ADC
Applaudo
Developers
Conference
2020

Powered by ApplaudoStudios

Rafael Paz

Technologies

Microsoft CERTIFIED AZURE DEVELOPER ASSOCIATE

Microsoft CERTIFIED AZURE SOLUTIONS ARCHITECT EXPERT

@rafapaz09

@rafapaz09

@rafapaz05

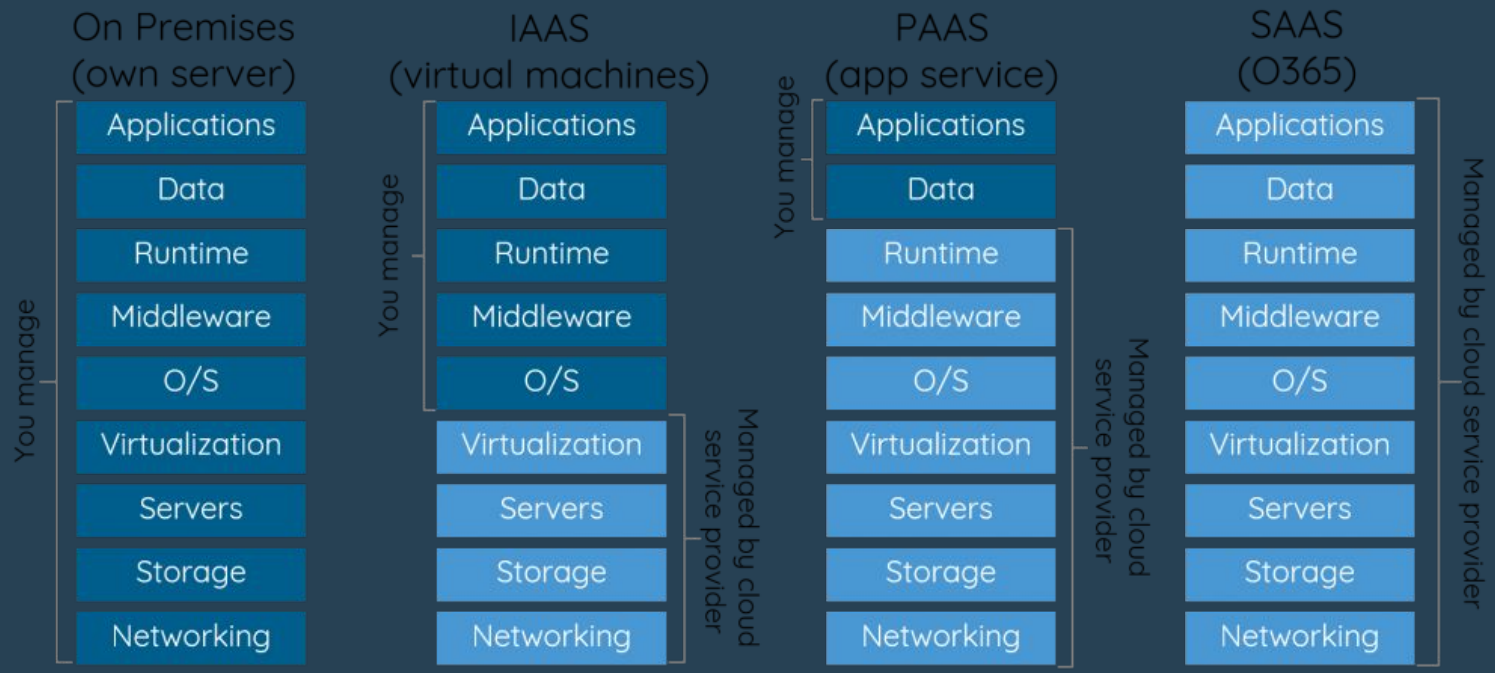
Azure PaaS Services

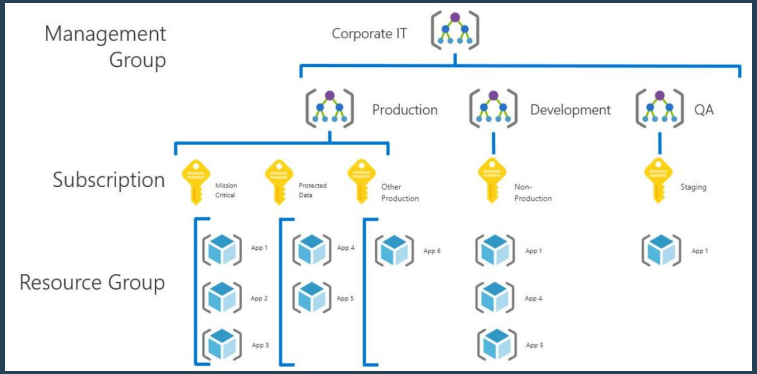
Discover what we can do with Azure to make our development process easier and faster

Agenda

1. Basic Cloud Concepts
2. Azure Hierarchy
3. Azure PaaS Services
4. Azure Architectures

Type of Services

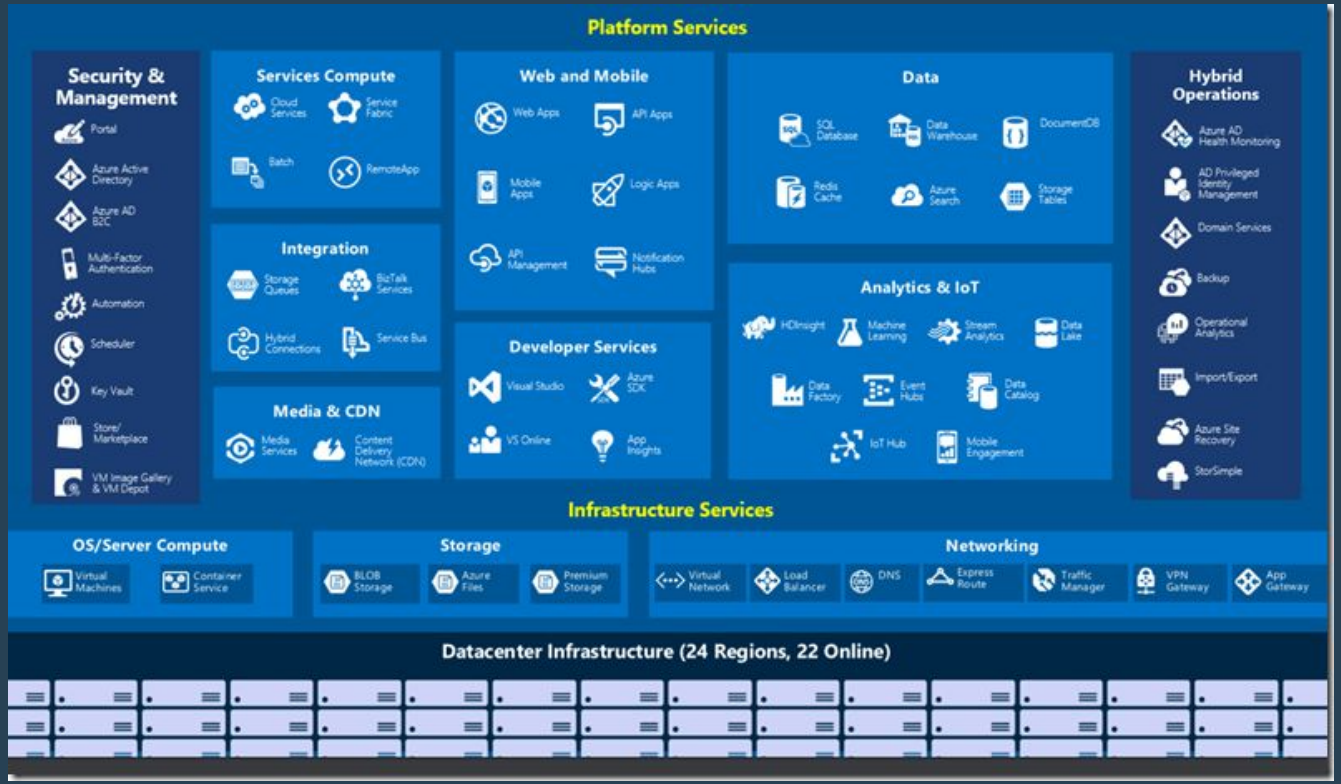




Azure Hierarchy

Management Groups
Subscriptions
Resource Groups

Azure PaaS Services



Data

Azure Cosmos DB

Microsoft's globally distributed, multi-model database service, with one click of a button, enables us to elastically and independently scale throughput and storage.

Azure Redis Cache

In-memory data stored based on the open-source software Redis.
Improves performance and scalability of an application.
Process large volumes of application request by keeping frequently accessed data in server memory.

Azure SQL

PaaS database engine that handles db management functions like patching, backup, upgrading and monitoring without user interaction.

Web And Mobile

Web Apps

Host web applications

App that runs and scales on Windows or Linux.

Logic Apps

Schedule, automate and orchestrate tasks.

Simplified design to create workflows.

Azure Functions

Run small pieces of code without worrying about application infrastructure.

Function is triggered by a specific event type.

API Management

Create API gateways for existing back-end services.

Create policies for the back-end to avoid throttling.

Integrations & CDN

Storage Queues

Store large numbers of messages

Create work to process asynchronously

Service Bus

Enterprise integration message broker

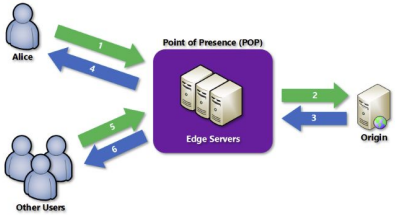
Decouple apps and services

Content Delivery Network

Efficiently deliver web content to users, stores cached content on edge servers in point-of-presence locations that are close to end users, to minimize latency.

Better performance for apps and services

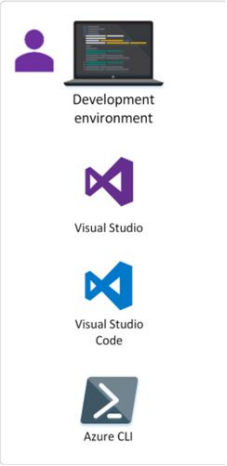
Highly scalable



Azure Architectures

Architectures examples by using different Azure PaaS services

DEVELOPER SERVICES - SECURITY



1

Local env with VS, Azure CLI

2

App sends auth req during debug, AAD returns a token

Azure AD

App Configuration

3

App config SDK sends a req with the token

4

The secrets are read by using the token

Key Vault

SECURITY - PRODUCTION

1

Azure Env (App Service, Azure Function, etc)

2

App sends auth req during debug, AAD returns a token

3

App config SDK sends a req with the token

4

The secrets are read by using the token

MSI Auth

Retrieving a KeyVault secret by using MSI Authentication from Azure.

```
// Creating Azure Service Token Provider
var serviceTokenProvider = new AzureServiceTokenProvider();

//Connecting to keyvault using MSI
var KeyVaultClient = new KeyVaultClient(new
KeyVaultClient.AuthenticationCallback(serviceTokenProvider.KeyVaultTo
kenCallback);

//Retrieving secret from keyVault
return await KeyVaultClient.GetSecretAsync(secretUri);
```

MSI Authentication Demo

Demo about using MSI auth for accessing a KeyVault and retrieving secrets
Git Repo URL: https://github.com/rafapaz09/Azure-MSI-KeyVault

Best Practices

Use MSI whenever is possible

Whenever is possible use MSI to connect to the Azure Services, this is going to allow access to the resources to only the services that you have granted access to.

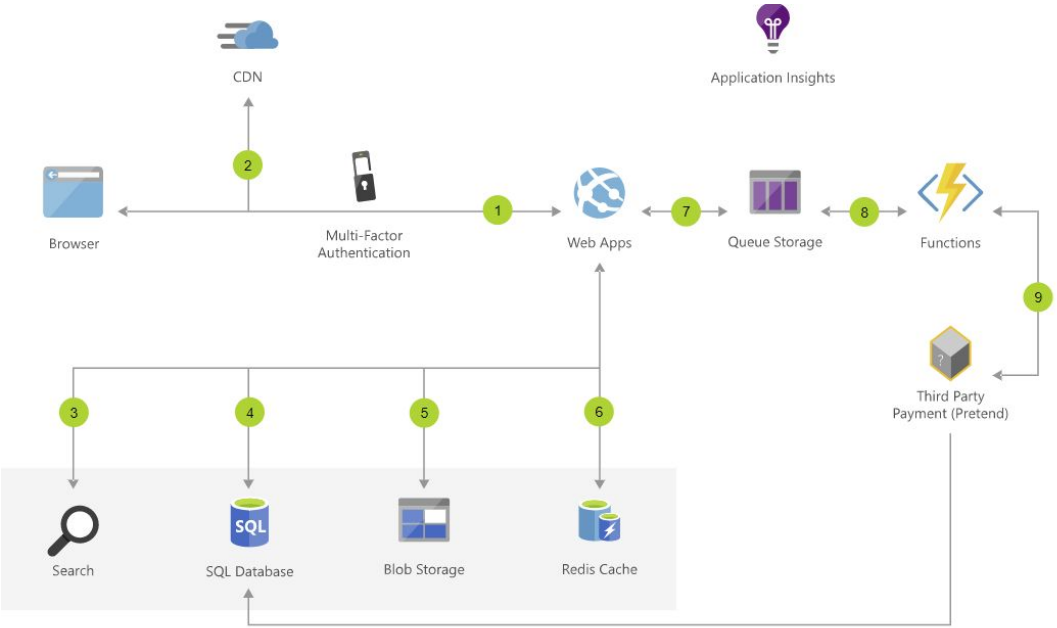
Use App Settings

Always avoid hardcoded values within your code and your repo, try to use Azure Application Settings from any information that you app needs, this will give you more flexibility if any value changes and when moving your application from one dev to prod.

Keep confidential info always within a KeyVault

Always keep application secrets, keys and certificates within a KeyVault to avoid any leak of information, KeyVault gives us more control regarding application access, user access and by turning on the firewall rules to a KeyVault, only the connection from certain Ips' will be allowed.

E-Commerce App

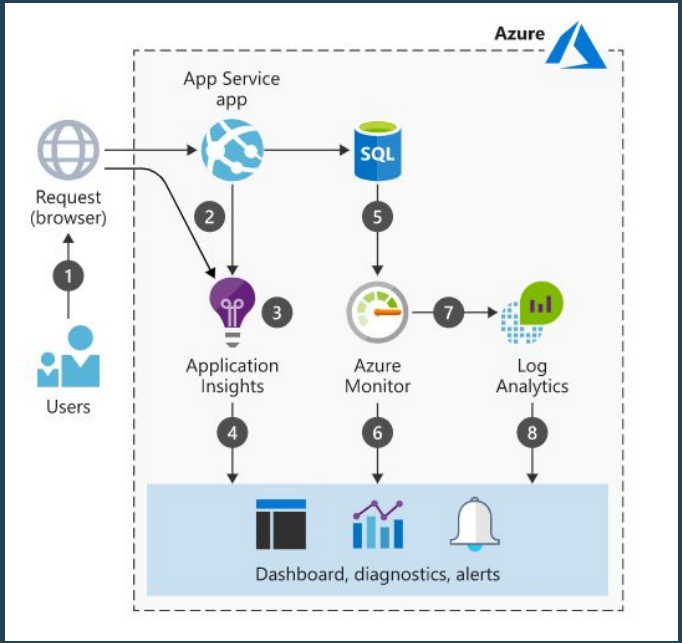


Pulls static resources from CDN → Products on Azure Search, and Azure Sql → Page output is cached in Redis → Submit order, app insert a message in a queue → Function reads message and do process

Storage Queue

Connecting an Azure Function with a Storage Queue

```
public static class Payment
{
    [FunctionName("PaymentFunction")]
    public static void Run
    (
        [QueueTrigger("payment",
        Connection = "AzureWebJobsStorage")]
        string myQueueItem,
        ILogger log
    )
    {
        log.LogInformation($"Queue trigger function");
        //Do the function work
        ..............
    }
}
```



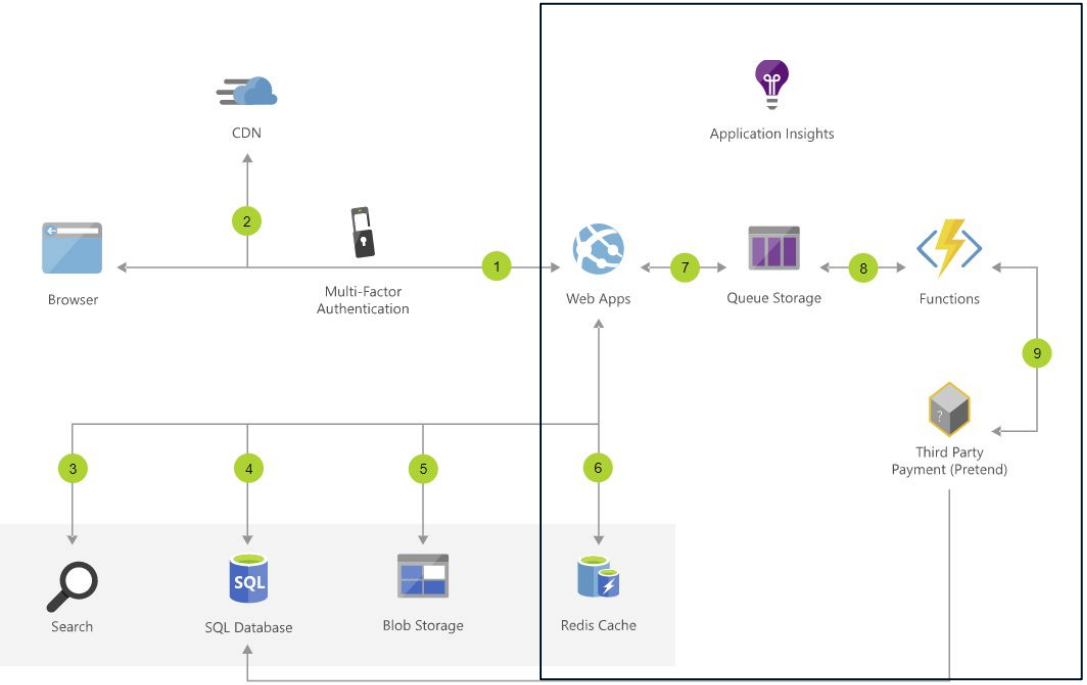
App Service Monitoring

- User interacts with the application
- App Service emits telemetry
- Application Insights collects performance and usage data from the application
- Azure Monitor collects and analyzes metrics and quotas
- Log Analytics collects log metrics
- Devs and admins can review health, performance and usage info

E-Commerce Demo

Demo about using Azure Redis Cache with a WebApp and Azure Function with Storage Queues
Git Repo URL: https://github.com/rafapaz09/E-Commerce

E-Commerce App Demo



Best Practices

Create autoscale rules and notification alerts

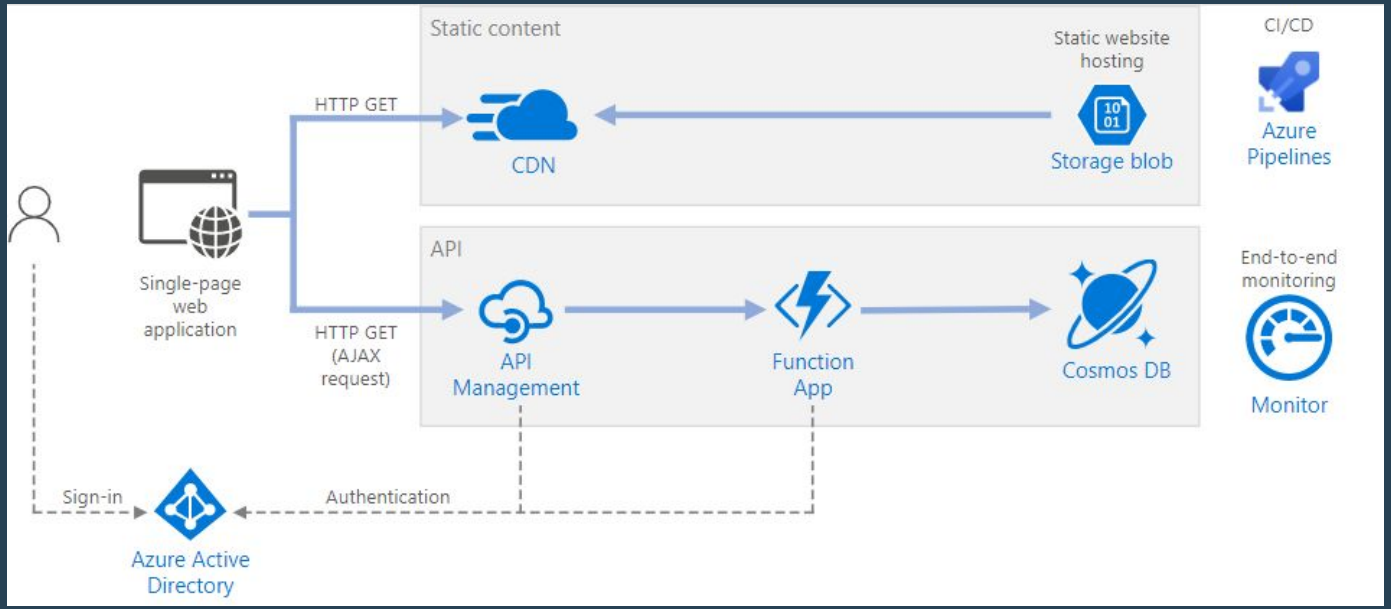
One of the advantages of using PaaS services in the cloud, is that allow us to create autoscale rules for the application, so whenever there is a spike in the traffic the app can easily scale out or scale up based on different rules. Always keep in mind that for each scale out rule there must exist a scale in rule.

Use Application Insights

Connect the application with Application Insights to visualize usage patterns and where the code is running slow or even any type of internal error.

Use Deployment Slots

Deployment slots for web applications are useful when we want to test any changes to our application before making it go live, it allows to have a pre-prod environment where we can easily test any new feature.



Serverless Architecture

Compute resources are allocated dynamically as needed by the platform
The compute resources scale on demand based on traffic, without the developer needing to do any configuration.

Rafael Paz

@rafapaz09

@rafapaz09

@rafapaz05

Applaudo
Developers
Conference
2020

Powered by ApplaudoStudios