Swiss Re

Where Scalability Meets Efficiency: Using MATLAB and Cloud Computing to Run Our Internal Risk Model

MathWorks Finance Conference 2022

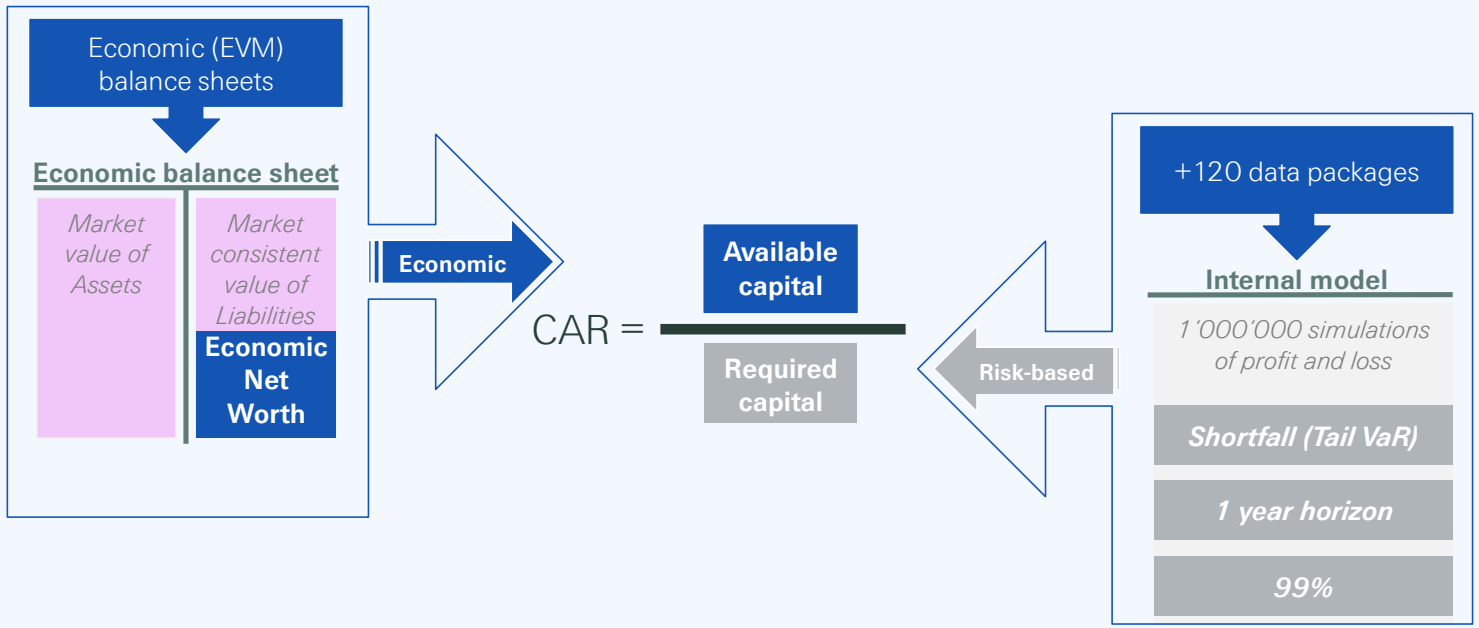
D. Rubio – 11 Oct 2022

Introduction

Background	• Swiss Re is one of the world's leading providers of reinsurance, insurance, and other forms of insurance-based risk transfer. • With headquarters in Switzerland, it uses a FINMA-approved internal capital adequacy model to comply with Swiss Solvency Test and Solvency II requirements. • The model is also used for business steering purposes and its usage has grown significantly over the past couple of years.
Status Quo	• The model is part of a complex technology ecosystem operated in a private cloud since 2019. • The current set-up is cost efficient and works well, improvements are regularly being introduced. • But some challenges remain which prevent the future ambitions to be achieved with the current platform.
Outlook	• After a successful exploration phase, we are kicking off a transformation program to migrate our risk models to Azure. • This transition will provide tangible benefits and fully aligns with Swiss Re's public cloud strategy.
Objective of today	• Outline our cloud transformation journey. • Discuss the operational and financial benefits of the current platform, and how we are addressing the challenges. • Discuss the transition to public cloud and how we are partnering with Mathworks to customize the reference architecture for Azure.

Swiss Re's capital position remained very strong in H1 2022, with the Group Swiss Solvency Test (SST) ratio above the 200–250% target range.

Capital Adequacy Ratio



Core Risks in internal model

PROPERTY & CASUALTY
- Natural catastrophe
- Man-made
- Costing & reserving
- Claims inflation

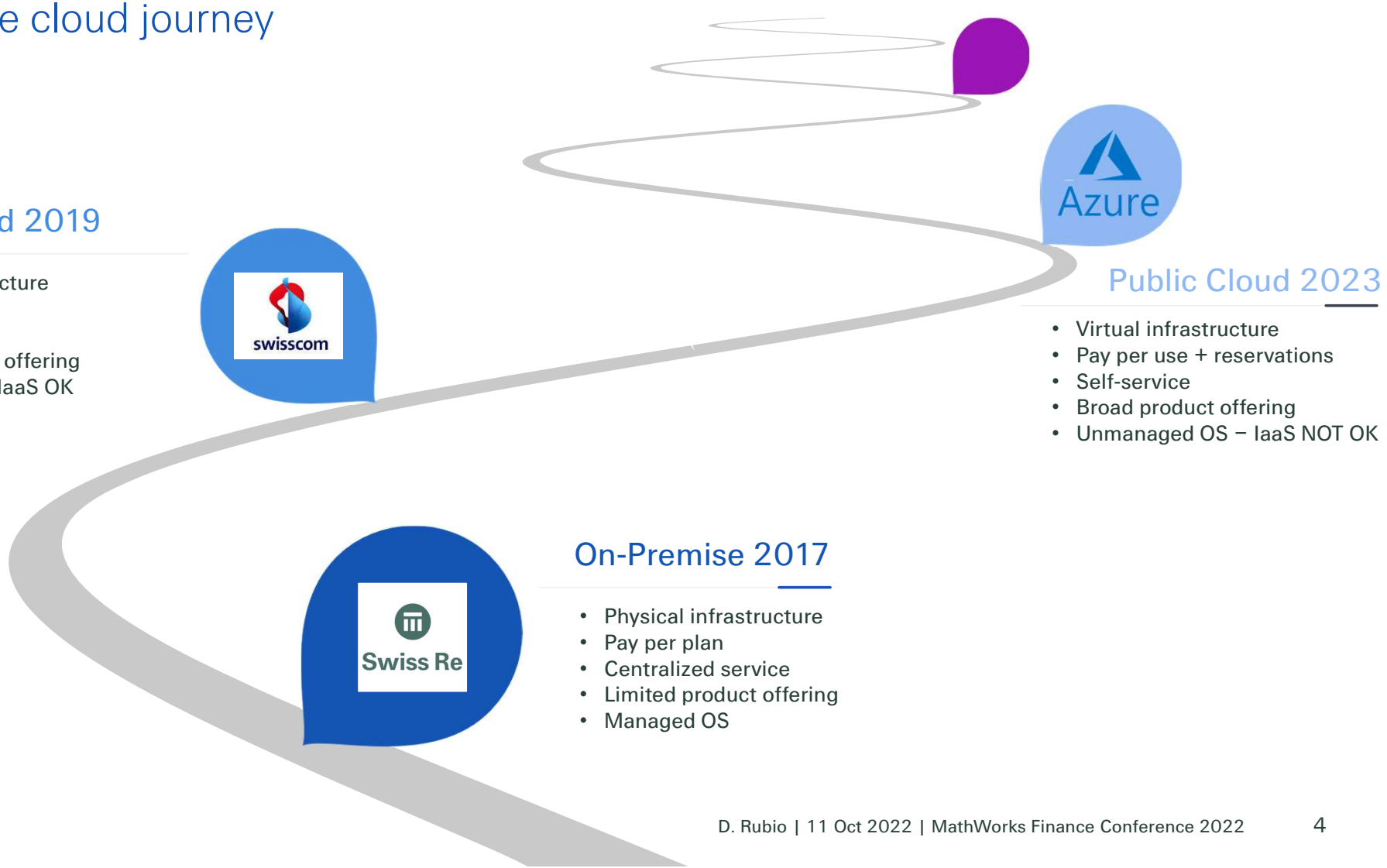
LIFE & HEALTH
- Lethal pandemic
- Mortality trend
- Longevity

FINANCIAL MARKET
- Credit spread
- Equity
- Foreign exchange
- FM inflation
- Interest rate
- Real estate

CREDIT
- Default risk
- Migration risk

Source: Swiss Re Risk Modelling

The private cloud has been a great learning experience, and has prepared us well for the next phase of the cloud journey



On-Premise 2017

- Physical infrastructure
- Pay per plan
- Centralized service
- Limited product offering
- Managed OS

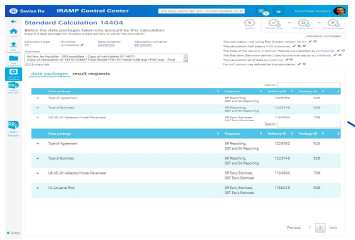
Private Cloud 2019

- Virtual infrastructure
- Pay per use
- Self-service
- Limited product offering
- Managed OS – IaaS OK

Public Cloud 2023

- Virtual infrastructure
- Pay per use + reservations
- Self-service
- Broad product offering
- Unmanaged OS – IaaS NOT OK

The internal model is deployed as the core of a complex technology ecosystem. A self-managed cluster provides the significant computing power required to perform large scale numerical calculations



Risk Analyst

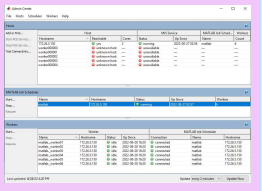
Head Node

Production Server

- RiskModelV1.ctf
- RiskModelV2.ctf
- RiskModelV3.ctf
- RiskModelV4.ctf
- RiskModelV5.ctf

Cluster Manager

Parallel Server



REST API

Cluster

Worker Node 1	Worker Node 2	Worker Node 3	Worker Node 4	Worker Node 5
Worker Node 6	Worker Node 7	Worker Node 8	Worker Node 9	Worker Node 10
Worker Node 11	Worker Node 12	Worker Node 13	Worker Node 14	Worker Node 15
Worker Node 16	Worker Node 17	Worker Node 18		

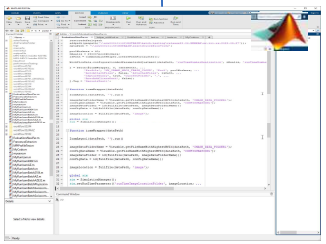
282 CPUs, 5.1 TB RAM, 3.5 TB storage

File Share

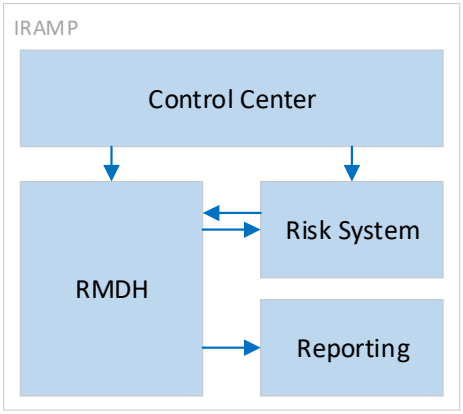
- RiskModelV1
- RiskModelV2
- RiskModelV3
- RiskModelV4
- RiskModelV5
- RiskModelV6
- RiskModelV7

10 TB storage

License Server

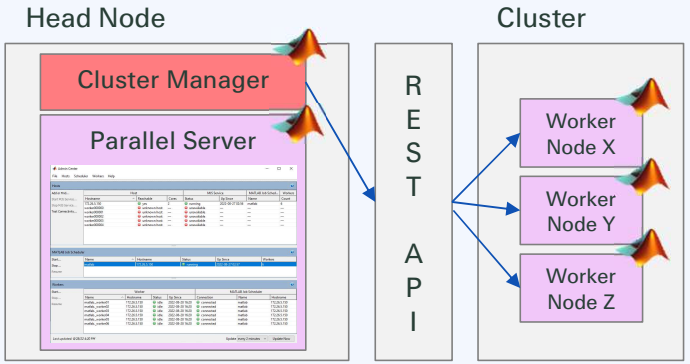


Power User



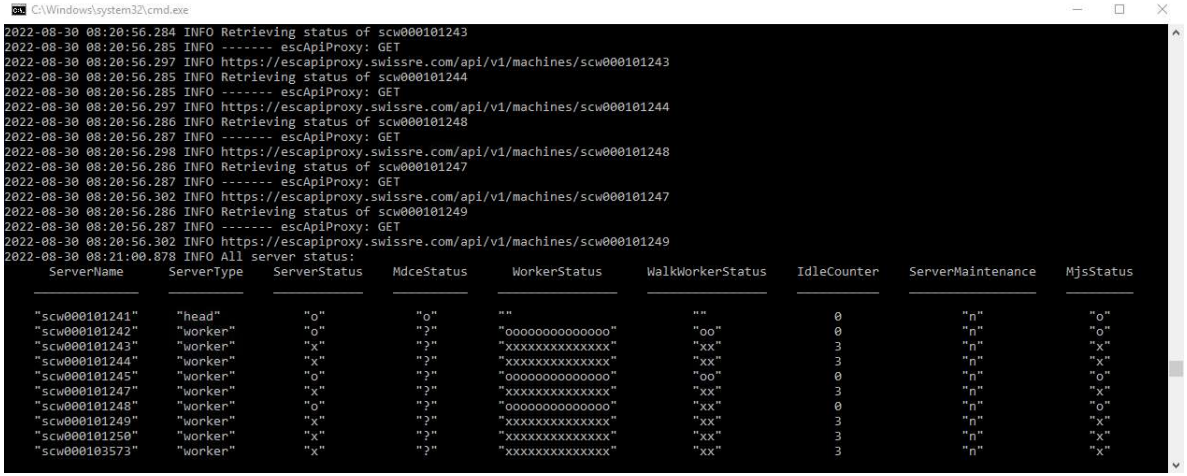
Our Cluster Manager scales programmatically the resources available and provides cost efficiency

How it looks



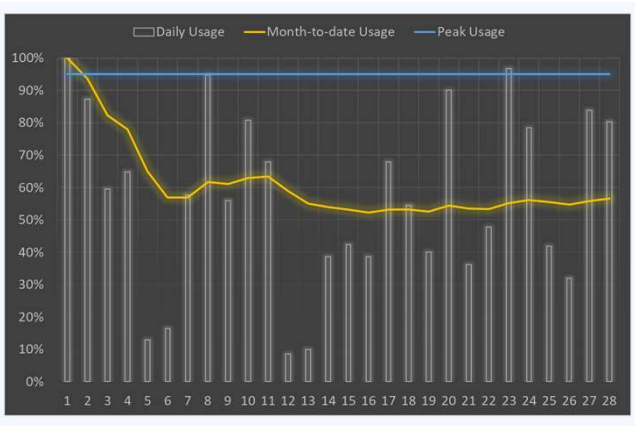
How it works

- Create blocker job
- Check cluster status (**VMs**, MJS, Worker)
- Check job queue
- **Start VM** / Install MJS / Start Workers
- Execute job
- Stop Workers / Uninstall MJS / **Stop VM**

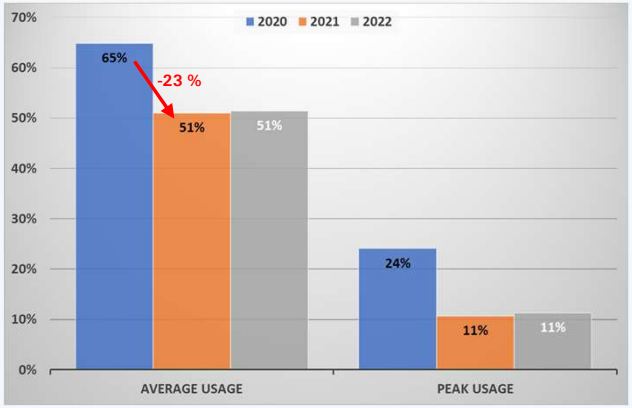


Significant savings can be achieved thanks to on-demand scaling and pay-per-use. Average usage reduced despite substantial increase in demand.

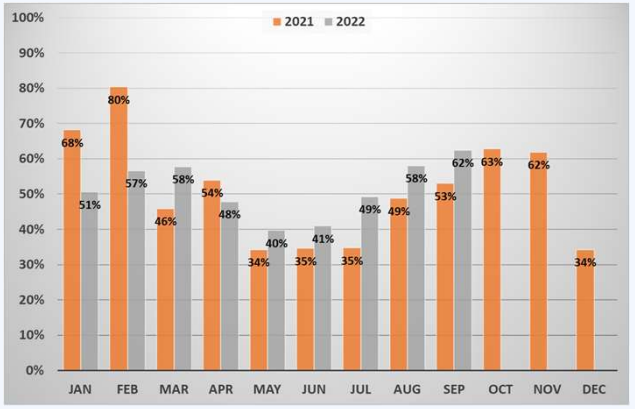
Daily Average Usage



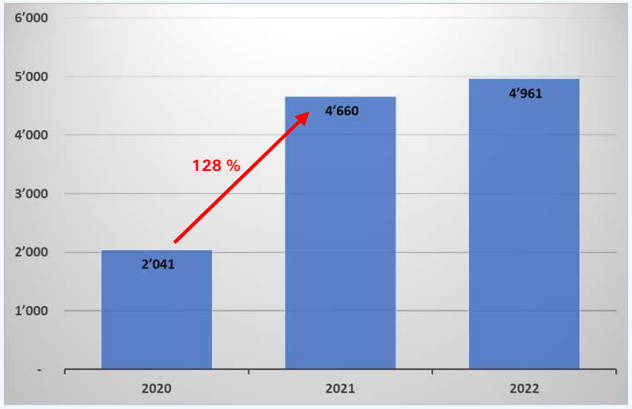
Annual Average/Peak Usage



Monthly Average Usage

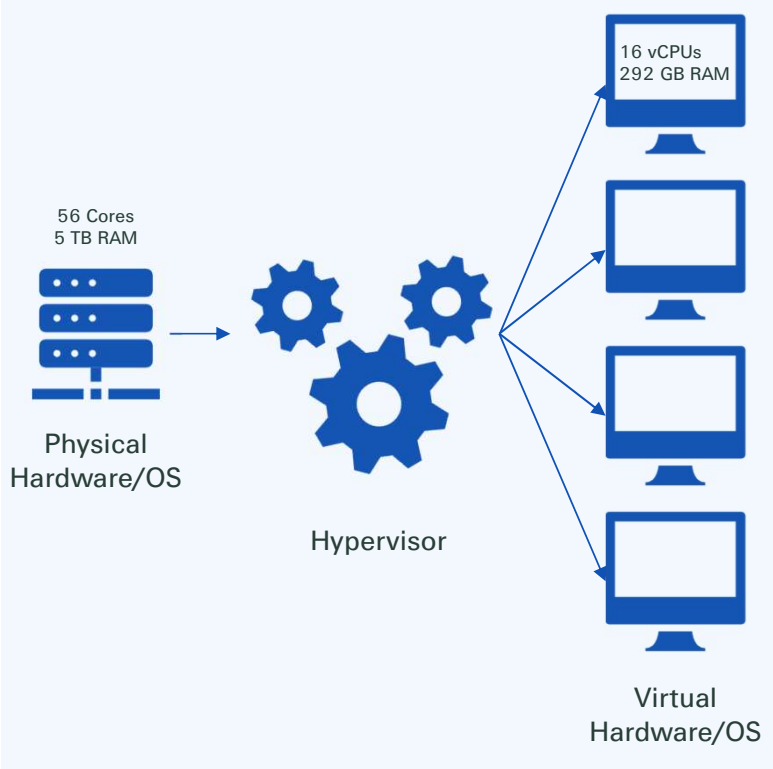


Annual Calculation Demand

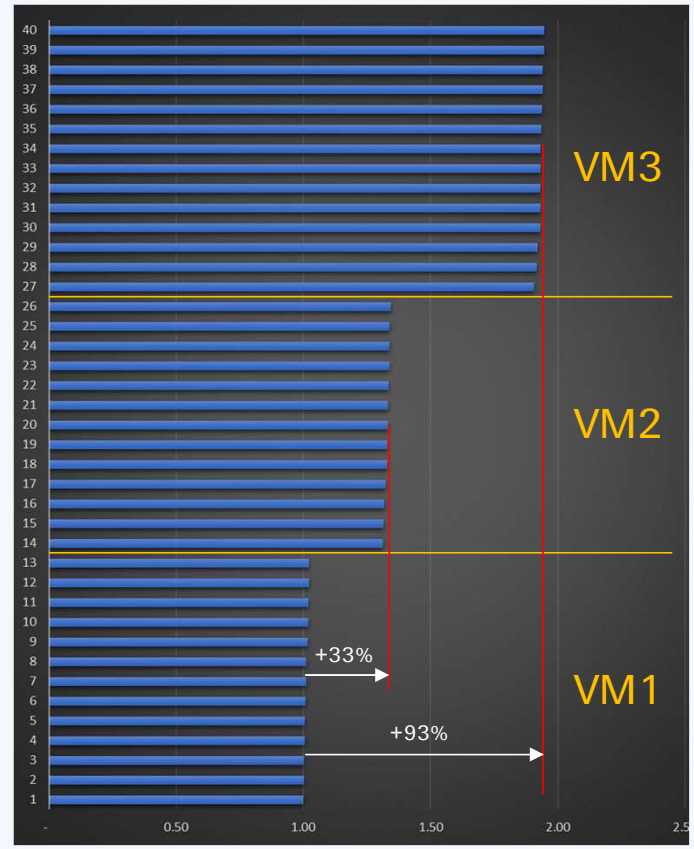


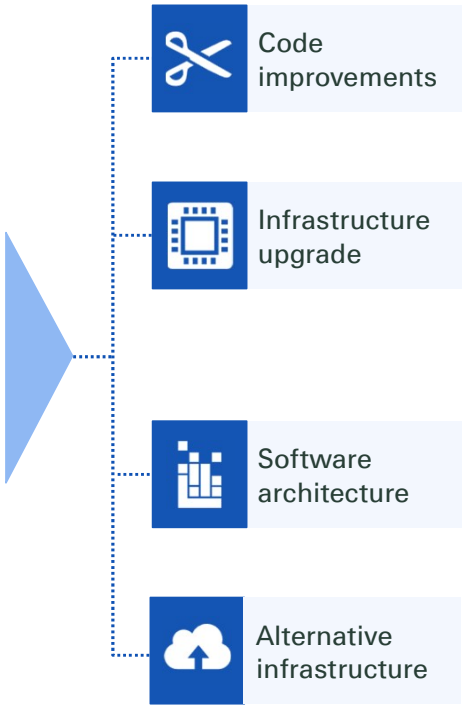
Overcommitment in a virtualized set-up may lead to performance volatility at worker level, particularly for CPU-intensive applications, leading to longer calculation times

Virtual Architecture



40 workers doing the same computational job



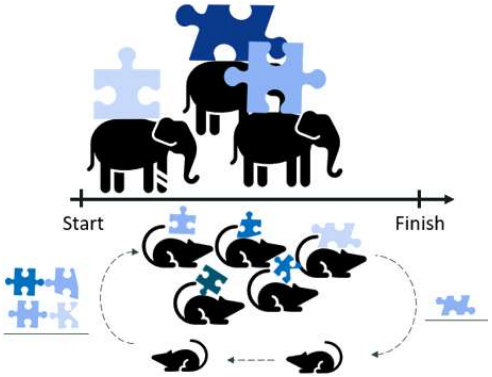


Transforming a few highly-dependent jobs into many independent jobs can be a game changer, particularly in a cloud context

Cloud-optimized parallelization scheme

Before:
Fixed pool of 40 workers processes 40 equal, massive jobs.

After:
Scalable pool of workers asynchronously processes ~3000 small jobs



Benefits

Scalability	Pool of workers can be enlarged (faster, more expensive) or made smaller (slower, cheaper) depending on business need.
Modularity	Individual jobs (=models) are disentangled. This increases flexibility for future model development.
Resilience	A single job failing no longer leads to a failure of the whole calculation. Instead, the failing jobs can be individually retried.
Performance	Performance is much less sensitive to a single low performing worker.

Resilience improvement

Old parallelization: An error in any part of the model crashed the entire calculation.
New parallelization: Failed jobs are retried.

```
21-Jun-2022 21:02:14: Calculating transactions in run-off…
    Error using Calculate2/setXlTransaction (line 1456)
    Out of memory. Type "help memory" for your options.
21-Jun-2022 21:02:17: End of log of job id 715.
21-Jun-2022 21:02:17: Failed job id 715. Retrying...
…
2022-06-21 21:24:00.279 INFO Workflow technically completed
2022-06-21 21:24:00.282 INFO Semantic status:
COMPLETED
```

Error → **Retry** → **Success!**

Performance improvement

Old parallelization: Slower workers delay the entire calculation.
New parallelization: Faster workers no longer wait.

```
09-Sep-2022 14:28:36: Running jobs are:
CriticalIllness for slice 1 (elapsed: 54.625s, JobID:149 )
CriticalIllness for slice 2 (elapsed: 54.614s, JobID:150 )
CriticalIllness for slice 3 (elapsed: 48.339s, JobID:151 )
CriticalIllness for slice 4 (elapsed: 38.010s, JobID:152 )
CriticalIllness for slice 5 (elapsed: 38.000s, JobID:153 )
CriticalIllness for slice 6 (elapsed: 35.781s, JobID:154 )
CriticalIllness for slice 7 (elapsed: 33.568s, JobID:155 )
CriticalIllness for slice 8 (elapsed: 29.310s, JobID:156 )
CriticalIllness for slice 9 (elapsed: 18.959s, JobID:157 )
CriticalIllness for slice 10 (elapsed: 12.629s, JobID:158 )
CriticalIllness for slice 11 (elapsed: 2.308s, JobID:159 )
CriticalIllness for slice 12 (elapsed: 0.065s, JobID:160 )
LifeProtection for slice 13 (elapsed: 813.775s, JobID:533 )
09-Sep-2022 14:28:36: Jobs outstanding: 716
```

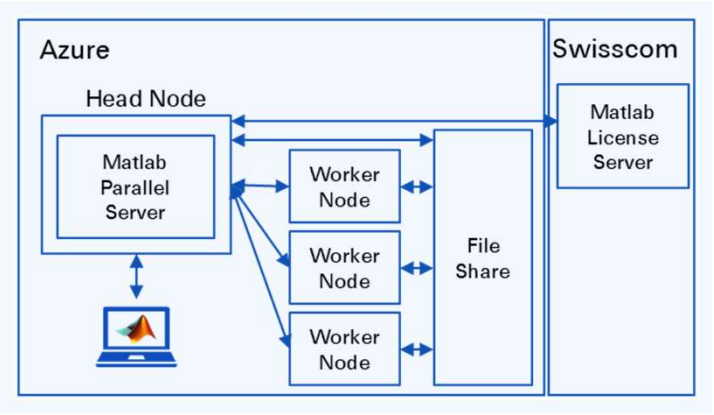
Take care, and so long

I am not very fit today!

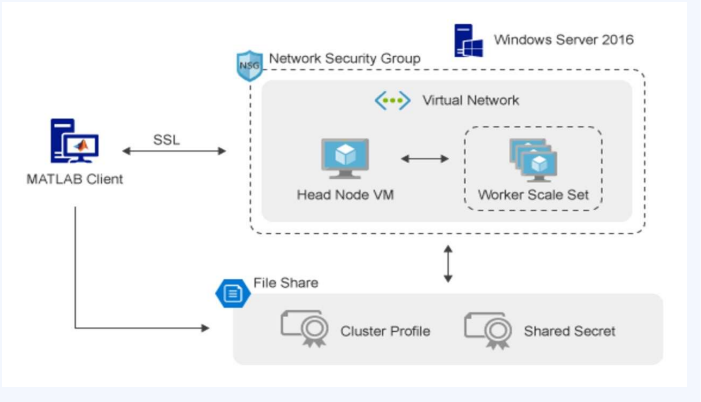
Source: Swiss Re Risk Modelling

We have used a self-managed cluster as a simple way to enter Azure, and used the insights we gained to subsequently adopt the reference architecture

Self-Managed Computing Cluster



Azure Marketplace: MATLAB Parallel Server (BYOL)



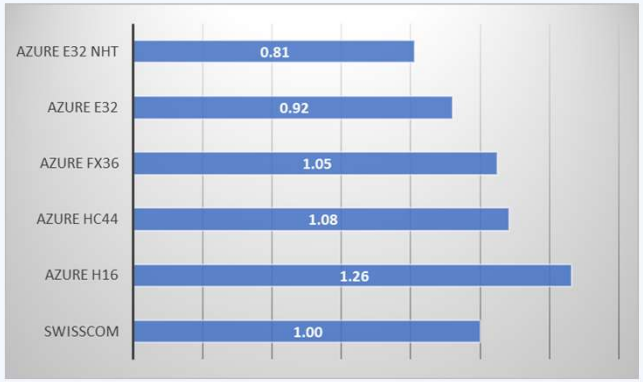
Ambitions

Performance	Scalability	Availability	Economics

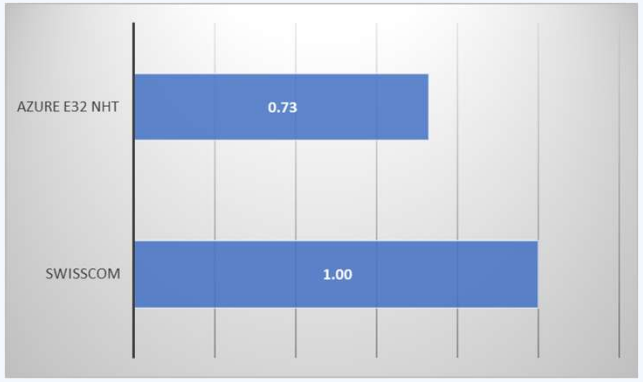
Supported by Microsoft, we have used a self-managed cluster to optimize the infrastructure choices in Azure for our particular use case

- **Computing**
  - High-performance computing (1:1): H-, HB- and HC-Series
  - Memory optimized (2:1): E-Series (also non-hyperthreaded)
- **Storage**
  - File share standard: hot, cold and transaction optimized
  - File share premium
  - Blob storage
- **Network**
  - Accelerated networking ON/OFF
- **OS**
  - Windows Server 2016, 2019 and 2022
- **Matlab version**
  - 2019b, 2021b and 2022b

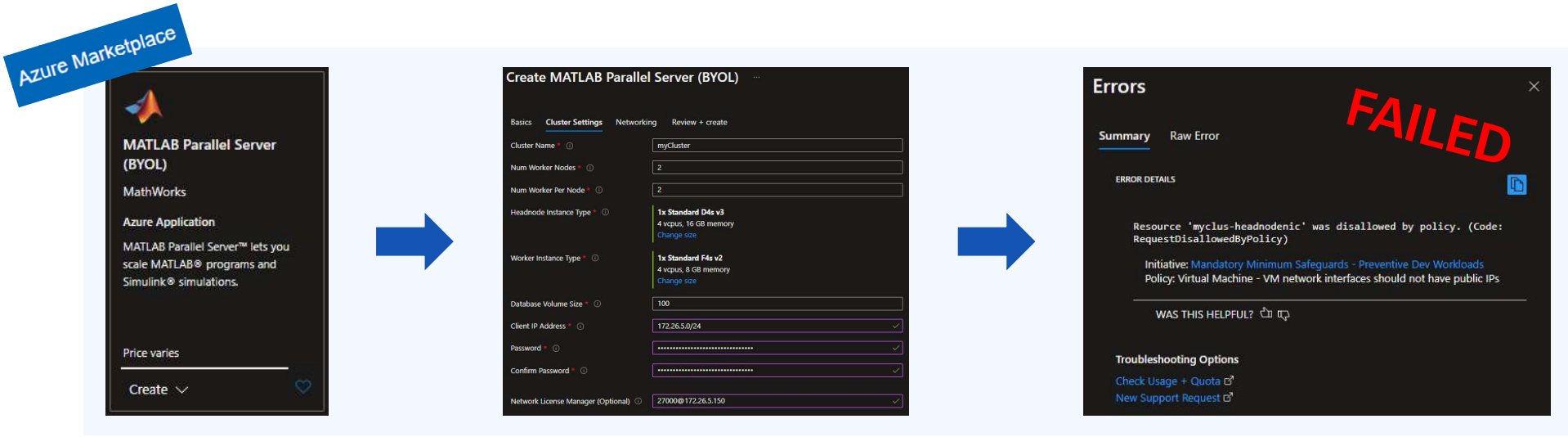
With the elephants



With the mice



We can't use the Azure Marketplace product out-of-the-box due to internal security restrictions, so we collaborated with Mathworks to customize the deployment process



Original deployment steps:

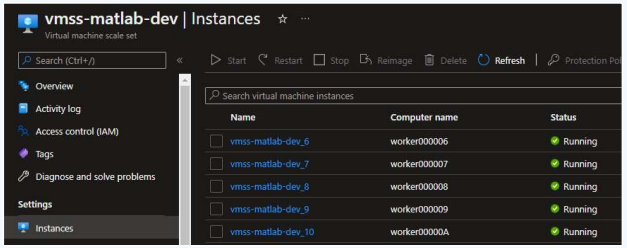
- Use Mathwork's own image with a given OS and Matlab version
- Deploy infrastructure components
- Apply Azure configuration (e.g. network)
- Run post-deployment scripts to install and configure Parallel Server

Mitigation actions:

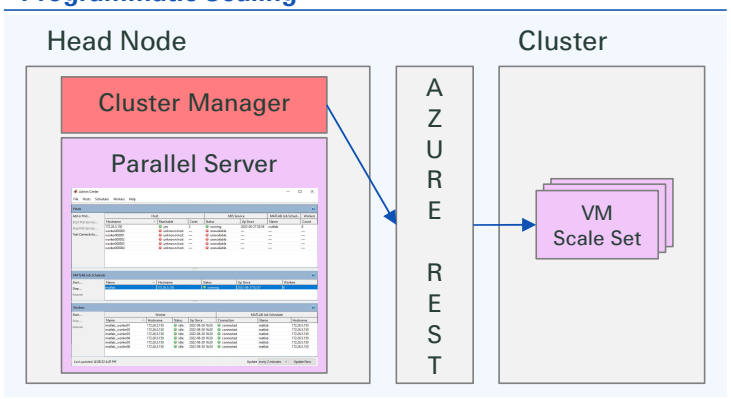
- Migrate deployment template from ARM to Terraform (optional)
- Replace default VNET with own private VNET
- Remove public endpoints
- Amend post-deployment scripts so that Parallel Server avoids public endpoints

Combining a scale set with a smart scaling agent is another game changer, as it unlocks unprecedented computing capacity in a highly-efficient manner

Manual Scaling



Programmatic Scaling

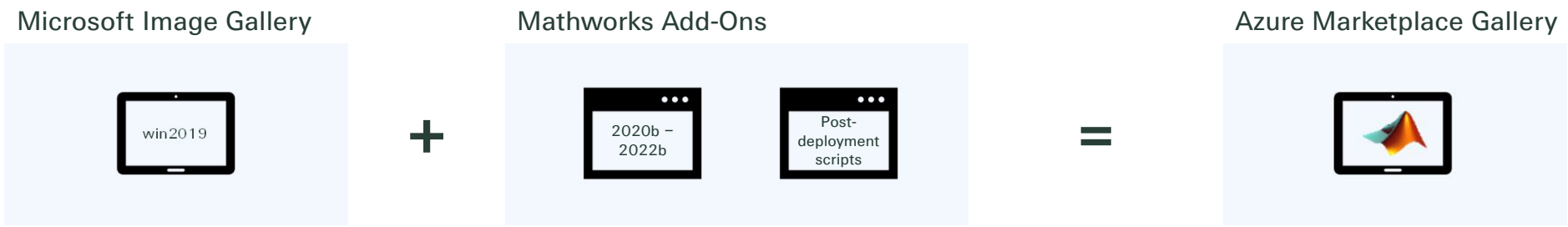


Custom Scaling

- **Standard autoscale**
- **Predictive autoscale**
- **Scheduled autoscale**

Scaling use case: load testing the File Share

We worked with Mathworks to understand the ingredients of the Azure Marketplace image, and we created our own to have full flexibility

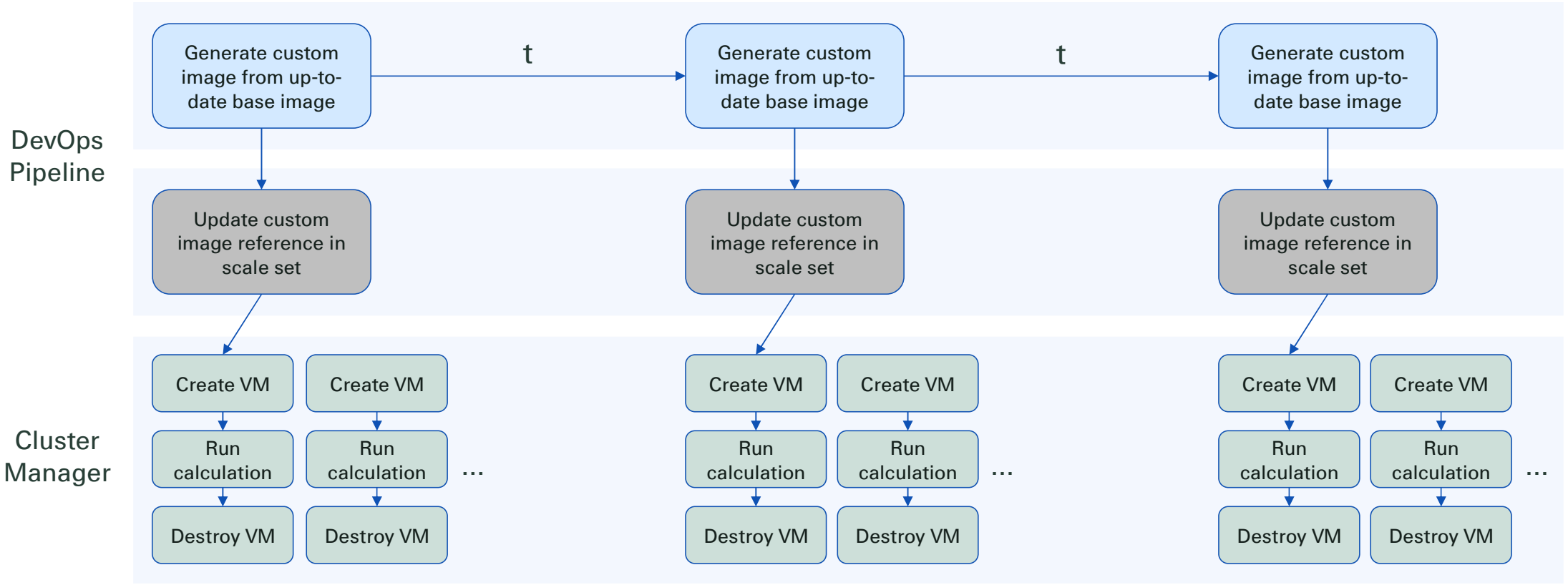


Packer Image Builder

Swiss Re

Performance | Scalability | Availability | Economics

An updated custom image removes the need for “online” OS updates, and results in increased availability and reduced operational support



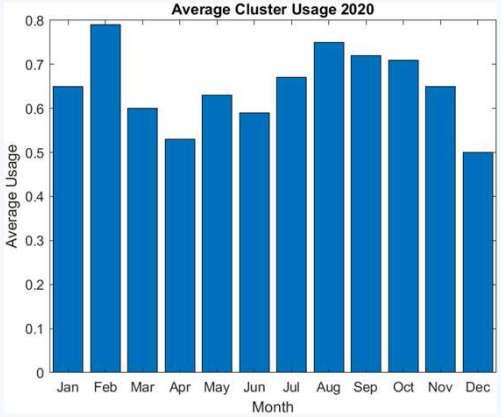
Swiss Re

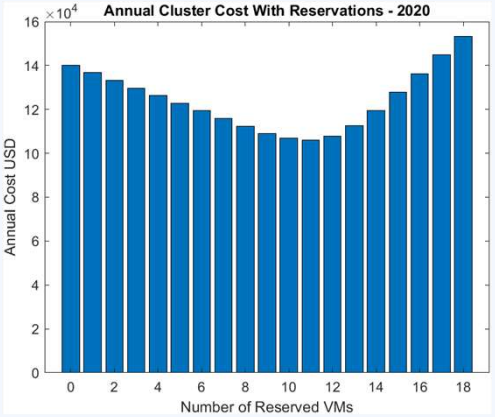
Performance | Scalability | Availability | Economics

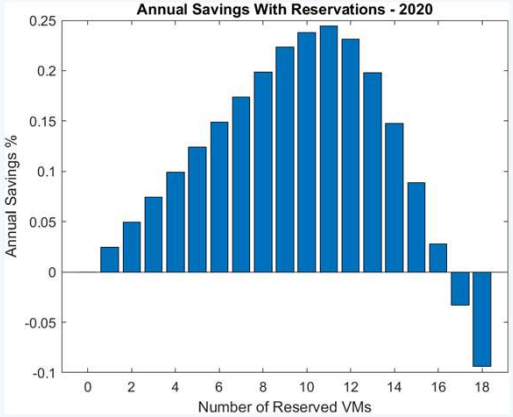
Infrastructure reservations represent an additional way to optimize the cost of computing resources, but predicting the right amount of reserved instances is not an easy task

2020

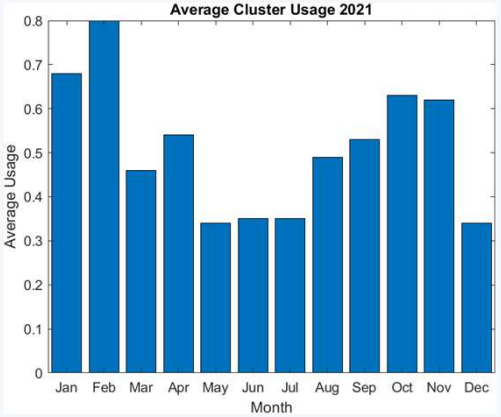
Cluster Size = 18 VMs
VM Cost = 1'000 USD
Discount = 29%

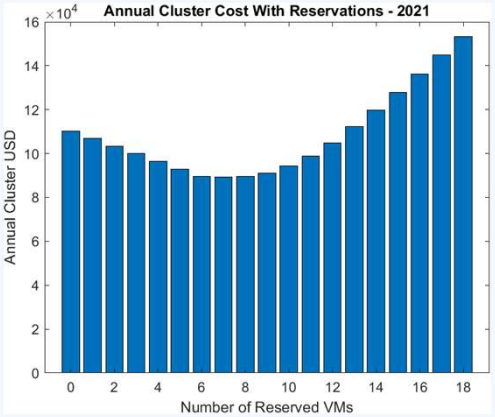


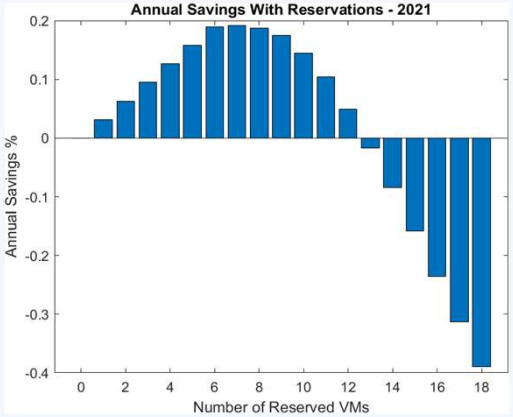




2021







Performance Scalability Availability Economics

Swiss Re

Any questions? Thank you!

Contact us

Daniel Rubio

Head Integrated Risk IT
Daniel_rubio@swissre.com

Acknowledgments

Danny Rasch, Salvi Amato, Francesco Atzeni, Tobias Frischknecht, Benjamin Troxler, Lionel Trebuchon

Edu Benet Cerda, Yannis Ben Ouaghrem, Alison Eele, Karen Tanner

Darko Mocelj, Daniele Rispoli, Shayne Kropp

Follow us