The following are implementation notes for the Microsoft Matter Center. This assumes you are following the "Matter Center for Office 365: Build and Deployment Guide" that Microsoft has posted up on https://github.com/Microsoft/mattercenter The following tech notes "fill in" points and solves issues that commonly arise from the base implementation.

Each environment and implementation is different, so this isn't a comprehensive list nor an assurance that these "fixes" will solve all issues, but these have come up as common problems for deployment

Pre-Requisite Items

Office 365 /Azure Subscription in the Same Tenant: The O365 AND the Azure subscriptions need to be the same tenant where the default directory in Azure is based off a sync or integration with the Office 365 directory. While the Microsoft documentation seems to imply that the Azure tenant and O365 tenants can be different, there is a known issue where KeyVault integration can ONLY be done when the O365 credentials are in the Default Director of Azure. This is most commonly addressed by having the O365 credentials AS the Azure administration credentials, so there might be a workaround, but for cleaner implementation start with an O365 tenant, and create an Azure subscription off that O365 user account.

Renaming the Azure Subscription to a Simple Name: Once you have your Azure Subscription, you'll want to RENAME the name of the Subscription to something that is a single word and simple to key in during the deployment phase. Subscriptions are commonly "Free Trial" or "Visual Studio Trial" or "Company XYZ Enterprise Agreement" and when you get to the deployment script, it'll prompt you to "enter the name of your subscription". You cannot simply type in a multi-word subscription name that has spaces and complex characters without the script erroring (or at least I couldn't easily figure it out, including I tried to add in quotes, etc), so if you rename your Azure subscription to something simple like AZURE (a single word), it'll ensure the implementation goes smoother down the line. Also note that when you key in the subscription name, it is case sensitive, so whatever you name your subscription, make sure to note uppercase/lowercase in the name. To rename your subscription: Go into the Azure Portal (https://portal.azure.com), click on the "More Services" at the bottom left column of the Azure console page and then choose Subscriptions. of the landing page and choose subscriptions (in the General section). Click on the subscription you plan to use, and in the Overview you'll see the "Subscription name" in the far right pane. Click on the subscription and you'll be able to rename it. Change it to a single word, maybe all uppercase so something like AZURE, then click Save. This will be the "subscription name" you'll be asked for in the deployment script process later in the deployment.

Pre-requisite Software: The pre-req software required to install/configure Matter Center (ie: Visual Studio, Azure SDK, .Net Core, etc) are very accurate, if you don't have the pre-reqs installed, you won't get very far with the deployment. Some of the decision points that commonly pop up is Visual Studio 2015 or 2017?, which versions or service updates, etc. In my implementation, I successfully had this all work with Visual Studio 2015, although I'd guess VS2017 would work as well but figured VS215 was more likely better tested with this 2015-created solution. The biggie is that VS2015 needs Update 3 or higher to be able to install the Office Developer Tools and various add-ins. There were MANY to install between the .NET

Core and Azure SDK to get the thing updated and working. Probably 2 hours alone to get VS with all the needed patches and updates if you start with nothing. When you have the choice of which version of the add-ins (SDKs, etc) to plug into Visual Studio, it seemed like all of the latest versions worked fine, so no issue loading up the latest. And a good thing is that if you are missing something, the scripts are very resilient to allow you to rerun the scripts over (and over). The scripts look to see if you started the script, it'll delete things you might have started with or already configured, and then reinstall the components and information. So you can install, get an error that you're missing a component, you can add the component and then rerun the script again. The thing to note is every time you reinstall, it will (likely) delete anything created prior, so think of it as a clean uninstall and install, which is fine for a clean installation, but just be aware that this could very well be a complete wipe and reinstall, thus the recommendation to build this for the first time in a test environment.

Pre-Deployment Items

Activating SharePoint Server Publishing Infrastructure Service: The deployment will activate the SharePoint Server Publishing Infrastructure Service for each Team Site, however in order for that functionality to work, this feature needs to be activated on the ROOT of your SharePoint Online tenant otherwise that sub-piece of the script will fail. So before you get too far, with SharePoint Online running, go to your https://<tenantname>.sharepoint.com, click on the Cog (in the upper right corner), click Settings, Site Collection Features, then scroll down to the SharePoint Server Publishing Infrastructure, and click to Activate

Deployment Items

Office 365 / Azure Same vs Different Tenant Implementation: During the deployment process, there will be a section of ADDING APPLICATIONS to Azure to cross-integrate Office 365 (namely Microsoft Graph, Microsoft SharePoint Online, and Azure Active Directory) between the Office 365 tenant and the Azure tenant. The documentation gives you the option of integrating with O365/Azure as the same tenant or different tenant. As I noted in the Pre-requisite notes, you'll have challenges trying to get this all work if your O365 tenant and your Azure tenant are different. It has to do with a piece on Azure KeyVault exchange where KeyVault integration of Office 365 presumes the Default Directory in Azure is your Office 365 tenant. So as you get to the application integration section, the presumption is your Azure tenant default directory IS your Office 365 user directory and is the Azure default.

Compiling and Deploying in Visual Studio: In the deployment process, you'll need to work in Visual Studio. The deployment guide quickly refers you to see general Microsoft Visual Studio documentation. For those who aren't completely familiar with Visual Studio, effectively you are going into Visual Studio. The first step is to "create build of the helper utilities", you will File/Open the "Microsoft.Legal.MatterCenter.HelperUtilities.sln" file in the tree\master\cloud\src\Helper Utilities\ Microsoft.Legal.MatterCenter.HelperUtilities folder where you unzipped the Matter Center files. The "building" process turns the code into EXE files that can be run. Microsoft doesn't ship EXE files, it ships solutions that can be compiled into EXE files, so with this Helper Utilities sln loaded, right-click on the solution (in the right hand (Solution Explorer) pane) and select BUILD. This runs through and builds a bunch of EXE files. This should complete without error. If you get errors (that'll show in the bottom Output pane),

usually you'll find you're missing some SDK or plug-in that was noted in the pre-req (that you might have missed installing). Download/install what is being requested, then do the BUILD again.

Later in the deployment steps (Deployment around Step 11), you'll be asked to Deploy the Web to Azure. In this part of using Visual Studio, you'll open the "Microsoft.Legal.MatterCenter.sln" that'll be on the (likely C>) of your computer in the tree\master\cloud\src\Solution\Deployment folder. When you load this solution, it'll go through a series of steps to "load", then "initialize" the various solution components. There's around 11 components that'll process and take a minute or so to process. Once all of the components have initialized (which you can see in the right hand pane) you can see if there are any errors in the Output pane at the bottom of Visual Studio. There are a few errors that'll pop up like the system won't be able to connect to the Team Foundation Server for change controls (if you're not using TFS change control by default, it'll look for a TFS server, but this error has no impact on the deployment of this service). If there are obvious errors, again it might note you need to install a .NET Core service, or some other plug-in. If there are no obvious errors, then you can deploy this Website.

First thing to do is make sure you are logged into your Azure account that you want this service deployed to. In Visual Studio, click File / Account and either logon or "add account" and type in your Office 365 (Azure) logon/password to load up your Azure subscription.

In Visual Studio, scroll down in the right pane (Solution Explorer) in Visual Studio to the Matter Center Website component, right click that component and select Deploy. You will be prompted where you want to deploy this Website. Select Microsoft Azure Web Apps (making sure that your Azure logon and subscription is the one being selected). With the proper subscription to Azure selected, you'll see the MatterCenterWeb that was created earlier in this process show up on the screen. Drill down 1 level so that you see the MatterCenterWeb "website" noted (so this push will layer on TOP of the existing Azure WebApp Website that you created earlier (which was the step that created the KeyVault keys, and created the security permissions to Office 365). From there, you can Next/Next, Publish using the defaults that'll push up to Azure.

Upon completion of the push (which can take 10-15 minutes), Visual Studio will "display your website" but if you logon at that point, it'll error. The reason is Visual Studio automatically launches a browser session using http:// (unsecured). You'll want to https:// (SSL secured) to the same URL and THAT will log you in properly to the Website. Continue with the first step of the Web instructions where you add /#/admin to the end of your website URL to wrap up the installation. Once you complete this /#/admin step, then you can https:// to your MatterCenter for use!

Post-Deployment Items

Integrating SharePoint Search to the Matter Center Site: You may find that while you get to the Matter Center website, when you click on MATTERS or DOCUMENTS the website just sits and you get a "wait" symbol, nothing happens. This is commonly solved by making sure your Azure Website has the SharePoint search string embedded. A tech note on this issue is referenced on the Matter Center GitHub issues tab: https://github.com/Microsoft/mattercenter/issues/642

The fix is to effectively go to https://<YourTenant>.sharepoint.com/_layouts/15/manageresultsources.aspx?level=sitecol and click on mattercenter. Then from that URL that you'll be taken to, at the end of the URL in your browser URL line you'll find a Source ID (a bunch of letters/numbers). Copy out that Source ID and replace the 3 instances of %2D in sourceid value with a hyphen. So effectively something like:

sourceid=9bc38fc9a%2Daf85%2D 373c-b78f%2Dca5cee7e79f9

And replacing the %2D with - it'll end up with your real Source ID is:

9bc38fc9a-af85-373c-b78f-ca5cee7e79f9

Now go to your Azure Portal (https://portal.azure.com), click on App Service in the left console menu, select your Matter Center App Service, under Settings / Application Settings there a thing called Search:SearchResultSourceID that you might have System.Object[] in that field. Delete that "System.Object[]" and replace it with the String you derived above (with the hyphens), click on SAVE.

When you go back and refresh your Matter Center page, you should be able to drill down into your Matter Center MATTERS and DOCUMENTS.

Matter Center Outlook Add-in Not Working: One of the nice features in Matter Center is the ability to leverage the Outlook add-in that pops Matter Center up within Outlook so that users can open and save content straight into Matters / Matter document libraries right from Outlook. What you might find is the Outlook Add-in works in your initial user/administrator you created, but the add-in doesn't show up on other users. This is because by default the Outlook Add-in is enabled FOR the "selected user" that you typed into the configuration spreadsheet (ie: your administration account, only).

Few things you need to check:

- Make sure your users have access to the Site Collections (Team Sites) with Owner or User rights to access the sites
- Also make sure the users have access to the SharePoint "Catalog" you created earlier (ie: https://<tenantname>.sharepoint.com/sites/catalog)

Assuming you don't have a "rigths" issue from the two above, go into the Office 365 Admin console (https://portal.office365.com), in the left column scroll down to Admin then select Exchange / Organization / Add-in and you'll likely see the Matter Center add-in is only applied to "specific users" . You'll want to change that to "everyone". To do so, do the following:

- On a workstation, Run remote Powershell to logon to Exchange Online (effectively on a workstation, run PowerShell as an Administrator. In Powershell, type the following…)
  - $Cred = Get-Credential
  - $s = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri https://ps.outlook.com/powershell -Credential $cred -Authentication Basic –AllowRedirection
  - Import-PSSession $s
  - Get-app -Organizationapp | Format-List DisplayName,appid

(grab the GUID from the above "Get-app" command associated with Matter Center, it'll look like a GUID like xxxxxxxx-xxxx-xxxx-xxxxxxxxx)

- set-app xxxxxxxx-xxxx-xxxx-xxxxxxxxx -organizationapp -providedto everyone

Once you do that, you'll be able to logon to any of the Matter Center users and you can click on the Cog (in the upper right of the Outlook browser), Settings, Mail, General/Manage Add-ins, and the Matter Center should be enabled.

Deleting Matter Center Install and Starting Again: You may find that after you installed Matter Center and fiddled with it, you now want to "start again" or you might find that something isn't quite working and you want to try the install again. You can do so by effectively walking through the install process all over again. The thing(s) you'll need to delete on the Azure/SharePoint site is effectively the Azure WebApp that was created so you don't confuse the old WebApp with the new one.

In the New Azure Portal (https://portal.azure.com), delete the Resource Group associated with the old portal/Web (this deletes like 7 Azure components like the Web, Insights, Storage, etc)

In the Old Windows Azure console (https://manage.windowsazure.com), under Active Directory, Apps, Apps my company uses, delete the MatterCenterWeb.

You don't need to go into Office 365 and delete SharePoint content, Term Store configurations, or other stuff. The Matter Center deployment script will clear up previous SharePoint and Office 365 components. You will notice that some stuff will continue on (like Team sites you might have created, and Matters/Documents you might have uploaded), but everything else gets cleaned up and starts over.

Support in Matter Center GitHub: If you are still experiencing problems, you can go into the GitHub MatterCenter "Issues" tab https://github.com/Microsoft/mattercenter/issues where you'll find previous questions (and answers) in both your Open and Closed sections.

Matter Center Administrative Tasks

Adding Users/Attorneys to Matter Center: When working with Matter Center, you will no doubt want to add new users/attorneys to Matter Center. To do so, presuming you have created the User to Office 365 as a normal email users, then all you need to do is associate that user to Matter Center. To do so:

- Go into SharePoint Site Collections (https://<TenantName-admin.sharepoint.com), click on the Site Collections in the left column, then for each client that the person should have access to, click on the Team site, Click on the Cog in the Upper Right of the Team Site page, select Site Permissions, and add the user to the Matter Center Users group

Adding Customers: To add new customers to Matter Center, do the following:
Go into SharePoint Site Collections (Portal / Admin / SharePoint Admin), click on the Term Store in the left column, expand the Matter Center Terms, Click on ClientID and Create Term a new # that'll be the next incremented up (ie: 100015). Then click on the Clients and "Create

Term". Enter in the name of the client and on the right hand pane click on Custom Properties tab, add in ClientID (and give it a number you just created under ClientID), and add in ClientURL and give it the URL for the Site Collection, Click Save

Now click on the Site Collections (still in the Portal/Admin/SharePoint Admin) and click to create a NEW Site collection.

- Give it a name of the client
- Select \TEAMS\ (as opposed to \sites\) and enter in the Client name you created above
- Make it a Team Site
- Once the site is created, add users to have rights to this Client

Still in the SharePoint Site Collections (Portal / Admin / SharePoint Admin), click on the Team site you just created, Click on the Cog (upper right), Site Settings, Site Permissions, and add the new user to the Owners group (ie: to the Company123 Owners group)
Now you STILL have to create this Matters file for this, so:

- Still in the SharePoint Site Collection for this newly created Site, click on the "Site Contents" in the lower left, select to use Classic Sharepoint, click on Add an App, Select Custom List
- Give it the name Matter Configurations
- Click on the newly created Matter Configurations app
- Click on the List at the top, click on Create Column, For Column Name call it ConfigurationValue (no space there) and select MultiLines of Text, Click OK
- Click on List Settings (in the upper right), click on Permissions for List
- Click on "Stop Inheriting" in the upper toolbar, then OK
- Select all groups (EXCEPT the Owners group), and click on Edit Permissions in the Tool Bar, choose Read, then OK