Porting Research Pipelines into Clouds

Architectural considerations (2/3)

David Yuan, Ph.D.

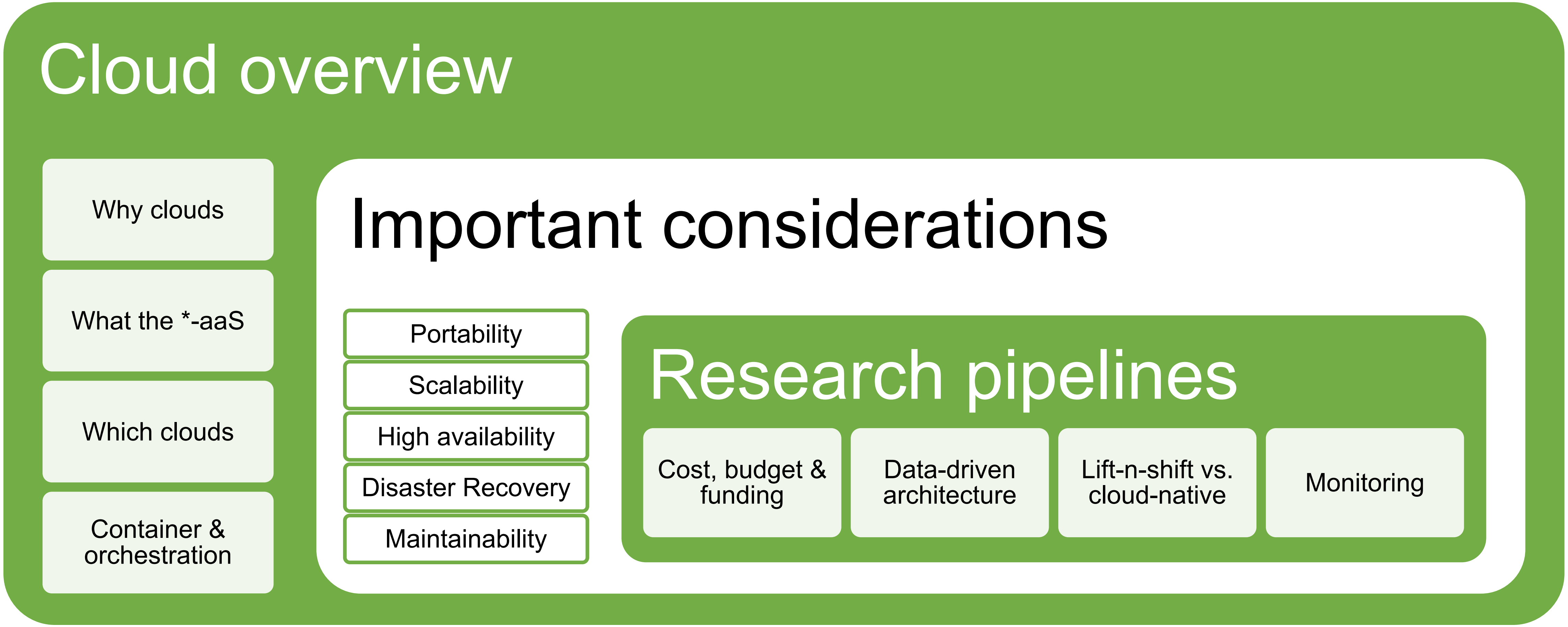
Cloud Bioinformatics Application Architect

Technology and Science Integration

European Bioinformatics Institute, EMBL

EMBL-EBI

Porting into clouds



Cloud overview

- Why clouds
- What the *-aaS
- Which clouds
- Container & orchestration

Important considerations

- Portability
- Scalability
- High availability
- Disaster Recovery
- Maintainability

Research pipelines

- Cost, budget & funding
- Data-driven architecture
- Lift-n-shift vs. cloud-native
- Monitoring

Important considerations

Portability

- Poor portability between clouds
- Docker & K8S: De-facto standards
- Major decision to be made as early as possible

Scalability

High availability

Disaster Recovery

Maintainability

Important considerations

Portability

- Poor portability between clouds
- Docker & K8S: De-facto standards
- Major decision to be made as early as possible

Scalability

- Cloud scaling up and scaling down limited by hardware
- Docker & K8S: vertical scaling, horizontal scaling, autoscaling across hardware boundary
- Storage IO often being the bottleneck

High availability

Disaster Recovery

Maintainability

Important considerations

Portability

- Poor portability between clouds
- Docker & K8S: De-facto standards
- Major decision to be made as early as possible

Scalability

- Cloud scaling up and scaling down limited by hardware
- Docker & K8S: vertical scaling, horizontal scaling, autoscaling across hardware boundary
- Storage IO often being the bottleneck

High availability

- Cloud better than traditional DCs
- K8S: ReplicaSet & StatefulSet across hardware boundary
- Shared POSIX filesystems: single point of failure

Disaster Recovery

Maintainability

Important considerations

Portability

- Poor portability between clouds
- Docker & K8S: De-facto standards
- Major decision to be made as early as possible

Scalability

- Cloud scaling up and scaling down limited by hardware
- Docker & K8S: vertical scaling, horizontal scaling, autoscaling across hardware boundary
- Storage IO often being the bottleneck

High availability

- Cloud better than traditional DCs
- K8S: ReplicaSet & StatefulSet across hardware boundary
- Shared POSIX filesystems: single point of failure

Disaster Recovery

- Double or triple redundancy: resilient to disaster
- Infrastructure-as-code: faster recovery
- K8S: clear separation of compute, configuration and data
- Shared POSIX filesystems: single point of failure

Maintainability

Important considerations

Portability

- Poor portability between clouds
- Docker & K8S: De-facto standards
- Major decision to be made as early as possible

Scalability

- Cloud scaling up and scaling down limited by hardware
- Docker & K8S: vertical scaling, horizontal scaling, autoscaling across hardware boundary
- Storage IO often being the bottleneck

High availability

- Cloud better than traditional DCs
- K8S: ReplicaSet & StatefulSet across hardware boundary
- Shared POSIX filesystems: single point of failure

Disaster Recovery

- Double or triple redundancy: resilient to disaster
- Infrastructure-as-code: faster recovery
- K8S: clear separation of compute, configuration and data
- Shared POSIX filesystems: single point of failure

Maintainability

- Cloud usually no scheduled downtime
- K8S: eliminating scheduled downtime
  - Rolling up upgrade K8S nodes, underlying hardware, application
  - Auto-recovery built in