

Azure for Developers

Audience Course Azure for Developers

The Course Azure for Developers is intended for developers who want to use Microsoft Azure to create and deploy cloud applications.

Prerequisites Course Azure for Developers

Experience in a modern programming language such as C#, Java, Python or PHP is required to participate in this course.

Realization Training Azure for Developers

The theory is discussed on the basis of presentation slides and demos. During the course theory and exercises are interchanged.

Certification Course Azure for Developers

The participants will receive a certificate Azure for Developers after successful completion of the course.

Duration: 4 days

Price: € 2650

[Open Schedule](#)



Azure Cloud Development



Content Azure for Developers

In the course Azure for Developers from SpiralTrain participants learn to use Microsoft Azure for the development of Cloud Applications and Services. Microsoft's Azure cloud computing platform allows you to create and manage applications in the cloud. The applications are then hosted in a network of Microsoft data centers and it is no longer necessary to purchase and manage a local server.

Azure Intro

The course Azure for Developers starts with an overview of the services and Availability Zones that the Azure Cloud Platform has to offer. The difference between Azure PAAS (Platform as a Service) and Azure IAAS (Infrastructure as a Service) is also discussed.

App Services

Next App Services and how Azure can be used to host Applications and in particular Web Applications will be treated. Attention is also paid to the support of various Development Environments such as for ASP.NET, Node JS and Python.

Data Access

The various forms of storage that Azure has to offer are also on the program of the course. Blob Storage and File Storage are discussed. And also accessing relational and NoSQL databases from Azure is covered.

VMs and Containers

Azure also supports deployment of Virtual Machines that can be provisioned entirely as desired. Both Windows and Linux VM's are possible. And containers based on Docker Images are also supported in Azure.

Azure Functions

Next attention is paid to how Azure supports serverless computing with Azure Functions. Serverless computing does use servers, but they run invisibly in the cloud. Functions scenarios, function pipelines and function chaining are treated.

Virtual Networks

Finally attention is paid to the Azure Network Service and setting up Virtual Networks. Inbound and outbound rules are discussed and load balancing and port forwarding are covered.

Modules Azure for Developers

Module 1 : Azure Intro	Module 2 : App Services	Module 3 : Azure Data Access
What is Azure? Azure Services Compute Services Data Services Application Services Network Services Azure CLI Regions Availability Zones Data Centers Fabric Controllers Azure PAAS Azure IAAS	App Service Plan Hosting Applications Web Applications API Apps Virtual Machines Security Monitoring Development Environments ASP.NET, Node JS Python, PHP Deploying in Visual Studio App Service Pricing Backup	Azure Storage Blob Storage Files and Tables Storage Accounts Storage Replication Azure Database Service SQL Databases Managed Instances Single Database Elastic Pool Database Configuration NoSQL Data Stores Azure Data Factory
Module 4 : VM's and Containers	Module 5 : Azure Functions	Module 6 : Azure Virtual Networks
Windows VM's Linux VM's VM Choices VM Sizes VM Configuration VM Storage Availability Set Scaling Containers on Azure Docker Images Running Containers Azure Service Fabric Container Scaling	Serverless Computing Creating Functions Deploying Functions Language Support Function Scenario's HTTP Triggers Process File Uploads Run Scheduled Tasks Function Pipelines Function Pricing Application Patterns Fan In Fan out Function Chaining	Azure Network Service Virtual Networks Creating Subnets Security Groups Inbound Rules Outbound Rules Azure Network Interface IP Configuration Hostname Resolution Application Gateway Load Balancer Port Forwarding Load Balancing Rules