The Gorilla Guide to...®

Application-Centric IT

For Private and Hybrid Cloud

Inside this Guide:

- Learn about the advantages of an application-focused approach to IT
- Discover application dependencies to simplify workload migration and resource planning
- Start the journey of developing a "full stack" mindset for managing applications

Nick Howell
James Green

Helping You Navigate
The Technology Jungle!

ActualTech Media
www.actualtechmedia.com

In Partnership With uila

The Gorilla Guide To…

Application-Centric IT

For Private and Hybrid Cloud

Authors
Nick Howell, DatacenterDude Services
James Green, ActualTech Media

Editor
Hilary Kirchner, Dream Write Creative, LLC

Layout and Design
Scott D. Lowe, ActualTech Media

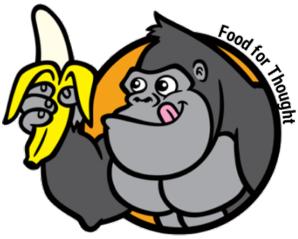
The information provided within this eBook is for general informational purposes only. While we try to keep the information up-to-date and correct, there are no representations or warranties, express or implied, about the completeness, accuracy, reliability, suitability or availability with respect to the information, products, services, or related graphics contained in this book for any purpose. Any use of this information is at your own risk.

ActualTech Media
Okatie Village Ste 103-157
Bluffton, SC 29909
www.actualtechmedia.com

Entering the Jungle

Callouts Used in This Book

The Gorilla is the professorial sort that enjoys helping people learn. In the Schoolhouse callout, you'll gain insight into topics that may be outside the main subject but are still important.



This is a special place where readers can learn a bit more about ancillary topics presented in the book.

When we have a great thought, we express them through a series of grunts in the Bright Idea section.

Takes readers into the deep, dark depths of a particular topic.

Icons Used in This Book

Definition. Defines a word, phrase, or concept.

Knowledge Check. Tests your knowledge of what you've read.

Pay attention. We want to make sure you see this!

GPS. We'll help you navigate your knowledge to the right place.

Watch out! Make sure you read this so you don't make a critical error!

Chapter 1

Refocusing IT on Applications

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 & 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 so many companies behind the 8-ball for years trying 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 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 their 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.

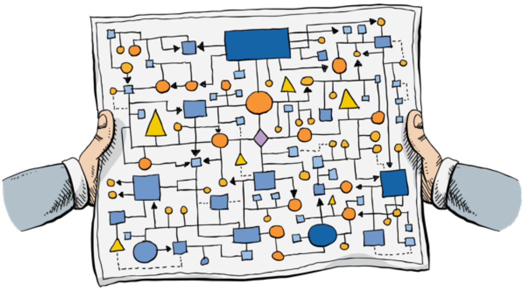


Figure 1-1: If you could see many of today's data centers represented on paper, they'd look something like this!

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 #1 ranked 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 also 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 pay only for what you consume model. The resources cloud brought to our fingertips was limited 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 to be run as subscription services in the cloud. Examples include Office 365 and SQL Server 2017. 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 AppDev 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 so much sense, doesn't it? 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 even 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 seeing 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 that could implement virtualization while still understanding the prerequisites and unique demands of storage and networking. This changing landscape only made the finger-pointing issues even worse than before, 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 their 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.

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.

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.

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 really 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 for these companies 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, as they are wont to do.

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%.

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 datacenters. 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.

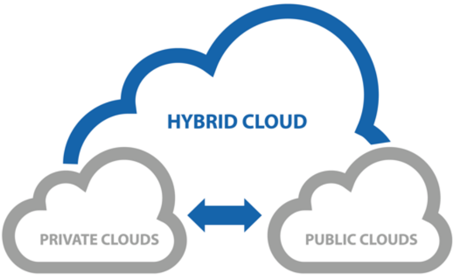


Figure 1-2. Because the Hybrid Cloud model seeks to extract the benefits of both public and private clouds, it's often the most beneficial, but also the most complex and tricky to manage.

When it comes to keeping track of applications as they exist across multiple infrastructures, it can get very tricky. As we'll discuss later, 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. Of course, 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 us with 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.

Knowledge Check

Answer the following questions to be sure you grasped everything in Chapter 1:

- What are some of the challenges that virtualization introduced?
- Define public, private, and hybrid cloud.
- How can leveraging cloud-like practices change an IT organization?

Chapter 2

Simplified Workload Migration

In the previous chapter, we went over the challenges associated with operating an application-centric infrastructure across all variants from private to public to hybrid clouds, and specifically how that affected IT teams. In this chapter, you'll learn about the challenges with moving workloads throughout and between these various infrastructure types.

Ask any admin who has managed a data center or IT department over the last 30 years what their biggest obstacle has been, and 9 out of 10 of them will tell you that it has something to do with moving data and applications around seamlessly. If you look at some of the most successful entry points IT vendors use to sell new products to businesses, many of the top features involve something to do with simplifying workload mobility and the agility to migrate quickly and without downtime.

Remember scheduled downtime? Those were the days. Sadly, gone are the days of being able to schedule 8- to 12-hour outage and maintenance windows. IT departments today have a pretty standard requirement of never going down, never deleting data, and seamless failover in the event of a problem. Their primary goal is to keep users online and conducting business. The ideal scenario involves the end user never experiencing delays or outages in any way, shape, or form.

In order to provide this level of availability, IT professionals may utilize infrastructure that spans multiple racks and systems, or even more broadly, multiple geographically separated locations. They may also use infrastructure from multiple cloud providers.

Data & Application Availability

Redundancy comes in multiple flavors, the first of which has to do with availability. In order to prevent interruption to the user's ability to access key systems and workloads during a failure scenario, a redundant set of infrastructure can be in place. This infrastructure can reside either on-site or in a different location, and it exists to allow users to continue to work despite an outage. This requires data and applications to be kept synchronized on as tight of a schedule as possible, usually limited only by the network capacity to do so. Network capacity demands increase as the amount of data being synchronized increases.

Load Balancing a Workload

Many business-critical applications require multiple instantiations of the same workload kept in lockstep in an active-active configuration in order to balance load at peak times. This, again, is usually limited by that same glass ceiling of network capacity. The key difference between load balancing for performance and synchronizing for availability is that the network performance required to run an active-active configuration is significant, and it's not feasible to loosen the constraints. Network requirements for availability-focused synchronization can be relaxed by adjusting the service level agreements; a higher RPO (recovery point objective) will require less network capacity.

In both of these traditional scenarios of workload migrations—redundancy and load balancing—massive amounts of data are constantly flying across the wire, and huge up-front CAPEX outlay is necessary in order to provide the underlying infrastructure to support either scenario (moving workloads for availability or for performance).

Availability Service Level Agreements

IT organizations often make commitments to the rest of the business to deliver certain levels of availability for applications and data. When it comes to recovering from a failure, two key metrics often define the commitment (or service level agreement) to the business:

- **Recovery Point Objective.** This metric is a measure of minutes, hours, or days *backward* in time from the failure. It tells the business how much data will be lost in the event of a failure. Will we be able to recover everything that happened up until 15 minutes prior to the failure? Four hours prior? Will we be able to recover to the data we had at close of business yesterday but lose everything that happened so far today? The gold standard for many organizations is 15 minutes, but it's not cheap to keep this promise for your users.

- **Recovery Time Objective.** This metric is a measure of minutes, hours, or days *forward* in time from the failure. It tells the business how long until the recovery will be completed in the event of failure. From the time that incident response begins, will the users be back online in 15 minutes? Will they need to leave for lunch and come back when the problem is resolved? Do we need to send everyone home for the day? Again, the gold standard is 15 minutes, but this requires serious organizational organization and advanced technology.

As more and more cloud services began to come online, businesses started to see opportunities to "rent" this infrastructure in an on-demand fashion, removing the need to spend millions on infrastructure they might only need in a contingency, and that they may realistically never even use.

As architectures matured, so did the migration methods of these workloads. Let's take a moment to go over some of the more popular use-cases, the problems they solve, and some of the new challenges they present.

On-Premises to Public Cloud Migrations

There are many reasons a company may wish to have a presence in one or more clouds outside of their on-premises data center. One of the most popular reasons is the long-term storage of "cold data," either in the form of offline backups, or simply data that has not been accessed in a specified period of time. Popular object storage systems and applications have built in algorithms to determine the age of data and they have capabilities to manage the migration of cold data to a particular cloud service.

What is Object Storage?

Object storage is unfamiliar to many enterprise IT admins who are accustomed to dealing with file- and block-based storage. Object-based storage stores data as unique objects (with a globally unique identifier) and generally attaches some metadata to each object. Dealing with large amounts of unstructured data is a common reason for implementing object-based storage systems.

In addition, more and more companies are leveraging cloud storage solutions for offline and off-site long-term backup. As the cost-per-gigabyte of storage has cratered, the sheer volume of storage available at an extremely discounted rate has driven IT leaders to consider cloud storage as a viable alternative, and companies like Iron Mountain have created entirely new lines of business around cloud offerings to replace more traditional tape vault archives. This has also forced the backup orchestration software vendors to build in functionality that includes backing up data to cloud-based destinations. Amazon S3 and Glacier, as well as Microsoft Azure Blob storage (Hot and Archive), are examples of some of these offerings.

At the end of the day, however, someone or something still needs to keep track of the location of these backups and cold data in order to facilitate future access should the need arise from a failure, audit, or user error.

Moving Entire Applications to the Cloud

While cold data storage may be a no-brainer, many companies are interested in tackling even bigger challenges. Often, IT leaders turn an eye towards the cloud when they're looking to solve elasticity and scalability issues. One of the key differentiators of the cloud as compared to traditional infrastructure is that it's inherently flexible. When it comes to applications such as eCommerce and big data analytics, the ability to scale up and down as demand changes can be a powerful capability and architects stand to improve performance and reduce costs.

Although the gains to be made are very real, moving a legacy application to the cloud isn't generally as easy as just picking it up and moving it. Due to the completely new architecture that cloud services are predicated on, a rewrite of the application is often needed, and at a bare minimum, configuration changes will be required to ensure the application transitions smoothly. Of course, all this is very difficult if not impossible without a proper understanding of how the application works and its dependencies.

Data Center to Data Center Migrations

Moving data between data centers is a long-standing tradition in IT, whether that is replication for passive backups, secondary workloads, or geo-located users. However, as data became entire workloads inclusive of VMs and multiple volumes, these migrations became more and more complex. Moving entire workloads between data centers today requires not only the movement of data, but the servers and applications that consume and leverage that data as well. This requires an unprecedented amount of monitoring, if only to keep track of where everything is at any given time.

This has a compounding effect over time from a cost perspective, and is one of the principal reasons companies began looking at third-party infrastructure (clouds) years ago, as it was much more cost-effective to use a service provider than to invest the capital expense required to stand up a duplicate data center.

Off-Site Migration for Disaster Recovery

Prepping for disaster recovery, or what is commonly referred to as a *smoking-hole scenario*, has always been and likely always will be one of the most common and pervasive use-cases for moving workloads off-site. As data became workloads and failover orchestration became more automated, this encouraged companies to leverage off-site tools more and more.

To be clear, this is different than storing cold data and backups as discussed previously. Rather, this is the ability to completely stand up your entire workload in an off-site scenario in the event of a catastrophic natural disaster or other situation where the on-site infrastructure is no longer available. This involves an incredible amount of planning, and often involves complex detailed runbooks with step-by-step procedures to follow in the event of a disaster.

Better to Be Prepared!

For many fortunate companies, a catastrophic disaster is something they'll never have to face. However, more common types of failures like a hardware malfuction or a power outage can still bring down entire data centers. Even if your data center is relatively safe and secure, you still need to be doing proper disaster recovery planning!

The kinds of goals you should shoot for to consider a migration successful are:

- Cost-effective and Fast to Deploy
- Application Agnostic
- Rollback Capability
- Zero Downtime

When performing a migration, it's also good to take stock of your technical debt, eliminating leftover and unnecessary resources, VMs, and instances. Think of it as a good housecleaning practice to free up infrastructure or save money on next month's cloud bill!

Workload Migration Challenges

Migrating workloads, especially complex, multi-tiered applications, can be a harrowing process. Undertaking the migration without a solid understanding of the interdependencies of applications and the exact resource requirements means almost certain failure. In today's world, it's understandable that failures can, do, and will happen, but what's not acceptable today is downtime. Cutovers of migrations need to be instant, or an hour at most.

A failure of this migration endeavor can often be attributed to a bottom-up infrastructure focus - simply trying to mirror hardware that is available on-site. The downfall of most workload migrations is lack of visibility into the configurations of the actual application(s), and how it consumes resources from that infrastructure. With a proper understanding of how applications interact, it becomes much clearer what the requirements at the secondary site really are.

Mapping Application Dependencies

Understanding and documenting the relationship between all components of an application in extensive detail is vital to the success of a migration, as well as to managing and maintaining said application. Most organizations lack this level of insight, as documentation is almost instantly out of date the minute it is written, and deep understanding of the application often exists only as tribal knowledge in the heads of the administrators maintaining the application. What happens to a critical application if there's a problem and the administrators with that tribal knowledge are unavailable? Perhaps those individuals are busy, on vacation, hospitalized, retired; there's an infinite number of reasons that tribal knowledge goes offline. Manually creating the workflows and processes necessary to avoid downtime or repair failures is still not enough in today's IT world. These types of recoveries can be managed and monitored for with automation and workflows can be triggered automatically in the event of a failure, eliminating this vulnerability.

No one person should hold the keys to your digital kingdom, but let's also not assume that there is even a single person that has the detailed knowledge necessary to migrate or restore application services. In a modern application-centric IT organization, you can use a tool to provide this insight into the interdependencies of applications for you.

Understanding Resource Requirements

When breaking down an application, you need to consider all of the services that application provides and connects with in order to properly recreate the resources required to run it. Oftentimes, an application *can* be run in a degraded state temporarily while a migration is in process, or during a failover scenario. Said another way, you don't always need the exact tier one hardware to stand up another copy of the application, but you definitely need in-sync data and access availability. Users are forgiving of a lack of speed more so than they are a complete lack of access.

Knowledge Check

So how do we go about measuring the sheer breadth of an application?

- Services running within the app
- External providers of data to the app
- External consumers of data from the app
- Servers and OS versions required to run the app
- Storage systems required to host the app data

All of these components are baseline requirements when it comes to understanding the needs of your application and determining its portability.

Migrate Successfully with Application-Centric Workload Visibility

For too long, we have placed focus on the underlying infrastructure of a data center and looked up to see which applications would be affected in a migration scenario. Having a complete understanding of the workload and all application dependencies prevents you from doing a lot of patch work and cleanup after the migration is complete.

When you use the application itself to define the underlying resources it is consuming, it is much easier to track supporting infrastructure and resources that need to migrate or travel with the application when it moves.

This top-down approach leads to much more successful migrations, and to only replicating and moving the data and supporting systems required to support the app or apps being migrated.

It is critical to the success of a migration that you understand the resource requirements of any and all applications that you predetermine to need to have the ability to move between data centers. This will empower IT departments to be prepared for just about any given situation or scenario they run into.

Working backwards can lead to the best results. Start at the top by outlining end goals, and backtrack down the stack to define any infrastructure, software, licensing, and other requirements necessary to allow you to migrate your workloads around seamlessly, whether that's between data centers or clouds, or a healthy mix of both.

These types of best practices can lead to a much smoother experience when fully migrating an application or workload to the public cloud, or doing iterative migrations between servers across myriad public and private cloud instances.

Chapter 3

Application-Centric Workload and Resource Planning

In the previous chapter, we outlined many best practices you could take advantage of in order to increase the chances of migration success, regardless of where you are moving your workload, whether it's a planned or unplanned migration. But if we take a big step back and look at the bigger picture, even aside from migrations, we all could do with a little more planning and reassessment of our needs on a regularly scheduled basis, as things are in a constant state of flux.

Modern IT budgets aren't as generous as they have been in the past, and IT professionals are commonly asked to do more with less. In such a time as this, it's more important than ever to be able to forecast and plan appropriately for the purposes of budgeting and setting expectations.

Transparency amongst teams, departments, and leaders is something that many companies could continually improve on. IT teams are almost always oversubscribed, have a never-ending task list, and are constantly being asked to move mountains. While the things they're asked to do are not impossible, they can be very daunting when not properly planned for or when the necessary resources are not available due to budget constraints.

You'll never save your job in a failure scenario by blaming the company for not giving you enough money to do something required to facilitate successful business continuity. This makes planning and resource evaluation one of the (if not *the*) most important considerations to help you avoid landing yourself in hot water when the inevitable creeps up on you!

Workload Planning Challenges

The most common phrase in IT is "it depends…" and no greater truth has ever been spoken. Many questions asked of IT professionals often require an IF/THEN/ELSE conversation to break down and determine requirements in order to properly understand business objectives and how high-level projects will impact infrastructure demands.

Let's go over a few things that have a tendency to influence these decision processes.

Lack of East-West Traffic Insight

If "it depends" is the most common phrase in IT, then "it's the network's fault" comes in a close second! We fear what we don't understand, and for infrastructure people, the network can be this maelstrom of wires, config files, protocols, and blinking lights that prove difficult to grasp. It takes a special mindset and certainly a specialized skillset to truly understand and identify networking architectures, the interdependencies of applications and underlying workloads, and how they all connect and speak to one another.

While many companies will invest in having multiple helpdesk employees and systems engineers, they often skimp on having one or more proper network engineers. In reality, this can be one of the most pivotal and foundational roles within any IT organization.

Properly connecting systems and the applications they host to their partner servers and apps, databases, and storage systems is the literal backbone of any data center and infrastructure design. Monitoring and shaping traffic, packet inspection, and documenting all of this is also one of the most vital parts of any migration scenario. Most organizations don't have a solid understanding of the networking interdependencies of a given application because they don't inspect network traffic at that deep of a level, and documentation is likely out of date.

What is East-West Traffic?

With regard to enterprise networking, practitioners sometimes use the directional terms north, south, east, and west to refer to which parts of the infrastructure are communicating. Northbound and southbound traffic is that which is going into and out of the data center, both logically and physically speaking. This is frequently end user traffic to communication with an external service. East and west traffic is a way of describing communications *inside* the data center and between systems.

Most enterprises have a network team and believe they have enough networking insight in the data center. Unfortunately, most of them would also agree that they have little-to-no east-west traffic insight. This would include more discrete networking components such as virtual switches, leaf switches, and spine switches.

Full Stack View? Nowhere in Sight.

In the past, it was (and possibly still is) common practice to simply duplicate the infrastructure and just re-install the apps to facilitate a restore should the need arise. This is incredibly shortsighted and gets a lot of companies in trouble, as they're not taking the full stack of the application and its dependencies into account.

It's impossible to plan without a big picture view of the infrastructure, which includes the applications, the hardware, and everything in between. Not to mention the fact that each of the pieces between the power plug and the application has their own methods and software and licenses required to run, back up, and restore.

Infrastructure resource planning is meaningless without considering applications and what their priorities are. Also, quantified resource usage information and identifying performance bottlenecks are necessary in order to properly address resource planning. This also provides you with a benchmark to measure against for the purpose of comparison.

Under/Overprovisioning Is Rampant

For a long time, it was common practice in IT when purchasing new gear to pad what you were buying with +100% of whatever you currently used or anticipated using in order to make the capital expense last longer. Organizations often use a 3-year refresh cycle. As IT professionals implementing these systems, knowing that an additional purchase mid-cycle would be an uphill battle ingrained a natural tendency to overprovision. In an interesting twist, one of the marketing tactics of VMware during the virtualization movement of the late 2000's was to highlight how servers and storage systems were only using 10% of the capacity on average. Storage systems proprietors began preaching to us about thin provisioning, so as to only provision what we needed, as we needed it, with things like auto-grow monitoring enabled.

So, we began to pack everything in again. We pushed ESX hosts to the limit with 60%+ CPU utilization, Memory Ballooning, vMotion/DRS, and other technologies that would keep things online should spikes occur. But even with all of that special vTechnology, we still needed to overprovision our hosts to account for other hosts in the cluster going offline. Even in the midst of a mass consolidation effort, we were still overprovisioning.

The practice of "what if" contingency planning is certainly healthy, but at a certain point just becomes counter-productive to the goals you're trying to accomplish. Right-sizing an environment for both optimal cost and performance is another key challenge with regards to resource planning. You never want to not have enough resources to run servers and applications at full power, but you also don't want to overspend for a “just in case.” Certain monitoring software vendors will present you with an “all green” view of your datacenter even if you’re ridiculously overprovisioned. As far as they’re concerned, you’re in good shape. But the reality is that you’ve grossly overspent in order to protect against a contingency. Had you and your team had proper utilization and monitoring software in order to plan for resources needed, you would be able to right-size your environment without unnecessary up-front capital outlay. Think of it as “Everything you need, and nothing you don’t.”

Simplify Workload Planning with Application-Centric IT

Historically, IT departments have operated in a way that requires underlying infrastructure to be in place before applications have resources to consume. This is often referred to as the bottom-up approach. Even through the last decade of virtualization, this remains a long-standing practice.

As we move into this next phase of the evolution of IT, which is a world of ubiquitous resources, a new approach needs to be adopted where the applications themselves are defining from the top-down which resources they need to consume, and the supporting infrastructure needs to have the agility and flexibility to adjust these resources on the fly. This is one of the biggest motivations for companies to migrate workloads to the cloud, as it provides this foundation for unlimited resources, and an operational payment model that doesn’t require excessive capital investment before the applications can even be turned on.

Knowledge Check

Answer the following questions to ensure you've understood this section on workload and resource planning:

- Why is it challenging to properly estimate resource utilization and future needs?
- How has the traditional IT procurement cycle limited our resource planning abilities?
- What is the solution to over-/underprovisioning?

Chapter 4

Security Analytics and Change Control

Security is the number one concern on many IT executives' minds these days. With the rampant data leaks, breaches, and ransomware attacks we've seen in recent years, it's no wonder that IT leaders walk on eggshells when it comes to changing anything regarding infrastructure, access, or security.

As we travel through the various evolutionary cycles of technology, even as things get simpler on the surface, under the covers they have a tendency to get more and more complex. Infrastructure sprawl and complexity has made the challenge of security and managing changes to the environment more difficult to stay on top of than ever.

Change Control and Security Challenges

Who's Talking to Whom? Who can do what to what? These two questions identify some of the biggest hurdles IT departments face.

Access.

It's not only related to individuals, employees, and users, but identifying applications themselves and their ability to access resources can be a huge challenge for companies. In fact, many breaches come from holes and weaknesses exploited from an admin giving an application a full access path to communicate with other sensitive resources. SQL injections, web server exploits, and buffer overruns all stem from an attacker's ability to trick an application into giving too much access, allowing them to run wild behind your firewall. Once they're inside, the next step is privilege escalation, likely giving themselves administrative access to *everything*.

Applications, to a certain extent, can be more dangerous than individual users. Without being able to actually see how applications are intercommunicating, it's next to impossible to adopt a strong security posture.

Keeping an Eye on Unauthorized Connections

Without insight into who's talking to whom, it can't be determined which network communication is sanctioned and which isn't. If something malicious was going on, would you know? Would you get an alert?

If you did get some sort of alert, what would your reaction be? How would you remove this threat and identify the attack vector?

Most enterprises focus all of their security efforts on the perimeter and user endpoints. It's fair to say that these are the two biggest attack surfaces, but what about inside your network, or "behind the firewall?" The east-west traffic between all of your servers, storage, virtualization platforms, etc., needs constant monitoring as well.

On the surface, it's easy to say, "We were hacked," as a sort of end-all be-all answer to why something happened. But at some point, harder questions are going to be asked, problem areas are going to have to be identified, and solutions will need to be implemented in order to shore up weaknesses in the infrastructure.

The issue at hand is the difference between a reactive response and a proactive one. Most responses are reactive. "Something happened. What do we do about it?" With a proactive approach, preventative measures can be put in place to not only alert admins that something is happening, but advances in monitoring software and intrusion prevention technology have allowed IT professionals to proactively thwart malicious attackers seeking access to your company's resources.

Device Management

Managing hardware infrastructure isn't anywhere near as bad as it used to be. Prior to virtualization, every single server had its own firmware, management software, device drivers, and operating system software that needed to be continually monitored and maintained. Although the outlook is better now, this challenge has not completely gone away with virtualization. We still have our VM Hosts to manage and maintain, but it is a far sight better than it was before when it comes to the sheer numbers of devices and the data center footprint to be managed.

Managing hosts and various devices within an infrastructure requires mounds of specialized and proprietary software from the device manufacturers. If not configured or interconnected properly, this is yet another silo of micromanagement that is required to be configured, maintained, and backed-up.

In addition, outside of widely deployed virtual desktop infrastructures, one cannot dismiss the attack surfaces inherent in end-user desktops and devices as well. More than ever, users are introducing their own laptops and mobile devices into their work life, connecting them to your networks, and storing company data on them. This is a component of a phenomenon known across the industry as "shadow IT."

Shadow IT

The term *shadow IT* can raise an eyebrow when it's mentioned, so let's define it!

Shadow IT is a phenomenon where users within an organization deploy and consume technology services that are not under the control of the corporate IT organization. For example, using a cloud-based file storage and sharing service like Dropbox apart from the blessing and oversight of IT would be a good example of shadow IT.

Shadow IT Survey Feedback

In a recent survey conducted by ActualTech Media, 49% of respondents said that cloud file sharing services like Dropbox, Box.com, and Microsoft OneDrive were in use within their organization. It's likely that responding in the affirmative indicates that the use of this service is sanctioned by IT. Is it possible that some of the remaining 51% are also using file sync and share services, but without the blessing of IT?

The practice of circumventing corporate IT policies and using alternative tools in order to get work done is understandable, even if it's not allowable. Users want to be able to do their jobs effectively and with as little friction as possible; if corporate IT can't or isn't willing to provide the experience they're looking for, users will probably look elsewhere. The problem with letting things get to this point is that it poses a massive security risk.

How Can IT Maintain Control?

Understanding where and how shadow IT practices are creeping in to your organization can be tricky. First of all, users typically know when they're side-stepping IT and are often less than forthcoming about the services they use from third parties other than corporate IT.

Aside from just asking, there are some not-so-pretty ways to develop an understanding of what's going on, such as manually reviewing firewall logs or web proxy logs. A better way to approach this challenge is by leveraging a monitoring tool that does comprehensive deep packet inspection as well as anomaly detection. By reviewing a rich set of metadata about the traffic and a baseline of what's normal within an organization, it becomes much easier to detect unsanctioned tools cropping up within a network.

End users employing applications outside the purview of IT exposes the organization to the risk that sensitive company data will be obtained by unapproved parties. It also provides an additional attack vector for bad actors to use to infiltrate the network and begin wreaking havoc inside the network. Needless to say, it's important that IT stays on top of shadow IT.

Transaction Forensics

Speaking of auditing, a phrase that has become more and more common in recent years is Transaction Forensics. While this is mostly found in global e-commerce markets and mergers and acquisitions, some of the principles can be applied to infrastructure, application, and user activity throughout the entire data center. Think of it as logging, but extremely organized, where every communication and every event are logged in a way that highlights who and what was interacted with.

Being able to instantly identify who did what, to what, and when, is one of the most complex auditing challenges an IT organization has to deal with on a daily basis. Having strong audit logging and transaction forensics in play can make or break some organizations, especially those that are public entities or government institutions. Considering the threat landscape today, this is not a nice-to-have feature, but a requirement—especially for those kinds of IT departments.

Microsegmentation Challenges

If we described IT like a chapter of Genesis in the Bible describes creation, we would say, "In the beginning, there was the computer. It had a hard drive. A software application was installed upon it. A user interacted with that software. Something came out." Simple, right?

Well, now we live in a world where millions of users can interact with the same application at the same time, producing completely different results. This requires a level of scale at the application layer never considered before. For a period of time, virtualization got the job done for us. It provided a way to consolidate the data center into centralized hosts that supported hundreds of servers, each running an application. This consolidation spawned an entire industry and third-party ecosystem to support that infrastructure archetype, and over time, the successful practice of supporting, maintaining, and securing that way of life was achieved.

With the re-emergence of cgroups (a.k.a. "containers"), we're now facing the microsegmentation of individual services, breaking up an application into many pieces and parts. While this gives us even more capability to scale, support, and stay online, it re-introduces the cyclical world again where an entirely new ecosystem needs to be spun up in order to support and secure this new standard of application development.

Microsegmentation and Hybrid Cloud

As if microsegmentation within a data center isn't hard enough, introducing microsegmentation in a hybrid cloud scenario in which different parts of the application could be scattered across multiple clouds is exponentially more complex.

As you read about in Chapter 2, mapping application dependencies is an important objective for many organizations, especially when a hybrid cloud architecture is in play. Understanding the way different parts of an application work together and which applications depend on each other is key to a successful hybrid cloud adoption. With the narrow, granular, and highly restrictive security policies that microsegmentation employs, your application dependency map has to be incredibly precise if you're going to successfully microsegement applications that cross clouds.

From a security standpoint, the idea of microsegmentation means that IT can allow communication between these very specific parts of an application and deny all other communications. By implementing a security posture that denies access by default and only exposes services very selectively and granularly, the data center is hardened significantly. Of course, while that sounds great in theory, it's incredibly hard to do in practice. Locking down the network without a deep and broad understanding of exactly who's talking to whom will result in so many failures and outages that the pushback from end users and the business will likely stall or completely undermine the initiative.

Securing a Business with Application-Centric IT

It needs to be made clear that running an application-centric IT organization cannot be handled the same way as a virtualized infrastructure, or even a more traditional client-server siloed infrastructure. While it is more of a philosophy and way of life within the department and its staff than a particular model of server or software you can buy, it's something that needs to be adopted wholly or not at all. Otherwise, you run the risk of becoming split-brained in the ways you monitor, maintain, and secure your company's data.

The best way to do this is by examining your entire department, identifying workloads and the underlying resources that support them, and securing workloads as a whole.

As much as people want to knock the old-school way in which siloed client-server stacks were built and maintained, they were a shining example of what is commonly referred to today as a self-contained "full-stack" centered around the application that had its own monitoring, maintenance, security, and backup software specific to that particular application and nothing else. The root difference between then and now is that shared resources underneath a particular application are likely also being used to support 1+n other applications in any given department. So, the same software used needs to be able to support *all* applications leveraging those resources.

Even though we haven't gotten to this Mecca yet, we are closer today than we ever have been.

Chapter 5

Private and Hybrid Cloud Migration, Monitoring, and Analytics with Uila

The ultimate arbiter of application performance is the user. Is the application working well for them? If not, users call the help desk. But it's notoriously hard for IT to analyze issues from the end user perspective. Even when multiple users report application performance issues, IT's suite of monitoring tools may not reveal a clear issue.

With Uila, IT can see performance from the user perspective - the way IT Operations expects to see it. Uila's full stack visibility enables IT to quickly identify root causes anywhere in the stack. If a user reports an outage or performance issues, IT can pinpoint where issues are occurring - whether it's the user's connection, a specific virtual switch, an over-provisioned host, application or database server, or underlying storage.

Uila provides deep insights and analytics into web and database application performance. This is done by collecting application response times through the network and by reading transaction codes and queries from network packets. The result is deeper insights into client and server errors or constraints so that the issues can be hunted down and mitigated.

In most virtualized data centers, no single tool provides full stack visibility from the end user to the application to the underlying infrastructure layer. But without this visibility, IT infrastructure and operations teams routinely struggle to identify root cause of application performance issues that pull in multiple teams and take too long to remediate.

Uila's full stack visibility for virtualized data centers reduces troubleshooting time from days to minutes, enabling lean IT teams to get time back for more strategic projects. IT teams can identify blind spots in the infrastructure to head off performance problems at the pass, and eliminate finger-pointing between infrastructure and application teams with automated root cause analysis and forensics.

Uila also provides IT teams the mandatory Pre-Migration Private Cloud Assessment and Post-Migration Validation of performance of the business-critical applications in a single pane of glass.

This includes:

- Application services inventory map and current performance baseline
- Comprehensive resource provisioning and usage for compute and storage
- Application dependencies and topology map
- Single pane of glass for correlated App and Infrastructure visibility
- Validate business-critical application availability and end-user performance improvements desired from the hybrid cloud migration

With Uila, IT teams are guaranteed the identification and availability of the required business-critical application dependencies and VMs after the actual disaster recovery.

In addition, you will gain real-time network security insight with comprehensive applications, network, and infrastructure, as well as East-West traffic statistics across VMs. This provides you with full visibility into what "normal" network security looks like by detecting abnormal patterns of connections and traffic flow, using advanced packet capture for deep packet inspection and forensic evidence, indicating a potential network security threat that could impact business service performance and operations.

Uila also gives IT teams the data they need to optimize application performance with the existing data center infrastructure with clear visibility into future infrastructure needs, based on existing bottlenecks and hotspots. Uila's analytics across the entire application and infrastructure stack also helps IT find underutilized resources and even unused infrastructure software licenses.

With Uila, IT can be much more strategic and selective when it comes time to upgrade infrastructure.

Knowledge Check

Answer these questions to confirm your knowledge of modern security challenges and how to address them.

- What is shadow IT?
- Why is migrosegmentation attractive but difficult to implement?
- What are some unique ways that Uila can address security concerns?