From The Architects

Veeam Backup & Replication

V12 Enhanced security and scalability with object storage Secure Mode

Written by Luca Dell’Oca

Table of Contents

Introduction

One of the most anticipated features of Veeam Backup & Replication v12 is the ability to write backups directly to Object Storage (also referred to as “Direct to Object” or D2O in short). This capability unlocks all the powerful use cases that involve object storage as now it can be used as a primary target for any backup and restore activity in Veeam Backup & Replication, Veeam Agents and Veeam Cloud Connect.

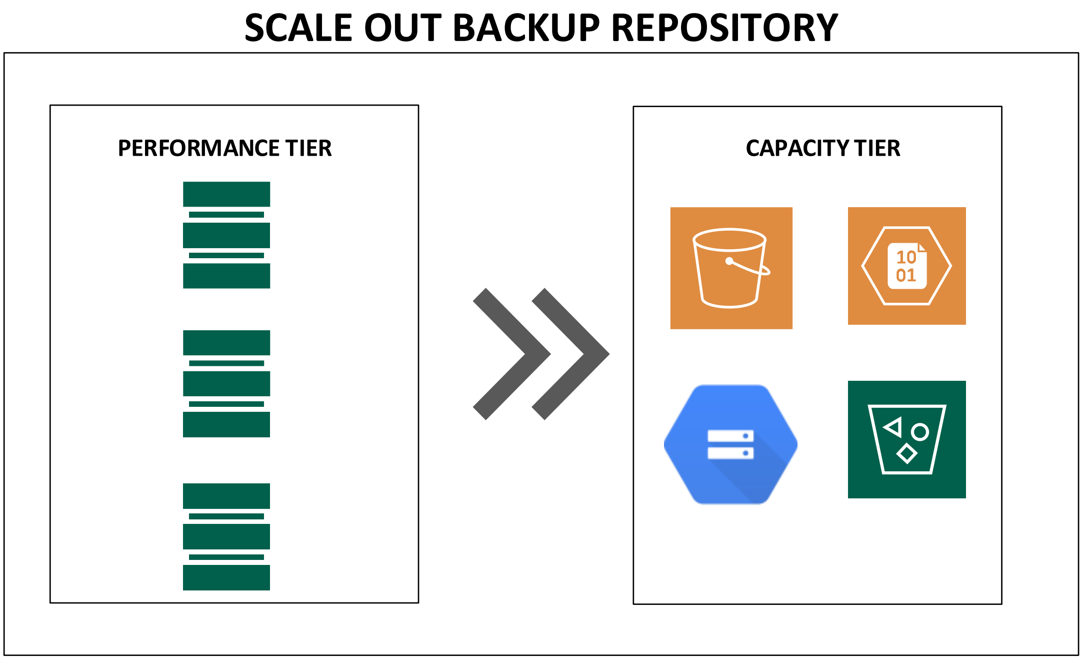
Support for block storage is not going away, but the possibility to choose between Block and Object adds even more design possibilities to Veeam architects.

In this document, while describing the design and use cases of D2O, we will focus on some advanced capabilities of Object Storage, specifically the multi-tenancy options. On AWS and S3-compatible storage Veeam Backup & Replication v12 can leverage IAM and STS. On Microsoft Azure we will leverage SAS. In the paper we will refer to these as **"Secure Mode"**. We’ll explain what it is, the benefits, security implications, and how it can be used in both end users and service provider environments.

The premise: Object Storage is ready for prime time

There is little doubt that Object Storage is the “storage of the future”. Capabilities like scale-out, multi-node, redundancy, replication, and transparent failover, make this solution a hassle free choice for building – or consuming – storage.

Veeam introduced support for Object Storage back in 2019 with Veeam Backup & Replication 9.5 Update 4. Initially it was usable only as a “Capacity Tier” inside a Scale-Out Backup Repository: multiple performance extents backed by Block Storage volumes would receive backup data from Veeam proxies and Agents, and then a move or copy operation would transfer the data to the Object Storage:



This made perfect sense at the time, as every customer had block storage available in their datacenters, and the operation of adding a Capacity Tier is easy and transparent, especially when consuming an off-site Capacity Tier from a public cloud provider.

With the advancements in technology, these days Object Storage has become more and more common, not only when consumed “from the Cloud”, but also as a powerful technology deployed at the customer premises or at the service provider offering storage services. One negative comment about Object Storage was around its performance: yes it was cheap and scalable, but speed was not comparable to Block Storage. This made it the perfect candidate for file and archive storage. But technology moves on, and the new solutions on the market and the improvements in Veeam capability to write to object storage has brought us to Veeam Backup & Replication v12 and Direct To Object, with performance totally comparable to Block Storage

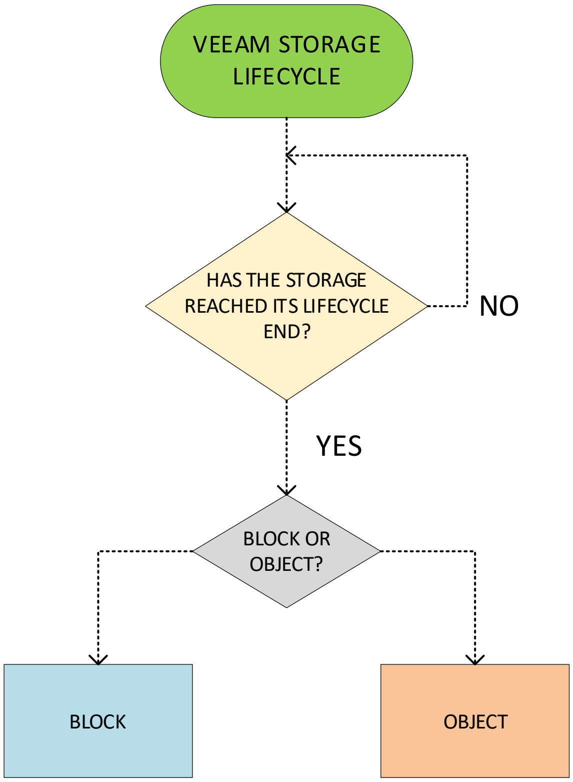
Planning: should I move immediately to Object Storage?

Veeam Backup & Replication is a software-based solution. This gives our partners and customers extreme flexibility when designing the best architecture for their own needs.

Direct backup to object storage can be seen as another great option when choosing which storage solution to use for a given Veeam environment. Existing block storage investments can be leveraged at their maximum potential, without the need to replace them to introduce a different technology. There is no pressure from Veeam to use Object Storage, and both solutions are going to be supported for the foreseeable future.

At some point in time however, the existing hardware will reach their lifecycle end, either because the hardware itself is too old to support the ever-growing environment, or simply because the 3- or 5-years support contract with the vendor will expire.

That could be a good time to think about introducing Object Storage as the new primary storage for backups, following a workflow like this:



Obviously, in new environments this decision can be taken during the planning and design phases. A new environment can also be an existing environment that needs additional storage space: maybe a server provider is already offering Veeam Cloud Connect with Block Storage. With the upgrade to Veeam Backup & Replication, they can “start small” with D2O by adding this new capability and offer it to v12 customer. Two options, maybe with different prices, to further expand the offered service. In this way, when Object Storage will become the main solution, the provider will have a great experience already in operating a D2O system.

Security considerations

Let’s suppose an architect has decided to use an Object Storage as the new primary backup target for Veeam Backup & Replication v12.

Aside of the choice of the storage vendor to use, the main architectural decision to make is: which access mode should be used? This may sound like a simple question, but there are multiple combinations, and each have its pros and cons, as well as different security implications. We will provide a more detailed explanation of each mode later in this document.

We will dig deeper into each mode later in the paper, but for now let’s focus on the Security point of view. It is important to understand the situation as well as why Secure Mode is so important and powerful.

The access to resources in Veeam Backup & Replication is done using credentials. For users these are usually called username and password. In Object Storage they have different names depending on the used technology, but in essence they are all a combination of two strings. AWS, Google and S3-Compatible storage solutions call them **Access Key / Secret Key**, while on Azure storage they are **Account / Shared key**.

These Cloud Credentials are registered into Veeam Backup & Replication so that the software can access the storage and operate on it, doing reads, writes and deletes.

These credentials allow their owner to read data of every workload that is stored in the bucket. While this is not a problem in common use cases, like a single environment storing backups of all the virtual machines under its protection, it can become a security issue in a multi-tenant environment.

For example, think about multiple Veeam agents, installed on all the laptops of a company, sending data to the same Object Storage bucket: by using the same set of credentials, each user can read the data of any other user’s backup stored in the same bucket.

A possible workaround could be to create multiple buckets, and assign a different bucket to each agent, but even with just a few managed agents, this design would quickly become an operational nightmare. We need something that can allow at the same time concurrent access to a shared storage resource, while guaranteeing confidentiality to the stored data.

We can apply two possible solutions:

- Abstraction (there is no native multitenancy, but a layer in between source and destination hides the resources not belonging to the requester)
- Segregation (there is native multitenancy, that allows the requester to only see data belonging to it)

Veeam Backup & Replication can use both, depending on the chosen configuration.

Data streams

Before digging into the different design options, let's talk about the three types of data streams we have in each use case. It's important to understand them in order to make a choice.

Authentication Path: in each scenario a different entity is responsible for authenticating the different Veeam data movers connecting to the storage solution (or the gateway in front of it) and determine to which data they have access to. It can be the Veeam Backup Server (if using abstraction) or the Object Storage (if using segregation).

Control Path: once the client has been authenticated, it receives commands from the controlling Veeam Server. The Control Path is the connection between remote agents and the central system to send control commands, push backup policies, retrieve backup reports, and so on. This is a light traffic.

Backup Path: the connection between remote Veeam agents (Windows / Linux / Mac) and the storage system to write backups and read them during a restore. This is a network-intensive traffic.

Direct to Object Storage configurations

Now we can have a look at the 5 possible ways to consume an Object Storage we mentioned before. Some of them will use abstraction, some will use segregation:

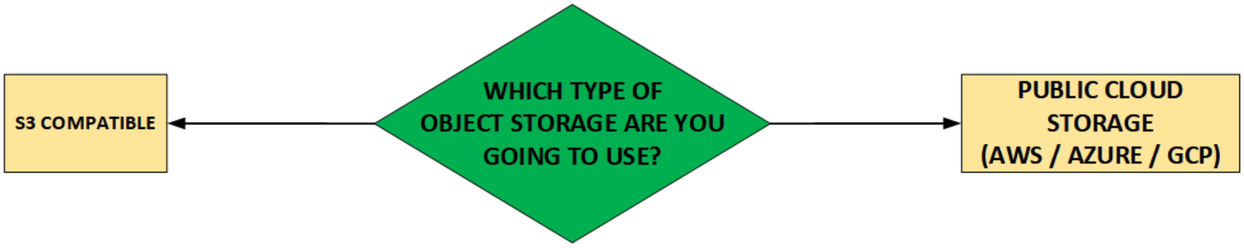
Gated Access (to public cloud storage)
Direct Access (to public cloud storage)
Secure Indirect Access (to S3-compatible storage)
Insecure Direct Access (to S3-compatible storage)
Secure Direct Access (to S3-compatible storage)

In each configuration, the three paths explained before are different.

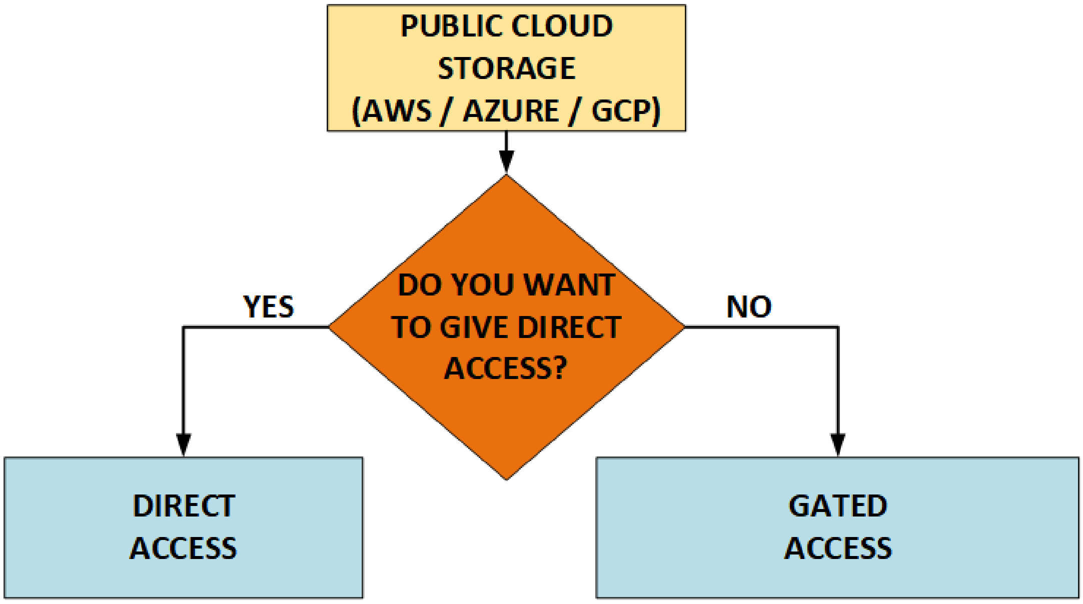
Note*: in this example, we will talk about a Service Provider using Object Storage for its Cloud Connect services. The same design process can be applied to end users by replacing Cloud Connect with Veeam Backup & Replication.*

Let's play the part of a backup architect. We have implemented Veeam Backup & Replication v12, we can now write backups directly to Object Storage, and we need to choose which Object Storage configuration we want to use.

The first choice to be done is whether we are going to consume one of the three hyperscaler offerings - Amazon Web Services S3, Azure Blob, or Google Cloud Storage – or an S3-compatible solution from a 3rd party – either on-premises or available via the Internet from a Service Provider.


S3 COMPATIBLE
WHICH TYPE OF OBJECT STORAGE ARE YOU GOING TO USE?
PUBLIC CLOUD STORAGE (AWS / AZURE / GCP)

We then have two options: **Gated** or **Direct**.


PUBLIC CLOUD STORAGE (AWS / AZURE / GCP)
DO YOU WANT TO GIVE DIRECT ACCESS?
YES
NO
DIRECT ACCESS
GATED ACCESS

```
Auth    : Public Cloud
Control: Agent -> VCC gtw -> VCC
Backup : Agent -> OBJS
```

```
Auth    : VCC Server
Control: Agent -> VCC gtw -> VCC
Backup : Agent -> VCC gtw -> OBJS gtw -> OBJS
```

Note on the used terms:

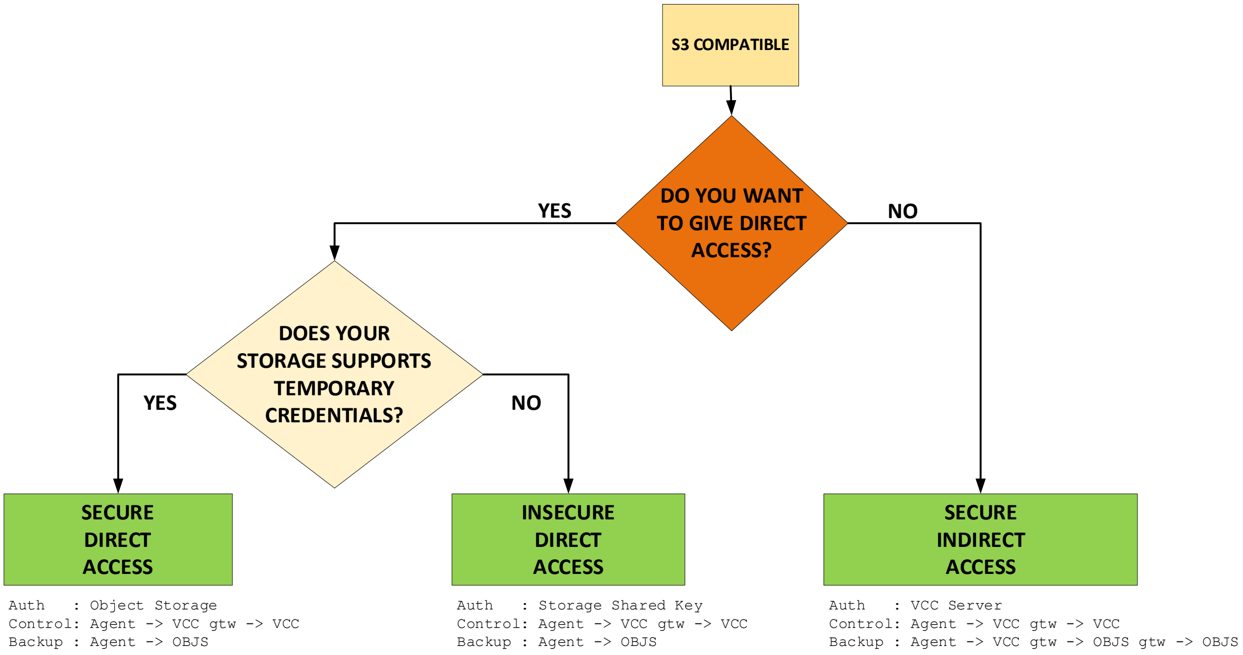
VCC (Server) = Veeam Cloud Connect
VCC gtw = Veeam Cloud Gateway
OBJS = Object Storage
OBJS gtw = Object Storage gateway

Auth: Authentication Path
Control: Control Path
Backup: Backup Path

Gated Access uses Abstraction to protect the access to the storage. The authentication of each agent is done by Cloud Connect (or Veeam Backup server in the end user scenario) using the same credentials for every connection. Backup traffic goes through the Object Storage gateway (the Cloud Connect Gateway in a service provider design, the Object Storage gateway in an end user design) and this is the key: the Cloud Connect gateway only shows to the remote agent the data that it has rights to access, hiding every other Veeam agents' data.

Direct Access instead uses Segregation to protect the access to the storage. The public cloud provider (we'll show the details of how each provider does this later in the paper) can authenticate the agent not just by using the shared keypair, but specifically recognizing the single remote Veeam agent, and then providing access only to the resources it has rights for.

For a Veeam Cloud Service Provider it would make much more sense to consume an on premises Object Storage to maximize the investment in a owned storage, so let's have a look at the other three possible use modes.


S3 COMPATIBLE
DO YOU WANT TO GIVE DIRECT ACCESS?
YES
NO
DOES YOUR STORAGE SUPPORTS TEMPORARY CREDENTIALS?
YES
NO
SECURE DIRECT ACCESS
INSECURE DIRECT ACCESS
SECURE INDIRECT ACCESS
Auth   : Object Storage
Control: Agent -> VCC gtw -> VCC
Backup : Agent -> OBJS
Auth   : Storage Shared Key
Control: Agent -> VCC gtw -> VCC
Backup : Agent -> OBJS
Auth   : VCC Server
Control: Agent -> VCC gtw -> VCC
Backup : Agent -> VCC gtw -> OBJS gtw -> OBJS

Secure Indirect Access.
In this scenario we still leverage Gated Access and the remote agent is authenticated by the Veeam backup server. This activity is secure thanks to the fact that the gateways will abstract the content of the Object Storage as explained before. This design is exactly like the previous versions of Veeam Cloud Connect, just replacing Block Storage with Object Storage.

If we want to leverage D2O and remove backup traffic from flowing through the Veeam Cloud Gateways and Object Storage gateways, we need to opt for Direct Access modes.

Here we have two sub-options: Insecure Direct Access and Secure Direct Access.
Depending on the capabilities of the chosen S3-compatible storage, they can offer the same authentication methods of the AWS S3 storage in terms of multi-tenancy. S3 systems in fact can leverage STS (Security Token Service) to create multi-tenant access to a shared S3 bucket.

If the chosen storage doesn't support STS, but we still want to offer D2O, we can choose **Insecure Direct Access**. The name is self-explanatory: access will still be direct to Object Storage, but there will be no authentication mechanism. Each agent with the authentication keypair will have complete access to the whole content of the bucket.

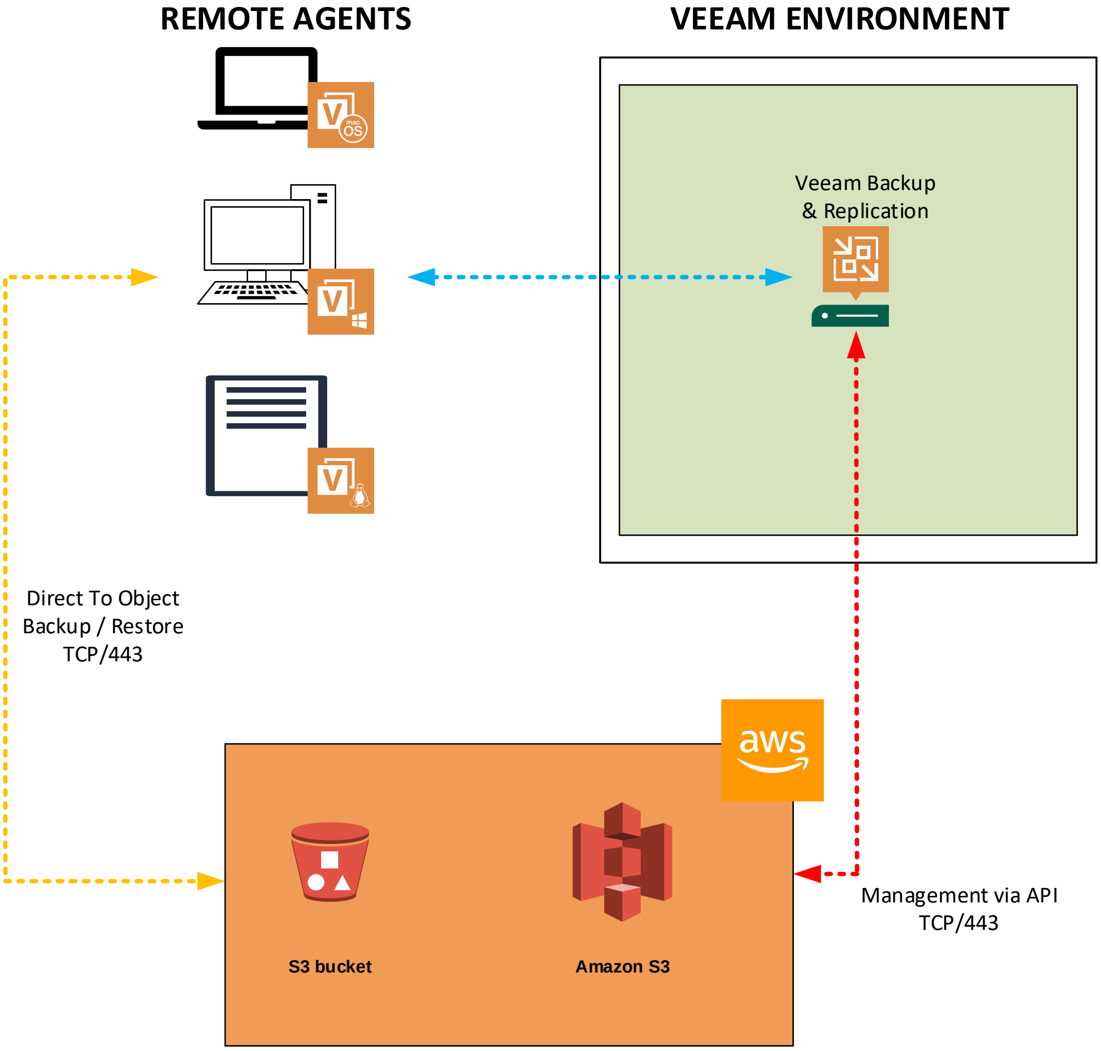
This is a sub-optimal solution and therefore we suggest to evaluate Secure Indirect instead.

If the chosen storage supports STS, we can use **Secure Direct Access**. In this final use case, the Object Storage and its IAM/STS services will be responsible for the authentication of each remote Veeam agent.

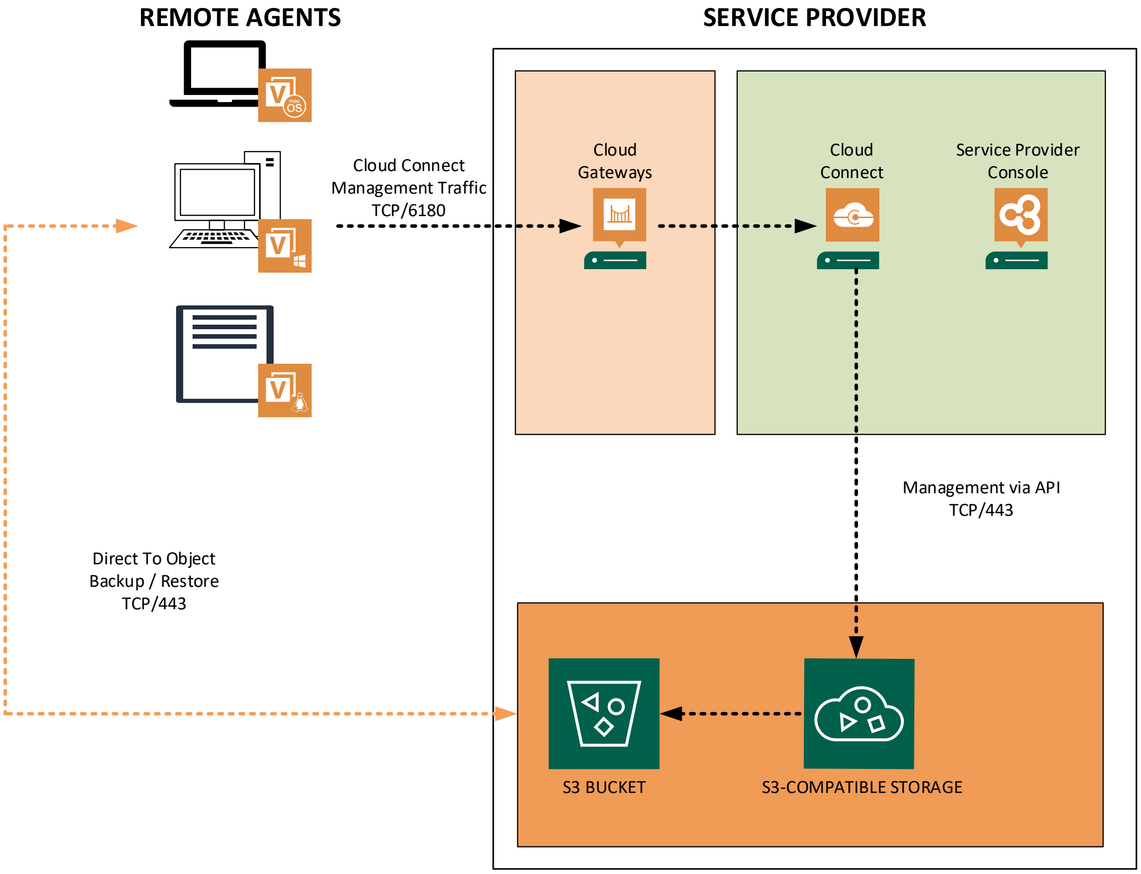
The backup data stream will flow directly from the remote agents directly to the Object Storage. It's the same design as Direct Access for public cloud, but this time served via a S3-compatible solution.

In the rest of this document we will focus on Secure Direct Access mode - which we'll refer to as **Secure Mode** - because it's the most effective option. The reason for this is that it is a totally different and powerful approach to the data path. In Gated Access mode data must flow through Veeam Cloud Gateways which can then become a bottleneck. In Direct Access mode data flows directly from each remote agent to the object storage directly thus removing the load from the Veeam infrastructure, which now becomes a pure Management Plane.

We see and expect two main use cases: end user sending backups to a public cloud Object Storage


REMOTE AGENTS
VEEAM ENVIRONMENT
Veeam Backup & Replication
Direct To Object Backup / Restore TCP/443
aws
S3 bucket
Amazon S3
Management via API TCP/443

and Service Provider using D2O to offer offsite backup services via Cloud Connect



In the following pages, we'll present the Service Provider use case. All the concepts explained can also be applied to end users.

How does Secure Mode works, and how we use it?

The entire idea of Secure Mode is to leverage an **“automatic” multi-tenancy**. There is no need to manually create and manage all the different access keys nor the related access permissions on the S3 bucket for every single remote agent.

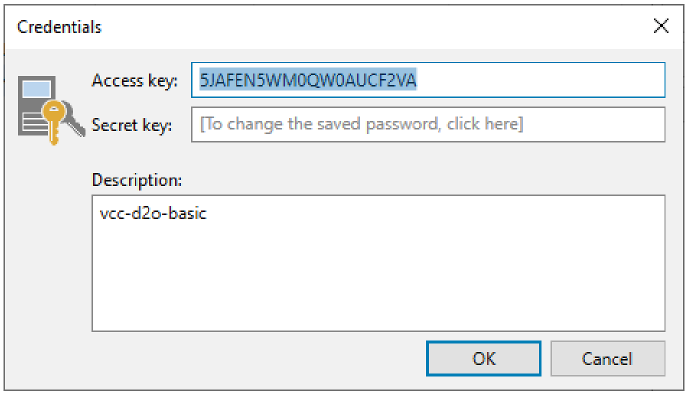
The AWS S3 API (and by inheritance, many S3-Compatible storage solutions) have an option called STS. Azure and Google have a different yet comparable solution. For simplicity, we will use the S3 API to explain Secure Mode and later we’ll show how it works in Microsoft Azure and Google Cloud.

Example scenario

To make things more understandable, we’ll follow a practical example: a Service Provider with local Object Storage is going to implement Secure Mode for its remote customers.

Configuration

During the configuration phase of a new Object Storage solution in Veeam Backup & Replication we register the pair of **Access Key** and **Secret Key**.



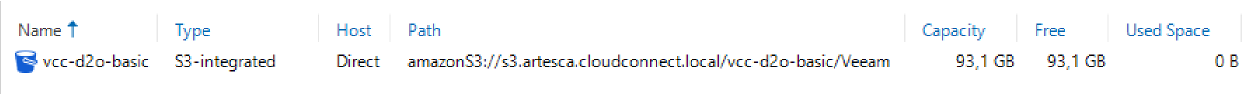
Access key: 5JAFEN5WM0QW0AUCF2VA

Secret key: [To change the saved password, click here]

Description:

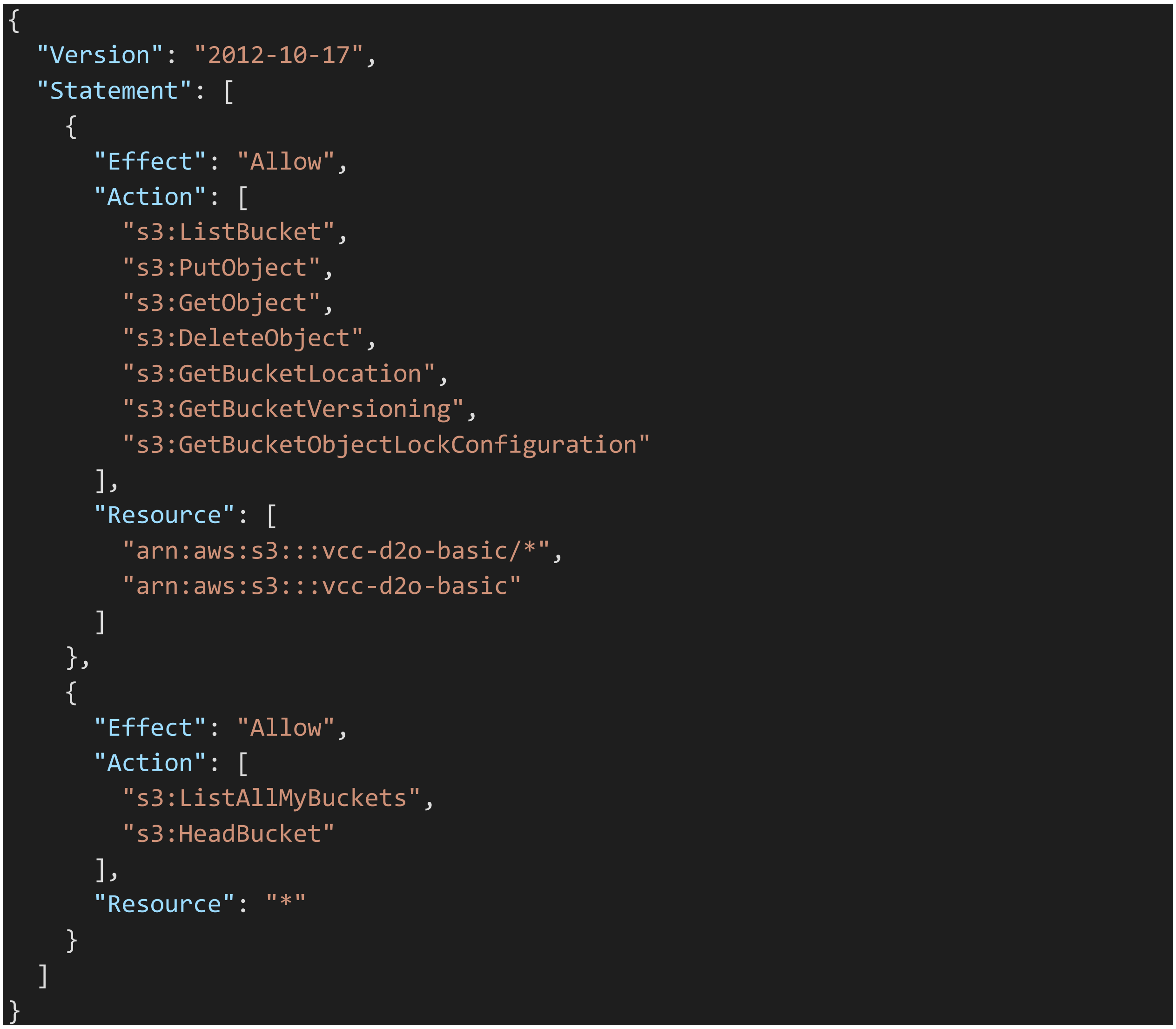
vcc-d2o-basic

With these credentials, Veeam Backup & Replication can now connect to the Object Storage and operate on a given bucket:



Name ↑	Type	Host	Path	Capacity	Free	Used Space
vcc-d2o-basic	S3-integrated	Direct	amazonS3://s3.artesca.cloudconnect.local/vcc-d2o-basic/Veeam	93,1 GB	93,1 GB	0 B

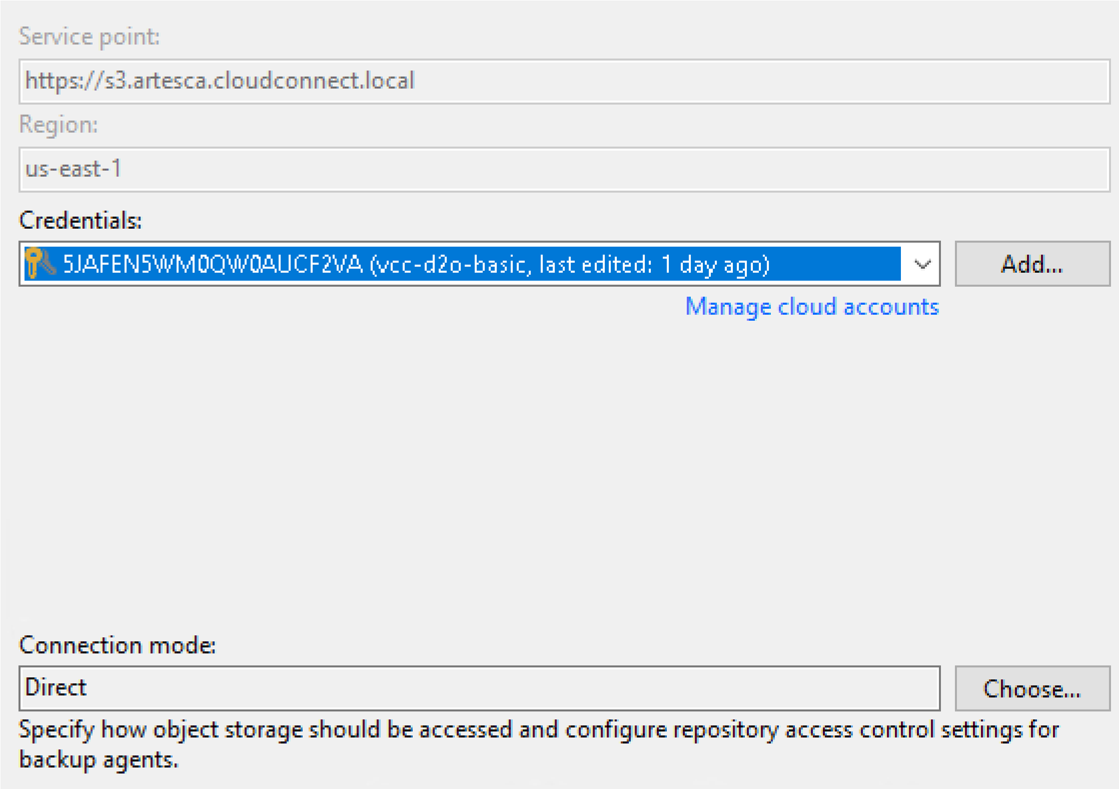
This is possible because a IAM policy has been previously created and mapped to the user to whom this access key belongs to:



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:GetBucketVersioning",
        "s3:GetBucketObjectLockConfiguration"
      ],
      "Resource": [
        "arn:aws:s3:::vcc-d2o-basic/*",
        "arn:aws:s3:::vcc-d2o-basic"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:HeadBucket"
      ],
      "Resource": "*"
    }
  ]
}
```

A typical IAM policy for S3 read/write access.

When adding the new bucket, we have to select “Direct” as the Connection mode:



Service point:

https://s3.artesca.cloudconnect.local

Region:

us-east-1

Credentials:

5JAFEN5WM0QW0AUCF2VA (vcc-d2o-basic, last edited: 1 day ago)

Connection mode:

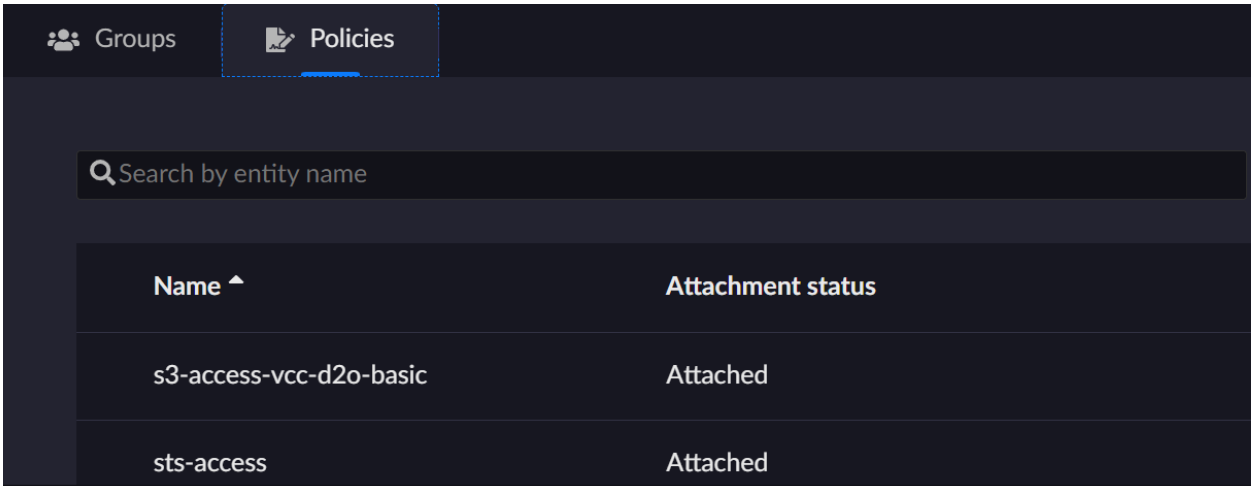
Direct

Specify how object storage should be accessed and configure repository access control settings for backup agents.

With these operations, the bucket is mounted, but it's still not using Secure Mode. In order to enable Secure Mode an additional policy is needed:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetPolicyVersion",
        "iam:DeleteAccessKey",
        "iam:GetPolicy",
        "iam:AttachUserPolicy",
        "iam:DeleteUserPolicy",
        "iam:DeletePolicy",
        "iam:DeleteUser",
        "iam:ListUserPolicies",
        "iam:CreateUser",
        "iam:TagUser",
        "iam:CreateAccessKey",
        "iam:CreatePolicy",
        "iam:ListPolicyVersions",
        "iam:GetUserPolicy",
        "iam:PutUserPolicy",
        "iam:ListAttachedUserPolicies",
        "iam:GetUser",
        "iam:CreatePolicyVersion",
        "iam:DetachUserPolicy",
        "iam:DeletePolicyVersion",
        "iam:ListAccessKeys",
        "iam:SetDefaultPolicyVersion"
      ],
      "Resource": "*"
    }
  ]
}
```

This policy is the key to the entire configuration. By mapping this policy to the same user we have created before for S3 access, the Veeam backup server will be able to connect to the Object Storage with the same keypair and be authorized to manage the Security Tokens (STS).

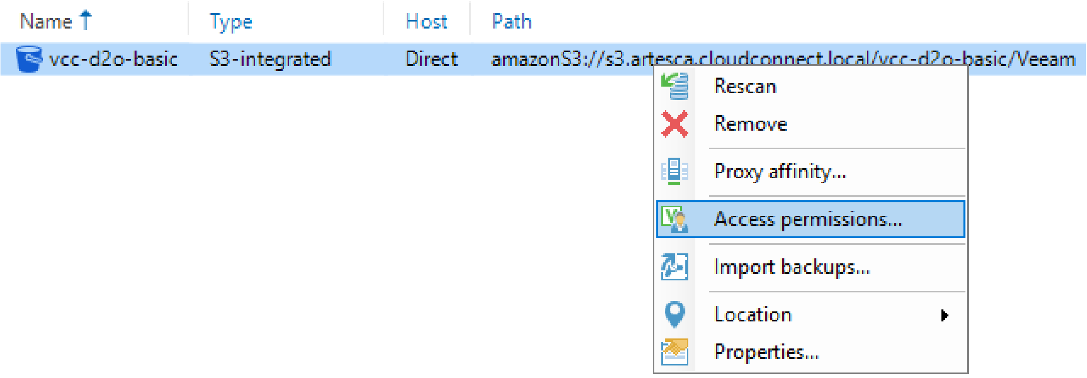


Name	Attachment status
s3-access-vcc-d2o-basic	Attached
sts-access	Attached

We can see in the IAM policies of the Object Storage, that the user has both policies. One to access the bucket and one to do STS operations.

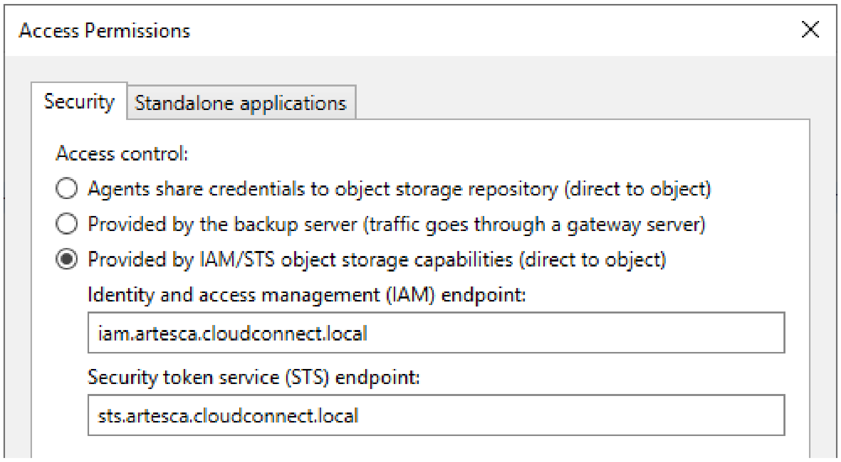
Once the policies are attached, it’s time to configure Secure Mode in the Veeam user interface!

Open the access permissions of the registered S3 bucket:



Name	Type	Host	Path
vcc-d2o-basic	S3-integrated	Direct	amazonS3://s3.artesca.cloudconnect.local/vcc-d2o-basic/Veeam

Here we see the three configuration modes mentioned before:



Access control:

○ Agents share credentials to object storage repository (direct to object)

○ Provided by the backup server (traffic goes through a gateway server)

● Provided by IAM/STS object storage capabilities (direct to object)

Identity and access management (IAM) endpoint: iam.artesca.cloudconnect.local

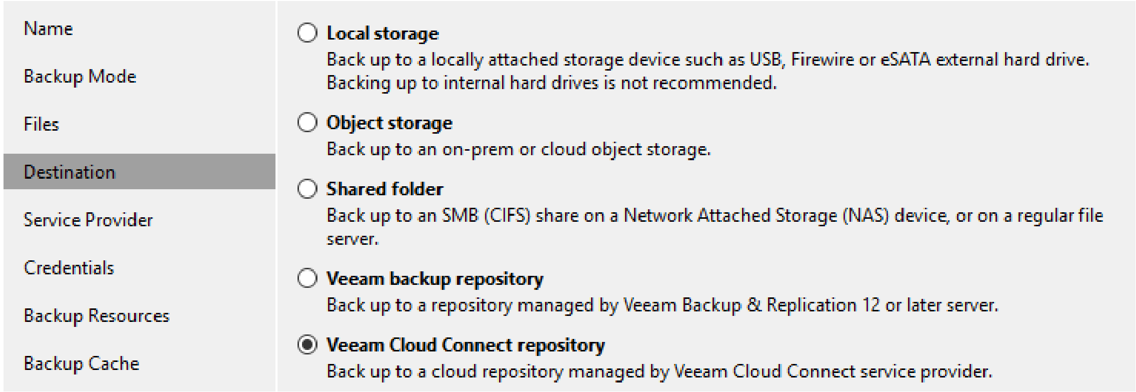
Security token service (STS) endpoint: sts.artesca.cloudconnect.local

First is Insecure Direct, second is Secure Indirect, and the last one is the Secure Direct Access. The last one is the one we want to use. Select it and enter the URL’s of the two endpoints: IAM and STS.

Let’s do backup!

Once the storage is configured, we can create a new tenant in Cloud Connect that will consume the Object Storage as its Cloud repository. Once it’s all set, we can configure our first remote agent to see what’s going to happen.

The job configuration is straightforward. Choose to send backups to a Cloud Connect repository:



Name
Backup Mode
Files
Destination
Service Provider
Credentials
Backup Resources
Backup Cache

○ **Local storage**
Back up to a locally attached storage device such as USB, Firewire or eSATA external hard drive. Backing up to internal hard drives is not recommended.

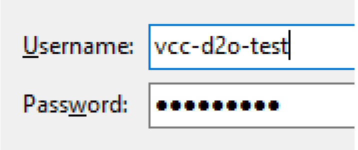
○ **Object storage**
Back up to an on-prem or cloud object storage.

○ **Shared folder**
Back up to an SMB (CIFS) share on a Network Attached Storage (NAS) device, or on a regular file server.

○ **Veeam backup repository**
Back up to a repository managed by Veeam Backup & Replication 12 or later server.

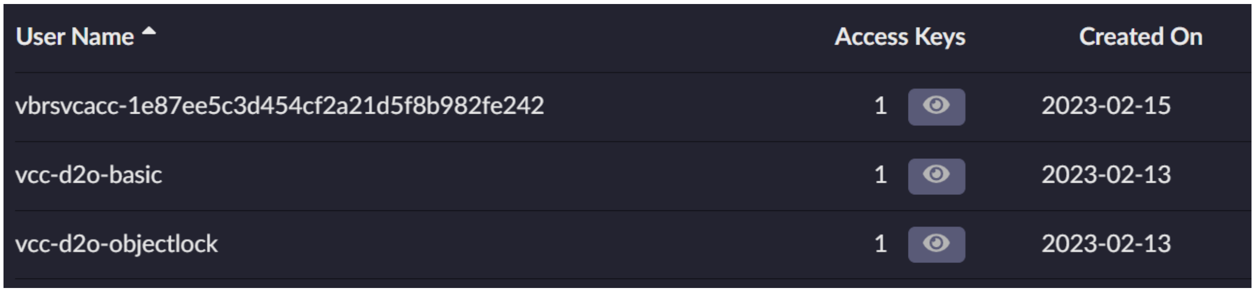
● **Veeam Cloud Connect repository**
Back up to a cloud repository managed by Veeam Cloud Connect service provider.

Enter the Cloud Connect credentials:



Username: vcc-d2o-test
Password: ●●●●●●●●●●

Note here that we are NOT using the Access/Secret keypair. That key is jealously kept inside the Veeam Backup Server and never passed to any remote agent. Instead we connect with the Cloud Connect tenant credentials. As soon as the first backup is started, we can notice one immediate change in the users list of the Object Storage:



User Name ⏶	Access Keys		Created On
vbrsvcacc-1e87ee5c3d454cf2a21d5f8b982fe242	1	👁	2023-02-15
vcc-d2o-basic	1	👁	2023-02-13
vcc-d2o-objectlock	1	👁	2023-02-13

Together with the two “VCC” users we created earlier to mount the two S3 buckets, a third user is now created automatically by the Veeam backup server. This user vbrsvcacc-1e87ee5c3d454cf2a21d5f8b982fe242 is the result of the STS process. This user has its own access key which is the result of a combination of Access Key, Secret Key, and Security Token.

This keypair is then passed to the remote agent to be used to connect to the Object storage.

Here is where **“automatic multi-tenancy”** happens:

- In Cloud Connect, each sub-tenant receives its own STS key
- In Veeam Backup & Replication, each agent receives its own STS key (when job mode is Managed by Agent or Standalone)
- For VM’s backups, all the VM’s are stored under the same account, regardless of whether they are targeting a locally registered Object storage or a remote Cloud Connect server exposing the object storage

When the agent connects to the bucket with this key, it can write its own backups. These backups are stored in a dedicated subfolder and the agent can only read what it wrote. No other agent can see the backup data created by another agent, even when sharing the same bucket.

Azure and Google Cloud

The previous scenario explains how Secure Mode works when using S3 storage. With Azure or Google Cloud object storage, there are some small differences that are worth mentioning.

Let’s look at this table:

	VBR	VBR + Immutability	Cloud Connect	VCC + Immutability
AWS S3/Compatible	✓	✓	✓	✓
Azure Blob	✓	✓	✓	✗
Google Cloud Storage	✓	✗	✓*	✗

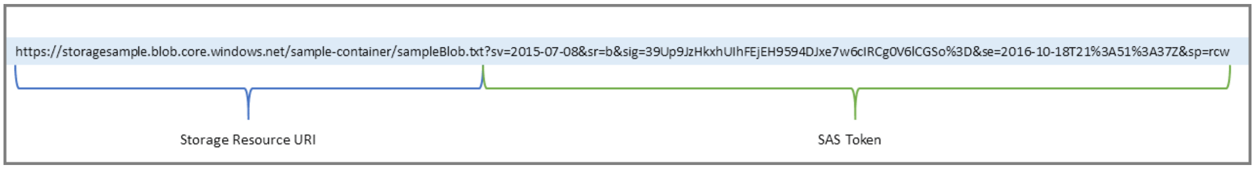
** Cloud Connect with Google storage works in direct mode, but additional credentials must be set in the appliance helper settings.*

End users and Service Providers can consume each of the three main storage platforms. If we also want to leverage Object Storage Immutability we need to shrink the list of available options: at the time of writing Google doesn’t offer this feature yet, but for Cloud Connect also Azure is not an option. As many of the service providers are probably going to leverage on-premises S3-compatible storage solutions, this should not be a real limit for them.

Azure

In Azure there is no creation of users, nor policies or permissions. Here we can grant limited access to Azure Storage resources using **shared access signatures** (SAS: https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview).

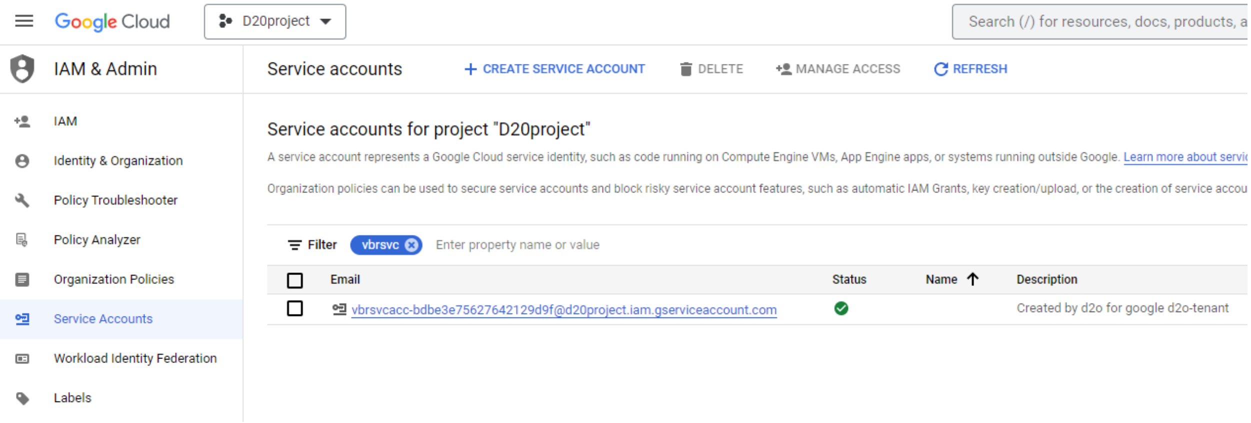
In SAS, everything is organized via private links: every agent gets a unique link to access data:



```
https://storagesample.blob.core.windows.net/sample-container/sampleBlob.txt?sv=2015-07-08&sr=b&sig=39Up9JzHkxhUIhFEjEH9594DJxe7w6cIRCg0V6lCGSo%3D&se=2016-10-18T21%3A51%3A37Z&sp=rcw
```

Google

Google is similar to S3. The Veeam backup server creates users under the IAM service and each user is also tagged as Veeam Account with value %server name%:



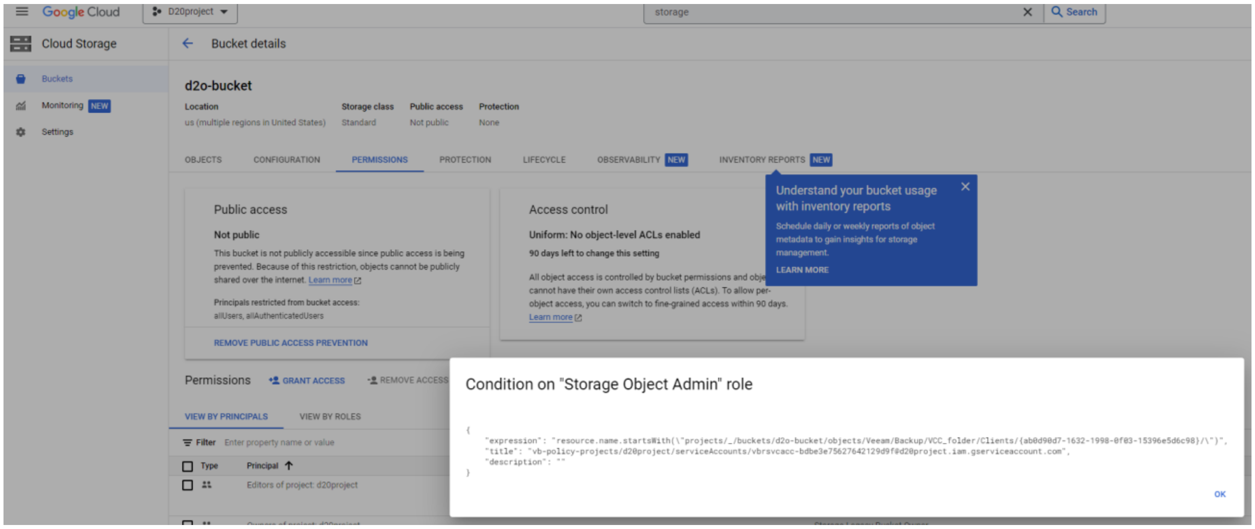
Service accounts for project "D20project"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more about servic

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accou

□	Email	Status	Name ↑	Description
□	vbrsvcacc-bdbe3e75627642129d9f@d20project.iam.gserviceaccount.com	✓		Created by d2o for google d2o-tenant

Then, we assign an access policy to the bucket:



d2o-bucket

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Not public	None

Public access

Not public

This bucket is not publicly accessible since public access is being prevented. Because of this restriction, objects cannot be publicly shared over the internet. Learn more

Principals restricted from bucket access: allUsers, allAuthenticatedUsers

Access control

Uniform: No object-level ACLs enabled

90 days left to change this setting

All object access is controlled by bucket permissions and obje cannot have their own access control lists (ACLs). To allow per-object access, you can switch to fine-grained access within 90 days. Learn more

Understand your bucket usage with inventory reports

Schedule daily or weekly reports of object metadata to gain insights for storage management.

LEARN MORE

□	Type	Principal ↑
□		Editors of project: d20project

Condition on "Storage Object Admin" role

```
{
    "expression": "resource.name.startsWith(\"projects/_/buckets/d2o-bucket/objects/Veeam/Backup/VCC_folder/Clients/{ab0d90d7-1632-1998-0f03-15396e5d6c98}/\")",
    "title": "vb-policy-projects/d20project/serviceAccounts/vbrsvcacc-bdbe3e75627642129d9f@d20project.iam.gserviceaccount.com",
    "description": ""
}
```

Unlike AWS, Google has different policy schemas. In Google the rights to access and consume storage are granted by bucket policy, not by user policy. Because of that there are a few limitations:

1. Veeam needs to work with policy after user creation: first the software creates a lot of users, then the policy will be modified for each user consequentially.

2. Google has a limit for policy size. We can find the Limits in the documentation. An additional problem is that - despite documentation states that limit is 1500 appearances in the policy - we have observed problems with policy size before we reach that limit. The explanation from the documentation is: *If you use IAM Conditions, or if you grant roles to many principals with unusually long identifiers, then IAM might allow fewer principals in the allow policy.*

For these reasons, we use 2 roles to grant access to resources:

Storage Object Viewer role that grants these rights:

- resourcemanager.projects.get
- resourcemanager.projects.list
- storage.objects.get
- storage.objects.list

Storage Object Admin role that grants these rights:

- orgpolicy.policy.get
- resourcemanager.projects.get
- resourcemanager.projects.list
- storage.objects.*
- storage.multipartUploads.*

Then, to limit rights to specific folders we use conditions.

Policy Example:

Condition for Storage Object Viewer. This expression grants read-only access to the config folder and grants rights to list buckets (without it - the bucket cannot be accessible at all).

```
{
    "expression": "resource.name.startsWith(\"projects/_/buckets/d2o-
bucket/objects/Veeam/Backup/VCC_folder/Clients/{ab0d90d7-1632-1998-0f03-
15396e5d6c98}/\")",
    "title": "vb-policy-projects/d20project/serviceAccounts/vbrsvcacc-
bdbe3e75627642129d9f@d20project.iam.gserviceaccount.com",
    "description": ""
}
```

Condition for Storage Object Admin. This expression grants full access to the client folder

```
{
    "expression": "resource.type == \"storage.googleapis.com/Bucket\" ||
resource.name.startsWith(\"projects/_/buckets/d2o-
bucket/objects/Veeam/Backup/VCC_folder/Config/\")",
    "title": "vb-VCC_folder-config-policy",
    "description": ""
}
```

NOTE: for Google object storage it's mandatory to configure the Appliance Helper. Without Appliance Helper customers won't be able to use Google Cloud Storage for agent jobs. Read more in the User Guide: https://helpcenter.veeam.com/docs/backup/cloud/cc_object_storage.html

Things to know

Any new technology must be adopted with attention. For this reason, it's important to learn and take into consideration some details that can influence the design and operations of an environment leveraging Secure Mode.

Compatibility
Which sources can consume Secure Mode?

Veeam Backup & Replication 12

Veeam Agent for Microsoft Windows 6.0

Veeam Agent for Linux 6.0

Veeam Agent for Mac 2.0

Cloud Connect Insider Protection
It doesn't work with object storage assigned as a cloud repository (https://helpcenter.veeam.com/archive/backup/120/cloud/cc_object_storage.html). We suggest in this case to use immutability, which can give the same result with some additional benefits: files are always visible to end users, and so restores can be started immediately, without the need to move deleted files in the original location.

Token duration

In case of AWS/Google and S3Compatible – VBR Server creates users with policies that allow agent to use only their own folder.

You may find in official AWS documentation (https://docs.aws.amazon.com/STS/latest/APIReference/API_GetSessionToken.html) that STS lifespan can range from 900 seconds (15 minutes) up to a maximum of 129,600 seconds (36 hours), with a default of 43,200 seconds (12 hours). In the Veeam Cloud Connect use case, Veeam DOES NOT set any expiration date. Cloud Connect server is responsible for the creation and refresh of credentials for each agent.

For Azure – VBR server creates a couple of SAS links. Each SAS link is valid for 30 days. When agents connect to VBR/VCC and there are 15 days till expiration or less – VBR recreates new links.

Key protection in the agents
For Cloud Connect: Agents DO NOT store STS keys, they receive them from VCC server every time a session is initiated. Agent stores only tenant/subtenant credentials.

For VBR server: agents can backup to Object Storage without connection to VBR – so, they need credentials. In this case each agent receives limited credentials and stores them.

Availability
VCC/VBR server is an absolutely needed component of the solution. Since STS keys are retrieved at each session, no operation can happen if the server is unreachable by agents, even if the Object storage is online.

Conclusions

We hope that this document has given a good and clear understanding of Veeam’s usage with Object Storage and the Secure Mode option, the scenarios in which it can be a good fit, and the huge potential that this design has specifically in distributed environments.

With the “new normal” of people working from remote locations, and thus the exponential increase of data stored in personal devices, we believe that a solution allowing multiple remote agents to consume a backup storage solution directly accessible via Internet, but without compromising on security, is a key point. Secure Mode allows Veeam admins to create such a scenario in an easy yet powerful way.

If you are in the phase of replacing your Veeam backup storage or are looking at key advantages to switch to Veeam, think about Secure Mode.