



# Learn Azure Step by Step

Every Saturday and Sunday

India time 9 PM, 1 hour

[www.questpond.com](http://www.questpond.com)

Why Cloud ?	<b>Why Cloud and Pay as you go model?</b> Understanding terms <b>High Availability, Scalability, Elasticity, Agility, Fault Tolerance, and Disaster Recovery</b> <b>IAAS , PAAS and SAAS</b> Private , Public and Hybrid cloud <b>Capex and Opex</b> <b>Consumption based model</b>	10 min
Opening an Azure Account	<b>Open and registering Azure account 13000 / 220\$</b>	10 min
Some basic Azure concepts	<b>Geography, Region, Zones and Availability Zones</b> <b>Resource</b> and Resource groups , Azure resource manager.	10 Min
Azure virtual machines	<b>Creating First Azure virtual Machine and understanding</b>	30 min
SQL Server	<b>Azure SQL Server</b> <b>DTU , EDTU</b>	1 hour
Azure Web APP	<b>Hosting your first ASP.NET core on Azure</b>	30 min
Azure Storage ( Intro)	<b>Blobs, Queues , tables and Files.</b> <b>(Tied up with Azure)</b>	1 hour
Functions and Logic Apps(Intro)	<b>Azure functions and Logic Apps</b>	
Azure tables	Create an Azure table C# Insert update , Query and delete 10 best practices. To the point queries , Partion key , row key Duplicate is ok Compund keys Hot partitions EGT Avoid new Azure tables Delete pattern Large entities  Intra partition Inter partition <b>Concurrency ETag why did not work.</b>  Azure table pattern designs	1 hour



Azure Blobs	Types of blobs Append , Page blob and Block blobs Simple blob upload Uploading block blob in chunks Append blob Page blob	
Azure Queues	Creating Azure Queues Send message . Receive message Peeking Messages Dequeue	
Azure Files	By using C# files	
Function App by using visual Studio	Log reading Blob – Function App	
Logic App by using visual studio	Visual logic app Complicated one <a href="https://github.com/MicrosoftDocs/azure-docs/issues/40647">https://github.com/MicrosoftDocs/azure-docs/issues/40647</a>	
Devops	What is Devops ? CI CD CU Azure piple line YAML Checked in , Build start , Release , updated in to Azure APP Service Unit testing inside the pipe line Basic YAML from Azure pipeline	1 hour
Azure networking	IP, Subnet Create a simple VNet NSG...Neywork level Ping from machine ping NSG	
Power Shell programming with Azure	Power Shell and Power Shell ISE CMDlets :- Get-Process Get Set kk-III Get-Service -Name x* Get-Service   Sort-Object -property Status \$MyVariable = 1, 2, 3 Azure PowerShell Az module Azure power Shell module Shw online power Shell  Open an elevated PowerShell prompt and in order to allow the execution of scripts signed by a trusted publisher run  Set-ExecutionPolicy RemoteSigned  Run Install-Module -Name Az -AllowClobber  Connect-AzAccount <a href="https://docs.microsoft.com/en-us/powershell/azure/azureps-vm-tutorial?view=azps-7.1.0">https://docs.microsoft.com/en-us/powershell/azure/azureps-vm-tutorial?view=azps-7.1.0</a>	



	<p>1. How and where to store the PowerShell script in Azure.</p> <p>2. How to schedule the PS script in Azure to do a specific job everyday at a particular time.? \$psversiontable</p>	
Messaging	<p>Azure Queues , Events and Service Bus</p> <pre>// if you want to add a filter var msg = new ServiceBusMessage(\$"Message {i}"); msg.To = "sub1"; if (!messageBatch.TryAddMessage(msg))</pre> <p>in the filter of sub sys.To='Sub1'</p>	1hour
Azure Automate	Run book , Attach schedule with it	
Azure AD , Two factor		
OAuth 2.0 . open ID MVC core	<p><a href="https://docs.microsoft.com/en-us/azure/active-directory/develop/tutorial-v2-asp-webapp">https://docs.microsoft.com/en-us/azure/active-directory/develop/tutorial-v2-asp-webapp</a></p> <p><a href="https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-v2-aspnet-core-webapp">https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-v2-aspnet-core-webapp</a></p> <p>Angular API <a href="https://docs.microsoft.com/en-us/rest/api/servicebus/get-azure-active-directory-token">https://docs.microsoft.com/en-us/rest/api/servicebus/get-azure-active-directory-token</a></p> <p><a href="https://docs.microsoft.com/en-us/graph/auth-v2-service#token-request">https://docs.microsoft.com/en-us/graph/auth-v2-service#token-request</a></p> <p>MVC Contoller</p> <p><b>Web API</b></p> <pre>"AzureAd": {   "Instance": "https://login.microsoftonline.com/",   "Domain": "https://localhost:44360/",   "ClientId": "1eb4d1a2-9673-49e2-8c2f-88fdf8cb3761",   "TenantId": "bf65742e-d37d-46f9-bc1d-24def2b7bc1d",   "CallbackPath": "/signin-oidc" }</pre> <p>Home work..</p>	
ARM	Demo...	
Docker	<p>Docker is a software which implements container</p> <p>Docker , enable docker for windows</p> <p>DockerFile , DockerIgnore</p> <p>Docker build</p> <p>Docker Run</p>	
Kubernetes	You can not install the main of Kuber on windows	
ACI/Aks/ACR		
APIM	<p>What is APIM ?</p> <p>Importing an API Service</p> <p>Understanding the inbound, output bound and backend</p> <p>Talk about products</p>	

	<p>Applying input policies.          Seeing some demos of output policies          Restrict IP Address          Limit rate calls          Removed asp headers  <b>Versioning</b> is for the end users          Revision is for code          APIM Security with OAuth          Revision and Versioning of APIM          Security in Azure APIM OAuth and openid          Error handling</p>	
	<p>For OAuth2.0 <a href="https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-protect-backend-with-aad">https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-protect-backend-with-aad</a>  <b>Step 1 :- Created APIM (1 hour)</b>  <b>Step 2:- Create WebAPI and publishing to App Service</b>  <b>Step 3:- AppService API I will add to APIM</b>  <b>Step 4:- Create Server application register in AAD</b>  <b>Step 5:- Create Client application Server register in AAD</b>  <b>Step 6 :- Create Secret key in the client app</b>  <b>Step 7:- Client App given Access to Server application , Grant permission in AAD</b>  <b>Step 8 :- Attach the Client key , Server Auth , Token..WebAPI</b>  <b>Step 9 :- Inbound Jwt check</b>          Step 10... Final call from developer Client          Using open id connect</p>	
Azure telemetry	<p><a href="https://docs.microsoft.com/en-us/azure/azure-monitor/app/data-model">https://docs.microsoft.com/en-us/azure/azure-monitor/app/data-model</a>          Azure Application Insights sends telemetry from your web application to the Azure portal, so that you can analyze the performance and usage of your application.</p> <p>Metric , Log , Event          Alert</p> <p>Azure telemetry          Application insight          Logging , Trace , Event Metrics , Alert, Telemetry          Trace - used either directly, or through an adapter to implement diagnostics logging using an instrumentation framework that is familiar to you, such as Log4Net or System.Diagnostics.          Event - typically used to capture user interaction with your service, to analyze usage patterns.          Metric - used to report periodic scalar measurements.</p> <p>Log text was not seen..          Dependency telemetry  <a href="https://squaredup.com/blog/kusto-101-a-jumpstart-guide-to-kgql/">https://squaredup.com/blog/kusto-101-a-jumpstart-guide-to-kgql/</a>           TableName</p>	



	<p>AppServiceAppLogs   where ResultDescription contains "Application"</p> <p>AppServiceAppLogs   where ResultDescription contains "Application"   order by OperationName</p> <p>AppServiceAppLogs   project ResultDescription , Level</p> <p>AppServiceAppLogs   where TimeGenerated &gt; todatetime('3/11/2022')</p> <p>AppServiceAppLogs   where TimeGenerated &gt; todatetime('3/11/2022')   summarize Count=count() by Source</p> <p><b>How can we query the logs / metrics in to .NET Core.</b></p>	
Frontdoor service, manager, Load Balancer Firewall , Load balancers, WAF( Web Application firewall) , CDN	<p>Demo of Azure front door service</p> <p>Dynamic site acceleration</p> <p>Azure CDN and Front door services</p> <p><a href="https://docs.microsoft.com/en-in/azure/cdn/cdn-create-new-endpoint">https://docs.microsoft.com/en-in/azure/cdn/cdn-create-new-endpoint</a></p> <p>All people seem to need data processing</p> <p><a href="https://www.networkworld.com/article/3239677/the-osi-model-explained-and-how-to-easily-remember-its-7-layers.html">https://www.networkworld.com/article/3239677/the-osi-model-explained-and-how-to-easily-remember-its-7-layers.html</a></p>	
Load balancer	<a href="https://docs.microsoft.com/en-us/azure/load-balancer/quickstart-load-balancer-standard-public-portal">https://docs.microsoft.com/en-us/azure/load-balancer/quickstart-load-balancer-standard-public-portal</a>	
Traffic Manager	<p><a href="https://docs.microsoft.com/en-us/azure/traffic-manager/quickstart-create-traffic-manager-profile">https://docs.microsoft.com/en-us/azure/traffic-manager/quickstart-create-traffic-manager-profile</a></p> <p><a href="https://loadium.io/">https://loadium.io/</a></p> <ol style="list-style-type: none"> <li>1. You can not use free app service</li> <li>2. Both app service should be in different geography</li> </ol>	
Firewall	<a href="https://docs.microsoft.com/en-us/azure/firewall/tutorial-firewall-deploy-portal-policy#configure-a-network-rule">https://docs.microsoft.com/en-us/azure/firewall/tutorial-firewall-deploy-portal-policy#configure-a-network-rule</a>	
Azure cosmosDB		
Azure Caching		
Azure monitoring		
Caching Azure..	What are option...	
Azure cognitive		
Storage	Cosmos DB	1hour
Dat lakes, Data bricks.		
Webjob		
Machin learning...		
Azure Data flow		
Azure Data service		



	Terraform/ Ansible	
	Mass transit..	

Azure load balancer	App gateway	Frontdoor	Traffic Manager
Works at Layer 4	Works at Layer 7	Works at Layer 7	DNS based load balancing
Global load balancing solution	Regional load balancing solution	Global load balancing solution	Global load balancing solution
Backends for regional azure load balancer are VMs, VM scalesets Backends for global azure load balancer is regional azure load balancer	Backends can be VMs, VM scaleset, azure app service, onpremises/external servers	Any internet facing service hosted inside or outside azure	Any internet facing service hosted inside or outside azure
Recommended for non HTTP(S) traffic	Recommended for HTTP(S) traffic	Recommended for HTTP(S) traffic	Recommended for non HTTP(S) traffic
SSL offloading is not supported	SSL offloading is supported	SSL offloading is supported	SSL offloading is not supported
WAF is not supported	WAF is supported	WAF is supported	WAF is not supported

Azure table code

<https://docs.microsoft.com/en-us/dotnet/api/overview/azure/data.tables-readme-pre>

Odata Query

<https://docs.microsoft.com/en-us/odata/concepts/queryoptions-overview>

Azure table cost

<https://azure.microsoft.com/en-in/pricing/details/storage/tables/>

Simple Azure blob

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-dotnet>

Azure table design patterns

<https://docs.microsoft.com/en-us/azure/storage/tables/table-storage-design-patterns>

<https://learnxinyminutes.com/docs/yaml/>

<https://docs.microsoft.com/en-us/azure/devops/pipelines/ecosystems/javascript?view=azure-devops&tabs=code>

Azure service bus portal and C# code

<https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-dotnet-get-started-with-queues>



questpond.teachable.com/courses/712305/lectures/12803068

0% COMPLETE

in 90 Minutes. (93:22)

- Lab 7 :- Azure tables, Partition and Row keys. (59:08)
- Lab 8 :- Block blobs, Append blobs & Page blobs. (55:29)
- Lab 9 :- Azure Queues, visibility timeouts, Peek & De-Queue. (28:21)
- Lab 10 :- Cloud services vs App services. (36:23)
- Lab 11 :- WebJob and background processing. (43:01)
- Lab 12 :- Azure DevOps using Azure Pipeline (79:46)

10 important best practices to remember

1. Think Read First, Point Queries
2. Data storage is cheap. Duplicate data with Multiple Keys and Aggregate Data.
3. Compound Keys
4. Hot Partition avoid.
5. Avoid unnecessary tables. Entity Group transactions
6. Intra Partition pattern
7. Inter -Partition Patten
8. Delete pattern
9. Large entities Pattern
10. Optimistic locking.

58:14

Download

	Intra			
Table	Employee			
Partition	Row Key	Name	Design	
Dept	Emp-1001	Shiv	sup	
Dept	Emp-1002	Raju	test	
Dept	<a href="#">EmpEmail-Shiv@yahoo.com</a>	Shiv	sup	
Dept	<a href="#">EmpEmail-Raju@yahoo.com</a>	Raju	test	
	Inter			
0 Dept-Emp		1001 Shiv	sup	
1 Dept-Emp		1002 Raju	test	
2 Dept-Email	<a href="#">Shiv@yahoo.com</a>	Shiv	sup	
3 Dept-Email	<a href="#">Raju@yahoo.com</a>	Raju	test	

Table Entity operations

```

class Patient : ITableEntity
{
    public string PatientName { get; set; }
    public string PatientAddress { get; set; }
    public string PartitionKey { get; set; }
    public string RowKey { get; set; }
    public DateTimeOffset? Timestamp { get; set; }
    public ETag ETag { get; set; }
}
class Program

```



```
{
    static void Main(string[] args)
    {
        Patient p = new Patient();
        p.PartitionKey = "OutPatient";
        p.PatientName = "Shiv123";
        p.RowKey = "r1009";
        p.PatientAddress = "Mumbai";
        var tableClient = new TableClient(new

Uri("https://storagequestpond.table.core.windows.net/Patients"), "Patients",
        new TableSharedKeyCredential("storagequestpond",
"xYiBpJZ8gN9znVvSKND+O6PULjHDkoUbOe0nXoBiAdmpI+UiTNfKw9qxckTpRtugg8Zj8RgWL/hFEe+eIewQJ
Q=="));

        //tableClient.AddEntity(p);

        Pageable<Patient> query = tableClient.Query<Patient>(filter: $"RowKey eq
'r1009'");
        foreach (var item in query)
        {

        }
        Patient entity = tableClient.GetEntity<Patient>("OutPatient", "r1009");
        entity.PatientName = "new name123";
        //tableClient.DeleteEntity()
        tableClient.UpdateEntity(entity, ETag.All, TableUpdateMode.Replace);
        Console.WriteLine("Hello World!");
    }
}
```

Upload and download

```
static async void Upload()
{
    string connectionString =
"DefaultEndpointsProtocol=https;AccountName=questpondstorage;AccountKey=T4mvdmtD9RouZT
niKnuxoSzVSbzKx60Zn0yRZZ0gQHdSTZC83Ez32EP+V+/xoSo0LNeG3QmMrh/+z6IoQRhmQ==;EndpointSuf
fix=core.windows.net";
    BlobServiceClient blobServiceClient = new
BlobServiceClient(connectionString);

    BlobContainerClient containerClient =
blobServiceClient.GetBlobContainerClient("blob123");

    BlobClient blobClient = containerClient.GetBlobClient("mp4");
    Console.WriteLine(blobClient.Uri);
    await blobClient.UploadAsync(@"E:\Untitled.mp4", true);
    //await blobClient.DownloadToAsync(downloadFilePath);
}
```

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-dotnet-legacy>

with file splits

```
public static async void Upload()
{
    string connectionString =
"DefaultEndpointsProtocol=https;AccountName=questpondstorage;AccountKey=T4mvdmtD9RouZT
```





```
niKnuxoSzVShbzKx60Zn0yRZZ0gQHdSTZC83Ez32EP+V+/xoSo0LNeG3QmMrh/+z6IoQRhmQ==;EndpointSuffix=core.windows.net";
```

```
CloudStorageAccount storageAccount =  
CloudStorageAccount.Parse(connectionString);  
// Step 2 :- get reference o the blob client  
CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();  
blobClient.DefaultRequestOptions = new BlobRequestOptions()  
{  
  
    SingleBlobUploadThresholdInBytes = 1024 * 1024, //1MB, the minimum  
    ParallelOperationThreadCount = 1  
};  
// Step 3 :- from the blob client you will get refernce to the container  
CloudBlobContainer container =  
blobClient.GetContainerReference("blob123");  
// Step 4 "- use the container and get access to block blob  
// Upload(container, @"e:\Untitled.mp4");  
CloudBlockBlob blockBlob = container.GetBlockBlobReference("myf123");  
//blockBlob.DownloadToFile(@"c:\my.pdf", FileMode.CreateNew);  
//foreach (var blockListItem in blockBlob.DownloadBlockList())  
//{  
//    Console.WriteLine("Block ID: " + blockListItem.Name);  
//    Console.WriteLine("Block size: " + blockListItem.Length);  
//}  
  
blockBlob.StreamWriteSizeInBytes = 1024 * 1024;  
// Step 5 :- Uploading a PDF 13 mb  
await blockBlob.UploadFromFileAsync(@"e:\Untitled.mp4");  
//blockBlob.StreamMinimumReadSizeInBytes = 1024 * 1024;  
//using (var blobStream = await blockBlob.OpenReadAsync())  
//{  
//    using (var fs = new FileStream(@"c:\mynew.pdf", FileMode.Create))  
//    {  
//        await blobStream.CopyToAsync(fs);  
//    }  
//}  
//await blobClient.DownloadToAsync(downloadFilePath);  
}
```

Append blob

```
CloudStorageAccount storageAccount =  
CloudStorageAccount.Parse(@"DefaultEndpointsProtocol=https;AccountName=shivblob;AccountKey=  
=nuGq71akmiGgxAwUpzuq4L+WocsioPcHPt1kGbVWekFf/9h+8ODxGAqnY387f7glNoftXd8oMgEHS6Z  
23wXUSg==;EndpointSuffix=core.windows.net");
```

```
// Step 2 :- get reference o the blob client
```

```
CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();
```

```
// Step 3 :- from the blob client you will get refernce to the container
```

```
CloudBlobContainer container = blobClient.GetContainerReference("mycontainer");
```

```
// Step 4 :- Append blob
```

```
CloudAppendBlob appendBlob = container.GetAppendBlobReference("myblobapp");
```

```
//appendBlob.DownloadToFile()
```



```
if (!appendBlob.Exists())
{
    appendBlob.CreateOrReplace();
}

appendBlob.AppendFromFile(@"d:\Hello1.txt");
appendBlob.AppendFromFile(@"d:\Hello2.txt");
```

#### Page blob

```
CloudStorageAccount storageAccount =
CloudStorageAccount.Parse(@"DefaultEndpointsProtocol=https;AccountName=shivblob;AccountKey=
nuGq71akmiGgxAwUpzuq4L+WocsioPcHPt1kGbVWekFf/9h+8ODxGAqnY387f7glNoftXd8oMgEHS6Z
23wXUSg==;EndpointSuffix=core.windows.net");
```

```
// Step 2 :- get reference o the blob client
```

```
CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();
```

```
// Step 3 :- from the blob client you will get refernce to the container
```

```
CloudBlobContainer container = blobClient.GetContainerReference("mycontainer");
```

```
CloudPageBlob pb = container.GetPageBlobReference("pageblob123");
```

```
if (!pb.Exists())
```

```
{
```

```
    pb.Create(3 * 512);
```

```
}
```

```
Stream blobStream = pb.OpenRead();
```

```
byte[] data = new byte[512];
```

```
blobStream.Seek(512, SeekOrigin.Begin); // start reading fro 512
```

```
blobStream.Read(data, 0, 512);
```

```
File.WriteAllBytes(@"d:\new2.txt", data);
```

```
//IEnumerable<PageRange> ranges = pb.GetPageRanges();
```

```
//foreach (PageRange range in ranges)
```

```
//{
```



```
// Console.WriteLine(range.StartOffset + " " + range.EndOffset);  
  
//}  
  
//byte[] bytearray = new byte[512];  
  
//MemoryStream m = new MemoryStream();  
  
//FileStream fs = new FileStream(@"d:\Hello1.txt", FileMode.Open);  
  
//fs.Read(bytearray, 0, 512);  
  
//m = new MemoryStream(bytearray);  
  
//pb.WritePages(m, 0);  
  
  
//fs = new FileStream(@"d:\Hello2.txt", FileMode.Open);  
  
//fs.Read(bytearray, 0, 512);  
  
//m = new MemoryStream(bytearray);  
  
//pb.WritePages(m, 512);
```

<https://docs.microsoft.com/en-us/azure/storage/queues/storage-dotnet-how-to-use-queues?tabs=dotnet>

```
string connectionString =  
"DefaultEndpointsProtocol=https;AccountName=questpondstorage;AccountKey=k/Td/CrBY1XHQP  
LKaRT+Ag1n7xukLn7gFEgsfkDzBaxP7o3TYi5LrhvLY9nhBU6gWA9QotdD3U1S7pK0aSQRDw==;EndpointSuf  
fix=core.windows.net";
```

```
the queue // Instantiate a QueueClient which will be used to create and manipulate  
  
QueueClient queueClient = new QueueClient(connectionString, "myqueue123");  
//queueClient.CreateIfNotExists();  
//queueClient.SendMessage("Hello 1");  
//queueClient.SendMessage("Hello 2");  
//PeekedMessage[] peekedMessage = queueClient.PeekMessages();  
//Console.WriteLine(peekedMessage[0].Body);  
QueueMessage[] retrievedMessage = queueClient.ReceiveMessages();  
  
// Process (i.e. print) the message in less than 30 seconds  
//Console.WriteLine($"Dequeued message: '{retrievedMessage[0].Body}'");  
  
retrievedMessage = queueClient.ReceiveMessages();  
foreach (var item in retrievedMessage)  
{  
    Console.WriteLine(item.Body);  
}  
// Process (i.e. print) the message in less than 30 seconds  
//Console.WriteLine($"Dequeued message: '{retrievedMessage[0].Body}'");  
// Delete the message  
queueClient.DeleteMessage(retrievedMessage[0].MessageId,  
retrievedMessage[0].PopReceipt);
```



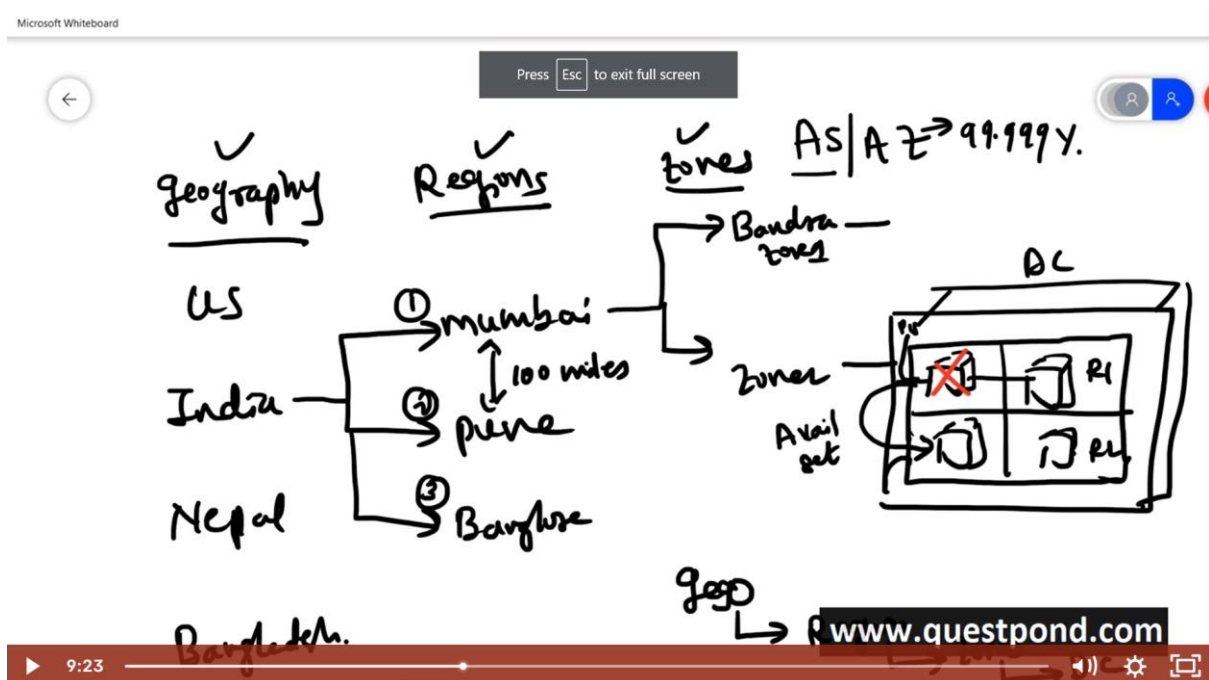
```
Console.WriteLine("Hello World!");
```

Azure files

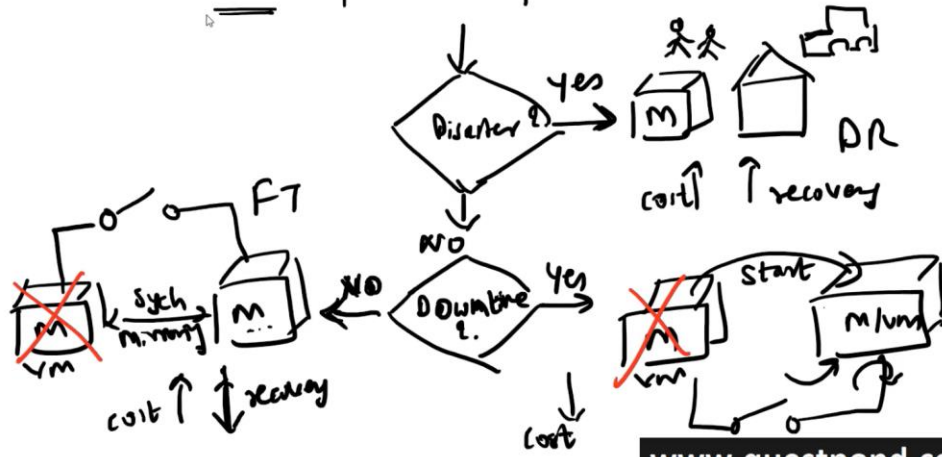
<https://docs.microsoft.com/en-us/dotnet/api/overview/azure/storage.files.shares-readme>

Home work..

<https://docs.microsoft.com/en-us/azure/logic-apps/quickstart-create-logic-apps-with-visual-studio>

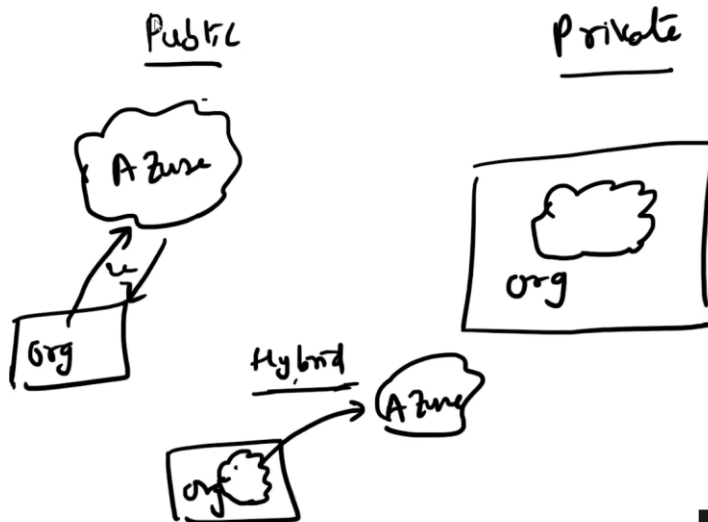


DR | FT | HA



www.questpond.com

7:48

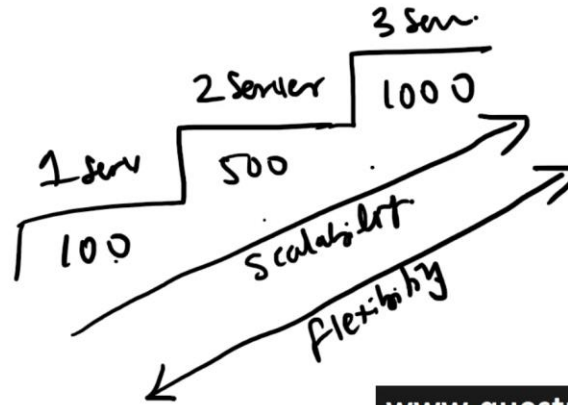


www.questpond.com

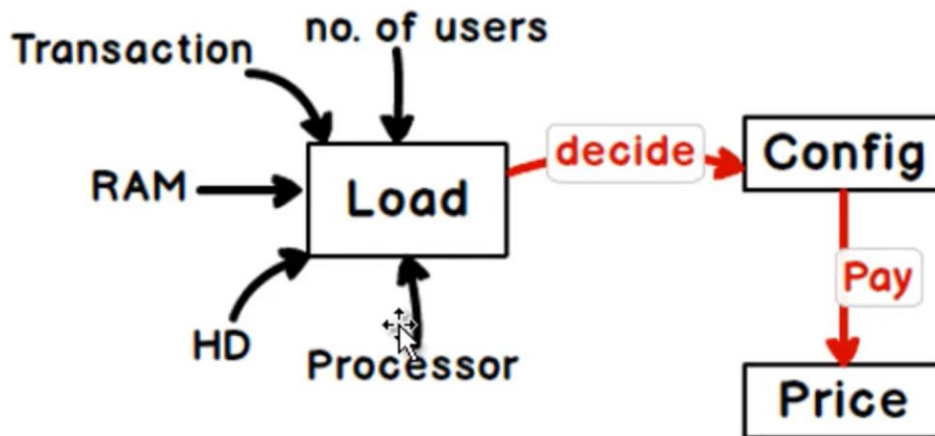
4:34

1s → 2s  
→ now

# Agility, Scalability & Flexibility



www.questpond.com



www.Questpond.com

questpond.teachable.com/courses/712305/lectures/12803029

0% COMPLETE

IAAS. (40:31)

Lab 2:- SQL Server on Azure ( DTU & EDTU) (32:41)

Lab 3:- Azure Functions. (25:12)

Lab 4:- Azure storage (Blobs,Table,File and Queue). (23:51)

Lab 5:- Azure Cosmos DB. (32:46)

Lab 6:- Microservices & Azure Fabric in 90 Minutes. (93:22)

Lab 7:- Azure tables, Partition and Row keys. (59:08)

no. of Writes  
RAM in GB  
no. of Reads

CPU %

Primary (MDF)

Amount of Log MB flushed

Log Trans (LDF)

DTU - Database Transaction Units

5:03

Download

questpond.teachable.com/courses/712305/lectures/12803029

0% COMPLETE

IAAS. (40:31)

Lab 2:- SQL Server on Azure ( DTU & EDTU) (32:41)

Lab 3:- Azure Functions. (25:12)

Lab 4:- Azure storage (Blobs,Table,File and Queue). (23:51)

Lab 5:- Azure Cosmos DB. (32:46)

Lab 6:- Microservices & Azure Fabric in 90 Minutes. (93:22)

Lab 7:- Azure tables, Partition and Row keys. (59:08)

Log MB/sec

CPU

no. of Reads

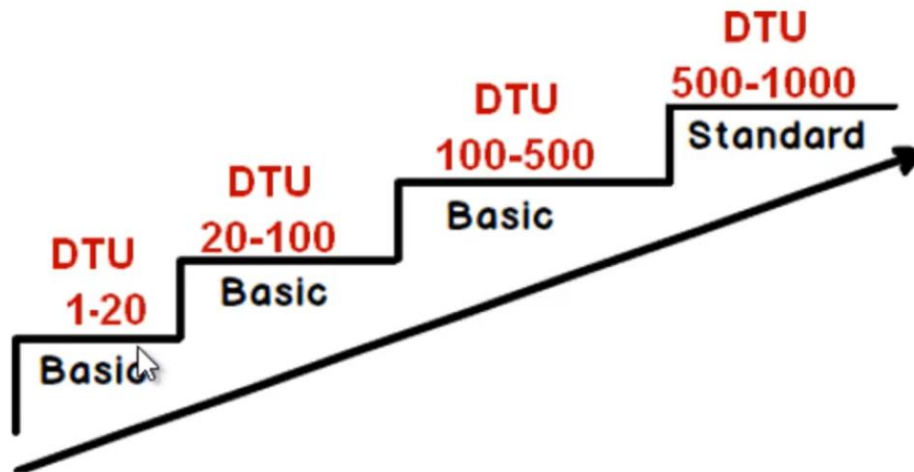
RAM

no. of Writes

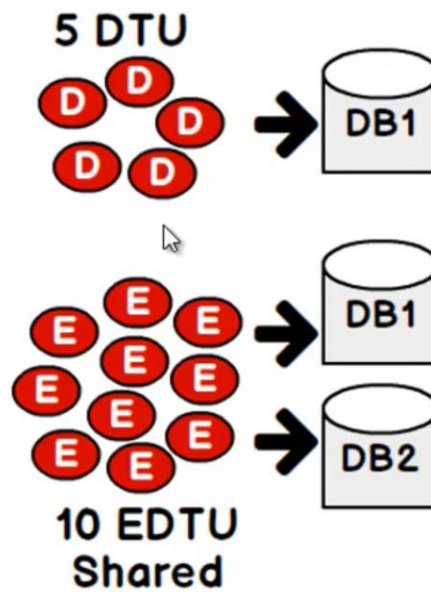
DTU

7:30

Download



www.Questpond.com



<https://dtucalculator.azurewebsites.net/>

<https://docs.microsoft.com/en-us/azure/active-directory/develop/tutorial-v2-asp-webapp>





<https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-v2-aspnet-core-webapp>

Angular API

<https://docs.microsoft.com/en-us/rest/api/servicebus/get-azure-active-directory-token>

<https://docs.microsoft.com/en-us/graph/auth-v2-service#token-request>

MVC Conttoller

### Web API

```
"AzureAd": {
  "Instance": "https://login.microsoftonline.com/",
  "Domain": "https://localhost:44360/",
  "ClientId": "1eb4d1a2-9673-49e2-8c2f-88fdf8cb3761",
  "TenantId": "bf65742e-d37d-46f9-bc1d-24def2b7bc1d",
  "CallbackPath": "/signin-oidc"
}

public void ConfigureServices(IServiceCollection services)
{
    services.AddAuthentication(OpenIdConnectDefaults.AuthenticationScheme)
        .AddMicrosoftIdentityWebApp(options => Configuration.Bind("AzureAd",
options));

    services.AddControllersWithViews(options =>
    {
        var policy = new AuthorizationPolicyBuilder()
            .RequireAuthenticatedUser()
            .Build();
        options.Filters.Add(new AuthorizeFilter(policy));
    }).AddMicrosoftIdentityUI();
}

// This method gets called by the runtime. Use this method to configure the
HTTP request pipeline.
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        // The default HSTS value is 30 days. You may want to change this for
production scenarios, see https://aka.ms/aspnetcore-hsts.
        app.UseHsts();
    }
    app.UseHttpsRedirection();
    app.UseStaticFiles();

    app.UseRouting();
```



```
app.UseAuthentication();
app.UseAuthorization();

app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Home}/{action=Index}/{id?}");
});
}
```

GET

▼

https://login.microsoftonline.com/bf65742e-d37d-46f9-bc1d-24def2b7bc1d/oauth2/v2.0/token

Params

Authorization

Headers (10)

Body ●

Pre-request Script

Tests

Settings

●

 none

●

 form-data

●

 x-www-form-urlencoded

●

 raw

●

 binary

●

 GraphQL

	KEY	VALUE	DESCRIPTION
<div>✓</div>	grant_type	client_credentials	
<div>✓</div>	client_id	1eb4d1a2-9673-49e2-8c2f-88fdf8cb3761	
<div>✓</div>	client_secret	Epi7Q~dNOXR2JG5lik2mvCqTp-SFWdl6~...	
<div>✓</div>	scope	.default	
	Key	Value	Description

## Docker help

dockerfile with out any extension

.dockerignore

docker build please put the . at the last

### 1. Important commands

To create image.

docker build -t mvcoreimage1 .

To create container



```
docker run -d -p 8080:80 --name mvcccontainernew1 mvcoreimage1
```

To list all images in a computer

```
docker images
```

To list all containers in a computer.

```
docker container list -a
```

To check container details like IP address

```
docker inspect mvccore5
```

To stop a image

```
docker stop imagename
```

Power shell commands to delete all images and containers.

```
docker ps -a -q | % { docker rm $_ }
```

```
docker images -q | % { docker rmi $_ }
```

Check if there are issues with the container

```
docker logs containername
```

with out stopping you can not delete

2. docker file for MVC core (dockerfile)

```
FROM mcr.microsoft.com/dotnet/core/sdk:3.1 AS build
```

```
WORKDIR /MyCore123
```

```
# copy csproj and restore as distinct layers
```



```
COPY *.sln .
```

```
COPY MyCore123/*.csproj ./MyCore123/
```

```
RUN dotnet restore
```

```
# copy everything else and build app
```

```
COPY MyCore123/. ./MyCore123/
```

```
RUN dotnet publish -c Release -o out
```

```
FROM mcr.microsoft.com/dotnet/core/aspnet:3.1 AS runtime
```

```
WORKDIR /MyCore123/MyCore123
```

```
COPY --from=build /MyCore123/out ./
```

```
ENTRYPOINT ["dotnet", "MyCore123.dll"]
```

2. docker file for MVC 5(dockerfile)

```
FROM microsoft/aspnet
```

```
COPY ./bin/Release/Publish/ /inetpub/wwwroot
```

Step 3 :- .dockerignore

```
# directories
```

```
**/bin/
```

```
**/obj/
```

```
K8s ... Coobernetes
```

```
coob-control
```

```
coob-let
```

<https://www.knowledgehut.com/blog/devops/install-kubernetes-on-windows>



```
. kubectl apply -f .\recommended.yaml
```

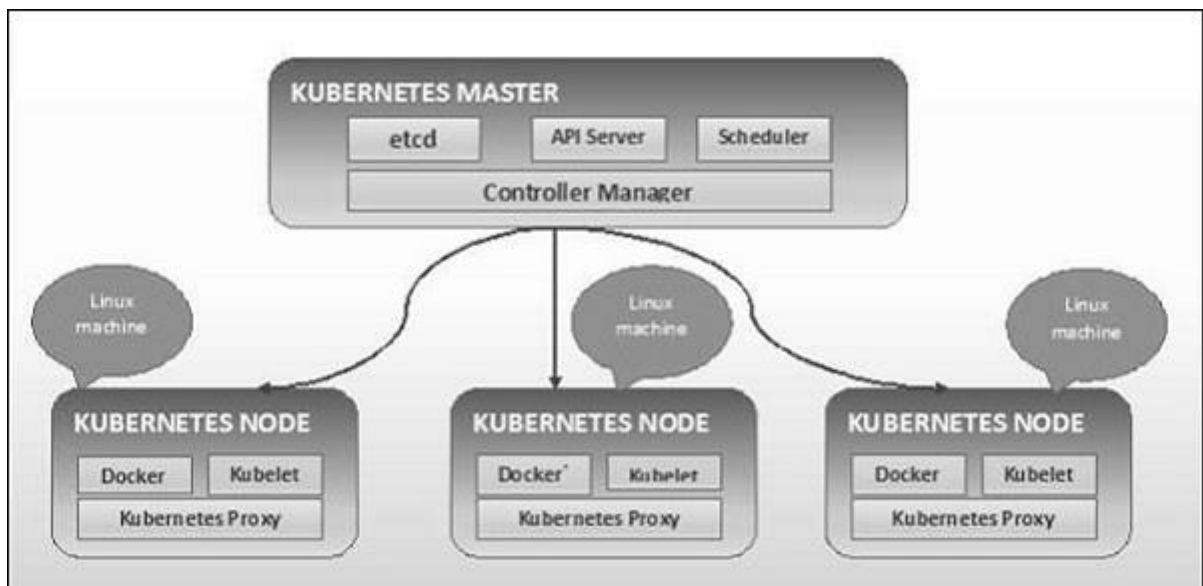
<https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-rc7/aio/deploy/recommended.yaml>

```
kubectl.exe get -f .\recommended.yaml.txt
```

```
((kubectl -n kube-system describe secret default |  
Select-String "token:") -split " ")[1]
```

```
kubectl proxy.
```

<http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/>



<https://github.com/kubernetes/minikube/releases/latest/download/minikube-installer.exe>

```
minikube start
```

```
kubectl get nodes
```

```
minikube dashboard
```

```
minikube dashboard -url
```

```
minikube status
```

```
kubectl get pods --all-namespaces
```

```
kubectl create deployment hello-minikube --  
image=k8s.gcr.io/echoserver:1.4
```

```
kubectl expose deployment hello-minikube --type=NodePort --  
port=8080
```



```
Minikube service hello-minikube --url
```

```
kubectl delete service mynginxapp
```

```
kubectl delete deployment mynginxapp
```

<https://alexanderzeitler.com/articles/running-asp-net-core-on-minikube/>

```
kubectl api-versions
```

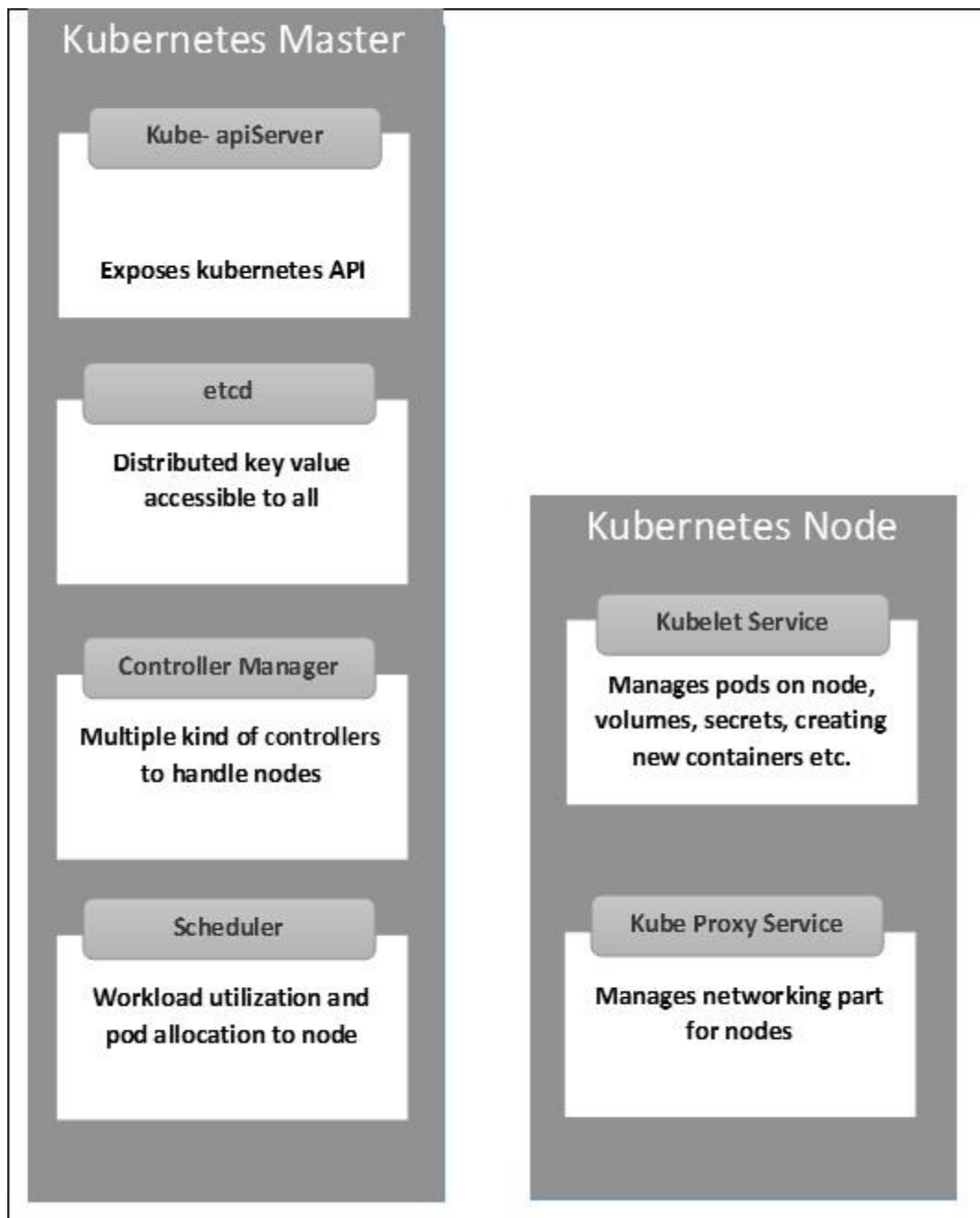
```
kubectl explain
```

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: hello-netcore-k8s
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: hello-netcore-k8s
    spec:
      containers:
        - name: hello-netcore-k8s
          image: hello-netcore-k8s
          ports:
            - containerPort: 80
```

```
kubectl create -f deployment.yaml
```

```
docker context use default
```

<https://docs.microsoft.com/en-us/azure/container-registry/container-registry-get-started-docker-cli?tabs=azure-cli>



<https://docs.microsoft.com/en-us/azure/container-registry/container-registry-get-started-portal>

<https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.2>

<https://docs.microsoft.com/en-us/azure/container-registry/container-registry-get-started-docker-cli?tabs=azure-cli>

```
Connect-AzAccount -DeviceCode
```

<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli>



```
docker tag mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine questpondacr.azurecr.io
docker run -it --rm -p 8080:80 questpondacr.azurecr.io/v1
```

```
kubectl create deployment hello3-minikube --image=
questponacr.azurecr.io/mvcoreimage:latest
```

```
kubectl expose deployment hello1-minikube --type=NodePort --
port=8081
```

```
minikube image load imagename:tag
```

```
kubectl create deployment hello3-minikube --image=
questponacr.azurecr.io/mvcoreimage:latest
```

```
--overrides='{ "spec": { "template": { "spec": { "imagePullSecrets": [{"name": " sec123 "}]} } } }
```

```
docker tag myimage questacr1001.azurecr.io/myimage:latest
```

```
docker push questacr1001.azurecr.io/myimage:latest
```

```
az acr login questpondacr
```

<https://docs.microsoft.com/pt-br/azure/container-registry/container-registry-auth-kubernetes>

<https://docs.microsoft.com/en-us/azure/aks/cluster-container-registry-integration?tabs=azure-cli>

## APIM

<https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-policies#how-to-configure-policies>

<https://docs.microsoft.com/en-us/azure/api-management/transform-api>

## Products

<https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-add-products?tabs=azure-portal>