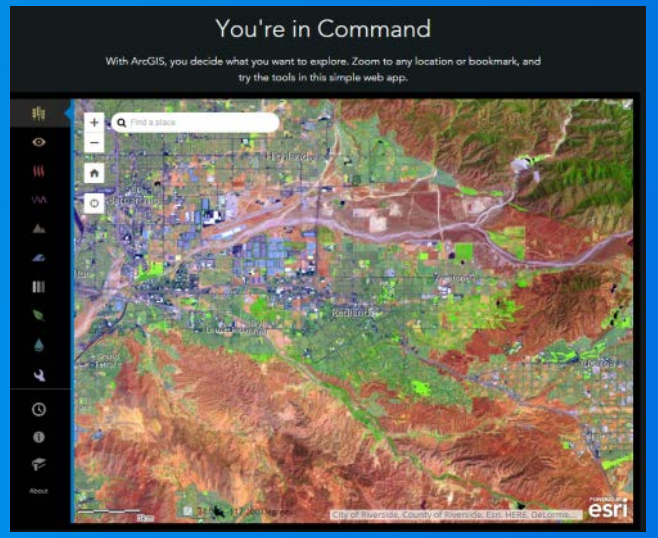
Best Practices for Sharing Imagery using Amazon Web Services

Peter Becker

Objectives

- Making Imagery Accessible
- Store massive volumes of imagery on inexpensive cloud storage
- Use elastic compute for image services

www.esri.com/landsatonaws

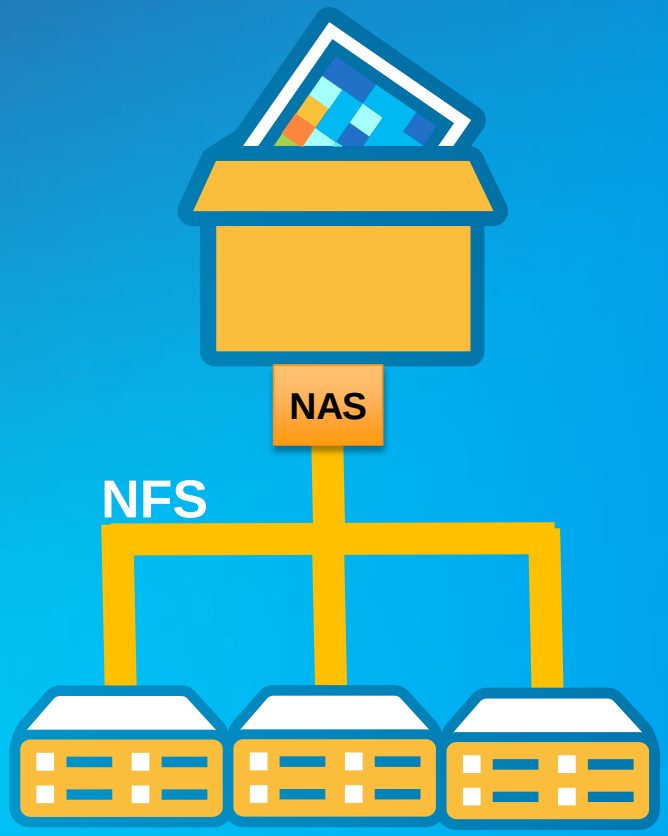


- Focus on Imagery
  - Large in Volume
  - Static
  - Basis for visualization and Analytics

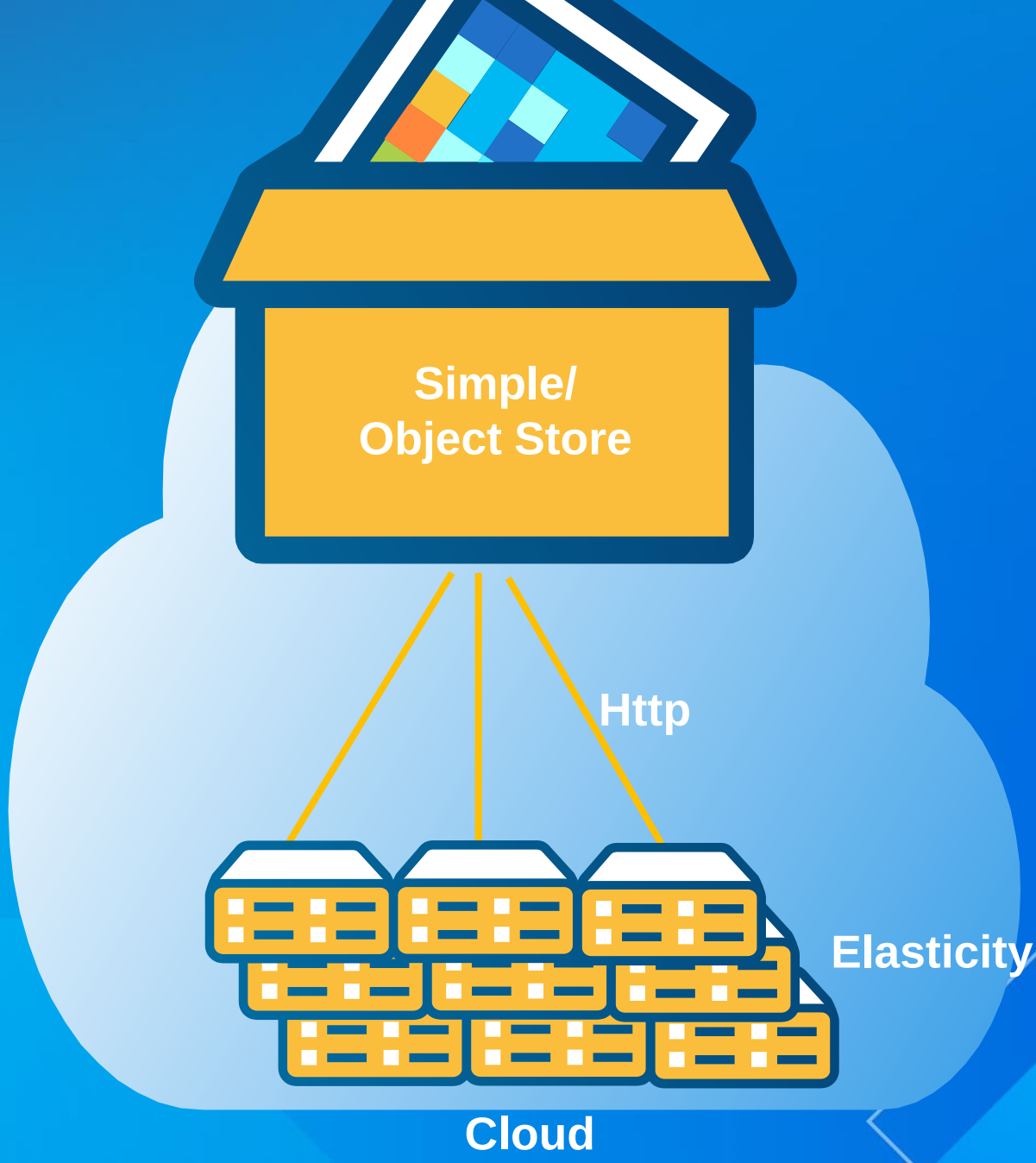
- Note:
- Implementation of these workflows currently require additional tools and a component that is not publically available. Details for access will be provided at the end

What's Changed

Standalone
Direct Access



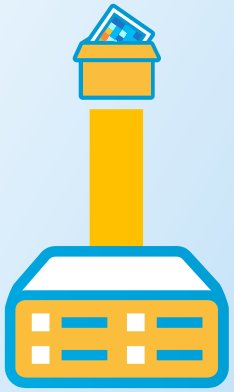
Enterprise



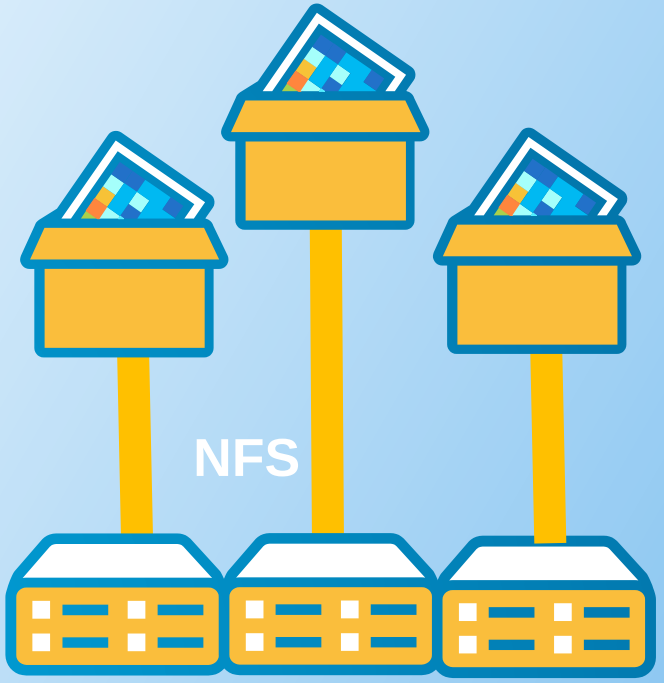
Cloud

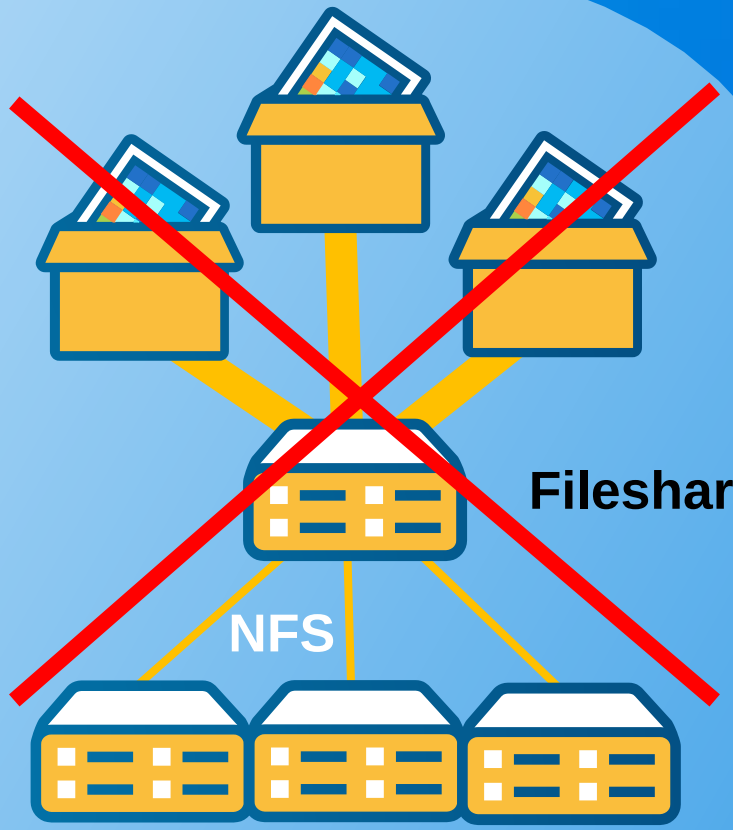
Amazon Storage Options

Ephemeral
(80GB)
$0 (inc. with EC2)



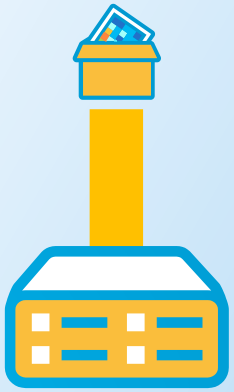
EBS (Elastic Block Storage)
1TB/Disk ($100/TB/Month)



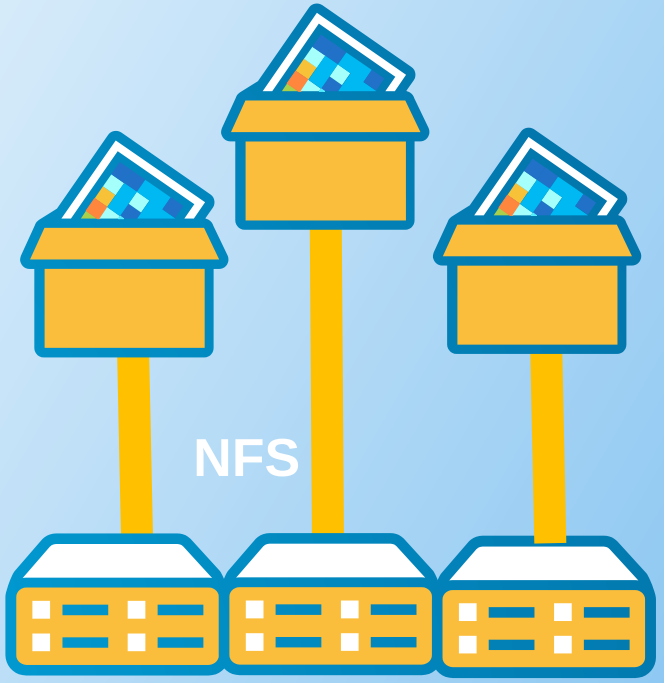


Amazon Storage Options

Ephemeral
(80GB)
$0 (inc. with EC2)



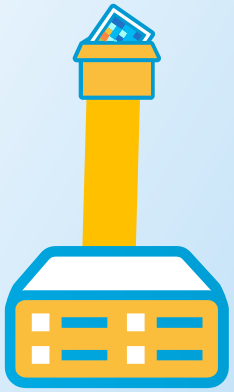
EBS (Elastic Block Storage)
1TB/Disk ($100/TB/Month)



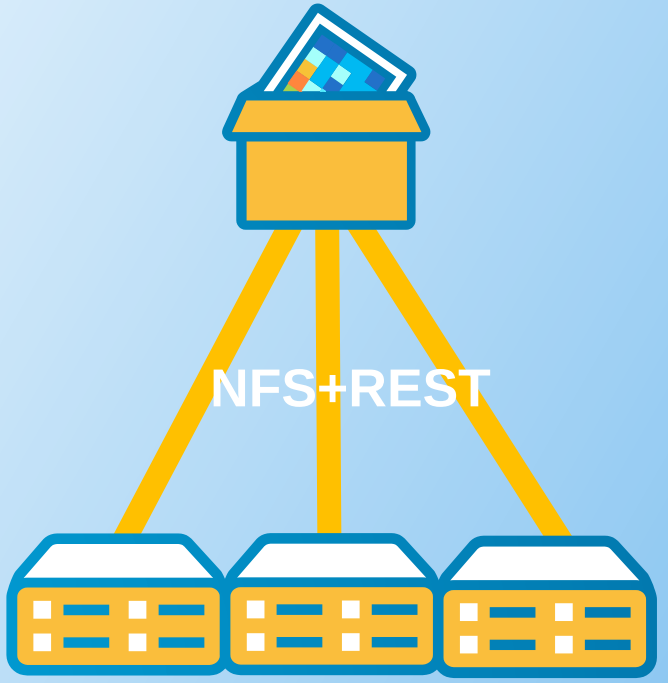
Simple Storage
$30/TB/Month
99.999999999% Durability
99.99% Reliability/year
≈10-50MB/s

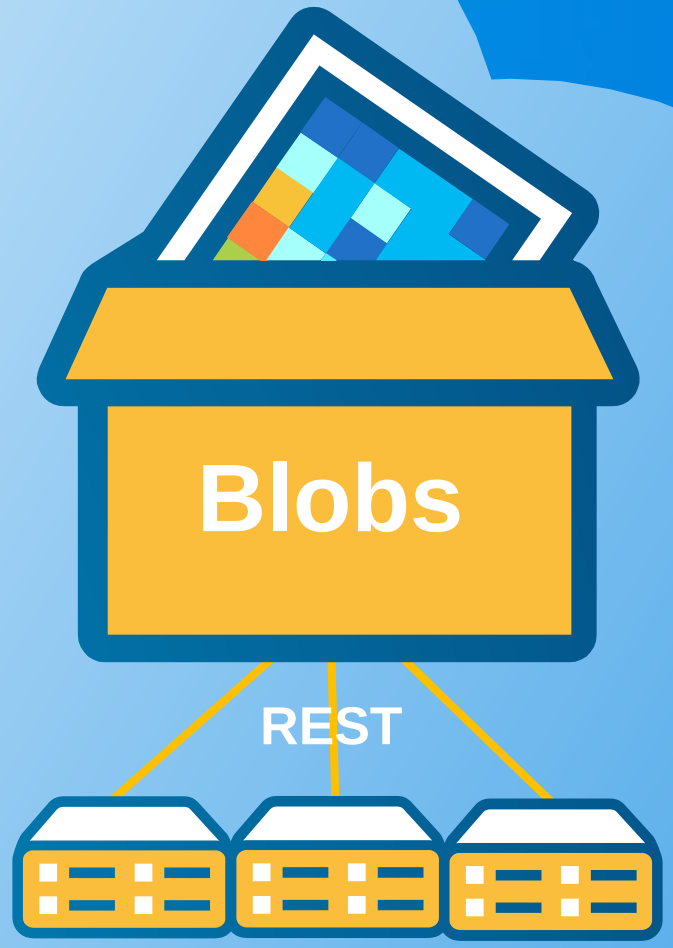
Azure Storage Options

Temporary Disk (80GB)



Azure Files
5TB, max 60MBps/Share

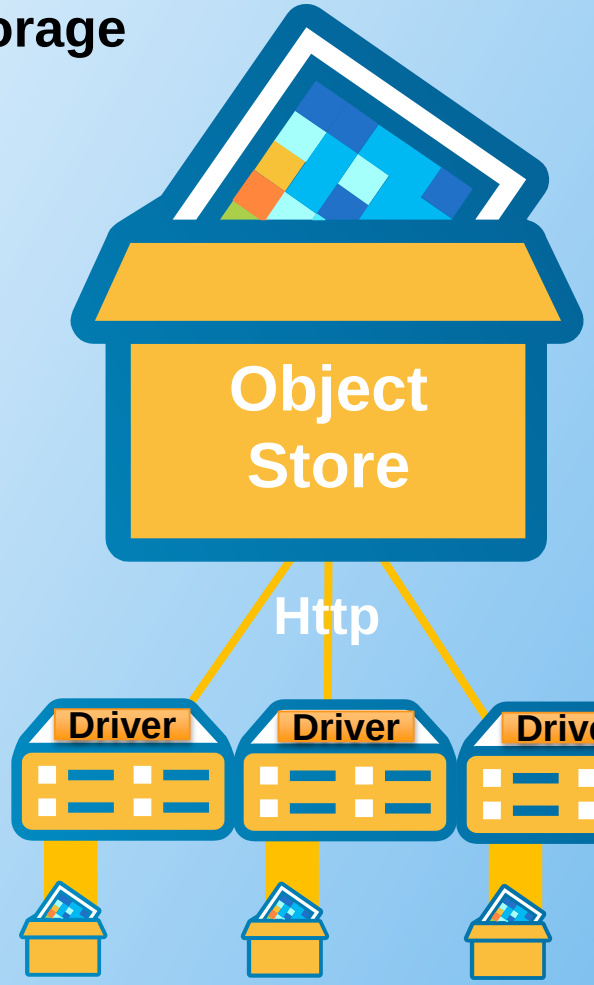




Azure Blobs
500TB/container
max 60MBps/Blob

Optimizations

- **Use Inexpensive Scalable Storage**
  - **Minimize Number of requests**
  - **Minimize Size of requests**
  - **Minimize Repeat requests**
- **Enable Elasticity**
- **Enable Caching**



Suitable data structure:
- **Enable access in single request**
  - **Metadata**
  - **Index**
  - **Data Tiles**

Enable http access

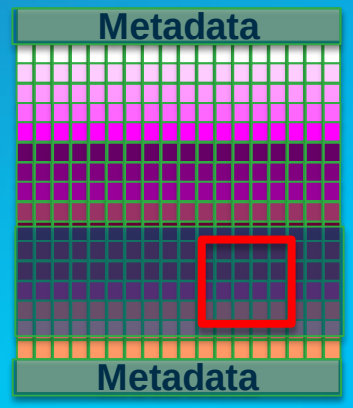
Cache to ephemeral/temporary storage

File Format Considerations / Optimization

- Tiling of imagery – Enables partial access
- Compression – Reduce storage and transfer – Weigh against additional compute
- Data access complexity – Reduce subsequent requests

Bad

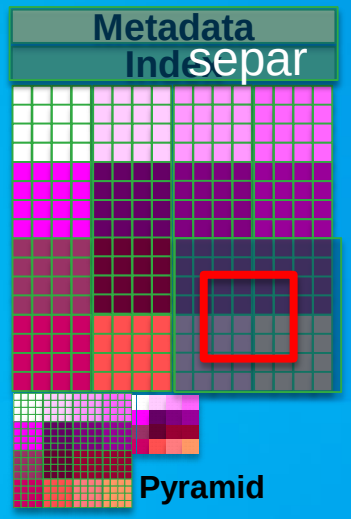
Striped TIF



Non Optimum Access

Good

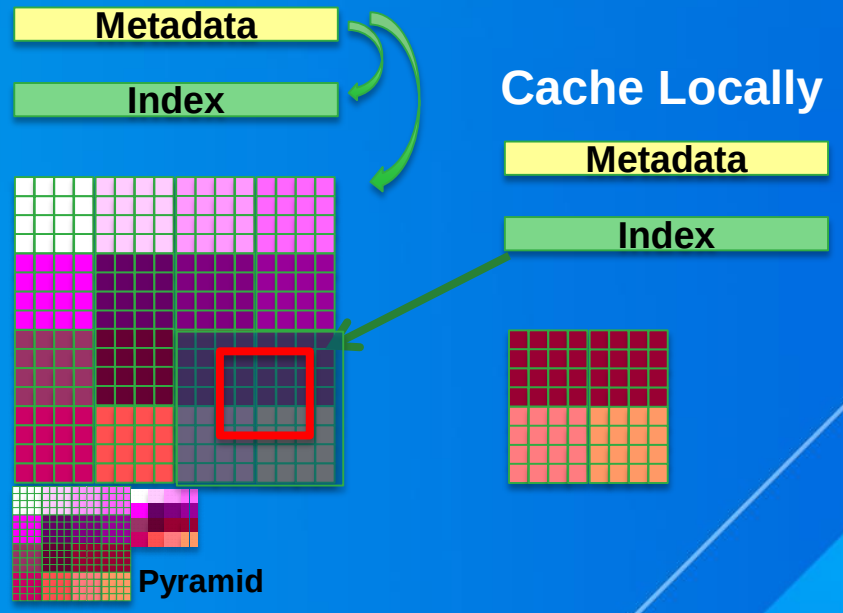
Tiled TIF



Enables partial Access

Optimum

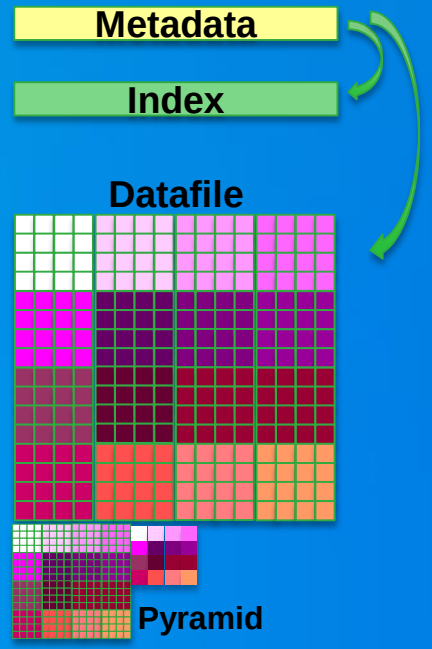
MRF



Enable separate caching of metadata and index

MRF (Meta Raster) Format

- NASA designed format
- Split raster into three (or more) files
  - MRF – Includes links to Index and DataFile
  - Index – Simple index to tiles
  - DataFile – Data file
  - Auxiliary files – e.g. Aux.xml
- Open Source Implemented in GDAL
- Multiple compression options
- Multiple modes: Static, Split, Cloned, Caching



- MRF file is seen as a Raster Dataset in ArcGIS
- MRF file and index is small and can easily be on a local file system
- Data can be on slower storage

Storage Formats & Compression

Advantages/Disadvantages

- **Raw/Striped TIF** — **Sequential access / No compression**
- **Tiled GeoTIF** — **Tiled Access**
  - **Compression Options**
    - **None**
    - **Lossy JPEG……Fast Lossy compression 8 & 12bit**
    - **Lossless Deflate/LZW…Generic lossless compression**
- **JPEG2000** — **Tiled Access**
  - **Compression Options**
    - **Lossy……Computationally expensive**
    - **Lossless……Computationally expensive**
- **MRF** — **Tiled Access**
  - **Compression Options**
    - **As geoTIF – JPEG,Deflate**
    - **PNG – Generic lossless**
    - **LERC – Controlled Lossy – Very Fast – Optimum for higher bit depth and catagorical**

Converting Data

- ArcGIS CopyRasters, GDAL_Translate/GDALaddo
  - Convert rasters, but don’t copy associated metadata
- OptimizeRasters
  - Python Script (Calls ArcGIS Version of GDAL)
  - From any GDAL supported format to MRF or TiledTIF
  - Parameters to define compression, tilesize etc
  - Currently command line tool

Getting Imagery to and from the Cloud

- AWS command line
- CloudBerry, FileZilla, ….
- OptimizeRasters
  - Copy and conversion in parallel
  - Option to creates Cloned MRF, Caching MRF
  - Check success and retries if necessary
- Import/Export
  - Ship hard drives for large projects

- Note: File Names on S3 are case sensitive

Demo

Conversion and Uploading Using OptimizeRasters

Accessing Cloud Imagery

ArcGIS Prerequisites

- ArcGIS typically access files through a file system
- Cannot access files in S3 directly

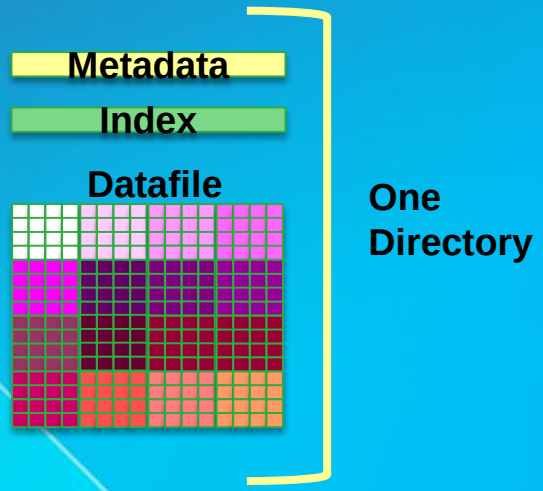
- Use MRF Driver to point to MRF file that points to data on S3
- MRF Driver is custom GDAL Driver for ArcGIS
- Available in 10.3.1* (with issues)
- Best to download – See later

- Note: Following can be used to Optimize Enterprise implementations as well
- Works with data on S3, but also from any local storage

ArcGIS MRF Driver

- Special MRF driver for ArcGIS
- Incorporated into ArcGIS 10.3.1* Desktop & Server
- 4 Modes

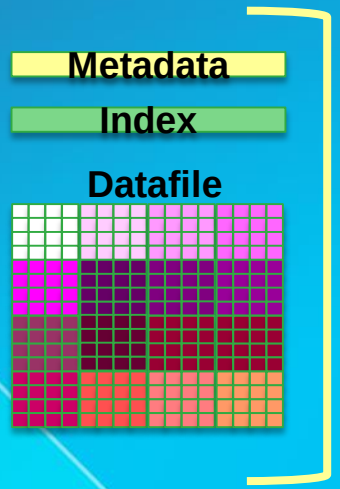
  StaticMRF



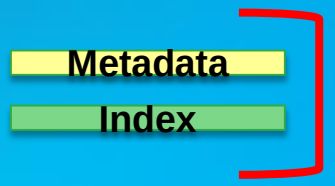
ArcGIS MRF Driver

- Special MRF driver for ArcGIS
- Incorporated into ArcGIS 10.3.1* Desktop & Server
- 4 Modes

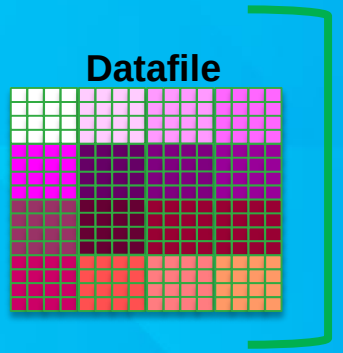
StaticMRF



SplitMRF



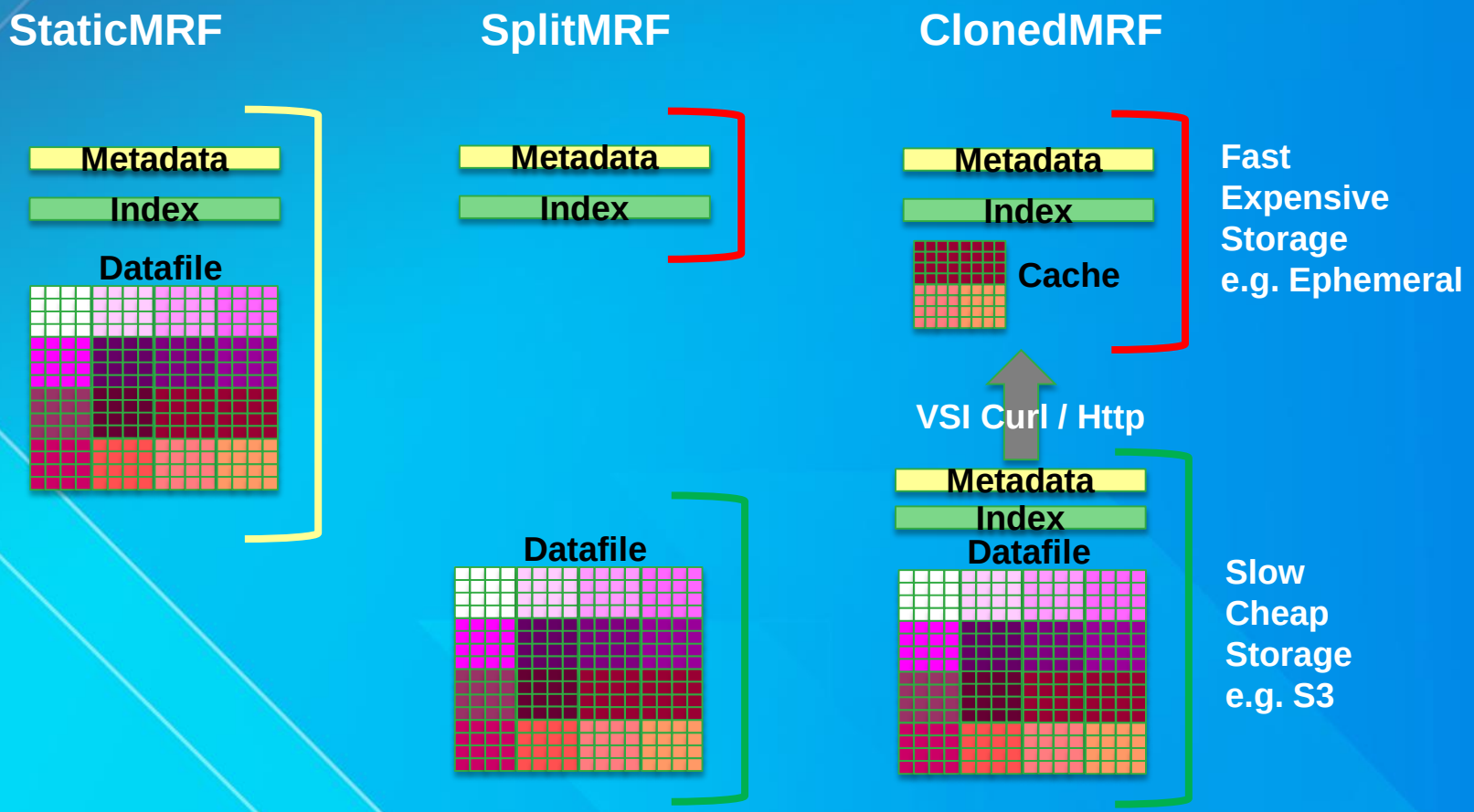
Fast
Expensive
Storage



Slow
Cheap
Storage

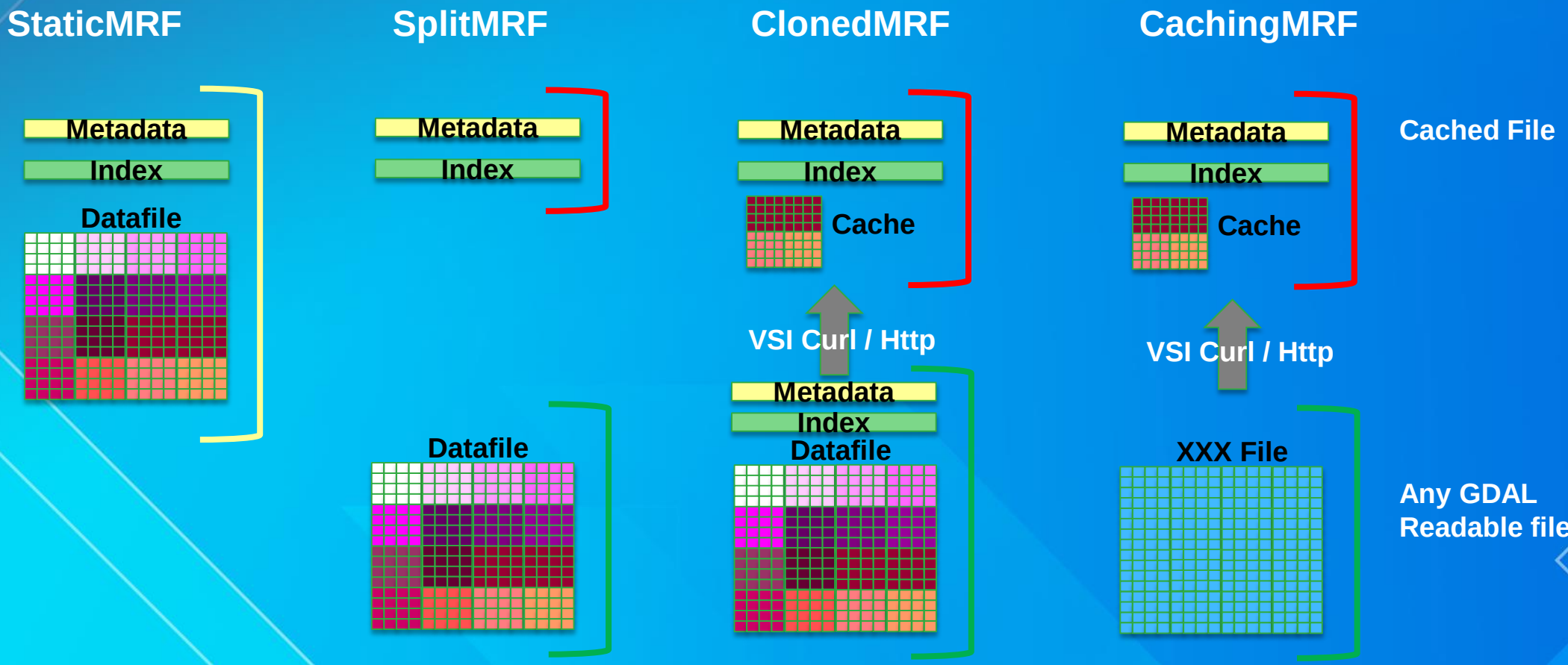
ArcGIS MRF Driver

- Special MRF driver for ArcGIS
- Incorporated into ArcGIS 10.3.1* Desktop & Server
- 4 Modes



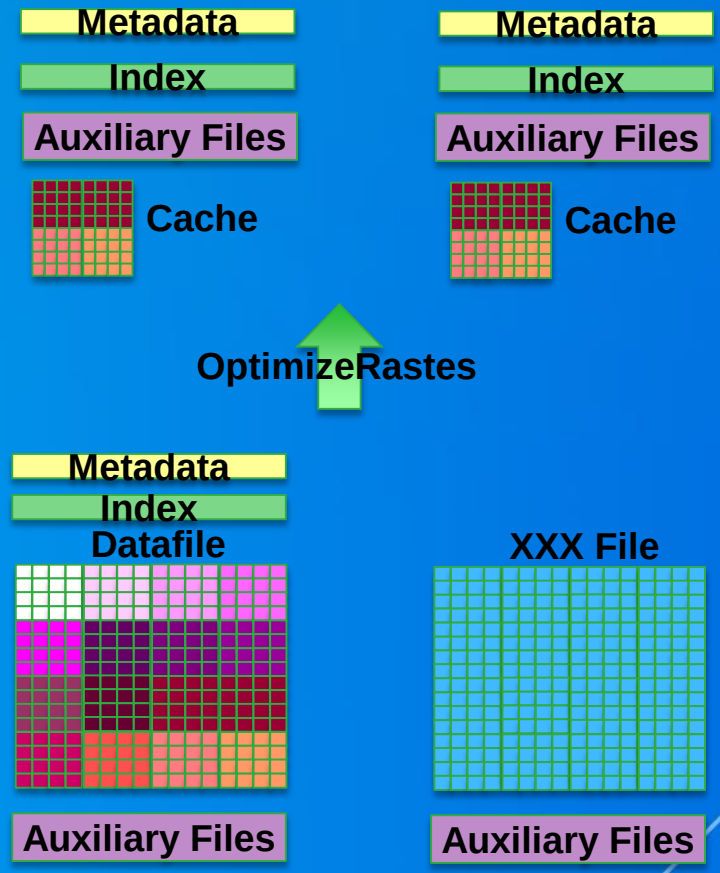
ArcGIS MRF Driver

- Special MRF driver for ArcGIS
- Incorporated into ArcGIS 10.3.1* Desktop & Server
- 4 Modes



Creating Local MRF Files that Point to Source

- Use OptimizeRasters
  - Creates MRF files on local file system (Ephemeral or EBS)
  - MRF points to Index and Data Files
  - Copies Auxiliary files for RasterType
  - Note MRF files can have any extension (incl. .tif)
- If source is MRF – Use ClonedMRF
  - Adds caching optimization
- If source is other format – Use CachingMRF
  - Enables source to be any Image on S3 (or SAN etc)
- Define Cache location as Ephemeral



Demo

Creating and Using CachingMRF

Creating and Using CachingMRF

Creating Mosaic Dataset

- Use Standard Workflow
  - Create Mosaic Dataset
  - Add Rasters
  - Define and Build Overviews
  - Set Properties
  - Perform QC
  - Publish

- More Advance Workflow
  - Create Source Mosaic Dataset for each project
    - As for Standard Workflow
  - Create Derived Mosaic Dataset

- For more details see: Image Management Workflows
  - Resource Center landing page http://esriurl.com/6005
  - Guidebook in Help System http://esriurl.com/6007
  - ArcGIS Online Group http://esriurl.com/6539

Handling Overviews

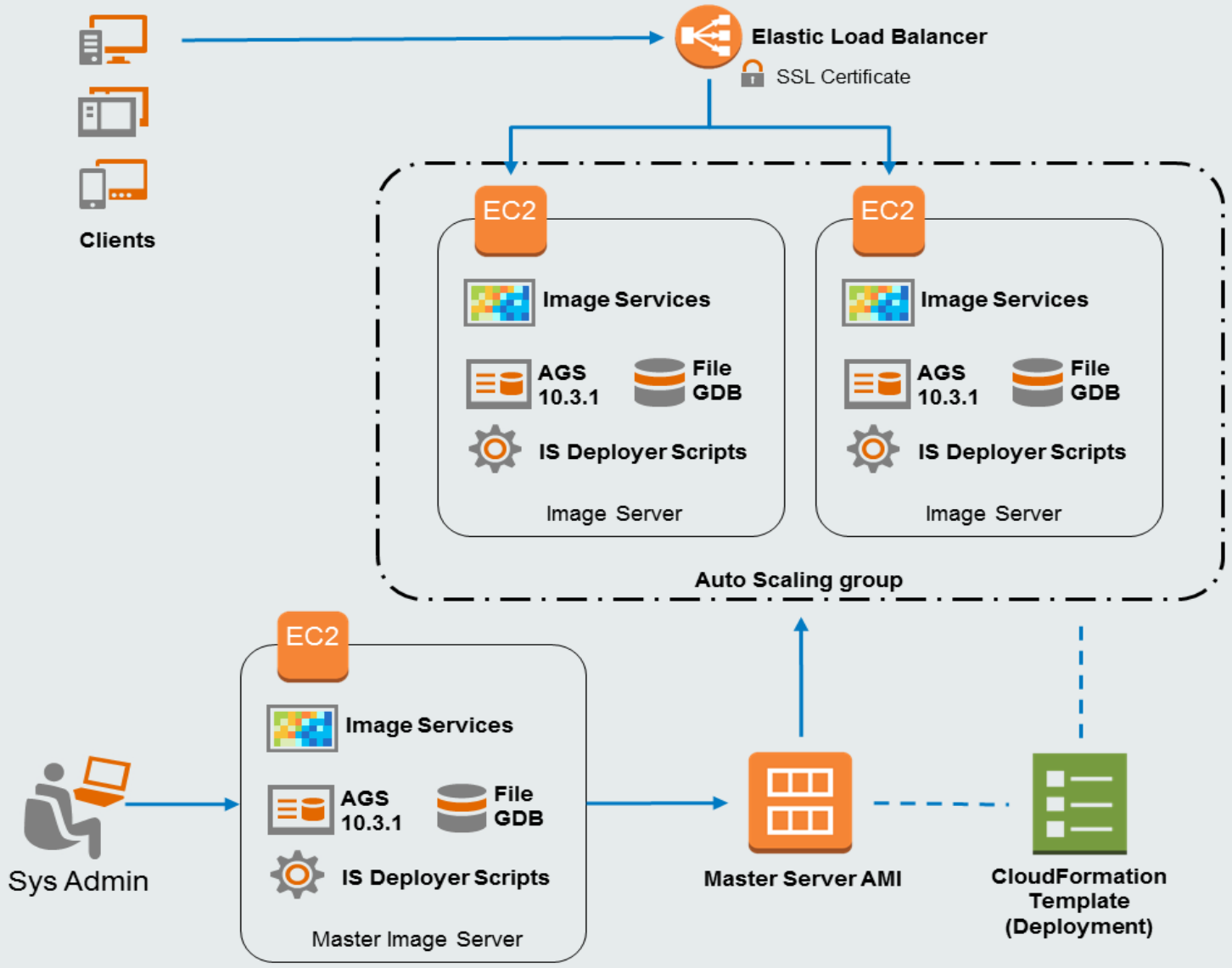
- Create Overviews on Ephemeral or EBS
- Use OptimizeRasters to copy to S3
- Delete Overviews
- Use Optimize Rasters to copy back MRF files
- Repair paths (if necessary)

- (If very large can generate on EBS and copy MRF back to Ephemeral)

Publishing Image Services

- For initial testing use Standard Workflow
  - Right click Publish as Image Service
- Publishing can be automated

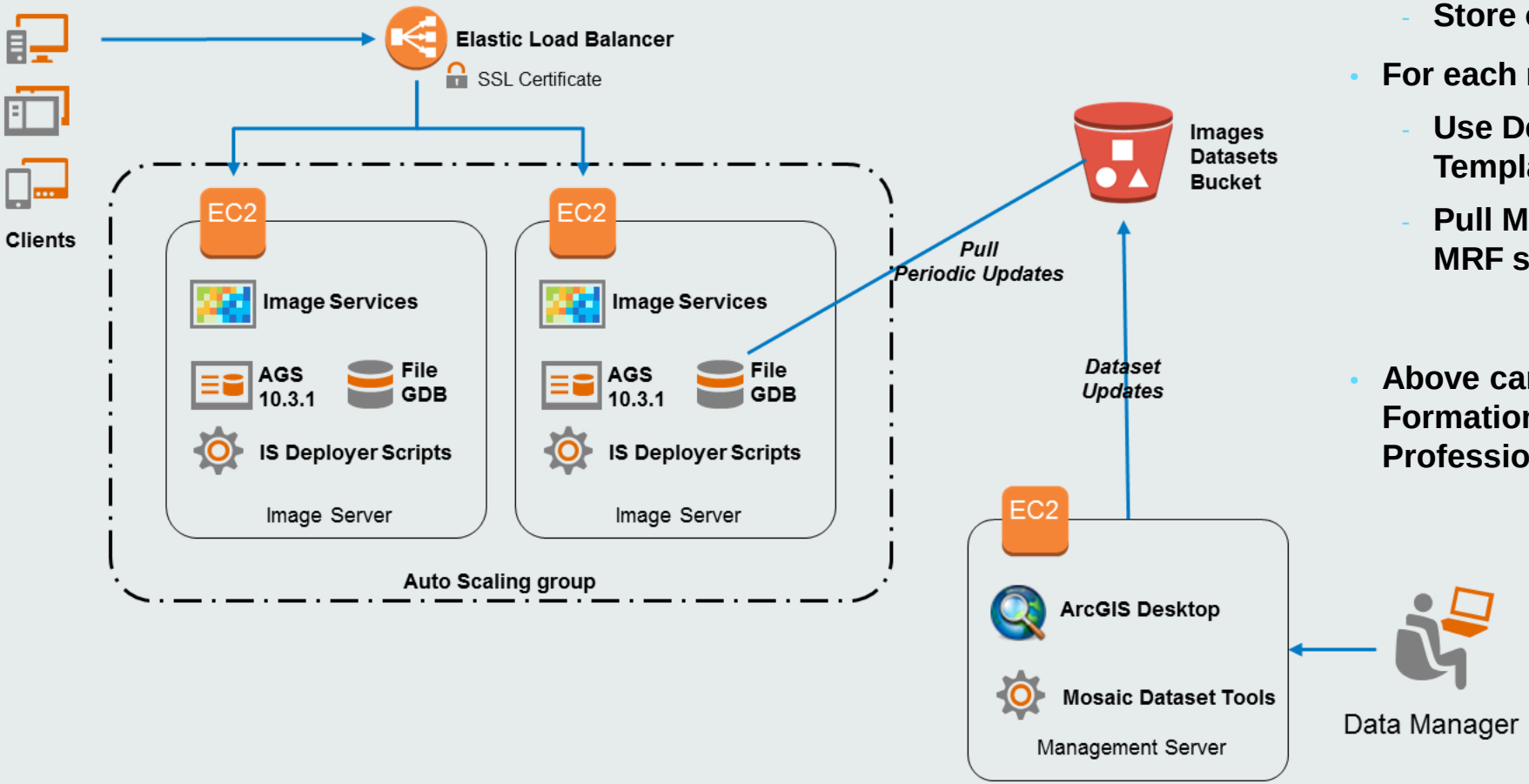
Configuring ArcGIS Server for Imagery in AWS



- Use AutoScaling Group
- Use Silo configuration of ArcGIS Server (each server is its own site)
- Load Balancer – User Regular Amazon Elastic Load Balancer (ELB)
  - Turn On Sticky session on using Duration-Based Session Stickiness
  - Listen on ports 80 (http) & 443 (https)
  - Install SSL certification on port 443
  - Forward Port 80 (ELB) -> 6080 (ArcGIS Server)
  - Forward Port 443 (EBL) -> 6443 (ArcGIS Server)
- On ArcGIS server turn on http & https

- Create Master Server
- Create AMI of Master Server
- Config AMI to Autoscaling group
- Scale out and scale in based on CPU Load (typically CPU Load >80% or <20% for 5min)
- Set Min and Max Instances

Publish Elastic Image Services



- Prepare Base
  - Delete all Cache files
  - ZIP MD and MRF files
  - Store on S3
- For each machine
  - Use Deployer Script / Cloud Formation Template to Start up
  - Pull Mosaic Datasets, MRF overviews, MRF source rasters from S3 in a zip file.

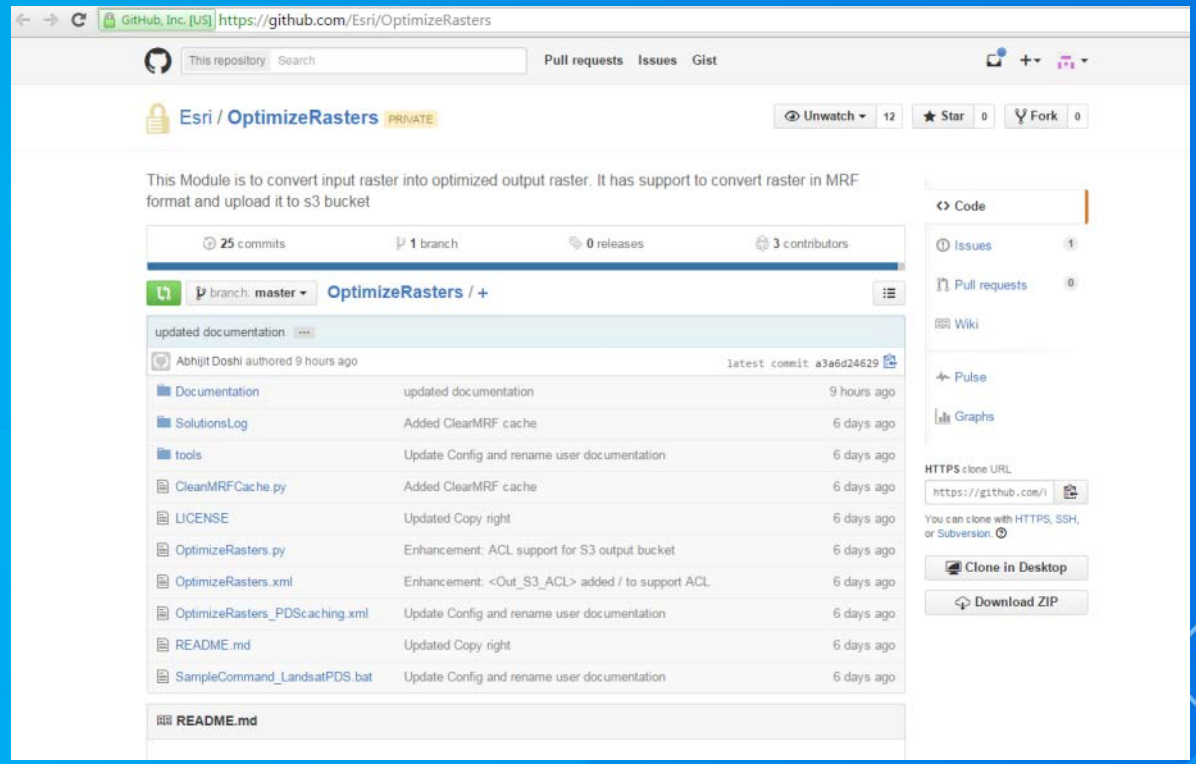
- Above can be automated with Cloud Formation templates available from Professional Services

Advanced Techniques

- S3 Security
  - Use ACL to define Public Read
  - Else use Bucket Policy to define which IPs etc. can read
- Embed MRF files into Mosaic Dataset
  - MRF files fully encapsulated in Mosaic Dataset
- PostgreSQL
  - Multiple machines from same Mosaic Dataset (dynamic updates)

Getting Access

- Currently the scripts are in Prototype state
- To get access to OptimzeRasters, MRF Drivers, PPT and doc:
  - Send email to - **ImageManagementWorkflows@esri.com**
    - Provide GitHub ID
    - Name & email
    - Brief explanation of project goals

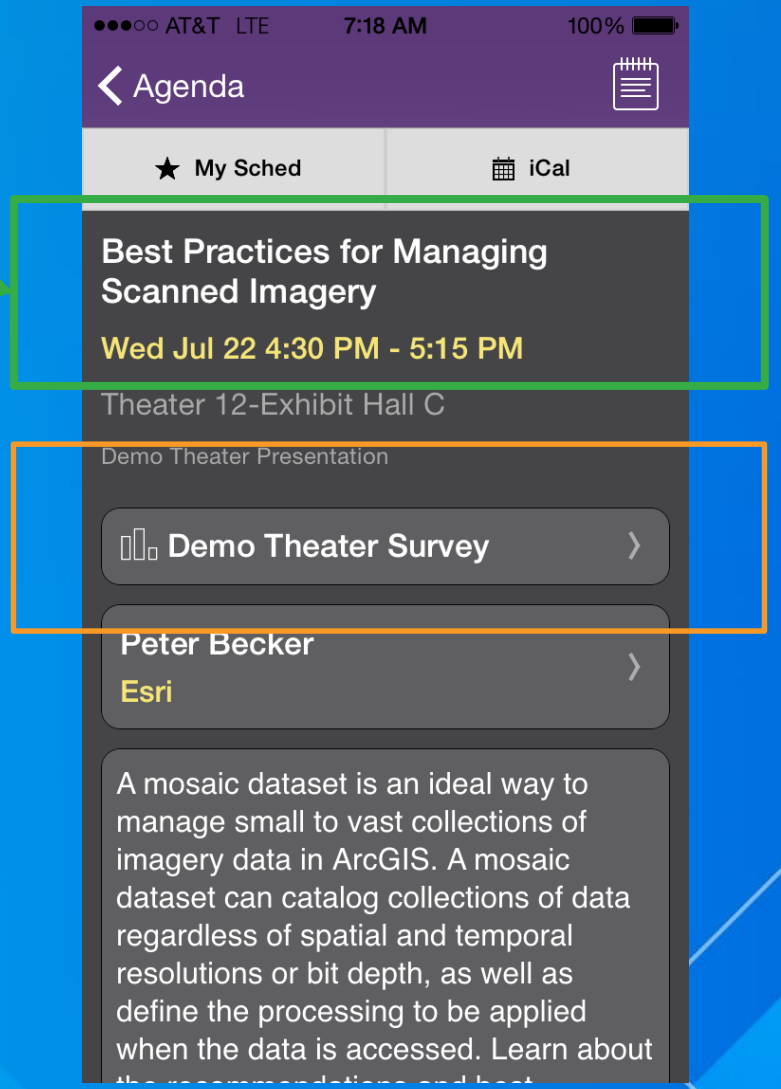


Thank you…

- Please fill out the session survey in your mobile app
- Select
- [Best Practices for Sharing Imagery using Amazon Web Services]

  in the Mobile App
  - Use the Search Feature to quickly find this title
- Click “Technical Workshop Survey”
- Answer a few short questions and enter any comments

Note: Incorrect time



Understanding our world.