



Industry:

Higher Education

Geography:

North America

"If there's one word to describe CloudBees' support team, it is "consistency." I always get expert and easy-to-understand answers to my questions no matter how trivial they may seem to me."

Bryan Teague
Senior Web Administrator
University of San Diego

Asking for Help Is a Best Practice When Building a CI/CD Pipeline

Contributed by Bryan Teague, Senior Web Administrator, University of San Diego

I try to learn something new every day. That's why I love this job. As the senior web administrator at the University of San Diego (USD), I am surrounded by smart people who are always challenging me to expand my knowledge of DevOps and programming.

I look forward to coming in every morning because my work here changes every day. USD's IT web services are centralized, and we do everything in-house. Nothing is outsourced. The university web services team handles all aspects of the university website, from the initial specification to UX design, coding, and deployment.

You Get One Chance to Make a First Impression

We have to get everything right because the website is often the first point of contact with the university for both internal and external users. If it isn't displaying and functioning correctly or taking too long to load, the user experience suffers, and this can have severe repercussions for the USD community.

For example, we have an online directory with profiles and pictures of faculty and staff that is also a gateway for various online tools and services. Instead of operating separate access points for individual areas within our system, we manage permissions at this central location. If our website fails or otherwise underperforms, our people can't do their day-to-day work.

USD's website is also an information portal for prospective students, and that gives us one chance to make a first impression. If our user experience is poor or our site fails to provide timely, relevant information, applicants may choose another school.

Shifting from Managing Content to Building User Experiences

I've been here since 2014. In that time, the web team's role has shifted from maintaining, managing, and hosting content to designing end-to-end user experiences. This approach to our work is satisfying, and we have won awards for our efforts. I am very proud of what we've accomplished here and am always looking for ways to improve our performance and our services.

One of the things I've done is to put together a new system that allows the web services team to focus on development instead of unit testing. In the past, my developers were validating code manually, but I decided to automate the process by offloading it to a CI/CD server. I also implemented a workflow that incorporates industry best practices and standard programming concepts like code reviews to accelerate our deployment cycle and to help our coders learn from one another.

Recognizing That We Needed Faster Answers

Our CI/CD pipeline combines a Jenkins server and a GitHub repository. I chose these tools because they're open source. The price was right and, in most cases, we can leverage the wisdom of the wider community to resolve our issues.

But there are also times when we need fast answers and a deeper level of support than the open source community can provide. For these reasons, we purchased a yearly Jenkins Support Plan from CloudBees.

CloudBees is the world's leading provider of commercial Jenkins pipeline solutions, and their people are also very active in the open source movement. I was already on a first-name basis with members of their team because I'd talked to them months before we subscribed to their LTS support service.

An Upgrade That Went Really Wrong

The web services team is on its second Jenkins environment. We began with an ancient pre-pipeline version that gave us code automation and a few other workflow improvements. That implementation worked right out of the box, but we started having trouble when I tried to upgrade our environment to a more recent build.

The update crashed our server. It took me several days to get it back up and running, but things were never the same, and I could only provide basic features to the team. Our auto code deployment plugin appeared to be incompatible with the new version. It seemed impossible to upgrade our environment any further, and so we limped along with an outdated build that lacked this mission-critical feature.

As this drama was unfolding, I discovered that another of our ITS teams had deployed an up-to-date Jenkins server of its own. It was not only stable but was also running the CI/CD pipeline I wanted to deploy for my web development team. So, I struck up a deal to work together, and I moved our projects to this new server.

Unfortunately, the new environment wasn't already loaded with the same auto code deployment plugin that had hindered our previous upgrade attempt, and I didn't know how to fix this. This time, I accepted that the issue was beyond my capacity to resolve, and I turned to CloudBees for help.

Securing Funding for CloudBees Jenkins Support

Securing funding for the maintenance plan turned out to be easier than I expected. My contact at CloudBees laid out our options and quoted an annual price. The Gold package proved more than enough for our needs, and I pitched it to my superiors.

It was a solid value proposition and an easy sell. Since we were using Jenkins, we could divert some of the money we were saving to support. For an annual fee that was less than many consultants charge per week, CloudBees provided 24x7 Jenkins support, four-hour turnaround times for our questions, and upgrade assistance.

That last part sealed the deal for my management team and me. The CloudBees support team mapped out the steps I needed to follow when upgrading our server to use the required plugin and supplied a fallback plan in case I needed to turn the back the clock to our older installation. They also threw in a 30-minute phone call before the actual upgrade so we could review the entire plan step-by-step ahead of touching the server.

"CloudBees has taught me that there's no such thing as a stupid question. Jenkins can become a tangled mess, especially when plugins aren't playing well with one another. On such occasions, a silly suggestion can spark the idea that resolves a complex issue."

Bryan Teague
Senior Web Administrator
University of San Diego

There Are No Stupid Questions

I've used this service for three upgrade cycles and countless other technical issues, and everyone there knows their stuff. If there's one word to describe CloudBees' support team, it is "consistency." I always get expert and easy-to-understand answers to my questions no matter how trivial they may seem to me.

I know it's a cliché, but the folks at CloudBees have taught me that there's no such thing as a stupid question. Jenkins can become a tangled mess, especially when plugins aren't playing well with one another. On such occasions, a silly suggestion can spark the idea that resolves a complex issue.

Three Key Steps to a Successful Jenkins Upgrade

One of the best things CloudBees support supplied was a sure-fire method to ensure that every upgrade is a success. There are three parts to this approach. First, make sure your backups are fully functional. If you're missing a component—even a relatively minor plugin—your Jenkins installation is going to fail.

Second, take a snapshot of your entire system just before you perform an upgrade. If something goes wrong, you can roll back to your state from a couple of hours ago instead of reloading a stale backup that is days or weeks old.

Finally, back up your actual Jenkins directory. That way, you have a double failsafe and two separate recovery points should things go wrong.

CloudBees Jenkins Support Is Right for You

If you're running a local Jenkins installation, you need CloudBees Jenkins Support. Especially if you're like me, and you don't know all the ins and outs of Java (yet).

If you're doing high-speed development and running a CI/CD pipeline, you don't have the luxury of time when it comes to repairing and rebuilding your Jenkins servers. CloudBees Jenkins Support will hold your hand and walk you through the upgrade or repair process.

Even if your company does everything else in-house, support is the one thing you should outsource. CloudBees provides year-round assistance at a fraction of the cost of adding a full-time member to your team.

The return on investment is tremendous, but let me put it another way. You buy maintenance contracts for your hardware, so why not also protect your pipeline?

Building the Bedrock of Our CI/CD Pipeline

CloudBees' support gave me the boost I needed to build a rock-solid Jenkins server as the foundation of our CI/CD pipeline. It has elevated our capabilities with Jenkins. As a result, we have gone from a rudimentary configuration to our ideal development environment. We now have the tools to concretize some of our long-standing goals.

Over the next two years, we are redeveloping all our sites and building a new design system using a React front-end. Our goal is to develop a workflow that will allow us to do front-end testing before we roll out changes to the public without derailing us while we work on our internal servers.

"The return on investment is tremendous, but let me put it another way. You buy maintenance contracts for your hardware, so why not also protect your pipeline?"

Bryan Teague
Senior Web Administrator
University of San Diego

Our CI/CD pipeline has given us the bedrock we need to automate unit and regression testing, and to find ways of ensuring backward compatibility while still moving forward. I am looking at a variety of front-end display, CSS, JavaScript, and PHP testing tools, and exploring ways to tie them into our automated code deployment workflow.

It helps to know that we can develop everything in-house and then deploy to our build server, to our staging server and have everything work the first time. I am also working with my developers to fully integrate Jenkins with GitHub. That will be the last step in our transition to full production deployments.

Asking for Help Is a Best Practice

Working at USD has its share of advantages. My work here is ever-evolving, and the university provides excellent opportunities for professional development. Every year, I get to attend two conferences or trainings of my choosing.

Last year, I selected the DevOps World|Jenkins World conference in San Francisco. I am learning everything I can about DevOps, and during that week, I was able to benefit from contact with the industry's leading developers.

For five days, I indulged my curiosity and asked a lot of questions. It was a privilege to be surrounded by so much brainpower and expertise. I never felt like the smartest person in the room, and that put me at ease because I was there to learn.

Don't get me wrong. I know my stuff, but I am also aware of my limitations. It's good to recognize where your knowledge ends because that's where learning begins. Some people are afraid to ask questions because it makes them look foolish, but the opposite is true. Not asking is even sillier.

If you want to improve personally and professionally, individually or collectively, you have to embrace a simple truth: Asking for help is a best practice.

"We have gone from a rudimentary configuration to our ideal development environment. We now have the tools to concretize some of our long-standing goals."

Bryan Teague
Senior Web Administrator
University of San Diego

Learn More About The University of San Diego

www.sandiego.edu