GRAHAM GILBERT / MACDEVOPS:YVR / 21.06.16

Imagr

airbnb

IRUNTOTHEFUTUREANDJUMP
I'm sittin' here
completely surrounded by no beer.

Some say...

Imaging is dead

Requirements

- **USB Drive or download of work files:** http://bit.ly/imagrlab
- **OS X Installer:** Matching the version of OS X your Mac is running (10.10 Minimum)
- **VMware Fusion Professional:** A trial version is fine. Must be Professional version to NetBoot.

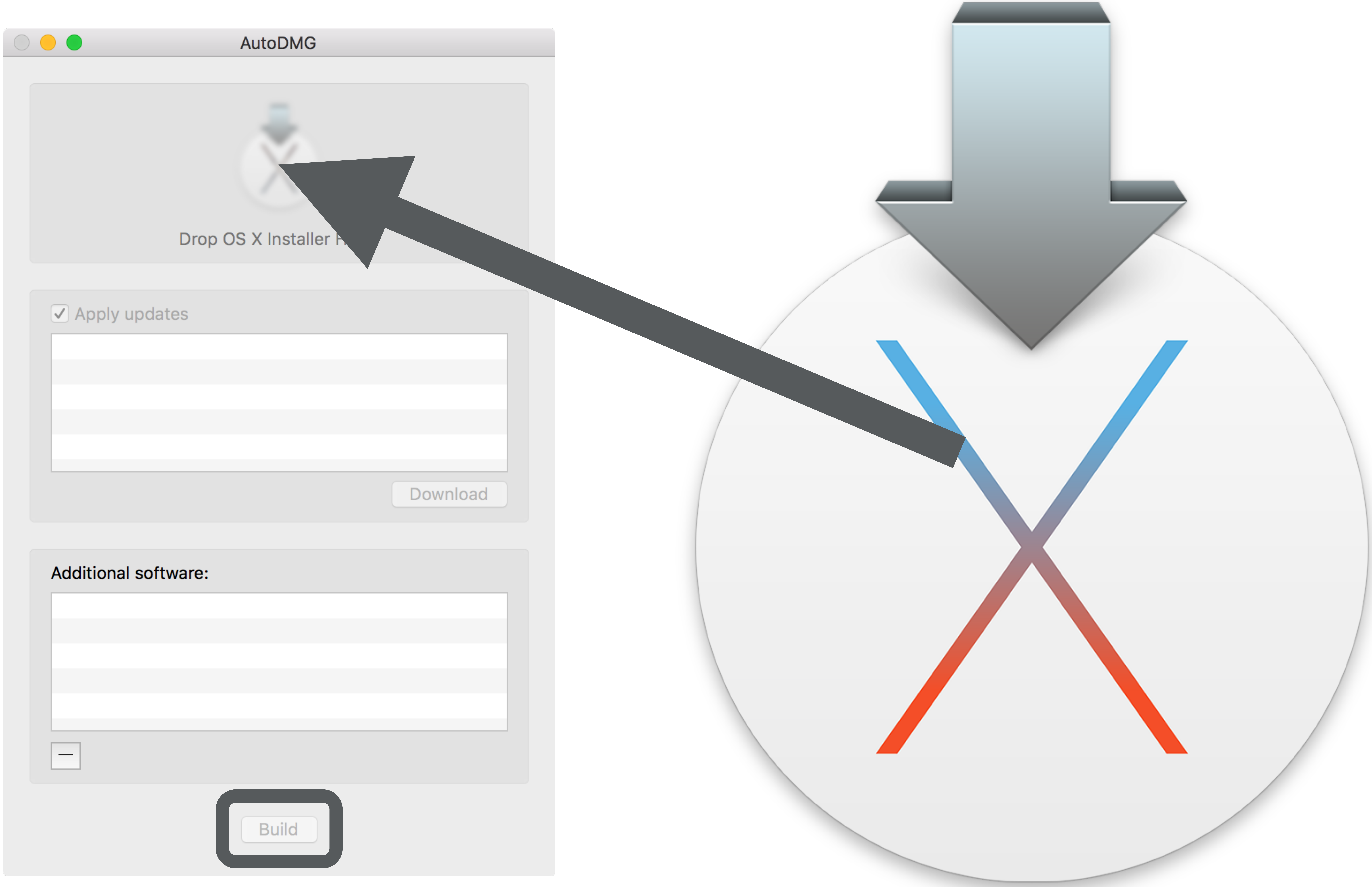
Agenda

Long Boring Prep

AutoDMG

- Creates a Never Booted Image.
- Really, really simple.
- I mean *really* simple..

Internal Hard
Handle the hard drive c
avoid damaging the circu
Make sure you are properl


AutoDMG
Drop OS X Installer H
☑ Apply updates
Download
Additional software:
—
Build

Last prep

Honest!

- Text Editor.
- Copy everything to your home directory.
- Install Docker Toolbox.

Start Docker Machine

```
$ cd somewhere/Imagr_Lab
$ Scripts/machine.sh
```

Start Docker Machine

```
$ docker-machine env vmwaredocker
$ eval "$(docker-machine env vmwaredocker)"
```

Start Docker Machine

```
$ cd somewhere/Imagr_Lab
$ Scripts/import.sh
```

```
@objc.IBAction
[illegible]
    self.targetVolume = self.chooseTargetDropDown.titleOfSelectedItem()
    for volume in self.volumes:
        if str(volume.mountpoint) == str(self.targetVolume):
            self.workVolume = volume
            break
    self.enableAllButtons_(sender)
    NSApp.endSheet_(self.chooseTargetPanel)
    self.chooseTargetPanel.orderOut_(self)
    self.selectWorkflow_(self)


@objc.IBAction
def closeImagingTarget_(self, sender):
    self.enableAllButtons_(sender)
    NSApp.endSheet_(self.chooseTargetPanel)
    self.chooseTargetPanel.orderOut_(self)
    self.setStartupDisk_(sender)

@objc.IBAction
def selectWorkflow_(self, sender):
    self.chooseWorkflowDropDown.removeAllItems()
    list = []
    for workflow in self.workflows:
        list.append(workflow['name'])

    self.chooseWorkflowDropDown.addItemsWithTitles_(list)
    self.chooseWorkflowLabel.setHidden_(False)
```

Choose Target: /Volumes/Macintosh HD 1

Choose Workflow: Base Workflow

What is Imagr?

Imagr is an open source deployment tool for Macs

Uses a simple web server

No OS X Server required

No OS X “Server” required

Open Source

Can replace DeployStudio for most people

Growing community

Configured with a plist

Scripts/ genpassword.sh

```
#!/bin/bash

python -c 'import hashlib; print
hashlib.sha512("YOURPASSWORDHERE")
.hexdigest()'
```

Generating the password

```
$ cd somewhere/Imagr_Lab
$ Scripts/genpassword.sh
```

Data/web_root/imagr_config.plist

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/
DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>password</key>

<string>b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b
1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86</string>
  <key>workflows</key>
  <array>
    <dict>
      <key>name</key>
      <string>Imagr Workflow</string>
      <key>description</key>
      <string>A demo workflow made in the Imagr Lab session.</string>
      <key>components</key>
      <array>

      </array>
    </dict>
  </array>
</dict>
</plist>
```

imagr_config.plist

Workflows

Components

imagr_config.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>password</key>
 <string>b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e59
76ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86</string>
  <key>workflows</key>
  <array>
    <dict>
      <key>name</key>
      <string>Imagr Workflow</string>
      <key>description</key>
      <string>A demo workflow made in the Imagr Lab session.</string>
      <key>components</key>
      <array>

      </array>
    </dict>
  </array>
</dict>
</plist>
```

imagr_config.plist

```
<dict>
  <key>type</key>
  <string>image</string>
  <key>url</key>
  <string>http://YOURIPADDRESS/your-autodmg.dmg</string>
</dict>
```

imagr_config.plist

```
<dict>
  <key>type</key>
  <string>package</string>
  <key>url</key>
  <string>http://YOURIPADDRESS/packages/clearReg.pkg</string>
  <key>first_boot</key>
  <false/>
</dict>
<dict>
  <key>type</key>
  <string>package</string>
  <key>url</key>
  <string>http://YOURIPADDRESS/packages/adminUser.pkg</string>
</dict>
```

imagr_config.plist

```
<dict>
  <key>type</key>
  <string>computer_name</string>
</dict>
```

imagr_config.plist

```
<dict>
  <key>type</key>
  <string>script</string>
  <key>content</key>
  <string>#!/bin/bash
/usr/bin/csrutil netboot add 10.10.10.10
/usr/bin/csrutil netboot add 10.10.10.11
</string>
    <key>first_boot</key>
    <false/>
</dict>
<dict>
    <key>type</key>
    <string>script</string>
    <key>content</key>
    <string>#!/bin/bash
/usr/sbin/systemsetup -f -setremotelogin on</string>
</dict>
```

That’s it!

Validating the plist

- **Apps/Imagr.dmg**
- **/Volumes/Imagr/Tools:** Copy to your internal drive

Validating the plist

```
$ cd /SOMEWHERE/Tools
$ ./validateplist http://
YOURIPADDRESS/imagr_config.plist
```

Has AutoDMG finished?

The NBI

Tools/config.mk

```
URL=http://YOURIPADDRESS/imagr_config.plist
DMGPATH=/SOMEWHERE/Imagr_Lab/Apps/Imagr.dmg
OUTPUT=/SOMEWHERE/Imagr_Lab/Data/web_root/Imagr.nbi
VALIDATE=false
```

Build your NBI

```
$ make update
```

Copy AutoDMG image to Data/web_root

Docker

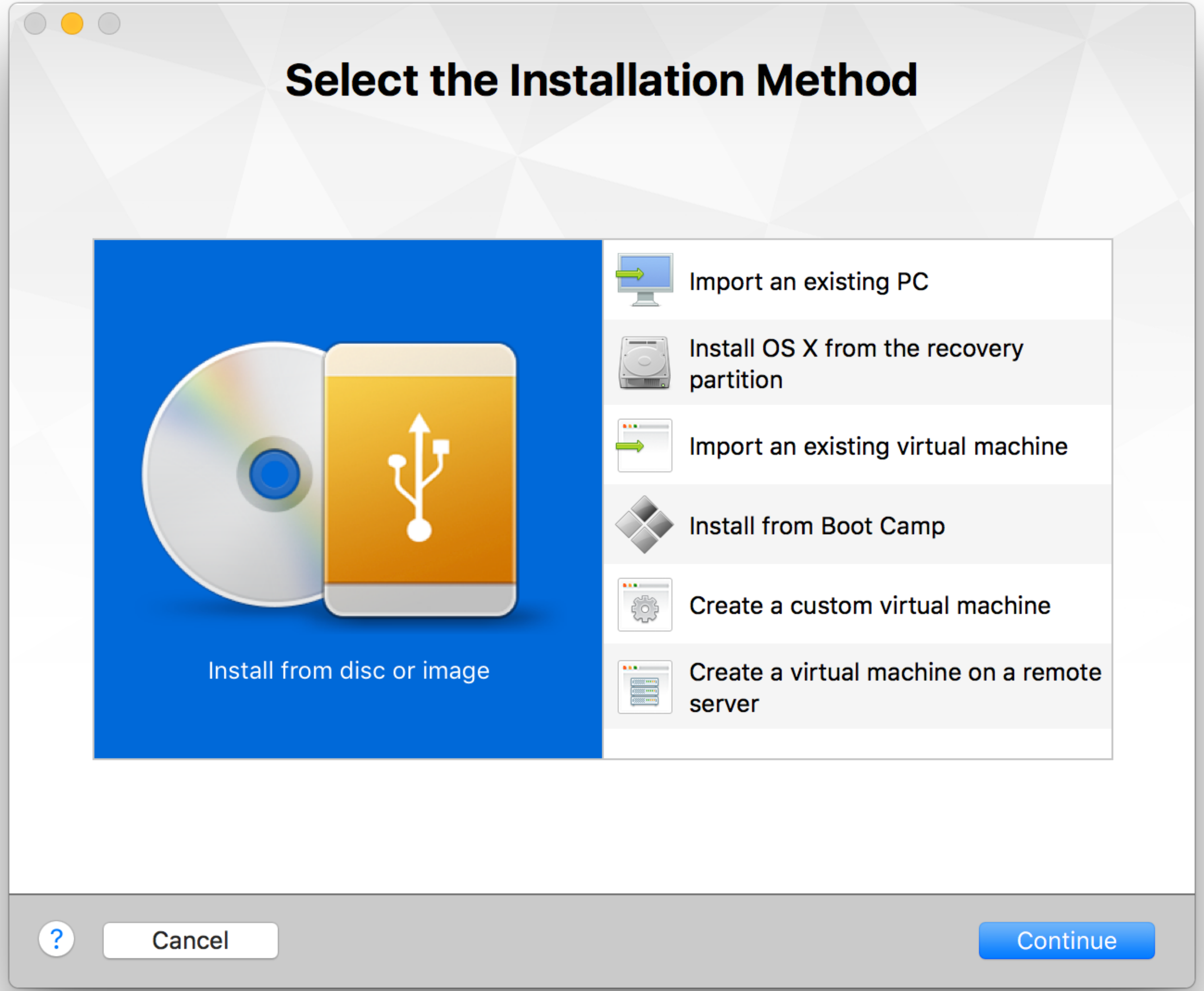
Start your containers

```
$ cd somewhere/Imagr_Lab
$ Scripts/start-docker.sh
```

Start your containers

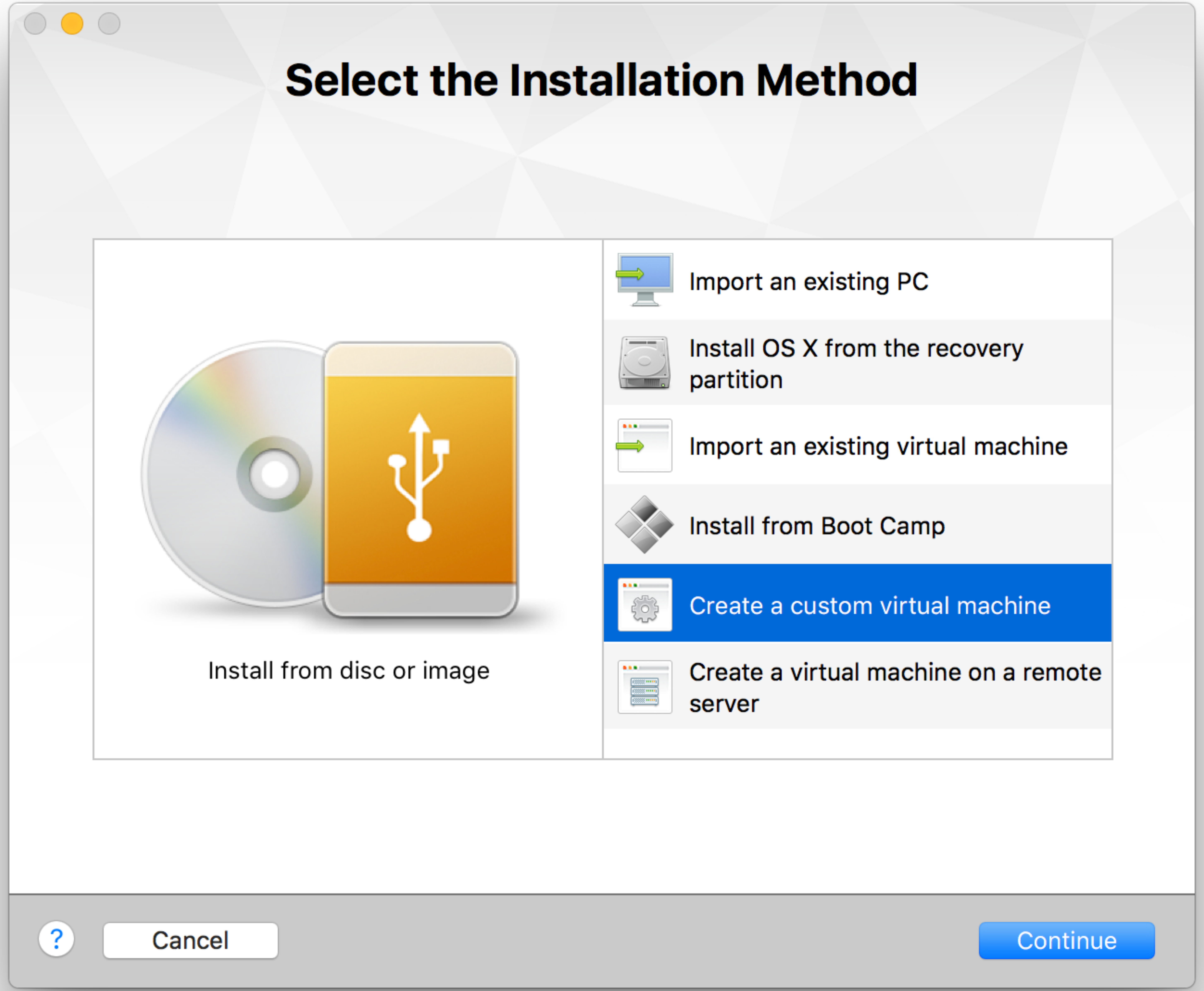
```
$ docker logs bsdpy
```

Time to NetBoot



Select the Installation Method

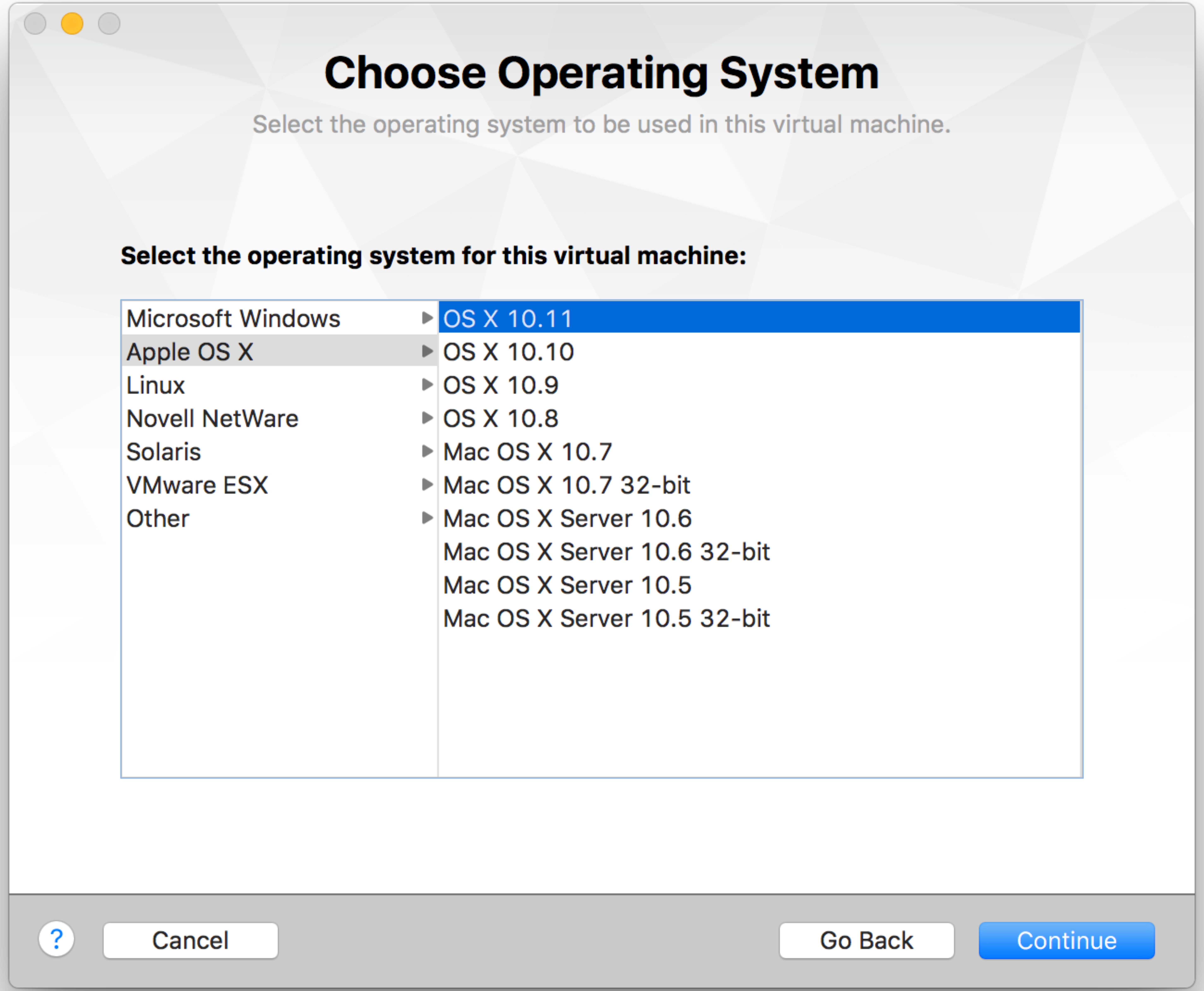
Install from disc or image
Import an existing PC
Install OS X from the recovery partition
Import an existing virtual machine
Install from Boot Camp
Create a custom virtual machine
Create a virtual machine on a remote server
?
Cancel
Continue



Select the Installation Method

Install from disc or image

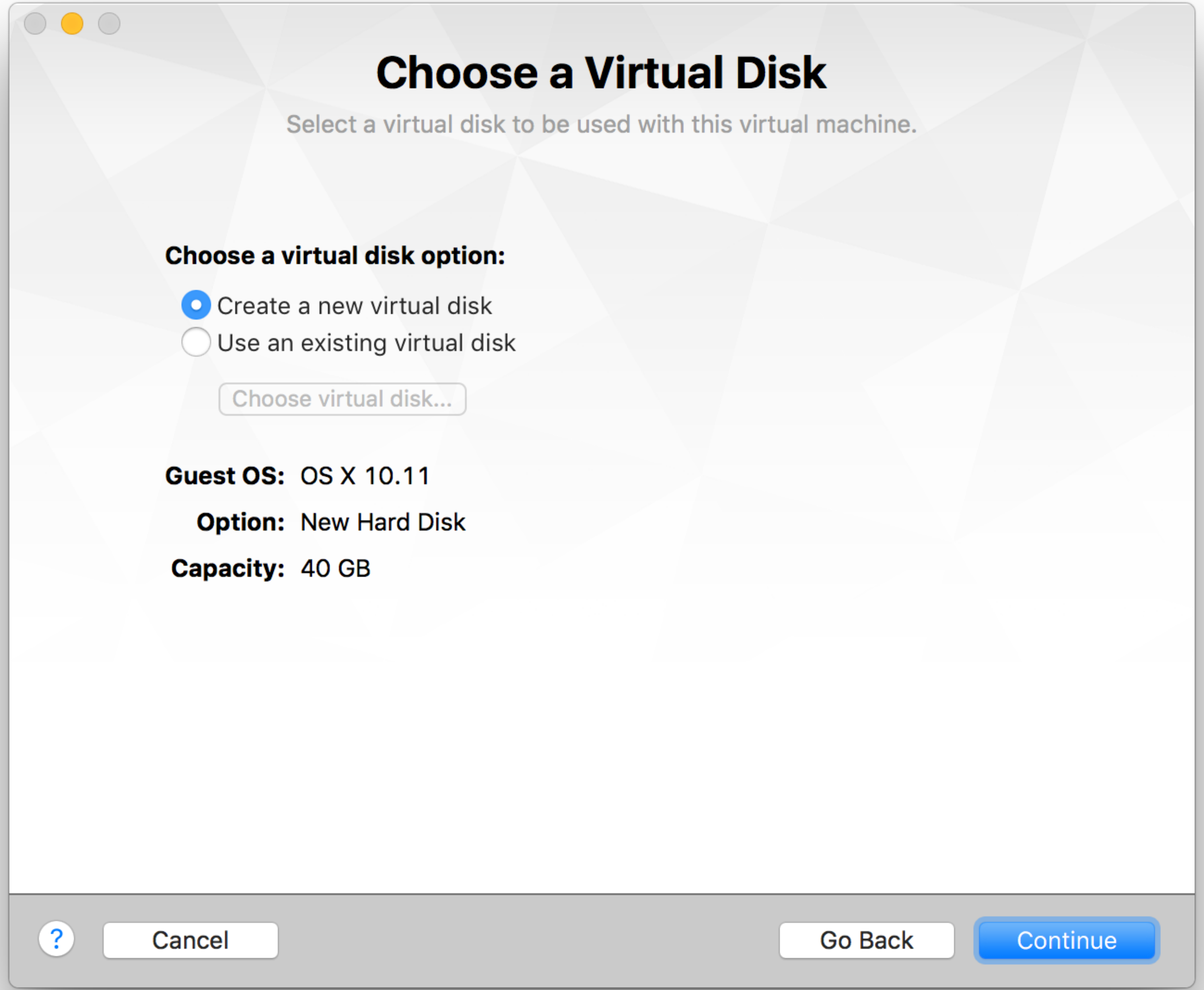
- Import an existing PC
- Install OS X from the recovery partition
- Import an existing virtual machine
- Install from Boot Camp
- Create a custom virtual machine
- Create a virtual machine on a remote server



Choose Operating System

Select the operating system to be used in this virtual machine.

Select the operating system for this virtual machine:



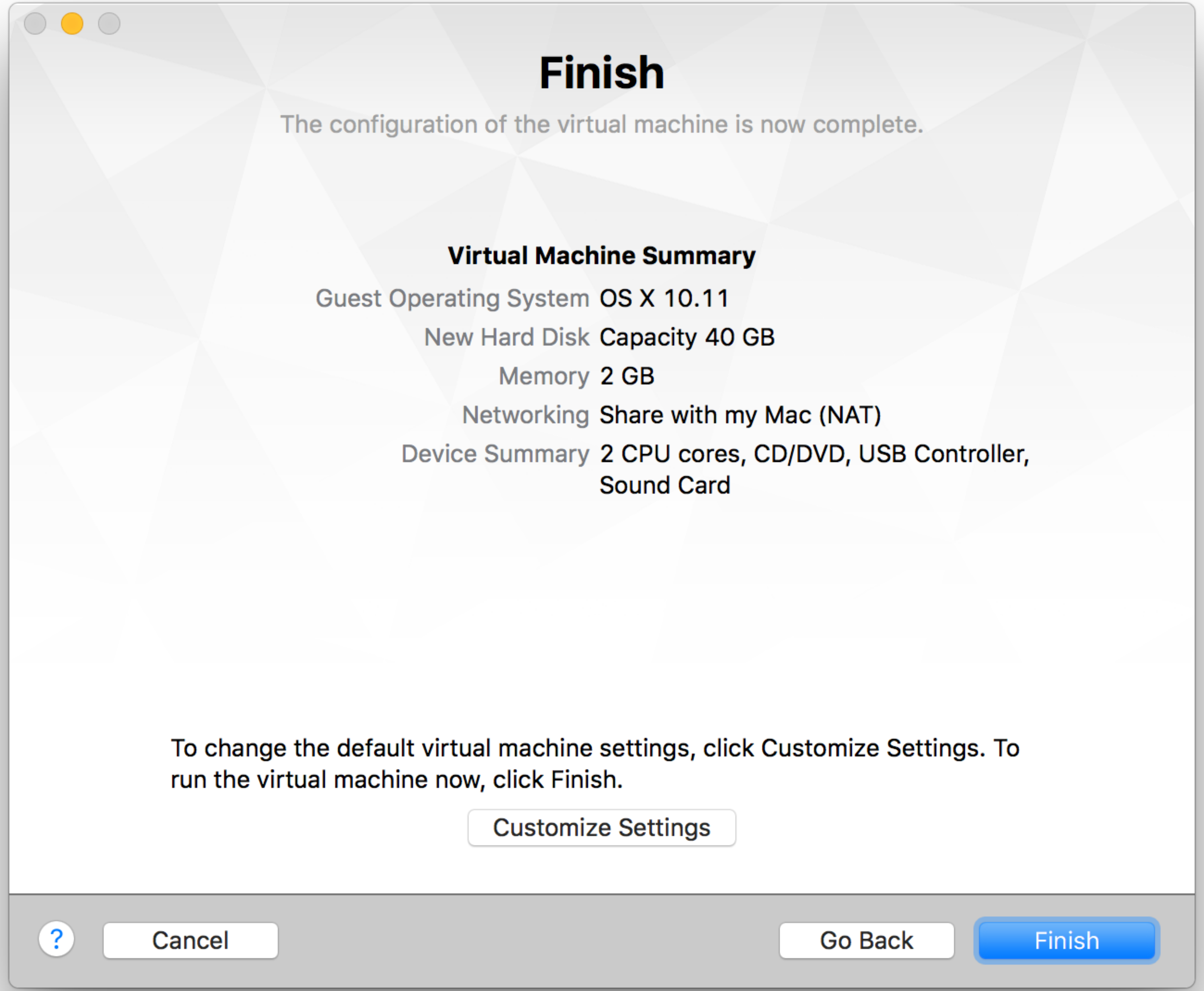
Choose a Virtual Disk

Select a virtual disk to be used with this virtual machine.

Choose a virtual disk option:

● Create a new virtual disk
○ Use an existing virtual disk

Guest OS: OS X 10.11
Option: New Hard Disk
Capacity: 40 GB



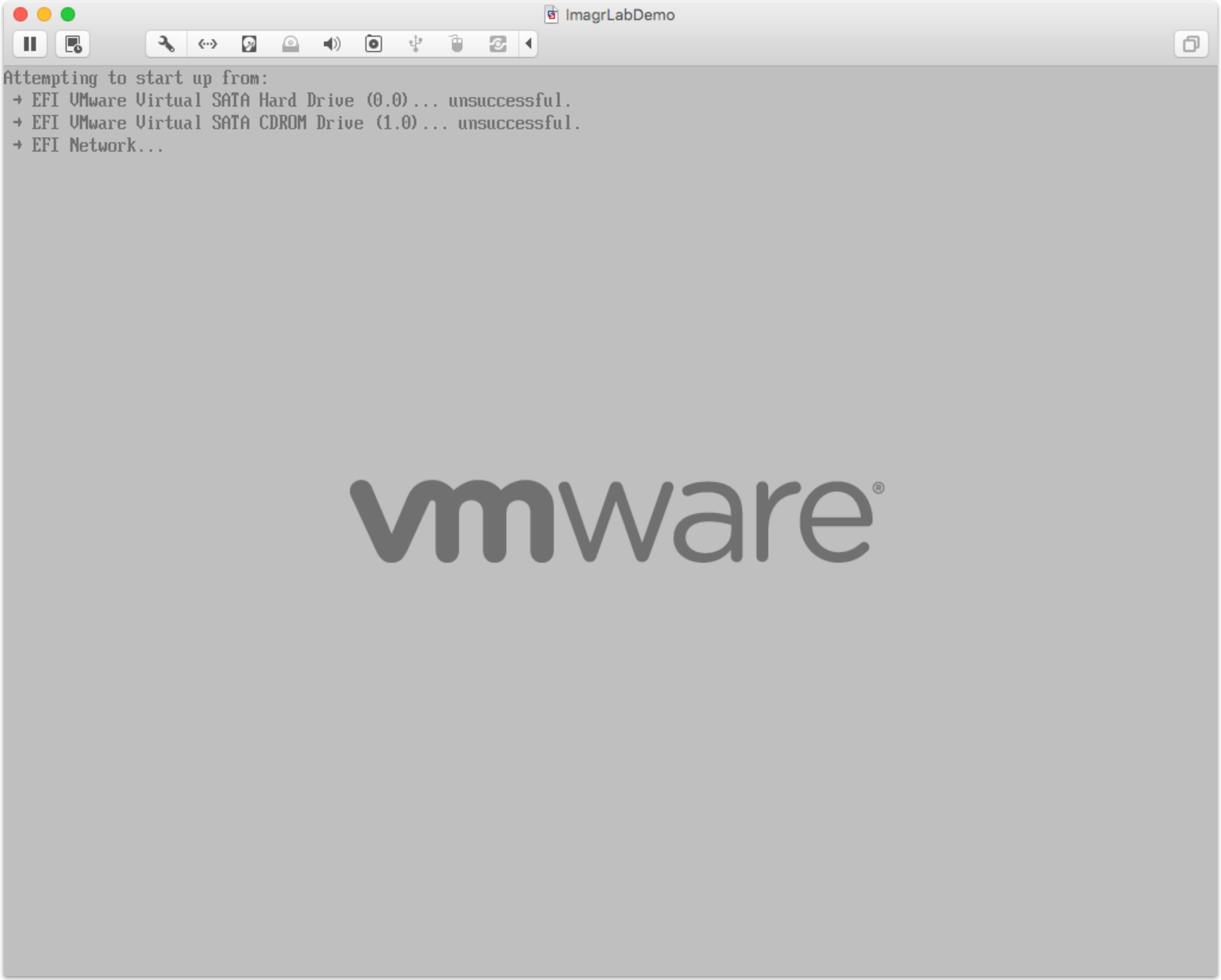
Finish

The configuration of the virtual machine is now complete.

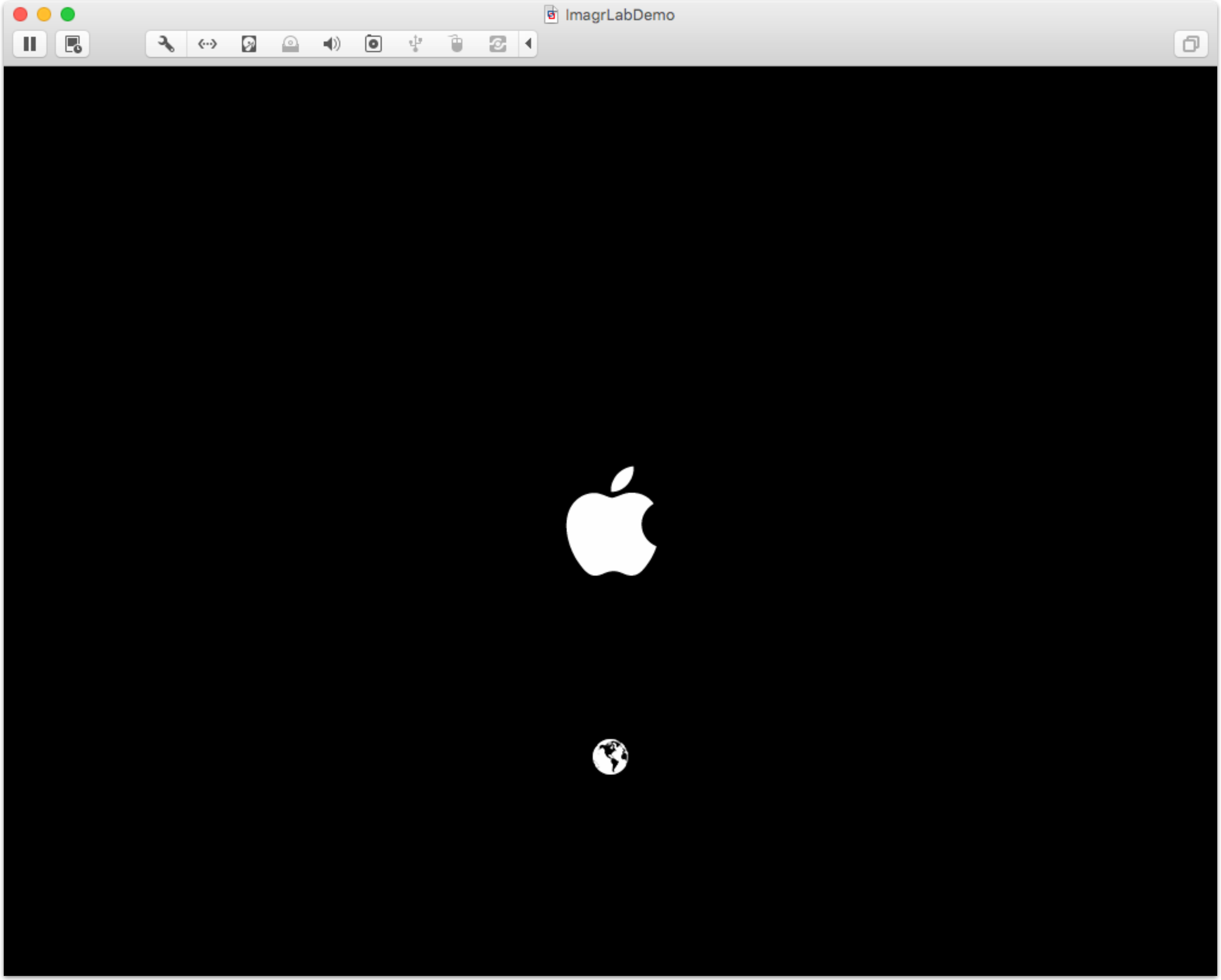
Virtual Machine Summary

Guest Operating System	OS X 10.11
New Hard Disk	Capacity 40 GB
Memory	2 GB
Networking	Share with my Mac (NAT)
Device Summary	2 CPU cores, CD/DVD, USB Controller, Sound Card

To change the default virtual machine settings, click Customize Settings. To run the virtual machine now, click Finish.

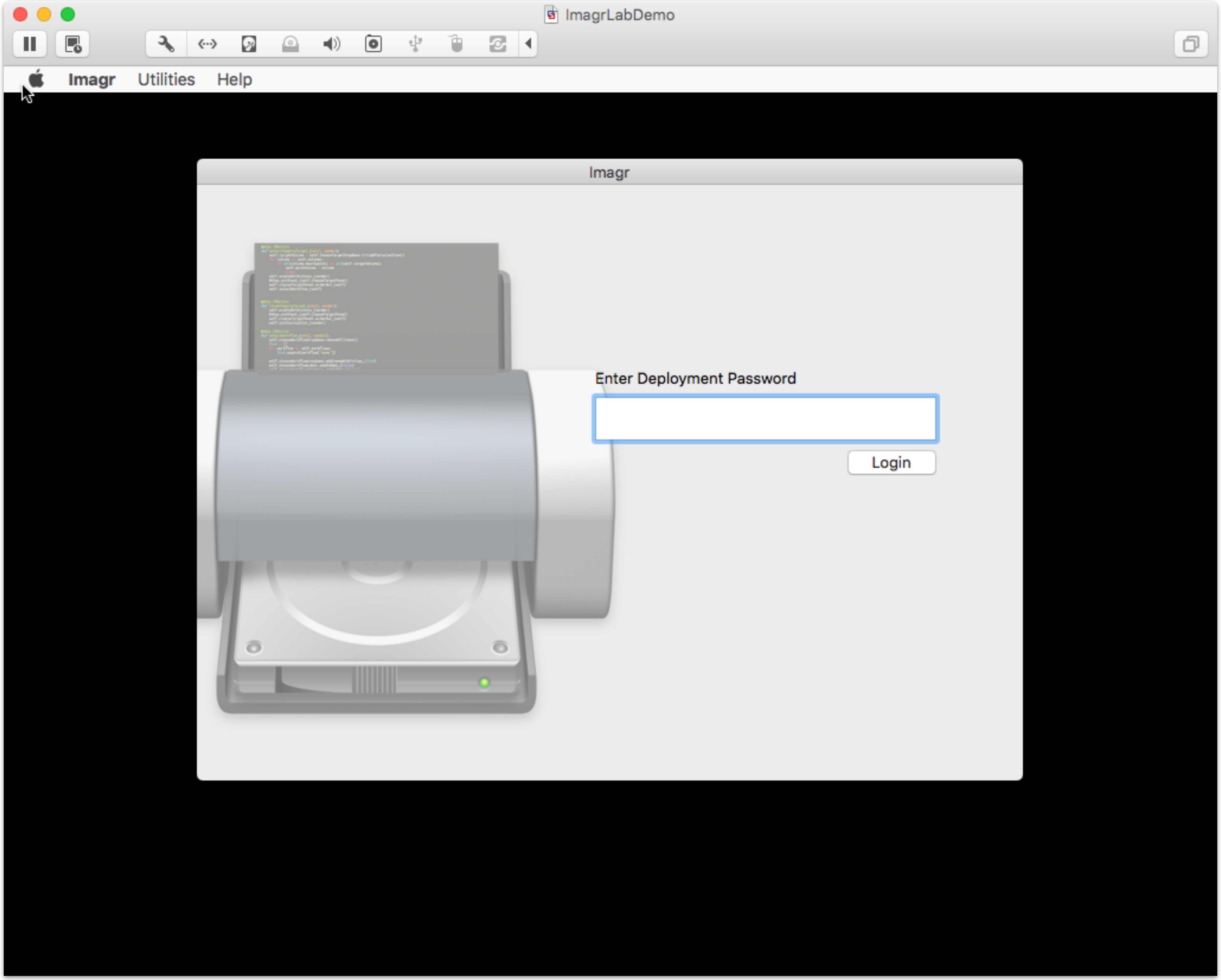


```
Attempting to start up from:
 → EFI VMware Virtual SATA Hard Drive (0.0)... unsuccessful.
 → EFI VMware Virtual SATA CDROM Drive (1.0)... unsuccessful.
 → EFI Network...
```


ImagrLabDemo

What’s happening?

```
$ docker logs bsdpy
```


ImagrLabDemo
Imagr
Utilities
Help
Imagr
Enter Deployment Password
Login

Questions?

Say hi!

- **Twitter:** @grahamgilbert
- **Github**: @grahamgilbert
- **Web:** grahamgilbert.com
- **Imagr:** github.com/grahamgilbert/imagr

airbnb