



Lab: Terraform Auto Complete

One of the features of the Terraform CLI is that you can install a tab-completion if you are using `bash` or `zsh` as your command shell. This provides you with tab-completion support for all command names and *some* command arguments.

- Task 1: Install Terraform Auto Complete
- Task 2: Test Auto Complete via CLI
- Task 3: Install VSCode Extension for Auto Completion

Task 1: Install Terraform Auto Complete

```
$ terraform -install-auto-complete
```

Task 2: Test Auto Complete via CLI

In your terminal, type `terraform` and hit the `tab` key. Notice how auto complete has provided you a list of all the sub-commands available for use:

```
$ terraform
apply          destroy        fmt            get            import
login          output         providers      refresh        state
test           validate      workspace
console        env            force-unlock   graph          init
logout         plan           push           show           taint
untaint        version
```

Next, type `terraform state` and hit the `tab` key (note the space after state). Notice how auto complete has provided with additional options available for the terraform state command:

```
$ terraform state
list  mv  pull  push  replace-provider  rm  show
```

Task 3: Install VSCode Extension for Auto Completion

In VSCode, select “Extensions” from the left navigation panel. Search for `terraform`. Install the HashiCorp Terraform extension verified from HashiCorp.





Task 4: Use Auto Completion in VSCode

In your `main.tf` file, scroll to the bottom and add a few new lines. Type in `resource` (or just part of the word) and hit the tab key. See how the extension creates a new resource block for you and suggests resource types as you start typing. Type in `aws_ins` and you can tab complete on `aws_instance` to specify a new EC2 instance resource.

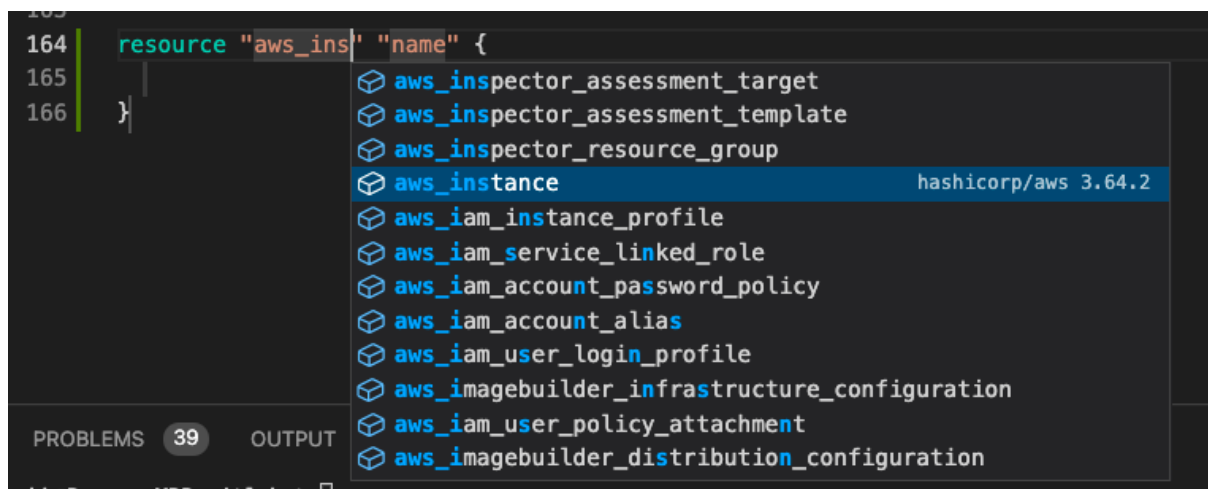


Figure 1: Auto Complete VSCode

Troubleshooting

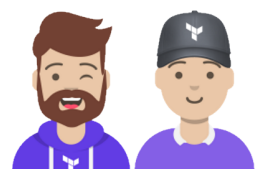
If you get an error message that says `Error executing CLI: Did not find any shells to install` you might need to update, or create, your `~/.zshrc` or `~/.profile` file to enable auto-complete. If you are using `zsh`, which is now the default on MacOS, you'll only need `~/.zshrc`. If you're using `bash`, you'll just need `~/.profile`. If this file does not yet exist, you can run the command:

```
# for zsh
touch ~/.zshrc

# for bash
touch ~/.profile
```

Try to install Terraform Auto Complete again.

If you get an error message that says `complete:13: command not found: compdef`, then you should add the following to the file mentioned above:





```
autoload -Uz compinit && compinit
```

With that now added, your `~/ .zshrc` file should now look similar to this:

```
autoload -U +X bashcompinit && bashcompinit
autoload -Uz compinit && compinit
complete -o nospace -C /usr/local/bin/terraform terraform
```

You can restart your session or simply run `exec zsh` to reload the configuration.

