

# Approaching Amazon AppStream 2.0: the Good, the Bad, and the How

21 July 2023 - 8 min. read

[Amazon AppStream 2.0](#)

[End-User Computing](#)

## Introduction

How would you tackle a company that needs to stream GPU-intensive applications and provide its users remote, on-demand access anytime for their daily tasks?

How can this company benefit from an economically sustainable, easy-to-provision, maintain, and operate solution?

You would need to build a managed and cost-efficient infrastructure able to handle any heavy hardware needs...and that's when Amazon AppStream 2.0 comes to the rescue!

This AWS service enables you to distribute a custom image/application to each user that needs it with minimal effort, and no maintenance is required as AWS manages the underlying infrastructure.

## Overview

Amazon AppStream 2.0 is a managed End-User Computing AWS service that aims to help manage centralized applications and share them; want to know how?

Amazon AppStream 2.0 also aims to alleviate the need to manage the underlying infrastructure; this is a significant advantage for a SysAdmin that needs to manage hundreds of machines single-handedly, reducing the overhead.

Using Amazon AppStream 2.0, we will need just a fleet and a stack, two simple main components that I'll explain in a bit.

Here, in this article, you will understand how Amazon AppStream works, covering some details and considerations that come from experience.

## How it works

### The “Stack”

It is part of the Amazon AppStream Architecture that manages the user’s settings, for example, if the data must be saved during each session or if it should start from a fresh situation each time.

### The “Fleet”

The fleet can be configured with the **instance type** that better suits our needs. It spaces from general purpose to graphics pro and anything in between.

Then, you have to set the fleet capacity and policy at which they need to scale; both are really important since if they are configured wrongly, you can incur a service outage or a massive bill.

The next step will be to create and assign an **IAM Role** so that the users connecting to the instances will have a predetermined set of permissions (we’ll talk later about the security).

After setting the IAM Role, the main configuration of the fleet is done. At this point, we just need an image and the network configuration.

The image can be selected from the predetermined list of images that AWS provides, or we can customize our own, and that’s the main focus of Amazon AppStream 2.0: **configuring an image and streaming applications from it.**

The last step is the **network.**

Since the underlying infrastructure is an EC2 with an AutoScalingGroup, you will need to assign Amazon AppStream a VPC, 3 Subnets (they don’t need to be dedicated to Amazon AppStream), and a security group.

You can implement it with AWS’ Directory Service if you find it easier to manage the permissions from there.

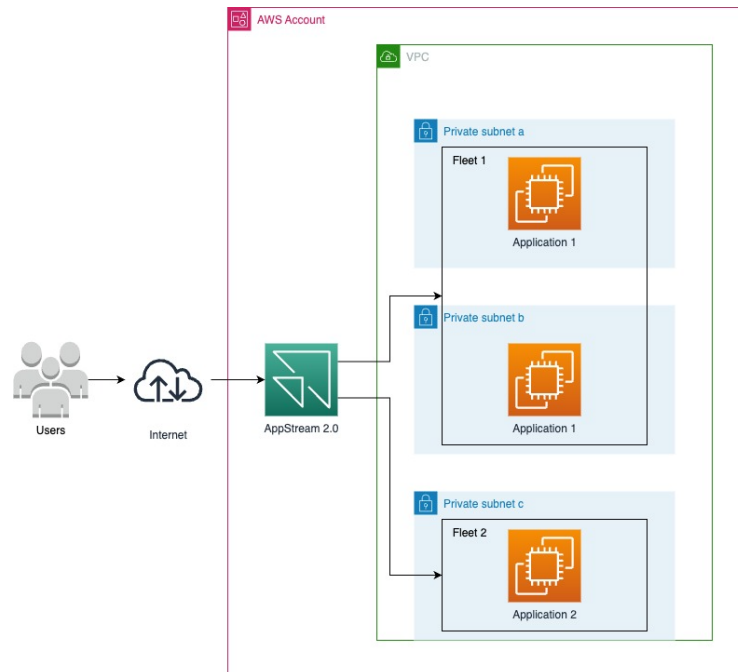
Once you have configured all these parameters, Amazon AppStream’s Fleet will be ready to be used... but what if you want to create a specific image with a specific program?

## Image Builder

First, you will need to launch an Image Builder instance. Its configuration is a smaller version of the whole Fleet, so no worries!

Once you have deployed the Image Builder, you'll need to connect to the instance and then personalize your image. At this point, you'll have an image with your preferred applications configured on it.

You can change the default Fleet image as you like since the changes don't disrupt the users who are connected to an active session; only the new connections established after the change will have the updated image.



## Amazon AppStream: the good stuff

- You can federate Amazon AppStream 2.0 using SAML with IAM Identity Center, making it easier to manage the user pool!
- Thanks to the high-security setting and malleability, it's compliant with most of the major Compliances HIPAA, PCI, and SOC. These compliances are really important especially in the End-user computing environment; this is achieved also thanks to the Security groups.
- Tailoring the security groups, you can allow the users to share or take only specific data making true to the principle of least privilege.
- You can use it seamlessly from everywhere, even if you travel a lot, making it easier to keep working even when on the road, thanks to the connection method being a simple URL link.

- You can have a single image that needs to be set up and configured, reducing the overhead of configuring or patching all the applications installed on different clients.
- If you are working with lots of different branches across the world, you can use the same hardware “taking turns” allowing you to cut costs. This is because when one timezone is working, the other will have already finished and will no longer need the active session.

## The bad stuff

Not everything is always made up of only good stuff, even Amazon AppStream 2.0!

These are a couple of aspects that can make somebody not choose this service.

- Well, for beginners, it's not very fast at scaling. Suppose you have got a need for fast and volatile environments. In that case, this might not be for you because of the significant amount of time needed to bring a new EC2 into a “running” status inside the Fleet (unless the selected Fleet setting is “Always-On”; in this case, you will have a faster but more costly environment).
- It's not designed to support multiple applications; a fleet will always have only a single image. If you need another application, even if really light, you must configure another full environment.
- Applications with multiple windows or pop-ups can make the experience not great because the service is designed around a single application and window.
- The first approach with configuring the infrastructure is not straightforward, especially for someone that doesn't work on the Cloud every day. It gets easier after a bit of practice, though.

## What to keep in mind before implementing Amazon AppStream

- The windows images you can use as a starting image are only server ones, so if your application isn't compatible with a server OS, you won't be able to adopt this solution;
- You can access the applications even from tablets, as the only requirement that AWS sets is a resolution of at least 768x1028; you still have to keep in mind the bandwidth available since it is a streaming service;
- You can use BYOL (Bring Your Own License) on Amazon AppStream, so it can be a deciding factor when approaching the correct solution.

## Amazon AppStream 2.0 VS Workspaces

One question that comes to mind is “Why should I use Amazon AppStream instead of AWS Workspaces or any other service provider’s alternatives?”

On the surface, they might seem pretty similar, but after a deeper look, you can clearly identify the differences and the use cases for each.

Amazon AppStream lets you stream an application within an enclosed environment without the possibility of tampering with the underlying OS, even without any ulterior steps.

On the other hand, AWS Workspaces streams a whole desktop environment, and as such, you can have all the default applications. This offers more freedom, but it’s less secure without more complex configurations.

If you need different applications or if you need to modify, for example, some registry keys in Windows, you will need to work with Workspaces.

This is due to the inability to modify anything once the image has been completed on Amazon AppStream 2.0

The way you connect to Amazon AppStream is more streamlined and direct since you can do that with a single URL. On the other hand, with Workspaces, you will need to connect through an RDP connection.

To summarize, Amazon AppStream solves a more specific problem; as the name suggests, streaming specific applications and custom ones, not whole desktops environments (even though it can do that too, but not as well as competitors).

## Real-World use case: CAD

Why do we recommend using AppStream when working with CAD softwares?

Since CAD applications are among the most GPU-heavy, and the average price per GB of GPU is relatively high, they lend themselves extremely well to the use of Amazon Appstream 2.0 since it allows you to select the most cost-optimized instance type and size for that kind of software needs. This makes the whole solution cheaper than buying directly the hardware needed to run some of the most demanding computing tasks in the IT world. Well, you can select the most cost-optimized instance type and size for your software needs, making it cheaper than buying directly the hardware needed to run some of the most demanding ones.

As mentioned above, it's quite useless to configure and reserve a whole desktop instance if you only use an application.

You can seamlessly save your outputs inside an Amazon S3 bucket, where, if configured correctly, you can use a VPC endpoint and move the heavy CAD files only inside the AWS region, reducing costs even more!

## Wrap Up

Overall Amazon AppStream is a solid option that can be considered when approaching the need to distribute applications easily, scalably, securely, and affordably, all while respecting an End-user environment.

I found its use relatively simple; it doesn't require much effort in setting up the environment and the application, and managing the images and fleets is streamlined and well-documented.

There are definitely some alternatives that are similar to Amazon AppStream, like Citrix, VMWare, and Microsoft. Still, they all have their preferred use case, so I recommend carefully evaluating each provider's pros and cons!

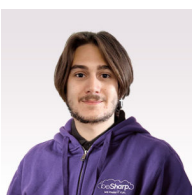
If you are interested in deepening this aspect, feel free to leave a comment below! We'll be happy to discuss this further in our following articles.

See you again in 14 days on Proud2beCloud!

---

## About Proud2beCloud

**Proud2beCloud** is a blog by [beSharp](#), an Italian APN Premier Consulting Partner expert in designing, implementing, and managing complex Cloud infrastructures and advanced services on AWS. Before being writers, we are Cloud Experts working daily with AWS services since 2007. We are hungry readers, innovative builders, and gem-seekers. On Proud2beCloud, we regularly share our best AWS pro tips, configuration insights, in-depth news, tips&tricks, how-tos, and many other resources. Join the discussion!



**Stefano Salvaneschi**

DevOps Engineer @ beSharp. I'm the kind of guy that can't let a question go unanswered and I can't resist making them too! Passionate about IT since i was a child, now landed on the Cloud side. In my free time i enjoy playing videogames, dungeons & dragons and everything geeky in between, all accompanied by a nice beer!

---

Copyright © 2011-2023 by beSharp spa - P.IVA IT02415160189