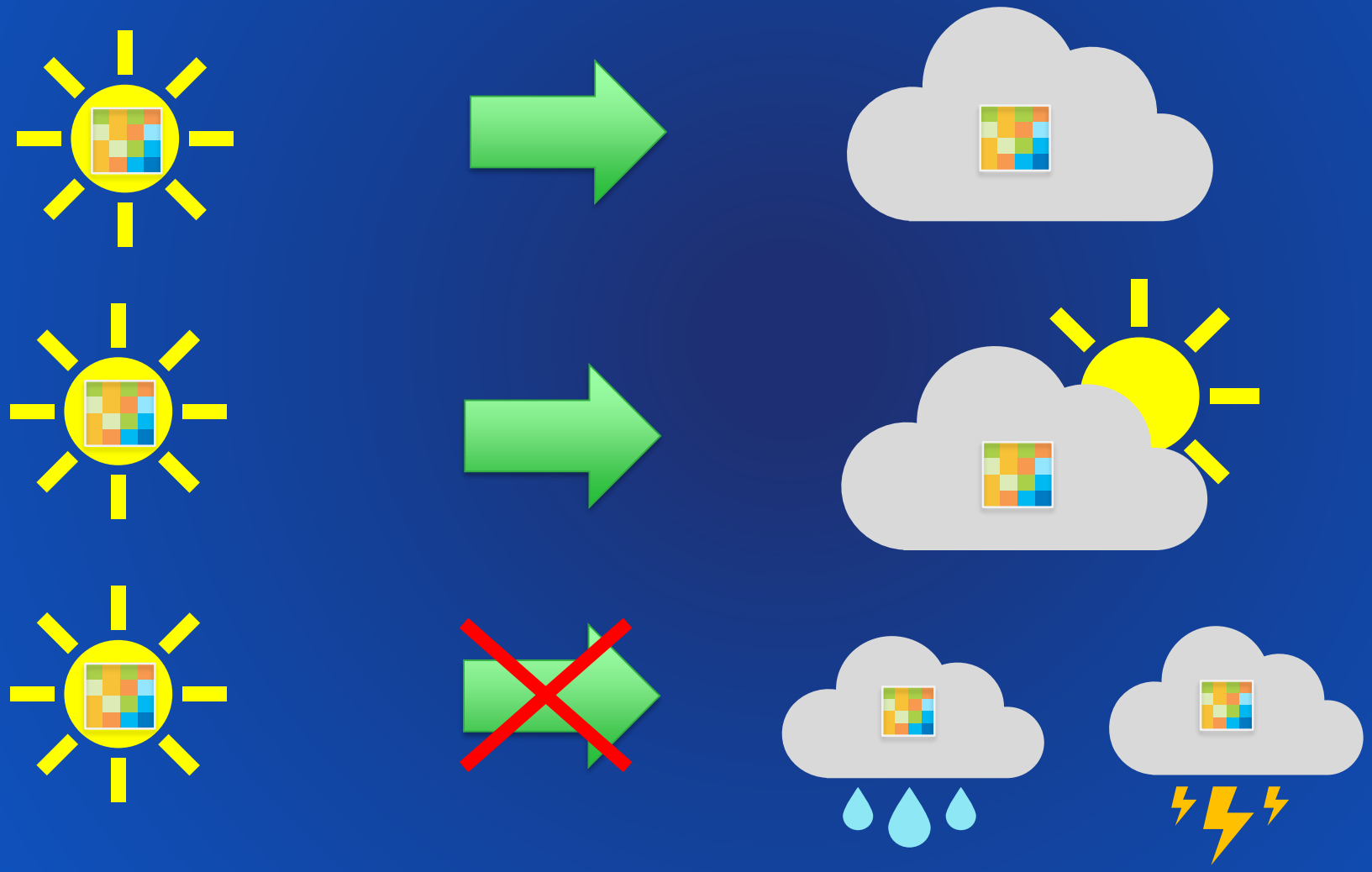
Managing and Serving Imagery in the Cloud Using ArcGIS Image Server

Peter Becker

SEE
WHAT
OTHERS
CAN'T

Make Imagery Accessible for Visualization and Analysis Using Cloud



Don’t just copy your data into the cloud and expect it all to work

What is Cloud Computing?

- Use network of remote servers hosted on the internet
  to store, manage and process data,
  rather than a local server or personal computer

What is different in Cloud?

- Implementation and Management of Infrastructure (EC2, ELB, Storage)
- Object Storage (Https) vs Local (SAN/NAS)
- Elasticity in scaling
- Security
- Cost : Pay for the use
- Test and Scale as required

Why Manage and Serve Imagery in the Cloud?

- You have large collection of imagery
- Infrastructure is getting expensive
- Have heard that Cloud is ： Cheap, Secure, Elastic, Resilient, Simple, … ??
- Organization is going “Cloud First”

Cloud Infrastructure

- ArcGIS Online (SaaS)
  - Imagery currently limited to Tile Cache (Base Maps & Elevation) for imagery
- AWS – EC2, S3
- Azure – VMs, BlobStore
- Other – VMs, CloudStorage

5 Key Imagery Capabilities of ArcGIS

Visualization & Exploitation

Management

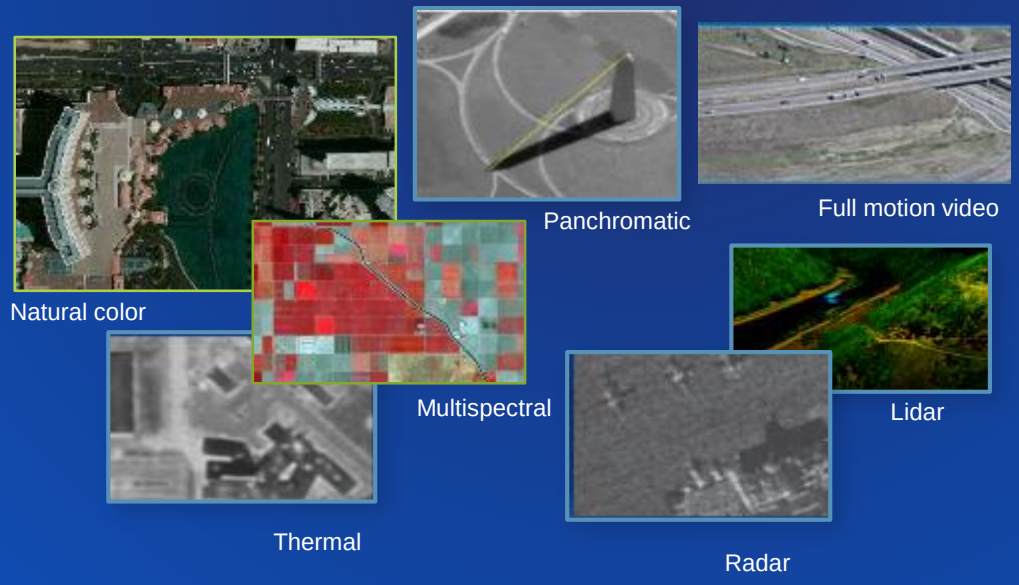
Analysis

Content

Map Production

ArcGIS Support for All Imagery and Raster Formats

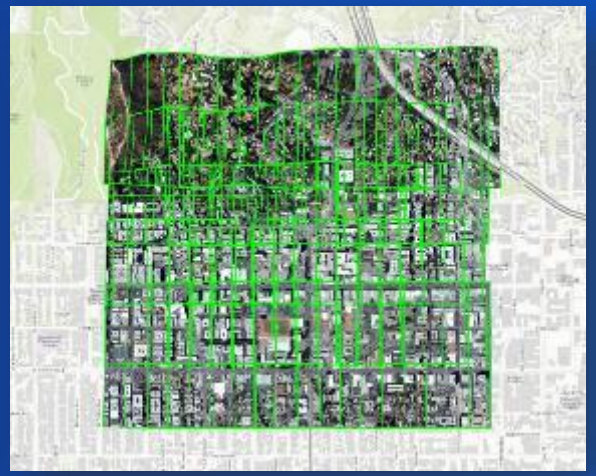
- Different sensor platforms – Satellite, Aerial, Drones, Categorical, Scientific, ..
- Different sensors – Optical, Thermal, Radar, Lidar, Scanned, Generated – Categorical,…
- Different modalities – Multispectral, Panchromatic, Complex,
- Different levels and ‘Products’ from vendors (Imagery+Metadata)
- Different formats. TIF, NITF, NetCDF, JP2, MrSID, ….. Some are better than others



Mosaic Datasets

Enable Management of Large Collections

- **Optimum Data Model for Image Management**
- **Highly Scalable, from Small to Massive Volumes of Imagery**
- **Defined in GeoDatabase (File or Enterprise)**
- **References sources**
- **Maintains metadata**
- **Defines processing to be applied at Item and Service Level**
- **Defines mosaicking rules**
- **Defines overview to be used at small scale**
- **Provides:**
  - **Dynamic Mosaicking**
  - **On-the-fly processing**
- **Access as Image or Catalog**

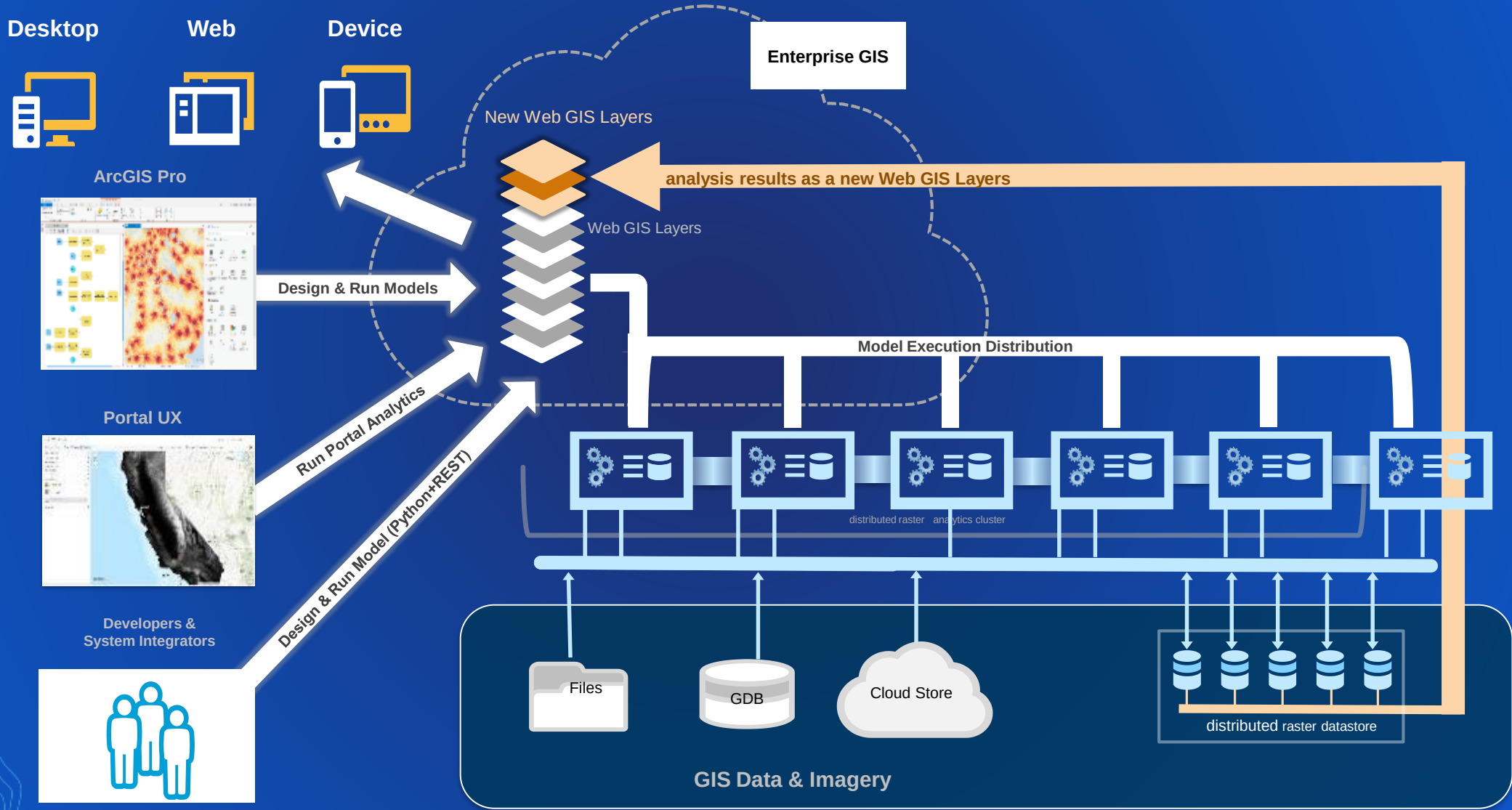


ArcGIS Image Server

Key Capabilities

- **Imagery hosting**
  - Enables users within organization to upload imagery into ArcGIS Enterprise and serve it as dynamic imagery layers
- **Dynamic image services**
  - Web accessible imagery which can have processing applied on-the-fly
- **Raster Analytics**
  - Quickly process and persist data to create new information products
- **Ortho mapping**
  - Processing of satellite, aerial or drone imagery into digital elevation models and ortho mosaics

Scaling ArcGIS Image Server with Raster Analytics

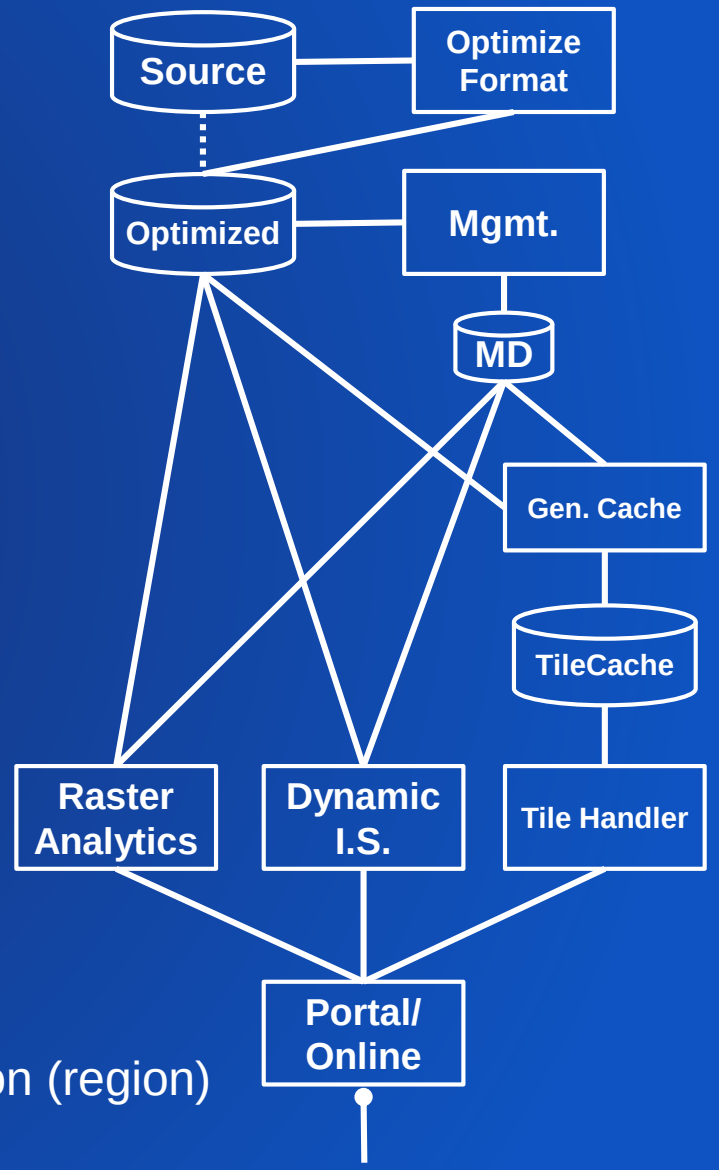


Components of a Complete Imagery Solution

On Premises or Cloud

- **Storage (Source and Optimized)**
- **Management**
- **Generate TileCache**
- **Serve TileCache**
- **Serve Dynamic Image Services**
- **Perform Analytics**
- **Access Control (Portal or ArcGIS Online)**

Where should each of these be performed? On premises or cloud?



Need to ensure that storage and processing are at same location (region)

On Premises vs On Cloud Options

		On-Premises	In-Cloud
Store imagery	File System		
	Object Store		
Manage Imagery	ArcGIS Pro		
	Portal Web		
	Automated		
Serve Tile Cache	ArcGIS Online		
	ArcGIS Enterprise		
Serve Dynamic Image Services	ArcGIS Enterprise + ArcGIS Image Server		
	ArcGIS Image Server stand alone		
Analysis + Ortho Mapping	ArcGIS Pro		
	ArcGIS Enterprise + Image Server		
Portal	ArcGIS Enterprise		
	ArcGIS Online		

On Premises vs On Cloud Options

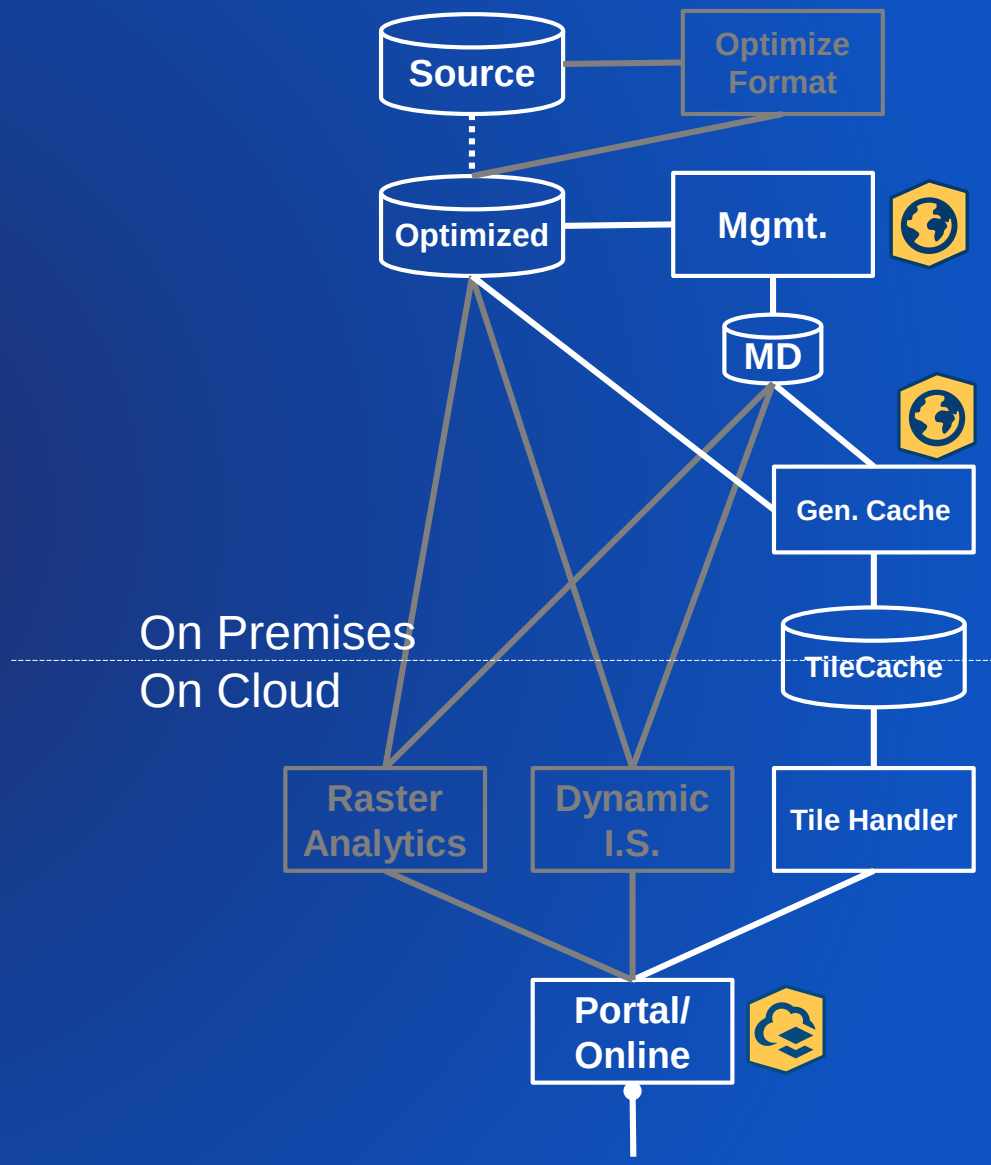
		On-Premises	In-Cloud
Store imagery	File System		
	Object Store		
Manage Imagery	ArcGIS Pro		
	Portal Web		
	Automated		
Serve Tile Cache	ArcGIS Online		
	ArcGIS Enterprise		
Serve Dynamic Image Services	ArcGIS Enterprise + ArcGIS Image Server		
	ArcGIS Image Server stand alone		
Analysis + Ortho Mapping	ArcGIS Pro		
	ArcGIS Enterprise + Image Server		
Portal	ArcGIS Enterprise		
	ArcGIS Online		

Best Practices			
1	2	3	4
☀	☀		
	☁	☁	☁
☀		☁	☁
	☁		
			☁
☁			
	☁	☁	
	☁	☁	
			☁
	☁	☁	
	☁	☁	
			☁

Best Practice – 1
Manage Local / TileCacheToArcGIS Online

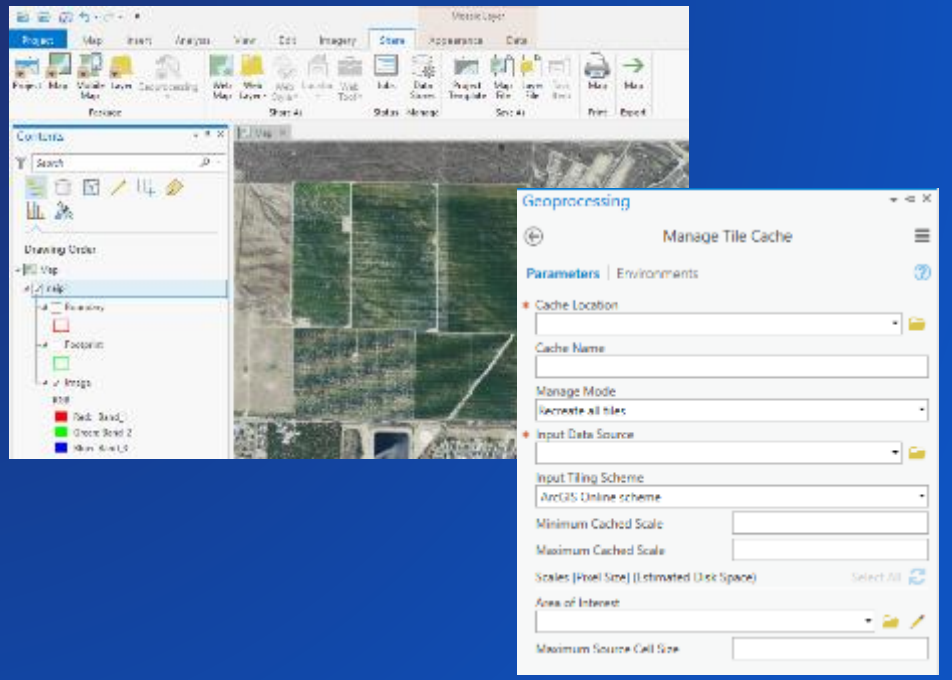
Only for 3band RGB compressed + Elevation

- **Keep imagery on Local File Systems**
- **Manage using Mosaic Datasets, Review and do QA**
- **Generate Tile Cache Using ArcGIS Pro 2.4**
- **Package (TPKX) and Upload to ArcGIS Online**
- **Publish to ArcGIS Online (C1.2/GB/Month)**
  (Very roughly: 100SqMiles@6” - $5/Month, 10,000Km2@15cm - $20/Month)
- **Access as BaseMaps in all applications + Elevation in Scenes**
- **Advantage**
  - **Simple, Inexpensive**
  - **No server to install**
- **Disadvantage**
  - **Need to pre-process all data**
  - **Only for 3band RGB compressed + Elevation**
  - **Not usable for analysis (but can be used for deep learning based)**
  - **No dynamic mosaicking, processing functions**



Tile Cache Tools

- **Use: ArcGIS Pro 2.4**
  - **Generate Tile Cache Tiling Scheme to define the tiling grid and image format**
  - **Manage Tile Cache to build the tiles**
  - **Export Tile Cache to create the tile package**
  - **Share Package to upload the tiles to ArcGIS Online or your ArcGIS Enterprise portal**
- **Note**
  - **Tile cache can be used as a large raster**
  - **TPKX can be used as a large raster**

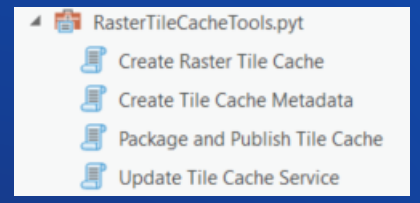


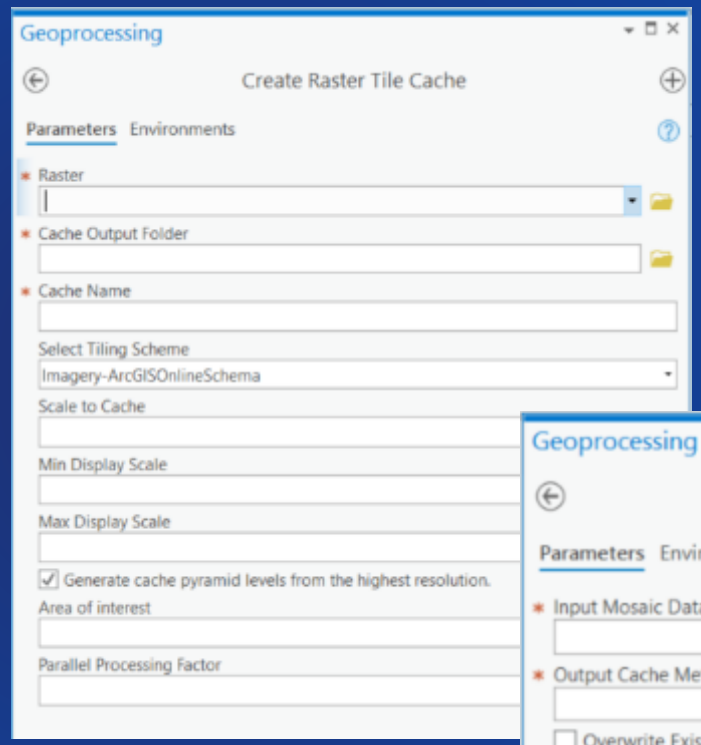
Recommend to use ArcGIS Pro 2.4
Includes number of enhancements with TPKX

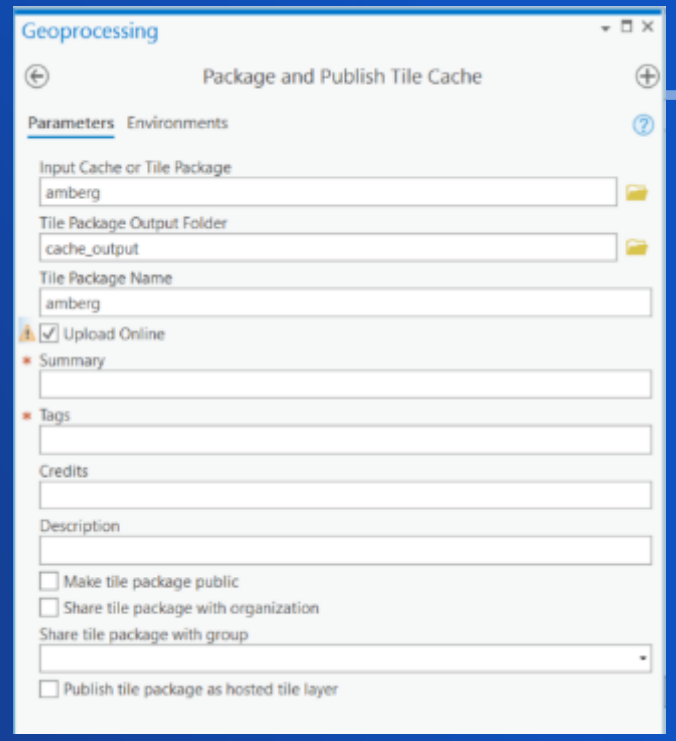
Raster Tile Cache Tools (Custom GPTool)

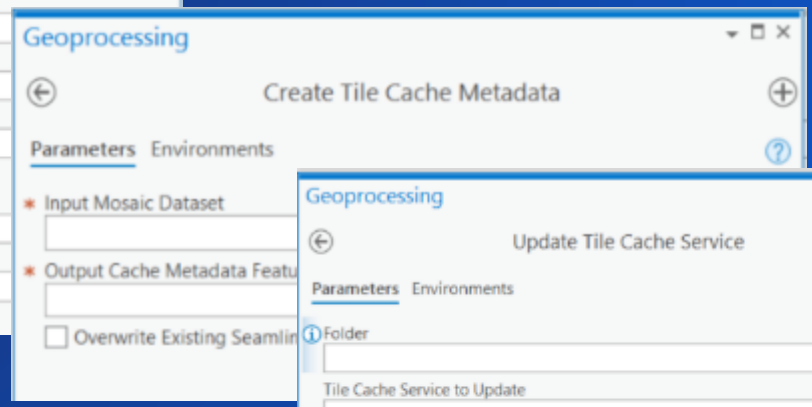
Custom tool to simplify caching – From Imagery Workflows

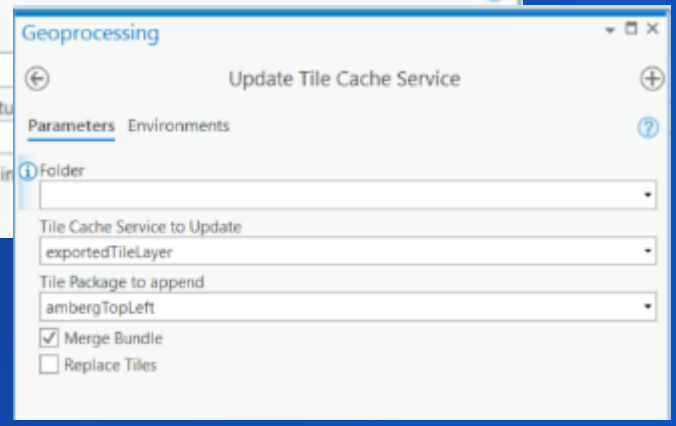
- **Use: Raster Tile Cache Tools**
  - **Create Raster Tile Cache**
  - **Package & Publish**
  - **Create Cache Metadata**
  - **Update Tile Cache**





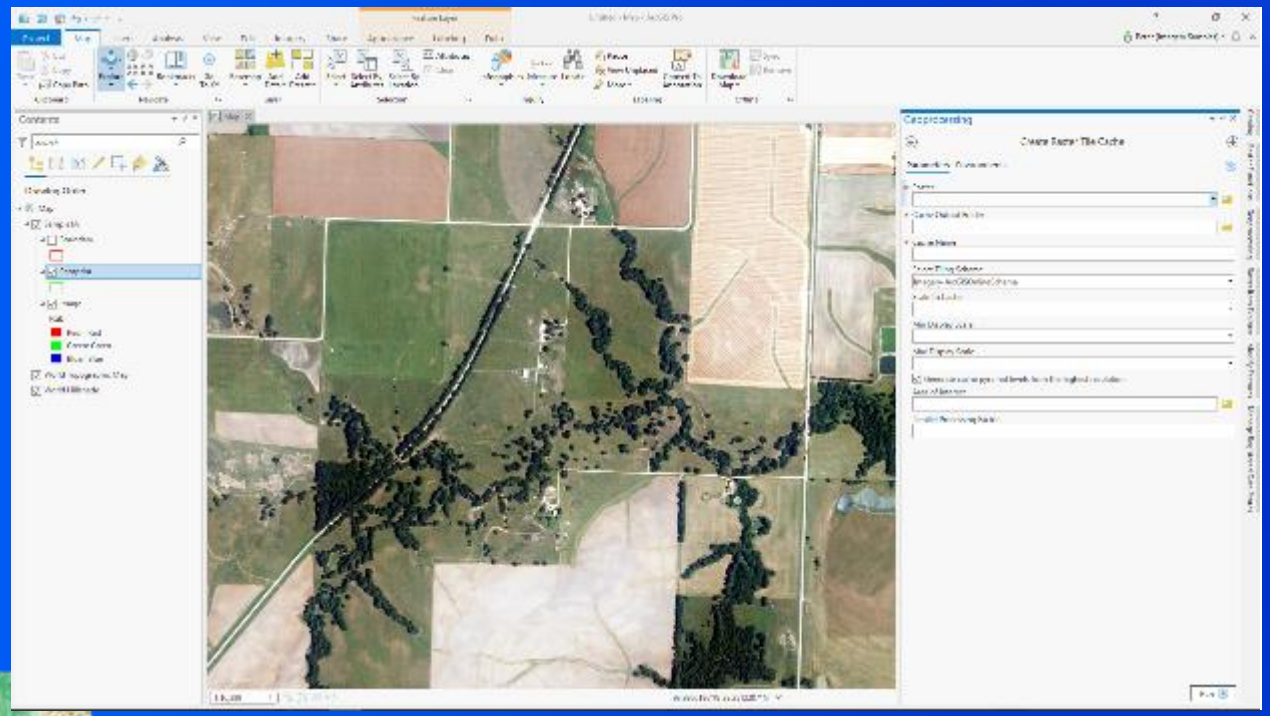






Requires ArcGIS Pro 2.4
Utilizes enhancements with TPKX

For more details see: http://doc.arcgis.com/en/imagery/workflows/resources/serving-cached-imagery.htm

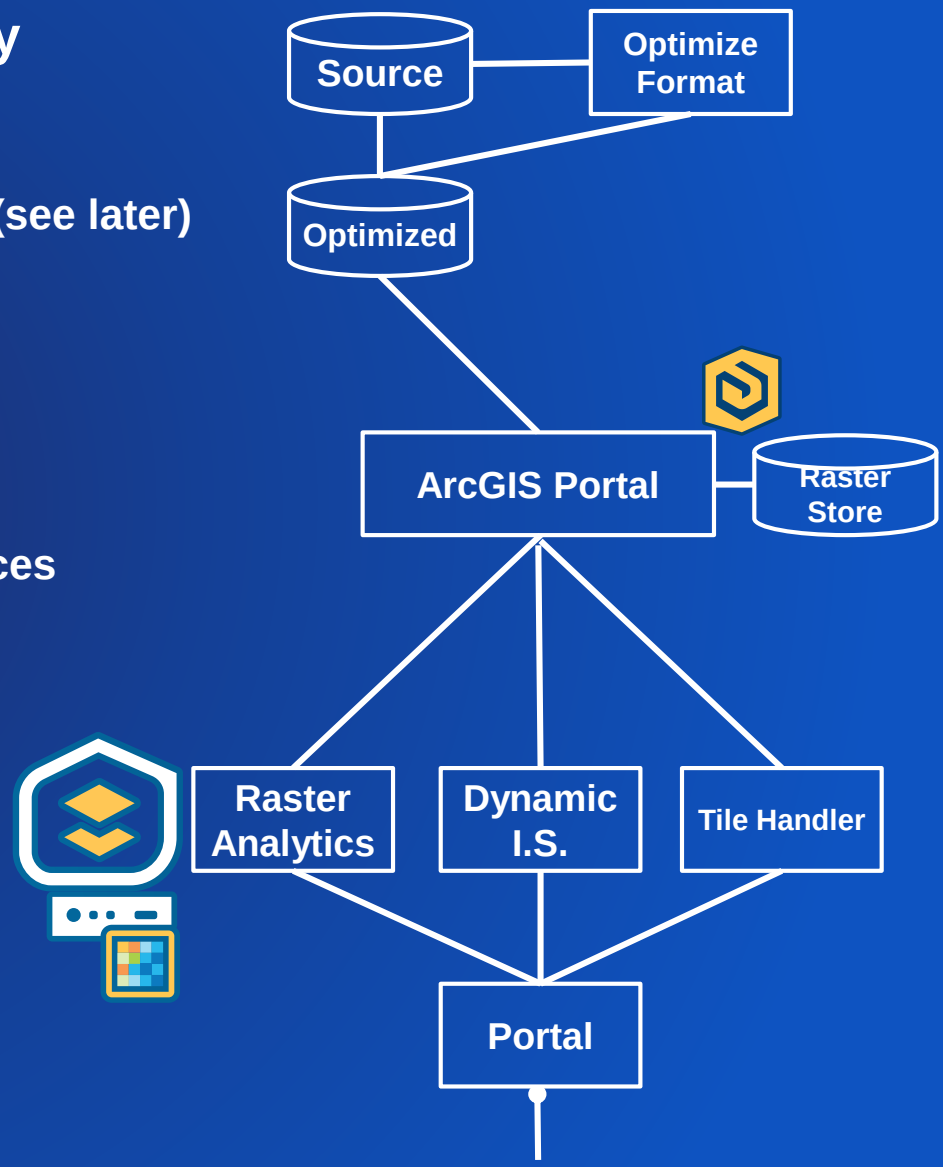


Create and Publish Tile Cache

Best Practice – 2

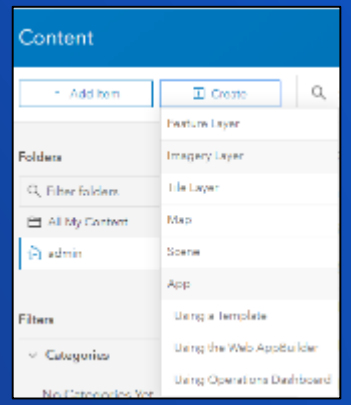
ArcGIS Enterprise + Image Server 10.7.1 - Hosted Imagery

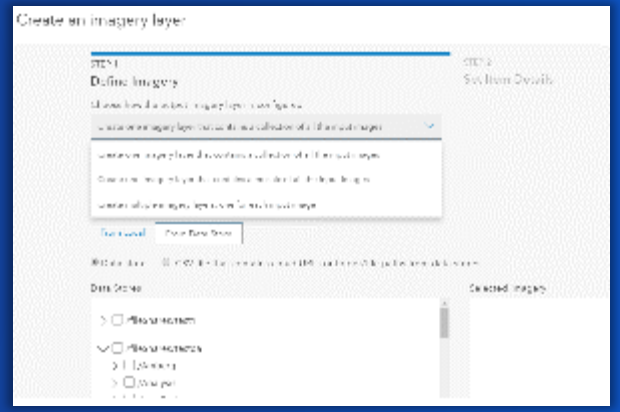
- Use: ArcGIS Enterprise Portal – Web
  - Source imagery can be local file or Referenced from Cloud Store (see later)
  - 3 Options: (One mosaic, Image Collection, Individual Layers)
  - Uses Shared instances (see later)
  - Utilize Raster Analytics for distribute processing
- Advantage
  - Simple: Anyone in organization can publish dynamic image services
  - Utilize full Raster Analytics and Ortho Mapping capabilities
  - Fully on-premises or on-cloud
- Disadvantage
  - Limited to simple mosaic dataset
  - No control over mosaic dataset properties
  - Limited Optimization
  - Requires DevOps

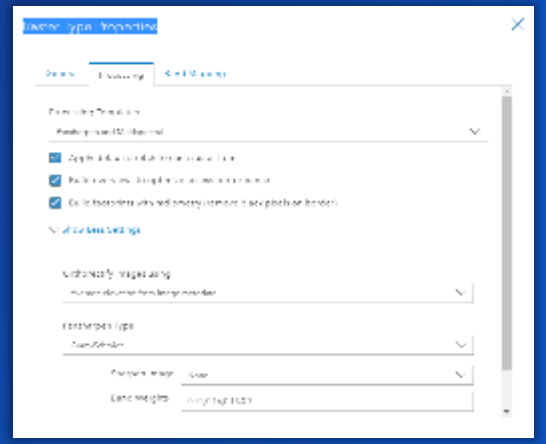


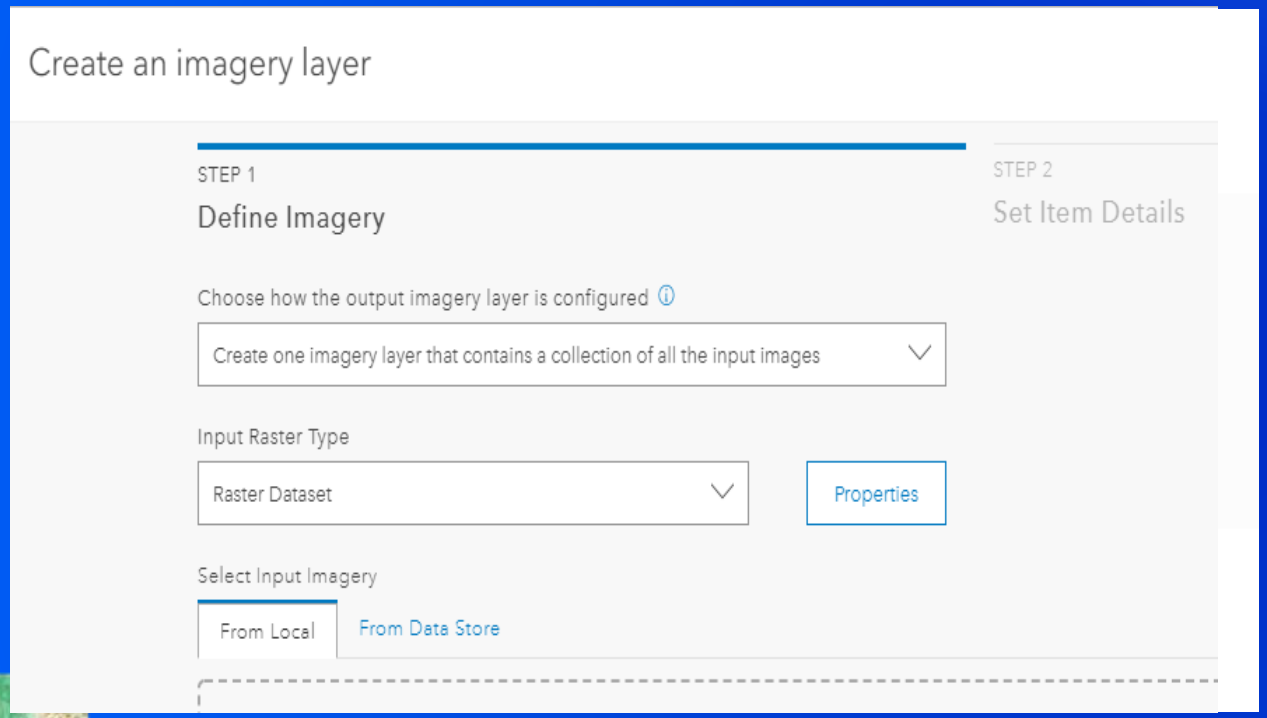
Hosted Imagery

- Portal User Interface
- Upload (or reference) imagery
- Create:
  - One imagery layer that is a mosaic of all the input images
    - Mosaics images into a CRF on raster store
  - One imagery layer that contains a collection of input images
    - Creates and publishes a mosaic dataset. Images remain in original format
  - Multiple imagery layers, one for each input image
    - Creates a single image layer for each. Remain in original format
- Optional Raster Types for some sensor products







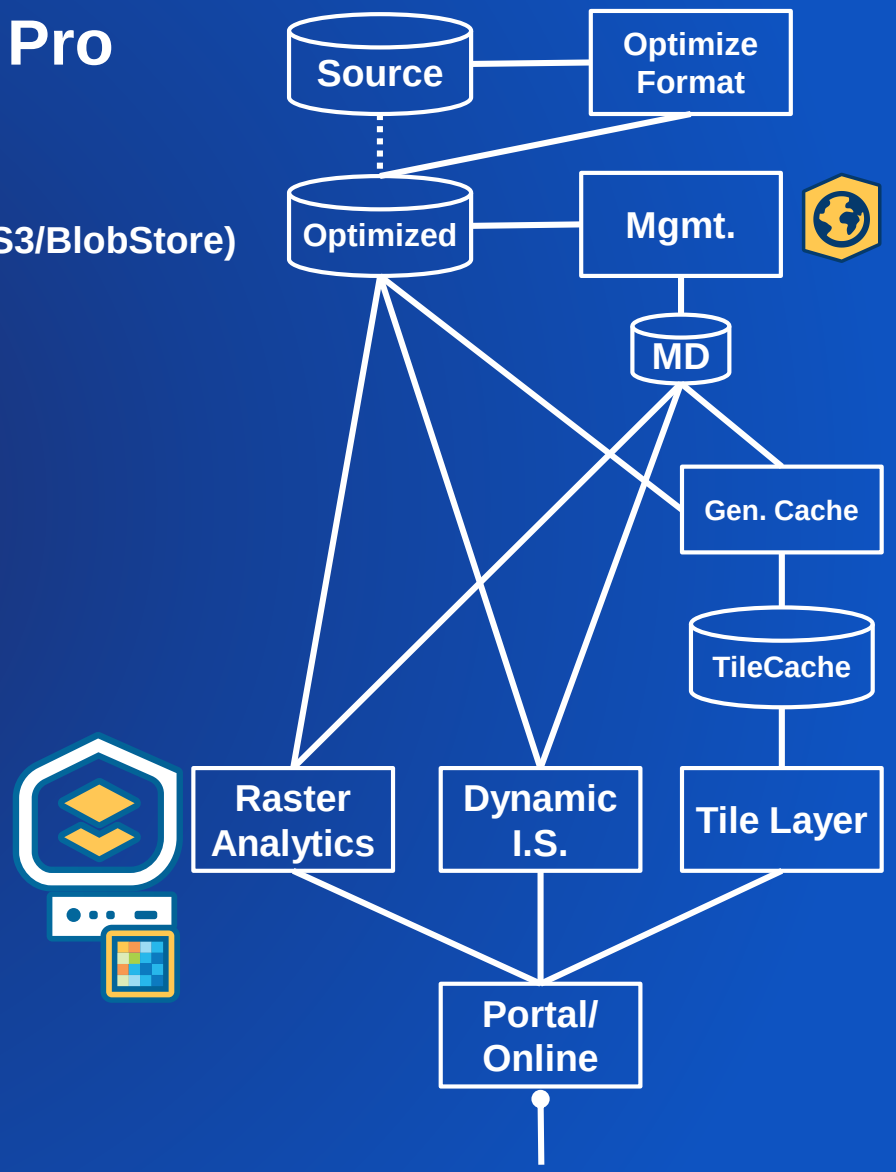


Hosted Imagery in ArcGIS Enterprise 10.7.1

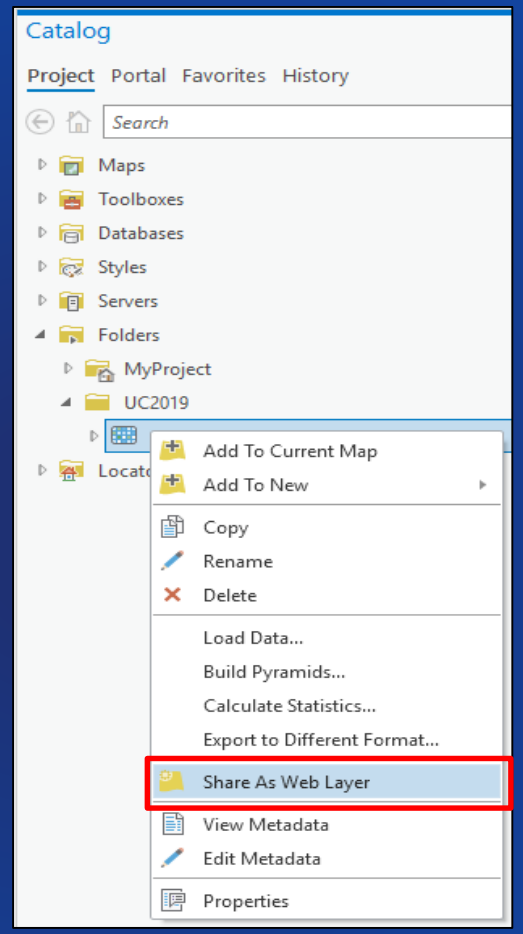
(With ArcGIS Image Server)

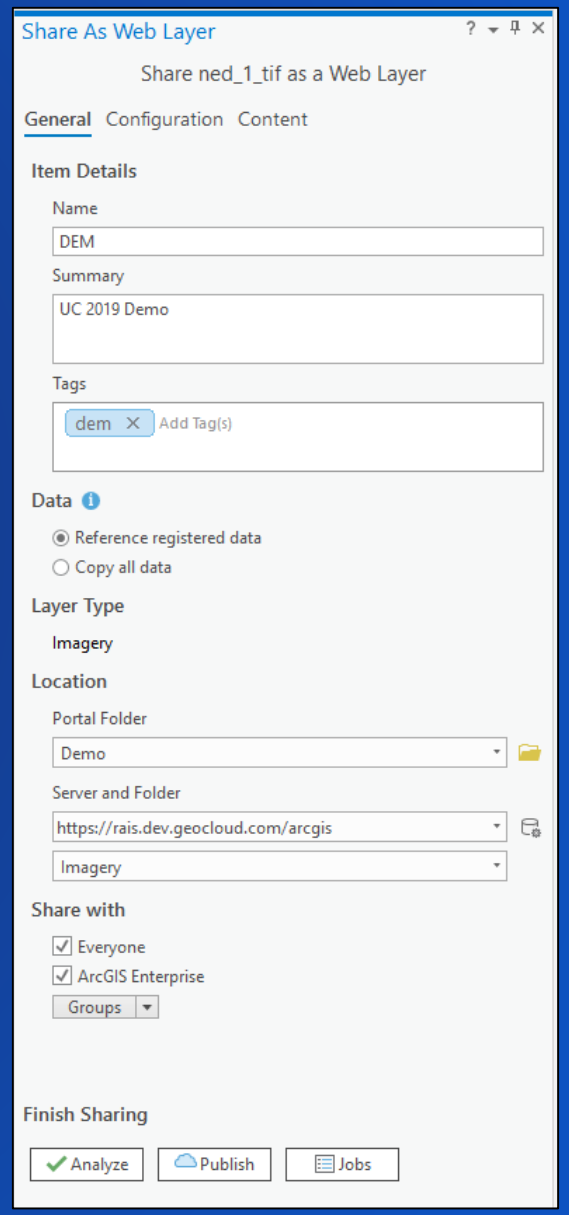
Best Practice – 3
ArcGIS Enterprise + Image Server – Use ArcGIS Pro

- Use: ArcGIS Enterprise + ArcGIS Pro (using RDP in cloud)
  - Source can be from Data store or Native cloud storage (eg public S3/BlobStore)
  - Use ArcGIS Pro to manage imagery and create mosaic dataset
    - Store mosaic dataset on Enterprise Geodatabase (eg RDS)
    - Reference imagery using (ACS, VSI or Raster Proxies)
  - Utilize Raster Analytics for distribute processing
- Advantage
  - Full capabilities of Mosaic Datasets
  - Utilize full Raster Analytics and Ortho Mapping capabilities
- Disadvantage
  - Requires separate cloud machine for ArcGIS Pro
  - Need DevOps



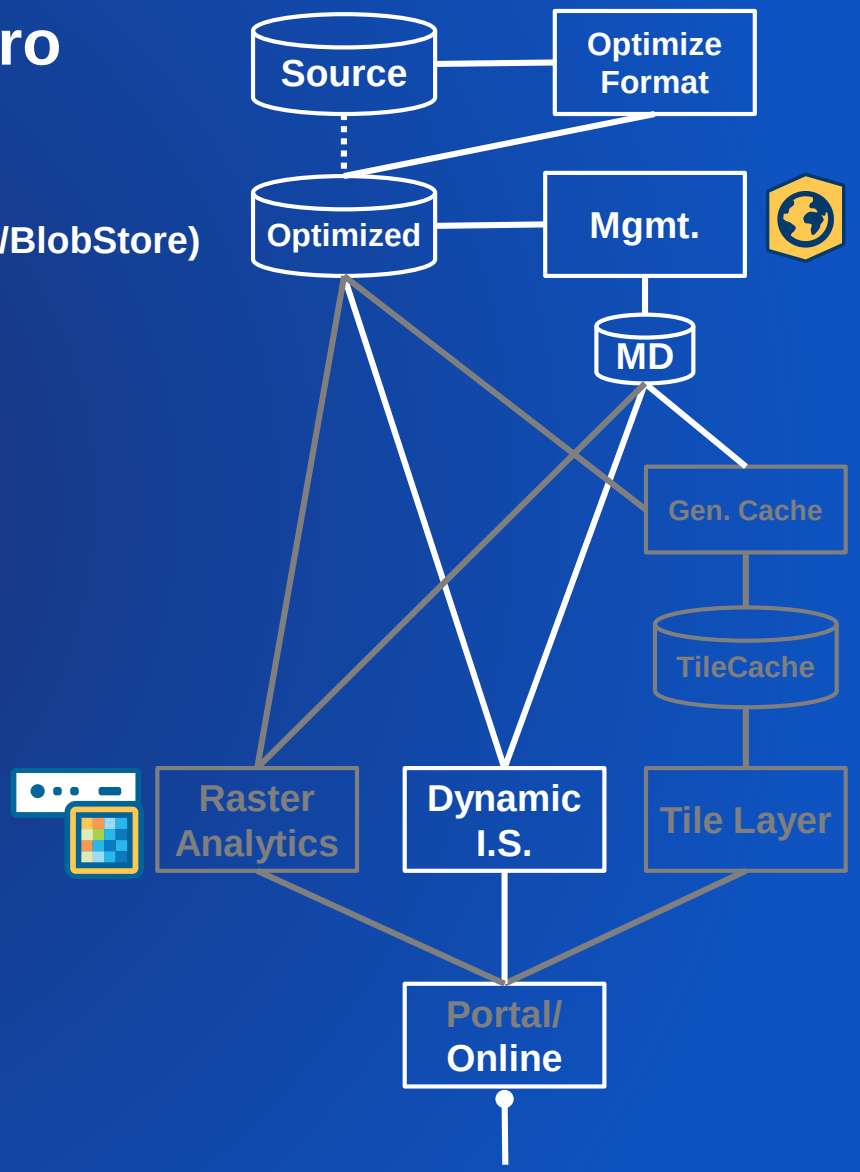
Share as Web Layer





Best Practice – 4
ArcGIS Image Server stand alone – Use ArcGIS Pro

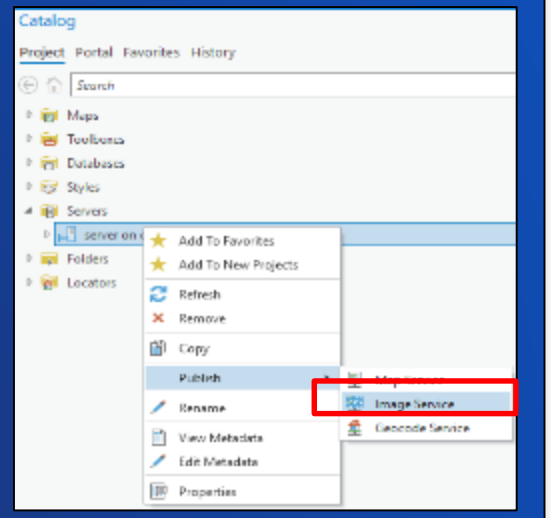
- Use: ArcGIS Pro (running RDP in cloud)
  - Source can be from Data store or Native cloud storage (eg public S3/BlobStore)
  - Use ArcGIS Pro to manage imagery and create mosaic dataset
    - Store mosaic dataset on Enterprise Geodatabase (eg RDS) of fileshare
    - Reference imagery using (VSI, ACS or Raster Proxies)
- Advantage
  - Full capabilities of Mosaic Datasets
  - Lower infrastructure costs
- Disadvantage
  - Requires separate cloud machine for ArcGIS Pro
  - Not utilize Raster Analytics, OrthoMapping or Image Hosting
  - Need to user direct server connection / else Item on ArcGIS online

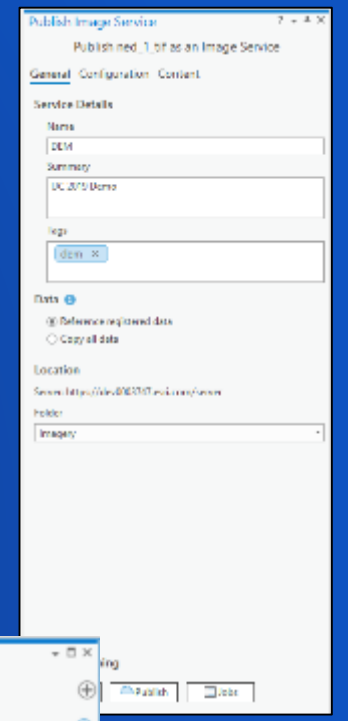


Publishing from Pro to Image Server Stand Alone

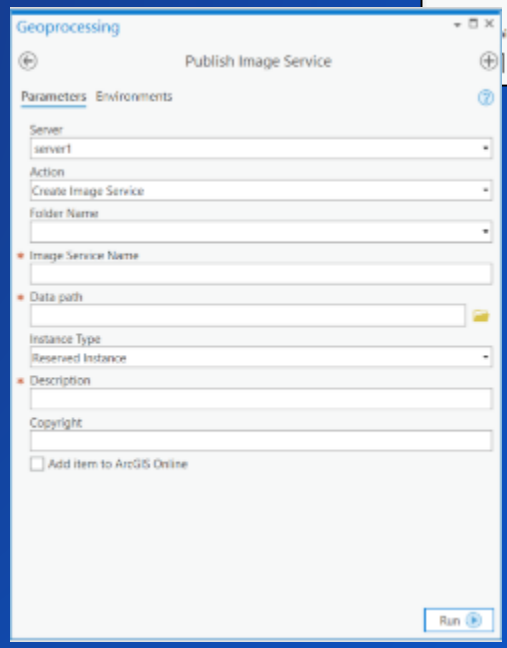
Two methods

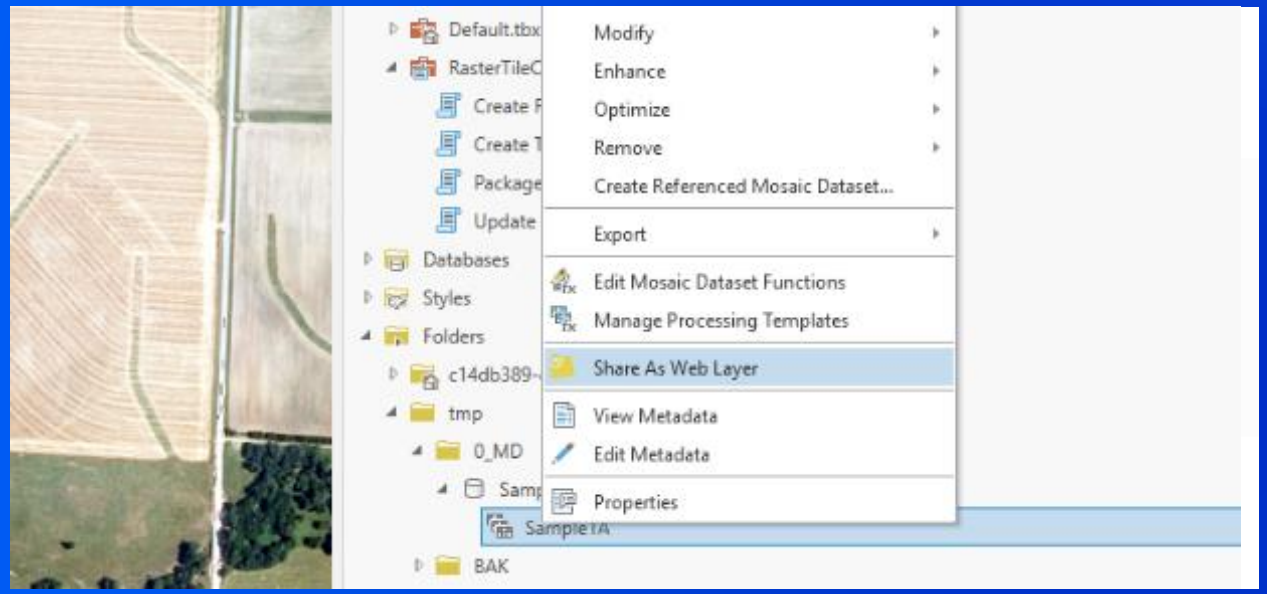
- **ArcGIS Pro 2.4**
  - **Share as Web Layer**





- **Publish Image Service Tool**
  - **Part of MDCS (see Imagery Workflows)**
  - **Will also create Imagery Layer Items in ArcGIS Online (or Portal)**
  - **Define if Dedicated or Shared Instance**





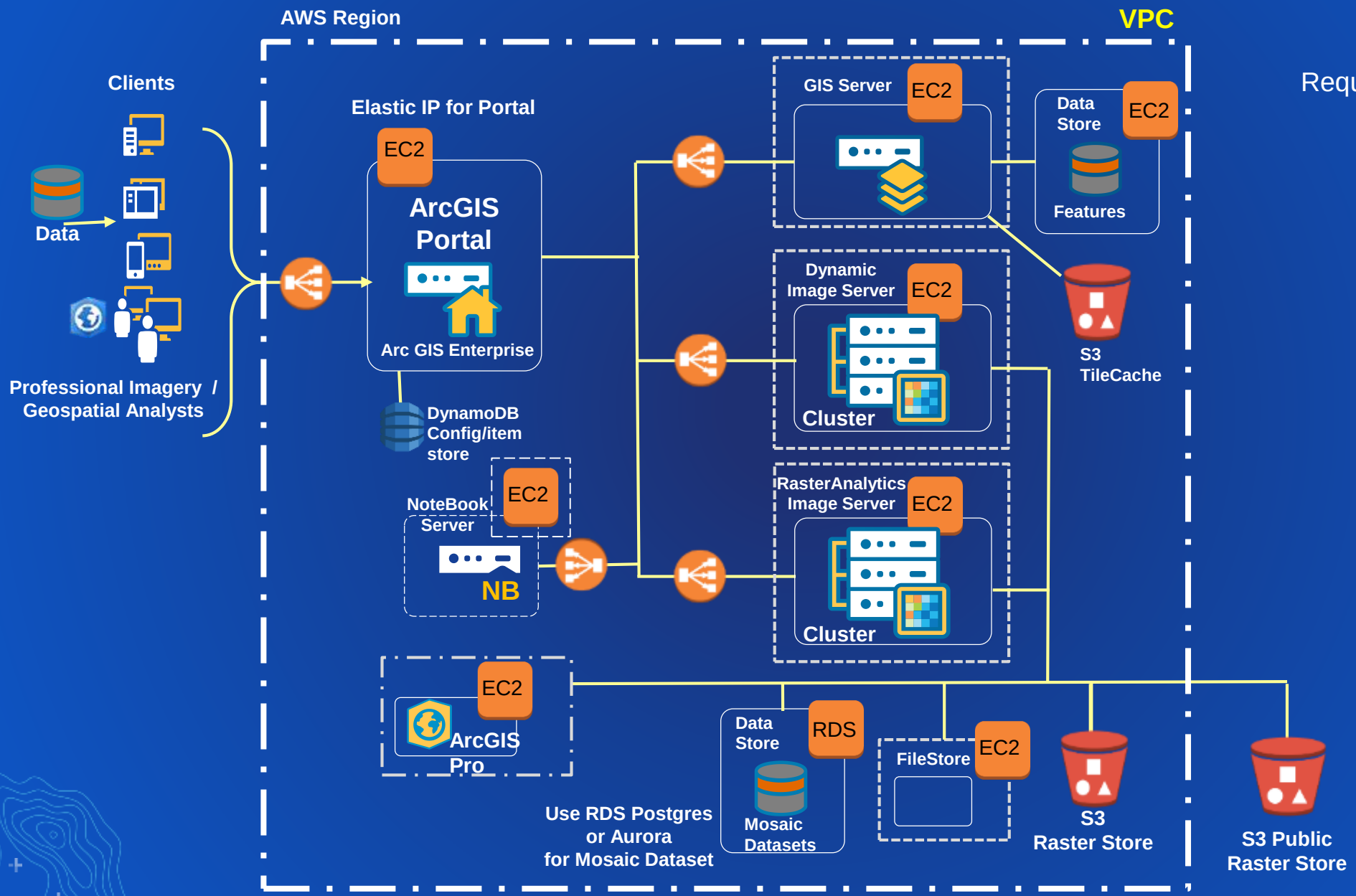
Publish to ArcGIS Image Server

Architecture & Implementation

Supporting Slides

www.esriurl.com/UC2019ImgMgmtCloud

ArcGIS Enterprise with ArcGIS Image Server on the Cloud (AWS sample)



Required for Options 2 and 3

Prerequisites – Before deployment

AWS Example, Azure is similar

- **Amazon account with full access to EC2 and other resources.**
- **If you are using ArcGIS Amazon Image (AMI) for the first time accept terms in the AWS Marketplace.**
- **A valid domain name for your site.**
- **A TLS (SSL) certificate for your domain, obtained from a certifying authority.**
- **An Elastic IP address that you will associate with the EC2 instance; you must map your domain name to the Elastic IP address.**
- **License file for Portal, ArcGIS server, Image server**
- **If the amazon account is new we recommend creating the below things**
  - **VPC, ElasticIP, KeyPair. Elastic Load Balancer, S3 Buckets and Configure the security groups for SSH and RDP access**
- **Also note down the VPC id, SiteEIPAllocationID for the elasticIP, keypair**

https://enterprise.arcgis.com/en/server/latest/cloud/amazon/introduction-to-aws-terminology.htm

ArcGIS Enterprise + Image Server Deployment

- **Enterprise**
  - **M5d.2xlarge (1 instance)**
  - **Cloud Formation Template : http://enterprise.arcgis.com/en/get-started/latest/windows/base-arcgis-enterprise-deployment.htm**
  - **Template : https://arcgisstore1071.s3.amazonaws.com/11595/templates/arcgis-allinone-windows.template**
  - **Readme : https://arcgisstore1071.s3.amazonaws.com/11595/docs/ReadmeAllInOneVPC.html**
  - **Image Server (Image Hosting)**
  - **M5d.2xlarge (2 instance) (in autoscaling mode) (for image server stack)**
  - **db.r4.xlarge (RDS) for high 20gb+ Mosaic datasize, or db.m4.xlarge**
  - **You will need a ELB, you can use the below Cloudformation template to create an ELB**
    - **Template https://arcgisstore1071.s3.amazonaws.com/11595/templates/arcgis-geoevent-server-elb.template**
    - **Then map the ELB name to the required domain name**
  - **Obtain following: Domain Name, SSL certificate, Image server License File (note you edit the file to mention your name and details)**
  - **If using RDS select DBengine and select postgres for it.**
  - **Template : https://s3.amazonaws.com/arcgisstore1071/11595/templates/arcgis-server-windows.template**
  - **ReadMe : https://s3.amazonaws.com/arcgisstore1071/11595/docs/ReadmeHAServerVPC.html**
  - **Image Server (Raster Analytics)** **(Separate site, Separate server function)**
  - **Similar to Image server**
  - **No need to setup RDS, use the above one.**

AWS Options to set up ArcGIS Enterprise

The below link gives an idea on what is ArcGIS Enterprise on AWS

https://enterprise.arcgis.com/en/server/latest/cloud/amazon/what-is-arcgis-server-on-aws.htm

- **Using AWS Cloud Formation Templates**
  - https://enterprise.arcgis.com/en/server/latest/cloud/amazon/deploy-arcgis-using-cloudformation.htm
  - https://enterprise.arcgis.com/en/server/latest/cloud/amazon/aws-cloud-formation-and-arcgis-server.htm
- **Using ArcGIS Server Cloud Builder**
  - https://enterprise.arcgis.com/en/server/latest/cloud/amazon/arcgis-enterprise-cloud-builder-cli-for-aws.htm
  - https://enterprise.arcgis.com/en/server/latest/cloud/amazon/arcgis-server-architectures-on-aws.htm#ESRI_SECTION1_04B5C34B31D54B8AB29AE14DD0F30F2F
- **Using AWS management Console (setup Manually)**
  - https://enterprise.arcgis.com/en/server/latest/cloud/amazon/use-aws-management-console-with-arcgis-server.htm
  - https://enterprise.arcgis.com/en/server/latest/cloud/amazon/arcgis-server-architectures-on-aws.htm#ESRI_SECTION1_E0C2D729F89048FF87A93D2114486A66

Type	Advantages	Disadvantage
CloudFormation Templates	Easy for Automation, Standard from Amazon, have better control on working with various AWS services, Easy to setup	Need Better understanding on how AWS services works, It's for more advance users. Managing can become tricky in cases
ArcGIS Server Cloud Builder	Easy to setup, easy to start and stop the site, can templatize	Difficult to integrate with other AWS services, need Secret Keys, advance customization might be difficult
Management Console (Manual)	Full control on how the system works	Can be used only by advance users. Needs to setup all components and connect them. Need to be proficient in the ArcGIS Enterprise and AWS

Azure Options to set up ArcGIS Enterprise

The below link gives an idea on what is ArcGIS Enterprise on Azure

https://enterprise.arcgis.com/en/server/latest/cloud/azure/overview-arcgis-server-on-microsoft-azure.htm

- **Using ArcGIS Server Cloud Builder**
  - **https://enterprise.arcgis.com/en/server/latest/cloud/azure/install-azure-cloud-builder.htm**
  - **https://enterprise.arcgis.com/en/server/latest/cloud/azure/deploy-web-gis-on-azure.htm**
  - **https://enterprise.arcgis.com/en/server/latest/cloud/azure/esri-images-on-azure-marketplace.htm**

ArcGIS Image Server stand alone on the Cloud (AWS sample)

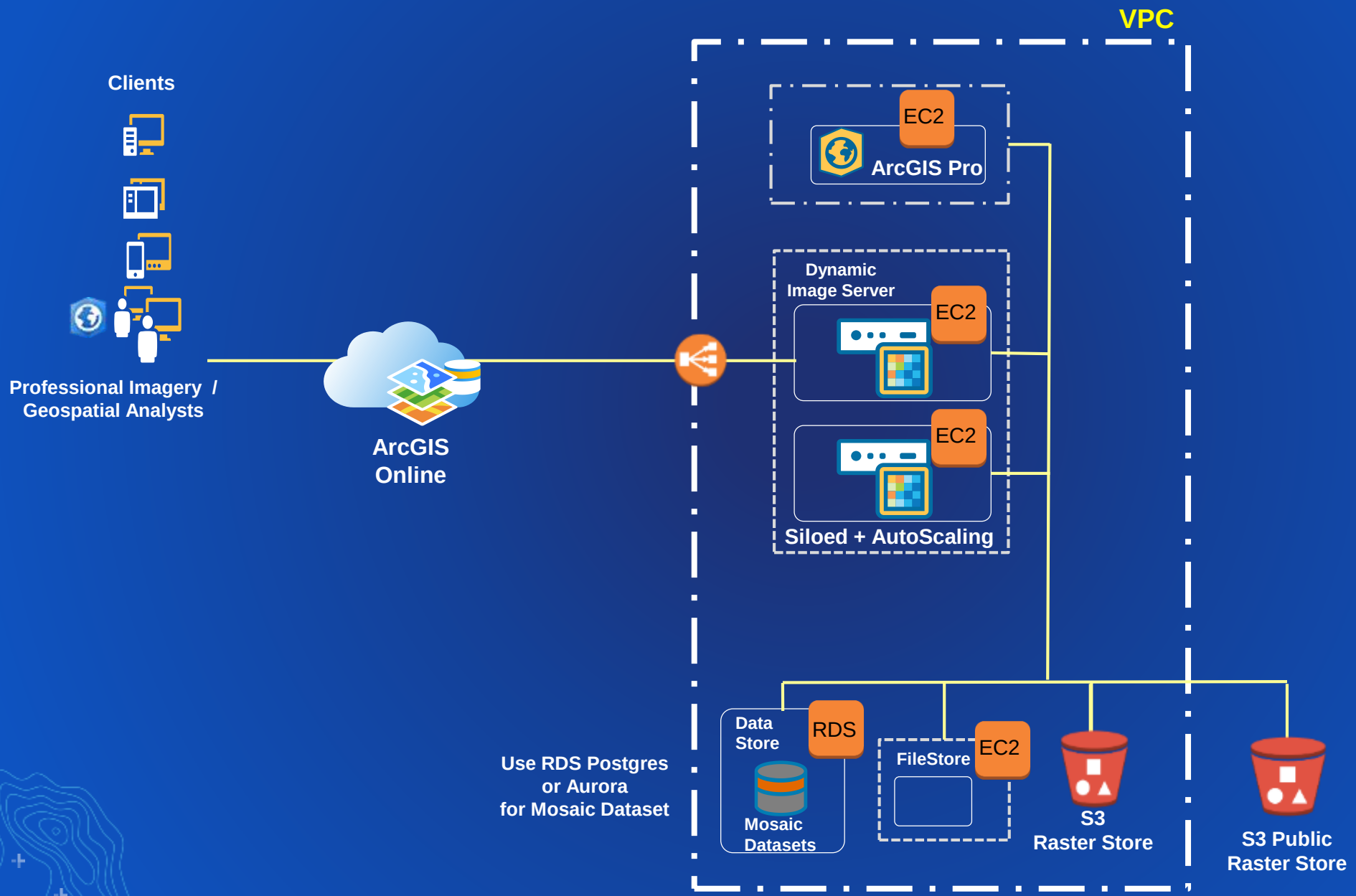


Image Server Deployment

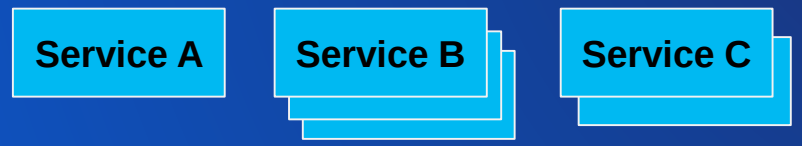
- **Image Server**
  - **M5d.2xlarge (1 instance) (in autoscaling mode, siloed)**
  - **db.r4.xlarge (RDS) for high 20gb+ Mosaic dataset, or db.m4.xlarge**
  - **You will need a ELB, you can use the below Cloudformation template to create an ELB**
    - **Template https://enterprise.arcgis.com/en/server/latest/cloud/amazon/deploy-arcgis-using-cloudformation.htm**
    - **Then map the ELB name to the required domain name**
  - **Obtain following: Domain Name, SSL certificate, Image server License File (note you edit the file to mention your name and details)**
  - **If using RDS select DBengine and select postgres for it.**

Dedicated vs Shared Instances

Dedicated Instances

- Traditional Instances
- Each Image Services has defined pool of SOC
- Better for reduced number of high load services

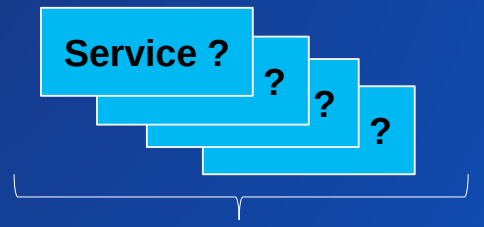
- Publish using Pro/ArcMap or Server Admin API



Shared Instances

- Pool of SOCs used for all Hosted Image Services (inc RA output)
- Used by all Hosted Imagery
- Can be defined using Admin API
- Note there is warm up time for Mosaic Datasets (about 12seconds)
- Best used for large number of services

- Publish using Pro or Server Admin API



File vs Enterprise GeoDatabase

- File GeoDatabase
  - OK for smaller mosaic datasets
  - Good for local mosaic datasets
  - Very Chatty – Not good to store on a shared drive

- In Cloud
  - Best to use AWS RDS (Aurora, PostgreSQL, SQL Server) or Azure (SQL Server, PostgreSQL)
  - Typically do not use portal Enterprise geodatabase (security/setup/portal load)
  - Can use FileGeodatabse on FileShare, but does not scale well
  - Alternative use FileGeodatabse, but copy from S3 to Ephemeral drive (simpler in Siloed)

Cloud Storage Options

- **File Storage**
  - **SMB access**
  - **Not Cloud Native**
  - **OK for smaller dataset, but does not scale well**
  - **Good for caching, if connected to directly server (eg Ephemeral)**

- **Object Storage - S3 / Azure Blob**
  - **Shared Access**
  - **REST Based (Http:)**
  - **Nearly unlimited size**
  - **Higher Latency/Higher throughput**

Optimum is to use Cloud Storage for Large images with caching local on ephemeral disk

Using Cloud Storage

- **Native** (directly use S3/BlobStore)
  - Create using Cloud Provider Console. Use AIM Role or Profile on machines
  - Use \\VSI or Raster Proxies
  - Only single secured account per machine (Credential or IAM Role)
  - Can use Canonical IDs for access control
  - More control (policies/rules), Read Only
- **ACS** (Cloud storage connection file)
  - Create in Pro (embed credentials)
  - Supports Credentials or IAM Role
  - Use in Pro, Add Rasters to Mosaic Dataset
  - Publish to Server (credential passed in Mosaic Dataset)
  - Supports multiple accounts, Read Only
- **Cloud Raster Store**
  - Create with ArcGIS Server Manager
  - Used to store Imagery Hosting and output from Raster Analytics (Raster Store)
  - Read/Write

IKONOS, QuickBird, GeoEye-1, WordView-1-4
GF-1 PMS, GF-1 WFV, GF-2 PMS
Landsat 1-5 MSS, 4-5 TM, 7 ETM+, 8
Pleiades-1, SPOT 5,6,7
KOMPSAT-2, KOMPSAT-3
Sentinel-2
DubaiSat-2
ZY3-SASMAC

Cloud Native Security – Many Options

- Public – No security
- Public - No List - Obfuscate – Public but hide URLs
- Requestor Pays – Same as Public, but user must have account (who pays for Egress)
- IAM role (Identity&Access Management) / Role-Based Access control (RBAC)
  - Set permission by users
- Presigned URLs, Amazon Query String Request Authentication / SAS (Shared Access Signature)
  - Token based. Token has expiry
- Access Control List – File Level permissions (Difficult for large numbers of files)
- Bucket policies - Fine control. (E.g. can set access by canonical IDs, use xx*, IPAddress etc.)

Note: Also Check Cross Origin Resource Sharing (CROS) if data accessed directly by apps.

What Affects Performance

- Volume of data read
  - Only read what is required (tiling/pyramids)
  - ArcGIS will only read pixels required
  - Use of Footprints vs. NoData
- Process Efficiency
  - Process only pixels required
  - Concatenate functions to reduce I/O
  - Use raster functions to speed processing
- Latency
  - Reduce number of requests (especially duplicate)
  - Tile Level Caching
- Bandwidth
  - Reduce data volume through compression (but don’t increase CPU)
- Data structure
  - Ensure efficient data access

Raster Storage Formats & Compression

- Raw, Stripped TIF
  - Disadvantage : Sequential access
  - Compression : None
- NetCDF, GRIB,
  - Advantage : Multidimension support
  - Compression : Varied
- Tiled GeoTIFF
  - Advantage : Tiled Access, can include Pyramids
  - Compression :
    - None
    - Lossy JPEG (8bit & 12bit)
    - Lossless Deflate/LZW
- COG (cloud optimized Geotiff)

  Same as Tiled GeoTIF only index is move to front

- MRF
  - Advantage : Tiled Access
  - Compression
    - As GeoTIFF +
    - LERC – Controlled Lossy – Very Fast

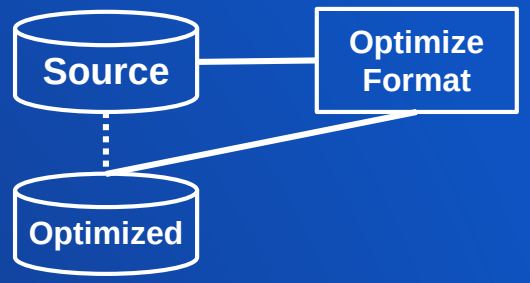
  Optimum for higher bit depth & categorical

  Best of compatibility with satellite scenes etc.

- CRF
  - Advantage : Tiled into Bundles, Multi-process write
  - Compression
    - None, LERC, JPEG

  Optimum for larger rasters

Structuring Imagery



- Correctly structuring imagery can have a massive effect on performance
- Many datasets directly from vendors are very generic (and slow to access)
- Can put high load on File System
- Often near impossible to use directly on Cloud storage

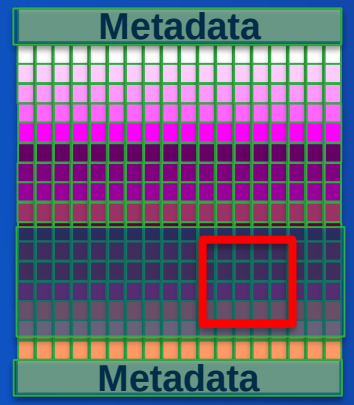
- Typically better to Optimize Imagery
  - Prior to Uploading to Cloud
  - As part of uploading process
  - Upload to Cloud and then Optimize
- Multiple Cloud Optimizations (Also works very well on-premises)

Raster Format Considerations / Optimization

- Tiling of imagery – Enables partial access
- Compression – Reduce storage and transfer – Weigh against additional compute requirements
- Data access complexity – Reduce subsequent requests
- Pyramids – Provide faster access as smaller scales

Slow

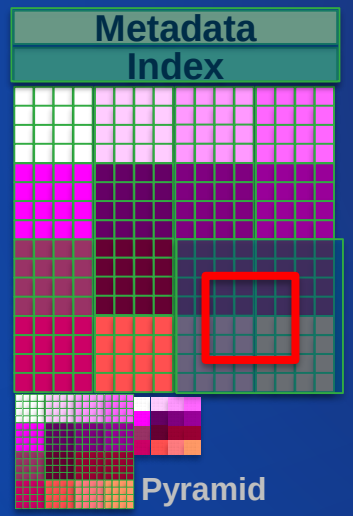
Striped TIF



Non optimum access

Good

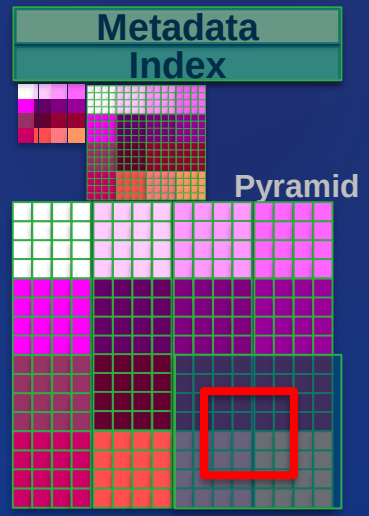
Tiled TIF



Enables partial access

Good

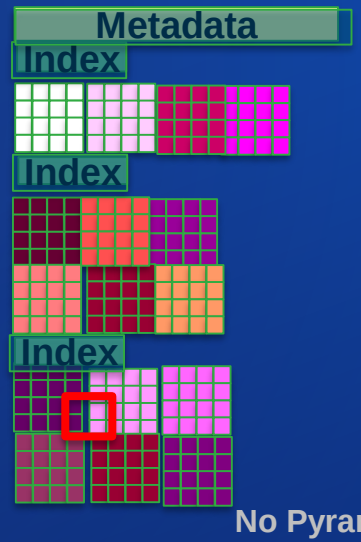
COG



Improved access to pyramids (often part of first request)

Not Good

NetCDF/HDF/GRIB (gridded)

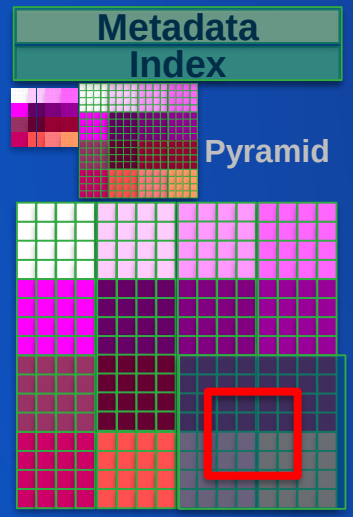


Raster Format Considerations / Optimization

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

Good

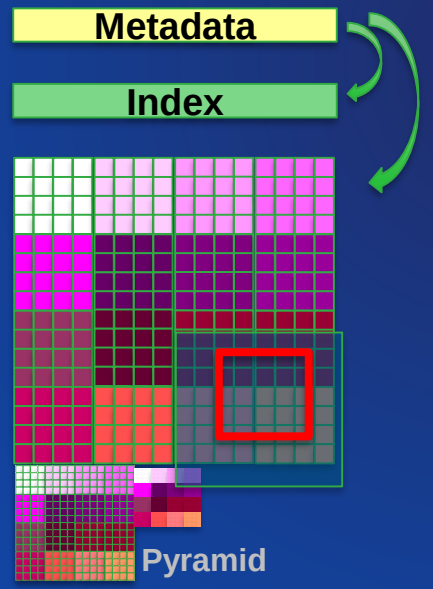
COG



Improved access to pyramids

Optimum

MRF



Separate files enable operating system caching

Files can be in separate locations (eg metadata and index on fast access)

CRF – Cloud Raster Format

- Similar to TPK, but multi-band, multi-bit depth
- Spit into bundles/packet each has set number of tiles. Each bundle has its own index
- Structure of tiles is inherent in the directory structure
- Supports various compression including LERC
- When accessed each bundle required is read and cached locally (managed)
- Advantage
  - Enables multi-processor write
  - Good for large rasters
- Disadvantage
  - Multiple files
  - Not good for many smaller rasters

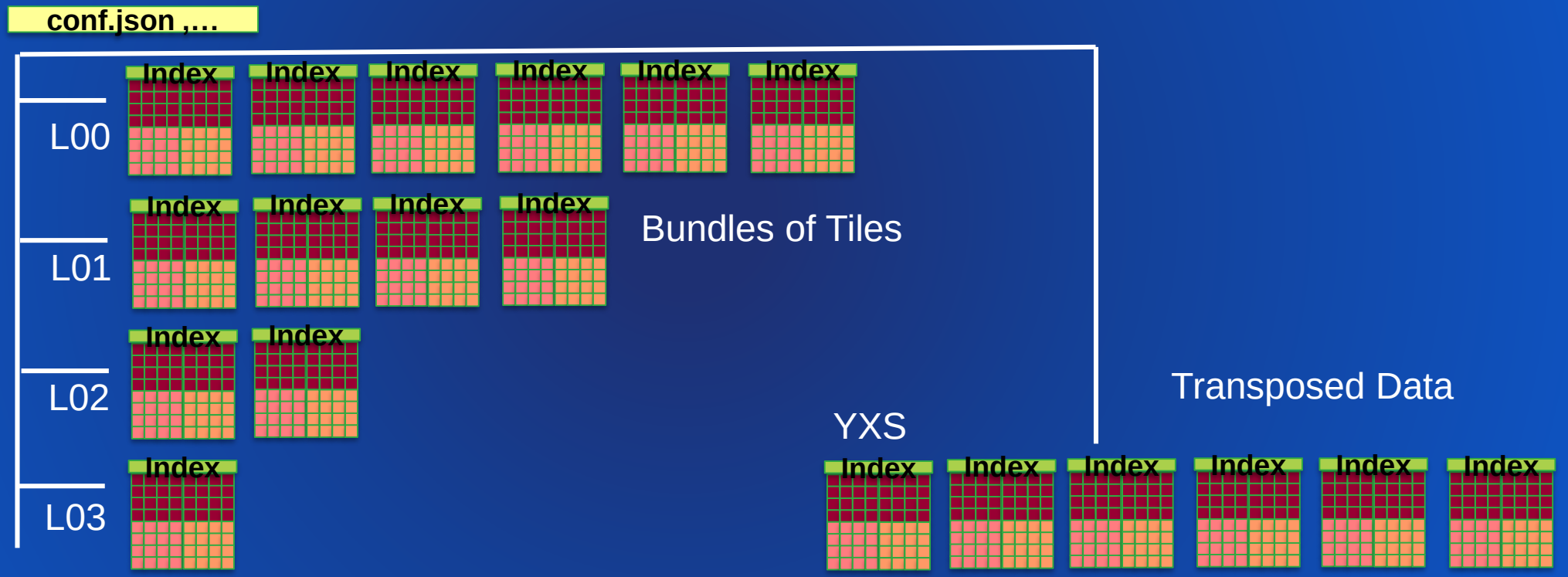
Simplicity of data structure enables both

MRF and CRF to be directly accessed through JavaScript etc

Transposed CRF

- **Extension to CRF (at V2.4) to optimally handle additional dimension**

GDAL has no suitable API for additional dimensions
CRF is not accessed through GDAL



Compression

- **Lossless**
  - **Deflate**
  - **LZW**
  - **LERC ***
  - **PNG**
  - **JPEG2000**
- **Lossy**
  - **JPEG (RGB / YCbCr)**
  - **JPEG2000**
- **Controlled Lossy**
  - **LERC**

	As a Factor of Tiled TIFF No Compression		
	Size	Time to Write	Time to Read Full
Tiled TIFF None	1.00	1.00	1.00
Tiled TIFF Deflate	0.67	2.34	0.96
Tiled TIFF LZW	0.58	5.18	0.98
Tiled TIFF LERC	0.53	1.69	1.02
MRF LERC	0.53	2.09	0.91
COG Deflate	0.61	8.48	1.07

The above are typical, Compression performance is dependent on source

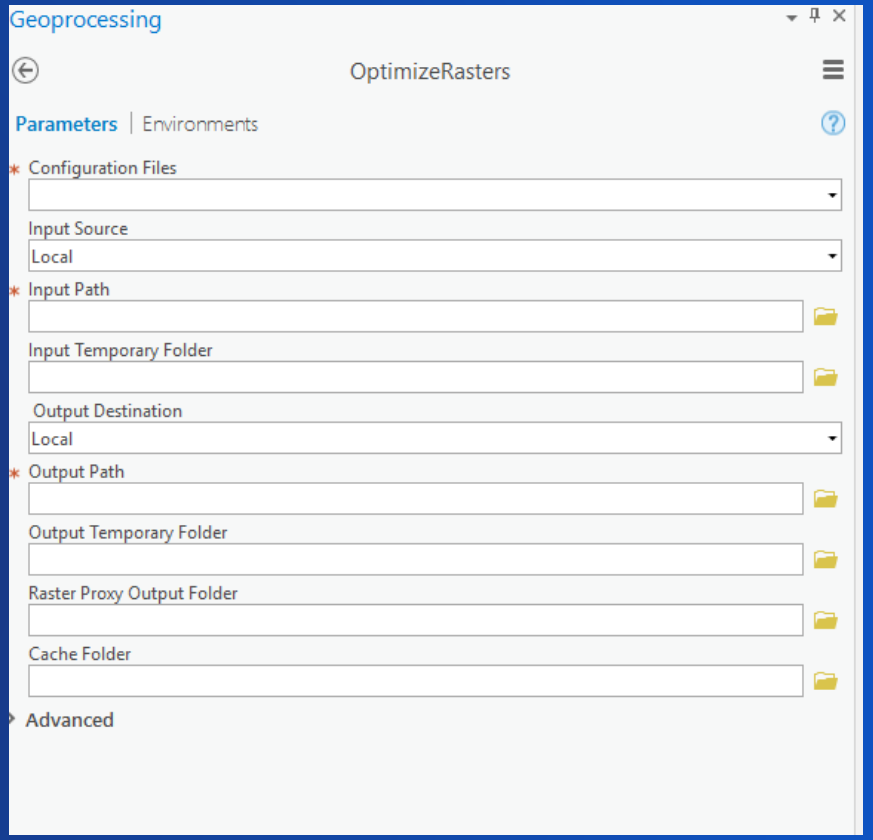
* LERC is Lossless when tolerance set lower than precision

Image Conversion

- Converting data options
  - ArcGIS Export – To TIF, CRF, MRF
  - Copy Raster (for CRF)
  - Use OptimizeRasters
    - https://github.com/Esri/OptimizeRasters
  - GDAL

Optimize Raster

- Converts Raster to optimal formats
- Transfer data to and from cloud storage (AWS, Azure, Google, Alibaba) or enterprise storage
- Runs with in ArcGIS (ArcGIS Pro or ArcMap) or as a standalone utility
- Supports for Windows and Linux
- Parallel Processing is supported by default
- Creates Raster Proxies
- Logging support
- Converts the data using the best practices (templates provided)
- …



https://github.com/Esri/OptimizeRasters

Getting data to the Cloud - How to transfer

- Upload using Portal (goes to raster store)
- CloudBerry, etc.
- Amazon cli Command or Console
- OptimizeRasters
- White Glove Services Snowball ($200/50TB), Snowmobile

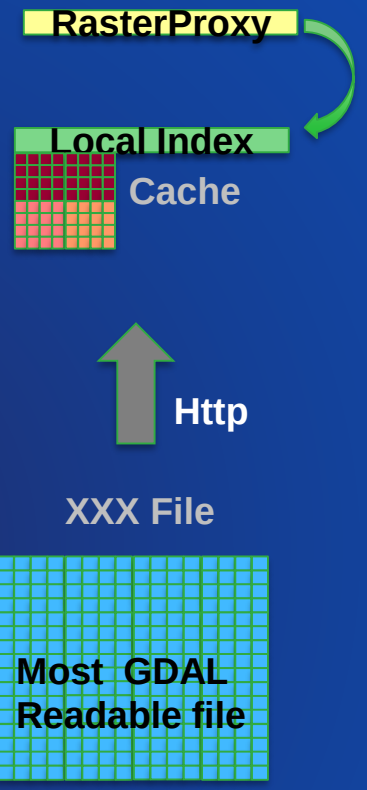
Accessing Imagery from Cloud Storage

- **Directly use \\VSICurl \\VSIS3 \\VSIAZ**
  - **As if local file. Multiple requests, No Caching.**
- **ACS**
  - **Advantage of multiple profiles**
  - **Caching Optional -** The pixel cache for a mosaic dataset can be generated by running the Add Rasters to Mosaic Dataset tool with the **Enable Pixel Cache** parameter checked. (Issues, Not recommended in current version)
  - **Managed Cache**
- **Raster Proxies** **(See Optimize Rasters)**
  - **Can be batch produced**
  - **Non Managed Cache (Manage on own)**
  - **Option to use as local files and then embed**
  - **Further optimize by setting environment variables**

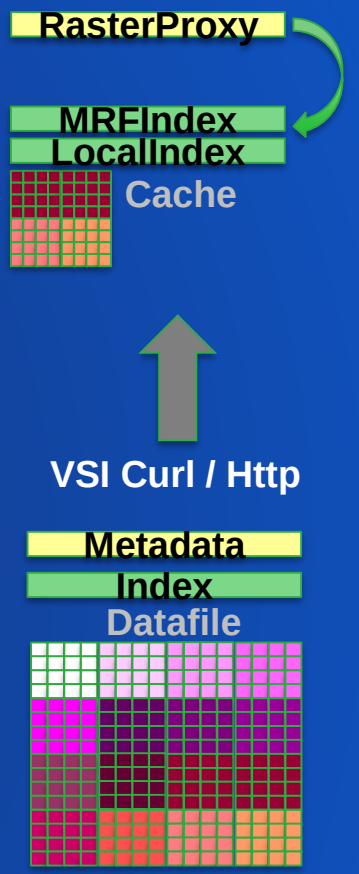
Raster Proxies

- Work with most GDAL Readable formats
- More optimal with MRF
- 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



When accessed,
tiles + index to tiles
are stored locally.
Tiles are store either
uncompressed
or using LERC



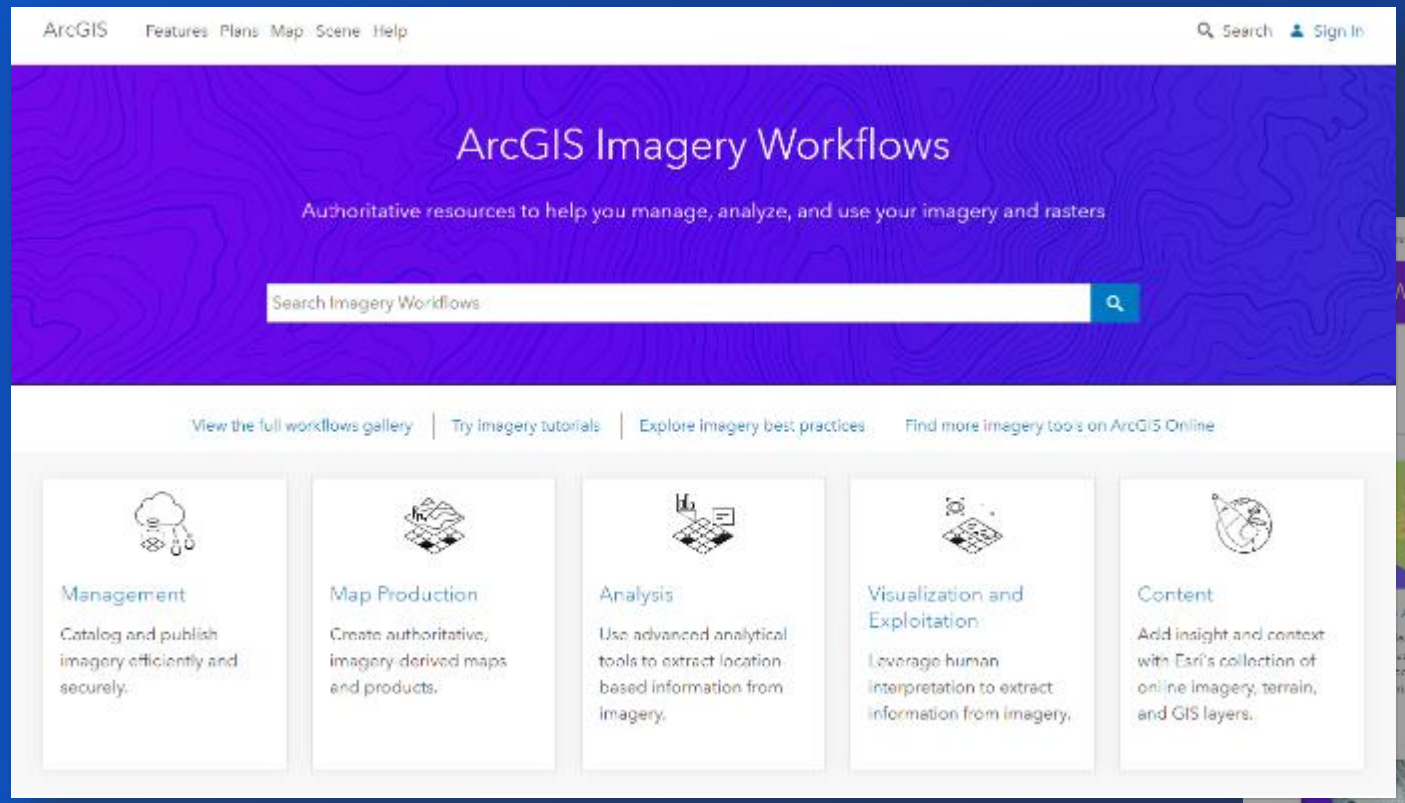
When accessed,
Tiles + index to tiles +
index to source tiles
are stored locally.
Tiles are original tiles.

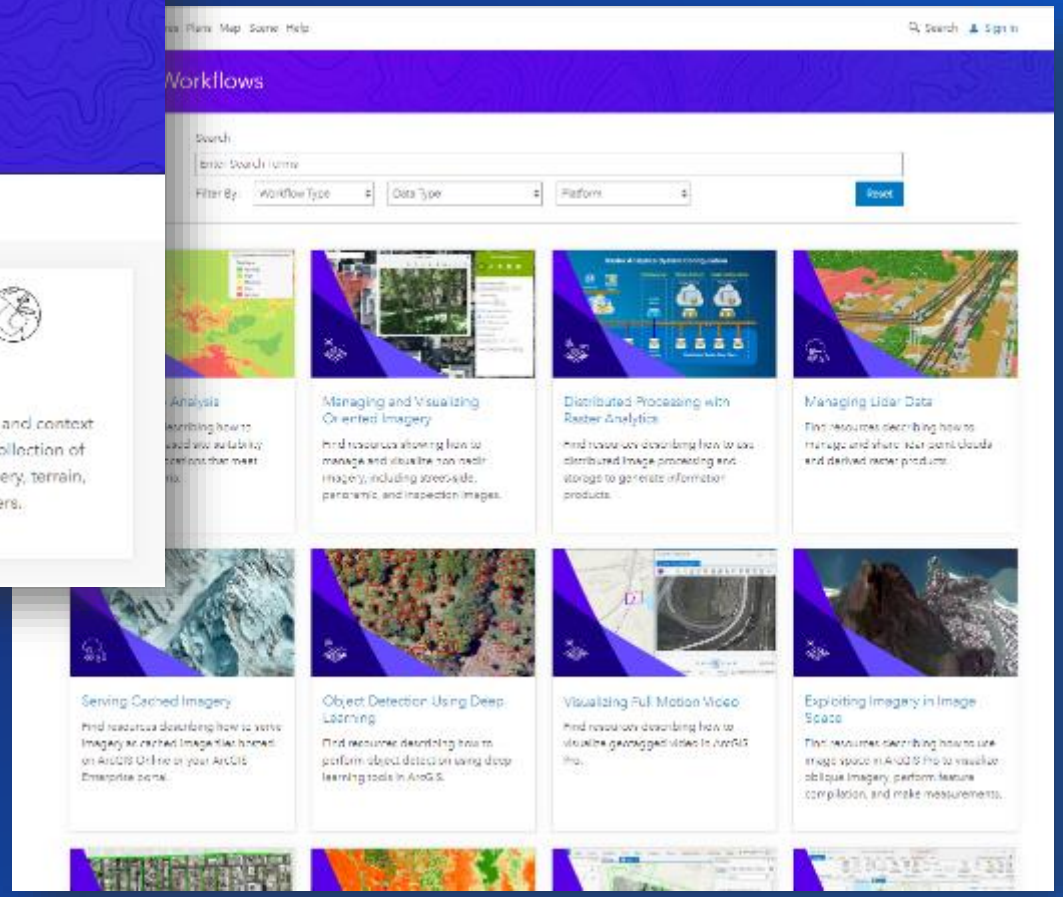
Referencing Rasters in a Mosaic Dataset

Where possible mosaic dataset should not reference files on disk

- **Options**
  - **Files share – Same path for authoring & server**
  - **Create ASC, Add data to MD and publish (don’t use caching yet)**
  - **File Share to Raster Proxies (which have linked to cloud storage)**
  - **Create using Raster Proxies, then embed into MD before Publishing (MDTools)**
  - **Create directly with \\VSI in paths (use table Raster)**
  - **Create directly with raster proxies embedded (user raster proxy as table in Optimize Rasters)**
  - **Use Raster Proxies then embed into mosaic dataset using MDTools (Part of MDCS)**

Imagery Workflows





www.esriurl.com/imageryworkflows

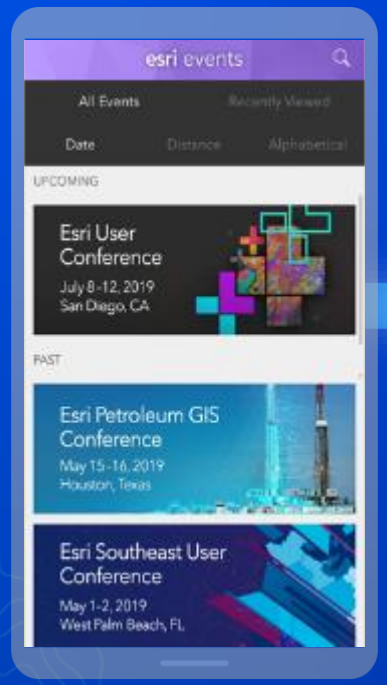
Questions?

Supporting Slides

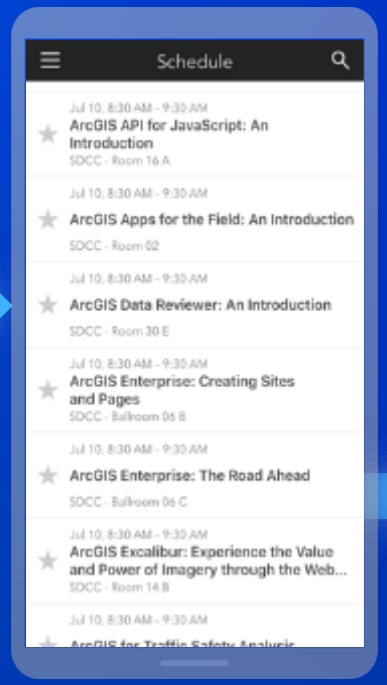
www.esriurl.com/UC2019ImgMgmtCloud

Please Share Your Feedback in the App

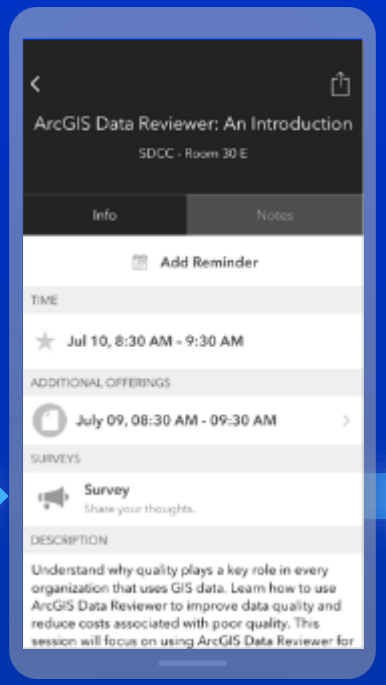
Download the Esri Events app and find your event



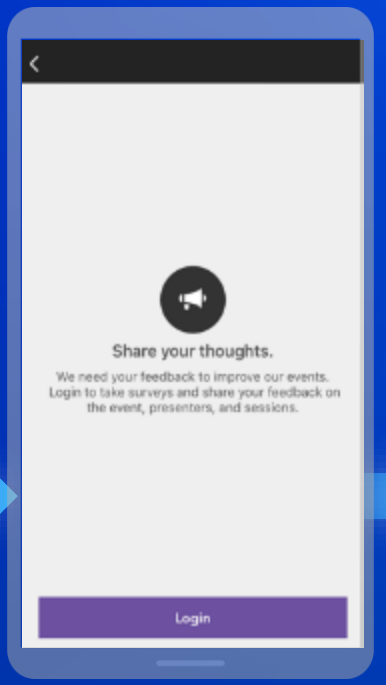
Select the session you attended



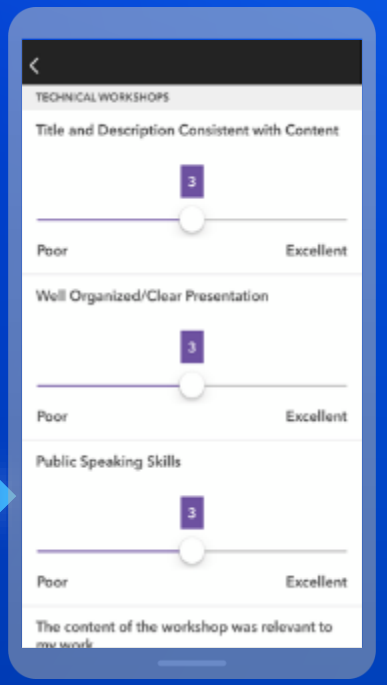
Scroll down to "Survey"



Log in to access the survey



Complete the survey and select "Submit"



Demo Title

Presenter(s)