# Challenge 0: Deploy the “on premise” environment

In this exercise, you will set up your environment for use for the rest of the exercises. This will involve downloading the assets from the repository and deploying them to Azure. We will use your workstation to set up the environment and connect to Virtual Machines via Remote Desktop.

# Proctor Notes & Guidelines

Pretty simple click deploy to Azure and the deployment takes about 30 mins to provision, load the images and configure the software.

### Default Password Settings

Admin User Name: sysadmin  
Admin User Password: Password$123

db Login Name: workshopServiceAcc  
db User Name: workshopServiceAcc  
db User Password: P2ssw0rd

Database Name: partsUnlimitedDB

Note: if you don't have a Visual Studio Enterprise License, please select the community edition for the Dev Workstation OS Version.

Also Do not modify the values for \_artifacts Location or \_artifacts Location SAS Token.

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Challenge 0.5: Verify the “on premise” environment

Once the deployment is complete, review the vmweb01 configuration to find the public URL.  Copy the URL, open a browser, and paste and go to verify that everything worked correctly.

@ Parts Unlimited 
a Fabrikam subsidiary 
Brakes 
Lighting Wheels & Tires 
Batteries 
New Arrival : Bugeye Headlights (2 Pack) 
(eyword 
0000 
Search 
Home Cert 
Log in 
Oil 
More 
NEW TIRES 
0 IMPROVED FUEL EFFICIENCY 
SUPERIOR WET WEATHER BREAKING. 
ADDED DURABILITY 
Shop

# Proctor Notes & Guidelines

Students can find the host name for the website using the Azure Portal and looking at the public DNS address for the vmweb01.

# Challenge 1 Migrate to Azure PaaS

Migrate the parts unlimited web application to Azure leveraging Azure PaaS like Azure App Service and Azure SQL.

There are multiple options using Azure Migrate or from within Visual Studio (accessible from your Dev)1 Box in your environment. your on-premise application and data to PaaS The

Hint’s: You may want to verify requirements for the particular tool you wish to use to migrate the web app it may require you to use the Microsoft Web Platform Installer (already installed in the machine, but available for download at [https://www.microsoft.com/web/downloads/platform.aspx](https://www.microsoft.com/web/downloads/platform.aspx%20)

# Proctor Notes & Guidelines

There ae multiple solutions to achieving this task,

* Azure Migrate Tool
* Visual Studio

The simplest might be to push students to using Azure Migrate. However, there are a couple of issues to be aware of:

1. Should use the tool from vmweb01
2. Verify that enhanced IE restrictions are turned off
3. Use the Web Platform installer to add two libraries:
   1. SQL Server 2012 Shared Management Objects (April 2012)
   2. Microsoft SQL Server Data-Tier Application Framework (DACFx) (June 2014)
4. Verify new unique names for the services, as the migration tool will migrate multiple services into the same Resource Group per Datacenter looks like “Server-Migration-WestUs2”

Note: Use the admin user and password that you used for the server and Use the same name for the database that you used when you created the environment

Default Settings:

Default DB Login Name: workshopServiceAcc

Default DB User Name: workshopServiceAcc

Default DB User Password: P2ssw0rd

Default Database Name: partsUnlimitedDB

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Challenge 2: Implementing DevOps Continuous Integration / Continuous Deployment (CI/CD)

Parts Unlimited wants to improve their deployment process to enable more agility. Currently, Parts Unlimited has a very manual deployment process. Developers document the install steps and operations performs the deployment. This process worked when they pushed code only a few times a year. However, the goal is to rapidly deliver value in days as opposed to months.

Currently, there is no automated deployment process. Visual Studio Team Services has been chosen as the ALM/DevOps platform moving forward. You will implement a CI/CD process from Team Services to Azure. Once the CI/CD process is in place you can quickly deploy changes to production with full traceability. The automation not only shortens your delivery time, it also dramatically improves the repeatability. Additionally, you can add unit testing, automated UI testing, Load Testing, Manual Testing, approval processes, etc. to ensure you do not sacrifice quality for speed.

# Proctor Notes & Guidelines

There are multiple paths to complete this task, using Azure Devops Portal, from Visual Studio and from the Azure Portal.

There is a wizard that will help here: [https://azure.microsoft.com/en-us/try/devops/](mhtml:file://C:\Users\danshue\Desktop\Workshop\AppMigrationAndModernizationWorkshopPresenter.mht!https://azure.microsoft.com/en-us/try/devops/)

The source code is also available of the Dev01 virtual machine in the c:\Source\AppWorkshop folder.

Note:

Default Admin: sysadmin

Default Password: Password$123

There may be an issue with the application connection string from using the migration tool, if you get a Server Error in ‘/’ Application check that they are pointing to the right resources. It is a bug in the tool.

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Challenge 3: Detect, Diagnose, Monitor Application Performance with Azure App Insights

Parts Unlimited understands that bad experiences online will quickly drive a customer to your competitor. With that in mind, they want to gain better visibility into issues customers are experiencing while using their site.

In addition to gaining better insights into the sites health, they are also interested in better understanding how customers are using the site and the sites effectiveness.

As there is no instrumentation on the site today, they have decided to add Application Insights telemetry.

Timely actionable telemetry is an important aspect of any modern application. Actionable information is a vital component of any DevOps practice.

* Add Application Insights to the web app
* Modify the code to instrument the app
* Test locally to verify the app works
* Push the changes to kick off a release
* Test that the production application works
* Review App Insights Data

# Proctor Notes & Guidelines

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Challenge 4: Optimize and Protect App

While the default settings for deploying web application may work in many situations, we want to examine our Azure App Service Web Application and ensure its secure, optimized, and able to meet the demands of our customers. We will use the Azure portal to configure the ensure backups, logging is enabled, and the application is sized and can scale as need.

This lab will demonstrate how to safe guard and protect your application and make it available by:

* Resource Group cannot be deleted by creating a lock
* Configure backups are setup with 120-day retention
* Ensure the average response time is < 2 seconds
* Configure alerts exist for CPU < 90 secs
* Configure diagnostics and exception logging is configured
* Ensure the average CPU < 20%
* Verify that there’s 35 GB free space available
* Scale up to a premium hosting model with > 3GB RAM available

Azure App Service services keeps you informed about the state of your application. It provides the ability to safe guard the application by performing automatic backups, provides a consolidate method for collection information about your application, has the ability of scaling up/down and notifying you when it does.

# Proctor Notes & Guidelines

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Challenge 5: Optimize & Protect Data

While the default settings for deploying SQL database may work in many situations, we want to examine the database and ensure its secure, optimized, and able to meet the demands of our customers. We will use the Azure portal to configure the ensure tuned, diagnostics is enabled, and the database is sized and can scale as need.

This exercise will demonstrate how to safe guard and protect your database and make it available by:

* Database DTU utilization is < 20%
* >= 3000 DTUs are available for the database
* Database is accessible only from Azure (not externally)
* Database is failover-ready to another region (without app code change)
* Threat Detection is enabled
* Automatic Tuning is enabled
* Diagnostics are enabled
* Alerts anytime DTU usage is > 60%
* Database vulnerabilities are automatically discovered

# Proctor Notes & Guidelines

About this time might be a good idea to request an API Management Instance so it can be running by the time it comes up as a challenge

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Exercise 6: Optimize Entire Web Application Performance with Content Distributed Network (CDN)

As the improvements have been rolling out to the Parts Unlimited site, marketing has been ramping up its advertising causing an uptick in traffic. While this is exactly what the company wants, it has caused some latency issues. Improvements need to be made to the Parts Unlimited site to reduce latency and improve end user experience. This must be done quickly as there are additional marketing campaigns on the horizon.

To solve the performance issues as quickly as possible, we are going to implement a Content Delivery Network for our site. It has been decided that the entire site will be cached with CDN.

* Create a CDN endpoint to Web App
* Configure for correct behavior of when using Query String
* Change for even faster performance by adding compression

With Azure CDN you have a choice of different CDN providers. Configuring the service is quick, usually around a few minutes, and billing is based on use as opposed to a contract.

# Proctor Notes & Guidelines

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Exercise 7: Optimize Partial Web Application Performance with Content Distributed Network (CDN)

In the previous CDN exercise, we showed how you can turn on CDN for the whole entire site. However, for some sites with special content such as large files (e.g. large documents, video, etc.), we may want to optimize the delivery for that specific type of content. This lab will show you how to take a directory where your content in stored offload that content to a CDN.

* Create Resource Group & Storage Account
* Create a CDN Profile
* Create a CDN Endpoint
* ouPublish Content: Copy images to storage account
* Change code to use CDN
* Explore additional features

## 

# Proctor Notes & Guidelines

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Exercise 8: Increase Application / Database Performance with Redis Cache

It has come to the businesses attention that customers are experiencing slow responses. Configure a Redis Cache to improve performance. Specifically:

* Provision the Azure Redis service
* Improve session state by using Redis as Session State Provider
* Speed up output by using Redis as Output Cache Provider
* Reduce database usage by using Redis as Data Cache

# Proctor Notes & Guidelines

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Exercise 9: Improve Quality & Performance of Search with Azure Search

It has come to the businesses attention that customers are not able to find what they are looking for easily. After some investigation it was determined that the existing search capabilities were not adequate and needed to be updated. To improve the application’s search capability as quickly as possible, with as little code as possible, while ensuring the ability to support the sites growing traffic, Azure Search will be implemented.

* Go to the Azure Portal
* Look at the Parts Unlimited DB config
* Add Azure Search
* Create a branch
* Modify the application to use Azure Search
* Test Local
* Commit and push to release
* Test in production

# Proctor Notes & Guidelines

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Exercise 10: Accelerate development and take advantage of serverless using Azure Functions

Parts Unlimited has been growing. The business wants to start building relationships with other retailers, so they can sell their product through multiple channels, not just Parts Unlimited.

To get started, the Parts Unlimited site needs to be improved architecturally so portions of its functionality can be reused when integrating with third parties. To accomplish this, the products model will be split into a reusable class. Functions will then be built using this class. Next, the Parts Unlimited website will be modified to use the Azure functions as opposed to accessing the database directly.

# Proctor Notes & Guidelines

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions

# Exercise 11: Monetize your data and services, and open new channels to customers using Azure API Management

Now that Parts Unlimited has an API available, the business wants to start building relationships with other retailers, so they can sell their product through multiple channels. We’ve decided to use Azure API Management to expose and manage our API.

#### 

# Proctor Notes & Guidelines

If you get stuck anywhere please see the Step By Step Guide for explicit instructions for each challenge in multiple solutions