

Managing Imagery In The Cloud

Peter Becker

Objectives

- Making Imagery Accessible
 - Store massive volumes of imagery on inexpensive cloud storage
 - Use elastic compute for ArcGIS image services
- Focus on Imagery
 - Large in Volume
 - Static
 - Basis for visualization and analytics

Running ArcGIS Image Server in the Cloud

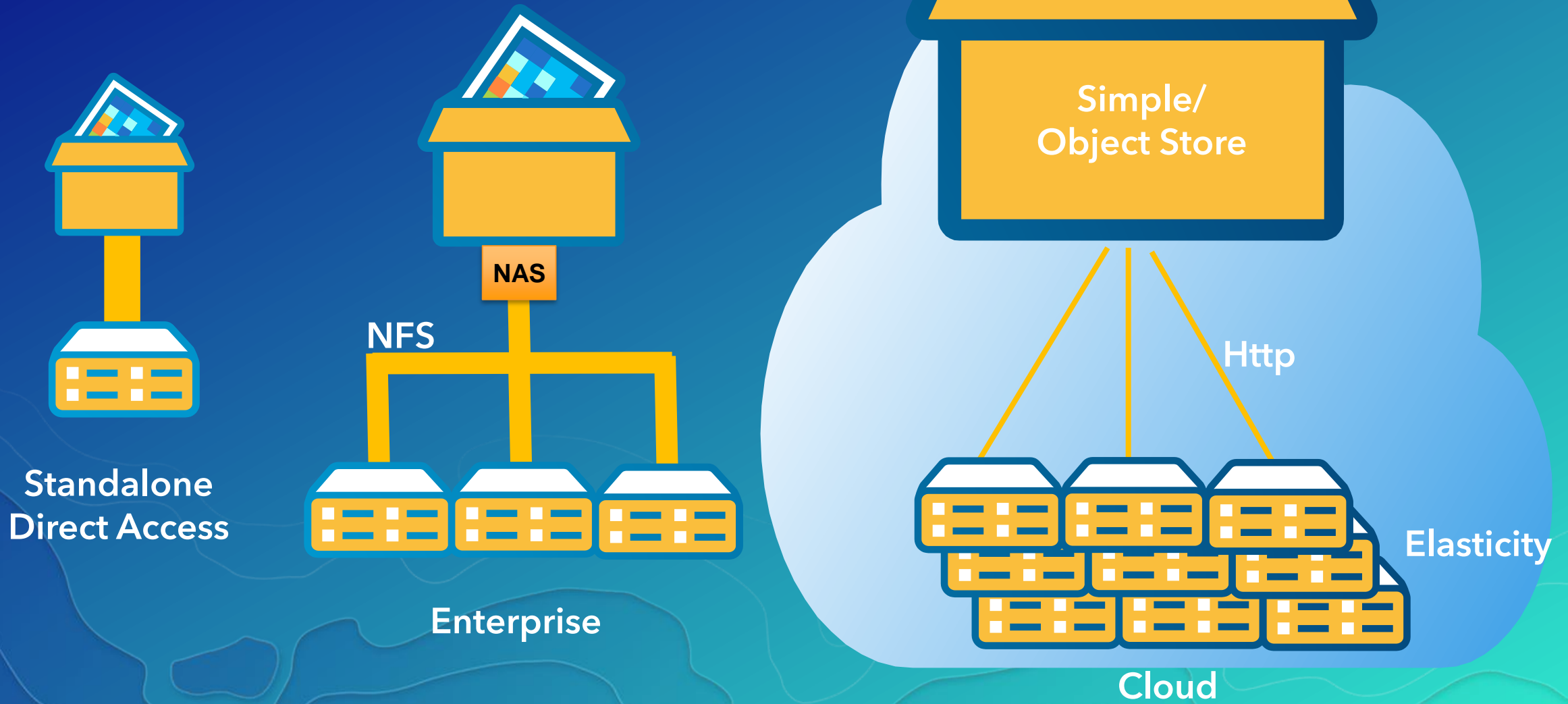
- **Advantages**

- Lower cost resilient storage
- Lower cost enterprise compute
- Simpler install
- Simpler scalability

- **Disadvantages**

- Data needs to be uploaded
- Different storage types
- Infrastructure changes
- Potential security concerns
- Potential for complex data access policies

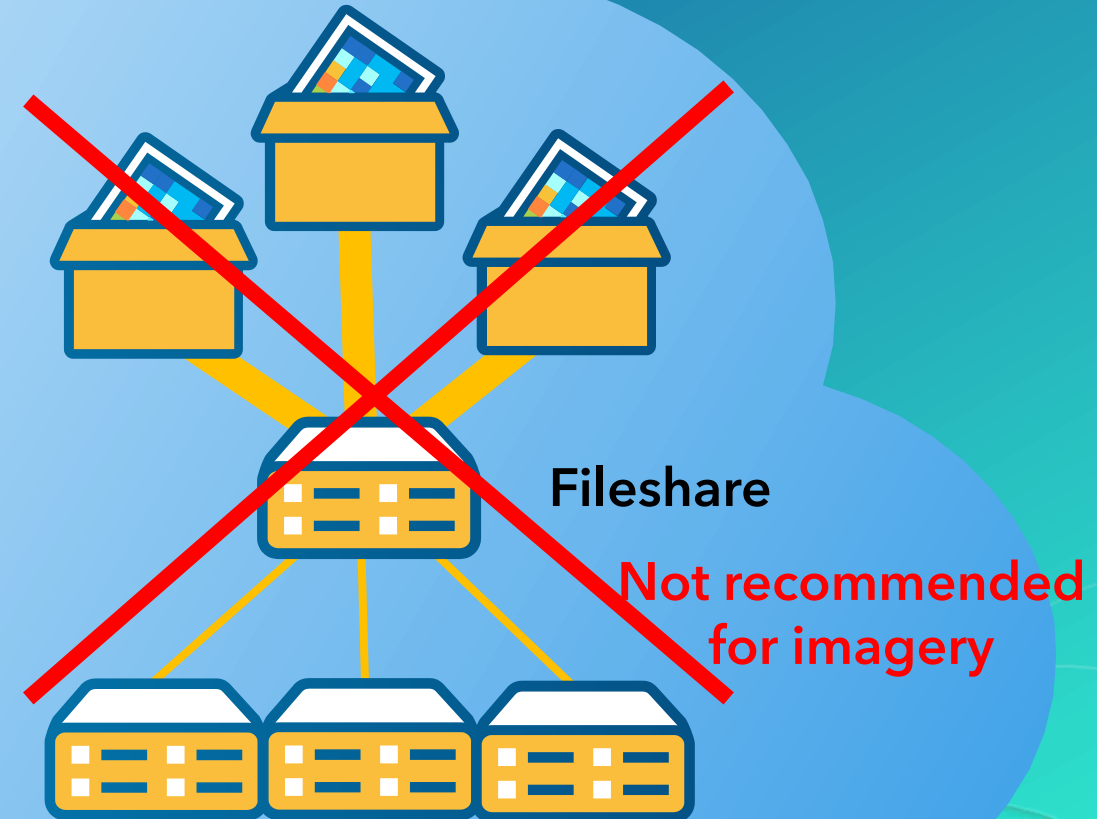
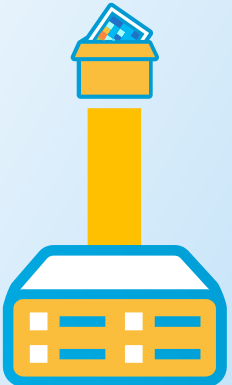
What's Changed



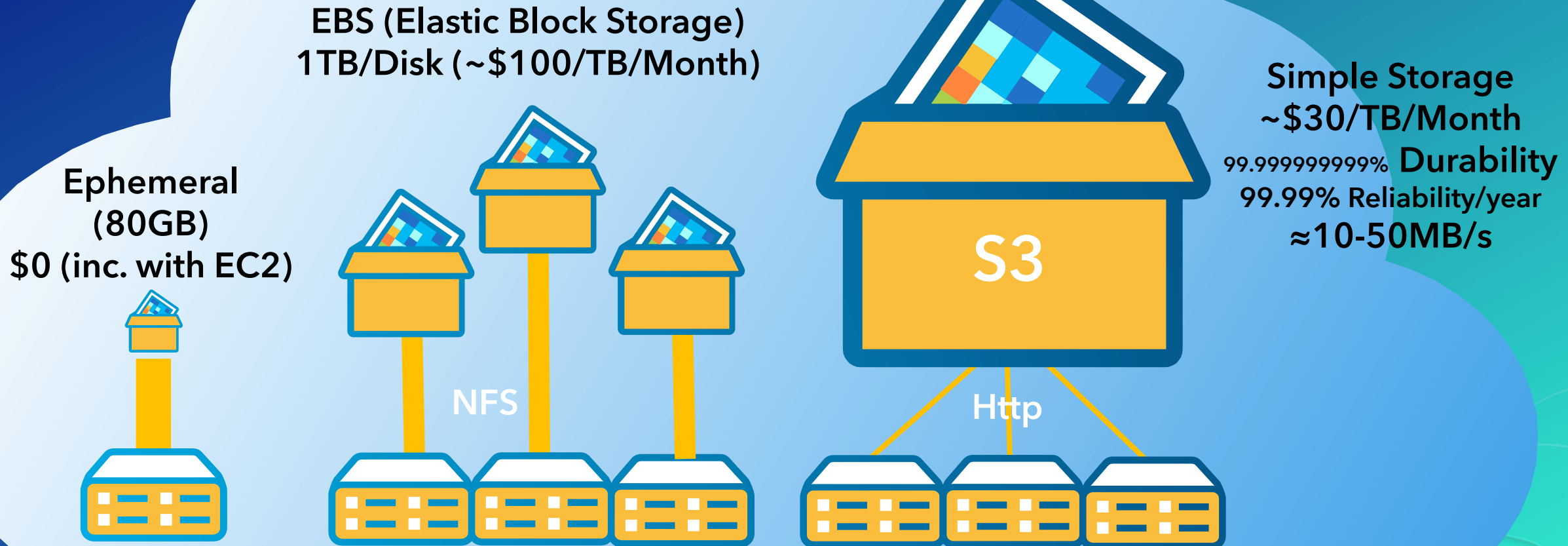
Amazon Storage Options

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

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

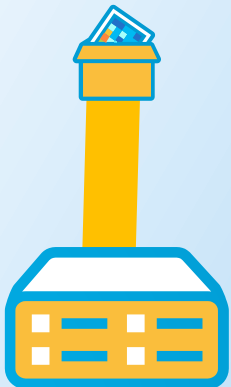


Amazon Storage Options



Azure Storage Options

Temporary
Disk
(80GB)



Azure Files
5TB, max 60MBps/Share



NFS+REST



Azure Blobs
500TB/container
max 60MBps/Blob

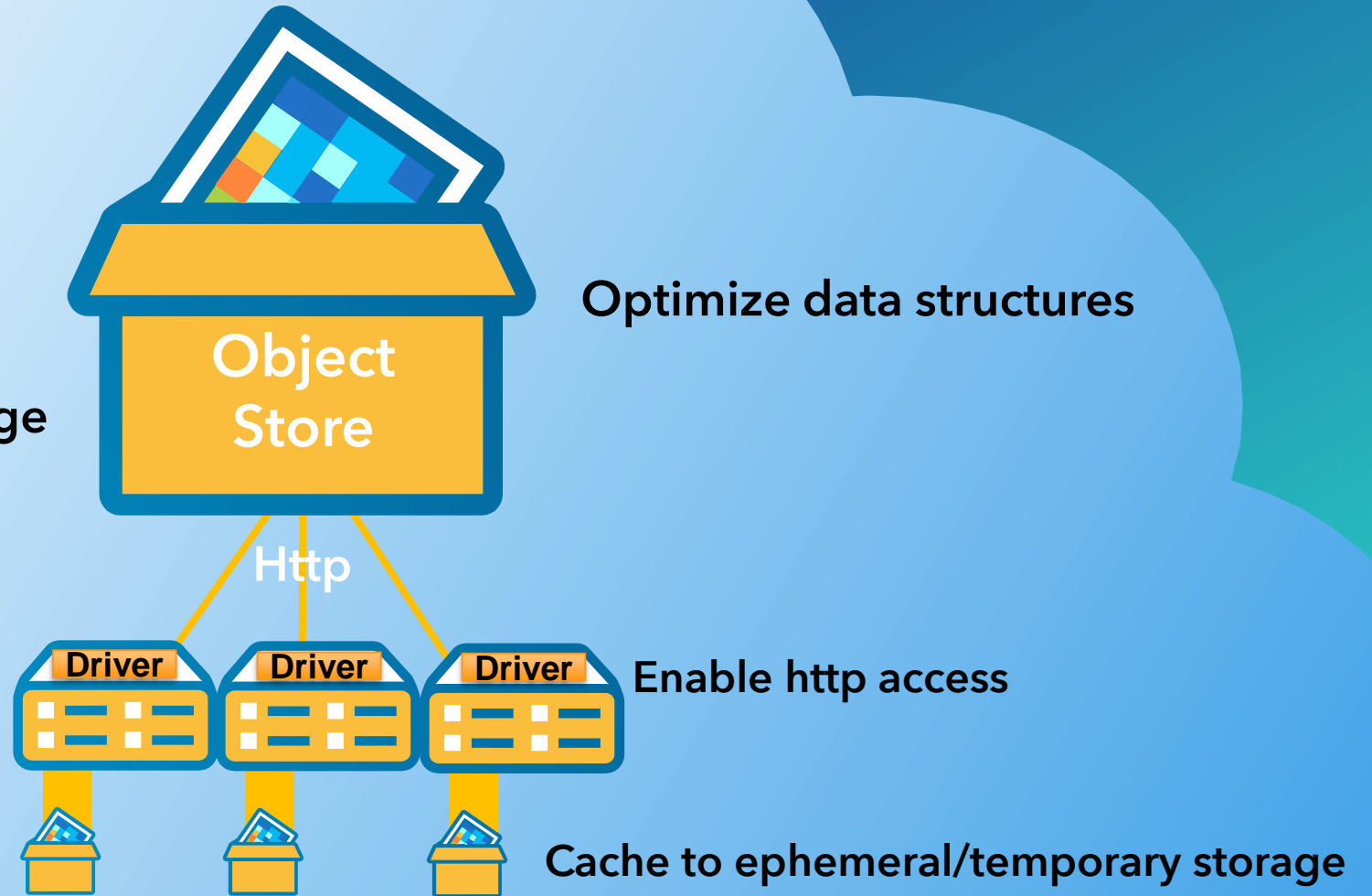


REST



Optimizations

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



Raster Storage Formats & Compression

- **Raw/Striped TIF**

- Disadvantage : Sequential access
- Compression:
 - None

- **JPEG2000**

- Advantage : Tiled Access, Higher Compression
- Disadvantage : Computationally expensive
- Compression
 - Lossy
 - Lossless

- **Tiled GeoTIF**

- Advantage: Tiled Access
- Compression
 - None
 - Lossy JPEG (8bit & 12bit)
 - Lossless Deflate LZW...Generic lossless

- **MRF**

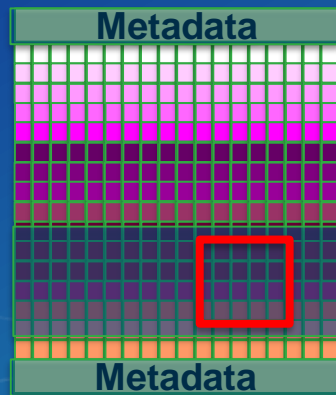
- Advantage : Tiled Access
- Compression
 - Same As geoTIF - JPEG,Deflate
 - PNG - Generic lossless
 - LERC - Controlled Lossy - Very Fast - Optimum for higher bit depth and categorical

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
- Pyramids

Bad

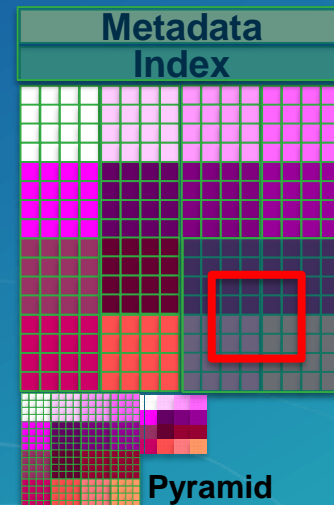
Striped TIF



Non optimum
access

Good

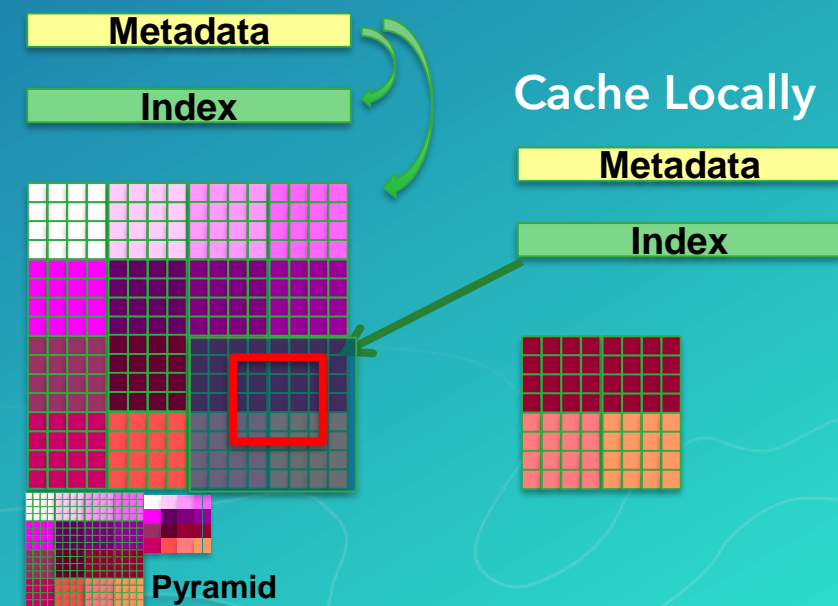
Tiled TIF



Enables partial
access

Optimum

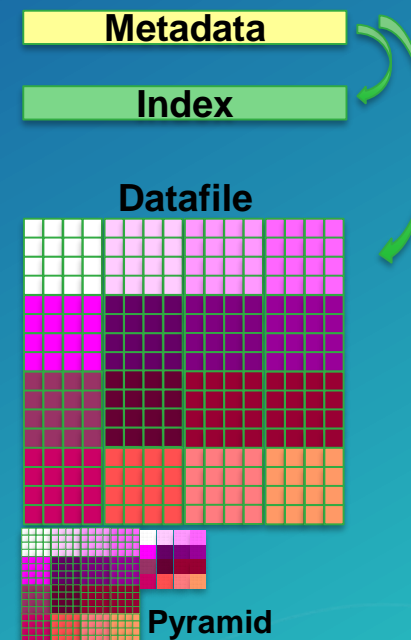
MRF



Enable separate caching of
metadata and index

MRF (Meta Raster) Format

- NASA designed format
- Split raster into three (or more) files
 - MRF - Small XML, with links to Index and DataFile
 - Index - Simple index to tiles
 - DataFile - Data file
 - Auxiliary files - e.g. Aux.xml
- Open Source Implemented in GDAL 2.1+
- Multiple compression options
- Supported for read and write in ArcGIS
- Multiple implementation modes
 - Metadata, Index and data file can be on different volumes
 - MRF file and index is small and can easily be on a local file system
 - Data can be on slower storage

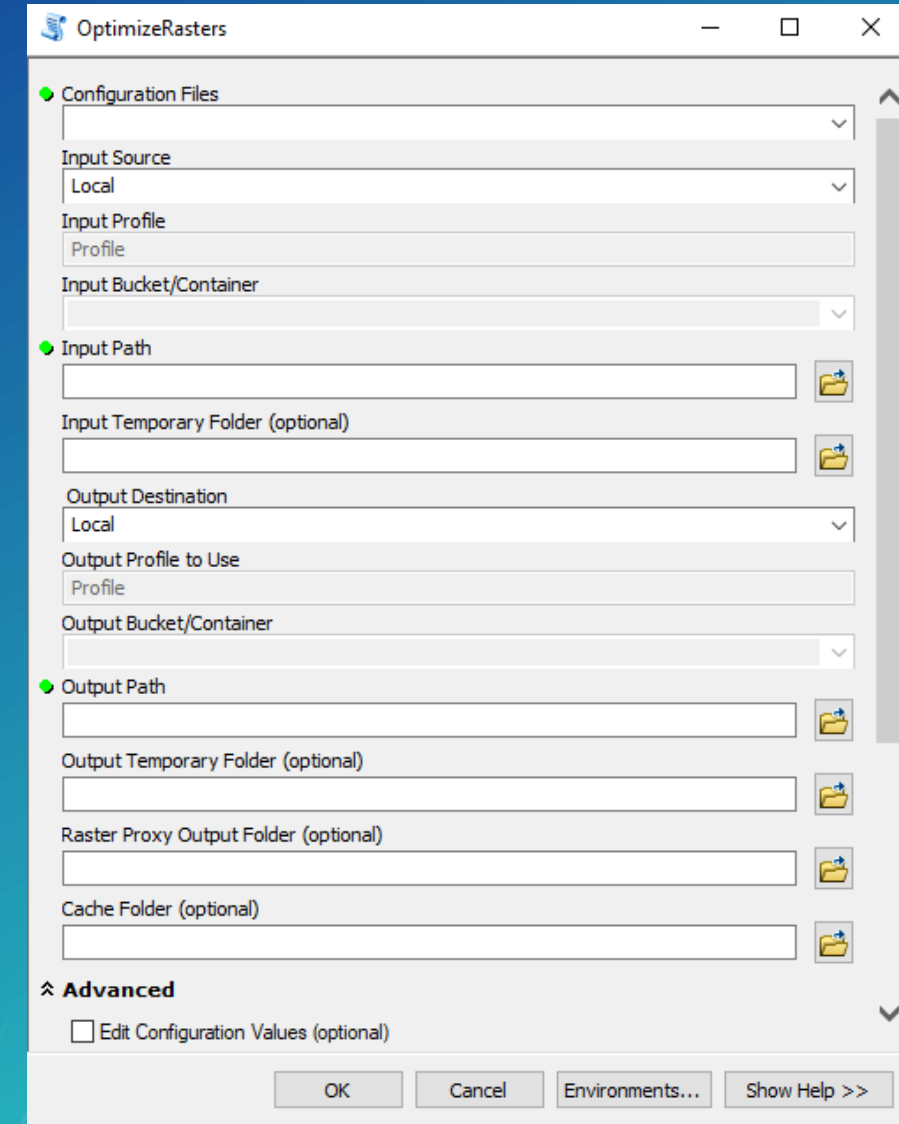


Converting Data to MRF

Using OptimizeRasters

- Available as Python Tool box and Command line python tool.
- Python Script
- Convert from any GDAL supported format to MRF or TiledTIF
- Predefined Templates with best practices to convert various satellite data
- Easy to modify set parameters to define compression, tilesize etc of output.
- Includes data upload and download options
- Works with data on S3,Azure, GoogleCloud storage and also from any local storage.
- Can run independent of ArcGIS.
- Can be scheduled to run periodically for new data.
- Logging support

<https://github.com/Esri/OptimizeRasters>



Getting Imagery to and from the Cloud

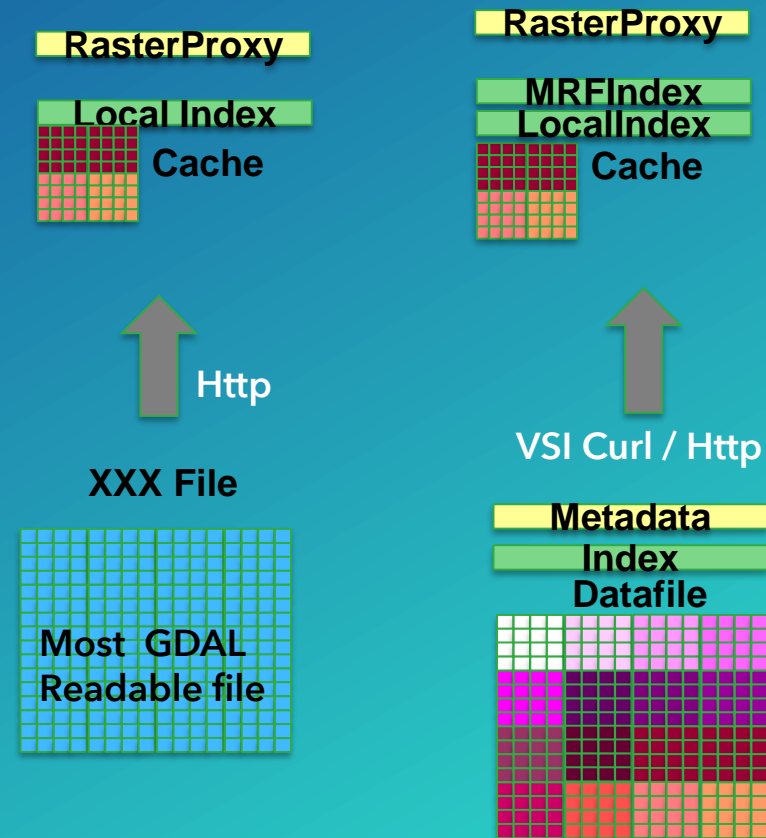
- AWS command line
 - CloudBerry, S3 Browser,
 - OptimizeRasters
 - Converts in optimized way to be used in cloud
 - Option to creates Raster Proxies
 - Check success and retries if necessary
 - Runs in parallel to speed up the processing
 - Supports Obfuscation
 - Supports Amazon S3, Microsoft Azure, Google Cloud storage
 - Support Lambda (serverless compute) (with some restrictions on data size)
 - Import/Export
 - Ship hard drives for large projects
- Note: File Names on S3 are case sensitive

Accessing Cloud Storage in ArcGIS

- ArcGIS typically access files through a file system
- Cannot generically access files in cloud storage
- Use “Raster Proxies” links imagery on cloud storage
- ArcGIS uses GDAL to read these Raster Proxies to access data on cloud
- Available from 10.5
- Requires install for Boto3 for S3

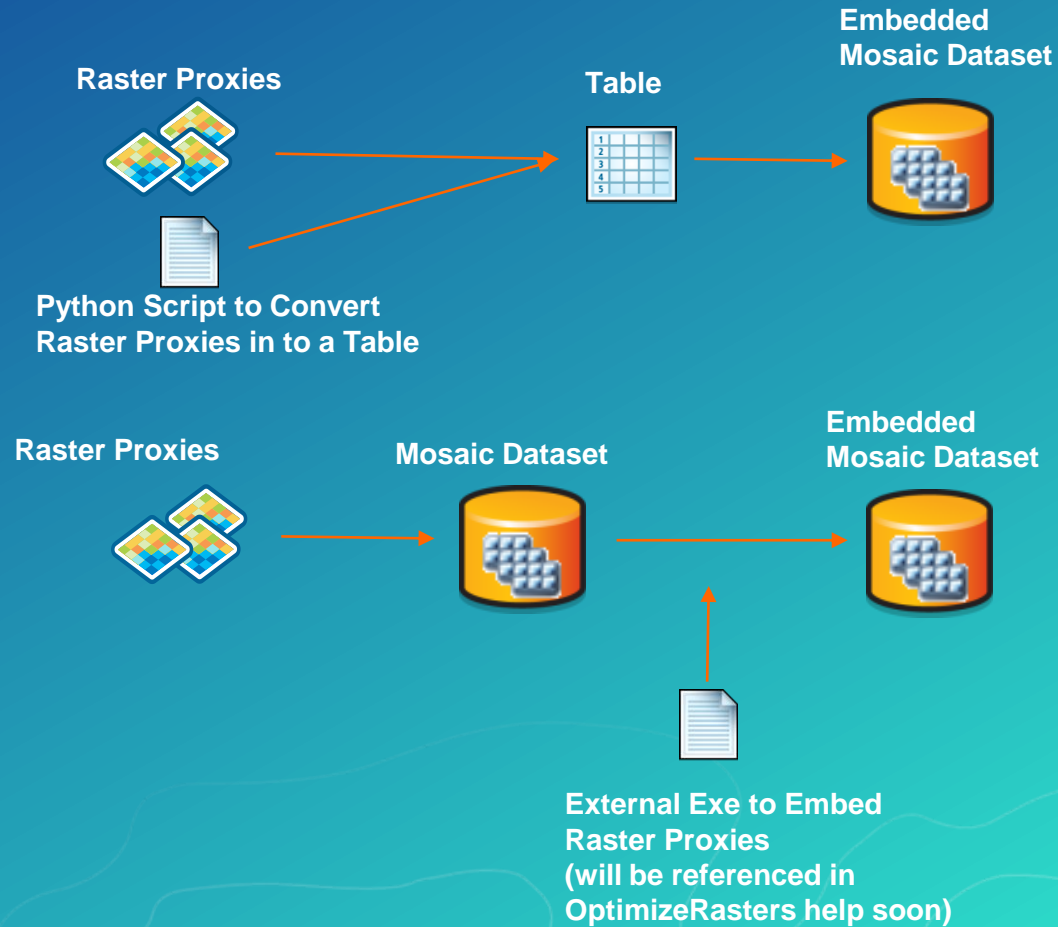
Raster Proxies

- Work with most GDAL Readable files
- Reference the source files
- Enables caching of data
 - Speed up subsequent reads
- Most Optimum when referencing MRF
- Can have any raster extension
- Use like any other Rasters
- Need to consider cache location
- Need to manage cache
- Create using OptimizeRasters
- Read Optimize Rasters Help



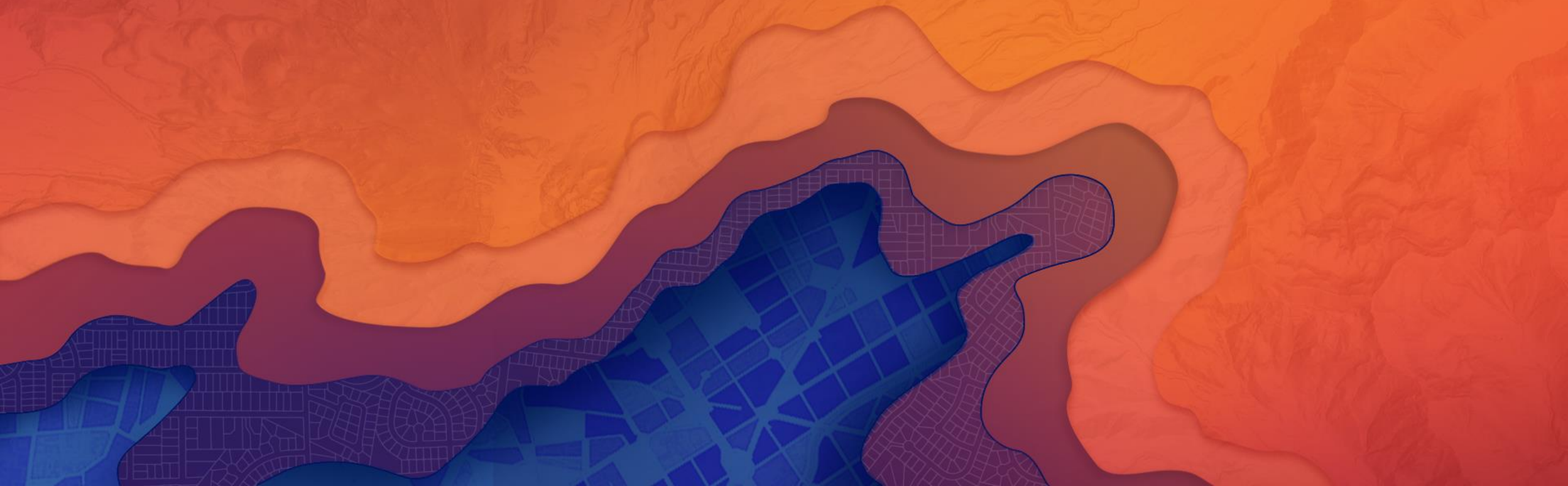
Embedding Raster Proxies into Mosaic Datasets

- Enables Mosaics with no file references
- Simplifies publishing in elastic environments
- Two ways to create:
 - Embedding the MRF string into a table
 - Use table raster type
 - Enables very fast load
 - Simple management for structured data
 - Does not support Aux files (aux.xml, RPC etc)
 - Embedding the Rater proxies just before publishing.
 - Can use all standard Raster types
 - Easy to modify the paths if needed.
 - Support all aux files
 - Need to run as a external command to embedded the MRF
 - Once embedding is done you can not repair the path or sync the mosaic dataset

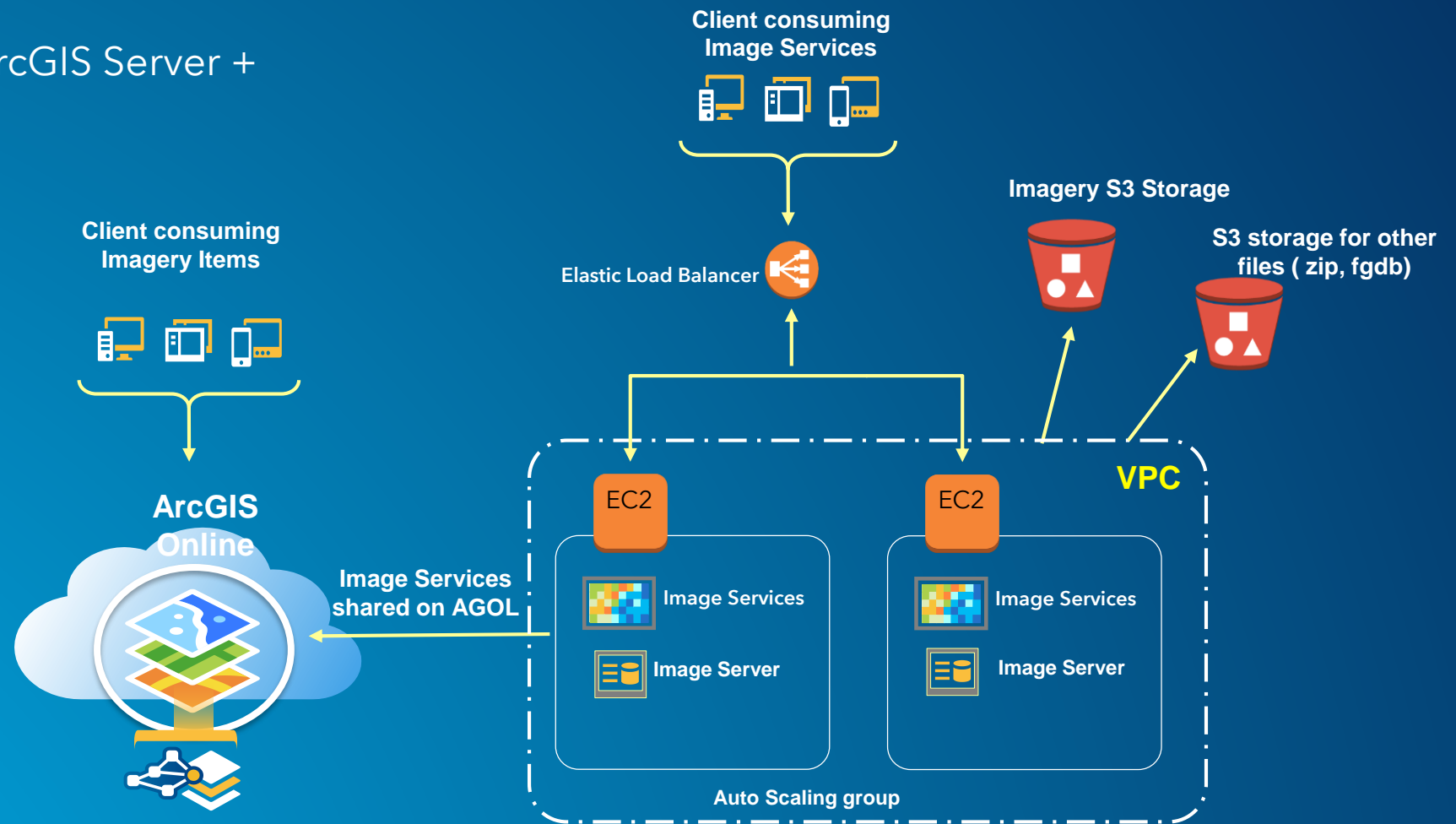


Cloud Based Architectures

For Image Server

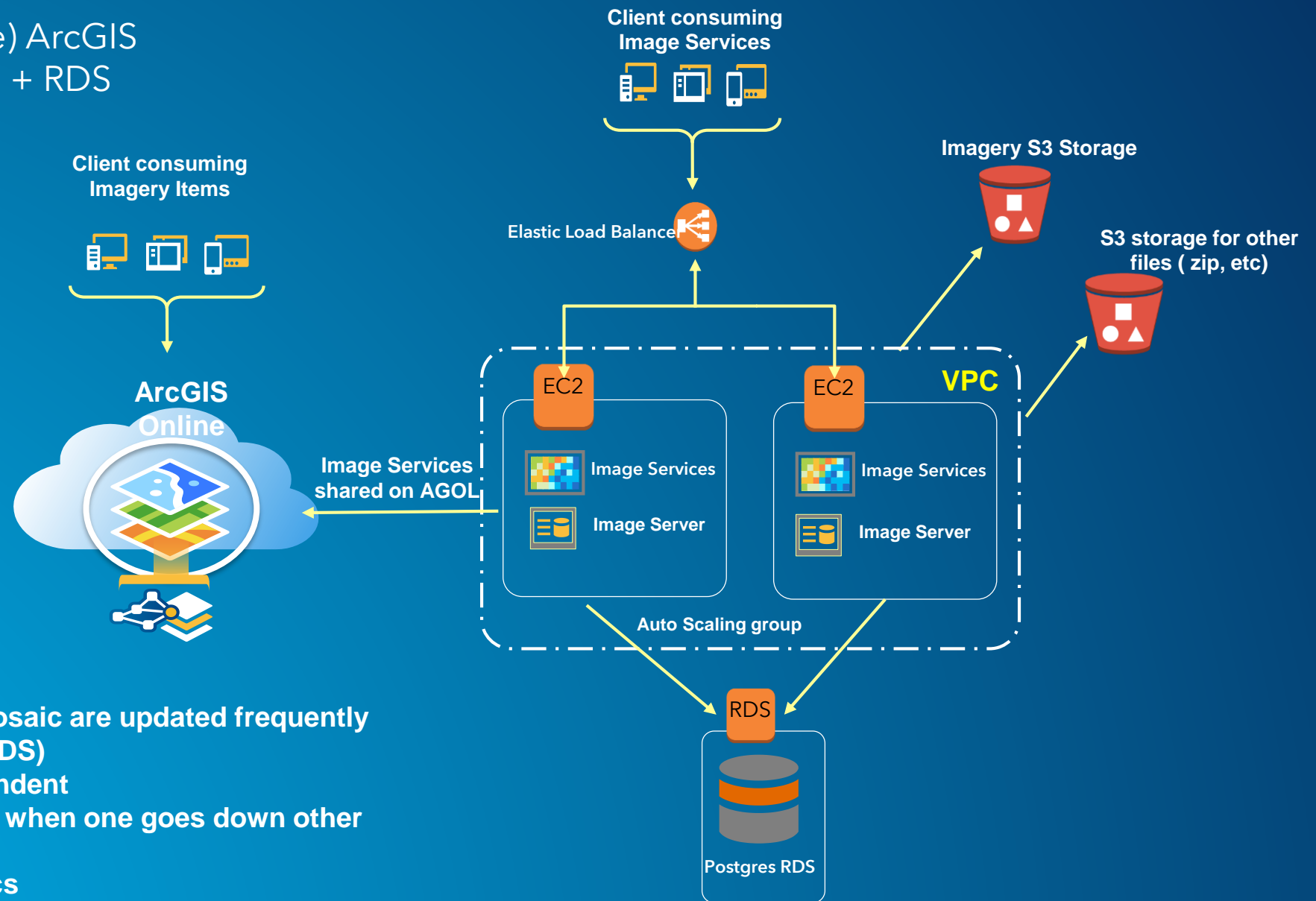


Siloed (single-machine) ArcGIS Server + Image server



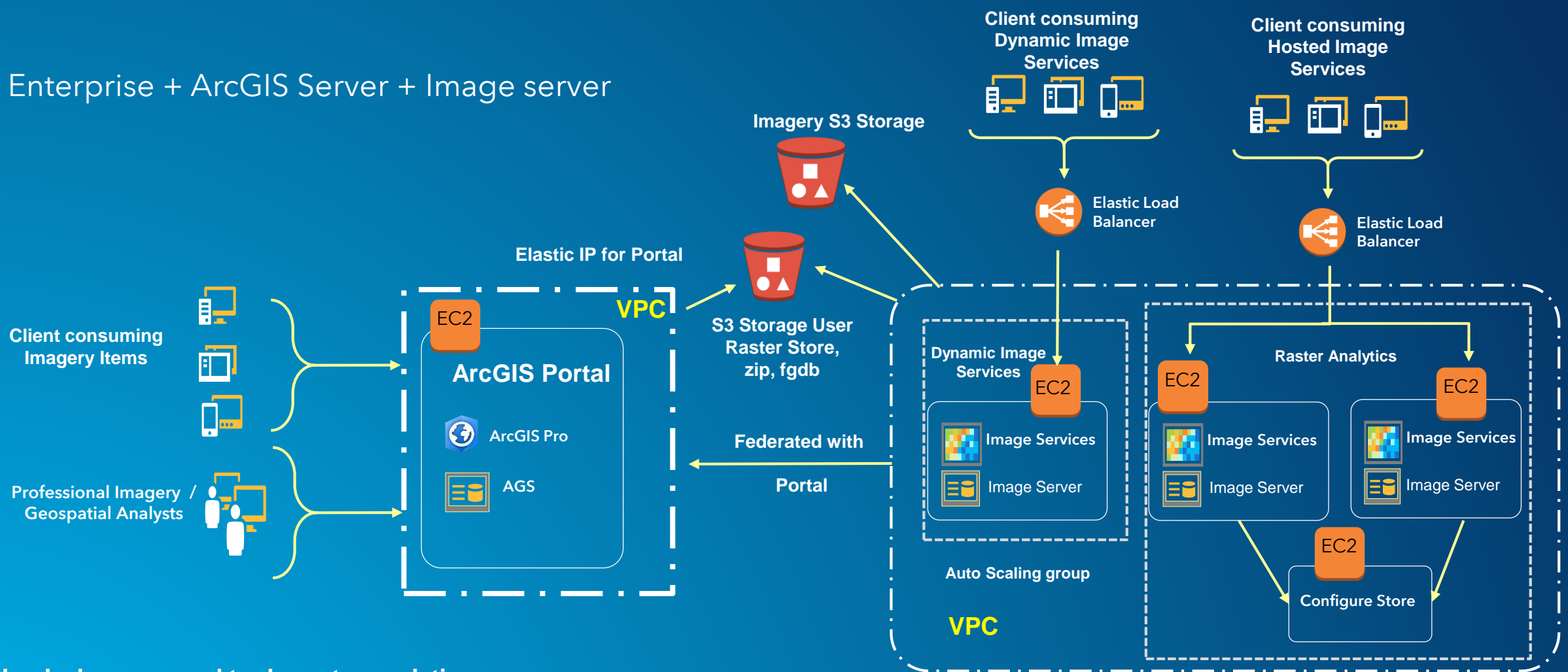
- Used in case where the mosaic are not updated frequently
- Mosaic sit in File GDB
- Each server site is independent
- Allows seamless recovery when one goes down other comes up
- Can not run raster analytics
- Load Balancer distributes the load on the server machines
- Each machine has its own config store

Siloed (single-machine) ArcGIS Server + Image server + RDS



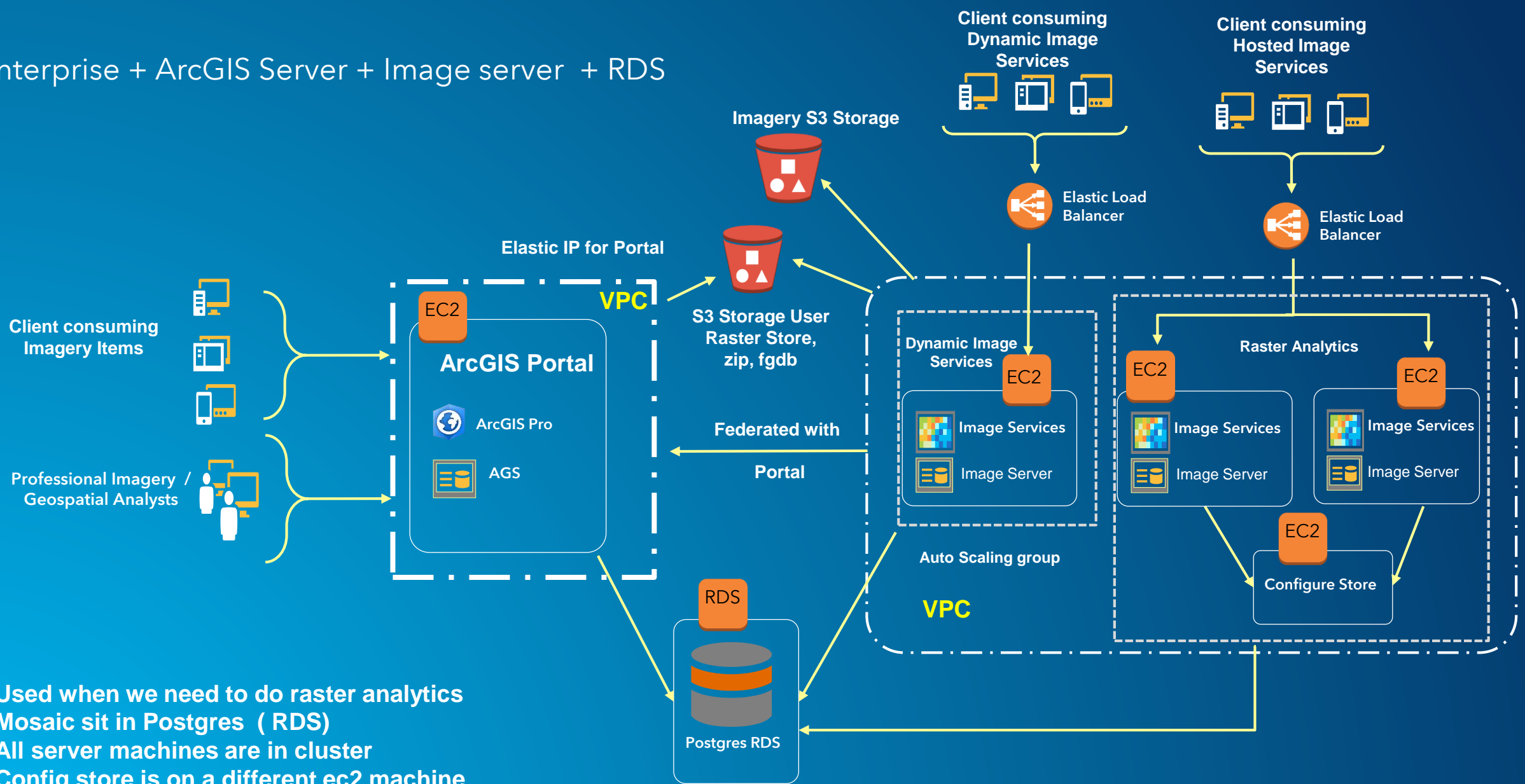
- Used in case where the mosaic are updated frequently
- Mosaic sit in Postgres (RDS)
- Each server site is independent
- Allows seamless recovery when one goes down other comes up
- Can not run raster analytics
- Load Balancer distributes the load on the server machines
- Each machine has its own config store

Enterprise + ArcGIS Server + Image server



- Used when we need to do raster analytics
- Different Stack for Dynamic services and raster analytic
- Mosaic sit in FileGDB in the config store,
- If the mosaic are not going to be updated they can sit each machine
- All server machines are in cluster
- Config store is on a different ec2 machine
- Portal and Image server same VPC

Enterprise + ArcGIS Server + Image server + RDS



- Used when we need to do raster analytics
- Mosaic sit in Postgres (RDS)
- All server machines are in cluster
- Config store is on a different ec2 machine
- Different Stack for Dynamic services and raster analytic

Cloud formation templates (10.5.1)

<https://s3.amazonaws.com/arcgisstore1051/7333/docs/index.html>

- **Siloed (single-machine) ArcGIS Server + Image server**
 - <https://s3.amazonaws.com/arcgisstore1051/7333/templates/arcgis-siloed-server-VPC.template>
- **Siloed (single-machine) ArcGIS Server + Image server + RDS**
 - <https://s3.amazonaws.com/arcgisstore1051/7333/templates/arcgis-siloed-server-VPC.template>
- **Enterprise + ArcGIS Server + Image server**
 - <https://s3.amazonaws.com/arcgisstore1051/7333/templates/arcgis-allinone-windows.template> (Portal)
 - <https://s3.amazonaws.com/arcgisstore1051/7333/templates/arcgis-server-windows.template> (AIS)
- **Enterprise + ArcGIS Server + Image server + RDS**
 - <https://s3.amazonaws.com/arcgisstore1051/7333/templates/arcgis-allinone-windows.template> (Portal)
 - <https://s3.amazonaws.com/arcgisstore1051/7333/templates/arcgis-server-windows.template> (AIS)

Cloud formation templates (10.5)

<http://arcgisstore105.s3.amazonaws.com/6491/docs/index.html>

- **Siloed (single-machine) ArcGIS Server + Image server**
 - <http://arcgisstore105.s3.amazonaws.com/6491/templates/arcgis-siloed-server-VPC.template>
- **Siloed (single-machine) ArcGIS Server + Image server + RDS**
 - <http://arcgisstore105.s3.amazonaws.com/6491/templates/arcgis-siloed-server-VPC.template>
- **Enterprise + ArcGIS Server + Image server**
 - <http://arcgisstore105.s3.amazonaws.com/6491/templates/arcgis-allinone-windows.template>
 - <http://arcgisstore105.s3.amazonaws.com/6491/templates/arcgis-image-server-windows.template>
- **Enterprise + ArcGIS Server + Image server + RDS**
 - <http://arcgisstore105.s3.amazonaws.com/6491/templates/arcgis-allinone-windows.template>
 - <http://arcgisstore105.s3.amazonaws.com/6491/templates/arcgis-image-server-windows.template>

Data Access Control

For Rasters



Data Access Control Options

- **Case 1 : Public**
 - Make the data public and have everyone access to read and list. In this case any one can access the data and list all the files only for read (no write)
 - (<https://aws.amazon.com/articles/5050/>)
- **Case 2 : Obfuscated**
 - Use Bucket policy to only read the data but not list. In this case if the user knows the path then only he can access the data
 - (<http://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html#example-bucket-policies-use-case-2>)
- **Case 3 : VPC**
 - Give permission only to the machine with the specific VPC (Virtual Private Cloud). Use the bucket policy to have an access to the data to list as well as read. All the request coming from the machine within this VPC will have the permission
 - (<http://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies-vpc-endpoint.html>)
- **Case 4 : Restricted IP**
 - Give Permission to a specific Ip address, in this case all the request coming from the given IP address will only be allowed
 - (<http://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html#example-bucket-policies-use-case-3>)
- **Case 5 : IAM Role - Not Currently Supported (soon)**
 - Access credentials updated by the amazon service, the access key will provide the access to s3. These key have to be stored as environmental variables.
 - Requires use of /vsis3/ vs /vscurl

Use of Obfuscation

- A simple alternative to common access control in cloud
- Less maintenance of user name and passwords
- Reduced security Overhead which can slow down the speed of access
- The folders in which the files are obfuscated so that it is not possible to guess the contents
- The bucket policy is set up so that users cannot query or get listings of the bucket content. Only users who know the URLs of the files can access them or share them
- The largest limitation to the file obfuscation method is that once a URL has been shared, access cannot be easily revoked for a single user

For More Details: www.esriurl.com/imageryworkflows

ArcGIS Features Plans Gallery Map Scene Help

Search Sign In

ArcGIS Imagery Workflows

Tools and best practices to help you manage, analyze, and use your imagery and rasters

Search Imagery Workflows



Image Management Tools

Image Analysis Tools

Image Use Tools

Manage Imagery

