opentext™

OpenText™ Core SAST Aviator

Software Version: 25.2.0

User Guide

Document Release Date: May 2025
Software Release Date: May 2025

Legal notices

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Copyright notice

Trademark notices

Documentation updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

This document was produced for OpenText™ Core SAST Aviator CE 25.2 on May 14, 2025.

Contents

Preface

Contacting Customer Support

Visit the Customer Support website to:

- Manage licenses and entitlements
- Create and manage technical assistance requests
- Browse documentation and knowledge articles
- Download software
- Explore the Community

For more information

For more information about OpenText Application Security Testing products, visit OpenText Application Security.

Product feature videos

You can find videos that highlight OpenText Application Security Software products and features on the Fortify Unplugged YouTube™ channel.

Change log

The following table lists changes made to this document. Revisions to this document are published between software releases only if the changes made affect product functionality.

Software Release / Document Version	Changes
25.2.0	Initial release of the document.

Introduction

OpenText™ Core SAST Aviator is a cloud-based enterprise service that audits, identifies, and classifies each issue received from SAST scan result as a true positive or a false positive.

SAST Aviator leverages Large Language Model (LLM) technology to evaluate each identified issue, predicting whether it is a true positive or a false positive. For both true and false positive cases, SAST Aviator provides a detailed explanation. When an issue is classified as a true positive, SAST Aviator offers remediation recommendations, enabling users to resolve code issues quickly and accurately.

SAST Aviator is accessible using SAST in an off-cloud setup and SAST through the Fortify Hosted model. In both scenarios, you can use the open-source Fortify CLI tool to transmit SAST scan results from the OpenText™ Fortify Software Security Center to SAST Aviator for processing. The results from SAST Aviator are stored as audit information in the Fortify Software Security Center.

Product name changes

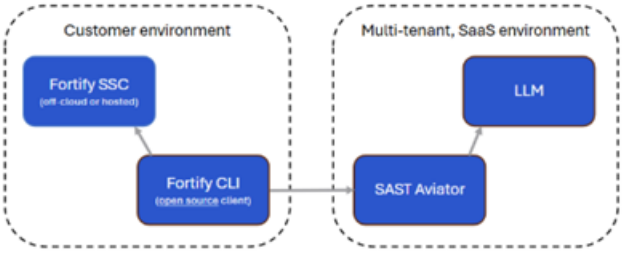
OpenText is in the process of changing the following product names:

Previous name	New name
Fortify Static Code Analyzer	OpenText™ Static Application Security Testing (OpenText SAST)
Fortify Software Security Center	OpenText™ Application Security
Fortify WebInspect	OpenText™ Dynamic Application Security Testing (OpenText DAST)
Fortify on Demand	OpenText™ Core Application Security
Debricked	OpenText™ Core Software Composition Analysis (OpenText Core SCA)
Fortify Applications and Tools	OpenText™ Application Security Tools

The product names have changed on product splash pages, mastheads, login pages, and other places where the product is identified. The name changes are intended to clarify product functionality and to better align the Fortify Software products with OpenText. In some cases, such as on the documentation title page, the old name might temporarily be included in parenthesis. You can expect to see more changes in future product releases.

SAST Aviator model

Fortify CLI tool sends the SAST scan results, including vulnerability information and source code snippets from the Fortify Software Security Center to SAST Aviator. The SAST Aviator sends the scan results to an LLM for auditing the scan results. The results are then returned to Fortify CLI. The vulnerability and audit information are not stored within the SAST Aviator. SAST Aviator retains only the statistical data indicating that an audit occurred, including the number of issues identified.



Advantages of SAST Aviator

SAST Aviator leverages Generative Artificial Intelligence (GenAI) in the form of a Large Language Model (LLM) to generate content and provide product functionality.

- **Hybrid model**: SAST Aviator model enables you to use the cloud service hosted by OpenText and the existing Fortify CLI utility and not host a heavy LLM.
- **Integration with Fortify Software Security Center**: A regular SAST scan is performed, and the results are uploaded to the Fortify Software Security Center. Subsequently, SAST Aviator audits the scan results stored in Fortify Software Security Center. This approach is designed to prevent redundant evaluations.

fcli capabilities

In addition to auditing the scan results using fcli, the following operations can be performed:

- View entitlements and entitlement consumption. See "Entitlement model" on the next page section.
- Create new applications and view available applications.
- Create access tokens for individual users.
- List and revoke the access tokens.

User roles

SAST Aviator includes sequential procedures and involves different user roles in your organization. The user roles involved are:

- Customer administrator: Customer administrator can create an admin configuration, create and manage tokens, and manage applications and entitlements.
- Customer user: User can create a user session and trigger an audit.

Note: Before using the fcli, you must be registered with OpenText. OpenText Support creates and registers the tenant upon your initial purchase of SAST Aviator.

Entitlement model

SAST Aviator is a paid service. The SAST Aviator models available for purchase are **per Developer** and **per Application**.

- **per Application**: The basic entitlement model is **per Application**. You can run any number of audits for a single application. Therefore, the customer administrator must ensure to register, that is, create applications in the tenant. SAST Aviator monitors the number of entitlements for the respective tenant every time a Fortify Project Report (FPR) is processed.
- **per Developer**: SAST Aviator cannot monitor developers. By default, a maximum of four applications per ten developers are allocated. If this allocation does not meet your requirements, contact your account representative.

For every new purchase of the SAST Aviator entitlement, OpenText Support updates the number of entitlements for the respective tenant.

Limitations

When presented with SAST results containing extremely large numbers of issues, certain limits will apply to SAST Aviator's auditing.

This is very similar to how auditing by a human practically works. If there are hundreds of issues, they can be audited manually. But if there are thousands, that's neither practical nor useful. The first response should be to look for patterns. A recurring bad coding pattern may cause many true positives. In that case, the code should be fixed in bulk. Alternatively, some rule in SAST may trigger massive false positives for that particular codebase. In that case, the SAST scan configuration should be adjusted. After these steps, a smaller number of remaining findings can be audited individually.

SAST Aviator has largely been designed as an AI-powered version of a human auditor. It will not blindly audit an unlimited number of issues. Practically, such a limit is also necessary, given the non-trivial resource consumption for every issue audited.

The following limits apply:

- For any FPR, at most 2,500 new issues will be audited in total.
- For any FPR, at most 500 new issues will be audited in a single category.

When SAST Aviator processes an FPR that exceeds one or more of these limits, any issue beyond the limit will be marked as “Excluded due to Limiting”, and the specific limit will be explained in the comment. Such issues will not be audited in a subsequent run either. When a subsequent SAST analysis of the same project yields new issues, these new issues will be audited.

A concrete example to illustrate the principle:

- A SAST scan of project X yields 3,000 issues.
- The SAST Aviator auditing will audit 2,500 of those, and mark 500 as “Excluded due to Limiting”.
- Now, development continues. 100 additional issues have been found, leading to an FPR with 3,100 issues.
- SAST Aviator auditing of this new FPR will:
  - Not touch the 2,500 previously audited issues.
  - Not touch the 500 issues previously marked as excluded.
  - Audit the 100 new issues.

Authentication model

To perform tasks using fcli, the user must be authenticated based on the role.

Customer administrator must use the private key to sign in to the administrator requests. See "Manage tenant information" on page 22 for the list of administrator tasks. The user must use the access token to sign in to the user request, such as to create a user session before performing a user task.

Audit tag mapping

The SAST Aviator algorithm predicts issues to be true positives or false positives and, in some cases, unsure.

In Fortify Software Security Center, the Fortify Software Security Center needs to set a certain audit tag value, and decide to suppress an issue or not. As a user, you may have customized the available audit tag values in Fortify Software Security Center resulting in deviations from the default values. For these reasons, SAST Aviator has functionality to map its predictions to audit tag values and suppression status in Fortify Software Security Center.

To configure this mapping, SAST Aviator considers the two tiers of support. See "Supported languages and vulnerability categories" on page 15 for more information. Considering there are two tiers and three different SAST Aviator outcomes (true positive (TP), false positive (FP), unsure), there are six different cases. These cases must be mapped to an Fortify Software Security Center audit value and a suppression status. The following is the default mapping performed by SAST Aviator:

Tier	Tier configuration name	Outcome	Audit value	Suppressed
Supported with automatic suppression	tier_1	TP	Exploitable	No
Supported with automatic suppression	tier_1	FP	Not an issue	Yes
Supported with automatic suppression	tier_1	Unsure	Not set	No
Supported without automatic suppression	tier_2	TP	Suspicious	No
Supported without automatic suppression	tier_2	FP	Not an issue	No
Supported without automatic suppression	tier_2	Unsure	Not set	No

To override the default tag mapping, use the **--tag-mapping** argument when you run an audit.

```
fcli aviator ssc audit --av <application_version_name:id> --tag-mapping=<file.yaml>
```

Note: SAST Aviator provides a clear and reasonable message even if the mapping file does not comply with the format.

The following tag mapping file is the default one that implements the mapping as explained in the table above. Use this mapping file as a basis to configure your required mapping. In addition to changing audit tag values and suppression status, you can also select a different audit tag altogether.

```
# Set the SSC tag to use to store Aviator results. Optional.
# If not set, defaults to "87f2364f-dcd4-49e6-861d-f8d3f351686b"
tag_id: "87f2364f-dcd4-49e6-861d-f8d3f351686b"


# Map Aviator results to SSC tag values. “tier_1” are issues that are
# high-confidence cases that by default are suppressed automatically.
# “tier_2” are the remaining issues.
# “value” is a String attribute that maps to a tag value in SSC. It may be
# omitted. In that case, Aviator will not set a value (but will still add a
# comment)
# “suppress” is a Boolean attribute that defaults to “false”


mapping:
  tier_1:
    fp:
      value: "Not an Issue"
      suppress: true
    tp:
      value: "Exploitable"
      suppress: false
    unsure:
      suppress: false
  tier_2:
    fp:
      value: "Not an Issue"
      suppress: false
    tp:
      value: "Suspicious"
      suppress: false
    unsure:
      suppress: false
```

Related documents

This topic describes documents that provide information about OpenText Application Security Software products.

Note: Most guides are available in both PDF and HTML formats.

All products

The following documents provide general information for all products. Unless otherwise noted, these documents are available on the Product Documentation website for each product.

Document / file name	Description
About OpenText Application Security Software Documentation appsec-docs-n-<*version*>.pdf	This paper provides information about how to access OpenText Application Security Software product documentation. **Note:** This document is included only with the product download.
OpenText Application Security Software Release Notes	This document provides an overview of the changes made to OpenText Application Security Software for this release and important information not included elsewhere in the product documentation.

Fortify ScanCentral SAST

The following document provides information about Fortify ScanCentral SAST. This document is available on the Product Documentation website at https://www.microfocus.com/documentation/fortify-software-security-center.

Document / file name	Description
OpenText™ Fortify ScanCentral SAST Installation, Configuration, and Usage Guide sc-sast-ugd-<*version*>.pdf	This document provides information about how to install, configure, and use Fortify ScanCentral SAST to streamline the static code analysis process. It is written for anyone who intends to install, configure, or use Fortify ScanCentral SAST to offload the resource-intensive translation and scanning phases of their OpenText SAST process.

Fortify Software Security Center

The following document provides information about Fortify Software Security Center. This document is available on the Product Documentation website at https://www.microfocus.com/documentation/fortify-software-security-center.

Document / file name	Description
OpenText™ Applicaton Security User Guide ssc-ugd-<*version*>.pdf	This document provides Fortify Software Security Center users with detailed information about how to deploy and use Fortify Software Security Center. It provides all the information you need to deploy, configure, and use Fortify Software Security Center. It is intended for use by system and instance administrators, database administrators (DBAs), enterprise security leads, development team managers, and developers. Fortify Software Security Center provides security team leads with a high-level overview of the history and status of a project.

OpenText SAST

The following documents provide information about OpenText SAST (Fortify Static Code Analyzer). Unless otherwise noted, these documents are available on the Product Documentation website at https://www.microfocus.com/documentation/fortify-static-code.

Document / file name	Description
OpenText™ Static Application Security Testing User Guide sast-ugd-<*version*>.pdf	This document describes how to install and use OpenText SAST to scan code on many of the major programming platforms. It is intended for people responsible for security audits and secure coding.
OpenText™ Static Application Security Testing Custom Rules Guide sast-cr-ugd-<*version*>.zip	This document provides the information that you need to create custom rules for OpenText SAST. This guide includes examples that apply rule-writing concepts to real-world security issues. **Note:** This document is included only with the product download.
OpenText™ Fortify License and	This document describes how to install, configure, and use

Document / file name	Description
Infrastructure Manager Installation and Usage Guide lim-ugd-<*version*>.pdf	the Fortify License and Infrastructure Manager (LIM), which is available for installation on a local Windows server and as a container image on the Docker platform.

OpenText Application Security Tools

The following documents provide information about OpenText Application Security Tools. These documents are available on the Product Documentation website at https://www.microfocus.com/documentation/fortify-static-code-analyzer-and-tools.

Document / file name	Description
OpenText™ Application Security Tools Guide sast-tgd-<*version*>.pdf	This document describes how to install application security tools. It provides an overview of the applications and command-line tools that enable you to scan your code with OpenText SAST, review analysis results, work with analysis results files, and more.
OpenText™ Fortify Audit Workbench User Guide awb-ugd-<*version*>.pdf	This document describes how to use Fortify Audit Workbench to scan software projects and audit analysis results. This guide also includes how to integrate with bug trackers, produce reports, and perform collaborative auditing.
OpenText™ Fortify Plugin for Eclipse User Guide ep-udg-<*version*>.pdf	This document provides information about how to install and use the Fortify Plugin for Eclipse to analyze and audit your code.
OpenText™ Fortify Analysis Plugin for IntelliJ IDEA and Android Studio User Guide iap-udg-<*version*>.pdf	This document describes how to install and use the Fortify Analysis Plugin for IntelliJ IDEA and Android Studio to analyze your code and optionally upload the results to Fortify Software Security Center.
OpenText™ Fortify Extension for Visual Studio User Guide vse-ugd-<*version*>.pdf	This document provides information about how to install and use the Fortify Extension for Visual Studio to analyze, audit, and remediate your code to resolve security-related issues in solutions and projects.

Supported languages and vulnerability categories

SAST Aviator is verified by OpenText to maximize accuracy. The extent to which a particular vulnerability category in a certain programming language is supported by SAST Aviator may differ based on the amount of verification and optimization that has already been performed. There are three classes:

Class	Description
Supported with automatic suppression	Cases with a high degree of confidence. By default, SAST Aviator will perform automatic suppression of false positives.
Supported without automatic suppression	Cases where confidence is yet to be established to the same standard. By default, SAST Aviator does not perform automatic suppression.
Not supported	A small set of cases that cannot be handled by SAST Aviator.

The underlying LLM used by SAST Aviator evolves over time. Because not every LLM version is immediately available in all cloud hosting locations used by SAST Aviator, different instances of SAST Aviator may use different LLM versions at any point in time. The LLM version in use determines the classification of cases. Generally, on newer LLMs, more classes can be moved to “automatic suppression”.

The following overview lists how language/category combinations are classified in the current version of SAST Aviator for off-cloud and hosted customers.

Supported language/category combinations with automatic suppression

- Java
  - All categories except explicitly non-supported ones.
- .NET
  - Dynamic Code Evaluation: Serializable delegate
  - Password Management: Password in Configuration File
  - System Information Leak: External
  - Header Manipulation
  - Credential Management: Hardcoded API Credentials

- Server-Side Request Forgery
- Value Shadowing
- Mass Assignment: Insecure Binder Configuration
- ASP.NET MVC Bad Practices: Model with Required Non-Nullable Property
- Value Shadowing
- SQL Injection
- ASP.NET MVC Bad Practices: Optional Submodel with Required Property
- Password Management: Hardcoded Password
- Privacy Violation
- Cross-Site Scripting: Reflected
- Path Manipulation
- Open Redirect
- Mass Assignment: Sensitive Field Exposure
- XML Injection
- XPath Injection
- Null Dereference
- Unreleased Resource: Unmanaged Object
- Portability Flaw: File Separator
- ASP.NET MVC Bad Practices: Controller Action Not Restricted to POST

Supported without automatic suppression

- All other language/category combinations supported by OpenText SAST, except explicitly excluded cases.

Not supported

- One vulnerability category is explicitly not-supported:
  - Privilege Management: Unnecessary Permission.

Note: The verification of this category issues requires access to the complete source code at once, which is not compatible with the way SAST Aviator functions.

Get Started

Perform the following steps to get started with SAST Aviator:

1. "Download fcli container" below
2. "Register customer administrator" below
3. "Generate tokens for individual users" on the next page
4. "Create application" on page 20
5. "Trigger audit" on page 21

Download fcli container

Download fcli **dev_v3.x** from https://github.com/fortify/fcli and extract it.

Register customer administrator

Prerequisite

Ensure the tenant is registered. Contact OpenText Support for the details.

To register a customer administrator:

1. Create a 4096-bit RSA key pair in PEM format using a cryptographic tool available in your organization.
   a. (Optional) Use OpenSSL (https://www.openssl.org/) to generate the RSA key pair.
      i. Generate a 4096-bit RSA private key (private_key.pem):

```
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:4096 -out private_key.pem
```

      ii. Extract the public key (public_key.pem) from the private key:

```
openssl rsa -in private_key.pem -pubout -out public_key.pem
```

2. Save the private key (private_key.pem) for later use.
3. Share the public key (public_key.pem) and other necessary details with OpenText Support to register the customer administrator.

You will be notified once the registration is complete. Administrator registration indicates that your order has been fulfilled, allowing you to use the tool functionality.

Generate tokens for individual users

Prerequisite

You must be a registered customer administrator.

To create an access token for a specified user:

1. In the command prompt, navigate to the path where fcli is extracted.
2. (Optional) View the various administrator and user operations available.

```
fcli aviator -h
```

Argument	Description
-h, --help	Shows the help message and exits.
admin-config	Manages the SAST Aviator administrator configurations (start here).
session	Manages the SAST Aviator user sessions (start here).
entitlement	Manages the SAST Aviator entitlements.
app	Manages the SAST Aviator applications.
ssc	Use SAST Aviator with SSC.
token	Manage SAST Aviator access tokens.

Note: Use **admin-config** to view entitlement, manage applications, and generate tokens and **session** to audit.
Ensure to create an **admin-config** before performing administrator tasks and a user **session** before auditing.

3. Create an administrator configuration for interacting with SAST Aviator.

```
fcli aviator admin-config create --url <aviator_server_url> --tenant <tenant_name> --private-
```

```
key <path_to_private_key.pem>
```

Note: The `--private-key` can be the file containing the key or the key itself. Ensure the key is in PEM format.

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

An admin session is created.

4. Generate the user access token.

```
fcli aviator token create --email <admin_email_id> --name <custom_token_name> --save-token
<output_file>
```

Optional arguments	Description	Default value
--save-token	Save the generated raw token string to the specified file. By default, the string is in json format.	NA
-o, --output	Specify the token format. The available formats are csv, table, expr, json, xml, and yaml.	NA
--to-file	Specify a file to save the output.	NA
--end-date	Specify the token expiration date in YYYY-MM-DD format.	30 days from the date of creation.

Note: OpenText recommends using `--save-token` optional argument to ease the user experience when using the access token.

Create application

Prerequisite

- You must be a registered customer administrator.
- Ensure to have a valid entitlement. See "Entitlement model" on page 8 section.

To create an application:

1. Create an administrator configuration for interacting with SAST Aviator.

```
fcli aviator admin-config create --url <aviator_server_url> --tenant <tenant_name> --private-
key <path_to_private_key.pem>
```

Note: The `--private-key` can be the file containing the key or the key itself. Ensure the key is in PEM format.

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

2. Create an application.

```
fcli aviator app create <aviator_application_name>
```

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

An application is created and assigned to the first available entitlement.

Trigger audit

Prerequisite

- You must be a customer user.
- Ensure you have the following:
  - At least one application is assigned to the entitlement.
  - Valid user access token.

To trigger an audit:

1. Log in to your Fortify Software Security Center session.

```
fcli ssc session login --url <ssc_url> -u <user_name> -p <ssc_password>
```

2. Create a user session to interact with SAST Aviator.

```
fcli aviator session login --url <aviator_server_url> --token <access_token>
```

Note: The default value for `--token` is a file path. To use other formats for the access token, prefix the value with `file:<local file containing key>` or `string:<key string value>` or `env:<env-var name containing key>`.

Ensure to create a user session before auditing.

If you cannot locate your access token, contact your customer administrator.

Optional argument	Description	Default value
--av-session, --aviator-session	Name of the Aviator user session.	default

3. Audit the application.

```
fcli aviator ssc audit --av <application_version_name:id>
```

Optional arguments	Description	Default value
--app	Name of the Aviator application. If the name is not specified, build ID of the FPR is considered.	FPR build ID
--tag-mapping	Override the default tag mapping using the YAML file. See "Audit tag mapping" on page 9.	tag mapping.yaml
--ssc-session	Name of the SSC session to use for auditing.	default
--av-session, --aviator-session	Name of the Aviator user session.	default

It may take a few minutes to process the FPR. The duration depends on the size of the FPR.

Note: You can use the same access token to audit multiple FPRs on different terminals at the same time.
You can audit an FPR only once.

4. Once the SAST Aviator processes the FPR, the Action status and the number of audited issues will be displayed. SAST Aviator uploads the audited FPR back to the Fortify Software Security Center application version.
   a. Open the Fortify Software Security Center application and go to **Applications** > **Artifacts**.
   b. Click each row to view the Audit details, such as analysis tag, remediation comment, and the highlighted vulnerable code segment.

Manage tenant information

Prerequisite

You must be a registered customer administrator.

You can perform the following administrator tasks using the SAST Aviator fcli:

- "admin-config" on the next page
  - Create an administrator configuration.
  - List the administrator configurations.
  - Delete an administrator configuration.
- "tokens" on page 24

- Create a user access token.
- List the access tokens.
- Validate, revoke, or delete the access tokens.

- "applications" on page 26
  - Create an application.
  - Get the application details.
  - List the available applications.
  - Update or delete the applications.
- "entitlements" on page 27
  - Retrieve the entitlement details for a specified tenant.

Note: You must create an **admin-config** to interact with SAST Aviator before performing an administrator task.

admin-config

The following tasks can be performed using the **fcli aviator admin-config** command.

- Create an administrator configuration for interacting with SAST Aviator.

```
fcli aviator admin-config create --url <aviator_server_url> --tenant <tenant_name> --private-key
<path_to_private_key.pem>
```

Note: The `--private-key` can be the file containing the key or the key itself. Ensure the key is in PEM format.

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

- List the available SAST Aviator administrator configurations.

```
fcli aviator admin-config list
```

- Delete the SAST Aviator administrator configuration.

```
fcli aviator admin-config delete --admin-config <administrator_configuration_name>
```

tokens

The following tasks can be performed using the **fcli aviator token** command.

- Create an access token for a user.

```
fcli aviator token create --email <admin_email_id> --name <custom_token_name> --save-token <output_
file>
```

Optional arguments	Description	Default value
--save-token	Save the generated raw token string to the specified file. By default, the string is in json format.	NA
-o, --output	Specify the token format. The available formats are csv, table, expr, json, xml, and yaml.	NA
--to-file	Specify a file to save the output.	NA
--end-date	Specify the token expiration date in YYYY-MM-DD format.	30 days from the date of creation.

Note: OpenText recommends using `--save-token` optional argument to ease the user experience when using the access token.

- Retrieve a list of access tokens for a specified user within an Aviator tenant.

```
fcli aviator token list --email <admin_email_id>
```

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

- Validate an existing access token for a user within an Aviator tenant, checking its authenticity and status.

```
fcli aviator token validate --email <admin_email_id> --token <access_token>
```

Note: The default value for `--token` is a file path. To use other formats for the access token, prefix the value with `file:<local file containing key>` or `string:<key string value>` or `env:<env-var name containing key>`.

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

- Revoke an access token for a user within an Aviator tenant. This task invalidates the access token without permanently deleting it.

```
fcli aviator token revoke --email <admin_email_id> --token <access_token>
```

Note: The default value for `--token` is a file path. To use other formats for the access token, prefix the value with `file:<local file containing key>` or `string:<key string value>` or `env:<env-var name containing key>`.

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

- Delete an existing access token for a user.

```
fcli aviator token delete --email <admin_email_id> --token <access_token>
```

Note: The default value for `--token` is a file path. To use other formats for the access token, prefix the value with `file:<local file containing key>` or `string:<key string value>` or `env:<env-var name containing key>`.

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

applications

The following tasks can be performed using the **fcli aviator app** command.

- Create applications.

```
fcli aviator app create <aviator_application_name> --tenant <tenant_name> --private-key <path_to_
private_key.pem>
```

Note: The `--private-key` can be the file containing the key or the key itself. Ensure the key is in PEM format.

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

- Get the details of an existing Aviator application for a particular tenant.

```
fcli aviator app get <application_id>
```

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

- List Aviator applications for a particular tenant.

```
fcli aviator app list
```

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

- Rename an existing SAST Aviator application identified by its ID.

```
fcli aviator app update <application_id> -n <new_application_name>
```

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

- Delete an Aviator application by its ID.

```
fcli aviator app delete <application_id>
```

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

entitlements

Retrieve the following entitlement details for a specified tenant in SAST Aviator.

- Total number of entitlements available.
- Number of active entitlements.
- Total number of applications linked to the entitlements that are already consumed.
- Number of remaining applications under the available entitlements.
- Expiration date for each entitlement.

```
fcli aviator entitlement list
```

Optional argument	Description	Default value
--admin-config	Name of the Aviator administrator configuration.	default

FAQs

What should you do if you disagree with SAST Aviator audit result?

OpenText recommends creating a support request and sharing the FPR.

What should you do if you cannot locate your access token?

Contact your customer administrator.

What should you do if you run out of entitlements?

Contact your account representative.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email.

Note: If you are experiencing a technical issue with our product, do not email the documentation team. Instead, contact Customer Support at https://www.microfocus.com/support so they can assist you.

If an email client is configured on this computer, click the link above to contact the documentation team and an email window opens with the following information in the subject line:

Feedback on User Guide (Core SAST Aviator 25.2.0)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to fortifydocteam@opentext.com.

We appreciate your feedback!