Munki

In depth documentation

Click the link - https://github.com/munki/munki/wiki/Demonstration-Setup (https://github.com/munki/munki/wiki/Demonstration-Setup)

Helpful video to get you started - https://youtu.be/ZDSf4bo5Zas (https://youtu.be/ZDSf4bo5Zas) 3 parts.

Software

Munki Admin (Server Computer)

```
https://github.com/hjuutilainen/munkiadmin/releases
(https://github.com/hjuutilainen/munkiadmin/releases)
```

Managed Software Center. All client computers (labs)

```
https://github.com/munki/munki/releases
(https://github.com/munki/munki/releases)
```

AutoPkgr

```
https://github.com/autopkg/autopkg/releases
(https://github.com/autopkg/autopkg/releases)
```

Creating Munki Repository

- Choose a location in memory where the repository shall reside
- In your designated location. /Volumes/Storage/ make a Data folder.
- In the Data folder make a folder called munki.
- Now open the terminal, designate to /Volumes/Storage/Data/munki/.
- Here you will create a directory. "mkdir munki_repo."
- Inside munki_repo make 4 more directories
- 1. "mkdir catalog" 2. "mkdir manifests" 3. "mkdir pkgs" 4. "mkdir pkgsinfo".

- Set the correct permissions on these directories "chmod -R a+rX ../munki_repo"

```
munkiserver:munki_repo admin$ pwd
/Volumes/DataHD/Munki/munki_repo
munkiserver:munki_repo admin$ chmod -R a+rX ../munki_repo
munkiserver:munki_repo admin$ █
```

Munki Admin Configuration

Munki Admin is the tool we use to choose which software goes where.
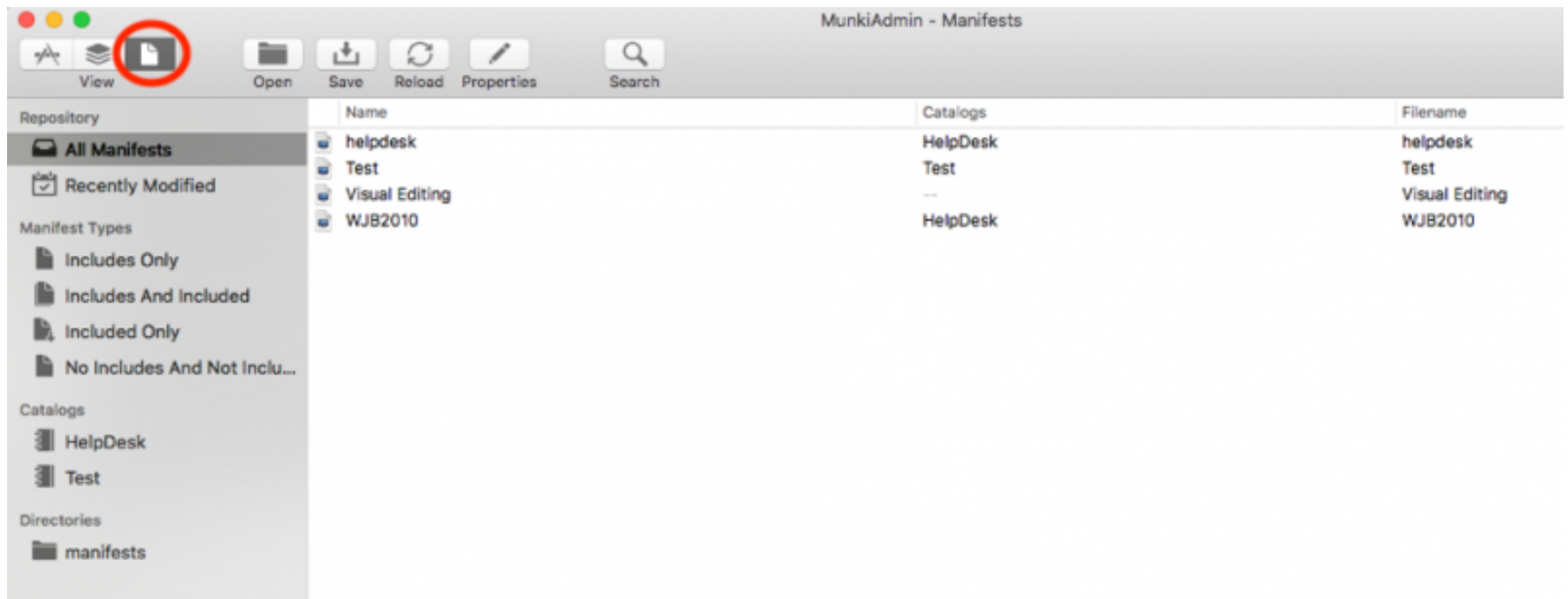
First things first. Choose your repo. Click Open and locate your repo.
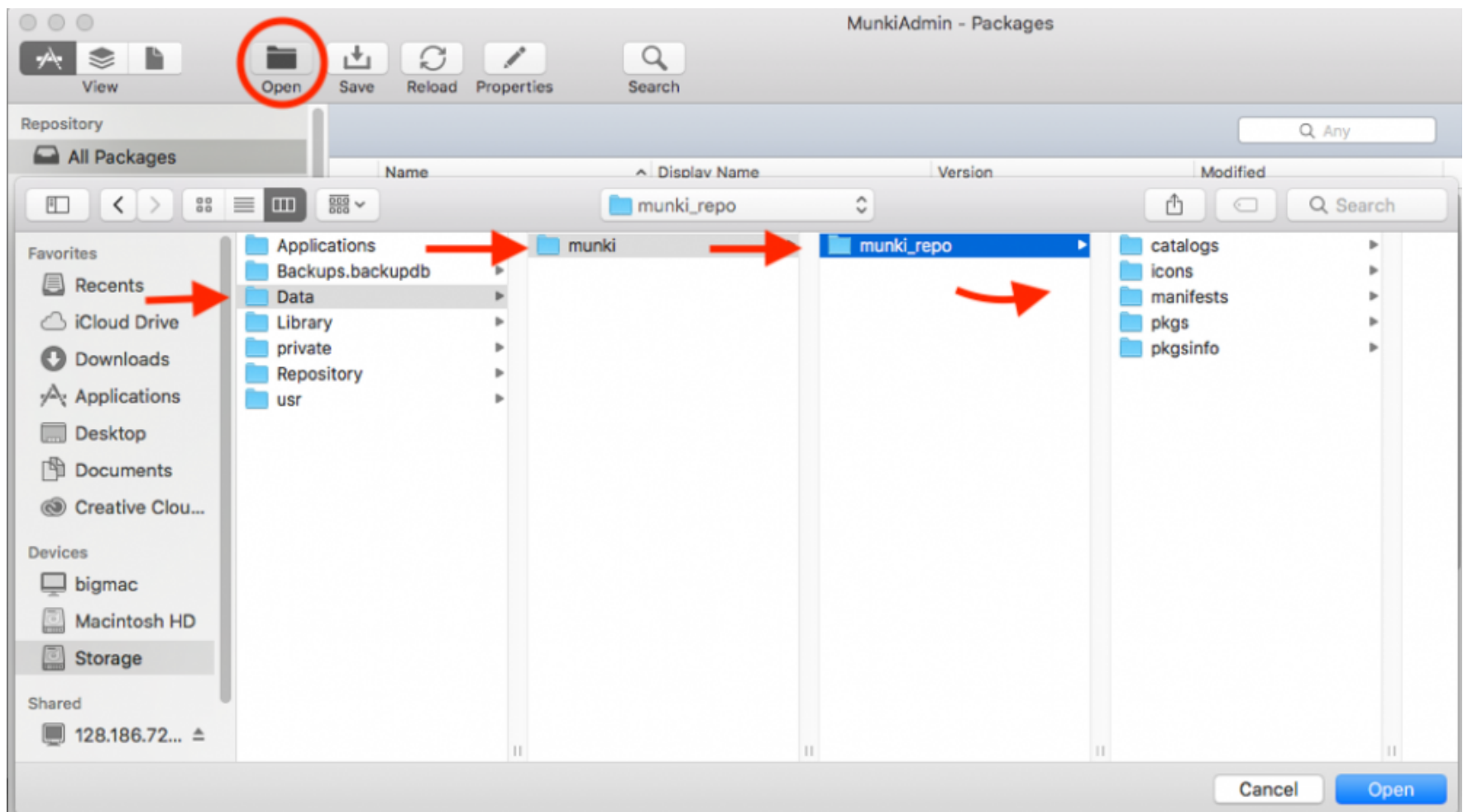

Manifest = lab

Catalogs = packages and icons.

Manifest Use a catalog. You really only need two catalogs. One for testing purposes, and the other to store all your software you'd like to use. Meaning we can make numerous Manifest (labs, rooms, or individual computers). Which all can use the same catalog.

- Through the manifest tab this is where you can choose what software is Installed straight to the computer (Managed Install).
- You can choose what you want to uninstall (Managed Uninstalls).
- You can manage what software you'd like to have auto update (Managed Updates).
- You can give an option to the client if they would like to install the software (Optional Installs).
- It is easy to add software into each table. Just click on the table you want.
- For example, click managed installs.
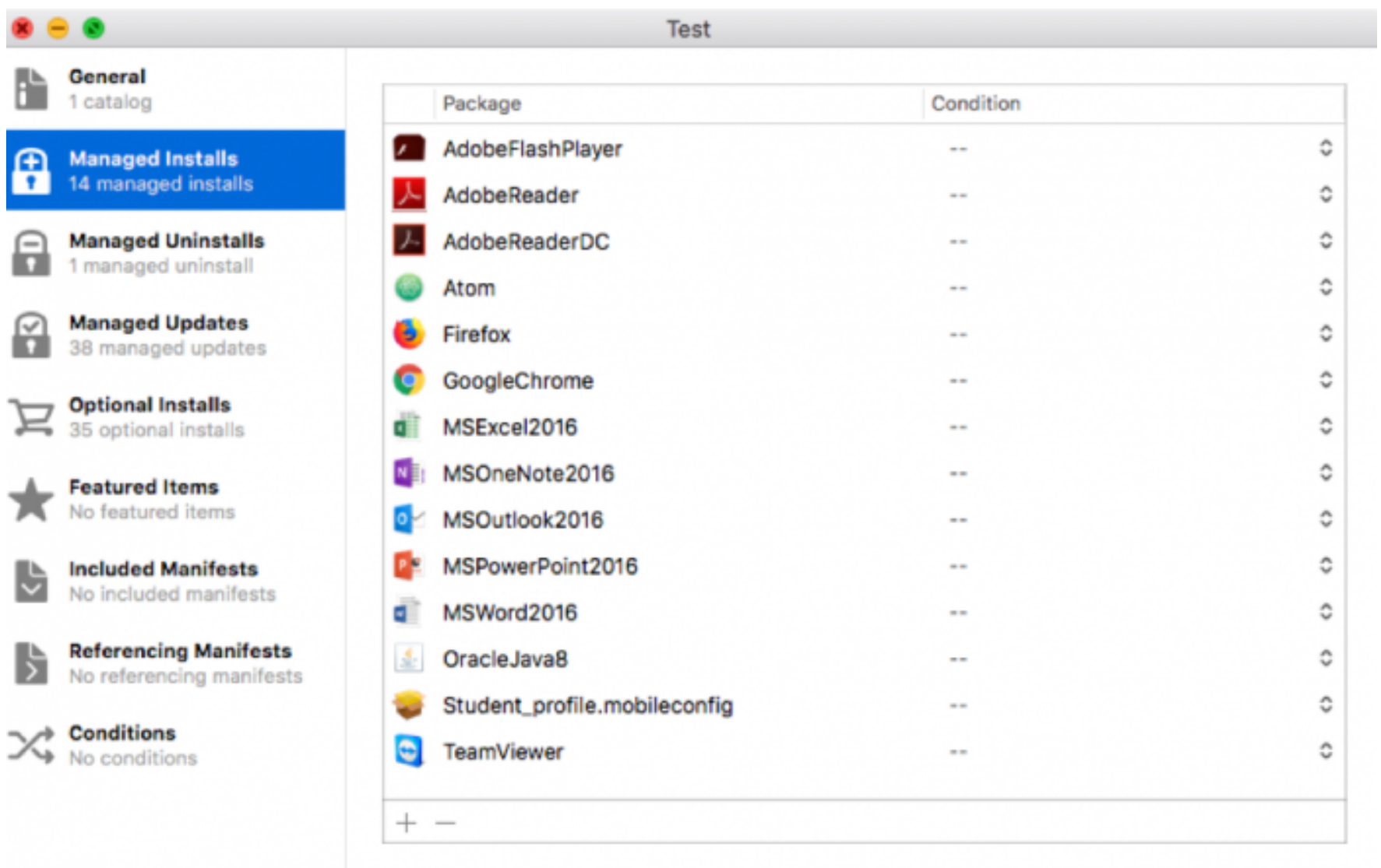
- To add software, click the Plus sign (+) to remove software click the minus sign (-).

- It works the same way for each table.
- Make sure to save regularly when you make changes. After saving hit reload.
- This can be found at the top next to the Open button.

Icons

- Everything added from Auto Pkgr will be added to the Show Packeges list.
- Here the software will not have an icon.
- Through munki admin you can easily search the package for icons.

 Then just click extract and it will

locate a image.

—-

Auto Pkgr

This is installed on the server computer. This application uses GitHub and pulls repositories from others using GitHub. You pull receipts from these repositories. Receipts can be update packages, installation files, and unistallers.

AutoPkgr

Install | **Repos & Recipes** | Schedule | Notifications | Folders & Integration

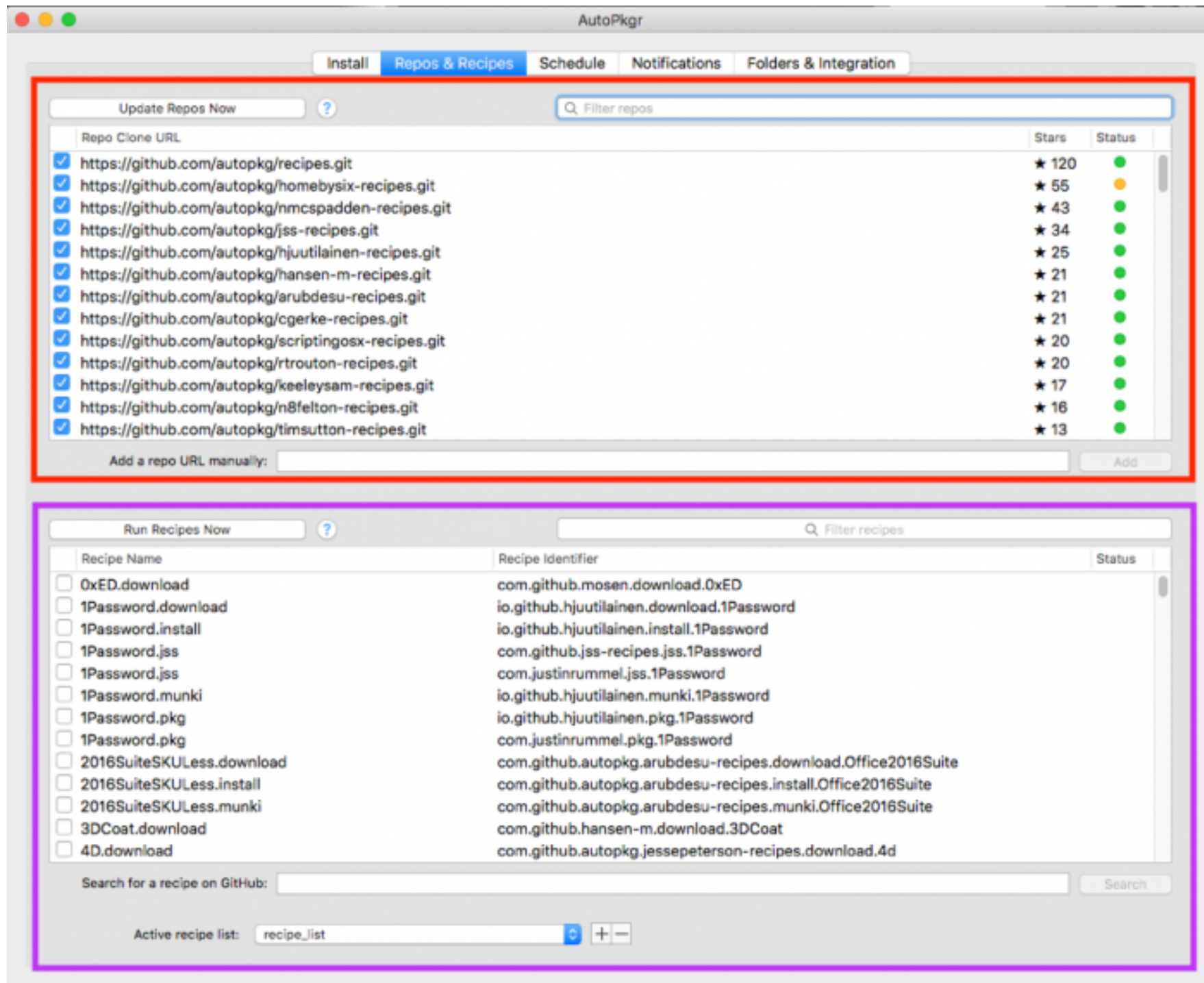| Repo Clone URL	Stars	Status
☑ https://github.com/autopkg/recipes.git	★ 120	●
☑ https://github.com/autopkg/homebysix-recipes.git	★ 55	●
☑ https://github.com/autopkg/nmcspadden-recipes.git	★ 43	●
☑ https://github.com/autopkg/jss-recipes.git	★ 34	●
☑ https://github.com/autopkg/hjuutilainen-recipes.git	★ 25	●
☑ https://github.com/autopkg/hansen-m-recipes.git	★ 21	●
☑ https://github.com/autopkg/arubdesu-recipes.git	★ 21	●
☑ https://github.com/autopkg/cgerke-recipes.git	★ 21	●
☑ https://github.com/autopkg/scriptingosx-recipes.git	★ 20	●
☑ https://github.com/autopkg/rtrouton-recipes.git	★ 20	●
☑ https://github.com/autopkg/keeleysam-recipes.git	★ 17	●
☑ https://github.com/autopkg/n8felton-recipes.git	★ 16	●
☑ https://github.com/autopkg/timsutton-recipes.git	★ 13	●

Update Repos Now   ?   Filter repos

Add a repo URL manually: Add

Run Recipes Now   ?   Filter recipes

Recipe Name	Recipe Identifier	Status
☐ 0xED.download	com.github.mosen.download.0xED	
☐ 1Password.download	io.github.hjuutilainen.download.1Password	
☐ 1Password.install	io.github.hjuutilainen.install.1Password	
☐ 1Password.jss	com.github.jss-recipes.jss.1Password	
☐ 1Password.jss	com.justinrummel.jss.1Password	
☐ 1Password.munki	io.github.hjuutilainen.munki.1Password	
☐ 1Password.pkg	io.github.hjuutilainen.pkg.1Password	
☐ 1Password.pkg	com.justinrummel.pkg.1Password	
☐ 2016SuiteSKULess.download	com.github.autopkg.arubdesu-recipes.download.Office2016Suite	
☐ 2016SuiteSKULess.install	com.github.autopkg.arubdesu-recipes.install.Office2016Suite	
☐ 2016SuiteSKULess.munki	com.github.autopkg.arubdesu-recipes.munki.Office2016Suite	
☐ 3DCoat.download	com.github.hansen-m.download.3DCoat	
☐ 4D.download	com.github.autopkg.jessepeterson-recipes.download.4d	

Search for a recipe on GitHub: Search

Active recipe list: recipe_list   + −

- The top window is your Repository. Check At least the top 45 then click update repos now.
- Next in the bottom window is all the software you have access to through your repositories.
- Search through the recipes for the software you want. You have to check the (.munki extensions). That will be the only files that work through munki.
- After you have chosen all your software click "Run Recipes Now". This can take a while.
- After that you might get a warning saying missing trust. That is ok, they still work so just ignore that prompt.
- Finally, all the software should be in your catalog. To check this, make your way to munki admin.

Server Set up

- Visit Mac OS () Server

- Make sure to turn on website services
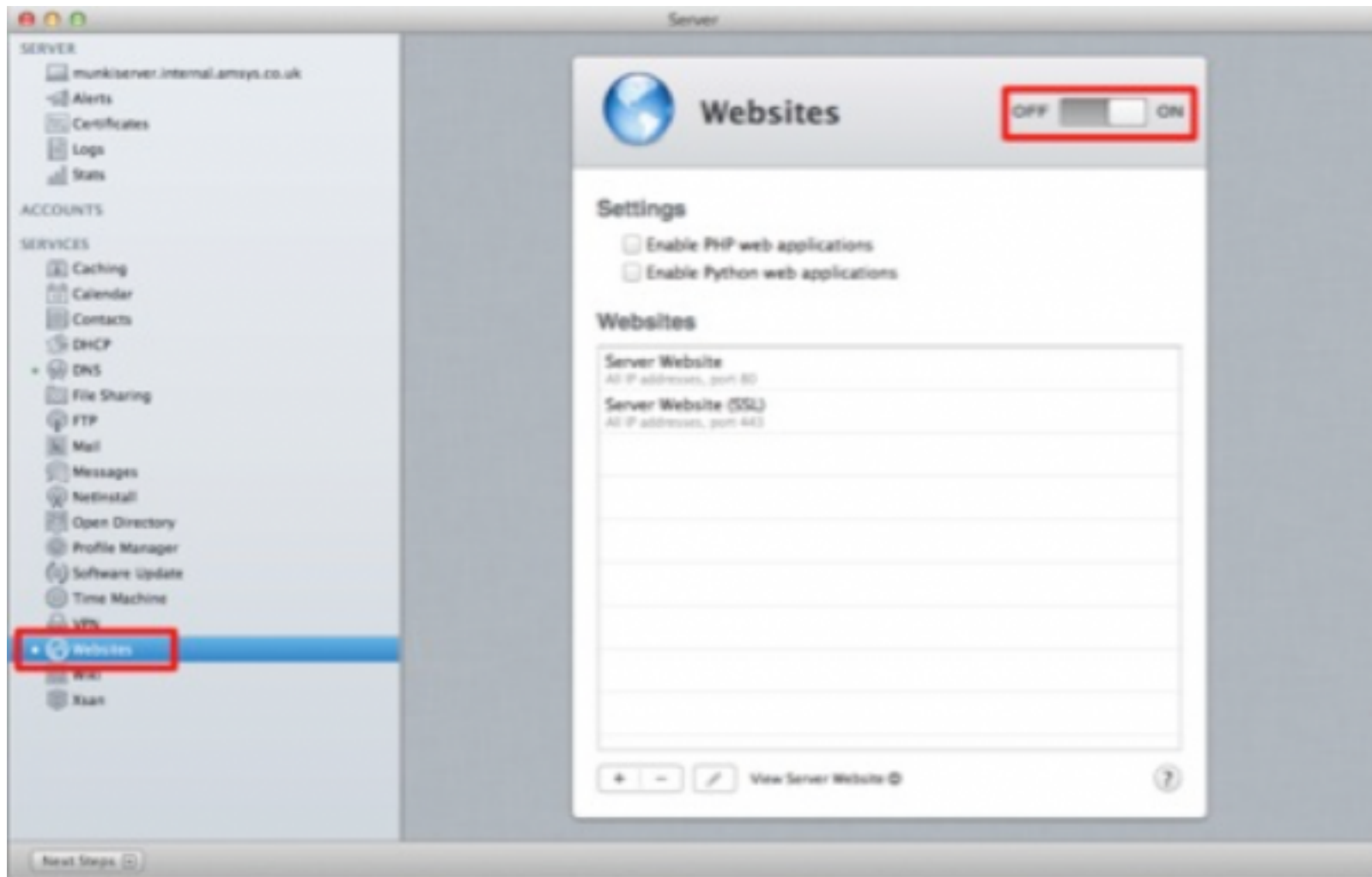
- Link the Munki repository with the web service.
- Click the arrow next to the "store site files in: Default". Make a note of where this default store is. By default, this will be "/Library/Server/Web/Data/Sites/Default/"
- Use the terminal to create a symlink called "munki_repo".
- sudo ln -s /Volumes/DataHD/munki/munki_repo [space] /Library/Server/Web/Data/Sites/Defualt/
- Close any open finder and terminal windows and switch back to the Server App.
- Click the "Edit Advanced Settings…" button.

- Select the box called "Allow Folder Listing" and click "OK". Click "Done" on the "Server
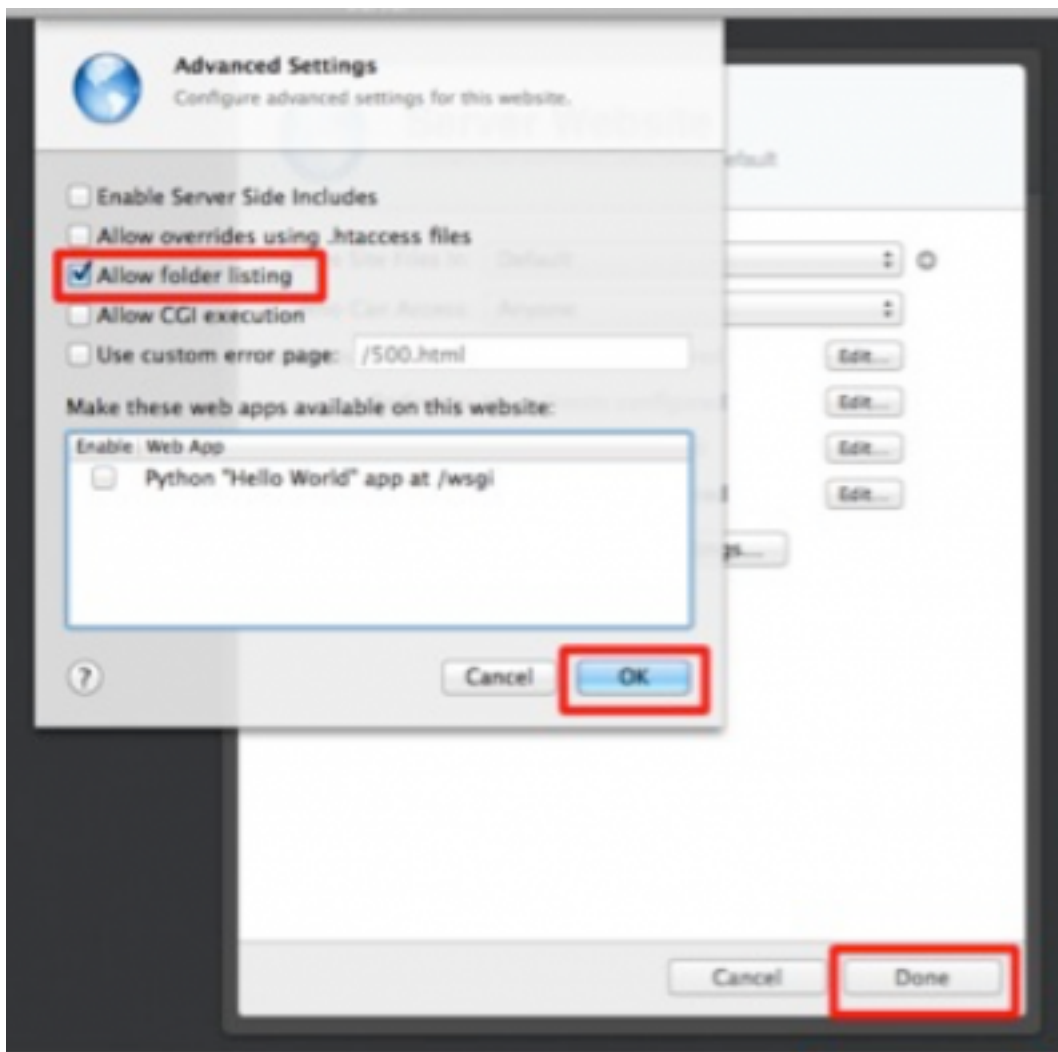
- Website" screen.
- Launch Safari and navigate to bigmac.cci.fsu.edu/munki_repo.
- You are all set up.

Setting up Clients

Open terminal and run these commands: Must download Managed Software Center on all Client Devices.

```
sudo defaults write /Library/Preferences/ManagedInstalls SoftwareRepoURL "http://bigmac.cci.fsu.edu/munki_repo"
```

```
sudo defaults write /Library/Preferences/ManagedInstalls ClientIdentifier Test
```

or "lab name"

```
sudo defaults write /Library/Preferences/ManagedInstalls InstallAppleSoftwareUpdates True
```

Munki has a a couple powerful command line tools

- makepkginfo
- munkiimport
- managedsoftwareupdate
- makecatalogs
- manifestutil

 To fully utilize these make your way to the in depth documentation at the
top. Control — F and search for command line tools. You will see links, each
link gives documentation on each command.

 Web Server: http://bigmac.cci.fsu.edu/munki_repo
(http://bigmac.cci.fsu.edu/munki_repo) —location

- JZ