Having to type out kubectl commands or their corresponding options can be tedious, time-consuming, and error-prone. You can configure command autocompletion for kubectl within your shell. With a simple press of the Tab key, you can list and select matching commands and options. In this lab, you will configure autocompletion on Ubuntu with apt-get.

Configuring Autocompletion
kubectl supports integration with your shell to enable tab completion for both commands and resources. Depending on your environment, you may need to install the bash-completion package before you activate command autocompletion. You can do this using the appropriate package manager:

MacOS: brew install bash-completion
CentOS/Red Hat: yum install bash-completion
Debian/Ubuntu: apt-get install bash-completion

When installing on Ubuntu, make sure to follow the instructions from brew about how to activate tab completion using your ${HOME}/.bash_profile.

Start by installing the bash-completion package using apt-get:

apt-get install bash-completion

Once bash-completion is installed, you can temporarily activate it for your terminal using:

source <(kubectl completion bash)

The command source <(kubectl completion bash) is used to enable Bash completion for the kubectl command, which is the command-line tool for managing Kubernetes clusters.

When you run this command, it loads the kubectl completion script for Bash into your current shell session, which allows you to use tab completion to quickly and easily type out kubectl commands and their arguments.

Note that in order for this command to work, you must have the bash-completion package installed on your system.

To make this automatic for every terminal, you can add it to your ${HOME}/.bashrc_ file:

echo "source <(kubectl completion bash)" >> ${HOME}/.bashrc

```
$ apt-get install bash-completion
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bash-completion is already the newest version (1:2.11-5ubuntu1).
The following packages were automatically installed and are no longer required:
  eatmydata libeatmydata1 libflashrom1 libftdi1-2 libnetplan0 python3-json-pointer python3-jsonpatch
  python3-jsonschema python3-pyrsistent
Use 'apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
$ source <(kubectl completion bash)
$ echo "source <(kubectl completion bash)" >> ${HOME}/.bashrc
$
```

The command echo "source <(kubectl completion bash)" >> ${HOME}/.bashrc appends the line source <(kubectl completion bash) to the end of the ${HOME}/.bashrc file.

The ${HOME}/.bashrc file is a script that is executed by Bash whenever a new interactive shell is launched. By appending the source <(kubectl completion bash) line to this file, you ensure that the kubectl completion script is loaded into your shell session automatically every time you launch a new terminal window or tab.

This can be a useful time-saver, as it eliminates the need to manually type the source <(kubectl completion bash) command every time you open a new terminal.
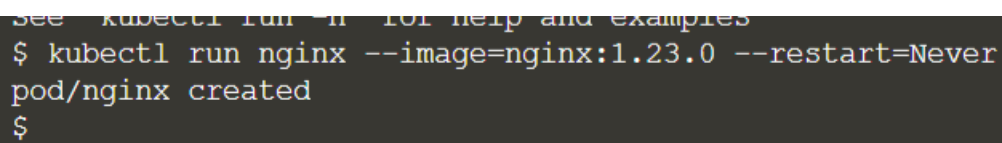
Using Autocompletion
You can now try out the autocompletion functionality for creating a Pod.

Type in kubectl ru and press the Tab key. The autocompletion functionality will simply select the run command, as it is the only matching option for the initial characters. Type in the name nginx.

Now type the option --im and press the Tab key. You will see that --image= is proposed. Enter the value nginx:1.23.0 after the = character. You can further explore the available values of the --restart command-line option.

The command could look as follows:

kubectl run nginx --image=nginx:1.23.0 --restart=Never

```
See "kubectl run -h" for help and examples
$ kubectl run nginx --image=nginx:1.23.0 --restart=Never
pod/nginx created
$
```

Using autocompletion commands will autocompleted on own .

kubectl is a powerful tool for managing objects in a Kubernetes cluster. You can configure autocompletion functionality so you don't have to remember every single command and CLI option. Make the Tab key a real time-saver!