Amplify Central

6 April 2020

API Gateway Agents

This documentation describes the following Axway software:

AMPLIFY Central API Gateway

Contents

AMPLIFY Central and Axway API Manager connected overview 1

Understand why you would want a connected / managed environment for AMPLIFY Central and Axway API Manager. Learn how you can govern and monitor the creation / deployment / publishing and subscriptions of AMPLIFY Central and Axway API Manager hosted APIs in one central location.

Note The Axway API Manager connectivity to AMPLIFY Central is currently available in an alpha review mode; current functionality and configuration may change before release. Therefore, this connectivity is available for trial use only and is not supported for production API management or connectivity.

What is Axway API Manager connected?

Connecting Axway API Manager to AMPLIFY Central enables you to have a connected environment for Axway API Gateway where two agents (Discovery and Traceability) are used with Axway API Gateway to:

- Create a new API Gateway environment that can generate configurations for agents, allowing them to interact with AMPLIFY Central.
- Detect changes in API Manager deployments using the Discovery Agent. The Discovery Agent pushes the service configuration as an API service for the environment, which can then be published as a catalog item to be used by consumers to subscribe to the service.
- Create / Update a subscription in API Gateway that is associated with the service and API key and is used by AMPLIFY Central.
- Filter the Axway API Manager logs that are related to discovered APIs and prepare the transaction events that are sent to AMPLIFY platform.

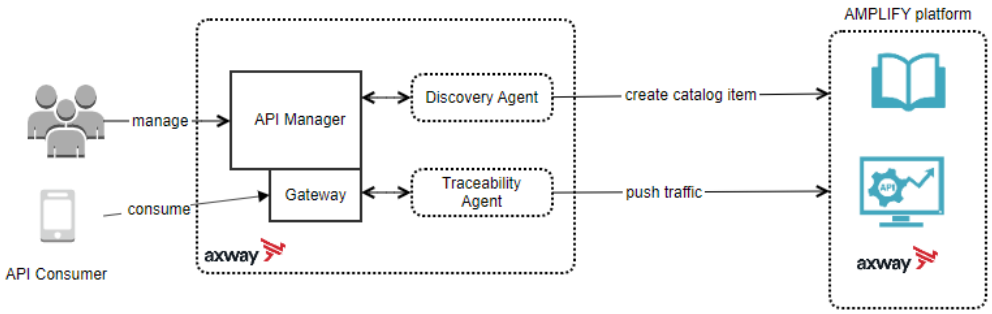
Discovery Agent

The Discovery Agent is used to discover new published API. The Discovery Agent pushes either an OAS3 or Swagger2 spec to AMPLIFY Central (depending on which was used to create the backend proxy in API Manager).

The Discovery Agent discovers APIs that have PassTrough / API Key / OAuth security.

The related APIs are published to AMPLIFY Central in either disconnected mode (catalog item publication) or connected mode (API Service publication). For additional information, see *Discovery Agent* on page 10.

Note Although both publication modes are functional, APIs cannot be fully managed from AMPLIFY Central before Q3-2020.



Traceability Agent

The Traceability Agent is used to filter the logs and prepare the transaction events that are sent to AMPLIFY Central.

Minimum requirements

- An AMPLIFY Central Service Account. See https://docs.axway.com/bundle/axway-open-docs/page/docs/central/cli_proxy_flow/index.html
- Axway API Manager / Axway API Gateway versions 7.6.2 SPx, 7.7 SPx or 7.8

Connect Axway API Manager to AMPLIFY Central quickstart

1. Create an environment object in AMPLIFY Central using either the UI or CLI.
2. Generate a key pair.
    a. Create a new Service Account user in AMPLIFY Central using the key pair. see Manage an API proxy using AMPLIFY CLI.
3. Create a Discovery Agent environment file.
    a. Move key files to a keys directory.
    b. Log into the Artifactory Repository and pull the latest binary of the Discovery Agent.
    c. Start the Discovery Agent.
4. Create a Traceability Agent environment file.
    a. Move key files to a keys directory.
    b. Log into the Artifactory Repository and pull the latest binary of the Traceability Agent.
    c. Start the Traceability Agent.

Prepare AMPLIFY Central 2

Learn how to create an environment and Service Account for Axway API Gateway within AMPLIFY Central.

Note The Axway API Gateway connectivity to AMPLIFY Central is currently available in an alpha review mode; current functionality and configuration may change before release. Therefore, this connectivity is available for trial use only and is not supported for production API management or connectivity.

Before you start

- Read *AMPLIFY Central and Axway API Manager connected overview* on page 5
- You will need a basic knowledge of Axway API Manager
- Verify that @axway/amplify-central-cli version is at minimum 0.1.3

Objectives

Learn how to create an environment and Service Account for Axway API Gateway within AMPLIFY Central.

Create an environment

Create an environment object in AMPLIFY Central that represents the effective Axway API Gateway environment. Depending on your needs, you can create as many environments as required.

Each discovered API or Traffic is associated to this environment and eases the filtering.

You can create your environment using either the UI, API or CLI.

Create environment using the UI

Create an environment in **AMPLIFY Central UI > Topology > Environments > create** and give it a relevant name. It is not necessary to have a real environment at this point, but it is important to have an environment name. You can find this environment name in your environment details in the UI.

Example: https:/<AMPLIFY Central URL>/topology/environments/apigtw-v77

Create environment using the CLI

Examples:

```
amplify central create -f <filename>
amplify central create env <name> -o json
```

Options:

```
-o, --output = yaml | json
-f, --file = (filename.yml, filename.yaml, or filename.json)
```

Sample file:

```
---
group: management
apiVersion: v1alpha1
kind: Environment
name: My beautifull environment name
title: Any usefull title
metadata:
  id: e4e084a66f86a7ea016f8c2ba1a40005
  audit:
    createTimestamp: '2020-01-09T21:17:47.302+0000'
    createUserId: DOSA_91cdec76c1084d86a6ee48f19bc
    modifyTimestamp: '2020-01-09T21:17:47.302+0000'
    modifyUserId: DOSA_91cdec76c1084d86a6ee48f19bc
  resourceVersion: '6'
  references: []
attributes:
  attr1: value1
  attr2: value2
tags:
  - Testing
  - another tag
spec:
  description: My wonderfull description to help me.
  icon:
    contentType: image/png
    data: "[optional base64 encoded image]"
```

For information, see Manage an environment using AMPLIFY CLI .

Create a Service Account

Create a Service Account in AMPLIFY Central.

1. Generate a private and public key pair:

```
openssl genpkey -algorithm RSA -out ./private_key.pem -pkeyopt rsa_keygen_bits:2048

openssl rsa -pubout -in ./private_key.pem -out ./public_key.pem
openssl rsa -pubout -in ./private_key.pem -out ./public_key.der -outform der
(optional) base64 ./public_key.der > ./public_key
```

Note The public key can be either of type .der format or of type base64 encoded of the .der format.

2. Create a new Service Account user in API Central using the key pair from above. For additional information, see Manage an API proxy using AMPLIFY CLI.

Deploy your agents 3

Learn how to deploy your Discovery Agent and Traceability Agent so that you can manage your Axway API Gateway environment within AMPLIFY Central.

Note The Axway API Gateway connectivity to AMPLIFY Central is currently available in an alpha review mode; current functionality and configuration may change before release. Therefore, this connectivity is available for trial use only and is not supported for production API management or connectivity.

Before you start

- Read *AMPLIFY Central and Axway API Manager connected overview* on page 5
- Prepare AMPLIFY Central
- You will need a basic knowledge of Axway API Gateway

Objectives

Learn how to create your Discovery Agent and Traceability Agent configuration files, then install and run your agents.

Discovery Agent

The Discovery Agent is used to discover new deployments and stage updates to existing deployments. Once they are discovered, the related APIs are published to AMPLIFY Central, in one of the following publication modes, so that they become available for any consumer. See *centralMode* on page 21.

- Catalog item publication (disconnected mode): Customers expose their APIs globally for their consumers but keep the API management at the Gateway level.
- API Service publication (connected mode): Customer manage their APIs from the AMPLIFY platform. WILL NOT BE FULLY IMPLEMENTED UNTIL Q3-2020.

As soon as an API is published, the identifier of the asset in AMPLIFY Central is kept in a custom field at the api level. The name of the custom field is defined in *APIMANAGER_PROXYAPICIDFIELD* on page 15.

The Discovery Agent only discovers APIs that have the tag(s) defined in the agent configuration file. See *Discovery Agent variables* on page 15.

The Agent can run in the following modes:

- With a configuration file
  - Default: located in the same directory as the agent binary.
  - Optional: use a dedicated folder where the configuration file is located (use the --pathConfig flag).

    Configuration file name should be the same as the agent binary.

    Properties inside the configuration file can reference environment variables. This enables you to set up only one configuration file that addresses different behaviors (depending on the environment variables). See *Discovery Agent variables* on page 15.
- With command line arguments. See *Discovery Agent flags* on page 20.

Create your configuration

To create an `env_vars` file, see *Discovery Agent variables* on page 15. Download the zip file:

```
curl -L "https://axway.bintray.com/generic-repo/v7-agents/v7_discovery_agent/latest/discovery_agent-latest.zip" -o discovery_agent-latest.zip
```

The Discovery Agent config yaml and Discovery Agent executable are included.

Install / run

1. Move the `private_key.pem` and `public_key` files that were originally created when you set up your Service Account to a keys directory. Make sure the directory is located on the machine being used for deployment. Note that the `public_key` comes from Steps 3 and 4 of *Create a Service Account* on page 8.
2. Download the zip file from https://axway.bintray.com/generic-repo/v7-agents/v7_discovery_agent/latest/discovery_agent-latest.zip.

Note The zip contains the Discovery Agent config yaml and the Discovery Agent executable.

3. Unzip the file and install the binary on a machine that can access the APIM Manager environment.

Traceability Agent

The Traceability Agent is used to filter the Axway API Gateway logs that are related to discovered APIs and prepare the transaction events that are sent to AMPLIFY platform. Each time an already discovered API is called by a consumer, an event (summary + detail) is sent to AMPLIFY Central and is visible in API Observer.

The Agent can run in the following modes:

- With a configuration file
    - Default: located in the same directory as the agent binary.
    - Optional: use a dedicated folder where the configuration file is located (use the --path.config flag). See *Traceability Agent flags* on page 27.

      Configuration file name should be the same as the agent binary.

      Properties inside the configuration file can reference environment variables. This enables you to set up only one configuration file that addresses different behaviors (depending on the environment variables). See *Traceability Agent variables* on page 23.
- With a YAML configuration file

Create your env_vars file

To create an `env_vars` file, see *Traceability Agent variables* on page 23.

Create your YAML config file

Most Traceability Agent configurations are overridden by the environment variable, except for the APIGateway event file path(s). Note that the default `traceability_agent.inputs.paths` is set to read multiple files using wildcard.

YAML config file template

```
################### Beat Configuration ########################
traceability_agent:
  inputs:
    - type: log
      paths:
        - /home/axway/axway/apigateway/events/group-*.log
      include_lines: ['.*"type":"transaction".*"type":"http".*']

# Send output to Central Database
output.traceability:
  enabled: true
  hosts: ${LOGSTASH_URL:ingestion-lumberjack.datasearch.axway.com:453}
  ssl:
    enabled: true
    verification_mode: none
  agent:
    central:
      tenantID: ${CENTRAL_TENANTID:""}
      deployment: ${CENTRAL_DEPLOYMENT:preprod}
```

```
      environment: ${CENTRAL_ENVIRONMENT:""}
      auth:
        url: ${CENTRAL_AUTH_URL:https://login.axway.com/auth}
        realm: ${CENTRAL_AUTH_REALM:Broker}
        clientId: ${CENTRAL_AUTH_CLIENTID:""}
        privateKey: ${CENTRAL_AUTH_PRIVATEKEY:/keys/private_key.pem}
        publicKey: ${CENTRAL_AUTH_PUBLICKEY:/keys/public_key}
        keyPassword: ${CENTRAL_AUTH_KEYPASSWORD:""}
        timeout: 10s
      ssl:
        minVersion: ${CENTRAL_SSL_MINVERSION:""}
        maxVersion: ${CENTRAL_SSL_MAXVERSION:""}
        nextProtos: ${CENTRAL_SSL_NEXTPROTOS:""}
        cipherSuites: ${CENTRAL_SSL_CIPHERSUITES:""}
        insecureSkipVerify: ${CENTRAL_SSL_INSECURESKIPVERIFY:""}
    apigateway:
      getHeaders: ${APIGATEWAY_GETHEADERS:true}
      host: ${APIGATEWAY_HOST:localhost}
      port: ${APIGATEWAY_PORT:8090}
      pollInterval: ${APIGATEWAY_POLLINTERVAL:1m}
      auth:
        username: ${APIGATEWAY_AUTH_USERNAME:""}
        password: ${APIGATEWAY_AUTH_PASSWORD:""}
      ssl:
        minVersion: ${APIGATEWAY_SSL_MINVERSION:""}
        maxVersion: ${APIGATEWAY_SSL_MAXVERSION:""}
        nextProtos: ${APIGATEWAY_SSL_NEXTPROTOS:""}
        cipherSuites: ${APIGATEWAY_SSL_CIPHERSUITES:""}
        insecureSkipVerify: ${APIGATEWAY_SSL_INSECURESKIPVERIFY:""}
    apimanager:
      host: ${APIMANAGER_HOST:localhost}
      port: ${APIMANAGER_PORT:8075}
      pollInterval: ${APIMANAGER_POLLINTERVAL:1m}
      apiVersion: ${APIMANAGER_APIVERSION:1.3}
      auth:
        username: ${APIMANAGER_AUTH_USERNAME:""}
        password: ${APIMANAGER_AUTH_PASSWORD:""}
      ssl:
        minVersion: ${APIMANAGER_SSL_MINVERSION:""}
        maxVersion: ${APIMANAGER_SSL_MAXVERSION:""}
        nextProtos: ${APIMANAGER_SSL_NEXTPROTOS:""}
        cipherSuites: ${APIMANAGER_SSL_CIPHERSUITES:""}
        insecureSkipVerify: ${APIMANAGER_SSL_INSECURESKIPVERIFY:""}
logging:
  metrics:
    enabled: false
  # Send all logging output to stderr
  to_stderr: true
  # Set log level
  level: ${LOG_LEVEL:info}
```

Multiple file paths

```
traceability_agent:
  inputs:
    - type: log
      paths:
        - /home/axway/axway/apigateway/events/group-2_instance-1.log
        - /home/axway/axway/apigateway/events/group-2_instance-2.log
```

File path with wildcard

```
traceability_agent:
  inputs:
    - type: log
      paths:
        - /home/axway/axway/apigateway/events/group-2_instance-*.log
```

Install / run

1. Move the `private_key.pem` and `public_key` files that were originally created when you set up your Service Account to a keys directory. Make sure the directory is located on the machine being used for deployment.
2. Download the zip file:

   ```
   curl -L "https://axway.bintray.com/generic-repo/v7-agents/v7_traceability_agent/latest/traceability_agent-latest.zip" -o traceability_agent-latest.zip
   ```

   Note The zip contains the Traceability Agent config yaml and the Traceability Agent executable.
3. Unzip the file and install the binary on a machine that can access the APIM Manager environment.

Discovery Agent variables 4

Use the following environment variables to create your Discovery Agent env_vars file. for additional information, see *Discovery Agent* on page 10.

Variable name	Description
API Manager variables	
APIMANAGER_HOST	The host API Manager is running on (localhost).
APIMANAGER_PORT	The port API Manager is listening on.
APIMANAGER_DISCOVERYIGNORETAGS	Comma-separated blacklist of tags that should not be on a Proxy before sending to AMPLIFY Central. Take precedence over APIMANAGER_FILTER
APIMANAGER_FILTER	Expression to filter the API you want the agent to discover. See *Filtering APIs to be discovered* on page 28 for conditional expression samples.
APIMANAGER_APIVERSION	The API version of the API Manager (1.3).
APIMANAGER_POLLINTERVAL	The frequency in which API Manager is polled for new endpoints (default=ns, us, ms, s, m, h). Set to **30s**.
APIMANAGER_PROXYAPICIDFIELD	The field name used to store AMPLIFY Central identifier for the frontend proxy in API Manager.
APIMANAGER_PROXYURL	The URL for the proxy for API Manager (http://username:password@hostname:port). If empty, no proxy is defined.

Variable name	Description
APIMANAGER_ SUBSCRIPTIONAPPLICATIONFIELD	The field name used to save subscription IDs to the API Manager application (default=subscriptions). To display this in the UI, add a custom property under applications in your API Manager configuration. See Customize API Manager.
APIMANAGER_AUTH_USERNAME	The API Manager username for this agent. Created in API Manager (must be API Manager Admin).
APIMANAGER_AUTH_PASSWORD	The password created for the API Manager username created for this agent (created in API Manager).
APIMANAGER_SSL_MINVERSION	String value for the minimum SSL/TLS version that is acceptable. If zero, empty TLS 1.0 is taken as the minimum. Allowed values are: TLS1.0, TLS1.1, TLS1.2, TLS1.3.
APIMANAGER_SSL_MAXVERSION	String value for the maximum SSL/TLS version that is acceptable. If empty, then the maximum version supported by this package is used, which is currently TLS 1.3. Allowed values are: TLS1.0, TLS1.1, TLS1.2, TLS1.3.
APIMANAGER_SSL_CIPHERSUITES	An array of strings. It is a list of supported cipher suites for TLS versions up to TLS 1.2. If CipherSuites is nil, a default list of secure cipher suites is used, with a preference order based on hardware performance. See *Supported Cipher Suites* on page 30.
APIMANAGER_SSL_NEXTPROTOS	An array of strings. It is a list of supported application level protocols, in order of preference, based on the ALPN protocol list. Allowed values are: h2, http/1.0, http/1.1, h2c.

Variable name	Description
APIMANAGER_SSL_INSECURESKIPVERIFY	Controls whether a client verifies the server's certificate chain and host name. If true, TLS accepts any certificate presented by the server and any host name in that certificate. In this mode, TLS is susceptible to man-in-the-middle attacks.
LOG_LEVEL	The log level for output messages (debug, info, warn, error).
LOG_FORMAT	The format to print log messages (json, line, package).
LOG_OUTPUT	The output for the log lines (stdout, file, both).
LOG_PATH	The path (relative or absolute) to save logs files, if output type file or both.
AMPLIFY Central variables	
CENTRAL_URL	The URL to the AMPLIFY Central instance being used for this Discovery Agent.
CENTRAL_TENANTID	The Organization ID from AMPLIFY Central. Locate this at Platform > User > Organization.
CENTRAL_TEAMID	The Team ID in AMPLIFY Central that all AWS APIs will be linked. Locate this at AMPLIFY Central > Access > Teams.
CENTRAL_MODE	Method to send endpoints back to Central. (connected = API Server, disconnected = Catalog).
CENTRAL_PROXYURL	The URL for the proxy for Amplify Central (http://username:password@hostname:port). If empty, no proxy is defined.
CENTRAL_ENVIRONMENT	Environment that is set by download kit in APIC
CENTRAL_AUTH_URL	The AMPLIFY login URL: https://login.axway.com/auth

Variable name	Description
CENTRAL_AUTH_REALM	The Realm used to authenticate for AMPLIFY Central.
CENTRAL_AUTH_CLIENTID	The name of the Service Account created in AMPLIFY Central. Locate this at AMPLIFY Central > Access > Service Accounts.
CENTRAL_AUTH_PRIVATEKEY	The private key associated with the Service Account.
CENTRAL_AUTH_PUBLICKEY	The public key associated to the Service Account. Extract using the following commands: • `openssl genpkey -algorithm RSA -out ./private_key.pem -pkeyopt rsa_keygen_ bits:2048` • `openssl rsa -pubout -in ./private_key.pem -out ./public_key.pem` • `openssl rsa -pubout -in ./private_key.pem -out ./public_key.der -outform der` • `base64 ./public_key.der > ./public_key` If the keys for APIC service account have already been generated, then only the 3rd and 4th bullet points need to be run using the public key that was previously generated.
CENTRAL_AUTH_KEYPASSWORD	The password for the private key, if applicable.
CENTRAL_AUTH_TIMEOUT	The timeout to wait for the authentication server to respond (ns - default, us, ms, s, m, h). Set to **10s**.
CENTRAL_ENVIRONMENT	Name of the AMPLIFY Central environment.
CENTRAL_APISERVERVERSION	Version of the API Server that the agent will communicate with

Variable name	Description
CENTRAL_ADDITIONALTAGS	Additional tag names to publish separated by a comma.
CENTRAL_SSL_MINVERSION	String value for the minimum SSL/TLS version that is acceptable. If zero, empty TLS 1.0 is taken as the minimum. Allowed values are: TLS1.0, TLS1.1, TLS1.2, TLS1.3.
CENTRAL_SSL_MAXVERSION	String value for the maximum SSL/TLS version that is acceptable. If empty, then the maximum version supported by this package is used, which is currently TLS 1.3. Allowed values are: TLS1.0, TLS1.1, TLS1.2, TLS1.3.
CENTRAL_SSL_CIPHERSUITES	An array of strings. It is a list of supported cipher suites for TLS versions up to TLS 1.2. If CipherSuites is nil, a default list of secure cipher suites is used, with a preference order based on hardware performance. See *Supported Cipher Suites* on page 30.
CENTRAL_SSL_NEXTPROTOS	An array of strings. It is a list of supported application level protocols, in order of preference, based on the ALPN protocol list. Allowed values are: h2, http/1.0, http/1.1, h2c.
CENTRAL_SSL_INSECURESKIPVERIFY	Controls whether a client verifies the server's certificate chain and host name. If true, TLS accepts any certificate presented by the server and any host name in that certificate. In this mode, TLS is susceptible to man-in-the-middle attacks.

Discovery Agent flags 5

Use the following parameters [flags] when issuing the discovery_agent command. View these flags with command `discovery_agent -h`.

Note These parameters are also environment variables referenced in *Deploy your agents* on page 10, with the exception of help and pathConfig. These parameters are not configuration values.

discovery_agent [flags] Flags	Description
apimanagerHost	String. Host of API Manager service (default "localhost").
apimanagerPassword	String. API Manager password.
apimanagerPort	Int. Port of API Manager service (default 8075).
apimanagerDiscoveryIgnoreTags	String. List of tags on frontend proxy to check for and ignore discovery.
apimanagerFilter	String. Filter condition for discovery
apimanagerPollInterval	Duration. The time interval at which the published proxies will be checked for publishing as catalog. (default 30s).
apimanagerUsername	String. API Manager username.
apiManagerSSLMinVersion	String. Minimum acceptable SSL/TLS protocol version (default "TLS1.2").
apiManagerSSLMaxVersion	String. Maximum acceptable SSL/TLS protocol version (default "0").
apiManagerSSLCipherSuites	Strings. A list of supported cipher suites, comma separated (default [ECDHE-ECDSA-AES-256-GCM-SHA384,ECDHE-RSA-AES-256-GCM-SHA384,ECDHE-ECDSA-CHACHA20-POLY1305,ECDHE-RSA-CHACHA20-POLY1305,ECDHE-ECDSA-AES-128-GCM-SHA256,ECDHE-RSA-AES-128-GCM-SHA256,ECDHE-ECDSA-AES-128-CBC-SHA256,ECDHE-RSA-AES-128-CBC-SHA256])

discovery_agent [flags] Flags	Description
apiManagerSSLNextProtos	Strings. List of supported application level protocols, comma separated.
apiServerEnvironment	String. The Environment that the APIs will be associated with in AMPLIFY Central.
apiVersion	String. The version of the V7 API. (default "1.3").
apiManagerSSLInsecureSkipVerify	Controls whether a client verifies the server's certificate chain and host name.
authClientId	String. Client ID for the service account.
authKeyPassword	String. Password for the private key, if needed.
authPrivateKey	String. Path to the private key for AMPLIFY Central Authentication (default "/etc/private_key.pem").
authPublicKey	String. Path to the public key for AMPLIFY Central Authentication (default "/etc/public_key").
authRealm	String. AMPLIFY Central authentication Realm (default "Broker").
authTimeout	Duration. Timeout waiting for AxwayID response (default 10s).
authUrl	String. AMPLIFY Central authentication URL (default "https://login-preprod.axway.com/auth").
centralMode	String. Agent Mode (default "disconnected").
centralPollInterval	Duration. The time interval at which the central will be polled for subscription processing (default 1m0s).
centralSSLCipherSuites	Strings. List of supported cipher suites, comma separated (default [ECDHE-ECDSA-AES-256-GCM-SHA384,ECDHE-RSA-AES-256-GCM-SHA384,ECDHE-ECDSA-CHACHA20-POLY1305,ECDHE-RSA-CHACHA20-POLY1305,ECDHE-ECDSA-AES-128-GCM-SHA256,ECDHE-RSA-AES-128-GCM-SHA256,ECDHE-ECDSA-AES-128-CBC-SHA256,ECDHE-RSA-AES-128-CBC-SHA256]).

discovery_agent [flags] Flags	Description
centralSSLInsecureSkipVerify	Controls whether a client verifies the server's certificate chain and host name.
centralSSLMaxVersion	String. Maximum acceptable SSL/TLS protocol version (default "0").
centralSSLMinVersion	String. Minimum acceptable SSL/TLS protocol version (default "TLS1.2").
centralSSLNextProtos	Strings. List of supported application level protocols, comma separated.
centralTeamId	String. Team ID for the current default team for creating catalog.
centralTenantId	String. Tenant ID for the owner of the environment.
centralUrl	String. URL of AMPLIFY Central (default "https://apicentral.preprod.k8s.axwayamplify.com").
help	Help for discovery_agent.
logFormat	String. Log format (json, line, package) (default "json").
logLevel	String. Log level (debug, info, warn, error) (default "info").
logOutput	String. Log output type (stdout, file, both) (default "stdout").
logPath	String. Log file path if output type is file or both (default "logs").
pathConfig	String. Configuration file path for the agent.
subscriptionApplicationField	String. The custom field name in V7 to track Subscription IDs. (default "subscriptions").
version	Version for discovery_agent.

Traceability Agent variables 6

Use the following environment variables to create your Traceability Agent env_vars file. For additional information, see *Traceability Agent* on page 11.

Variable name	Description
API Gateway variables	
APIGATEWAY_GETHEADERS	Call the API Gateway API to get additional transaction details (headers, useragent). If false, API Gateway config does not need to be set. Default is True.
APIGATEWAY_HOST	The host that Axway API Gateway is running on.
APIGATEWAY_PORT	The port that Axway API Gateway is listening on.
APIGATEWAY_ POLLINTERVAL	The frequency in which the agent polls the logs in us, ms, s, m, h. Default=ns. Set to 1m.
APIGATEWAY_PROXYURL	The URL for the proxy for Axway API Gateway (http://username:password@hostname:port). If empty, no proxy is defined.
APIGATEWAY_AUTH_ USERNAME	Your Axway API Gateway username.
APIGATEWAY_AUTH_ PASSWORD	Your Axway API Gateway password.
APIGATEWAY_SSL_ MINVERSION	String value for the minimum SSL / TLS version that is acceptable. If zero, empty TLS 1.0 is taken as the minimum. Allowed values are: TLS1.0, TLS1.1, TLS1.2, TLS1.3.
APIGATEWAY_SSL_ MAXVERSION	String value for the maximum SSL / TLS version that is acceptable. If empty, then the maximum version supported by this package is used, which is currently TLS 1.3. Allowed values are: TLS1.0, TLS1.1, TLS1.2, TLS1.3.

Variable name	Description
APIGATEWAY_SSL_ CIPHERSUITES	An array of strings. It is a list of supported cipher suites for TLS versions up to TLS 1.2. If CipherSuites is nil, a default list of secure cipher suites is used, with a preference order based on hardware performance. See *Supported Cipher Suites* on page 30.
APIGATEWAY_SSL_ NEXTPROTOS	An array of strings. It is a list of supported application level protocols, in order of preference, based on the ALPN protocol list. Allowed values are: h2, http/1.0, http/1.1, h2c.
APIGATEWAY_SSL_ INSECURESKIPVERIFY	InsecureSkipVerify controls whether a client verifies the server's certificate chain and host name. If true, then TLS accepts any certificate presented by the server and any host name in that certificate. In this mode, TLS is susceptible to man-in-the-middle attacks.
API Manager variables	
APIMANAGER_HOST	The host API Manager is running on.
APIMANAGER_PORT	The port API Manager is listening on.
APIMANAGER_APIVERSION	The API version for the API Manager.
APIMANAGER_ POLLINTERVAL	The frequency in which API Manager is polled for new endpoints (ns - default, us, ms, s, m, h). Set to 1m.
APIMANAGER_ PROXYAPICIDFIELD	The field name used to store the AMPLIFY Central identifier for the frontend proxy in API Manager.
APIMANAGER_PROXYURL	The URL for the proxy for API Manager (http://username:password@hostname:port). If empty, no proxy is defined.
APIMANAGER_AUTH_ USERNAME	The API Manager username created for this agent. Must be API Manager Admin.
APIMANAGER_AUTH_ PASSWORD	The password for the API Manager username created for this agent.
APIMANAGER_SSL_ MINVERSION	String value for the minimum SSL/TLS version that is acceptable. If zero, empty TLS 1.0 is taken as the minimum. Allowed values are: TLS1.0, TLS1.1, TLS1.2, TLS1.3.

Variable name	Description
APIMANAGER_SSL_MAXVERSION	String value for the maximum SSL/TLS version that is acceptable. If empty, then the maximum version supported by this package is used, which is currently TLS 1.3. Allowed values are: TLS1.0, TLS1.1, TLS1.2, TLS1.3.
APIMANAGER_SSL_CIPHERSUITES	An array of strings. It is a list of supported cipher suites for TLS versions up to TLS 1.2. If CipherSuites is nil, a default list of secure cipher suites is used, with a preference order based on hardware performance. See *Supported Cipher Suites* on page 30.
APIMANAGER_SSL_NEXTPROTOS	An array of strings. It is a list of supported application level protocols, in order of preference, based on the ALPN protocol list. Allowed values are: h2, http/1.0, http/1.1, h2c.
APIMANAGER_SSL_INSECURESKIPVERIFY	Controls whether a client verifies the server's certificate chain and host name. If true, TLS accepts any certificate presented by the server and any host name in that certificate. In this mode, TLS is susceptible to man-in-the-middle attacks.
Amplify Central variables	
LOG_LEVEL	The log level for the agent.
LOGSTASH_URL	The URL of the logstash to forward the transaction log entries.
LOGSTASH_PROXYURL	The URL for the proxy for logstash (socks5://hostname:port). If empty, no proxy is defined.
CENTRAL_URL	The URL to the AMPLIFY Central instance being used for this Traceability Agent.
CENTRAL_TENANTID	The Organization ID from AMPLIFY Central. Locate this ID in the Platform > Organization > Org ID.
CENTRAL_DEPLOYMENT	The APIC deployment environment.
CENTRAL_ENVIRONMENT	Name of the AMPLIFY Central environment.
CENTRAL_PROXYURL	The URL for the proxy for Amplify Central (http://username:password@hostname:port). If empty, no proxy is defined.
CENTRAL_AUTH_URL	The AMPLIFY login URL: https://login.axway.com/auth

Variable name	Description
CENTRAL_AUTH_REALM	The Realm used to authenticate for AMPLIFY Central. Locate this in AMPLIFY Central > Access > Service Accounts.
CENTRAL_AUTH_CLIENTID	The name of the Service Account created in AMPLIFY Central. Locate this in AMPLIFY Central > Access > Service Accounts. `DOSA_XXXXXXXXXx`
CENTRAL_AUTH_ PRIVATEKEY	The private key associated with the Service Account.
CENTRAL_AUTH_PUBLICKEY	The public key associated to the Service Account.
CENTRAL_AUTH_ KEYPASSWORD	The password for the private key, if applicable.
CENTRAL_SSL_MINVERSION	String value for the minimum SSL/TLS version that is acceptable. If zero, empty TLS 1.0 is taken as the minimum. Allowed values are: TLS1.0, TLS1.1, TLS1.2, TLS1.3.
CENTRAL_SSL_MAXVERSION	String value for the maximum SSL/TLS version that is acceptable. If empty, then the maximum version supported by this package is used, which is currently TLS 1.3. Allowed values are: TLS1.0, TLS1.1, TLS1.2, TLS1.3.
CENTRAL_SSL_ CIPHERSUITES	An array of strings. It is a list of supported cipher suites for TLS versions up to TLS 1.2. If CipherSuites is nil, a default list of secure cipher suites is used, with a preference order based on hardware performance. See *Supported Cipher Suites* on page 30.
CENTRAL_SSL_ NEXTPROTOS	An array of strings. It is a list of supported application level protocols, in order of preference, based on the ALPN protocol list. Allowed values are: h2, http/1.0, http/1.1, h2c.
CENTRAL_SSL_ INSECURESKIPVERIFY	Controls whether a client verifies the server's certificate chain and host name. If true, TLS accepts any certificate presented by the server and any host name in that certificate. In this mode, TLS is susceptible to man-in-the-middle attacks.

Traceability Agent flags 7

Use the following parameters [flags] when issuing the traceability_agent command. View these flags with command `traceability_agent -h`.

discovery_agent [flags] Flags	Description
path.config	String. Configuration path.
path.data	String. Data path.
path.home	String. Home path.
path.logs	String. Logs path.
strict.perms	Strict permission checking on config files (default true).

Filtering APIs to be discovered 8

You can set up tag-based condition expression(s) using the *Deploy your agents* on page 10 environment variable to discover APIs that could be added to AMPLIFY Central.

Conditional expressions statements use logical operators to compare values. This section provides sample syntax for defining expressions.

Notes:

For tag-based filtering, the conditional expression should have "tag" as the prefix / selector in the symbol name: `tag.<tagName> == <tagValue>`

The expression can be a simple condition or a compound condition in which multiple conditions are evaluated using logical operators. See *Logical operators* on page 29 and *Comparative operators* on page 29.

Filter based on tag name

```
tag.<tagName>.Exists() == true | false
```

Filter based on tag value

```
tag.Any() == | != <tagValue>
```

Filter based on tag name and tag value

```
tag.<tagName> == | != <tagValue>
```

Filter based on partial value

```
tag.<tagName>.contains(<value>) == true | false
```

Filter using MatchRegEx

```
tag.<tagName>.matchRegEx(<regularExpression>)
```

Logical operators

Logical operators are used for evaluating multiple conditions. These can be combined with comparative operators and any of the above expressions.

Operator	Description
&&	Logical AND operator, returns true if conditions on both sides are true.
||	Logical OR operator, returns true if the condition on either side is true.

Comparative operators

Comparative operators are used for comparing two values. These can be combined with logical operators and any of the above expressions.

Operator	Description
==	Equal to operator, returns true if values on both sides are equal.
!=	Not equal to operator, returns true if the value on the left side is not equal to the value on the right side.

SSL / TLS advanced 9

By default, the Axway API Manager connection to AMPLIFY Central is SSL secured using TLS1.2 and the appropriate secured default Cipher Suite. However, you can choose to change the default to use different variable values for protocols and supported Cipher Suites that meet your own requirements. See the variable tables in *Deploy your agents* on page 10.

Note TLS 1.3 is not yet supported by AMPLIFY Central.

Default Cipher Suites

ECDHE-ECDSA-AES-256-GCM-SHA384
ECDHE-RSA-AES-256-GCM-SHA384

ECDHE-ECDSA-CHACHA20-POLY1305
ECDHE-RSA-CHACHA20-POLY1305

ECDHE-ECDSA-AES-128-GCM-SHA256
ECDHE-RSA-AES-128-GCM-SHA256

ECDHE-ECDSA-AES-128-CBC-SHA256
ECDHE-RSA-AES-128-CBC-SHA256

Supported Cipher Suites

ECDHE-ECDSA-AES-128-CBC-SHA
ECDHE-ECDSA-AES-128-CBC-SHA256
ECDHE-ECDSA-AES-128-GCM-SHA256
ECDHE-ECDSA-AES-256-CBC-SHA
ECDHE-ECDSA-AES-256-GCM-SHA384
ECDHE-ECDSA-CHACHA20-POLY1305
ECDHE-ECDSA-RC4-128-SHA
ECDHE-RSA-3DES-CBC3-SHA
ECDHE-RSA-AES-128-CBC-SHA
ECDHE-RSA-AES-128-CBC-SHA256
ECDHE-RSA-AES-128-GCM-SHA256
ECDHE-RSA-AES-256-CBC-SHA
ECDHE-RSA-AES-256-GCM-SHA384
ECDHE-RSA-CHACHA20-POLY1305
ECDHE-RSA-RC4-128-SHA

RSA-RC4-128-SHA
RSA-3DES-CBC3-SHA
RSA-AES-128-CBC-SHA

RSA-AES-128-CBC-SHA256
RSA-AES-128-GCM-SHA256
RSA-AES-256-CBC-SHA
RSA-AES-256-GCM-SHA384

TLS-AES-128-GCM-SHA256
TLS-AES-256-GCM-SHA384
TLS-CHACHA20-POLY1305-SHA256

Subscription for the consumer

10

A subscription provides the consumer, or subscriber, with the required security, quota and endpoint materials to correctly consume the API.

Subscription workflow

1. An administrator creates an application on Axway API Manager that provides the necessary security feature (API key / OAuth...) and quota, if needed:
    - Add a custom field to the application to track the AMPLIFY Central subscription. Refer to `<API_Gateway_install_dir>/apigateway/webapps//apiportal/vordel/apiportal/app/app.config` file in the **customPropertiesConfig** section. For more details, see Customize API Manager.

      Sample application:

      ```
      customPropertiesConfig: {
              user: {
                  // custom properties...
              },
              organization: {
                  // custom properties...
              },
              application: {
                  subscriptions: {
                     label: 'Subscriptions'
                  },
              },
              api: {
                  // custom properties...
              }
          }
      ```
2. A consumer initiates the subscription in AMPLIFY Central:
    a. Open an AMPLIFY Catalog item.
    b. Click **Subscribe**.
    c. Enter the *Team* and API Manager *Application* name (created in Step 1). **Warning**: The names must match. Otherwise, the subscription will fail.

    For additional information, see Manage AMPLIFY Catalog subscriptions.

3. The Discovery Agent receives the subscription event:

   Subscription status: **Subscribing...**

   a. Associate the API to the selected application.

   b. Send back the subscription status.

   Subscription status: **Active**

   - Subscription ID is automatically added to the **Custom** field of the application.
   - If failure, subscription status: **Subscription failed**. Refer to the Discovery Agent log for more information. You can delete the subscription and start again from Step 2.

4. The subscriber consumes the API:

   - The API can be consumed once the subscription details are received.

Note The API Manager application and the API must be in the same organization. Otherwise, an error message is displayed in the Discovery Agent log.
Workaround: You can grant the API access to the organization where the application belongs:

1. In the UI, select the API.
2. Expand **Manage selected**.
3. Select **Grant access**.

Unsubscribe workflow

1. A consumer initiates unsubscribe:

   a. Open the AMPLIFY Catalog and navigate to the **Subscription** tab.

   b. Delete the subscription

   For additional information, see Manage AMPLIFY Catalog subscriptions.

2. The Discovery Agent receives the Unsubscribe event:

   - The subscription ID is removed from the application's Custom field.

Tips, troubleshooting and limitations 11

Note The Axway API Gateway connectivity to AMPLIFY Central is currently available in an alpha review mode; current functionality and configuration may change before release. Therefore, this connectivity is available for trial use only and is not supported for production API management or connectivity.

Tips

API summary

In Axway API Manager, you can edit an unpublished API to include an API summary (description) that displays in the API's Catalog Overview once the API is published. If the API summary field is empty when the API is published, then "API From V7 APIManager" is displayed in the Catalog Overview. To update the API summary once the API is published, you must unpublish the API.

Troubleshooting

Question	Answer
Why can't Discovery Agent connect to Axway API Manager?	Make sure that the Axway API Manager hostname and port are correct. Also Make sure that the connected user has the Axway API Manager Administrator role.
Why can't Traceability Agent connect to Axway API Gateway?	Make sure that the Axway API Gateway hostname and port are correct. Also Make sure that the connected user has the Axway API Gateway Operator role.
Why can't my agents connect to AMPLIFY Central?	Go to *AMPLIFY Central UI* > **Access** > **Service Accounts** and make sure that the Service Account is correctly named and valid. Make sure that the tenantID and teamID are correct.
Why don't I see traffic in AMPLIFY Central?	Make sure that the Condor URL is accessible from the machine where Traceability Agent is installed.

Question	Answer
Why isn't my API discovered?	Check that the tag set on the frontend API has a correct name and its value is **True**. See *Deploy your agents* on page 10.

Limitations

Web services or APIs including AWS signing / HTTP Basic / Invoke policy / 2-way SSL security cannot be discovered, as these features are not currently supported in AMPLIFY Central.