



“CircleCI allows us to be confident in the software we build before taking it out for a drive.”

Adrian Macneil,

Director of Infrastructure Engineering
Cruise Automation



Why CircleCI Enterprise and GitHub Enterprise?

The race to be the first to create a self-driving car for the masses is on. Cruise Automation (a subsidiary of GM) is in charge of taking the fully electric Chevy Bolt and making it fully autonomous. What does it take to lead? For Cruise, one strategy they use to zoom past the competition is setting up key systems and tools that allow their team to accelerate progress, without causing unnecessary friction. Cruise chose CircleCI Enterprise and GitHub Enterprise as the tools that would help their team move quickly and safely to getting autonomous vehicles on the road.

The Challenge Prior to CircleCI

Building a self-driving car is a complex challenge, to say the least. The driving software needs to control the physical vehicle, obey traffic laws, follow map directions, and respond instantaneously to any situation on the road. Prior to using GitHub Enterprise, the team at Cruise was using GitHub.com. While the level of familiarity the team had with the tool was great, Cruise needed a tighter level of control over their code, and who had access to it.

“

If you want to get up and running quickly, without a lot of pain, CircleCI makes it easy for you to get building, instead of spending time trying to make your tools work.

”

On the continuous integration side, their previous tool was unreliable, and had a substantial amount of custom code which their team had to maintain at scale. In addition, it was difficult for engineers to see which builds were currently running, and when the team needed to diagnose the cause of failures, they had no easy way to access their log output or build artifacts.

When Adrian Macneil started as Director of Infrastructure Engineering at Cruise, he thought there could be a better way. Their time was much better spent on building software than on maintaining tools, and he knew there should be a way to balance the ease-of-use of GitHub with the necessary security features.

About Cruise Automation

Cruise is building the world's best autonomous vehicles to connect people to the places, things, and experiences they care about.

Industry: Automotive

Languages: C++, Python, Node.js

Stack: GitHub Enterprise, CircleCI

Enterprise, Docker, Slack, Coveralls

Deploys to: Cars!

Past CI/CD: BuildBot

The Solution

Adrian quickly landed on GitHub Enterprise and CircleCI Enterprise as the right tools for the Cruise team. The seamless integration between the two made the choice even easier. The team feels a sense of relief knowing that their code is only accessible behind their firewall on GitHub Enterprise. “Our code is our secret sauce,” said Adrian. “It’s nice knowing that it’s running on our own servers and network that we have full control over.”

On the CircleCI Enterprise side, they gained flexibility and performance in order to run their very complicated build and test suite. “As you can imagine, it’s kind of complicated to build an autonomous vehicle,” said Adrian. “We need to run really powerful machines to simulate our vehicle software. We’ve loaded up CircleCI with AWS GPU instances, which allows us to run many more tests and simulations than we were able to previously.”

For Cruise, the biggest benefit of bringing in CircleCI was that it enabled the team to move faster, without compromising quality or safety. At Cruise, every piece of code gets a physical road test with an engineer in the car before it is merged into master. CircleCI enables Cruise to run many more simulations than they could previously, which gives them more confidence in taking code to a road test. The ability to easily split tests and simulations across multiple containers is one area that Adrian really appreciates. “The ultimate test for us is taking our code on the road. Every piece of code gets a road test before it is merged into master. CircleCI allows us to be confident in the software we build before taking it out for a drive.”

Support from the CircleCI team is also something that Cruise values. “CI is constantly evolving. We’re constantly tweaking our environment, and running custom workloads,” said Adrian. “It’s really important for us to be able to ping someone at

CircleCI and say, ‘here’s what we’re trying to achieve, any advice?’”

Adrian also feels that using GitHub Enterprise and CircleCI in combination has increased team collaboration. “One side effect from using both GitHub Enterprise and CircleCI is that collaboration between teams and projects has increased. Our engineers can easily create new repositories for smaller projects and libraries, and set up CI without any infrastructure support. We can allow our teams to have that creativity and flexibility because we trust that everything is internal to Cruise.”

“

With CircleCI and GitHub as your tools, ramp up time shrinks, and we see that as a competitive advantage.

”

Onboarding is also faster because GitHub and CircleCI work so seamlessly together, which allows every new employee to be productive on day 1. “New developers aren’t sitting there fighting the tools, or going into a course on how to commit source code to the repo. This is a legitimate issue at most large companies – they run tools that aren’t familiar to the team, and so they have to invest a lot in training and onboarding. With CircleCI and GitHub as your tools, ramp up time shrinks, and we see that as a competitive advantage.”

“If you want to get up and running quickly, without a lot of pain, CircleCI makes it easy for you to get building, instead of spending time trying to make your tools work. That was a big reason for us to choose CircleCI, instead of building something custom,” said Adrian. “Sure...but, that would take months of work and multiple engineers. It doesn’t make sense to do that when you have a pain point now. Engineering time is more valuable if spent solving your unique business problems.”

