CLOUDBEES JENKINS

DOES COMMUNITY JENKINS EDITION STACK UP

Dept: CTP

Author: Rock Mutchler, Navin Chaudhar, Kirti Kalankar, Jason Fisher

Topic: Solution Analysis Decision

Date: 09/12/2018

CONNECTED TECHNOLOGIES

Jenkins Problem Statement

Community Jenkins Doesn't Stack Up

Open source is good in technical stacks but for enterprise class tools, it is a nightmare to support, that goes against all the DevOps and Agile principles. With all the instability coming from uncontrolled and untested development in these open source tools, it becomes an operational nightmare. There is no up-time guarantee and days and weeks are spent troubleshooting issues that just simply can be a bug in the tool code. This directly impacts the ability to deliver faster and quality code to the customer. This becomes more critical for a tool that will be utilized by each and every developer in the company, multiple times every single day. A small outage or performance issue can lead to the waste of thousands of costly man hours, with no official support to help.

Community Jenkins Bottlenecks

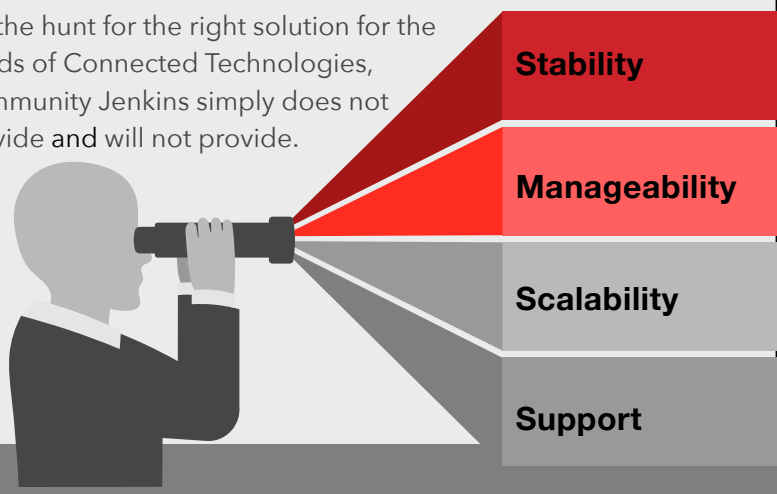
From the start because of the lack of evolving development and support the application requires antiquated and expensive infrastructure just to run. This is the start of the first bottleneck, **inconsistent infrastructure**. This means a different process to run the Jenkins application than other applications on the system. This is a whole separate set of infrastructure to then maintain and support which requires a different skill set. This leads to another bottle neck, **inconsistent environments**. The lack of enterprise support for plugins along with not having access to subject matter experts while running on an outdated system leads to lots of issues. One of these is errors not due to errors within the code but rather with application building and deploying the code to the environments. **This can cause days or even weeks of pure waste chasing down bugs**. This is one of if not the largest killers of agility. These hot fixes and constant context switching move us to the next bottleneck, **manual intervention**. This directly impacts the ability to deliver software in their software development lifecycle (SDLC). It does this by interruption, which then blocks the ability to deliver more often with a high degree of reliability and quality. It also causes mistrust of the pipeline.

Mistrust In CT's Pipeline Past

That's right, Connected Technology has felt the waste of an open source pipeline piece in the past, not long ago. Spinnaker caused all types of waste for the infrastructure team as well as the application team. Requiring a full, dedicated team to setup, personalize, manage and maintain. It's odd how if you add that up, a few hours and some time waste from a few teams how much more open source ends up costing. This is a recurring theme, and even with the past experiences and warnings it won't be the last we see of it.

Key Requirements

On the hunt for the right solution for the needs of Connected Technologies, Community Jenkins simply does not provide and will not provide.



Missing Items In The Box Necessary For Any Trust

The open source, community Jenkins simply does not and will not have many key items critical to any enterprise company. Here are a few items listed below, the entire list is too large for this document.

Distributed Architecture	Optimized infrastructure performance
Team enablement at exponential scale	Automated Proactive Updates and Validation
Enables parallelism for development teams	Release Management at Scale
On the Fly Resource Allocation	Scaled support without the need for additional resources
Manageability	Secure & Compliant freedom with governance for the entire organization

Community Jenkins Found Lacking

As with many quick, easy, or free technology short cuts something has to give. While using these shortcuts can be great for initial POCs or success for an unknown company, such as a startup, it quickly becomes a huge pain point. This has held true for decades, yet the process repeats as it is always thought using something else will not be needed or that it can be done without. It's truly sad to watch over and over, so collectively we decided to break this cycle on the CTP team in Toyota Motors NA Connected Technologies.

Supporting information of this decision can be found here:
https://confluence.sdlc.toyota.com/x/XGG7Bg

Wrong Steps With Open Source Jenkins

- So we have entrusted a critical operational function to an open source community, will it be there tomorrow?
- Many plugins, which one do we pick? Does it exists? Is someone still maintaining it?
- How many times are we going to let a single open source plugin in a large list of 'needed' plugins break all builds and halt all the teams while yet a more expensive in-house fix is found or created?
- 4 - 6 full time engineers to manage, maintain and fix the ever growing complexity with an outdated Jenkins platform?
- Stuck with old infrastructure which is expensive to run and even more expensive to keep maintaining. Tried using the dedicated engineers for other projects but the context switching back to Jenkins all the time creates an unproductive and unfulfilling position.

Painful Lessons Learned

01

Upgrades are a Nightmare

Because community Jenkins releases versions bi-weekly, it becomes an operational challenge to identify a stable and workable version that doesn't break existing pipelines and jobs. No official support makes it very complex. Enterprise version releases only tested versions.

02

Buggy Plugins

Like the product itself, the plugins are open sourced where developers can contribute to any plugin and there is no way to check if it has been tested properly. This problem is known to generate issue where all the pipelines can just vanish because of a broken or buggy plugin that just misbehaved or it can be as serious as Jenkins crashes where the Jenkins doesn't start at all.

03

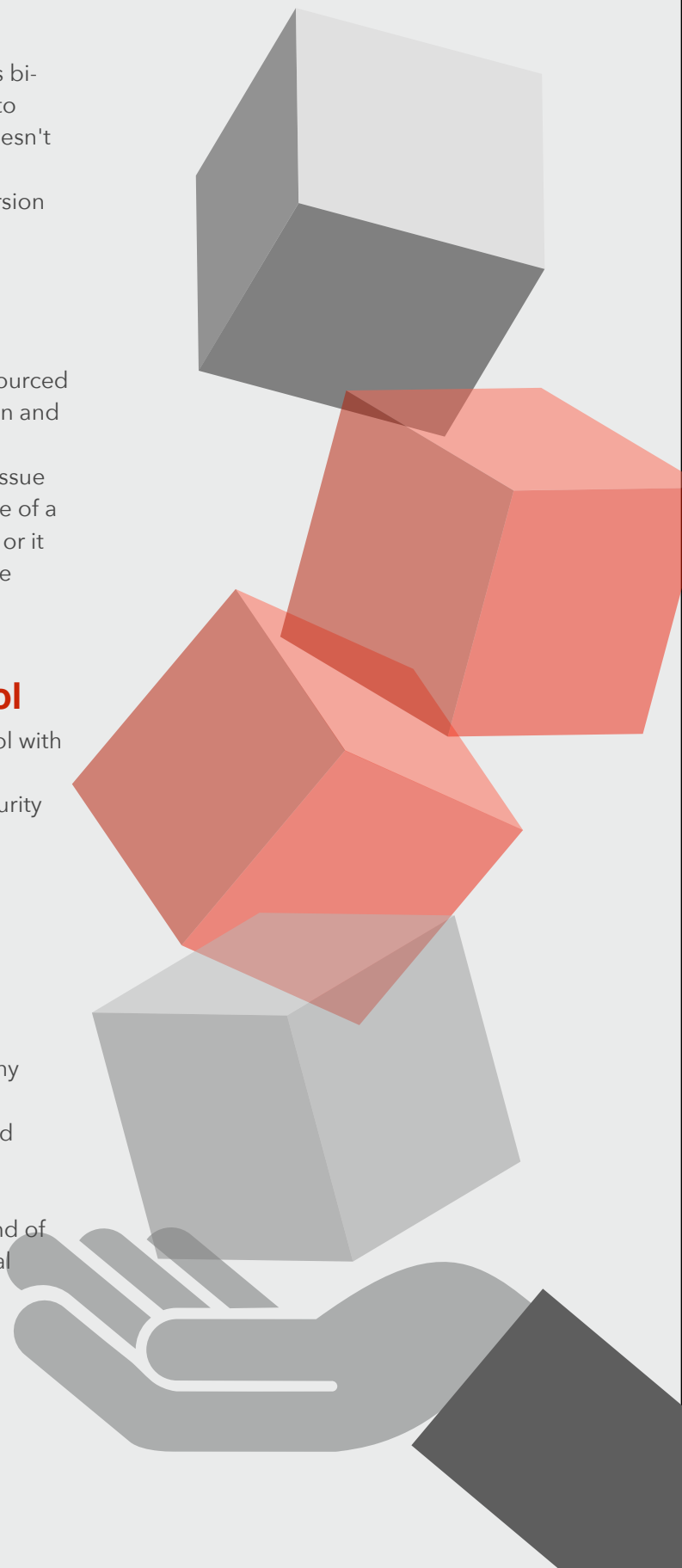
Security and Access Control

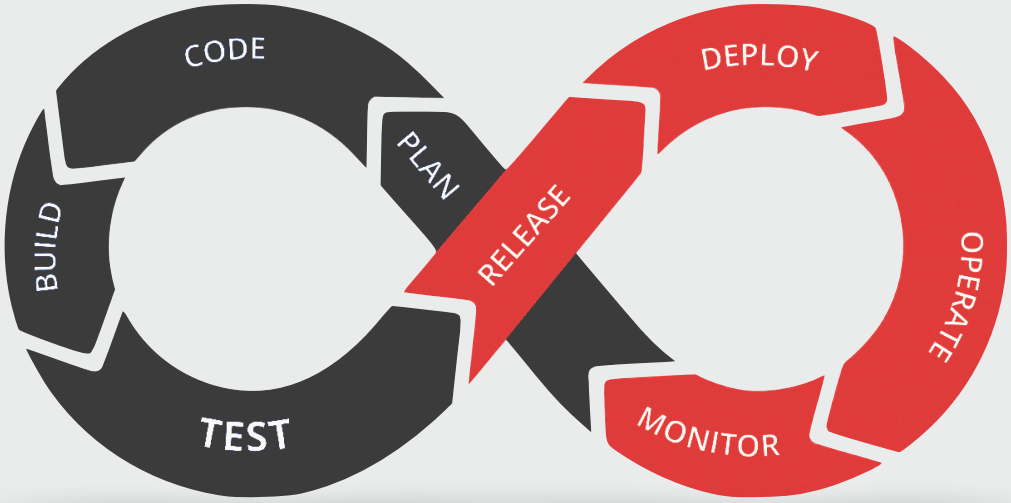
Centralized governance for user access control with simplified management tools and utilities. Reducing waste in user management and security errors from manual mistakes on permissions. Audit-able central management for security

04

Scalability Issues

Community version of Jenkins doesn't offer any scalability. A single master Jenkins starts experiencing extreme slowness, perm-gen and heap issues because of limited hardware resources. Enterprise version brings in the possibility of multiple masters where these kind of issues can be avoided, without any operational





Quality by Toyota Motors NA - Connected Technologies

There is simply too much risk for no reward in not purchasing Enterprise CloudBees Jenkins. Members of the team with decades of experience with Jenkins Community as well as research and Proof of Concepts have been completed. Past nightmares of outages, data loss, hours of manual intervention, and plenty of sleepless nights all come at not only a very high price it is also against everything that Toyota stands for and represents to the World.

Producing Winners Every Second

Right around the corner, Toyota CT will be joining the ranks of many that have benefited greatly from the support of the leading pipeline company, Cloudbees Jenkins. There are 100's if not 1000s of stories out there. The accounts below are just a few examples of the real impact of having a supported, stable, and scaleable pipeline. Conversely, there are not these kinds of accounts for using the open source solution.

Sourced from https://www.cloudbees.com/customers

accenture
High performance. Delivered.

"Our development teams that use the Accenture DevOps Platform with CloudBees Jenkins Enterprise can concentrate on innovation. They don't have to worry about setting up and managing their tooling environment."

– Andi Strain, Accenture DevOps platform lead

Results

- ✓ 10,000 engineering days gained annually for innovation
- ✓ 80+% time reduction in Jenkins maintenance across the platform
- ✓ DevOps tool provisioning reduced from weeks to minutes

Capital One

"CloudBees Jenkins Platform is by far the most heavily used. That fact alone leads me to believe it is either the easiest to use, it has the most community love or it is simply the best from a technology perspective."

– Brock Beatty, Director of software engineering

Results

- ✓ 90% of pipeline automated
- ✓ Deployment frequency increased 1,300%
- ✓ Engineers focused on application development, not infrastructure
- ✓ Quality and security ensured through repeatable processes