

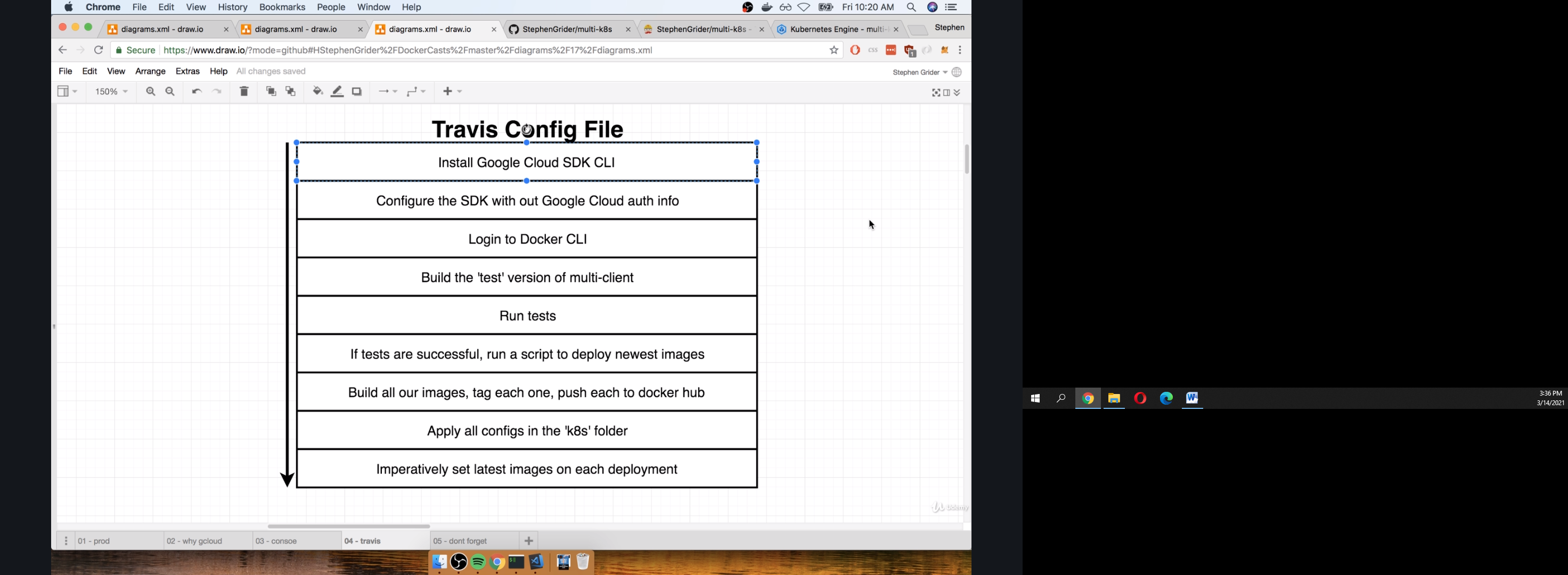
Updated GKE creation steps for new Google Cloud UI

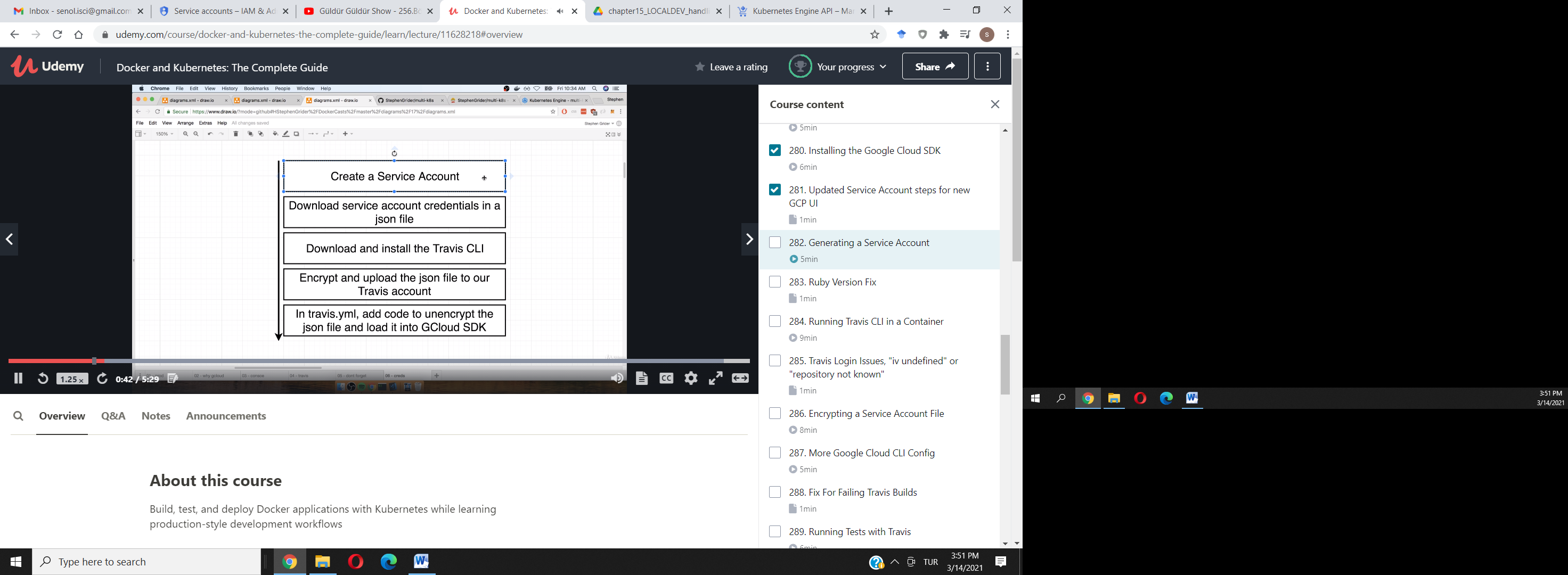
*updated 3-11-2021*

Part of the Google Cloud interface for enabling the GKE API and creating a cluster has seen some changes in the past few weeks, so, here is an updated walkthrough with screenshots. These instructions pick up after you have successfully created the new multi-k8s project.

1. Click the Hamburger menu on the top left-hand side of the dashboard.
2. Click **Kubernetes Engine**
3. Click the **ENABLE** button to enable the Kubernetes API for this project.
4. After a few minutes of waiting, clicking the **bell** icon in the top right part of the menu should show a **green** checkmark for **Enable services: container.googleapis.com**
5. If you refresh the page it should show a screen to create your first cluster. If not, click the hamburger menu and select **Kubernetes Engine** and then **Clusters**.  
   Once you see the screen below, click the **CREATE** button.
6. A **Create Cluster** dialog will open and provide two choices. Standard and Autopilot. Click the **CONFIGURE** button within the **Standard** cluster option
7. A form similar to the one shown in the video will be presented. Set the **Name** to **multi-cluster** (step 1). Confirm that the **Zone** set is actually near your location (step 2). The Node Pool that is discussed in the video is now found in a separate dropdown on the left sidebar. Click the downward-facing arrow to view the settings. No changes are needed here (step 3). Finally, click the **CREATE** button at the bottom of the form (step 4).
8. After a few minutes, the cluster dashboard should load and your multi-cluster should have a **green** checkmark in the table.

TRAVIS SETUP :





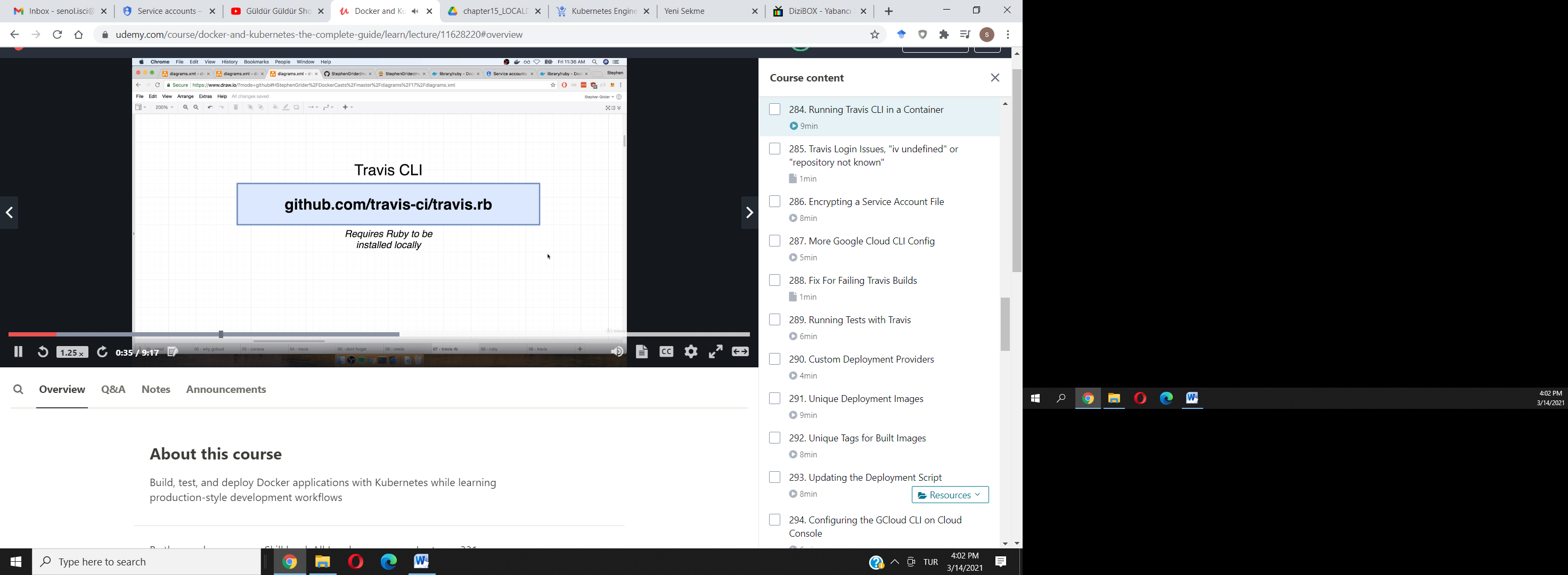
Updated Service Account steps for new GCP UI

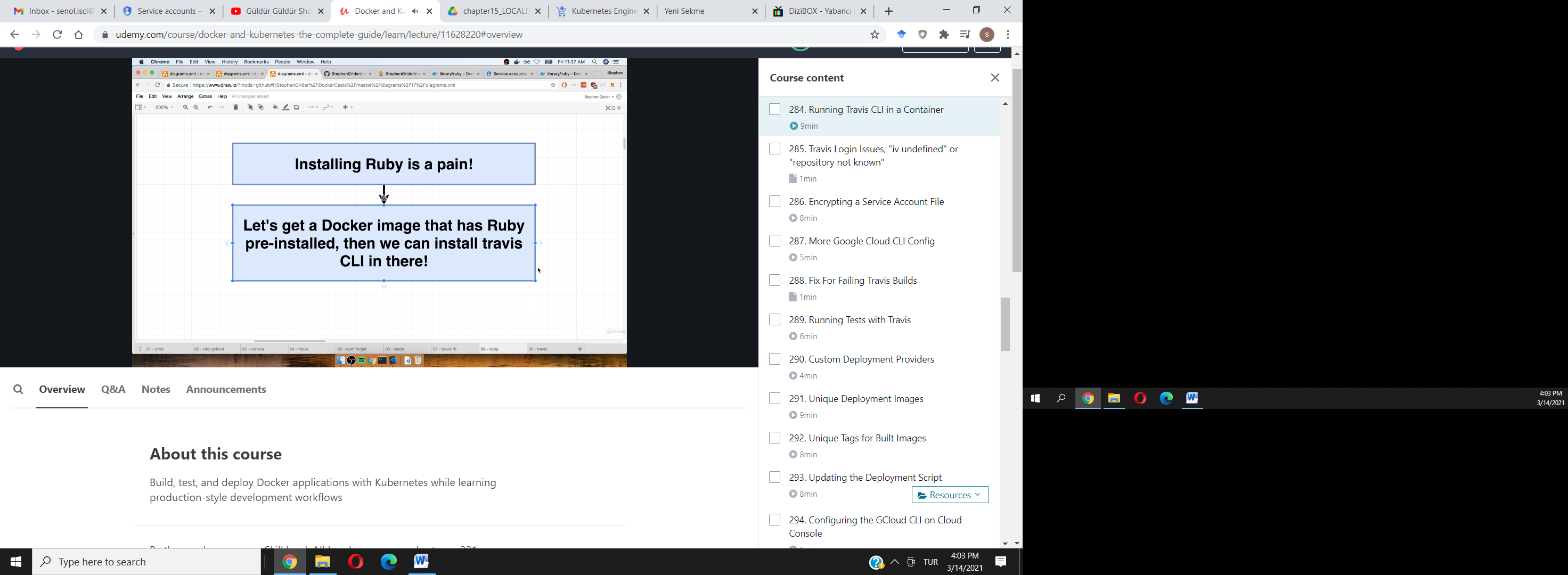
*updated 3-11-2021*

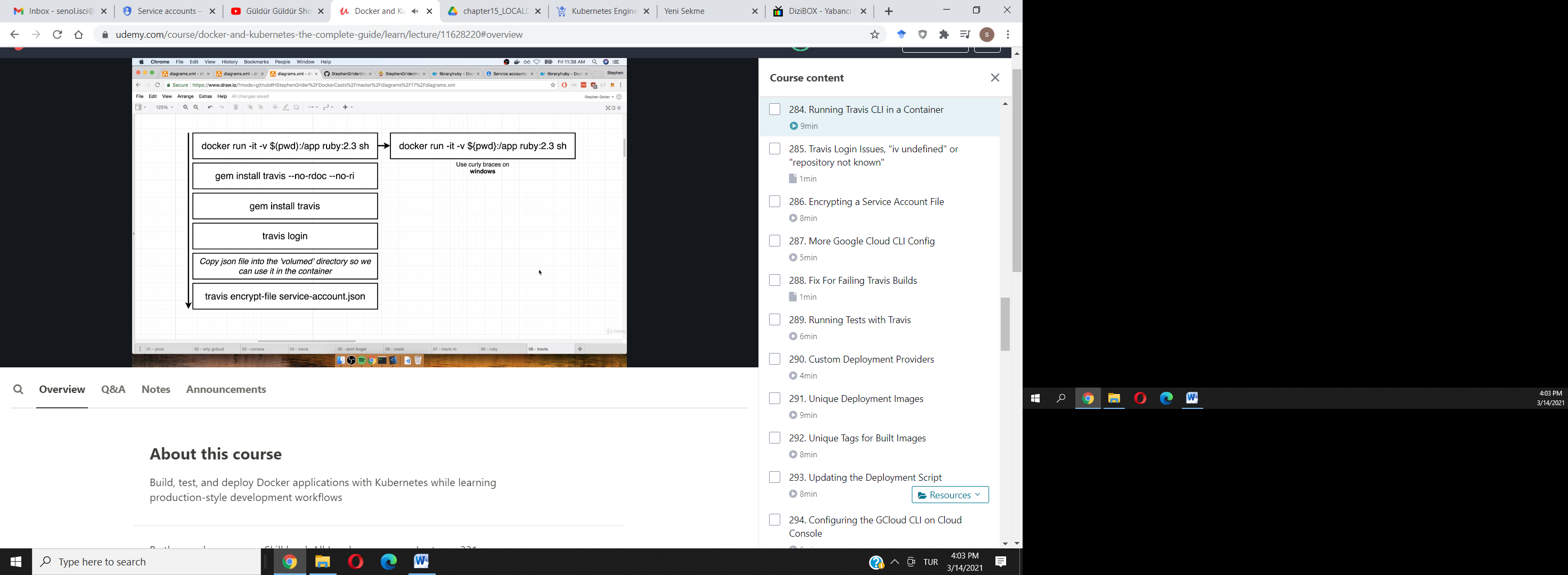
Part of the Google Cloud interface has seen some changes in the past few weeks, so, here is an updated walkthrough with screenshots to generate a Service Account.

1. Click the Hamburger menu on the top left-hand side of the dashboard, find **IAM & Admin**, and select **Service Accounts**. Then click the **CREATE SERVICE ACCOUNT** button.
2. In the form that is displayed, set the **Service account name** to **travis-deployer** (step 1), then click the **CREATE** button (step 2).
3. Click in the **Select a role** filter and scroll down to select **Kubernetes Engine** and then **Kubernetes Engine Admin**.
4. Make sure the filter now shows **Kubernetes Engine Admin** and then click **CONTINUE**
5. The Grant users access form is optional and should be skipped. Click the **DONE** button.
6. You should now see a table listing all of the service accounts including the one that was just created. Click the **three dots** to the right of the service account you just created. Then select **Manage Keys** in the dropdown.
7. In the **Keys** dashboard, click **ADD KEY** and then select **Create new key**.
8. In the **Create private key** dialog box, make sure **Key type** is set to **JSON,** and then click the **CREATE** button.
9. The JSON key file should now download to your computer

INSTALLING TRAVIS CLI:







In the upcoming lecture, we will be installing Travis inside a Docker container. This will now require using the v2.4 version of Ruby instead of v2.3 as shown in the video. The command will now look like this:

docker run -it -v ${pwd}:/app **ruby:2.4** sh

With this version, we will no longer be passing the --no-rdoc or --no-ri flags when installing Travis. The command will simply be:

gem install travis

token created in github:

name:

travisgooglecloud

token:

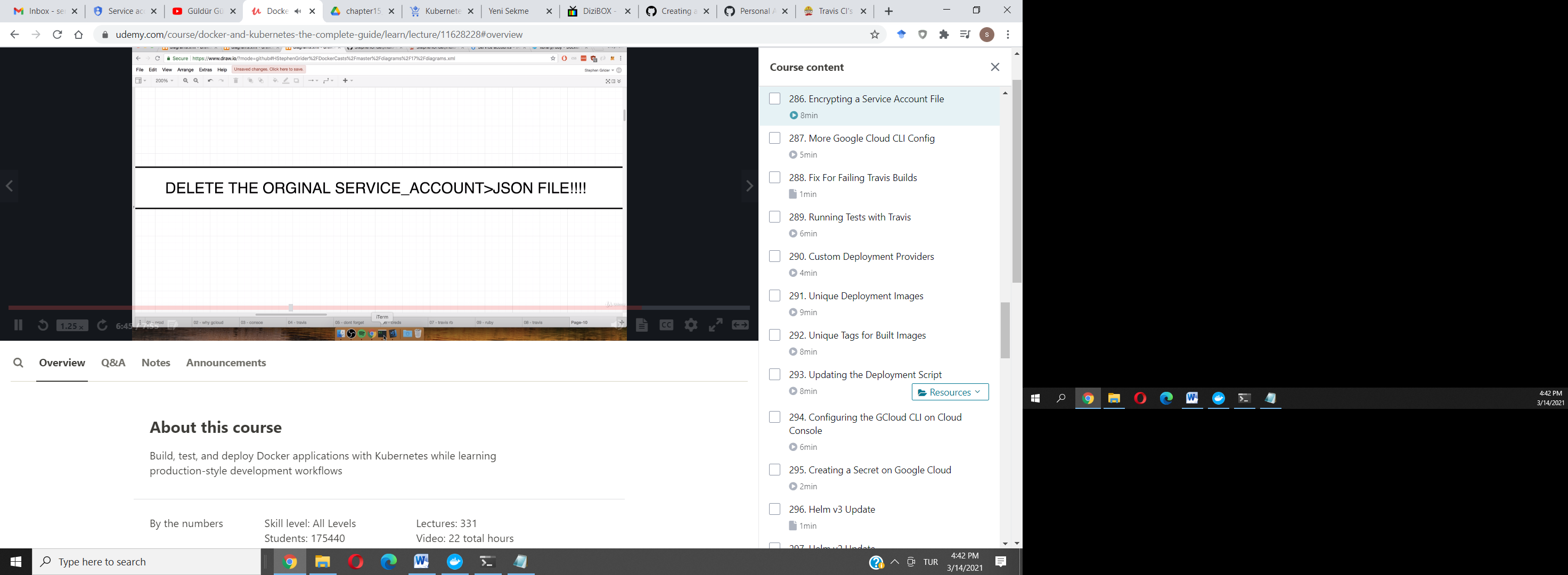
dee15708c99f425a8cf3999089d044420a3a6f4f

write this in travis cli in ruby container shell:

travis login --github-token dee15708c99f425a8cf3999089d044420a3a6f4f --com

travis encrypt-file service-account.json -r SenolIsci/multi-k8s --com

service.account.json.enc file created and do git add and commit



ravis Login Issues, "iv undefined" or "repository not known"

*updated 1-8-2021*

In the upcoming lecture, we will be encrypting a service account file in the Travis container we created in the previous lecture. Travis is finalizing its transition from .org to .com, and you may end up getting errors when attempting to log in or during deployment.

The Travis login now requires a Github Token. Please follow these instructions to create a Personal Token for Travis to use here:

<https://docs.github.com/en/free-pro-team@latest/github/authenticating-to-github/creating-a-personal-access-token>

This will also require setting the scope. Travis requires the permissions noted here:

<https://docs.travis-ci.com/user/github-oauth-scopes/#repositories-on-httpstravis-cicom-private-and-public>

**The login command will now look like this:**

*travis login --github-token YOUR\_PERSONAL\_TOKEN --com*

or

*travis login --github-token YOUR\_PERSONAL\_TOKEN --pro*

**When you encrypt the file, you must pass the same --com or --pro flag you used to log in:**

*travis encrypt-file service-account.json -r USERNAME/REPO --com*

or

*travis encrypt-file service-account.json -r USERNAME/REPO --pro*

If you are getting **iv undefined** errors, you may have missed passing the --com or --pro flags to both the login and encryption commands. Also, if you still have a .org Travis account these old accounts will need to be migrated to .com ASAP.

Please visit the migration guide here:

<https://docs.travis-ci.com/user/migrate/open-source-repository-migration#migrating-a-repository>

You can also get an **iv undefined**error if you've passed the wrong repo to the file encryption or passed a repo name with a typo in it. Please note, after the migration, or after fixing a typo, you'll need to run through the entire encryption process again.