

## ACIT 4850 – Lab 1 – Source Code Management

<b>Instructor</b>	Mike Mulder ( <a href="mailto:mmulder10@bcit.ca">mmulder10@bcit.ca</a> )
<b>Total Marks</b>	10
<b>Due Dates</b>	Demo and Submission Due by End of Lesson 2 Class: <ul style="list-style-type: none"><li>• Jan. 15th for Set C (Monday)</li><li>• Jan. 16<sup>th</sup> for Set B (Tuesday)</li><li>• Jan. 18<sup>th</sup> for Set A (Thursday)</li></ul>

### Applicable Requirements

- **REQ1010** - The Enterprise Development Environment shall be prototyped on Microsoft Azure cloud infrastructure.
- **REQ1020** – The Enterprise Development Environment shall have a Source Code Management capability. The Source Code Management tool will be GitLab. Note: The GitLab Community Edition (free) will be sufficient for the prototype environment.
- **REQ1030** - The Source Code Management tool shall be integrated with the single sign-on services provided by the Microsoft Azure cloud infrastructure. Note: The oauth integration from Azure Active Directory (AD) will be used for the prototype.
- **REQ1040** – The Source Code Management tool shall allow software development teams to create source code repositories and logically group them. Note: Source code repositories are Projects in GitLab and groupings of repositories are Groups in GitLab. Logical groupings may be by software product and common infratructure.
- **SEC1010** – All password credentials will be stored in a password safe.
- **SEC1020** – All web applications and API endpoints shall be encrypted (i.e., https endpoints). Note: Self-signed certificates are sufficient for the prototype environment.

### Group Formation

For this lab and many subsequent labs you will work with a partner in order to:

- Share computing resources (and any associated costs).
- Work together for more complex tasks.
- Divide up the work for larger tasks.

Find a partner and signup with that partner in a Lab Group for your set on the Learning Hub.

For most labs, you will work with your partner and can choose to share Azure computing resources. So you can decide to only use the Azure account for one group member for each lab, or balance resources across both accounts.

It is **your responsibility** to manage the resources on Azure as you will be billed for your usage above the free tier and any credits you are provided by Microsoft. *Make sure you disable any resources, such as virtual machines when not being used.*

## Cloud Provider Setup

Signup for the Microsoft Azure for Student Free Account Credit. No credit card is required. This gives you the basic Azure free services plus a \$100 account credit (if you have not already signed up for this offer).

<https://azure.microsoft.com/en-us/free/students/>

You will then be using for Azure for Students subscription to cover most of your Azure costs for this course. Once you have exceeded the credits, you will have to provide a credit card and pay for any non-free services yourself.

**Note: Do not use your BCIT e-mail for your Azure user as this can cause problems since it is associated with BCIT's active directory. Create an Azure account using another e-mail. If you do use an Azure account associated with your BCIT e-mail, you will have to create a new Tenant in Azure Active Directory and apply your Student Subscription under that tenant.**

If you decide to work on the same Azure account with your partner, one person will need to sign-up for Azure and add the other as a co-administrator. To add a co-administrator, login to your Azure dashboard.

Add your partner to Azure Active Directory (AD):

- All services -> Azure Active Directory
- Users
- New Guest User
- Add your partner's information and invite them to your Azure AD
- Your partner should then get an e-mail inviting them to Azure.

Make your partner a Co-administrator:

- All services -> Subscriptions -> Azure for Students
- Access control (IAM)
- Add -> Add co-administrator
- Select the user you just added to Azure AD as a co-administrator.

The Azure for Student credit balance can be monitored here:

<https://www.microsoftazuresponsorships.com/>

## Familiarize Yourself with Azure

Microsoft Azure has most of the equivalent services to AWS that you may be familiar with. The following website provide the equivalent Azure service for most common AWS services:

<https://docs.microsoft.com/en-us/azure/architecture/aws-professional/services>

The two main Azure services you will use for this lab are:

- Azure Active Directory – for single sign-on
- Virtual Machines – to host a GitLab instance

## Source Code Management Setup

### Step 1 - VM Creation

*Note: GitLab will be quite slow to install and run on the free tier virtual machine as it likely does not have enough RAM. Recommend using a more capable VM (such as B2ms with 8GB of RAM) and stopping the instance when not actively being used.*

Select the Virtual Machines service in Azure and Add a new Virtual Machine.

Create a Linux Virtual Machine (using the Virtual Machines Service) with the following specifications under your Azure for Students subscription:

Basics:

- Subscription: Azure for Students
- Resource group (Create new): GitLabCE
- Virtual machine name: GitLabCE
- Region: (US) West US (or equivalent)
- Image: Ubuntu 20.04 LTS – x64 Gen2
- Size: Standard\_B2ms
- Authentication Type: SSH public key (create an SSH key if necessary and paste in the public key or have it create one for you)
- Inbound Ports: Allow HTTP (80), HTTPS (443), SSH (22)

Networking:

- Make sure Inbound Ports 80, 443 and 22 are open

Leave everything else at the defaults and Create the Virtual Machine. This may take a few minutes.

### Step 2 - Set the VM's DNS Name

When the VM is created, view the newly created VM's resources. It should already be assigned a Public IP address. However, this address may change when the VM is stopped and restarted.

Select the *Not Configured* link next to the DNS name label under the VM's details. Set the DNS name label to the following:

gitlab-acit4850-groupX (where X is your Group number)

Your full DNS name will be something like:

gitlab-acit4850-group1.westus2.cloudapp.azure.com

Make sure you record it as you will need it for the GitLab installation.

Select Save and then close the Configure view (i.e., select X in the top right corner)

### Step 3 - GitLab CE Installation

Using your ssh key, ssh into your VM.

Follow the instructions at the following URL to install GitLab CE your Ubuntu 18.04 LTS (or 20.04 LTS) VM. Read the notes below before running through the instructions.

<https://about.gitlab.com/install/#ubuntu>

Notes:

- **Make sure you replace “ee” with “ce” to install the free Community Edition.**
- You can skip the Postfix installation. We won’t be using the e-mail notifications.
- Use the DNS name you recorded above for your GitLab URL (i.e., <https://gitlab-acit4850-group1.westus2.cloudapp.azure.com>). Make sure to include the leading https://
- A LetsEncrypt SSL certificate should automatically be provisioned for your GitLab CE installation. If you get a validation error, verify the Networking configuration of your VM. Ports 80 and 443 should be open for inbound access.
- You can skip Steps 4 and 5 in the GitLab installation instructions.

Once the installation is done, go to the specified file on your VM to find the root password (/etc/gitlab/initial\_root\_password). Make sure you record the root password somewhere safe so you can find it again (i.e., a password safe).

You can now go to <https://<Your VM DNS>> and login to the GitLab web application with root as the username and your root password.

#### Step 4 – Single Sign-on

Because our Enterprise Development Environment will eventually be used by many product team members and will contain multiple applications, we do not want them to have to manage usernames and passwords for each application. Therefore, we want to integrate with a single sign-on capability.

For this prototype, we will use Azure Active Directory to manage users and to enable single sign-on through OAuth2.

In the Azure Portal:



- Go to Azure Active Directory (use the search bar to find it as necessary). You should be in your Default Directory (this won’t work in the BCIT Directory since it’s locked down).
- Select App registrations in the left side menu
- Select New Registration
  - Name: GitLabCE
  - Supported Account Types: Accounts in this organizational directory only (the default)
  - Redirect URI Web - [https://<Your VM DNS>/users/auth/azure\\_oauth2/callback](https://<Your VM DNS>/users/auth/azure_oauth2/callback)
- Click Register. The details of your new registration should show up.

Display name	: GitLabCE	Supported account types	: My organization only
Application (client) ID	: eee805ae-6ec3-462d-a88a-159a3594871f	Redirect URIs	: 1 web, 0 spa, 0 public client
Directory (tenant) ID	: 28f302e9-5ae6-4011-9484-cce45b537891	Application ID URI	: Add an Application ID URI
Object ID	: 06c712d2-4bc8-4f06-9d26-a2bce5c4b3ae	Managed application in I...	: GitLabCE

- Select Certificates & secrets on the left side menu

- Add a New client secret
  - Set the Description to client\_secret and set Expires to 6 months
  - Note the Value of the client\_secret can only be copied when first created. If you don't get the value at that time, you'll need to delete it and recreate it.

+ New client secret

Description	Expires	Value	ID
client_secret	12/31/2299	1~*****	373e6316-ff83-4c44-9de5-145286fa4ed0  

- You need the following from the details of your new registration:
  - Application (client) ID
  - Directory (tenant) ID
  - client\_secret

The above will not work if your Azure account is associated with your BCIT e-mail.

SSH into your GitLabCE VM:

- Open /etc/gitlab/gitlab.rb as root (i.e., use sudo). We're going to make the file look something like this:

```
# OAuth Config
gitlab_rails['omniauth_enabled'] = true
gitlab_rails['omniauth_allow_single_sign_on'] = ['azure_oauth2']
gitlab_rails['omniauth_sync_email_from_provider'] = 'azure_oauth2'
gitlab_rails['omniauth_sync_profile_from_provider'] = ['azure_oauth2']
gitlab_rails['omniauth_sync_profile_attributes'] = ['name', 'email']
gitlab_rails['omniauth_block_auto_created_users'] = false
gitlab_rails['omniauth_auto_link_ldap_user'] = true
gitlab_rails['omniauth_external_providers'] = ['azure_oauth2']
gitlab_rails['omniauth_providers'] = [
  {
    "name" => "azure_oauth2",
    "args" => {
      "client_id" => "$CLIENT_ID",
      "client_secret" => "$CLIENT_SECRET",
      "tenant_id" => "$TENANT_ID",
    }
  }
]
```

**Note: There are underscores above in azure\_oauth2.**

There is an existing section in the file with the above settings. You can uncomment it and update the values to match those above.

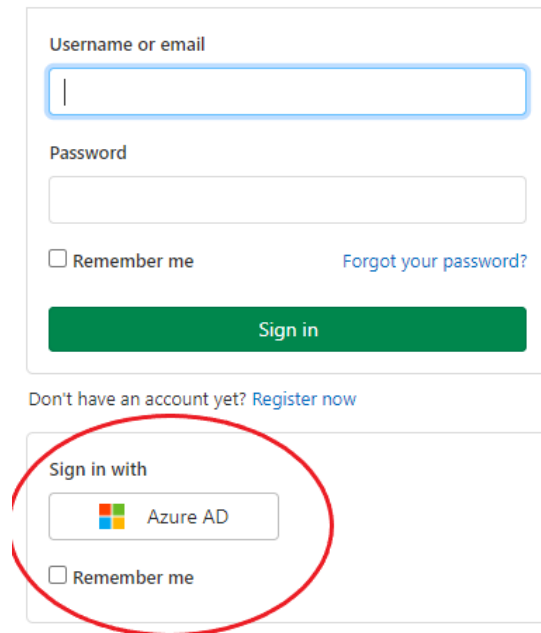
The \$CLIENT\_ID, \$CLIENT\_SECRET and \$TENANT\_ID should be replaced with your values from your App Registration.

- Reconfigure GitLab by running the following as root (i.e., use sudo):

```
gitlabctl reconfigure
```

This command will report any errors with your configuration that you need to fix.

When you go to your GitLab application, it should now allow you to login with Oauth.



The image shows the GitLab login interface. At the top, there is a form with two input fields: 'Username or email' and 'Password'. Below these fields are two checkboxes: 'Remember me' and 'Forgot your password?'. A green 'Sign in' button is positioned below the checkboxes. Below the main form, there is a link that says 'Don't have an account yet? Register now'. Below this link, there is a section titled 'Sign in with' which contains a button for 'Azure AD' (with the Microsoft logo) and another 'Remember me' checkbox. This entire 'Sign in with' section is circled in red.

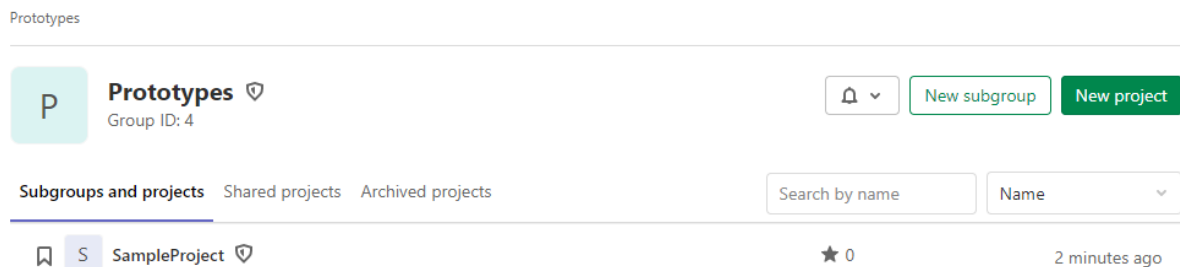
Make sure both you and your partner can login to GitLab using Oauth. If not, then you need to debug your configuration.


**Also make sure to disable self sign-up in GitLab as we only want access to GitLab from our Azure users.**

#### Step 5 – GitLab Groups and Projects

Under your user account (not root), login to GitLab and create the following:

- A Group called Prototypes (Visibility should be Internal)
  - Make sure both users, if applicable, are added to the group
- A Project called SampleProject (Visibility should be Internal) in the Prototypes group
- A Readme file in SampleProject





**SampleProject**
Project ID: 2


🔔

🌟 Star 0

🍴 Fork 0

1 Commit
1 Branch
0 Tags
143 KB Files
143 KB Storage

master
sampleproject / +
History
Find file
Web IDE
Clone


Update README.md  
Michael Mulder authored 1 minute ago

5dc93c7c

README
Auto DevOps enabled
Add LICENSE
Add CHANGELOG
Add CONTRIBUTING
Add Kubernetes cluster

Name	Last commit	Last update
📄 README.md	Update README.md	1 minute ago

📄 README.md

This is a test

**Note: When you login in for the first time with your user account using Oauth, an account is created for your user in GitLab. You must login as root after this account is created and make your user an admin in order to be able to create Groups and Projects.**

Verify that you can clone the SampleProject repository locally to both yours and your partner's laptops.

Note: To be able to clone the repo locally you need to setup credentials in GitLab for your user. You can either upload your ssh public key, create an access token or (I believe) there is a way to use your Oauth credentials directly.

You are now done and ready to demo next class. **Make sure you shutdown any Azure resources (i.e., the VM) to conserve your credits and free tier usage.**

### Demo, Grading and Submission

A demo of your lab against the applicable requirements which will determine your grade on the lab. All mandatory requirements must be met otherwise you will receive zero on the lab. You can re-demo the lab if you haven't met the mandatory requirements, up to the last class before the midterm week, but you will lose 20% every week late.

Req.	Mandatory	Demo	Marks
REQ1010	Yes	<ul style="list-style-type: none"> <li>Show your Azure dashboard (both students).</li> <li>Show your running VM.</li> </ul>	2
REQ1020	Yes	<ul style="list-style-type: none"> <li>Show your running GitLab instance at the DNS name of your server from Azure</li> </ul>	2
REQ1030	Yes	<ul style="list-style-type: none"> <li>Show that you can login to GitLab with your Azure user using Oauth (both students).</li> </ul>	2
REQ1040	Yes	<ul style="list-style-type: none"> <li>Show your Group and Project in GitLab and your local clone of the Project (both students).</li> </ul>	2
SEC1010	Yes	<ul style="list-style-type: none"> <li>Show that your GitLab endpoint is encrypted (https). Note: It can be self-signed for this lab.</li> </ul>	1

SEC1020	No	<ul style="list-style-type: none"> <li>• Show that you have used ssh keys for ssh access to your VM.</li> <li>• Describe how you are securely storing your GitLab root password</li> </ul>	1
<b>Total</b>			<b>10</b>

**Submit a screenshot of your GitLab Group/Project to the Lab1 dropbox under Activities -> Assignments on D2L.**