Open edX on Azure - Ficus Stamp Deployment

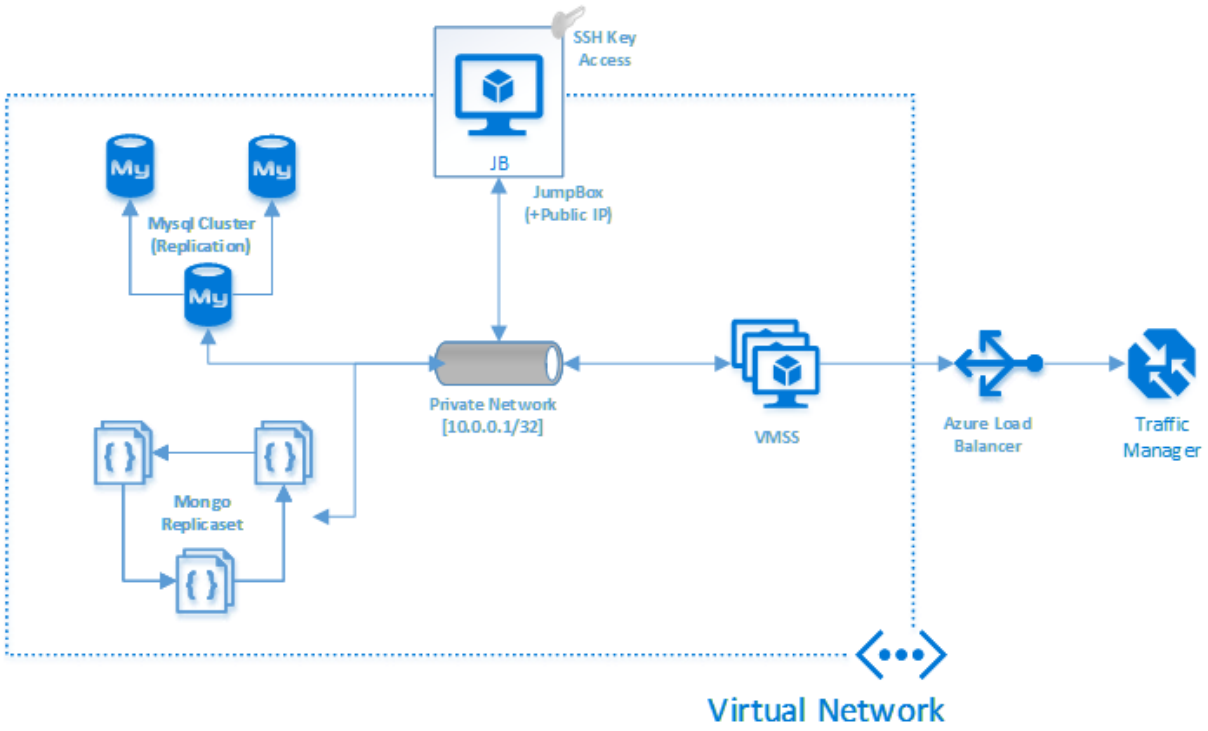
Table of Contents

Welcome to the Open edX Planning and deployment guide for the Open edx "Ficus" edition on Azure ™

This document will help you

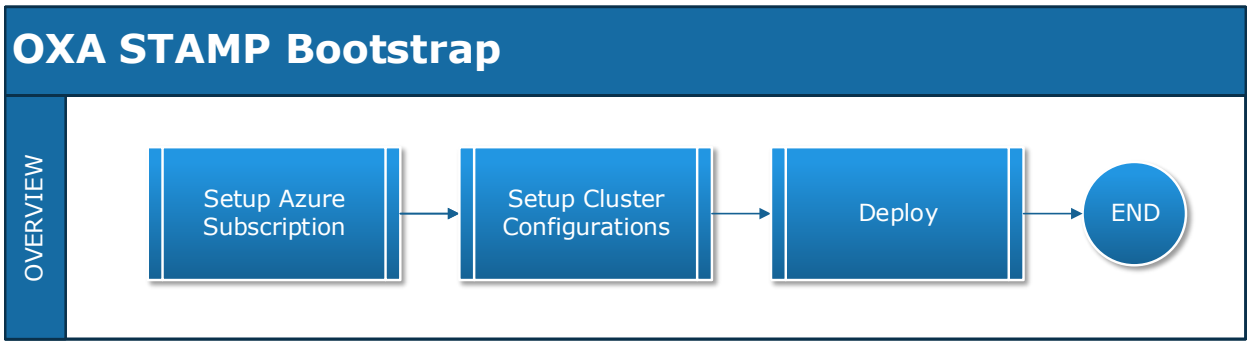
- Understand pre-requisites
- Plan your deployment
- Execute the deployment steps
- Validate the deployment

Architecture for Open edX (STAMP) deployment on Azure



Process Overview

The stamp deployment of Openedx Ficus release is a 3-step process as shown below:



Step 1: Pre-Requisites

1.1. You need an azure subscription to work against for this installation

1.2. Ensure that azure-cli is installed (You can install it from https://go.microsoft.com/?linkid=9828653)

1.3. Ensure Azure Powershell Cmdlets are installed (https://docs.microsoft.com/en-us/powershell/azureps-cmdlets-docs/)

1.3.1. You may need to change the default execution policy on your machine

```
Set-ExecutionPolicy -ExecutionPolicy Bypass
```

1.3.2. *Install-Module AzureRM* (Installs the Azure Resource Manager modules)

1.4. Install bash

1.4.1. Install Ubuntu bash on your Windows 10 machine
https://msdn.microsoft.com/en-us/commandline/wsl/install_guide

(or)

1.4.2. Download and Install Git Bash for Windows
https://git-scm.com/download/win

1.5. Create SSL certificate and key files

1.5.1. We have included sample certificates in the default configuration folder https://github.com/Microsoft/oxa-tools/tree/oxa/master.fic/config/stamp/default. Please create your own public and private keys.

1.5.2. Generate your certificates for SSL (via a cert authority)

- Access openssl via the command openssl on Ubuntu bash or Git bash
- Create the .key and .crt file (via openssl)

1. Export the private key:
openssl pkcs12 -in [PATH-TO-PFX] -nocerts -out ~/key.pem -nodes

2. Export the certificate:
openssl pkcs12 -in [PATH-TO-PFX] -nokeys -out ~cert.crt

3. Remove the passphrase from the private key:
openssl rsa -in ~/key.pem -out ~/cert.key

4. Copy the cert.crt and cert.key to the [configuration folder]

Step 2: Plan the Deployment

To prepare your cluster configuration, you must first understand the stamp architecture as shown above.

2.1. Next you need to identify the clouds you want to deploy to e.g. bvt, int or prod. For the purpose of this documentation we will go with bvt environment.

2.2. Sync the configuration files from the repository https://github.com/Microsoft/oxa-tools/tree/oxa/master.fic

2.2.1. From your Bash console (Git Bash or Ubuntu Bash), run the following command
git clone -b oxa/master.fic https://github.com/Microsoft/oxa-tools.git

2.3. Files that will be part of **oxa-tools\config\stamp\default** folder:

2.3.1. bvt.sh

- Name it after the cloud you are deploying to; bvt in this case
- Make sure the file has unix line ending

2.3.2. id_rsa/id_rsa.pub

2.3.2.1. We have provided sample keys. However, you need to create your own public and private keys. These keys provide front door access to the jumpbox; please change them.

2.3.2.2. Run the command ssh-keygen on your bash command prompt

- **Command**: *ssh-keygen -b 4096 -t rsa*
- Specify the file in which to save the key
- Do not specify any passphrase for the keys

2.3.2.3. **Copy these files** over to the folder where you have the bvt.sh file. The key is required to access the jumpbox; key file will be associated with the admin user of the jumpbox to give necessary permissions

2.3.3. cert.crt/cert.key files

- Copy cert.crt and cert.key files generated at pre-requisite (section 1.5) over to the folder where you have the bvt.sh file.

2.3.4. parameters.json

2.3.4.1. This contains the stamp configuration parameters; each parameter is defined in the template file

https://github.com/Microsoft/oxa-tools/blob/oxa/master.fic/config/stamp/default/parameters.json

2.3.4.2. You may want to change the SKU of the VMs as per Azure cost for the resources

Step 3: Execute the Deployment Steps

3.1. Create AAD client and grant permissions to your subscription; this same web client can be used for OAuth. See appendix A for instructions if you want to create an AAD client.

3.2. Run the command below from Powershell command (with admin privileges)

3.2.1. Run command "Login-AzureRmAccount"

- This would open up a browser window requesting you to login to Azure.
- After successful login, you will be returned to the PowerShell window which would display your Account, Tenant Id (used as AAD Tenant Id below), and default subscription details.

3.2.2. Run "Get-AzureRmSubscription" to view all Azure subscriptions for your account.

3.2.3. Construct the following command by replacing the highlighted tags with appropriate values and then execute.

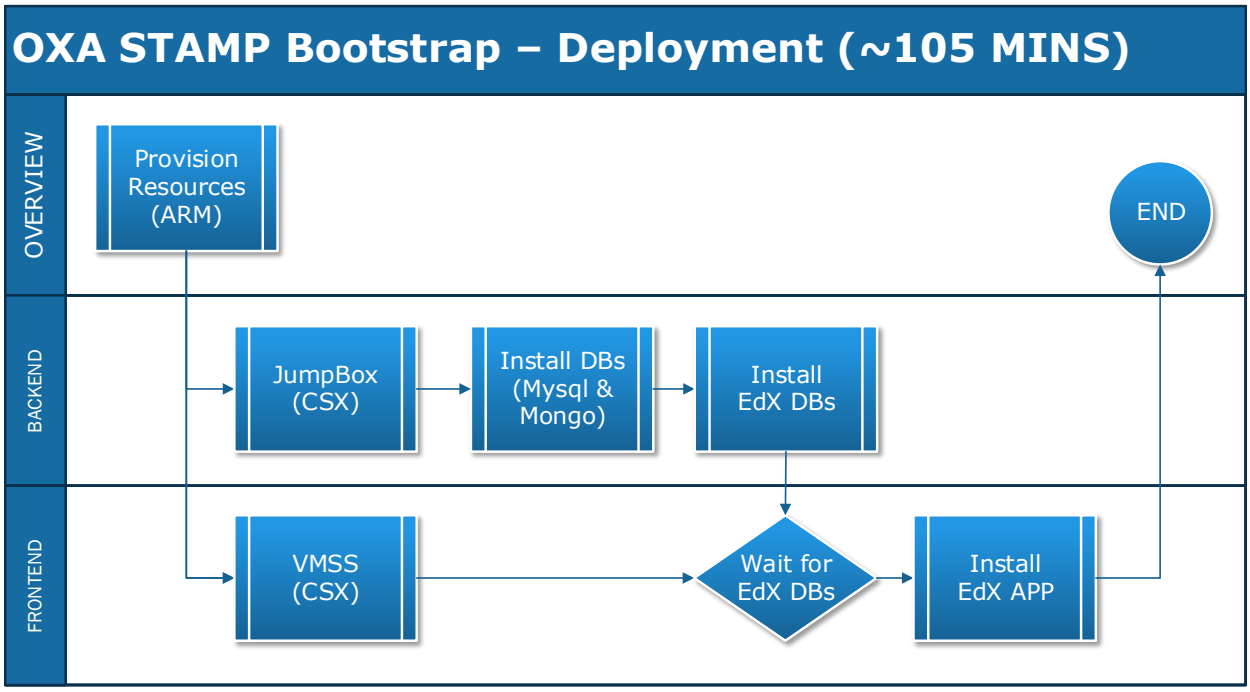
[Enlistment Root]\oxa-tools\scripts\Deploy-OxaStamp.ps1 -AzureSubscriptionName [Subscription Name] -ResourceGroupName [Cluster Name] -Location "central us" -TargetPath "[Enlistment Root]\oxa-tools\config\stamp\default" -AadWebClientId <AAD web client ID from Azure> -AadWebClientAppKey <AAD web client app key from Azure> -AadTenantId <AAD tenant id> -KeyVaultDeploymentArmTemplateFile "[Enlistment Root]\oxa-tools\templates\stamp\stamp-keyvault.json" -FullDeploymentParametersFile "[Enlistment Root]\oxa-tools\config\stamp\default\parameters.json" -FullDeploymentArmTemplateFile "[Enlistment Root]\oxa-

tools\templates\stamp\stamp-v2.json" -ClusterAdministratorEmailAddress [Your Email Address] -SmtpServer <SMTP server name> -SmtpServerPort <SMTP server port> -SmtpAuthenticationUser *<SMTP auth user> -SmtpAuthenticationUserPassword <SMTP auth user password> -ServiceAccountPassword* <Service Account Password> *-EnableMobileRestApi*

Notes about the highlighted tags:

- [Enlistment Root] = root of your local git hub repositories
- [Subscription Name] = Name of your azure subscription
- [Cluster Name] = unique cluster name created on Azure (limit to 8-10 characters)
- *<AAD web client ID from Azure>* - Your AAD Web Client Id
- *<AAD web client app key from Azure>* - Your AAD Web Client Id
- *<AAD tenant id>* - Tenant Id from section 3.2.1
- *[Your Email Address]* – Your/Admin email address
- *<SMTP server name>* - SMTP Server Name
- *<SMTP server port>* - SMTP Server Port
- *<SMTP auth user>* - SMTP auth User
- *<SMTP auth user password>* - SMTP auth Password

3.2.4. The deployment is a two-step process:



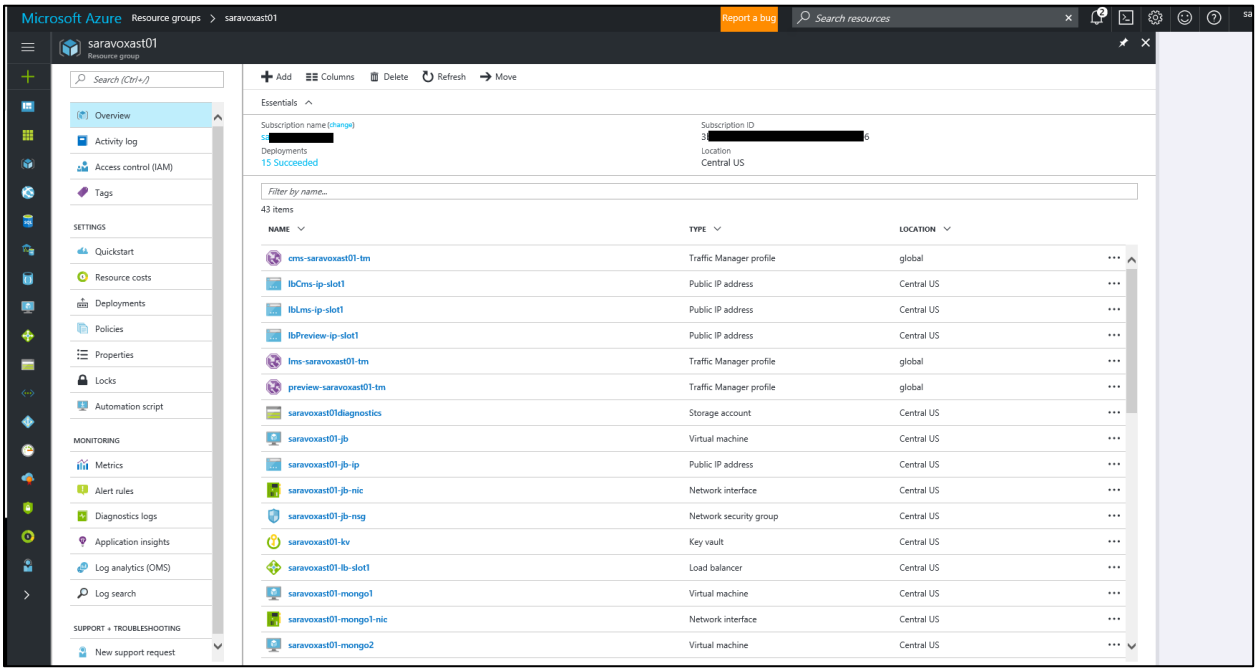
i. Provisioning of resources takes around 15 minutes
ii. Deploying the bits to the stamp configuration takes about 1.5 hrs
    - Email is generated at regular intervals of the process
        - Start of installation of edx app (vmss)
        - Installation and configuration of backend database applications (mysql and mongo)
        - Installation of EDX database
        - Completion of installation of edx app (vmss)

3.2.5. Completion and Testing

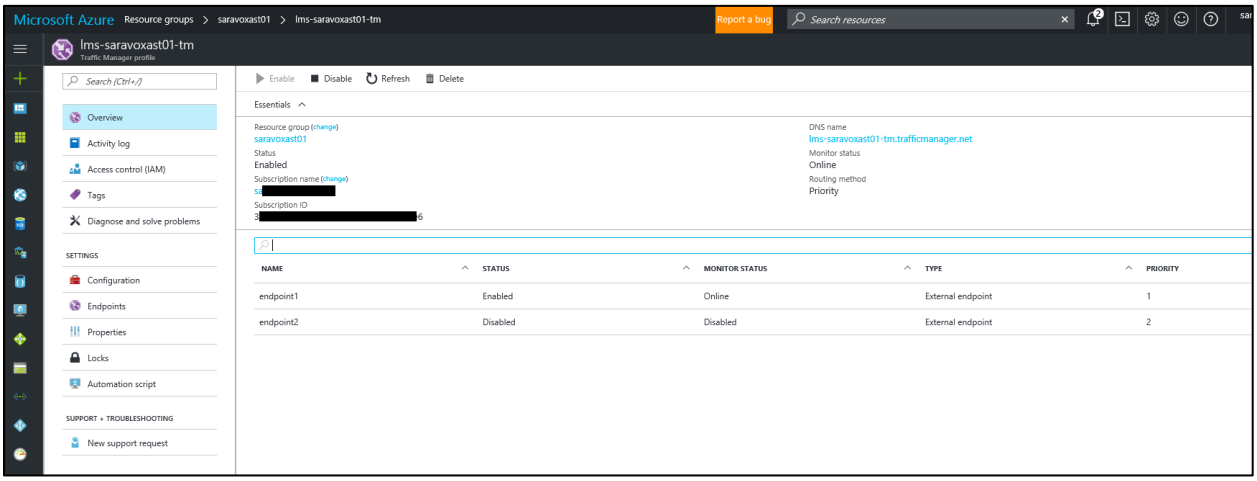
iii. Once the deployment is complete, you can access the LMS and CMS

iv. The URLs would look similar to this.
https://lms-[Cluster Name]-tm.trafficmanager.net
https://cms-[Cluster Name]-tm.trafficmanager.net

v. You can also access it from the Azure portal; check the resources under type "Traffic Manager profile"



- Click the lms and cms resources to get the details of the DNS name



If you can access the LMS and CMS, the installation is successful.

Please follow the Program Guide for post deployment work to enable oAuth, importing content, and etc.

APPENDIX

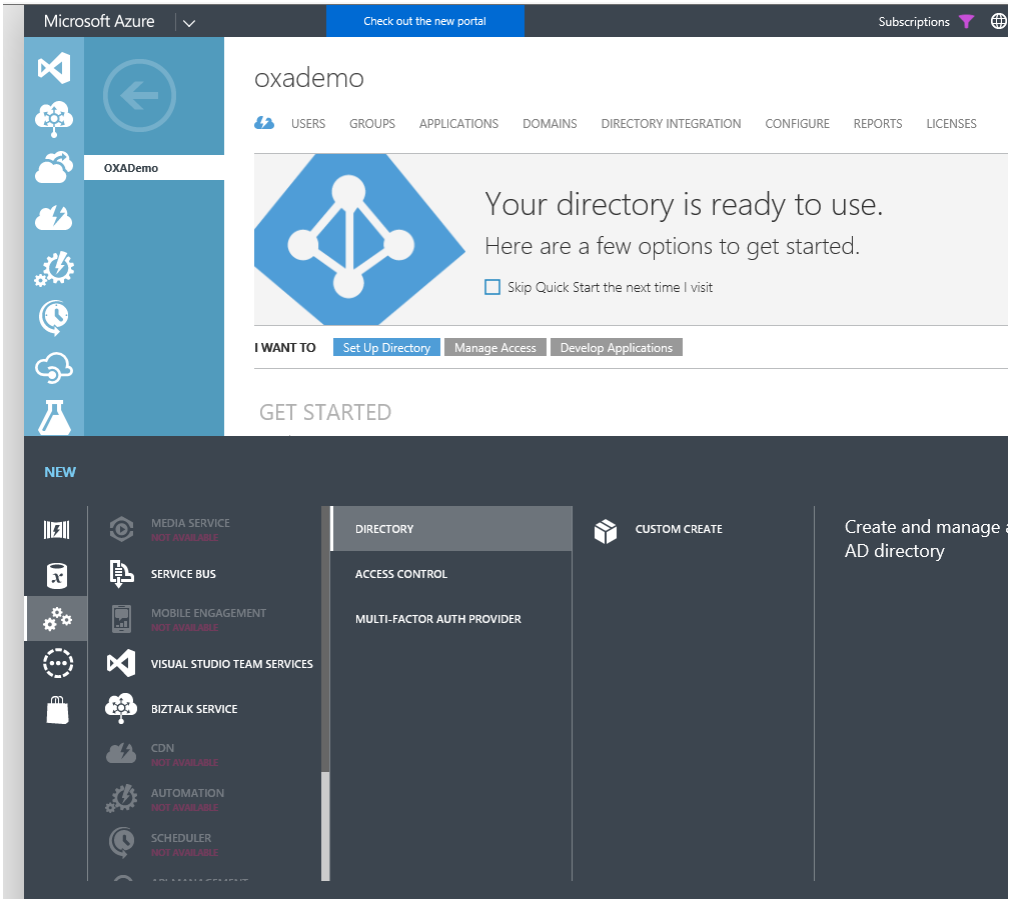
APPENDIX A

a) Resources

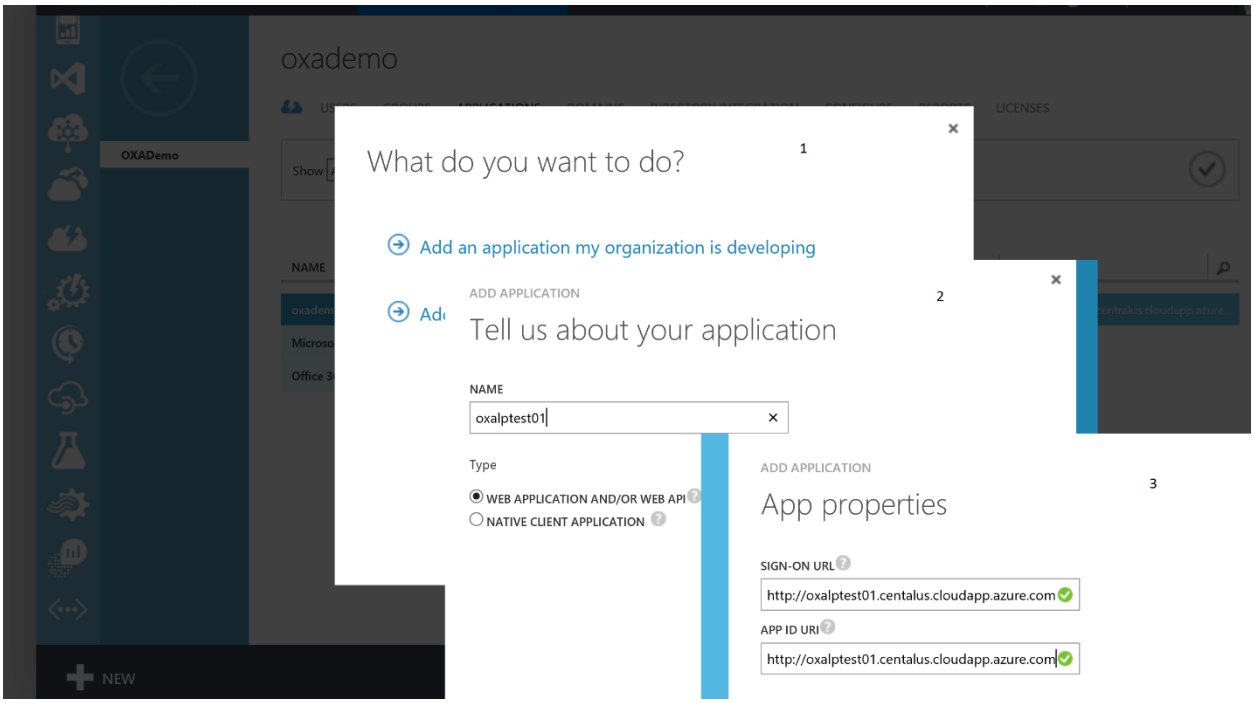
Resources	URL / Link	Remarks
Azure-CLI	https://go.microsoft.com/?linkid=9828653	Azure Command Line Interface
Azure Powershell Cmdlets	https://docs.microsoft.com/en-us/powershell/azureps-cmdlets-docs/	Azure Power shell
Open edX Configuration Files	https://github.com/Microsoft/oxa-tools/tree/oxa/master.fic.eltonc	Open edX "Ficus" Build Configuration Files
Ubuntu Shell (BASH)	https://msdn.microsoft.com/en-us/commandline/wsl/install_guide	Ubuntu Shell
Old Azure Portal	https://manage.windowsazure.com	Legacy Azure portal
New Azure Portal	https://ms.portal.azure.com	New Azure portal

b) Create an Azure Active Diretory (Skip this step if you already have an Active Directory)

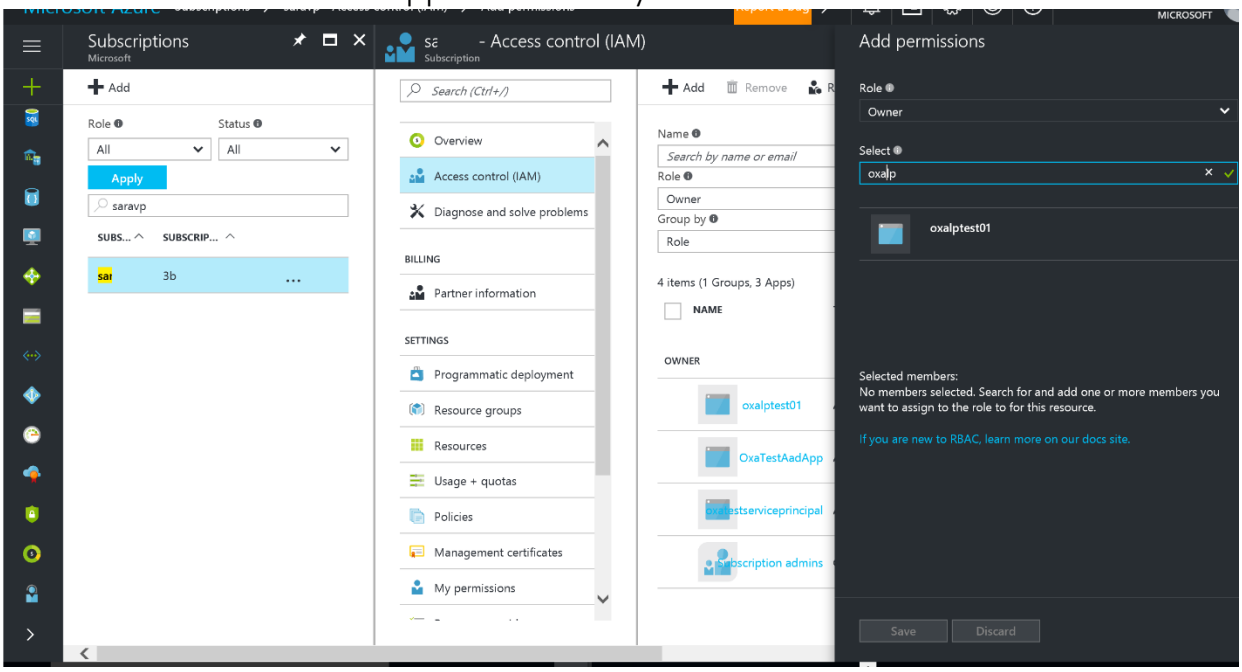
i. Access the classic Azure portal https://manage.windowsazure.com, under Active Directory create a new Directory by clicking on the "New"


Microsoft Azure
Check out the new portal
Subscriptions
OXADemo
oxademo
USERS
GROUPS
APPLICATIONS
DOMAINS
DIRECTORY INTEGRATION
CONFIGURE
REPORTS
LICENSES
Your directory is ready to use.
Here are a few options to get started.
Skip Quick Start the next time I visit
I WANT TO
Set Up Directory
Manage Access
Develop Applications
GET STARTED
NEW
MEDIA SERVICE
NOT AVAILABLE
SERVICE BUS
MOBILE ENGAGEMENT
NOT AVAILABLE
VISUAL STUDIO TEAM SERVICES
BIZTALK SERVICE
CDN
NOT AVAILABLE
AUTOMATION
NOT AVAILABLE
SCHEDULER
NOT AVAILABLE
DIRECTORY
ACCESS CONTROL
MULTI-FACTOR AUTH PROVIDER
CUSTOM CREATE
Create and manage
AD directory

c) Add an application by clicking on the +Add

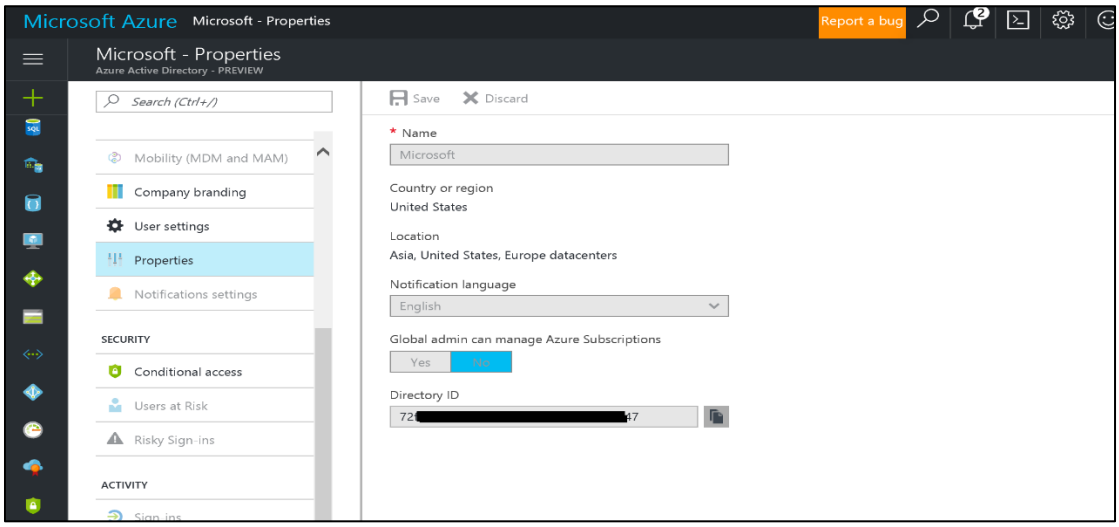


d) On the new Azure portal https://ms.portal.azure.com, select your subscription, under Access Control (IAM), "+Add" Permissions to grant "Owner" access to the application that you have created

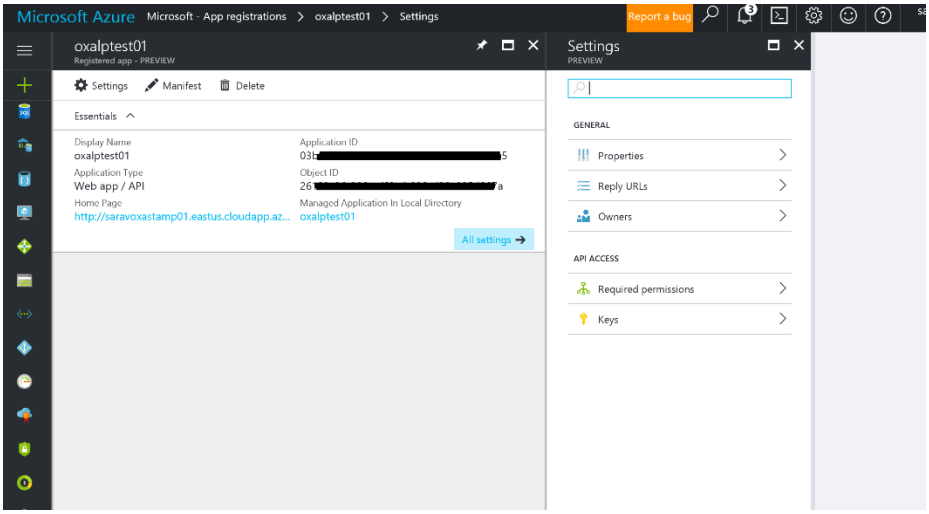


e) Steps to get values for AadTenantId , AadWebClientId, and AadWebClientAppKey

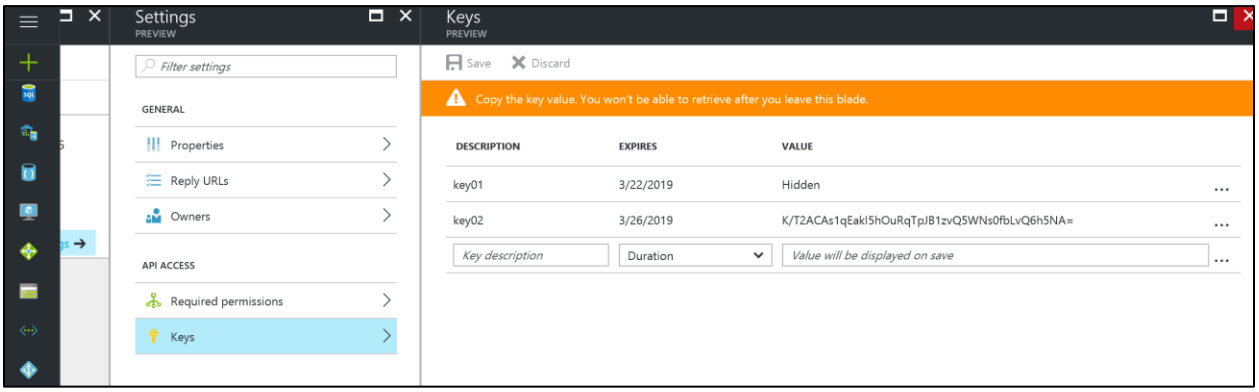
i. *AADTenantId can be fetched from section 3.2.1. You can also get this value from the portal under properties of your active directory. It's called "Directory ID"*



ii. For AADWebClientId, under App Registrations, select your application. You can find Applcation Id. It's the value for AADWebClientId.



iii. For AadWebClientAppKey, generate a Key by providing Key Desciritption and Duration. When you save, it will automatically generate a Key which will be displayed in the UI. Copy this value and store it securely. When you visit keys next time, this value will be hidden. You will not be able to access it unless you have stored it somewhere safe. But, you can create a new key.



APPENDIX B

FAQ Contents

Based on feedback from partners and our own testing, we have made the following changes to deployment guide and process

1. Deployment status emails are not working:

We have worked on this issue and verified that Office 365 emails and gmail smtp settings are working now. Pls follow the following guidance for configuring the email to receive deployment notifications.

Third-Party SMTP Relay - Gmail & O365

During the STAMP deployment, we allow users to provide an SMTP relay that will allow them to relay deployment notification and other system emails to the cluster administrative user(s). It has come to attention that this doesn't work well with third-party email providers like Google or Outlook/O365. Therefore, we have made additional updates to support two providers: Gmail & O365.

There are five (5) deployment email parameters (see STEP 3 above):

- *ClusterAdministratorEmailAddress* – this is any address or distribution list where you'd like all notification emails will be sent
- *SmtpServer* – this is the SMTP server fully qualified address
- *SmtpServerPort* – this is the communication port on the SMTP server specified above
- *SmtpAuthenticationUser* – this is the user name to authenticate with on the SMTP server specified above
- *SmtpAuthenticationUserPassword* – this is the corresponding password for authentication

Server Settings

The SmtpServer & SmtpServerPort details for Gmail can be found here:

https://support.office.com/en-us/article/POP-and-IMAP-settings-for-Outlook-Office-365-for-business-7fc677eb-2491-4cbc-8153-8e7113525f6c

(see the "POP and IMAP settings for Office 365 for business email" section)

The SmtpServer & SmtpServerPort details for GMAIL can be found here:

https://support.google.com/a/answer/176600?hl=en

(see the "Use the Gmail SMTP Server" section)

User Credentials

The *SmtpAuthenticationUser* is typically the email address of the account with SMTP relay access. This applies to both Gmail and O365.

The SmtpAuthenticationUserPassword:

for O365, this is the password of the account with SMTP relay access.

for Gmail, this is an application password (see more details below)

Additional Details for Gmail

If you are using Gmail, the password for the email address you are using for SmtpAuthenticationUser will not work. You must instead create and use an App Password that is associated with the email address. Creating this application password has a pre-requisite: your account must have 2-Step Verification enabled. Here's how to configure your account:

Enable 2-Step Verification (pre-requisite): https://support.google.com/accounts/answer/185839?hl=en

Create an App Password: https://support.google.com/accounts/answer/185833?hl=en (see the "How to generate an App password" section)

Examples

O365: I have an office 365 account oxa-admin@contoso.com. To login to this account, I use the following password: 123@contoso_com. I'd however like to send all notifications to oxanotifications@contoso.com which is a distribution list to my engineering team.

My OXA deployment email parameters would be:

-ClusterAdministratorEmailAddress oxanotifications@contoso.com -SmtpServer "smtp.office365.com" -SmtpServerPort 587 -SmtpAuthenticationUser "oxa-admin@contoso.com" -SmtpAuthenticationUserPassword "123@contoso_com"

Gmail: I have a Gmail account oxa-admin-team1@gmail.com. To login to this account, I use the following password: 123@contoso_com. I want to send all notifications to oxanotifications-team1@gmail.com. I also need a separate App password which I generated as eekqiutsqrvliube under my "oxa-admin-team1@gmail.com" account.

My OXA deployment email parameters would be:

-ClusterAdministratorEmailAddress oxanotifications-team1@gmail.com -SmtpServer "smtp.gmail.com" -SmtpServerPort 587 -SmtpAuthenticationUser "oxa-admin-team1@gmail.com" -SmtpAuthenticationUserPassword "eekqiutsqrvliube"

2. Core quota limits have exceeded

Message=Operation results in exceeding quota limits of Core. Maximum allowed: 10, Current in use: 5, Additional requested: 12.

This error typically is shown if your subscription doesn't have capacity support enough cores. You should file a ticket with Azure to increase more VM Capacity (cores) to your subscription.

3. How to access the VMs after deployment

Accessing the VMs is done via SSH. There is only one entry point and that is the jumpbox.

It is assumed you have logged into the azure portal (portal.azure.com) and selected your target azure subscription.

Here's how to proceed:

1. From the azure portal, click on resource groups icon and select the resource group you created as part of the bootstrap. It will be the name of your cluster ([Cluster Name] deployment variable).
2. From within the list of resources, search for "**jb**".
3. The search should return a list of resources associated with your jumpbox.
4. Click on the resource named "[**Cluster Name**]**-jb-ip**" and copy the value of its DNS Name.

5. From your bash console type the following:
    a. **ssh [the admin user name from your parameters.json file]@[domain name of your jumpbox] -i [path to your ssh private key that was generated in Step 2.3.2]**
6. This should log you into the jumpbox

Once you have access to the jump box, all other servers will be available via the private network. If you'd like to access a specific machine, do the following:

- From the azure portal, click on resource groups icon and select the resource group you created as part of the bootstrap. It will be the name of your cluster ([Cluster Name] deployment variable).
- From within the list of resources, search for "**vnet**".
- The search should return the Virtual Network Resource named "[Cluster Name]-vnet"
- Click on the Virtual Network Resource. It should list all network interfaces (NICs) associated with all resources connected to your virtual network. These are private ip addresses. For the lms/cms frontend, the resource will be named like **"[Cluster Name]-vmss-[deploymentVersionId from your parameters.json file]"**

- Once you determine which NIC you'd like to connect to, do the following:
    - **ssh [IP Address]**

      where [**IP Address**] is the private ip address of the NIC associated with server you'd like to connect to.

4. I am seeing degrading status on the VMs in the Azure portal

This typically means something went wrong with the deployment. The only way to know the details of error is to have correct email configuration where you will see notifications and details of failed deployments. Please see #1 item in this document on configuring emails correctly

5. Where do I specify the service passwords?

We have added an additional parameter to command line section 3.2.3 to specify the service account password. Please make sure that this password doesn't have any non-alpha numeric characters. Mongo DB has some restrictions. The default password we used earlier has been changed to take this into account.

[Enlistment Root]\oxa-tools\scripts\Deploy-OxaStamp.ps1 -AzureSubscriptionName [Subscription Name] -ResourceGroupName [Cluster Name] -Location "central us" -TargetPath "[Enlistment Root]\oxa-

tools\config\stamp\default" -AadWebClientId <AAD web client ID from Azure> -AadWebClientAppKey <AAD web client app key from Azure> -AadTenantId <AAD tenant id> -KeyVaultDeploymentArmTemplateFile "[Enlistment Root]\oxa-tools\templates\stamp\stamp-keyvault.json" -FullDeploymentParametersFile "[Enlistment Root]\oxa-tools\config\stamp\default\parameters.json" -FullDeploymentArmTemplateFile "[Enlistment Root]\oxa-tools\templates\stamp\stamp-v2.json" -ClusterAdministratorEmailAddress [Your Email Address] -SmtpServer <SMTP server name> -SmtpServerPort <SMTP server port> -SmtpAuthenticationUser <SMTP auth user> -SmtpAuthenticationUserPassword <SMTP auth user password> -ServiceAccountPassword <Service Account Password>

6. What updates effect installations prior to July 2017?

Open edX deployments prior to July 7, 2017 need few configuration updates to have end-to-end LaaS flow working. **The below changes are ONLY to be used if you already have Open edX running with users taking courses.**

Sync the configuration files from the repository to new local folder. This path will become your [Enlistment Root]

https://github.com/Microsoft/oxa-tools/tree/oxa/master.fic

From your Bash console (Git Bash or Ubuntu Bash), run the following commands:

git clone -b oxa/master.fic https://github.com/Microsoft/oxa-tools.git

Run an update script which sets up right configurations.

From a powershell session in admin mode, execute the following commands:

Note: Replace all the highlighted parameters with your own settings. Then run the following commands. It will approximately take 2-5 minutes for these commands to run and update the settings.

```
[array]$upgradeParameters = @( @{"name"="target-user"; "value"="[the adminUsername from your parameters.json file]"}, @{"name"="cluster-admin-email"; "value"="[Your Email Address]"})

[Enlistment Root]\scripts\Deploy-CustomScriptsExtension-v2.ps1 -AzureSubscriptionName [Subscription Name] -ResourceGroupName [Cluster Name] -AadWebClientId "[AAD web client ID]" -AadWebClientAppKey "[AAD web client app key]" -AadTenantId "[AAD tenant id]" -TemplateFile "[Enlistment Root]\templates\stamp\stamp-v2-backend-upgrade.json" -TemplateParameterFile "[Enlistment Root]\templates\stamp\stamp-v2-backend-upgrade-parameters.json" -ClusterAdmininistratorEmailAddress [Your Email Address] -InstallerPackageName "enablemobileapi" -UpgradeParameters $upgradeParameters
```

where:

1. [Your Email Address] The email address is used to send notifications regarding the update failures. If deployment succeeds you will not receive any emails
2. [OS User Account]: the existing operating system user account whose authorized key you want to rotate
3. [Path to SSH Public Key]: the full path to the replacement public key
4. [Enlistment Root]: location where the oxa-tools repository was cloned
5. [Subscription Name] = Name of your azure subscription
6. [Cluster Name] = name of the existing azure STAMP cluster/ resource group you intend to update
7. [AAD web client ID] - Your AAD Web Client Id
8. [AAD web client app key] - Your AAD Web Client Id
9. [AAD tenant id] - Tenant Id of the AAD entity in which you have the web client
10. [Your Email Address] - Your/Admin email address

Once these commands are executed, the configurations on your VMs will be updated and your end-to-end integration with academy.microsoft.com will work.