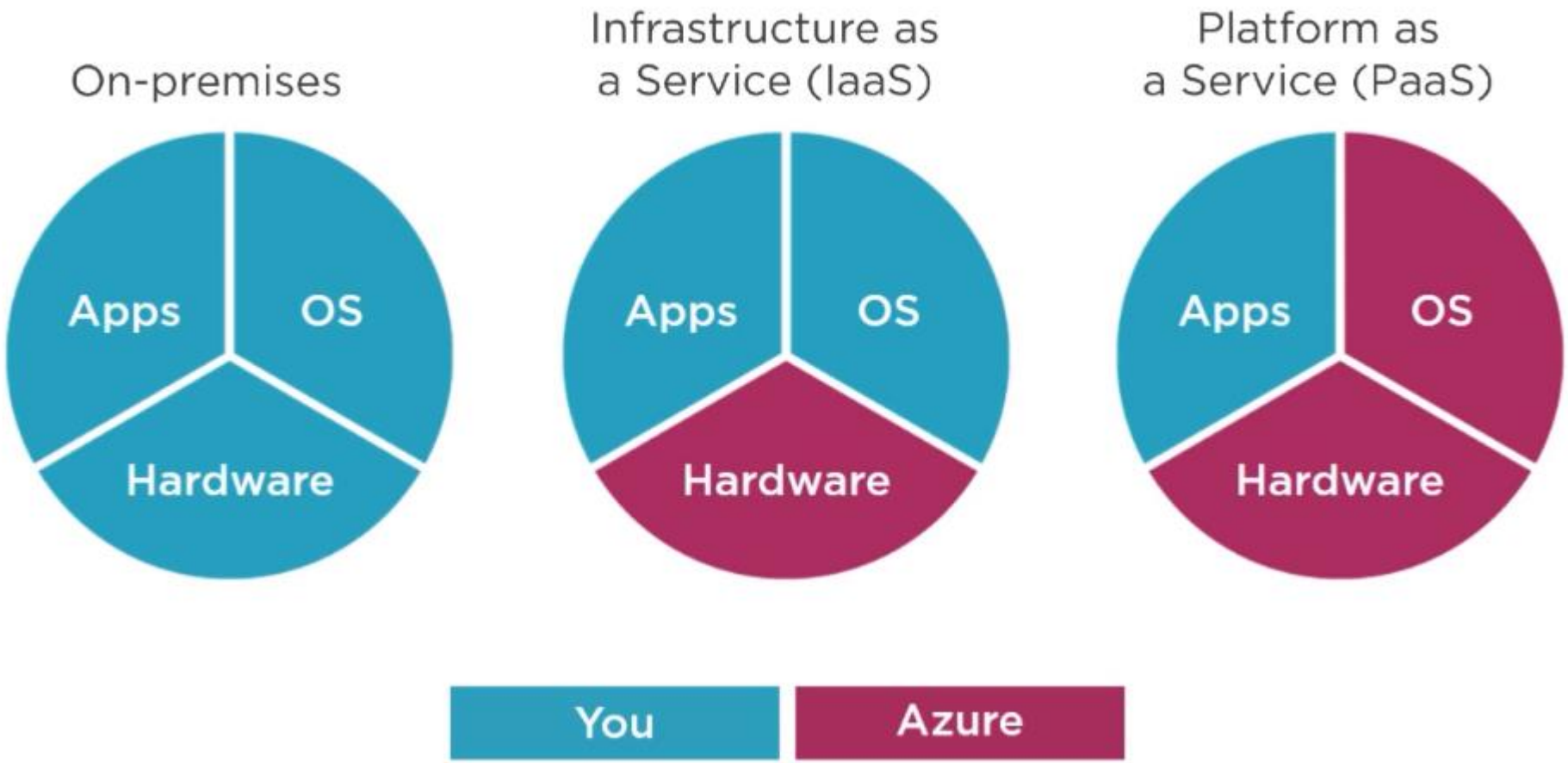
Azure App Service Architecture


On-premises
Apps
OS
Hardware
Infrastructure as
a Service (IaaS)
Apps
OS
Hardware
Platform as
a Service (PaaS)
Apps
OS
Hardware
You
Azure

Azure Apps Services

App Service Web **Apps** is a fully managed compute platform that is optimized for hosting websites and web applications. This platform-as-a-**service** (PaaS) offering of Microsoft **Azure** lets you focus on your business logic while **Azure** takes care of the infrastructure to run and scale your **apps.**

Web Apps is a fully managed compute platform that is optimized for hosting **websites and web applications.**
Mobile App: Use the **Mobile Apps** feature of **Azure App** Service to rapidly build engaging cross-platform and native apps for iOS, Android, Windows or Mac; store app data in the cloud or on-premises; authenticate customers; send push notifications; or add your custom back-end logic in C# or Node.js.
Logic App: Logic Apps are managed service that provides the Lego blocks of Azure server less architecture that you can use to create, develop and deploy cloud-based integrations and workflows.

Web Apps not only adds the power of Microsoft Azure to your application, such as security, load balancing, autoscaling, and automated management. You can also take advantage of its DevOps capabilities, such as continuous deployment from VSTS, GitHub, Docker Hub, and other sources, package management, staging environments, custom domain, and SSL certificates.

Why use Web Apps?

Here are some key features of App Service Web Apps:

- **Multiple languages and frameworks** - Web Apps has first-class support for ASP.NET, ASP.NET Core, Java, Ruby, Node.js, PHP, or Python. You can also run PowerShell and other scripts or executables as background services.
- **DevOps optimization** - Set up continuous integration and deployment with Visual Studio Team Services, GitHub, BitBucket, Docker Hub, or Azure Container Registry. Promote updates through test and staging environments. Manage your apps in Web Apps by using Azure PowerShell or the cross-platform command-line interface (CLI).
- **Global scale with high availability** - Scale up or out manually or automatically. Host your apps anywhere in Microsoft's global datacenter infrastructure, and the App Service SLA promises high availability.
- **Connections to SaaS platforms and on-premises data** - Choose from more than 50 connectors for enterprise systems (such as SAP), SaaS services (such as Salesforce), and internet services (such as Facebook). Access on-premises data using Hybrid Connections and Azure Virtual Networks.
- **Security and compliance** - App Service is ISO, SOC, and PCI compliant. Authenticate users with Azure Active Directory or with social login (Google, Facebook, Twitter, and Microsoft). Create IP address restrictions and manage service identities.
- **Application templates** - Choose from an extensive list of application templates in the Azure Marketplace, such as WordPress, Joomla, and Drupal.
- **Visual Studio integration** - Dedicated tools in Visual Studio streamline the work of creating, deploying, and debugging.
- **API and mobile features** - Web Apps provides turn-key CORS support for RESTful API scenarios, and simplifies mobile app scenarios by enabling authentication, offline data sync, push notifications, and more.
- **Serverless code** - Run a code snippet or script on-demand without having to explicitly provision or manage infrastructure, and pay only for the compute time your code actually uses.

Azure App Service plan overview

In App Service, an app runs in an *App Service plan*. An App Service plan defines a set of compute resources for a web app to run. These compute resources are analogous to the server farm in conventional web hosting. One or more apps can be configured to run on the same computing resources (or in the same App Service plan).
When you create an App Service plan in a certain region (for example, West Europe), a set of compute resources is created for that plan in that region. Whatever apps you put into this App Service plan run on these compute resources as defined by your App Service plan. Each App Service plan defines:

- **Region (West US, East US, etc.)**
- **Number of VM instances**
- **Size of VM instances (Small, Medium, Large)**
- **Pricing tier (Free, Shared, Basic, Standard, Premium, PremiumV2, Isolated, Consumption)**

The *pricing tier* of an App Service plan determines what App Service features you get and how much you pay for the plan. There are a few categories of pricing tiers:

- **Shared compute**: **Free** and **Shared**, the two base tiers, runs an app on the same Azure VM as other App Service apps, including apps of other customers. These tiers allocate CPU quotas to each app that runs on the shared resources, and the resources cannot scale out.
- **Dedicated compute**: The **Basic**, **Standard**, **Premium**, and **PremiumV2** tiers run apps on dedicated Azure VMs. Only apps in the same App Service plan share the same compute resources. The higher the tier, the more VM instances are available to you for scale-out.
- **Isolated**: This tier runs dedicated Azure VMs on dedicated Azure Virtual Networks, which provides network isolation on top of compute isolation to your apps. It provides the maximum scale-out capabilities.

App Service plans

	FREE	SHARED	BASIC	STANDARD	PREMIUM	ISOLATED *
– Limits **						
Apps	10	100	Unlimited	Unlimited	Unlimited	Unlimited
Disk space	1 GB	1 GB	10 GB	50 GB	250 GB	1 TB
Max instances			Up to 3	Up to 10	Up to 20	Up to 100
SLA			99.95%	99.95%	99.95%	99.95%
Functions on App Service Plans *			✔	✔	✔	✔

App Service

=

WEB APPS

Web apps that scale with your business

MOBILE APPS

Build Mobile apps for any device

LOGIC APPS

Automate business process across SaaS and on-premises

API APPS

Easily build and consume APIs in the cloud

WEB APPS

Web apps run as-is
no changes required

Full capability set available including:

- .NET, Node.js, Java, PHP, and Python
- Web Jobs for long running tasks
- Integrated VS publish, remote debug…
- CI with GitHub, Bitbucket, VSO
- Auto-load balance, Auto scale, Geo DR
- Virtual networking and hybrid connections
- Site slots for staged deployments

MOBILE APPS

Mobile services plus
a whole lot more

New capabilities for Mobile apps:

- Webjobs for long running tasks
- CI with GitHub, BitBucket, VSO
- Auto-load balance, Autoscale, Geo DR
- Virtual networking and hybrid connections
- Site slots for staged deployments

LOGIC APPS

Automate SaaS and on-premises systems

New Logic Apps for easy automation

- No code designer for rapid creation
- Dozens of pre-built templates to get started
- Out of box support for popular SaaS and on-premises apps
- Use with custom API apps of your own
- Biztalk APIs for expert integration scenarios

API APPS

Create, consume and host APIs more easily

Easily use cloud or custom APIs:

- Dozens of built-in APIs for popular SaaS
- An ecosystem of APIs for any need
- Create and publish custom, reusable APIs
- Visual Studio tooling with one click publish and remote debugging
- Automatic client SDK generation for many languages

WEB APPS

Customer site

Service Advisor

LOGIC APPS

Customer Appointment

MOBILE APPS

Technician App

API APPS

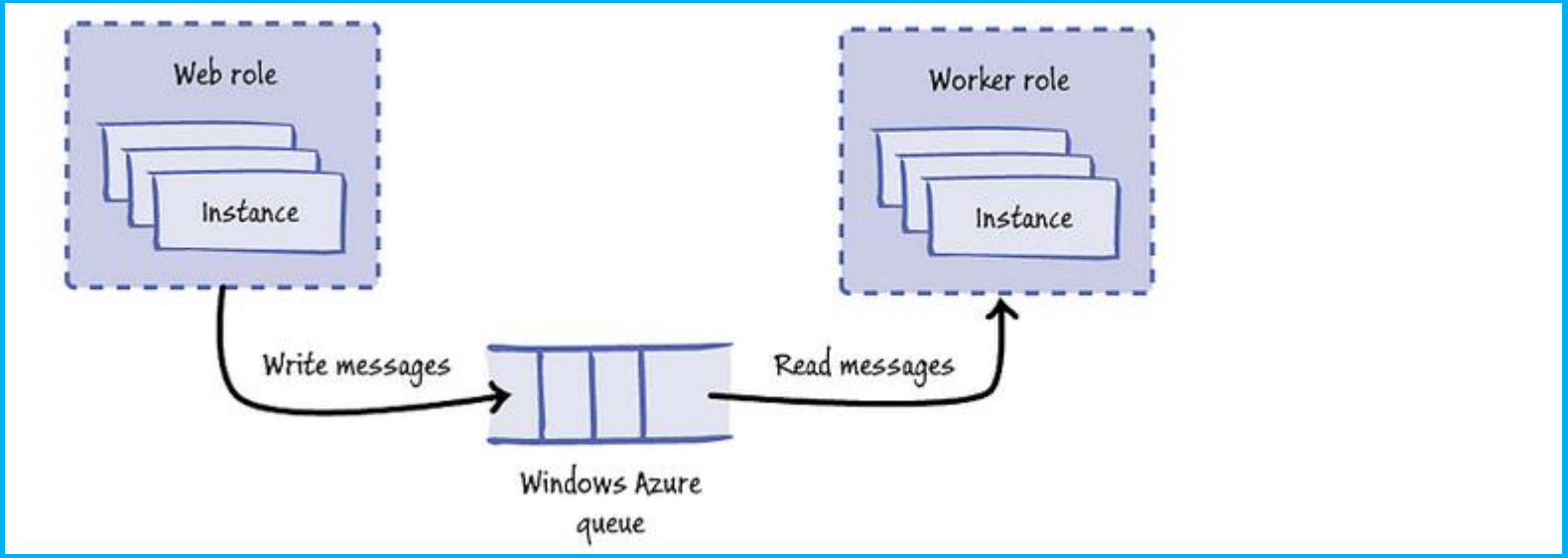
Web Role and Worker Role

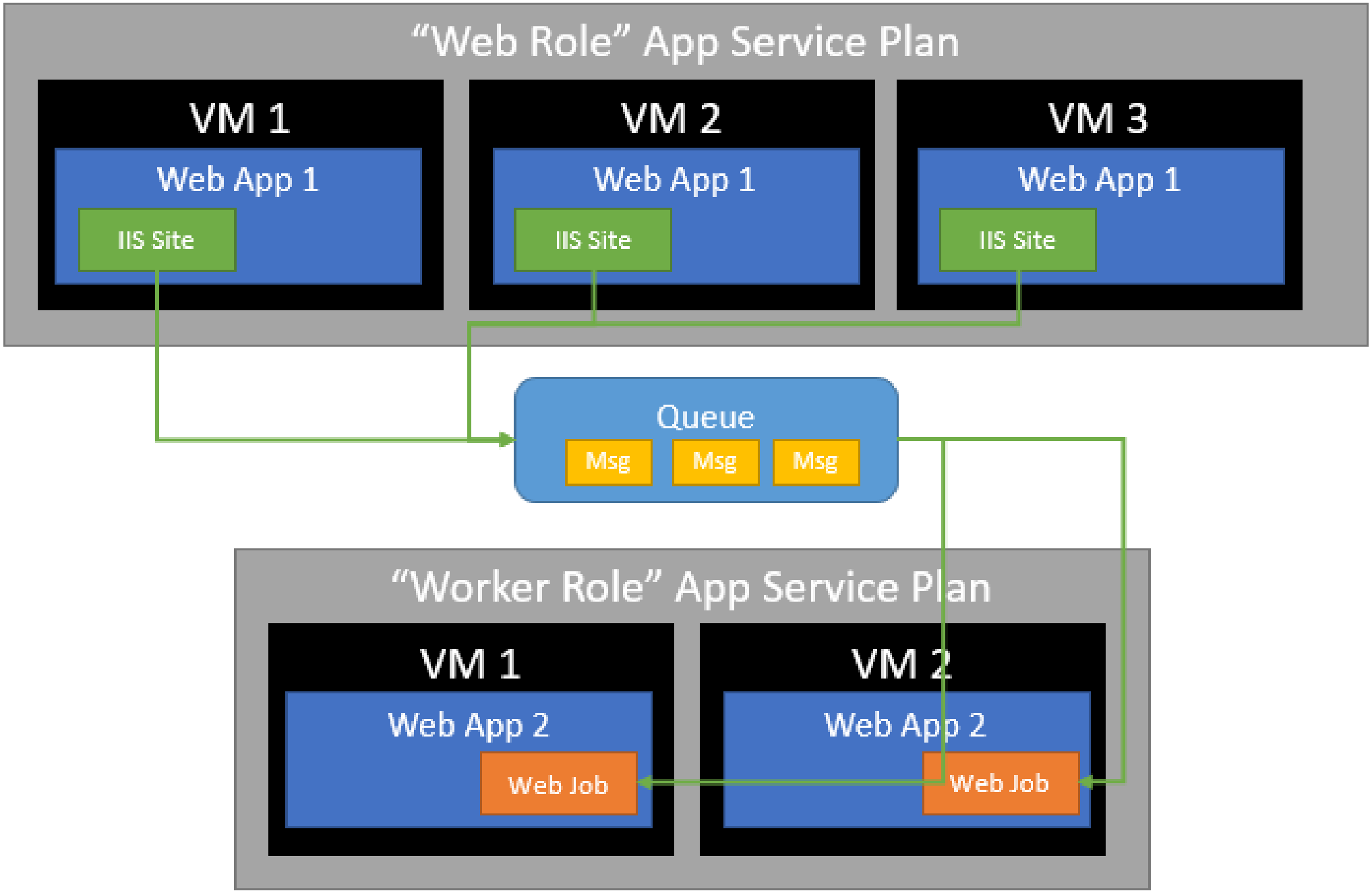
The Azure App Service is a great container for a wide range of application types, but often I see people overlook this option because they want to use the “web and worker role” pattern made famous by Cloud Services. This pattern is shown in the following diagram and involves:

•A **Web Role** (consisting of 1 or more instances) that hosts a website or an API and is accessed by the application’s users. The web role can perform simple tasks synchronously, but whenever any complex processing is required it will create a message and drop it on a Queue.

•One or more **Queues**, which support asynchronous communication between the Web Role and the Worker Role.

•A **Worker Role** (consisting of 1 or more instances) which pull messages of the Queue and perform slow or complex processing tasks.



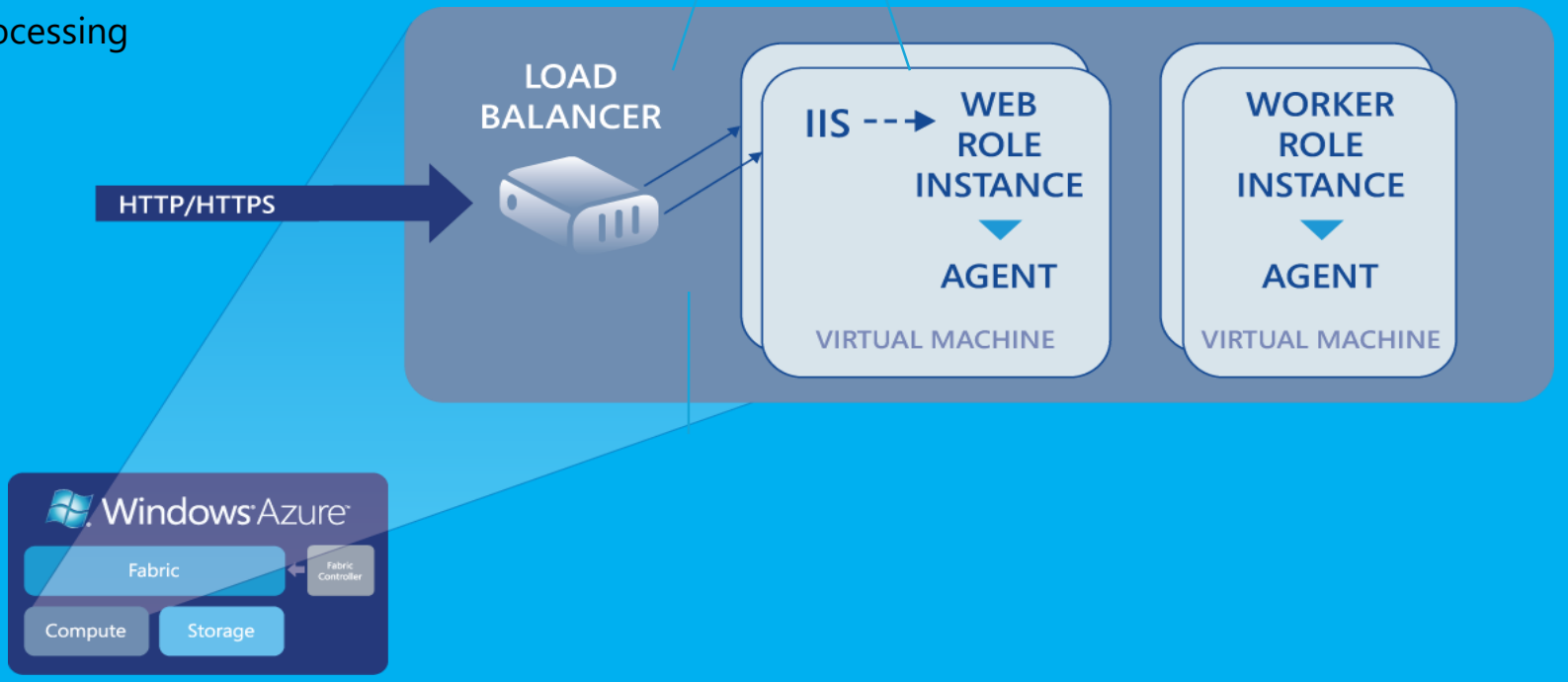

"Web Role" App Service Plan
VM 1
Web App 1
IIS Site
VM 2
Web App 1
IIS Site
VM 3
Web App 1
IIS Site
Queue
Msg
Msg
Msg
"Worker Role" App Service Plan
VM 1
Web App 2
Web Job
VM 2
Web App 2
Web Job

Compute Service in Microsoft Azure
Goal: Massive Scalability

Scale out – not up – by replicating worker instances as needed.

Allow applications to scale user and compute processing independently.

Two role types: Web Role & Worker Role
Windows Azure applications are built with web roles, worker roles, or a combination of both deployed to a number of instances.



Each instance runs on its own VM (virtual machine), replicated as needed

Defining Windows Azure Roles

- ✓ Build services and solutions with any combination of Web and Worker Roles
- ✓ Communicate via HTTP/HTTPS, TCP/IP, and ports other than 80 and 443
- ✓ Develop with Microsoft and non-Microsoft tools:
  ASP.NET, WCF, other .NET tools, Java, Python, Ruby, etc.

WEB ROLE

- ✓ Interacts with end-user or web services
- ✓ Communicate with worker roles directly or via queues

WORKER ROLE

- ✓ Initiates their own requests for data or tasks from the queue
- ✓ Similar to a "batch job" or Windows service