Lesson 8 Demo 2: Publish to Puppet Forge

This section will guide you to:

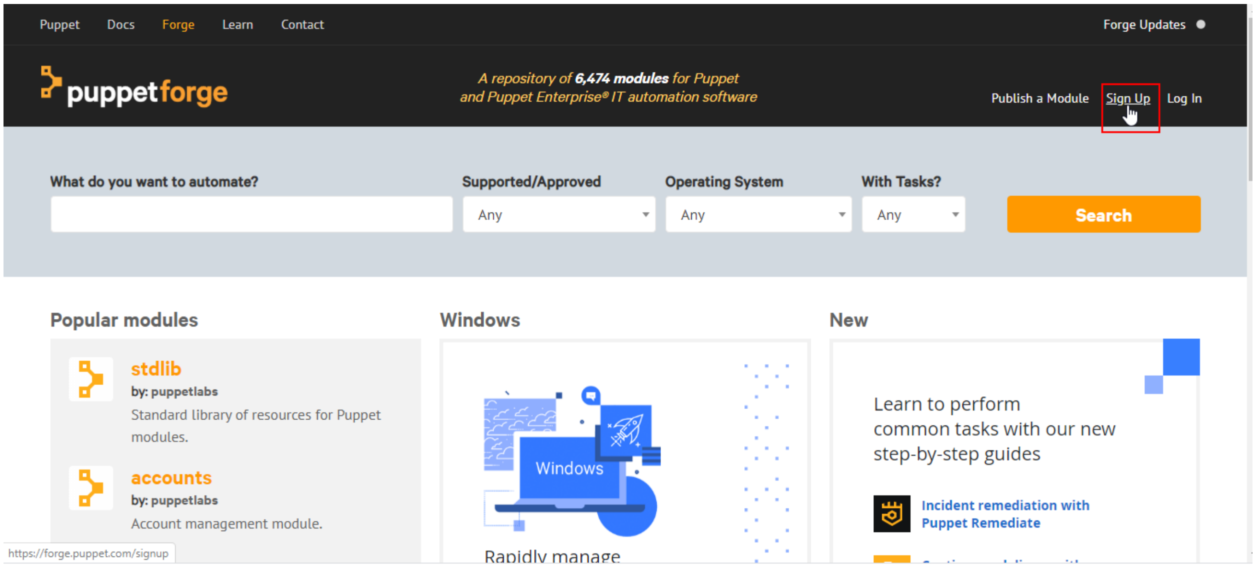
- Upload and publish a custom module to Puppet Forge

This lab has five subsections, namely:

1. Creating a Forge account
2. Preparing the custom module
3. Adding metadata.json file in the module
4. Building a module package
5. Uploading the module package to Puppet Forge

Step 1: Creating a Forge account

- Go to forge.puppet.com and click on **Sign Up** as shown in the following screenshot:



*A repository of **6,474 modules** for Puppet and Puppet Enterprise® IT automation software*

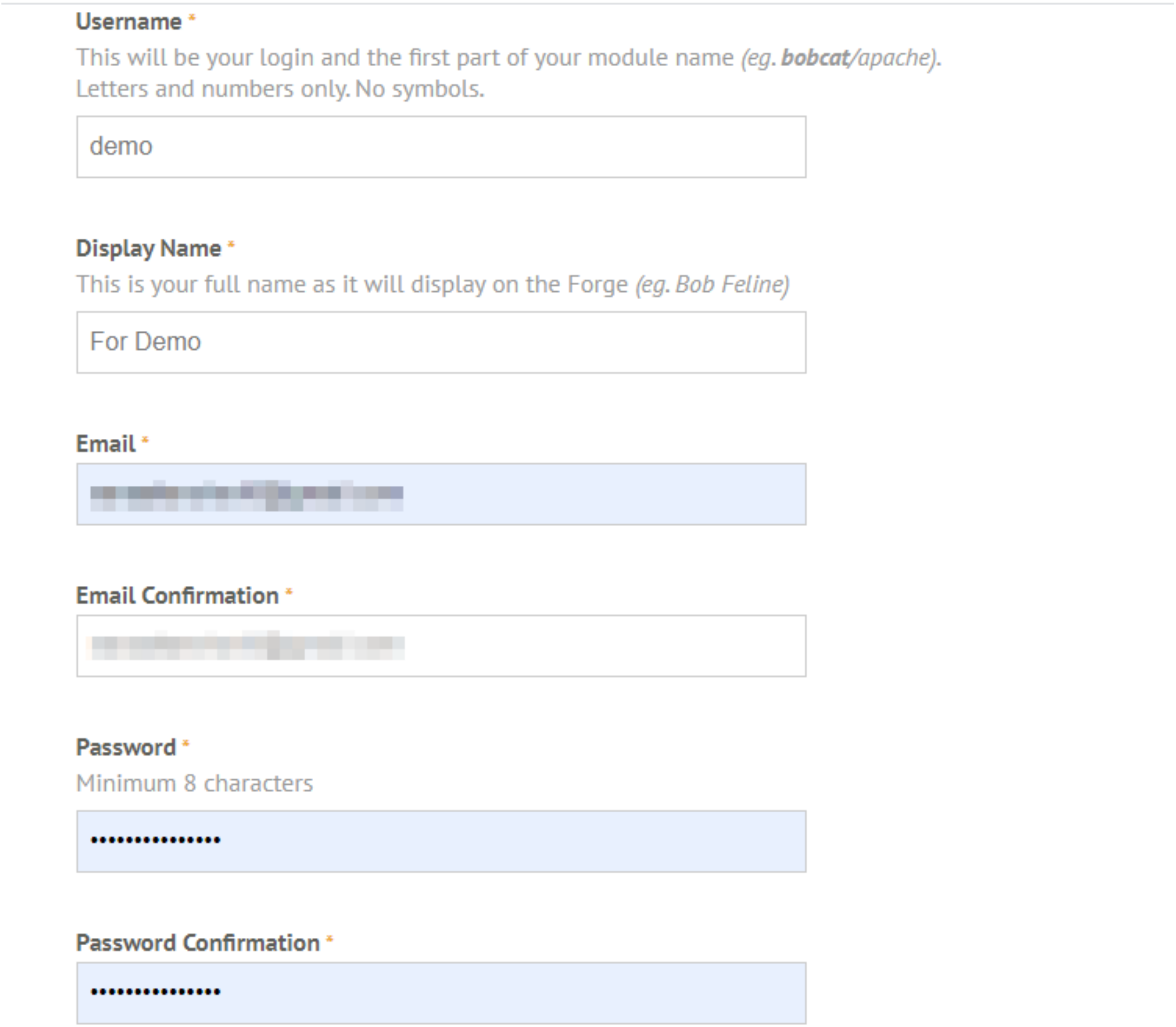
Standard library of resources for Puppet modules.

Account management module.

Rapidly manage

Learn to perform common tasks with our new step-by-step guides

- Fill in the information on the sign-up page.
- Note: The username you choose will become a part of your module long name



Username *

This will be your login and the first part of your module name *(eg. **bobcat**/apache)*. Letters and numbers only. No symbols.

demo

Display Name *

This is your full name as it will display on the Forge *(eg. Bob Feline)*

For Demo

Email *

[redacted]

Email Confirmation *

[redacted]

Password *

Minimum 8 characters

•••••••••••••••

Password Confirmation *

•••••••••••••••

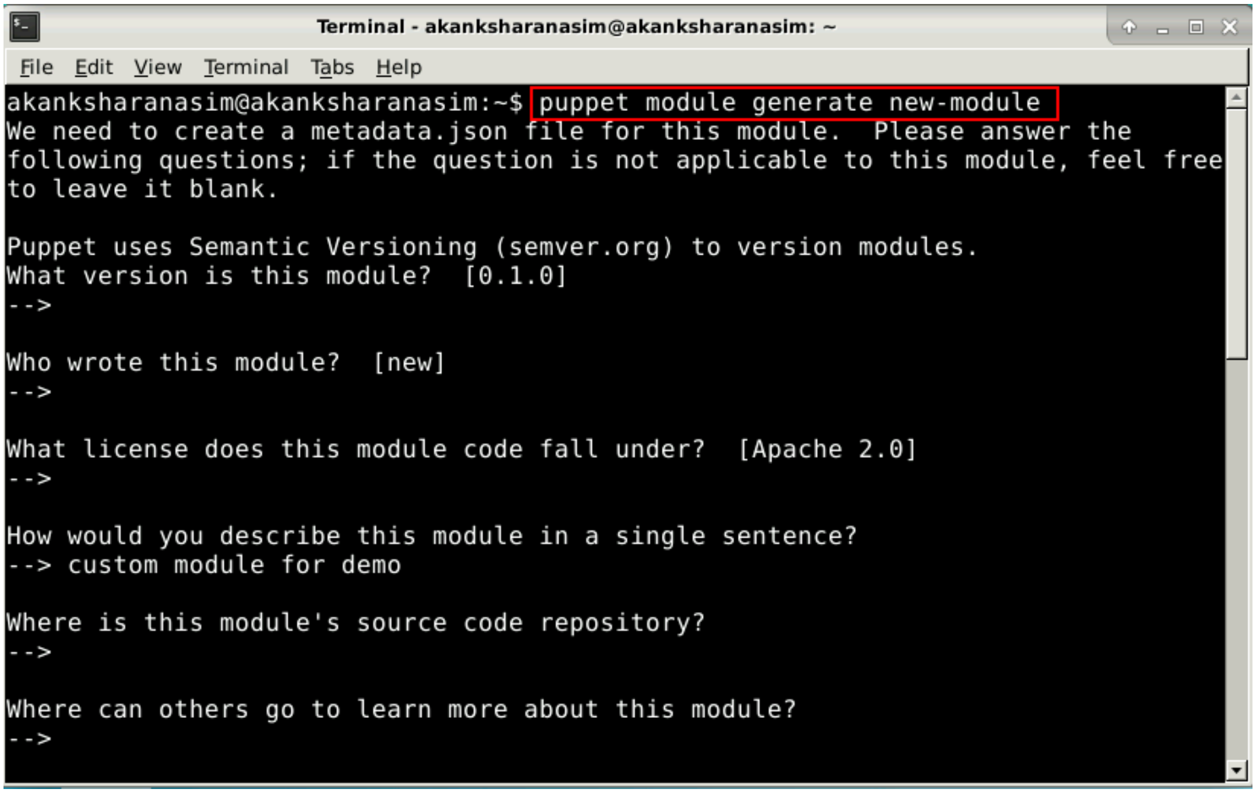
- Follow the email verification steps sent to the email you provided in the previous step
- Go to forge.puppet.com and log in to your account

Step 2: Preparing the custom module

- Open the command line terminal on the system where Puppet has been installed
- Type in the following command to create the module structure:

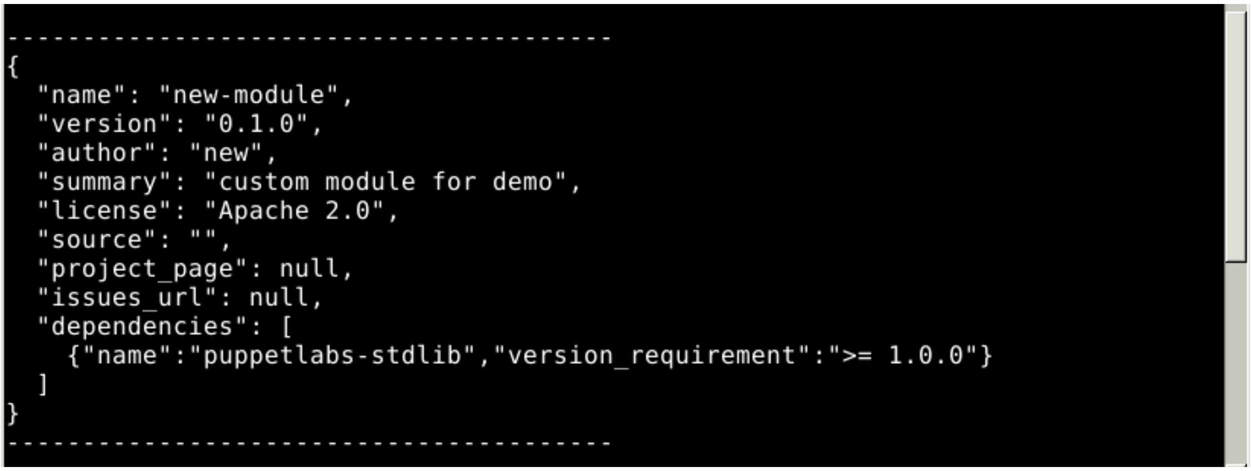
```
puppet module generate newmodule
```

- The above modified command will prompt a series of questions that will be used to create the metadata.json file for the module
- Press **Enter** to use the default answers for the questions


Terminal - akanksharanasim@akanksharanasim: ~
File Edit View Terminal Tabs Help

```
akanksharanasim@akanksharanasim:~$ puppet module generate new-module
We need to create a metadata.json file for this module.  Please answer the
following questions; if the question is not applicable to this module, feel free
to leave it blank.

Puppet uses Semantic Versioning (semver.org) to version modules.
What version is this module?  [0.1.0]
-->

Who wrote this module?  [new]
-->

What license does this module code fall under?  [Apache 2.0]
-->

How would you describe this module in a single sentence?
--> custom module for demo

Where is this module's source code repository?
-->

Where can others go to learn more about this module?
-->
```

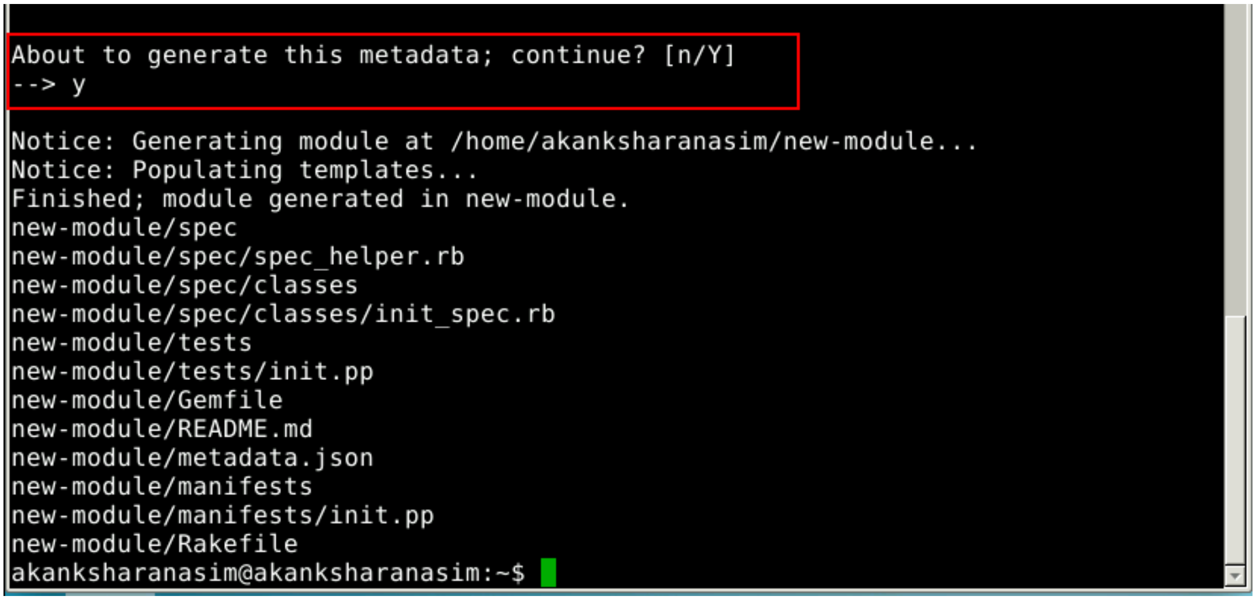
- Once the json file is created it will be displayed on the terminal as shown in the screenshot below:



```
----------------------------------------
{
  "name": "new-module",
  "version": "0.1.0",
  "author": "new",
  "summary": "custom module for demo",
  "license": "Apache 2.0",
  "source": "",
  "project_page": null,
  "issues_url": null,
  "dependencies": [
    {"name":"puppetlabs-stdlib","version_requirement":">= 1.0.0"}
  ]
}
----------------------------------------
```

Step 3: Adding metadata.json file in the module

- To generate a metadata.json file, type y when prompted with the question: about to generate metadata; continue?



```
About to generate this metadata; continue? [n/Y]
--> y

Notice: Generating module at /home/akanksharanasim/new-module...
Notice: Populating templates...
Finished; module generated in new-module.
new-module/spec
new-module/spec/spec_helper.rb
new-module/spec/classes
new-module/spec/classes/init_spec.rb
new-module/tests
new-module/tests/init.pp
new-module/Gemfile
new-module/README.md
new-module/metadata.json
new-module/manifests
new-module/manifests/init.pp
new-module/Rakefile
akanksharanasim@akanksharanasim:~$ 
```

- The basic module structure will be created at /home/rootuser/modulename path

- The default files and templates will be populated inside the module's respective directories

Step 4: Building a module package

- Use the following command to navigate to the module directory:

```
cd
```

- Build a package using the following command:

```
pdk build
```

- Use the default answers for the questions prompted by pressing **Enter**
- At the confirmation prompt, confirm the package creation

Step 5: Uploading the module package to Puppet Forge

- At the confirmation prompt, confirm the package creation
- Click on **Publish** button
- On the upload page, click **Choose File**
- Locate and select the tarball built in previous steps, then click **Upload Release**