Lesson Description - Building and Sharing a Wheel Distribution

For our internal tools, there’s a good chance that we won’t be open sourcing every little tool that we write, but we will want it to be distributable. The newest and preferred way to distribute a python tool is to build a ‘wheel’.

Let’s set up our tool now to be buildable as a wheel so that we can distribute it.

Documentation For This Video

- The `wheel` documentation
  </ul>

Adding a `setup.cfg`

Before we can generate our `wheel`, we’re going to want to configure `setuptools` to not build the wheel for Python 2. We can’t build for Python 2 because we used string interpolation. We’ll put this configuration in a `setup.cfg`:

setup.cfg

```
[bdist_wheel]
python-tag = py36
```

Now we can run the following command to build our wheel:

```
(pgbackup-E7nj_BsO) $ python setup.py bdist_wheel
```

Next, let’s uninstall and re-install our package using the wheel file:

```
(pgbackup-E7nj_BsO) $ pip uninstall pgbackup
(pgbackup-E7nj_BsO) $ pip install dist/pgbackup-0.1.0-py36-none-
any.whl
```

Install a Wheel From Remote Source (S3)

We can use pip to install wheels from a local path, but it can also install from a remote source over HTTP. Let’s upload our wheel to S3 and then install the tool outside of our virtualenv from S3:

```
(pgbackup-E7nj_BsO) $ python
>>> import boto3
>>> f = open('dist/pgbackup-0.1.0-py36-none-any.whl', 'rb')
>>> client = boto3.client('s3')
>>> client.upload_fileobj(f, 'pyscripting-db-backups',
'pgbackup-0.1.0-py36-none-any.whl')
>>> exit()
```

We’ll need to go into the S3 console and make this file public so that we can download it to install.

Let’s exit our virtualenv and install pgbackup as a user package:

```
(pgbackup-E7nj_BsO) $ exit
$ pip3.6 install --user https://s3.amazonaws.com/pyscripting-db-
backups/pgbackup-0.1.0-py36-none-any.whl
$ pgbackup --help
```