# Installing MedICI and CodaLab



This document explains how to install CodaLab and is an alternative to the more detailed user manual on the CodaLab Wiki. CodaLab is under active development, so the procedures here are subject to change.

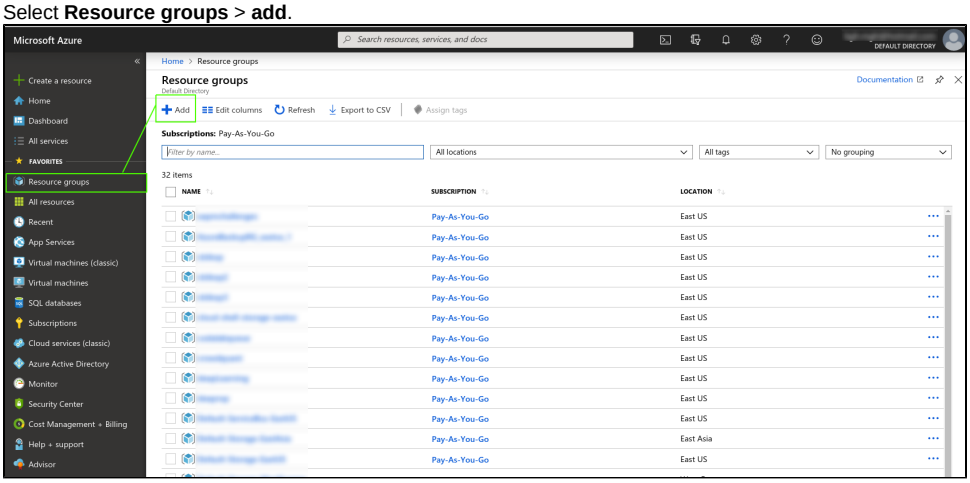
Your first step is to decide which virtual machine or computer you want to install MedICI.

## Creating a Virtual Machine on Microsoft Azure\Ubuntu Server 18.04 LTS

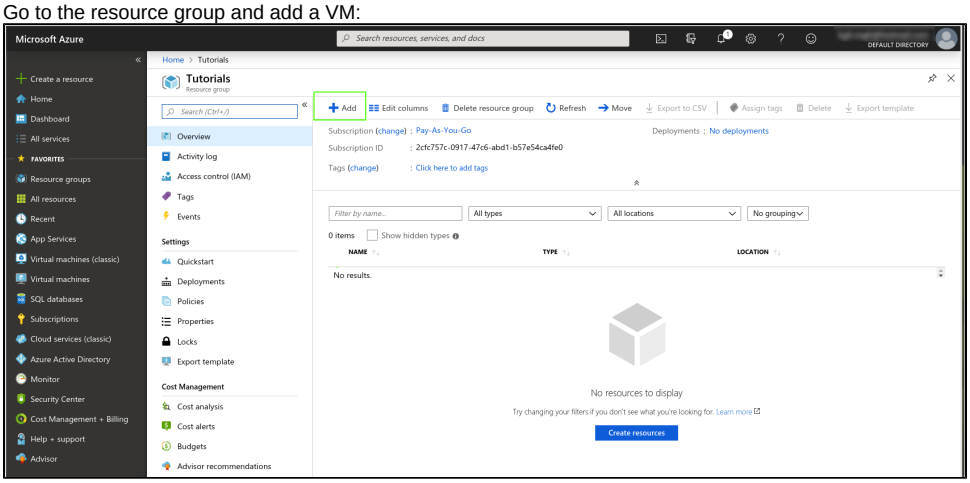
This tutorial focuses on Ubuntu-based installations on Azure, although instructions on how to do this on Microsoft and Apple operating systems and in AWS are also included.

Create a VM in Azure or use a computer with Ubuntu Server 18.04 installed. Once you have an Azure account, go to the Azure home page. A sidebar with many options or categories appears. You can create either a resource or a resource group and then add resources into it. This tutorial uses the resource group method to demonstrate good organizational practices.

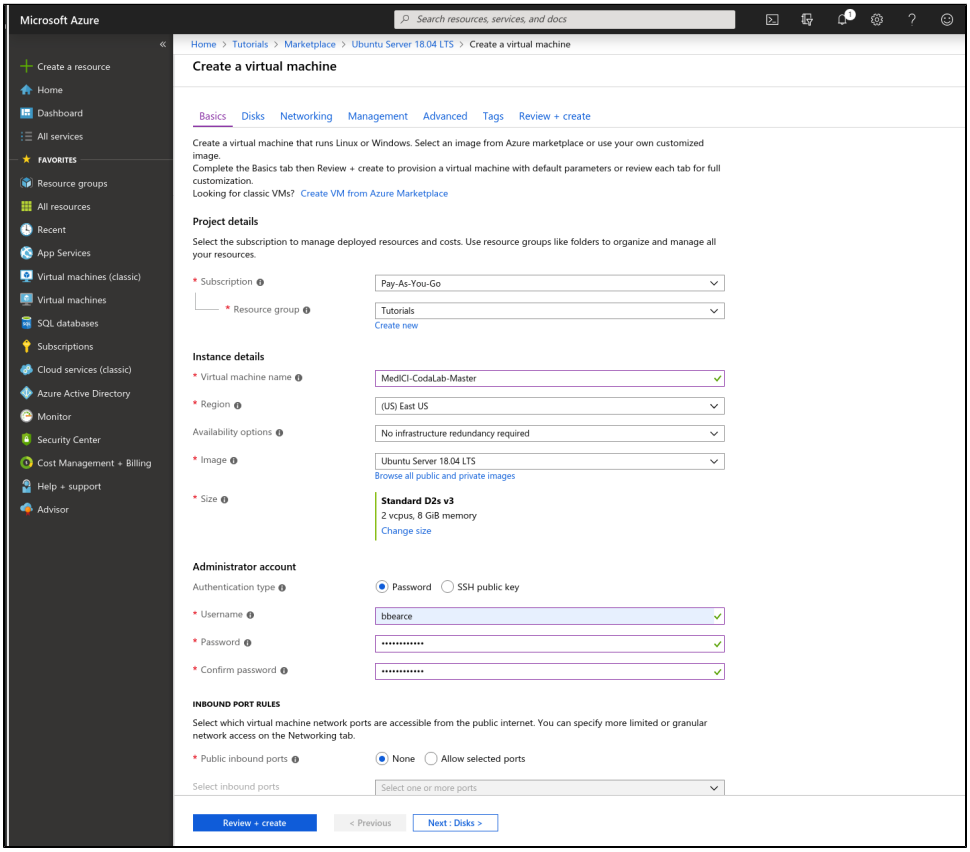
1. Select **Resource groups** > **add**.



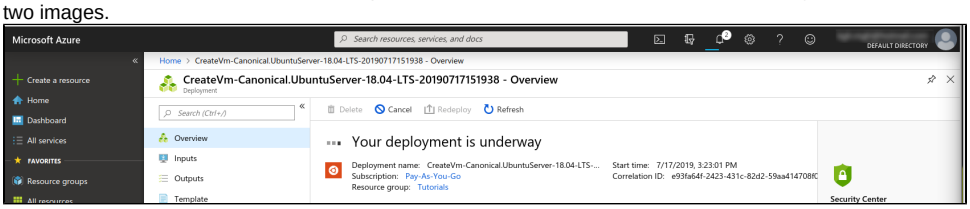

2. Name the resource group and then decide how you will pay for this service.
3. Click **Review + create**.
4. Go to the resource group and add a VM:

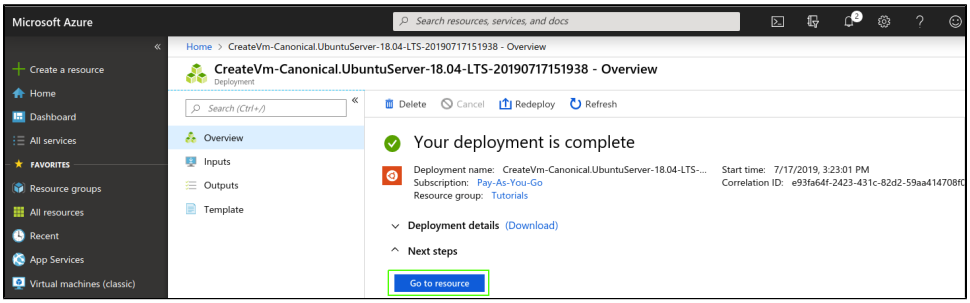


5. Choose Ubuntu Server 18.04 LTS.
   The VM creation page appears.
6. Choose the following settings.



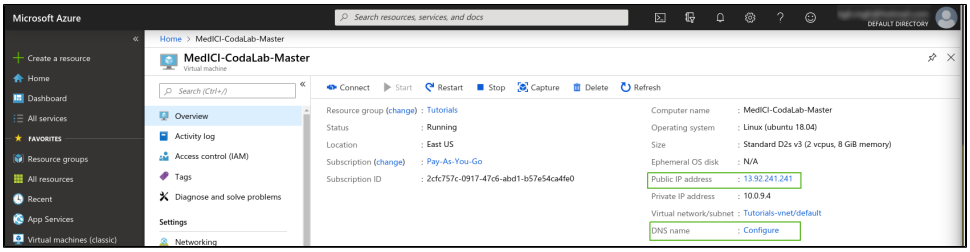

7. Now click **Review + create**. Review the options and then click **create**. Azure starts creating the VM and then finishes, as shown in the following two images.



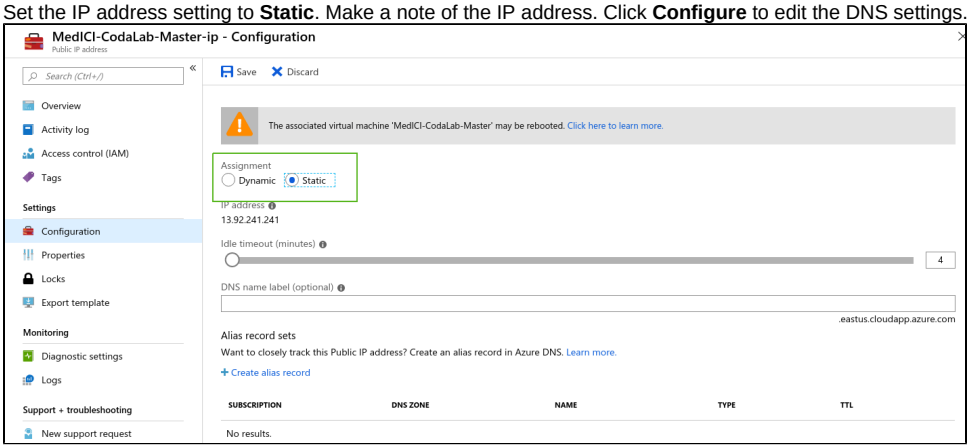




# IP Address Config

1. Navigate to the VM by clicking **Go to resource**. Go to the resources home page to get the public IP address so that you can access the machine.

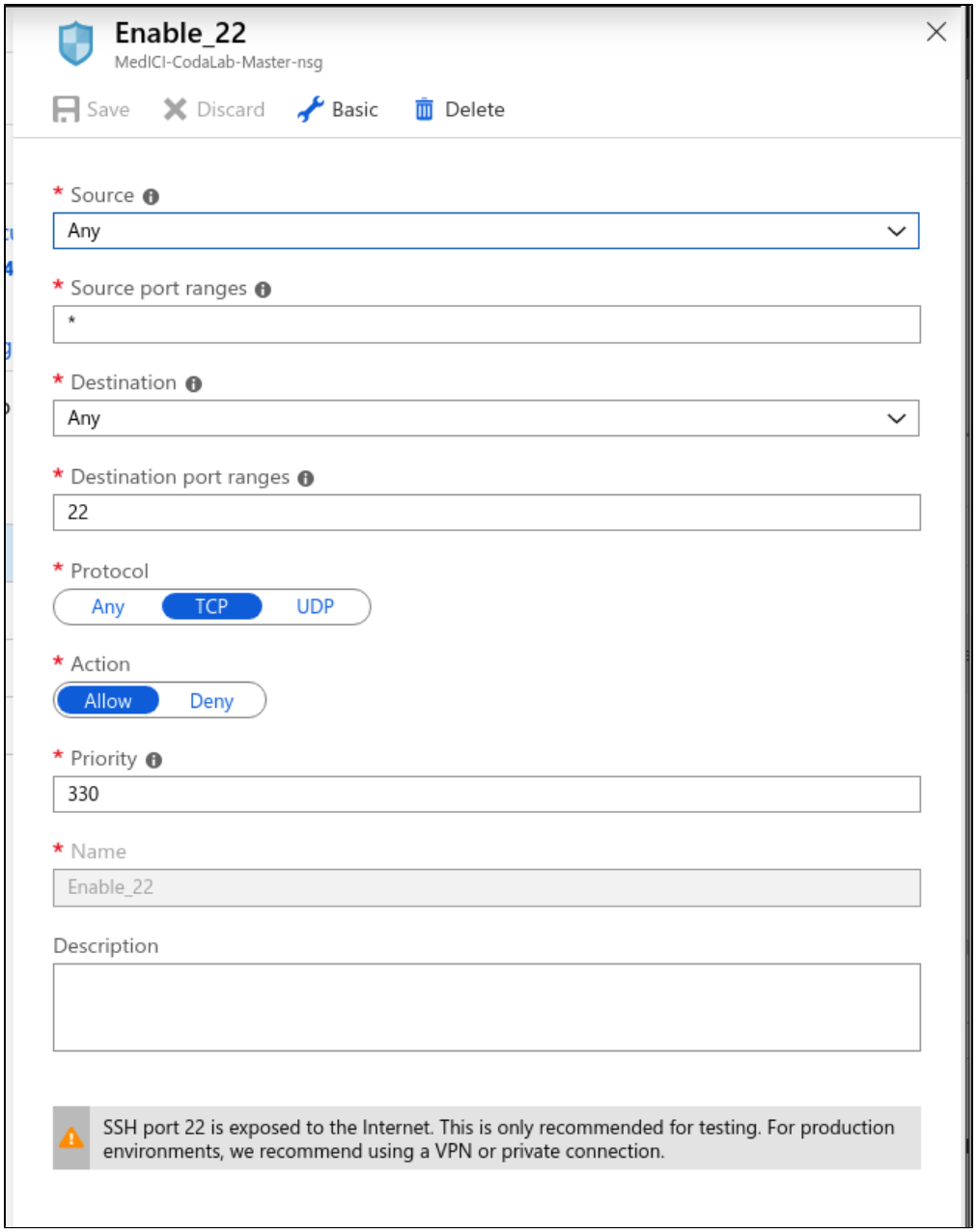



2. Set the IP address setting to **Static**. Make a note of the IP address. Click **Configure** to edit the DNS settings. Click **Save**.

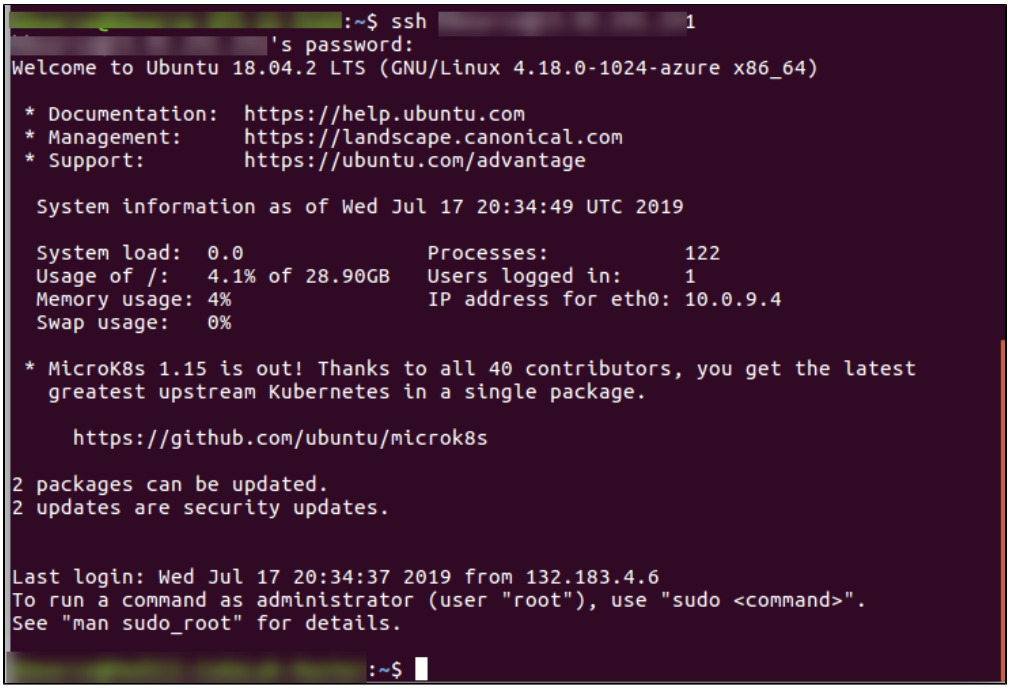



## SSH Port

We need to configure the machine to be ssh accessible (port 22) and open http (port 80) (https://medium.com/techinpieces/practical-azure-how-to-enable-ssh-on-azure-vm-84d8fba8103e). Go to the VM and click *Networking.* On the right find the button *Add inbound port rule.* Change *Destination port ranges* to 22, *Protocol* to TCP, *Priority* to 330, and *Name* to anything you'd like:



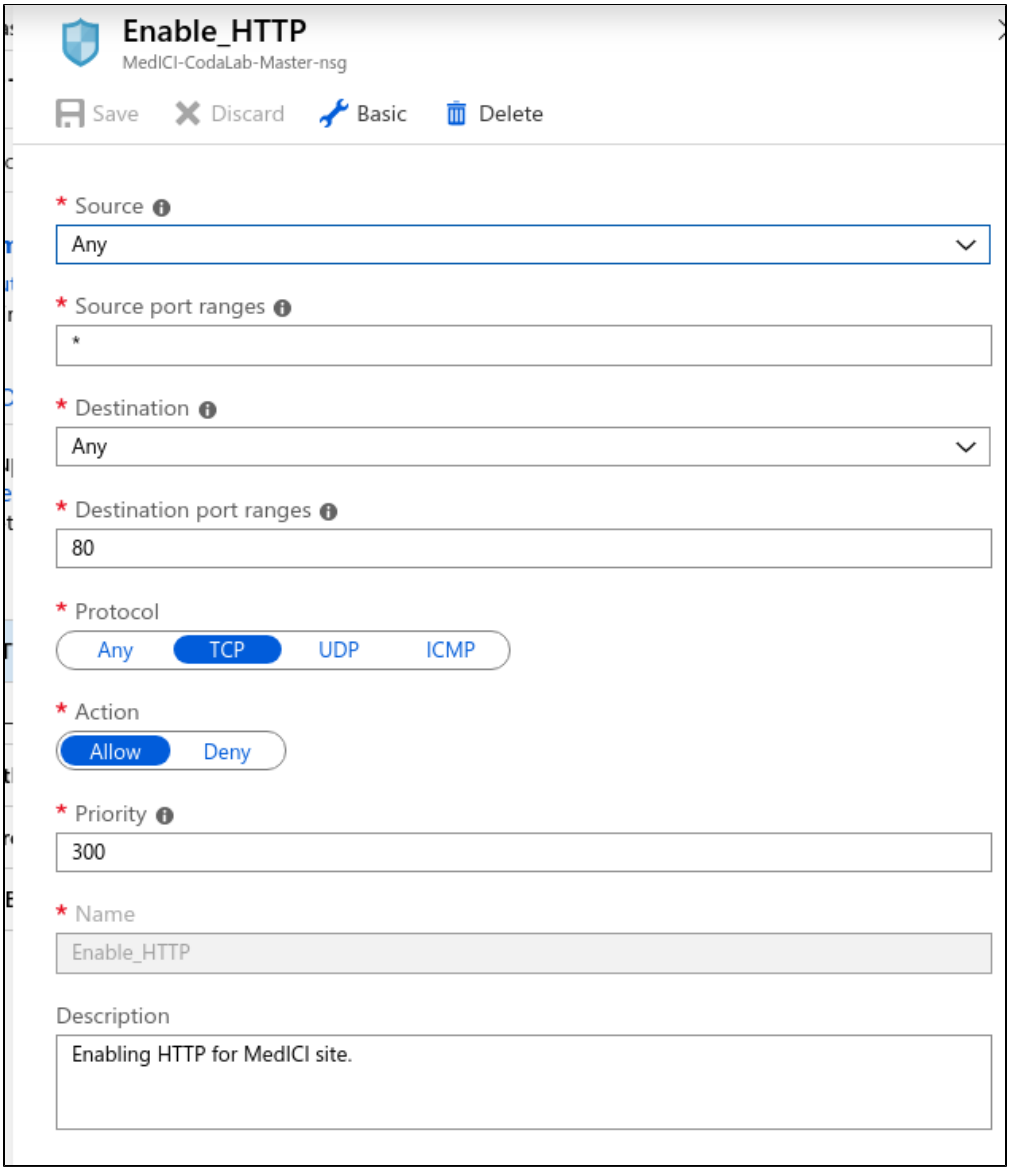

Once you have that let's open a terminal and get inside the machine by typing *$ ssh <user>@<public IP address>.*




If you get a message regarding encryption keys, enter y\yes as this is the first time signing into this machine.

## HTTP Port

On the right find the button **Add inbound port rule**. Change the **Destination port ranges** to 80, **Protocol** to TCP, **Priority** to 300, and **Name** to anything you like.




This step is necessary to access the site once you deploy it.

# Installing CodaLab

CodaLab has documentation regarding their preferred installation configuration. The two best sources are the following:

- https://github.com/codalab/codalab-competitions/wiki
- https://codalab-competitions.readthedocs.io/en/latest/

The base code for CodaLab can be set up on different cloud providers (Google Cloud, AWS, Azure). The CodaLab team has made Docker images for AWS and Google cloud that in theory should just run once some settings are established, but require a good working knowledge of Docker. For the purposes of MedICI, it is recommended to follow the "Configure CodaLab from scratch (harder documentation)" procedure found in the first link above.

## Step 1 - Install Docker and Docker-Compose

Before we clone GitHub, we must install docker and docker-compose. Follow the Docker installation instructions to install on Ubuntu.

### Docker

The following are the commands for a basic installation of Docker on Ubuntu.

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
$ sudo apt-get update
```

```
$ sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
$ sudo apt-key fingerprint 0EBFCD88

$ sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
   $(lsb_release -cs) \
   stable"
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
$ sudo docker run hello-world
```

This creates the following result.

```
bbearce@MedICI-CodaLab-Master:~/src/MedICI$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:6540fc08ee6e6b7b63468dc3317e3303aae178cb8a45ed3123180328bcc1d20f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Now if we run the `docker` command, we see the options list that tells us how to use the command, which verifies that Docker is installed and we can use it.

## Docker-Compose

Check to see if docker-compose is installed from the terminal on your new virtual machine. If not, rather than use sudo apt install docker-compose, the CodaLab documentation from Configure CodaLab from scratch (harder documentation) says to install it from Docker's website. Use the following command.

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.25.5/docker-compose-$(uname -s)-$(uname -
m)" -o /usr/local/bin/docker-compose
```

```
bbearce@MedICI-CodaLab-Master:~$ sudo curl -L "https://github.com/docker/compose/releases/downloa
d/1.24.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   617    0   617    0     0   3085      0 --:--:-- --:--:-- --:--:--  3085
100 15.4M  100 15.4M    0     0  37.8M      0 --:--:-- --:--:-- --:--:-- 37.8M
bbearce@MedICI-CodaLab-Master:~$
```

Now change the permissions in /usr/local/bin/docker-compose to use docker-compose:

```
sudo chmod +x /usr/local/bin/docker-compose
```

```
bbearce@MedICI-CodaLab-Master:~$ sudo chmod +x /usr/local/bin/docker-compose
bbearce@MedICI-CodaLab-Master:~$
```

Run `docker-compose` in the terminal window. The options list appears that tells us how to use this command. This verifies that docker-compose is installed and ready to use.

## Step 2 - Get the Source Code

The latest code is located in the codalab-competitions folder in the codalab repository (repo) in GitHub. Both sets of instructions direct you to clone this repository locally to the virtual machine. Since we will be customizing our installation, fork the repository and do the following:

1. Pull the latest changes from the master repo https://github.com/codalab/codalab-competitions into your own repo periodically to make sure you have the latest base code. (repo: https://github.com/QTIM-Lab/MedICI/). If you need help doing this on your own, see forking a repo, but you can just clone the QTIM-Lab/MedICI repo.
2. Store your customizations on GitHub so that you can clone from this repo for future projects.
3. Clone the MedICI project. This follows CodaLabs instructions but with our GitHub repo instead. Below is a picture of the commands I've executed from "Configure CodaLab from scratch (harder documentation)". The green circles represent once inside the "src" folder, the changes that are not in the raw CodaLab github repo. This is the start of our customizations.

```
bbearce@MedICI-CodaLab-Master:~$ cd ~
bbearce@MedICI-CodaLab-Master:~$ pwd
/home/bbearce
bbearce@MedICI-CodaLab-Master:~$ mkdir src
bbearce@MedICI-CodaLab-Master:~$ cd src
bbearce@MedICI-CodaLab-Master:~/src$ git clone https://github.com/QTIM-Lab/MedICI.git
Cloning into 'MedICI'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 76582 (delta 7), reused 4 (delta 2), pack-reused 76560
Receiving objects: 100% (76582/76582), 61.46 MiB | 25.49 MiB/s, done.
Resolving deltas: 100% (32829/32829), done.
bbearce@MedICI-CodaLab-Master:~/src$ ls
MedICI
bbearce@MedICI-CodaLab-Master:~/src$ cd MedICI/
bbearce@MedICI-CodaLab-Master:~/src/MedICI$ mkdir var && sudo chown bbearce:bbearce -R var
bbearce@MedICI-CodaLab-Master:~/src/MedICI$ mkdir /tmp/codalab && sudo chown bbearce:bbearce -R /
tmp/codalab
bbearce@MedICI-CodaLab-Master:~/src/MedICI$ sudo chmod 777 /tmp/codalab
bbearce@MedICI-CodaLab-Master:~/src/MedICI$ cp .env_sample .env
bbearce@MedICI-CodaLab-Master:~/src/MedICI$ ls -la
total 112
drwxrwxr-x  9 bbearce bbearce 4096 Jul 26 18:46 .
drwxrwxr-x  3 bbearce bbearce 4096 Jul 26 18:40 ..
-rw-rw-r--  1 bbearce bbearce 3570 Jul 26 18:46 .env
-rw-rw-r--  1 bbearce bbearce 4316 Jul 26 18:40 .env_production_sample
-rw-rw-r--  1 bbearce bbearce 3570 Jul 26 18:40 .env_sample
-rw-rw-r--  1 bbearce bbearce   48 Jul 26 18:40 .flake8
drwxrwxr-x  8 bbearce bbearce 4096 Jul 26 18:40 .git
-rw-rw-r--  1 bbearce bbearce 2518 Jul 26 18:40 .gitattributes
-rw-rw-r--  1 bbearce bbearce 1600 Jul 26 18:40 .gitignore
-rw-rw-r--  1 bbearce bbearce 8958 Jul 26 18:40 .pylint-conf
-rw-rw-r--  1 bbearce bbearce  353 Jul 26 18:40 Dockerfile
-rw-rw-r--  1 bbearce bbearce  628 Jul 26 18:40 ISSUE_TEMPLATE.md
-rw-rw-r--  1 bbearce bbearce  353 Jul 26 18:40 LICENSE.TXT
-rw-rw-r--  1 bbearce bbearce 1408 Jul 26 18:40 README.md
-rw-rw-r--  1 bbearce bbearce  299 Jul 26 18:40 VERSIONS.md
drwxrwxr-x  2 bbearce bbearce 4096 Jul 26 18:40 certs
-rw-rw-r--  1 bbearce bbearce 1268 Jul 26 18:40 circle.yml
-rw-rw-r--  1 bbearce bbearce  443 Jul 26 18:40 circlec_dev_setup.py
drwxrwxr-x 13 bbearce bbearce 4096 Jul 26 18:40 codalab
drwxrwxr-x  6 bbearce bbearce 4096 Jul 26 18:40 docker
-rw-rw-r--  1 bbearce bbearce 5079 Jul 26 18:40 docker-compose.yml
drwxrwxr-x  3 bbearce bbearce 4096 Jul 26 18:40 docs
drwxrwxr-x  3 bbearce bbearce 4096 Jul 26 18:40 scripts
drwxrwxr-x  2 bbearce bbearce 4096 Jul 26 18:44 var
```

4. Push these changes into your branch. At this point the app should be ready to deploy.
5. Run `$ docker compose up -d.` You may encounter the following error:

```
bbearce@MedICI-CodaLab-Master:~/src/MedICI$ docker-compose up -d
WARNING: The SSL_CERTIFICATE variable is not set. Defaulting to a blank string.
WARNING: The SSL_CERTIFICATE_KEY variable is not set. Defaulting to a blank string.
ERROR: Couldn't connect to Docker daemon at http+docker://localhost - is it running?

If it's at a non-standard location, specify the URL with the DOCKER_HOST environment variable.
```

   The solution to this error is to run with sudo or add yourself to the docker group:
   `$ sudo docker-compose up -d`

   Or
   https://docs.docker.com/install/linux/linux-postinstall/

   - `$ sudo groupadd docker` (this may be done already)
   - `$ sudo usermod -aG docker $USER` (now restart the VM)
   - Run `$ docker-compose up -d`
   - The flag '-d' means to run in detached mode. If you don't use it, you will notice your command prompt is in a live feed mode telling you everything that is going on with this Docker.

## Step 3 - Tweak the .env file and Azure Storage

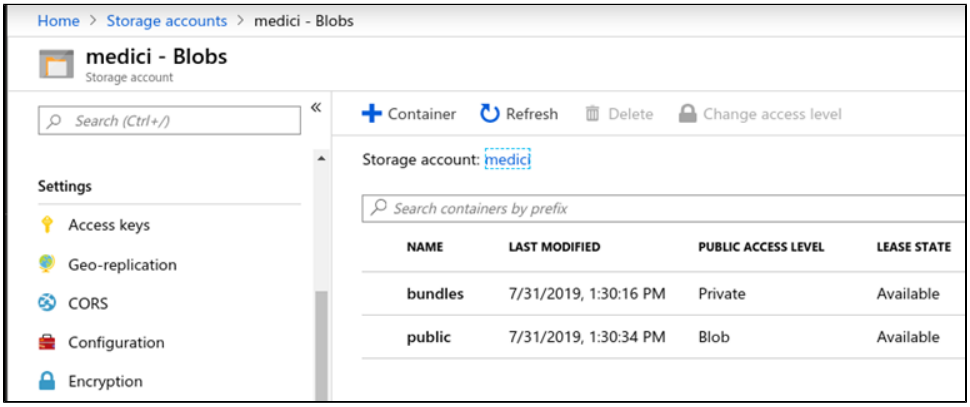
The following documentation explains this step: https://codalab-competitions.readthedocs.io/en/latest/

### Create a Storage Account

Below is the blob storage section for Azure:

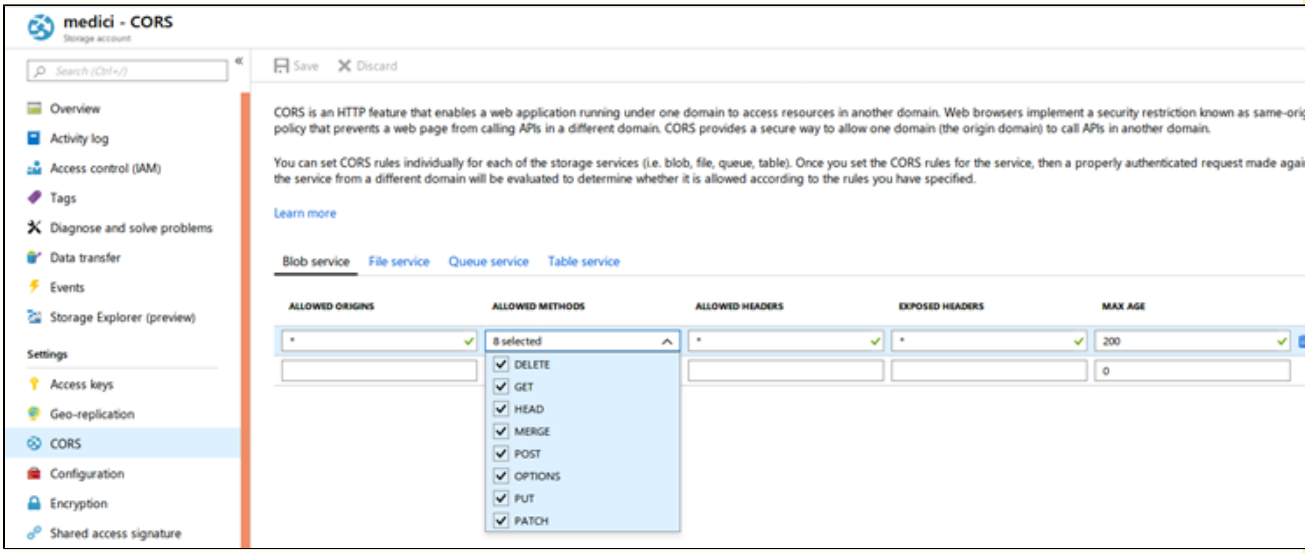
You may sign up for an Azure account, then follow the directions below. You do not have to do this if you've already set up S3.

1. Log on to the Azure Portal.
2. From the Dashboard, click **Storage accounts** on the left.
3. Click **Add** at the top of the page to create a new storage account.
4. If you don't already have a subscription, create one now. The free trial requires a credit card, and **deletes all your storage containers after 90 days**, unless you upgrade to a different plan such as 'Pay as You Go'.
5. Select the **Classic** storage account. Refer to the following image for settings.

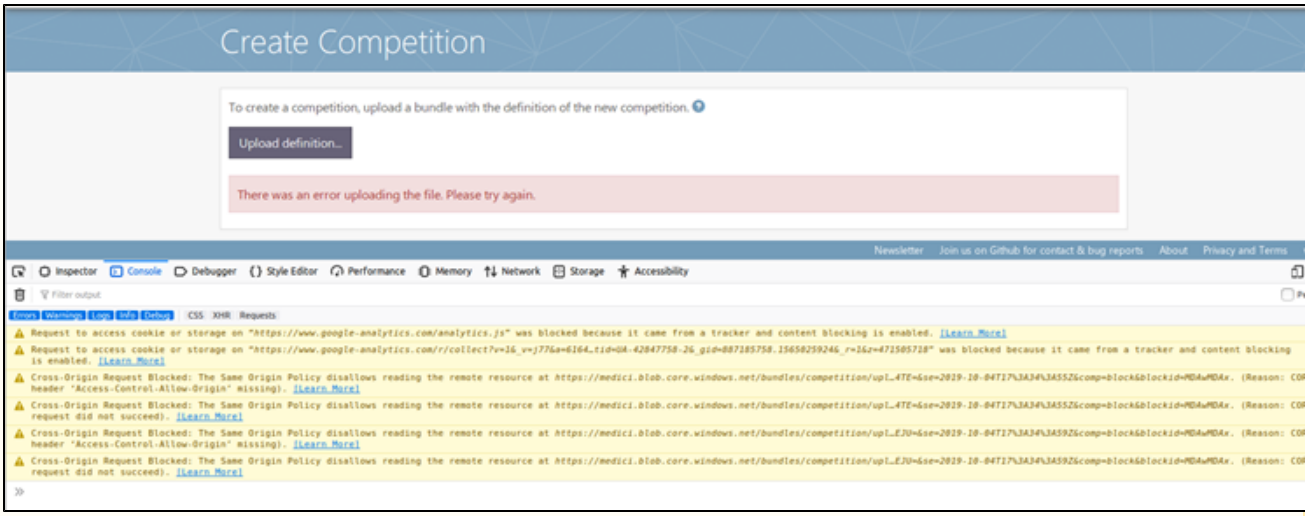



6. In the dashboard, click **All Resources/All Subscriptions** and then click your username. Click **Access Keys** and copy your account name and access key to .env under AZURE_ACCOUNT_NAME and AZURE_ACCOUNT_KEY.
7. Within that same user account, click on **Containers** and **Add a new container**.
8. Create a new container named "bundles". Set the **Access** to "Private".
9. Add another container named "public". Set the **Access** to "Public Blob".
10. Make sure the DEFAULT_FILE_STORAGE .env option is set to codalab.azure_storage.AzureStorage.

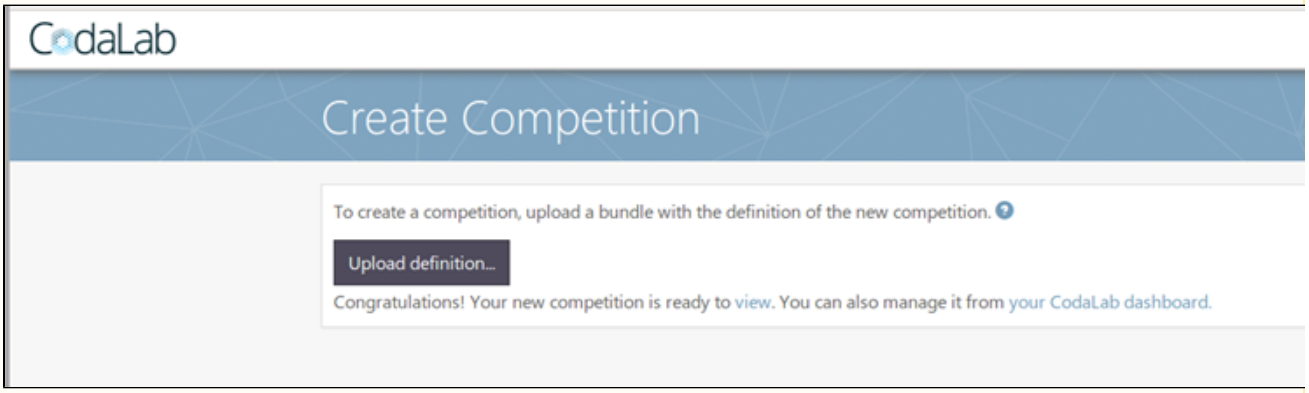
⚠ The CodaLab documentation leaves one final step out. We NEED to enable CORS. Click CORS in the image in the previous procedure and add these settings (https://stackoverflow.com/questions/50785255/azure-storage-got-403-url-cors-not-enabled-or-no-matching-rule-found-for-thi):




If you don't, uploading a competition will present a CORS errors in the browser:




Once applied, uploading works:




## Adding a Custom Logo

To create a custom logo, follow the Codalab instructions but note that you must log in to the database as the super user to use the customizer module. Django doesn't have a super user by default, so you must create it. To do so, create a secure connection to the virtual machine that hosts your site and run the following command.

```
$ docker exec -it django python manage.py createsuperuser
```

```
bbearce@MedICI-CodaLab-Master:~/src/MedICI$ docker exec -it django python manage
.py createsuperuser
/usr/local/lib/python2.7/site-packages/django_extensions/db/fields/__init__.py:4
25: DeprecationWarning: Django 1.8 features a native UUIDField, this UUIDField w
ill be removed after Django 1.7 becomes unsupported.
  warnings.warn("Django 1.8 features a native UUIDField, this UUIDField will be
removed after Django 1.7 becomes unsupported.", DeprecationWarning)

Username: bbearce
Error: That username is already taken.
Username: admin
Email address: bbearce@gmail.com
Password:
Password (again):
Superuser created successfully.
```

You can now return to the CodaLabs instructions and navigate to http://your-instance.com/customizer. Log in with your super user credentials and choose a new file as your logo.