# Welcome to Project Thor.

Thor is an open source project undertaken by the Twin Cities Cloud Computing User Group under the leadership of Adam Grocholski, Architect, RBA Consulting.

Thor is designed to be a mobile utility that allows schedules and scheduling services to be accessed quickly by end users.

## The scenarios Thor addressed include:

* **Internal conference room scheduling** – Ever have a person visit your office and you need a place to meet? You make a quick pass around the usual suspects looking for an open room. You find one but you note sure how long it’s open… Then you want to book it… but what’s the alias to do that? Thor makes it easy.
* **External Schedule of Events for a Conference** – Provide people a live calendar of events direct to their mobile phone without the need to create a new web site or management application
* **Public Conference room Schedules** – Allow Anonymous users to access room schedules securely without needing direct access to exchange!

## Requirements:

* Azure Compute Account
* Azure Storage Account
* Access to an Instance of Microsoft Exchange Server 2007 or newer
* MS Tag Account

## Installation Notes:

This document covers the installation and configuration of Version 1.1 of Thor.

This document assumes that you have a Windows Azure account and Access to Exchange 2007+ server whether hosted or on premise. This document also assumes you understand the basics of deploying into and managing an Azure application.

Thor stores logs in Azure Storage so it is recommended that you have access to some kind of Azure storage utility to view and manage these log files.

## Breaking Changes:

If you are upgrading from version 1.0, you will need to take the following steps:

1. Stop your existing deployment of Thor.
2. Delete your existing deployment of Thor.
3. Delete the following tables from the storage account used by Thor:
   1. Settings
   2. AdministratorAccounts
   3. Calendars
   4. Appointments
   5. InvalidLogins
4. After you deploy and start the deployment of Thor in Azure, you will need to reconfigure Thor via the configuration UI (http://<<yourappname>>.cloudapp.net/configuration).

## Installation

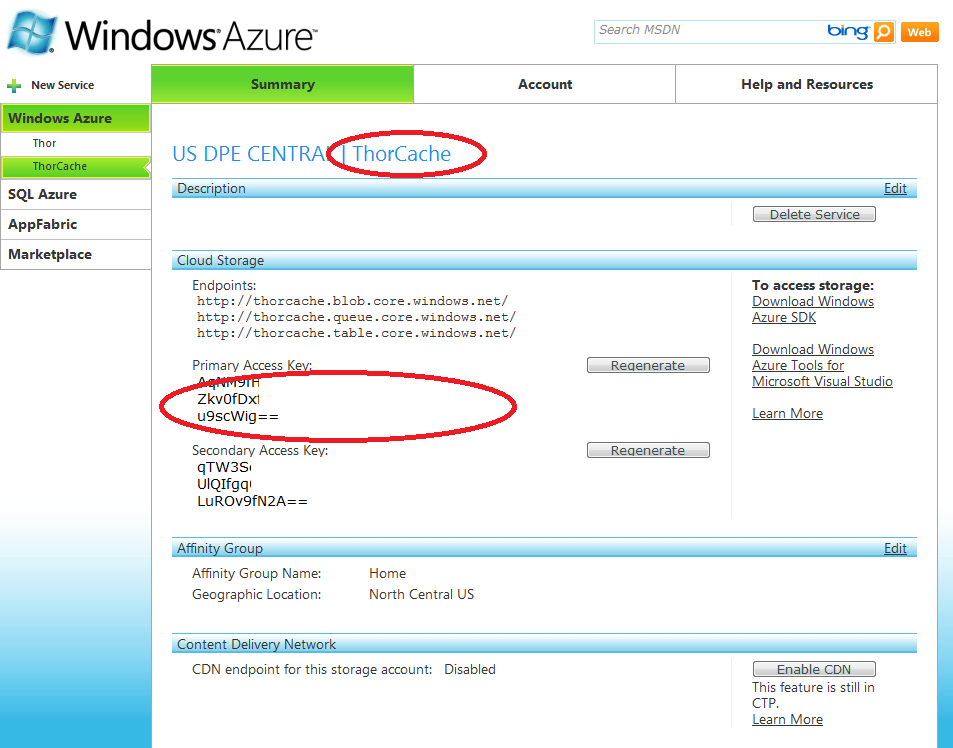
1. Download the Thor Azure Deployment package from <http://thor.codeplex.com/releases/view/40406> you’re looking for the Thor.Deploy file. This contains the code ready for deployment. You can also get the Thor source code from the same site
2. Download the appropriate calendar provider: Exchange 2007 Interop, Exchange 2010 Interop or the Exchange Web Services provider from <http://thor.codeplex.com/releases/view/40406>. The providers are named:
   * Thor.exchange.interop.exchage2007.zip
   * Thor.exchange.interop.exchange2010.zip
   * Thor.exchange.webservices.zip
3. Once the Thor files are down loaded you will need to modify the data source specification in the ServiceConfiguration.cscfg file. There are PowerShell scripts to perform this function but there is currently a bug in the API required. So we have to do this the hard way. Open the ServiceConfiguration.cscfg file and change:



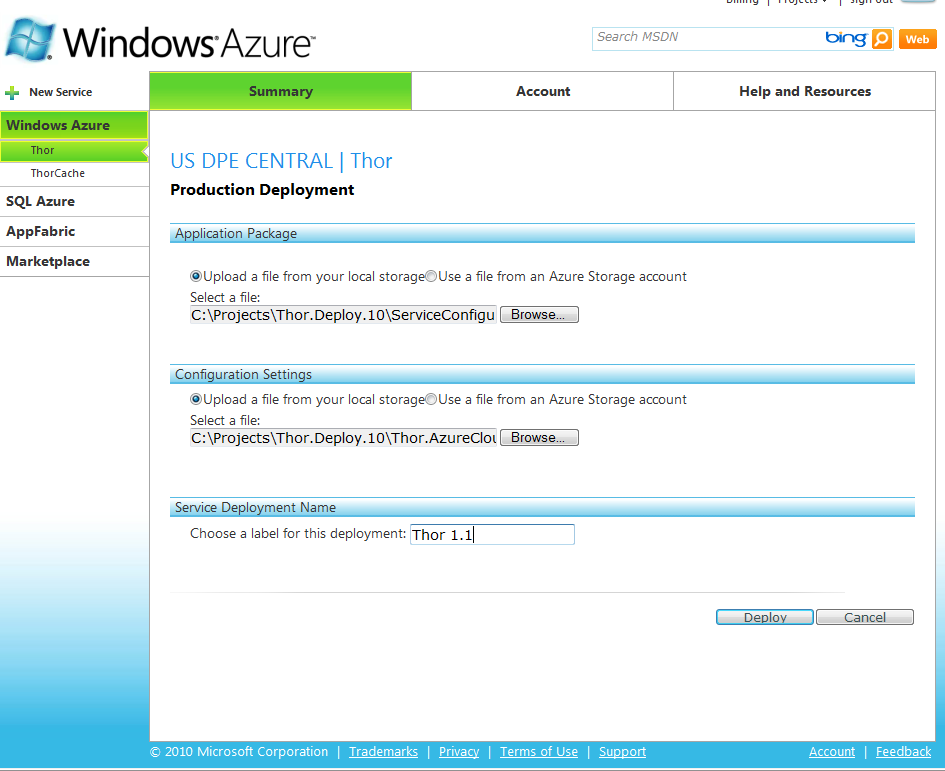
You need to change this value to point to your Azure storage account. For example:

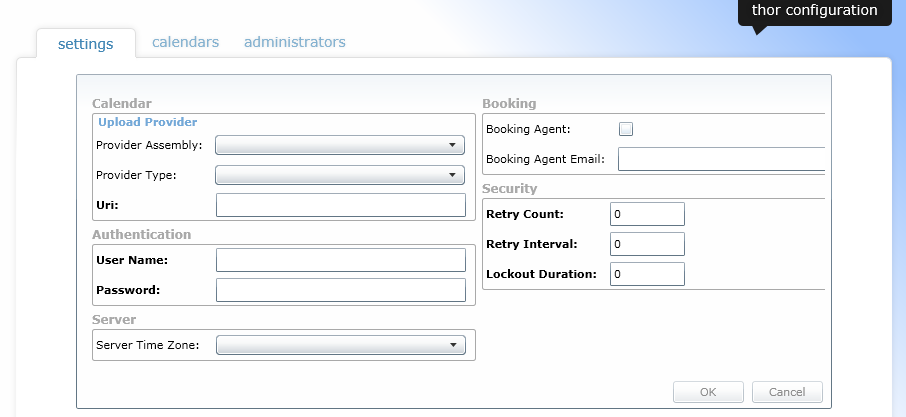
value="DefaultEndpointsProtocol=https;AccountName=**thorcache**;AccountKey=**AqNM9fHWEaMTQKebXXXXYXYXYXYYSYSYYXYXYXYXYXYYXYXYXYXYXYXYXYXYXYXYXYXYXYXYXXYg**==" />

You get these values from the Azure portal:



Once your modification have been completed save the ServiceConfiguration.cscfg

1. Now deploy the solution to Azure using the portal… Thor also includes scripts to perform this but that will be covered in a different document. 
2. Once the solution has been deployed. Launch the URL provided and Thor’s management screen will appear. Click on the Upload Provider link and select one of the Zip providers you downloaded earlier



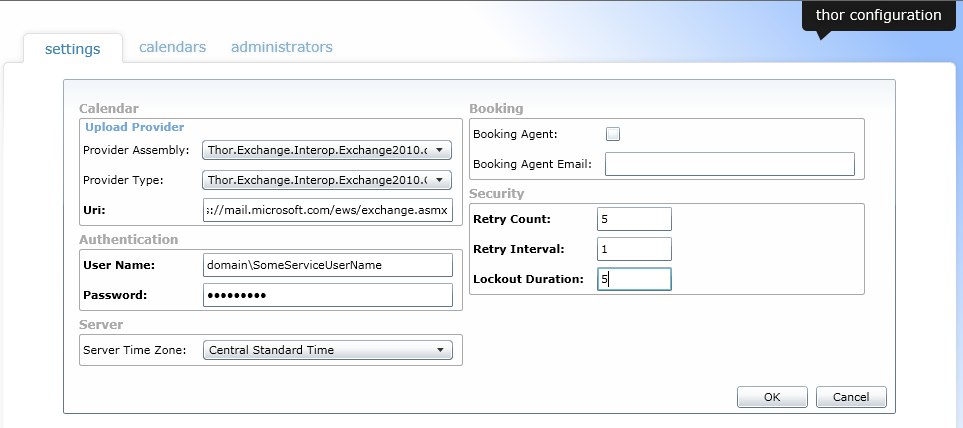
1. Now that the provider has been uploaded lets complete the configuration. This data is persisted in Azure Storage in an encrypted format. Once you’re done entering the data press OK before moving on to the next step.

Specifies whether or not to use a booking agent. If a booking agent is used, appointment requests will be emailed to the specified agent.



Email address to send appointment requests to.

Set Exchange Server Web Service ASMX URL



Specify how many tries a user gets during log in

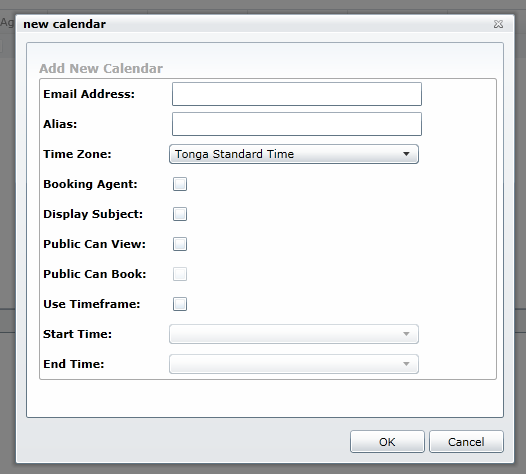
How many minutes between retries

If the user fails the login attempts how long should their accounts be locked out in minutes?

Select time zone to be used when calendar data is stored in Azure Storage

Specify domain account used for calendar processing by the Azure worker role

1. Configure the calendars Thor should monitor. Once again when you’re done adding calendars make sure you press OK to save those settings.



Whether or not unauthenticated users can book appointments on the calendar.

Whether or not to display the subject of the calendar’s appointments.

Whether or not unathenticated users can view the calendar.

Whether or not to use a booking agent when booking appointments for this calenar.



End time for the calendar time frame

Whether or not use a timeframe when displaying appointments for the calendar

Start time for the calendar time frame

1. Configure who will be the administrators for the Thor instance. Once again when you’re done adding administrators make sure you press the OK button to save the settings.
2. If you’ve done everything then you should be able to get a calendar by simply adding [/calendars/MyAccount@DomainName.com](mailto:/calendars/MyAccount@DomainName.com) and get a calendar.



1. Now to make the magic happen with MS Tag go to <http://www.microsoft.com/tag/> and setup an account. Create a new tag and just copy in the URL to the Thor managed calendar (http://myApp.CloudApp.net/calendars/mycalendar@MyDomain.com and presto you’re ready to amaze your friends!