CD with Kubernetes: The Prequel
Speaker

Priyanka Sharma
Director of Technical Evangelism - GitLab
Board Member - Cloud Native Computing Foundation
@pritianka
The Real Hero

Marin Jankovski (and his team)
Engineering Manager, Delivery
@maxlazio
Cloud native development and accelerating cycle time is increasingly important to business survival.

To succeed, engineering leaders must embrace CI/CD and be thoughtful of how to leverage new tech. This is GitLab’s story to moving a large scale production system to continuous delivery.

- The business imperative - cycle time compression
- How we did it - legacy first
- The value CD brought to GitLab
- DevOps - achievement unlocked
Business survival depends upon a radically faster DevOps lifecycle which can only be enabled by shifting left.
Speeding Up Release Cycle Time is Critical to Business

**Cycle time compression** may be the most underestimated force in determining **winners & losers in tech**.

— Marc Andreessen
How: Small, very small changes unlock velocity
But our reality was different

Release management at GitLab (2013-2018)

1. Monthly release on the 22nd
2. Rotating Release Manager role
3. Manual actions
4. Semi-automated
Cloud native development and accelerating cycle time is increasingly important to business survival.

To succeed, engineering leaders must embrace CI/CD and be thoughtful of how to leverage new tech. This is GitLab’s story of moving a large scale production system to continuous delivery.

- The business imperative - cycle time compression
- How we did it - legacy first
- The value CD brought to GitLab
- DevOps - achievement unlocked
Let’s make releases easier!

Release management at GitLab (2018)

- July - Release management team formed
- October - renamed to Delivery team with expanded mandate - Kubernetes migration
Our constraints

1. GitLab.com is a live system -> **No downtime**
2. GitLab still needs to release at the same cadence -> **No delays**
3. Migration to Kubernetes is a multi month project -> **No time**
4. Engineering organization ready for Continuous Delivery? -> **???
What the delivery team spent time (2018)

- Deploys: 60.0%
- Release tasks: 26.1%
- CE/EE merge: 10.3%
- RM QA: 3.6%
The game plan

Tackling 80% of the pie

This would bring:

1. **No** release **delays**
2. Repeatable and faster deploys to enable **no downtime**
3. More **time** for our Kubernetes migration
4. More space to **prepare** the org for Continuous Delivery
GitLab

GitLab runs on GitLab

VISIBLE
Real time view across the entire lifecycle

EFFICIENT
Collaborate without waiting

GOVERNED
Develop and operate with confidence
GitLab runs on GitLab, uses Ansible, and does not yet use k8s for CD.
Before
Good enough solution
Automation with existing tools

We’ve used a number of GitLab features:

1. GitLab CI scheduled pipelines with GitLab API to automate Deploys
   a. Create branches
   b. Cherry-pick merge requests
   c. Create issues
2. GitLab project mirroring for CE and EE
3. GitLab CI for reducing manual RM work
   a. Multi project pipelines
   b. Deployment
   c. Automated QA
4. GitLab Chatops to reduce context switching and do everything from Slack
   a. All release publishing tasks are triggered through Chatops
Expanded pipeline

1. Create an issue
2. Commit your changes
3. CI pipeline runs
4. Review app
5. Peer review & discussion
6. Approve changes
7. Merge, issue closed
8. Create a merge request

Syntax
- gtsk-prepare
- gtsk-migrations-assets
- gtsk-gitaliy
- gtsk-fleet
- gtsk-peek

@pritianka #velocityconf
Cloud native development and accelerating cycle time is increasingly important to business survival.

To succeed, engineering leaders must embrace CI/CD. This is GitLab’s story to moving a large scale production system to continuous delivery.

- The business imperative - cycle time compression
- How we did it - legacy first
- The value CD brought to GitLab
- DevOps - achievement unlocked
GitLab
Commit to Canary in 2 hours

@pritianka
@pritianka
#velocityconf

@pritianka
#velocityconf
GitLab
From weekly to daily deploys

@pritianka #velocityconf
GitLab

All developers in on call rotation within 3 weeks
Cloud native development and accelerating cycle time is increasingly important to business survival.

To succeed, engineering leaders must embrace CI/CD. This is GitLab’s story to moving a large scale production system to continuous delivery.

- The business imperative - cycle time compression
- How we did it - legacy first
- The value CD brought to GitLab
- DevOps - achievement unlocked
What does the Release team spend their time in 2019?

- CE/EE merge: 0.3%
- Release tasks: 17.8%
- Other: 82.0%
More frequent deploys

Number of deployments

- May 2019
- June 2019
- July 2019
- August 2019
Our culture shift

Quality is a priority

No hot-patching unless p1 and s1

Every engineer is on the on-call rotation
All aboard the k8s boat!
<table>
<thead>
<tr>
<th>Environment</th>
<th>Deployment</th>
<th>Job</th>
<th>Commit</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>gprd</td>
<td>#763 by</td>
<td>gprd-cny:upgrade #63...</td>
<td>master --&gt; ca50cc4b</td>
<td>14 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Revert &quot;Merge branch 'testfors...&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>instances (38)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>94%</td>
<td>Complete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gstd</td>
<td>#762 by</td>
<td>gstd:upgrade #630627</td>
<td>master --&gt; ca50cc4b</td>
<td>14 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Revert &quot;Merge branch 'testfors...&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>instances (2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td>Complete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pre</td>
<td>#761 by</td>
<td>pre:upgrade #630626</td>
<td>master --&gt; ca50cc4b</td>
<td>14 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Revert &quot;Merge branch 'testfors...&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>instances (3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>66%</td>
<td>Complete</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Resources

- Automated scheduled deployments design
- Automated deployments process
- Automated deployment transition
- Report from GitLab 10.4 release
- Kubernetes work
  - Kubernetes configuration decisions
  - Container Registry production readiness review
  - Container Registry work epic
Rate today’s session!

Cyberconflict: A new era of war, sabotage, and fear

9:55am-10:10am Wednesday, March 27, 2019
Location: Ballroom
Secondary topics: Security and Privacy

Rate This Session

We’re living in a new era of constant sabotage, misinformation, and fear, in which everyone is a target, and you’re often the collateral damage in a growing conflict among states. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. Moving from the White House Situation Room to the dens of Chinese, Russian, North Korean, and Iranian hackers to the boardrooms of Silicon Valley, David reveals a world coming face-to-face with the perils of technological revolution—a conflict that the United States helped start when it began using cyberweapons against Iranian nuclear plants and North Korean missile launches. But now we find ourselves in a conflict we’re uncertain how to control, as our adversaries exploit vulnerabilities in our hyperconnected nation and we struggle to figure out how to deter these complex, short-of-war attacks.

David Sanger
The New York Times

David E. Sanger is the national security correspondent for the New York Times as well as a national security and political contributor for CNN and a frequent guest on CBS This Morning, Face the Nation, and many PBS shows.
Thank you!

Priyanka Sharma
Director of Technical Evangelism - GitLab
Board Member - Cloud Native Computing Foundation
@pritianka
GitLab is the first single application for the entire DevOps lifecycle

<table>
<thead>
<tr>
<th>Manage</th>
<th>Plan</th>
<th>Create</th>
<th>Verify</th>
<th>Package</th>
<th>Secure</th>
<th>Release</th>
<th>Configure</th>
<th>Monitor</th>
<th>Defend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle Analytics</td>
<td>DevOps Score</td>
<td>Audit Management</td>
<td>Continuous Integration (CI)</td>
<td>Container Registry</td>
<td>Continuous Delivery (CD)</td>
<td>Auto DevOps</td>
<td>Metrics</td>
<td>Runtime</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Authentication and Authorization</td>
<td>Code Quality</td>
<td>Maven Repository</td>
<td>Release Orchestration</td>
<td>Kubernetes Configuration</td>
<td>Application Self Protection</td>
<td>Application Self Protection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Authorization</td>
<td>Performance Testing</td>
<td>Dependency Scanning</td>
<td>Pages</td>
<td>ChatOps</td>
<td>Web</td>
<td>Application Self Protection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NPM Registry</td>
<td>Container Scanning</td>
<td>Review Apps</td>
<td>Serverless</td>
<td>Firewall</td>
<td>Application Self Protection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>License Management</td>
<td>Incremental Rollout</td>
<td>Coming soon:</td>
<td>Threat Detection</td>
<td>Application Self Protection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Feature Flags</td>
<td>PaaS</td>
<td>Behavior Analytics</td>
<td>Application Self Protection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Release Governance</td>
<td>Chaos Engineering</td>
<td>Analytics</td>
<td>Application Self Protection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Runbook</td>
<td>Vulnerability Management</td>
<td>Application Self Protection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Configuration</td>
<td>Data Loss Prevention</td>
<td>Application Self Protection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cluster Cost Optimization</td>
<td>Container Network Security</td>
<td>Application Self Protection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Governance</td>
<td>Status Page</td>
<td>Application Self Protection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Coming soon:
- Code Analytics
- Workflow Policies
- Cycle Analytics
- DevOps Score
- Audit Management
- Agile Portfolio Management
- Service Desk
- Kanban Boards
- Project Management
- Source Code Management
- Code Review
- Wiki
- Snippets
- Web IDE
- System Testing
- Usability Testing
- Accessibility Testing
- Compatibility Testing
- Rubygem Registry
- Linux Package Registry
- Helm Chart Registry
- Dependency Proxy
- Secret Detection
- IAST
- RASP
- Container Scanning
- License Management
- System Testing
- Usability Testing
- Accessibility Testing
- Dependency Proxy
- Secret Detection
- IAST
- RASP
- License Management

#velocityconf
Ticketmaster migrated from Jenkins to GitLab to speed up build time.

15X faster builds

Public Case Study
Paessler AG automated QA tasks down from 1 hour to 30 seconds.

120X

Increased QA efficiency

Public Case Study
SVN was a blocker to adopting DevOps. Axway implemented GitLab and went from yearly to biweekly deployments.

26X faster release cycles

OKAY for now - replace with Goldman Sachs when we can