

# Security

Security is increasingly important. As PMA.core has been deployed in increasingly complex scenarios over the years, its security features have evolved, too.

Security pertaining to root-directories is situated at two levels:

- Security features that enable root-directories to access content, such as:
  - Configure public/secret key combinations for S3 resources
  - Configure account credentials to be used when accessing a UNC network resource path
- Prevent [users](#) from access mounted content through root directories that they are or are not allowed to do
  - Define Access control lists

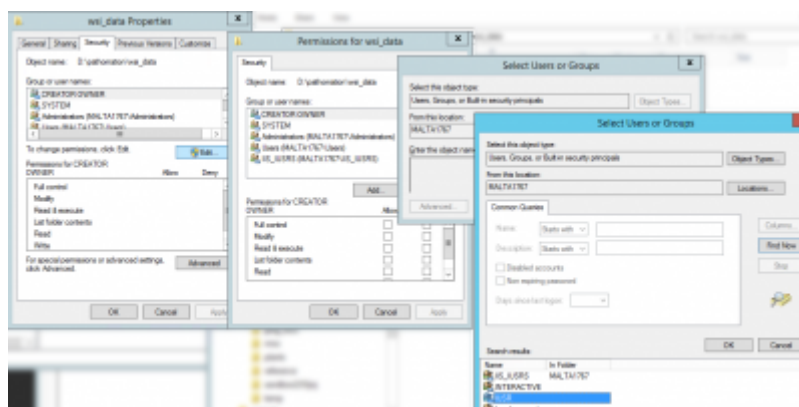
The following paragraphs elaborate on these respective subjects:

## Accessing secured content

Based on the type of data storage that a root directory's mounting point refers to, the configuration offers different options:

### Local hard disk entry points

If you want to expose a local folder on the server's hard disk as a root directory in PMA.core, you have to give the IIS user account access rights to the folder using the Windows Explorer:



Note that even though the dialog shows impersonation options, you can't use these in a local path reference context. The impersonation properties are reserved for networked content, and if you fill them in, PMA.core tries to interpret your local reference as a network path, and subsequently fails trying to access it.

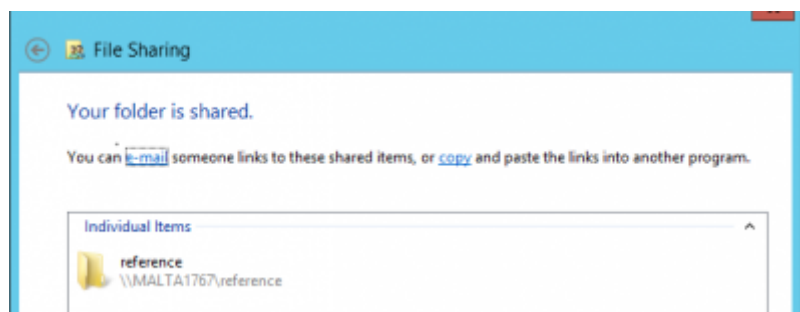
So when defining local hard disk paths, make sure the impersonation options are left blank.

## Network storage (UNC paths)

Pathomation runs under a certain application pool. This application pool is associated with a user identify, which may not have access to the network path that you try to access. Giving access for the application pool to access the network resource may be difficult for a variety of reasons.

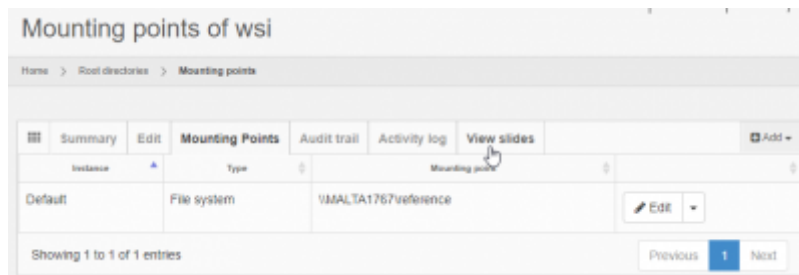
If you can't immediately access the network path with default (i.e. application pool) credentials, you can provide additional information.

In the case below we've created a dedicated pma\_read user that is permitted to access the shared [\\MALTA1767\reference](#) path:



We can enter this as a path for the mounting point, and add the impersonation information for our pma\_read user:

The mounting point shows up, and you can activate the View slides tab to inspect its content:

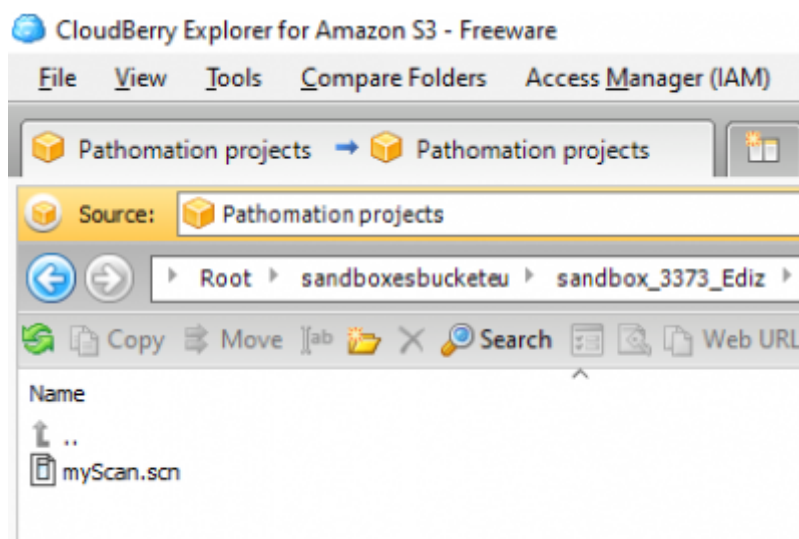


If the credentials are faulty, an error appears

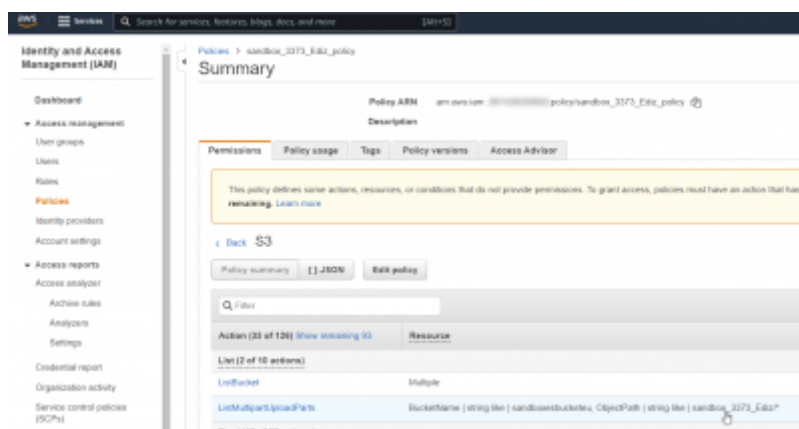
## S3 storage

PMA.core is one of the few vendors that [supports cloud storage natively](#).

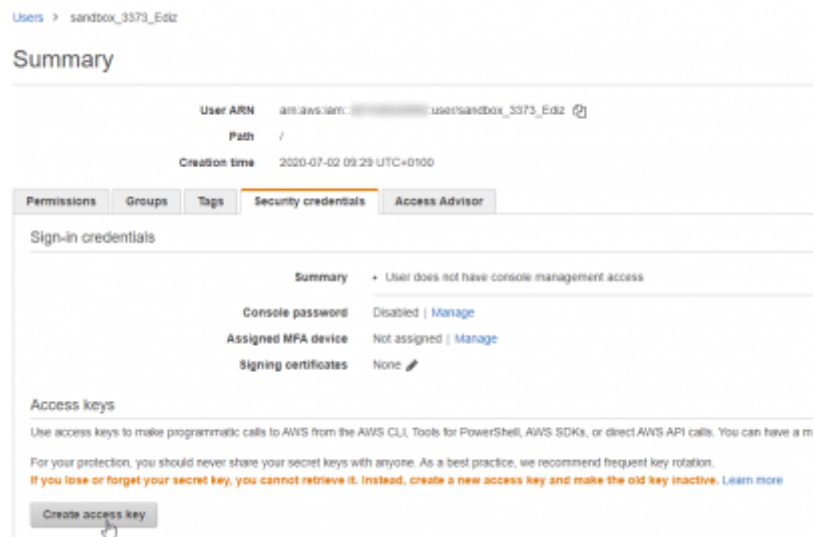
Let's say that you have an S3 bucket and put slides in it:



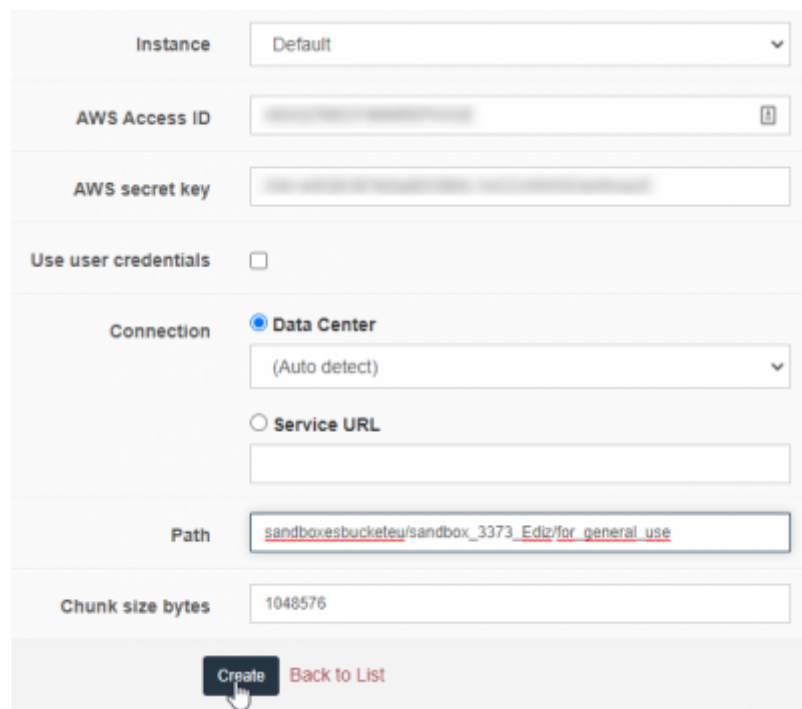
To protect access, you should create a dedicated entity that can only access that content.



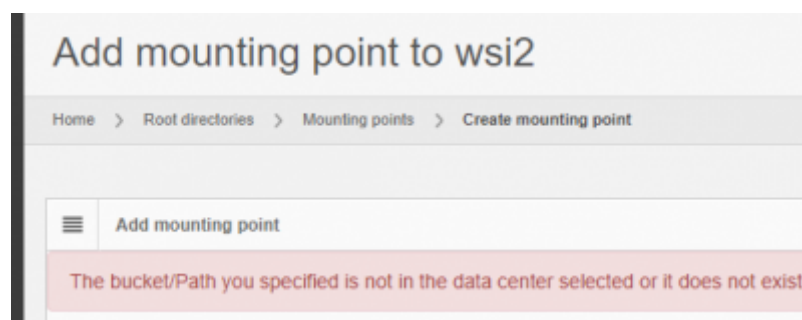
You can then create a pair of dedicated access / secret keys for the new entity:



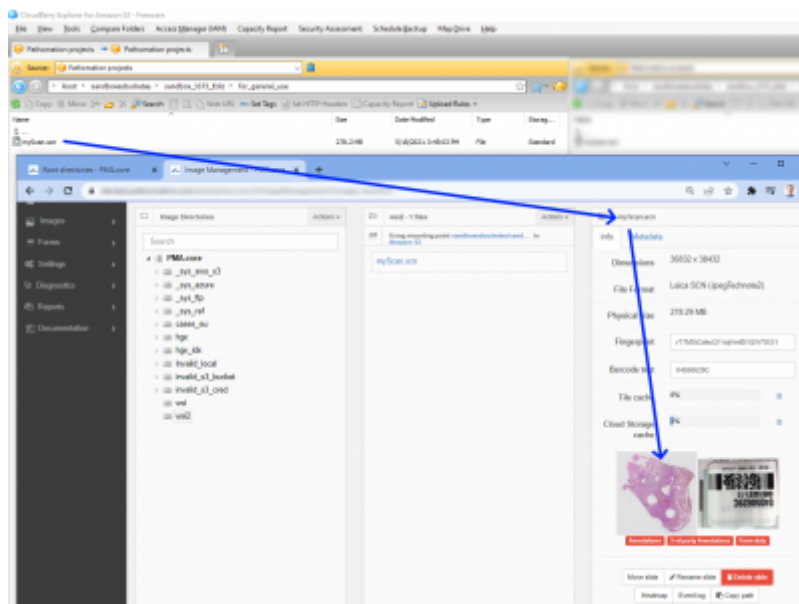
These keys are then used to configure the S3 mounting point at the PMA.core side:



The mounting point only functions when the provided credentials are still active on the S3 storage side. If not, an error message ensues:



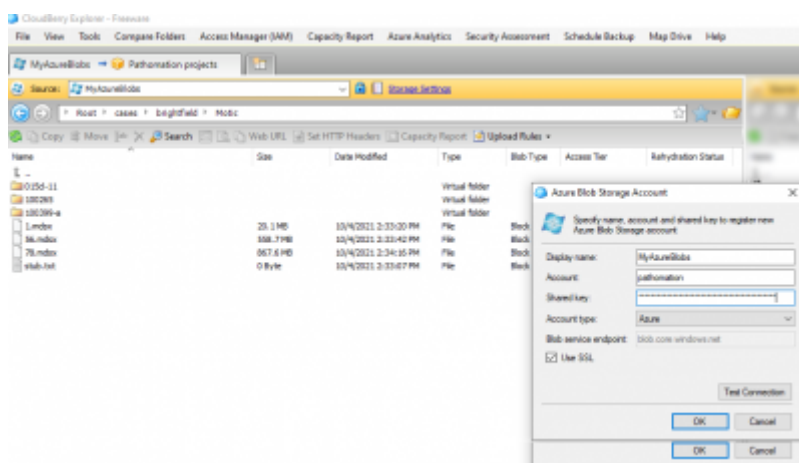
If all is well, you can now browse your slides directly from your S3 content.



## Azure storage

Microsoft Azure has its own protocol, and so we provide a separate mounting point type of it.

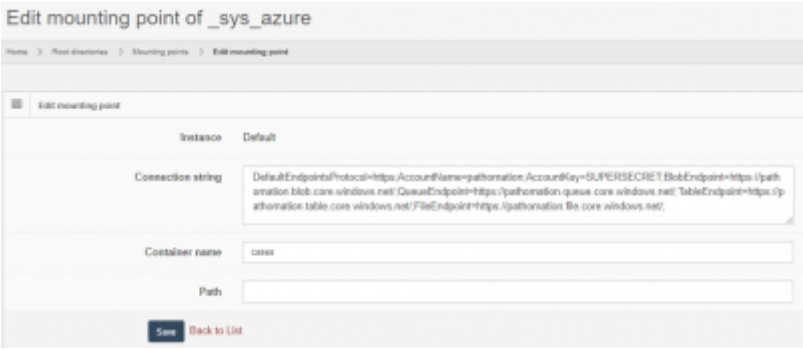
Let's say that you have an Azure container defined and put some slides in it already:



You can convert these credentials in a connectionstring:

```
DefaultEndpointsProtocol=https;AccountName=pathomation;AccountKey=UPERSECRET;BlobEndpoint=https://pathomation.blob.core.windows.net/;QueueEndpoint=https://pathomation.queue.core.windows.net/;TableEndpoint=https://pathomation.table.core.windows.net/;FileEndpoint=https://pathomation.file.core.windows.net/;
```

This text snippet is then pasted in the connection string field of the mounting point properties:

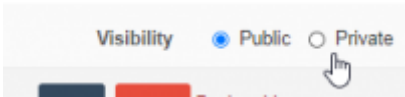


If all goes well, you can now serve your slides from your Azure storage repositories.

Public vs private

As you have more users and more root-directories, it becomes undesirable that everybody is allowed to see everything.

Therefore, root-directories can be marked “public” or “private”:

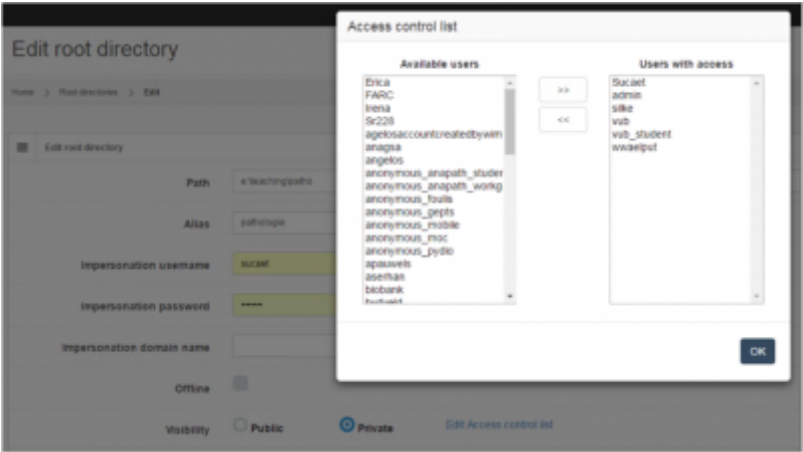


Public root directories are marked “public”, it means every user has access to them. They can be accessed by anybody who is a registered user in the PMA.core user repository.

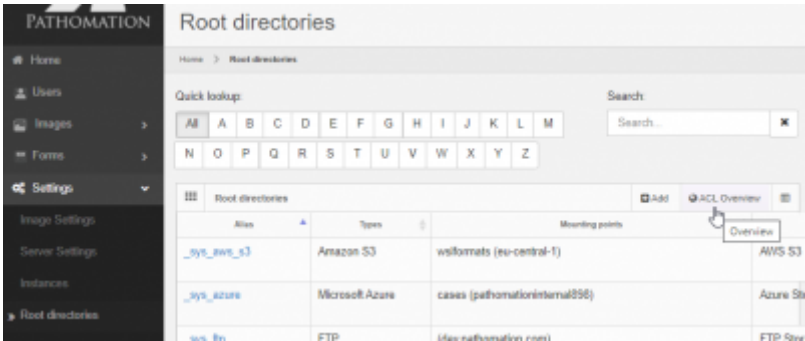
Private root directories are marked “private”, it means only select users can see the content. They are only accessible by those who have been explicitly given access to be allowed to access the folder through the directory's access control list.

Access control lists

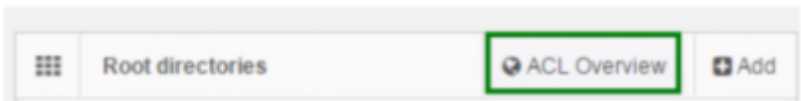
Once marked private, you can select what users are allowed to see the content of the root directory, and which ones aren't: Do this by pressing the “Edit access control list” link after you selected the “private” option:



An interactive overview grid is available via the Root directories management view:



As you get even more root-directories and more users, it is useful to get an overview of who has access to what. For that, you can request the ACL report from the root-directories view.



The resulting report looks like this:

Root directories ACL overview							
Home > Root directories > Overview							
Root directories ACL overview							
User / Directory	ICON	MOC	Workgroup	_in_aperio	_in_fluo_bd	_in_fluo_nikon	_in_fluo_zeiss
Erica							
FARC				✓	✓	✓	✓
Irena							
Sr228	✓			✓	✓	✓	✓
Sucaet	✓	✓	✓	✓	✓	✓	✓
admin	✓			✓	✓	✓	✓
agelosaccountcreatedbywiminydio							
anagsa			✓				
angelos							
anonymous_anapath_student							
anonymous_anapath_workgroup							
anonymous_fouls							
anonymous_gepts							
anonymous_mobile							
anonymous_moc		✓					

From:  
<https://docs.pathomation.com/pma.core/2.0/> - **PMA.core 2.x**

Permanent link:  
[https://docs.pathomation.com/pma.core/2.0/doku.php?id=rootdir\\_security&rev=1647254887](https://docs.pathomation.com/pma.core/2.0/doku.php?id=rootdir_security&rev=1647254887)

Last update: **2022/03/14 13:48**