


# Deploy a basic workload in Azure

12/31/2018 • 3 minutes to read • Contributors 

## In this article

[Basic workload](#)

[Basic web application \(PaaS\)](#)

[Single Windows or Linux VM \(IaaS\)](#)

[Next steps](#)

The term *workload* is typically defined as an arbitrary unit of functionality, such as an application or service. It helps to think about a workload in terms of the code artifacts that are deployed to a server, and also other services specific to an application. This may be a useful definition for an on-premises application or service, but for cloud applications it needs to be expanded.

In the cloud a workload not only encompasses all the artifacts, but it also includes the cloud resources as well. Included is cloud resources as part of the definition because of the concept known as "infrastructure as code". As you learned in [how does Azure work?](#), resources in Azure are deployed by an orchestrator service. This orchestrator service exposes functionality through a web API, and you can call the web API using several tools such as PowerShell, the Azure CLI, and the Azure portal. This means that you can specify Azure resources in a machine-readable file that can be stored along with the code artifacts associated with the application.

This enables you to define a workload in terms of code artifacts and the necessary cloud resources, thus further enabling you to isolate workloads. You can isolate workloads by the way resources are organized, by network topology, or by other attributes. The goal of workload isolation is to associate a workload's specific resources to a team, so that the team can independently manage all aspects of those resources. This enables multiple teams to share resource management services in Azure while preventing the unintentional deletion or modification of each other's resources.

This isolation also enables another concept, known as DevOps. DevOps includes the software development practices that include both software development and IT operations above, and adds the use of automation as much as possible. One of the principles of DevOps is known as continuous integration and continuous delivery (CI/CD). Continuous integration refers to the automated build processes that are run every time a developer commits a code change. Continuous delivery refers to the automated processes that deploy this code to various environments such as a development environment for testing or a production environment for final deployment.

## Basic workload

A *basic workload* is typically defined as a single web application or a virtual network (VNet) with virtual machine (VM).

### ⓘ Note

This guide does not cover application development. For more information about developing applications on Azure, see the [Azure Application Architecture Guide](#).

Regardless of whether the workload is a web application or a VM, each of these deployments requires a *resource group*. A user with permission to create a resource group must do this before following the steps below.

## Basic web application (PaaS)

For a basic web application, select one of the 5-minute quickstarts from the [web apps documentation](#) and follow the steps.

#### ⓘ Note

Some of the Quickstart guides will deploy a resource group by default. In this case, it's not necessary to create a resource group explicitly. Otherwise, deploy the web application to the resource group created above.

Once you deploy a simple workload, you can learn more about the proven practices for deploying a [basic web application](#) to Azure.

## Single Windows or Linux VM (IaaS)

For a simple workload that runs on a VM, the first step is to deploy a virtual network. All infrastructure as a service (IaaS) resources in Azure such as virtual machines, load balancers, and gateways, require a virtual network. Learn about [Azure virtual networks](#), and then follow the steps to [deploy a Virtual Network to Azure using the portal](#). When you specify the settings for the virtual network in the Azure portal, be sure to specify the name of the resource group created above.

The next step is to decide whether to deploy a single Windows or Linux VM. For Windows VM, follow the steps to [deploy a Windows VM to Azure with the portal](#). Again, when you specify the settings for the virtual machine in the Azure portal, specify the name of the resource group created above.

Once you've followed the steps and deployed the VM, you can learn about [proven practices for running a Windows VM on Azure](#). For a Linux VM, follow the steps to [deploy a Linux VM to Azure with the portal](#). You can also learn more about [proven practices for running a Linux VM on Azure](#).

## Next steps

See [Architectural decision guides](#) for how to use core infrastructure components in the Azure cloud.