**Docker Driver and Ingress - IMPORTANT**

We've noticed many macOS and Windows students attempting to take the course with the docker driver when using Minikube instead of Docker Desktop.

The docker driver **is not** supported for use in this course if you are on macOS or Windows. It currently does not work with any type of ingress:

<https://minikube.sigs.k8s.io/docs/drivers/docker/#known-issues>

If you have started the course with the docker driver, you will need to switch to a different driver in order to continue.

Delete your cluster:

minikube delete

Restart with a different driver:

**macOS:**

minikube start --driver=hyperkit

or

minikube start --driver=virtualbox

**Windows:**

Windows students should be using Docker Desktop with WSL2 and not Minikube. A VM driver will not work since it would require virtualization that is in conflict with WSL2. You should stop and head back to the [instructions to enable Docker Desktop's Kubernetes](https://www.udemy.com/course/docker-and-kubernetes-the-complete-guide/learn/lecture/25113234#questions) instead.

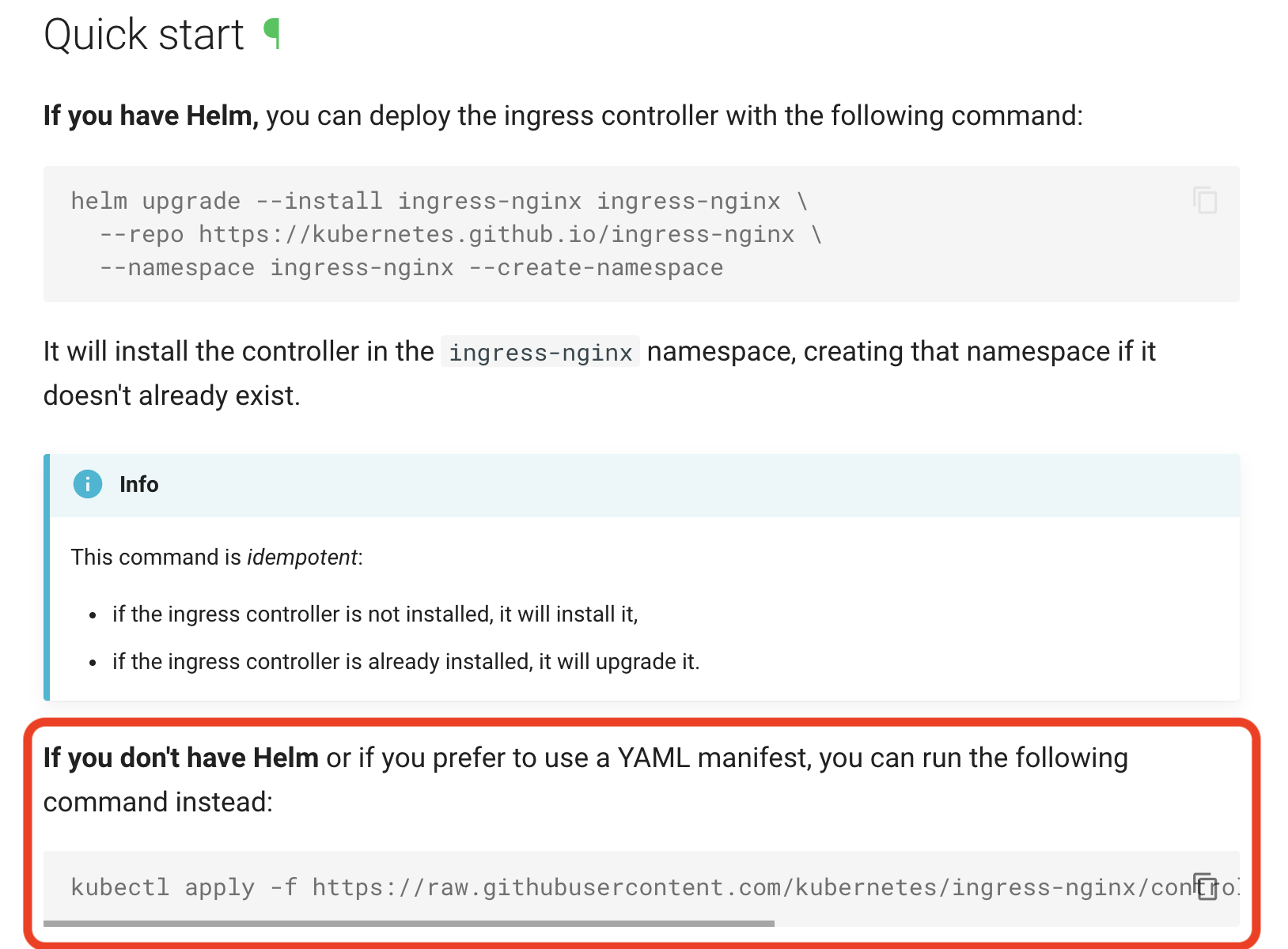
**Linux:**

If you are a Linux user, the ingress add-on should be supported when using the docker driver.

**Important - DO NOT SKIP - Ingress Nginx Installation Info**

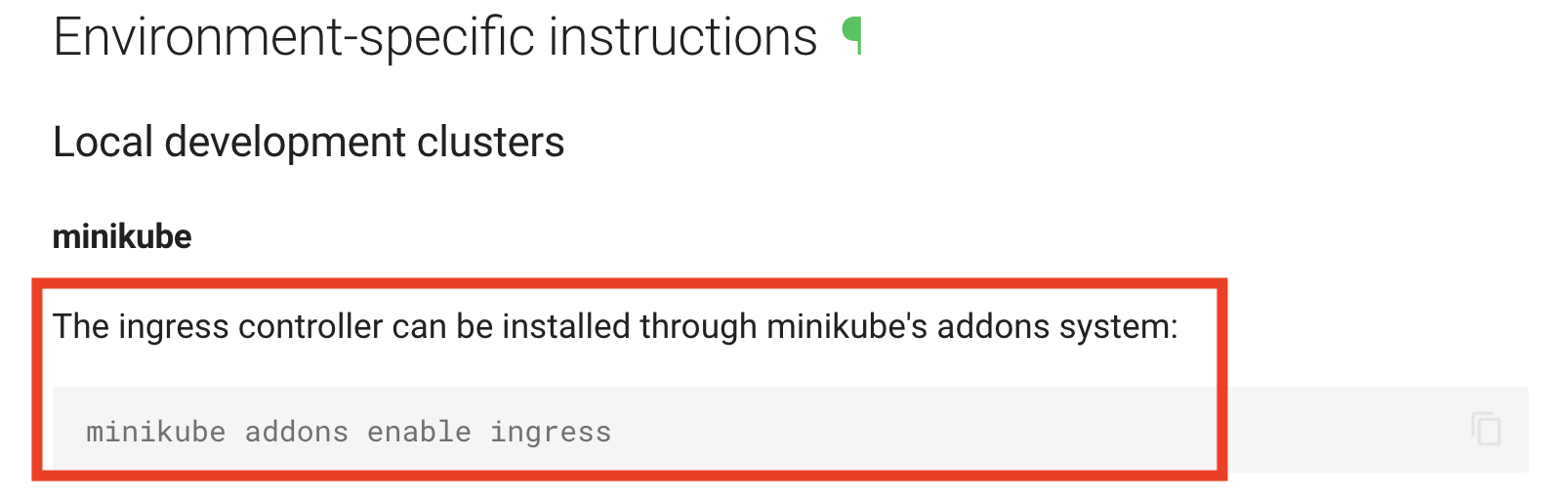
**In the upcoming lecture, we will be installing** [**Ingress Nginx**](https://kubernetes.github.io/ingress-nginx/)**. In the video, it is shown that there is a required mandatory command that needed to be run for all providers. This has since been removed, so, the provider-specific commands (Docker Desktop, Minikube, etc) are all that are required. Please triple-check that you are installing** [**Ingress Nginx**](https://kubernetes.github.io/ingress-nginx/) **and not** [**Nginx Ingress**](https://docs.nginx.com/nginx-ingress-controller/)**, which is a totally different and incompatible library.**

#### **Installation - Docker Desktop (macOS and Windows)**

****

[**https://kubernetes.github.io/ingress-nginx/deploy/#quick-start**](https://kubernetes.github.io/ingress-nginx/deploy/#quick-start)

#### **Installation - Minikube**

****

[**https://kubernetes.github.io/ingress-nginx/deploy/#minikube**](https://kubernetes.github.io/ingress-nginx/deploy/#minikube)

**Docker Desktop's Kubernetes Dashboard**

**This note is for students using Docker Desktop's built-in Kubernetes. If you are using Minikube, the setup here does not apply to you and can be skipped.**

**If you are using Docker Desktop's built-in Kubernetes, setting up the admin dashboard is going to take a little more work.**

**1. Grab the most current script from the install instructions:**

[**https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/#deploying-the-dashboard-ui**](https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/#deploying-the-dashboard-ui)

**eg:**

**As of today, the kubectl apply command looks like this:**

**kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.5.0/aio/deploy/recommended.yaml**

**2. Create a dash-admin-user.yaml file and paste the following:**

* **apiVersion: v1**
* **kind: ServiceAccount**
* **metadata:**
* **name: admin-user**
* **namespace: kubernetes-dashboard**

**3. Apply the dash-admin-user configuration:**

**kubectl apply -f dash-admin-user.yaml**

**4. Create dash-clusterrole-yaml file and paste the following:**

* **apiVersion: rbac.authorization.k8s.io/v1**
* **kind: ClusterRoleBinding**
* **metadata:**
* **name: admin-user**
* **roleRef:**
* **apiGroup: rbac.authorization.k8s.io**
* **kind: ClusterRole**
* **name: cluster-admin**
* **subjects:**
* **- kind: ServiceAccount**
* **name: admin-user**
* **namespace: kubernetes-dashboard**

**5. Apply the ClusterRole configuration:**

**kubectl apply -f dash-clusterrole.yaml**

**6. In the terminal, run kubectl proxy**

**7. Visit the following URL in your browser to access your Dashboard:**

**http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/**

**8. Obtain the token for this user by running the following in your terminal:**

**First, run kubectl version in your terminal.**

**If your Kubernetes server version is v1.24 or higher you must run the following command:**

**kubectl -n kubernetes-dashboard create token admin-user**

**If your Kubernetes server version is older than v1.24 you must run the following command:**

**kubectl -n kubernetes-dashboard get secret $(kubectl -n kubernetes-dashboard get sa/admin-user -o jsonpath="{.secrets[0].name}") -o go-template="{{.data.token | base64decode}}"**

**9. Copy the token from the above output and use it to log in at the dashboard.**

**Be careful not to copy any extra spaces or output such as the trailing % you may see in your terminal.**

**10. After a successful login, you should now be redirected to the Kubernetes Dashboard.**

**The above steps can be found in the official documentation:**

[**https://github.com/kubernetes/dashboard/blob/master/docs/user/access-control/creating-sample-user.md**](https://github.com/kubernetes/dashboard/blob/master/docs/user/access-control/creating-sample-user.md)