Cloud Computing Guide

Why use cloud computing?

We've been asked regularly about running OrcaFlex in cloud computing environments, particularly by customers who need to:

- Run simulations faster than their current hardware allows
- Process large batches without tying up office workstations
- Scale computational capacity up and down based on project demands
- Reduce the time from model setup to results

Cloud platforms allow you to provision servers with hundreds of cores in just a few mouse clicks. Whilst we don't currently offer cloud computing as a service ourselves, many of our customers are successfully using cloud platforms to accelerate their workflows. This guide shares practical approaches for deploying OrcaFlex in cloud environments.

Before you start

Licensing is the key consideration

The most common challenge is ensuring your FlexNet licence server is accessible from your cloud environment.

This typically requires either connecting your cloud resources to your on-premises network via VPN, or deploying a licence server in the cloud. All cloud solutions consume OrcaFlex licences just like machines in your office.

There is no additional cost associated with cloud processing.

We don't recommend specific providers

Whilst we're aware of customers successfully using various approaches with Amazon Web Services (AWS), Microsoft Azure, Google Cloud, Rescale and other platforms, we don't endorse any particular cloud provider.

Three approaches to cloud deployment

Which approach suits you depends on how you currently work with OrcaFlex and your team's experience with cloud technologies. Here are the three main options:

Option 1: Virtual machines (easiest to get started)

This is the most straightforward approach and works exactly like running OrcaFlex on a powerful office workstation. You get a Windows server in the cloud, install OrcaFlex, and connect to it using Remote Desktop. Batch processing works exactly as it does on your local machine.

When to choose this

You want faster hardware but minimal changes to your existing workflow. Good for getting started quickly or if you're unfamiliar with containers and APIs.

Setup steps:

1. Create a virtual network that can reach your licence server (via VPN or by deploying the licence server in the cloud)
2. Provision a Windows-based virtual machine (VM)
3. Install OrcaFlex and configure it to point to your licence server
4. Connect via Remote Desktop and work as normal

Performance tip

Compute-optimised instances often give the best cost-per-simulation performance, whilst High Performance Compute (HPC) families deliver the fastest individual simulation speeds. Experiment with different instance sizes to find the right balance for your workload. You can also save "images" to avoid reinstalling and configuring OrcaFlex each time you need a new VM - just deploy from your saved image and you're ready to go immediately.

Cloud Computing Guide

Licence servers

If you choose to put your licence server on a VM then you need to ensure that this machine persists while all the licences are active. You cannot use the “VM Image” technique described above for these servers and if the licence server VM is destroyed (or if ephemeral storage detaches) then you will lose access to these licences. You can contact us to assist with disaster recovery under these circumstances.

Option 2: Managed HPC platforms (fully managed)

Third-party HPC platforms provide a fully managed service where you submit jobs without managing infrastructure yourself. This removes setup complexity but introduces platform costs and can make monitoring simulation progress less straightforward.

When to choose this

You want a comprehensive solution and don't want to manage servers or infrastructure. You're comfortable with less visibility into and control of running simulations.

Option 3: Docker containers with Python API (most flexible)

We provide pre-configured Docker containers on Docker Hub that include OrcaFlex, Python and OrcFxAPI ready to use. These containers are headless (no graphical interface), so you control simulations programmatically using Python scripts rather than through the OrcaFlex GUI or Batch.

When to choose this

You're comfortable with Python scripting and want maximum flexibility in how you distribute and manage simulations. Particularly useful if you're already automating workflows with OrcFxAPI.

What to know

Containers provide a standardised environment that runs consistently across different platforms. You write Python scripts to set up and run simulations, then use cloud orchestration tools to distribute the work across multiple containers.

Find our containers at: https://hub.docker.com/u/orcina

We provide several variants:

- Windows-based containers (recommended starting point)
- Ubuntu-based containers with WINE for cost reduction (see below)
- Containers with specific Python and OrcaFlex version combinations
- Benchmark containers for performance testing

Cost reduction with Linux containers

A common question from customers already running workloads in the cloud is how to reduce infrastructure costs.

Windows Server instances are typically more expensive than Linux instances, and this difference adds up quickly when running large compute workloads.

OrcaFlex is a Windows native application, but we provide Docker containers that run on Linux hosts using WINE (an open-source Windows compatibility layer). These Ubuntu-based containers come with Python, OrcaFlex and OrcFxAPI pre-configured, eliminating the complexity of setting up WINE yourself.

The trade-off

Our testing shows WINE, which introduces computational overhead, reducing simulation throughput by up to 30% compared to native Windows execution. For workloads with frequent API calls (such as external functions), this overhead may be more noticeable.

When it makes sense

The cost savings from Linux instances can be substantial for workloads requiring significant compute capacity over extended periods. You should evaluate whether being able to provision more compute capacity at lower cost compensates for the individual performance reduction.

Recommendation

Start with Windows-based containers to establish your workflow, then explore Linux-WINE containers if cost optimisation becomes important.

Practical considerations

Data transfer costs and performance

This can matter for large simulation files. Consider the bandwidth between your cloud instances and storage, particularly if you're writing large result files frequently. Post-calculation actions can be used to extract required results to avoid the need for simulations to be saved in the cloud.

Instance sizing

Latest-generation CPUs with fast SSDs deliver the best performance, but at higher cost. Start with compute-optimised instances and adjust based on your cost-performance requirements.

Model optimisation

You can sometimes reduce simulation time more effectively by optimising the models to run faster rather than improving the hardware. We're always happy to review models to investigate optimisation opportunities.

Getting more help

For Docker containers, comprehensive documentation is available on Docker Hub covering container setup, licence configuration, data management and integration with your own code.

If you have questions about deploying OrcaFlex in cloud environments or encounter issues, please contact us at orcina@orcina.com. We always welcome questions and feedback about your experiences with cloud deployments, as this helps us improve our documentation and containers.