



CHASE THOMPSON-BAUGH

IMAGING IN 50+ LOCATIONS

IMAGING



NETBOOT



TOOLS



ALL THESE THINGS ARE DEAD





DEP



MDM + VPP

8,500+ COMPUTERS

ZERO

WHAT ABOUT HIGH SIERRA?

ASR + APFS = 

STEP 1: IDENTIFY THE PROBLEM



STEP 2: FIX IT



SYNC NBI FILES TO SERVERS

WHAT NOT TO DO



SYNC NBI FILES TO SERVERS

WHAT NOT TO DO

```

Synchronize () {

    # Get the directory path of running script, added '2' to not
confuse the script... potentially.
DIR2="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
# Location of directory containing .NBI files
syncFolder="$DIR2/StoreNBIs/"
# Find OS version
OSXVersion=$(sw_vers -buildVersion)
# Administrator password for `sudo` commands
adminPass="PASSWORD"

start=$(date +%s)
echo -e
"\n#####
##### Syncing to $1 #####
####"
echo -e "===== $(date) ====="
echo -e "===== Syncing to $1 =====\n"

# Actual `rsync` command, designed with rsync v3.1.2 in mind
/usr/local/bin/rsync \
    -aHAXxv \
    -e 'ssh -i /Users/localadmin/.ssh/id_rsa -T -o Compression=no
-x' \
    --exclude '.DS_Store' \
    --rsync-path="/usr/local/bin/rsync" \
    "$syncFolder" \
    admin@$1

# Determine location of the `serveradmin` binary
if [[ "$OSXVersion" < "12A" ]]; then
    ServerAdminPath="$4/usr/sbin/serveradmin"
else
    ServerAdminPath="$4/Applications/Server.app/Contents/
ServerRoot/usr/sbin/serveradmin"
fi
if [ ! -e "$ServerAdminPath" ]; then
    echo "ERROR: Unable to find 'serveradmin' tool. Is Server.app in
the Applications folder?"
fi

# Cycle the NetBoot Service
ssh -i /Users/localadmin/.ssh/id_rsa admin@$1 "echo $(echo $adminPass
| base64 -D) | sudo -S $ServerAdminPath stop netboot"
ssh -i /Users/localadmin/.ssh/id_rsa admin@$1 "echo $(echo $adminPass
| base64 -D) | sudo -S $ServerAdminPath start netboot"

end=$(date +%s)
# Compare $start and $end to get total time in minutes and seconds
runtime=$(python -c "print '%u:%02u' % ((${end} - ${start})/60, (${end} - ${start})%60)")
echo -e "\n===== Sync Completed ====="
echo -e "===== Runtime was $runtime ====="
echo -e "#####
##### Syncing to $1 #####
####"
}

# Export the above function so `parallel` can use it.
export -f Synchronize

# Simultaneously sync to as many servers as there are cores in this server
/usr/local/bin/parallel Synchronize ::: "${Servers[@]}" >> "$DIR/
rsync.log"

```



Re: Managing NetBoots on 70+ Servers

★ Chase Thompson-... ▾ Morgan is right.

12/1/16, 10:01 AM

★ Chase Thompson-... ▾ I guess I have to do some research on Imagr today.

11/30/16, 9:14 AM

★ Chase Thompson-... ▾ Kostas, Glad to hear someone is in the same boat, if only on a smaller scale. 11/29/16, 8:52 PM

★ Chase Thompson-... ▾ We use Jamf Pro and I do deploy AST 2 using that method since Apple pro... 11/29/16, 8:46 PM

★ Chase Thompson-... ▾ To: MacEnterprise

11/29/16, 4:45 PM

I'm looking for advice on managing/standardizing the NetBoot/NetInstall/NetRestore images on 70+ macOS servers. My company is an Apple authorized service provider and we have to use macOS servers for use with Apple diagnostic tools. This is also where we keep all of our netboot sets to update/restore/diagnose customer computers. Keeping the images up-to-date on all those servers is a challenge. I currently have site-to-site VPN to all stores (Thanks Meraki) so I have one master computer that uses a custom rsync script to push images to the store servers in parallel (i.e. 12 core computer syncs to 12 stores at a time). The issue with that is that it takes a long time to sync to all servers and many time out, despite having a fast fiber connection on the master end. The script runs twice a week via a cron job so we can always try again. I also keep a log file with the outcome of each server sync. This works well enough and if a server is being particularly stubborn with the script, we keep copies of all netboot sets in an S3 bucket so we could just download them manually.

I've thought about setting up a DeployStudio Master on my master computer and setting up all the store servers as replicas to be a more streamlined experience. I have experience with DeployStudio but not at this scale. I imagine the sync would still be slow but more managed at least. DeployStudio would replace the individual NetInstall and NetRestore images in favor of workflows which would significantly reduce the number of available boot options at the boot picker. A solution for syncing NetBoots would still have to be worked out though as we still need an environment to run things like DiskWarrior.

I've also thought about using the S3 bucket as a source of truth and having a local script on the servers compare their current neboot sets and download the required changes using authenticated URLs. This would address the issue of speed as I could use AWS CloudFront to speed up downloads. It would also serve as the solution for NetBoot/Install/Restores. I've never used S3 in this way so more research would be needed.

Has anyone else managed a high number of netboot servers that could give me some pointers?

All Devices

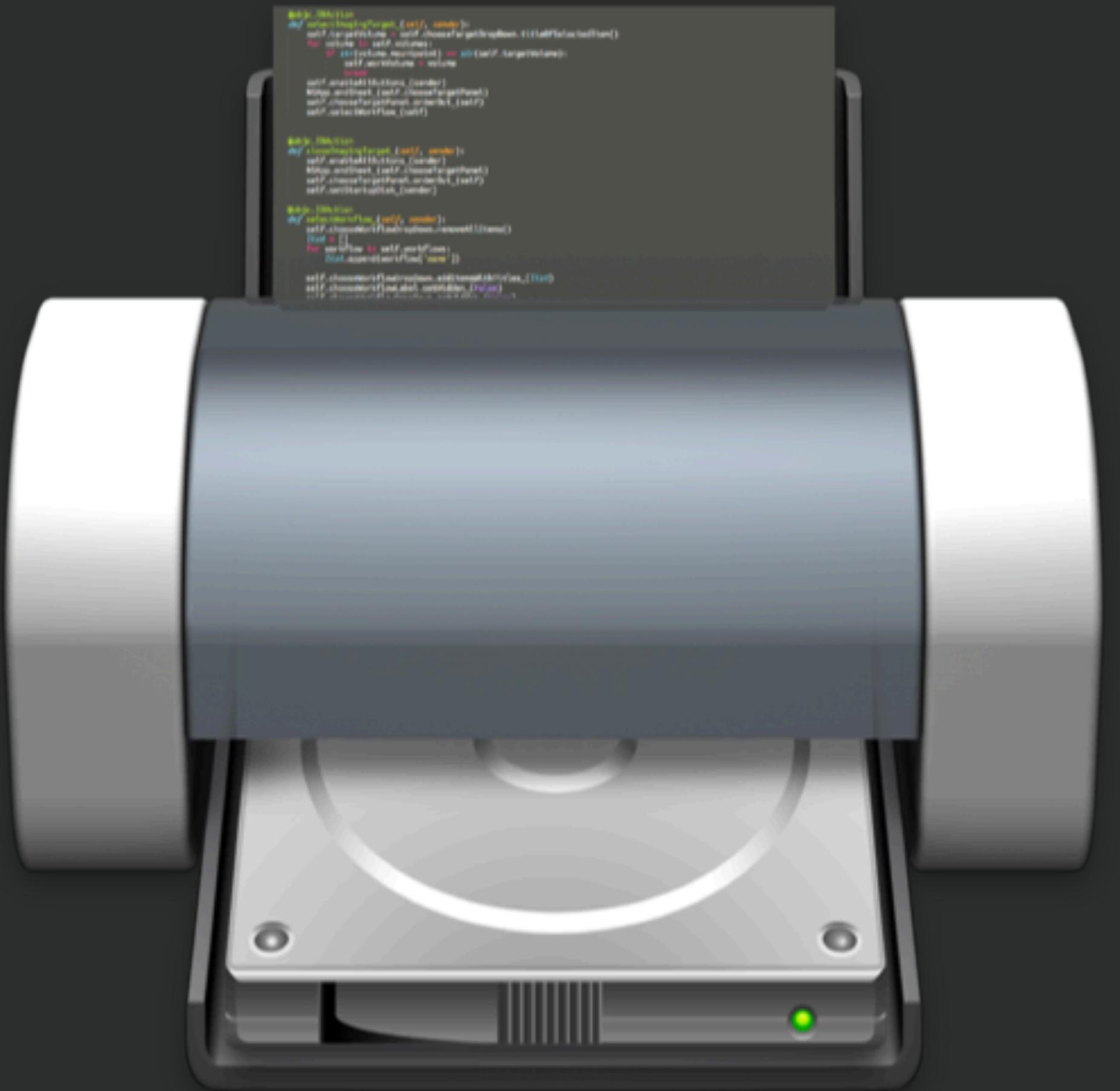
Back View Add Blueprints Prepare Update Back Up Tag Help

All Devices Supervised Unsupervised Recovery

Name	Product Version	Model	Capacity	Is Supervised	Organization Name	Up
Customer MacBook		MacBookPro13,1				

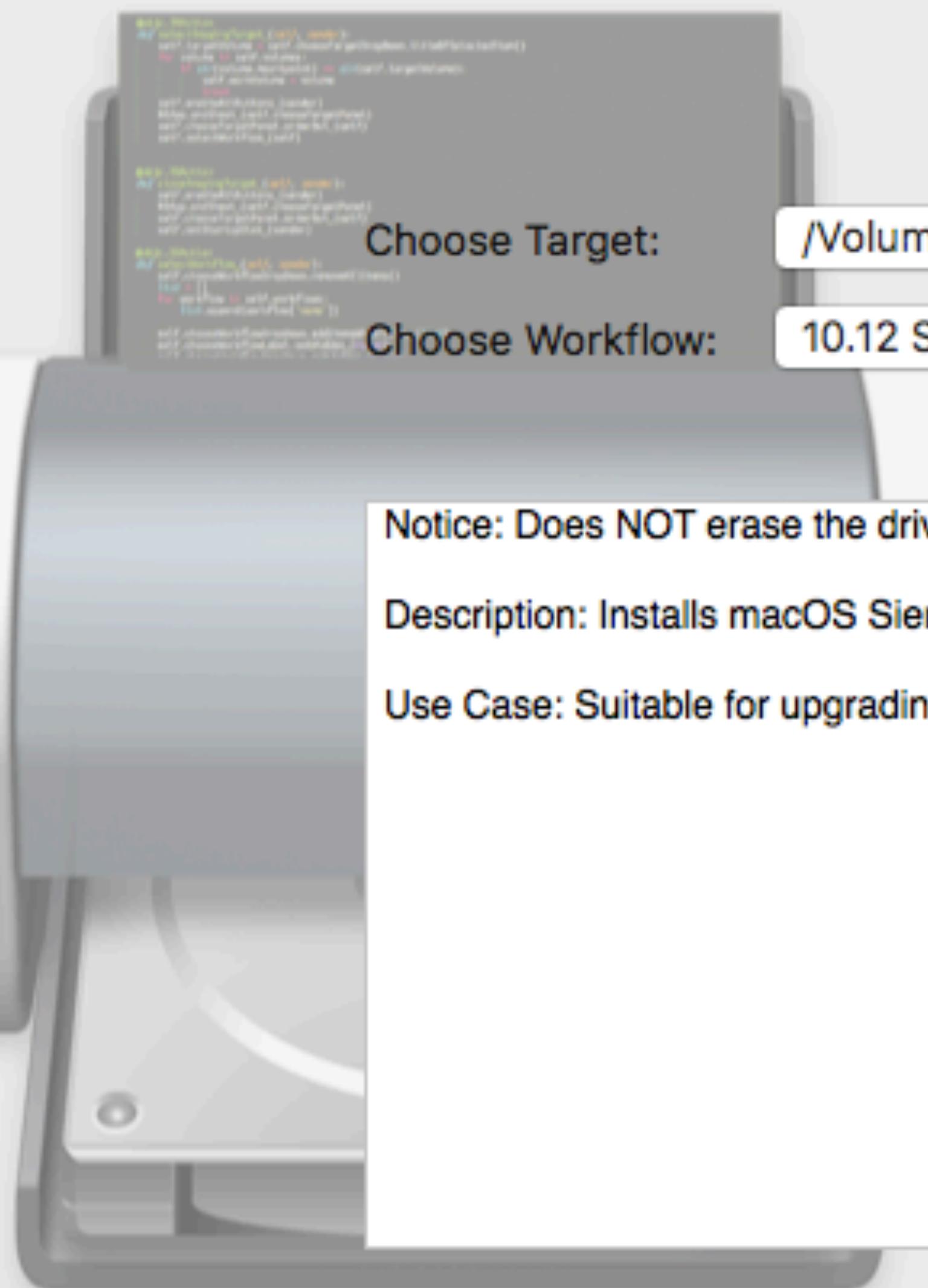
⚠ Customer MacBook

- Add
- Remove
- Modify
- Export
- Restore
- Update...
- Prepare...
- Restore from Backup...
- Apply ▶**
 - 10.7 Lion Restore
 - 10.7 Lion Upgrade
 - 10.8 Mountain Lion Restore
 - 10.8 Mountain Lion Upgrade
 - 10.9 Mavericks Restore
 - 10.9 Mavericks Upgrade
 - 10.10 Yosemite Restore
 - 10.10 Yosemite Upgrade
 - 10.11 El Capitan Restore
 - 10.11 El Capitan Upgrade
 - 10.12 Sierra Restore
 - 10.12 Sierra Upgrade
- Back Up
- Advanced ▶
- Get Info
- Quick Look
- Tags...



WHAT IS IMGR?

**IMAGR IS AN OPEN SOURCE
DEPLOYMENT AND IMAGING TOOL**

Choose Target:

Choose Workflow:

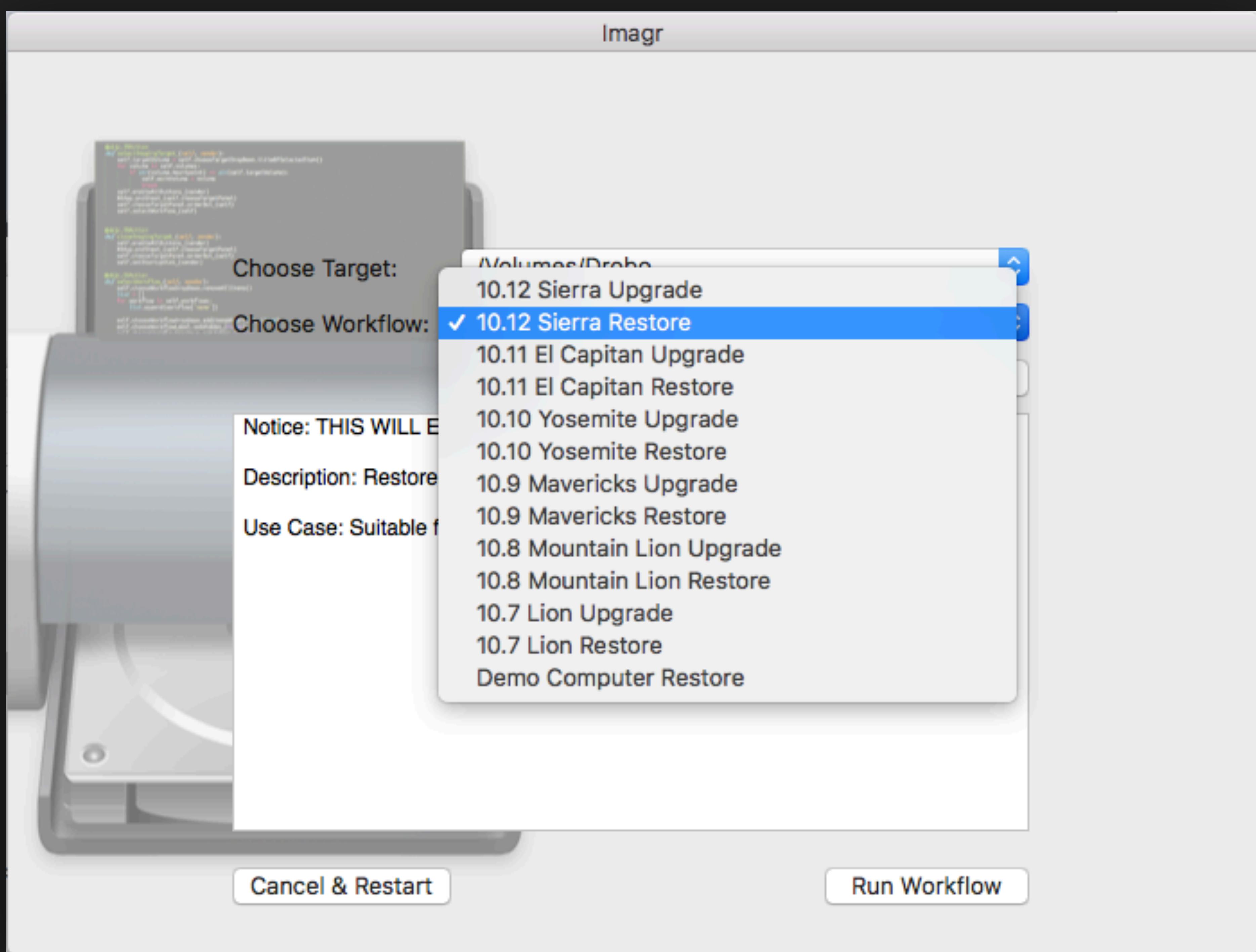
Notice: Does NOT erase the drive first. Cannot be run from "10.12 Boot" NBI.

Description: Installs macOS Sierra 10.12.2 16C68 and restarts the computer.

Use Case: Suitable for upgrading the OS without erasing customer data.

SAMPLE IMAGR WORKFLOW

```
<dict>
  <key>bless_target</key>
  <true/>
  <key>components</key>
  <array>
    <dict>
      <key>type</key>
      <string>image</string>
      <key>url</key>
      <string>http://{{SERVER_URL}}/imagr/images/osx_10.12.5_16F73.hfs.dmg</string>
      <key>verify</key>
      <true/>
    </dict>
  </array>
  <key>description</key>
  <string>Restores macOS 10.12.5 16F73 and restarts the computer.</string>
  <key>hidden</key>
  <false/>
  <key>name</key>
  <string>10.12 Sierra Restore</string>
  <key>restart_action</key>
  <string>restart</string>
</dict>
```







10.10 Boot



10.12 Boot



10.12 Imagr

[Mac](#)[iPad](#)[iPhone](#)[Watch](#)[TV](#)[Music](#)[Support](#)[Q](#)

About System Integrity Protection on your Mac

OS X El Capitan and later includes security technology that helps protect your Mac from malicious software.



Paths and apps that third-party apps and installers can continue to write to include:

- /Applications
- /Library
- /usr/local

System Integrity Protection is designed to allow modification of these protected parts only by processes that are signed by Apple and have special entitlements to write to system files, such as Apple software updates and Apple installers. Apps that you download from the Mac App Store already work with System Integrity Protection. Other third-party software, if it conflicts with System Integrity Protection, might be set aside when you upgrade to OS X El Capitan or later.

System Integrity Protection also helps prevent software from selecting a startup disk. To select a startup disk, choose System Preferences from the Apple menu, then click Startup Disk. Or hold down the Option key while you restart, then choose from the list of startup disks.

Information about products not manufactured by Apple, or independent websites not controlled or tested by Apple, is provided without recommendation or endorsement. Apple assumes no responsibility with regard to the selection, performance, or use of third-party websites or products. Apple makes no representations regarding third-party website accuracy or reliability. Risks are inherent in the use of the Internet. [Contact the vendor](#) for additional information. Other company and product names may be trademarks of their respective owners.



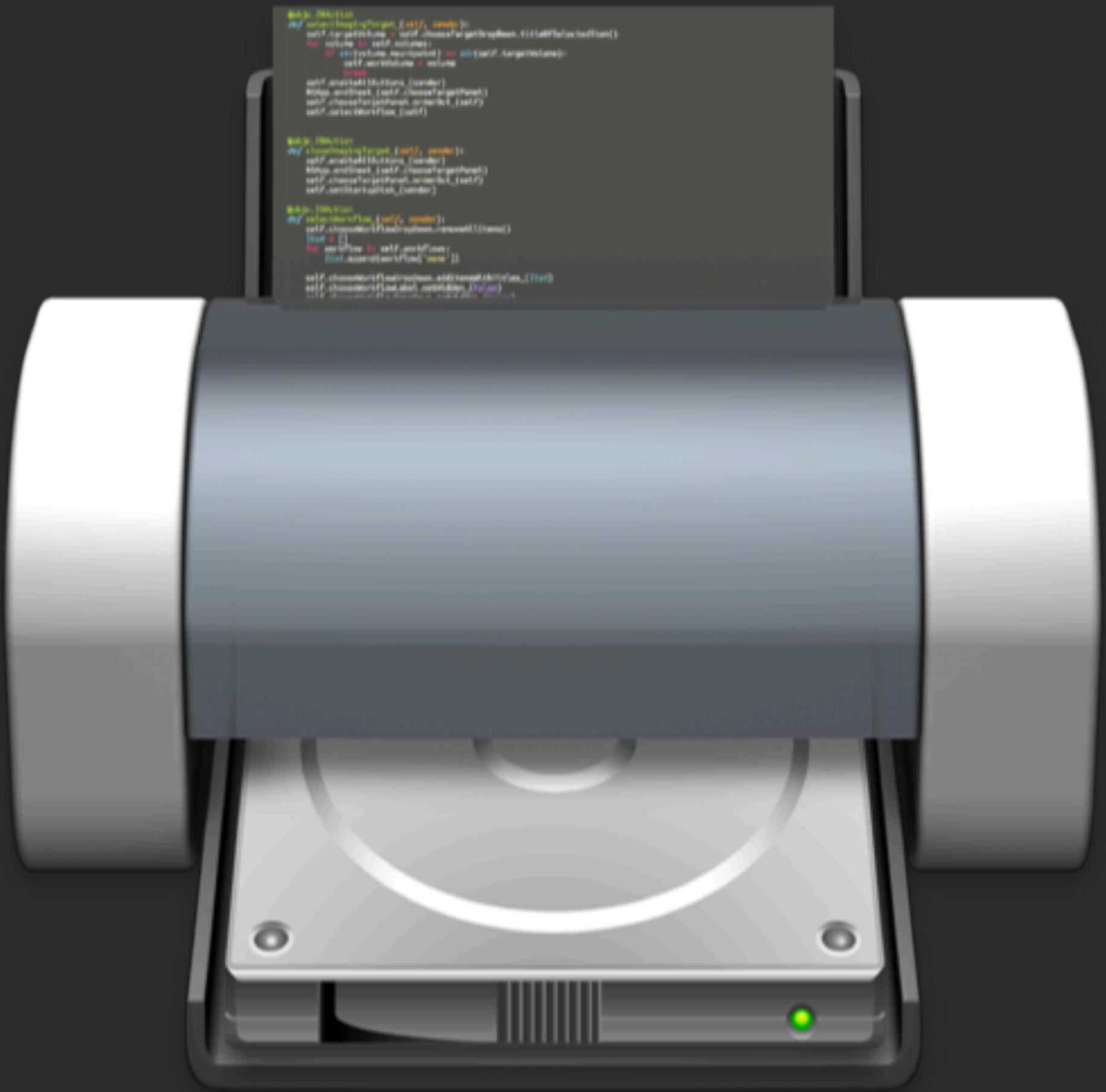
10.10 Boot
(N/A)



10.12 Boot
(SIP)



10.12 Imagr
(No SIP)



SETUP IMAGR ON MAC OS SERVER

IMAGR SETUP

EACH STORE ALREADY HAS:



IMAGR SETUP

WHERE DOES IMAGR LIVE?

/Library/Server/Web/Data/Sites/
Default/imagr

IMAGR SETUP

FIVE BASIC DIRECTORIES

▶ backgrounds

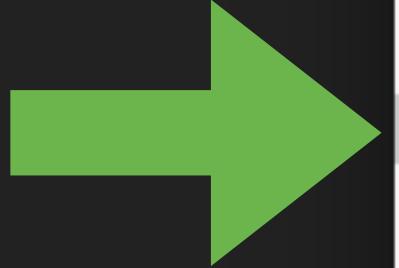
▶ config

▶ images

▶ pkgs

▶ scripts

Name	Date Modified	Size	Kind
▼ backgrounds	Dec 27, 2016, 9:08 PM	--	Folder
10.7.jpg	Dec 15, 2016, 7:24 AM	424 KB	JPEG image
10.10.jpg	Dec 15, 2016, 7:24 AM	1.8 MB	JPEG image
10.12.jpg	Dec 15, 2016, 7:24 AM	442 KB	JPEG image
10.13.jpg	Dec 15, 2016, 7:24 AM	382 KB	JPEG image
▼ config	Today, 12:09 AM	--	Folder
imagr_config_boot.plist	Jun 1, 2017, 4:47 PM	17 KB	property list
imagr_config.plist	Today, 12:09 AM	21 KB	property list
▼ images	May 16, 2017, 3:33 AM	--	Folder
osx_10.7.5_11G63.hfs.dmg	Dec 16, 2016, 1:35 PM	5.29 GB	Disk Image
osx_10.8.5_12F45.hfs.dmg	Dec 16, 2016, 1:31 PM	5.47 GB	Disk Image
osx_10.9.5_13F34.hfs.dmg	Dec 16, 2016, 1:15 PM	6.68 GB	Disk Image
osx_10.10.5_14F27.hfs.dmg	Dec 16, 2016, 9:17 AM	7.85 GB	Disk Image
osx_10.11.6_15G31.hfs.dmg	Dec 16, 2016, 1:49 PM	8.37 GB	Disk Image
osx_10.12.5_16F73.hfs.dmg	May 15, 2017, 3:25 PM	6.77 GB	Disk Image
▼ pkgs	Jun 2, 2017, 6:24 AM	--	Folder
ClearReg.pkg	May 11, 2017, 10:52 AM	2 KB	Installable package
create_inventory-1.0.pkg	Feb 21, 2017, 4:20 PM	17 KB	Installable package
create_service-2.0.pkg	Mar 29, 2017, 3:58 PM	21 KB	Installable package
create_simplyanswers-1.0.pkg	Feb 8, 2017, 4:00 PM	17 KB	Installable package
demo_admin-1.0.pkg	Dec 30, 2016, 4:06 PM	46 KB	Installable package
dockutil-2.0.5.pkg	Dec 6, 2016, 10:31 AM	13 KB	Installable package
InstallOSX_10.7.5_11G63.dmg	Dec 9, 2016, 4:07 PM	4.72 GB	Disk Image
InstallOSX_10.8.5_12F45.dmg	Dec 9, 2016, 4:34 PM	4.45 GB	Disk Image
InstallOSX_10.9.5_13F34.dmg	Dec 9, 2016, 5:00 PM	5.33 GB	Disk Image
InstallOSX_10.10.5_14F27.dmg	Dec 9, 2016, 2:28 PM	5.72 GB	Disk Image
InstallOSX_10.11.6_15G31.dmg	Dec 9, 2016, 3:01 PM	6.21 GB	Disk Image
InstallOSX_10.12.3_16D32.dmg	Jan 23, 2017, 7:12 PM	4.95 GB	Disk Image
JamfQuickAdd-9.99.pkg	Jun 1, 2017, 4:47 PM	3.4 MB	Installable package
outset-2.0.3.pkg	Dec 6, 2016, 10:29 AM	9 KB	Installable package
sa_background-1.0.pkg	Feb 13, 2017, 1:58 PM	218 KB	Installable package
tune_play-5.8.8.pkg	Mar 9, 2017, 1:05 PM	3.3 MB	Installable package
▼ scripts	May 31, 2017, 6:05 AM	--	Folder
demo_settings.sh	Jan 27, 2017, 11:47 AM	1 KB	Plain Text
disable_apple_icloud_diagnostic_and_siri_pop_ups.sh	Oct 6, 2016, 1:55 PM	3 KB	Plain Text
inventory_settings.sh	Feb 21, 2017, 4:20 PM	2 KB	Plain Text
manager_settings.sh	Feb 21, 2017, 4:20 PM	690 bytes	Plain Text
reenable_apple_setup_assistant.sh	May 30, 2017, 3:08 PM	134 bytes	Plain Text
service_settings.sh	Feb 21, 2017, 4:20 PM	2 KB	Plain Text
simplyanswers_settings.sh	Feb 16, 2017, 11:29 AM	2 KB	Plain Text



Server

- TBSC Server
- Alerts
- Certificates
- Logs
- Stats

Accounts

- Users
- Groups

Services

- Caching
- Calendar
- Contacts
- File Sharing
- Mail
- Messages
- Profile Manager
- Time Machine
- VPN
- Websites
- Wiki
- Xcode

Advanced

- DHCP
- DNS
- FTP
- NetInstall
- Open Directory
- Xsan

 Websites

Status: Available on your local network at TBSC-Server.local
Learn about configuring this service [+](#)

Permissions: Private Networks
[Edit Permissions...](#)

Web Applications: Enable PHP
 Enable Python

Websites:

Name	Address	Port
Server Website ⊕	All IP addresses	80
Server Website (SSL) ⊕	All IP addresses	443

[+](#) [-](#) [!\[\]\(b62f6c6d3fab664b4f8ad485ea9de244_img.jpg\)](#)



Server

- TBSC Server
- Alerts
- Certificates
- Logs
- Stats

Accounts

- Users
- Groups

Services

- Caching
- Calendar
- Contacts
- File Sharing
- Mail
- Messages
- Profile Manager
- Time Machine
- VPN
- Websites
- Wiki
- Xcode

Advanced

- DHCP
- DNS
- FTP
- NetInstall
- Open Directory
- Xsan

 NetInstall

ON

Status: ● Available in the Startup Disk pane of System Preferences for macOS clients
[Learn about configuring this service](#)

Storage Locations:  Macintosh HD
[Edit Storage Locations...](#)

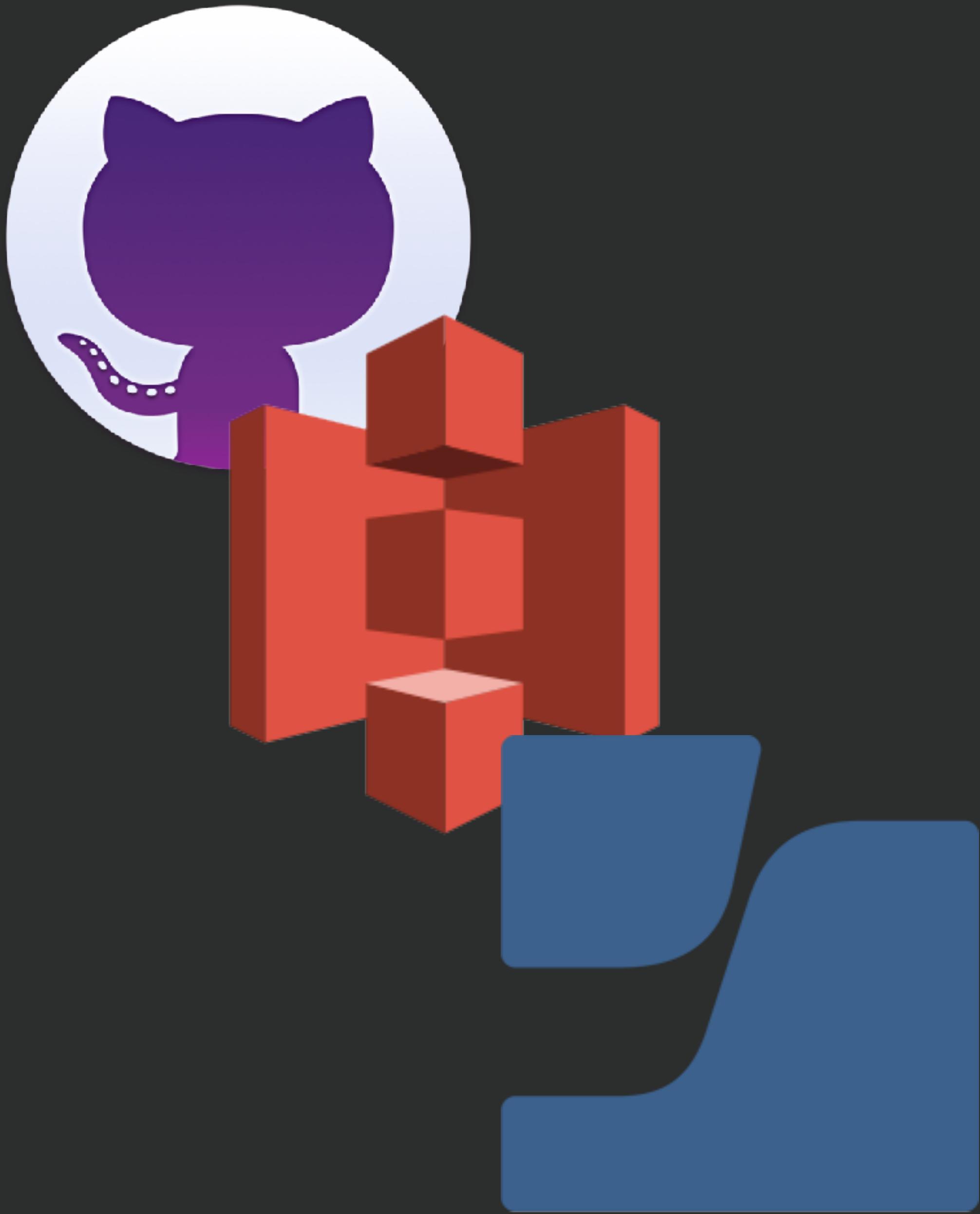
Enable NetInstall on: Ethernet
[Choose Ports...](#)

Global Access Filter: Restrict access to images
[Edit Access Filter...](#)

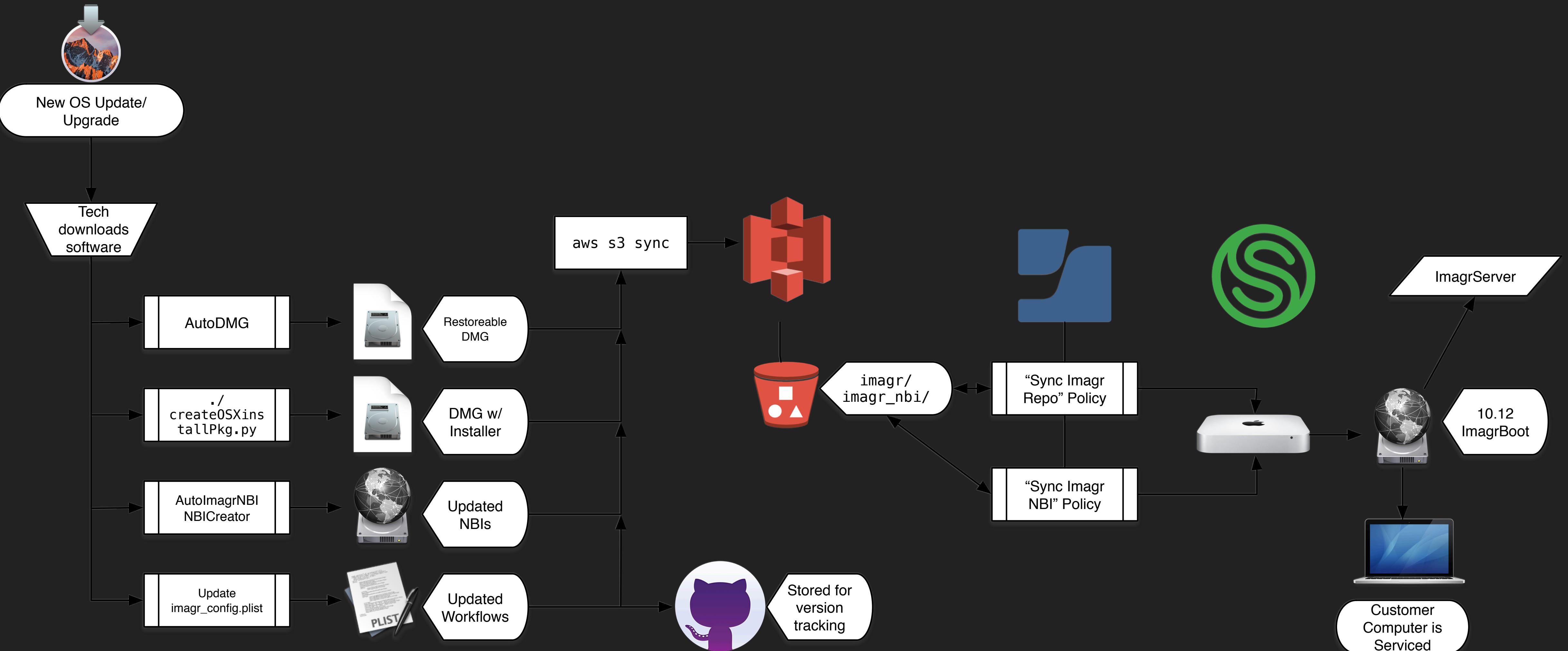
Images:

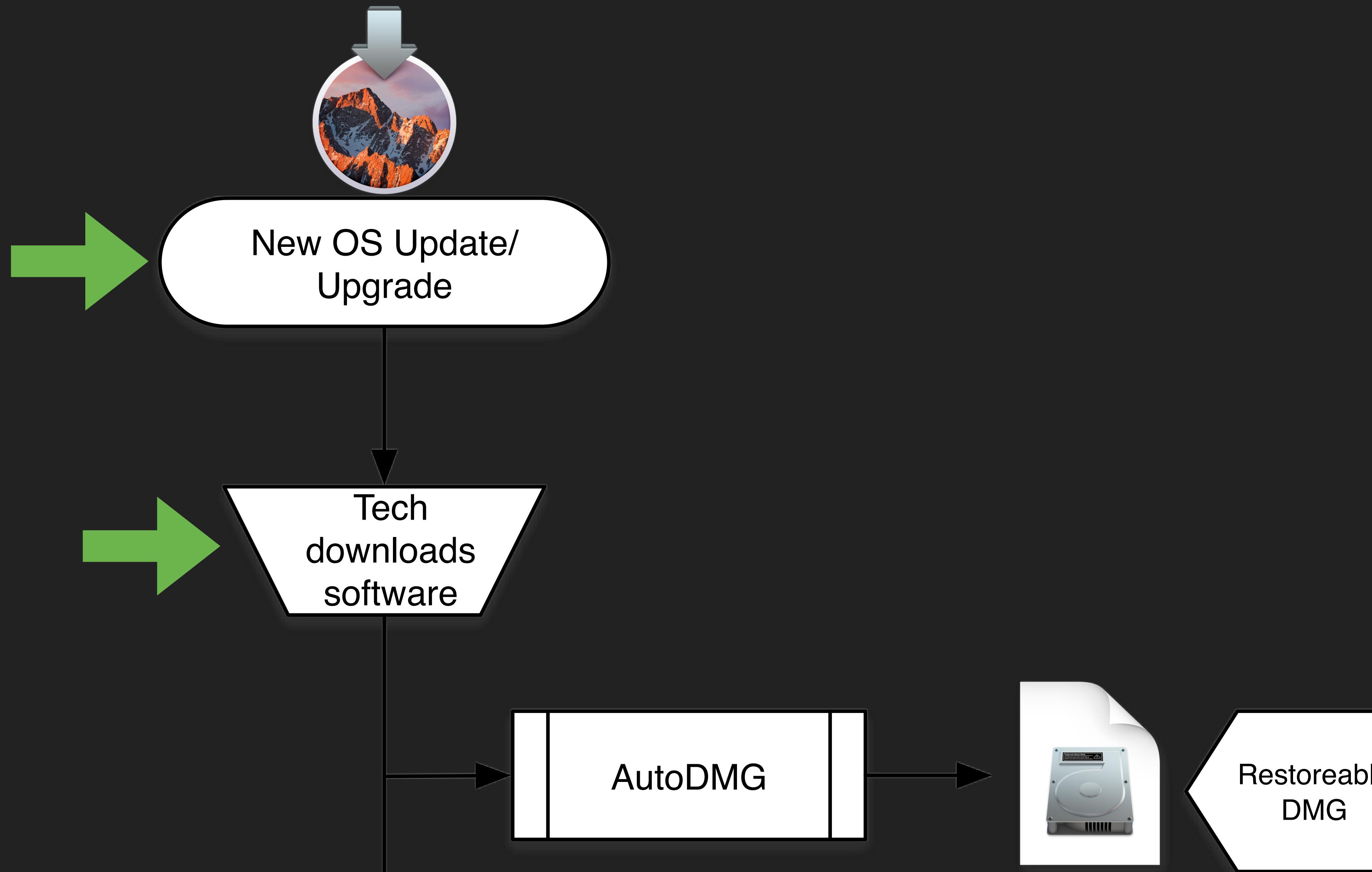
- 10.10 Boot
- 10.12 Boot
- 10.12 Imagr
- 10.13 Boot
- 64bit Apple Service Toolkit v1.5.22v11 access restricted (default)
- Apple Service Toolkit v1.5.22v8 access restricted (default)
- Diagnostics access restricted (default)
- Legacy Apple Service Toolkit v1.5.13v7 access restricted (default)
- TAOS_15E2066_v12 access restricted
- TAOS_16B2657_v4 access restricted
- TAOS_16F2073_v18 access restricted

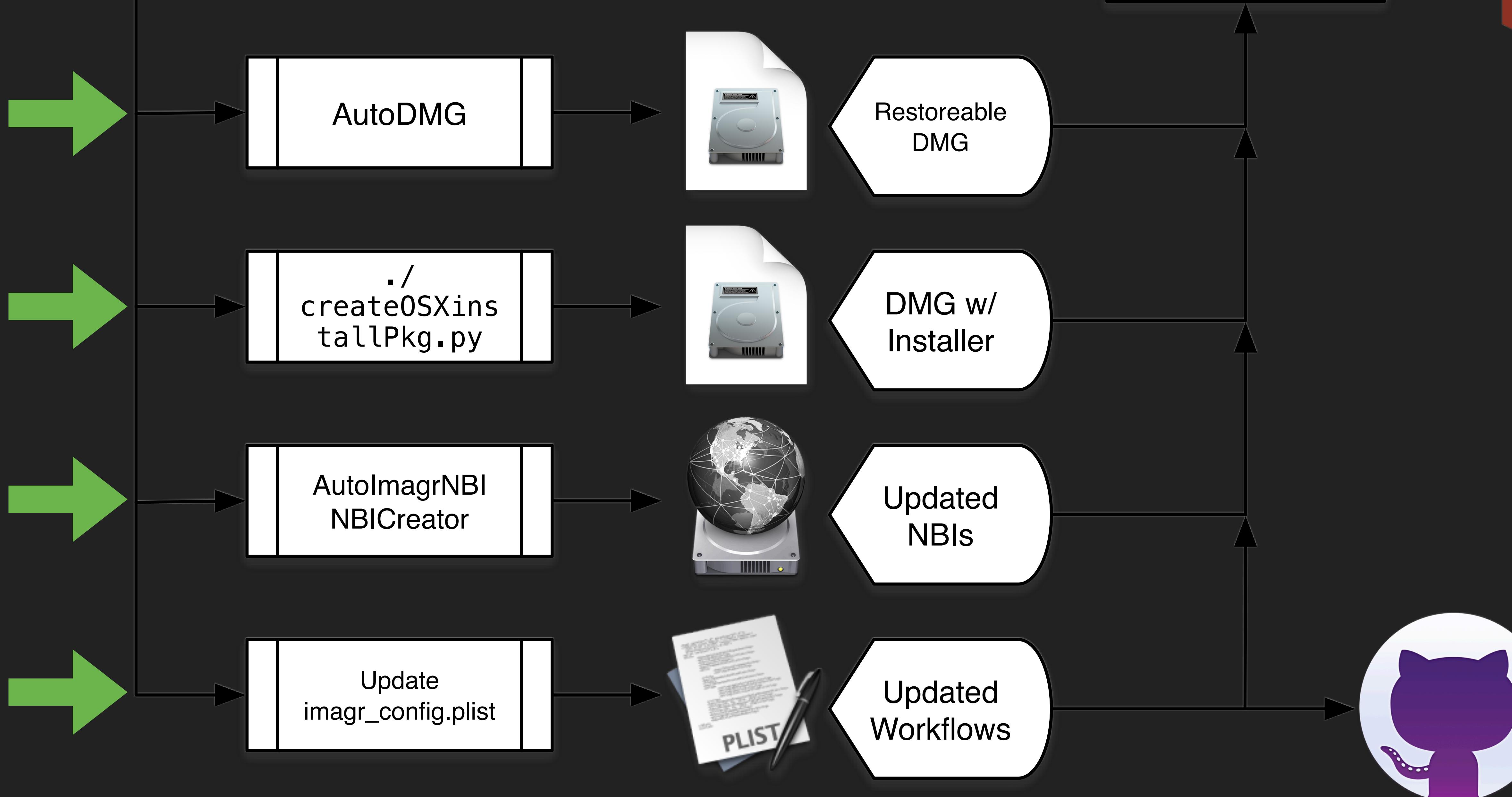
 Search

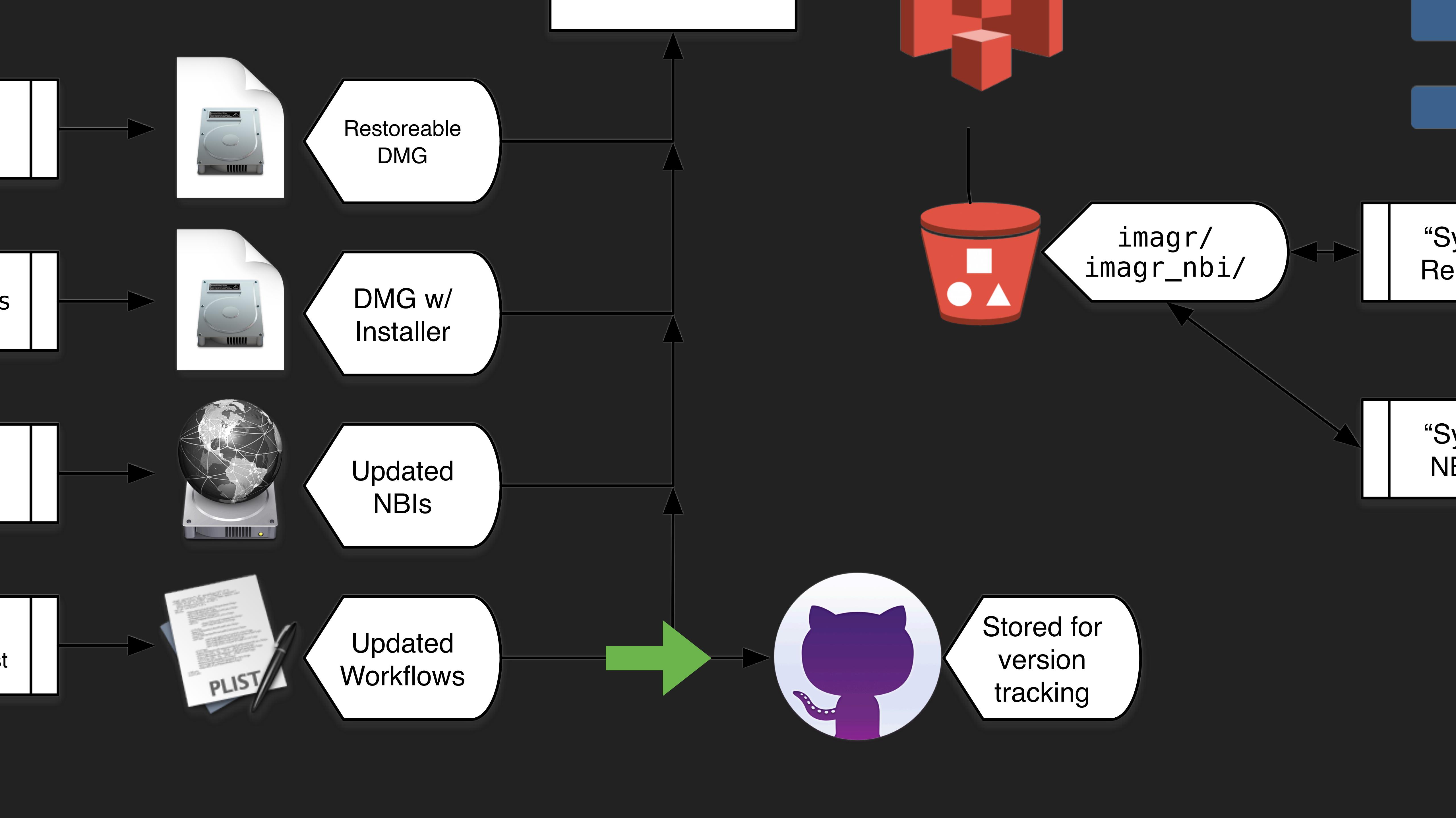


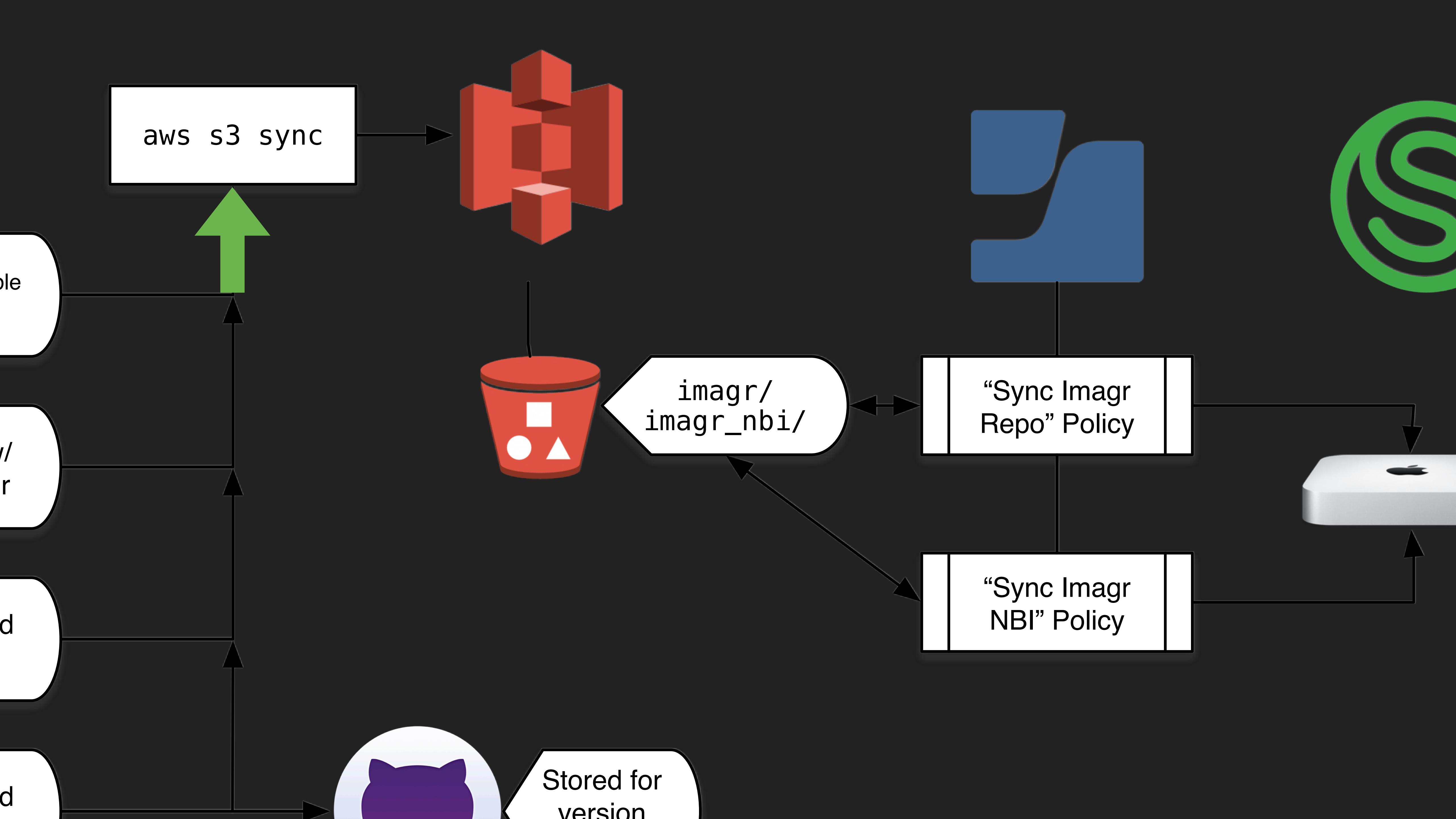
DEPLOY IMAGR TO
50+ LOCATIONS











CREATE AWS IAM USER



Services ▾

Resource Groups ▾



N. Virginia ▾

Support ▾

AWS services

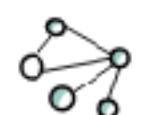
Find a service by name or feature (for example, EC2, S3 or VM, storage).



Recently visited services



EC2



VPC



RDS

All services



Compute

EC2

EC2 Container Service

Lightsail ↗

Elastic Beanstalk

Lambda

Batch



Developer Tools

CodeStar

CodeCommit

CodeBuild

CodeDeploy

CodePipeline

X-Ray



Internet of Things

AWS IoT

AWS Greengrass



Contact Center

Amazon Connect

Helpful tips



Manage your costs

Get real-time billing alerts based on your cost and usage budgets. [Start now](#)



Create an organization

Use AWS Organizations for policy-based management of multiple AWS accounts. [Start now](#)

Explore AWS



Services

Resource Groups



N. Virginia

Support

AWS services

iam



IAM

Manage User Access and Encryption Keys



IAM



Route 53



API Gateway



Certificate Manager



Lambda

All services



Compute

EC2

EC2 Container Service

Lightsail

Elastic Beanstalk

Lambda

Batch



Developer Tools

CodeStar

CodeCommit

CodeBuild

CodeDeploy

CodePipeline

X-Ray



Internet of Things

AWS IoT

AWS Greengrass



Contact Center

Amazon Connect

<https://console.aws.amazon.com/console/home?region=us-east-1#>

Helpful tips



Manage your costs

Get real-time billing alerts based on your cost and usage budgets. [Start now](#)

Create an organization

Use AWS Organizations for policy-based management of multiple AWS accounts. [Start now](#)

Explore AWS

New Product Announcements

View the latest announcements from the AWS Summit - San Francisco. [Learn more](#).

**Services** ▾**Resource Groups** ▾

Global ▾

Support ▾

Search IAM

Add user**Delete user**

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Add user

1

Details

2

Permissions

3

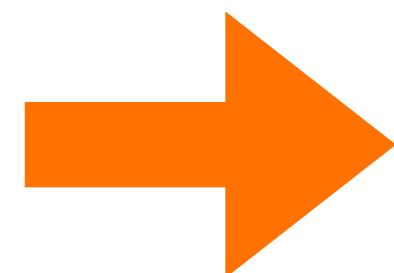
Review

4

Complete

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)



User name*

imagr-demo

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

[Feedback](#)[English](#)

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Select AWS access type

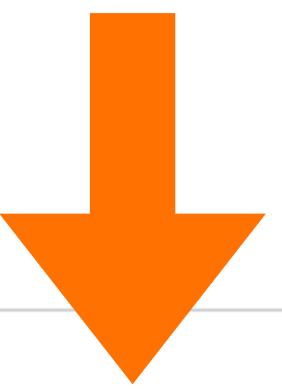
Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

**Access type*****Programmatic access**

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

**AWS Management Console access**

Enables a **password** that allows users to sign-in to the AWS Management Console.

*** Required**[Cancel](#)[Next: Permissions](#)[Feedback](#)[English](#)

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

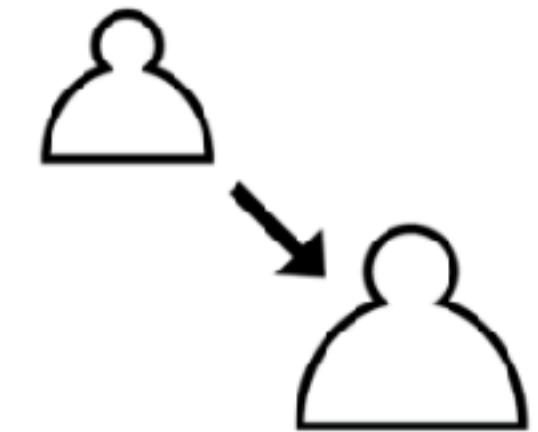
[Privacy Policy](#)[Terms of Use](#)

[Services](#)[Resource Groups](#)[Global](#)[Support](#)

Add user

[Details](#)[Permissions](#)[Review](#)[Complete](#)

Set permissions for **imagr-demo**

[Add user to group](#)[Copy permissions from
existing user](#)[Attach existing policies
directly](#)[Feedback](#)[English](#)

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)

[Services](#)[Resource Groups](#)

[REDACTED]

[Global](#)[Support](#)

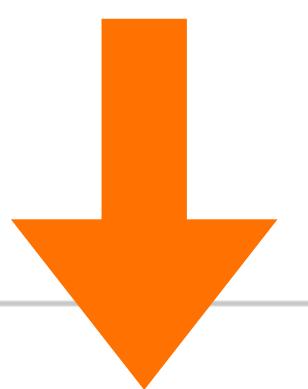
This user has no permissions

You haven't given this user any permissions. This means that the user has no access to any AWS service or resource. Consider returning to the previous step and adding some type of permissions.

User details

User name imagr-demo

AWS access type Programmatic access - with an access key

[Cancel](#)[Previous](#)[Create user](#)[Feedback](#)[English](#)

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)



Services ▾

Resource Groups ▾



Global ▾

Support ▾

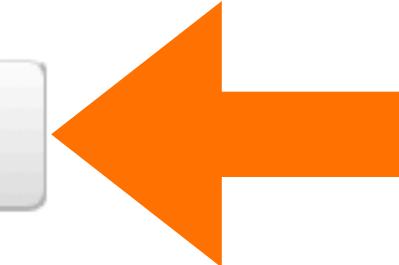


Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://tech-brands.signin.aws.amazon.com/console>

Download .csv



User	Access key ID	Sec
► imagr-demo	AKIAIBJS7JOQJAD5U3GA	*****

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Search IAM

Dashboard

Groups

Users

Roles

Policies

Identity providers

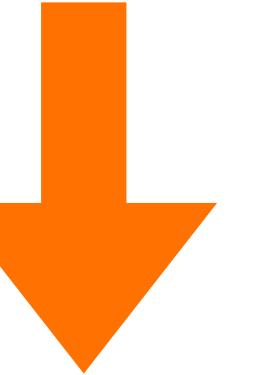
Account settings

Credential report

Encryption keys

Users > imagr-demo

Summary

**User ARN** arn:aws:iam::382659350623:user/imagr-demo**Path** /**Creation time** 2017-07-12 19:06 EDT**Permissions****Groups (0)****Security credentials****Access Advisor**

Get started with permissions

This user doesn't have any permissions yet. Get started by adding the user to a group, copying permissions from another user, or attaching a policy directly. [Learn more](#)

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Search IAM

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

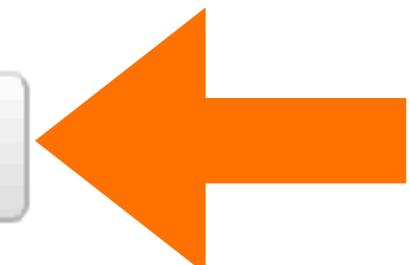
Credential report

Encryption keys

Signing certificates **None**

Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. [Learn more](#)

[Create access key](#)

Access key ID	Created	Last used	Status	
AKIAIBJS7JOQJAD5U3GA	2017-07-12 19:06 EDT	N/A	Active	Make inactive

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate access to AWS CodeCommit repositories. [Learn more](#)

[Feedback](#)[English](#)

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)

AWS IAM ACCESS KEY

ACCESS KEY ID:

AAAAAAAAAAABB BBBB BBBB BBBB

ACCESS SECRET KEY:

jsdkjkas&67as3m38ja/
m11n/sd6fm+n23ad9s/K

**CREATE AWS S3
BUCKET**



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Welcome to Amazon S3. Create new buckets or select an existing bucket to view and configure properties.

[Documentation](#)

Amazon S3

[Switch to the old console](#)[Discover the new console](#)[Quick tips](#) Search for buckets [+ Create bucket](#)[Delete bucket](#)[Empty bucket](#)

0 Buckets

0 Regions



Bucket name

Region

Date created

Create bucket

1 Name and region

② Set properties

③ Set permissions

4 Review

Name and region

Bucket name 

► imagr-assets-demo

Region

► US East (N. Virginia)

Copy settings from an existing bucket

Select bucket (optional)

Create

Cancel

Next

**Services** ▾**Resource Groups** ▾

Global ▾

Support ▾



Amazon S3 > imagr-assets-demo

Objects**Properties****Permissions****Management****Upload****+ Create folder****More ▾**

US East (N. Virginia)



This bucket is empty. Upload new objects to get started.

**GRANT PERMISSIONS
TO IAM USER**



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Search IAM

Create policy

Policy actions ▾



Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys



Create policy

Policy actions ▾

	Policy name ▾	Type	Attachments ▾	Description
<input type="checkbox"/>	AdministratorAccess	Job function	1	Provides full access to AWS services
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	AWS managed	0	Provides full access to create/
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAcc...	AWS managed	0	Provides full access to invoke
<input type="checkbox"/>	AmazonAPIGatewayPushToCloud...	AWS managed	0	Allows API Gateway to push lo
<input type="checkbox"/>	AmazonAppStreamFullAccess	AWS managed	0	Provides full access to Amazo
<input type="checkbox"/>	AmazonAppStreamReadOnlyAccess	AWS managed	0	Provides read only access to /

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use



Create Policy

Step 1 : Create Policy

Step 2 : Set Permissions

Step 3 : Review Policy

A policy is a document that formally states one or more permissions. Create a policy by copying an AWS Managed Policy, using the Policy Generator, or typing your own custom policy.

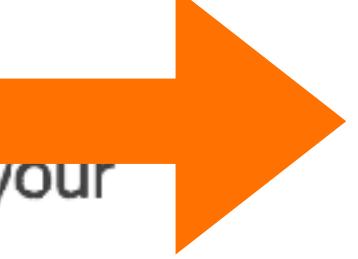
Copy an AWS Managed Policy

Start with an AWS Managed Policy, then customize it to fit your needs.

Select

Policy Generator

Use the policy generator to select services and actions from a list. The policy generator uses your selections to create a policy.



Select

Create Your Own Policy

Use the policy editor to type or paste in your own policy.

Select

Cancel



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Create Policy

[Step 1 : Create Policy](#)**Step 2 : Set Permissions**[Step 3 : Review Policy](#)

The policy generator enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [Overview of Policies](#) in Using AWS Identity and Access Management.

The diagram illustrates the sequential steps in creating a policy:

- Effect**: Allow Deny
- AWS Service**: Amazon S3
- Actions**: 4 Action(s) Selected
- Amazon Resource Name (ARN)**: arn:aws:s3:::imagr-assets-demo
- Add Conditions (optional)**
- Add Statement**

[Cancel](#)[Previous](#)[Next Step](#)

AWS IAM ACCESS KEY

ACTIONS:

ListBucket

GetObject

PutObject

DeleteObject



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Create Policy

Step 1 : Create Policy

Step 2 : Set
Permissions

Step 3 : Review Policy

The policy generator enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see [Overview of Policies](#) in Using AWS Identity and Access Management.

Effect Allow Deny

AWS Service

Amazon S3

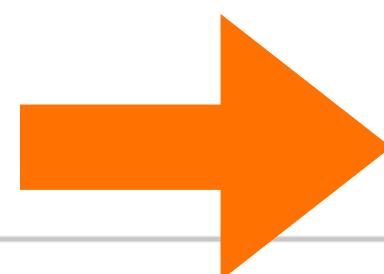
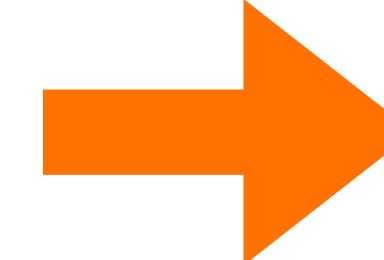
Actions

4 Action(s) Selected

arn:aws:s3:::imagr-assets-demo

Add Conditions (optional)

Add Statement



Cancel

Previous

Next Step



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Create Policy

Step 1 : Create Policy

Step 2 : Set
Permissions

Step 3 : Review Policy

Actions

-- Select Actions --

Amazon Resource Name
(ARN)

*

Add Conditions (optional)

Add Statement

Effect	Action	Resource	
Allow	s3:DeleteObject s3:GetObject s3>ListBucket s3:PutObject	arn:aws:s3:::imagr-assets-demo	Remove

[Cancel](#)[Previous](#)[Next Step](#)



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Create Policy

Step 1 : Create Policy

Step 2 : Set
Permissions

Step 3 : Review Policy



Customize permissions by editing the following policy document or use the IAM Policy Simulator to learn more about the effects of your policy. To test the effects of this policy before applying your changes, see [Overview of Policies](#) in the *Using IAM* guide. To test the effects of this policy before applying your changes, use the [IAM Policy Simulator](#).

Policy Name

Description

Allows read/write access to the [imagr-assets-demo](#) S3 bucket.

Policy Document

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [
```



Use autoformatting for policy editing

Cancel

Validate Policy

Previous

Create Policy

AWS IAM POLICY DOCUMENT

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Stmt1499565196000",  
            "Effect": "Allow",  
            "Action": [  
                "s3:DeleteObject",  
                "s3:GetObject",  
                "s3>ListBucket",  
                "s3:PutObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::imagr-assets-demo"  
            ]  
        }  
    ]  
}
```



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Create Policy

Step 1 : Create Policy

Step 2 : Set
Permissions

Step 3 : Review Policy

Customize permissions by editing the following policy document or paste an existing document. For more information about the policy language, see [Overview of Policies](#) in the *Using IAM* guide. To test the effects of this policy before applying your changes, use the [IAM Policy Simulator](#).

Policy Name

imagr-assets-demo

Description

Allows read/write access to the [imagr-assets-demo](#) S3 bucket.

Policy Document

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [
```



Use autoformatting for policy editing

Cancel

Validate Policy

Previous

Create Policy



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Create Policy

Step 1 : Create Policy

Step 2 : Set Permissions

Step 3 : Review Policy

This policy is valid.

Policy Name

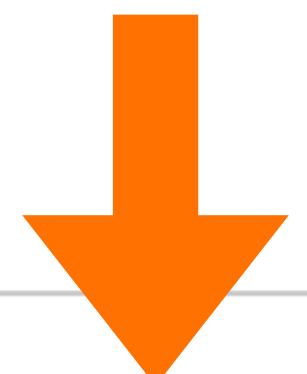
imagr-assets-demo

Description

Allows read/write access to the ~~imagr-**assets**-demo~~ S3 bucket.

Policy Document

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Action": "s3:GetObject",  
6             "Resource": "arn:aws:s3:::mybucket/*",  
7             "Effect": "Allow",  
8             "Principal": "*"  
9         }  
10    ]  
11 }
```



Use autoformatting for policy editing

Cancel

Validate Policy

Previous

Create Policy



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Search IAM

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

**imagr-assets-demo has been created.**

You are now ready to attach your policy to users, groups, and roles.

**Create policy****Policy actions ▾**

Filter: Policy type ▾



Search

	Policy name ▾	Type	Attachments ▾	Description
<input type="checkbox"/>	AdministratorAccess	Job function	1	Provides full access to AWS services
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	AWS managed	0	Provides full access to create, update, and delete API resources
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	AWS managed	0	Provides full access to invoke API endpoints
<input type="checkbox"/>	AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	0	Allows API Gateway to push logs to CloudWatch Logs



Services

Resource Groups



Global

Support

Search IAM

Dashboard

Groups

Users

Roles

Policies

Identity providers

Account settings

Credential report

Encryption keys

Users > imagr-demo

Summary

**User ARN** arn:aws:iam::382659350623:user/imagr-demo**Path** /**Creation time** 2017-07-12 19:06 EDT**Permissions****Groups (0)****Security credentials****Access Advisor**

Get started with permissions

This user doesn't have any permissions yet. Get started by adding the user to a group, copying permissions from another user, or attaching a policy directly. [Learn more](#)

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)



Services

Resource Groups



Global

Support

Add permissions to imagr

1

Permissions

2

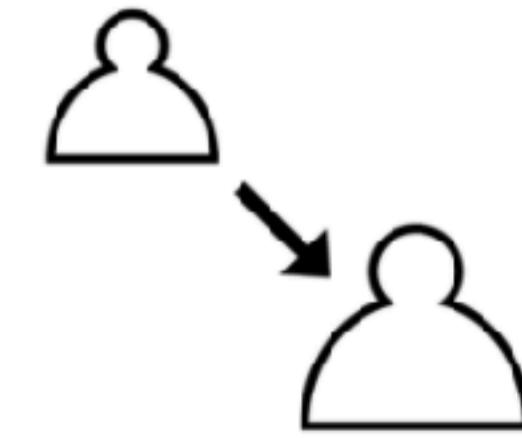
Review

Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.



Add user to group

Copy permissions from
existing userAttach existing policies
directly

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use



Services

Resource Groups



Global

Support

Attach one or more existing policies directly to the user or create a new policy. [Learn more](#)

[Create policy](#)[Refresh](#)

		Policy name	Type	Attachments	Description
<input type="checkbox"/>	▶	imagr-assets-demo	Customer managed	0	Allows read/write access to the imagr-assets-...

[Feedback](#)[English](#)

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)



Services ▾

Resource Groups ▾



Global ▾

Support ▾



Cancel

Next: Review

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use



Services ▾

Resource Groups ▾



Global ▾

Support ▾

Add permissions to imagr

1

Permissions

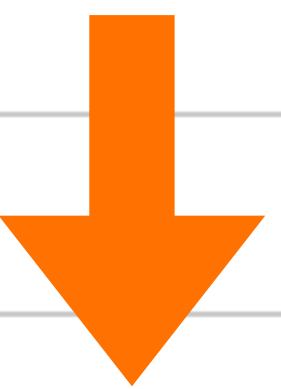
2

Review

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	imagr-assets-demo



Cancel

Previous

Add permissions

[Feedback](#)[English](#)

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)

SYNC TO AWS

SYNC TO AWS

INSTALL AWS_CLI

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
```

```
$ unzip awscli-bundle.zip
```

```
$ sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

SYNC TO AWS

INSTALL AWS_CLI

```
$ brew install awscli
```

SYNC TO AWS

EDIT BASH_PROFILE

```
$ nano ~/.bash_profile
```

BASH PROFILE

INCLUDE SOMEWHERE

```
# AWS IAM Credentials
export AWS_ACCESS_KEY="AAAAAAAAABBBBBBBBBB"
export AWS_SECRET_KEY="jsdkjkas&67as3m38ja/
m11n/sd6fm+n23ad9s/K"
```

SYNC TO AWS

LOAD BASH_PROFILE

```
$ source ~/.bash_profile
```

SYNC TO AWS

USE AWS CLI

```
$ aws s3 sync /imagr/repo/path/ \
  s3://imagr-assets-demo/repo/ \
  --exclude \"*.DS_Store\" \
  --delete
```

BASH PROFILE

INCLUDE SOMEWHERE

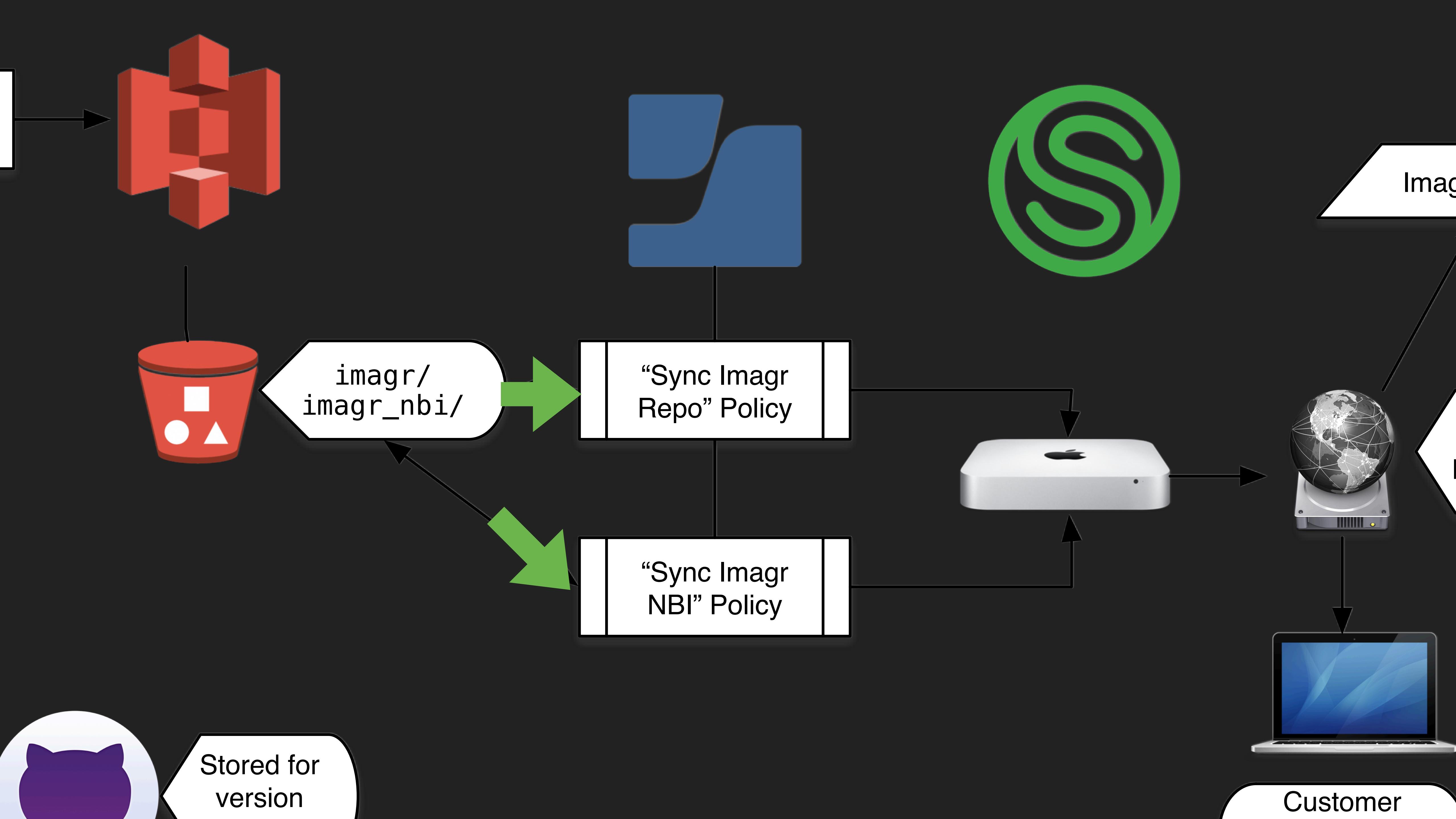
```
# Imagr upload commands
alias upload-imagr-repo="aws s3 sync /imagr/
repo/path/ s3://imagr-assets-demo/repo/
--exclude \"*.DS_Store*\" --delete"

alias upload-imagr-nbi="aws s3 sync /imagr/nbi/
path s3://imagr-assets-demo/nbidir/ --exclude
\"*.DS_Store*\" --delete"
```

SYNC TO AWS

USE AWS CLI

```
$ upload-imagr-repo && upload-imagr-nbi
```



CREATE JAMF POLICIES

SYNC_IMAGR_REPO.SH

```
#!/bin/bash

#####
# Variables
#####

# AWS Credentials
AWS_ACCESS_KEY_ID="$4"
AWS_SECRET_ACCESS_KEY="$5"
AWS_DEFAULT_REGION="us-west-2"

# Location of AWS CLI binary
aws_binary="/usr/local/bin/aws"

# Directory of local Imagr repo on server
imagr_path="/Library/Server/Web/Data/Sites/Default/imagr/"

# S3 bucket and path containing master Imagr repo
s3_bucket_url="$6"
```

SYNC_IMAGR_REPO.SH

```
#####
# Functions
#####

verify_binary () {

    # Check that the aws cli is installed

    if [[ ! -f "$aws_binary" ]]; then
        echo "Installing AWS CLI..."

        # Download AWS CLI zip file
        /usr/bin/curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "/tmp/awscli-bundle.zip"

        # Unzip the downloaded file
        /usr/bin/unzip /tmp/awscli-bundle.zip -d /tmp

        # Install unzip'd files
        /tmp/awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws

        # Check again, exit if not found
        if [[ ! -f "$aws_binary" ]]; then
            echo "AWS CLI still not found. Exiting."
            exit 1
        fi
    fi
}
```

SYNC_IMAGR_REPO.SH

```
export_creds () {  
  
    # Export AWS credentials for use with the  
    # AWS CLI  
  
    export AWS_ACCESS_KEY_ID="$AWS_ACCESS_KEY_ID"  
    export AWS_SECRET_ACCESS_KEY="$AWS_SECRET_ACCESS_KEY"  
    export AWS_DEFAULT_REGION="$AWS_DEFAULT_REGION"  
}  
  
sync_repo () {  
  
    # Sync the S3 directory to local fileshare and  
    # delete deprecated images and pkgs.  
  
    if [[ ! -d "$imagr_path" ]]; then  
        /bin/mkdir "$imagr_path"  
    fi  
  
    "$aws_binary" s3 sync "$s3_bucket_url" "$imagr_path" --delete --quiet  
}
```

SYNC_IMAGR_REPO.SH

```
update_config () {

    # Update the imagr_config.plist file
    # replacing {{SERVER_URL}} with the IP
    # of the local server

    # Find IP address
    serverIP=$(/usr/bin/facter ipAddress)

    if [[ ! -f "${imagr_path}config/imagr_config.plist" ]]; then
        echo "Config file is not found. Is the Data volume missing?"
        exit 1
    else
        # Replace {{SERVER_URL}} with actual IP of server
        /usr/bin/sed -i '' -e "s/{{SERVER_URL}}/${serverIP}/g" \
            "${imagr_path}config/imagr_config.plist"
    fi
}
```

SYNC_IMAGR_REPO.SH

```
#####
# Code Execution
#####

verify_binary
export_creds
sync_repo
update_config

exit 0
```



Computers



Mobile Devices



Users

Notifications 1

Full JSS ▾

chase.thompson-baugh ▾



Search Inventory



Search VPP Content



Licensed Software



Policies



Configuration Profiles



Managed Preferences



Restricted Software



PreStage Imaging



Mac App Store Apps



Patch Reporting



eBooks



Smart Computer Groups



Static Computer Groups



Enrollment Invitations



PreStage Enrollments



Management Settings



Sync Imagr Repo

Options

Scope

Self Service

User Interaction

Show in JSS Dashboard

General

Scripts

1 Script

General

Display Name

Display name for the policy

Sync Imagr Repo

 Enabled

Site

Site to add the policy to

None

Category

Category to add the policy to

Server

Trigger

Event(s) to use to initiate the policy

 Startup

When a computer starts up. A startup script that checks for policies must be configured in the JSS for this to work

 Login

When a user logs in to a computer. A login hook that checks for policies must be configured in the JSS for this to work

Done

History

Logs

Clone

Delete

Edit

Sync Imagr Repo

Options

Scope

Self Service

User Interaction



General



Scripts

1 Script

When a computer starts up. A startup script that checks for policies must be configured.

- Login

When a user logs in to a computer. A login hook that checks for policies must be configured.

- Logout

When a user logs out of a computer. A logout hook that checks for policies must be configured.

- Network State Change

When a computer's network state changes (e.g. when the network connection changes, when the IP address changes)

- Enrollment Complete

Immediately after a computer completes the enrollment process

- Recurring Check-in

At the recurring check-in frequency configured in the JSS

- Custom

At a custom event

Custom Event

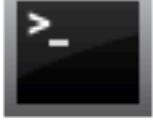
Custom event to use to initiate the policy. For an iBeacon region change event, use

sync-imagr-repo

Sync Imagr Repo

Options Scope Self Service User Interaction

 General

 Scripts
1 Script

Server-Side Limitations Client-Side Limitations

Client-side limitations are enforced based on the settings on computers

Do Not Run Between
Time range during which the policy should not run
8 00 a.m. to 8 00 p.m.

Do Not Run On
Days on which the policy should not run

Sun.

Mon.

Tue.

Wed.

Thu.

Fri.

Sat.

Sync Imagr Repo

Options

Scope

Self Service

User Interaction



General



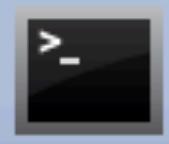
Packages

0 Packages



Software Updates

Not Configured



Scripts

1 Script



Printers

0 Printers



Disk Encryption

Not Configured



Dock Items

0 Dock Items



Local Accounts

0 Accounts

Scripts

sync_imagr_repo.sh

Priority

Priority to use for running the script in relation to other actions

After

Parameter Values

Values for script parameters. Parameters 1-3 are predefined as mount point, computer na

AWS Access Key ID

AWS Secret Access Key

AWS S3 Bucket URL (s3://)

Parameter 7

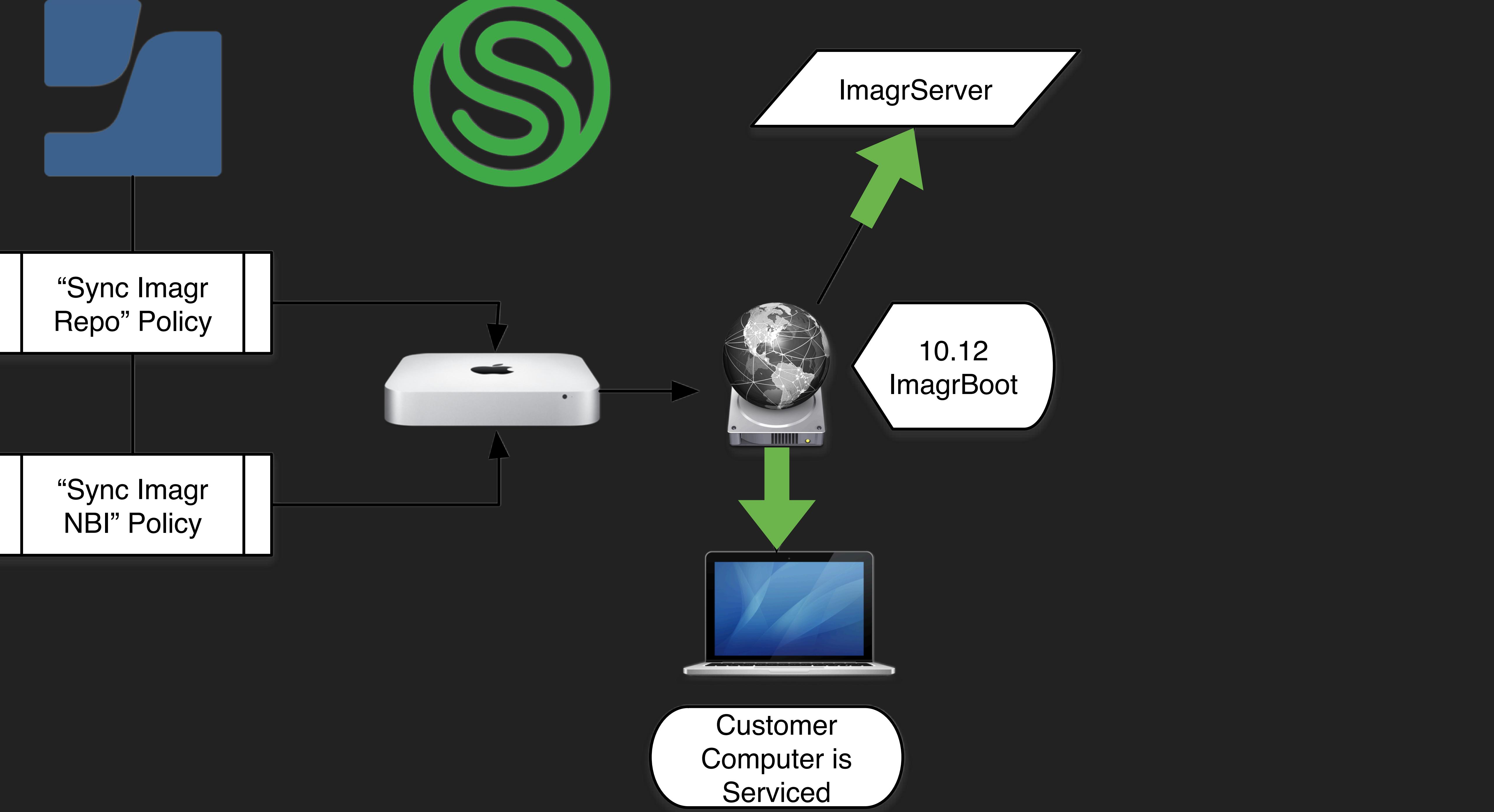
SYNC_IMAGR_REPO.SH

DON'T HAVE JAMF?:

- ▶ Launch Daemon
- ▶ Munki
- ▶ Config management (Puppet,
Chef, etc.)



You just got Jamf'd

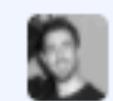




IMAGR SERVER FOR REPORTING

[Code](#)[Issues 0](#)[Pull requests 0](#)[Projects 0](#)[Wiki](#)[Insights ▾](#)

Branch: master ▾

[imagr_server / Readme.md](#)[Find file](#)[Copy path](#)

grahamgilbert Readme

81dfa96 on Jun 11, 2015

1 contributor

7 lines (4 sloc) | 471 Bytes

[Raw](#)[Blame](#)[History](#)

Imagr Server

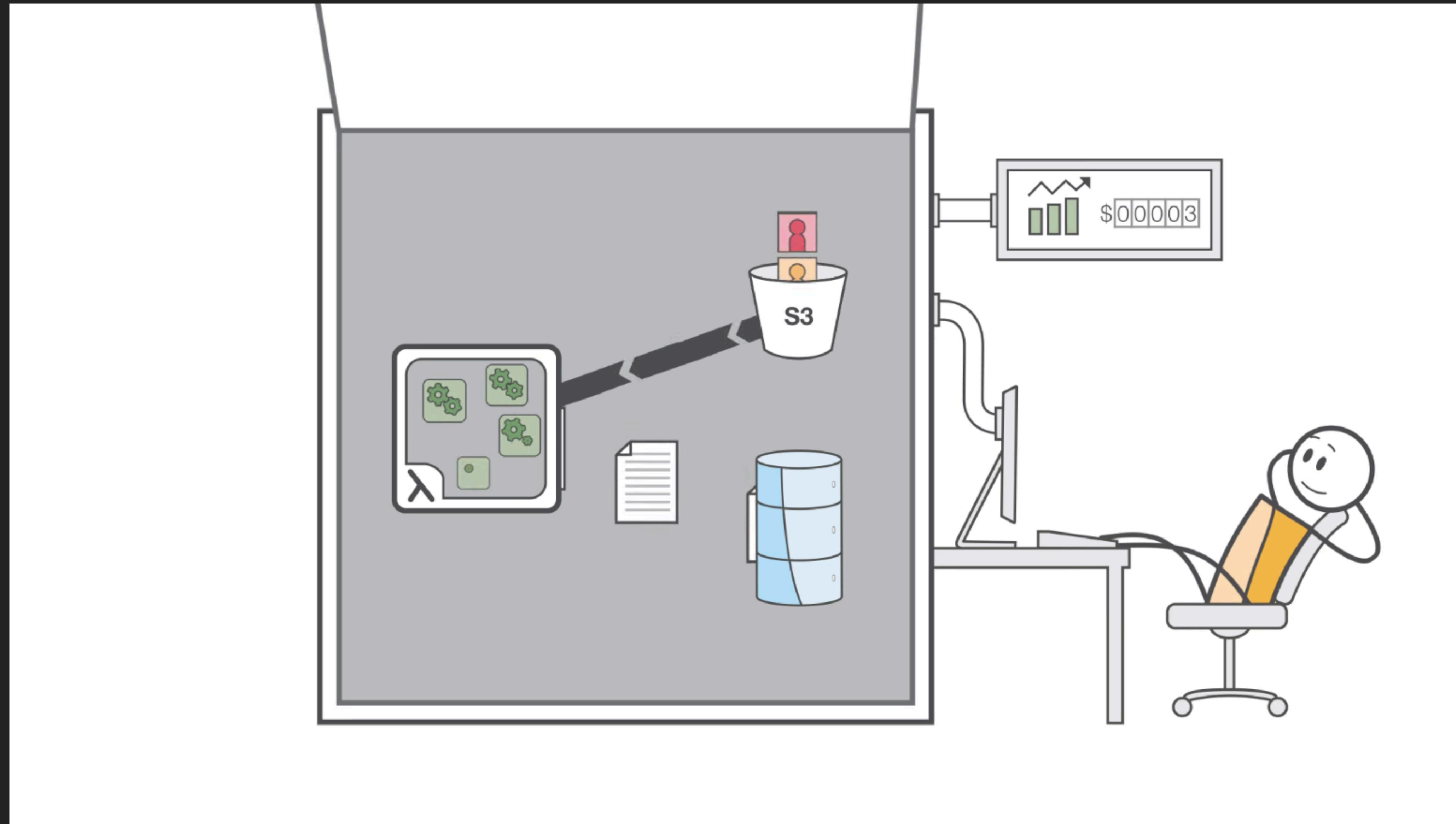
This is a Django application that will accept reports from Imagr. The recommended way of running it is via the [Docker Image](#), but it can be run using Nginx, Apache - but running it in this manner is an exercise left to the reader.

Configuring Imagr

Set your `reporturl` in Imagr's configuration to `http://yourimagrserver/report/`. Imagr will then send its reports to Imagr Server.



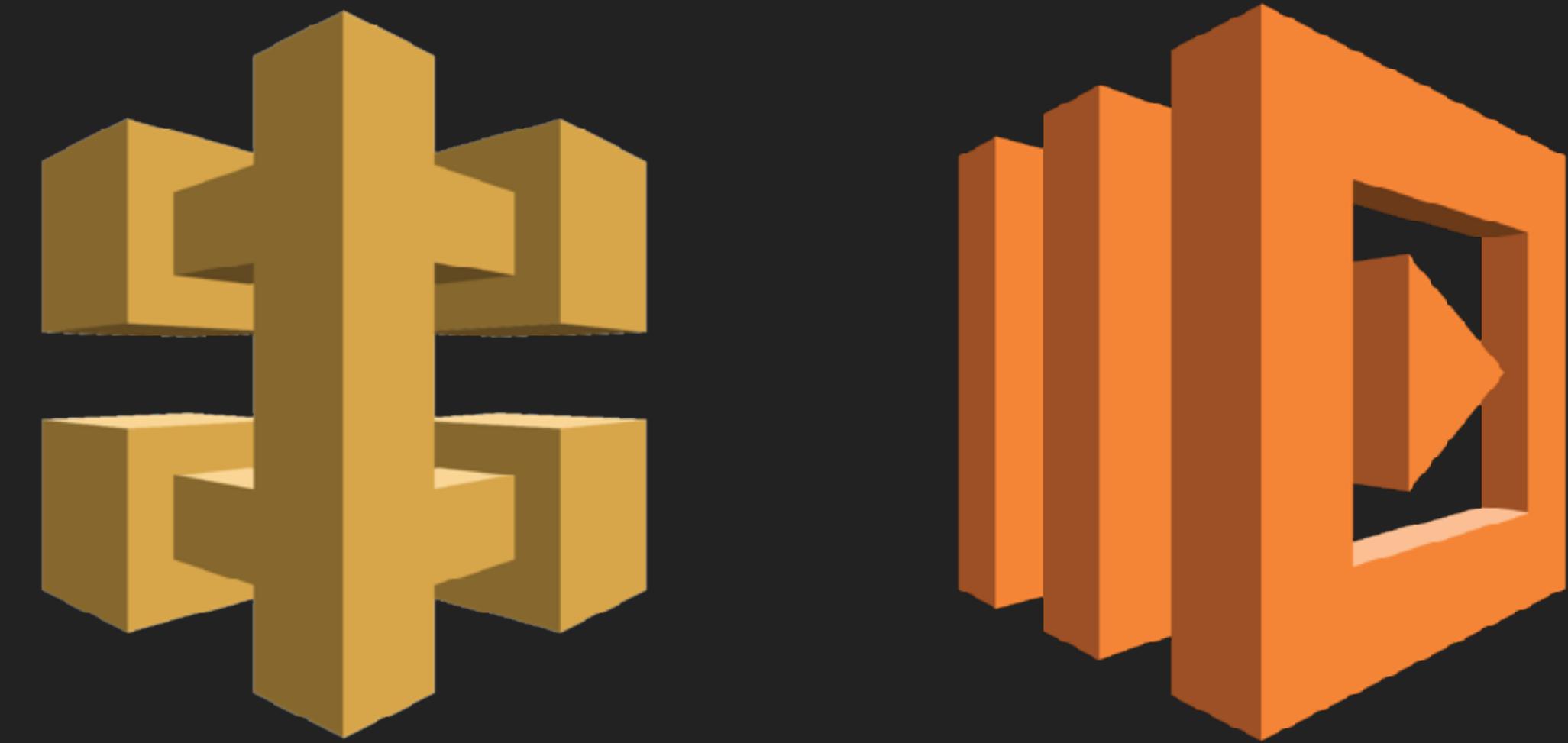
lambda



IMAGR SERVER ON AWS

ZAPPA MAKES IT SUPER EASY TO BUILD
AND DEPLOY ALL PYTHON WSGI
APPLICATIONS ON AWS LAMBDA + API
GATEWAY.

<https://github.com/Miserlou/Zappa>



api lambda

UPDATED LIST

REQUIREMENTS.TXT

Django==1.8.2

django-bootstrap3==5.4.0

wsgiref==0.1.2

requests

django-storages

psycopg2

zappa

IMAGR SERVER ON AWS

SETUP ZAPPA

```
$ virtualenv imagr_env && cd imagr_env
```

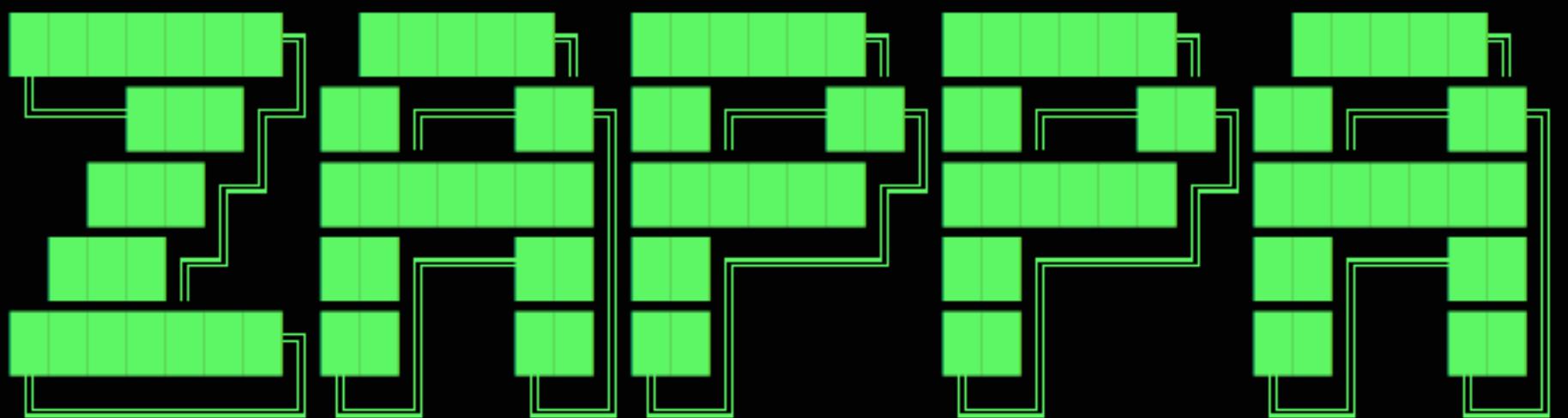
```
$ source /bin/activate
```

```
$ git clone https://github.com/chasetb/imagr\_server/tree/aws imagr_server && cd imagr_server
```

```
$ pip install -R requirements.txt
```

```
$ zappa init
```

```
(imagr_env) Chase-Thompson-Baughs-MacBook-Pro:imagr_server chase$ zappa init
```



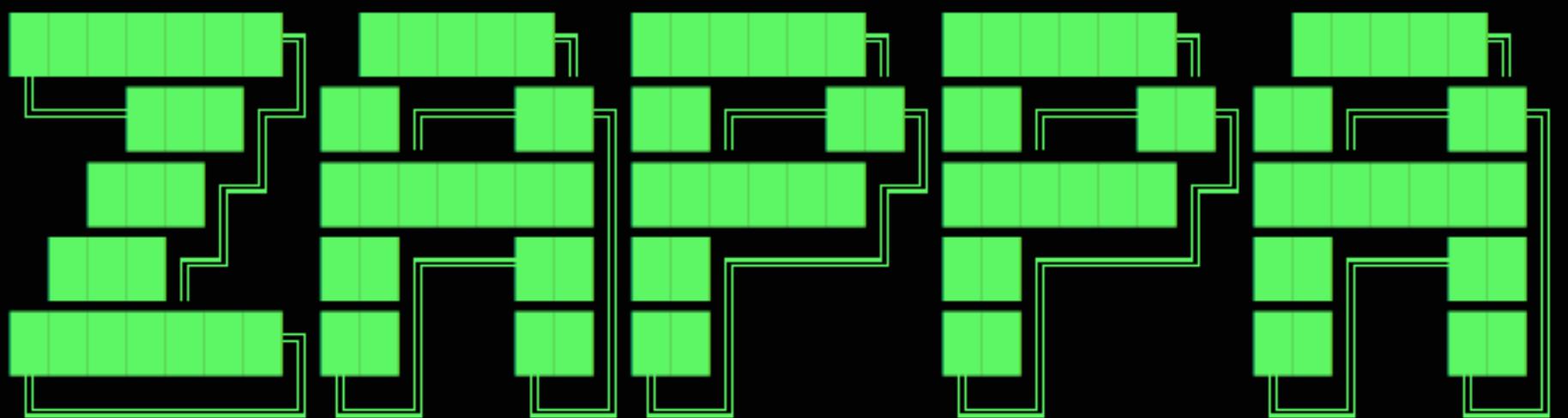
Welcome to Zappa!

Zappa is a system for running server-less Python web applications on AWS Lambda and AWS API Gateway.
This `init` command will help you create and configure your new Zappa deployment.
Let's get started!

Your Zappa configuration can support multiple production stages, like '**dev**', '**staging**', and '**production**'.
What do you want to call this environment (default 'dev'): █



```
(imagr_env) Chase-Thompson-Baughs-MacBook-Pro:imagr_server chase$ zappa init
```



Welcome to Zappa!

Zappa is a system for running server-less Python web applications on AWS Lambda and AWS API Gateway. This `init` command will help you create and configure your new Zappa deployment. Let's get started!

Your Zappa configuration can support multiple production stages, like '**dev**', '**staging**', and '**production**'. What do you want to call this environment (default 'dev'): dev

AWS Lambda and API Gateway are only available in certain regions. Let's check to make sure you have a profile set up in one that will work.

Okay, using profile **default**!

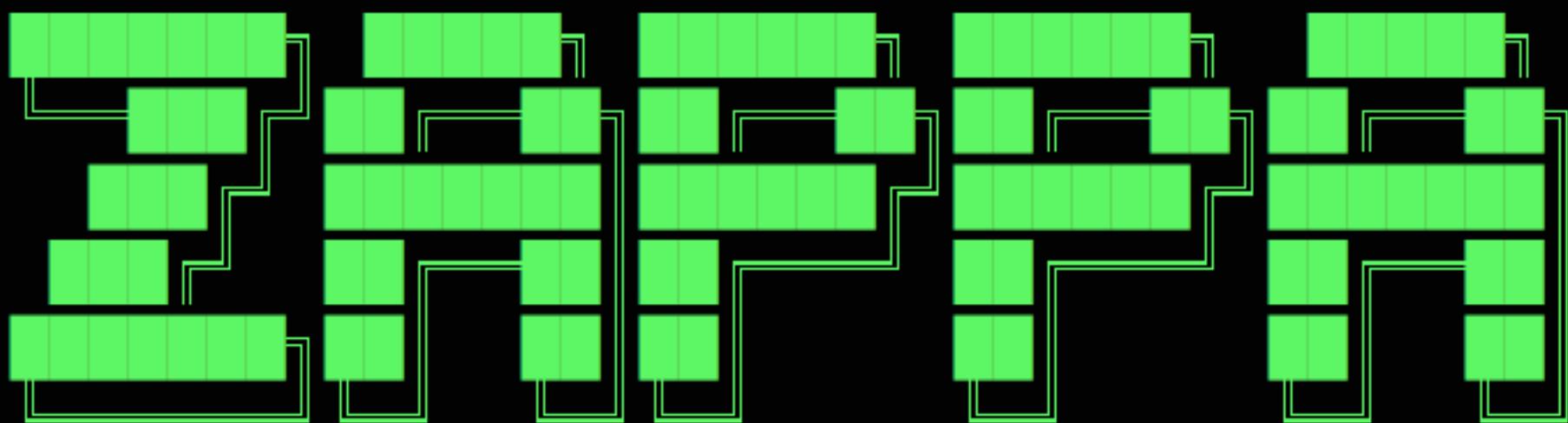
Your Zappa deployments will need to be uploaded to a **private S3 bucket**.

If you don't have a bucket yet, we'll create one for you too.

What do you want call your bucket? (default 'zappa-obz4miok9'): imagr-assets-demo



```
(imagr_env) Chase-Thompson-Baughs-MacBook-Pro:imagr_server chase$ zappa init
```



Welcome to Zappa!

Zappa is a system for running server-less Python web applications on AWS Lambda and AWS API Gateway. This `init` command will help you create and configure your new Zappa deployment. Let's get started!

Your Zappa configuration can support multiple production stages, like '**dev**', '**staging**', and '**production**'. What do you want to call this environment (default 'dev'): dev

AWS Lambda and API Gateway are only available in certain regions. Let's check to make sure you have a profile set up in one that will work.

Okay, using profile **default**!

Your Zappa deployments will need to be uploaded to a **private S3 bucket**.

If you don't have a bucket yet, we'll create one for you too.

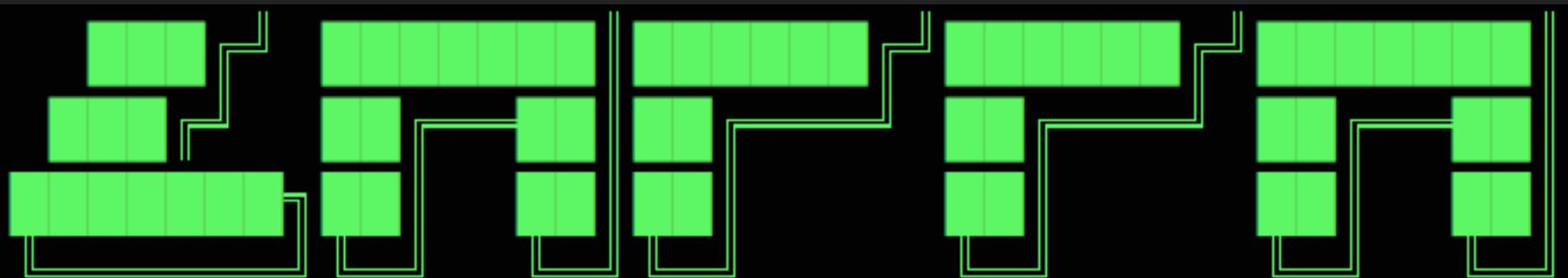
What do you want call your bucket? (default 'zappa-obz4miok9'): imagr-assets-demo

It looks like this is a **Django** application!

What is the **module path** to your project's Django settings?

We discovered: **imagr_server.example_settings**, **imagr_server.settings**

Where are your project's settings? (default 'imagr_server.example_settings'): █ ←



Welcome to Zappa!

Zappa is a system for running server-less Python web applications on AWS Lambda and AWS API Gateway. This `init` command will help you create and configure your new Zappa deployment. Let's get started!

Your Zappa configuration can support multiple production stages, like '**dev**', '**staging**', and '**production**'. What do you want to call this environment (default 'dev'): dev

AWS Lambda and API Gateway are only available in certain regions. Let's check to make sure you have a profile set up in one that will work.

Okay, using profile **default**!

Your Zappa deployments will need to be uploaded to a **private S3 bucket**.

If you don't have a bucket yet, we'll create one for you too.

What do you want call your bucket? (default 'zappa-obz4miok9'): imagr-assets-demo

It looks like this is a **Django** application!

What is the **module path** to your project's Django settings?

We discovered: **imagr_server.example_settings**, **imagr_server.settings**

Where are your project's settings? (default 'imagr_server.example_settings'):

You can optionally deploy to **all available regions** in order to provide fast global service.

If you are using Zappa for the first time, you probably don't want to do this!

Would you like to deploy this application **globally**? (default 'n') [y/n/(p)primary]: n



AWS Lambda and API Gateway are only available in certain regions. Let's check to make sure you have a profile set up in one that will work.
Okay, using profile **default**!

Your Zappa deployments will need to be uploaded to a **private S3 bucket**.

If you don't have a bucket yet, we'll create one for you too.

What do you want call your bucket? (default 'zappa-obz4miok9'): **imagr-assets-demo**

It looks like this is a **Django** application!

What is the **module path** to your project's Django settings?

We discovered: **imagr_server.example_settings**, **imagr_server.settings**

Where are your project's settings? (default 'imagr_server.example_settings'):

You can optionally deploy to **all available regions** in order to provide fast global service.

If you are using Zappa for the first time, you probably don't want to do this!

Would you like to deploy this application **globally**? (default 'n') [y/n/(p)primary]: **n**

Okay, here's your **zappa_settings.json**:

```
{  
  "dev": {  
    "aws_region": "us-east-1",  
    "django_settings": "imagr_server.example_settings",  
    "profile_name": "default",  
    "s3_bucket": "imagr-assets-demo"  
  }  
}
```

Does this look **okay**? (default 'y') [y/n]: **y**

```
If you are using Zappa for the first time, you probably don't want to do this!
Would you like to deploy this application globally? (default 'n') [y/n/(p)primary]: n
```

Okay, here's your **zappa_settings.json**:

```
{  
    "dev": {  
        "aws_region": "us-east-1",  
        "django_settings": "imagr_server.example_settings",  
        "profile_name": "default",  
        "s3_bucket": "imagr-assets-demo"  
    }  
}
```

```
Does this look okay? (default 'y') [y/n]: y
```

Done! Now you can **deploy** your Zappa application by executing:

```
$ zappa deploy dev
```

After that, you can **update** your application code with:

```
$ zappa update dev
```

To learn more, check out our project page on **GitHub** here: <https://github.com/Miserlou/Zappa> and stop by our **Slack** channel here: <https://slack.zappa.io>

Enjoy!,
~ Team **Zappa**!

```
(imagr_env) Chase-Thompson-Baughs-MacBook-Pro:imagr_server chase$ █
```

WAIT

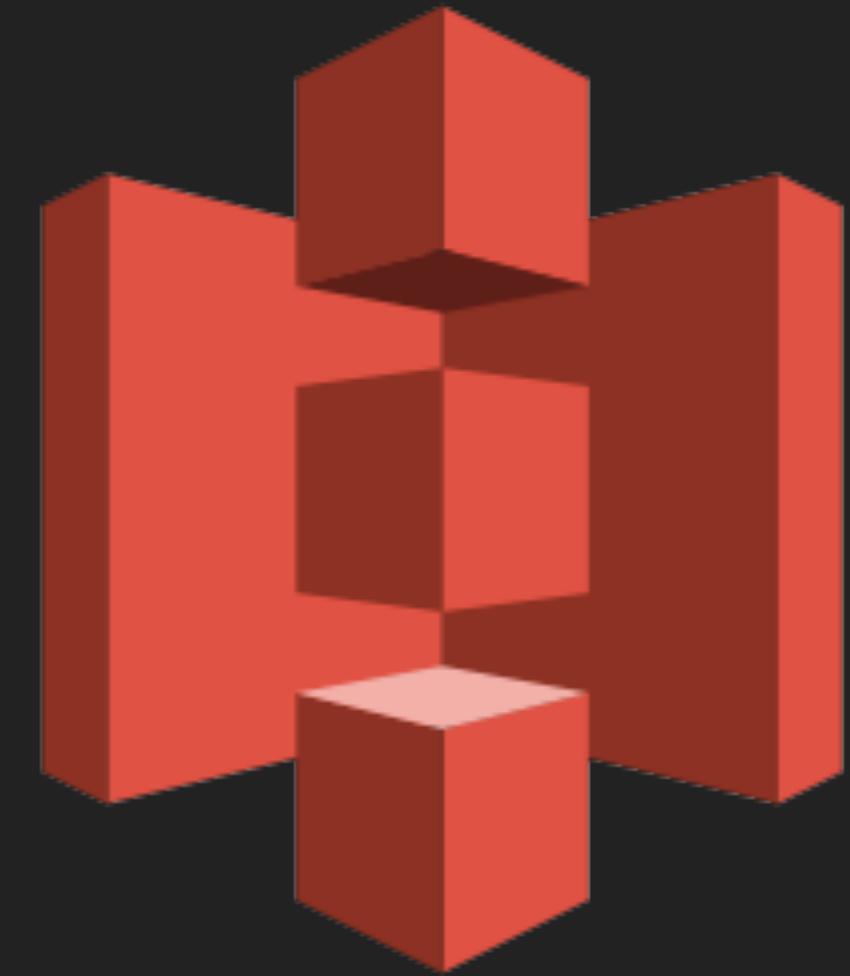
IMAGR SERVER ON AWS

ZAPPA_SETTINGS.JSON

```
{  
  "dev": {  
    "aws_region": "aws-region-name",  
    "django_settings": "imagr_server.settings",  
    "domain": "imagr.yourdomain.com",  
    "lambda_description": "Imagr Serverless Server",  
    "profile_name": "default"  
    "project_name": "ImagrServer",  
    "s3_bucket": "yourbucket",  
  },  
}
```



rds



s3

STATIC ASSETS

DJANGO SETTINGS.PY

```
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.admindocs',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'bootstrap3',
    'report',
    'dashboard',
    'storages'
)
```

STATIC ASSETS

DJANGO SETTINGS.PY

```
AWS_STORAGE_BUCKET_NAME = 'BUCKET_NAME'
```

```
AWS_ACCESS_KEY_ID = 'xxxxxxxxxxxxxxxxxx'
```

```
AWS_SECRET_ACCESS_KEY = 'yyyyyyyyyyyyyyyyyyyyyyyy'
```

```
AWS_S3_CUSTOM_DOMAIN = '%s.s3.amazonaws.com' %
```

```
AWS_STORAGE_BUCKET_NAME
```

```
STATIC_URL = "https://{}/" % AWS_S3_CUSTOM_DOMAIN
```

```
STATICFILES_STORAGE =  
'storages.backends.s3boto.S3BotoStorage'
```

IMAGR SERVER ON AWS

SETUP DJANGO-STORAGES

```
$ python manage.py collectstatic
```

DATABASE STORAGE

DJANGO SETTINGS.PY

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql_psycopg2',  
        'NAME': '',  
        'USER': '',  
        'PASSWORD': '',  
        'HOST': '',  
        'PORT': ''  
}
```



Services

Resource Groups



Oregon

Support

Step 1: Select Engine

 Free tier eligible only 

Select Engine

To get started, choose a DB Engine below and click Select.

Amazon
Aurora



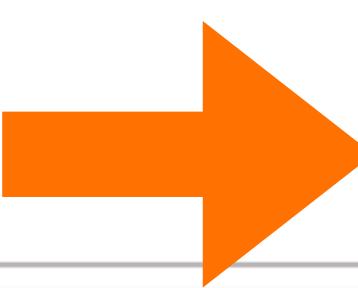
PostgreSQL

Amazon Aurora

Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <\$1/day.

- Up to 5 times the throughput of MySQL and 3 times the throughput of PostgreSQL.
- Up to 64TB of auto-scaling SSD storage.
- 6-way replication across three Availability Zones.
- Up to 15 Read Replicas with sub-10ms replica lag.
- Automatic monitoring and failover in less than 30 seconds.

MySQL-compatible edition

**Select** Feedback English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

Step 1: [Select Engine](#)**Step 2: Specify DB Details**

Step 3: Configure Advanced Settings

i Estimated monthly costs for your instance are as follows:

DB Instance	\$ 13.14
Storage	\$ 0.58
Total	\$ 13.72

Billing estimate is based on on-demand usage as described in [Amazon RDS Pricing](#). Estimate does not include costs for backup storage, IOs (if applicable), or data transfer.

Estimate your monthly costs for the DB Instance using the [RDS Instance Cost](#)

Specify DB Details

Free Tier

The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).

Only show options that are eligible for RDS Free Tier

Instance Specifications

DB Engine postgres

License Model postgresql-license

DB Engine Version PostgreSQL 9.6.2-R1

DB Instance Class db.t2.micro — 1 vCPU, 1 GiB RAM

Multi-AZ Deployment No

Storage Type General Purpose (SSD)

(Minimum: 5 GB, Maximum: 20 GB) Higher allocated storage [may improve IOs](#)

Specify DB Details

Free Tier

The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).

- Only show options that are eligible for RDS Free Tier

Instance Specifications

DB Engine	postgres
License Model	postgresql-license
DB Engine Version	PostgreSQL 9.6.2-R1
DB Instance Class	db.t2.micro — 1 vCPU, 1 GiB RAM
Multi-AZ Deployment	No
Storage Type	General Purpose (SSD)

(Minimum: 5 GB, Maximum: 20 GB) Higher allocated storage may improve IOPS
[View details](#)

DB Instance Class

db.t2.micro — 1 vCPU, 1 GiB RAM

Multi-AZ Deployment

No

Storage Type

General Purpose (SSD)

Allocated Storage*

5 GB

Settings**DB Instance Identifier***

imagrserver-demo

Master Username*

dbadmin

Master Password*

.....

Confirm Password*

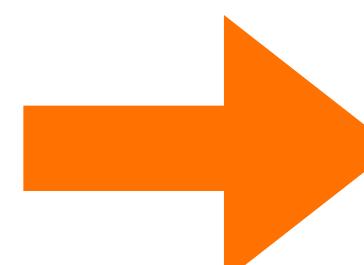
.....|

* Required

Retype the value you specified
for Master Password.**Cancel****Previous****Next Step**

Configure Advanced Settings

Network & Security



VPC*

Default VPC (vpc-31fc2554)



Subnet Group

default



Publicly Accessible

Yes



Availability Zone

No Preference



VPC Security Group(s)

launch-wizard-3 (VPC)
rds-launch-wizard (VPC)
rds-launch-wizard-1 (VPC)
rds-launch-wizard-4 (VPC)

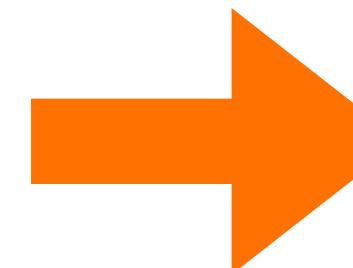
Select the security group or groups that have rules authorizing connections from all of the EC2 instances and devices that need access the data stored in the DB instance. By default, security groups do not authorize any connections; you must specify rules for all instances and devices that will connect to the DB instance.

[Learn More](#).

Database Options

Database Name

Database Options



Database Name

Database Port

DB Parameter Group



Option Group



Copy Tags To Snapshots



Enable Encryption



Backup

Backup Retention Period

Backup Window



Monitoring

Enable Enhanced Monitoring

Select the DB parameter group that defines the configuration settings you want applied to this DB instance. [Learn More.](#)

Enable Encryption

No

Backup

Backup Retention Period

7 days

Backup Window

No Preference

Monitoring

Enable Enhanced Monitoring

No

Maintenance

Auto Minor Version Upgrade

Yes

Maintenance Window

No Preference

* Required

Cancel

Previous

Launch DB Instance



- Step 1: [Select Engine](#)
- Step 2: [Specify DB Details](#)
- Step 3: [Configure Advanced Settings](#)

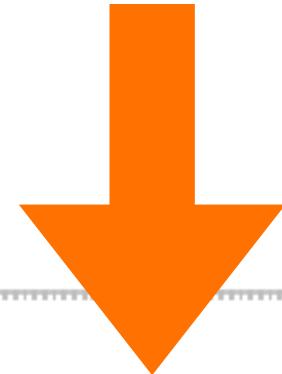
✓ Your DB Instance is being created.

Note: Your instance may take a few minutes to launch.

Connecting to your DB Instance

Once Amazon RDS finishes provisioning your DB instance, you can use a SQL client application or utility to connect to the instance.

[Learn about connecting to your DB instance](#)



[View Your DB Instances](#)



Services

Resource Groups



Oregon

Support

RDS Dashboard

Instances

Clusters

Reserved Instances

Snapshots

Parameter Groups

External Licenses

Option Groups

Subnet Groups

Events

Event Subscriptions

Notifications

Launch DB Instance

Show Monitoring

Instance Actions



Filter: All Instances

Search DB Instances...

Viewing 2 of 2 DB Instances

	Engine	DB Instance	Status	CPU	Current Activity	Maintenance	Class
		PostgreSQL imagrserver-demo	backing-up		0 Connections	None	db.t

Endpoint: [imagrserver-demo.co7hrut0ypq1.us-west-2.rds.amazonaws.com:5432](#) authorized)

	Configuration Details	Security and Network
	ARN arn:aws:rds:us-west-2:382659350623:db:imagrserver-demo	Availability Zone us-west-2c
	Engine PostgreSQL 9.6.2	VPC (vpc-31fc2554)
	License Model Postgresql License	Subnet Group default (Complete)
	Created Time July 9, 2017 at 12:06:52 AM UTC-6	Subnets subnet-d62eda8f subnet-3703d440 subnet-7348f916
	DB Name imagr	Security Groups rds-launch-wizard-4 (sg-ef05c997) (active)
	Username dbadmin	Publicly Accessible Yes
	Option Group default:postgres-9-6 (in-sync)	Endpoint imagrserver-demo.co7hrut0ypq1.us-
	Parameter Group default postgres9_6 (in-sync)	

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

DATABASE STORAGE

DJANGO SETTINGS.PY

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql_psycopg2',  
        'NAME': 'imagr',  
        'USER': 'dbadmin',  
        'PASSWORD': 'supersecret10',  
        'HOST': 'imagrserver-demo.co7hrut0ypq1.us-  
                 west-2.rds.amazonaws.com',  
        'PORT': '5432'  
    }  
}
```

IMAGR SERVER ON AWS

SETUP DJANGO DATABASE

```
$ python manage.py makemigrations
```

```
$ python manage.py migrate
```

```
$ python manage.py createsuperuser
```

IMAGR SERVER ON AWS

NOW YOU CAN DEPLOY

```
$ zappa deploy dev
```

IMAGR SERVER ON AWS

```
(imagr_env) Chase-Thompson-Baughs-MacBook-Pro:imagr_server chase$ █
```



Google Cloud



Services ▾

Resource Groups ▾



N. Virginia ▾

Support ▾

AWS Lambda

[Dashboard](#)[Functions](#)[Lambda > Functions](#)[Create a Lambda function](#)[Actions ▾](#)

Filter by tags and attributes or search by keyword



Function name	Description	Runtime	Code size	Last Modified
imagr-server-dev	Zappa Deployment	Python 2.7	18.5 MB	27 minutes ago

[Feedback](#)[English](#)

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)



Services ▾

Resource Groups ▾



N. Virginia ▾

Support ▾



Amazon API Gateway

APIs

Show all hints

**APIs**

imagr-server-dev

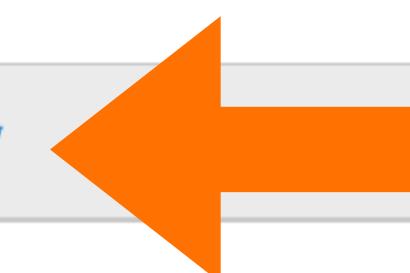
Usage Plans

API Keys

Custom Domain Names

Client Certificates

Settings

Create API**imagr-server-dev***Created automatically by Zappa.*

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)



Services ▾

Resource Groups ▾



N. Virginia ▾

Support ▾



Amazon API Gateway

APIs

> imagr-server-dev (8yvn1qv15i)

> Stages

> dev

Show all hints



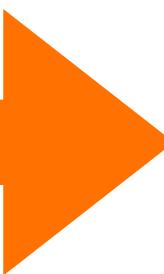
APIs

imagr-server-dev

Stages

Create

dev



Stages

Authorizers

Gateway Responses

Models

Documentation

Binary Support

dev Stage Editor

Delete Stage

Invoke URL: <https://8yvn1qv15i.execute-api.us-east-1.amazonaws.com/dev>

Settings

Stage Variables

SDK Generation

Export

Deployment History

Documentation History

Configure the metering and caching settings for the **dev** stage.

Cache Settings

Enable API cache

← → C

Secure

<https://8yvn1qv15i.execute-api.us-east-1.amazonaws.com/login/?next=/dev/>

{"message": "Forbidden"}



Services ▾

Resource Groups ▾



N. Virginia ▾

Support ▾



Amazon API Gateway

Custom Domain Names

Show all hints



APIs

imagr-server-dev

Usage Plans

API Keys

Custom Domain Names

Recent Certificates

Settings

+ Create Custom Domain Name

No Custom Domain Names exist. Click the Create Custom Domain Name button above.

To use your own domain name for an API and Stage, create a Custom Domain Name backed by an ACM certificate. Add Base Path Mappings to map your custom domain to an API and Stage. You can leave the path empty to use the root (no additional mappings will be allowed) and you can leave the Stage empty to use the root URL (e.g. www.example.com/my-base-path/MyStage). To use an ACM certificate with API Gateway, you must request or import the certificate in the (N. Virginia) region.

+ Creating Custom Domain Name...

New Custom Domain Name

Domain Name

ACM Certificate (us-east-1 only)

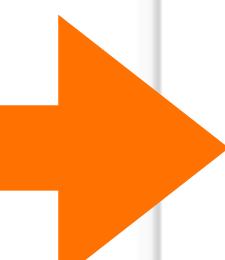
Base Path Mappings

Path	Destination
------	-------------

Add mapping

Cancel

Save





Services ▾

Resource Groups ▾



N. Virginia ▾

Support ▾

Request a certificate

Import a certificate

Actions ▾



« < Viewing 1 to 1 of 1 certificates > »

<input type="checkbox"/>	Name ▾	Domain name ▾	Additional names	Status ▾	Type ▾	In use? ▾

« < Viewing 1 to 1 of 1 certificates > »

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)[Terms of Use](#)





Services ▾

Resource Groups ▾



N. Virginia ▾

Support ▾

Request a certificate

Import a certificate

Actions ▾



« < Viewing 1 to 1 of 1 certificates > »

Name ▾ Domain name ▾ Additional names Status ▾ Type ▾ In use? ▾

« < Viewing 1 to 1 of 1 certificates > »

Feedback

English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

Import a certificate

Step 1: Import certificate

Step 2: Review and import

You can use AWS Certificate Manager certificates with other [AWS Services](#).

Select certificate



Paste the PEM-encoded certificate body, private key, and certificate chain below. [Learn more](#).

Certificate body*

Certificate private key*

Certificate chain

* Required

[Cancel](#)

[Review and import](#)



URL (e.g. www.example.com/my-base-path/MyStage). To use an ACM certificate with API Gateway, you must request or import it (N. Virginia) region.

+ Creating Custom Domain Name...

New Custom Domain Name

Domain Name

imagr-demo.simplymac.com

ACM Certificate (us-east-1 only)

*.simplymac.com (48e5075a)

Base Path Mappings

Path	Destination
/	imagrserver-de... : dev

Add mapping

Cancel Save

The screenshot shows the 'New Custom Domain Name' configuration page. It includes fields for the domain name (imagr-demo.simplymac.com), an ACM certificate (*.simplymac.com), and base path mappings (a table with one entry: '/'). There are also 'Add mapping' and 'Cancel' buttons, and a prominent 'Save' button at the bottom.

imagr-demo.simplymac.com

Uploaded on 5/22/2017



Distribution Domain Name

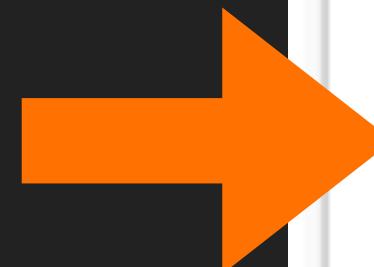
d99m3hfvv2ka7.cloudfront.net

ACM Certificate

*.simplymac.com (48e5075a)

Show Base Path Mappings

Edit





Secure

<https://imagr-demo.simplymac.com/login/?next=/>



Imagr Server

Sign in to start your session

Username



Password



Sign In



slack

SLACK SETTINGS

DJANGO SETTINGS.PY

```
# slack notifications
SLACK_NOTIFY=True
SLACK_WEBHOOK_URL='https://slackwebhook'
SLACK_CHANNEL='macops'
SLACK_BOT_NAME='imagr'
```

IMAGR SERVER ON AWS

NOW UPDATE LAMBDA

```
$ zappa update dev
```

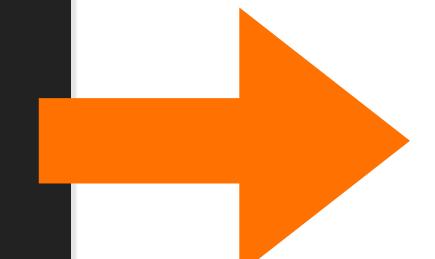
#imagr_report

★ | 2 | 0 | Add a topic

 Search

Message

Preparing to run workflow 10.12 Sierra Restore...



Serial

C02I

Status

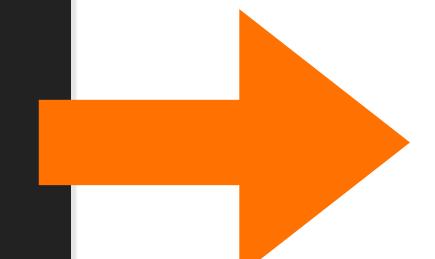
in_progress

Message

Restoring DMG:

http://

/imagr/images/osx_10.12.5_16F73.hfs.dmg



Serial

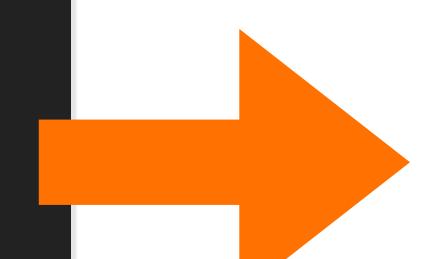
C02I

Status

in_progress

Message

Finished running 10.12 Sierra Restore.



Serial

C02I

Status

success

Message

Imagr is starting up...

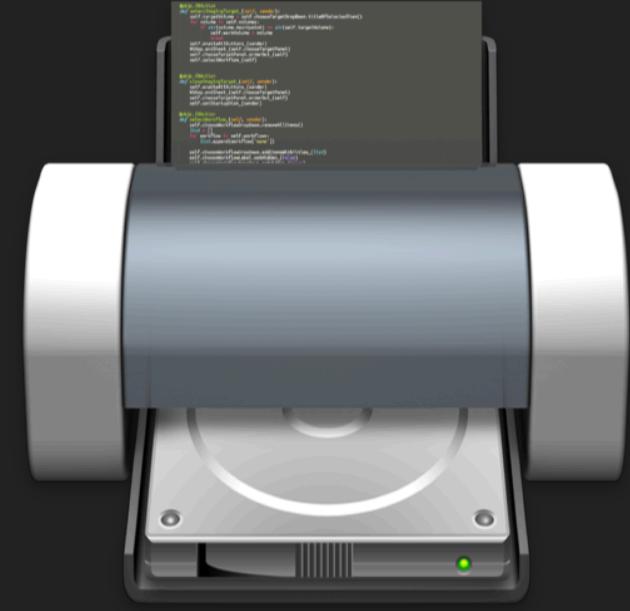


Message #imagr_report



RECAP

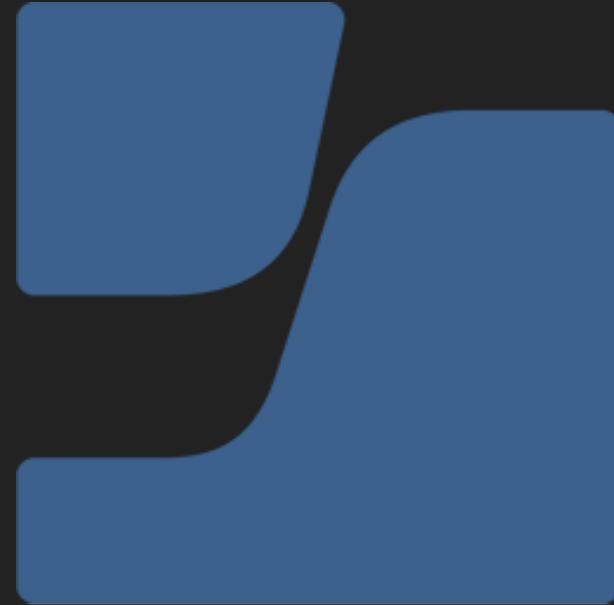
RECAP



imagr



git



jamf



server



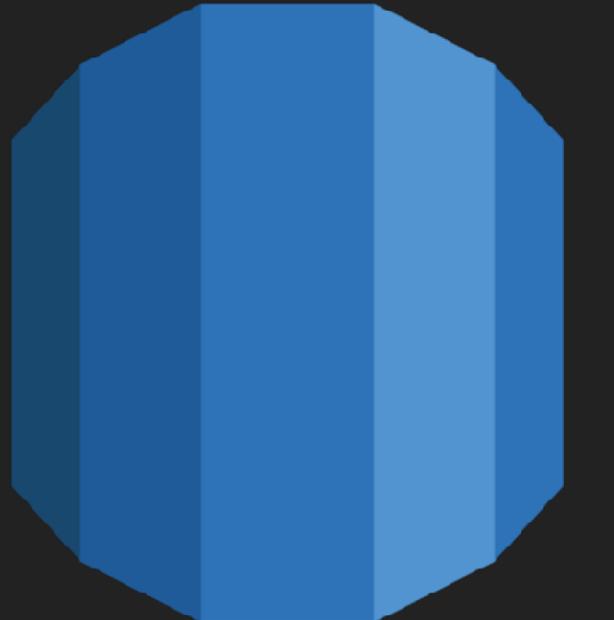
slack



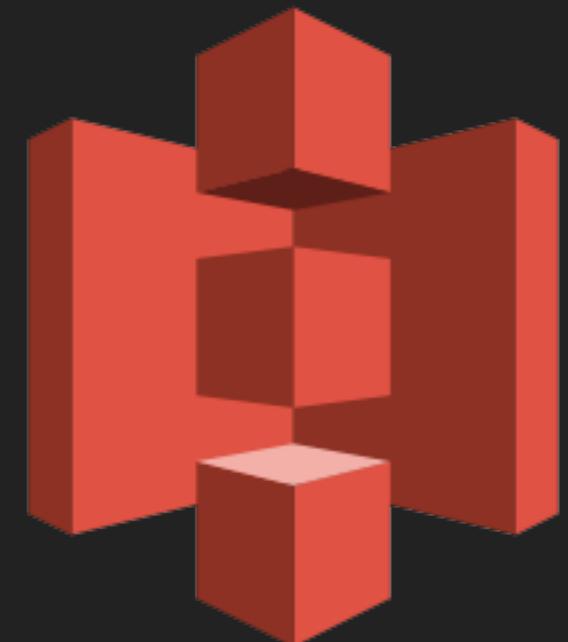
lambda



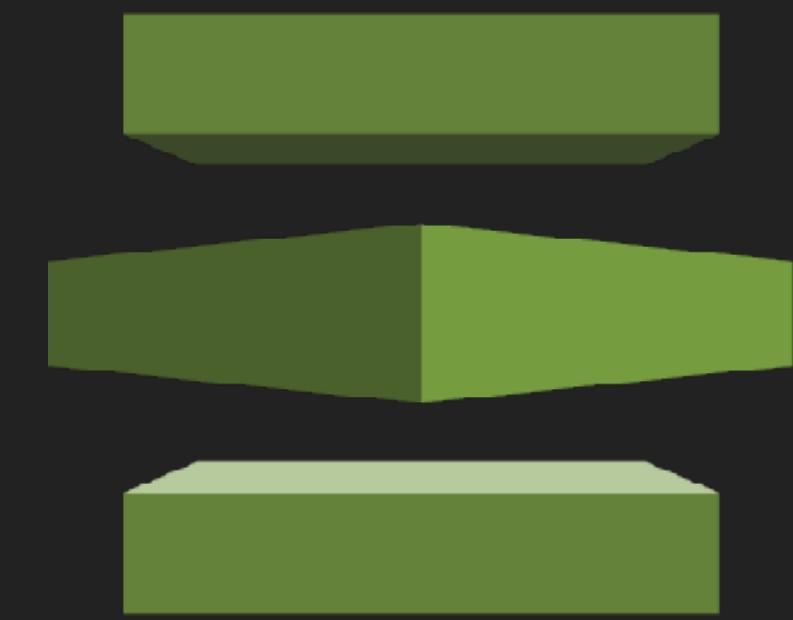
api



rds



s3



cert



IMAGING IS ONLY

MOSTLY DEAD

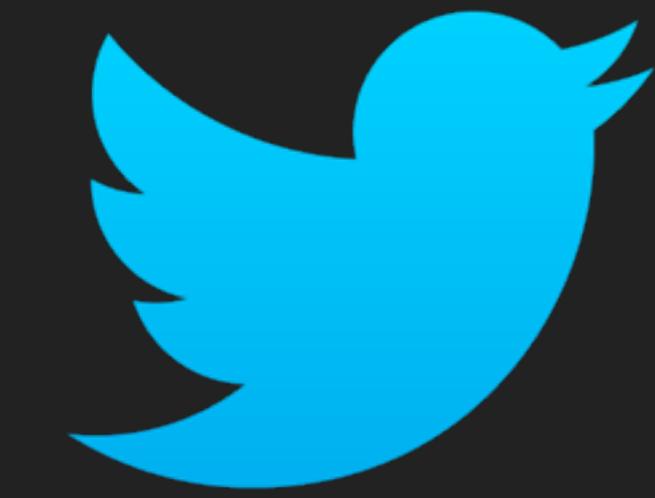
QUESTIONS?

GET THE SAMPLES

SCRIPTS AND DOCUMENTATION

[https://github.com/chasetb/
psumac-imaging](https://github.com/chasetb/psumac-imaging)

CONTACT ME



@chase_tb



@chasetb

feedback:

<https://bit.ly/psumac2017-209>