Audio file

Your Recording 37.wav

Transcript

00:00:13 Speaker 2

So from that URL we gain access to the Sagemaker Studio and straight away we can see that we are in dark mode and that we have a more complete interface than just Jupyter lab. I can see in the left hand side I have my navigation bar and most importantly I can see I have more than one application. I can see I have Jupyterlab and if I click on that, that will get me through to a hosted Jupyterlab interface.

00:00:42 Speaker 2

Wonderful.

00:00:42 Speaker 2

But it's not the only app.

00:00:45 Speaker 2

I also have access to Rstudio. Remember, statisticians love the R programming language and the Rstudio. So if I clicked on that link, I would gain access to the Rstudio, but again, hosted inside of a managed instance by Sagemaker. I would need an Rstudio license to do that. There is that small caveat first.

00:01:09 Speaker 2

We also have Code Editor now that is essentially Microsoft Visual Studio Code, again hosted by Sagemaker Studio. So in all cases, these applications are hosted and run inside of managed instances within your AWS region. Code Editor provides a more developer-like experience, so maybe as you're moving out of the experimentation phase of your ML project.

00:01:36 Speaker 2

And more into the productionization of your project so that it needs to become more.

00:01:42 Speaker 2

Automated rather than human driven. You probably will refactor some of your Jupyter notebooks into Python scripts. At that point you kind of feel more like a Python developer and you should probably be working in a more suitable integrated development environment for plain Python coding, and code editor fits that job perfectly.

00:02:05 Speaker 2

We also have access to Canvas here. Sagemaker Canvas we're going to have a look at later on, but for now all we need to know is that it is a low code tool that allows me to visually build a model without having any Python skills. Really rather clever, but it's it. I don't ever fluffed it. So from that URL we are now in Sagemaker Studio.

00:02:31 Speaker 2

And straight away we can see that we are in a completely different user interface. We are in dark mode by default.

00:02:39 Speaker 2

Now what I can see in the top left side of the user interface is the applications panel. Now the applications panel tells me straight away that I have more than just Jupyter Lab as an available application.

00:02:55 Speaker 2

Now let's look at each application in turn. We have Jupyter Lab so we can gain access to a separate Jupyterlab interface that will be running as a hosted instance in Sagemaker, and again we can choose the size of that instance.

00:03:13 Speaker 2

Alternatively, if we are using the R programming language and want to use Rstudio, we could use that instead. Now this is loved historically by many data scientists and statisticians, but one caveat if you want to use Rstudio is that you will need an Rstudio license you need that needs to be supplied into the interface first before that will work.

00:03:38 Speaker 2

But again, it's just running our studio in a managed virtual machine for you, hosted in your AWS region.

00:03:47 Speaker 2

We also have Code Editor, which is essentially Microsoft Visual Studio Code, a more developer-focused IDE. Now this can be very good for us as we start to productionize and we move away from the Interactive Data exploratory phase that we perform in Jupyter and move to more automated pipelines. We might refactor some of our Jupyter notebooks into Python scripts and that point having a more.

00:04:17 Speaker 2

Develop a focused IDE would be good.

00:04:20 Speaker 2

They may have Sagemaker Canvas, a low code tool that allows me to build models without knowing any Python coding at all. Now obviously that has some limitations, but it's pretty powerful for rapid prototyping and we have a dedicated lesson on that later on.

00:04:38 Speaker 2

And we have ML Flow.

00:04:41 Speaker 2

Now ML Flow is a complete end to end machine learning management framework.

00:04:47 Speaker 2

And in some ways there is some overlapping functionality between ML Flow and Sagemaker Studio.

00:04:54 Speaker 2

But Siegemaker make it available as an application hosted within Sagemaker, allowing you to take advantage of its capabilities, particularly around experimentation, management and model registry capability.

00:05:12 Speaker 2

Now over on the right here, we've got a link to say, access your Studio Classic applications. So if you had used Sagemaker maybe more than one or two years ago, you would have been using Sagemaker Classic. Now again, Sagemaker Classic is being deprecated, but if you need access to that legacy interface, you can gain access to it from there.

00:05:36 Speaker 2

Now, as we expand the left navigation bar, we can see some of the things it exposes. For example, we can look at running instances. Now. Right now I don't have any running instances. I've just launched Sagemaker Studio, and I've not yet started up Jupyter Lab or code Editor or Canvas. But once you start to run those applications, each one will show up as a separate running instance that could be stopped and started.

00:06:06 Speaker 2

As you please. We've also got access in here to Hyper Pod. And Hyper pod, well we talk about it later on, is going to allow us to have extremely large resilient clusters ideal for training large language models.

00:06:22 Speaker 2

We can see under the Data section we've got access to certain data tools, such as the Data Wrangler, a tool that allows me to perform data preparation and feature engineering in a graphical, low-code way. It's actually now integrated as part of Sagemaker Canvas. I've also got access to the Feature Store, a way that I can store engineered features from my data so that I can reuse them for other projects.

00:06:51 Speaker 2

Or to accelerate inference, I've also got access to EMR clusters. EMR is Elastic MapReduce, essentially managed Hadoop clusters, and we've had Emrs separate AWS service for many years. But it's a great environment for running distributed processing if you've got large datasets and you want to run Pyspark scripts against them, so in a distributed way, then EMR clusters.

00:07:21 Speaker 2

Or a perfect place to run them and you can gain access to them directly from here in Sagemaker Studio.

00:07:28 Speaker 2

Slide down to the bottom where we can see deployments. This is where we can view and manage any endpoints. Now, an endpoint is where we are choosing to use Sagemaker to host our model. In other words, it deploys A managed virtual machine with our model artifact deployed inside it, along with some inference handling code. And right at the bottom, we've got projects, and that's going to allow us to have complete turnkey projects that will deploy.

00:07:58 Speaker 2

Everything needed for a complete CI CD, automated training pipeline and deployment pipeline. Something again, we have a complete lesson on later in the course. So let's say I want to use Jupyter Lab in the Sagemaker Studio user interface. I go to the applications panel and I find the application I want. So there's Jupyterlab and I would click on it.

00:08:22 Speaker 2

At that point, I'm presented with what Jupiter Lab spaces I have. But what is a space? A space is essentially A managed virtual machine instance that is running our Jupiter Lab software. You can think of it almost as an equivalent of a notebook instance that we saw in our previous demonstration.

00:08:44 Speaker 2

Now right now I can see I have no Jupyter lab spaces, so I'm going to have to create one.

00:08:51 Speaker 2

So I'd click on Create Jupiter Lab space.

00:08:54 Speaker 2

When I do that, it's going to prompt me for a name for that space. So I'm going to call this one Jupiter Lab Space 1, and it's going to prompt me for the sharing mode. Do I want this to be private just to me, or do I want to share it with other user profiles in the same Sagemaker domain?

00:09:12 Speaker 2

For now I'm going to select private.

00:09:15 Speaker 2

And then I can click createspace.

00:09:18 Speaker 2

Now we saw that choice of private space versus shared space, So what's the real difference?

00:09:25 Speaker 2

Well, we saw that if I choose a private space is just private to me, but if it's shared then multiple users can collaborate and share that space. Good start.

00:09:37 Speaker 2

So we can think of that solo work versus real time Co editing.

00:09:43 Speaker 2

And this is an important point because in a shared space with Jupiter Lab, we can have a kind of Google Docs shared document experience. So it can both be working on the same Jupyter notebook at the same time, and that can make a huge difference for productivity.

00:10:03 Speaker 2

Now, if I use a private space for my Jupyter lab, I've got a dedicated virtual machine resource that only I can use. But if I choose shared space, then that instance could be used by multiple users at the same time.

00:10:19 Speaker 2

Now here is where a lot of confusion arises, and that's in storage. Private space uses an EBS volume per user. OK, because you get a dedicated managed virtual machine just to yourself. Then the Manage Virtual Machines virtual hard disk is all yours. So the virtual hard disk is called an EBS volume in AWS terminology.

00:10:46 Speaker 2

In a shared space, you've got multiple users all accessing the same.

00:10:50 Speaker 2

Virtual machine instance and therefore you are sharing the same virtual machine, virtual hard disk or EBS volume. Now at this point you might think, well, what about the EFS volume in the domain that run NFS? Yeah, we don't use that anymore, not for shared spaces using Jupyter Lab. This is a really important point because a lot of people think they're still using EFS and they're not.

00:11:18 Speaker 2

We transitioned away from that when Sagemaker Classic.

00:11:21 Speaker 2

Was end of supported?

00:11:24 Speaker 2

So when we think about the use cases of private versus shared, if I'm working individually, then sensitive work just for me, private space makes sense. But if I'm working with other data scientists, I'm looking to collaborate, then I need a shared space.

00:11:42 Speaker 2

If I want to do customization, I can because I'm more isolated in a private space, it makes sense I can fully customize that to my own personal needs. But if I am in a shared space, remember that some settings that I change will affect not just me.

00:11:59 Speaker 2

From a security perspective, very strict access control to a private space. It's just for your user profile. If I go with a shared space then you're going to need permissions management to be really clear about who can perform what action.

00:12:14 Speaker 2

So just a note here about EFS in Sagemaker domains. As of 2025, we are no longer using Sagemaker Classic and therefore EFS is no longer being used by default. We are just using EBS, the virtual hard disks of the virtual machines that you provision for your spaces. Be careful when you look in the documentation when it's talking about shared spaces in EFS because of the documentation is written as of.

00:12:44 Speaker 2

2024 or earlier, it might suggest that you're using EFS when you're not.

00:12:50 Speaker 2

Now I mentioned Sagemaker Classic again there, so it's important to differentiate Sagemaker Classic features from Sagemaker Studio features.

00:13:01 Speaker 2

So with Sagemaker Classic, if you have just started you working with Sagemaker, you are only using Sagemaker Studio and you can safely ignore references to Sagemaker Classic. Sagemaker Classic is still maintained in order to support customers who started out with that a number of years ago, but AWS has been strongly encouraging customers to move away from Sagemaker Classic.

00:13:27 Speaker 2

Now, in terms of default creation behavior, Sagemaker Classic automatically created a private space per user profile. Sagemaker Studio, we don't do that. You need to create a Jupyter lab space, or a code editor space, or an Rstudio lab space. You've got to create the spaces that you need. We don't make assumptions.

00:13:51 Speaker 2

From a storage perspective, in Sagemaker Classic we used the EFS volume, the shared NFS volume that was defined as part of your Sagemaker domain. But in Sagemaker Studio now we only use EBS volumes. In other words, the virtual hard disks of the virtual machine that is hosting your Jupyter Lab or your Rstudio or your code editor. This was primarily for performance reasons.

00:14:20 Speaker 2

We were getting far superior performance from EBS than from EFS.

00:14:26 Speaker 2

From a collaboration perspective, both Sagemaker Studio and Classic both support public and private spaces. But on Sagemaker Classic, it was all on the EFS file system. On Sagemaker Studio, it's not, it's just EBS.

00:14:43 Speaker 2

From an execution role perspective, the IAM role that is used to perform your actions that will assume the User Profile execution role in Sagemaker Classic and it will do the same in Sagemaker Studio.

00:14:59 Speaker 2

Further customization options available in Sagemaker Studio in terms of how it operates.

00:15:05 Speaker 2

And there are greater range of applications available in Sagemaker Studio compared to Sagemaker Classic. O. Again, avoid Sagemaker Classic if you can and justice use the newer Sagemaker Studio. Let's take a look at the storage alignment between spaces and user profiles, just to be clear about how they interact.

00:15:28 Speaker 2

In the Sagemaker domain, we have our Sagemaker Studio. We have our user profiles defined.

00:15:35 Speaker 2

And then we can create spaces. So let's say we've created a Jupyter Lab space that's marked as private for user one. We've then created a Jupyter Lab space separately that is marked as being shared. So user 1 profile would see listed there is a private space and they would see a public space. The second user, user profile two would only see the public shared space.

00:16:02 Speaker 2

They would not see the private space at all. They had no idea.

00:16:06 Speaker 2

There was a private space in existence, so it's completely hidden from the other user. They'll only see the shared space.

00:16:14 Speaker 2

Now when you create a space, you define the size of managed EC2 instance that will be used to host the Jupyter Lab software. Here we've chosen for private space MLT 3 medium. OK, well then the storage will be in the EBS volume or virtual hard disk attached to that instance.

00:16:35 Speaker 2

Now if I choose for my shared space, again I've gone for the same instant size, a MLT 3 medium, then that instance will also have local hard disk, in other words an EBS volume, and that's where the data will be written to. Which raises a question, well, what is the EFS volume doing in the Sagemaker domain when we know that EFS volume was primarily there for Sagemaker Classic, the previous version?

00:17:04 Speaker 2

Of Sagemaker Studio. This starts to make sense when no longer using it. It wasn't fast enough. If you want it to consume it, we won't stop you.

00:17:15 Speaker 2

You would have to create a manual mount point from within your space to do an NFS mount to that target EFS system. So we won't stop you from mounting it. So there's an option, but there is that caveat on performance. If we compare that with Sagemaker Classic, we would have our Sagemaker domain, We would have your Sagemaker Classic application, and you would have your user profiles, but this time your user profiles would align to.

00:17:45 Speaker 2

Storage that was on the EFS volume. Each user profile would get their own private directory. An application and your use of storage would align directly with your own private directory. If you chose to have a shared space, then there would be a shared directory space where you could both sharefile assets. So there's user profile one only seeing their directory, user profile 2 only seeing theirs and both users.

00:18:16 Speaker 2

Seeing the shared space, let's now look at how we can stop and start our Jupyter Lab spaces and how we can control what software is available within them.

00:18:29 Speaker 2

When I look at a Jupyter lab space I've created, I created one here that was private and was of size MLT 3 medium. Now I can see right now that this is a sense because it's of type ML, it indicates that it's an ML fully managed instance. So I can't get into that instance directly because it's being looked after by Sagemaker. I can see it's currently.

00:18:57 Speaker 2

Its status is stopped.

00:18:58 Speaker 2

So I can't connect to it until I start it. Now, when we created our Jupyter lab space, not only did we specify the compute size in terms of MLT 3 dot medium, but we also had to specify a container image. Now, when Sagemaker performs any action with managed instances, it's not actually deploying software into the operating system.

00:19:27 Speaker 2

Is not installing things into Linux. Instead, what it's doing with those managed instances is it's using containerization, so it's actually running containers on those instances to provide that software. Now in the majority of cases, that will be one container per EC2 managed instance, so that Jupyter Lab software has been obtained from a Jupyter Lab container image.

00:19:56 Speaker 2

We obtained that image from a repository of container images within your AWS account. So within your AWS account we have a service called the Elastic Container Registry. And the Elastic Container Registry allows you to store container images. Now if you've ever worked with Docker or similar container run times, then containers won't make sense to you. But if you've not worked with containerization before, the basic premise.

00:20:26 Speaker 2

Is here that you will be able to run software in a managed environment where we can control and abstract the underlying file system, CPU, networking into separate namespaces. Now I can see here the Sagemaker Distribution 2.1 container image resides in the container registry within my account. I don't need to do anything, Sagemaker has made sure that it's already there.

00:20:55 Speaker 2

When you spin up a Jupiter Lab space, what is happening is that Sagemaker is provisioning a managed EC2 instance with a container runtime in it.

00:21:06 Speaker 2

And it will pull that container image in to that instance and launch the container. And it's the container that is running your Jupyter lab server process.

00:21:19 Speaker 2

So when we look at the full properties of our Jupyter Lab space, I can say, yes, I've specified my instance size and yes, I've specified my image. Now, by default here, we're looking at a Sagemaker distribution 2.1 image. Over time, that will be updated. There'll be a 2.22 point 4 and eventually a distribution 3. So it's up to you to make sure that you are picking an up-to-date image that contains all of the.

00:21:49 Speaker 2

Components that you would expect to have now whether you're using.

00:21:54 Speaker 2

Tensorflow or Pytorch, Keras, Numpy, Pandas or psychic learn? Then the default container image will contain most of the tools you need. But for some of the deep learning frameworks, there are some dedicated container images just for you that go beyond what we give in our general purpose container image.

00:22:17 Speaker 2

So make sure that you familiarize yourself with walk container images Sagemaker makes available to you and pick the appropriate one that's going to deliver you the software that you need.

00:22:31 Speaker 2

So the Sagemaker Python SDK that allows us to have high level abstractions for training jobs, for data processing jobs for inference and deployment of endpoints that is accessible and already present in that container image. So in your Python scripts within your Jupyter lab space, you could simply say import Sagemaker or import photo 3 and it will just work.

00:23:01 Speaker 2

You don't need to do any pip installs in this place, it's already there.

00:23:07 Speaker 2

Once your Jupiter Lab space is up and running, we'll be able to see that under the status indicator. We'll then see that we've got a button to open Jupyter Lab, and if we click on that, it will spawn our new Jupyterlab interface, just like we saw before in our familiar interface for managing Jupyter Notebooks. But notice when it's running, we also now have a button to stop the space. And if we were to stop the space, then we'll get a warning.

00:23:36 Speaker 2

And the warning is just telling us that yes, this will stop the space. So my MLT 3 medium.

00:23:43 Speaker 2

Virtual machine instance will shut down and they'll no longer be paying for it. But it's just warning you here that if you did anything from within your notebooks, such as provision A Sagemaker endpoint hosting your model, while you may be provisioned an EMR cluster to run some Pyspark job or maybe you created an S3 bucket to host your data set, those resources will still be there and you'll still be billed for those. So it's just giving you an awareness of, yes, you're saving money by.

00:24:13 Speaker 2

This, but be aware anything you've already provisioned, unless you deprovision it, that will continue to be billed against your account.

00:24:23 Speaker 2

If we look at our Jupyter lab space, this should look very familiar to us. Now we can see we are in our launcher tab and we can create new Jupyter notebooks using a variety of different kernels. We've just been using the Python 3 kernel so far in this course. We can also see in the left hand side in our navigation bar we've got our extension for git so that we can clone repositories and push back our changes back to git servers.

00:24:51 Speaker 2

Now, if we choose to use a Jupyter Lab space that was shared rather than private, the first indication that you would be in a shared space would be this indicator on the launcher page RTC indicating real-time collaboration.

00:25:10 Speaker 2

There would be a further clue that you are in a shared Jupiter Lab space in the top right hand corner of the interface. What we would see is this kind of indicator giving an indication of the user's name, just with a single letter and a color. Now again, the idea of is that we can easily identify if there are multiple collaborators working on the same document at the same time. So again, if you've ever worked in Google Docs.

00:25:40 Speaker 2

Same document at the same time. Yeah, like different coloured cursors, so we know who's making what changes. It's a very similar concept to what we have here. So if you're never not sure, check the top right hand corner. If there's no coloured circle there or RTC indicator on the launcher page, then you're probably just in a private space instead. Now, in order to create a shared space during the creation of a Jupyter Lab space, we would just need to indicate.

00:26:10 Speaker 2

That you are sharing this with your domain.

00:26:14 Speaker 2

When you have multiple Jupyter lab spaces, there are two places where we can see and manage that.

00:26:22 Speaker 2

1st is in the Sagemaker AI Web Management Console. In other words, outside of the Sagemaker Studio UI in the original web console UI.

00:26:35 Speaker 2

When you look at your Sagemaker domain, we saw previously the Domain Settings tab and the User Profiles tab.

00:26:43 Speaker 2

But here we're looking at the Space Management tab.

00:26:48 Speaker 2

And because I've created 2 spaces, one that was private and one that was shared, I could see all those spaces listed here and notice in the owner user profile name we can see who created them.

00:27:03 Speaker 2

When you have multiple Jupyter lab spaces.

00:27:07 Speaker 2

In your Sagemaker domain.

00:27:09 Speaker 2

There are two places where we can see and manage that.

00:27:13 Speaker 2

Firstly, if we look in AWS Management Console outside of the Sagemaker Studio, we can see the properties of our Sagemaker domain. We have already seen domain settings for the readiness of our domain and its name. We have seen the User Profiles tab where our default user was defined and where we could add new ones.

00:27:36 Speaker 2

But now we're looking at space management. Under space management we can see how many Jupyter lab spaces we have. Now here I can see I've got a Jupyter lab space that's in service that is shared.

00:27:50 Speaker 2

And I have a second one, which is the Jupiter Lab space, which is private. So if you imagine you are managing a team of data scientists, you might have 10 private spaces, one each, and maybe one or two shared depending on what collaboration projects they were working on. The 2nd place where I could see my spaces would be within the Sagemaker Studio. Within Sagemaker Studio, if I've clicked on Jupyter Lab, I'd be able to see what spaces I have.

00:28:20 Speaker 2

So here I can see I have two spaces. One of them is running. I can see in the action column that I could stop it because it is running. I can see I could open it. That would launch my new tab into Jupyter lab. I can also see there under the type column that space is shared. I can see I've got another Jupyter lab space defined. This one is currently stopped.

00:28:46 Speaker 2

It's private just to me, so only I would see it under my user profile.

00:28:51 Speaker 2

And over in the action column, because space is currently stopped, the only action available to me is to start it by clicking Run.

00:29:01 Speaker 2

So if we click into the shared space there, we'll be able to see the properties of that shared space. And because this shared space is currently running, the link is there to open Jupyterlab and gain access to the regular Jupyter interface. Notice here in the properties we can see the actual properties of the EC2 instance. In this case here we've got a 5 GB virtual hard disk. We could change that to be larger if we were working with a larger.

00:29:31 Speaker 2

Set and we needed more local storage, but we would do that when the instance was stopped.

00:29:37 Speaker 2

Remember, with public and private spaces, visibility is limited. If I've got a private space, then that private space is only visible to my user profile. No other user profile will be able to see my private space. They will only see their private space and any shared spaces.

00:30:00 Speaker 2

If we were to compare private spaces with shared spaces.

00:30:04 Speaker 2

Remember that private spaces you have your own view too, only you have access to them. And even administrators of the Sagemaker domain do not have direct access.

00:30:15 Speaker 2

That a shared space will be visible to all users in that domain. If you don't want a user to have visibility of a domain then create another domain for them. Remember we are not limited to a single Sagemaker domain.

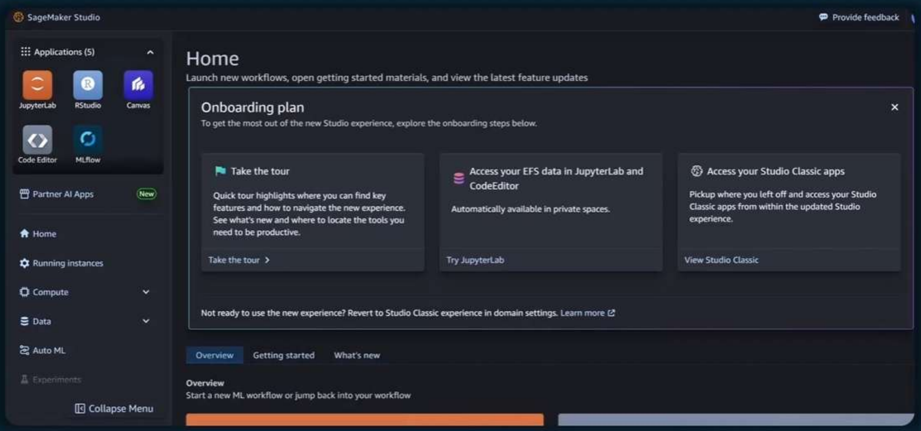
00:30:31 Speaker 2

We do need the proper permissions to be accessible, so while a shared space might be visible, it might not be consumable depending on the permissions you have.

00:30:42 Speaker 2

It will be listed though for all users. So to put it another way, if I have user A and user B, then any private space either user creates would only be visible to themselves, whereas a shared space would be visible to all users in the domain, including user A&B.

Workflow: SageMaker Studio



Home

Launch new workflows, open getting started materials, and view the latest feature updates

Onboarding plan

To get the most out of the new Studio experience, explore the onboarding steps below.

Take the tour

Quick tour highlights where you can find key features and how to navigate the new experience. See what's new and where to locate the tools you need to be productive.

Take the tour >

Access your EFS data in JupyterLab and CodeEditor

Automatically available in private spaces.

Try JupyterLab

Access your Studio Classic apps

Pickup where you left off and access your Studio Classic apps from within the updated Studio experience.

View Studio Classic

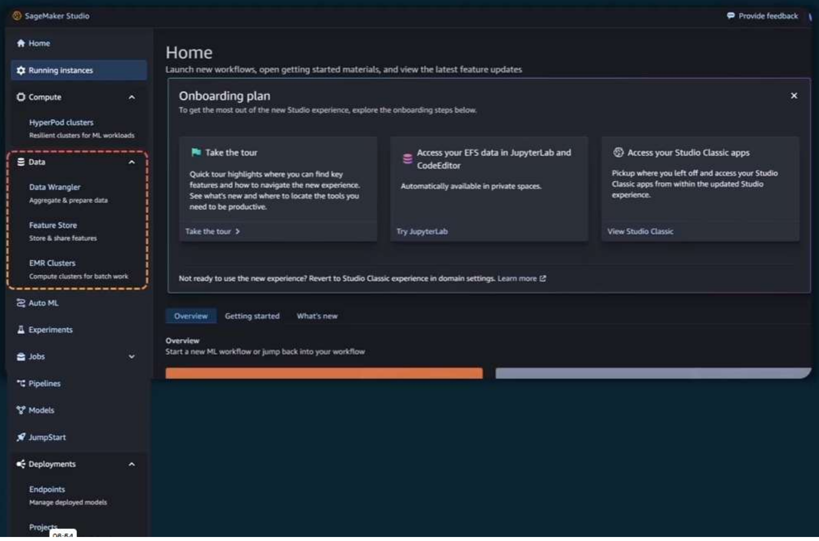
Not ready to use the new experience? Revert to Studio Classic experience in domain settings. Learn more

Overview | Getting started | What's new

Overview

Start a new ML workflow or jump back into your workflow

Workflow: SageMaker Studio



Home

Launch new workflows, open getting started materials, and view the latest feature updates

Onboarding plan

To get the most out of the new Studio experience, explore the onboarding steps below.

Take the tour

Quick tour highlights where you can find key features and how to navigate the new experience. See what's new and where to locate the tools you need to be productive.

Take the tour >

Access your EFS data in JupyterLab and CodeEditor

Automatically available in private spaces.

Try JupyterLab

Access your Studio Classic apps

Pickup where you left off and access your Studio Classic apps from within the updated Studio experience.

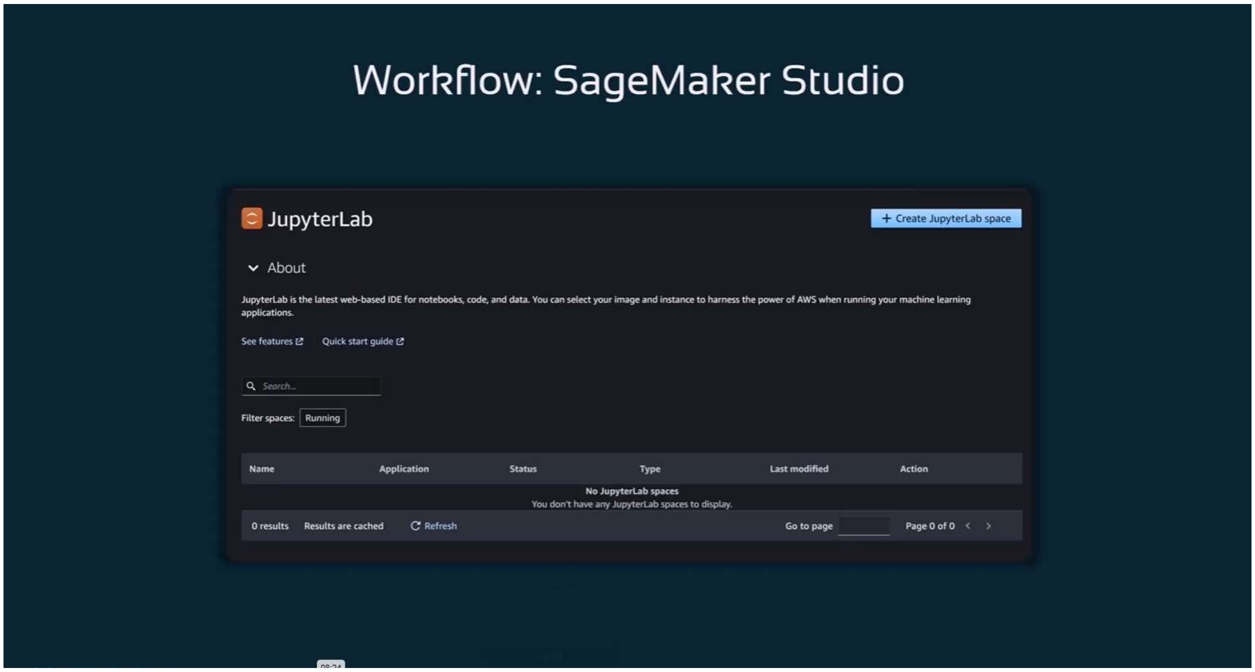
View Studio Classic

Not ready to use the new experience? Revert to Studio Classic experience in domain settings. Learn more

Overview | Getting started | What's new

Overview

Start a new ML workflow or jump back into your workflow



Workflow: SageMaker Studio

JupyterLab

About

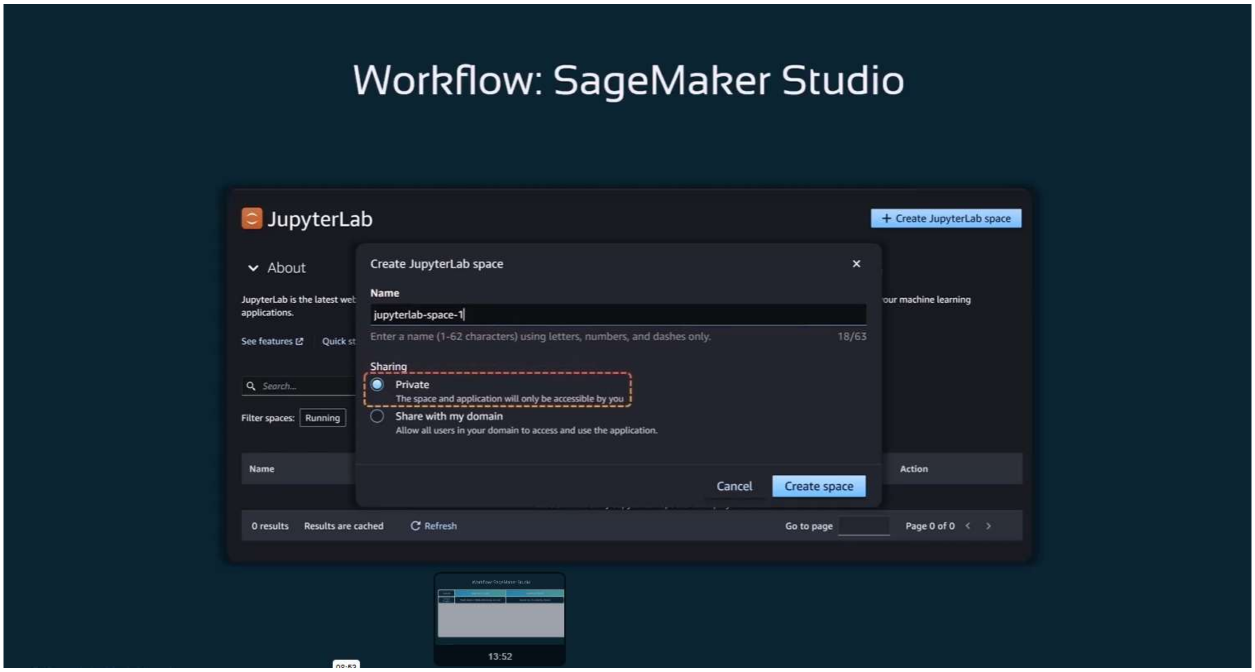
JupyterLab is the latest web-based IDE for notebooks, code, and data. You can select your image and instance to harness the power of AWS when running your machine learning applications.

Filter spaces: Running

Name	Application	Status	Type	Last modified	Action

No JupyterLab spaces
You don't have any JupyterLab spaces to display.

0 results Results are cached Refresh Go to page Page 0 of 0



Workflow: SageMaker Studio

JupyterLab

Create JupyterLab space

Name

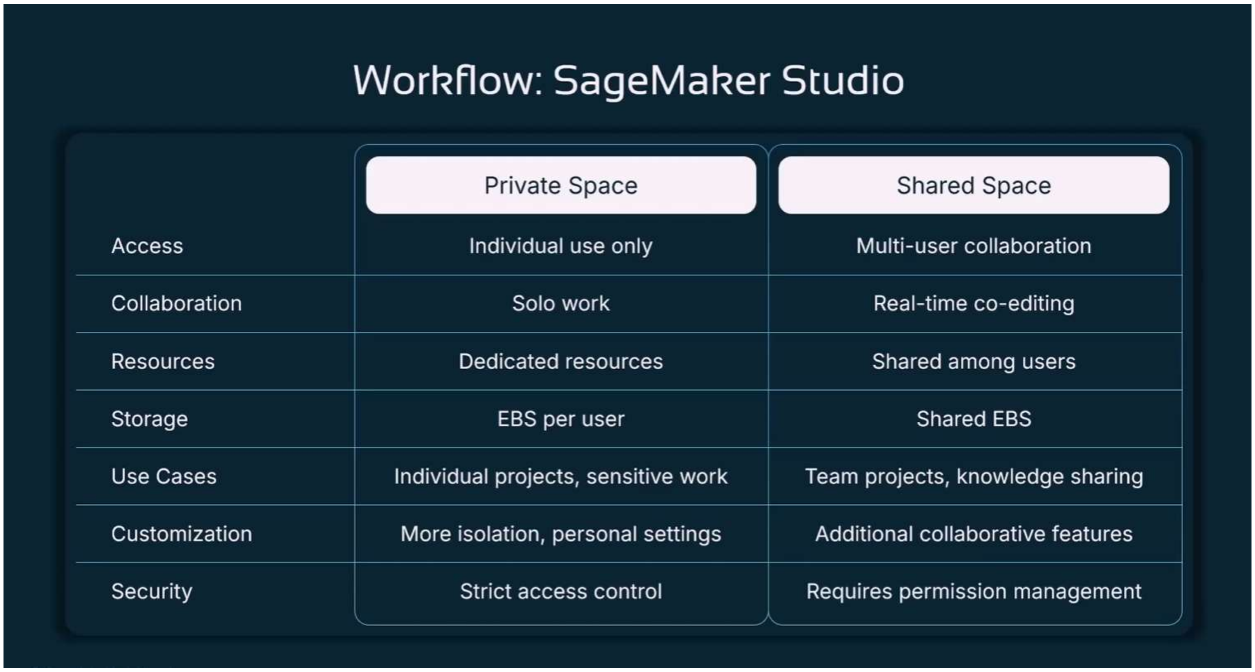
jupyterlab-space-1

Enter a name (1-62 characters) using letters, numbers, and dashes only. 18/63

Sharing

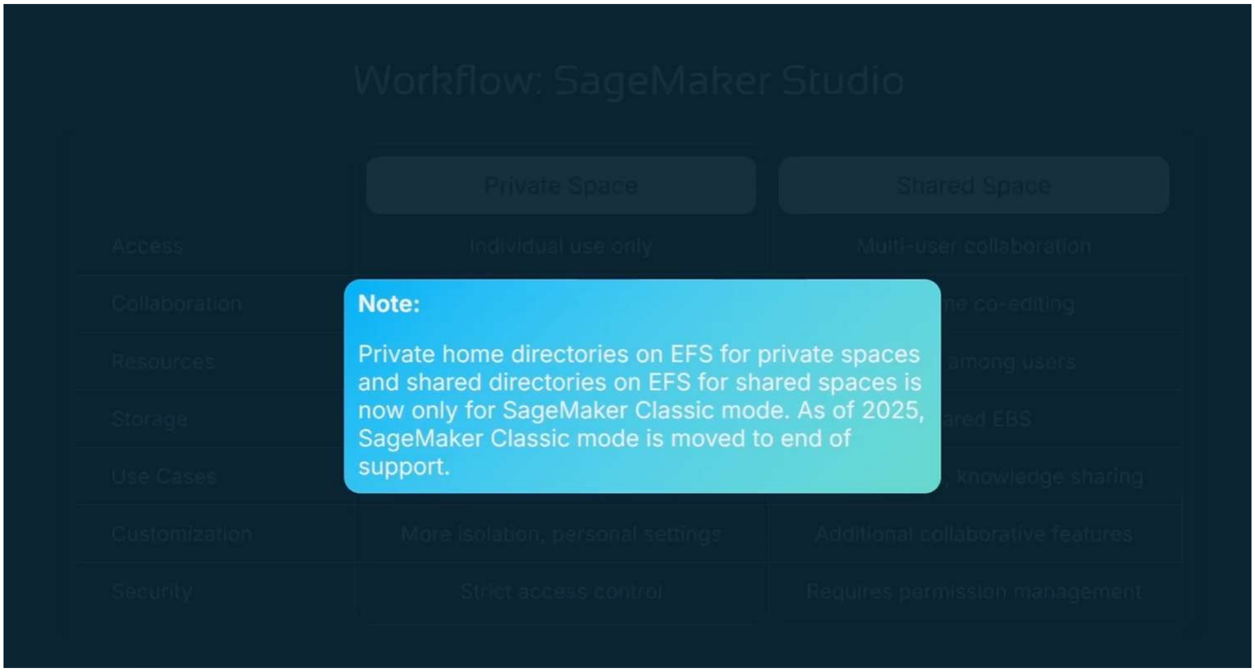
● Private
The space and application will only be accessible by you

○ Share with my domain
Allow all users in your domain to access and use the application.



Workflow: SageMaker Studio

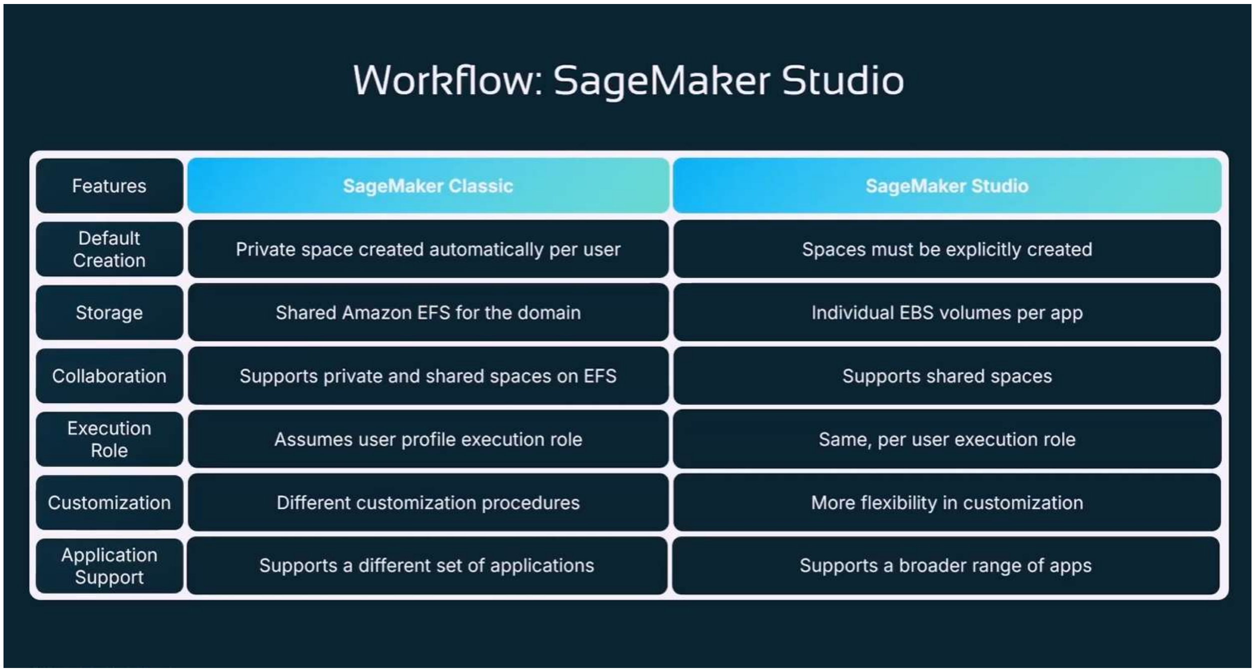
	Private Space	Shared Space
Access	Individual use only	Multi-user collaboration
Collaboration	Solo work	Real-time co-editing
Resources	Dedicated resources	Shared among users
Storage	EBS per user	Shared EBS
Use Cases	Individual projects, sensitive work	Team projects, knowledge sharing
Customization	More isolation, personal settings	Additional collaborative features
Security	Strict access control	Requires permission management



Workflow: SageMaker Studio

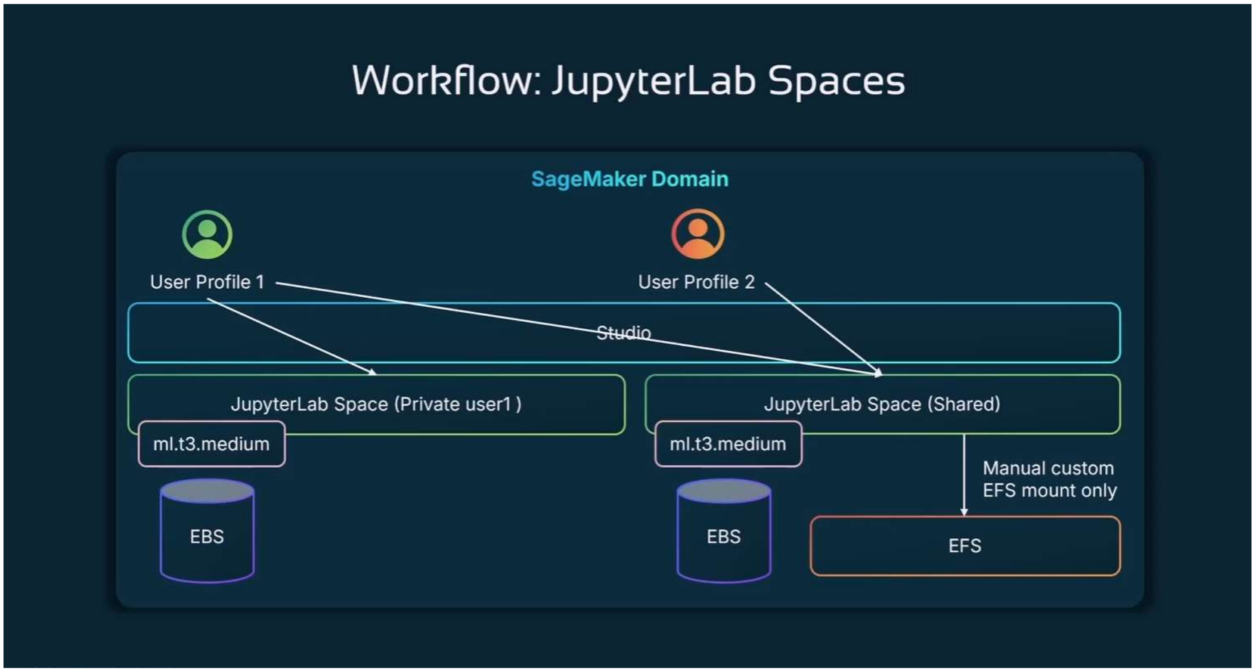
Note:

Private home directories on EFS for private spaces and shared directories on EFS for shared spaces is now only for SageMaker Classic mode. As of 2025, SageMaker Classic mode is moved to end of support.



Workflow: SageMaker Studio

Features	SageMaker Classic	SageMaker Studio
Default Creation	Private space created automatically per user	Spaces must be explicitly created
Storage	Shared Amazon EFS for the domain	Individual EBS volumes per app
Collaboration	Supports private and shared spaces on EFS	Supports shared spaces
Execution Role	Assumes user profile execution role	Same, per user execution role
Customization	Different customization procedures	More flexibility in customization
Application Support	Supports a different set of applications	Supports a broader range of apps



Workflow: JupyterLab Spaces

SageMaker Domain

User Profile 1

User Profile 2

Studio

JupyterLab Space (Private user1)

JupyterLab Space (Shared)

ml.t3.medium

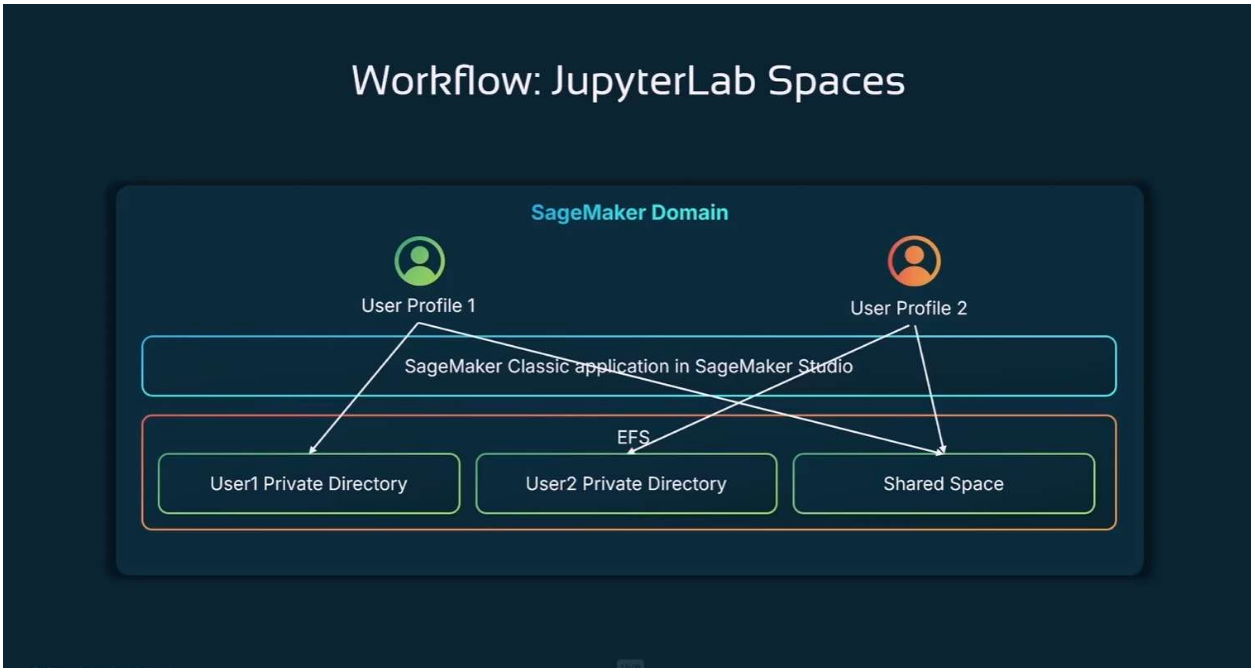
ml.t3.medium

EBS

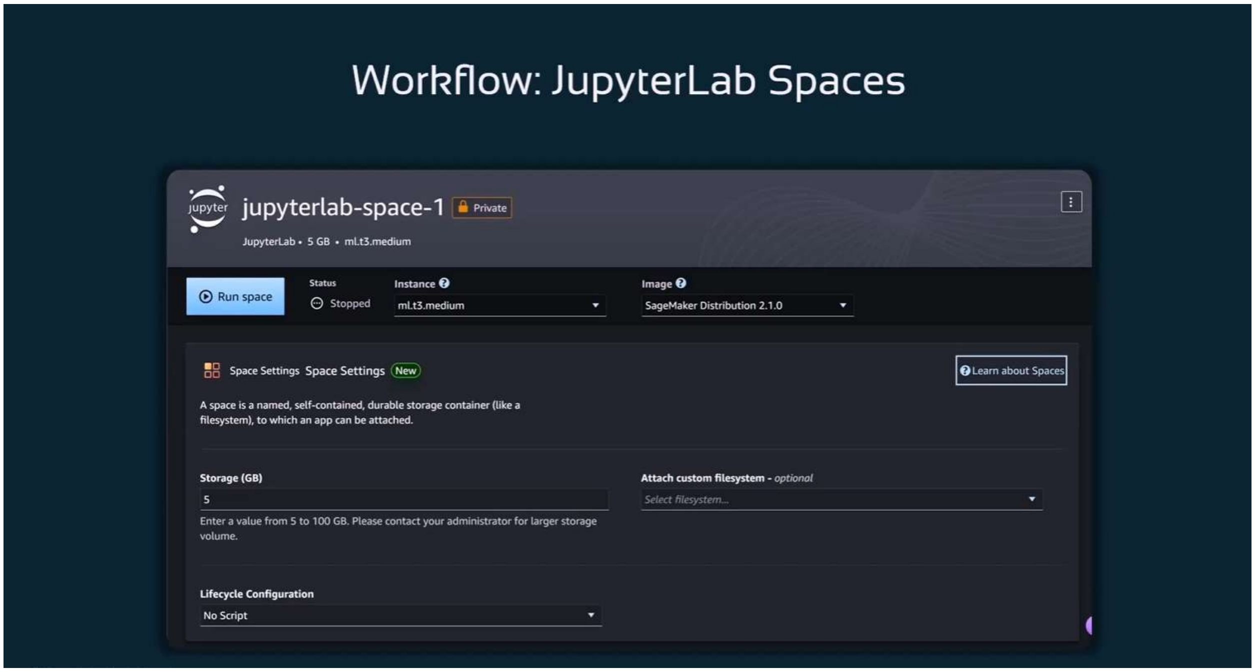
EBS

Manual custom EFS mount only

EFS



Workflow: JupyterLab Spaces



Workflow: JupyterLab Spaces

jupyterlab-space-1 🔒 Private

JupyterLab • 5 GB • ml.t3.medium

Status: Stopped

Instance: ml.t3.medium

Image: SageMaker Distribution 2.1.0

Space Settings Space Settings New

A space is a named, self-contained, durable storage container (like a filesystem), to which an app can be attached.

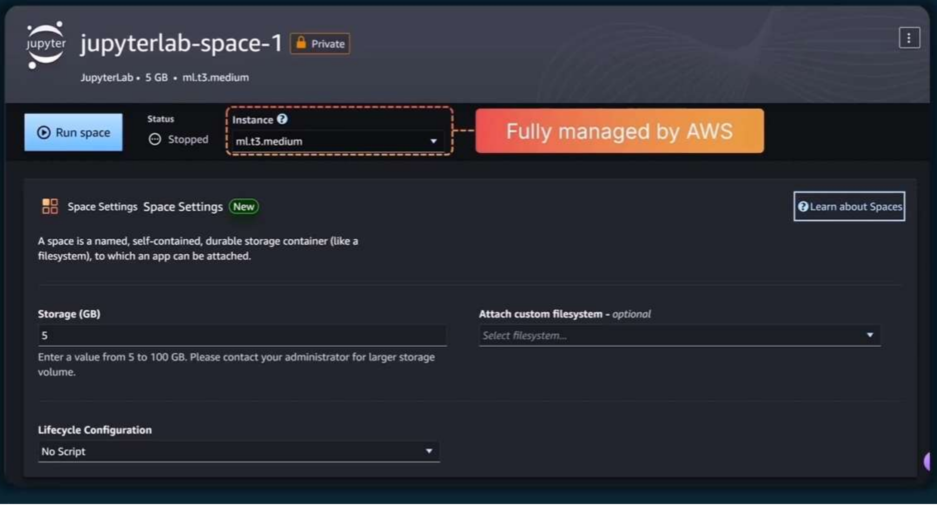
Storage (GB): 5

Enter a value from 5 to 100 GB. Please contact your administrator for larger storage volume.

Attach custom filesystem - *optional*: *Select filesystem...*

Lifecycle Configuration: No Script

Workflow: JupyterLab Spaces



jupyterlab-space-1 Private

JupyterLab • 5 GB • ml.t3.medium

Status: Stopped

Instance: ml.t3.medium

Fully managed by AWS

Space Settings Space Settings New

A space is a named, self-contained, durable storage container (like a filesystem), to which an app can be attached.

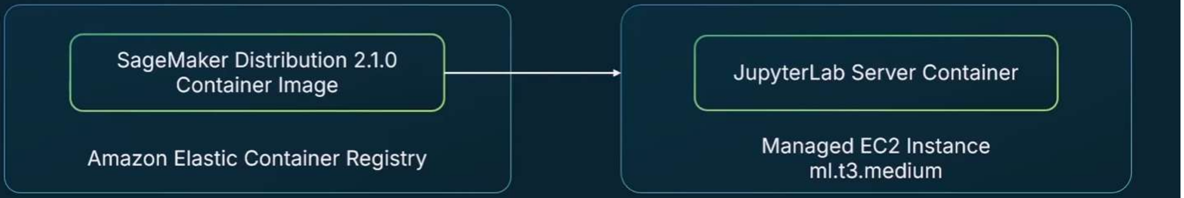
Storage (GB): 5

Enter a value from 5 to 100 GB. Please contact your administrator for larger storage volume.

Attach custom filesystem - *optional*: *Select filesystem...*

Lifecycle Configuration: No Script

Workflow: JupyterLab Spaces



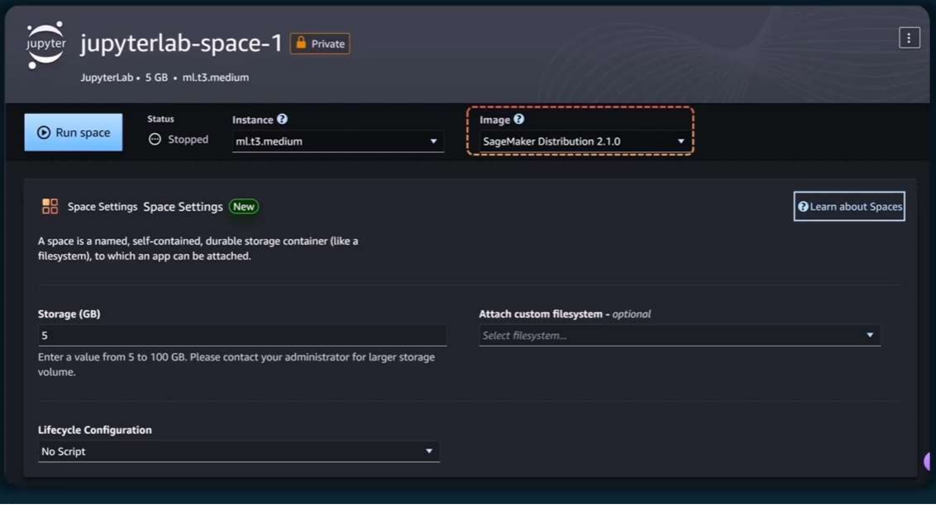
SageMaker Distribution 2.1.0
Container Image

Amazon Elastic Container Registry

JupyterLab Server Container

Managed EC2 Instance
ml.t3.medium

Workflow: JupyterLab Spaces



jupyterlab-space-1 Private

JupyterLab • 5 GB • ml.t3.medium

Status: Stopped

Instance: ml.t3.medium

Image: SageMaker Distribution 2.1.0

Space Settings Space Settings New

A space is a named, self-contained, durable storage container (like a filesystem), to which an app can be attached.

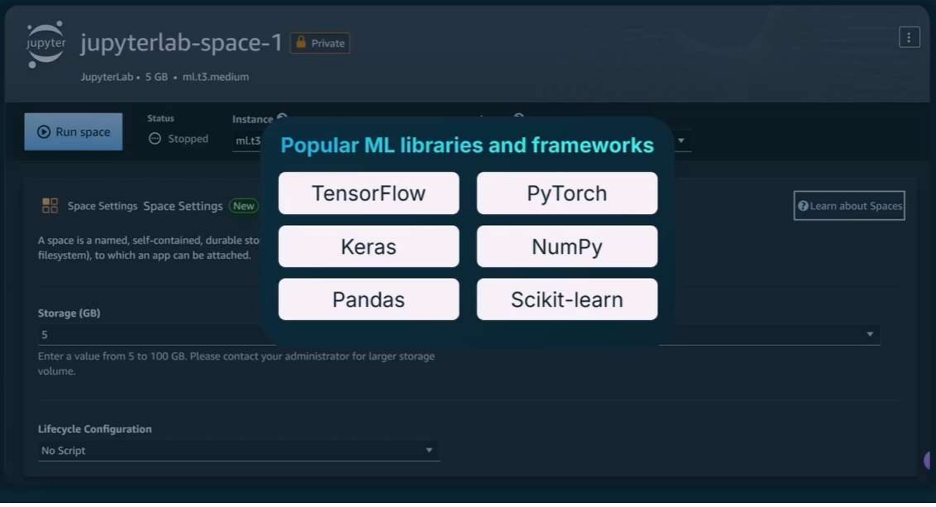
Storage (GB): 5

Enter a value from 5 to 100 GB. Please contact your administrator for larger storage volume.

Attach custom filesystem - *optional*: *Select filesystem...*

Lifecycle Configuration: No Script

Workflow: JupyterLab Spaces



Popular ML libraries and frameworks

- TensorFlow
- PyTorch
- Keras
- NumPy
- Pandas
- Scikit-learn

jupyterlab-space-1 Private

JupyterLab • 5 GB • ml.t3.medium

Status: Stopped

Space Settings Space Settings New

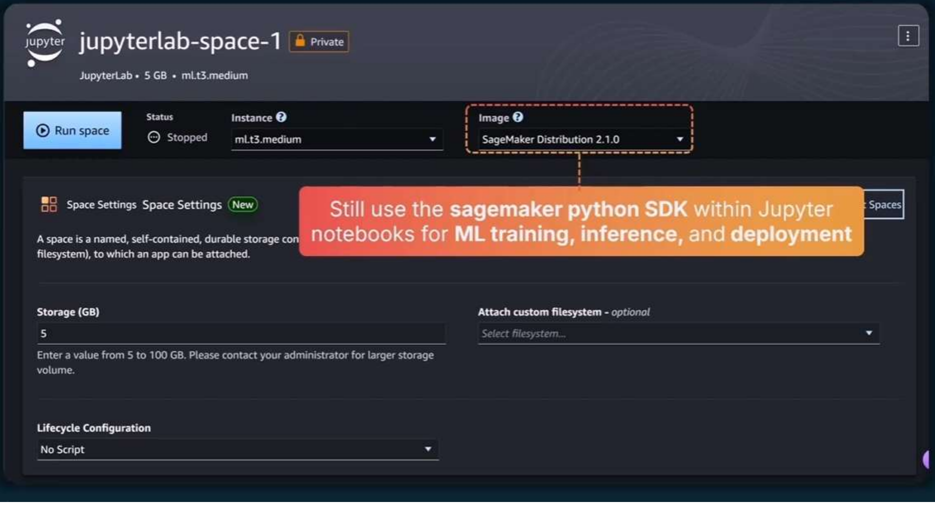
A space is a named, self-contained, durable sto filesystem), to which an app can be attached.

Storage (GB): 5

Enter a value from 5 to 100 GB. Please contact your administrator for larger storage volume.

Lifecycle Configuration: No Script

Workflow: JupyterLab Spaces



jupyterlab-space-1 Private

JupyterLab • 5 GB • ml.t3.medium

Status: Stopped

Instance: ml.t3.medium

Image: SageMaker Distribution 2.1.0

Space Settings Space Settings New

A space is a named, self-contained, durable storage con filesystem), to which an app can be attached.

Still use the **sagemaker python SDK** within Jupyter notebooks for **ML training, inference,** and **deployment**

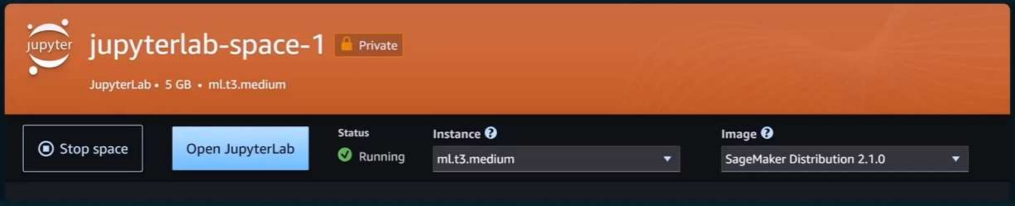
Storage (GB): 5

Enter a value from 5 to 100 GB. Please contact your administrator for larger storage volume.

Attach custom filesystem - *optional*: *Select filesystem...*

Lifecycle Configuration: No Script

Workflow: JupyterLab Spaces



jupyterlab-space-1 Private

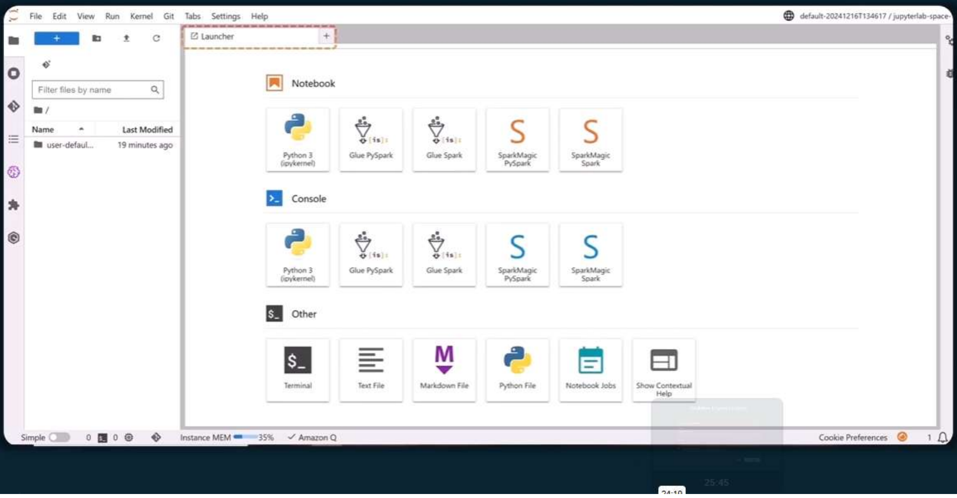
JupyterLab • 5 GB • ml.t3.medium

Status: Running

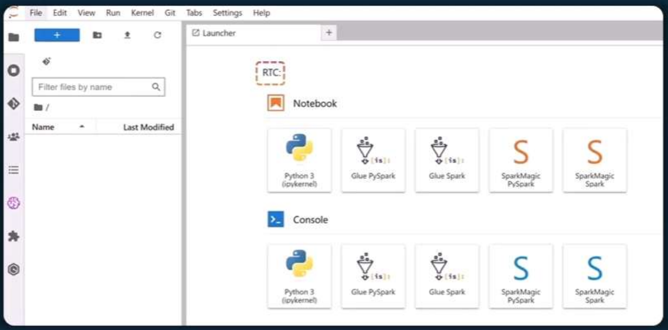
Instance: ml.t3.medium

Image: SageMaker Distribution 2.1.0

Workflow: JupyterLab Spaces

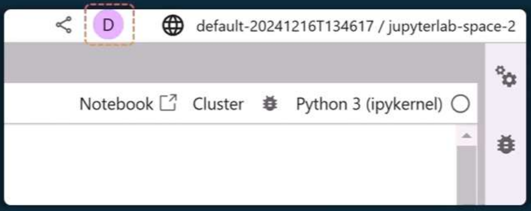


Workflow: JupyterLab Spaces



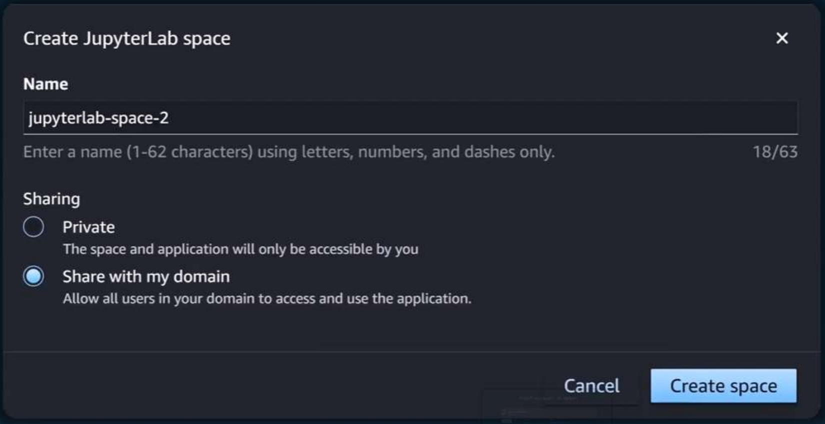
Shared space shows an **RTC indicator** on the launcher page.

Workflow: JupyterLab Spaces



Each user gets a **unique circle identity** in the top right toolbar.

Workflow: JupyterLab Space



Create JupyterLab space

Name

jupyterlab-space-2

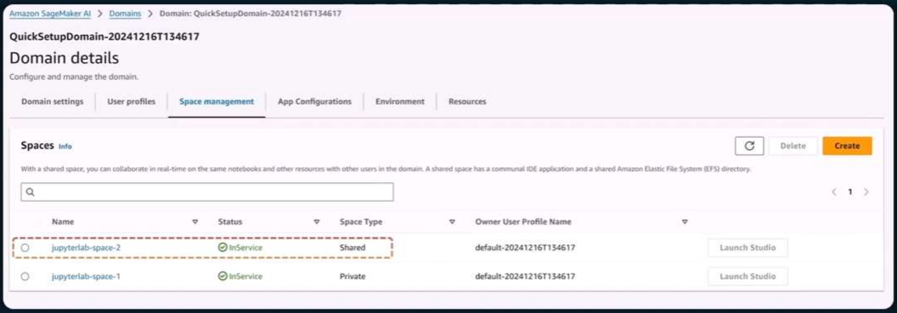
Enter a name (1-62 characters) using letters, numbers, and dashes only. 18/63

Sharing

○ Private
The space and application will only be accessible by you

● Share with my domain
Allow all users in your domain to access and use the application.

Workflow: JupyterLab Space



QuickSetupDomain-20241216T134617

Domain details

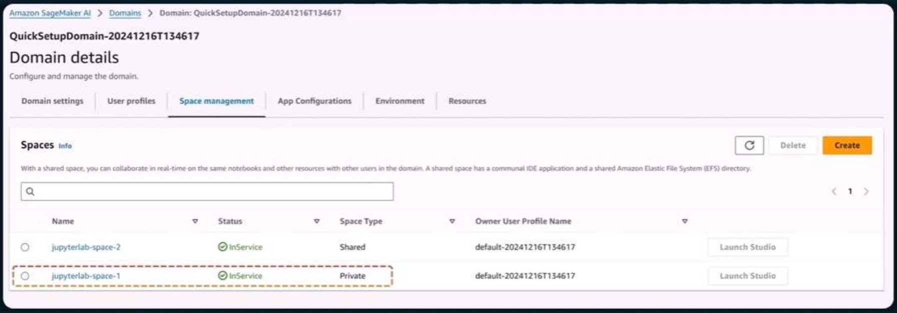
Configure and manage the domain.

Spaces

With a shared space, you can collaborate in real-time on the same notebooks and other resources with other users in the domain. A shared space has a communal IDE application and a shared Amazon Elastic File System (EFS) directory.

	Name	Status	Space Type	Owner User Profile Name	
○	jupyterlab-space-2	✓ InService	Shared	default-20241216T134617	Launch Studio
○	jupyterlab-space-1	✓ InService	Private	default-20241216T134617	Launch Studio

Workflow: JupyterLab Space



QuickSetupDomain-20241216T134617

Domain details

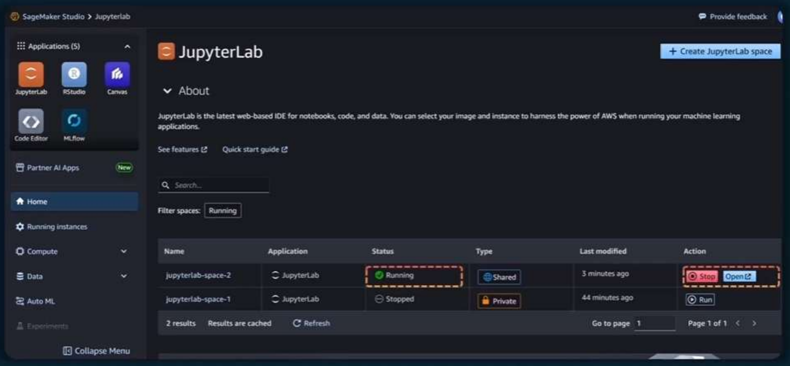
Configure and manage the domain.

Spaces

With a shared space, you can collaborate in real-time on the same notebooks and other resources with other users in the domain. A shared space has a communal IDE application and a shared Amazon Elastic File System (EFS) directory.

	Name	Status	Space Type	Owner User Profile Name	
○	jupyterlab-space-2	✓ InService	Shared	default-20241216T134617	Launch Studio
○	jupyterlab-space-1	✓ InService	Private	default-20241216T134617	Launch Studio

Workflow: JupyterLab Space



JupyterLab

+ Create JupyterLab space

About

JupyterLab is the latest web-based IDE for notebooks, code, and data. You can select your image and instance to harness the power of AWS when running your machine learning applications.

See features · Quick start guide

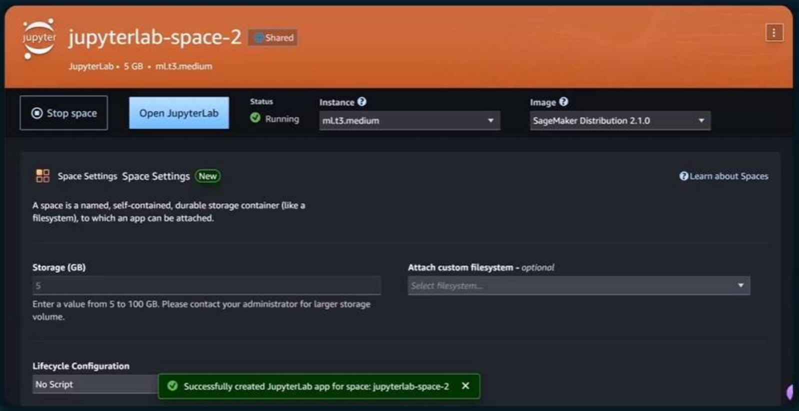
Search...

Filter spaces: Running

Name	Application	Status	Type	Last modified	Action
jupyterlab-space-2	JupyterLab	Running	Shared	3 minutes ago	Stop Open
jupyterlab-space-1	JupyterLab	Stopped	Private	44 minutes ago	Run

2 results · Results are cached · Refresh · Go to page 1 · Page 1 of 1

Workflow: JupyterLab Space



jupyterlab-space-2 Shared

JupyterLab • 5 GB • ml.t3.medium

Status: Running

Instance: ml.t3.medium

Image: SageMaker Distribution 2.1.0

Space Settings Space Settings New

A space is a named, self-contained, durable storage container (like a filesystem), to which an app can be attached.

Storage (GB): 5

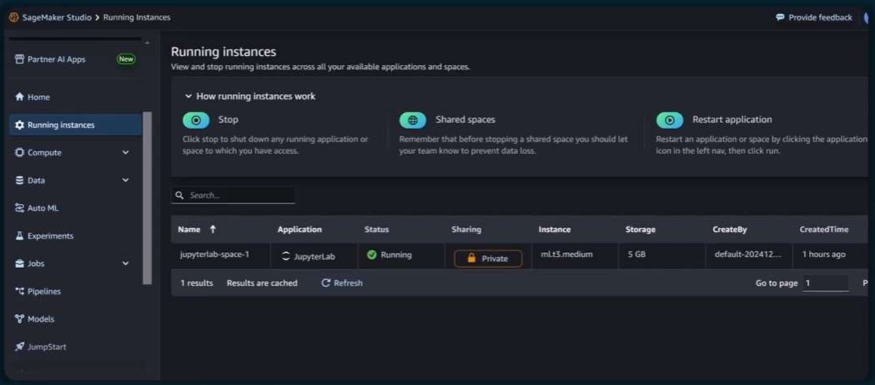
Enter a value from 5 to 100 GB. Please contact your administrator for larger storage volume.

Attach custom filesystem - *optional*: Select filesystem...

Lifecycle Configuration: No Script

Successfully created JupyterLab app for space: jupyterlab-space-2 ✕

Workflow: JupyterLab Spaces



Running instances

View and stop running instances across all your available applications and spaces.

How running instances work

Stop

Click stop to shut down any running application or space to which you have access.

Shared spaces

Remember that before stopping a shared space you should let your team know to prevent data loss.

Restart application

Restart an application or space by clicking the application icon in the left nav, then click run.

Name ↑	Application	Status	Sharing	Instance	Storage	CreateBy	CreatedTime
jupyterlab-space-1	JupyterLab	✓ Running	Private	ml.t3.medium	5 GB	default-202412...	1 hours ago

1 results Results are cached Refresh

Go to page 1

Private spaces are limited to a single **user profile**, with each user able to view only their own spaces in **SageMaker Studio**.

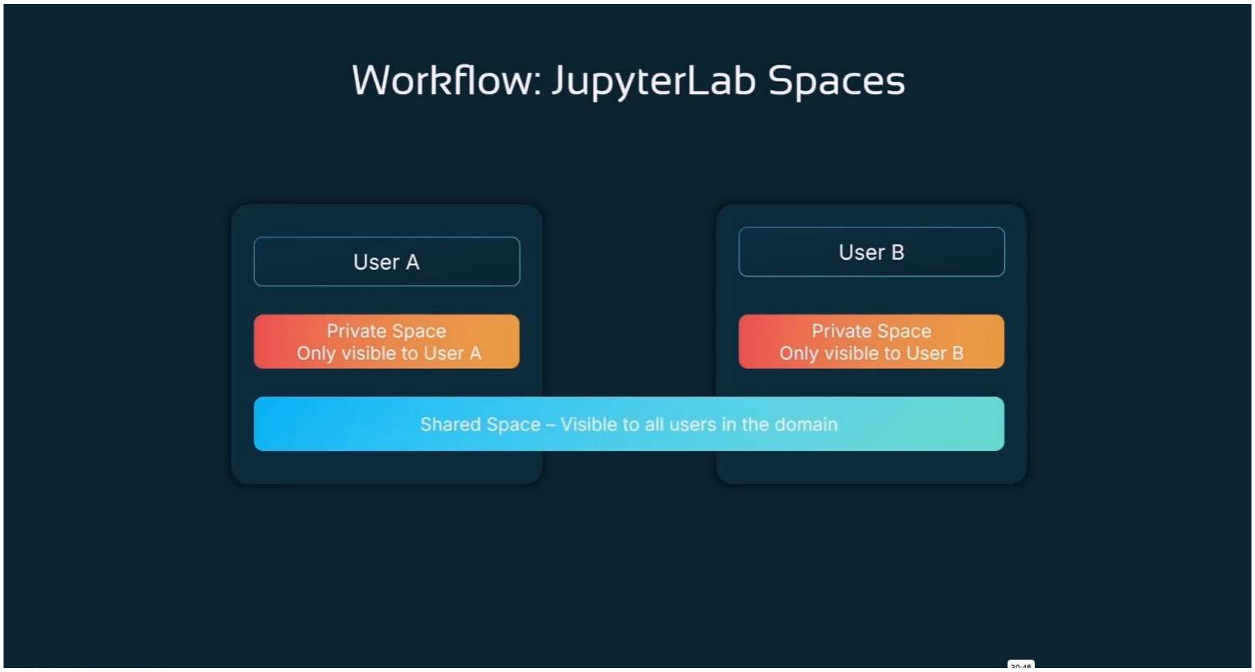
Workflow: JupyterLab Spaces

Private Spaces

- Each user has their own view
- Only creator can access
- Even admins have no direct access

Shared Spaces

- Visible to all users in the domain
- Accessible with proper permissions
- Listed for all users



Workflow: JupyterLab Spaces

User A

Private Space
Only visible to User A

User B

Private Space
Only visible to User B

Shared Space – Visible to all users in the domain