

Working with the Code in the DevPod Using Browser-Based IDE

This lesson shows how we can modify the code using the built-in browser-based IDE provided by the DevPod.

We'll cover the following

- How can we modify the code?
- Working with the code remotely
 - ⚠ A note to Windows users
- Testing automatic build and deployment

How can we modify the code?

We could go back to the DevPod and modify the code from a terminal. We could use `vi` or a similar editor for that. While I do use terminal editors quite often, I find them sub-optimum when working on a project. I believe that `vi`, `emacs`, `nano`, and similar editors are useful when working on individual scripts, but not that great when working on a full-fledged project. Call me lazy, but I need an IDE like the Visual Studio Code, IntelliJ, Eclipse, or something similar. I need syntax highlighting, code complete, the ability to jump into a function with a single click, and other goodies provided by IDEs.

The problem is that the code we're interested in a DevPod running inside our cluster. That means that we need to synchronize our local files from a laptop to the DevPod or we can work with the code remotely. For now, we're interested in the latter option (we'll explore the former later). If we are to work with remote files, and we are not (yet) going to synchronize files between our laptop and the DevPod, the only available option is to use a remote IDE (unless you want to stick to `vi` or some other terminal-based editor).

Working with the code remotely

We can open the IDE through the command that was given to us when we created

the DevPod. But, since we already saw how easy it is to port forward with

Kubernetes, we don't need to go back to that output. We can just as easily construct our own command.

```
kubect1 port-forward \  
service/$GH_USER-go-ide 8086:80 &
```



We created yet another port forwarding. Now we should be able to open the IDE through localhost on the port 8086.

A note to Windows users

Git Bash might not be able to use the `open` command. If that's the case, replace `open` with `echo`. As a result, you'll get the full address that should be opened directly in your browser of choice.

```
open "http://localhost:8086"
```



What you see in front of you is the Visual Studio Code. It is a browser-based IDE (it can run as a desktop app as well). It is, in my experience, the best browser-based IDE today. If you've already used the Visual Studio Code on your laptop, this will feel familiar. If you haven't, it's intuitive and straightforward, and you'll have no problem adapting it (if you think it's useful).

Let's give Visual Studio Code a spin.

Testing automatic build and deployment

Our next mission is to modify a few Go files and observe that the changes are built and deployed without us executing any additional commands. Remember that the watcher (`watch.sh`) is still running.

Please expand the `go-demo-6` directory, and double-click the `main.go` file to open it in the main body of the IDE. Next, change `hello, world` (or whatever else you changed it to previously) to `hello, devpod`.

Since we have tests that validate that the correct message is returned, we'll need to change them as well. Open the `main_test.go` file next, search for `hello, world` (or whatever else you changed it to previously) and change it to `hello, devpod`.

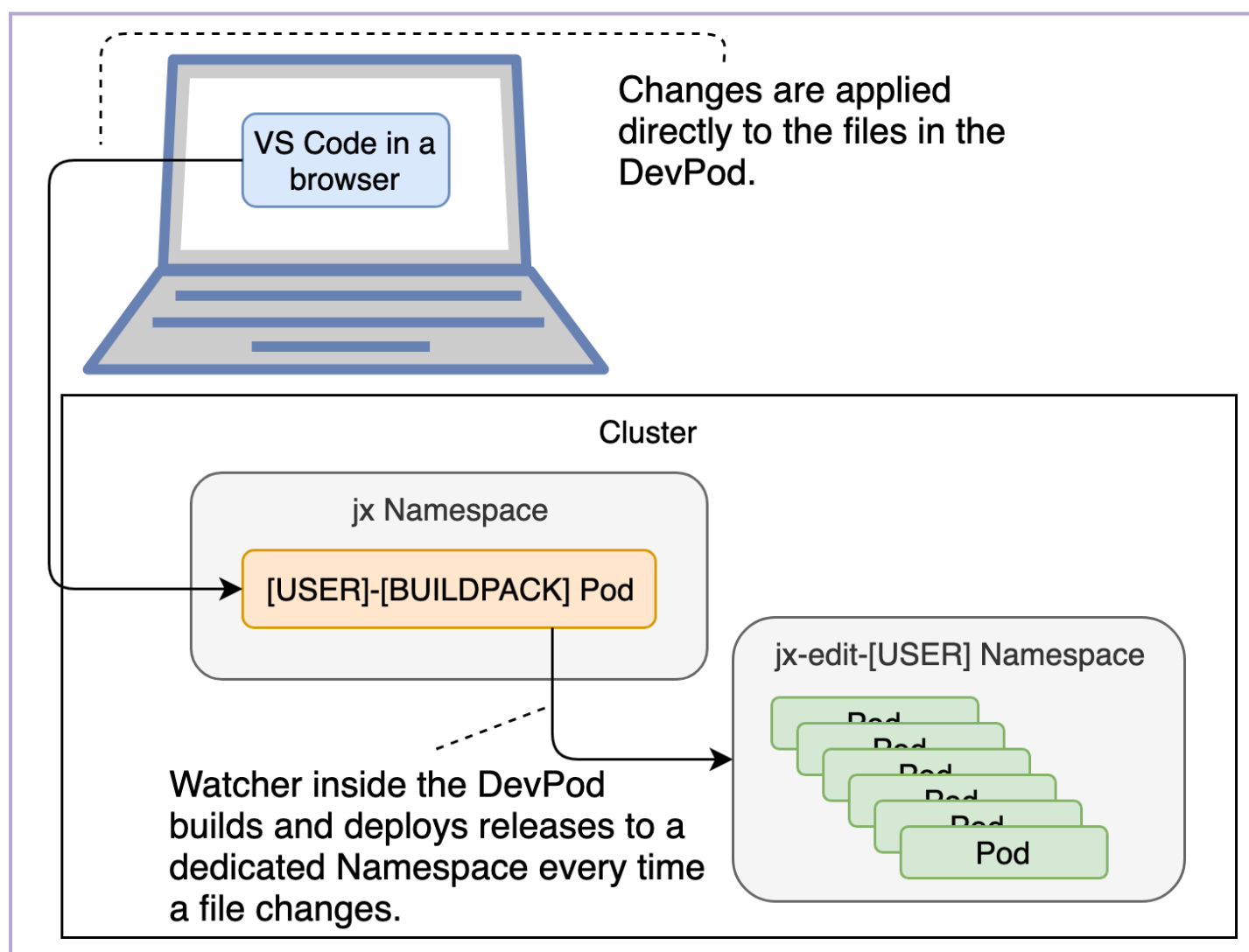
Whatever else you changed it to previously), and change it to `hello, devpod`.

Make sure to save the changes to both files.

Now we can confirm that our changes are automatically built and deployed every time we change a Go source code. However, since a new Pod was created as a result of the change, we'll need to port forward again.

```
kubectl --namespace jx-edit-$GH_USER \  
  port-forward service/go-demo-6 \  
  8087:80 &  
  
curl "http://localhost:8087/demo/hello"
```

The output should be `hello, devpod`!



The process of using Visual Studio Code to modify files inside a DevPod and building and deploying applications

We saw how we could work using personal development environments and modifying them from a browser.

I used Visual Studio Code quite a few times. It is beneficial when we don't have all the tools we need in our local environment (e.g., GH, Docker, Kubernetes, etc.).

the tools running inside our laptops (except `jx` CLI). But, a browser-based editor might not be your cup of tea. You might find a desktop IDE easier and faster. Or, maybe you are emotionally attached to a desktop version of Visual Studio Code, IntelliJ, or whatever else is your coding weapon of choice is. Fear not, we can use them as well. Our next mission is to connect your favorite IDE with DevPods. But, before we do that, we'll delete the DevPod we're currently running and start a new one with a twist.

```
killall kubectl
```

```
jx delete devpod $GH_USER-go
```



The first command deleted all the processes that were doing port forwarding, while the second started the process of deleting the DevPod. Please type `y` when asked whether you want `to delete the DevPods` and press the enter key.

The DevPod is no more.

The problem is that we did not push the code changes to GitHub. If we did that, we could pull them to our local hard disk. Since we forgot that vital step, our changes were lost the moment we deleted the DevPod.

That was silly of me, wasn't it? Or maybe I did that intentionally just to show you that we can also synchronize files from your laptop into the DevPod, and vice versa. We are going to do it the other way around in the next lesson.