Imperial College London

RCS RESEARCH COMPUTING SERVICE
Gateway to Discovery

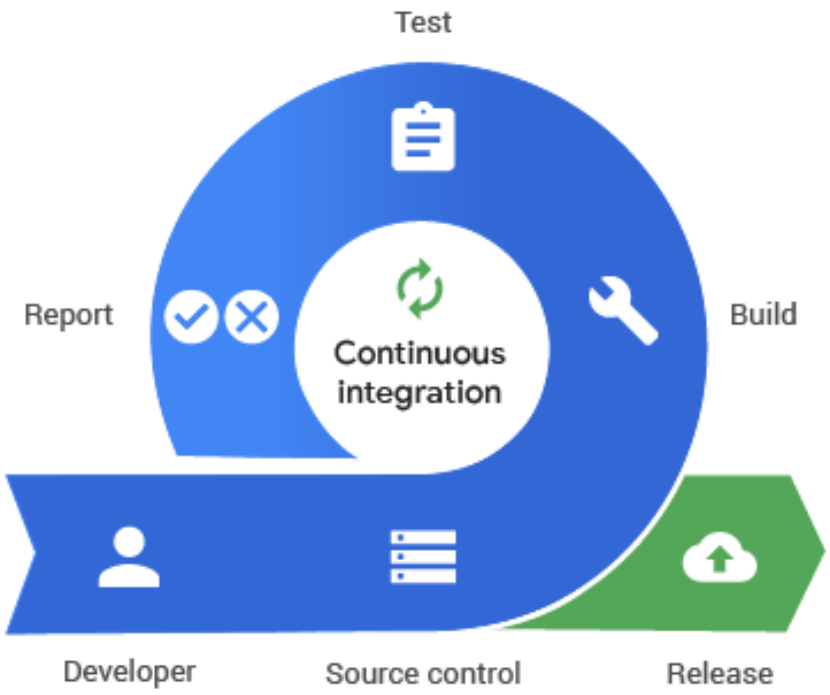
Continuous Integration for Research Software

Dr Christopher Cave-Ayland

c.cave-ayland@imperial.ac.uk
@ImperialRSE
Imperial College Research Computing Service, DOI: 10.14469/hpc/2232

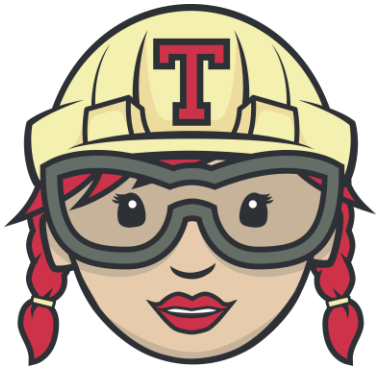
What is Continuous Integration

- Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project – Atlassian



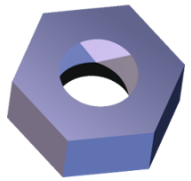
https://cloud.google.com/solutions/continuous-integration/

Imperial College London



RCS RESEARCH COMPUTING SERVICE
Gateway to Discovery

Lots of options!



Bamboo

Azure Pipelines

circleci

ANVIL
Research Software Testing Platform

CI/CD

Google Cloud

GitHub Actions

Bitbucket Pipelines

Jenkins

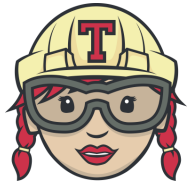
More usefully

Cloud hosted services (usually including compute environments)

CI/CD

ANVIL
Research Software Testing Platform

Azure Pipelines



circleci

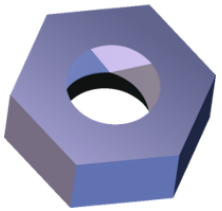
GitHub Actions

Google Cloud

Bitbucket Pipelines

Software

Jenkins



Buildbot

CI/CD

Bamboo

Challenges for Research Software and CI

- Computationally intensive – cpu/memory
- Use of accelerators
- Complex dependencies
- Multi-platform
- Specialist compilers + operating systems
- Multi-node execution

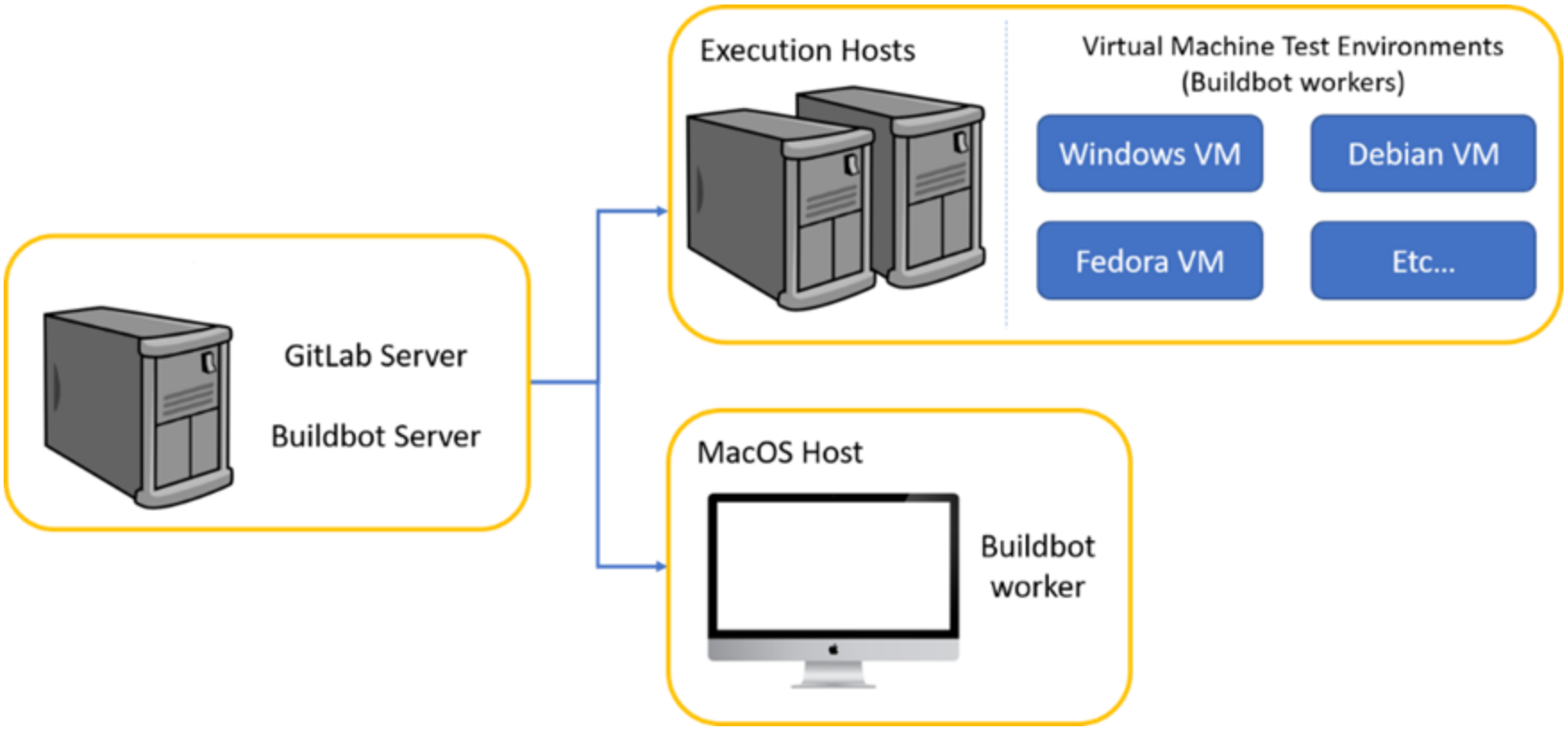
How do these interact with available CI implementations?

Nektar++ - www.nektar.info

- Finite element/computational fluid dynamics code
- ~15 years old
- Open-source C++
- 2 full time developers – Imperial + Exeter
- Variable number of PhD/project student developers

- Computationally intensive (compile + test)
- **Multi**-platform
- Complex dependencies

Existing Nektar++ CI Setup

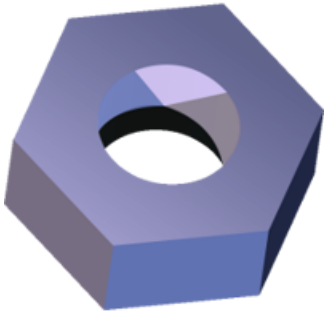


Criteria

- Reduced maintenance burden
- Work with on-premise GitLab code repository
- Greater reproducibility
- Test on Windows, Mac and 6 Linux distros
- Optimised build times (build cache)
- Rapid debugging of failures
- Infrastructure-as-code
- Easy to setup new environments
- No recurrent costs – preferably will make use of existing infrastructure

Review some alternatives

ANVIL
Research Software Testing Platform



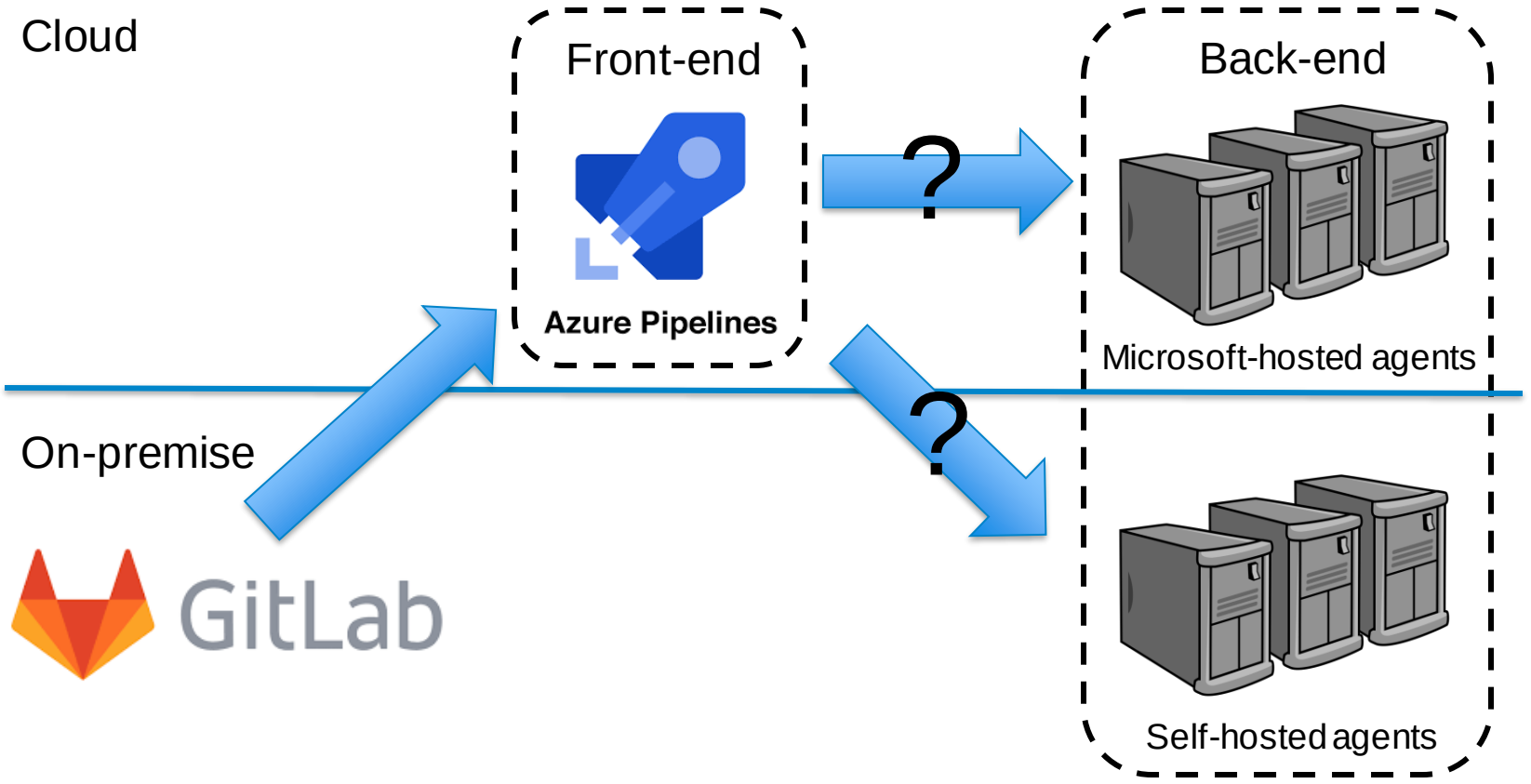
Buildbot
(with tweaks)

Azure Pipelines

CI/CD

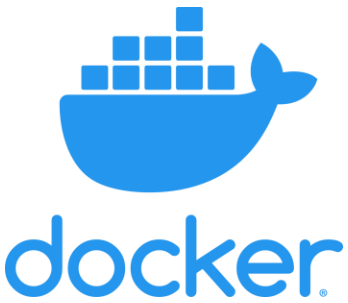
- Specialised CI service for research software
- STFC hosted (project restrictions)
- Based on Jenkins
- Can run workloads on SCARF (HPC cluster)
- Scientific software + compilers available in environment
  - Intel compilers

Front-end vs Back-end



Back-end alternatives

On-premise



"Cloud"

Azure Pipelines

Scores

Front-End

	Heterogeneous workloads	Gitlab Integration	Sustainability	Ease of Use	IaC	Advanced Gitlab Integration	CD
Buildbot	3	3	0	2	0	2	2
Gitlab CI	3	3	1	2	1	3	2
Azure Devops	3	1	1	2	1	0	2
Anvil	3	2	1	2	0	1	2

Back-End

	Multi-platform	Sustainability	Ease of Use	Infrastructure	Cost	IaC
On-prem VMs	2	0	1	2	3	0
On-prem Docker	2	1	1	3	3	2
Gitlab CI	2	2	0	2	1	2
Azure Devops	3	2	0	2	2	2
Anvil	1	2	0	2	3	1

Front-End	Back-End	Total Score
Gitlab CI	On premise Docker	26
Buildbot	On premise Docker	24
Azure	Azure	21
Anvil	Anvil	20

Beyond the scores

Azure Pipelines + Microsoft agents

- A good offering
- Every platform
- 10 concurrent free builds
- Lowest maintenance
- Held back by GitLab integration
- Unclear what cost would be

Buildbot + On-premise Docker

- Swapping VMs for Docker is a no brainer
- CI configuration is separate from code base
- Separate server to maintain
- Support for building rpms/debs
- Custom integration with GitLab

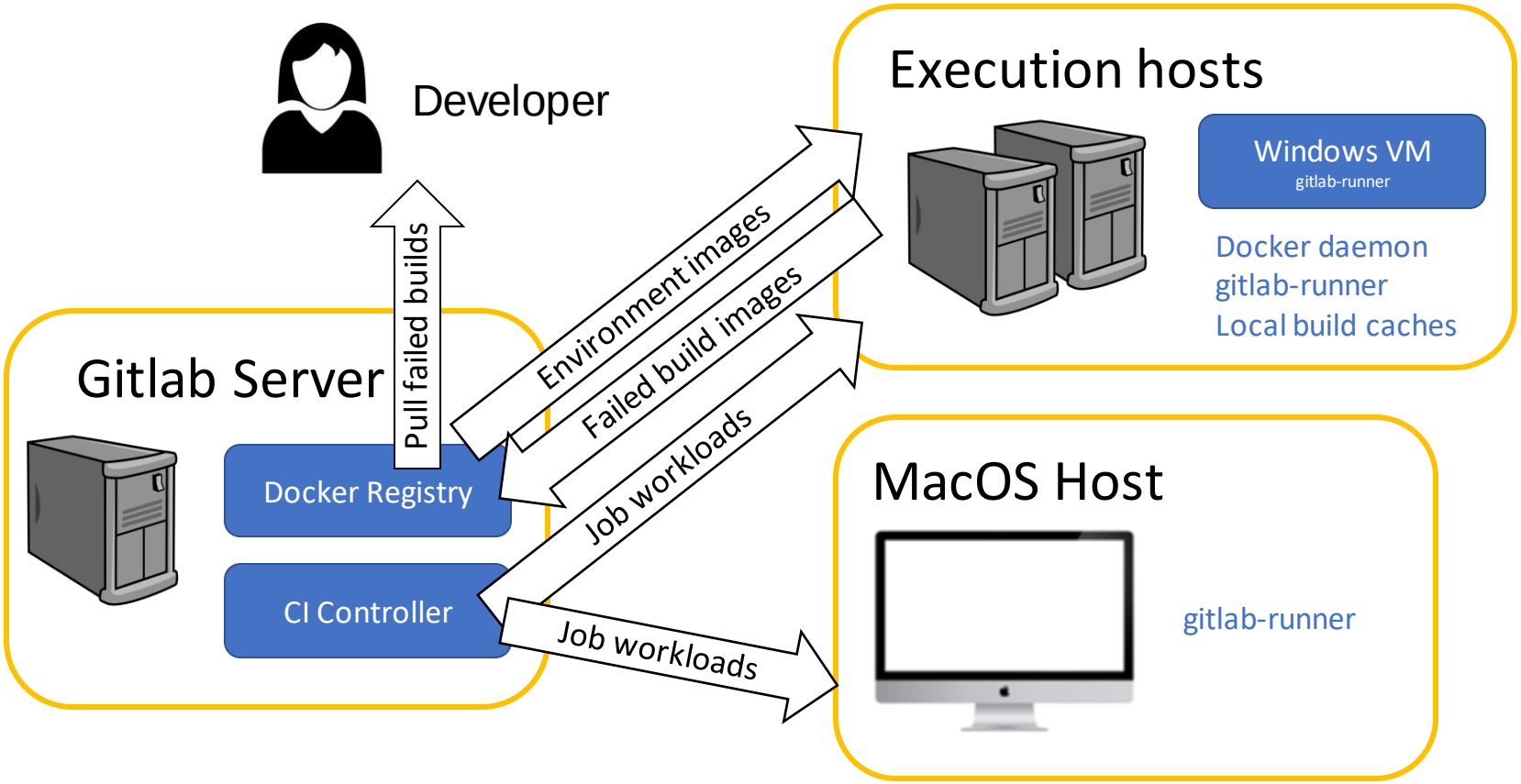
GitLab CI + On-premise Docker

- Integrated with code hosting
- GitLab.com runners would be expensive
- Container registry
- Conditional pipeline execution

Anvil + Anvil

- No container support
- Specialised environments not relevant to Nektar++
- No relevant dependencies available
- Questionable longevity

Our work-in-progress solution



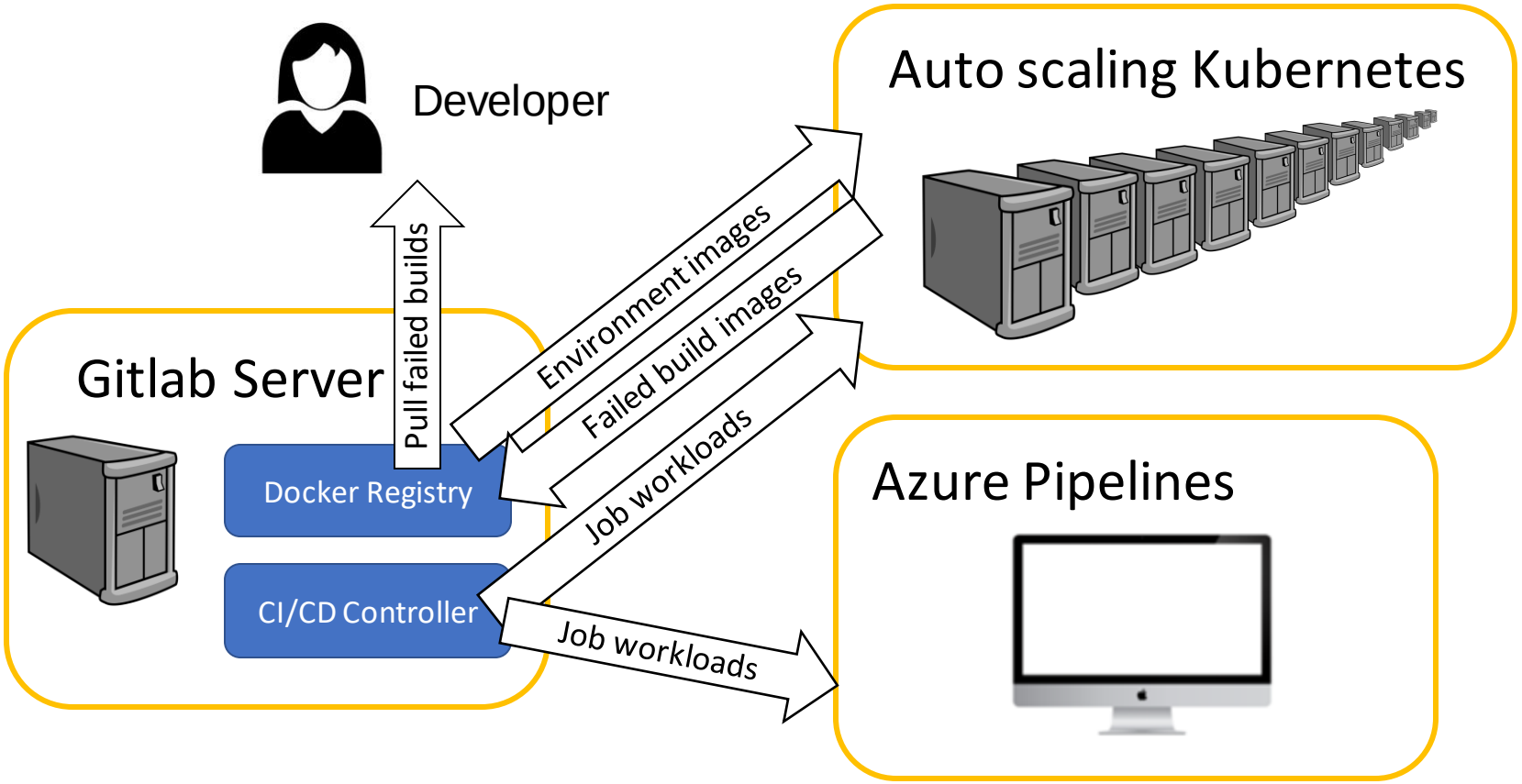
The benefits

- Reduced maintenance – 12 VMs down to 1
- CI configuration is under version control
- Non-admins can change the CI configuration
- Non-admins have access to rapid debugging workflow
- Linux builds are now fully reproducible
- Adding new Linux distros is easy
- Much more agnostic to execution host
- Faster and more flexible execution
- All in part of GitLab

Insights

- One size does not fit all
  - Individual project requirements
  - Existing constraints
- Not much to choose between different CI workflow languages – you're going to write a yml file
- Use Docker
- Don't underestimate time required to maintain infrastructure
- Existing cloud CI services still don't meet all use cases for research software

Cloud based Possibilities



Thank you!

- Nektar++ development team
  - Chris Cantwell
  - Dave Moxey
  - Spencer Sherwin
- Research Software Reactor
  - Tania Allard
  - Sarah Gibson
  - Gerard Gorman
  - Microsoft
- Research Computing Service
- Research Computing Service
  - Diego Alonso Alvarez
  - Mayeul d'Avezac de Castera
  - Mark Woodbridge

Questions?