira Value Stream Analytics interface. On the left, a table displays lead times for various stages: Issue (1 minute), Plan (Not enough data), Code (Not enough data), Test (Not enough data), Review (Not enough data), Staging (Not enough data), Total (Not enough data), and In Development (25 days). A blue box highlights the "Add a stage" button at the bottom of this table. On the right, a modal window titled "New stage" is open, allowing users to define a new stage named "In Review". It includes fields for "Start event" (Issue label was added) and "Start event label" (In review), and "Stop event" (Issue closed). A green "Add stage" button is at the bottom of the modal. Below the modal, another table shows "Issue start to finish" times for different stages: Plan (about 10 hours), Test (about 5 hours), Review (3 days), and Staging (2 days). The last row of this table is also highlighted with a blue box around the "Add a stage" button.

Stage ⓘ	Median ⓘ
Issue	1 minute
Plan	Not enough data
Code	Not enough data
Test	Not enough data
Review	Not enough data
Staging	Not enough data
Total	Not enough data
In Development	25 days

Stage ⓘ	Median ⓘ	Issue start to finish
Plan	about 10 hours	Implement user #5 · Opened 22 d
Test	about 5 hours	
Review	3 days	
Staging	2 days	
		Issue start to finish 21 days

CI/CD Analytics - “Deployment Frequency”



What's in it?

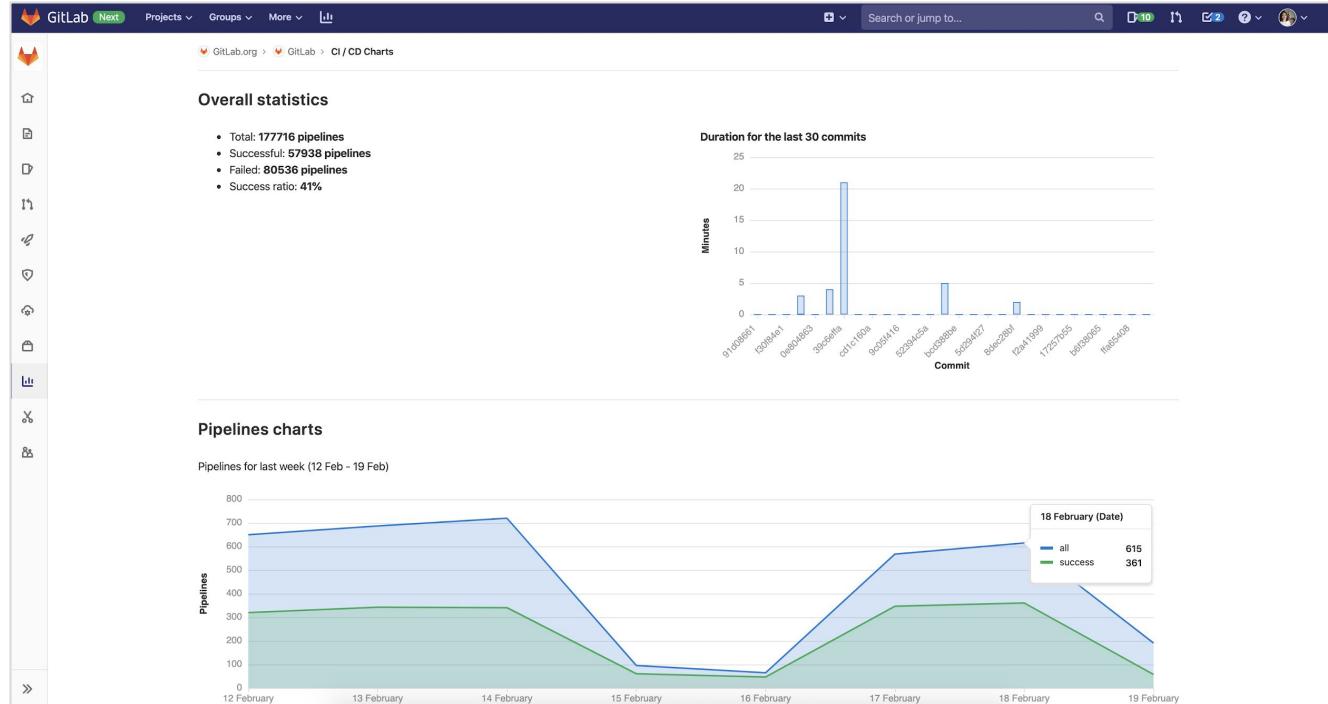
- Overall pipelines run & duration, success ratios by week, month, year

Helps customers answer

- What is my CI/CD pipeline success rate?

How to navigate?

- Project: Analytics -> CI/CD Analytics



Issue Analytics “Change Failure Rate”



What's in it?

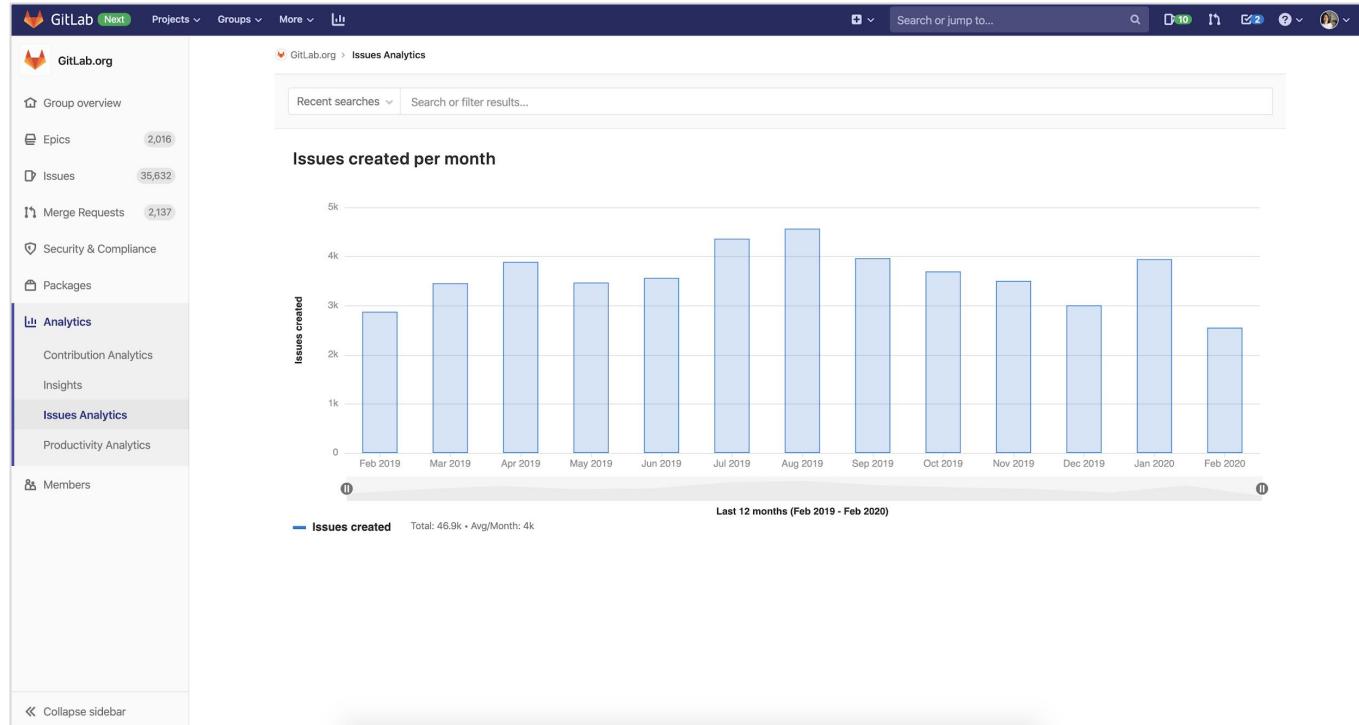
- Issues created per month by milestone, release, author etc

Helps customers answer

- What is the developer / QA productivity of my team?

How to navigate?

- Group: Analytics -> Issue Analytics



Group & Project Insights “Change Failure Rate”



What's in it?

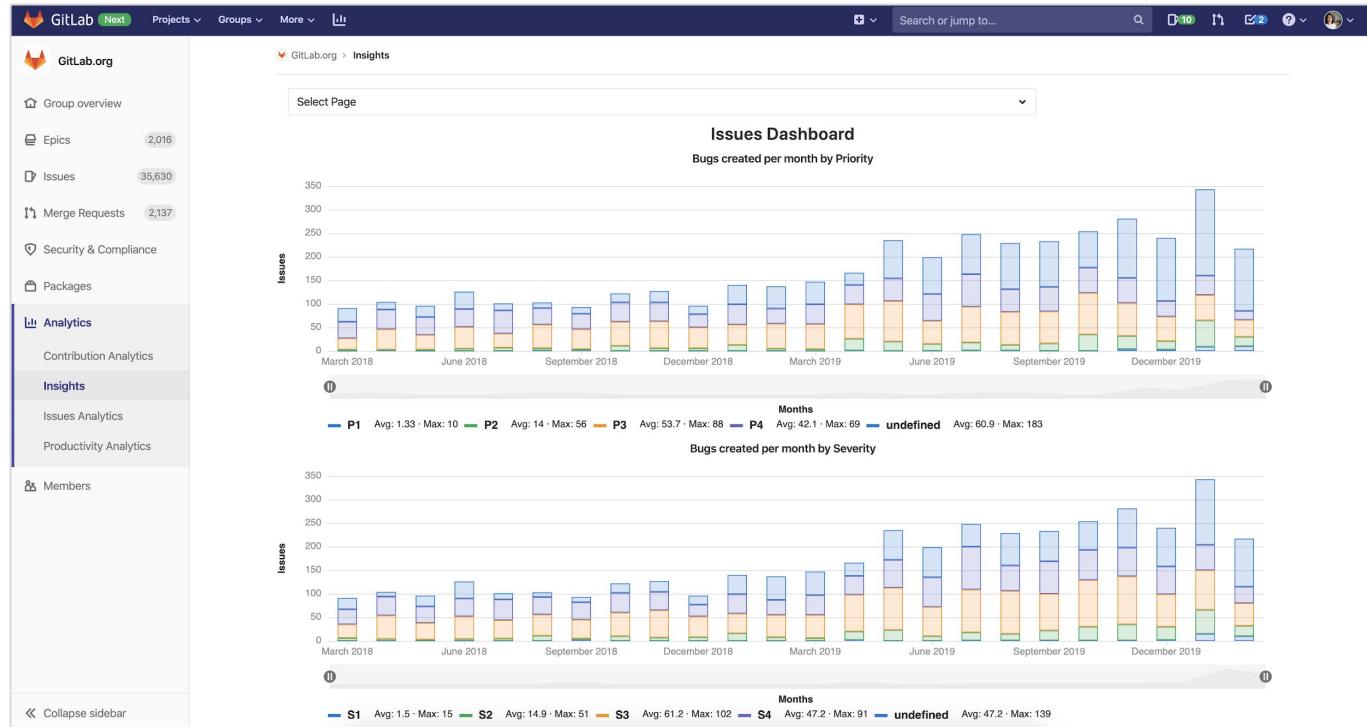
- Bugs created, merge requests, regressions, missed deadlines, track labeled issues

Helps customers answer

- What are our metrics on tracked labels? E.g., bugs, strategic issues etc

How to navigate?

- Group/Project: Analytics -> Insights



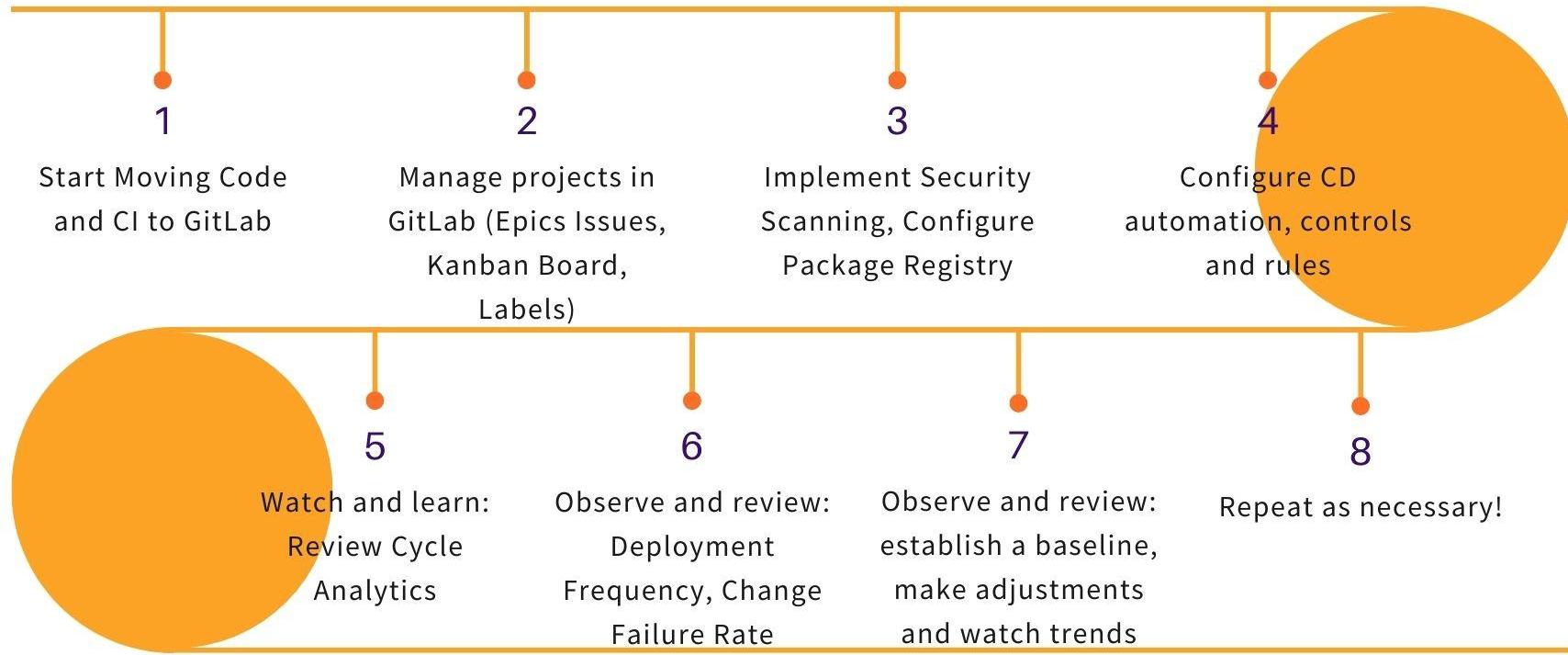
Currently Unavailable “Time to Restore Service”



But multiple options are available to derive these:

1. Incident management in GitLab could capture that and the Issue analytics can present the information.
2. Incident ticket system (assuming it is outside of GitLab can offer these metrics)

Where to begin





GitLab

—

But wait... There is more...



GitLab

Executive Insights

DevOps Score



What's in it?

- Your organization's maturity in DevOps Adoption

Helps customers answer

- How are we doing compared to other organizations?

How to navigate?

- Analytics -> DevOps Score

The screenshot shows the GitLab DevOps Score dashboard. On the left, a sidebar menu includes 'Analytics', 'Productivity Analytics', 'Cycle Analytics', 'DevOps Score' (which is selected), and 'Cohorts'. The main content area displays the 'DevOps Score' page with the title 'Introducing Your DevOps Score'. It states: 'Your DevOps Score gives an overview of how you are using GitLab from a feature perspective. View how you compare with other organizations, discover features you are not using, and learn best practices through blog posts and white papers.' Below this, a large red box highlights the '12.6% index score'. The dashboard then lists various metrics comparing 'You' and 'Lead' organizations:

Metrics	You	Lead	Score	Details
Issues created per active user	0.12	3.3	3.6%	i
Comments created per active user	3.6	19.0	18.7%	i
Milestones created per active user	0.00	0.50	0.0%	i ?
Boards created per active user	0.00	0.33	0.0%	i ?
Merge Requests per active user	0.14	3.8	3.6%	i ?
Pipelines created per active user	0.00	17.8	0.0%	i ?
Environments created per active user	0.83	0.83	100.0%	i ?
Deployments created per active user	0.00	10.2	0.0%	i ?
Monitoring fraction of all projects	0.00	0.12	0.0%	i ?
Service Desk issues created per active user	0.00	3.3	0.0%	i ?

At the bottom, there are two rows of icons representing different DevOps processes: planning, development, testing, deployment, monitoring, and service management.

Roadmaps



What's in it?

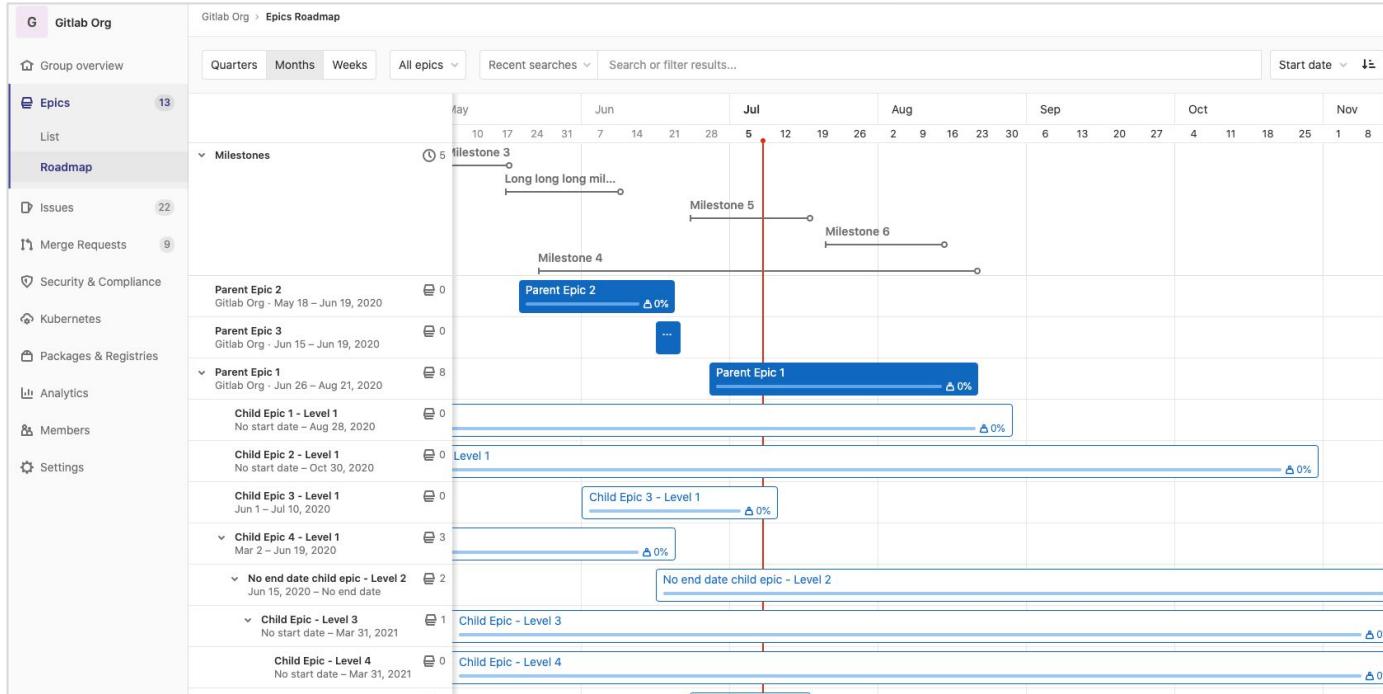
- Timelines for epics - helps with portfolio planning, tracking & execution

Helps customers answer

- Are we building the right thing at the right time?

How to navigate?

- Epics -> Roadmap





GitLab

Productivity Insights

Productivity Analytics



What's in it?

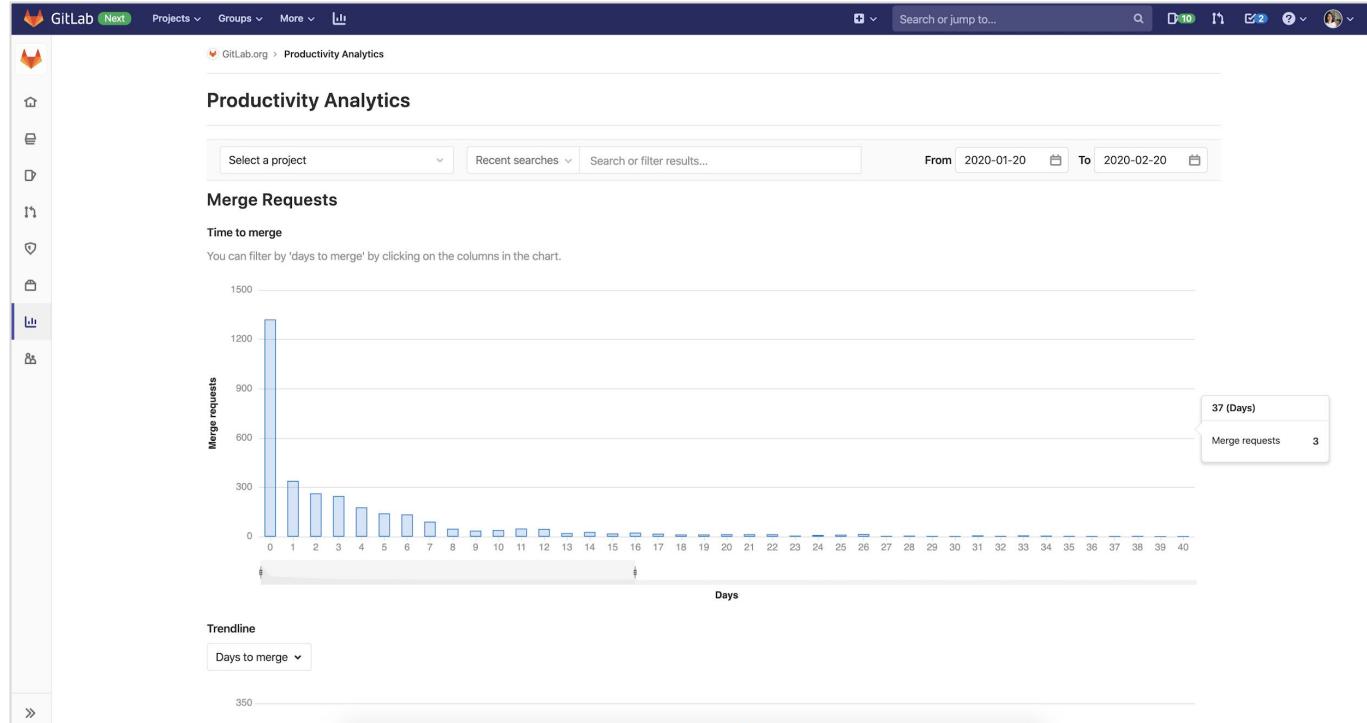
- Time to merge, commits per MR, time for first comment, lines of code per commit etc

Helps customers answer

- What is our Development Velocity at a group & project level?

How to navigate?

- Group/Project: Analytics -> Productivity Analytics



Code Review Analytics



What's in it?

- Open merge requests, review time

Helps customers answer

- What are our longest running reviews impacting cycle time?

How to navigate?

- Analytics -> Code Review Analytics

The screenshot shows the GitLab interface with the sidebar collapsed. The main header reads "Code Review". Below it, a sub-header says "Review time is defined as the time it takes from first comment until merged." A search bar is present. The main content area is titled "Merge Requests in Review 1002". It lists several merge requests with columns for Merge Request ID, Review time, Author, Approvers, Comments, Commits, and Line changes. The last column includes a green "+312 -1" and a red "+2 -10" respectively.

Merge Request	Review time	Author	Approvers	Comments	Commits	Line changes
WIP: Introduce release assets with debian r... !2234 · opened 2 years ago	967 days	[User]	-	18	1	+312 -1
EE::ProtectedRef inherits has_many and acc... !2768 · opened 2 years ago	907 days	[User]	-	4	1	+2 -10
WIP: Use case-insensitive lookup for Kerber... !3047 · opened 2 years ago · ⏪ Backlog	869 days	[User]	-	9	2	+18 -1
Add note about HTTP/2 causing issues with ... !3111 · opened 2 years ago	861 days	[User]	-	4	1	+12 -0
Secret variable settings update !5974 · opened 1 year ago · ⏪ 12.9	622 days	[User]	-	77	53	+245 -114
Sanitized RelayState redirect for Group SAML !6108 · opened 1 year ago · ⏪ Backlog	616 days	[User]	-	10	1	+43 -1
Perform sql matching of tags ee !3628 · opened 2 years ago	552 days	[User]	-	2	3	+114 -7



GitLab

Developer Insights

Contribution Analytics



What's in it?

- Push events, merge requests & issues opened/closed

Helps customers answer

- At a group level, who are our top contributors and areas of improvement?

How to navigate?

- Group: Analytics -> Contribution

The screenshot shows the GitLab Contribution Analytics page with the following sections:

- Push**: 7795 pushes, more than 456029.0 commits by 339 people contributors. A chart shows pushes from GitLab Bot, Mark Chao, Miranda Fluharty, Lindsay Kerr, and Konrad Borowski.
- Merge Requests**: 1073 created, 839 accepted. A chart shows merge requests from Isaac Dawson, Nicolas Dular, and Satya.
- Issues**: 1040 created, 848 closed. A chart shows issues closed from GitLab Bot, Jeremy Elder, and Chad Woolley.
- Contributions per group member**: A table showing contributions per group member, including Name, Pushed, Opened issues, Closed issues, Opened MR, Accepted MR, and Total.

Name	Pushed	Opened issues	Closed issues	Opened MR	Accepted MR	Total
GitLab Bot	817	40	29	37	20	942
Isaac Dawson	499	1	0	2	23	520
GitLab Release Tools Bot	374	9	8	2	4	397
Rémy Coutable	162	15	11	23	20	234
Build Trigger	222	0	0	0	0	222
Julian Thome	154	3	3	44	2	200
Manoj M J	11	182	2	2	0	197
Ian Baum	175	3	1	2	2	183
Alexandru Croitor	136	5	1	7	0	150
Stan Hu	101	4	8	14	11	142
GitLab Alert Bot	0	121	21	0	0	142
Marcel Amirault	87	5	3	11	25	131
John Cai	101	4	3	10	4	128
Natalie Tschirhart	91	1	11	1	25	119

Contributors



What's in it?

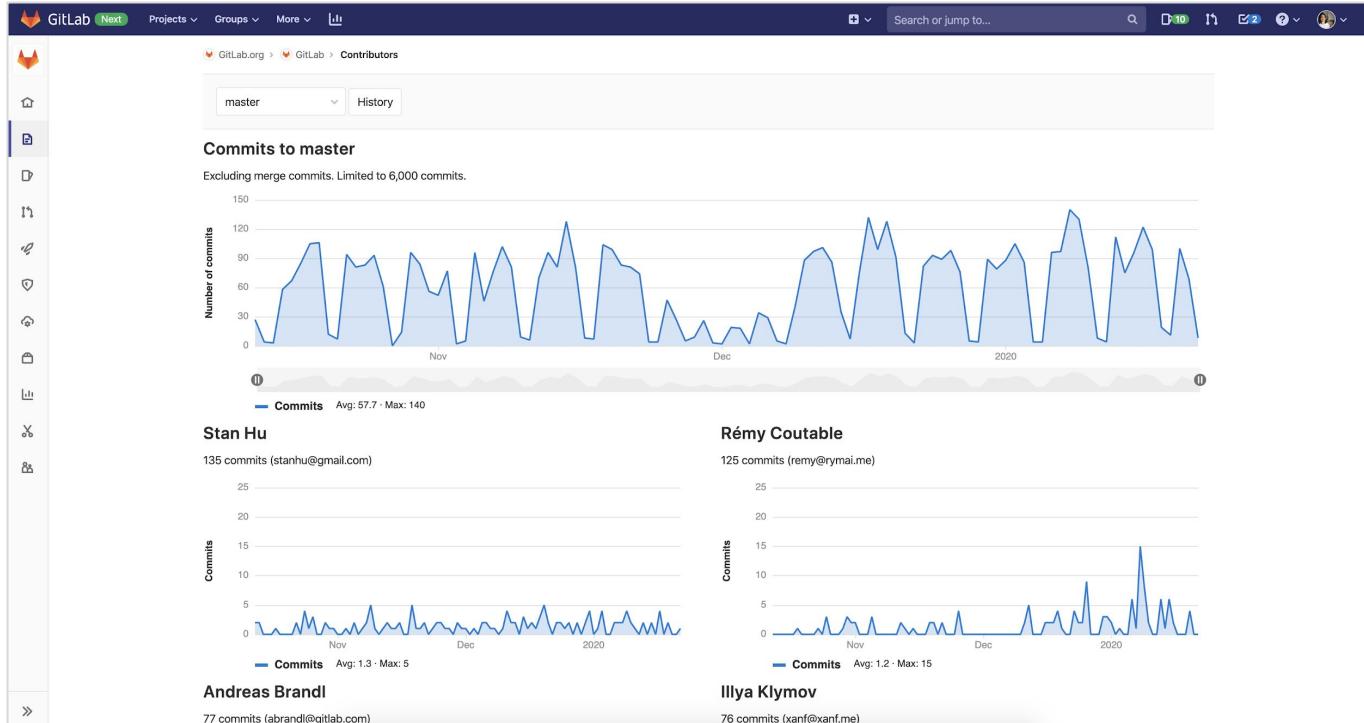
- Overall Commits to master & commits by users

Helps customers answer

- At a project level, who are our top contributors?

How to navigate?

- Project: Repository -> Contributors



Repository Analytics / Contribution Charts



What's in it?

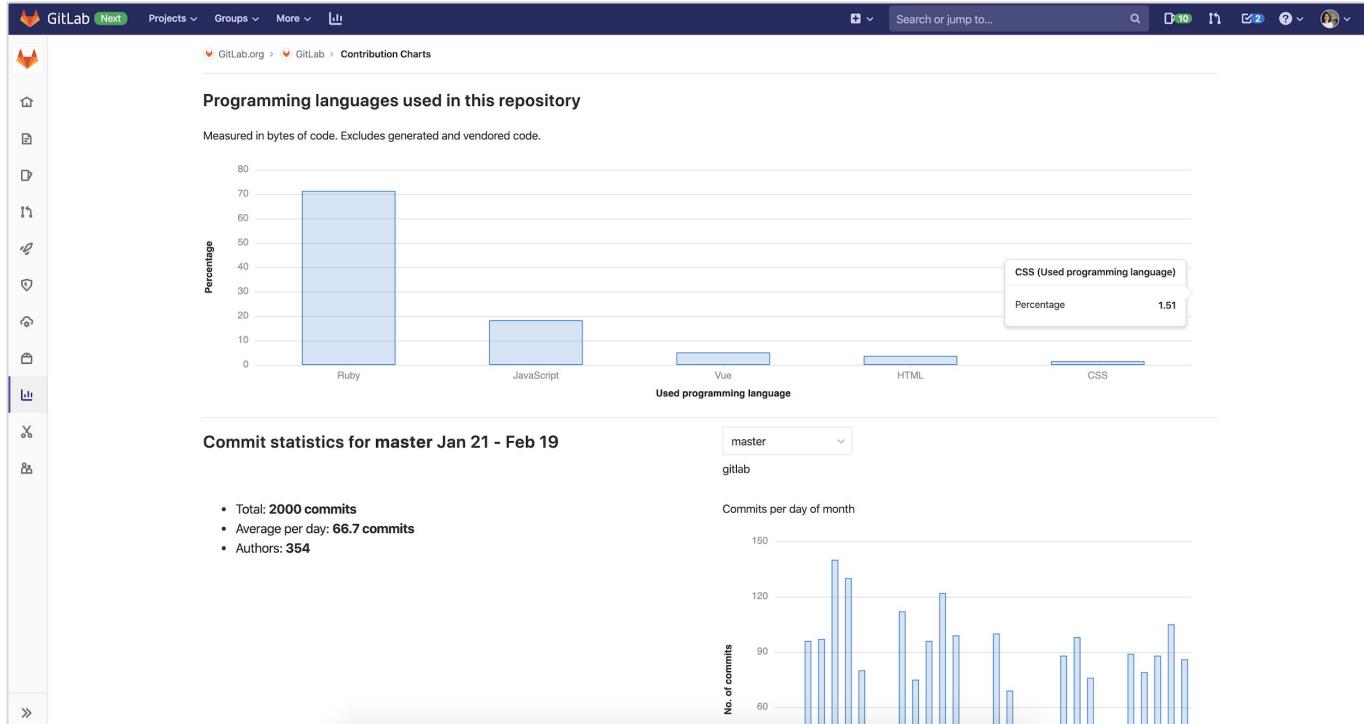
- Total commits, average per day commits, programming languages used in repository

Helps customers answer

- What programming languages do developers prefer to use?

How to navigate?

- Project: Analytics -> Repository Analytics



Code Quality Report



What's in it?

- Code Quality across the project

Helps customers answer

- What is the quality of my entire project?

How to navigate?

- Project: CI/CD -> Pipelines -> Code Quality Tab

Add new file

⌚ 10 jobs for **feature-branch** in 36 seconds (queued for 5 minutes and 17 seconds)

⟳ latest

⌚ 0f395a0e ⋮ ⌂

❗ 1 related merge request: !1 WIP: Add new file

Pipeline Jobs 10 Code Quality

❗ Found 2258 code quality issues

- ✖ Similar blocks of code found in 2 locations. Consider refactoring. in [ee/spec/features/admin/geo/admin_geo_projects_spec.rb:152](#)
- ✖ Similar blocks of code found in 2 locations. Consider refactoring. in [ee/spec/finders/geo/lfs_object_registry_finder_spec.rb:491](#)
- ✖ Similar blocks of code found in 2 locations. Consider refactoring. in [ee/spec/finders/geo/lfs_object_registry_finder_spec.rb:512](#)
- ✖ Similar blocks of code found in 2 locations. Consider refactoring. in [ee/spec/finders/security/pipeline_vulnerabilities_finder_spec.rb:129](#)
- ✖ Similar blocks of code found in 2 locations. Consider refactoring. in [ee/spec/finders/security/pipeline_vulnerabilities_finder_spec.rb:147](#)
- ✖ Similar blocks of code found in 2 locations. Consider refactoring. in [ee/spec/lib/atlassian/jwt_spec.rb:56](#)



GitLab

Operations Insights

Operations Dashboard



What's in it?

- Snapshot view of a list of projects active alerts, last commit, pipeline status, and last deployment

Helps customers answer

- What is the health of my projects?

How to navigate?

- More -> Operations

A screenshot of the GitLab Operations Dashboard. The interface features a dark header bar with the GitLab logo and a search bar. Below the header is a title "Operations Dashboard". The main area contains a grid of project cards, each representing a different repository. Each card includes the project name, a small icon, the last commit message, the time since the commit, the number of alerts (indicated by a triangle icon), and the pipeline status (e.g., "running", "passed").

Project	Last Commit	Time Ago	Alerts	Pipeline Status
GitLab.com / www-gitlab-com	master -> d606993a	17 minutes ago	0 Alerts	running
GitLab.org / GitLab FOSS	master -> c1fc5da1	17 minutes ago	0 Alerts	passed
GitLab.org / gitlab-runner	master -> 058d0c5e	1 hour ago	0 Alerts	passed
GitLab.org / gitaly	master -> fde7ec76	2 hours ago	0 Alerts	passed
monitor / Tanuki Inc	master -> 3e7c9d3a	1 day ago	0 Alerts	passed
monitor / monitor-sandbox	master -> 234ad0fc	3 weeks ago	19 Alerts	passed

Environments Dashboard



What's in it?

- Environments across various projects

Helps customers answer

- Are there issues with one or more of my project environments?

How to navigate?

- More -> Environments

The screenshot shows the GitLab Environments Dashboard. At the top, there is a navigation bar with links for GitLab, Next, Projects, Groups, More, and a search bar. A green button labeled "Add projects" is also visible.

The main title is "Environments Dashboard". Below it, a sub-header states: "This dashboard displays a maximum of 7 projects and 3 environments per project. Read more." There is a breadcrumb navigation path: M monitor > M monitor-sandbox.

The dashboard displays three environments:

- production**: Shows a recent merge request from 'master' to '234ad0fc' (Merge branch '671-create-...') 3 weeks ago. The status is "passed".
- staging**: Shows a recent merge request from 'master' to 'fb4654d1' (Add new issue template) 3 months ago. The status is "blocked".
- production**: Shows a recent merge request from 'master' to '3e7c9d3a' (Merge branch 'use-basic-r...') 1 day ago. The status is "passed".

Each environment card includes a "View app" button and a three-dot menu icon.

Environments



What's in it?

- Single project environment details, ability to deploy, roll-back, monitor, stop applications in specific environment, access pod logs, web terminal & deployment history

Helps customers answer

- What is the status of my project environments?

How to navigate?

- Operations-> Environments

The screenshot shows the GitLab interface for managing environments. The top navigation bar includes 'GitLab Next', 'Projects', 'Groups', 'More', and a search bar. On the left, there is a sidebar with various icons. The main content area is titled 'Environments' and shows two entries:

- Available 9 Stopped 4**: This section header is followed by a table with columns: Environment, Deployment, Job, Commit, Updated, and Auto stop in.
- production**: Details: #147 by [user] (production #440437807). Commit: master -> 3e7c9d3a (Merge branch 'use-basic-report...'). Updated: 1 day ago. Actions: edit, delete, copy, and a red square icon.
- staging**: Details: #104 by [user] (staging #350205462). Commit: master -> fb4654d1 (Add new issue template). Updated: 3 months ago. Actions: edit, delete, copy, and a red square icon.

Each environment entry includes a summary card with the following details:

- Instance (1)
- 100% Complete
- Succeeded (green bar)
- Running (green bar)
- Failed (red bar)
- Pending (grey bar)
- Unknown (white bar)
- Canary (yellow bar)



GitLab

Security Insights
DevSECOps

Project Security Dashboard



What's in it?

- Overview of security vulnerabilities by criticality in the project - access static, dynamic application security, dependency & container scanning results

Helps customers answer

- How secure is my application?

How to navigate?

- Security & Compliance> Vulnerability Report

The screenshot shows the GitLab Vulnerability Report interface. The left sidebar lists project navigation options like Project overview, Repository, Issues (28), Merge Requests (3), Requirements, CI/CD, Security & Compliance (selected), and Vulnerability Report (selected). The main content area displays a summary of vulnerabilities: Last updated 4 hours ago (#251630637), with counts for Critical (25), High (51), Medium (127), Low (131), Info (0), and Unknown (0). Below this is a table of detected vulnerabilities:

Status	Severity	Scanner	Description	Identifier	Scanner	Activity
Detected	Critical	All severities	Exposure of Resource to Wrong Sphere in org.apache.tomcat.embed/tomcat-embed-core pom.xml	CVE-2017-5648 + 1 more	Dependency Scanning	1
Detected	Critical	All severities	Information Exposure in org.apache.tomcat.embed/tomcat-embed-core pom.xml	CVE-2017-5651 + 1 more	Dependency Scanning	1
Detected	Critical	All severities	Deserialization of Untrusted Data in com.fasterxml.jackson.core/jackson-databind	CVE-2020-9546	Dependency	1

Group Compliance Dashboard



What's in it?

- Latest merge request and approvers

Helps customers answer

- Am I able to gate changes made to my environments & view a history?

How to navigate?

- Security & Compliance -> Compliance

Merge Request	Approval Status	Pipeline	Updates
Resolve "Move CSS files into static resources" tanuki-corp-saf13 - created by: Sameer Kamani	ⓘ	✓	approved by: Pointy Haired Boss 26-move-css-files-into-static-resources into master merged 4 months ago
Resolve "Change the password field to fix the issue" my-tanukitech-spring!42 - created by: Sameer Kamani	ⓘ	✗	approved by: Pointy Haired Boss 59-change-the-passw... -field-to-fix-the-issue into m... ter merged 8 months ago

License Compliance



What's in it?

- List of licenses used in the application

Helps customers answer

- Am I compliant to my organizational policies on licenses?

How to navigate?

- Security & Compliance>
License Compliance

The screenshot shows the GitLab interface for the 'Tanuki Corp - Spring App' project. The sidebar on the left is collapsed. The main content area is titled 'License Compliance' and displays a table of detected licenses in the project. The table has two columns: 'Name' and 'Component'. The detected licenses include:

Name	Component
unknown	hibernate (2.1.8)
GNU Lesser General Public License	logback-classic (1.1.7) and logback-core (1.1.7)
Apache Software Licenses	log4j-over-slf4j (1.7.21)
MIT License	jcl-over-slf4j (1.7.21), jul-to-slf4j (1.7.21), and 1 more
GNU Lesser General Public License v2.1 only	javassist (3.20.0-GA)
Eclipse Public License 1.0	logback-classic (1.1.7) and logback-core (1.1.7)
Apache License 2.0	classmate (1.3.3), commons-dbcp (1.4), and 34 more
Mozilla Public License 1.1	javassist (3.20.0-GA)

Dependency List



What's in it?

- List of dependencies and known vulnerabilities for the latest pipeline run

Helps customers answer

- Am I vulnerable due to any of my dependencies?

How to navigate?

- Security & Compliance> Dependency List

The screenshot shows the GitLab interface with the sidebar open. Under the 'Security & Compliance' section, the 'Dependency List' option is selected. The main content area displays a table of dependencies with their details and vulnerability counts.

Component	Packager	Location	License
ch.qos.logback/logback-core 1.1.7	Java (Maven)	pom.xml	1 vulnerability detected
com.fasterxml.jackson.core/jackson-databind 2.8.4	Java (Maven)	pom.xml	10 vulnerabilities detected
org.apache.tomcat.embed/tomcat-embed-core 8.5.6	Java (Maven)	pom.xml	9 vulnerabilities detected
org.springframework.boot/spring-boot 1.4.2.RELEASE	Java (Maven)	pom.xml	2 vulnerabilities detected
org.apache.tomcat.embed/tomcat-embed-websocket 8.5.6	Java (Maven)	pom.xml	1 vulnerability detected
org.hibernate/hibernate-validator 5.2.4.Final	Java (Maven)	pom.xml	1 vulnerability detected
org.springframework/spring-core 4.3.2.RELEASE	Java (Maven)	pom.xml	2 vulnerabilities detected
org.springframework/spring-web 4.3.2.RELEASE	Java (Maven)	pom.xml	3 vulnerabilities detected
org.springframework/spring-webmvc 4.3.2.RELEASE	Java (Maven)	pom.xml	5 vulnerabilities detected
junit/junit 4.12	Java (Maven)	pom.xml	1 vulnerability detected
ch.qos.logback/logback-classic 1.1.7	Java (Maven)	pom.xml	
com.fasterxml.jackson.core/jackson-annotations 2.8.4	Java (Maven)	pom.xml	

Web Application Firewall (WAF) Statistics Report



What's in it?

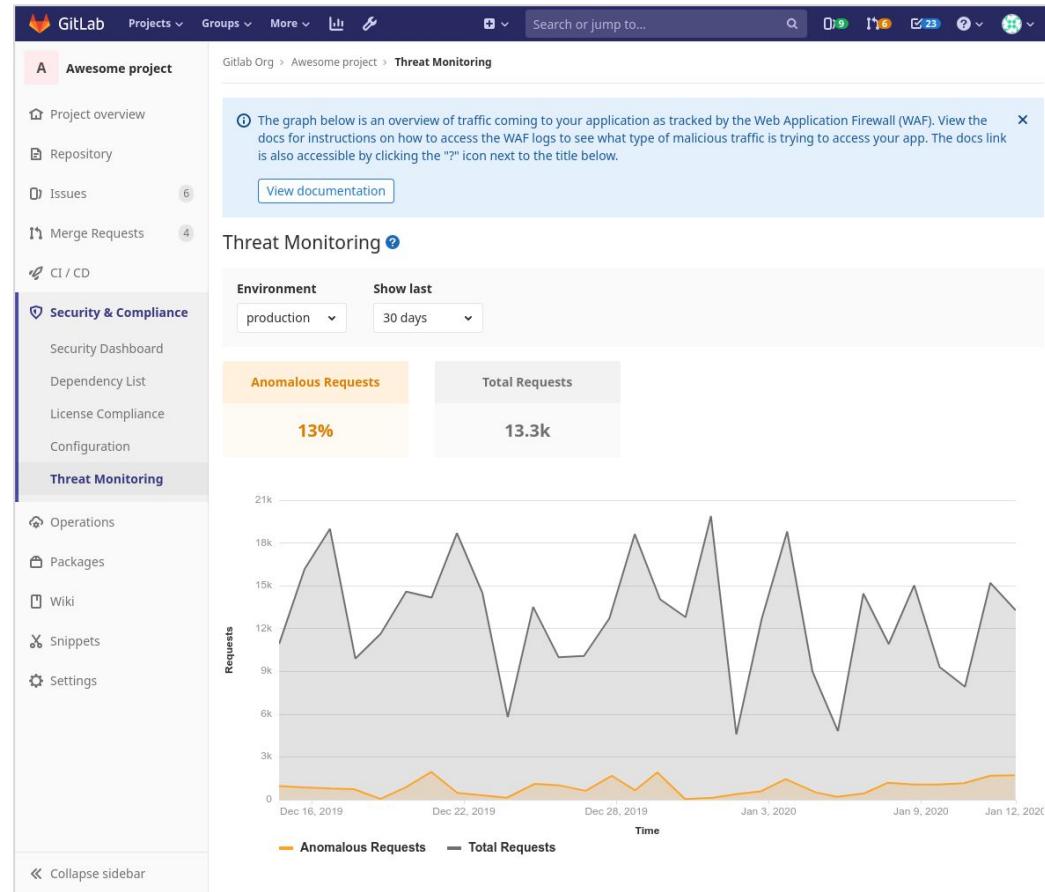
- Total and blocked traffic to determine how to configure, tune, and evaluate the Web Application Firewall

Helps customers answer

- What is my threat exposure?
Higher % of Anomalous requests imply higher threat exposure

How to navigate?

- Security & Compliance > Threat Monitoring





GitLab

Thank you!