ABBYY Vantage

System Administrator's Guide

© 2022 ABBYY Development, Inc. All rights reserved.

Table of Contents

About ABBYY Vantage

ABBYY Vantage is a comprehensive Content Intelligence platform that provides AI-powered cognitive services and pre-trained and trainable skills that can "understand" business documents and extract actionable data and insights.

This no-code / low-code platform makes today's digital worker and processes smarter and empowers the new citizen developer to accelerate digital transformation initiatives and expand automation to new processes in a fast and simple way, making an immediate impact on business results and customer experience.

Vantage is capable of processing structured, semi-structured, and unstructured documents in a variety of input formats and languages.

The Vantage platform comes with a set of , which can extract data from certain document types out-of-the-box (i.e. invoices, purchase orders, receipts, bills of lading, delivery notes). These skills can be adjusted according to specific requirements and further trained based on customer-specific documents.

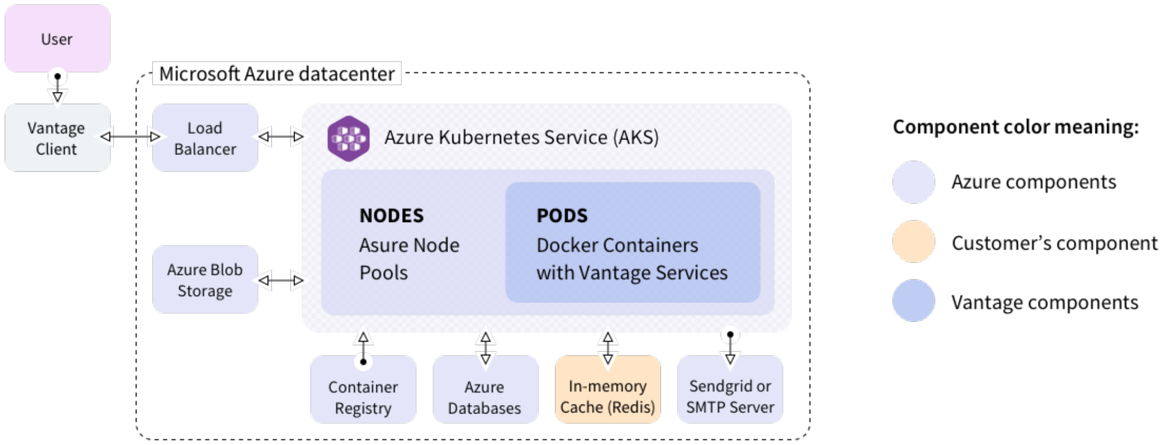
If the desired Document Skill is not available in the Vantage Skill Catalog more skills and technology components can be found in the ABBYY Marketplace.

Vantage users have also the option to design and train a completely new Document Skill, Classification Skill, and/or Process Skill based on their own document set.

Installing ABBYY Vantage in Azure

ABBYY Vantage is provided as a set of Docker containers running in the Kubernetes cluster and interacting with external components.

The installed Vantage scheme is as follows:



ABBYY Vantage installations can be implemented in one of two deployment options:

- **Highly available**. This configuration is designed to process a large number of pages and includes redundancy fail-safes. In this configuration, the Redis cluster is hosted on virtual machines outside of the Kubernetes cluster.
- **Without high availability**. A Proof of Concept (PoC) configuration recommended for demo, test, and trial deployments. It is designed to process a relatively small number of pages (up to several thousand pages per day). This configuration's requirements are minimal and fault tolerance is not implemented. In this configuration, the Redis cluster is hosted inside the Kubernetes cluster.

For more information about scaling tests for **Highly available** and **Without high availability** configurations, see the Performance Guide.

Requirements for both installation environments are listed in the System Requirements section. For instructions on how to install Vantage, please refer to the Installation section. For instructions on how to configure Vantage once it has been installed, please refer to the Initial Setup section.

System Requirements

The following external components should be configured before installing ABBYY Vantage (detailed requirements for each component are listed below).

- Microsoft Azure Managed Kubernetes Service (AKS) as a delivery option for Kubernetes (container orchestration system).
- Azure Node Pools to be used as Kubernetes nodes.
- Azure Public IP Address to provide access to Vantage via the Internet.
- Azure Databases for Vantage operation and for Business Processing Reporting Warehouse (if required).
- Azure Storage Accounts to store uploaded documents and extracted data.
- Container Registry to store Docker images of Vantage workers and services.
- Redis Cluster (for **Highly Available** configuration only) for communication between Vantage components.
- SMTP Server to send e-mail messages to Vantage users.
- TLS Certificate to establish an encrypted connection between Vantage and users.
- DNS record for further access to Vantage.

Microsoft Azure Managed Kubernetes Service (AKS)

Kubernetes Version 1.24.x is required.

Azure Node Pools

The number of machines to be used as Kubernetes nodes:

Machine Type	Maximum number of machines in a nodepool	Virtual CPU number	RAM, GB	Ephemeral Disk Size, GB
Standard_F8s_v2	10	8	16	100
Standard_E8_v4	3	8	64	100

Note: Depending on the load, Vantage automatically scales services and workers to process documents efficiently. The table above lists the maximum number of nodes of each type that Vantage can use when scaling. Once the product has been installed, the **Without high availability** configuration will not use all of the provided nodes, and will use 6 machines of the **Standard_F8s_v2** type and 3 machines of the **Standard_E8_v4** type instead, while the **Highly available** configuration will immediately begin using all allowed nodes. For more information about scaling and the configurations needed for different input loads, see the Performance Guide.

The Standard_E8_v4 pool must have the following labels and taints:

Labels:

- k8s.abbyy.com/techcore:true
- k8s.abbyy.com/overprovision-label:workers

Taints:

- k8s.abbyy.com/techcore=true:NoSchedule

Azure Public IP Address

This should be a Standard Static IPv4 address.

Azure Databases

The requirements are the following:

Database type	Series	Number of vCores	Max Storage Size, GB
Azure SQL Database with elasticpool	General Purpose: Standard-series (Gen 5)	4	100

Azure Storage Accounts

Four Storage Accounts of the following types are required:

Storage purpose	Storage type
Skills Storage	Premium
Processing Storage	Premium
Temporary Storage	Standard
Shared Folders Storage	Standard

The Shared Folders Storage must have at least 10 GB of space available.

Container Registry

You can use any container registry. Azure Container Registry is recommended.

Redis Cluster

This requirement is for **Highly available** configuration only.

Redis Streams support is required.

The following requirements should be met:

Version	Number of nodes	RAM, GB (for each node)	Cluster password	Specific options other than the standard redis.conf

6.2 and later	6 or more	4 (at least)	Should be set	`appendonly yes`
				`cluster-config-file nodes-6379.conf`
				`cluster-enabled yes`
				`cluster-node-timeout 5000`
				`cluster-require-full-coverage no`
				`maxclients 10000`
				`maxmemory 2048mb`
				`maxmemory-policy noeviction`
				`repl-backlog-size 256mb`
				`repl-ping-slave-period 5`
				`save ""`
				`slave-serve-stale-data yes`
				`stop-writes-on-bgsave-error no`
				`supervised auto`
				`masterauth` ***password***
				`requirepass` ***password***
				The value of the **maxmemory** parameter should be set to half of the memory available on the machine for each Redis node. Redis should be available for configuration on SSH port 22.

Note: Azure Cache for Redis is currently not supported. It will be supported in a future release.

SMTP Server

You can use:

- The SendGrid platform, or
- Any SMTP server that
  1. supports SMTP
  2. uses authentication

TLS Certificate

A wildcard or a domain-specific certificate is required.

DNS record

No specific requirements.

Creating an Azure Infrastructure to install Vantage

Note: While you are creating an infrastructure, you can begin downloading container images, as this is a lengthy operation.

Note: You need to install Redis on your virtual machines (see System requirements) if you are going to install Vantage in the **Highly available** configuration.

Described below are Azure Command-Line Interface (CLI) commands that should be used to create the required infrastructure using ARM templates. However, you can also implement your own infrastructure provision solution, as long as it meets our requirements.

1. Since the infrastructure is created inside the installer container, first you need to log in to the Vantage Docker registry and download the installer image to your Docker registry by running the following command:

```
docker login vantageonprem.azurecr.io -u MyToken -p J8Sj2qRjKv8whEM38w/rD38BSnlhwoJE
docker pull vantageonprem.azurecr.io/vantage-azure:2.3
docker tag vantageonprem.azurecr.io/vantage-azure:2.3
registry.yourdomain.tld/vantage/vantage-azure:2.3
docker push registry.yourdomain.tld/vantage/vantage-azure:2.3
```

2. Run the installer from a Docker image:

```
docker run -it registry.yourdomain.tld/vantage/vantage-azure:2.3
```

3. Sign in to your Azure account:

```
az login
az account set --subscription subscription_id
```

4. Create a resource group specifying the region where you want to deploy the cluster (the **location** parameter):

```
az group create --location location --resource-group resource_group_name
```

5. Create a Kubernetes cluster. First, check which resources will be created:

```
az deployment group what-if --name aks-cluster \
   --resource-group resource_group_name \
   --template-file files/infrastructure/azure/arms/Cluster.Manual.json \
   --parameters kubernetesClusterName=cluster_name
```

Create a cluster:

```
az deployment group create --name aks-cluster \
--resource-group resource_group_name \
--template-file files/infrastructure/azure/arms/Cluster.Manual.json \
--parameters kubernetesClusterName=cluster_name
```

6. Create storage accounts. First, check which resources will be created:

```
az deployment group what-if --name storage \
--resource-group resource_group_name \
--template-file files/infrastructure/azure/arms/StorageAccounts.Template.json
```

Create a storage account:

```
az deployment group create --name storage \
--resource-group resource_group_name \
--template-file files/infrastructure/azure/arms/StorageAccounts.Template.json
```

Get **accessKeys** and **secretKeys** for the storage accounts and populate the **s3storage** section in env_specific.yaml using these settings.

```
az deployment group show --name storage --resource-group resource_group_name --query
properties.outputs -o yaml
```

7. Create an Azure SQL Server and Azure SQL Databases if you don't have SQL Server. First, check which resources will be created for the server:

```
az deployment group what-if --name dbservers \
--resource-group resource_group_name \
--template-file files/infrastructure/azure/arms/Databases/DB.Server.Template.json \
--parameters serverBaseName=cluster_name \
--parameters dbAdminLogin=dblogin \
--parameters dbAdminPassword=dbpass
```

The **dblogin** and **dbpass** parameters correspond to the login and password that will be used to access the databases.

Create the Azure SQL Server:

```
az deployment group create --name dbservers \
--resource-group resource_group_name \
--template-file files/infrastructure/azure/arms/Databases/DB.Server.Template.json \
--parameters serverBaseName=cluster_name \
--parameters dbAdminLogin=dblogin \
--parameters dbAdminPassword=dbpass
```

Check which resources will be created for the database:

```
az deployment group what-if \
--name databases \
--resource-group resource_group_name \
--template-file files/infrastructure/azure/arms/Databases/SQL.Databases.Template.json
\
--parameters serverBaseName=cluster_name
```

Create the Azure SQL Database:

```
az deployment group create \
--name databases \
--resource-group resource_group_name \
--template-file files/infrastructure/azure/arms/Databases/SQL.Databases.Template.json
\
--parameters serverBaseName=cluster_name
```

8. Get nodeResourceGroup for the cluster:

```
az aks show --name cluster_name -g resource_group_name --query nodeResourceGroup -o
tsv
```

Save and populate the loadbalancer.external_ip inventory with the latest public IP address.

Installation

Note: Your cluster must have Internet access to install the product.

The ABBYY Vantage installation procedure consists of the following steps:

1. Preparing resources
2. Specifying resource credentials
3. Downloading container images
4. Running the installation scripts

Preparing Resources

Before you begin, make sure that all the requirements listed in System Requirements are met and that on your administrator's machine:

- Azure Kubernetes Service and all required external resources are accessible
- The Docker is installed
- The kubeconfig file needed to connect to the Azure Kubernetes Service cluster is located in the ~/ .kube/abbyy folder

Specifying resource credentials

Prior to running the installer, do the following:

1. Create a directory from which the installation will be carried out and navigate to this directory. For example:

   ```
   mkdir /opt/azure-install && cd /opt/azure-install.
   ```

2. Create an env_specific.yml file and replace the strings highlighted in red with the credentials of the external components required for the installation.

   You can either copy the code from the example below (be sure to keep the indents in your file the same as in this example) or download the file here.

   ```
   env: vantage
   poc: false
   domain: yourdomain.tld

   product_host: "vantage.{{ domain }}"

   loadbalancer:
     external_ip:  X.X.X.X

   container_registry_host: "registry.yourdomain.tld"
   container_registry_user: "service"
   container_registry_password: "password"
   container_registry_name: "{{ container_registry_host }}/vantage"

   install_nn_extraction_training_workers: false

   platform_namespace: abbyy-vantage
   ```

```
logging:
enabled: true
elasticsearch:
  enabled: false
  host: null
  port: 9200
  username: null
  password: null
  scheme: https
  ilm:
   create: false
file:
  enabled: true

platform:
  infra_namespace: abbyy-infrastructure
  monitoring_namespace: abbyy-monitoring

platform_admin_email: admin@yourdomain.tld

sendgrid:
  enabled: false
  apiKey: ""

smtp:
  host: X.X.X.X
  login: null
  password: ""
  port: 587
  useSSL: false

mailFrom: noreply@yourdomain.tld

database:
  type: sqlserver
  host: X.X.X.X
  username: login
  password: password
  encrypt: false

s3storage:
  skills:
    accessKey: access_key
    secretKey: secret_key
  processing:
    accessKey: access_key
    secretKey: secret_key
  temporary:
    accessKey: access_key
    secretKey: secret_key
  sharedfolder:
    accessKey: access_key
    secretKey: secret_key
    resourcegroup: resource_group

redis:
  ips: ['172.16.10.101', '172.16.10.102', '172.16.10.103', '172.16.10.104',
'172.16.10.105', '172.16.10.106']
  port: 6379
```

```
    password: redispassword
    ssl: false

  reporting:
    enabled: false
```

Parameter values

Parameter	Description
env	The installation prefix. Used as subdomain name by default.
poc	Specifies whether the **Without high availability** configuration will be installed. If the value of the parameter is **True**, the **Without high availability** configuration will be installed. If the value of the parameter is **False**, the **Highly available** configuration will be installed.
domain	The primary domain.
product_host	The DNS name that will be used to access the product. For example, if **product_host** is set to **vantage.{{ domain }}** and domain variable is "example.com," Vantage will be accessible via the following address: vantage.example.com.
loadbalancer.external_ip	An additional floating IP address for the balancer, which will be referenced by the primary domain name to access Vantage. This can be any free IP address on the subnetwork hosting the balancer virtual machines.
container_registry_host	The domain name (FQDN) of the Docker registry. Set to **registry.yourdomain.tld** by default.
container_registry_user	The name of the user with the permissions to download images from the Docker registry. This parameter is left blank if there is no Docker registry.
container_registry_password	Password/token for downloading images from the Docker registry. This parameter is left blank if there is no Docker registry.
container_registry_name	The directory of the Docker registry where the images are hosted. By default, the value of the property is **registry.yourdomain.tld/vantage**.

Parameter	Description
install_nn_extraction_training_workers	Specifies whether extraction workers will be installed. This parameter can be set to either **True** or **False**.
platform_namespace	Namespace used for the platform.
logging.enabled	Specifies whether logging should be enabled. This parameter can be set to either **True** or **False**.
logging.elasticsearch.enabled	Specifies whether Elasticsearch should be enabled. This parameter can be set to either **True** or **False**. 🔖 **Note:** Set the value of the parameter to **True** only if you already have Elasticsearch and want to connect Vantage logs to it. If you don't have existing Elasticsearch and Kibana instances, you can deploy them in a cluster using instructions in Elasticsearch and Kibana. This must be done before installing the product.
logging.elasticsearch.host	The server IP address.
logging.elasticsearch.username	The username used to connect to Elasticsearch.
logging.elasticsearch.password	The password used to connect to Elasticsearch.
logging.elasticsearch.scheme	The protocol scheme to connect to Elasticsearch.
logging.elasticsearch.ilm.create	Specifies whether index_templates and ilm policies for logs.abbyy.*indexes in elasticsearch should be created. This parameter can be set to either **True** or **False**.
logging.file.enabled	Specifies whether logs should be written as files. This parameter can be set to either **True** or **False**.
platform.infra_namespace	Namespace used for the infrastructure.
platform.monitoring_namespace	Namespace used for monitoring.
platform_admin_email	E-mail used to log in to the System Administrator's interface.
sendgrid.enabled	Specifies whether SendGrid for sending e-mails will be used. This parameter can be set to either **True** or **False**.
sendgrid.apiKey	The API key of SendGrid.

Parameter	Description
smtp.host	The IP address or name of the SMTP server host. **Note:** Fill in the SMTP block parameters only if the value of the **sendgrid.enabled** parameter is **False**.
smtp.login	The username used to connect to the SMTP server.
smtp.password	The password used to connect to the SMTP server.
smtp.port	The port of the SMTP server.
smtp.useSSL	Specifies whether an encrypted connection should be used. This parameter can be set to either **True** or **False**.
mailFrom	E-mail address of the sender of e-mail templates from the server.
database.type	The external SQL server type (sqlserver).
database.host	The IP address of the SQL server.
database.username	The username used to connect to the database (the user must have privileges required to create databases).
database.password	The password used to access the database.
database.encrypt	Specifies whether encryption is enabled in the database. This parameter can be set to either **True** or **False**.
s3storage.skills.accessKey	The name of the storage account for skills.
s3storage.skills.secretKey	The secret name of the storage account for skills.
s3storage.processing.accessKey	The name of the storage account for processing.
s3storage.processing.secretKey	The secret name of the storage account for processing.
s3storage.temporary.accessKey	The name of the storage account for intermediate transactions.
s3storage.temporary.secretKey	The secret name of the storage account for intermediate transactions.
s3storage.sharedfolder.accessKey	The name of the storage account for shared folders.
s3storage.sharedfolder.secretKey	The secret name of the storage account for shared folders.
s3storage.sharedfolder.resource_group	The resource group of the storage account for shared folders.

Parameter	Description
Note: Do not fill in the following 4 parameters related to the Redis cluster if you are using the **Without high availability** configuration.	
redis.ips	The Redis cluster IP address in the following format: ['172.16.10.101', '172.16.10.102', '172.16.10.103', '172.16.10.104', '172.16.10.105', '172.16.10.106'].
redis.port	The port used to connect to the Redis cluster.
redis.password	The password used to connect to the Redis cluster.
redis.ssl	Specifies whether an encrypted connection to the Redis cluster should be used. This parameter can be set to either **True** or **False**.
reporting.enabled	Specifies whether the Warehouse reporting service should be deployed. This parameter can be set to either **True** or **False**. **Note:** This service only supports Microsoft SQL server databases.

All other parameters are unchanged.

3. Create a directory named **ssl**. Place the certificate (along with the intermediate certificate) corresponding to the main domain name and the key in PEM format into the following files, respectively: **./ssl/cert.pem**, **./ssl/key.pem**. You should convert your CRT file to PEM by changing the contents of the file to the following format:
   -----BEGIN CERTIFICATE-----
   [your certificate]
   -----END CERTIFICATE-----

If an external authentication provider (Active Directory Federation Services) with a certificate signed by an internal certificate authority is used, place the root certificate into ./ssl/adfs-root.pem.

Downloading container images

To download container images, do the following:

1. Log in to the Vantage Docker registry and download the installer image to your Docker registry by running the following command:

```
docker login vantageonprem.azurecr.io -u MyToken -p J8Sj2qRjKv8whEM38w/rD38BSnlhwoJE
docker pull vantageonprem.azurecr.io/vantage-azure:2.3
docker tag vantageonprem.azurecr.io/vantage-azure:2.3
registry.yourdomain.tld/vantage/vantage-azure:2.3
docker push registry.yourdomain.tld/vantage/vantage-azure:2.3
```

2. Execute a script that will download the images to your Docker registry. The script does not run inside the container and runs on your administrator's machine instead.

You can find a list of relevant docker images in the images.list file. The file is already in the installation container, and you do not need to download it separately.

```
#!/bin/bash
images=./images.list
registry=registry.yourdomain.tld/vantage
for image in $(cat $images); do
  docker pull $image
  docker tag $image $registry/${image#vantageonprem.azurecr.io/}
  docker push $registry/${image#vantageonprem.azurecr.io/}
done
```

Running the installation scripts

ABBYY Vantage is installed automatically using the Ansible tool, which is installed and set up inside the Docker container used for the installation. Ansible interacts with the machines using various configuration scenarios (YAML playbooks).

To install ABBYY Vantage, follow the steps below.

1. Run the installer from a Docker image if you have not run it yet.

```
docker run -it \
-v current
directory/env_specific.yml:/ansible/inventories/azure/group_vars/all/env_specific.yml
\
-v current directory/ssl:/ansible/files/ssl:ro \
registry.yourdomain.tld/vantage/vantage-azure:2.3
```

Parameter values

Parameter	Description
-v current directory/ssl:/ansible/files/ssl:ro	The path to the folder with SSL certificates, which should contain the following files: cert.pem, key.pem, adfs-root.pem.

Sign in to your Azure account:

```
az login
az account set --subscription subscription_id
```

Point kubecontext to your cluster:

```
az aks get-credentials --resource-group resource_group_name --name cluster_name
```

Run the following command to make sure the correct context is being used:

```
kubectl cluster-info
```

2. Do a pre-deploy check:

```
ansible-playbook -i inventories/azure playbooks/0-PreflightCheck-azure.yml
```

The playbook will carry out a preliminary check regarding the resources whose parameters are specified in the **env_specific.yml** file of the playbook. If the playbook runs with errors, fix the errors before proceeding to the next installation step.

3. Accept the terms of the EULA and ABBYY Privacy Policy:

   ```
   ansible-playbook -i inventories/azure playbooks/legal.yml
   ```

   The playbook will ask the administrator to accept the terms of the EULA and ABBYY Privacy Policy.

4. Run the following command to install the product:

   ```
   ansible-playbook -i inventories/azure playbooks/site-azure-deploy.yml
   ```

   The playbook will deploy the metric and log collection system and deploy ABBYY Vantage.

The time required to complete the installation process will depend on the performance of the selected machines. On average, the process will take about 10 minutes. You can monitor the pods and job execution in a cluster using any tool for working with Kubernetes (for example, the Lens utility).
Once the deployment is finished, you will have a URL for the provided domain name, as well as a login and password that can be used to log in, create tenants, and process documents.

The Business Processing Reporting Warehouse is installed together with Vantage if the **reporting.enabled** parameter in the **env_specific.yml** file is set to **True**. If you choose not to install reporting initially, you can install it at a later point using the following command inside the **site-azure-deploy.yml** playbook:

```
ansible-playbook -i inventories/os/inventory -v playbooks/deploy-reporting.yml
```

If you want to use Grafana for analyzing data, you can deploy it in a cluster using the instructions in Grafana.

Initial Setup

After ABBYY Vantage is installed:

1. Open ABBYY Vantage and log in using the credentials of the default system administrator account displayed immediately after the installation.

2. Create a new tenant and assign a subscription (for more information, see Managing a Tenant).

Managing a Tenant

ABBYY Vantage is a multitenant system that operates using the following two scopes: **System Administrator's Scope** and **Tenant's Scope**.

Note: To ensure the different clients are logically isolated, users with permissions and roles for one of the scopes are not provided access to the other, and vice versa. Similarly, tenants are not able to access data from other tenants.

The **System Administrator's Scope** covers system administration. Only users with the **System Administrator** role are able to manage the system tenants.

This section contains instructions on how to manage a tenant:

- Creating and Deleting a Tenant
- Subscriptions

Creating and Deleting a Tenant

Creating a tenant

To create a new tenant, do the following:

1. Navigate to the **Tenants** tab in the left pane.
2. Click **New Tenant**.
3. In the dialog that will open, specify the following for your tenant: name, description, and email address of the tenant administrator. Next, upload the subscription you were provided with by clicking **Upload Subscription File**. For more information about subscriptions, see Subscriptions.

Once you have completed the above steps, a new tenant will be created in the system, and an invite with a registration link will be sent to the specified email.

A list of all tenants created in the system can be viewed in the **Tenants** tab. Tenants with invites that have not yet been accepted by the tenant administrator are denoted with an ✉ icon in front of their names. Hovering the mouse cursor over the icon displays the date when the invite was sent.

Note: Invites remain valid for 14 days, during which the user has to register by clicking the link in the invite. Otherwise, the tenant will be deleted from both the system and the list, and the link will become invalid.

If required, you can also send the same invitation again by doing the following:

1. Select the appropriate tenant by marking it in the tenant list.
2. Click ✉ **Resend Invite**.
3. In the dialog box that will open, modify the email address if required, and click **Send**.

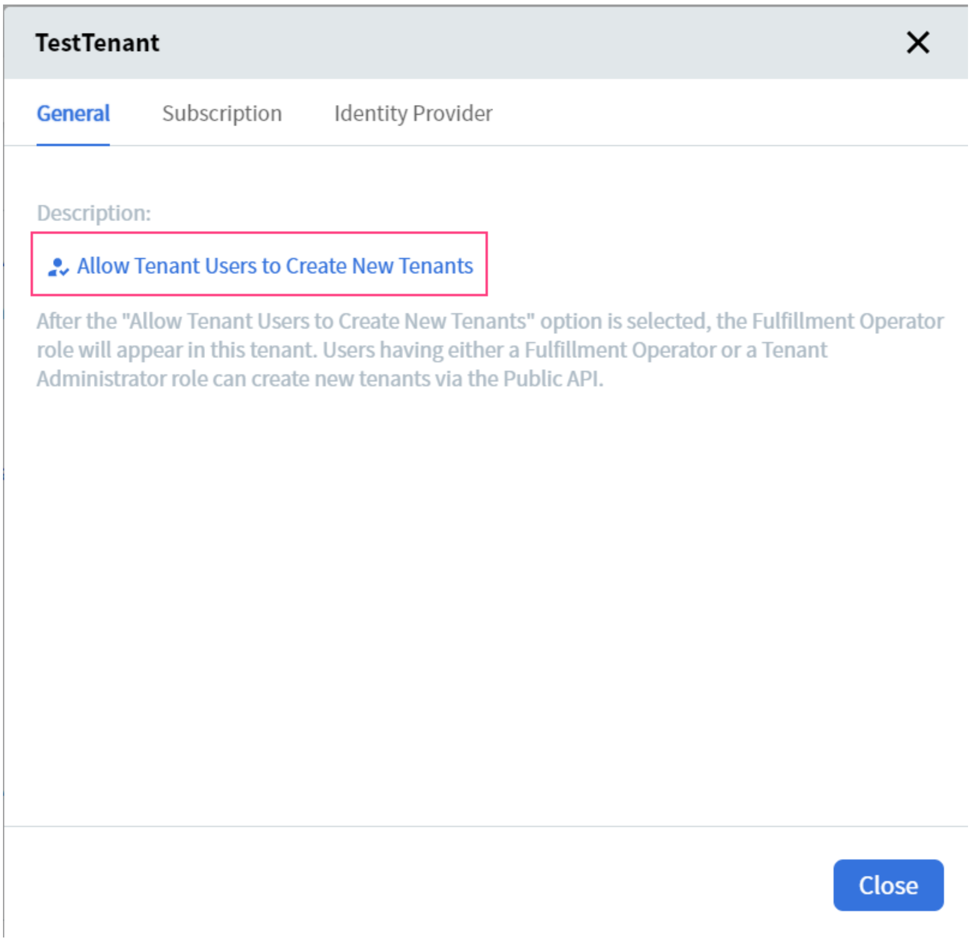
Creating a tenant via REST API

A System Administrator can assign the Fulfillment Operator role to a Tenant Administrator. With this role is assigned, the Tenant Administrator can add other tenants to the system via REST API as well as assign this role to other users in the tenant.

To assign the Fulfillment Operator role, a System Administrator needs to do the following:

1. Navigate to the **Tenants** tab in the left pane and click on the required tenant.

2. In the dialog that will open, navigate to the **General** tab and click **Allow Tenant Users to Create New Tenants**.



TestTenant

Description:

After the "Allow Tenant Users to Create New Tenants" option is selected, the Fulfillment Operator role will appear in this tenant. Users having either a Fulfillment Operator or a Tenant Administrator role can create new tenants via the Public API.

⚠**Important!** Once this role has been assigned to a tenant, it cannot be unassigned.

A Tenant Administrator that has been assigned this role will be able to create tenants via REST API requests, as well as assign this role to other users in the tenant.

Deleting a tenant

To delete a tenant, do the following:

1. Select the appropriate tenant by marking it in the tenant list.
2. Click 🗑 **Delete** and confirm the deletion.

Once you have completed the above steps, the tenant will be removed from the list, and access to its data will be blocked. According to the ABBYY retention policy, tenants are deleted from the system 30 days after their deletion from the list. Before that period has elapsed, it is not possible to create a new tenant with the same name.

Subscriptions

For a tenant to work with ABBYY Vantage, it needs to have an active subscription. Tenant subscriptions are used to determine which application features the tenant will have access to.

In particular, tenant subscriptions determine the following:

- the amount of time during which access to skills will be provided,

- the number of pages that can be processed while the subscription is active,
- the available skills.

To count pages processed using various skills, counters are included in a subscription. For more information, see How Pages Are Counted in ABBYY Vantage.

Note: A subscription applies to all users in a tenant. If an ABBYY Vantage subscription is tied to several tenants within a single Vantage installation, these tenants will have shared counters for each skill used as part of the subscription.

Subscription Parameters

A system administrator can view information about any tenant’s subscription.

This can be done as follows:

1. Click the name of the appropriate tenant in the **Tenants** tab.
2. In the dialog that will open, navigate to the **Subscription** tab.

This tab contains an overview of the following subscription parameters:

- the subscription type, its serial number and expiry date,
- counters included in the subscription,
- information about remaining available pages for each of the skills,
- the date of the next counter update if counters are renewable,
- the number of pages that will become available with the next update,
- any additional subscription options.

How Pages Are Counted in ABBYY Vantage

ABBYY Vantage keeps count of all document pages processed by users with license-specific counters. The counter type used to calculate the number of processed pages depends on the skill used and on whether the skill is licensed or not. Licensed skills are skills that have been created by ABBYY or its partners and have undergone the licensing procedure at ABBYY.

There are several types of counters used in ABBYY Vantage:

1. Licensed skill counters are used to count pages processed using licensed skills. Each licensed skill has its own special counter, for example **ABBYY Air Waybill**, **ABBYY Bill Of Lading**, etc.
2. A **Core Cognitive Skills** counter is used to calculate the number of pages processed by all unlicensed ABBYY Vantage skills.
3. An **OCR Skill** counter is used to calculate the number of pages processed using OCR skills.
4. **Licensed Skills** counter is used to calculate the number of pages processed by trial versions of licensed skills only. This lets you process documents using licensed skills whose counters are either not part of your license or have run out of pages to process. This way, you can try out a licensed skill before purchasing it.

All pages processed by non-licensed skills are counted using a single Core Cognitive Skills counter. Pages processed by licensed skills are counted using separate Licensed Skill counters.

▾ List of licensed skills

Skill Name	License Name
Invoice US	ABBYY.Invoice
Invoice EU	
Invoice CA	
Invoice ES	
Invoice AU-NZ	
Invoice CN	
Invoice Classifier Skill	
Invoice Document Splitter Skill	
Purchase Order US	ABBYY.PurchaseOrder
Purchase Order EU	
Purchase Order Classifier Skill	
Receipt	ABBYY.Receipt
Commercial Invoice	ABBYY.CommercialInvoice
Bill Of Lading	ABBYY.BillOfLading
Air Waybill	ABBYY.AirWaybill
Remittance Advice	ABBYY.RemittanceAdvice
Utility Bill	ABBYY.UtilityBills
Bank Statement	ABBYY.BankStatements
Personal Earning Statement	ABBYY.PersonalEarningStatements
Arrival Notice	ABBYY.ArrivalNotice
IRS Tax Form W-2, Wage and Tax Statement	ABBYY.IrsTaxFormW-2
IRS Tax Form 1040, U.S. Individual Income Tax Return	ABBYY.IrsTaxForm1040

Pages are counted for documents processed during design time when modifying document and classification skills and during runtime when using any type of skill (document, classification, process, OCR).

Design time document processing for document and classification skills includes the following:

- importing demo documents to a document skill,
- importing files to a document set for training document and classification skills.

Runtime document processing includes the following:

- uploading pages to the **Documents** section,
- processing documents in transactions,
- document uploads when using the **Try Skill** feature.

All document pages processed during both design time and runtime are counted in the corresponding skill counters.

Note: Pages will be counted even if the transaction fails after the import of their respective documents.

All subscription counters are updated once during a set period of time called the renewal period. At the end of this period, all the counter values are reset and the maximum number of pages again becomes available to the user.

The duration of the renewal period and its start date are the same for all counters in a subscription, regardless of when any of the skills were used for the first time.

A single image of a page of any format is counted as one page, even if the image contains several different documents.

If the number of pages available for processing by a specific skill has run out, or if the currently active subscription does not provide access to the required processing skill, users will not be able to create new transactions using such skills. For more information, see Subscription Limits.

How pages are counted for process skills

Generally, process skills contain one or more OCR, document and/or classification skills (which can be both licensed and unlicensed). Depending on the exact set of skills contained within a process skill, pages for the document being processed will be counted as follows:

- **If the process skill uses only unlicensed skills**
  Each page will increment the Core Cognitive Skills counter once, regardless of the number of skills used to process it.
- **If the process skill uses one licensed skill**
  Each page will increment the licensed skill’s counter once, regardless of how many times the licensed skill was used to process the document.
- **If the process skill uses unlicensed skills and one licensed skill**
  Each page will increment the licensed skill's counter or the Core Cognitive Skills counter, depending on what skill was used to process the documents.

- **If the process skill uses several licensed skills**
  Each document page will increment each licensed skill's counter.

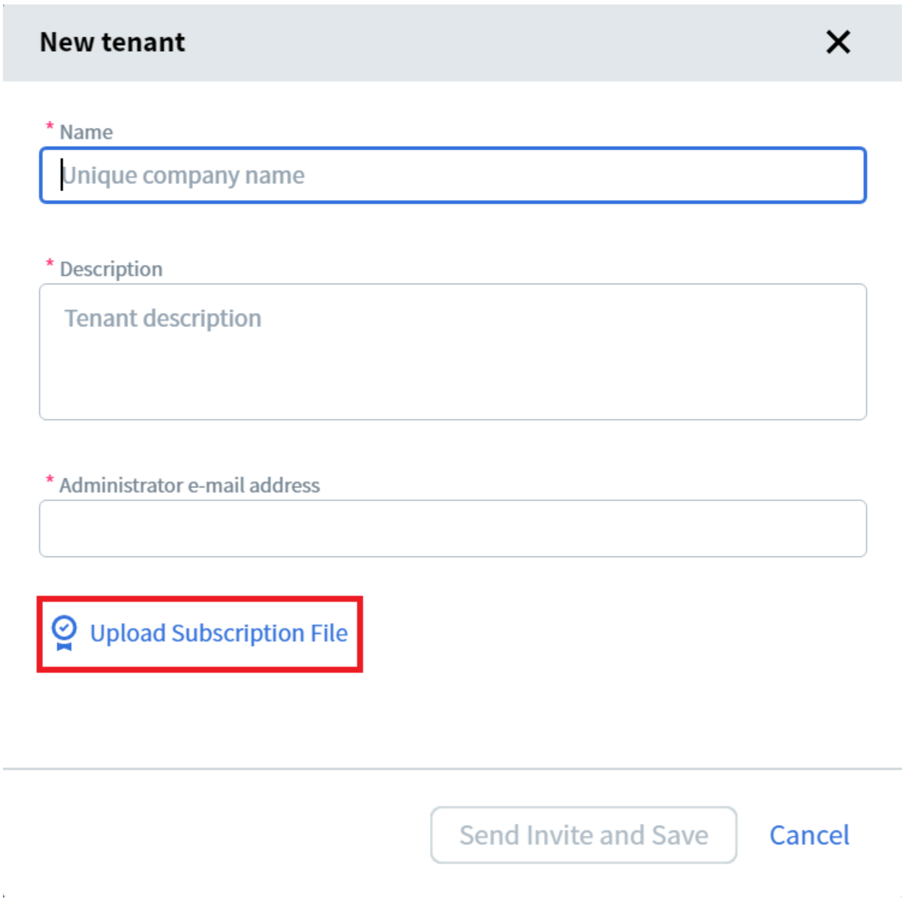
Managing Subscriptions

The system administrator can manage tenant subscriptions as follows: link, modify, and delete subscriptions.

Linking a subscription

A subscription file that you have received from ABBYY can not only be linked to a new tenant when it is created, but also to an existing tenant.

In the case of the former, the subscription file needs to be uploaded when a new tenant is created. To do so, click **Upload Subscription File**.



Modifying or deleting a subscription

If the number of pages available for processing for each of the skills has run out, or if the currently active tenant subscription lacks the required skill to process documents, the system administrator may swap the current subscription file for a new subscription file with different page limits and counters. If the tenant needs to be suspended, the subscription can be deleted.In this case, tenant users will not be able to create transactions using any of the skills.

To modify or delete a subscription, do the following:

1. Open the **Tenants** tab and click the name of the appropriate tenant.
2. In the dialog that will open, navigate to the **Subscription** tab.

3. Click the ⋮ icon displayed next to the subscription type name and select either **Change Subscription File** or **Delete Subscription**, depending on your desired action.

Setting up a Database Connection

ABBYY Vantage uses databases hosted on external servers and may become inoperable if those servers fail. In this case, the system administrator is able to restore such databases on a different server and set up a connection to the new databases using Consul.

Note: Before starting, make sure that the kubectl command line tool is installed and that a connection to the Kubernetes cluster has been established.

To set up a connection to a new database in the ABBYY Vantage settings, do the following:

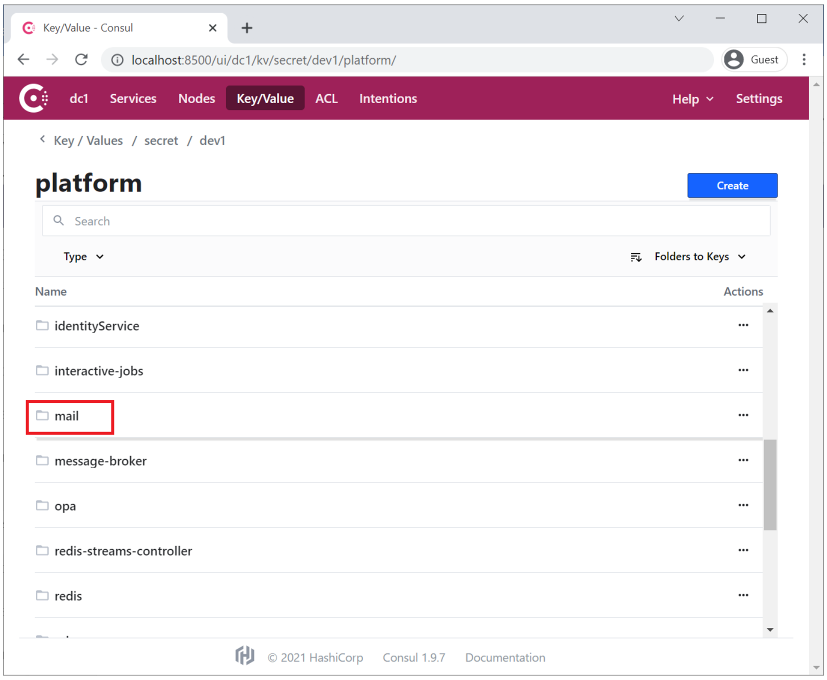
1. Access the Consul web interface by running the command below

```
kubectl port-forward -n abbyy-infrastructure service/consul-ui 8500:80
```

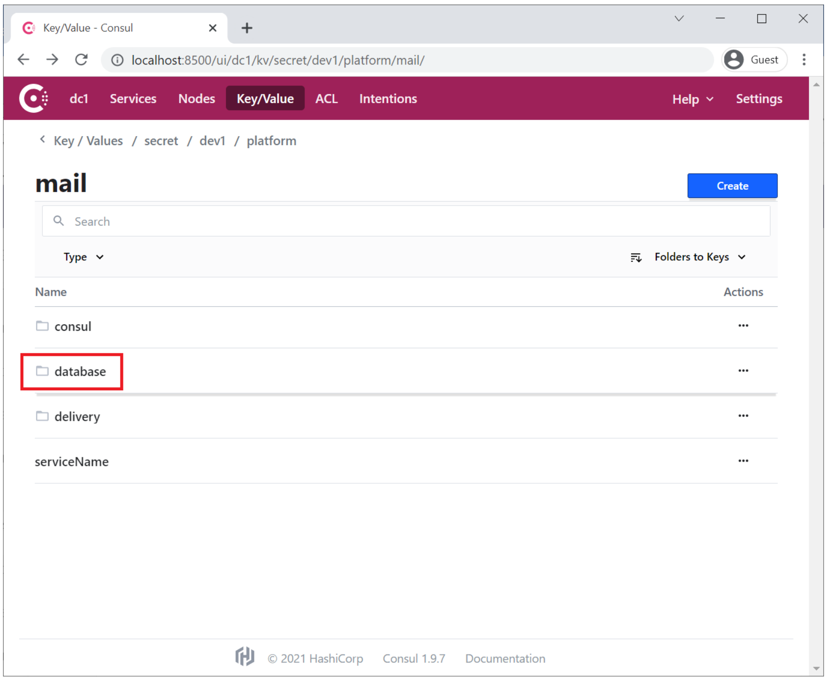
and then navigating to http://localhost:8500/ui/dc1/kv/secret/.

2. Use the **Key/Value** tab that will open to select the correct Vantage environment.

3. Select either the **platform** or the **vantage** project, as well as the appropriate service that uses the database, e.g. **mail**.

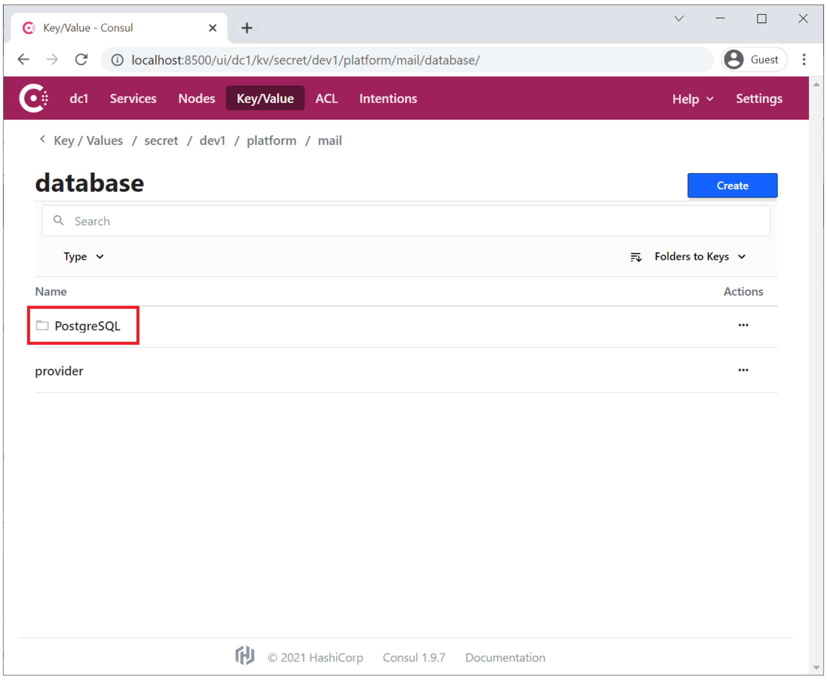


4. Navigate to the **database** section that every service contains.



Name	Actions
consul	···
database	···
delivery	···
serviceName	···

5. Open the **PostgreSQL** section.



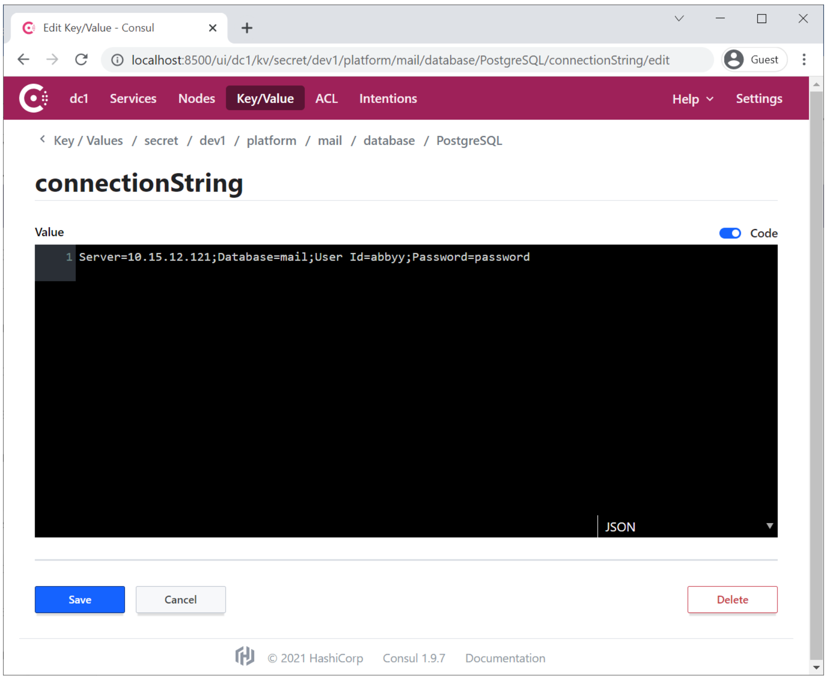
Name	Actions
PostgreSQL	···
provider	···

6. In the **connectionString** key:

    1. Replace the old value of **Server** with the address of the new server.

    2. Specify the new database in the **Database** parameter.

    3. Specify the login credentials for the database in the **User Id** and **Password** parameters.



```
Server=10.15.12.121;Database=mail;User Id=abbyy;Password=password
```

7. Click **Save**.

8. Restart the modified service by running the following command:

```
label=mail
kubectl -n abbyy-vantage rollout restart $(kubectl -n abbyy-vantage get deployments –
l app.kubernetes.io/component=$label -o name)
```

Note: When a server address changes, this procedure has to be carried out for every database.

Below is a table listing all services that use the database, as well as their label that can be used to find and restart each service.

Name of the Consul section name	Service label	Notes
platform		
api-gateway-registry	api-gateway-registry	
api-registry	api-registry	
auth-adminapi2	auth-adminapi2	
auth-identity	auth-identity	
auth	auth-sts-identity, auth-adminapi2	This database is used by two services.
blob-storage	blob-storage	
cron-service	cron-service	
documentsetstorage	documentsetstorage	

Name of the Consul section name	Service label	Notes
mail	mail	
skill-monitor	skill-monitor	
storage	storage	The **database** section is stored in the **fileMetadata** catalog.
workflow-facade	workflow-facade	
vantage		
catalogstorage	catalogstorage	
folderimport	folderimport	
mailimport	mailimport	
onlinemlservice	onlinemlservice	
publicapi	publicapi	
secretstorage	secretstorage	
skill-monitor	skill-monitor	
skillinfo	skillinfo	
subscriptions	subscriptions	
tokenmanagement	tokenmanagement	
transactions	transactions	
workspace	workspace	

Setting up a Manual Review Inactivity Timeout

In Manual Review, if no actions are taken by the operator for a period of 15 minutes with regards to an open task, a timeout is triggered. The ABBYY Vantage System Administrator can change the length of inactivity required for a timeout using Consul.

This parameter can be set up by doing the following:

1. Access the Consul web interface by running the command below

   ```
   kubectl port-forward -n abbyy-infrastructure service/consul-ui 8500:80
   ```

   and then navigating to http://localhost:8500/ui/dc1/kv/secret/.

2. Use the **Key/Value** tab that will open to select the correct Vantage environment.

3. Change the values of the following two keys:

Key	Description
secret/abbyy-vantage/vantage/verification/interactiveJobsOptions/**popTimeout**	The minimum period of time a user is inactive before a task will be returned to the interactive task queue. Any interactive action (mouse movement, keyboard input, patch processing, etc.) will reset the inactivity period countdown. The default value of the key is "**00:15:00**" (15 minutes).
secret/abbyy-vantage/vantage/verification/interactiveJobsOptions/**processingPopTimeout**	The minimum period of user inactivity after which the task will be returned to the queue of interactive tasks if there are long-term operations in the queue of this task (applying a skill, turning pages, etc.). When a long-running operation starts, this key value is set to the maximum allowable inactivity period. When the operation completes, the inactivity period is again reset to the **popTimeout** key value. The default value of the key is "**1.00:00:00**" (24 hours).

4. Click **Save**.

5. Restart the **verification** and **manualverification** services by running the following command:

```
label=verification
kubectl -n abbyy-vantage rollout restart $(kubectl -n abbyy-vantage get deployments -
l app.kubernetes.io/component=$label -o name)
```

and

```
label=manualverification
kubectl -n abbyy-vantage rollout restart $(kubectl -n abbyy-vantage get deployments -
l app.kubernetes.io/component=$label -o name)
```

Setting up OAuth 2.0 Authentication for Connecting to the IMAP Server

By default, users setting up document import from an email service using an Input activity in a Process Skill only have access to basic IMAP server authentication. In order for Google and Microsoft email services authentication to also become available via the OAuth 2.0 protocol, do the following:

1. Register the applications on Google Cloud Platform and the Azure portal;

2. Generate account credentials for these applications (Client ID and Client secret);

3. Pass the generated credentials to Consul.

The above can be done both when preparing to install Vantage, as well as after the installation.

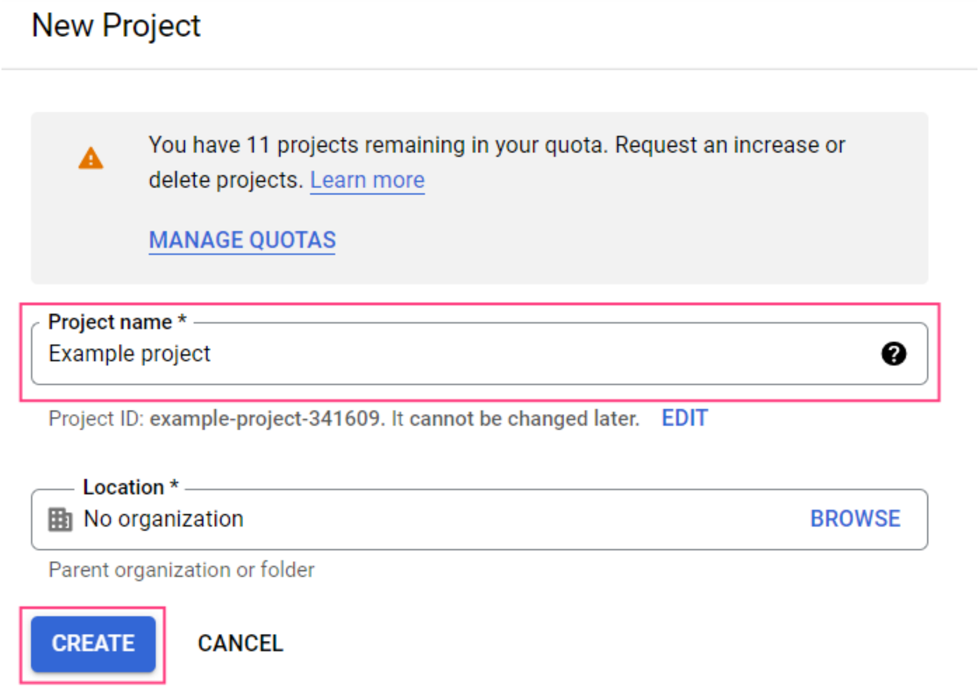
Registering the Application in Google

Creating an application requires a Google account.

Creating a project on the Google Cloud Platform

1. Navigate to the Google Cloud Platform New Project page.

2. Specify a name for your project and click **Create**.



New Project

You have 11 projects remaining in your quota. Request an increase or delete projects. Learn more

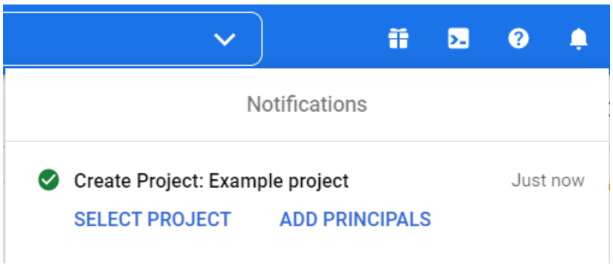
Project name *
Example project

Project ID: example-project-341609. It cannot be changed later.

Location *
No organization

Parent organization or folder

Wait for a notification saying that your project has been created.

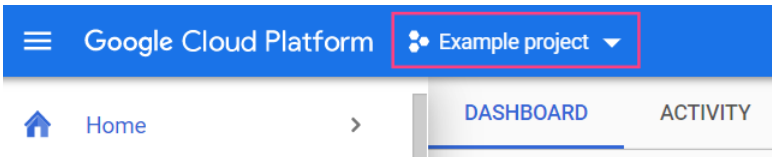


Notifications

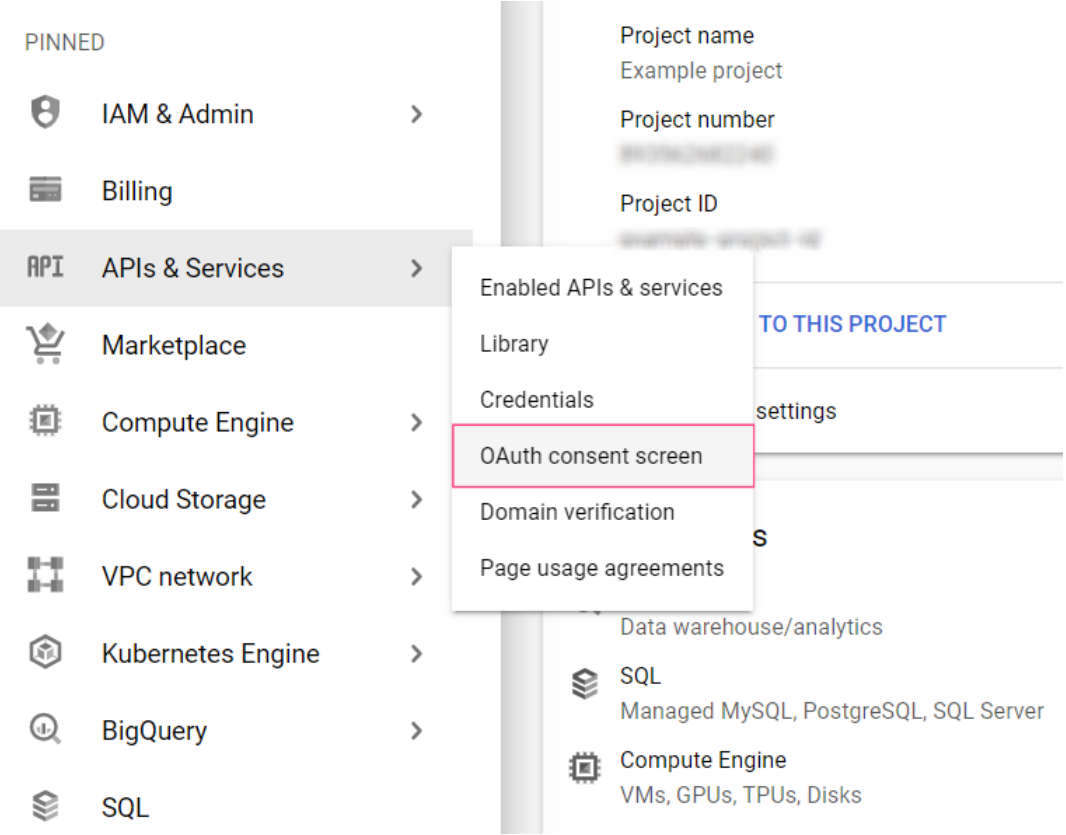
Create Project: Example project Just now

Setting up the application

1. Navigate to the Google Cloud Console and select the appropriate project.

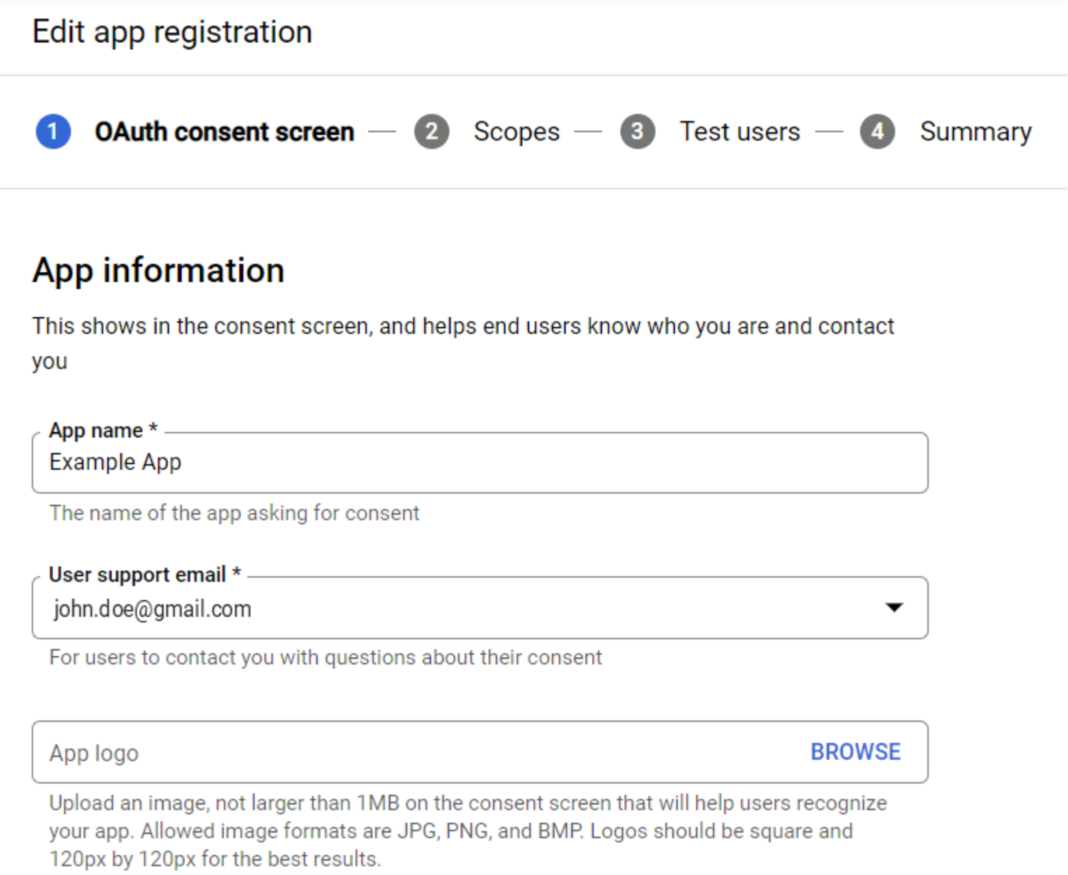


2. In the menu on the left side of the screen, select **APIs & Services** > **OAuth consent screen**.



3. Select the **External** user type and click **Create**.

4. Specify a name for your application. In the **User support email** drop-down list field, select your Gmail address.



Edit app registration

1 **OAuth consent screen** — 2 Scopes — 3 Test users — 4 Summary

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *: Example App

The name of the app asking for consent

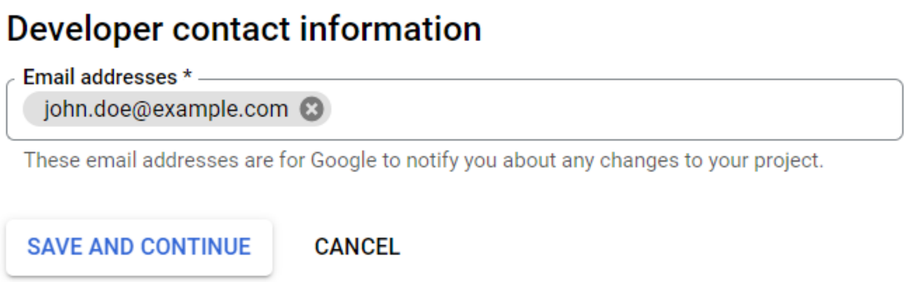
User support email *: john.doe@gmail.com

For users to contact you with questions about their consent

App logo BROWSE

Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 120px by 120px for the best results.

5. Specify the developer's email in the **Developer contact information** section at the bottom of the page and click **Save and continue**.



Developer contact information

Email addresses *: john.doe@example.com

These email addresses are for Google to notify you about any changes to your project.

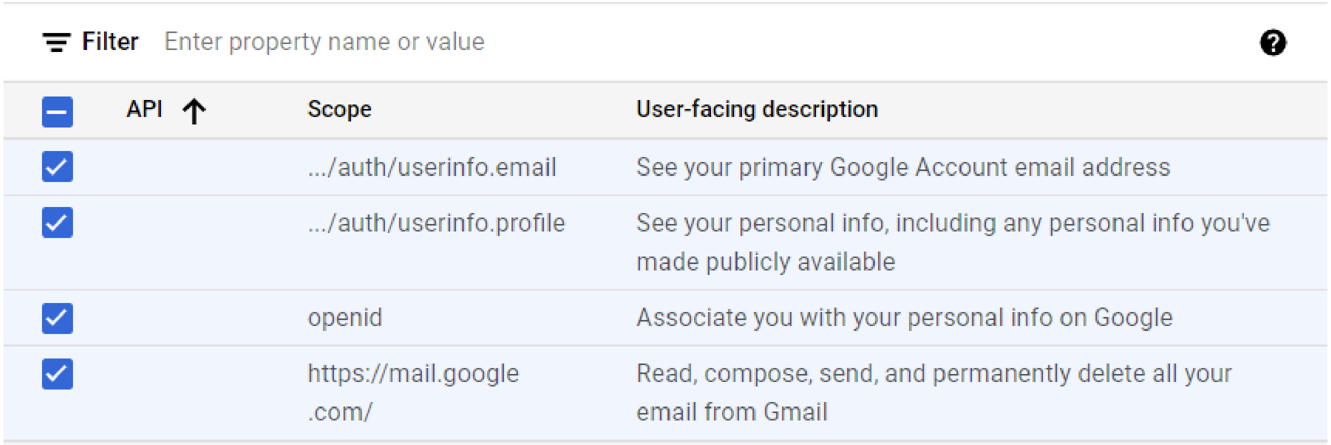
6. Click **Add or remove scopes**. This will open the **Update selected scopes** dialog on the right.

7. Copy and paste the following text into the **Manually add scopes** field in the bottom part of the dialog and click **Add to table**:

```
openid https://www.googleapis.com/auth/userinfo.email
https://www.googleapis.com/auth/userinfo.profile https://mail.google.com/
```

Note: You can also select scopes manually. The following scopes need to be selected:

- openid
- https://mail.google.com/
- ../auth/userinfo.email
- ../auth/userinfo.profile



☐	API ↑	Scope	User-facing description
☑		.../auth/userinfo.email	See your primary Google Account email address
☑		.../auth/userinfo.profile	See your personal info, including any personal info you've made publicly available
☑		openid	Associate you with your personal info on Google
☑		https://mail.google .com/	Read, compose, send, and permanently delete all your email from Gmail

8. Click **Update**. This will close the **Update selected scopes** dialog and display the selected scopes.

9. Click **Save and continue** at the bottom of the screen.

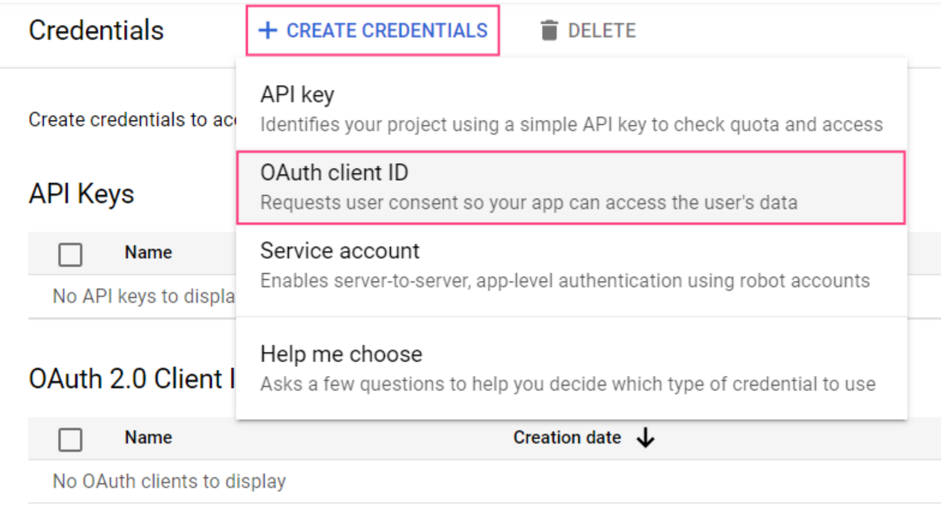
10. Click **Save and continue** to skip the **Test users** page settings and navigate to the **Summary** page.

On the **Summary** page, the following is displayed: information about the application, email addresses, and permissions that have been set up.

Creating account credentials

1. Select **Credentials** in the menu on the left side of the screen.

2. Click **+ Create credentials** and select **OAuth client ID**.



Credentials | + CREATE CREDENTIALS | DELETE

Create credentials to ac

API Keys

□ Name

No API keys to displa

OAuth 2.0 Client I

□ Name | Creation date ↓

No OAuth clients to display

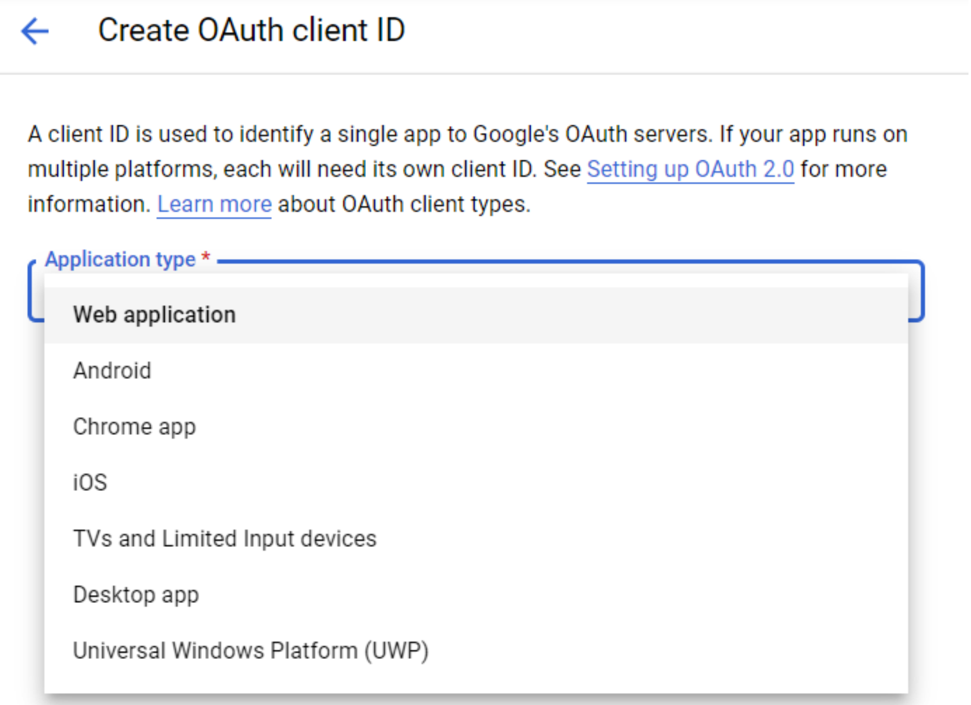
API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so your app can access the user's data

Service account
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

3. Select the Web application type.



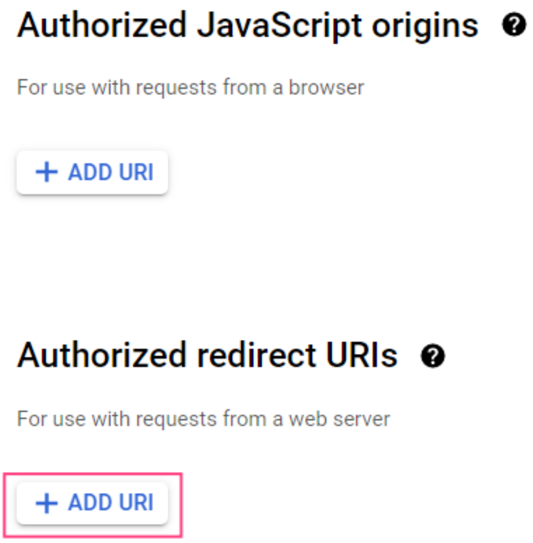
Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 for more information. Learn more about OAuth client types.

Application type *

- Web application
- Android
- Chrome app
- iOS
- TVs and Limited Input devices
- Desktop app
- Universal Windows Platform (UWP)

4. In the **Authorized redirect URIs** section, select **+ Add URI**.



Authorized JavaScript origins

For use with requests from a browser

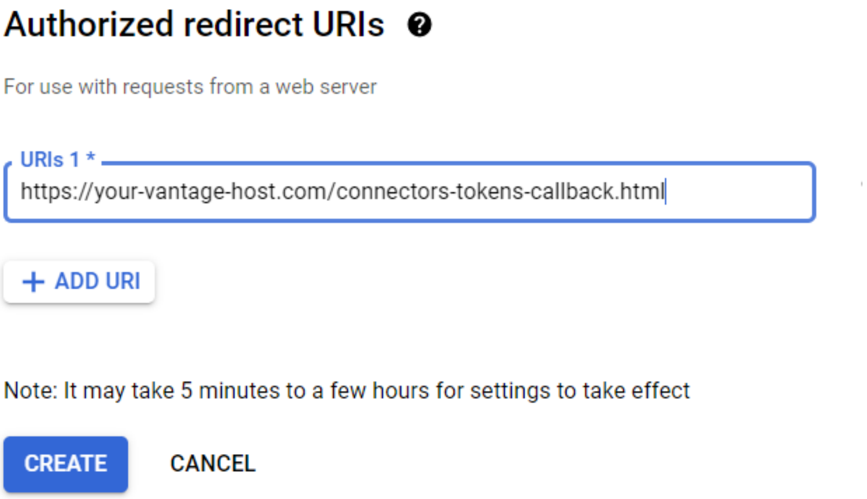
+ ADD URI

Authorized redirect URIs

For use with requests from a web server

+ ADD URI

5. In the field that will appear, specify the redirect URI: https://<Vantage host name>/connectors-tokens-callback.html.



Authorized redirect URIs

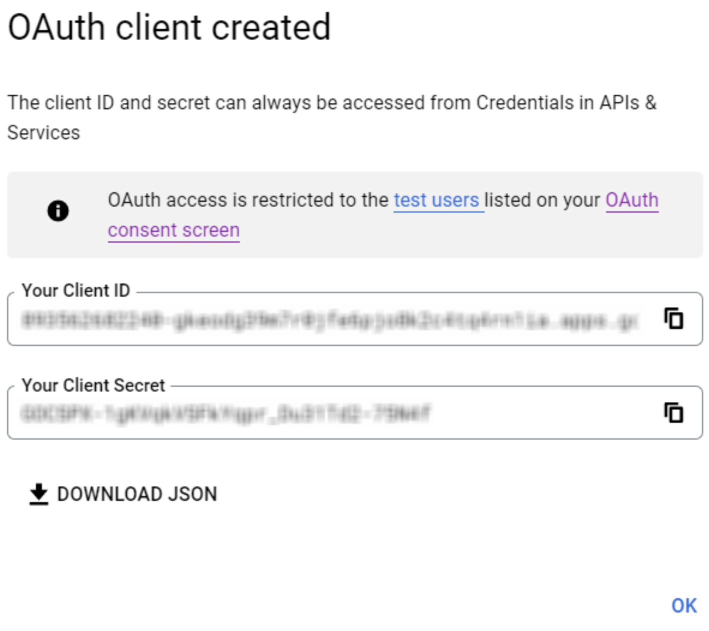
For use with requests from a web server

URIs 1 *: https://your-vantage-host.com/connectors-tokens-callback.html

Note: It may take 5 minutes to a few hours for settings to take effect

6. Click **Create**.

The pop-up dialog box that will appear will contain the Client ID and Client secret values.



OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

OAuth access is restricted to the test users listed on your OAuth consent screen

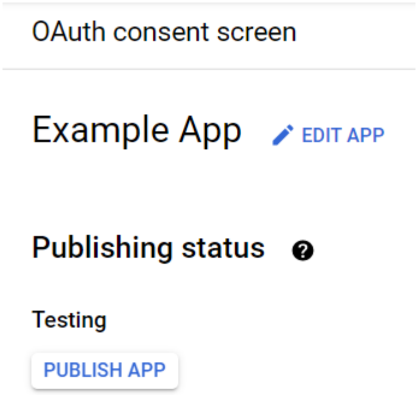
Your Client ID: [redacted]

Your Client Secret: [redacted]

This data is required for setting up the tokenmanagement service in Vantage. You can save it immediately or copy it later by navigating to the **APIs & Services** > **Credentials** page in the menu on the left of the screen and selecting the OAuth 2.0 client identifier you have created.

Publishing and verification

The publishing status of the application is displayed in the **APIs & Services** > **OAuth consent screen** section.



OAuth consent screen

Example App

EDIT APP

Publishing status

Testing

PUBLISH APP

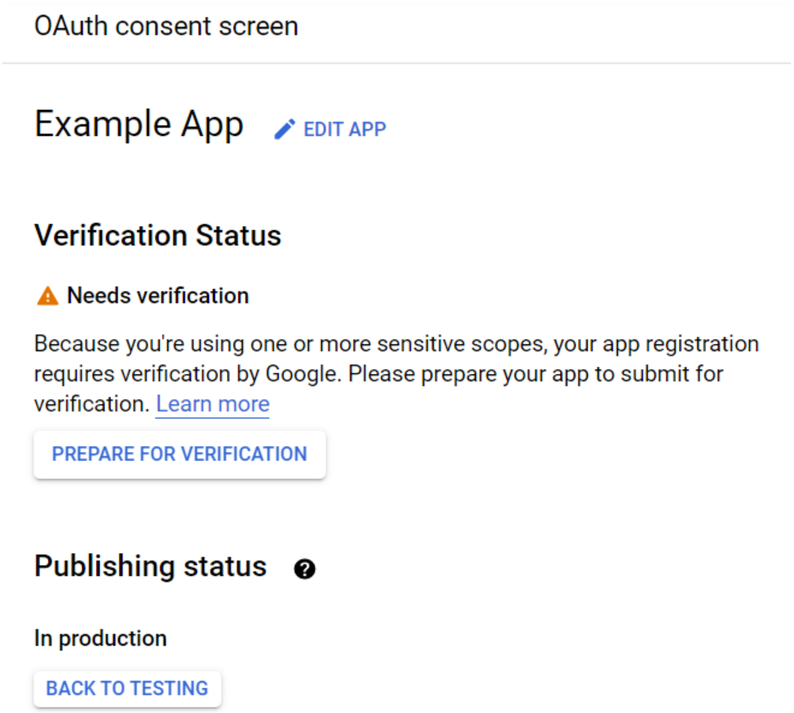
Applications with the **Testing** status are only available to users that have been added to the testers list. Only publishing an application makes it available to any user with a Google account.

Click **Publish app**. The https://mail.google.com/ scope allows the application to access confidential user data, which is why a message saying that the application needs to be verified will be displayed. To verify the application, you will need to provide the following:

- An official link to the application's Privacy policy,
- A YouTube video demonstrating the stated purpose of obtaining Google user data using the application,
- A text addressed to Google that contains a description of why you require access to confidential user data,
- A full list of all your domains verified in the Google Search Console.

Click **Confirm**. The status of your application will change to **In Production**.

The **Prepare for verification** button will also appear. This button lets you provide all required verification data.



OAuth consent screen

Example App

EDIT APP

Verification Status

⚠ Needs verification

Because you're using one or more sensitive scopes, your app registration requires verification by Google. Please prepare your app to submit for verification. Learn more

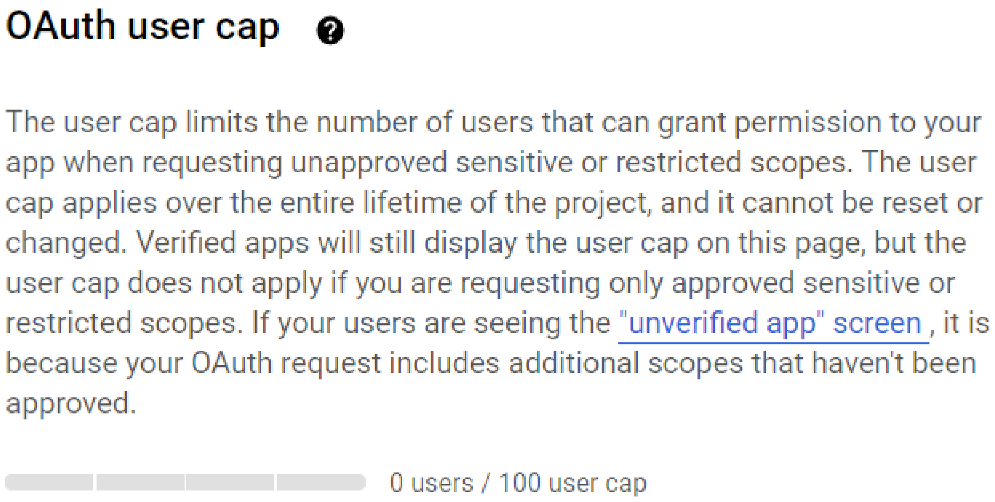
PREPARE FOR VERIFICATION

Publishing status

In production

BACK TO TESTING

Note: Before your application has been verified, only 100 users are able to use it. The user counter is located in the bottom part of the **OAuth consent screen** section and cannot be reset throughout the projects's lifetime.



OAuth user cap

The user cap limits the number of users that can grant permission to your app when requesting unapproved sensitive or restricted scopes. The user cap applies over the entire lifetime of the project, and it cannot be reset or changed. Verified apps will still display the user cap on this page, but the user cap does not apply if you are requesting only approved sensitive or restricted scopes. If your users are seeing the "unverified app" screen , it is because your OAuth request includes additional scopes that haven't been approved.

0 users / 100 user cap

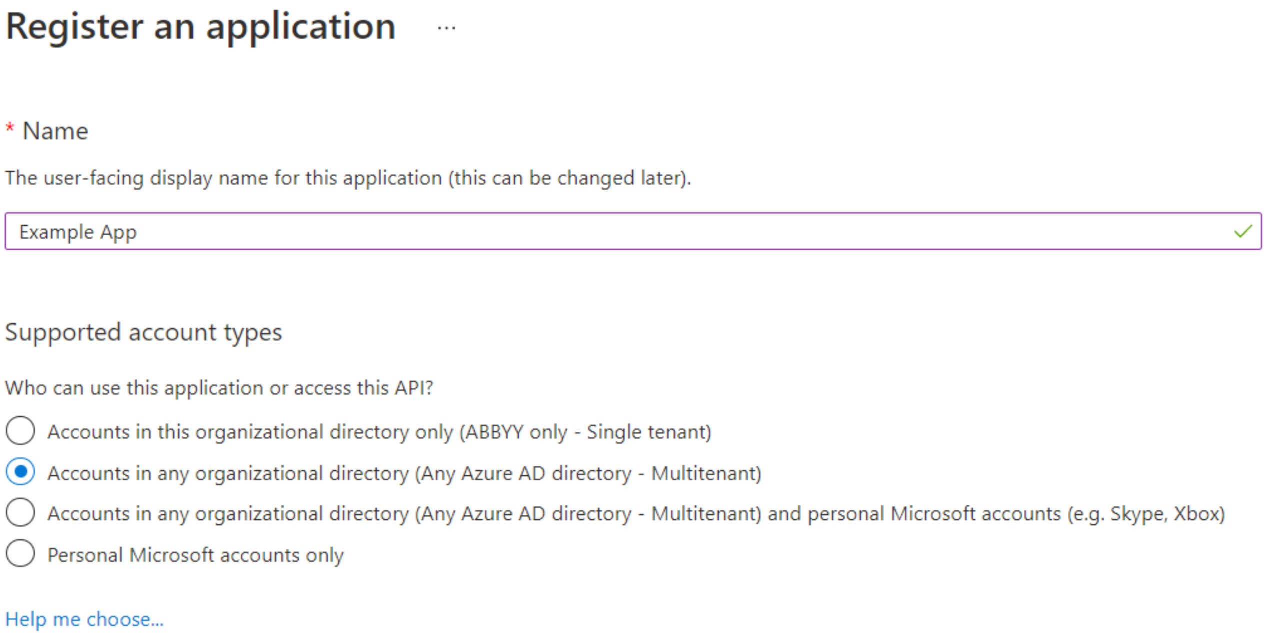
Registering the Application in Microsoft Azure

To create an application, an Azure Active Directory tenant with application registration and editing permissions is required.

You can switch to the correct directory on the Portal settings | Directories + subscriptions page.

Registering the application

1. Navigate to the App registrations page.
2. Click **New registration**.
3. Specify a name for your application and select the supported account types.



Register an application ...

* Name

The user-facing display name for this application (this can be changed later).

Example App ✓

Supported account types

Who can use this application or access this API?

○ Accounts in this organizational directory only (ABBYY only - Single tenant)

● Accounts in any organizational directory (Any Azure AD directory - Multitenant)

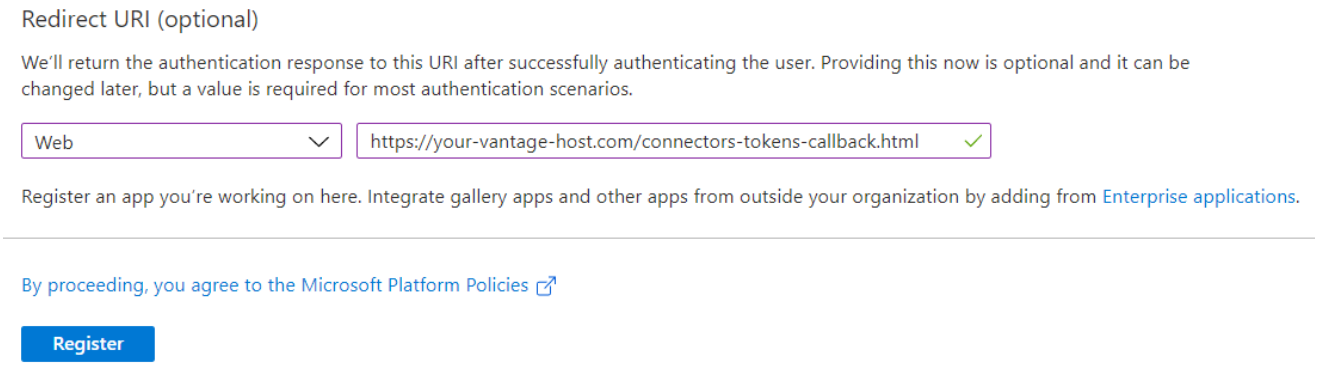
○ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

○ Personal Microsoft accounts only

Help me choose...

Note: If the Multitenant type is selected for the application, it will be available for users in any Azure AD tenant. Such applications need to be verified, which is only available for Microsoft Partner Network participants. If you are not a participant, select Single tenant, which will only make your app available to users in your own Azure AD tenant.

4. In the **Redirect URI** section, select the Web platform and specify the redirect URI: https://<Vantage host name>/connectors-tokens-callback.html.



Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web | https://your-vantage-host.com/connectors-tokens-callback.html ✓

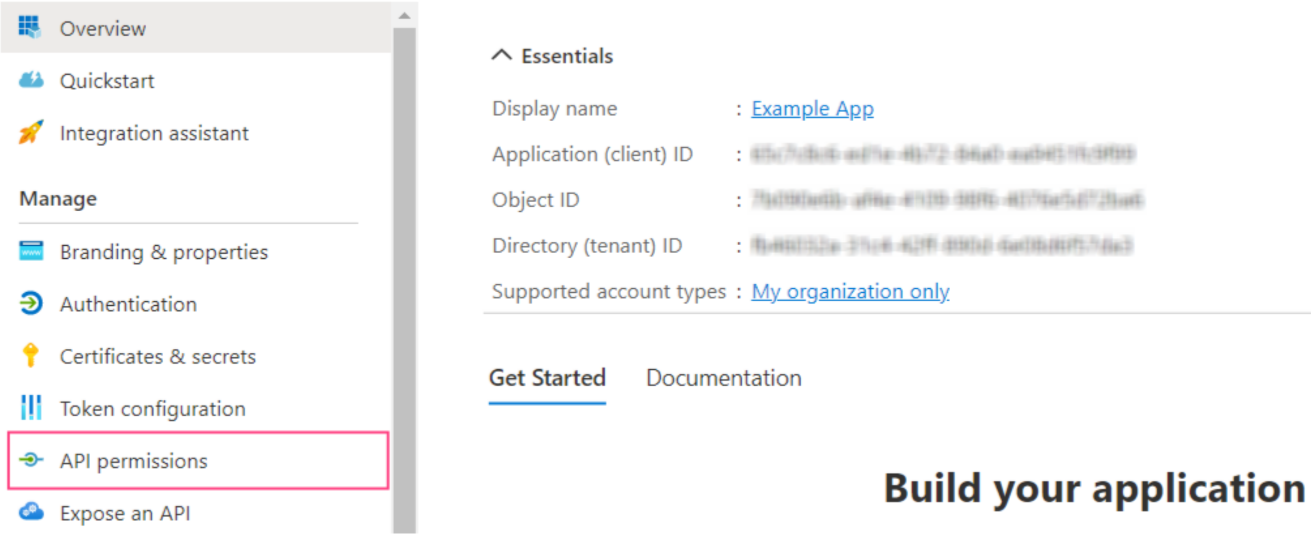
Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from Enterprise applications.

By proceeding, you agree to the Microsoft Platform Policies

5. Click **Register**.

Setting up application permissions

1. Navigate to the **API permissions** tab.



Essentials

Display name : Example App

Application (client) ID : [redacted]

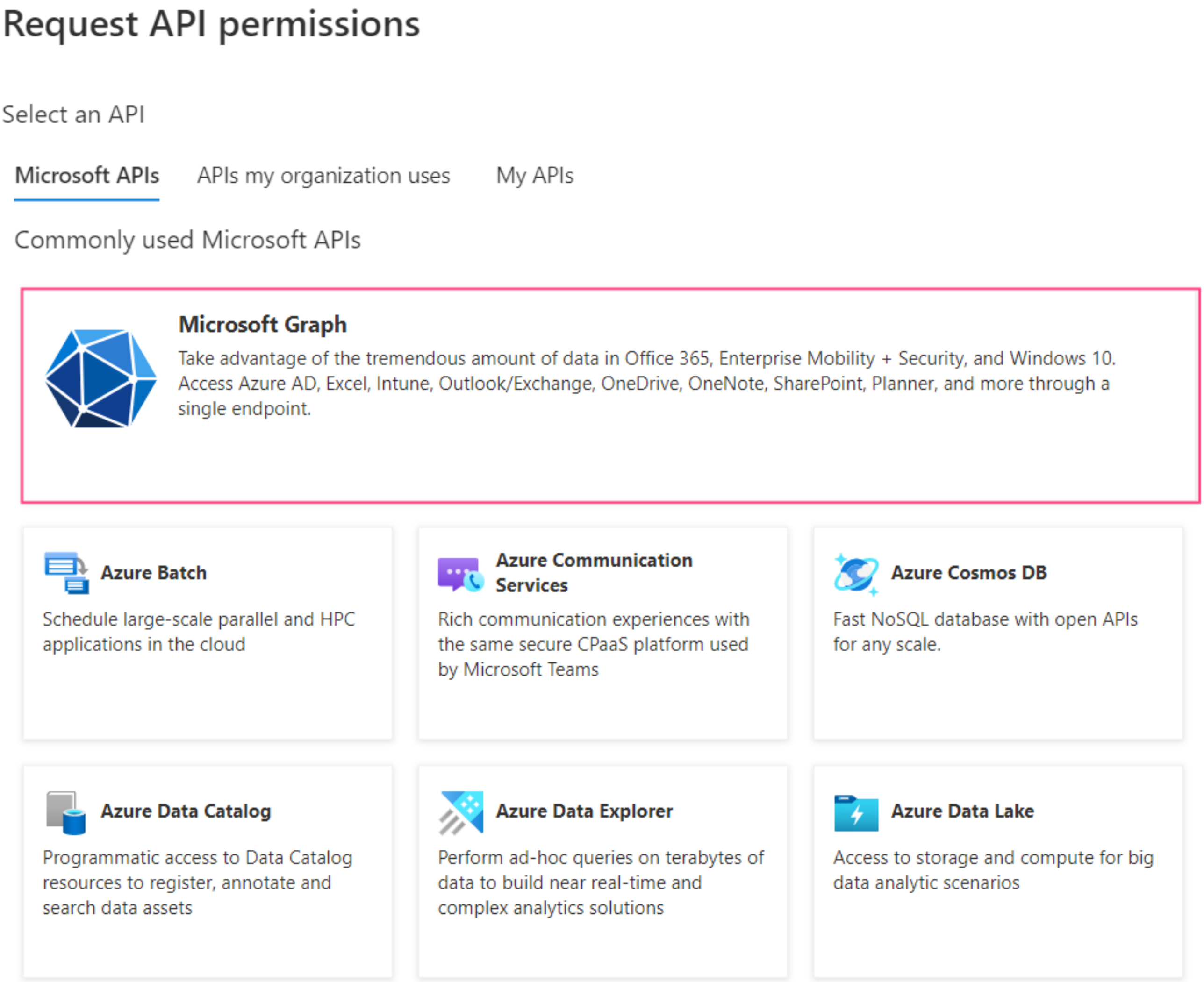
Object ID : [redacted]

Directory (tenant) ID : [redacted]

Supported account types : My organization only

2. Click **Add permission**.

3. In the dialog that will open, select the **Microsoft Graph** section.



Request API permissions

Select an API

Microsoft APIs APIs my organization uses My APIs

Commonly used Microsoft APIs

Microsoft Graph
Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.

Azure Batch
Schedule large-scale parallel and HPC applications in the cloud

Azure Communication Services
Rich communication experiences with the same secure CPaaS platform used by Microsoft Teams

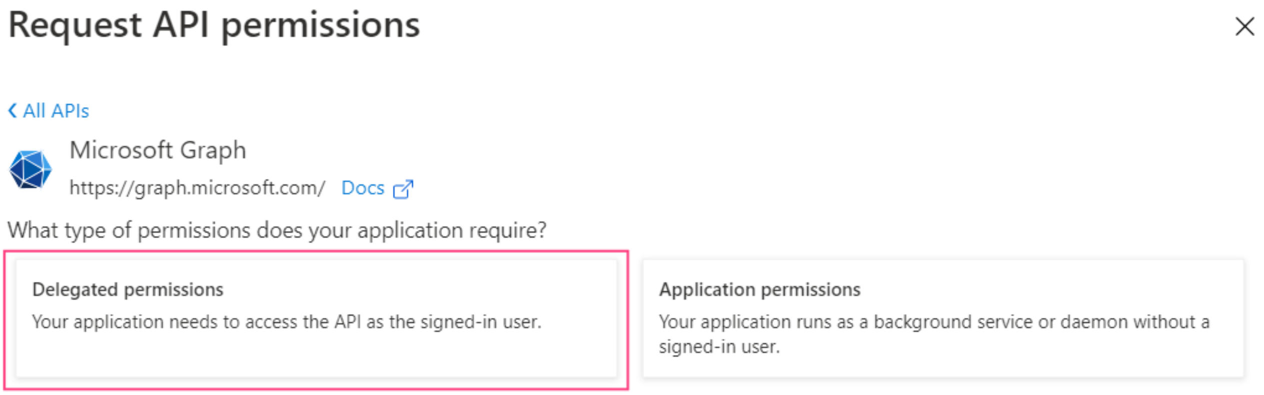
Azure Cosmos DB
Fast NoSQL database with open APIs for any scale.

Azure Data Catalog
Programmatic access to Data Catalog resources to register, annotate and search data assets

Azure Data Explorer
Perform ad-hoc queries on terabytes of data to build near real-time and complex analytics solutions

Azure Data Lake
Access to storage and compute for big data analytic scenarios

4. Select **Delegated permissions**.



Request API permissions

‹ All APIs

Microsoft Graph
https://graph.microsoft.com/ Docs

What type of permissions does your application require?

Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.

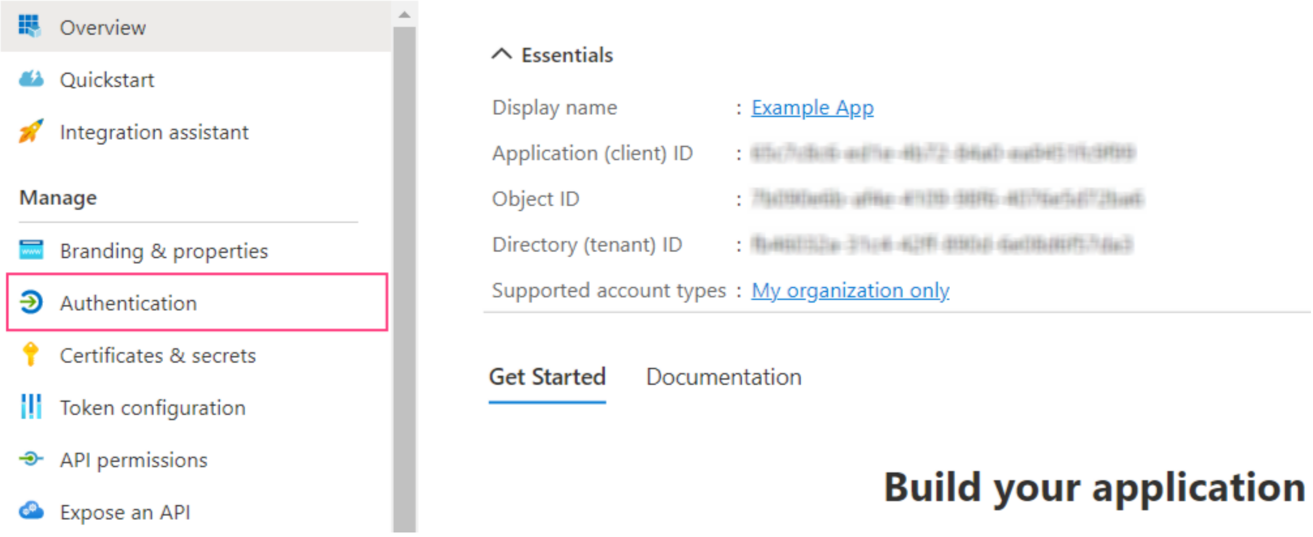
5. Add the following permissions:

- email
- IMAP.AccessAsUser.All
- offline_access
- openid
- profile

6. Click **Add permissions**. This will close the dialog and display the selected permissions.

Creating client secrets

1. Navigate to the **Authentication** tab.



Display name : Example App

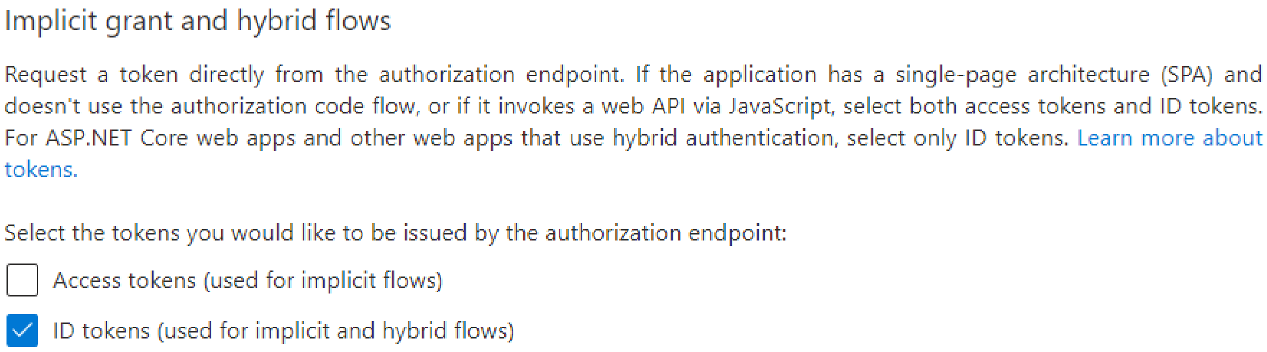
Application (client) ID : [redacted]

Object ID : [redacted]

Directory (tenant) ID : [redacted]

Supported account types : My organization only

2. In the **Implicit grant and hybrid flows** section, mark **ID tokens (used for implicit and hybrid flows)**.



Implicit grant and hybrid flows

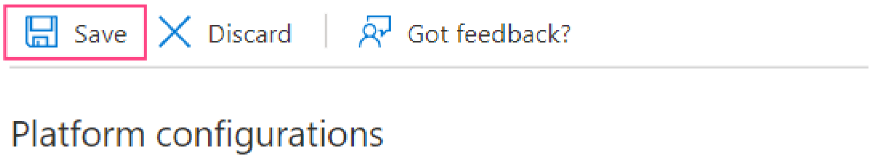
Request a token directly from the authorization endpoint. If the application has a single-page architecture (SPA) and doesn't use the authorization code flow, or if it invokes a web API via JavaScript, select both access tokens and ID tokens. For ASP.NET Core web apps and other web apps that use hybrid authentication, select only ID tokens. Learn more about tokens.

Select the tokens you would like to be issued by the authorization endpoint:

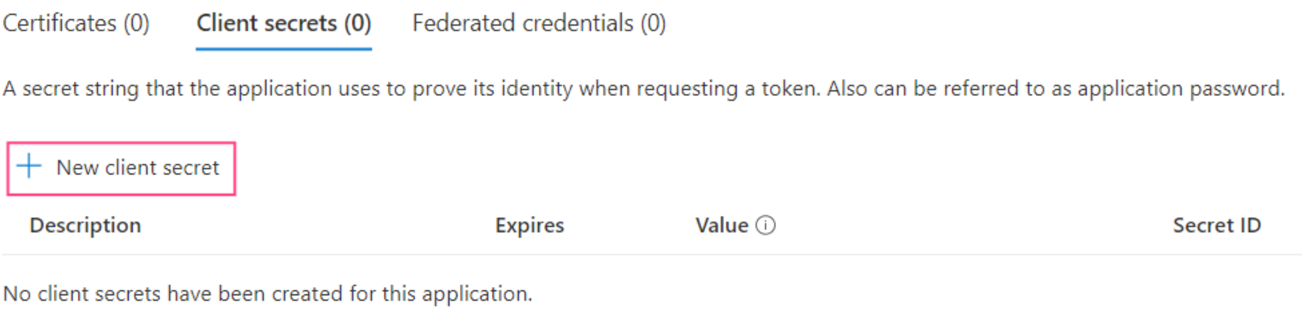
□ Access tokens (used for implicit flows)

☑ ID tokens (used for implicit and hybrid flows)

3. Click **Save** at the top of the screen.



4. Navigate to the **Certificates & secrets** tab and click **New client secret**.



A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

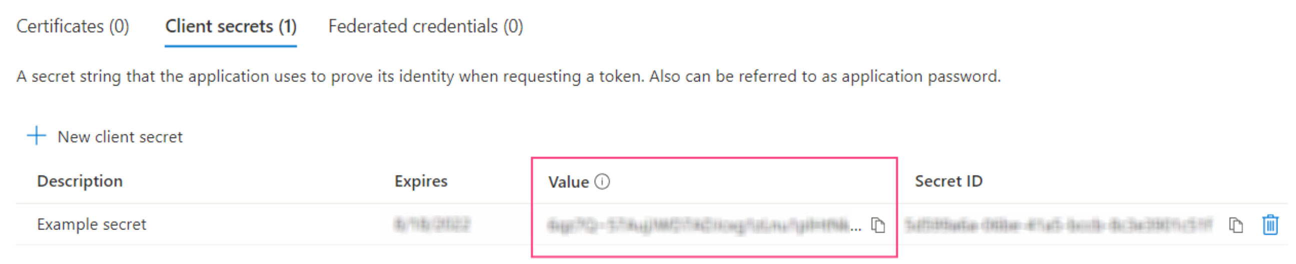
Description	Expires	Value ⓘ	Secret ID

No client secrets have been created for this application.

5. In the dialog box that will open, specify a name for the client secret and an expiration date.

Note: The maximum expiration date is 24 months.

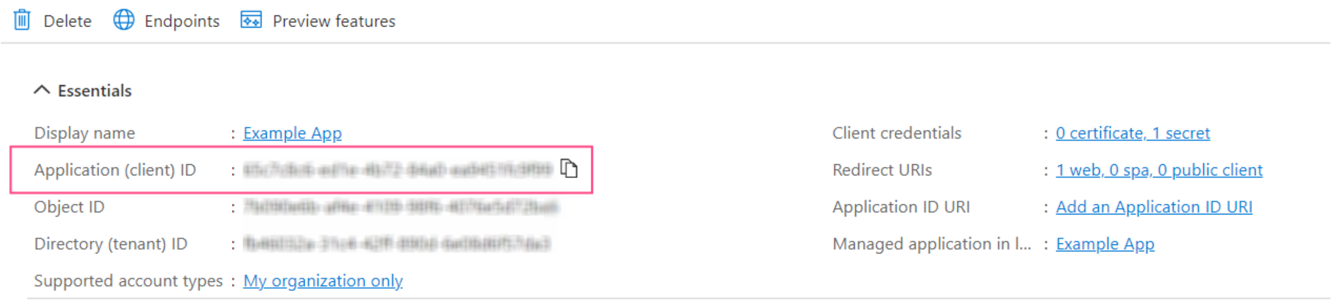
6. Click **Add**. This will close the dialog and display information about your new client secret. It is important that you copy and save the **Value**, since you will not be able to access it again once you close the page. **Value** is required when configuring the tokenmanagement service in Vantage.



A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value ⓘ	Secret ID
Example secret	[redacted]	[redacted]	[redacted]

You will also need a client identifier, which can be copied from the **Application (client) ID** field in the **Overview** tab. The copy icon will appear once you hover the mouse cursor over the value of the identifier.



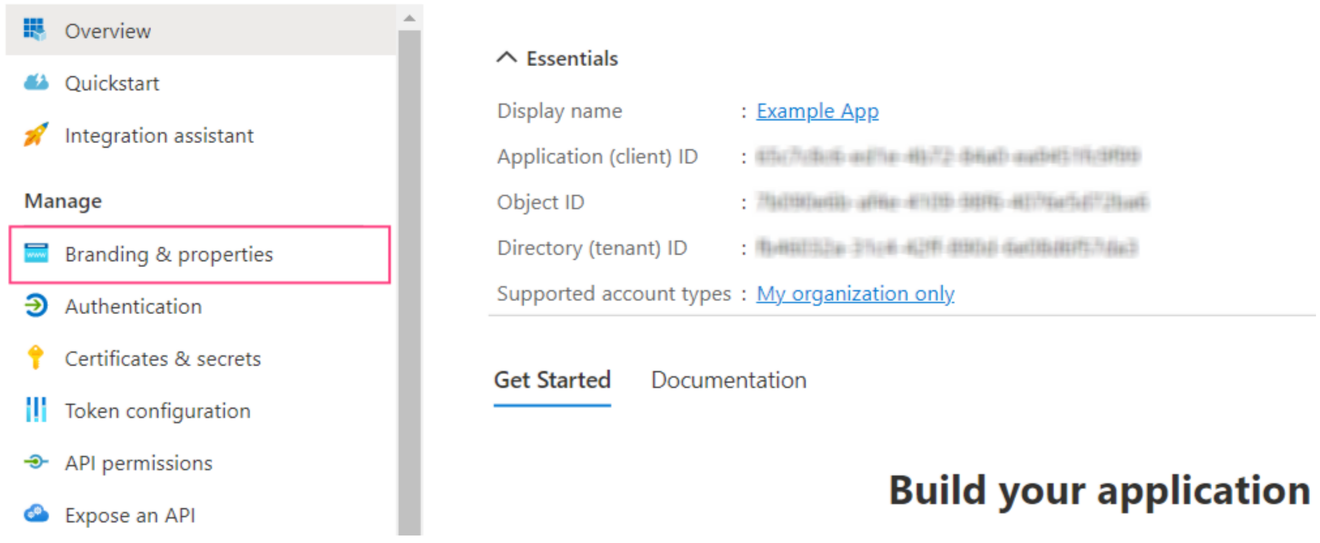
Display name	: Example App	Client credentials	: 0 certificate, 1 secret
Application (client) ID	: [redacted]	Redirect URIs	: 1 web, 0 spa, 0 public client
Object ID	: [redacted]	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: [redacted]	Managed application in l...	: Example App
Supported account types	: My organization only		

Verifying the application

To make the application available to users from any Azure AD tenant, verification is required. Verification is not required if accounts from a single Azure AD tenant are used.

Only Microsoft Partner Network participants can undergo verification.

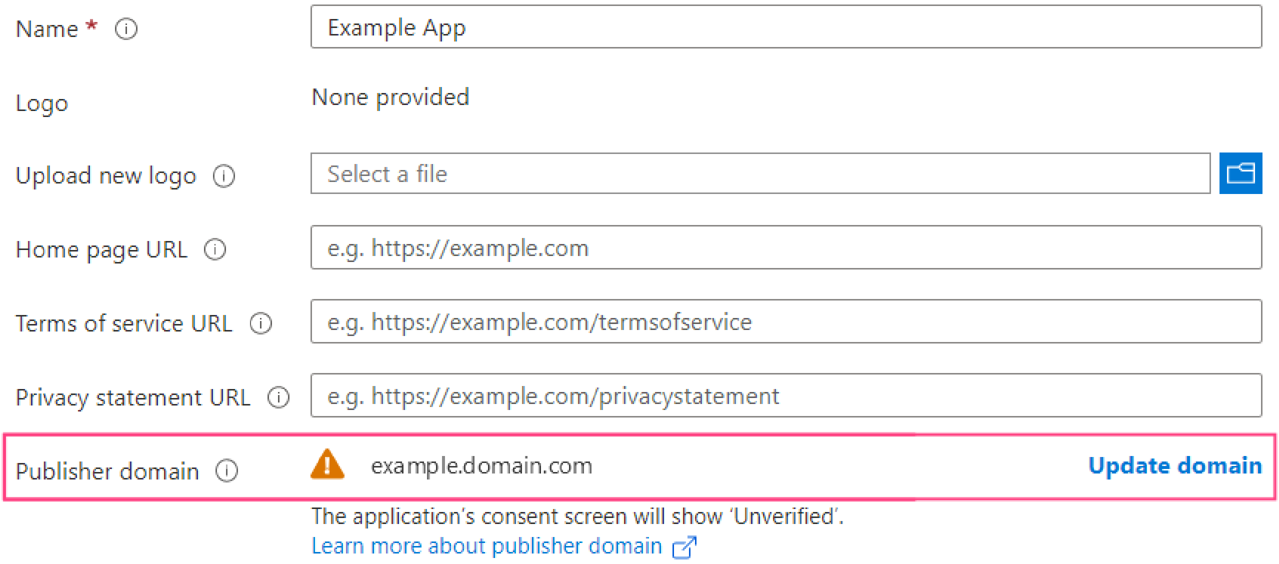
1. Navigate to the **Branding & properties** tab.



Display name	: Example App
Application (client) ID	: [redacted]
Object ID	: [redacted]
Directory (tenant) ID	: [redacted]
Supported account types	: My organization only

2. Verify that the domain is specified in the **Publisher domain** field. If required, configure your domain by clicking **Configure a domain**.

   The warning icon displayed next to the domain name means that an application with the specified domain cannot be verified. Click **Update domain** to specify a different valid domain related to the Azure Active Directory tenant. Alternatively, verify a new domain.



Name * ⓘ: Example App

Logo: None provided

Upload new logo ⓘ: Select a file

Home page URL ⓘ: e.g. https://example.com

Terms of service URL ⓘ: e.g. https://example.com/termsofservice

Privacy statement URL ⓘ: e.g. https://example.com/privacystatement

Publisher domain ⓘ: ⚠ example.domain.com Update domain

The application's consent screen will show 'Unverified'.

Learn more about publisher domain

3. In the **Publisher verification** section, specify your MPN ID and click **Verify and save**.

Note: If you do not have the required permissions to add an MPN ID, verify that all publisher verification requirements are satisfied.

Once your verification is successful, the appropriate icon will be displayed next to the **Publisher display name** field.

Passing Credentials to Consul

If Vantage is already installed, you need to use Consul to manually enter account credentials generated for Microsoft and/or Google email service authentication.

Features specific to the OAuth 2.0 protocol are listed in the TokenManagement service.

Note: Before setting up, verify that the kubectl command line tool is installed and that you are connected to the Kubernetes cluster.

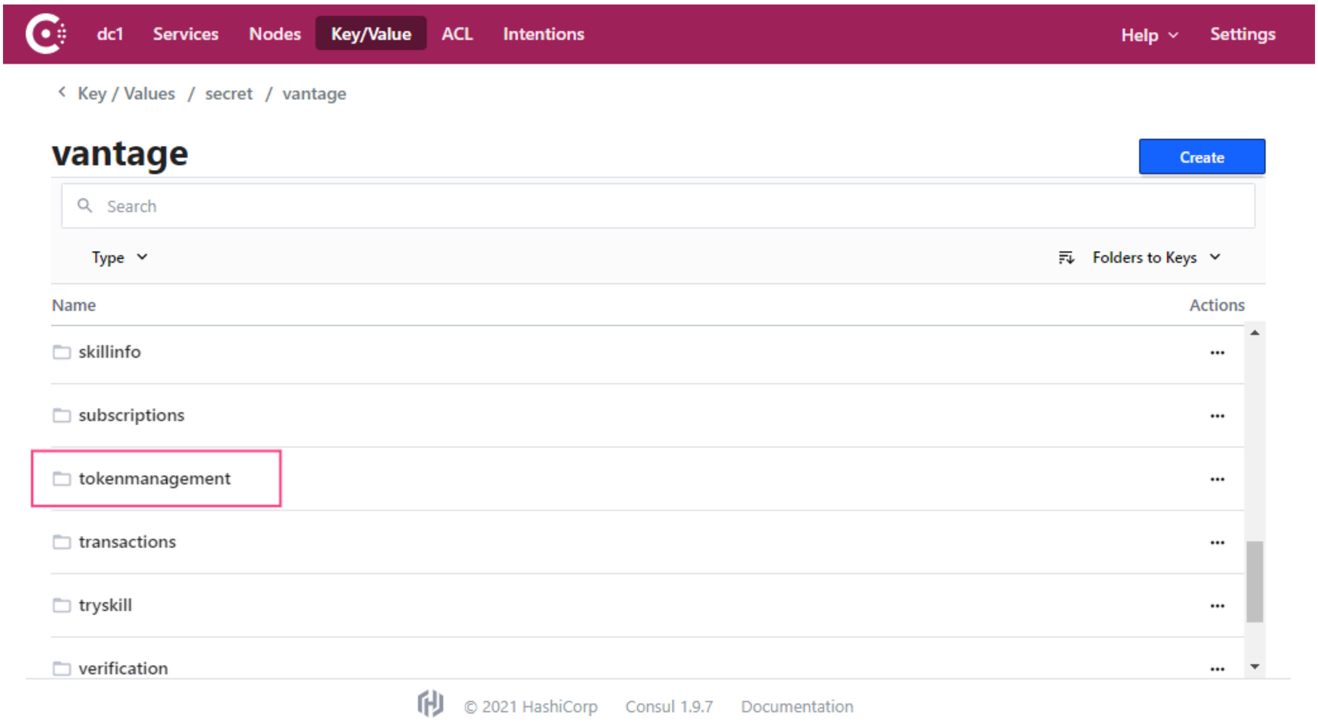
1. Get access to the Consul web interface by running the following command:

   ```
   kubectl port-forward -n abbyy-infrastructure service/consul-ui 8500:80
   ```

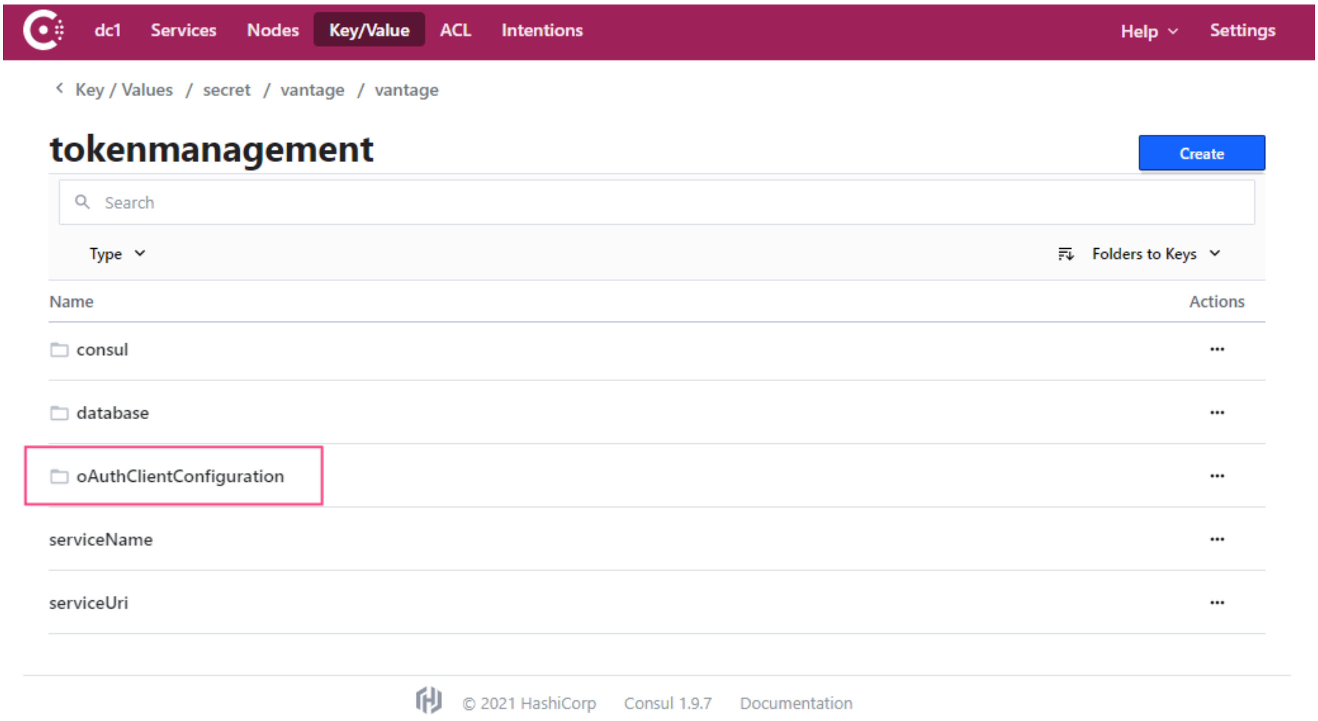
   Next, navigate to http://localhost:8500/ui/dc1/kv/secret/.

2. In the **Key/Value** tab that will open, select the appropriate Vantage deployment scope. Then, select the **vantage** project.

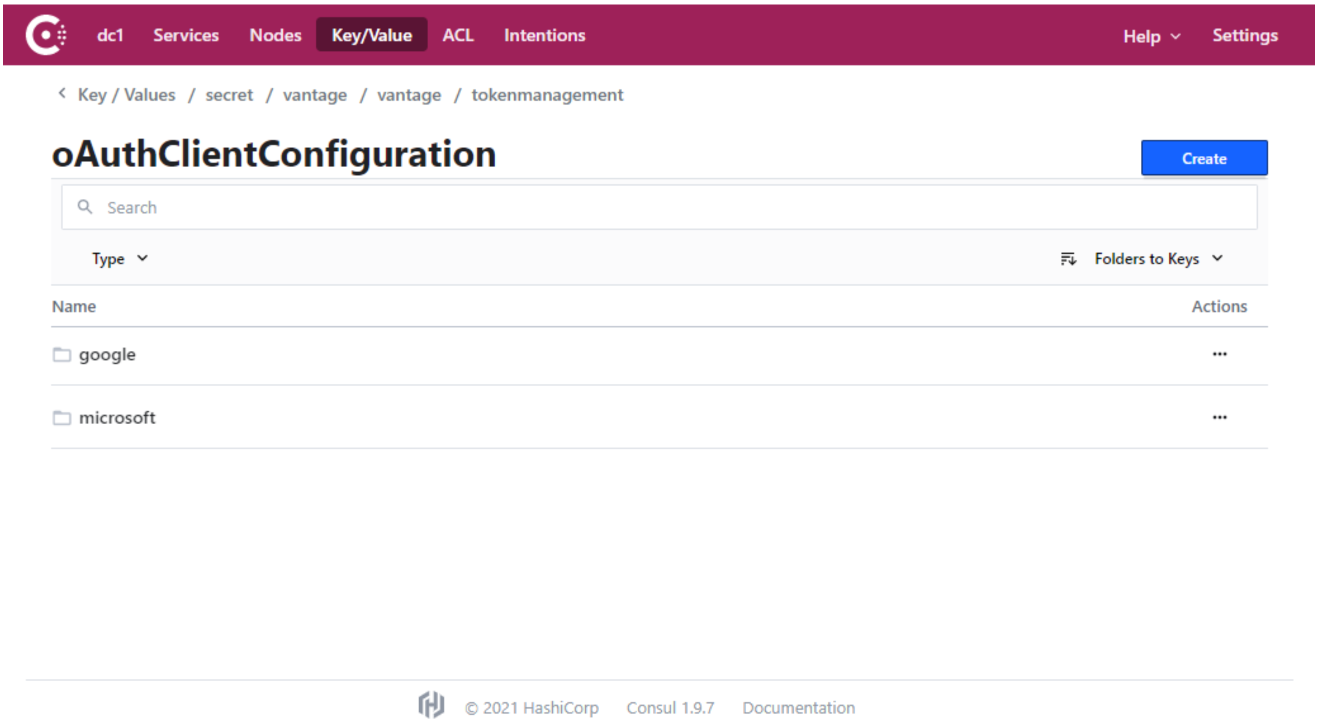
3. Select the **tokenmanagement** service.



4. Navigate to the **oAuthClientConfiguration** section.



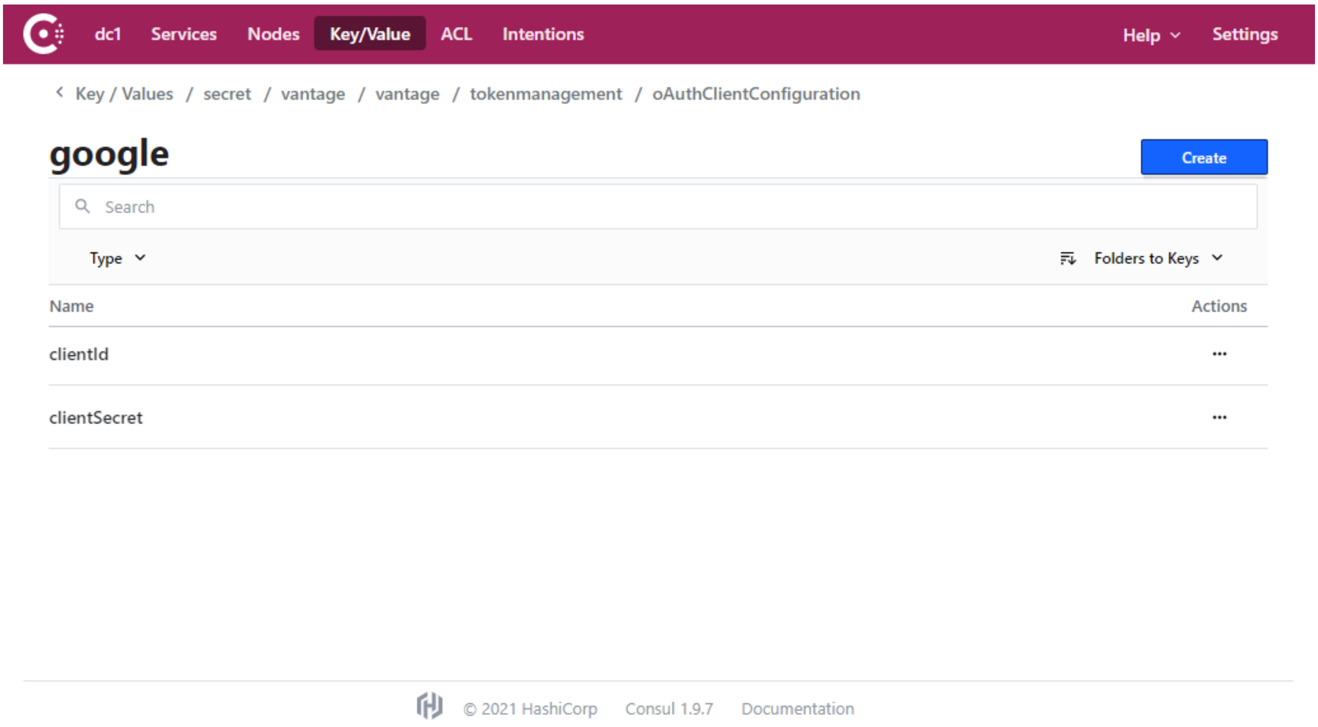
5. Select the service for which you want to specify user data.



oAuthClientConfiguration

Name	Actions
google	•••
microsoft	•••

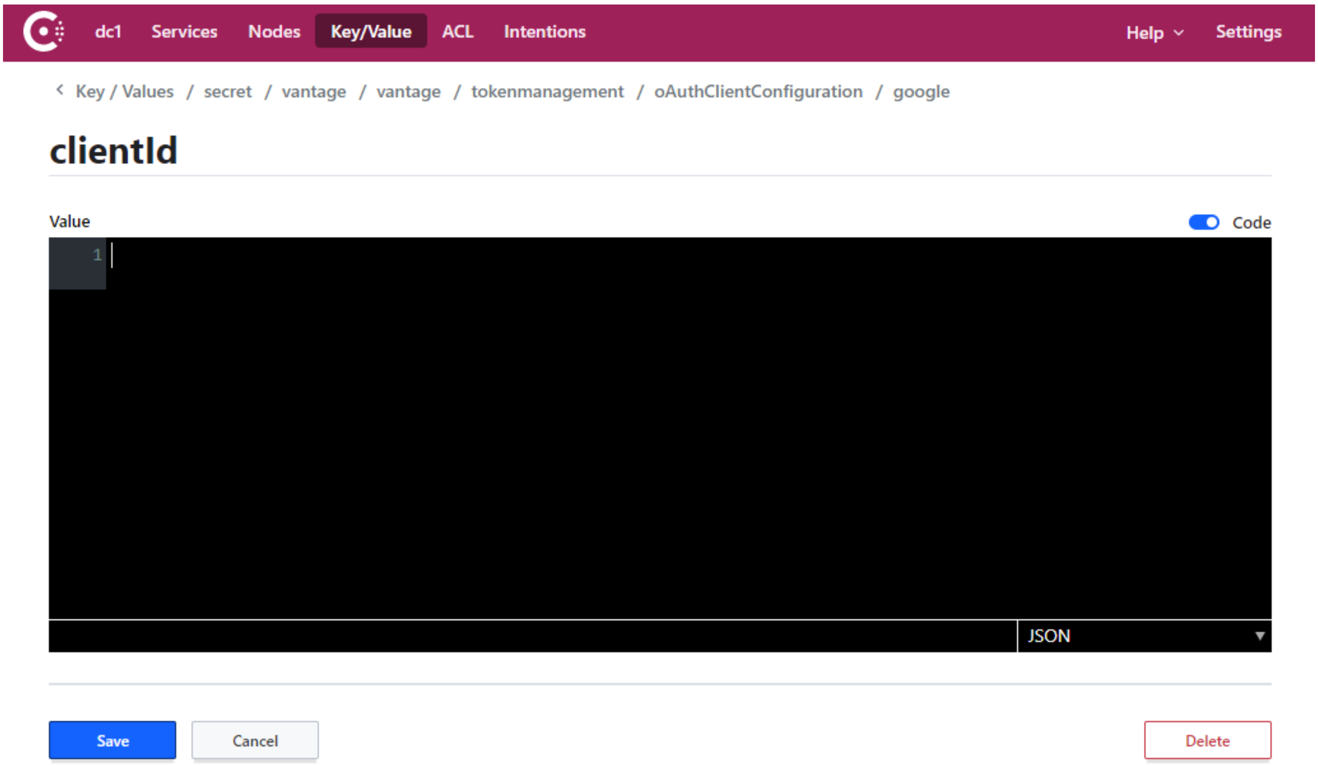
6. Select the **clientId** key.



google

Name	Actions
clientId	•••
clientSecret	•••

7. Copy and paste the Client ID value you saved earlier to the entry field and click **Save**.


dc1
Services
Nodes
Key/Value
ACL
Intentions
Help
Settings
Key / Values / secret / vantage / vantage / tokenmanagement / oAuthClientConfiguration / google
clientId
Value
Code
JSON
Save
Cancel
Delete

8. Repeat steps 6 and 7 for the **clientSecret** key.

If required, repeat steps 5 through 8 for a different email service.

9. Restart the **tokenmanagement** service by running the following command:

```
label=tokenmanagement
kubectl -n abbyy-vantage rollout restart $(kubectl -n abbyy-vantage get deployments –
l app.kubernetes.io/component=$label -o name)
```

Updating Client secret

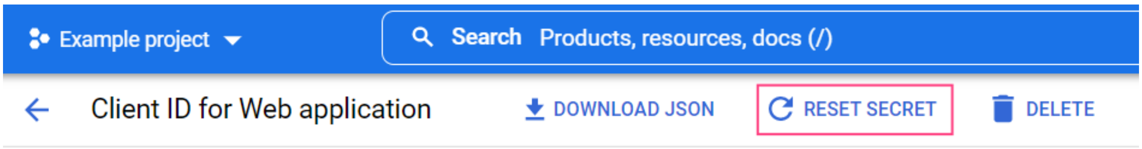
The Client secret value is used for serverside client identification and constitutes confidential information. For security purposes, data like this should periodically be updated. Some services like Azure Active Directory limit the validity period for such data.

Once a new Client secret has been created, the value of the corresponding Consul key should also be updated.

Note: Once Client secret has been updated, users will need to set up connections to their email service in the Input activity of the Document Skill from scratch. Otherwise, Vantage will not be able to connect to the mailbox and import emails from it.

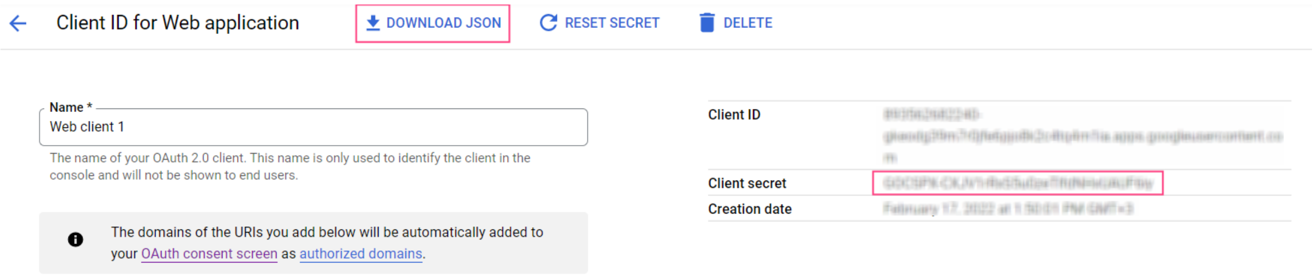
Updating Client secret in Google

1. Navigate to the Google Cloud Console and select the appropriate project.
2. In the menu on the left, select **APIs & Services** > **Credentials**.
3. In the **OAuth 2.0 Client IDs** section, select the identifier used to authenticate when connecting to the IMAP server.
4. Click **Reset secret**.


Example project
Search Products, resources, docs (/)
Client ID for Web application
DOWNLOAD JSON
RESET SECRET
DELETE

5. Click **Reset** in the pop-up dialog box. This will update the Client secret value and recall its previous value.

6. Download the JSON file containing the credentials. Alternatively, copy the Client secret value from the right side of the screen.



Name *
Web client 1

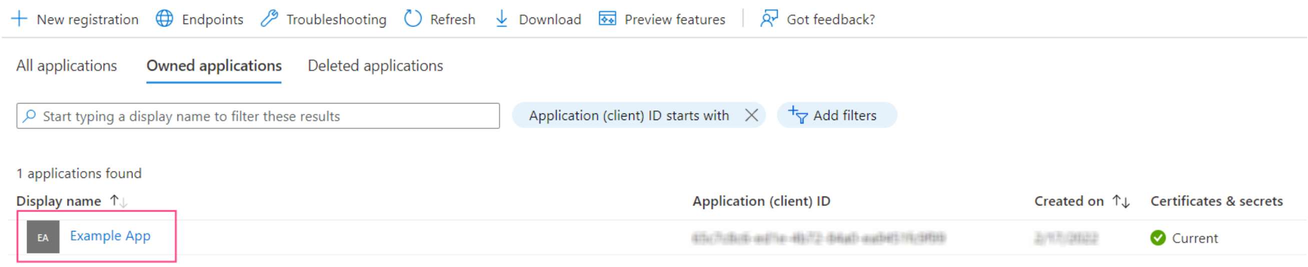
The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorized domains.

Client ID	[redacted]
Client secret	[redacted]
Creation date	[redacted]

Updating Client secret in Microsoft Azure

1. Navigate to the [App registrations]() page and select the application used for authentication using the IMAP server.



1 applications found

Display name	Application (client) ID	Created on	Certificates & secrets
EA Example App	[redacted]	[redacted]	Current

2. Navigate to the **Certificates & secrets** tab and click **New client secret**.

3. In the dialog box that will open, specify a name for the client secret and its expiration date.

4. Click **Add**. This will close the dialog and display information about the new client secret. It is important that you copy and save the **Value**, since you will not be able to access it again once you close the page.

5. If the current client secret has not expired yet, you can delete it in order to only be able to use the new client secret to identify the client.

Updating Client secret in Consul

Follow the steps listed in the [Passing Credentials to Consul]() section, omitting steps 6 and 7 (copying the **clientId** value).

E-mail Template Modification

Default e-mail texts sent by Vantage include ABBYY-specific information, for example: ABBYY customer support e-mails, GDPR notices from ABBYY, footers with ABBYY copyrights, etc. You can modify the templates for these e-mails to include your own details and make them specific to your company.

Vantage offers the following e-mail templates:

Email template	ID	Culture	Subject Template Parameters	Body Template Parameters
Invitation e-mail that is sent to the tenant administrator of a newly created tenant	E6F03F64-B982-49C6-B336-251CA6C45FFE	en	• productName	• userName - the name of the user to whom the e-mail is being sent, • productName - the name of the product, • invitationUri - the invitation link to create account.
Invitation e-mail that is sent to the newly created tenant	6C5DBE28-0A2C-4EB7-82DC-04173DC75418	en	• productName	• userName - the name of the user to whom the e-mail is being sent, • productName - the name of the product, • invitationUri - the invitation link to create account.
E-mail with a password reset link requested by a user on the Vantage login page	701B077A-20EA-42B8-A71E-AB3EA5996039	en	N/A	• displayName - optional display username provided during registration, • callbackUrl - the link to reset password, • expireHours - the number of hours until the link will expire.
E-mail about a successful skill export to a shared folder	B5A03F64-B982-49C6-B336-251CA6C45FFE	en	N/A	• userName - the user e-mail address, • skillName - the skill that is being exported, • folderPath - the path to the shared folder, • login - the SFTP folder access login,

Email template	ID	Culture	Subject Template Parameters	Body Template Parameters
				• password - the SFTP folder access password.
E-mail about an unsuccessful skill export to a shared folder	F3C4BD68-B9FF-439F-A719-5B4F62263C4E	en	N/A	• userName - the user e-mail, • skillName - the skill that is being exported, • errorMessage - the error message body.

To modify an e-mail template, follow the steps below:

1. Get access to the Vantage mail service.
2. Get the e-mail template details.
3. Update the e-mail template.

Getting access to the Vantage mail service

To get access to the mail service, do the following:

1. Get the access to the Mail service API through http://localhost:8080 or other port:

Bash

```
kubectl -n abbyy-vantage port-forward $(kubectl get service -n abbyy-vantage --
selector='app.kubernetes.io/name=mail' -o name) 8080:80
```

Terminal output sample

```
bash-4.4# kubectl -n abbyy-vantage port-forward $(kubectl get service -n abbyy-vantage
--selector='app.kubernetes.io/name=mail' -o name) 8080:80
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
```

Getting the e-mail template details

To get the markup for an existing e-mail template, send a **GET** request to the **templates** resource (http://localhost:8080) as follows:

```
GET http://localhost:8080/api/v1/templates/templateId
```

Sample response

```
{
    "localizedTemplates": {
        "en": {
            "subjectTemplate": "Your invitation to {{productName}}",
            "bodyTemplate": "<body>...</body>"
        }
    },
    "id": <templateId>,
    "createTime": "2022-04-06T07:23:52.903261+00:00",
    "updateTime": "2022-04-06T07:23:53.137142+00:00"
}
```

Note: The response does not contain attachments, which are instead are in the e-mail template. They must be specified and can additionally be modified when updating the template.

Updating an e-mail template

To replace an email template, send a **PUT** request to the **templates** resource:

```
PUT http://localhost:8080/api/v1/templates/templateId
```

Execute the following command:

```
{
    "culture": "en",
    "subjectTemplate": "subject template",
    "bodyTemplate": "body template",
    "attachments": [
        {
            "contentType": "attachment MIME",
            "contentId": "guid",
            "fileName": "image name used in body as cid",
            "content": "file content in base64 string"
        }
    ]
}
```

Template elements

Field	Type	Description
templateId	GUID	Identifier of the e-mail template to be modified.
culture	string	Template language. Currently, only English (en) is supported.
subjectTemplate	string	E-mail header text template. See the previous section for information on how to get the current value.
bodyTemplate	string	E-mail body template.

Field	Type	Description
		See the previous section for information on how to get the current value.
attachments[].contentType	string	Identifier of a particular attachment. The identifier must be unique for each attachment within the same template.
attachments[].contentId	string	Unique string.
attachments[].fileName	string	Attachment file name, can be used in the body as CID.
attachments[].content	base64string	Attachment file encoded in base64 format.

Note: The subject and body template parameters must remain unchanged. Do not add, remove, or modify them.

Note: Always include all attachments, even if you don't intend to modify them. The request completely replaces the email template.

Every Vantage e-mail template contains these four attached images:

cid	MIME-type	image
logo-vantage-logo-normal.png	image/png	ABBYY Vantage
600-px-copy-6.png	image/png	
16-headset-16@2x.png	image/png	🎧
16-global-outline-16@2x.png	image/png	🌐

▾ Sample request

```
PUT http://localhost:8080/api/v1/templates/E6F03F64-B982-49C6-B336-251CA6C45FFE

 {
    "culture": "en",
    "subjectTemplate": "Your {{productName}} account information",
    "bodyTemplate": "<body><table ...>...<img ... src="cid:logo-vantage-logo-
normal.png" alt="ABBYY Vantage" />...Dear {{userName}},...</body>",
    "attachments": [
        {
            "contentType": "image/png",
            "contentId": "4dcc3114-b7ff-48b5-902a-8fbd673d6acd",
            "fileName": "logo-vantage-logo-normal.png",
            "content":
"iVBORw0KGgoAAAANSUhEUgAAALoAAAAcCAMAAADhlVUwAAAABGdBTUEAALGPC/xhBQAAAAFzUkdCAK7OHOkAA
ACEUExURUdwTAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA..."
        },
        {
            "contentType": "image/png",
            "contentId": "e243efa2-55ed-4f07-a1e0-27d55460decc",
            "fileName": "600-px-copy-6.png",
            "content":
"iVBORw0KGgoAAAANSUhEUgAAAlgAAACkCAMAAAB8d6ClAAAABGdBTUEAALGPC/xhBQAAAAFzUkdCAK7OHOkAA
AJnUExURWSR8lgs/Vk0/l5a+P8gOGJ89GWZ8GaU8mKA9Fgy/Vx..."
        } ,
        {
            "contentType": "image/png",
            "contentId": "d57f26bb-43b6-41a3-b356-3f53dfbd28d7",
            "fileName": "16-headset-16@2x.png"
            "content":
"iVBORw0KGgoAAAANSUhEUgAAACAAAAAgCAMAAABEpIrGAAAABGdBTUEAALGPC/xhBQAAAAFzUkdCAK7OHOkAA
AAzUExURUdwTAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA..."
        } ,
        {
            "contentType": "image/png",
            "contentId": "df64ab94-9c7f-49d1-93bf-f7ba48eb2a98",
            "fileName": "16-global-outline-16@2x.png",
            "content":
"iVBORw0KGgoAAAANSUhEUgAAACAAAAAgCAMAAABEpIrGAAAABGdBTUEAALGPC/xhBQAAAAFzUkdCAK7OHOkAA
AA/UExURUdwTAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...."
        }
    ]
}
```

You can get the base64 string representation of a file by executing the following command:

PowerShell 7

```
[Convert]::ToBase64String([IO.File]::ReadAllBytes("full path to file"))
```

Monitoring and Administration

As a system administrator, you are tasked with monitoring ABBYY Vantage at all times, managing it, discovering any errors that may occur during document processing, as well as the causes of such errors.

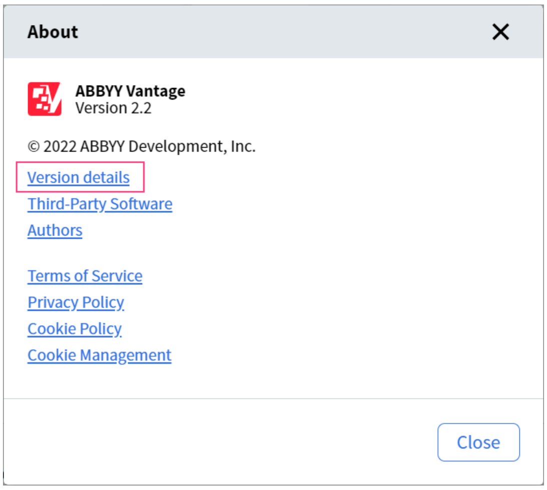
You can do this by using:

- Vantage log files.
- The built-in **Skill Monitor** service, which collects statistics for existing Vantage skills and provides detailed information regarding completed and ongoing transactions. This service also lets you get transaction event information required by technical support.

- A range of more advanced third-party services which allows you to: monitor internal Vantage processes, monitor specific workflows, analyze collected data to further fine-tune and optimize document processing, collect and analyze logs.

When contacting technical support, in addition to information about errors, you can also provide the version of the product and its components by doing the following:

1. Click **About** on the left pane and select **Version details**.



About

ABBYY Vantage
Version 2.2

© 2022 ABBYY Development, Inc.

2. Copy the details.

Log Files

Logs in Azure Files

To access the logs stored in Azure files, run the following command:

```
PV=$(kubectl -n abbyy-monitoring get pvc fluentd-pvc -o jsonpath='{.spec.volumeName}')
kubectl -n abbyy-monitoring get pv $PV -o jsonpath='{.spec.csi.volumeHandle}'
```

The response will specify the location where the logs are being stored. The resource group name, storage account name, and fileshare name are separated using the # character.

Example

```
mc_cluster_cluster_westeurope#fd100c1d01b43456d84dce8#pvc-5c78dc53-6508-4af0-9c2f-
3d308a04c374###abbyy-monitoring
```

If you want to connect the fileshare as a network drive, click **Connect** in the appropriate file sharing window in the Azure portal and follow the instructions.

For more information, see here.

Logs in shared folders in Azure Files

To access the logs stored in shared folders in Azure files, do the following:

1. Navigate to the storage account created for shared folders (**s3storage.sharedfolder.accessKey** parameter in the env.specific file).

2. Open **Files** and choose the folder named **sharedfolder**.

3. For Vantage logs, navigate to a folder named **abbyy-vantage** (**platform_namespace** parameter in the env_specific file).
4. Logs are stored in compressed gzip files with names in the Y-M-DD-H format (example, 2022-12-09-0800.log.gz). Copy files related to the time period when the problem occurred.
5. Send the files to ABBYY technical support.

If you want to connect the fileshare as a network drive, click **Connect** in the appropriate file sharing window in the Azure portal and follow the instructions.

For more information, see here.

Elasticsearch and Kibana

Elasticsearch and Kibana are tools for searching, analyzing, and visualizing logs. Elasticsearch and Kibana are not installed together with ABBYY Vantage and have to be installed and set up separately. You can use any existing installation.

Note: The installation and configuration of Elasticsearch and Kibana must be done before installing the product.

Note: The sample setup procedure below has been simplified and is provided only as an example.

To install Elasticsearch and Kibana, do the following:

1. Clone the repository:

```
git clone https://github.com/elastic/cloud-on-k8s.git
cd cloud-on-k8s
git checkout 2.5
cd deploy/eck-operator
```

2. Install the operator that deploys the resources:

```
helm -n elastic upgrade -i eck-operator . --create-namespace
```

3. Create a file named elastic.yaml. Copy and paste the following code into the file and save it:

```
cat << "EOF" > elastic.yaml
apiVersion: elasticsearch.k8s.elastic.co/v1
kind: Elasticsearch
metadata:
  name: elasticsearch
  namespace: elastic
```

```
spec:
  version: 8.5.1
  nodeSets:
    - config:
        indices.fielddata.cache.size: 38%
        xpack.ml.enabled: false
        xpack.security.enabled: true
      count: 3
      name: default
      podTemplate:
        spec:
          containers:
            - name: elasticsearch
              resources:
                limits:
                  memory: 1Gi
                  cpu: '1'
                requests:
                  cpu: '1'
                  memory: 1Gi
          initContainers:
            - command:

                - sh
                - '-c'
                - sysctl -w vm.max_map_count=262144
              name: sysctl
              securityContext:
                privileged: true
                runAsUser: 0
          nodeSelector:
            kubernetes.io/os: linux
      volumeClaimTemplates:
        - metadata:
            name: elasticsearch-data
          spec:
            accessModes:
              - ReadWriteOnce
            resources:
              requests:
                storage: 128Gi
            storageClassName: default
EOF
```

4. Create a file named kibana.yaml. Copy and paste the following code into the file and save it:

```
cat << "EOF" > kibana.yaml
apiVersion: kibana.k8s.elastic.co/v1
kind: Kibana
metadata:
  name: kibana
  namespace: elastic
spec:
  version: 8.5.1
  count: 1
  elasticsearchRef:
    name: elasticsearch
  podTemplate:
    spec:
      containers:
      - name: kibana
        env:
          - name: NODE_OPTIONS
            value: "--max-old-space-size=2048"
        resources:
          requests:
            memory: 512Mi
            cpu: 0.5
          limits:
            memory: 1Gi
            cpu: 1
      nodeSelector:
        kubernetes.io/os: linux
EOF
```

5. Run the following command to install Elasticsearch:

```
kubectl -n elastic apply -f elastic.yaml
```

Check the deployment status:

```
kubectl -n elastic get statefulset
```

6. Run the following command to install Kibana:

```
kubectl -n elastic apply -f kibana.yaml
```

Check the deployment status:

```
kubectl -n elastic get deployment
```

7. Get the password for an Elasticsearch user:

```
kubectl -n elastic get secret elasticsearch-es-elastic-user -o go-
template='{{.data.elastic | base64decode }}'
```

8. Place the values of the following parameters in your env_specific.yaml file:

```
logging:
  enabled: true
  elasticsearch:
    enabled: true
    host: elasticsearch-es-http.elastic.svc.cluster.local
    username: elastic
    password: elastic_user_password
    scheme: https
```

Grafana

Grafana is a tool for visualizing, monitoring, and analyzing data. Grafana is not installed together with ABBYY Vantage and has to be installed and set up separately. You can use any existing installation.

Note: The sample setup procedure below has been simplified and is provied only as an example.

To install Grafana, do the following:

1. Create a file named grafana.yaml.
2. Copy and paste the following code into the file and save it:

```
persistence:
  enabled: false

rbac:
  create: true
  namespaced: false
serviceAccount:
  create: true

podLabels:
  app.kubernetes.io/component: grafana
nodeSelector:
  kubernetes.io/os: linux

adminUser: admin
adminPassword: password

plugins:
  - grafana-piechart-panel
  - flant-statusmap-panel

grafana.ini:
  server:
    root_url: "%(protocol)s://%(domain)s:%(http_port)s/grafana/"
    enable_gzip: "true"
```

```
ingress:
  enabled: true
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/rewrite-target: /$2
  tls:
    - secretName: platform-wildcard
      hosts:
        - {{ env }}.{{ domain }}
  hosts:
    - {{ env }}.{{ domain }}
  path: "/grafana(/|$)(.*)"

sidecar:
  dashboards:
    enabled: true
    label: grafana_dashboard

datasources:
  datasources.yaml:
    apiVersion: 1
    datasources:
    - name: Prometheus
      editable: true
      isDefault: true
      jsonData:
        timeInterval: 5s
        tlsSkipVerify: true
      type: prometheus
      url: 'http://prometheus-scaling.abbyy-monitoring.svc.cluster.local:9090
```

Replace the host parameter value with the domain name of your Vantage cluster and change the initial administrator password.

3. Run the following command:

```
helm repo add grafana https://grafana.github.io/helm-charts
helm -n abbyy-monitoring upgrade -i  grafana grafana/grafana -f values.grafana.yaml
```

ABBYY Vantage © 2022 ABBYY Development, Inc.
ABBYY, ABBYY Vantage, Vantage are either registered trademarks or trademarks of ABBYY Development Inc. and/or its affiliates in the USA or other countries. These designations can also be logos, product or company names (or part of any of the above) of ABBYY Development Inc. and/or its affiliates and may not be used without consent of their respective owners.

Information in this document is subject to change without notice and does not bear any commitment on the part of ABBYY.

The software described in this document is supplied under a license agreement. The software may only be used or copied in strict accordance with the terms of the agreement. It is a breach of the United States copyright law and international laws to copy the software onto any medium unless specifically allowed in the license agreement or nondisclosure agreements.

No part of this document may be reproduced or transmitted in any from or by any means, electronic or other, for any purpose, without the express written permission of ABBYY.