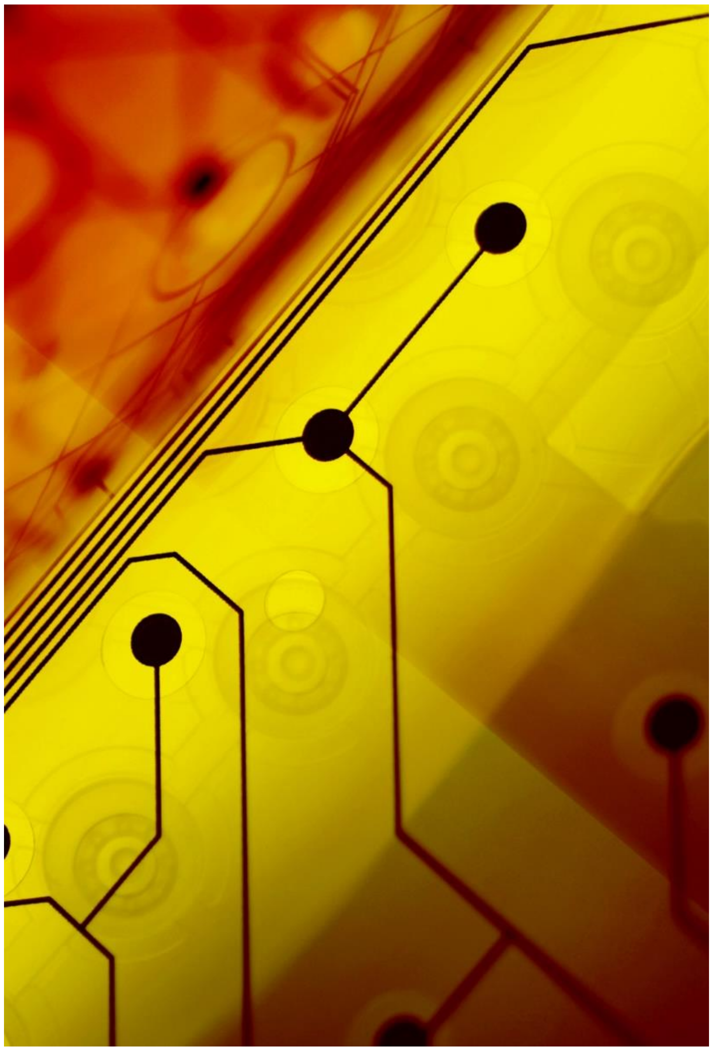

Helping obstinate On-premise clients move to Azure using Service Fabric

Rohan Wadiwala

Introduction

Cloud platform has drastically changed the IT landscape in the last five years. Thanks to various advantages provided by cloud platforms we have seen phenomenal growth in cloud adoption by our clients. One of the leading platforms leading this disruptive phenomenon has been Microsoft's Azure cloud platform. Azure with its wide range of SaaS and PaaS offerings has provided many organizations that are primarily Microsoft shops with a simple and easy way to migrate their application to cloud.

Due to these features, many new businesses are basing their complete IT infrastructure on cloud. As they don't have any on-premise assets, it's easier for them to start from scratch and reap the benefits of size and scale. Unfortunately, many customers have already invested heavily on their on-premise environments. This investment might be due to the long-term vision of the company or due to legal limitations. Many of these customers are either locked out due to above reasons or simply reluctant to change. This whitepaper focuses on providing a low-risk migration path for such customers focusing mainly on Microsoft platform.

Cloud Migrations

Current Methodologies

Generally, the current migration strategies depend on the size and type of application. In case of a smaller non-critical application customers generally opt for a bit risky but low-cost option to directly migrate their on-premise application to SaaS or PaaS services. This strategy ensures the upgradation takes minimum time as no intermediate stages are involved.

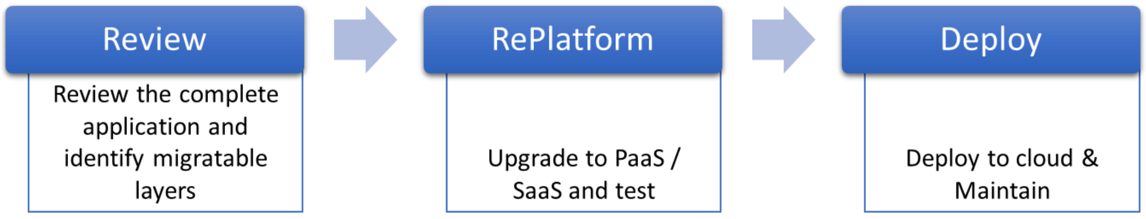
Review
Review the complete application and identify migratable layers
RePlatform
Upgrade to PaaS / SaaS and test
Deploy
Deploy to cloud & Maintain

Figure 1: Upgrade process for small / Non-LOB application

In case of a medium to large application with high business impact, a more cautious approach is preferred where-in the application is first migrated as-is to cloud infrastructure (IaaS) and not many changes are done to the application code and architecture. This gives the customer a chance to test the waters as well as upgrade their own cloud capabilities. Once the customer is confident enough in both viability of cloud for his business and internal employee skillset upgrade then the more elaborate migration occurs where-in the application is migrated/re-architected to a SaaS or PaaS solution.

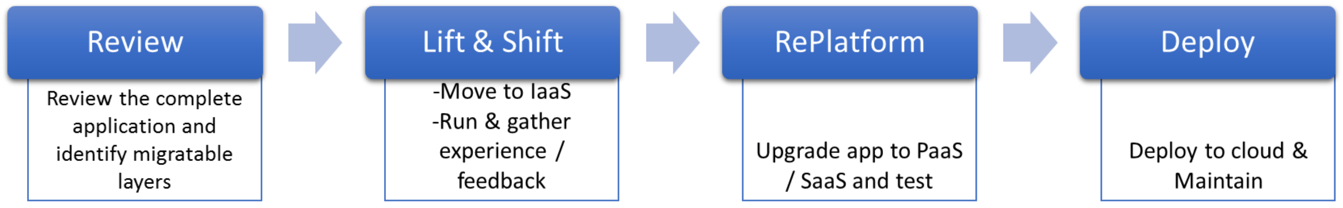
Review
Review the complete application and identify migratable layers
Lift & Shift
-Move to IaaS
-Run & gather experience / feedback
RePlatform
Upgrade app to PaaS / SaaS and test
Deploy
Deploy to cloud & Maintain

Figure 2: Upgrade process for large / LOB application

Issues with migration

Though, both the approaches provide excellent results; as mentioned earlier some clients cannot move to cloud or adopt any of the above methodologies because of:

1. Legal issues
2. Client's short-term vision
3. Internal resistance
4. Low confidence due to lack of knowledge
5. Risk to business continuity in short term

There is not a lot we as consultants can do when it comes to internal resistance, but for customers who are restricted only in short term or feel intimidated by the migration efforts or continuity issues; we can provide an option to build-up confidence as well as improve their familiarity to cloud by migrating to an intermediate technology which is available on both on-premise and on cloud. This way they not only get a taste for working on recent technologies and cloud but are rest assured that there will be very little or no impact to their business.

Enter Service Fabric

So, what can we do help such customers; well one way of assisting them is to move them to an on-premise platform which can work on cloud too. The advantage of moving to such a system agnostic platform is threefold:

1. Application gets upgraded to a more flexible platform and thus clients can maintain the upgraded application on their on-premise setup indefinitely (or till the time they will want to move to cloud).
2. Secondly, this upgraded application & platform can be scaled much more easily both horizontally and vertically as compared to the original application. In many cases, we

can just increase the density of the deployment to get the same benefit as that of horizontal scaling without the need to add more hardware.

3. Lastly and more importantly, clients won't have to reinvest in redevelopment when they move to cloud as their application will work seamlessly.

One of the technological platforms which can be considered in this scenario is **Azure Service Fabric**. Azure Service Fabric is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable services, applications and containers. Service Fabric also addresses the significant challenges in developing and managing cloud-native applications. Developers and administrators can work around infrastructure problems and put their energies behind implementing mission-critical, demanding workloads that are scalable, reliable, and manageable. Service Fabric represents the next-generation platform for building and managing these enterprise-class, tier-1, cloud-scale applications running in containers.

Three most important feature of Azure Service Fabric which we can utilize in this scenario are:

- Any OS, Any Cloud:

  This feature basically means we can run Service fabric either on-premise or on Azure and any operating system (Linux or Windows)
- Open Source

  Service Fabric is an open-source project which inherently makes it secure; as well as ensures that client does not have to pay any royalty thus helping to maintain cost.
- Write once, Deploy anywhere

  It's not necessary to write service fabric specific code for it to work on Service fabric. A regular web app or even an app created for containers can work seamlessly. This feature encourages easy migration from on-premise to Azure, as no extra coding effort are needed to port the binaries.

Migration strategy using Azure Service Fabric

As mentioned earlier the basic strategy is not to MIGRATE the user to cloud directly but to RE-PLATFORM to Service fabric for the intermediate term. Once re-platforming to Azure Service Fabric is done, the client can review, consolidate and enhance the application as before as the application is still in his data-center. Once the client is confident enough this environment can be extended to cloud or can be moved directly to cloud. In case the client does not find value in migrating to cloud then they can persist with the on-premise deployment of service fabric which itself provides high scaling functionality which replicates horizontal scaling without incurring any extra cost.

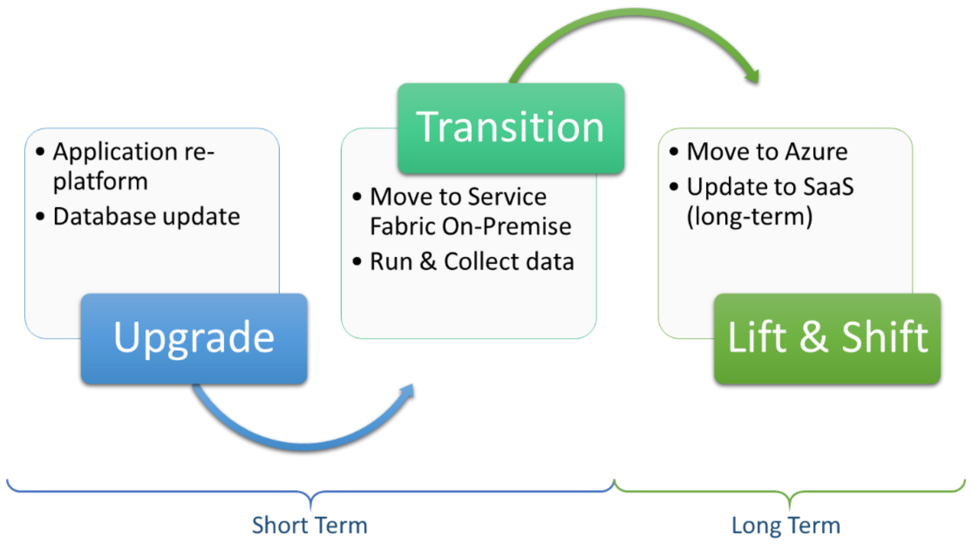

Figure 3: Overall Approach

The steps mentioned above can be detailed as below:

1. UPGRADE

This step encapsulates upgrading the applications' code to a newer platform. This step helps to create a ground-work for the next steps. In Microsoft based projects, it's always preferred to move from traditional .net framework to modern .net core framework.
This step can also involve moving from on-premise databases to a SaaS database if it suits the needs of the application.

2. TRANSITION

The second and probably the most important step of the flow is transitioning the application deployment from the on-premise OS environment to service fabric (or Docker if you like). There are no code changes needed during this step and this can be achieved only using minor configuration tweaks, but to gain full advantages of the Service fabric platform, some minor changes can be done to the application (e.g. using queuing components of Service Fabric or Caching feature of SF)
The most important task which needs to be undertaken during this step is deploying the application to service fabric and monitoring the state and throughput of the application. Results from this monitoring phase can be interpreted and decisions can be made to change service fabric configuration and implementation to suit the requirements and stabilize it.

As mentioned earlier, clients can continue using this on-premise implementation indefinitely and can reap various benefits provided by Service Fabric inherently like scaling, densification, fault-tolerance, etc.

3. LIFT & SHIFT

The last step of this journey will obviously be to move the whole service fabric environment to cloud. As Service fabric works and performs in the same manner on both on-premise and cloud environments, this is quite risk-free movement (which is generally one of the major concerns of the client).
This step should only be undertaken once all the predicament faced by the client for cloud implementation are resolved and the client is confidence enough to migrate to cloud.

Once this step is implemented, the client can get an actual feel for the advantages gained by moving to cloud and hopefully will be impressed enough to move the application to a more flexible SaaS-based model.

Titbits to consider

Service fabric can provide exceptional ROI if implemented correctly but like all technologies, it is just a tool and using it incorrectly can lead to some unfavorable outcomes. Hence care needs to be taken when setting up the environment and the deployment strategy.

Similarly, Service fabric is relatively new and is geared towards automated environment setup, so a lot of the service fabric setup & configuration is done using xml like files, unfortunately not many GUI tools are available to create these files (though some tricks can make life easier), so consider adding some time for creating/editing these scripts manually while estimating for such migrations.

Finally, service fabric is a powerful platform providing plethora of configuration options. It is unlike other platforms which your time might be acquainted with (e.g. Docker) hence it does have a learning curve (though not a very steep one).

Many ways one goal

Thus, to conclude, the mentioned methodology is just one of the many ways we can ensure a seamless transition from on-premise application by ensuring reliable deployment for clients who are reluctant or a bit skeptical about cloud migration. This strategy can be modified according to the timelines and venturousness of the client and his end goal. But, this strategy cannot be implemented in all scenarios. It’s always advisable to review the scenarios on a case by case basis. Capgemini internally has multiple programs which can help develop these strategies and units like Sogeti’s OneDeliver and Capgemini’s own cloud services can aid in this endeavor.