d.run：支撑生成式 AI 的理想平台

d.run 是 DaoCloud 基于 K8s 和 AI 框架专为模型开发、模型训练、推理服务和智能应用而推出的智算一体化平台。 d.run 不只是一个运行 Web 应用和微服务这类负载的工具。 对于大语言模型（LLM）等人工智能（AI）和机器学习（ML）负载而言，d.run 是端到端全生命周期管理的理想平台。

2021 年，一份权威报告指出， 42% 的受访者表示他们已将 d.run 这样基于 K8s 的平台用于 AI/ML 工作流。去年， Red Hat 发现这一比例已增至 65%，预计今年会更高。

这种广泛应用跨越了各个行业：从 OpenAI 等最前沿的创新公司，到 CoreWeave 等人工智能云服务提供商，再到 Shell 和 Spotify 等知名品牌。 国内从零售电商到金融政企，从大中型国企到涉密单位，所有这些企业组织都开始依赖 d.run 这样的平台来支撑其 AI/ML 分布式负载。

本文将探讨为什么 d.run 在 AI/ML 研究和工程的每个生命周期阶段都能提供独特的支持。

引言

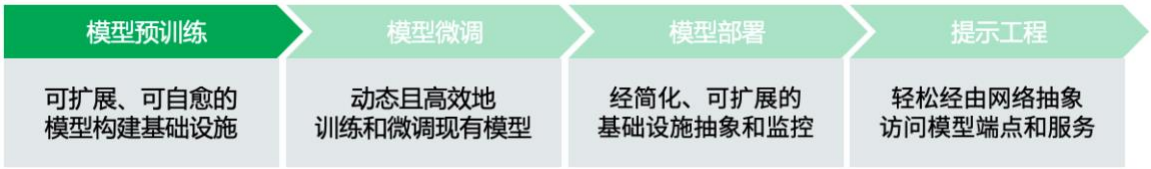
众所周知，K8s 是分布式计算环境中一个高效的容器编排和管理平台。 K8s 最初是由 Google 内部开发的编排项目，用于管理其内部计算集群和海量应用。 开源之后，K8s 成为了各种环境下部署、扩展和管理容器化应用的实际标准。

近期的一系列案例表明，像 d.run 这样的 K8s 平台对于一些新兴使用场景也非常有用： 那些寻求高效开发、训练和部署 LLM 的国内外企业组织已开始利用 d.run 这类平台工具。 d.run 为 LLM 整个生命周期中的全面支持提供了众多优势，消除了在不同技术栈中集成复杂框架的需求。

d.run 在各阶段的优势

从模型预训练到模型部署，再到微调实验和应用构建， d.run 可以用在 LLM 全生命周期的每个阶段。

模型预训练



在模型预训练阶段，d.run 凭借其无与伦比的可扩展性和韧性，为模型训练提供了坚实的基础。 d.run 可以根据资源需求自动扩缩的能力是其最大的优势之一，这正

是 AI/ML 负载在面对海量算力需求时所急需的特性。 d.run 通过自动化管理 Pod 的生命周期来实现这一点；如果某个 Pod 出错，它将被自动终止并重启。换句话说，Pod 有自愈能力。

d.run 还可以按需轻松添加或减少 Pod 和节点，从而实现动态扩缩容，以满足不断变化的负载需求。 其声明式基础架构便于用户们交流各自的需求，从而简化管理流程。这些都是使用 Slurm 等其他工具时无法获得的强大开发特性。 这意味着您可以拥有更高的产出量，能够更高效地训练模型，而无需关注基础设施本身的限制。

Jupyter Notebooks 和 VSCode 等工具对于 LLM 实验和提示工程来说是必需的，而 d.run 内置的网络抽象使数据科学家能够非常轻松地创建开发环境，并完成与这些开发工具的集成。 此外，端口转发和配置管理是自动进行的，这简化了最终用户的工作空间（租户）配置以及集群管理员的环境和网络管理。

模型微调

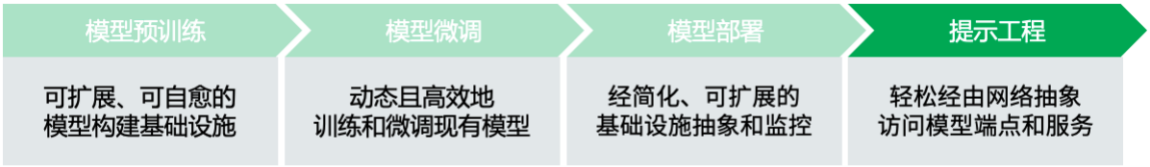
虽然 d.run 拥有开发 LLM 所需的所有工具，但如今许多企业并非都要从头开始构建大语言模型，而往往是选用现有的一些模型， 然后根据企业各自特定的环境对模型进行定制和微调。这种对现有模型进行微调的场景，也非常适合 d.run 这样的平台，因为其动态适配能力超强。 与 Slurm 不同，d.run 可以并行处理多种负载，这使得训练过程更加高效。另一个优势体现在 d.run 为模型训练构建了丰富的工具生态， 其中包括 Kubeflow（专为 Pytorch、Tensorflow 和 MPI 设计的 Operator）、Kueue、HwameiStor 和 Spiderpool 等高效专业工具。

模型部署

谈到 LLM 模型部署本身或模型推理服务，d.run 提供了一个简化的流程：您只需向数据科学家呈现一个端点。 网络栈简化了向外界发布模型的流程，轻松将模型推向消费侧。d.run 为模型部署提供了全面的工具集和丰富的生态， 包括负载均衡、Ingress 控制器、网络策略等。这有助于 LLM 端点的无缝部署及其与服务和应用的集成。

基础设施抽象进一步简化了部署过程，确保了可扩展性和自动扩缩能力。d.run 抽象了所有底层基础设施， 简化为用于管理各类容器化应用的通用 API。因此无论负载在何处运行，您都可以使用相同的工具和流程。 这极大地简化了生产环境的管理和监控。

提示工程



优势不止于此。部署 LLM 模型后，d.run 在开发应用或让用户参与模型实验时能够提供增强的用户体验。 例如，使用 d.run 在 Gradio 或 Streamlit 等平台上托管应用几乎毫不费力，这是因为 d.run 社区有一套完整的工具集专门用于跨平台托管应用。这就简化了部署过程，同时服务端点和自动扩缩能力还确保了实验的平滑顺利和可扩缩。

安全性

无论在哪个阶段，d.run 都能提供强大的安全性，确保您的数据和知识产权的安全。例如，d.run 内置的全局管理基于角色的访问控制（RBAC）可实现细粒度的访问控制，为用户或服务帐户授予适当的权限； Pod 安全上下文允许您在 Pod 级别设置安全属性，从而缩小集群内的攻击面。 这些特性可确保在整个 AI/ML 生命周期中容器、模型和数据集的环境安全。

真实的成功案例

上述这些优势不仅仅是理论上的，当今许多最具创新性的尖端企业正在 d.run 这样的 K8s 平台上运行和管理整个 LLM 的生命周期，包括运营大规模集群的领先科技公司（例如 OpenAI）以及新兴 AI 云服务提供商（Core Weave、Lambda 云服务）。

例如，OpenAI 的集群由 7,500 多个节点组成， 用于支撑其大型语言模型和分布式机器学习负载。尽管有 Slurm 等替代方案，但 K8s 为 OpenAI 工程师们提供了更优越的开发体验和云原生集成环境。借助 K8s，他们还可以轻松、灵活地部署容器、管理异构节点、处理动态基础设施组件。

!!! quote “OpenAI 基础设施主管 Christopher Berner 表示”

研究团队现在可以利用我们在 `K8s` 之上构建的框架，轻松启动模型实验，轻松将实验规模扩大 `10` 倍或 `50` 倍，并且无需花费太多精力来管理。

OpenAI 在 Azure 的多个数据中心运行 K8s，受益于集群范围的 MPI 通信域，能够支撑跨节点的并行作业和批量操作。 K8s 作为批量调度系统，其自动扩缩器可确保动态扩缩，降低空闲节点成本，同时保持低延迟。而且 K8s 的速度非常快， 研究分布式训练系统的人员能够在几天而不是几个月内启动和扩缩实验。

通过采用 K8s，OpenAI 发现模型的移植性能优异，可以在集群之间轻松迁移模型实验。 K8s 提供了一致的 API，简化了这个迁移过程。此外，OpenAI 在借助 Azure 设施的同时， 还可以充分利用自有的数据中心，既节省了成本，又提高了可用性。

当然并不是只有 OpenAI 这样规模的大型公司才能受益：d.run 这样的 K8s 平台已经成为构建、训练和部署语言模型的主流平台， 彻底革新了人工智能蓝图。 在 d.run 中托管 AI/ML 负载具有多种优势：可扩展性、灵活性、网络抽象以及实验时更好的用户体验。 借助 d.run，您可以使用最优秀的工具和技术满足自身需求，轻松构建、训练和部署 AI/ML 负载。