# It's Time to Refocus IT on Applications

**Nick Howell, DatacenterDude Services**
**James Green, ActualTech Media**

## CONTENTS



## INTRODUCTION

Information technology certainly isn't what it used to be, is it? Sometimes we long for the days when our biggest problems were a user's inability to log in or their printers not working. Do you remember the first time you pulled a cable? Crimped a connector? Successfully authenticated to Active Directory? Delivered an e-mail to an inbox?

How about creating virtual machines (VMs)? That was awesome! Remember your first "P2V" and how cool it was to see more than one OS running on a single physical server? And how cool was it to see all your VMs sharing the same pool of storage!? Across the same "virtual" network!?

Each of these questions represent an important mark in the last 20 years of IT history and the evolution of enterprise IT technology. These milestones seem to happen about every 10 years, and we have officially entered another one of these evolutionary cycles.

# DevOps and the Era of the Application

If you think about it, it's always been about the application. Everything else was just a necessity – a means to an end – to run the application. Setting up, running, and maintaining the entire underlying infrastructure consumed 90% of our staff, resources, and capital expenditure each year... all to run a suite of applications.

This was due to the complexity involved in infrastructure that led us to the massive consolidation that happened during the era of virtualization from roughly 2007 to 2013, and is still going to this day. There are still companies struggling to reach 100% virtualization, and the irony of this is that the usual limitation preventing a server from being virtualized is the application itself!

Even though we've jumped this hurdle (for the most part) across the industry, it has hindered a lot of companies from moving forward with virtualization projects, especially for proprietary and custom applications that could not be updated to support modern infrastructure methodologies.

### 100% Virtualization Is Not Always Necessary!

Although many companies implement a "virtualize first" strategy and most of their applications benefit most from virtualization, there are certainly reasons and workloads that make more sense to run on bare metal. Don't virtualize simply for virtualization's sake!

Even then, the application was the master. This delay has put many companies behind the eight ball over the years, forcing them to play catch-up. As the infrastructure evolved, so too did the various virtualization technologies and the underlying third-party virtualization ecosystem of storage systems, management suites, and business continuity and disaster recovery software.

The unfortunate side effect of this lag is that you end up running two or more sets of infrastructure in parallel. With disparate architecture, software, and licensing, running multiple sets of infrastructure to accomplish one set of goals introduces excessive operational complexity and expense. The number of things you have to keep track of compounds exponentially; so even though virtualization moved things forward from a technology perspective, it greatly increased the complexity of running an IT department efficiently. As virtualized infrastructures scale up, we ultimately end up with a problem just as big as the one that led to virtualization in the first place.

It's funny how all of this comes full circle, as technology begets more technology. As fast as things ramp up these days, it can feel like a flooding waterfall from a broken dam of new software, tools, and hardware that seems impossible to stop. There is always a bigger, better, stronger, and faster tool, widget, or gizmo that "can fix problem XYZ for you!" But every one of those comes with its own set of requirements, demands, and dependencies to take advantage of the latest and greatest. It's very easy to fall into the trap of the never-ending upgrade cycle. Just as you finish one set of upgrades, it's time to upgrade and migrate something else. And on, and on... and on.

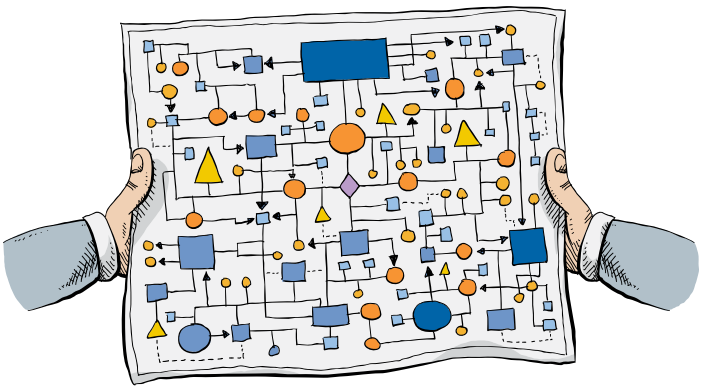

**Figure 1-1:** If you could see many of today's data centers represented on paper, they'd look something like this!

We've also begun to think about new ways of handling application development. A new way to get dependencies of infrastructure and red tape out of the way, to let coders just code. It isn't some magical box or software you can buy, but more a mantra of methodology leading to a faster, more efficient way for developers to build and test software without having to involve change management cycles and infrastructure team approvals.

In 2008, at the Agile Toronto conference, Andrew Shafer and Patrick Debois gave it a name: DevOps.

### What Is DevOps?

DevOps, according to Wikipedia, is a set of practices intended to reduce the time between committing a change to a system, and that change being placed into production.

It would be many years before DevOps became a mainstream buzzword used across the industry. IT teams and vendors were still very focused on virtualization, and another new technology and way of doing things outside of corporate IT was top-of-mind for everyone in the industry.

### IT Priorities Survey Feedback

In a broad survey recently conducted by ActualTech Media, respondents were asked to rank their priorities for the next 12 to 18 months. The No. 1 priority was to improve operational efficiency. This shows that companies are becoming increasingly burdened by complexity, and that they're motivated to eliminate it.

## EMERGENCE OF THE CLOUD

And then came the mystical "cloud."

In reality, it wasn't so mystical; it actually made a ton of sense. With the advent of a new generation of web development and mobile applications came a new way to host and run those modern applications. The cloud won the hearts of IT professionals by leveraging seemingly infinite amounts of on-demand resources, as well as an OPEX-based, pay-as-you-go and payonly-for-what-you-consume model. The resources the cloud brought to our fingertips was restricted only by the limit on our credit cards.

Today, we see many versions of this from major players such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, and others. The funny thing is that with the rise of modern mobile and web-based apps, business leaders took note; and the first thing they wanted to do was run all the corporate IT applications in the cloud.

Very quickly it was discovered that, while not impossible, this was certainly not a trivial or inexpensive venture. Common heavyweight enterprise apps like Microsoft Exchange, SQL Server, and Oracle Database cannot just be dropped into an EC2 instance in AWS and run the same way they've traditionally been operated on-premises in your data center.

Some refactoring has certainly occurred over the years to enable these technologies, like Office 365 and SQL Server 2017, to be run as subscription services in the cloud.. However, the concept of forklifting your mission-critical servers and applications into the cloud has been a long-running, yet nearly impossible, obsession of CIOs and IT leaders.

Another irony of modern infrastructure is that we've spent the last decade consolidating everything into single platforms to reduce footprint, and we're now trying to break everything up again at the application level. The re-introduction of Linux cgroups as "containers," made popular by Docker, has opened the door to isolation of multiple applications on a single OS, breaking their package of services into individual microservices that run independently of one another, and leveraging a shared pool of resources.

### Containers Alone Don't Break Down Silos

There's still some minor confusion surrounding who is responsible for what in a container-based application architecture. In this sort of environment, there is underlying infrastructure, similar to that of the virtualization environment, that is run by the infrastructure or VM admins. From the bottom of the stack all the way up to your repositories, someone still has to manage all of that! On the other side, the application owners simply log in to whatever repository management system you're using, such as Docker Hub, and generate pulls or instantiate containers. Very rarely do these roles cross over (perhaps in dev-test environments), but there is still typically a traditional separation of duties in place between development and infrastructure teams.

This is, at a high level, similar to the concepts we used to justify virtualization to pool resources, but the end result is quite different. We've moved up the stack even further and are now isolating components, as opposed to consolidating them, in an effort to remove dependency and overhead.

Leveraging containers in this way makes a lot of sense. It's one of those scary "why didn't we think of this sooner" ideas that will ultimately change the entire landscape of modern IT again, just like virtualization has done and continues to do. And just as companies are finally getting their heads around their shiny new virtual infrastructures, and as admins and engineers are seeing the dust settle from 10 years of P2Vs, endless new VM requests, and entirely new suites of management software, here comes the IT Cycle to kick off the next round of full-circle evolution all over again, making things more complex than ever before.

## Application and Infrastructure Complexity Deepens Silos

Due to this ever-growing complexity, many operators have begun to focus on infrastructure as the main issue, without a proper level of focus on the workloads the infrastructure is supporting. Tooling and workflows have been built to support infrastructure operations. When an application struggles to run properly, there's quickly a finger-pointing blame-game that happens toward IT and the underlying infrastructure as the root of the problem.

This conflict can often be attributed to the continued siloing of teams within an organization. One of the main goals of the virtualization movement was consolidation. Not only of infrastructure resources, but also of skillsets, teams, and staff. In a twist of fate, it had mostly the opposite effect, phasing out veteran IT staff who were used to performing one job or having a single area of focus, and leading to the rise of the "generalist" or "data center engineer," charged with running server/OS/storage/virtualization stacks which were traditionally run by 2-3 different teams.

But this added new types of admins with different types of skillsets. You needed network engineers with virtual networking understanding, storage admins that understood the demands of virtualization, and server admins who could implement virtualization while still understanding the prerequisites and unique demands of storage and networking.

This changing landscape only made the finger-pointing issues worse, and an industry-wide solution designed to consolidate and make teams more efficient led to expensive retraining of staff, and re-tooling of the entire datacenter.

### What Does "Full Stack" Mean?

A term you'll often hear around IT organizations these days is full stack. People talk about full stack visibility, full stack thinking, or about being a "full stack engineer."

This term is somewhat confusing because it can mean two different things, depending on the audience.

When a software developer uses the term full stack, they're referring to all the various parts of an application from the front-end user interface (like a website) through to the back-end database, and all the software and exchanges of information in between.

However, when an infrastructure architect uses the term full stack, they're referring to all components from the physical data center up to the front end of the application. This includes everything that a full stack developer is looking at plus the operating system, hypervisor, server, network, storage, and more.

Either way, full stack thinking is really an IT-specific way of saying to step back and look at the big picture. It's easy – especially for technical folks – for IT thinking to get very narrow in scope. The modern push toward always thinking about the big picture benefits all parties.

This re-tooling had some unfortunate side effects. It has led IT teams to focus efforts on troubleshooting, endless cascading upgrades, and capital expenditure on expansion of hardware and support as a result of application demand. The inherent flaw in this approach is that the infrastructure team's view of the application is always an afterthought. Its capabilities and functionality are limited to the available resources of the infrastructure, regardless of what they might be, or how many upgrades are performed.

IT needs to begin building and managing with the full stack in mind. There is certainly an element of infrastructure management that needs to remain, but the focus needs to shift to that of full stack implementations from the top down, and with consideration for the application first. When designing modern infrastructure with a full stack perspective, the application needs to be properly specified first; then the demands of the application can drive the infrastructure and resource requirements underneath.

The magnitude and velocity of growth in technology is simply out-pacing IT departments' abilities to keep up. Therefore, although this is going to be a tough transition for the more grizzled IT veterans, it is the only way forward for many organizations. This tough transition is only made worse by the lack of any real centralized single pane of glass from which to manage this new-age infrastructure from a full-stack perspective.

## IT with Applications at the Center

An IT organization with laser focus on the applications they're delivering has some key characteristics:

- An unyielding focus on the end-user experience
- A constant quest for better application visibility for the infrastructure team
- A culture that reinforces full stack thinking

## FOCUS ON END-USER EXPERIENCE

Getting a phone call from end users that something isn't working can be one of the most time-consuming problems for IT departments and their helpdesks. We all know applications can run slow at times, and this may be attributed to any number of possible issues, such as:

- **Problems within the data center**
    - The application itself
    - Lack of compute/memory resources
    - Storage bottlenecks
- **Problems with the End User**
    - Network Connectivity
    - Issues with user's endpoint device

The end user is most likely uninterested in tech jargon justifying why they're not having a pleasurable and productive experience. They just want to be able to do their job without interruption by malfunctioning technology.

In most cases, the real purpose of the IT department (in businesses where IT isn't the primary offering) is to empower the rest of the business to do its job. Internal consumers will turn on you very quickly and throw IT under the bus if they are unable to perform their duties. Thus, this user experience is critical to the delivery of any application and should be used as a prerequisite measuring stick to determine the necessary resources, infrastructure, and scale required for success.

There is a reason many consumer applications (even video games) do extensive alpha and beta testing with their user base. Ensuring that most of the kinks are worked out before the offering is made generally available ensures that you don't blow it on your one chance to make a good first impression.

Beyond user testing cycles, it's equally important to constantly monitor the real-time user experience. During times of high demand, the IT staff must be ready to expand capacity, bandwidth, and any other infrastructure constraints in order to accommodate end users accessing their applications.

### Proper Prior Performance Testing Prevents Potential Problems in Production

It's a good idea to learn to incorporate these types of methodologies and practices into your development and rollout cycles of new applications and always involve users in the process. Proper tests are one of the best indicators of your application's performance, and test results can be one of the first signs of trouble when something either doesn't make sense or isn't running as expected.

## APPLICATION VISIBILITY

Troubleshooting any issue with an application can often feel like throwing a bunch of balls of masking tape at the wall and waiting for one to stick. The time and money we've directed at modifying, upgrading, and monitoring our infrastructure over the last decade has left the applications behind, leading to a lack of visibility and understanding of what is actually going on under the hood of an application that isn't running properly. It is paramount that your infrastructure teams have a deep understanding of the inner-workings and requirements of all applications being run in the data center. Just like prioritizing real-time monitoring of infrastructure resources and assets, it is necessary to provide your infrastructure team with tools aimed at application visibility as well!

We can monitor logs all day long and hope we run across something (see: masking tape ball analogy), but the reality is that this is an attempt to use a reactive approach, where a proactive one is required.

Constant real-time monitoring of an application and its interconnected infrastructure is desirable, but it's hard to measure and optimize what we don't understand. Moreover, without this understanding and insight, troubleshooting can feel impossible. That being said, when deploying modern applications, it is imperative that a built-in understanding of application workflows and visibility of the full stack be implemented at the time of launch.

### FULL DATA CENTER SCOPE

Every single data center component has a part to play in the success of an application, and it is important to include the entire stack in day-to-day operations and monitoring.

In more traditional IT departments, this scope exists but is broken up across separate teams managing servers, storage, and networking independently. This simply cannot exist as isolated bubbles anymore, especially in IT departments that choose to place their applications at the center of their focus. An application-centric IT organization requires that these usually separated teams be in complete lockstep with the objectives of the application, aware of application performance, and have an overall meta-view of the broad inner-workings of the data center.

## Cloud-Induced Application-Centricity Challenges

When companies began entertaining the idea of running their applications in the cloud, the immediate focus was on trying to forklift entire application suites. In reality, it was going to require a step back to re-evaluate the deployment of the entire application stack, leveraging resources from a cloud provider – similar to how those one-tap-launch mobile applications were being run on smartphones. It could be argued that the evolution of consumer tech was the biggest thing that spurred the use of the cloud in commercial IT situations.

Many departments began experimenting with moving applications to the cloud, only to see a disconnect between those applications and the on-premises infrastructure. None of their management and monitoring tools worked anymore, so new tools were required. In fact, some of the tools that early adopters needed had not even been created yet, leaving CIOs shrugging their shoulders wondering what to do next.

As this demand from smartphone-trained users came more into focus and enterprises began to get their head around this new, cloudy way of life, more and more workloads found their way into the public cloud providers, and IT as an industry started to add some new words to our vocabulary. But as you'll learn, with these new words came some new challenges.

### Public Cloud

One of the best examples of public cloud use is the mobile apps that run on our smartphones. These apps operate completely independent of any on-premises infrastructure in most cases, are installed and updated directly from the cloud, and the only impact on the local device is when they are run. Data, state, and logs are all stored in the cloud, providing a central place of collection for management and troubleshooting. The cloud-centric nature empowers development teams to quickly test and deploy updates and maintenance releases more frequently, addressing bugs and rolling out improvements at an accelerated rate compared to a more traditional on-premises approach.

For an example that hits closer to home, let's use Microsoft's Office 365. Where IT administrators used to have to install and maintain a monolithic central infrastructure of Microsoft Exchange for e-mail, companies and users can now simply subscribe to Microsoft's service and get the exact same experience they had before, including all Office applications like Word, Excel, PowerPoint, and begin using Outlook for e-mail with one swipe of a credit card. Moving from on-premises Microsoft applications to Office 365 is, ideally, a transparent transition for the end user. Remember how critical a focus on end user experience is?

It took Microsoft a few years to nail this service down, but once they did, it really took off. It makes so much sense to operate things this way: offloading infrastructure responsibilities completely from IT departments and merely requiring them to become account managers for user access.

### Cloud Economics

Provisioning Microsoft Office services this way erases all capital expenditure on maintaining an Exchange infrastructure, and converts that CAPEX (capital expenditure) into OPEX (operating expenses), which is easier to stomach for many organizations. Considering that almost every user in an organization uses Office and e-mail, this change alone can have a huge impact on budgets, allowing departments to pivot precious capital to other areas of focus. Applying this principle to various aspects of IT gets to the heart of what the public cloud is all about.

## Private Cloud

For very large enterprises that do not want to place sensitive data in the cloud, it became a common practice to build their own cloud-like services, empowering end users with self-service access to request resources, and have automation take over for deployment tasks.

While these types of deployments are usually reserved for only the largest of companies with large IT teams, it became more common with various platforms that came with pre-designed templates that IT departments could build upon to design their own self-service infrastructure offering. While the advantages of a private cloud are obvious, this type of infrastructure comes with downsides, as well. For one, private clouds are almost universally complex. This complexity is frequently a distraction and causes IT organizations to lose focus on the applications.

## Hybrid Cloud

Hybrid cloud adoption is quickly becoming commonplace. In a survey conducted by ActualTech Media in 2016, 34% of respondents were currently evaluating their hybrid cloud options, and 10% had already implemented something. It's quite possible that, by this point, the percentage of companies either adopting or about to adopt a hybrid cloud posture has exceeded 50%.

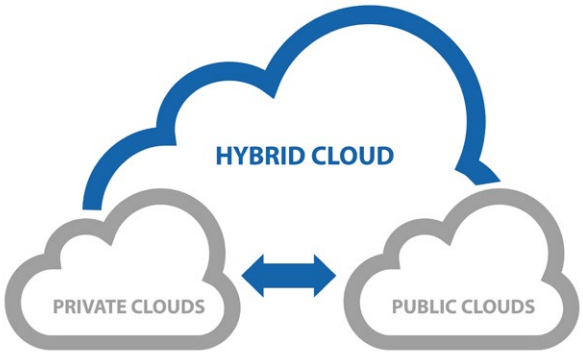


**Figure 1-2:** Because the Hybrid Cloud model seeks to extract the benefits of both public and private clouds, it's often the most beneficial, but also the most complex and tricky to manage.

This is exactly what you think it is: a combination of the use of both public and private cloud resources and methodologies. What, when, where, and how much are the common questions. The dilemmas are different and usually weigh each other out based on cost and overall management or privacy requirements.

When you're running applications in one place, whether it's the public cloud or your own private data center, it is exponentially less complicated than running that same application in two or more different locations at the same time. You could have users logging in from multiple places in the world, hitting different data centers. But things need to be kept in sync as much as possible, and centralized monitoring and management of all available resources and user sessions becomes even more important.

Hybrid cloud presents some of the most unique demands on IT staff of any current technology. Private and public infrastructure must be monitored for availability, and applications running across all locations must be managed for connectivity and user experience, as well as the load they're placing on the infrastructure or cloud instance they live upon.

When it comes to keeping track of applications as they exist across multiple infrastructures, it can get very tricky. Things like application dependency mapping and resource planning get extremely tricky in a hybrid cloud world.

## APPLYING APPLICATION-CENTRIC IT TO THE ENTERPRISE

When we think of applications running in the cloud, we tend to think of several common key concepts:

- CAPEX vs. OPEX
- Agility
- Performance
- Granularity of microservices

As application-centric infrastructure management becomes more and more prevalent, we can begin to draw some corollary between how things are done with apps in the cloud and how they should be done with apps anywhere in the enterprise. When deploying an app in the cloud, you're required to define the resources your app is going to need. You'll be careful to not overprovision, because that quickly gets expensive. Thankfully, you can often design cloud resources to scale up automatically as the demand on the application grows.

The cloud provides a nice roadmap of directives as we chart a course to build our new applications in enterprise IT departments with the full stack in mind.

# About Uila

Uila (https://www.uila.com) with its Application-Centric Infrastructure Monitoring and Analytics identifies performance bottlenecks for business-critical services & plans Workload Migration strategies for Private & Hybrid Cloud environments. Uila provides service dependency mapping, full stack correlation with 1-click root cause analysis and patented deep packet inspection technology that understands over 3,000 application protocols for transactional meta data analysis. Businesses use Uila to align themselves with their IT Operations team and cut time to resolution from days to minutes, keep their application at peak performance at all time and ensure end-user satisfaction to the fullest.

Uila also offers a comprehensive Application & Infrastructure Assessment program, that is designed to provide an assessment of the health of the entire stack in your Data Center environment. This includes both the Application & the Infrastructure (Compute, Network & Storage). Click below to learn more.