



Flux7: Building Self-Driving IT on Amazon Cloud

There's a lot of talk in the industry about DevOps and infrastructure as code, but that's not the full journey today. The road to Self-Driving IT is. This paper details how more enterprises are building automated, Self-Driving IT organizations that are more agile and cost effective than traditionally manually-driven enterprises.

Self-driving cars are not only a technical reality today; they are a practical reality well on its way. And they likely will soon be a widely available reality. For instance, Business Insider's Business Intelligence unit estimates there will be 10 million self-driving cars on the road in the United States by 2020. And these cars are going to do a lot more than make it more convenient for their owners to shuttle themselves around town. Self-driving cars are going to help make roadways much less congested, help to reduce accidents, and make it possible for many city dwellers to not even own a car, and free their money for other things in their lives. DevOps - and, pushing those concepts further into what we like to call Self-Driving IT - are doing the same thing for enterprises today.

Self-Driving IT is a way of automating away all of the drudgery and tediously repetitive events. It's about streamlining the technological overhead, automating the approval processes, automating the auditing and logging systems, and automating deployment so when someone needs a new application or service they can summon it, and it's ready. Just like a self-driving taxi.

When it comes to Self-Driving IT, many organizations have tried to make the transition themselves. Some are successful; however, many are not. That's because there's nothing inherently easy or straightforward when it comes to an organization moving to DevOps. And while many try to make it seem so, DevOps is much more than a mere collection of toolsets or processes. An organization doesn't simply install new configuration-management tools, hold more meetings requesting better collaboration among development and operations teams, and transform itself into a DevOps organization.

Successful DevOps is actually the culmination and proper orchestration of a collaborative mindset among teams and the right set of continuous processes, all supported by the right set of tools – and getting there isn't easy, especially for existing enterprises with legacy processes and technology in place. In many organizations, some of the greatest challenges come from people replacing bad habits with better processes, achieving efficiency by better collaboration, and permanently breaking the barriers between Dev and Ops.

The Vision of Self-Driving IT

Such a high level of collaboration is absolutely essential. Unfortunately, considering the subpar state of communication and teamwork in many enterprises, it's something much easier said than done. Too often, security, operations, internal compliance, and others are offices of "no." They tell developers "No, this isn't possible to do within policy," or "No, you can't do this." Most of the time, it's simply not accurate. And, fortunately, Self-Driving IT can help to alleviate a lot of this friction.

Consider how self-driving cars can automate many of the practices we want to enforce on the vehicle: it would stay within certain geographies, not exceed certain speed limits, return home after a certain time, and so forth. The solution with enterprises and Self-Driving IT is much the same thing: automate as much of one's IT efforts as can be automated. Rather than IT enforcing every rule manually, the staff instead spends that time building the right tools to enable developers to function as autonomously as possible and, in this way, instead of the personal interactions that too often can lead to communication and process breakdowns, the rules are enforced through automation.

What this also means is that with Self-Driving IT, there is much less time wasted waiting. Developers don't wait on operations, and operations does less manual work – and it never waits on any tasks that can be automated. This works in real production deployments by having one-click creation buttons for the infrastructure so that developers can have the resources they need, as well as automated fail-over mechanisms so that developers don't have to focus on IT. If developers need an environment to create a new application, they push a button for that specific environment. If they need multiple test environments, they can be created just as simply. Just like self-driving cars, users can often just push a button and get to where they need to go whenever they need to get there.

This is how automation helps to reduce friction between developers and IT. This is the promise of DevOps and of Self-Driving IT. In the next section we detail the technologies that help enable Self-Driving IT to become reality.

The Technology that Powers Self-Driving IT

At Flux7, we build our clouds on Amazon Web Services, the cloud-computing platform from Amazon.com. Most IT professionals are aware of the basic services provided in AWS which include Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3).

There are many things that we at Flux7 appreciate about AWS. The most important, considering self-service IT being the goal, is that AWS is straightforward to automate, and its automation tools are growing increasingly more powerful.

Rather than IT enforcing every rule manually, the staff actually spends that time building the right tools to enable developers to function as autonomously as possible.

Below are some tools that are key to building Self-Driving IT on Amazon Web Services:

AWS CloudFormation

AWS CloudFormation provides developers and operations teams a way to create and manage a collection of AWS resources, including well-orchestrated updating and provisioning. We've invested considerably in building CloudFormation tools that make it straightforward to create AWS resource-allocation templates.

AWS Command Line Interface

The AWS Command Line Interface (CLI) provides a centralized way to manage AWS resources from a command line that makes it easy to automate tasks through scripts.

Amazon Route 53

As a cloud Domain Name System (DNS) web service, Amazon Route 53 provides a reliable way to direct users to Internet applications by translating numeric IP addresses to memorable names, just like on the Web.

AWS Identity and Access Management in the Cloud

AWS Identity and Access Management (IAM) enables organizations to control access to their AWS resources with legacy identity tools such as managing users within groups, and establishing permission-based access.

AWS CodeDeploy

AWS CodeDeploy automates code deployments to on-premises systems and Amazon EC2 instances. This makes releasing new features, without downtime, much easier. Organizations can utilize this feature to avoid time-consuming manual procedures.

AWS CloudTrail

AWS CloudTrail provides visibility into cloud services that users need for security and regulatory compliance requirements. CloudTrail records AWS API calls and sends the appropriate user the log files. The events logged include identity and time of calls to APIs, source IP addresses, what was requested, and more. This is crucial for security-minded enterprises, organizations that are subject to regulatory compliance mandates, or those that want to track resource changes.

Flux7 Recommended AWS Ecosystem Toolsets

Docker

Moving to the extended Amazon AWS ecosystem, we find Docker essential. When you consider how companies achieve Self-Driving IT, it's through self-service portals. Docker, as an open-source tool that automates software deployment within containers, provides a way to do exactly that. Docker provides a way for secure, automated deployments and helps to separate duties between Dev and Ops. Developers, for example, can do whatever they need to do for their job within the container's environment, but IT controls what those container environments are and how they operate.

Ansible

The next tool that we find essential to DevOps and Self-Driving IT is Ansible. Ansible provides powerful software-deployment, task-execution, and configuration-management capabilities. And while Docker is powerful, it can't meet all environmental use cases. One such use case is Windows support. Another use case we see from time to time is in organizations that want to manage their systems more like traditional virtual machines than containers.

Additionally, there are situations where legacy applications prohibit the move to containers. For example, we work with a company in the energy sector whose application is using UDP packets, which Docker networking doesn't yet support.

Jenkins

Jenkins is a way for organizations to create continuous integration software development pipelines. Jenkins supports AccuRev, ClearCase, CVS, Git, and Mercurial, among many other software configuration-management toolsets. Flux7 uses Jenkins because it is very flexible. Flux7 also uses Jenkins for IT automation, which is a use case Flux7 created in response to customer demand. As always: if Dev and IT use the same tool, it inherently improves the move to DevOps and Self-Driving IT. Users can create in any language they need and it is very easy for new hires and clients to learn. Jenkins saves everything on the fly, making backup disaster recovery very easy. Jenkins can be completely automated. And the ecosystem supporting it is vast, so there are plug-ins for everything.

In fact, every time a new service or feature is created in other tools like Ansible, Docker and AWS, the user would see a Jenkins plug-in released nearly immediately.

When creating Self-Driving IT with toolsets such as these, at this point there's really no piece left in your architecture that is manual. These tools, along with the right processes, help create an extremely clean and fully Self-Driving aspect of your architecture. In the next section we will look at what these environments look like in action with real-world customers.

These tools, along with the right processes, help create an extremely clean and fully Self-Driving aspect of your architecture.

Self-Driving IT in Action

Flux7 Self-Driving IT case study: State Health Agency

This large state health agency found itself held back by its traditional IT services provisioning system. While the state agency was utilizing cloud services, it was not able to get the agility and rapid-deployment capabilities that it had initially sought.

Provisioning time after filing a request for IT services, such as a server, would take as long as four

months because of the approvals that had to be gathered. While that sort of latency is certainly common in organizations today, there's no longer any technical reason why it should take so long to deliver new IT services. The agency called Flux7 to have a look at its IT operation, and see what could be streamlined and improved.

As Flux7 assessed the state health agency's environment, it identified a pattern of applications being created for use on a few, standard set of architectures. After Flux7 identified a core set of application and infrastructure patterns, its team created sets of deployable public cloud templates. Flux7 placed these templates on a dashboard for teams to simply select the template that was needed; when the template was selected, its specific infrastructure – one that fully matches policy and regulatory needs – launched into production. The agency's security team and the office of the CTO closely vetted each template before it was placed on the dashboard.

This created very powerful benefits for this state agency. First, no time is wasted in approvals: everything, except for budget, is approved. Second, it's fully automated. The new environment is created and deployed into production with all the security and best practices preapproved.

Additionally, all of that automation didn't negatively impact the agency's ability to monitor its systems. In fact, its ability to monitor and audit its systems improved dramatically. The agency can now track and log the entire process, from the creation of the infrastructure to its use throughout production.

The result: Provisioning time was dramatically reduced – what used to be four months actually became about four minutes – and provided for significant agility, cost savings, and auditability.

Flux7 Self-Driving IT case study: Clinical research organization

This clinical research organization (CRO) wanted to improve upon its development time. The process to get new apps and services from idea to production was just taking too long. One of the biggest bottlenecks was its manual Quality Assurance (QA) process. When new apps or services were ready for review, they were submitted to the QA team for assessment. And despite the clinical research organization having an adequately staffed QA department, the time for QA to complete its review was taking too long and slowing the enterprise's ability to innovate.

The CRO's 20-person QA group, however, wasn't spending its time reviewing code. No, the members were actually spending a good portion of their time waiting on the QA test environments they needed to conduct their work. Further, the QA teams didn't have the autonomy to create their own test environments, and it wasn't possible for any single QA tester to have more than a single test environment at a time.

The CRO requested Flux7 have a look. Flux7 created a dashboard that would automatically provision the testing environments the QA teams needed. Then, every time someone in QA needed an environment they could simply select the environment they needed, and their preapproved environment became available. The dashboard also logged all QA activity. Through role-based test environment provisioning, QA testers also can now have multiple environments concurrently, which increased QA throughput even more.

The result: This CRO now enjoys a number benefits from its Self-Driving IT effort. First, there is no more

time wasted within the QA process. When software is ready for QA assessments, the teams are ready to get started.

There is no more time wasted within the QA process.

Flux7 case study: Software enterprise brings Self-Driving IT to corporate training

This leading software-maker for board-and chip-design manufacturers had an interesting challenge. Twice a year, for about 16 days, this company provides in-depth training to its customers and new hires, where all parties come to the organization's HQ and get in-depth product and service training.

The company tried everything it could to provide students the computing resources needed for their class. But it was trickier than it seemed at first blush. The IT team would get a list of all students and instructors ahead of training, and virtual machines would be created. And, on the first day of class, students would receive their own notebooks. But this meant that enrollment had to be frozen 15 days prior to the start of class. This enrollment freeze wasn't very agile: The number of students attending would change, instructors would need to change their virtual machine images for class, and 600 or more students trying to download virtual machines on the day of class stressed servers and bandwidth to their limits.

The manufacturer asked Flux7 to evaluate how the process could be improved. Flux7 automated the entire instructor course-creation process, so that instructors could select the course and the infrastructure they needed from a dashboard. Within their dashboard, instructors can customize the environment to suit their needs. The instructors are free to customize their classes up until the very day their class is scheduled to begin.

When students arrive for class, they simply log into the student portal, log into their class and launch the instance they need.

The result: The bottlenecks around server and class creation were eliminated, with the added bonus that instructors can customize their classes up until the last minute, and students get the environments they need to learn. This led to a 70 percent savings over previous years for the training program.

Flux7 on Self-Driving IT

We've conducted a lot of these engagements, over the course of which we've perfected a number of processes that we now follow as best practices. We pair these "lessons learned" with a very extensive tool chain that we have built in-house—all of which result in engagements moving quickly. To succeed, we work with our clients in three phases: Assess, Attune, and Engage."

Many of our lessons learned enable things to move fast. Here are the three phases:

The Assess Phase

In this phase, Flux7 assesses the state of the environment, workflow, and business requirements, and then designs future technology, workflow, and the game plan to get there. Once we have the game plan, we optimize it into a course of action that makes the most sense for a specific organization's situation. This is trickier than it may at first seem. If an organization moves too slowly, the move to Self-Driving IT can falter and stall. If the move is from zero to 60 in two seconds, oftentimes wheels spin and can't absorb the acceleration.

The Attune Phase

Here, we execute on the plans developed during the Assess phase, and create the Self-Driving infrastructure for internal consumption, and optimize the automated delivery of IT services based on the goals established. This Self-Driving IT is created in-line with all internal policies and is fully regulatory compliant. It is a way for organizations to rapidly move forward on cloud initiatives, including so-called "lift and shift" cloud migration, greenfield cloud infrastructure deployment, and specialized environments in the most stable, secure, compliant and auditable way possible.

The Engage Phase

This phase comes after the enterprise has reaped value from the Self-Driving IT initiatives built for them, and is ready to internalize its own expertise to create the same architectures. To provide this knowledge transfer we follow a model familiar to anyone who has worked in the culinary industry. In culinary, if one wants to learn from the course they don't just sit in a class. They observe in the chef's kitchen for a few weeks, and then the students will cook and the chef will watch and guide them. We take the same approach and the goal is to get the customer involved as much as possible.

Self-driving cars are going to help decongest roadways, reduce accidents and automobile-related fatalities, and make it possible for people to only pay for driving services as they use them. Self-Driving IT is doing the same thing for IT and enterprises today. Self-Driving IT is helping to automate the drudgery and tediously repetitive events. Self-Driving IT is all about streamlining the technological overhead, automating the approval processes, and automating as much as possible when it comes to internal security and regulatory and policy-compliance testing. But most important, it means when someone needs a new application or service, they can summon it just like a taxi. For more information about how your organization can benefit from Self Driving IT, and how Flux7 can help you get there as quickly as possible, contact (844)-358-9700 or sales@flux7.com today.

Contact us today for a needs assessment: <http://bit.ly/1GHDZvf>

www.flux7.com | Info@flux7.com | 844.358.9700



www.flux7.com/cloud-infrastructure-assessment/

