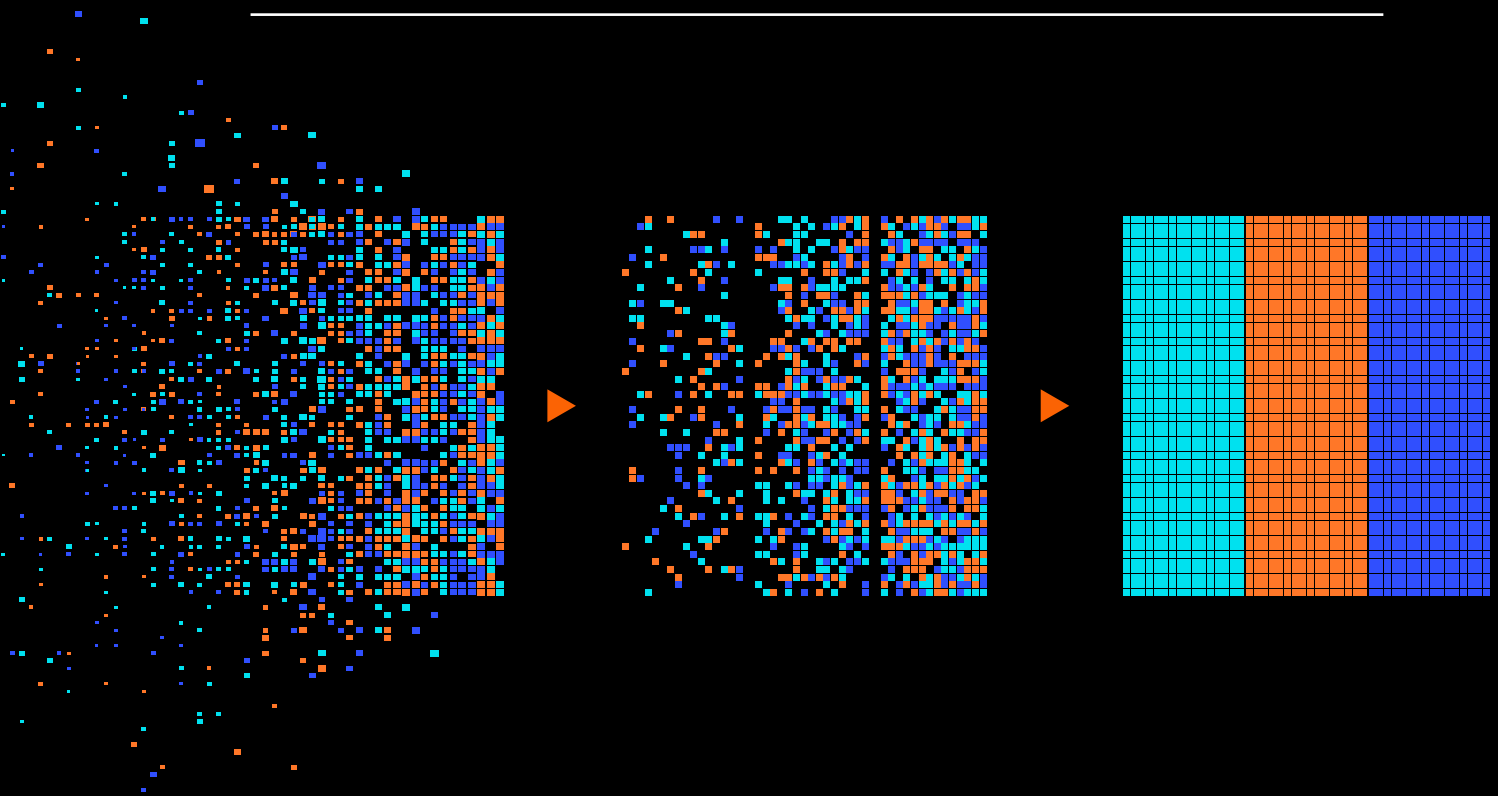




# Principles and Strategies for Getting Started on Microservices and Containers to Drive Your Business

---



# Introduction

The business world is moving faster as we near the third decade of the 21st century. Enterprises face competitive challenges that can come from anywhere at any time. Opportunities arise just as quickly and might be fleeting, if not acted upon immediately. Two overriding management priorities have emerged as companies navigate these swift waters: business models built around enhancing the customer experience; and a focus on operational efficiency. Implementing those priorities is complicated by several fluid situations:

- Customer expectations are rising
- Competition is accelerating
- New technologies constantly promise better experiences and more efficiency

Cloud migration has been a common response among many organizations as a means for improving operational agility and transforming customer experiences. While this frees up resources that might otherwise have gone toward server and application management, this alone does not solve the problem of staying nimble, flexible and responsive to business units and their changing needs.

Re-architecting the enterprise with technology requires rearchitecting the way that IT services are created and delivered. Companies need to adopt modern application development processes to shorten the build, code, test, delivery cycle and expedite time to market. The first step in this process involves migrating to microservices architecture and containers. These two technologies can be a powerful combination that enables increased agility, speed and resource efficiency for many tasks that developers work on daily. But the move is not without risks.

Inserting the microservices mindset into operations and culture, can be challenging, as the approach sometimes conflicts with the more methodical ways of IT. To avoid pitfalls, this eBook outlines principles and strategies for getting started on MSA and containers.

It will help you understand:

- How to work with microservices and containers productively
- The top challenges that organizations face in their microservices journey
- How people, processes and technology can be brought together for business success in this new paradigm.

To set context, let's look first at the fundamental shift driving these new considerations.

# The advent of microservices architecture

Monolithic applications have been the traditional mainstay of IT services. While they generally come from vendors already fully developed and integrated with libraries and databases, this model also comes with inherent drawbacks. As the application grows in size and complexity, it becomes more difficult to update and takes longer to implement changes. You must redeploy the entire application with every update making continuous deployment difficult. Reliability also becomes an issue as a bug anywhere in the application can bring the entire process to a halt.

Shifting traditional monolithic applications into the cloud is only a first step to digital transformation. The application will benefit from some efficiencies gained from running in a virtualized environment. Yet it is still an all-encompassing application, often containing its own databases, libraries and execution pipeline. For any given business function, it might be more than is necessary or at times has to be supplemented with additional functionality. It runs by its own internal logic and is therefore architected to limit flexibility. Essentially, business functions must navigate through the application's logic whether every step is required or not. That is not conducive to speed and flexibility. Ideally, applications should adapt to the business situation.

The full benefits of cloud-native IT approaches require more than just cloud hosting. Application architects have developed ways to disaggregate these monolithic applications into smaller building blocks that match specific business functions that are standardized and widespread in use. This approach is known as services architecture. As the building blocks grew more fine-grained, the term micro-services architecture (MSA) arose to describe an environment geared toward numerous tightly defined functions that can independently be deployed and combined on the fly to match a specific business need. MSA can significantly accelerate the production and deployment of an application with little to no wasted processes or components found in the monolithic approach. As the name implies, a microservice is a single building block highly optimized to accomplish one specific function.

For example, the process of ordering items over the web has numerous steps: verifying inventory, confirming price, taking shipment information, logging revenue, confirming credit card information, etc. Monolithic applications could certainly do this. However, the individual functions cannot be shared with other applications that have similar steps. In MSA, these functions become microservices that can be called on to perform their tasks for many business needs that require inventory verification etc. If demand increases, they can be replicated to match function supply with demand.

A given microservice still needs data and operating system functions to execute. While it's possible to create an application based on microservices alone, placing them into containers improves security and makes it easier to monitor, manage and scale the individual pieces. MSA is usually associated with containerization. Containers provide a lightweight means of bundling the necessary data and an OS kernel into a package that can execute while isolated from other pieces of the system or other workloads in the environment. Containers do not require the system overhead of virtual machines. IT operations managers can scale functions much faster and more smoothly than virtual machines by simply launching more containers with the same microservice, alleviating the more elaborate provisioning virtual machines entail.

For many enterprises, MSA and containers address the needs of the new state of competition. Practitioners, however, caution against leaping into this approach without a carefully constructed plan. Managing an MSA environment is not necessarily easier than traditional monolithic applications. Proliferating microservices and containers without proper frameworks can create an environment that's hard to manage and has a great deal of overhead. It can slow execution and create faults. The following sections outline the issues to consider at the outset of the transition.

## People

For a traditional IT staff steeped in maintaining systems and developing monolithic applications, the move to MSA is a major culture shift. It also requires new skills that must either be developed or acquired through new hires. The IT department must be prepared for this transition.

Culturally, MSA and containerization needs an Agile model of development to be successful. This means speed and efficiency must be prioritized. Quality cannot suffer, of course, but a minimum viable product approach to application development and delivery – with incremental improvements over time – becomes an acceptable way to view any given program. That is quite a change for staff accustomed to taking months or even years to deliver a project.

The mindset of your people needs to shift from a “project mentality” to a “product mentality.” Integrated cross-functional teams are responsible for building and deploying an application constructed out of these MSA building blocks. They are jointly responsible for the entire life cycle of that application, rather than a single step in a more traditionally segmented production pipeline.

That integrated team approach can only happen in a management environment where silos are broken down. New communication approaches are necessary as team members must work together in new ways. This new team must also have a greater connection to the business units demanding new applications. This sense of connection could go so far as having business unit personnel who are either formally or informally members of the integrated team. Speed and efficiency demand this interconnection with the business unit to understand their real needs and how they translate into the MSA or if they are better served by a monolithic application.

Navigating this culture shift and operating productively in this new way requires new skills among the IT staff. Coding will always be important, but this new environment places a premium on architectural skills. Deconstructing monolithic applications into their most efficient, productive services becomes a new discipline that might well be unique to any given enterprise (i.e. a retail operation might approach it differently from a manufacturing concern). In creating new applications within the MSA, the building blocks are largely pre-formed rather than coded by hand. The ability to envision the various required components of a larger application and how data flows among those components is highly valuable.

Likewise, staff need to draw on established and emerging practices in the Agile development and DevOps spaces such as extreme programming, the frequent release methodology to improve performance and response to customer demands, and continuous integration/continuous delivery. The sharing of responsibilities under these methods (e.g. testing, customer communication) becomes part of the dynamic in constructing teams. Teams also become responsible for assessing further automation steps and the need for new applications.

“The mindset of your people needs to shift from a ‘project mentality’ to a ‘product mentality.’ Integrated cross-functional teams are responsible for building and deploying an application constructed out of these MSA building blocks.”

---

## Technologies

The complexity of managing multiple containers, each with its own microservice, can quickly spiral out of control. With so many moving parts, security, scalability, and reliability become more complicated. A new technology framework is necessary to manage this distributed architecture. The key technologies necessary for building this framework consist of:

- Containerization
- Orchestration
- Monitoring

Unlike many monolithic applications that are products that can be purchased from established corporations (e.g. Microsoft, SAP) most of the container and microservices technologies are available as open source technology. They can be very effective in their function, but they don't come with a built-in service contract. You might need to adapt them to your specific needs and contribute your adaptations back to the open source community. Tools and technologies that have broad industry support will provide the best options for updates and new modules to address specific needs.

Containerization tools package a microservice with the additional data and components, such as an OS kernel, they need to execute. The important point about using such a tool is that it standardizes this process of constructing containers across your environment. This makes the use of code, run-times, libraries and communications consistent in all the many building blocks populating the environment.

That consistency of construction translates into the consistency of operation in production. By far, the most popular containerization tool is Docker, an open-source project that automates the process of developing, deploying, and running applications inside containers. It provides a consistent development environment for the entire team and enjoys broad support among developers.

Orchestration is a management layer that interfaces the individual containers to each other and to the ultimate application being constructed on the fly. The orchestration layer is responsible for maintaining service level agreements (SLAs) and scheduling services into optimal execution resources on the host. Orchestration implements the “loose coupling” function of the MSA model, maintaining all the necessary linkages while still allowing the individual building blocks to operate independently. Kubernetes, an open source contribution from Google, is the most widely used orchestration technology though Docker Swarm is also a well-known entry.

Monitoring tools provide visibility into how microservices are being used. Traditional monitoring tools from a monolithic application model are not built to comprehend the loose coupling, communication and multi-component nature of MSA and containers. Monitoring tools allow you to pinpoint where any errors occur and which microservice is causing the issue. Prometheus is a well-regarded option for monitoring.

These three technologies form the core of the MSA and container environment. While they are all necessary, there are options for establishing the complete framework. Essentially, this is a classic “build or buy” decision. You can construct a customized version of this framework on your own. This will take time but has the advantages of customization to your particular needs. Otherwise, Platform as a Service (PaaS) providers offer preconfigured, open source components that make it easier to implement a container framework, though they are often standardized in ways that set some limitations on flexibility so they can be applied across a variety of users. Many of these use Docker and Kubernetes for managing the different components. Amazon Web Services and Microsoft Azure both offer PaaS options as well as Cloud Foundry.

## Processes

Changes in people and technologies imply changes in processes. Yet, several important process points go beyond the use of any given tool or the addition of a skill. They involve management approaches to building the use of MSA and containers as a core competency of the IT organization.

Choose a greenfield project to launch an MSA/containers initiative rather than an existing mission-critical application. This offers several advantages, the most important of which is avoiding major disruption in the business. It also reduces pressure on the team or teams as they work out their internal dynamics and new roles. This core team will be tasked with documenting the way this new architecture functions. Data flows differently in MSA than it does with monolithic applications so data governance and analytics standards must be developed for inter-service data flow as well as API calls and any needs for database interaction.

By starting small it will be possible to work out all the important parts of building this new core competency across the organization. The principles of Agile development and DevOps disciplines such as continuous integration/continuous delivery can be pioneered in a more controlled way than attempting to change the entire IT program at once. Once this new product is delivered and the teams have gained experience, they can become a de facto center of excellence to spread these techniques and approaches across the organization as new programs arise and more ambitious undertakings are targeted.

As this eBook has noted, one of the biggest opportunities in moving to MSA and containers is the ability to reduce the complexity of monolithic applications to speed the execution of the core functions the business actually relies on. Care must be taken in this disaggregation process, however. Think of it as a Goldilocks approach. You are looking for the size of service that is neither too large nor too small. Both rob some of the efficiencies of the overall MSA approach. An incremental approach to this dismantling process is warranted. The basic concept is to start with the monolith and break it into macro services, still relatively large but offering a view into the workings of the applications and the components that make it up. The next step is to break those macro services down into mini services – considerably smaller and more focused but not necessarily single-function yet. The next step, then, is the single-function micro service.

By taking this incremental approach to breaking down the monolithic applications, IT organizations can find the mix of monolithic applications, services architecture in general and specific MSA functionality that works for the business. Most large enterprises will eventually run with this hybrid approach. Changing business needs will undoubtedly change the mix on a regular basis. The experience of working through this balance can guide future appraisals of what needs to scale up or down.

“Think of it as a Goldilocks approach. You are looking for the size of service that is neither too large nor too small.”

---

# Conclusion

Change is not only happening faster; it is becoming the defining condition of business. Creating a nimble, flexible, learning organization is the key to sustainable success. New imperatives require new approaches.

It is a given that IT provides the underlying platform for success. However, simply moving to the cloud is not enough. Cloud-native approaches to development, deployment and operations must be employed to optimize customer experience and overall efficiency.

MSA and containerization have emerged as two intertwined approaches to creating this new type of organization. They offer tremendous benefits. Yet, blindly shifting to these technologies without a tailored plan and the appropriate resources is a recipe for potential disaster.

Effective implementation of MSA and containers requires a thorough plan for the necessary transitions in people dynamics, technologies and processes. Trusted advisors can help with evaluating the choices and tailoring this new model for your particular business.







**Go2Group**, an IT consulting firm focused on providing superior quality professional services and digital solutions, helps enterprises achieve business agility. Founded in 2002, the company offers enterprise-level strategic services and solutions around application modernization, Agile and DevOps practices, and cloud.

Organizations in 70 countries benefit from its expert IT services that drive teams toward high-performance, empower teams to seamlessly accelerate application delivery, and enable increased efficiency through cross-team collaboration.

Building on our platinum-level partnerships with companies such as Atlassian, CloudBees and AWS, and with our strong and dedicated team of trained and certified technical consultants, enterprises can leverage the right mix of consulting-led strategies and technology-led solutions to direct their IT initiatives toward business growth.