Storage

- **Tools**
- **Storage on Azure and AWS**

Tools

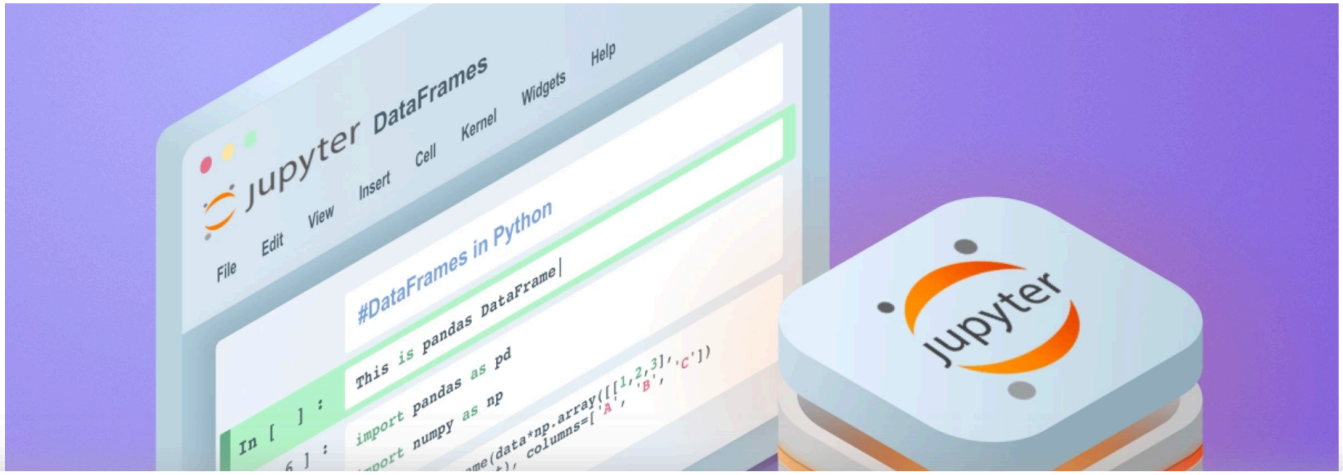
- Python (Anaconda), PHP, SDK's for Azure, AWS
- Jupyter Notebooks
  - DataCamp tutorial link on CourseLink under Software Links
- GitHub: SciEngCloud.github.io
  - https://github.com/SciEngCloud/SciEngCloud.github.io
- Docker

Karlijn Willems
November 12th, 2019

MUST READ PYTHON +2

Jupyter Notebook Tutorial: The Definitive Guide

This tutorial explains how to install, run, and use Jupyter Notebooks for data science, including tips, best practices, and examples.



github.com/SciEngCloud/SciEngCloud.github.io

Search or jump to... / Pull requests Issues Marketplace Explore

SciEngCloud / SciEngCloud.github.io

Watch 4 Star 7 Fork 5

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights

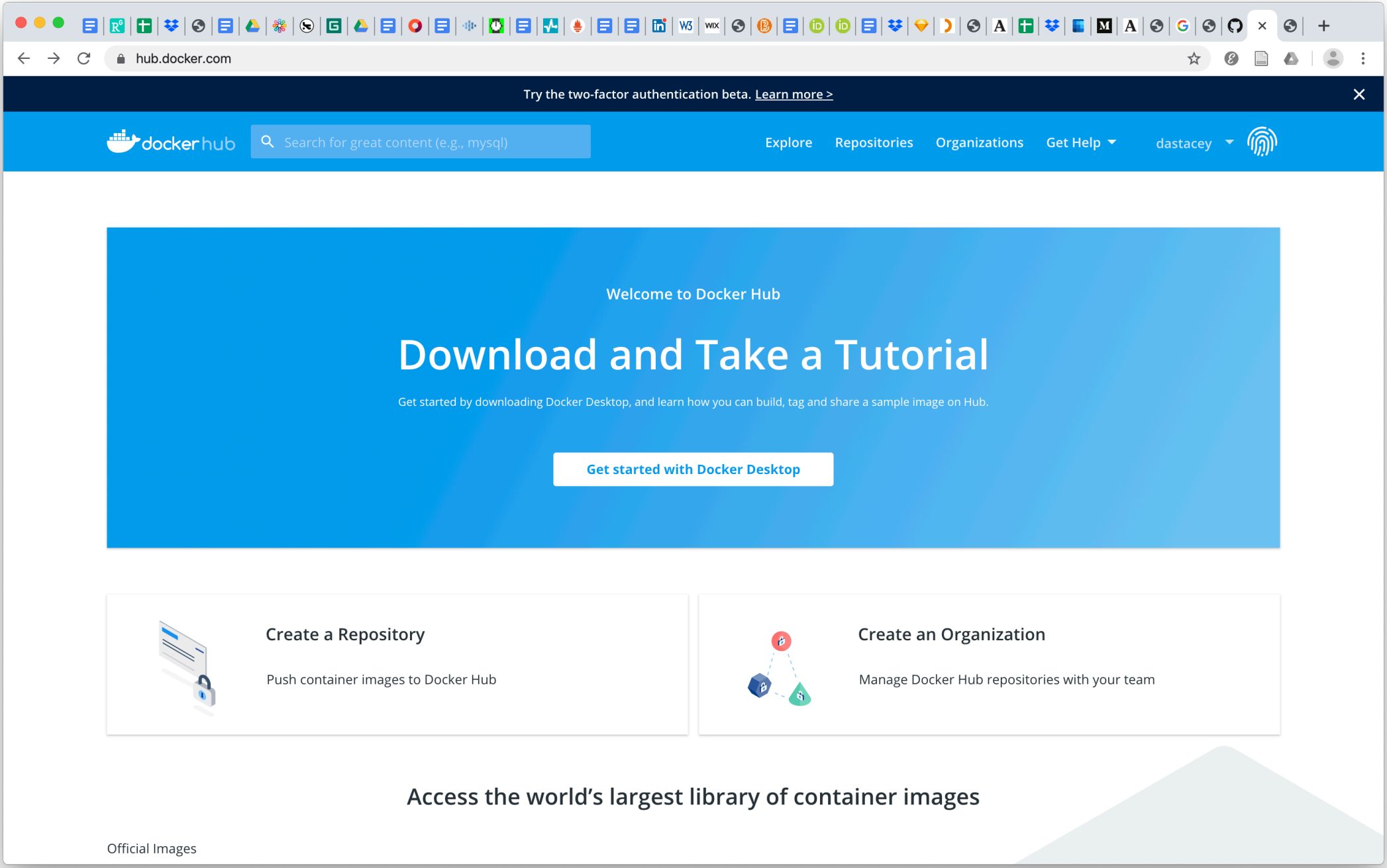
Cloud Computing for Science and Engineering web site

103 commits **1** branch **0** packages **0** releases **2** contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

dbgannon added fixed version of mxnet.ipynb — Latest commit `269f09d` on Dec 7, 2017

Docker-demo	more notebook work	3 years ago
arxiv_data	This is the data set for the arxiv experiment set.	2 years ago
aws-hpc-cluster	shorten urls	3 years ago
aws-ml-container	improvements to ecs demo	3 years ago
datadir	adding datadir for azure.ipynb	3 years ago
gcloud-container	fixing datafile in cloud	3 years ago
kinesis-spark-AoT	improved kinesis example	3 years ago
notebooks	added fixed version of mxnet.ipynb	2 years ago
sc-tutorial/final-pdfs	added sc-tutorial	2 years ago
singularity	added singlarity	2 years ago
README.md	updated readme	2 years ago
ServerlessComputing.pdf	added the serverless report	2 years ago
datadir.tar	fixing rm mistake	3 years ago
lectures_and_exercises.zip	added lectures and exercies and funcion ipynb	2 years ago


hub.docker.com
Try the two-factor authentication beta. Learn more >
dockerhub
Search for great content (e.g., mysql)
Explore
Repositories
Organizations
Get Help
dastacey
Welcome to Docker Hub
Download and Take a Tutorial
Get started by downloading Docker Desktop, and learn how you can build, tag and share a sample image on Hub.
Get started with Docker Desktop
Create a Repository
Push container images to Docker Hub
Create an Organization
Manage Docker Hub repositories with your team
Access the world's largest library of container images
Official Images

hub.docker.com/repository/docker/dastacey/4010-repo

Try the two-factor authentication beta. Learn more >

docker hub | Search for great content (e.g., mysql) | Explore | Repositories | Organizations | Get Help | dastacey

Repositories / dastacey / 4010-repo

Using 1 of 1 private repositories. Get more

General | Tags | Builds | Timeline | Collaborators | Webhooks | Settings

dastacey / 4010-repo

Containers for CIS4010

Last pushed: a few seconds ago

Docker commands

To push a new tag to this repository,

```
docker push dastacey/4010-repo:tagname
```

Tags

This repository contains 1 tag(s).

latest		an hour ago

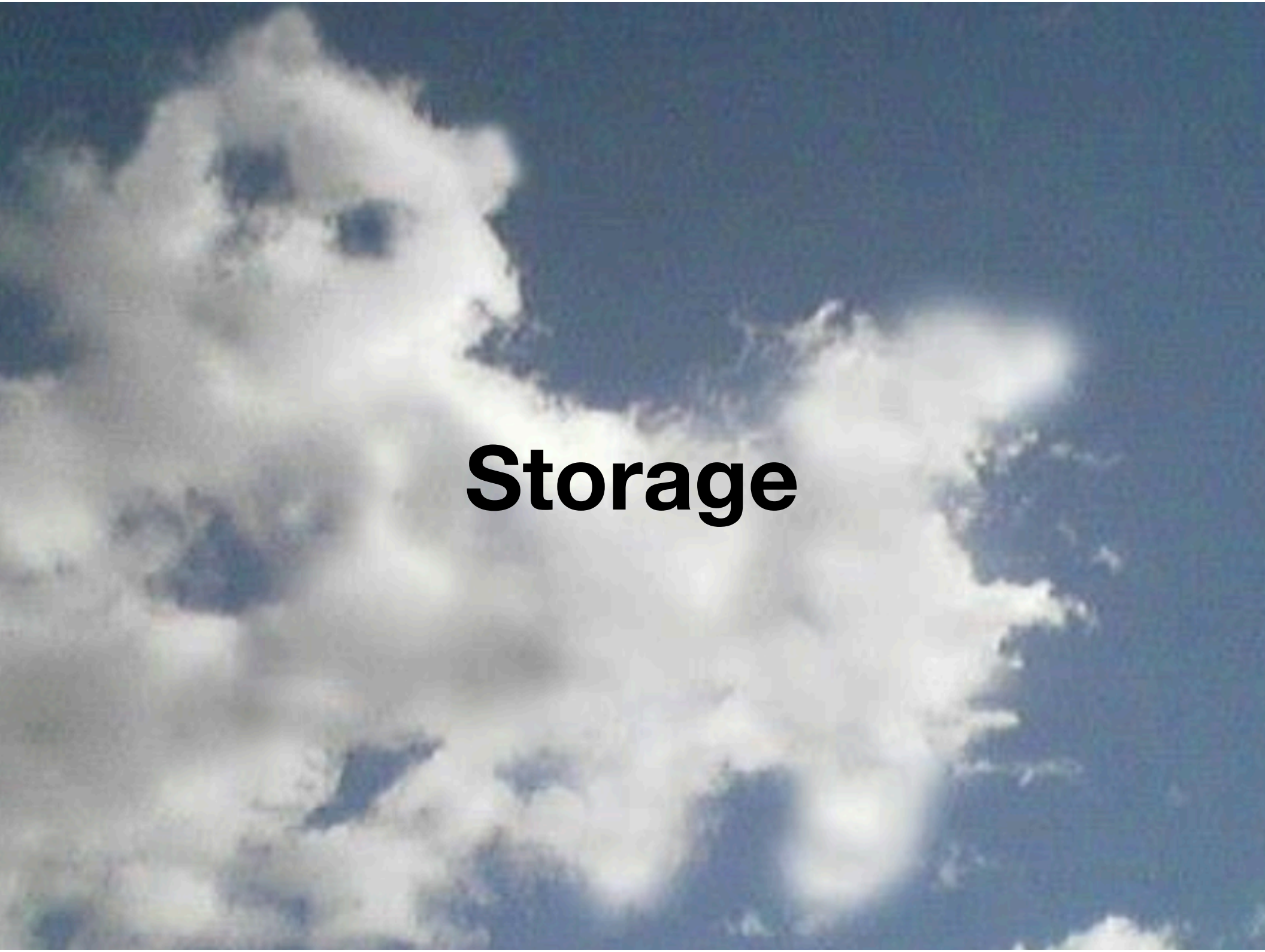
See all

Recent builds

Link a source provider and run a build to see build results here.

Readme

Repository description is empty. Click here to edit.


Storage

Types of Cloud Storage

- File systems
- Object stores
- Databases
  - relational, NoSQL, graph
- Data Warehouses
  - designed for running analytics and queries over large collections of data

File Systems

- Standard API for the Unix-derived file systems is called the Portable Operating System Interface (POSIX)
- File systems allow us to create, read, write, and delete files located within directories using command line tools, graphical user interfaces, or APIs.
- **Advantages**:
  - direct use of many existing programs without modification
  - provides a straightforward mechanism for representing hierarchical relationships among data—directories and files

File Systems

- **Disadvantages**:
  - provides no support for enforcing conventions concerning the representation of data elements and their relationships
  - the rigid hierarchical organization enforced by a file system often does not match the relationships that you want to capture
  - the file system model also has problems from a scalability perspective; the need to maintain consistency as multiple processes read and write can lead to bottlenecks

Object Stores

- Store unstructured binary objects or **blobs** (binary large objects)
- Eliminate hierarchy and **forbids updates to objects** once created - only allows deletes or replaces (if versioning allowed)
- Support a two-level folder-file hierarchy that allows for the creation of object **containers**, each of which can hold zero or more objects
- Each object is identified by a **unique identifier** and can have various metadata associated with it

Object Stores

- **Advantages**:
  - simplicity, performance, and reliability
  - since objects cannot be modified once created makes it easy to build highly **scalable** and **reliable** implementations
  - **replication without synchronization**

Object Stores

- **Limitations**:
  - provides little support for organizing data and no support for search: you must know an object's identifier to access it
  - cannot easily be mounted as a file system or accessed with existing tools in the ways that a file system can

Object Store Model

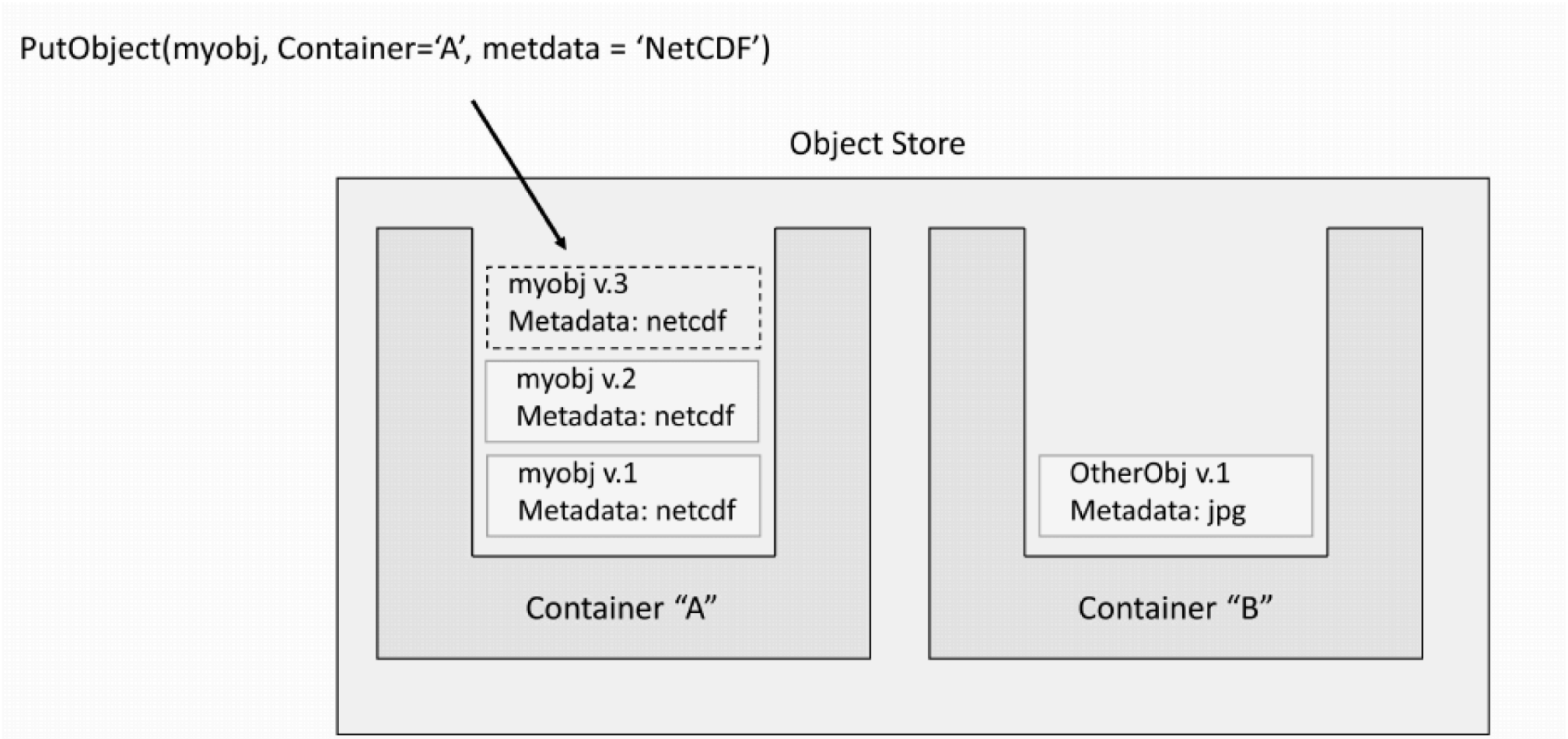


Figure 2.1: Object storage model with versioning. Each NetCDF file is stored in a separate container, and all versions of the same NetCDF file are stored in the same container.

Databases

- The use of a DBMS simplifies data management and manipulation and provides for
  - efficient querying and analysis
  - durable and reliable storage
  - scaling to large data sizes
  - validation of data formats
  - management of concurrent accesses

Types of Databases

- Relational databases
  - useful for searching data based on relationships among data items
  - MySQL and Postgres are available in cloud-hosted forms
  - cloud vendors offer specialized relational DBMSs that are designed to scale to particularly large data sizes
- NoSQL databases
- Graph databases
  - often graph databases are built on top of existing NoSQL databases

Relational Databases

- Support a relational algebra that provides a clear, mathematical meaning to the SQL language, facilitating efficient and correct implementations
- Support **ACID** semantics:
  - **A**tomicity (the entire transaction succeeds or fails)
  - **C**onsistency (the data collection is never left in an invalid or conflicting state)
  - **I**solation (concurrent transactions cannot interfere with each other)
  - **D**urability (once a transaction completes, system failures cannot invalidate the result)

NoSQL Databases

- Motivations
  - scaling the quantities of data and number of users that can be supported
  - dealing with unstructured data not easily represented in tabular form
- **Key-Value** store can organize large numbers of records, each of which associates an arbitrary key with an arbitrary value
  - a document store permits text search on the stored values

NoSQL Databases

- NoSQL databases in the cloud are often distributed over multiple servers and also replicated over different data centres
- They often fail to satisfy all of the **ACID** properties
- **Consistency** is often replaced by *eventual consistency*, meaning that database state may be momentarily inconsistent across replicas

The CAP Theorem

- It is not possible to create a distributed system with all three of the following properties:
  - **Consistency** - all computers see the same data at the same time
  - **Availability** - every request receives a response about whether it succeeded or failed
  - **Partition tolerance** - the system continues to operate even if a network failure prevents computers from communicating
- DBMS designer must choose between *high consistency* or *high availability* for a particular system

I. Foster, R. Ghani, R. S. Jarmin, F. Kreuter, and J. I. Lane, editors. Big Data and Social Science: A Practical Guide to Methods and Tools. Taylor & Francis Group, 2016. See also http://www.bigdatasocialscience.com

Table 2.1: Storage as a service options from major public cloud vendors.

Model	Amazon	Google	Azure
Files	Elastic File System (EFS), Elastic Block Store (EBS)	Google Cloud attached file system	Azure File Storage
Objects	Simple Storage Service (S3)	Cloud Storage	Blob Storage Service
Relational	Relational Data Service (RDS), Aurora	Cloud SQL, Spanner	Azure SQL
NoSQL	DynamoDB, HBase	Cloud Datastore, Bigtable	Azure Tables, HBase
Graph	Titan	Cayley	Graph Engine
Warehouse analytics	Redshift	BigQuery	Data Lake

Amazon File System Storage

- **E**lastic **B**lock **S**tore (EBS) and **E**lastic **F**ile **S**ystem (EFS) services offer related but different services

- **EBS** is a device that you can mount onto a single Amazon **EC2** compute server instance at a time

  - low-latency access to data from a single EC2 instance

- **EFS** is a general-purpose file storage service that provides a file system interface, file system access semantics , and concurrently-accessible storage for *many* Amazon EC2 instances

- Both **EBS** and **EFS** can be accessed directly only by **EC2** instances (from inside the Amazon cloud)

Azure File System Storage

- Allows users to create file shares in the cloud that can be accessed by a special protocol, **SMB**, that allows Microsoft Windows VMs and Linux VMs to mount these file shares as standard parts of their file system
  - In the cloud and outside of the cloud

Amazon Object Storage

- Amazon's Simple Storage Service (**S3**) was its first cloud service.
- As of 2016, it reportedly contained trillions of objects in billions of containers
- S3 containers are called **buckets**
- S3 is a classic object store

Azure Object Store

- **Azure Blob** storage service is concerned with highly reliable storage of unstructured objects (**blobs**)
  - similar to Amazon's S3
- Azure Blob storage has tiered storage and pricing
  - **hot** for frequently accessed data
  - **cool** for data accessed less often
- Azure Storage Explorer
  - *https://azure.microsoft.com/en-us/features/storage-explorer/*