teo-ma / MCW-Cloud-native-applications Public

forked from microsoft/MCW-Cloud-native-applications

Code Pull requests Actions Projects Wiki Security Insights Settings

main

MCW-Cloud-native-applications / Hands-on lab / **Before the HOL - Cloud-native applications.md**

teo-ma update script

14 contributors +2

```
305 lines (195 sloc) | 18.7 KB
```

Microsoft Cloud Workshop

Cloud-native applications
Before the hands-on lab setup guide
November 2021

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

Contents

Cloud-native applications before the hands-on lab setup guide

Overview

Before the hands-on lab, you will need to prepare the environment by deploying the database and the application locally on a virtual machine using Docker and MongoDB. You will also need to fork the GitHub repository containing the lab to your own GitHub account to be able to set up the CI/CD pipeline.

Requirements

1. Microsoft Azure subscription must be pay-as-you-go or MSDN.
   - Trial subscriptions will *not* work.
   - To complete this lab setup, ensure your account includes the following:
     - Has the Owner built-in role for the subscription you use.
     - Is a Member user in the Azure AD tenant you use. (Guest users will not have the necessary permissions.)
   - You must have enough cores available in your subscription to create the build agent and Azure Kubernetes Service cluster in Task 5: Deploy ARM Template. You'll need eight cores if following the exact instructions in the lab, more if you choose additional agents or larger VM sizes. Execute the steps required before the lab to see if you need to request more cores in your sub.
2. An account in Microsoft GitHub.
3. Local machine or a virtual machine configured with:
   - A browser, preferably Chrome for consistency with the lab implementation tests.
4. You will be asked to install other tools throughout the exercises.

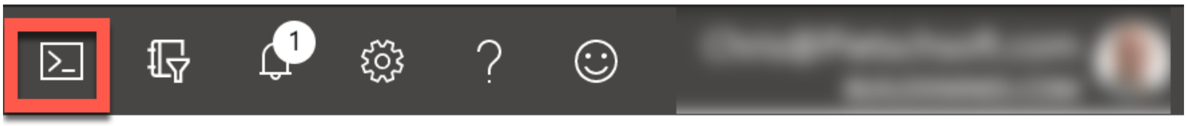
Before the hands-on lab

Duration: 60 minutes

You should follow all the steps provided in this section *before* taking part in the hands-on lab ahead of time as some of these steps take time.

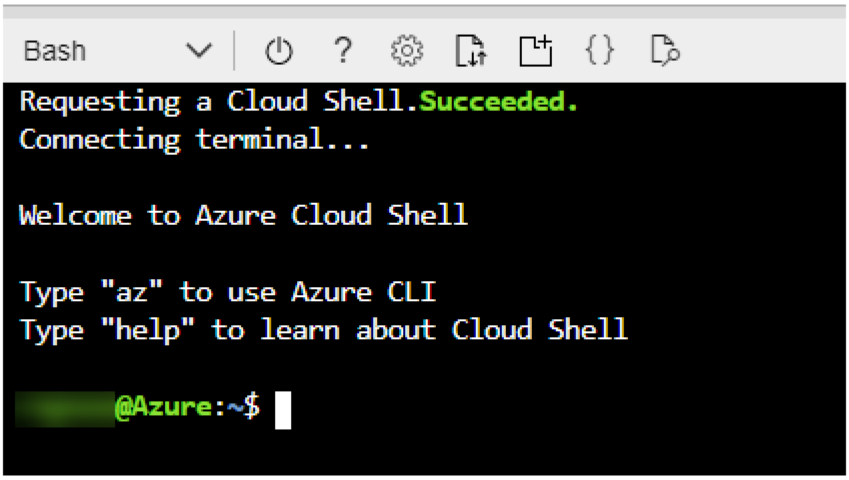
Task 1: Set up Azure Cloud Shell

1. Open a cloud shell by selecting the cloud shell icon in the menu bar.



[redacted]

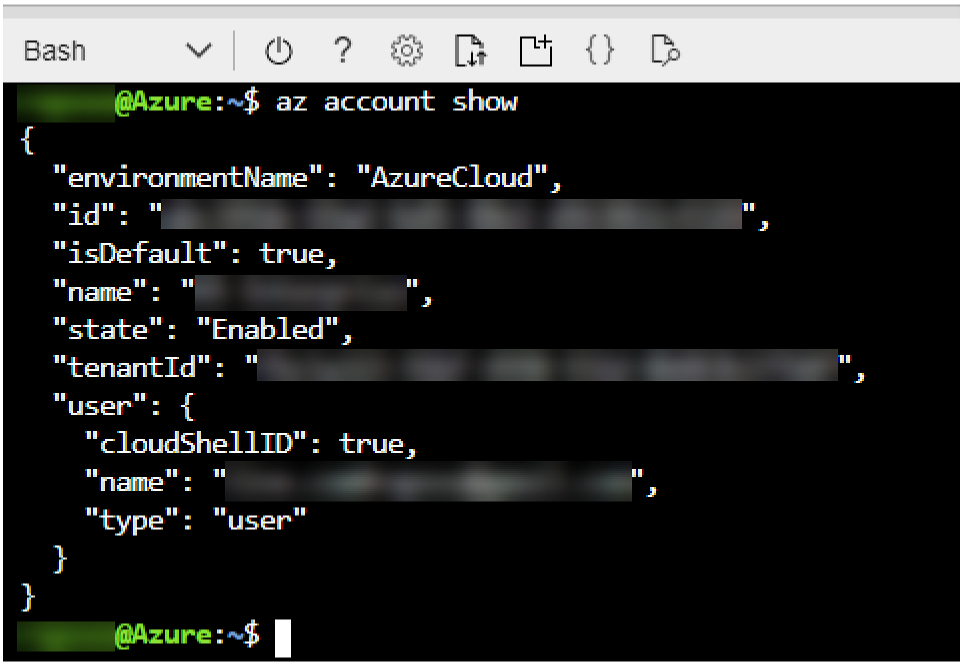
2. The cloud shell opens in the browser window. Choose **Bash** if prompted or use the left-hand dropdown on the shell menu bar to choose **Bash** from the dropdown (as shown). If prompted, select **Confirm**.



```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

[redacted]@Azure:~$
```

3. Make sure to set your default subscription correctly. To view your current subscription type:

```
az account show
```



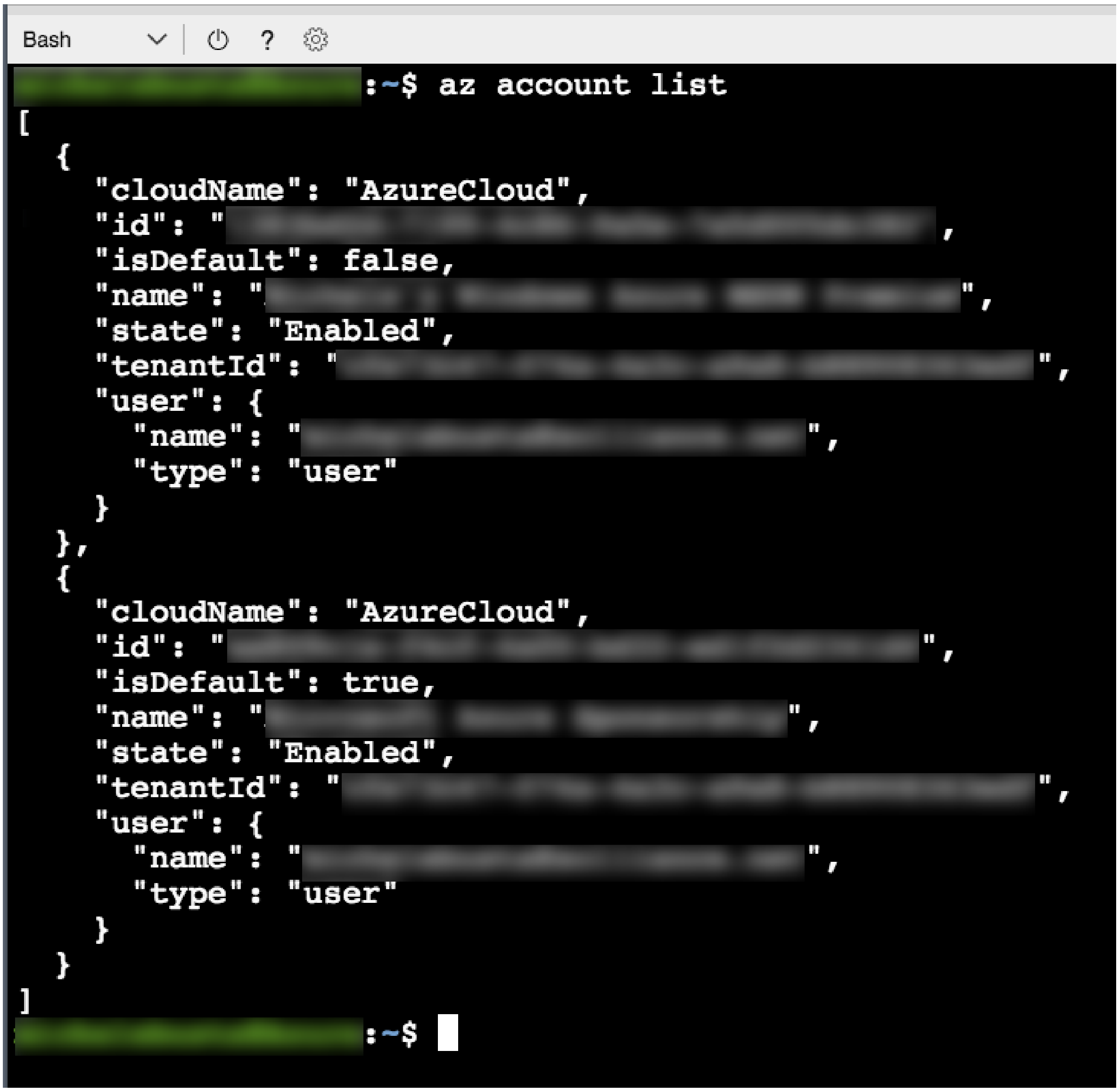
```
[redacted]@Azure:~$ az account show
{
  "environmentName": "AzureCloud",
  "id": "[redacted]",
  "isDefault": true,
  "name": "[redacted]",
  "state": "Enabled",
  "tenantId": "[redacted]",
  "user": {
    "cloudShellID": true,
    "name": "[redacted]",
    "type": "user"
  }
}
[redacted]@Azure:~$
```

4. If the subscription displayed in the previous step is not the subscription you plan on using for this workshop, you will need to set the current subscription to your desired subscription. To set your default subscription to something other than the current selection, type the following, replacing {id} with the desired subscription id value:

```
az account set --subscription {id}
```

> **Note**: If you do not know the id of your desired subscription, you can list the subscriptions available to you along with their ids via the following command:

```
az account list
```



```
[redacted]:~$ az account list
[
  {
    "cloudName": "AzureCloud",
    "id": "[redacted]",
    "isDefault": false,
    "name": "[redacted]",
    "state": "Enabled",
    "tenantId": "[redacted]",
    "user": {
      "name": "[redacted]",
      "type": "user"
    }
  },
  {
    "cloudName": "AzureCloud",
    "id": "[redacted]",
    "isDefault": true,
    "name": "[redacted]",
    "state": "Enabled",
    "tenantId": "[redacted]",
    "user": {
      "name": "[redacted]",
      "type": "user"
    }
  }
]
[redacted]:~$
```

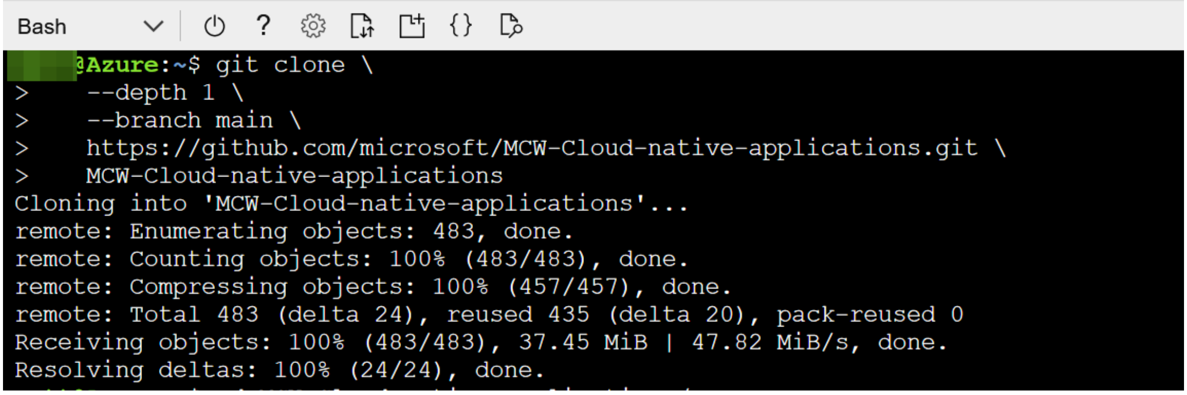
Task 2: Download Starter Files

> **Note**: You will need access to your Azure subscription via Azure Cloud Shell to proceed further in the workshop. If you don't have a cloud shell available, refer back to Task 1: Set up Azure Cloud Shell for set up instructions.

1. Check out the starter files from the MCW Cloud-native applications GitHub repository and detach them from the existing remote repository via the following commands:

```
cd ~
git clone \
   --depth 1 \
   --branch main \
   https://github.com/teo-ma/MCW-Cloud-native-applications.git \
   MCW-Cloud-native-applications
cd MCW-Cloud-native-applications
```

> **Note**: If you do not have enough free space, you may need to remove extra files from your Cloud Shell environment. Try running `azcopy jobs clean` to remove any `azcopy` jobs and data you do not need.



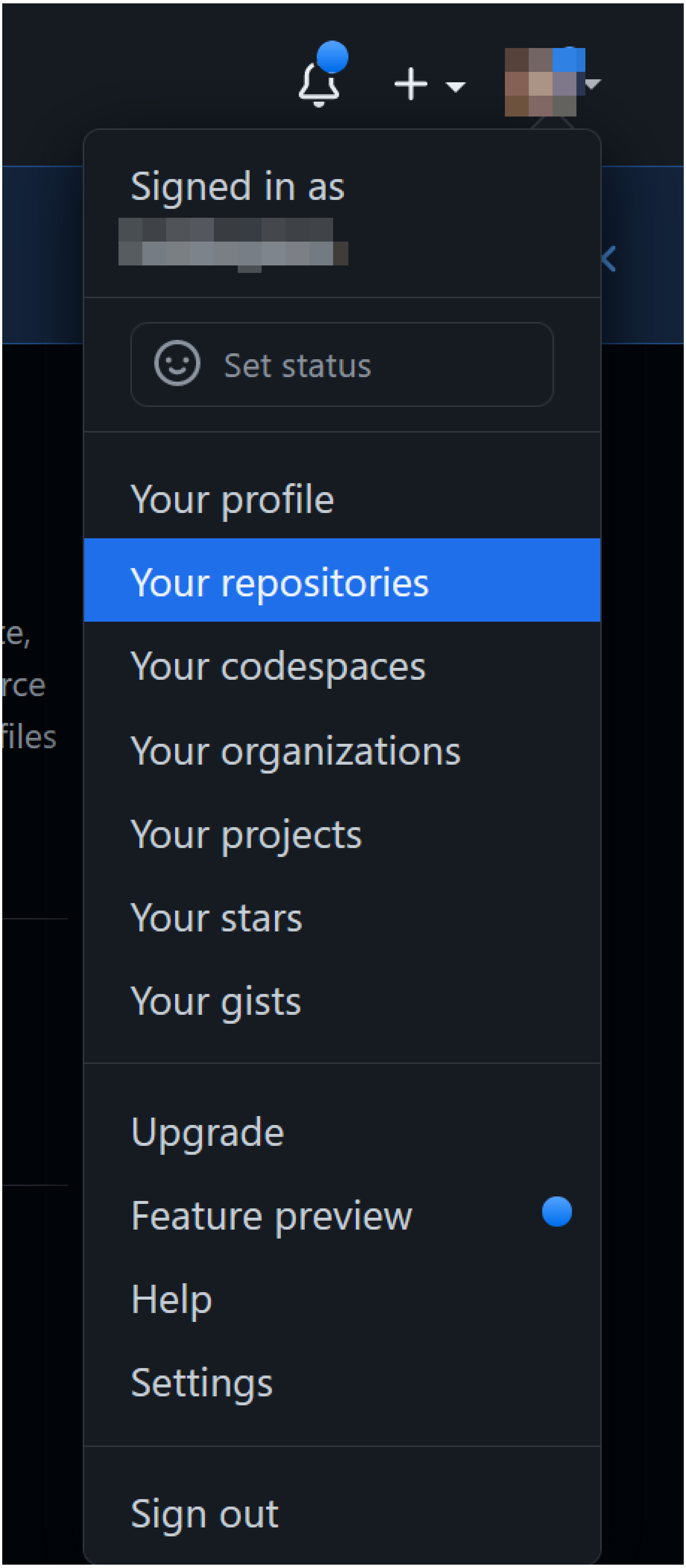
```
[redacted]@Azure:~$ git clone \
>    --depth 1 \
>    --branch main \
>    https://github.com/microsoft/MCW-Cloud-native-applications.git \
>    MCW-Cloud-native-applications
Cloning into 'MCW-Cloud-native-applications'...
remote: Enumerating objects: 483, done.
remote: Counting objects: 100% (483/483), done.
remote: Compressing objects: 100% (457/457), done.
remote: Total 483 (delta 24), reused 435 (delta 20), pack-reused 0
Receiving objects: 100% (483/483), 37.45 MiB | 47.82 MiB/s, done.
Resolving deltas: 100% (24/24), done.
```

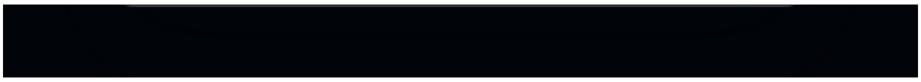
Task 3: Create a GitHub repository

FabMedical has provided starter files for you. They have taken a copy of the websites for their customer Contoso Neuro and refactored it from a single node.js site into a website with a content API that serves up the speakers and sessions. This refactored code is a starting point to validate the containerization of their websites. Use this to help them complete a POC that validates the development workflow for running the website and API as Docker containers and managing them within the Azure Kubernetes Service environment.

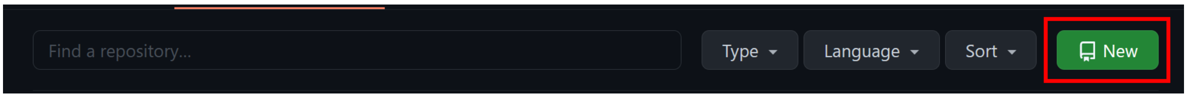
1. Open a web browser and navigate to https://www.github.com. Log in using your GitHub account credentials.

2. In the upper-right corner, expand the user drop-down menu and select **Your repositories**.

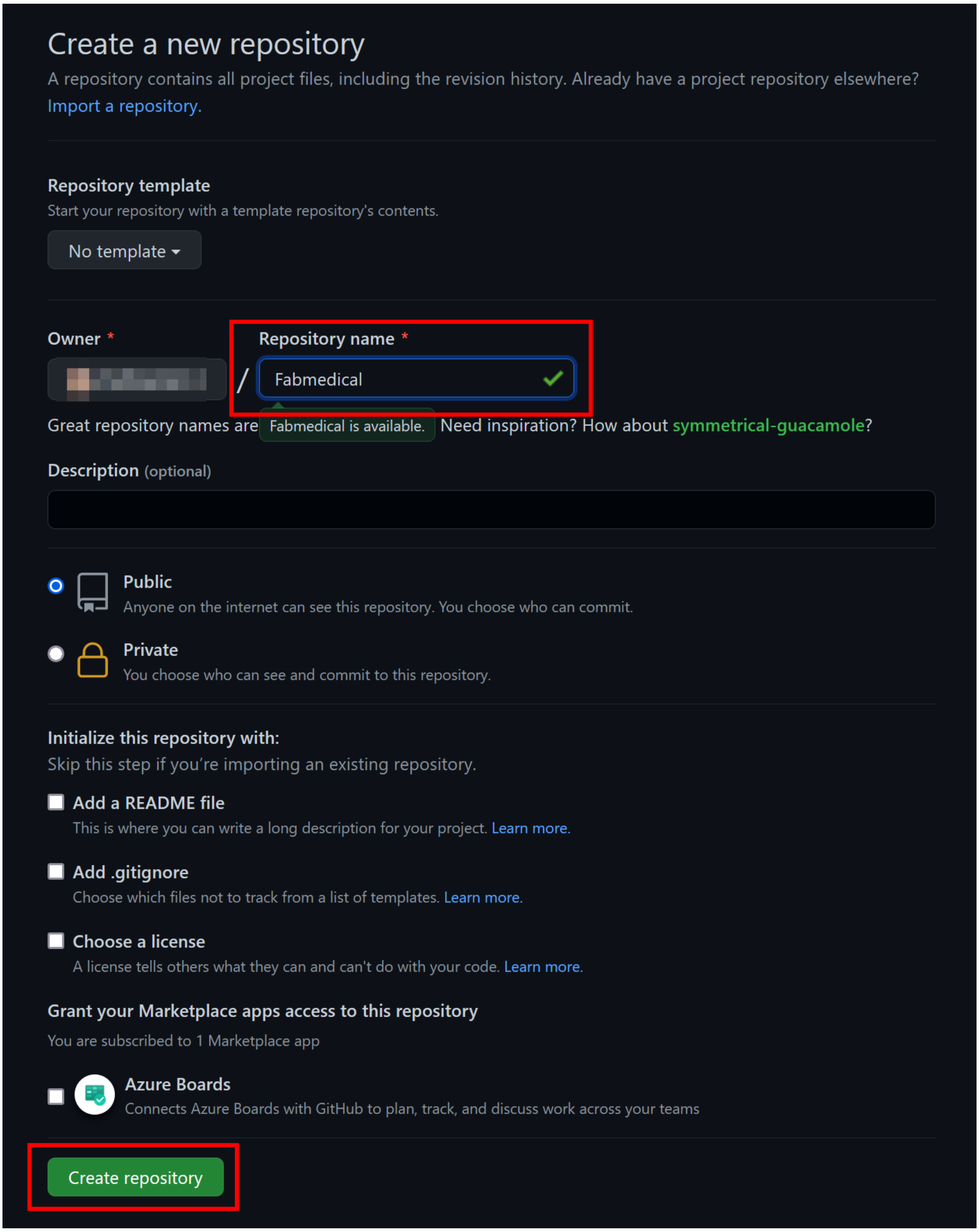

Signed in as
[redacted]
Set status
Your profile
Your repositories
Your codespaces
Your organizations
Your projects
Your stars
Your gists
Upgrade
Feature preview
Help
Settings
Sign out



3. Next to the search criteria, locate and select the **New** button.



4. On the **Create a new repository** screen, name the repository **Fabmedical** and select the **Create repository** button.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template

Start your repository with a template repository's contents.

No template

Owner * [redacted] / Repository name * Fabmedical ✓

Great repository names are Fabmedical is available. Need inspiration? How about symmetrical-guacamole?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

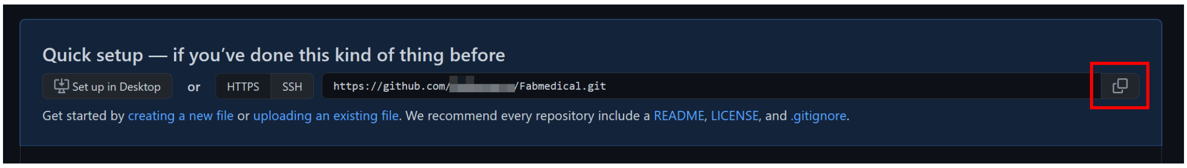
□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

Grant your Marketplace apps access to this repository
You are subscribed to 1 Marketplace app

□ Azure Boards
Connects Azure Boards with GitHub to plan, track, and discuss work across your teams

Create repository

5. On the **Quick setup** screen, copy the **HTTPS** GitHub URL for your new repository, paste this into notepad for future use.



Quick setup — if you've done this kind of thing before

https://github.com/[redacted]/Fabmedical.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

Task 4: Set up Azure Cloud Shell environment

1. A GitHub personal access token (PAT) with appropriate permissions is required to set up and complete this lab - Follow this link to quickly set up a GitHub personal access token with the required permissions. Save the obtained PAT as it will be needed by future steps.

> **Note**: Make sure to select the `workflow` and `admin:org read:org` permission scopes in addition to the `repo` scopes already selected when visiting the aforementioned link.

2. Set the following environment variables in an Azure Cloud Shell terminal.

```
export MCW_SUFFIX=<SUFFIX>                         # Needs to be a
export MCW_GITHUB_USERNAME=<GITHUB USERNAME>       # Your Github ac
export MCW_GITHUB_TOKEN=<GITHUB PAT>               # A personal acc
export MCW_AZURE_SUBSCRIPTION=<AZURE SUBSCRIPTION ID> # The target Azu
```

> **Note**: The following environment variables can also be set if their defaults are not appropriate for the lab setting or environment.

- MCW_GITHUB_URL - Defaults to `https://github.com/$MCW_GITHUB_USERNAME/Fabmedical`
- MCW_PRIMARY_LOCATION - Defaults to `eastus`
- MCW_PRIMARY_LOCATION_NAME - Defaults to `East US`
  - Note that the value needs to be placed in quotes (e.g. `export MCW_PRIMARY_LOCATION_NAME="West US"`)
- MCW_SECONDARY_LOCATION - Defaults to `westus`
- MCW_SECONDARY_LOCATION_NAME - Defaults to `West US`

> **Note**: If you run into the error below, you may have to either use a different regional pair or increase your regional core quotas in your current regions. This lab's cloud resources require at least eight available cores in your regional core quota. An up to date list of Azure Region Pairs can be found at this link.

```
{"error":{"code":"InvalidTemplateDeployment","message":"The template d
```

3. Configure your git email and name.

```
git config --global user.email "your@email.com"
git config --global user.name "Your Name"
```

4. Run the `create_azure_resources.sh` script in the `MCW-Cloud-native-applications` repository that was cloned in a previous step. This will provision all of the Azure cloud resources necessary to execute the workshop.

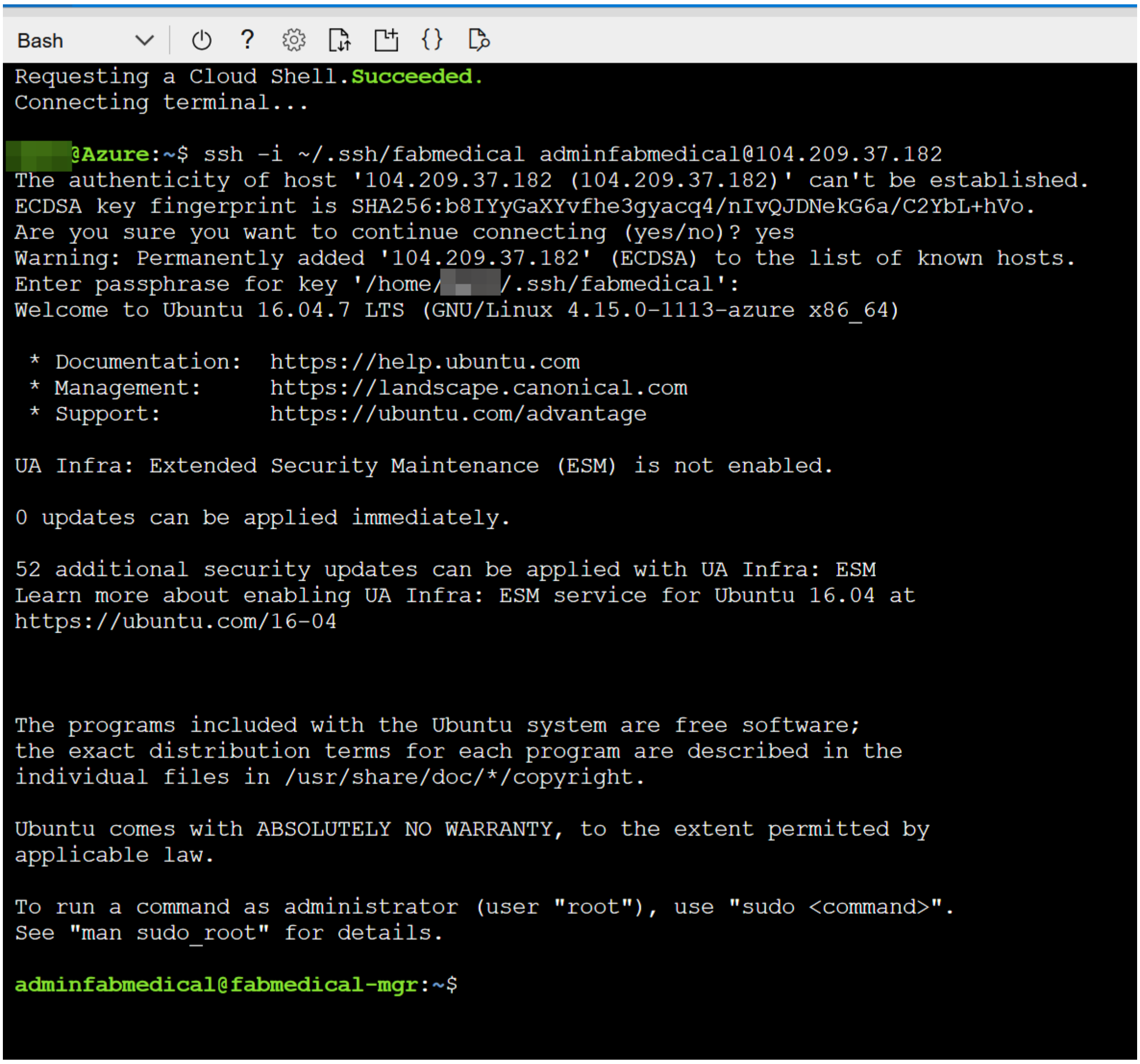
```
cd ~/MCW-Cloud-native-applications/Hands-on\ lab/lab-files/developer/s
bash create_azure_resources.sh
```

5. Upon successful execution of the `create_azure_resources.sh` script, a command for establishing an SSH session to the build agent VM should be present in the output.

```
Command to create an active session to the build agent VM:

    ssh -i ~/.ssh/fabmedical adminfabmedical@<PUBLIC IP OF VM>
```

6. Use the SSH command output in the previous step to establish an SSH session to the build agent VM. You should be presented with a prompt similar to the following:

```
adminfabmedical@fabmedical-SUFFIX:~$
```



```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

[redacted]@Azure:~$ ssh -i ~/.ssh/fabmedical adminfabmedical@104.209.37.182
The authenticity of host '104.209.37.182 (104.209.37.182)' can't be established.
ECDSA key fingerprint is SHA256:b8IYyGaXYvfhe3gyacq4/nIvQJDNekG6a/C2YbL+hVo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '104.209.37.182' (ECDSA) to the list of known hosts.
Enter passphrase for key '/home/[redacted]/.ssh/fabmedical':
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.15.0-1113-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

UA Infra: Extended Security Maintenance (ESM) is not enabled.

0 updates can be applied immediately.

52 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

adminfabmedical@fabmedical-mgr:~$
```

Task 5: Complete the build agent setup

1. From an Azure Cloud Shell terminal, use the SSH command output from the previous task and start an active SSH session to the build agent VM.

2. Clone the FabMedical GitHub repository created in the previous task.

```
git clone https://github.com/<GITHUB_USERNAME>/Fabmedical
```

3. Set the following environment variables in the active SSH session to the build agent VM. Use the same GitHub access token and Azure subscription ID used in a previous task.

```
export MCW_SUFFIX=<SUFFIX>                              # Needs to be a
export MCW_GITHUB_USERNAME=<GITHUB USERNAME>            # Your Github ac
export MCW_GITHUB_TOKEN=<GITHUB PAT>                    # A personal acc
export MCW_AZURE_SUBSCRIPTION=<AZURE SUBSCRIPTION ID> # The target Azu
```

4. Run the `create_build_environment.sh` script to set up the build agent VM environment. This script installs necessary dependencies on the build agent VM and applies the configuration settings to the VM's environment necessary for proper execution of the workshop.

```
cd ~/Fabmedical/scripts
bash create_build_environment.sh
```

> **Note**: Ignore any errors you encounter regarding the Docker client. That will be resolved after joining a new SSH session in the following steps.

5. After the script completes execution, type `exit` to exit the SSH session. We will need to join a new SSH session to ensure the docker environment on the build agent VM has completed set up.

```
adminfabmedical@fabmedical-???:~/Fabmedical/scripts$ exit
logout
Connection to 20.123.9.212 closed.
matt@Azure:~$ ssh -i ~/.ssh/fabmedical adminfabmedical@20.123.9.212
matt@Azure:~$ ssh -i .ssh/fabmedical adminfabmedical@20.123.9.212
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.15.0-1113-azure x86_64)

* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

UA Infra: Extended Security Maintenance (ESM) is not enabled.

0 updates can be applied immediately.

65 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Sat Dec  4 11:47:31 2021 from 23.100.27.176
adminfabmedical@fabmedical-mgy:~$
```

6. After reestablishing an SSH session to the build agent VM, run the `create_and_seed_database.sh` script to create and seed the MongoDB database for use in the workshop.

```
cd ~/Fabmedical/scripts
bash create_and_seed_database.sh
```

Task 6: Build Docker Images

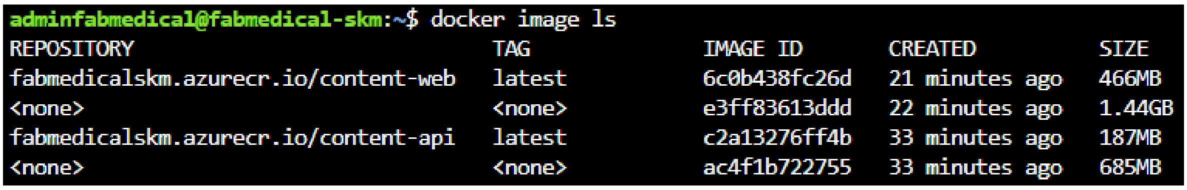
1. Navigate to the `content-api` directory and build the `content-api` container image using the Dockerfile in the directory. Note how the deployed Azure Container Registry is referenced. Replace the `SUFFIX` placeholder in the command.

   ```
   cd ~/Fabmedical/content-api
   docker image build -t fabmedical[SUFFIX].azurecr.io/content-api:latest
   ```

2. Repeat this step for the `content-web` image, which serves as the application front-end.

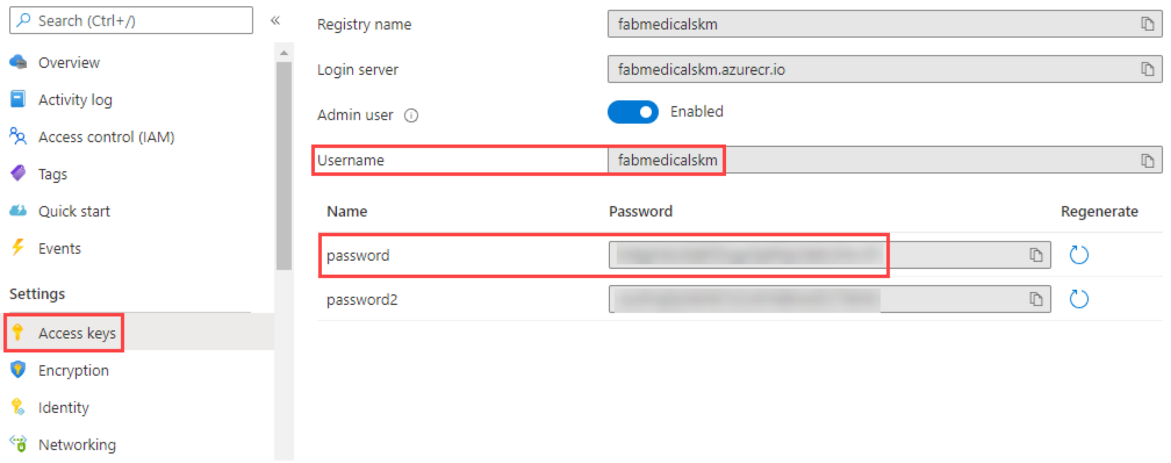
   ```
   cd ~/Fabmedical/content-web
   docker image build -t fabmedical[SUFFIX].azurecr.io/content-web:latest
   ```

3. Observe the built Docker images by running `docker image ls`. The images were tagged with `latest`, but it is possible to use other tag values for versioning.



```
adminfabmedical@fabmedical-skm:~$ docker image ls
REPOSITORY                            TAG      IMAGE ID      CREATED          SIZE
fabmedicalskm.azurecr.io/content-web  latest   6c0b438fc26d  21 minutes ago   466MB
<none>                                <none>   e3ff83613ddd  22 minutes ago   1.44GB
fabmedicalskm.azurecr.io/content-api  latest   c2a13276ff4b  33 minutes ago   187MB
<none>                                <none>   ac4f1b722755  33 minutes ago   685MB
```

4. Log in to Azure Container Registry using `docker login fabmedical[SUFFIX].azurecr.io`. Fetch the credentials from the **Access keys** tab of the ACR instance in the Azure portal.



Registry name: fabmedicalskm

Login server: fabmedicalskm.azurecr.io

Admin user: Enabled

Username: fabmedicalskm

Name	Password	Regenerate
password	[redacted]	
password2	[redacted]	

5. Push the two images you built.

    ```
    docker image push fabmedical[SUFFIX].azurecr.io/content-api:latest
    docker image push fabmedical[SUFFIX].azurecr.io/content-web:latest
    ```

You should follow all steps provided *before* performing the Hands-on lab.