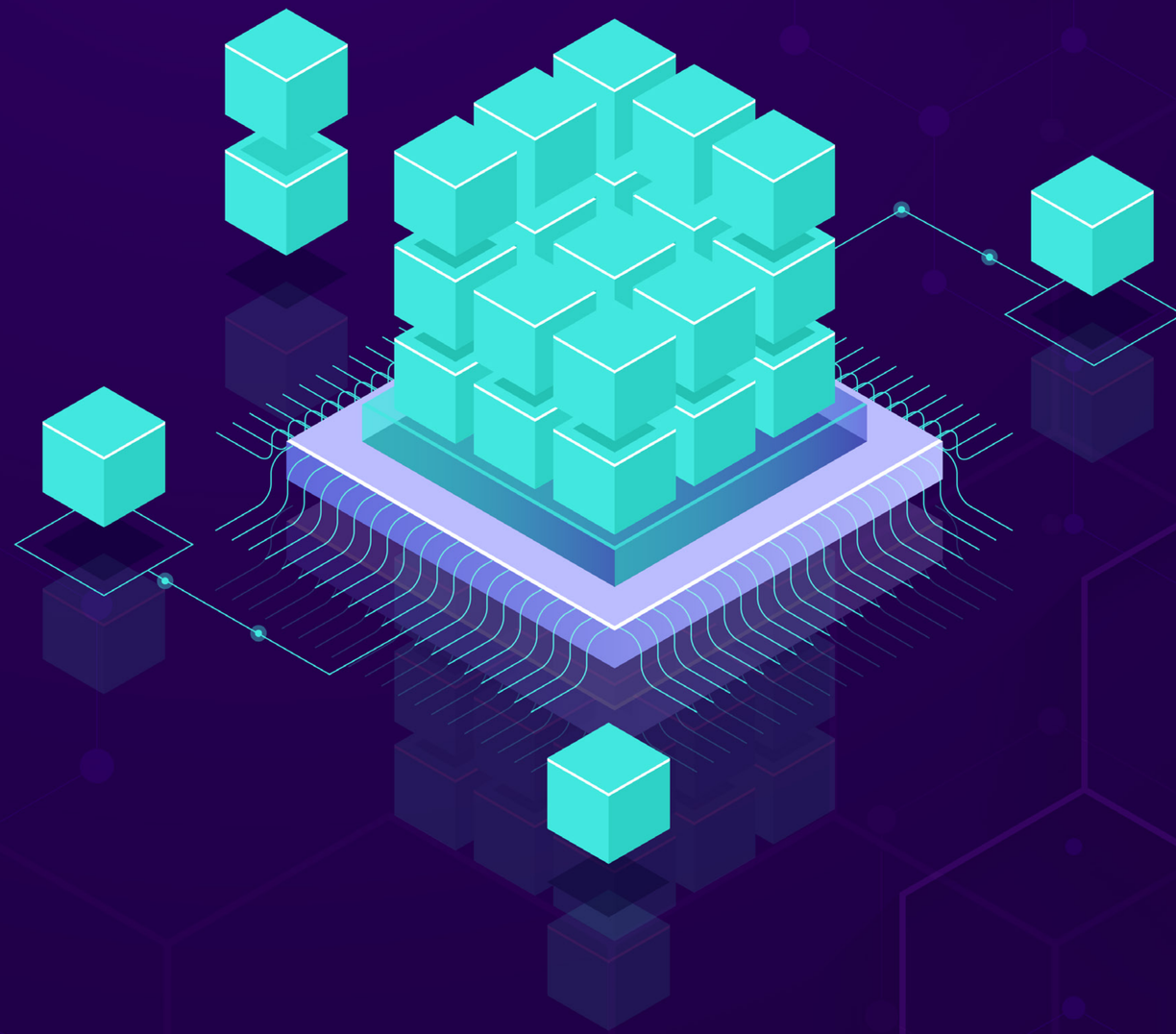
Ebook

Best Practices for Building a Scalable Machine Learning Infrastructure



cnvrg.io

Contents

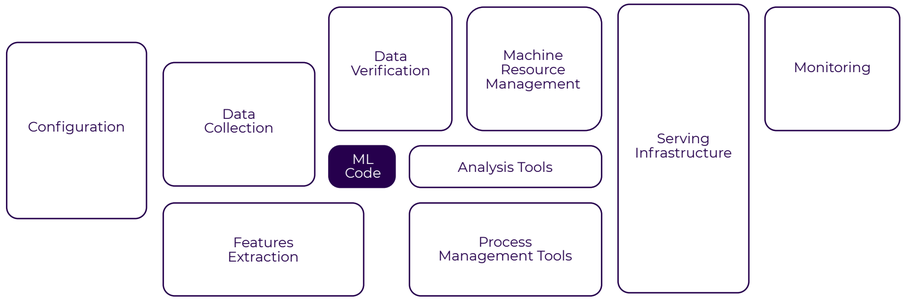
Executive Summary

Machine learning has matured and now data science teams demand more from their machine learning infrastructure. In the past machine learning was mostly for research, today it is driving businesses. While the base of a machine learning platform remains the same (manage, monitor, track experiments and models) to achieve scalability, elasticity and operationalization of machine learning development there are various capabilities that need to be considered before building a modern machine learning infrastructure. Today's machine learning infrastructures must be built for production, with as little technical debt as possible to accelerate machine learning development.

This guide will give a comprehensive understanding of what a modern machine learning infrastructure looks like, and how to build it for scale.

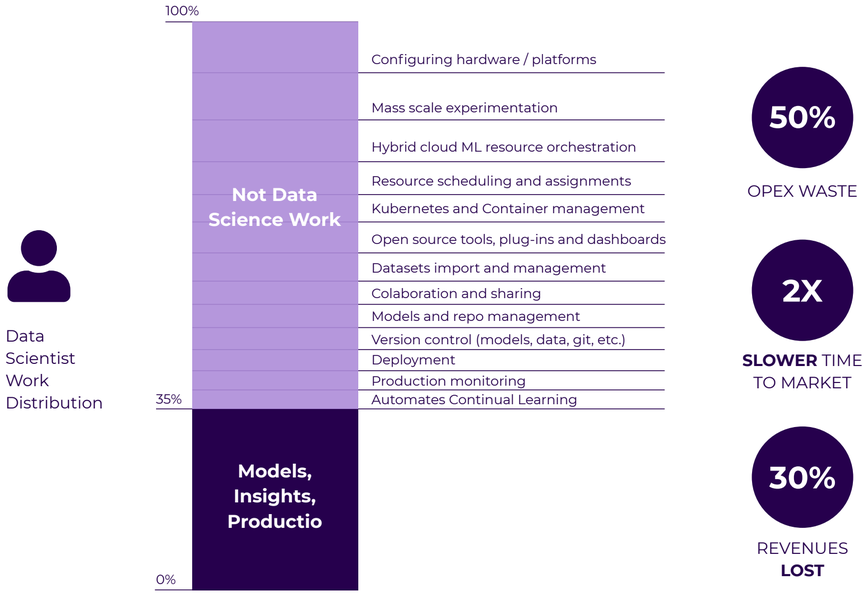
How do you build a scalable machine learning infrastructure?

There are a few critical elements when building a machine learning infrastructure. You need your machine learning infrastructure to be built for scalability, and to provide you with visibility so you can build plans on top of your existing stack. We'll first talk about the AI fabric comprising your compute resources, orchestration platforms like Kubernetes or OpenShift, and learn how to integrate that to your machine learning workflows. Components of a machine learning infrastructure also require solutions for data management, data version control and should provide a ML workbench for data scientists to give a simple way to train models, work on their research, and optimize models and algorithms. The last component of a scalable machine learning infrastructure is offering an easy and intuitive way to deploy models to production. One of the biggest challenges today, is that a lot of the models don't make it to production because of hidden technical debt that the organization has. Your machine learning infrastructure should be agnostic, and easily integrate into your existing and future stack. It should be portable and utilize containers for simple deployments, and allow your data scientists to run experiments and workloads in one click. In the following sections we will dive into the main aspects of building a scalable machine learning infrastructure.



What are the biggest machine learning infrastructure challenges?

The biggest challenge today facing AI and machine learning at scale is that data scientists are doing very little data science. When you look at a data scientist's day-to-day, you'll find that most of their time is spent on non-data science tasks like configuring hardware, configuring GPUs, CPUs, configuring machine learning orchestration tools like Kubernetes and OpenShift, and containers. In addition, hybrid cloud infrastructures have also grown in popularity for scaling AI. Operating in a hybrid cloud infrastructure adds complexity to your machine learning stack, as you need a way to manage all the diverse resources across cloud, multi cloud, hybrid clouds and other complicated setups.



Resource management has become a major part of a data scientist's responsibilities. For example, it is a challenge having a GPU server on-prem for a team of five data scientists. A lot of time is spent figuring out how to share those GPU's simply and efficiently. Allocation of compute resources for machine learning can be a big pain, and takes time away from doing data science tasks.

Managing machine learning models can also take a lot of time. Tasks like data versioning, model versioning, model management, deployment of models, using and streaming your open source tools and frameworks. In order to accelerate machine learning, data scientists should be able to focus on building the machine learning models, building the core IP over your technology, and monitoring model performance.

What are the biggest machine learning infrastructure *business* challenges?

An ill equipped machine learning infrastructure can greatly impact the business results of AI and ML. It can cause a high amount of OPEX waste, specifically underutilized clusters on-premises. In addition, time wasted on DevOps bottlenecks can slow time to market. This wasted time is often called 'technical debt'.

The second major business challenge for machine learning is concerning the machine learning workflow. When you look at AI in the enterprise today, there are two main workflows that are disconnected and broken. The first is the DevOps workflow, also known as MLOps. This workflow is focused on resource management, infrastructure, orchestration, visualization of models in production, integrating to the existing IT stack such as Git or Jira etc. Then there is the data science workflow, which is more concerned with data selection, data preparation, model research, running a lot of different experiments, training models, validation of models, tuning models, and eventually model deployment. There are so many steps and components in each of those pipelines. Today, those two flows are completely disconnected, and often are managed by completely different teams. As a result of these broken workflows, enterprises

experience a large technical debt. This challenge can have an effect on time to production, and have an overall effect on cost. As your organization scales, often these workflows become more complex. If you have teams across the world working on different projects, the infrastructure is completely siloed. This is why having a scalable machine learning infrastructure means having a streamlined machine learning infrastructure across all projects and teams in the organization.

How do you use MLOps best practices in your machine learning infrastructure?

In order to begin tackling these challenges, you must understand what MLOps is. MLOps, or machine learning operations reduce friction and bottlenecks between ML development teams and engineering teams in order to operationalize models. As the name indicates, MLOps combines DevOps practices for the unique needs of machine learning and AI development. It is a discipline that seeks to systematize the entire ML lifecycle. MLOps in the context of enterprises helps teams productionize machine learning models, and helps to automate DevOps tasks, so data scientists can focus less on technical complexity and more on delivering high impact machine learning models.

There are two main questions you should be considering when building your machine learning infrastructure. 1: how can it be built in a way that offers an intuitive experience for data scientists that do not have a DevOps background? 2: How can we build an enterprise stack that provides scalability and high performance for DevOps engineers that manage the overall machine learning stack?

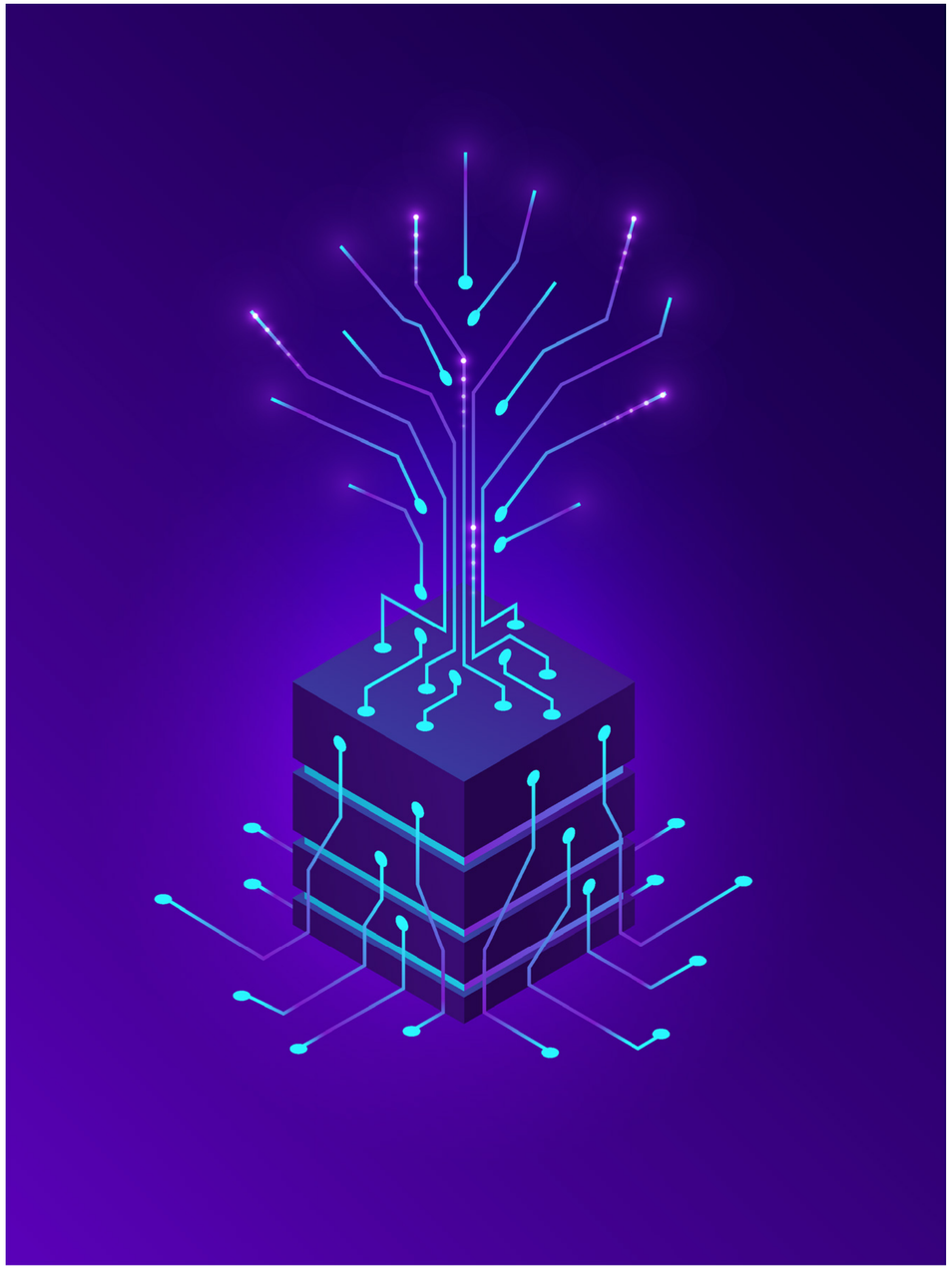
MLOps tackles a major part of these questions. With machine learning, it all starts with the compute. Machine learning is compute intensive. A scalable machine learning infrastructure needs to be compute agnostic. Whether your infrastructure is with GPU clusters, CPU clusters, Spark clusters, or cloud resources. We often see with enterprise customers that there is a pool of resources that is used for building machine learning applications. It may be split for different teams but still you have a desktop with GPUs, GPU clusters, CPU clusters and also cloud resources. This diverse pool of resources can be used for model training, for data preprocessing, model serving, inferences, and other machine learning workloads.

In a real machine learning pipeline example, you always have a data set, and one of the workers is allocated for data preprocessing. Then, some workers are allocated for model training, which could be ResNet, VGG16, YOLO, InceptionV3, or InceptionV4. In this case, we'll be using GPU workers for better deep learning performance. These GPU workers are

being used for the actual model training. In parallel you may have some Jupyter Notebooks up and running, consuming some compute power, plus model deployment and serving on a cloud instance. Now, this is just one single pipeline, but at the enterprise scale it can be much more complicated with multiple pipelines and projects running in parallel.

Now if you dive even deeper into this single pipeline, the compute consumption can get even bigger. In each algorithm you will have hyper parameter optimization. So with VGG16, it's not just a single run of TensorFlow, it is almost 500 runs of TensorFlow code, running on the AI infrastructure. That means it is going to allocate 500x the compute resources. A scalable machine learning infrastructure should support running 500 experiments in 500 runs. The more models you run, model training with more tweaking, you're probably going to get better results and more optimized accuracy.

Schedulers and meta schedulers in your machine learning infrastructure helps to improve both the data scientists interface by providing self service workload management, and offers scalability because it is compute agnostic.



What architecture can support machine learning at scale?

Now we will go over the steps to building an architecture that can support enterprise machine learning workloads at scale.

1. Containers

Containers are key to providing a flexible and portable machine learning infrastructure. With containers you can assign machine learning workloads to different compute resources. So GPUs, cloud GPUs, accelerators, any resource that you have can be assigned to each workload. Using containers can help distribute jobs on any of the resources that you have available. It is great for DevOps engineers because it provides a more portable and flexible way to manage workloads.

Containers help you to define an environment and are also great for reproducibility and reproducible data science. You can launch the containers anywhere on any cloud native technology. So, on-premise, Kubernetes cluster, bare-metal, using Docker simply and also cloud resources that have extensive support for all the different containers. You can also operate orchestration platforms like OpenShift, that make it easier for you to run and execute containers in the cluster.

2. Orchestration

When it comes to orchestration, you need to build something that is compute resource agnostic. While Kubernetes is becoming the standard way of deploying machine learning and for orchestration, there are so many flavors of Kubernetes. There is Rancher, there is OpenShift, there is Vanilla Kubernetes. Even for small deployments, there is MicroK8, and MiniKube. So when you're designing your own infrastructure, you need to decide what kind of orchestration platform you're aiming to support now and in the future. So you need to be able to design the stack in a way that fits your existing infrastructure while considering future infrastructure needs.

Also, whatever infrastructure you're designing, you need to be able to leverage all the compute resources that you already have in your enterprise. So, if you have a large Spark cluster, Hadoop environment or you have bare-metal servers that are not running on Kubernetes - like large CPU clusters - then you need to be able to support those as well. You need to build an infrastructure that can integrate to the Hadoop cluster, that can leverage Spark, that can leverage YARN, and can leverage all the technology that your organization has. Not only that, but additionally you should consider how to manage all your compute resources in one place for all your data scientists across the industry to access and use in one click.

3. Hybrid cloud multi cloud infrastructure

What are the benefits of a hybrid cloud infrastructure for machine learning? This is a big topic that could easily take on its own post. But specifically in machine learning, a hybrid cloud infrastructure is ideal because usually machine learning workloads are stateless. That means that you may run a machine learning training for a day, or for two weeks, and terminate the machine. As long as all the models and data are being stored, you can simply terminate the machine, and forget about it. Hybrid cloud deployment for machine learning is unlike software in this way. In software you need to persist and make sure the database is shared across the hybrid environment. For hybrid cloud machine learning, it's beneficial to control your resources in order to utilize the existing compute you already have. For example, let's say an organization has eight GPUs on-premise, and 10 data scientists. Your organization would want to be able to utilize all of the eight GPUs, and only burst to cloud when it reaches 100% utilization or allocation. Cloud bursting is an essential capability that allows organizations to increase parameterization, and also reduces cloud costs. Not only that, but cloud bursting allows data scientists to easily scale machine learning activities.

4. Agnostic & open infrastructure

Flexibility and being able to easily extend your base platform is critical, because machine learning is evolving extremely fast. So you need to design your machine learning infrastructure in a way that enables you to easily extend it. That means that if there is a new technology, a new operator, a new platform that you want to integrate, you can easily do that without reconfiguring your entire infrastructure. If there is one thing you take from this guide on machine learning infrastructure is to pick your technologies carefully, make sure it is agnostic, and built for scale. That way you can quickly adopt new technologies and operators as they evolve.

Second, if your infrastructure is agnostic, you also need to think about your interface with data scientists. If your interface is not intuitive, then you will miss the benefits of the new technology into your infrastructure. Remember, data scientists are not DevOps engineers or IT. Often they are PhD's in math and don't want to work with YAML files or namespaces or deployments etc. They want to do what they were hired to do which is to work on their models. So you need somehow be able to abstract the interface for data scientists, especially if you're using Kubernetes while providing them the flexibility and control that they need. Meaning that if there are data scientists or DevOps on your team who want to get into the internals of Kubernetes, you need to be able to allow that as well. In the end, it is all about supporting your data science and engineering teams to make them better professionals.

How do you schedule jobs on each of the different interfaces?

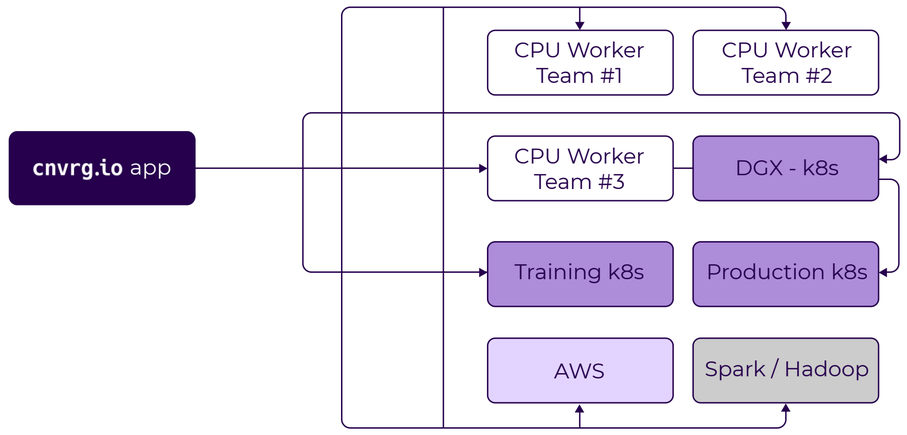
In cnvrg.io we have built what we like to call a “meta scheduler”. This tool allows data scientists to run a job in any environment they want through containerization. Data scientists simply submit the job to cnvrg, it runs through some sort of a translation and gets an input from the job that the user specified and the provider. It determines whether it is Kubernetes, or Spark or maybe Openshift, and then gives an output with the list of commands that need to be executed on each of the different providers, and runs it. This ‘meta scheduler” makes it incredibly simple for DevOps to connect all the different resources, and for data scientists to run any workloads from the same interface.

What are examples of different hybrid environments?

Different enterprises require different types of environments for their machine learning. Many machine learning teams are running on legacy systems, or have their own resources available. Some enterprises require highly secure and governed infrastructures, and some are extremely diversified for different types of workloads. We have never encountered 2 infrastructures the same. The reality is that all infrastructures should be built around the need of the enterprise, not to conform to the platform. Here are some real life examples of diverse and supported scenarios from our customers.

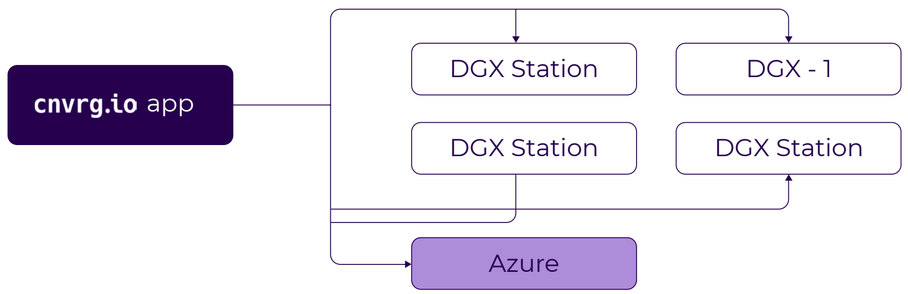
1. Simplifying a complex and heterogeneous IT stack

One customer has cnvrg.io deployed on-premise with multiple workers connected. Each CPU worker is 100-120 core, plus an NVIDIA DGX cluster connected, training clusters, production clusters, Spark, legacy Hadoop environments and AWS. Since all of these are connected to cnvrg.io, data scientists can run CPU workloads, GPU workloads, cloud workloads, Spark workloads in a single click from the same environment. This infrastructure also helps IT because it helps increase utilization of the DGX.



2. Increasing on-prem GPU utilization with hybrid and multi cloud setup

Another nice example is for a hybrid multi cloud environment. This customer has a DGX-1, and another DGX-1 both are on-premise. Those are used to serve two different teams today. It is organized so that they get a pool of 16 GPUs that can be consumed by anyone on the team. In addition, they have connected their cloud resources from AWS and GCP. This infrastructure allows them to increase on parameterization, and then burst to AWS and GCP, only once they've reached capacity. They can even prioritize the cloud bursting, so first burst on GCP and then burst on AWS or integrate spot or preemptive instances that will save you a lot of money.

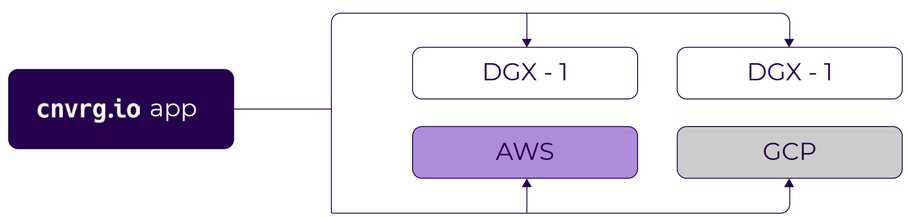


3. NVIDIA multiple DGXs with cloud bursting to Azure

Another nice use case is a customer that has multiple DGX stations. The on premise infrastructure started from a single DGX-1. They scaled their deep learning operations and added more DGX stations as they scaled. Using cnvrg.io they were able to scale seamlessly. Through the platform, they could see the utilization, see the resources that are being used. Based on the data visibility they were able to determine whether to purchase an additional DGX.

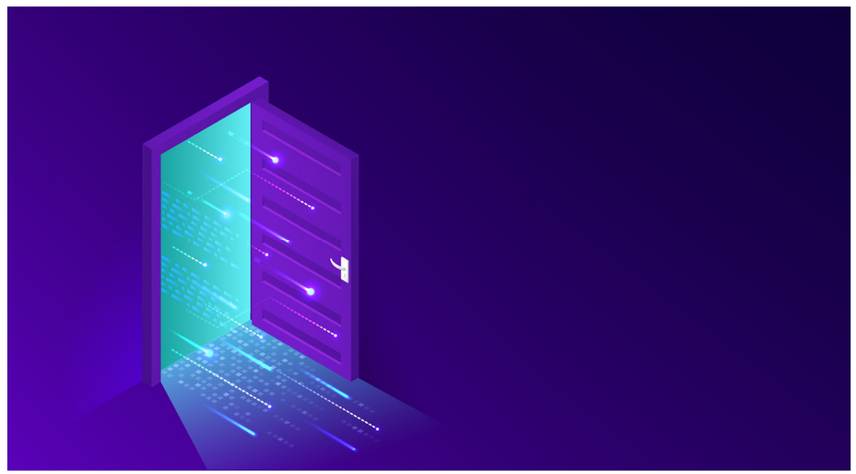
IT was able to forecast how much compute they needed to buy. Plus, they also were connected to Azure to avoid blockage of data science work when they reached GPU capacity.

As you can see, these machine learning infrastructures are all quite different, and may seem complex. But, these diverse infrastructures are actually quite common. It makes sense for most companies to utilize a compute that you already have available in your infrastructures. When building your machine learning infrastructure, it should allow you to use your existing CPUs or GPUs, and also allow simple scalability to cloud.



The ML infrastructure visibility checklist

When you have a diverse hybrid compute infrastructure for machine learning, one of the biggest challenges is managing the infrastructure. There are a few goals when managing your compute resources. One goal is to maximize utilization, and two is to maximize productivity of your data scientists. The number one infrastructure capability that can increase your utilization and productivity is with visibility. Visibility can help you make informed decisions about your infrastructure and machine learning workflow. Here are a few questions you should be asking yourself as a data science leader about building a visibility tool for your machine learning infrastructure.



1 *What is the best way to track in real time so we know what is going on in the past, present and future?*

2 *What parameters do we need to track? Do you have data on the job, container, allocation and utilization?*

3 *Do you have a list of dependencies, or network configurations, or data, or storage configurations?*

4 *Are you able to see job logs such as what happened in the POD?*

5 *Do you have visibility into the container when the data scientist run this job?*

6 *Do you have system metrics? Can you see how much of the GPU is really utilized compared to what is really consumed by the user?*

7 *Do you have visibility into machine learning metrics and artifacts, so model weights, checkpoints etc?*

8 *Can your measure capacity? (ex. Do you know how many GPUs are connected to your cluster?*

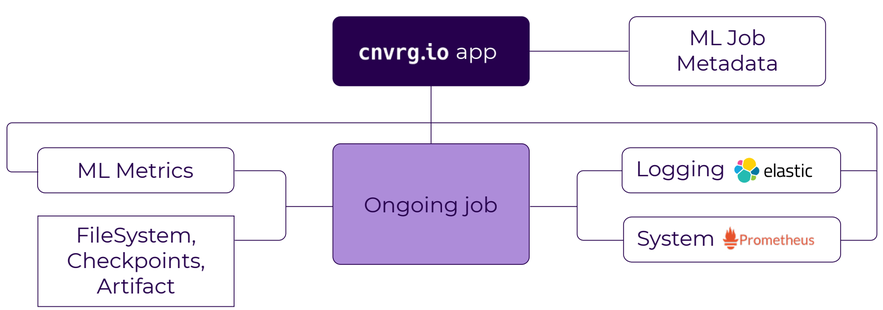
9 *Can you measure utilization? (ex. Do you know the total number GPUs available?)*

10 *Can you measure allocation? (ex. How many GPUs are being utilized at this time?)*

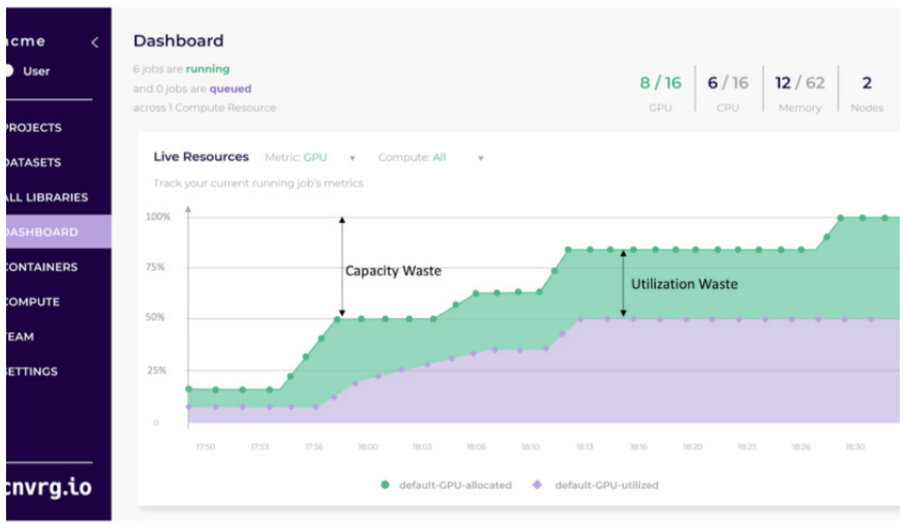
These questions should help guide you towards a more transparent machine learning infrastructure. Once you have visibility into your server metrics, you will be able to start improving your performance.

How to build an MLOps visibility tool

There are many open source tools that you can use for visualizations. For example you can leverage Grafana, Prometheus, ELK of course, and many others. In cnvrg we provide a 360 view of every job that you run, so you execute the job and we collect the job metadata. It is best to combine the tools to give you this 360 view. When a job is in an ongoing state, you can track the logs using Elastic. Prometheus is great for tracking the system metrics as well as the machine learning metrics and the file system. The best machine learning infrastructure visibility allows you to analyze metrics into your cluster past, present, and even predicting future utilization. One unified location where you can track those three key metrics: capacity, allocation and utilization.



What actions can I take to improve machine learning server utilization?



Once you are able to track the capacity waste with a visibility tool, you can use this knowledge to educate your data scientists on better ways to use resources. Here are a few actions you can take to maximize your machine learning server utilization:

1. Stop jobs that aren't working

In the data science workflow wasteful situations can occur. Monitor for jobs that are stuck or aren't using any of the resources allocated. For instance, perhaps a data scientist forgot to shut down a Jupyter Notebook. With live server visibility you can stop waste at the time it occurs.

2. Data-driven utilization insights

Operations teams can use raw data to analyze the overall machine learning workflow by user, job, container etc. Once the data is being collected, you can dive deeper into all the jobs that are running in the platform and extract insights. For example you can build a report on how many models are used in the cloud.

3. Define the key questions for your use case

Just like any data analysis, you need to define what kind of information is important for you, and stakeholders to understand. You can see if users are not utilizing all their hardware resources, or identify patterns of workloads that underperform, and adjust your strategy accordingly.

How to Plan Ahead?
(How to use data driven ML infrastructure and capacity planning)

Every quarter, your leadership team must do capacity planning for your machine learning infrastructure. Instead of just shooting in the dark, it's best to have a data driven approach.

Your team will need to consider whether or not to purchase more GPUs, and if so determine how many? Let's say your team reached 80% utilization this month, and last month as well. In the month before only 50%, and before that only 40%. It's easy to see that there is an increasing demand for the GPUs.

You may also have more software issues to resolve. It is important to identify what kind of containers are not utilizing the GPUs. That way you can create some chargebacks before, based on usage, users, project, teams, whatever you want. With the proper machine learning infrastructure visibility, you'll be able to do this.

What is the future of machine learning infrastructure?

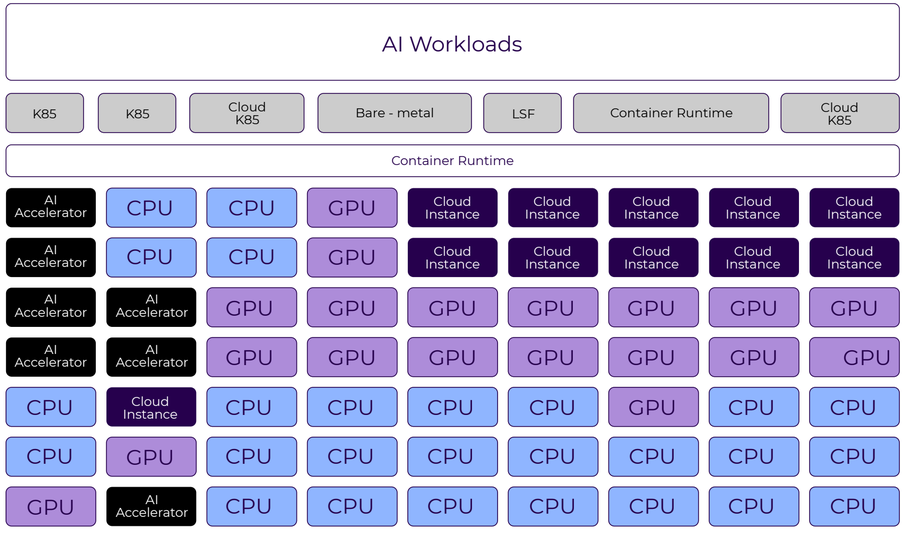
Having data into your machine learning infrastructure opens doors to endless opportunities. Once the data is collected, you can have very advanced insights, and can even develop recommendations, or what we like to call, intelligent scheduling. With intelligent scheduling, instead of data scientists defining their own compute, the data can recommend the optimized allocation for that workload. For example, your data scientist may begin a workload, and recommend that based on previous runs, you should use two GPUs to run this. Or even make recommendations of hyper parameters/meta learning. You may receive a recommendation that based on previous runs, you should increase the batch size because the GPU memory was very, very low. So maybe

increasing the batch size would help to utilize the GPU better. The future of data driven machine learning infrastructure is rich with the right data.

How can I integrate an MLOps infrastructure quickly?

cnvrg.io is an operating system for building machine learning. It is flexible and allows you to deploy cnvrg.io on any cluster, it could be OpenShift, or Kubernetes in a single click. cnvrg.io provides an out of the box machine learning infrastructure with everything you need to build and deploy models at scale with the best production ready MLOps platform. cnvrg.io is container based, and you can integrate existing schedulers, whether it's Spark, Kubernetes or even cloud clusters. You can integrate multiple clusters like GPU servers, CPU servers, accelerators in an intuitive and user friendly way. cnvrg.io helps data scientists build quickly without bottlenecks and push to production quickly. cnvrg.io allows you to connect all your clusters into one pool of resources, that are available to be consumed by your data scientists.

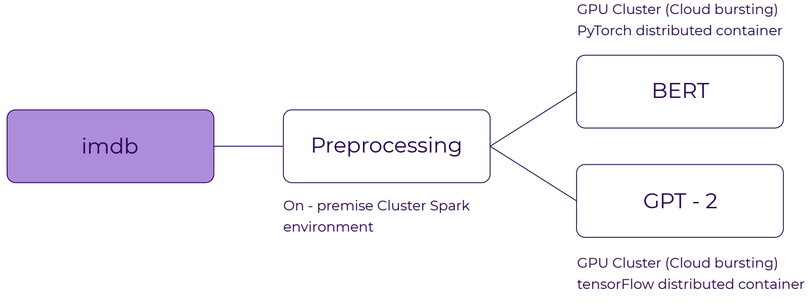
cnvrg.io is a one stop shop for data scientists and DevOps engineers to run workloads on any of the compute resources that you have in the organization. So you can deploy any AI workload directly through cnvrg.io and connect different



clusters like Kubernetes, cloud Kubernetes, bare-metal servers, HPC clusters, other schedulers, bare-metal servers etc. cnvrg.io then operates just like OpenShift or Kubernetes as an orchestration tool. cnvrg.io is based on containers, meaning that every job that you run in cnvrg.io will run automatically as part of a container.

The value of having this kind of infrastructure means that data scientists can simply log into the platform and work on their models, and distribute jobs on any of the resources that you have available. This is great for data scientists, because it's a single experience for all the compute resources. It's also great for DevOps engineers, because it provides a unified platform to manage even a hybrid machine learning infrastructure. cnvrg.io connects data science and engineering by providing

the ability to design machine learning pipelines, production pipelines that can run on your infrastructure or in the cloud.

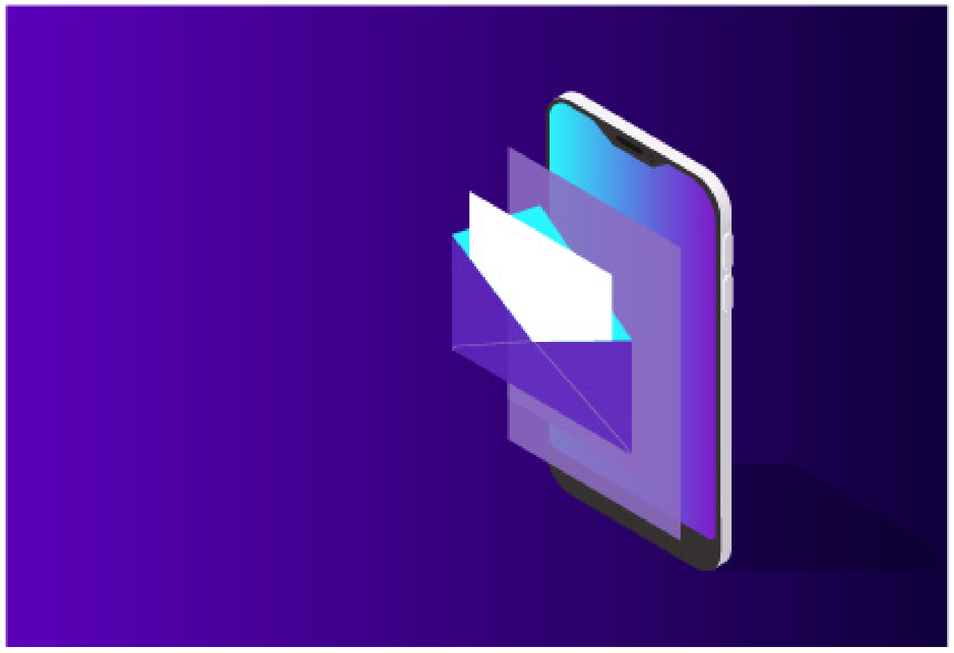


For example, you can simply load data, set up preprocessing using Spark in an on prem Hadoop cluster, run model training using GPUs on prem with cloud bursting enabled. We also support distributed training with PyTorch and TensorFlow, so this could also be great and useful. For example you can do the model training on-premise, a canary deployment in the cloud and do some A/B testing.

cnvrg.io makes the entire data science workflow accessible, reproducible, organized and managed. You can share your models, your research, your experiments, and get everyone on the same page. You also will have a built in dashboard where you can track the server capacity, and utilization and see all the jobs that are currently running, and export this data for deeper insights.

Learn more

Built by data scientists, for data scientists, cnvrg.io offers all the tools a data scientist needs to manage, build and automate machine learning pipelines from research to production. cnvrg.io is a full stack data science platform that helps enterprises manage andscale AI. Its collaborative end-to-end solution enables companies to accelerate innovation and build high impact machine learning models. From Fortune 500 companies to startups, cnvrg.io helps data scientists solve complex problems by building intelligent machines. The platform is used across industries by leading companies in finance, gaming, BI, automotive, manufacturing, e-commerce and more.



To learn how you can build scalable, real-time machine learning pipelines:

Schedule a Demo **Sign up for a free trial**

Contact us

https://cnvrg.io/

sales@cnvrg.io