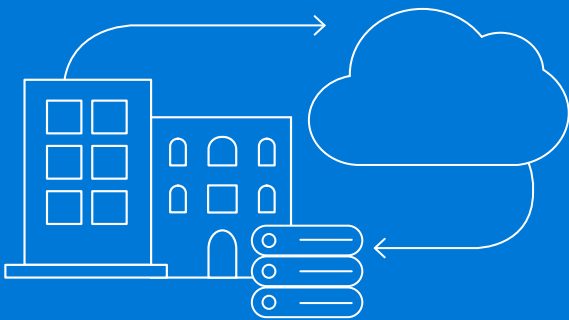


# Maximizing productivity through unified development

Development in hybrid cloud requires simple and flexible infrastructure



---

Organizations encounter a variety of reasons to build and deploy an application on-premises versus in the cloud, and vice versa. Many factors, such as regulatory requirements, the sensitivity of the data, and whether cloud services support the level of customization needed may play into a business's decision to adopt one approach over another. It is possible to develop in both environments in a flexible way, as the following strategies outline.

---

## Not all clouds are alike—navigating the nuances of hybrid models

A significant challenge in adopting a hybrid cloud application model, with software running in the cloud and on-premises, is that the cloud development platform—and a company's approach to that platform—differs from traditional on-premises approaches. While similarities exist—virtual machines still host Windows Server and Linux, for example—many of the issues are unique to the hybrid cloud and may require different developer skill sets, impeding the adoption and use of DevOps processes.

You can streamline development productivity by pursuing an infrastructure that is both clear and flexible:

1. Look for a single, flexible toolset for simplicity
2. Deploy software to containers for maximum flexibility
3. Adopt DevOps processes and tools that work whether an application is being developed on-premises or in the cloud
4. Provide integrated solutions across edge and cloud

Read more about getting the most out of your development efforts in a hybrid environment.



### 1. Look for a single, flexible toolset for simplicity

Companies should focus on creating a single set of tools for developers to maximize developer productivity with a common approach. By using tools that have the flexibility to develop and deploy applications in the cloud or on-premises, your company can meet any regulatory or policy requirements without changing the code or maintaining parallel code bases.



### 2. Deploy software to containers for maximum flexibility

Containers and microservices can be combined with platform as a service (PaaS) and serverless computing to allow developers to create applications and then choose where to deploy them. Such infrastructure will allow your company to develop common DevOps processes, train developers on specific skill sets, and maintain common tools across on-premises and cloud infrastructure.



### 3. Adopt DevOps processes and tools that work whether an application is being developed on-premises or in the cloud

Differences between a cloud platform and your on-premises environment can make it hard to create a common DevOps pipeline for applications deployed in both places. Look for tools and processes that allow your team to implement consistent DevOps mechanisms across your hybrid cloud.



### 4. Provide integrated solutions across edge and cloud

Edge scenarios are important for many businesses. For example, in manufacturing, a real-time control application might need to be located very close to the robots it controls, making even minor cloud latencies problematic. In other cases, a continuous connection to the cloud may not be practical. In these situations, organizations still want to create applications using modern platform technologies. They also might want to create applications that can run in the cloud if needed. Avoid locking these applications into an on-premises environment if it's not required.

#### Find solutions for unifying development in a hybrid cloud environment

##### Azure Cloud Services →

- [Azure Stack](#) extends on-premises applications
- [Azure Resource Manager](#) organizes components of the software development process
- [Azure Portal](#) maximizes collaboration

##### Visual Studio →

- The CI/CD pipeline works well with open-source tools
- Simple integration with [Visual Studio Team Services](#)