Lab 17: Terraform Variables

In this lab, we're going to cover a pretty foundational concept in TerraForm and well, pretty much any programming construct, and that is variables.

Variables allow us to configure our deployments dynamically and more succinctly. We define them in one place and they can define our entire deployment.

So let's take a look at how to configure and import our variables.

You basically define your variable and you can set defaults, you can use objects, maps, pretty much any type of thing you can do in TerraForm, you can do with variables.

Now you can input from the command line, you can input from directly from your main.tf file, you can input from Variables.tf files.

But we're going to start it from our `Main.tf` file and then we'll go from there.

So let's go ahead and take a look at some of the places where we might want to have variables. Now, you can put variables for just about any argument you want.

So let's start out by just specifying the external port. Is I'm going to create a variable. We're going to name that variable, we'll name it `ext_port`. Then we'll open and close some brackets.

```
terraform {
  required_providers {
    docker = {
      source  = "terraform-providers/docker"
      version = "~> 2.7.2"
    }
  }
}

provider "docker" {}

variable "ext_port" {}

resource "docker_image" "nodered_image" {
  name  = "nodered/node-red:latest"
}

resource "docker_container" "nodered_container" {
  image = docker_image.nodered_image.latest
  name  = "nodered"

  ports {
    internal = 1880
    external = var.ext_port
  }
}
```

Go to external port and let's replace that with var.ext_port. That's how you reference your variable.

And first up, Let's do a TerraForm Destroy.

```
terraform destroy --auto-approve
```

And as you can see, it's going to pop up and ask you for that variable. type in 1880.

Let's do a TerraForm plan once again asks for the variable.

If you do not define a default, it's going to ask you every single time. Now, let's take a look at the TerraForm plan.

```
terraform plan
```

Now another way you can simplify things is to go ahead and type in var here, do var, XT port equals So let's try that.

```
terraform plan -v ext_port=1880
```

There we go. It didn't ask and it went ahead and substituted the var properly.

So now that we've got our TerraForm plan, that's exactly what we want to do. We want to create a new image. And now we're going to terraform apply.

```
terraform apply
```

Verify the existence of the container by visiting `http://YOUR_VM_DNS_NAME.courseware.io:1880` in your web browser or running docker ps to see the container.