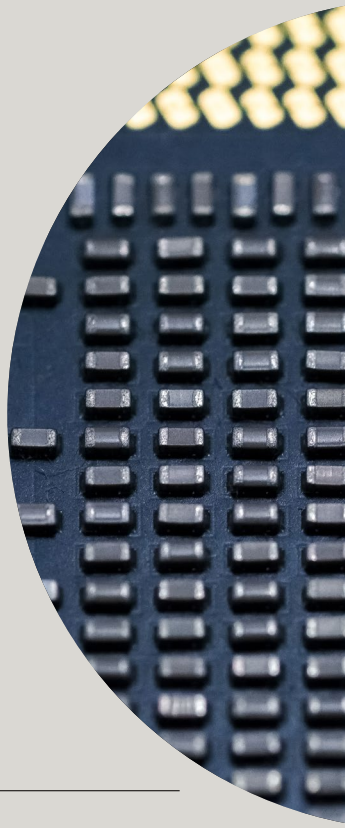
Build vs. Buy Dask Clusters

Your team uses Python for data pipelines, data science, machine learning, and deep learning workloads because Python has all the tools to easily get started on their laptop to build out everything from simple to complex analysis.



Once you've vetted your code, you're ready to scale beyond your desktop to use more data, and that requires more processing power. How can you leverage your code without a lot of refactoring? You're in luck because Dask, the premier native Python distributed computing framework, will allow you to easily scale beyond your desktop! So how do you do that? And how do you make it production-ready?

There are three possible deployment scenarios:

1) **Dask on a single machine or large cloud virtual machine**
2) **Manually deploying Dask on a cluster**
3) **Managed Dask-as-a-Service in the cloud**

Try Coiled Cloud Now

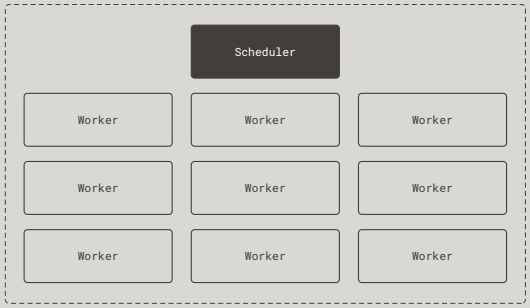
Dask on Single Machine or Large Cloud VM

Laptop
8 cores, 32 GB RAM

VS.

Manually Deploying Dask on a Cluster

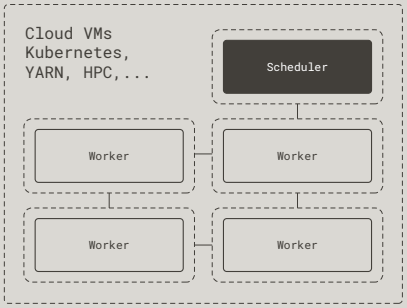
Large Cloud VM
128 cores, 2 TB RAM



VS.

Managed Dask-as-a-Service in the Cloud

Cluster of machines
4 cores, 16 GB RAM



	1 Dask on Single Machine or Large Cloud VM	2 Manually Deploying Dask on a Cluster	3 Managed Dask-as-a-Service in the Cloud
Security	Low	Limited	High
Cost	Low	High	Low
IT Management	High	High	Low
Time-to-Deployment	Low	High	Low
Performance	Limited	High	High
Scalability	Limited	High	High
Developer Experience	High	Low	High

Dask on a Single Machine or Large Cloud VM

Simpler but Limited

Running Dask on a single machine or large cloud virtual machine is an easy way to take advantage of parallelism while leveraging existing data scientist investment in Python code development without any additional costs. However, there are two major implications:

- **Performance** will be limited by disk I/O, memory limits, CPU, or network bottlenecks.
- **Scalability** will be constrained by the performance of the single machine.

Learn More

Other considerations include:

- **Security** vulnerabilities and risks are introduced by connecting a single machine to production data.
- **IT management** is burdened with supporting one machine per data scientist that typically doesn't conform to all the IT standards and there's no sharing of computing resources.
- **Developer experience** is ideal, but no other team members benefit from the knowledge and experience of data scientists working in different isolated environments.
- **Time-to-deployment** is elongated. There's a fast time to develop but moving to deployment involves roadblocks and delays due to security and IT concerns.

Manually Deploying Dask on a Cluster

Buyer Beware: Build (and maintain) clusters at your own risk

If and when you hit the performance and scalability wall, the good news is that you can build Dask clusters to get beyond those limitations. And there are a myriad of possible Dask deployment options, including:

- Docker containers on one or more VMs
- A cluster of VMs (bare metal, VMWare, OpenStack)
- A cluster of containers in Kubernetes (EKS, AKS, GKE, OpenShift)
- A cluster of managed containers (ECS, Docker Swarm)
- A cluster of YARN workers/containers (Cloudera, Google DataProc)
- A cluster of workers in an HPC cluster (Slurm, PBS, OpenGrid)
- An infinite number of other possible implementations

However, building Dask clusters requires trial and error, a significant investment to set up and test your infrastructure, test and debug your code, load test and debug the infrastructure. Plus, all security, authentication, SSO, access control, and credential management have to be built and tested. Then to gain access to production data, thorough InfoSec reviews are performed to review the architecture, validate security, and ensure compliance checks are being performed. Since security has been built and isn't standardized, additional pentesting, vulnerability, and fault tolerance testing are typically required.

Learn More

The major implications of building your own Dask clusters include:

- **Time-to-Deployment** increases as the team have to balance analytics work vs. building/maintaining infrastructure capabilities.
- **Security** checks, authentication, SSO, access control, and credential management have to be built and maintained to conform to evolving threats.
- **Costs** start to escalate due to the additional time to build and maintain security. There is a lack of proactive cost management controls such as quotas, limits, and cost reporting. Plus, there are hidden costs associated with the reduced data scientist productivity to support the building and maintaining of the infrastructure and security. This loss of productivity also impacts time available to data scientists, DevOps, and IT to analyze data.

Other considerations include:

- **Developer experience** and workflow are interrupted to manage packages and build infrastructure. This decreases their time to analyze data and develop models. This is also a deterrent to prototyping since even the simplest of prototypes require building out the infrastructure first. Since the analytics team doesn't build out infrastructure regularly, oftentimes, the complex infrastructure negatively impacts the end-user experience.
- **IT management** and support increases as IT is typically unaware of the environment until there is a problem or security vulnerability is exposed. Then IT has to scramble to support the shadow IT built infrastructure that doesn't conform to IT standards and isn't consistent with other IT supported data science environments.

Managed Dask-as-a-Service in Private or Public Cloud

Simpler, Scalable, Secure

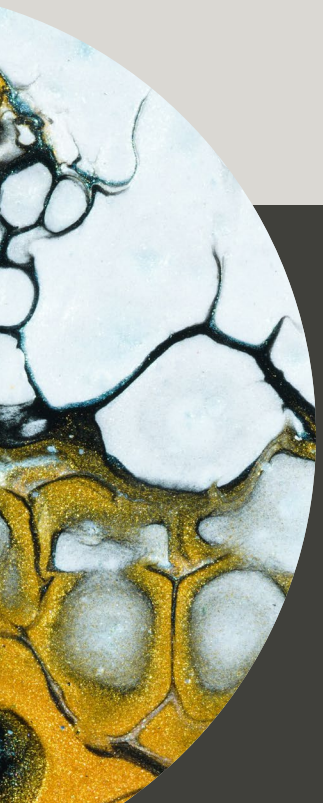
Coiled Cloud can operate within a multi-tenant environment or within a private cloud running a VPC in your preferred cloud platform, including AWS, GCP, and Azure. Coiled Cloud standardizes your Dask infrastructure and gives you the peace of mind to know that your Dask clusters are secure. Coiled Cloud automatically creates and manages Dask clusters and their underlying infrastructure. You get out-of-the-box industry standard security, authentication, SSO, access control, and credential management. User and team controls include cost controls, idle timeouts, quotas, limits, usage reporting, and cost tracking. Coiled Cloud makes it easy to manage packages and dependencies using your preferred approach with pip, conda, or Docker. Coiled Cloud takes care of scaling up, handling multiple users and clusters, and abstracts away low-level cloud resources and networking configuration. Coiled Cloud is a commercial product with well-documented security and operating model that makes it easier and faster to go through InfoSec and AppSec reviews.

The major implications of using Coiled Cloud as your standardized cloud environment for Dask clusters includes:

- **Time-to-Deployment** decreases as the team can dedicate their efforts to building additional analytics features and capabilities instead of building/maintaining infrastructure.
- **Security** controls are standardized to protect sensitive information. Industry-standard best practices to manage TLS/SSL certificates on the Dask scheduler/workers is included along with end-to-end network security and encrypted communications. Authentication is handled via email/password authentication, Google sign-in, GitHub sign-in, and by utilizing API tokens for authorization to Dask clusters. Pass-through authentication handles forwarding of credentials to Dask clusters by delegating tokens via Amazon STS to access AWS data sources.
- **Costs** are controlled by setting spending limits and quotas for the team or by a user. Reporting on historical costs and usage can be used to tune cost savings. Idle timeouts prevent runaway cloud bills.
- **Developer Experience** is streamlined, and productivity is increased. Running secure data workloads at scale is extremely easy. Working in their native environment – Jupyter notebook, VS Code, or favorite workflow orchestration tools such as Airflow, Prefect, or batch job – is a snap. Now there is a separation of responsibilities. The data scientist and data engineer focus on their core skills and the infrastructure building has been done for them and IT can easily control quotas and costs.

Additional considerations include:

- **Performance** isn't limited and can be adjusted to meet the business and SLA requirements. Better performance leads to rapid iterations since jobs take less time to complete. Enabling the use of fast-to-fail techniques that can radically change business processes by moving to near real-time processing.
- **Scalability** on-demand is achieved to meet the computational demands required by workloads. Autoscaling occurs with fine grain optimization that proactively scales based on the demand from Dask computations.
- **IT management** is centralized for the full stack – infrastructure and analytic workloads – with consistent secure environments for compute resources. The infrastructure is fully documented and meets your IT standards eliminating shadow IT.



Learn More

Try Coiled Cloud Now

About

Coiled scales Python to the cloud for data professionals. Based on Dask, the leading Python-native solution for distributed computing, Coiled has hosted more than 100M tasks for data professionals, scientists, and researchers around the globe including Capital One, Anthem Health, and the Air Force to solve challenges in business, research, and science.

Coiled is a remote-first company with the best and brightest working from around the globe. Founded by the initial author of Dask, Coiled is on a mission to create a platform that gives Data Scientists the power of the cloud and machine learning, freeing them from today's limitations so they can solve important problems.