# Building microservices on Azure

Microservices are a popular architectural style for building applications that are resilient, highly scalable, independently deployable, and able to evolve quickly. But a successful microservices architecture requires a different approach to designing and building applications.

**What are microservices?**
How do microservices differ from other architectures, and when should you use them?

**Microservices architecture style**
High-level overview of the microservices architecture style

## Examples of microservices architectures

**Use Service Fabric to decompose monolithic applications**
An iterative approach to decomposing an ASP.NET web site into microservices.

**Scalable order processing on Azure**
Order processing using a functional programming model implemented via microservices.

## Build a microservices application

**Use domain analysis to model microservices**
To avoid some common pitfalls when designing microservices, use domain analysis to define your microservice boundaries.

### Reference architecture for Azure Kubernetes Services (AKS)

This reference architecture shows a basic AKS configuration that can be the starting point for most deployments.

### Reference architecture for Azure Service Fabric

This reference architecture shows recommended configuration that can be the starting point for most deployments.

### Design a microservices architecture

These articles dive deep into how to build a microservices application, based on a reference implementation that uses Azure Kubernetes Services (AKS).

### Design patterns

A set of useful design patterns for microservices.

## Operate microservices in production

### Logging and monitoring

The distributed nature of microservices architectures makes logging and monitoring especially critical.

### Continuous integration and deployment

Continuous integration and continuous delivery (CI/CD) are key to achieving success with microservices.