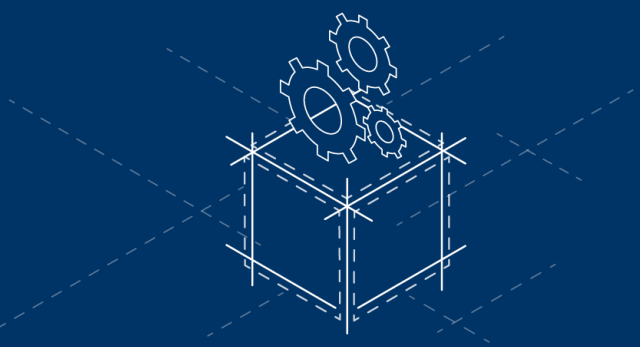
Cloudockit

INSTALLATION & CONFIGURATION GUIDE



GENERATED BY: Cloudockit

Contents

Introduction

Cloudockit Desktop is a tool that allows local document creation.

The document creation can be triggered manually or scheduled.

Cloudockit Desktop code is the same as Cloudockit Website, giving you the exact same output.

Pre-Requisites

Here is the list of Pre-Requisites to execute Cloudockit Desktop:

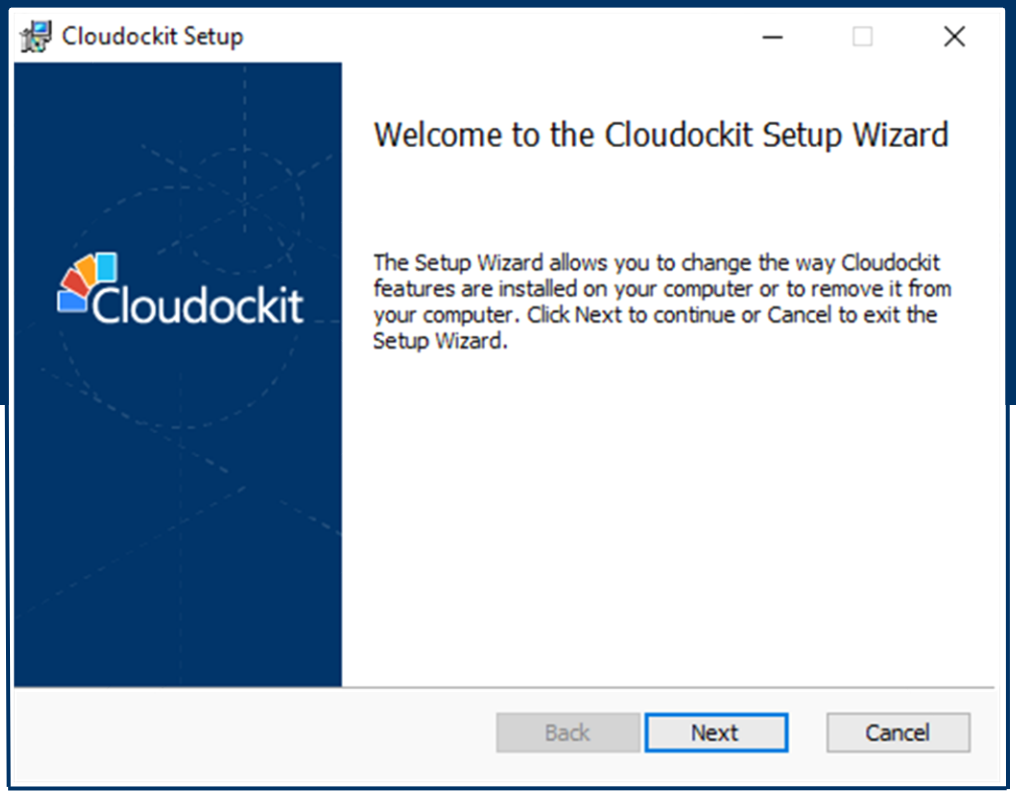
- Windows OS is required, Linux/MacOS are not supported
- Minimum Spec is 4 CPU and 12 GB of RAM. If you want to speed up the report generation, you can go higher than that
- Minimum resolution required: 1280 x 960
- Local Administrative Privileges is required to install and run the Cloudockit application
- Firewall must be open to communicate with
    - During the document generation process, ensure that the Azure/AWS/GCP API can be called. As an example, here are the endpoints used to retrieve Azure information (Public Cloud):
        - https://graph.windows.net/
        - https://login.microsoftonline.com/
        - https://management.azure.com
        - https://management.core.windows.net/
    - *(Optional)* https://generate.cloudockit.com on port 443. This communication is used for license validation purpose, the data retrieval and document generation are done locally. Cloudockit now offers Offline Activation mode allowing you to activate the license manually if you do not open this communication channel
    - *(Optional)* sengrid.net (used for email delivery if you specify an email)
    - *(Optional)* If you plan to create custom Compliance Rules or custom Tailored Diagrams, you need to open communication to https://generatebeta.cloudockit.com/CDKExternalControls as Cloudockit Desktop displays a web control for those components creation. You can also do it offline with Compliance Rules and Tailored Diagrams import feature.

Installation

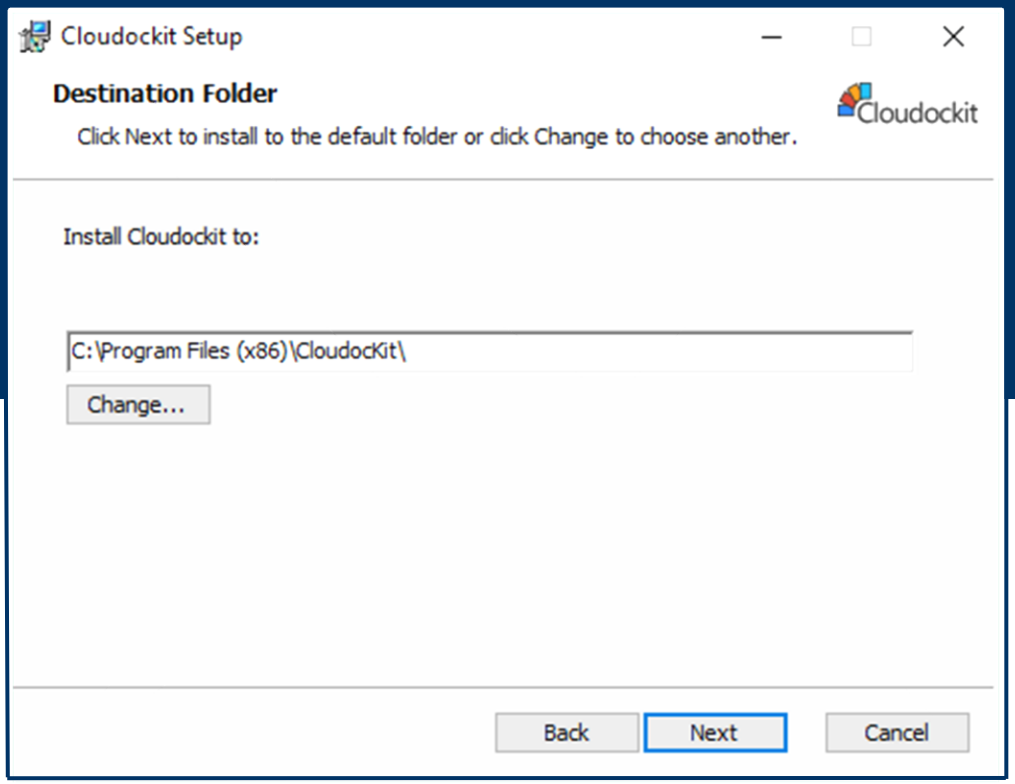
First, you need to install the tool. The tool is a simple *.msi* file.

Follow these steps:

1. Double click on Cloudockit.msi
2. Click on Next



3. Accept the Agreement
4. Select the installation Path

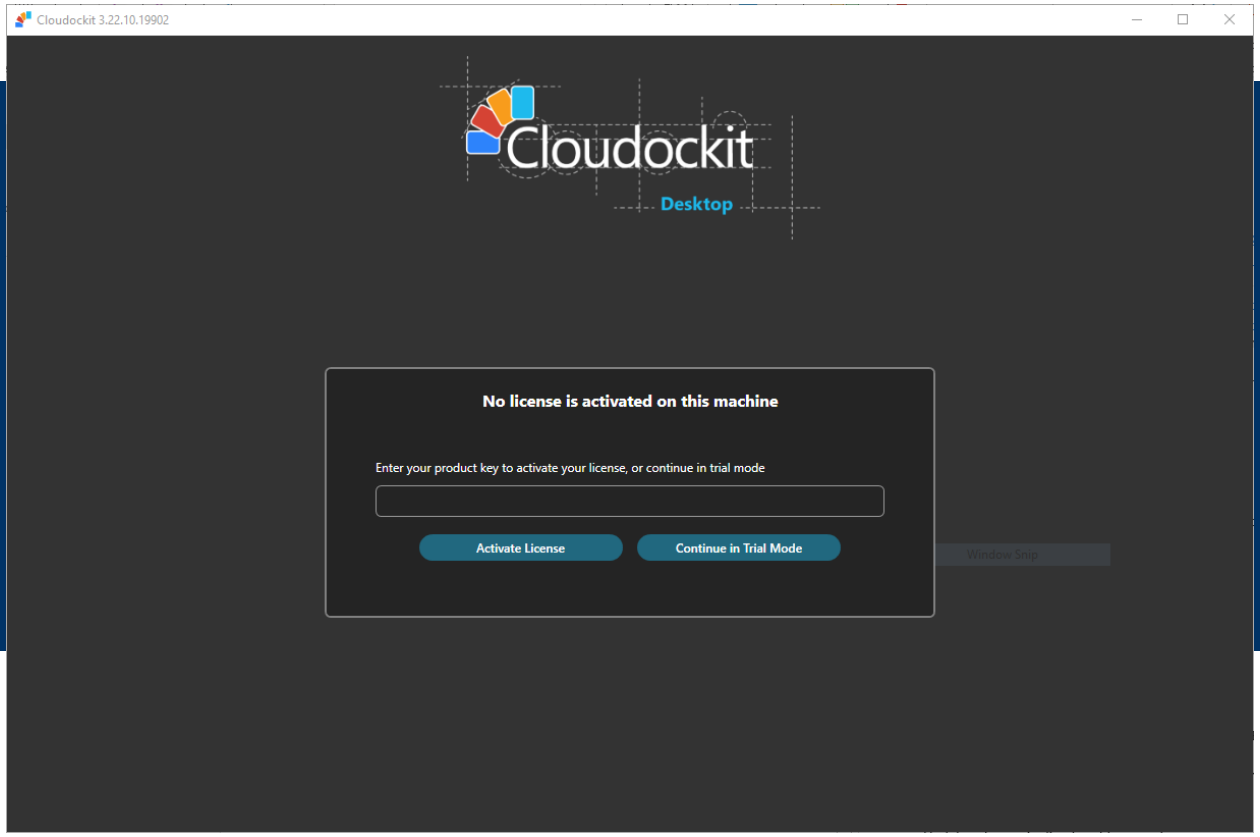


5. Click on Install
6. Click Finish
7. A shortcut has been generated on your desktop. Double click on it to start Cloudockit.

Please note that Cloudockit Desktop has an auto-update feature: it will automatically detect a new update and install it for you.

License Activation

Once you have installed Cloudockit tool, enter the product Key to activate the product.



Enter your product key to activate you license.

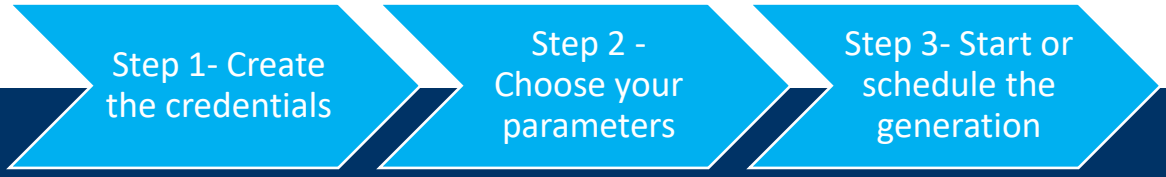
If you do not have one, simply select trial mode and enter your email. Please enter a valid email: we will send you a validation email with a link for the activation. Then restart Cloudockit Desktop and you will be in Trial mode.

Please note that in Trial mode you will have ** in the generated documents at random positions.**

If you DO NOT Receive the validation email, it is probably due to spam filtering issues, please contact us at support@cloudockit.com and we will fix that for you.

Start or Schedule a Document Generation

This is a 3 step process



Let's have a look at each of these steps.

Step 1 - Create the credentials

First step is to create the credentials that will be used to connect to your cloud environment.

These steps depend on the cloud provider. Please refer to the appropriate section.

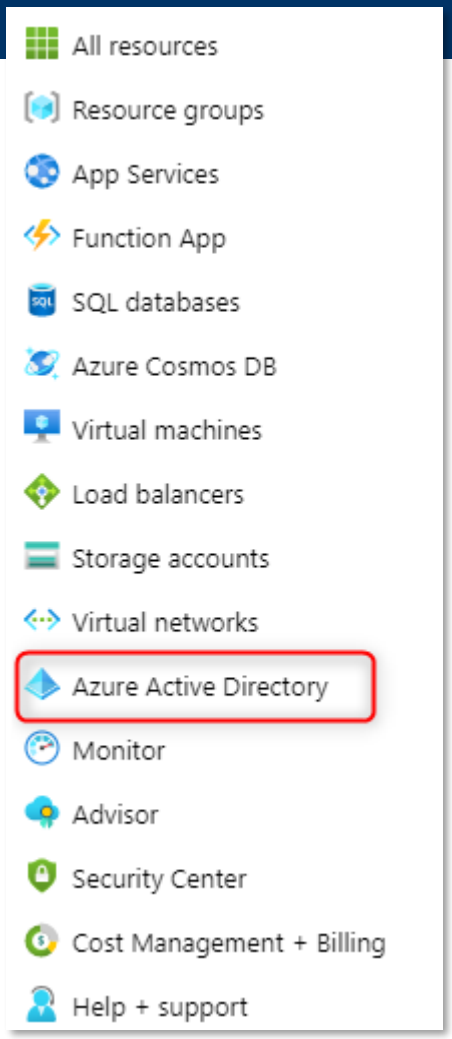
Azure – Create an AAD App

To allow Cloudockit to access the information from your subscription, create the credentials that CloudocKit.exe will use to scan your subscriptions.

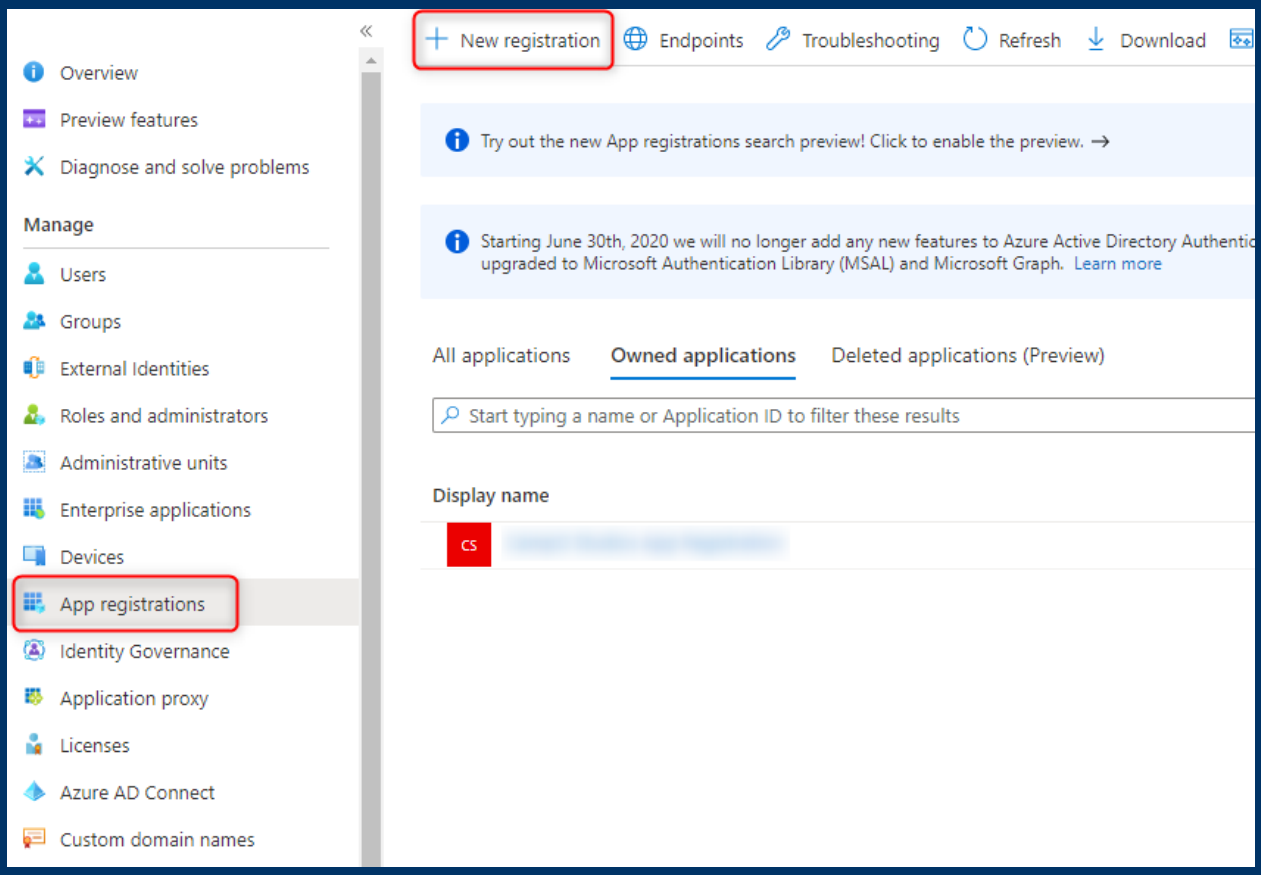
As this is a scheduled process, an Azure Active Directory Application (also known as Service Principal) is required. You cannot automate the document generation process with user credentials.

So, let's start by creating this AAD App.

Navigate to https://portal.azure.com and select the Azure Active Directory blade:

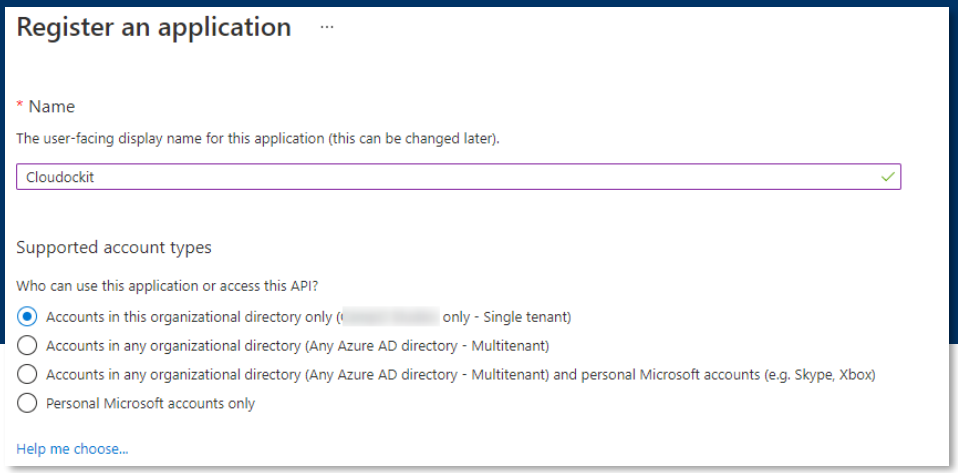


Click on App Registration:



Then, click on New Registration

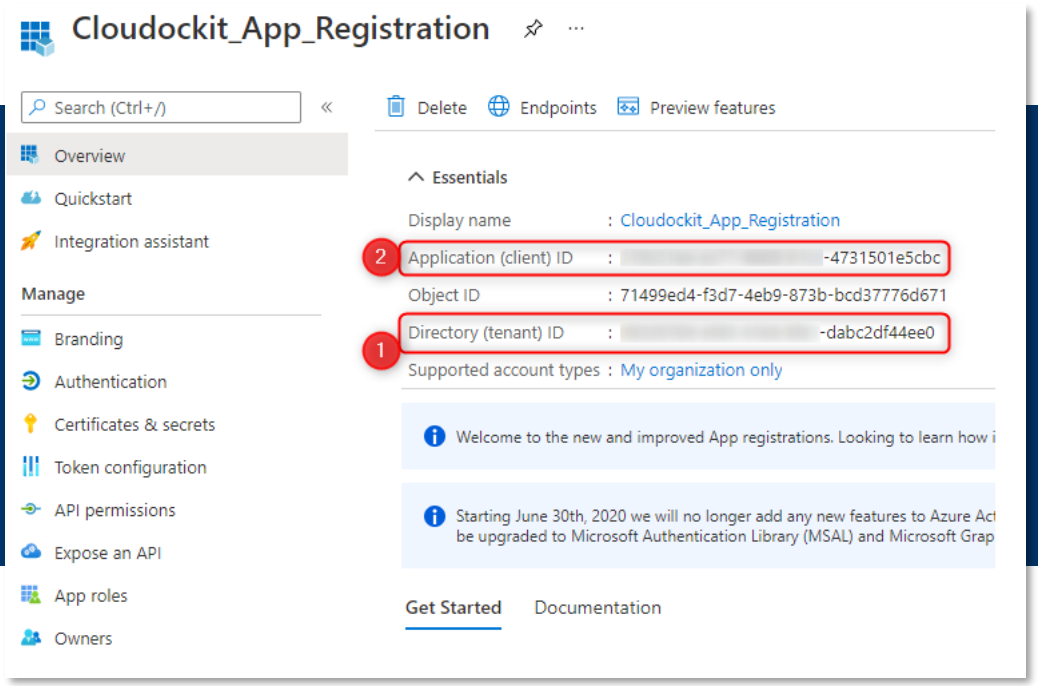
and enter the following settings:



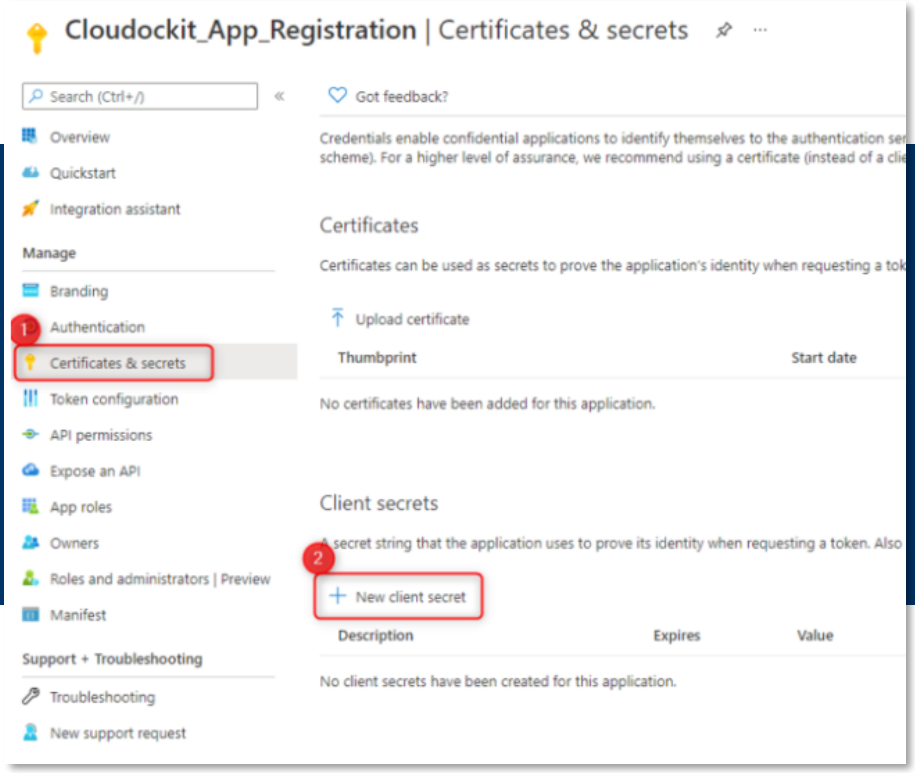
Name: Unique name for your app registration

Supported account types:
Select Accounts in this organizational directory only

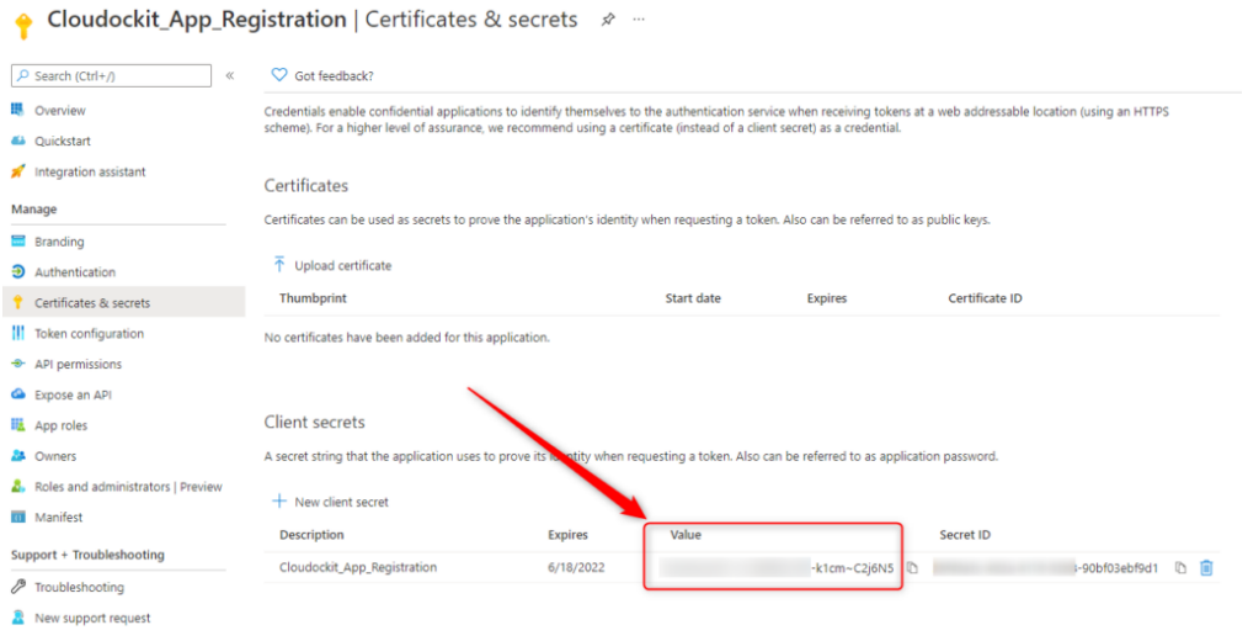
Once the application has been created, click on it, and take note of the Tenant ID and the Application ID. They are required when scheduling a document generation.



Then click on Certificates and secrets and create a new client secret that never expires:

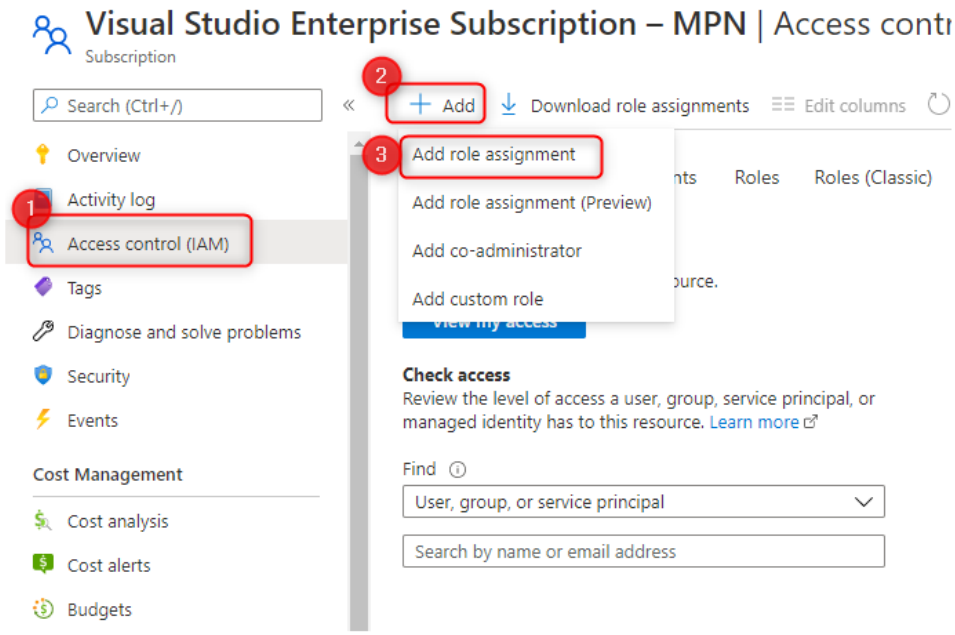


Copy and paste the value attribute. Keep it with the Directory ID and Application ID.



Note: This value is only visible after its creation. You will have to create a new one if you did not saved it.

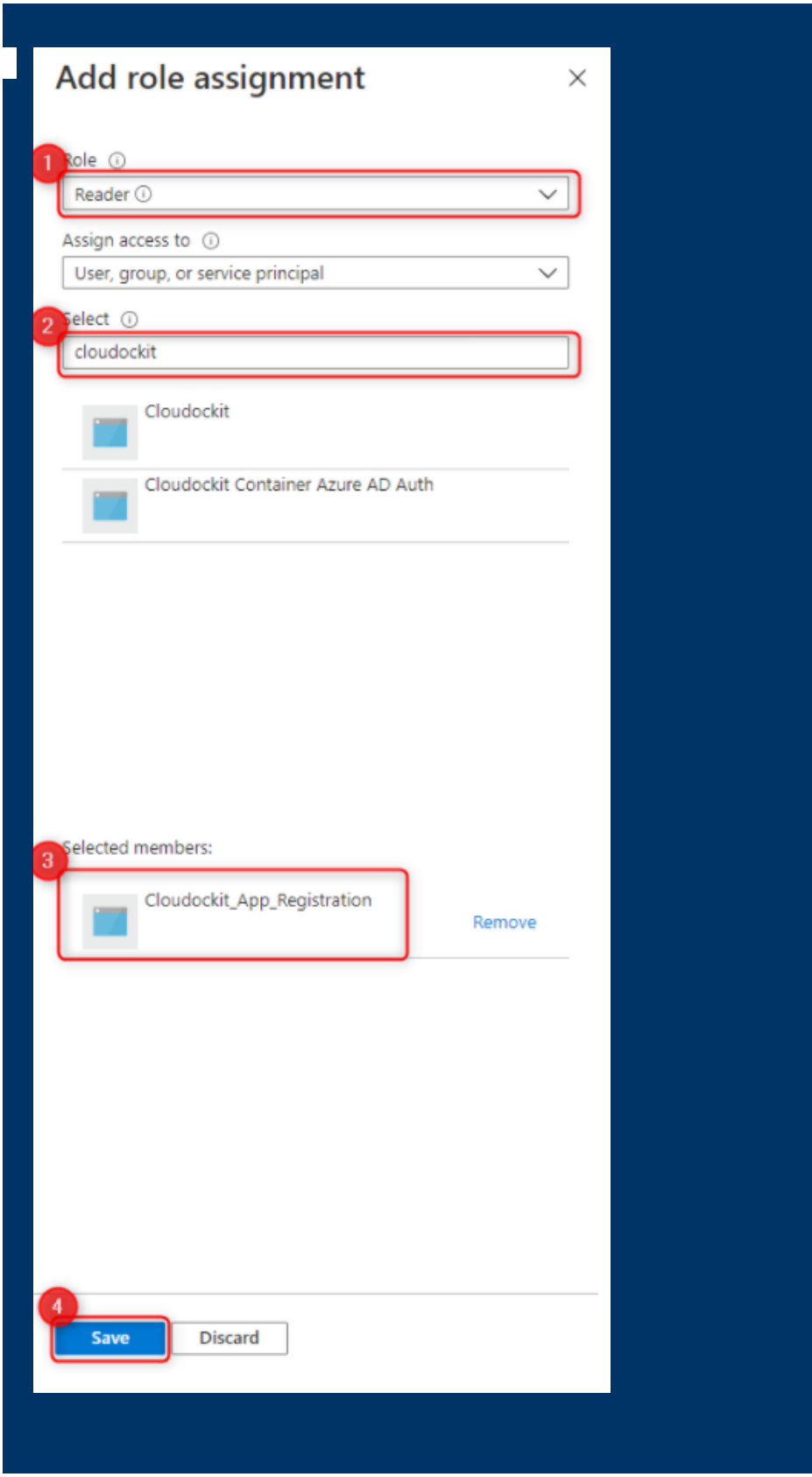
Select the appropriate subscription, select Access Control (IAM), click Add and Add role assignment.



Fill in the following fields:

Role: Reader

Select: Enter the name of the app registration. Select it from the list and add it to



AWS

You have two options to connect to your AWS Accounts depending on the scenario:

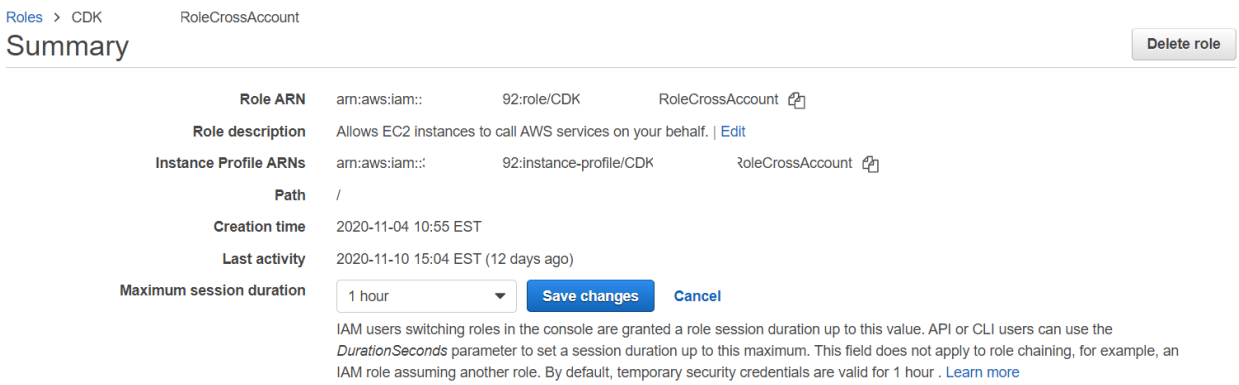
- **Option 1 -** If you want to connect to a specific **single AWS Account**, create Access Key and Secret Key with read permissions
- **Option 2 -** If you want to connect to a **multiple AWS Accounts**, create Access Key and Secret Key with permissions to list the account and roles in each account you want to scan

Option 1 - Single AWS Account - Create AWS Access Key and Secret Key
Option 2 - Multiple AWS Accounts - Create AWS Access/Secret Key and Cross Account Roles

Important note: to support scanning multiple AWS Accounts, Cloudockit Desktop MUST be installed in an EC2 instance.

To scan multiple AWS Accounts, we need to have the following components:

- An IAM User and its keys to list the Account in the OU organization
    - Note that you can also manually enter a list of Account IDs if you do not want to retrieve it automatically with an IAM User
    - This user is referred as ***CloudockitMultiAccountScan*** in the following sections.
- An AWS Cross-Account Role in each AWS Account that need to be scanned. The name of this Cross-Account Role needs to be the same in all AWS Account.
    - This role is referred as ***CloudockitScanRole*** in the following sections.
- An EC2 instance that has the privilege to AssumeRole to allow Cloudockit Desktop to Assume Roles defined in the AWS Accounts
    - This role is referred as ***CloudockitEC2RoleCrossAccount*** in the following steps
- The property “Maximum session duration” of this role should be set to 12h, in each of the AWS accounts to scan



- In the CloudockitDesktopConfiguration.json file located in CloudocKit\CloudockitConfiguration, please set the property AWSTokenDuration to be equal less or than Maximum session duration property set previously. AWSTokenDuration values between 1 and 12 – default is 12h.

Here are more details on how to create those components:

IAM User

This IAM User will be used to list all the accounts in the organization to allow Cloudockit Desktop to loop trough all these accounts and then Assume Roles in each of them.

To Create this IAM User, follow these steps:

- Login to IAM Console
- Click on User and then Add User
- In the Username field, enter ***CloudockitMultiAccountScan*** (or any name that follows your naming convention)
- In access type, select Programmatic Access
- Click Next, Permissions
- Click on Attach existing policies directly
- Click on Create Policy, this will open a new tab to create a new policy
    - In Service, select *Organizations*
    - In action, select *Access Level* / *ListAccounts* and *ListAccountsForParent*
    - In Resources, select *All resources*
- Click Review Policy
- In name enter *CloudockitMultiAccountScanPolicy* and click on Create Policy
- Close the tab that has been open to create the new policy
- Comme back to the user and refresh your browser
- In the search box, enter *CloudockitMultiAccountScanPolicy*, click the checkbox to select it and then click on *Next:Tags*, then *Next:Review* and finally click on *Create User*

Once the role is created, get the Access Key / Secret Key.

Select Users and choose *CloudockitMultiAccountScan.* Click on Security Credentials and then click on Create Access Key. Save the Access Key ID and Secret Access Key.

EC2 instance

Before creating the EC2 Instance, let's create the Policies and Role required for the EC2 Instance.

Follow these steps:

- Create a Policy named *CloudockitAssumePolicy* (or any convention you have)
  - Click on Policies
  - Click on Create Policy
  - Click on JSON and enter the following JSON Definition:

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Resource": "*"
    }
}
```

  - Click on Review Policy
  - In the name, enter *CloudockitAssumePolicy*

- Create a role named ***CloudockitEC2RoleCrossAccount*** (or any convention you have)
  - Click on Roles
  - Click on Create role
  - In Select type of trusted entity, select AWS Service
  - In Choose the service that will use this role, select EC2
  - Click on Permissions
  - Attach the two following policies to this Role:
    - *ReadOnlyAccess*
    - *CloudockitAssumePolicy*
  - Click on Tags and then Review and then enter *CloudockitEC2RoleCrossAccount* in the name and click Create Role

Now, we are ready to create the EC2 Instance where you will install Cloudockit Desktop. This EC2 instance will assume all the Roles in all the AWS Account.

Follow these steps to create the EC2 Instance:

- From the EC2 Console, click on Launch Instance. Select Windows Server 2019 Base, take the 64-bit edition for the Image
- In the step2, Choose an instance type, select t2.large. Click Next for the instance details
- In IAM Role, select the role *CloudockitEC2RoleCrossAccount* that you have created before
- Then click Review and Launch
- Login to the EC2 Instance and install Cloudockit Desktop

AWS Cross-Account Roles

In each AWS Account you want to scan, create a role named ***CloudockitScanRole*** (or any name that you prefer).

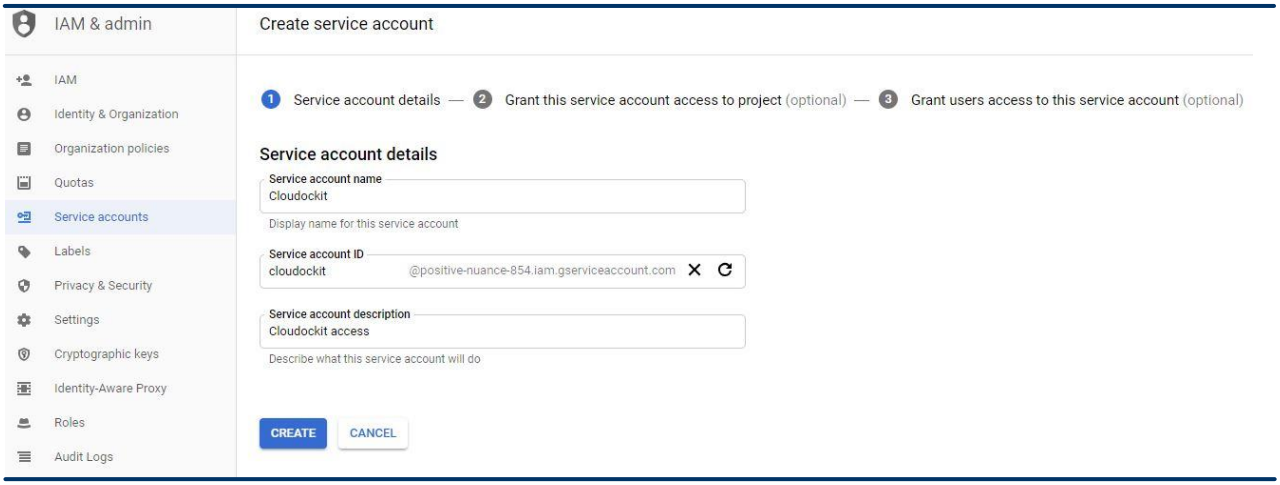
Here are the steps to create this role:

- From IAM console, click on Roles and then Create role
- Select Another AWS Account. Enter the Account ID where you are installing the EC2 instance that will run Cloudockit Desktop
- Then, click Next and select the *ReadOnlyAccess* permission
- Click Review
- Enter the name: CloudockitScanRole
- Click on create role
- Repeat the steps for all AWS Accounts

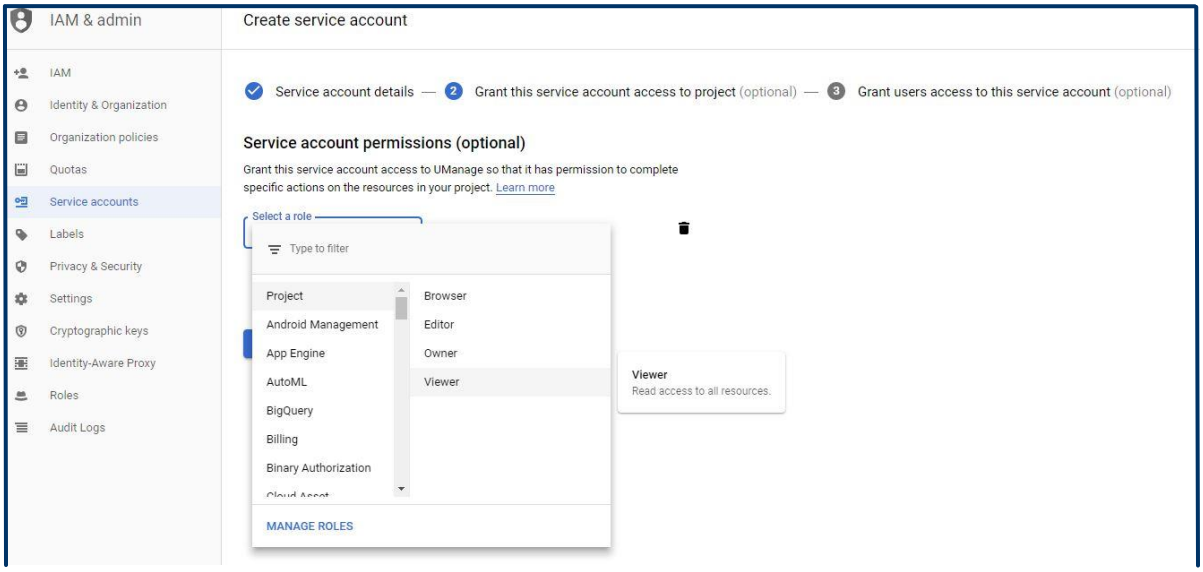
GCP – Create JSON Credential files

Step 1 – **Create a Service Account**

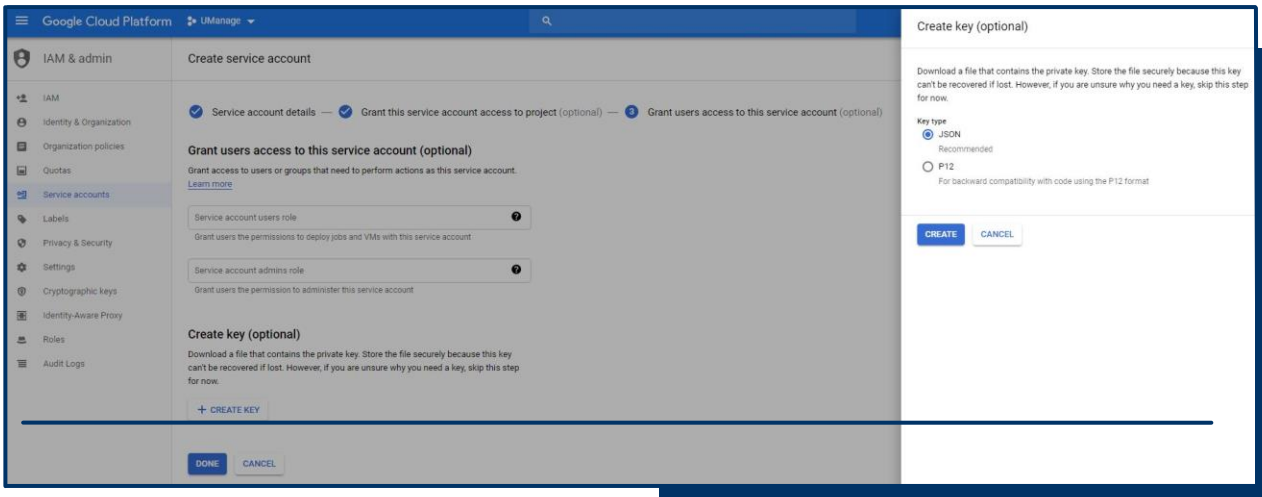
- Sign in to the GCP Console and click on IAM & Admin / Service Accounts: https://console.cloud.google.com/iam-admin/serviceaccounts
- Select the project where you want to create the Service Account (you will then be able to give the appropriate permissions to the other projects with the same service account)
- Click on Create Service Account and enter the Service Account Name. (For instance, use Cloudockit). Then click on create



- Then, click on create and select the role Project / Viewer. Click on Continue



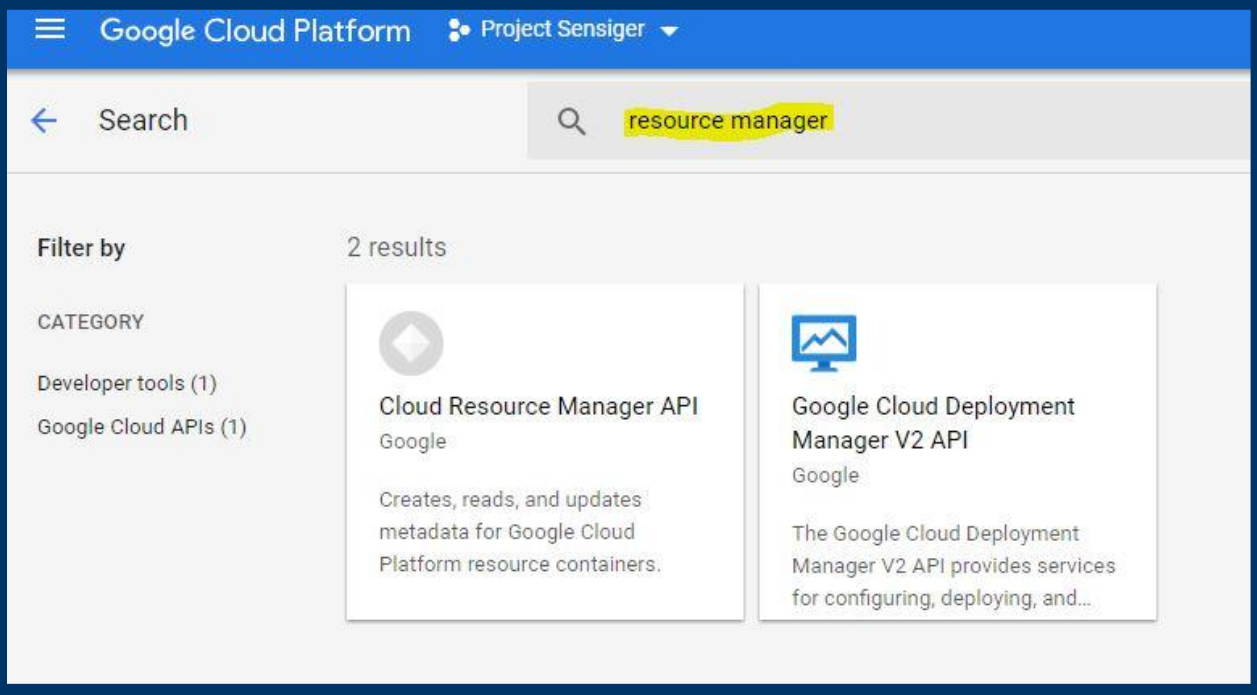
- Click on the Create Key button, select JSON and click Create. Save the file locally



- To save the service account, click on Done

Step 2 – Activate the appropriate API

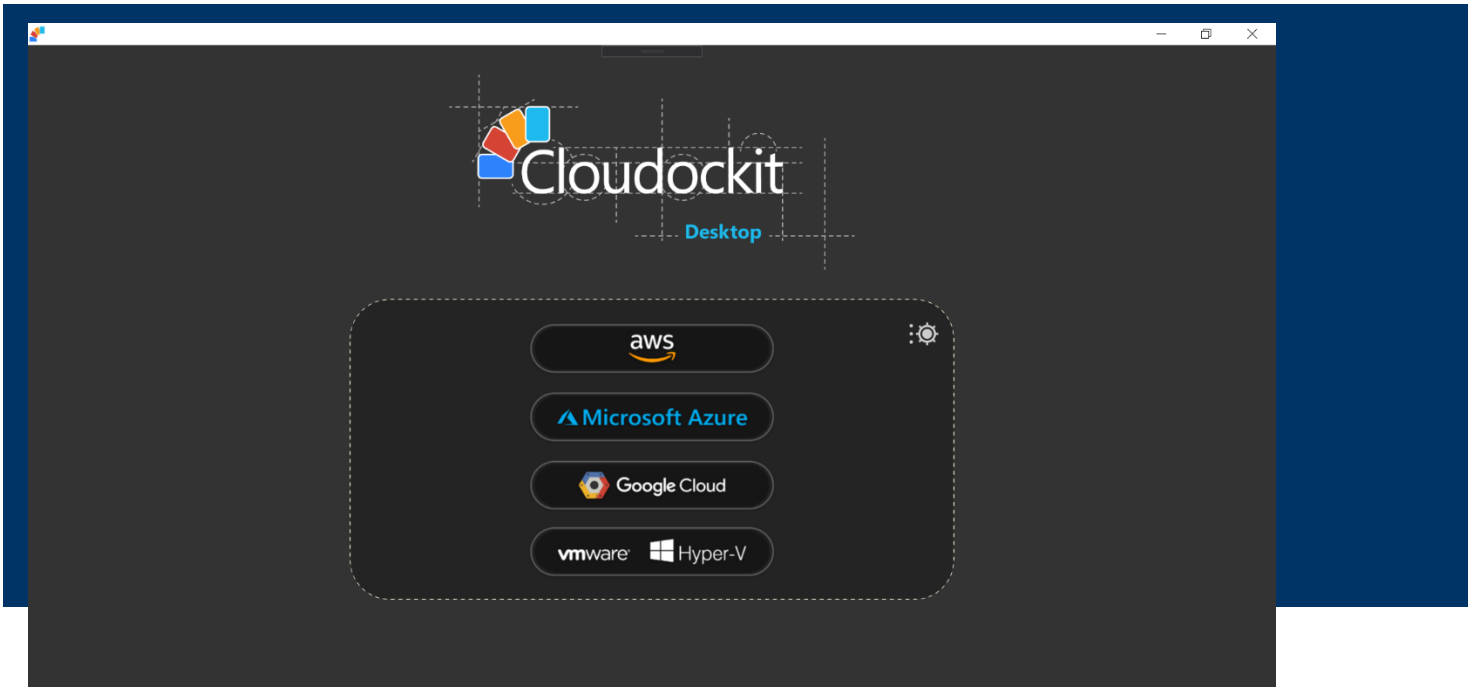
- As Cloudockit is using the Cloud Resource Manager API to list all the projects, you need to Enable this API
- To do so, click on API & Services and click on Enable APIs and Services
- In the search box, enter Resource Manager



- Then click on Cloud Resource Manager API and click ENABLE.

Step 2 – Choose your platform and your parameters

Choose your platform



Depending on the Platform you select, you will see different authentication option:

Azure

AAD Application

Proceed by clicking on AAD Application, obtained from step 1.

Enter your information in the list and don't forget to choose your cloud type depending on your location. Complete this form by clicking on "Login".

Afterwards, you will be able to choose your environment.

User

This option uses Code authentication. Firstly, enter your AAD Tenant Name, then Cloudockit will give you a Code to enter after navigating to https://microsoft.com/devicelogin.

Please note that with this option you cannot schedule a document generation. It is an interactive process therefore it cannot be used for automation.

Managed Identity Authentication (Preview)

This login option allows you to Log into Azure using credentials extracted automatically from the Azure environment from which you have created a Virtual Machine that will contain Cloudockit Desktop.

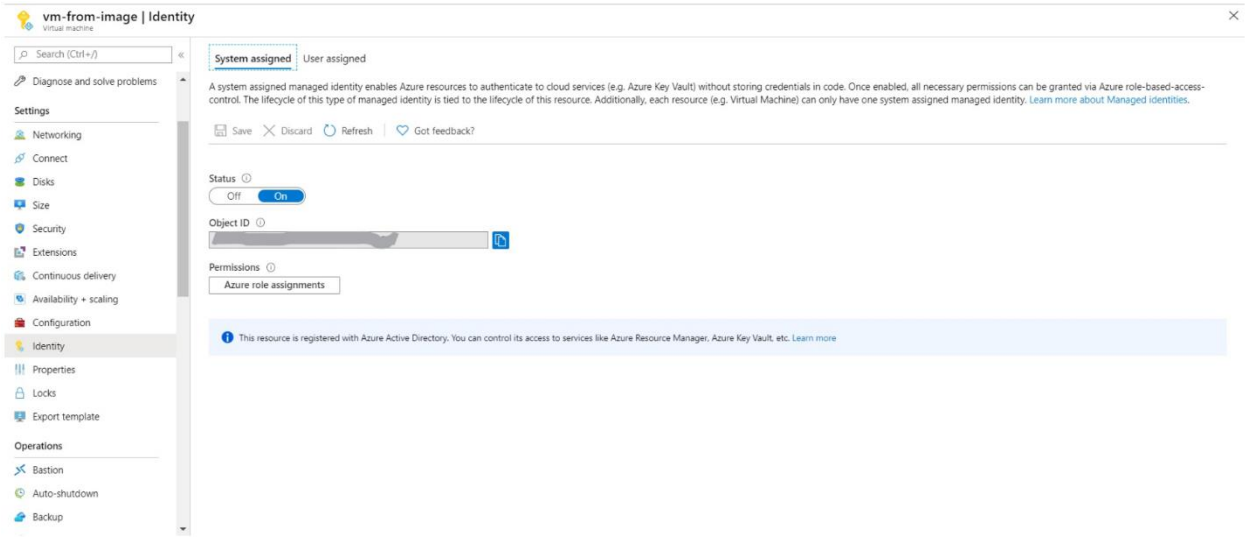
This way you won't have to manually enter or save your credentials in Cloudockit. Once you are connected to your Virtual Machine, it's all managed in the background by Azure.

To learn more about Azure Managed Identity, visit the link below:

https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/overview

This option is only functional if you completed the following steps:

1- Create a Virtual Machine in the Azure Directory you wish to scan (or use an existing one)
2- In Azure Portal, select the Virtual Machine and go to the "Identity" Tab under "Settings"
3- Under "System Assigned" Enable the option by clicking on Status -> ON, then Save



4- Go to the page of the subscription you want to scan, click on Access Control (IAM) Tab, then add “Reader” Role to the Virtual Machine. You can add multiple subscriptions to be scanned by assigning the “Reader” Role to your Virtual Machine from the different subscriptions.

Once these steps are completed, you can connect to your Virtual Machine and install Cloudockit (or continue using it if it’s already installed and has the Managed Identity option displayed).

The Managed Identity login option should be functional now.

You can add or revoke the “Reader” Role whenever you want from the Azure portal. This will add or remove the corresponding subscriptions to be scanned in Cloudockit.

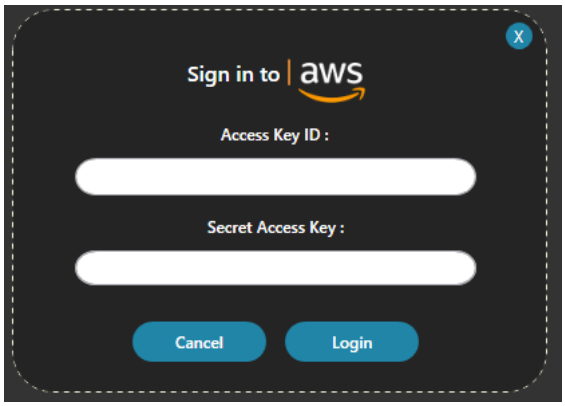
AWS

To authenticate to AWS, choose one of the following two options:

- Access Keys (single account)
- Cross-Account Role (multiple accounts)

Access Keys

Proceed by clicking “Use Access Key”.



There, you will be asked to enter your access key ID and your secret access key obtained from step 1. Continue by clicking Login.

Afterwards, you will be able to choose your environment.

Use Cross-Account Role

Proceed by clicking “Use Cross-Account Role”. Now you have two options

Option a – Accounts automatic discovery

If you have created an IAM User (referred as ***CloudockitMultiAccountScan*** above), then you need to enter its Access Key ID and Secret Access Key.

You also need to enter the Role To Assume in the accounts while doing the scan (referred as CloudockitScanRole above):

Option b – Manual entering of the Accounts

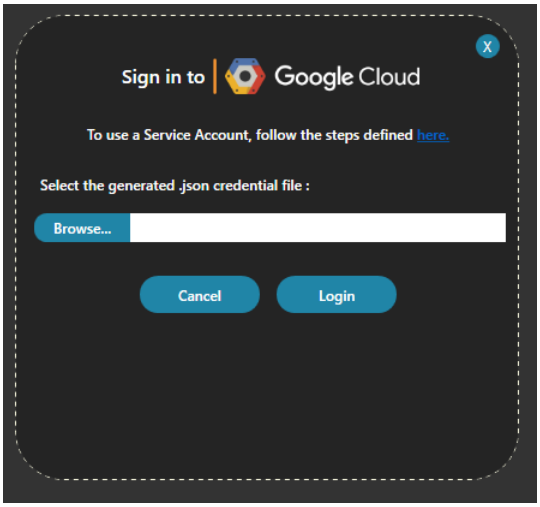
If you do not want to use an IAM User to dynamically list the account, you can enter the account list manually.

You also need to enter the Role To Assume in the accounts while doing the scan (referred as CloudockitScanRole above):

Continue by clicking Login.

Afterwards, you will be able to choose your environment.

GCP



Proceed by clicking “Use Service Account”

There, you will be asked to select your *.json* credential file obtained from step 1. Browse through your folders to find the required file.

Afterwards, you will be able to choose your environment.

Select your environments to document

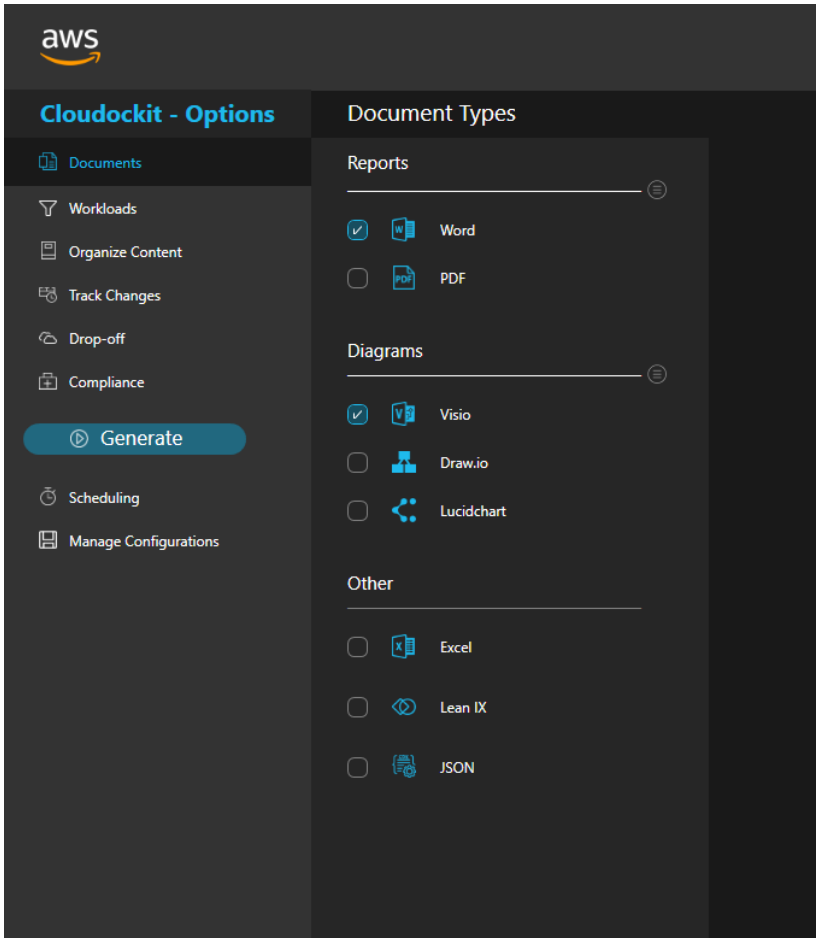
Once authenticated, you can select the environment you want to document:

Parameters selected from Cloudockit Desktop

Once you selected your environment, you can choose the settings for the document generation.

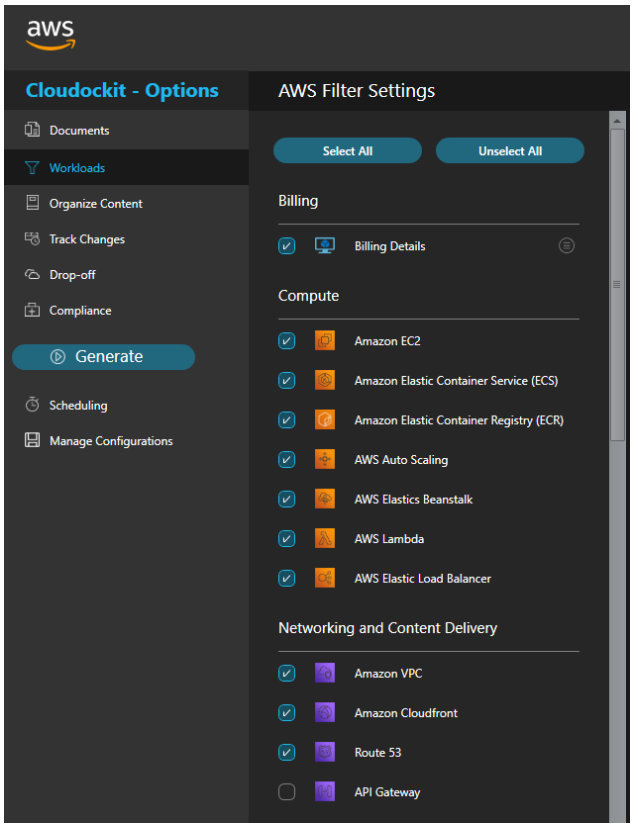
Here are the sections available:

Documents section



In this section, you can choose the type of document you want to generate. You have the choice between Word, PDF, Audit, Excel, Visio, Draw.io, Lucidchart, LeanIX, JSON …

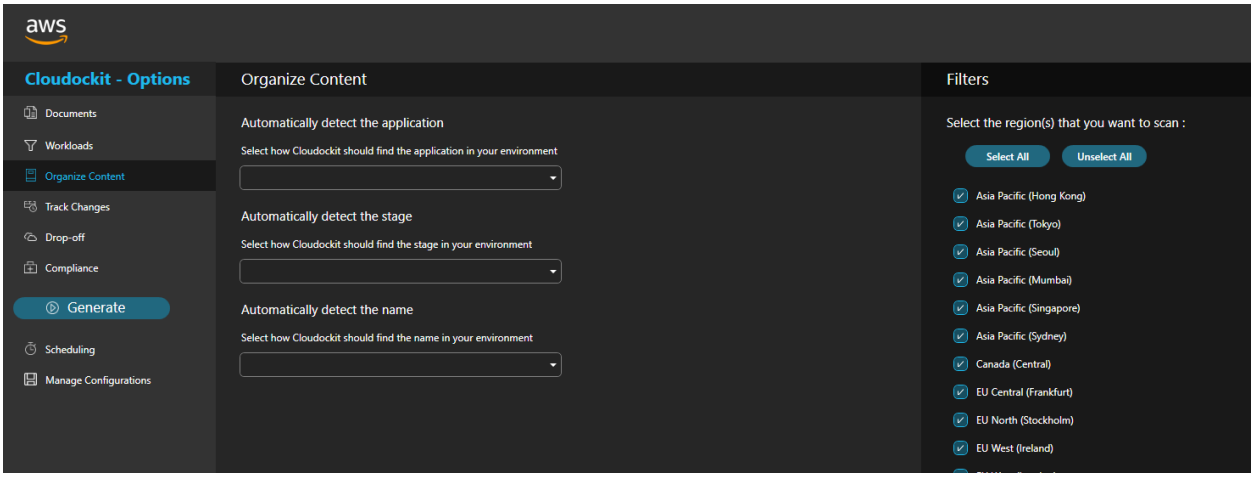
Workloads



This section allows you to choose between the different applications you would like to see linked to the rest of your cloud platform. Depending on your platform, you will have different workloads to choose from.

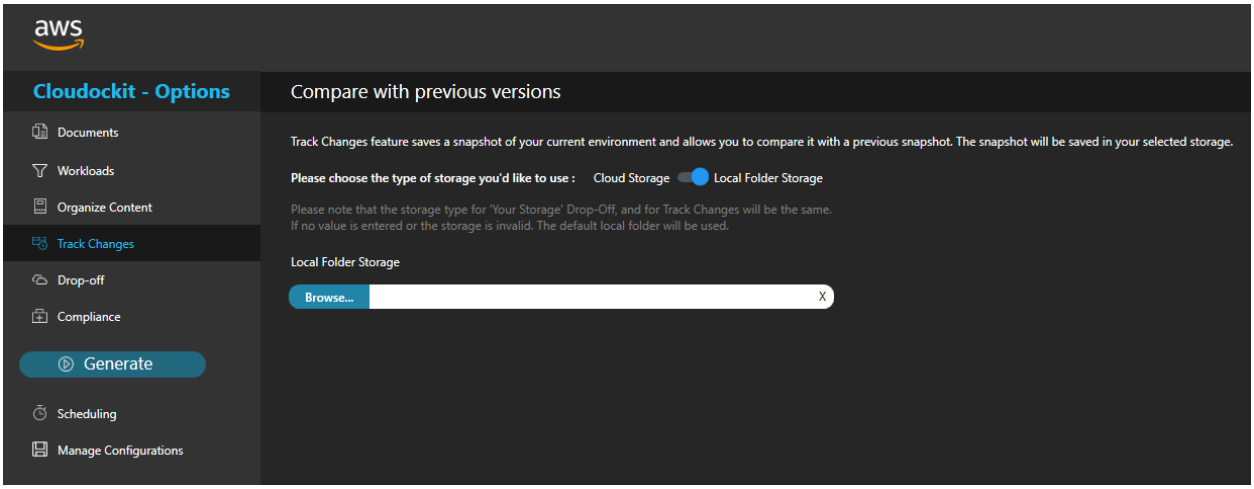
Note: Some workloads will contain additional settings that you may modify.

Organize Content



This section gives you the ability to filter what you want to scan and to automatically detect Application, Stage or Name. By doing so, you specify for example, a Tag that represent your Business Application and Cloudockit will automatically create diagrams per business application.

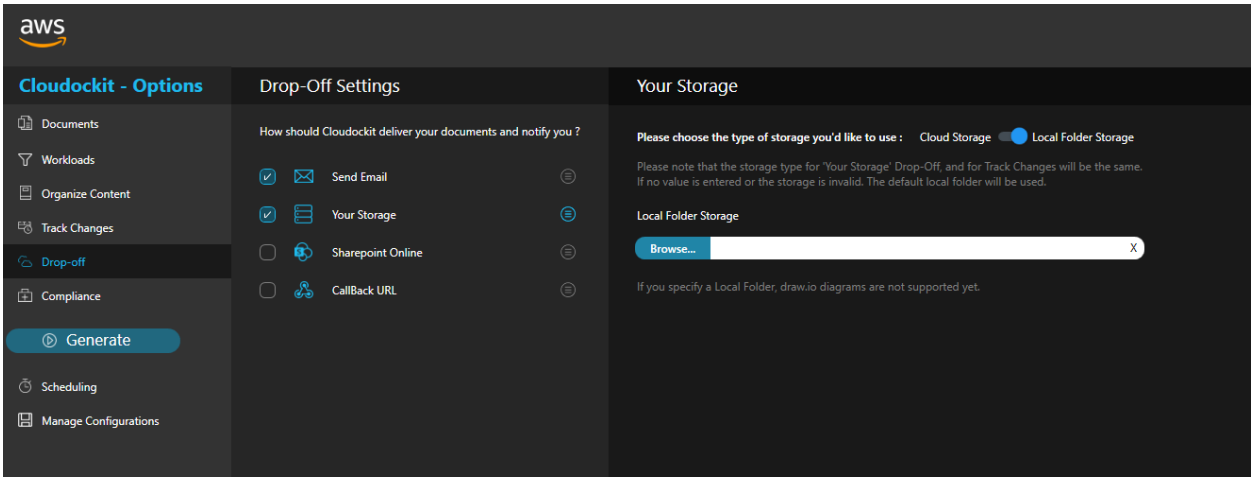
Track Changes



Track changes allows you to see the differences between two different environments. It will save a snapshot and compare it to a previous one. This will give you a better understanding of the changes that happened.

Note that you can choose to use a Cloud Storage or a Local Folder.

Drop-Off



The Drop-Off section allows you to send the generated documents to different sources. You can choose between:

- Email
- Your storage
    - Specify if you want to use Local Folder Storage (where Cloudockit is installed) or a Cloud Storage (your own Cloud Storage)

    - Cf. the section bellow for more details about Cloud Storage setup
- SharePoint
- Callback URL

Cloud Storage Setup

Here are the different options and required configuration for the cloud providers

AWS S3 Bucket

There are two options for the S3 bucket you specify:

- Option 1: **Bucket short name**
    - Specify the short name of the bucket like ***myCloudockitDropOffBucket***
    - This option can be used only if you scan one Account and the bucket is in the Account being scanned. Otherwise, you need to use option 2
- Option 2: **Bucket Arn**
    - Specify the Arn of your S3 bucket like this:
        - ***arn:aws:s3::349224196492:myCloudockitDropOffBucket***
        - The important part is the AccountID and Bucket name that are used internally by Cloudockit to locate the Bucket
    - This option must be used for a multi-Account scan scenario where multiple accounts are scanned, and documents are generated in a consolidated S3 Bucket
    - Please note that you can use this option when scanning non-AWS environments (Azure / GCP). You will need to specify a last part that contains the Role To Assume when connecting to the S3 Bucket:
        - ***arn:aws:s3::349224196492:myCloudockitDropOffBucket:CloudockitScanRole***

Permissions required for S3 Bucket

You will need to setup the appropriate permissions to the S3 Bucket like this:

For option 1

1. Create the following policy and add it to your IAM User

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::YOURS3Bucket"
            ]
        }
    ]
}
```

From the Storage Account, click on Permissions and then Bucket Policy. Ensure you have the following statement (replace the IAM User Arn and Resource)

```
{
    "Version": "2008-10-17",
    "Id": "Policy1335892530063",
    "Statement": [
        {
            "Sid": "Stmt1335892526597",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::xxxx:user/xxxxxxx"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::yourS3Bucket/*"
        }
    ]
}
```

For option 2

1. Create the following policy and add it to your ScanRole that is used for the cross-account authentication

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::YOURS3Bucket"
            ]
        }
    ]
}
```

From the Storage Account, click on Permissions and then Bucket Policy. Ensure you have the following statement (replace the IAM User Arn by the ARN of your role and Resource by your bucket)

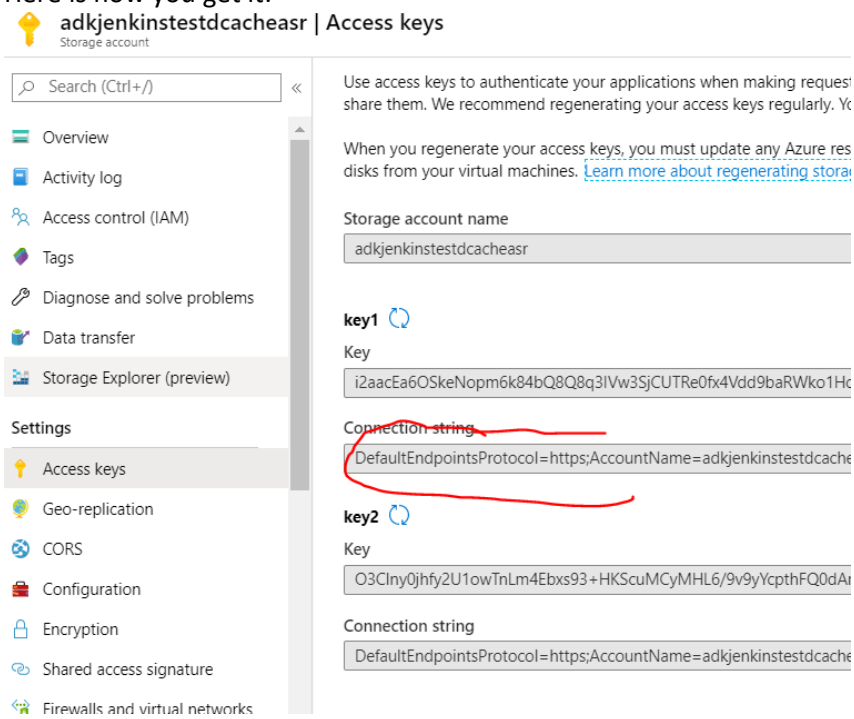
```
{
    "Version": "2008-10-17",
    "Id": "Policy1335892530063",
    "Statement": [
        {
            "Sid": "Stmt1335892526597",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::xxxx:user/xxxxxxx"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::yourS3Bucket/*"
        }
    ]
}
```

Azure Storage Account

There are two options for the Azure Storage Account you specify:

- Option 1: **Storage short name**
    - Specify the short name of the bucket like ***myCDKStorage***
    - This option can be used only if the Storage Account is in one of the subscriptions being scanned and the AAD tenant of the subscription is the same as the AAD tenant of the subscriptions being scanned
    - You need to ensure that the account being used has contribute permissions on the storage to write documents
- Option 2: **Storage Access keys**

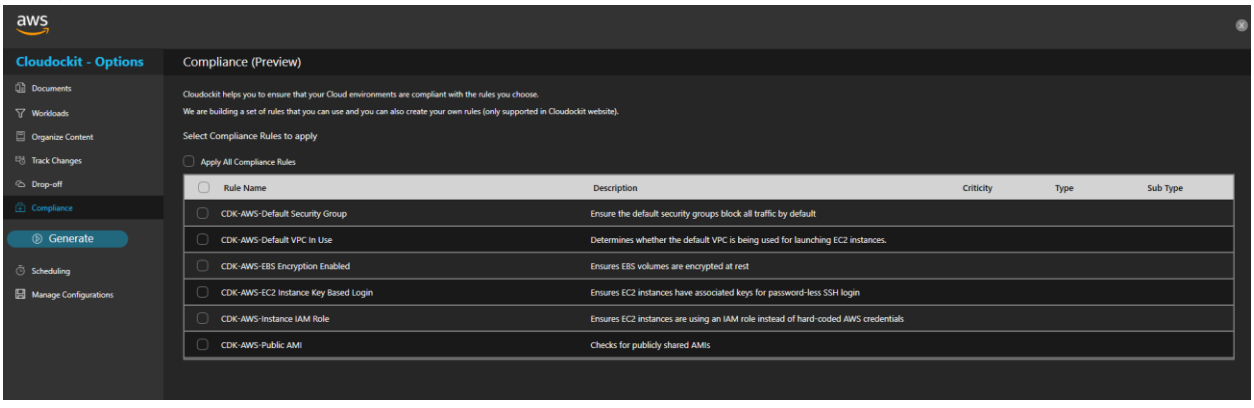
- Specify the connection string of the storage. This allows you to use Storage Account without any type of links to the subscriptions that you scan
- Here is how you get it:



GCP Bucket

You can specify your GCP Bucket account for drop off by entering the name of the GCP Bucket (short name). The Bucket needs to be available to the service account you are using. You also need to ensure the service account has privileges to write to this bucket.

Compliance



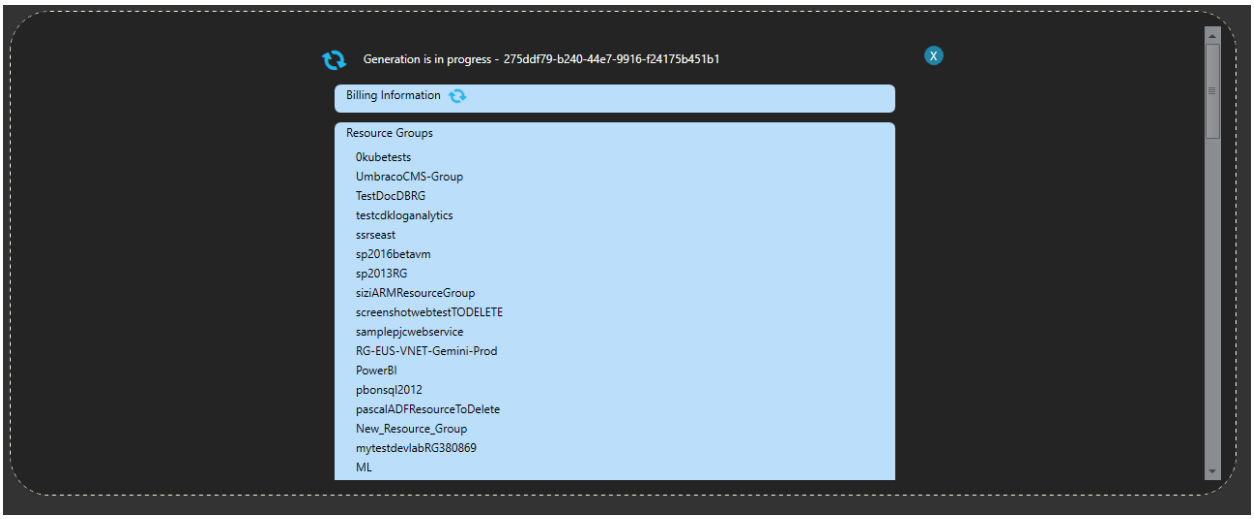
With the built-in set of rules, you can choose your Cloud environment setup. If the given set of rules aren't matching your personal needs, you can also a create your personal rules.

Step 3 – Start or schedule the generation

The last step is to start or schedule the document generation.

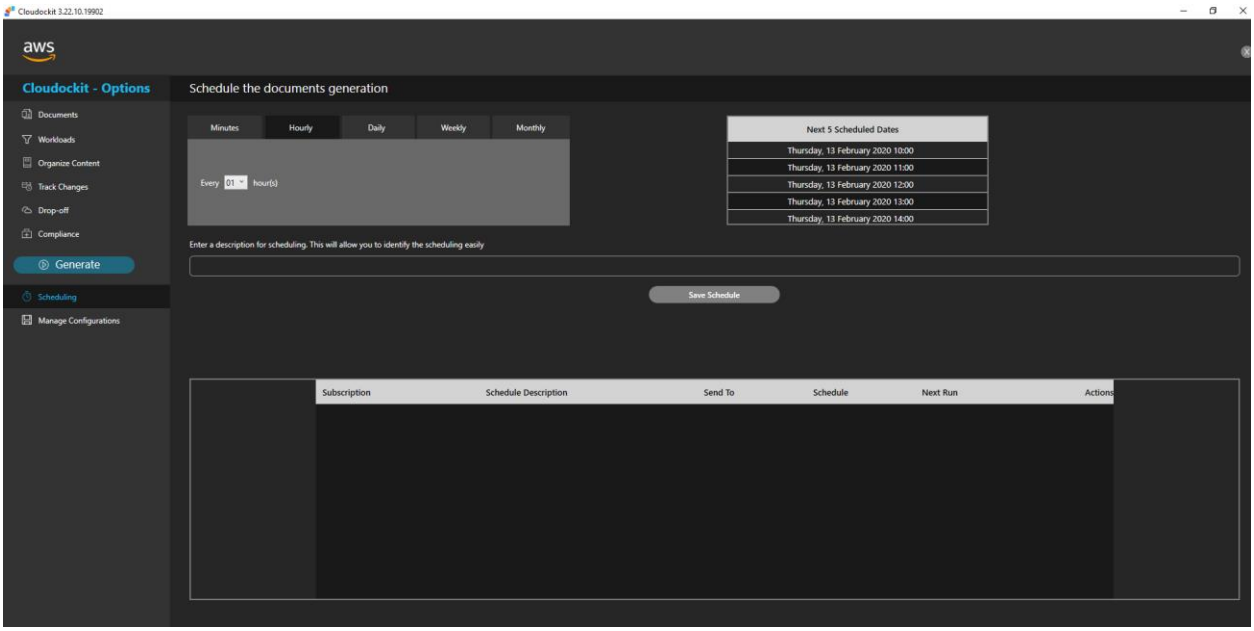
Start Document Generation

To start the document generation, click on **Generate** and it will start the process:



Schedule Document Generation

To Schedule the document generation, click on the Scheduling tab.



Then, enter the schedule you want to use.

Next, enter a description for this scheduling. Before saving it, go ahead and choose the parameters you would want your scheduling to have. The saving will take place automatically as you press on “Save Schedule”. If you want to modify this after saving, simply click on “Edit”, modify the parameters, and save the changes.

Note: The parameters will always be coming from the previous schedule parameters you modified. It will not revert to the previous parameters you selected before applying changes to a schedule.

Troubleshooting

Most Common causes of issues

Most of the time, issues that you may have with Cloudockit are the following:

- The output storage you specify is not accessible. Remember that the account you use MUST have WRITE permission on this storage
- The Keys you are using to connect to your environment have no permissions on any Azure Subscription or AWS Account, therefore, you cannot see any of your environment

Upgrade Issues

If you have an issue after an upgrade, please refer to the Upgrade Instructions section below.

General Procedure

If you still have an issue, please contact us at support@cloudockit.com and send us the file named Cloudockit.log located in the folder where you installed Cloudockit.

Upgrade Instructions

We try to minimize the breaking changes when we can. Most of the time, the upgrade instructions are basically empty.

In some circumstances we introduce breaking changes that require some manual changes. Please refer to the following section.

Upgrade to v3.20

Cloudockit Desktop v3.20 is now 100% 64 bits which introduces a new folder location C:\Program Files instead of C:\Program Files(x86).

If you created automation profiles with a version prior v3.20 and you migrated to v3.20, follow these steps:

- Copy the ADKSettings folder from C:\Program Files(x86)\Cloudockit to C:\Program Files\Cloudockit
- Open the Windows Task Scheduler and for each Task named AzureDockit, edit the properties to change the Action path and the Start Location

Upgrade to v3.22 or more

Cloudockit has been completely rewritten to no longer use the Task Scheduler. We are working on a tool to migrate existing schedules.

As the old notion of On-Prem Automation Profile is now deprecated and not accessible through the Cloudockit Website, you can contact us if you need help in the migration of that.