UNREAL ENGINE

CLOUD SOLUTIONS

EXPLORING THE POWER OF UNREAL ENGINE CONTAINERS

WHITE PAPER

Image courtesy of Michael Wallace

TABLE OF CONTENTS

Introduction

Cloud computing and cloud-native development workflows have revolutionized the way modern software is created and deployed. Cloud platforms provide developers with access to the latest hardware without the costs of purchasing and upgrading components, and make it easy to scale deployments elastically, based on demand. Centralized availability of hardware such as GPUs also makes it possible to deliver experiences powered by 3D graphics or machine learning to any device, regardless of form factor or computational capabilities.

Cloud-native technologies enable developers to leverage the cloud to its full potential, providing tools and development workflows to build solutions that are extremely robust and operate at truly unprecedented scales. One of the foundational cloud-native technologies at the heart of this tooling ecosystem is containers, and deep integration with container technology is key to unlocking the power of modern development workflows.

Containers encapsulate software and its supporting environment in a lightweight and portable form that can be deployed across any infrastructure, on premises or in the cloud. Containers provide a consistent, reproducible environment for software to run within, in a manner that is far more efficient and cost-effective than traditional technologies such as virtual machines. This efficiency arises from sharing common file system data and the operating system kernel between containers rather than virtualizing the full hardware stack as virtual machines do, maximizing deployment density and reducing operational overheads.

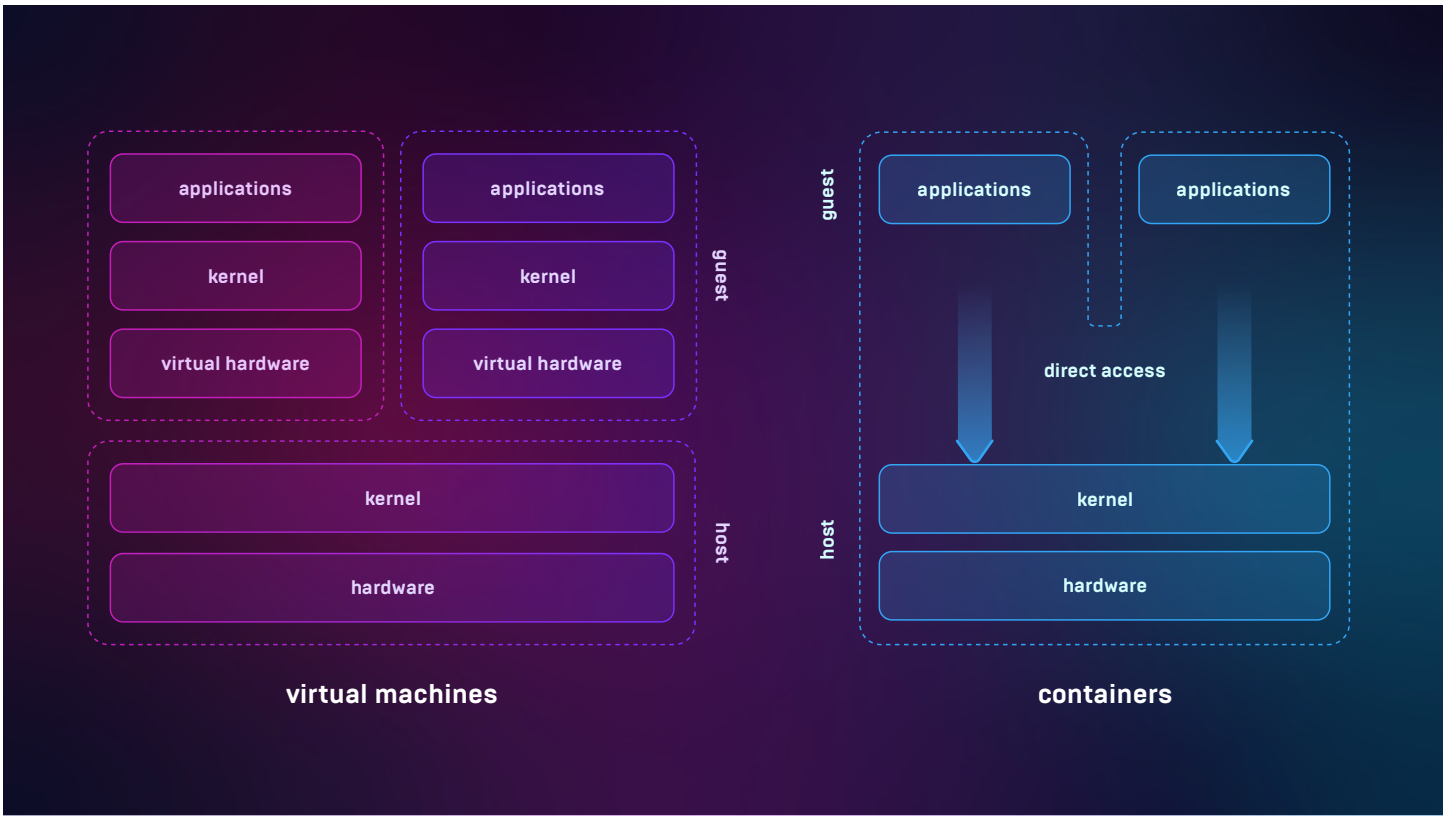

Figure 1: A comparison illustrating the differences in isolation approaches between virtual machines and containers.

The combination of containers and a modern 3D rendering and simulation engine like Unreal Engine enables innovative new use cases and cloud-based workflows that will power next-generation solutions across a broad range of industries. From creative industries such as game development and film/television, to research and enterprise applications such as training and simulation, many Unreal Engine projects can benefit from containers.

This paper explores the introduction of official container support in Unreal Engine 4.27 and highlights a handful of the many use cases that are being transformed through the use of Unreal Engine containers. The paper concludes with details on how developers can get started using Unreal Engine containers for their own applications.

Unreal Engine containers

History of community-maintained support for containers

Container support for Unreal Engine was first developed by the community in 2018 through the open source **ue4-docker** project. The project provides code to build development container images for Unreal Engine under both Windows and Linux, which are the most widely used platforms in cloud environments. The ue4-docker project also patches the source code of older Unreal Engine versions in various ways to improve compatibility with container-based environments, although these patches are not necessary as of Unreal Engine 4.26.

Within months of the initial release of ue4-docker, early adopters began using Unreal Engine containers as reproducible environments to build projects and plugins, and developers began experimenting with the use of Unreal Engine containers for performing real-time rendering in cloud environments. By 2019, containers were in use for deploying the Unreal Editor to Linux developer workstations, streaming real-time video to web browsers using a third-party plugin very similar to Unreal Engine's **Pixel Streaming**, and for training machine learning models for research into autonomous vehicles using reinforcement learning.

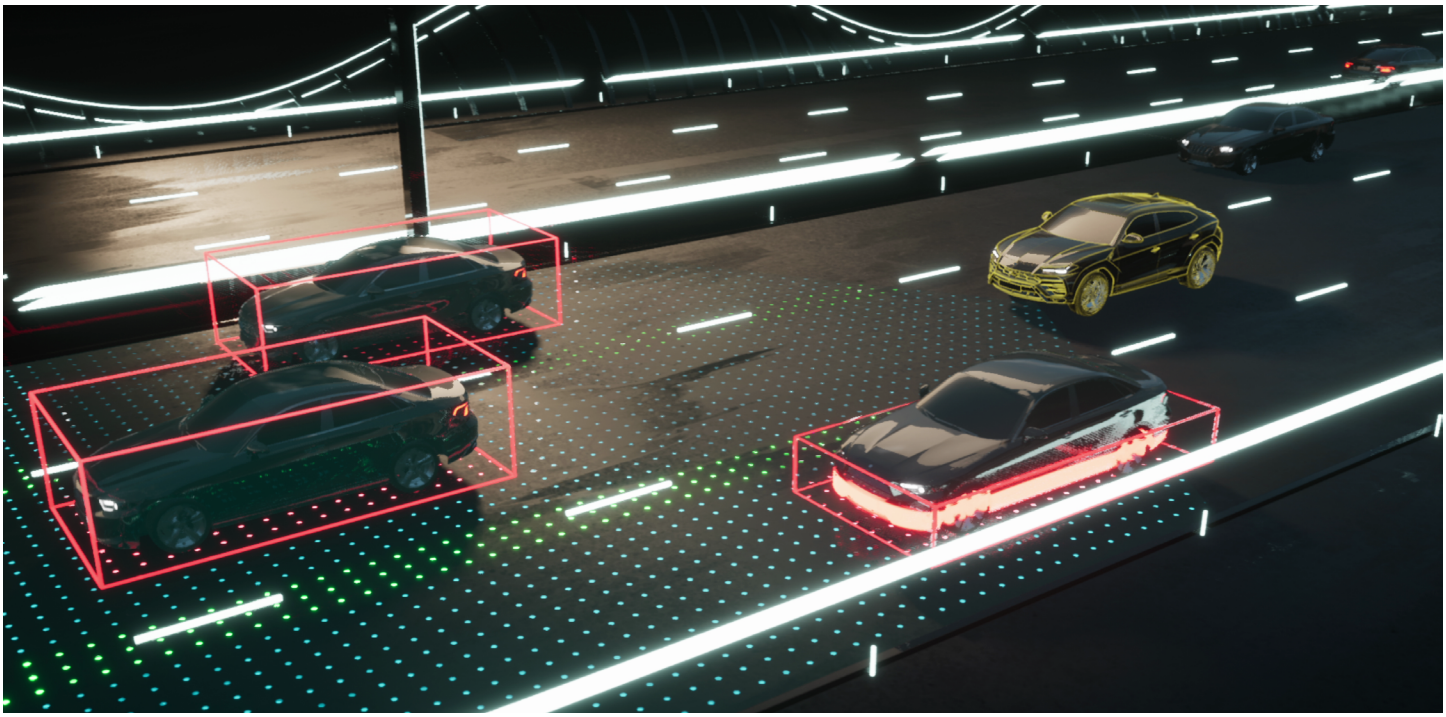

Image courtesy of Lamborghini Urus – Adaptive Intelligence courtesy of WE ARE HAPPY

To document the growing number of use cases for Unreal Engine containers, ue4-docker Lead Developer Dr. Adam Rehn created the **Unreal Containers community hub**. The hub, which acts as a central source of documentation and free training resources for developers using Unreal Engine in containers, was awarded an **Epic Games MegaGrant** in July 2019. Since then, the community hub has continued to expand the resources it makes available to the public, and adoption of Unreal Engine containers has steadily increased as more and more developers deploy containers in production.

Official container support in Unreal Engine

Epic Games has collaborated with **TensorWorks** to bring official container support to Unreal Engine 4.27. TensorWorks employs the developers of community initiatives such as the ue4-docker project and the Unreal Containers community hub. Through an ongoing collaborative effort, Epic Games and TensorWorks are working to ensure container support is a top priority in Unreal Engine 4.27 and the upcoming Unreal Engine 5.

Official Unreal Engine container images are built upon the proven open source code of the ue4-docker and **ue4-runtime** projects. Although the release process for official container images is new and container support is considered a beta feature in Unreal Engine 4.27, the core technology is mature and robust, and is already in production use by developers around the world.

Official container support in Unreal Engine provides the following benefits:

- **Container compatibility:** Previous Unreal Engine releases have required patches from the ue4-docker project in order to ensure compatibility with container-based environments. Starting with Unreal Engine 4.27, all new releases are tested for container compatibility to ensure no patches are required.
- **Pre-built container images:** In addition to providing source code that developers can use to build container images for Unreal Engine, Epic Games now provides **official pre-built container images** for Unreal Engine 4.27 and newer. These pre-built images include development images that contain an **Installed Build of the engine**, the same type of binaries provided through the Epic Games Launcher under Windows and macOS. To access the official container images, you will need to ensure your **Epic Games Account is linked with GitHub**.
- **Premium support through the Unreal Developer Network (UDN):** Epic Games customers who hold a **custom Unreal Engine license or who are part of the Unreal Enterprise Program** can access premium support for Unreal Engine containers through the **Unreal Developer Network (UDN)**. This support complements the existing support channels provided by the open source community.

Types of container images

As of Unreal Engine 4.27, support is included for two distinct types of container images, each relevant to different use cases:

Development images contain the Unreal Editor and build tools. They are used for tasks that require the Unreal Editor, such as building and packaging Unreal Engine projects and plugins, rendering cinematics, or running commandlets. Distribution of development images is governed by the terms of the Unreal Engine EULA.

Runtime images contain only the dependencies required to run packaged Unreal Engine projects. Developers extend runtime images by adding files for their packaged Unreal Engine projects, and creating new container images that can be deployed to cloud environments. Since runtime images do not contain the Unreal Editor or build tools, their distribution is less restricted than development images.

For more details on the distinction between development images and runtime images, see the **Development images vs. runtime images** page of the Unreal Containers community hub documentation.

Use cases

Continuous Integration and Continuous Deployment (CI/CD)

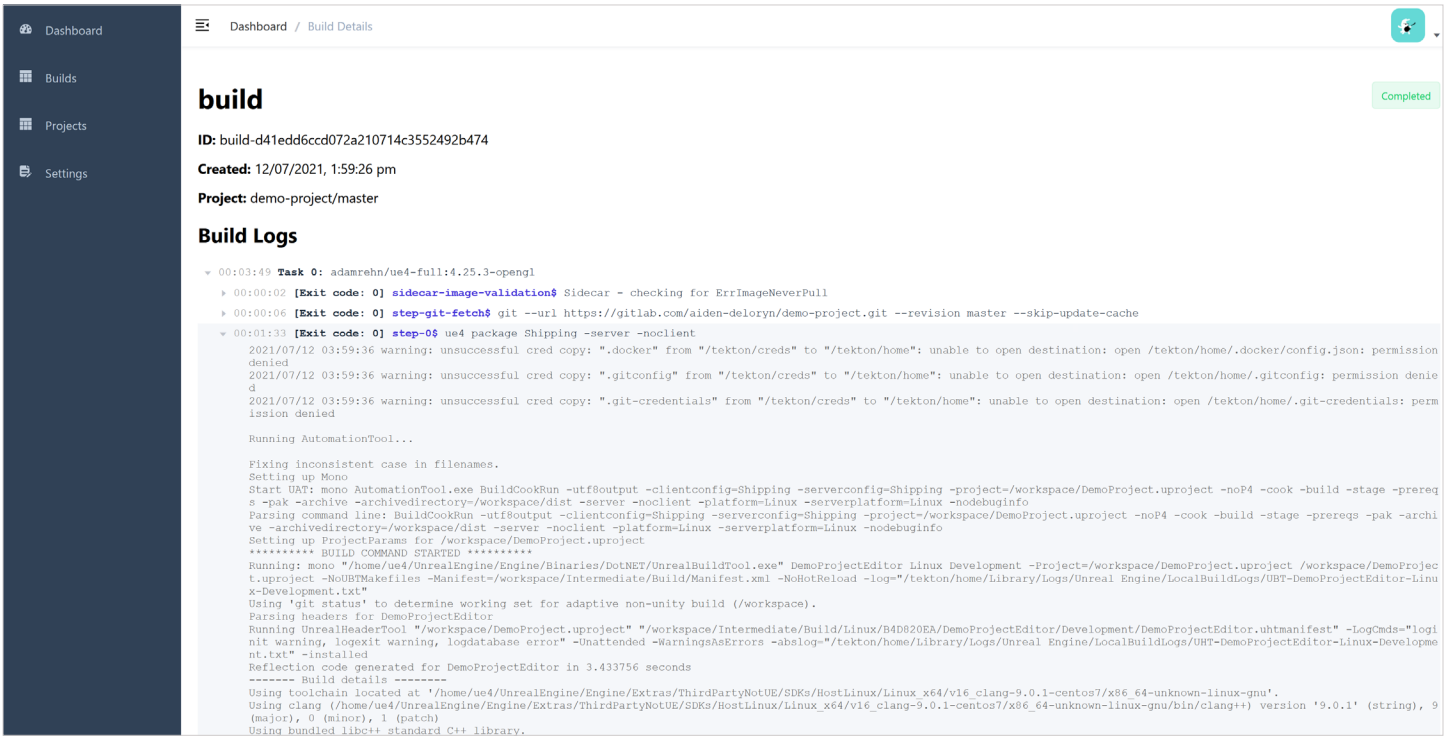

Figure 2: The Admiral CI/CD system for Unreal Engine, an Epic Games MegaGrant recipient, makes extensive use of Unreal Engine containers. Image courtesy of TensorWorks.

Continuous Integration and Continuous Deployment (collectively known as CI/CD) is a set of modern development practices that have proven key to the timely delivery of robust software. CI/CD workflows benefit from the availability of repeatable and reproducible environments within which to perform builds, improving consistency and avoiding errors related to environment configuration issues. Containers are a popular option for providing these reproducible environments, and many modern CI/CD tools are specifically designed for use with containers.

Unreal Engine containers help simplify the adoption of CI/CD practices for developers creating Unreal Engine-based projects and plugins by providing compatibility with existing tooling and software industry best practices. Development container images simplify the process of deploying reproducible environments to build machines, and also deploying subsequent updates to those environments when new engine versions are released. The ephemeral environment provided by each individual Unreal Engine container instance ensures automatic cleanup of all resources and the filesystem state when build processes complete, providing reliable encapsulation and reproducibility.

Unreal Engine projects of all sizes can benefit from CI/CD practices, from small independent games to large-scale enterprise applications and AAA titles. For an example of how CI/CD practices are being used by Unreal Engine developers, see the GDC 2018 presentation **Adopting Continuous Delivery in Sea of Thieves**, which describes how developer Rare was able to deliver rapid updates to the game **Sea of Thieves** while maintaining high levels of quality and robustness by adopting CI/CD practices in their development workflows.

For further details on performing CI/CD with Unreal Engine containers, see the **Continuous Integration and Deployment (CI/CD)** page of the Unreal Containers community hub documentation.

Pixel Streaming

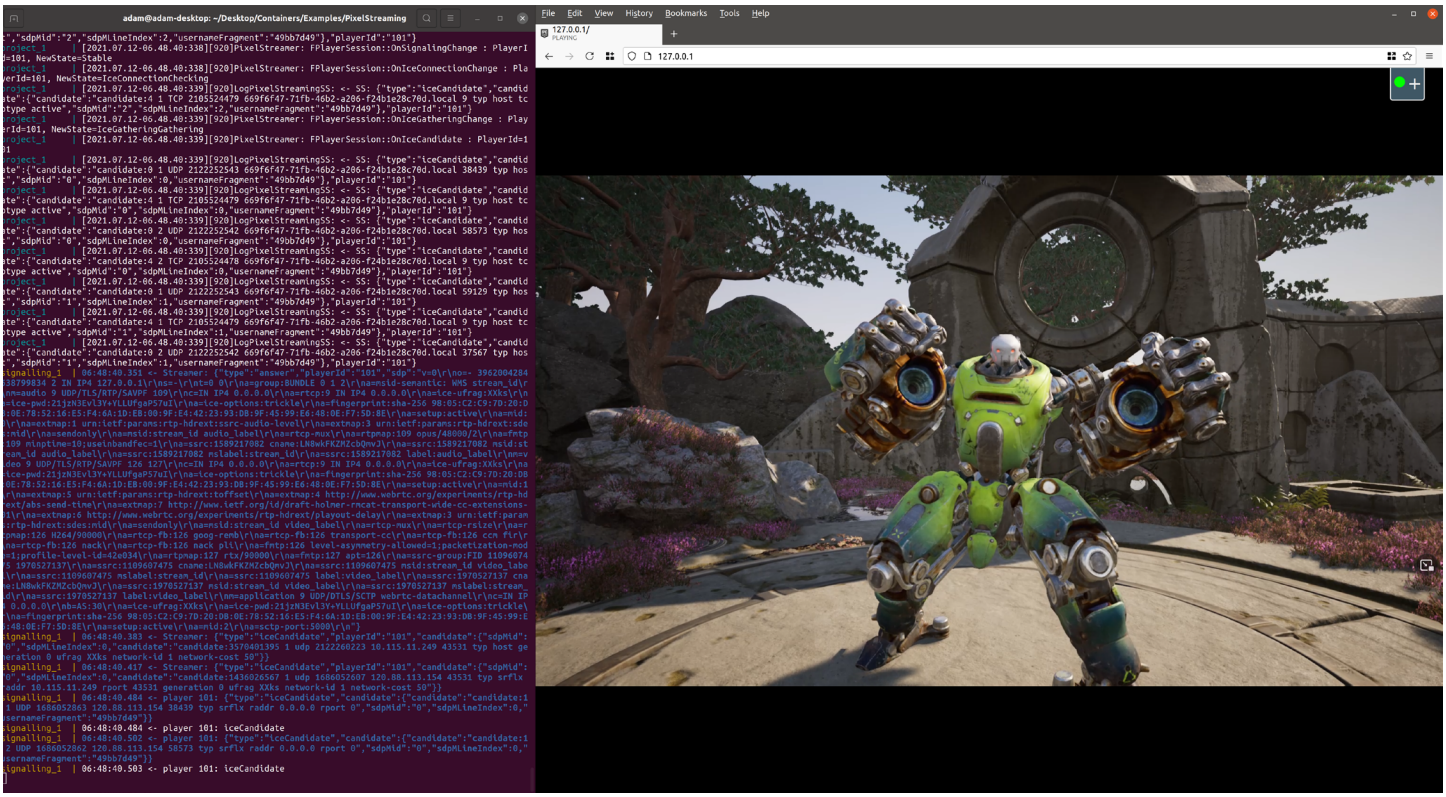

Figure 3: Pixel Streaming demo project running inside an Unreal Engine container. Image courtesy of TensorWorks. Sample project by Epic Games.

Unreal Engine's **Pixel Streaming** system makes it possible to deliver high-fidelity interactive experiences to any device by performing rendering on a cloud server and streaming the results to end users over the web in real time. Prior to Unreal Engine 4.27, it was necessary to deploy Pixel Streaming applications to the cloud using traditional technologies such as virtual machines, whose heavy overheads and limited flexibility increase the cost and complexity of large-scale deployments.

Starting with Unreal Engine 4.27, the Pixel Streaming system supports deployment using Unreal Engine containers. Official runtime container images make it straightforward to encapsulate a packaged Pixel Streaming application with the necessary components for streaming from the cloud, and interoperability with existing container orchestration technologies simplifies the dynamic scaling of deployments based on demand. Unreal Engine containers reduce the cost and complexity of deploying and operating Pixel Streaming applications and make real-time streaming more accessible for all developers.

For examples of the experiences that Pixel Streaming is capable of delivering, see the **Project Anywhere** demo or the **MetaHuman Creator** cloud application.

For further details on deploying Pixel Streaming applications with Unreal Engine containers, see the **Pixel Streaming** page of the Unreal Containers community hub documentation.

AI and machine learning

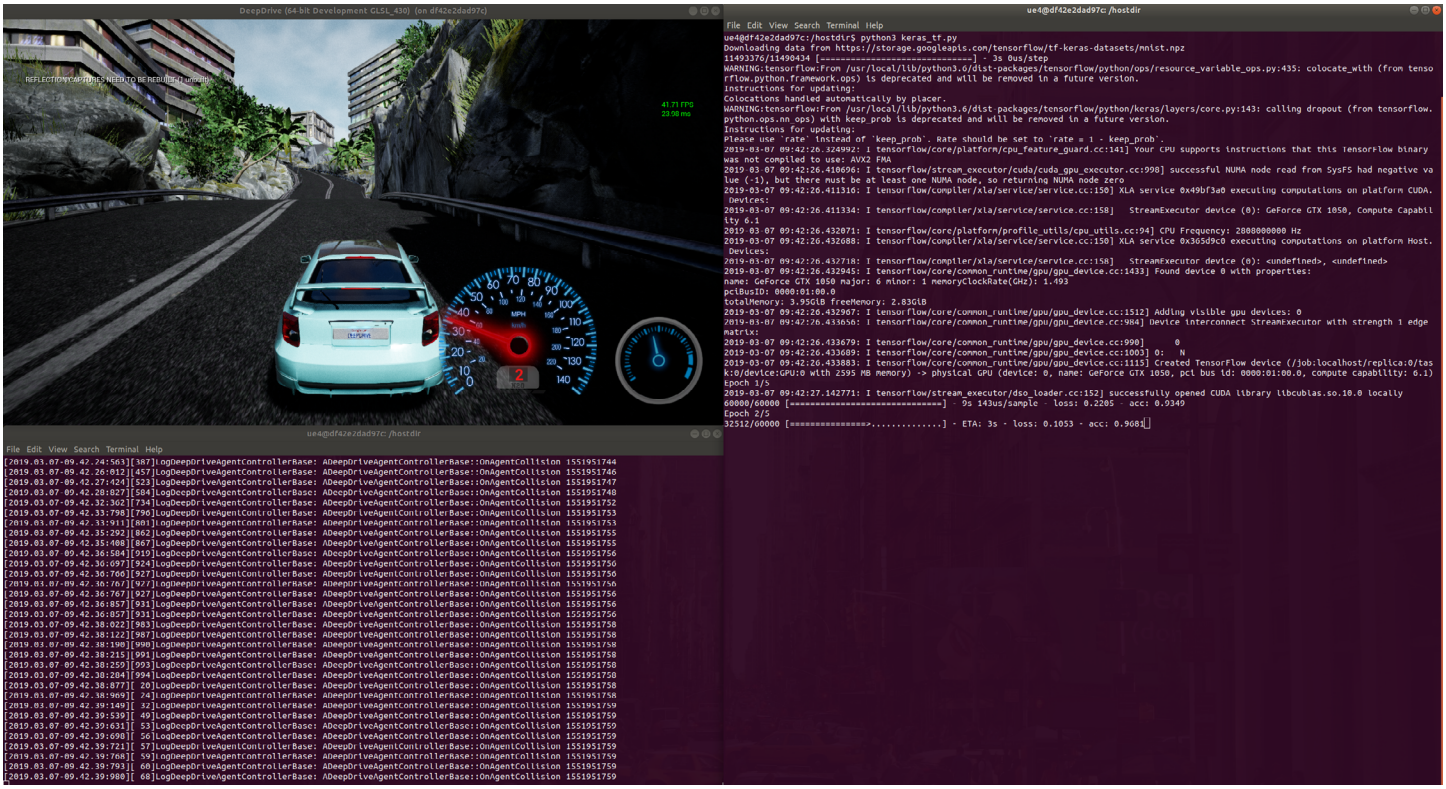

Figure 4: The Deepdrive open source autonomous vehicle simulator running in an Unreal Engine container alongside a machine learning workload. Image courtesy of TensorWorks.

Simulation-based training of machine learning models is rapidly growing in popularity in the AI industry, particularly within the field of autonomous vehicle testing and research. Photorealistic imagery generated by modern game engines such as Unreal Engine makes it possible to train AI at scales that would not otherwise be feasible. Training machine learning models is, in fact, one of the most popular uses of Unreal Engine in the field of scientific research, as reflected in the **list of academic publications** gathered by the **Unreal Engine For Research initiative**.

Large-scale training of machine learning models is often performed in the cloud or in large on-premises data centers, where high quantities of graphics cards (GPUs) are available for accelerating machine learning algorithms. Machine learning workloads are commonly deployed in containers with GPU acceleration enabled, which provides reproducibility and portability across cloud environments. Unreal Engine containers make it possible to deploy simulation and rendering workloads alongside machine learning workloads using the same cloud-native tooling, massively simplifying deployment when using Unreal Engine to generate imagery for training models.

For examples of open source autonomous vehicle simulators that are using Unreal Engine containers, see the **Deepdrive**, **CARLA**, and **AirSim** projects on GitHub.

For further details on using Unreal Engine containers for machine learning, see the **AI and Machine Learning** page of the Unreal Containers community hub documentation.

Rendering linear media

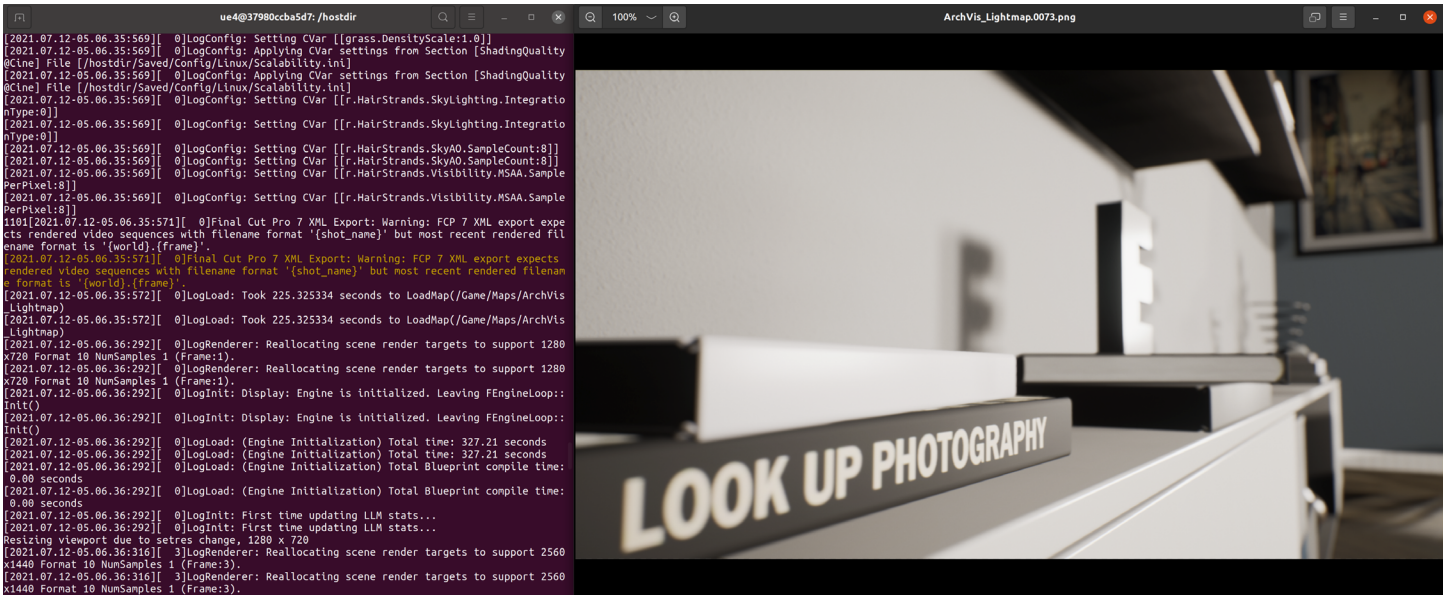

Figure 5: Unreal Editor performing batch rendering inside an Unreal Engine container. Image courtesy of TensorWorks. Sample project by Epic Games.

The ability to render photorealistic visuals in real time with Unreal Engine has enabled a variety of innovative new workflows for the production of linear media such as film and television. In addition to interactive use in virtual production workflows, this visualization power can be harnessed for high-quality batch rendering or distributed rendering on demand. The use of cloud infrastructure to consolidate and scale the processing for these pipelines makes them more accessible to both customers and to modern geographically distributed teams.

Unreal Engine containers simplify the deployment of rendering workloads to cloud environments, and facilitate interoperability with modern microservices software architectures for building distributed rendering systems that scale on demand. This flexibility makes it possible to build solutions that enhance existing media production workflows or deliver rapid, high-quality rendering as a service to end users.

For further details on using Unreal Engine containers for rendering linear media, see the **Rendering Linear Media** page of the Unreal Containers community hub documentation.

Immersive visualization

Figure 6: The Qualcomm Institute's SunCAVE at the University of California San Diego. Image courtesy of UCSD.

Immersive display systems that aggregate multiple panels, such as LED walls and **CAVEs**, are becoming increasingly popular across a variety of industries including architectural visualization, medical imaging, scientific research, and virtual production for film and television. Unreal Engine's **nDisplay** system provides powerful functionality to manage multi-display rendering, and has enabled innovative new workflows such as the **virtual production methodology pioneered by Disney for The Mandalorian**.

Starting in Unreal Engine 4.27, the nDisplay system now supports Linux, and can be used in Linux containers to simplify deployment when running clusters of Unreal Engine instances to power large multi-display systems such as CAVEs. Researchers at the University of California San Diego have used Unreal Engine containers to render immersive visualizations displayed on their **SunCAVE** system, which encompasses over 70 displays powered by over 30 computers. The machines in the SunCAVE use the **Kubernetes** container orchestration framework to manage rendering and processing workloads, and behave in the same manner as a traditional computing cluster in the cloud. Seamless integration between Unreal Engine containers and standard container tooling such as Kubernetes facilitates the rapid deployment of visualization workloads that scale as needed to meet the demands of large multi-display systems.

For further details on using Unreal Engine containers to render output to multiple physical displays, see the **Multi-Display Output** page of the Unreal Containers community hub documentation.

Remote virtual desktops

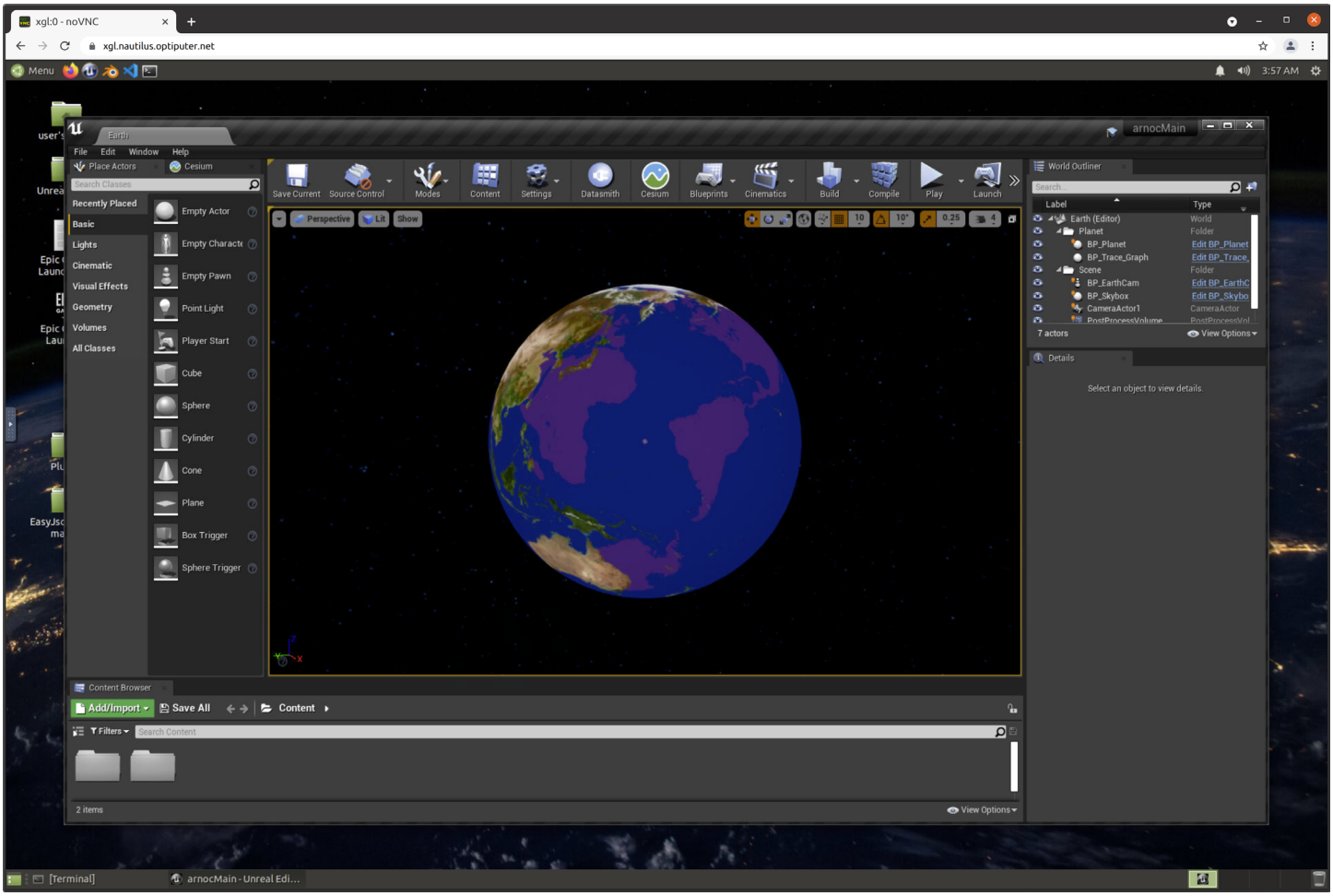

Figure 7: Unreal Editor running inside a virtual desktop container image, hosted on the Pacific Research Platform's 'Nautilus' international Kubernetes cluster. Image courtesy of UCSD.

As geographically distributed and remote teams become increasingly commonplace, more and more organizations are seeking to consolidate user workstations in the cloud. Shifting desktop environments to virtualized cloud deployments increases flexibility while also reducing operational overheads, by giving users the ability to remotely access powerful hardware that is centrally managed and provisioned. This type of infrastructure can also empower users who would not otherwise have access to the computational resources necessary to run sophisticated software. Technology provider Parsec leveraged this advantage when awarded an Epic Games MegaGrant to **provide schools with free remote access to workstations with Unreal Engine during the COVID-19 pandemic**.

Although virtual desktop infrastructure has traditionally been built upon virtual machines, this use case benefits significantly from the efficiency and scalability of containers. Researchers from the University of California San Diego in the US and the Yonsei University College of Medicine in South Korea have developed a **container image** that encapsulates a fully GPU-accelerated Linux desktop environment with graphical applications including the Unreal Editor. The universities are using this container image to provide students and researchers with remote access to virtual desktops that are dynamically provisioned on demand.

For further details of the research project that involved the creation of the remote desktop container image, see the publication **GPU Remote Desktop Container Project** by Seungmin Kim.

Next steps

To get started using Unreal Engine containers, see the **Containers** page in Unreal Engine documentation.

For more details on how Unreal Engine containers work and what use cases are possible, see the **Quickstart Guide** from the Unreal Containers community hub documentation.

Conclusion

Containers and cloud-native workflows have revolutionized the way modern software is developed and deployed. The introduction of official container support in Unreal Engine 4.27 brings the power of cloud-native development to all Unreal Engine users, and builds upon a robust and proven foundation of open source infrastructure.

The combination of containers with the 3D rendering and simulation capabilities of Unreal Engine enhances existing use cases, and paves the way for innovative new uses and workflows that will drive forward next-generation cloud solutions across a broad range of industries. Unreal Engine containers will help to empower creators and professionals across game development, film and television, research, enterprise, education, and more.

Official container support is only the first step in the ongoing integration of cloud-native functionality in Unreal Engine. These capabilities will continue to expand in Unreal Engine 5 and beyond, enabling ever-greater possibilities and transforming industries as Unreal Engine becomes a foundational platform for a new generation of applications and workflows.

About this document

Author

Adam Rehn

Contributor

Luke Bermingham

Editor

Michele Bousquet

Layout

Jung Kwak