



Migrating Your Workloads to Azure

So, you've just signed up for a Microsoft Enterprise Azure agreement. Your management team or business partners have told you that you need to reduce your dependency with on-premises datacenters. Your developers have been working on the next generation applications that will change the fortunes of the company, but have requested a large amount of high-performance infrastructure to develop and test the application at scale, as well as service multiple code branches. Whatever the motivation, navigating the journey to the cloud can often present a daunting array of choices to the IT professional when considering how to migrate to the cloud.

Related Technologies



Microsoft Azure



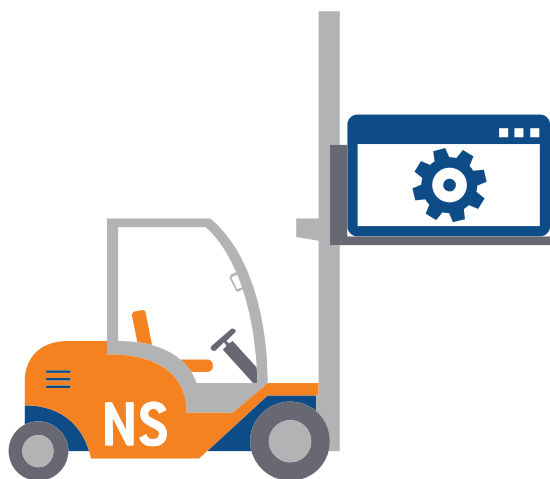
Infrastructure



Cloud Migrations

Direct Migrations

With any cloud migration, it is always useful to reflect on the compelling driver for the migration. If the answer is that the datacenter contract anniversary is due, perhaps the migration needs to be accelerated, thus moving assets to Azure more commonly known as Lift and Shift, forklifting or, the modern phrase, Direct Migration. Clearly, shipping physical servers to Microsoft Azure datacenters is not an option, but shipping VMs just might be. This approach has the key advantage that you can migrate workloads relatively quickly but the existing on-premise problems, configuration, support challenges are also migrated to the cloud.

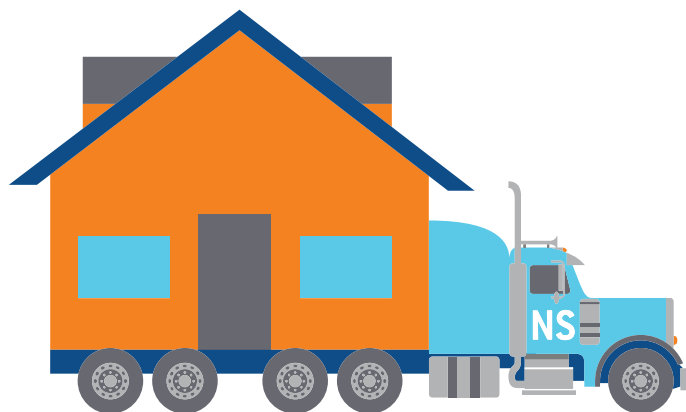


In fact, the problems are often exacerbated due to the previous availability assumptions and architectural interfaces. Operational charging models all present different challenges in the cloud. Assuming no reasonable alternative, if there is still a compelling short-term date to migrate, then there are many commercial tools available, ranging from the Microsoft Azure migration tool set, such as Velostrata, Bitnami and Corent. Some toolsets will provide fully detailed cloud assessment based on existing on-premises workloads whilst other tools offer an online and offline bandwidth-efficient block-level migration of an entire virtual machine workload. The ultimate choice is mandated by the speed of migration and supported workloads. For example, the native tooling from Microsoft supports operating systems from Windows 2008R2 and above. Velostrata, however, supports the migration of older operating systems such as Windows 2003.

Rehoming or Re-platforming

Assuming then that your compelling event is to remove the reliance on the datacenter, or perhaps the hardware in that datacenter is reaching a natural end of life, but you have a reasonable execution window and budget. Perhaps the SAN of virtualization host requires considerable care and attention to maintain your daily business SLAs for uptime and availability. Let's also assume that you have many off-the-shelf applications and you still have the means to install said applications. This scenario allows for the construction of a brand-new cloud-hosted platform with all the operational support features required for a production system, such as backup, monitoring, connectivity and, of course, security.

There are several considerations before you successfully re-home your application. First and foremost is the application supported in the cloud. There is no point in moving your production ERP system to the cloud only to find that you have no support available. The assumption is that the installation media is available, and of course it is not possible to ship CDs to Microsoft, so a media image will be required. It is also assumed those applications will also run on a modern operating system. Azure machines currently run 2008R2, 2012, 2012R2, 2016 and a range of Linux platforms. 2003 is only supported for direct migration but with best endeavors, you cannot create a 2003 machine from the Azure gallery.

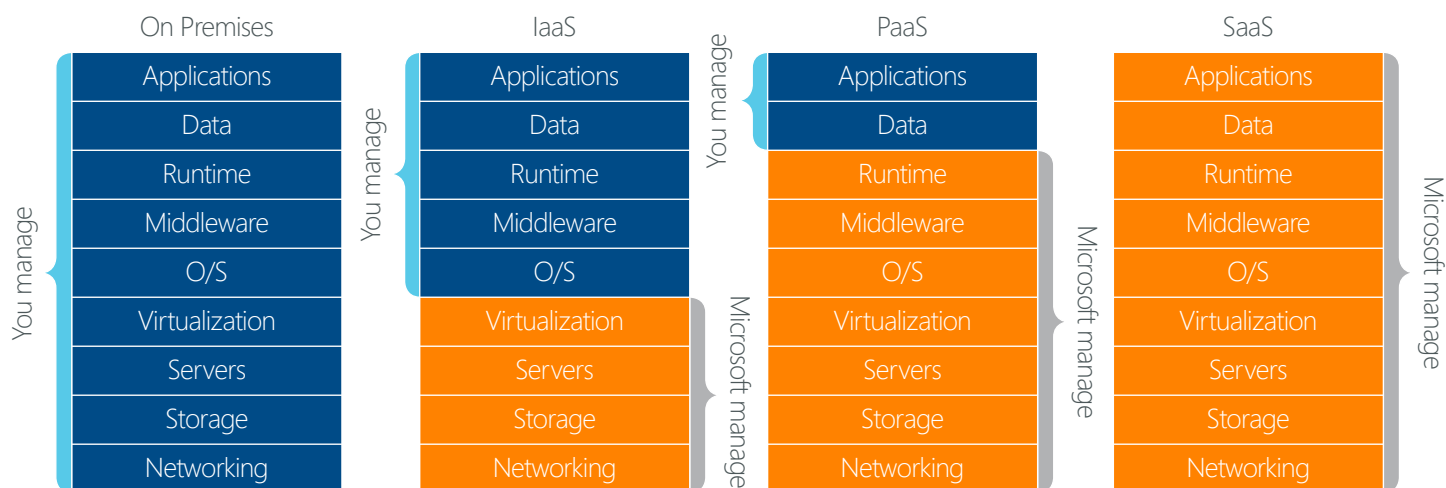


Other technical challenges generally relate to the application integration requirements. For example, it may depend on some internal or third-party data integration feeds. Perhaps there was an email

capability that was taken for granted on-premises. You discover that it has a tight dependency on its incumbent cluster hosting technology. Perhaps the application requires a license dongle attached to a physical server. These and many other architectural questions must be challenged and satisfied as part of an application discovery exercise. Only then can you proceed with the migration.

Re-Architecting for the Cloud

One of the key benefits of moving workloads to the cloud is the ability to defer some operational responsibility to the cloud provider, refer to the following diagram:



For on-premises workloads, the customer is responsible for everything from purchasing and maintaining hardware, or datacenter to license management, performance tuning backups, security, software and hardware refresh. When you migrate your VM to the cloud or re-platform your application onto a gallery VM, you still have the responsibility of maintaining the operating system, and any operational responsibilities above that. This modus operandi clearly provides the greatest level of control for any customer, but is not necessarily the cheapest or most efficient way of consuming cloud services.

For example, the on-premises application has a SQL database placed on an enterprise SQL server cluster, but does not require any of the additional SQL features such as SSIS or SSRS or SSAS. In this case, consider using the platform-as-a-service SQL

offering from Azure. It supports a range of scale and performance options according to the application demands. Encryption at the database level comes standard and creating a redundant instance in another Azure Region, is a configuration choice. Of course, your application may have been developed as an ASP .Net or a PHP application. The on-premises solution will invariably be distributed across many web application servers configured for high availability. Azure, again, offers a platform solution for hosting your application in a web app service. With this feature, there are many fault-tolerant and auto-scaling features that can be leveraged out-of-the-box. Perhaps, for example, a mail shot has been distributed to the customer

base and hits to the website have been forecasted to sky-rocket over the weekend. Scaling a webapp can thus be configured to automatically increase compute capability on a schedule, resulting in increased availability to customers, less abandonment from the website due to poor performance, and, from an operational perspective, costs have been controlled according to the schedule. Invariably, this business model works very well if the web transactions lead to conversion into revenue.

Re-Develop

By adopting modern agile practices, the migration can be dramatically accelerated by utilizing a microservices architecture. Aspects of the core application can potentially be modularized. For example, a low-priority function could be used as a



confidence-building development allowing developers, testers and Devops teams to become familiar with the new work environment. Success or failure in the early phases can be remediated with little impact to the overall budget. Key changes for any organization will be the cohesion between development and it operational staff to form a new highly agile Devops capability rather than traditional “analysis paralysis”, procurement lead or waterfall processes.

Retire, Replace, Tolerate

Again, if the application is from a third-party, the vendor clearly needs to support the PaaS technology stack. In-house authored applications are generally appropriate for this migration model. It also worth asking the vendor whether they have a PaaS appliance in the Azure Gallery or a Software-as-a-Service version of their application. For example, a Microsoft Dynamics on-premises solution could conceivably map to the Dynamics


365 solution from Microsoft. Of course, it may not always be relevant or beneficial to migrate a particular workload to Azure. An application fulfilling a legacy function for a small group of users may not be appropriate for the cloud without some remediation. In this case, it may be beneficial to “tolerate” the application and retain it on-premises somewhere. While considering migrations to the cloud, this is often a perfect opportunity to determine if there is a more current version of the application that is already cloud-ready. Perhaps there is a SaaS version available, thus the migration project transforms into a data migration with configuration activity.

Of course, when evaluating the IT estate, there may be multiple instances of the application serving different business functions or departments. Again, as a migration and transform activity, one should always look to consolidate where appropriate. If not, consider how using cloud qualities to scale out or scale up could be beneficial to these applications.

Take steps toward your best future with Azure. Contact a New Signature expert today.

NEWSIGNATURE

 www.newsignature.com
 [@newsignatureuk](https://twitter.com/newsignatureuk)

 +44 (0) 845 402 1752
 hello@newsignature.com