



Siemplify

How-Tos

Table of Contents

- Cases 3
 - How to get Insights on a Case..... 4
 - Changing Alert Priority instead of Case Priority 6
 - Cases FAQ 8
 - How can I perform a batch action on several cases at once? 9
 - Can we measure how long security analysts take to close or raise a case? 10
 - How do I send an email from Siemplify and log the response?..... 11
 - How do I mark a Case as an Incident?..... 12
 - How can I view the contents of closed cases? 13
 - How can I view original SIEM data in a case? 14
 - How can I merge multiple cases? 15
- Integrations..... 16
 - Building a Custom Integration 17
 - Creating a Custom Action 27
 - Supporting Multiple Instances 35
- Playbooks 45
 - Understanding Playbook Metrics 46
 - Using Alert Type Trigger in a Playbook 52
 - Using the Expression Builder 54
 - Bulk Actions and Filters in Playbooks..... 68
 - Creating Playbook Blocks 71
 - Boost playbook development efficiency with the Playbook Simulator..... 72
 - Playbooks FAQ 79
 - How do I scan multiple URLs in VirusTotal?..... 80
 - Is there an auto-assignment feature that assigns cases to users? 81
 - How do I put elements of the case data into an email message? 82
 - How do I export and import Playbooks? 83
 - Can I scan URLs received by email? 84
 - Is it possible to send messages to a phone number? 85
 - How many playbooks can be assigned to a single alert?..... 86
 - How is an SLA calculated? 87
 - What is the difference between Alert Grouping and Alert Overflow?..... 88
- Multi-Tenancy Features..... 89
 - Supporting Multi-Tenancy (Environments)..... 90
- Management 91
 - Using Advanced Reports 92
- Administrative 96

Alert Grouping Rules	97
Siemplify Logs	100
Set Shared Folder for Remote Backup (Linux)	103
Change SSL Certificates for Server and Client.....	104
Import and Export PostgreSQL Databases	105
White List for HTML Templates	109
How to increase the timeout of Jobs/Connectors/Actions in Siemplify	110
Administrative FAQ	111
How do I change a password for a DB user?	112
What are Networks used for?	113
Can I upgrade Java in Siemplify?.....	114
What is Siemplify's Password Policy?	115
Authentication, Permissions and Access	116
Configure SAML Provider in Siemplify.....	117
SAML Configuration for G Suite	120
SAML Configuration for Azure	126
Using Permissions in Siemplify	129
How to Configure LDAP	132
Create View Only Users	136
Authentications and Permissions FAQ	139
How can I prevent users from changing playbooks?	140
How can I disable a user account in Siemplify?.....	141
How do I generate an API key to access Siemplify's API?	142
Daily Tasks	143
Using the Search page.....	144
Open a ticket for Siemplify Support	147
Notifications	154
Daily Tasks FAQ	155
Does Siemplify have custom lists?.....	156
Where can I see the Siemplify Version number?	157
How can I manage contacts in Siemplify?	158
Connectors	159
ElasticSearch Connector: Mapping Custom DateTime	160
Defining Environments in Connectors.....	162
Creating a Custom Connector	164
Siemplify Installation Guide.....	170
Installer Attributes	171
Basic Installation (All-In-One).....	173
Basic Installation (External Database).....	174
Scaled Installation (Multi-Node)	175

Upgrading Siemplify	177
Database Maintenance	178
Troubleshooting	181
Hardening and Security Procedures	182
High Availability Installation.....	184
Overview	185
Prerequisites	187
Deployment Components.....	188
Deployment Process.....	189
Server Specifications	190
Upgrade Considerations	191
Shutdown and Restart HA Servers.....	192
Disaster Recovery Installation	197
Overview	198
Prerequisites	200
Deployment Process.....	201
PostgreSQL & Repmgr Installation Procedure	202
Extract and Install.....	203
Configure PostgreSQL.....	204
Restart PostgreSQL	206
Configure REPMGR (primary node).....	207
Configure REPMGR (replica node).....	208
Configure REPMGR (replica node – DR)	210
PreFailover Actions	212
Install Siemplify application Nodes.....	213
Install Siemplify DB availability detection script.....	216
Make DR system primary	217
Return to main site	218

Cases

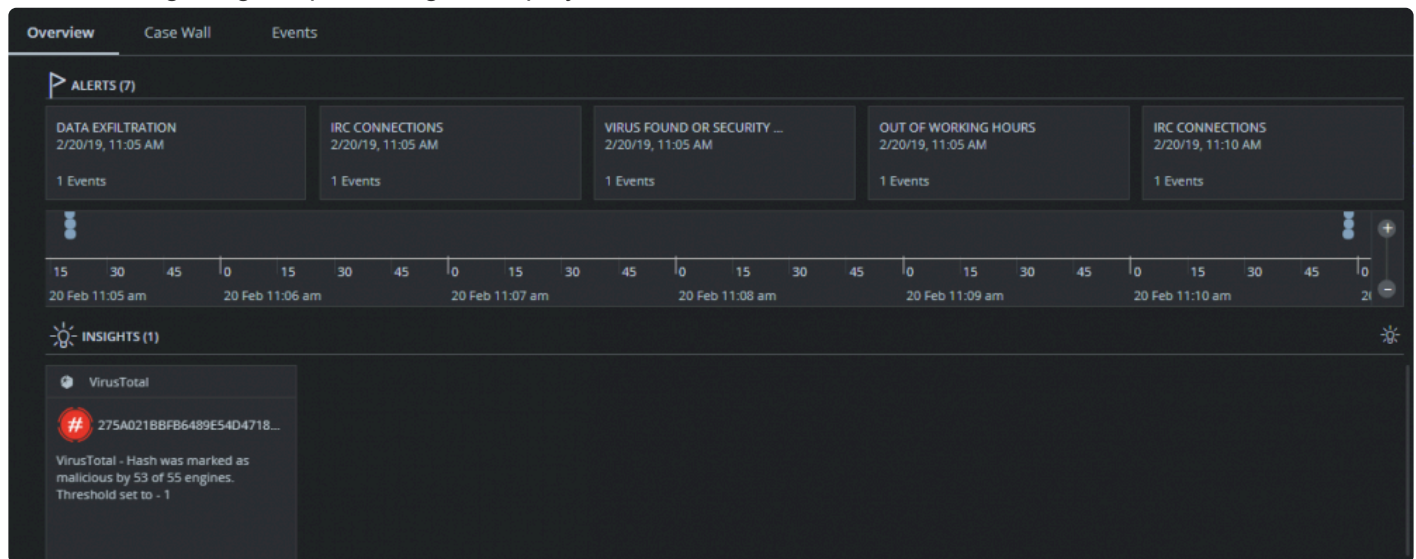
How to get Insights on a Case

Insights provides the ability to pull out key information for any part of a case and display it to analysts so that they can easily analyze the information without having to click to pull up information that is relevant to what they are investigating. Insights can be created as part of a Manual Action or they can also be created within a playbook.

Entity Insights

Entity Insights are created within a manual action, playbook action or as part of an integration. They contain the entity information and its metadata. You normally use this type of insight when performing enrichment on an entity and want to show specific enrichment data to the analyst when the entity hits a threshold or has information that is important for the analyst to know.

The following image depicts Insights displayed on the Case Overview screen.



To add an Entity Action within a Playbook:

To run as an action within a playbook, drag and drop the Add Entity Insight action into the playbook at the desired location.

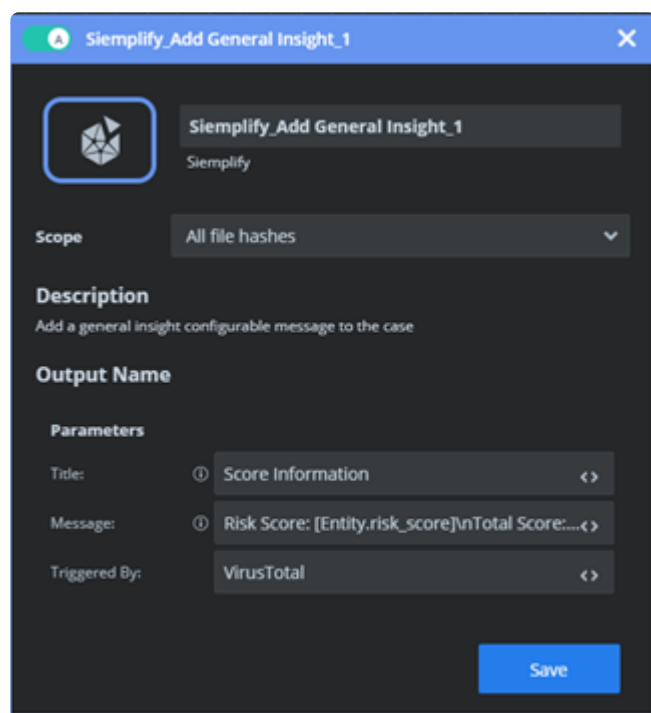
Click to open the Add Entity Insight Action and fill out the following fields:

Action Description: This is the text that you will see when looking at the action in the playbook designer. It is recommended to write a meaningful description so that when you review the playbook later you will be able to identify the Action at a glance.

Action Scope Select the required entity so that you only run an action on the information that you are interested in using. For example, Scan Hash by Virus total only needs to have hash information. So here, you would choose FileHash from the drop-down list because that is the information needed for the action.

Message The message field can have either static text or you can use placeholders. To use placeholders click <>. This opens the placeholders screen. The available placeholders that I can use are Alert, Case, Entity, Environment, Event, and Playbook. Note that you can also drill down to specific JSON results if required.

Select the object that you want. For example, select Entity. You can use as many placeholders (and free text) as you need in the Message.



The screenshot shows a configuration window titled "Siemplify_Add General Insight_1". It features a dark theme with a blue header bar. The window contains the following sections:

- Header:** A green status icon, the title "Siemplify_Add General Insight_1", and a close button (X).
- Icon:** A blue square icon with a white geometric design.
- Name:** A text field containing "Siemplify_Add General Insight_1" and a sub-label "Siemplify".
- Scope:** A dropdown menu currently set to "All file hashes".
- Description:** A section titled "Description" with the text "Add a general insight configurable message to the case".
- Output Name:** A section titled "Output Name".
- Parameters:** A section containing three rows:
 - Title:** A text field with "Score Information" and a help icon (i) on the left and a double arrow icon (↔) on the right.
 - Message:** A text field with "Risk Score: [Entity.risk_score]\nTotal Score:...." and a help icon (i) on the left and a double arrow icon (↔) on the right.
 - Triggered By:** A dropdown menu set to "VirusTotal" with a double arrow icon (↔) on the right.
- Save Button:** A blue button labeled "Save" at the bottom right.

Note that General insights can only be created within a playbook action. They give you control over identifying what triggered the insight, the title of the insight and the message of the insight.

✿ This article is relevant for 4.3 Software Version

Changing Alert Priority instead of Case Priority

Up until Release 5.6.0, the analyst could only change the priority of a case and could not touch the priority of an alert. The drawback with this approach is that once you have different alerts grouped into a case, each incoming alert with an attached playbook could alter the Case priority. So for example, if an Alert is ingested at 10:01 with a Playbook that defines the Case as Critical; and then another Alert is grouped into the same case at 10:05 with a Playbook that defines the Case as low priority, the entire Case would be classified as low priority causing important issues to go undetected.

With Release 5.6.0, a new method has been adopted which will solve this problem and allow greater flexibility with case priorities in general.

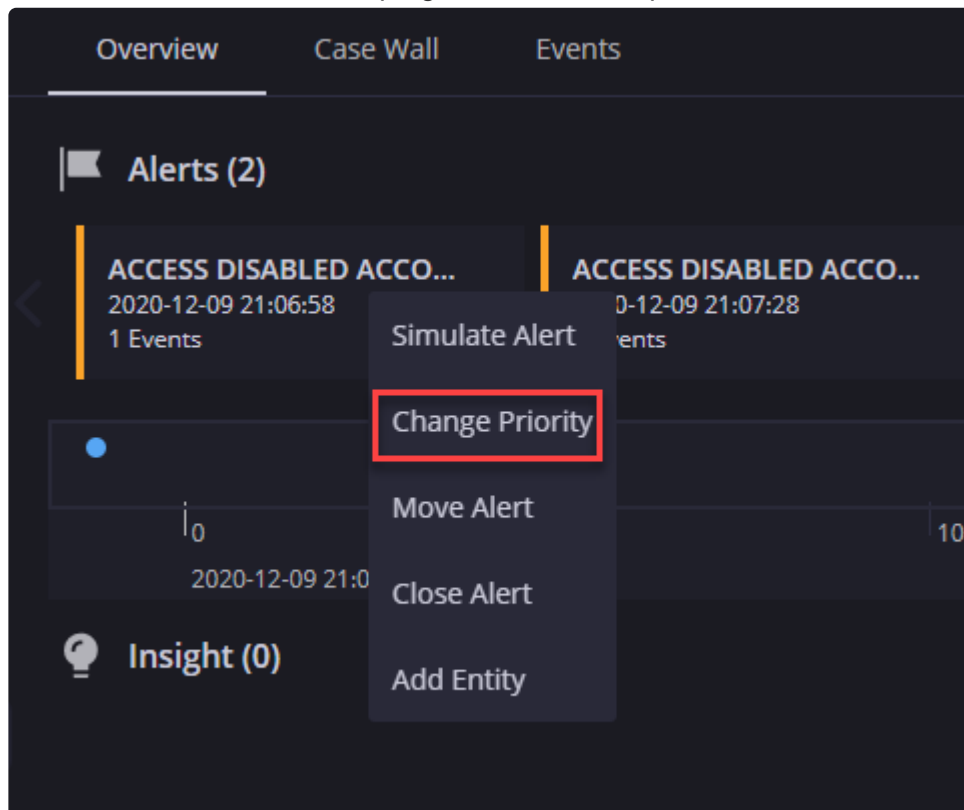
From now on – you can change the Alert Priority within a Case. Each case will inherit the highest priority of the grouped alerts. This way, going back to the example above, even if a later alert had a priority of low, this would no longer override the critical priority assigned to the case by the previous alert.

How can I change the priority of the alert?

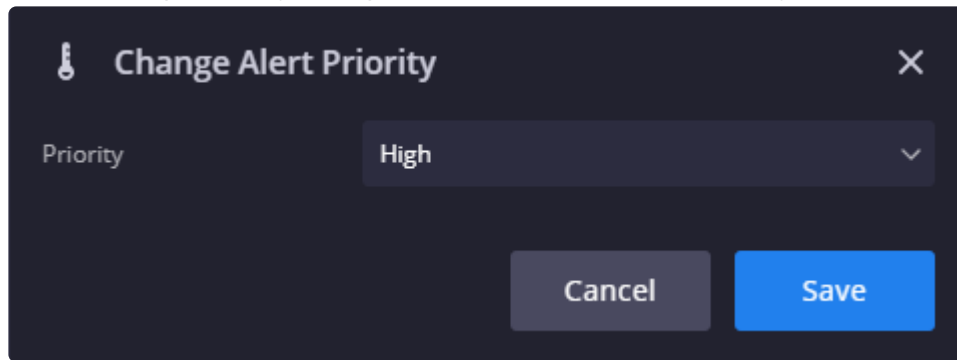
There are two ways you can change the priority of the Alert:

- Using the new Change Alert Priority Action – either in a Playbook or used as a manual Action.
- Change Priority through Alert itself as in the procedure below:

1. In the Cases screen in the Simplify Platform, hover over an Alert in the case.
2. Hover over the alert in the top right and in the drop-down list, select Change Priority.



3. In the Change Priority dialog box, select the required Priority and click Save.

A dark-themed dialog box titled "Change Alert Priority" with a close button (X) in the top right corner. Below the title, there is a label "Priority" followed by a dropdown menu currently showing "High" with a downward arrow. At the bottom of the dialog, there are two buttons: "Cancel" and "Save".

Change Alert Priority X

Priority High

Cancel Save

* You can still change the priority of the Case if required but Siemplify does not recommend this as best practice.

Cases FAQ

How can I perform a batch action on several cases at once?

Answer

1. Navigate to the Search page and search for the cases using the required filter.
2. Tick the checkboxes of the required cases.
3. Select one of the drop-down actions.

Search Results 1-21 of 21

<input type="checkbox"/>	ID	Assigned User	Title	Time	Tags	Products	Ports	Outcomes	Status	Environment	Priority	Stage	Ticket ID
<input type="checkbox"/>	1	Admin	DLP_Prod...	2/20/19, 11:05 AM	4TimesTag,plp...		633,770	Allowed,blocked	Open		High	Triage	8c280635-
<input checked="" type="checkbox"/>	5	Im-viewer	Phishing e...	2/20/19, 11:07 AM	4TimesTag,3TL...			allowed	Open		High	Triage	d824993d-
<input checked="" type="checkbox"/>	9	Im-ciso	IPS_Product	2/20/19, 4:51 PM	4TimesTag		633,770	Allowed,blocked	Open		High	Triage	e0e2e8d4-
<input type="checkbox"/>	12	Im-socmanager	Manual ca...	2/20/19, 7:08 PM	Manual Case,4T...				Open		Informative	Triage	
<input type="checkbox"/>	10010	Admin	Phishing e...	2/21/19, 11:08 AM				allowed	Open	AAA	High	Triage	b9223b5e-
<input checked="" type="checkbox"/>	10012	Abc	Phishing e...	2/21/19, 11:08 AM	3TimesTag,1TL...			allowed	Open	ABC	High	Triage	99d123dd-
<input type="checkbox"/>	10013	Aaa	Phishing e...	2/21/19, 11:08 AM				allowed	Open	DC	High	Triage	1264ad0d-
<input type="checkbox"/>	10014	Im-tier-2	Phishing e...	2/21/19, 11:09 AM				allowed	Open	123-asd	High	Triage	5fec9818-
<input type="checkbox"/>	10015	Tier-1	DLP_Prod...	2/21/19, 11:19 AM	Simulated Case		770	blocked	Open		Informative	Triage	7509fc4-
<input type="checkbox"/>	10016	Tier-2	DLP_Prod...	2/21/19, 11:19 AM	Simulated Case		770	blocked	Open		Informative	Triage	8805c596-
<input type="checkbox"/>	10017	Tier-3	DLP_Prod...	2/21/19, 11:19 AM	Simulated Case		770	blocked	Open		Informative	Triage	aec7c042-
<input type="checkbox"/>	10018	Soc Manager	DLP_Prod...	2/21/19, 11:19 AM	Simulated Case		770	blocked	Open		Informative	Triage	d37771c0-
<input type="checkbox"/>	10019	Ciso	DLP_Prod...	2/21/19, 11:19 AM	Simulated Case		770	blocked	Open		Informative	Triage	6edfb231-
<input type="checkbox"/>	10020	Tier-1	DLP_Prod...	2/21/19, 11:19 AM	Simulated Case		770	blocked	Open		Informative	Triage	0a9f55d9-

Export to CSV
Close case
Reopen case
Change priority
Assign case
Add tag
Merge cases

Can we measure how long security analysts take to close or raise a case?

Answer

Siemplify allows building processes for cases and measuring the time it takes to complete them as well as the stages of each process. Security analysts can take a case through the stages by following a playbook or manually. Automation can also follow a predefined process.

Let's use the dashboards to measure the stages each case is going through. To do that, we can either create a new table widget that shows cases avg. handling time or use the built in ROI widgets.

1. Navigate to the Dashboard screen and add a new widget with the following details.

The screenshot shows the 'Widget Settings' dialog for a 'Table' widget. The title is 'Case Handling Time'. The environment is set to 'All Environments' and the time range is 'Last Year'. The widget width is set to 1 column. The preview shows a table with the following data:

	Data Exfiltration	Phishing Email	Working Hours
Triage	2m	-	-

The settings on the left are: Number of: Cases, Calculate field: Handling avg time, Axis A: Tag, Axis B: Stage. The filters on the right include: Tags, Analysts, Case Status, Priority, Importance, Stages, Case Close Reasons, and Case Root Causes. The bottom right has 'Cancel' and 'Save' buttons.

2. Now look at the Dashboard and you will see this widget displays that the average handling time is 15 minutes.

How do I send an email from Siemplify and log the response?

Answer

This can be done either by running a manual action or via a playbook, by selecting the Send Email and Wait action.

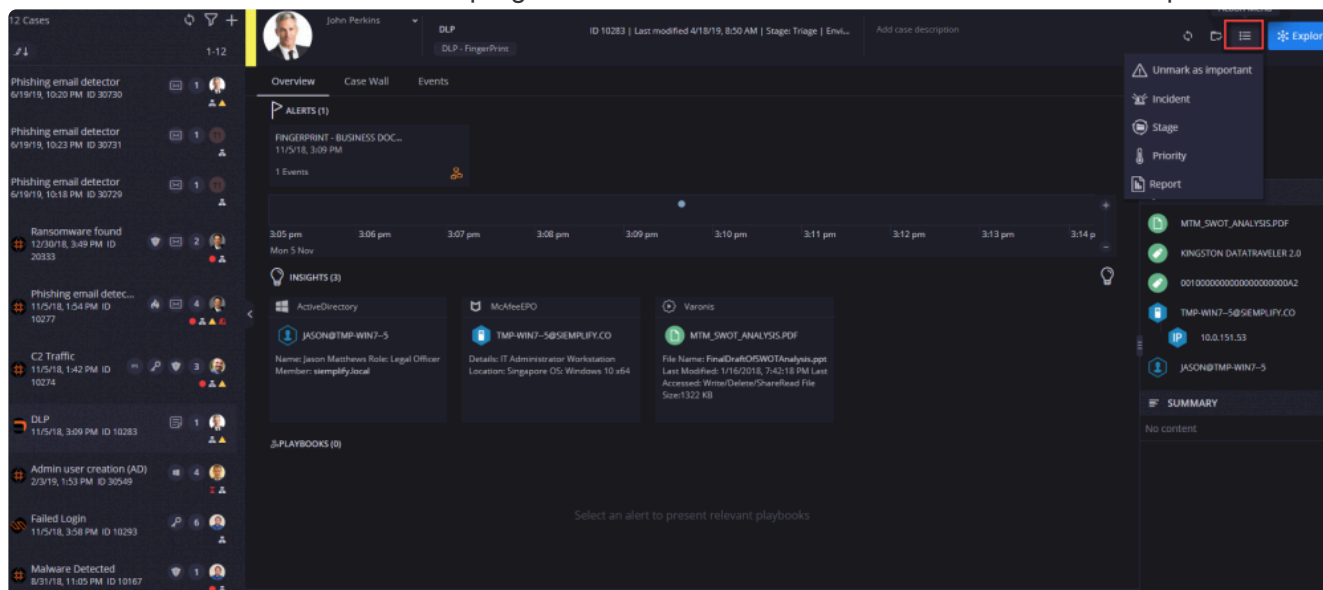
This action will send an email via SMTP, and then query the mailbox periodically for a response (looking for a unique ID to identify the correspondence). Once received, the action will fetch it into Siemplify.

The response can be seen on the Case Wall and used as an input for other actions in the playbook.

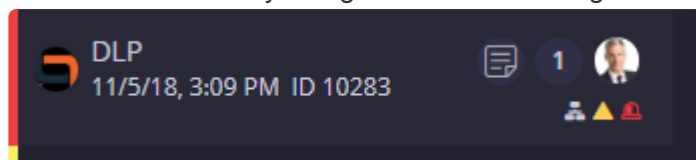
How do I mark a Case as an Incident?

Answer

1. Navigate to the Cases screen.
2. Highlight the required case.
3. Click on the Actions menu on the top right of the screen and choose Incident from the drop-down list.



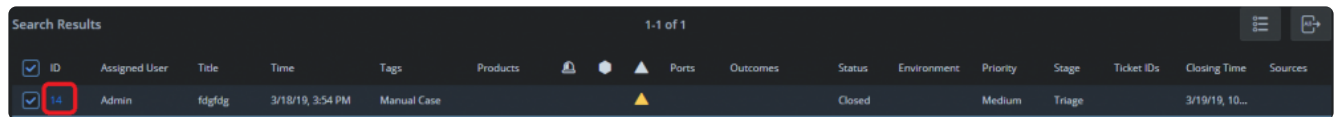
4. Click Yes in the confirmation dialog box. Note that the screen refreshes and the new Incident now appears with the Incident icon in the Cases list marked with the red Critical side bar. In addition, the case is automatically assigned to SOC Manager user or the user with SOC Manager role.



How can I view the contents of closed cases?

Answer

1. Navigate to the Search page.
2. In the Filter section, select Status > Closed.
3. Click Apply.
4. In the list of Closed Cases, select the required case and click on the ID number.



Search Results 1-1 of 1

<input checked="" type="checkbox"/>	ID	Assigned User	Title	Time	Tags	Products				Ports	Outcomes	Status	Environment	Priority	Stage	Ticket IDs	Closing Time	Sources
<input checked="" type="checkbox"/>	14	Admin	fdgfdg	3/18/19, 3:54 PM	Manual Case							Closed		Medium	Triage		3/19/19, 10...	

5. You will be redirected to the original case contents.

How can I view original SIEM data in a case?

Answer

Select a case from the queue. The Overview tab will show you the alerts in it, their timeline, extracted entities and insights that were collected by automation.

To see the original event that generated the alerts, go to the Events tab. Clicking on each event will open all details associated with it in the right Context Details pane.

The screenshot displays the Simplify SIEM interface. The top navigation bar includes 'Overview', 'Case Wall', and 'Events'. The 'Events' tab is active, showing a table of alerts. The table has columns: Alert ID, Alert Name, Event Name, Event Type, Product, Artifacts, Port, Outcome, and Time. A single alert is listed with ID 1, Name 'SUSPICIOUS PHISHING EMAIL', Event Name 'Email check', Product 'Phishing email detector', Outcome 'allowed', and Time '3/18/19, 4:54 PM'. The right pane, titled 'Context Details', shows the event 'Email check' and a search bar. Below this, 'HIGHLIGHTED FIELDS' are listed in a table:

FIELD NAME	VALUE
Device/Vendor	Email server
Start Time	3/18/19, 4:54 PM
End Time	3/18/19, 4:54 PM
Name	Your New Salary Notification
Category/Outcome	allowed
Source User Name	f.attacker4@gmail.com
Destination User Name	vickie.b@siemplify.co

Below the highlighted fields, 'DEFAULT' fields are listed in another table:

FIELD NAME	VALUE
sourcetype	Phishing email detector
Mail_OriginalEmailBody	Hello, You have an important email from the Human Resou...
DestinationURL	http://markossolomon.com/F

The bottom of the interface features a 'Write a comment...' text area and icons for adding comments, attachments, and other actions.

How can I merge multiple cases?

You can merge cases from the Search screen.

The screenshot displays the Simplify interface with the 'SEARCH' tab selected in the top navigation bar. The left sidebar contains a 'Filter' section with a search bar and various filter categories like Status, Environments, Tags, Users, Category Outcomes, Ports, and Products. The main area shows 'Search Results' with a table of cases. A red box highlights the 'SEARCH' tab (1), the filter search bar (2), the selection checkboxes in the table (3), and the context menu (4) which includes the 'Merge cases' option. The context menu also lists other actions: Export to CSV, Close case, Reopen case, Change priority, Assign case, and Add tag.

Selected	Assigned User	Title	Time	Tags	Products	Ports	Outcomes	Status	Environm	Priority
<input checked="" type="checkbox"/>	Admin	Win...	7/14/19, ...	Simulat...				O...	Info...	Info...
<input checked="" type="checkbox"/>	Admin	Win...	6/23/19, ...	Out of ...			633	O...	High	High
<input checked="" type="checkbox"/>	Tier-1	DLP...	6/23/19, ...	Data Ex...			770	O...	High	High
<input checked="" type="checkbox"/>	Tier-1	Win...	6/17/19, ...	Simulat...				Audit Suc...	O...	Info...
<input checked="" type="checkbox"/>	Admin	File ...	1/28/19, ...	Data Ex...			84...	O...	Acme...	High

Integrations

Building a Custom Integration

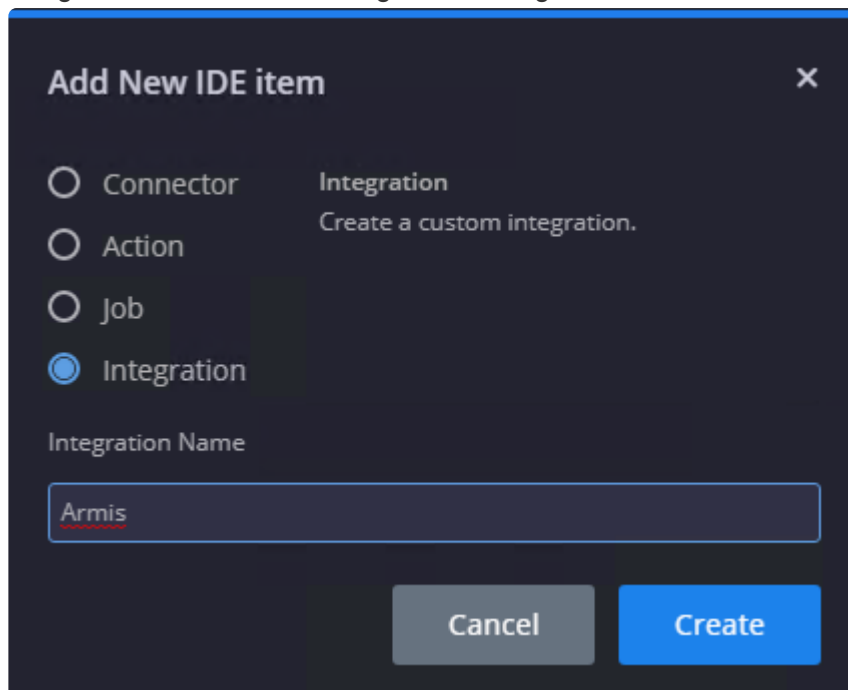
Building a Custom Integration

Simplify users can create custom integrations inside the Simplify IDE with the same structure as Simplify commercial integrations. The custom integrations will appear in the Marketplace and can be configured for different environments so they can be used in Playbooks, manual actions and remote agents. They can also be imported and exported as with other IDE items.

In this How To we will build a custom integration for the Armis product and create a Manager along with a Ping action. Knowledge of Python and object oriented programming is assumed for this tutorial.

Creating the Custom Integration in the IDE

From the IDE screen click the + icon in the upper left hand corner to add a new IDE item. Select the Integration radio button and give the integration a name.



Add New IDE item ✕

☐ Connector **Integration**
Create a custom integration.

☐ Action

☐ Job

☒ Integration

Integration Name

Armis

Cancel Create

The integration will be created and listed on the left hand side with a cog icon that designates it as a custom integration.

Clicking on the Cog Icon will bring up the Integration Settings where the Icon, Description, Python Dependencies and Integration Parameters can be defined. In the following screenshot, an image has been uploaded (this image will appear in the Marketplace with the integration), a brief description has been added and two parameters have been defined. There are no Script Dependencies that need to be added because the only non-standard Python library that will be utilized in this integration is the `requests` library which is already installed on the system.

✿ Script Dependencies are Python libraries that the custom integration will need to import. Dependencies can be added as wheel files, tarballs, gunzip format or python files (.whl, .tar, .gz, .py extensions are supported). Every integration runs in its own virtual environment so feel free to add different versions of libraries even if one is already installed on the system. For example, if there is a newer (or older!) version of `requests` that you would like to use instead of the default on the system (2.20.0 at the time of writing), download the dependency from a reliable source such as PyPi or GitHub and add it to the Script Dependencies for this integration. If a dependency is not installed in the virtual environment, the integration will import it from the system installation if the dependency is installed there.

Creating the Manager

Managers are not strictly necessary for an integration to function, but they are a great idea, especially for integrations that involve third party tools. Managers are essentially wrappers that contain the API logic for the third party tool. Managers should not have any imports from the Simplify SDK. Once the Manager is created, it can be imported as a module into Connectors, Actions and Jobs and its methods can be utilized.

To create the custom manager:

1. In the IDE screen, click the plus icon on the top left.
2. Select Manager and select the Armis integration, and name the Manager.

Add New IDE item ×

☐ Connector

☐ Action

☐ Job

☐ Integration

☒ Manager

Manager
Create a new manager to use with your actions. The manager is an integration module which contains API calls and functions that can be imported and used in your actions.

Manager Name

Integration

Armis ▼

Cancel

Create

3. Edit the script as it appears below.

```

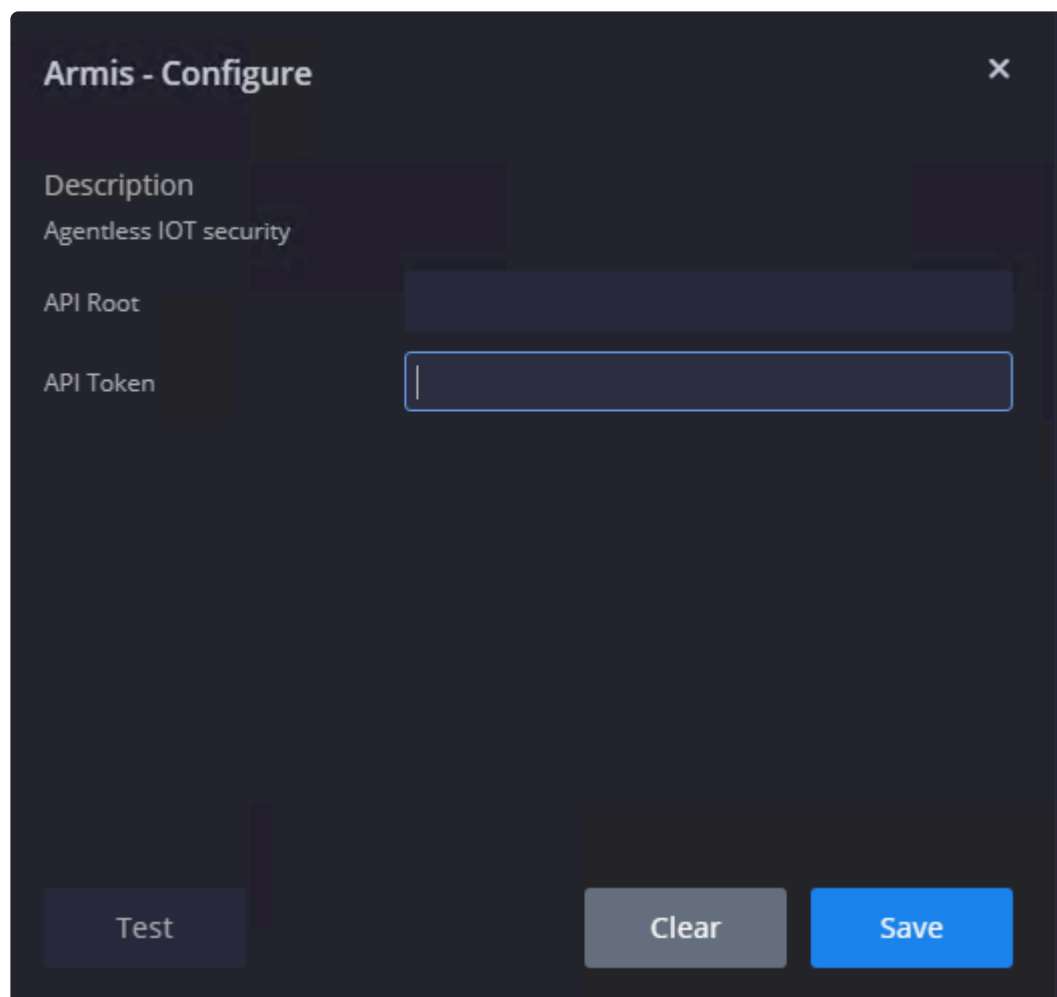
1 import requests
2
3 HEADERS = {'Accept': 'application/json'}
4
5 class ArmisManager(object):
6     def init(self, api_root, api_token):
7         self.api_root = api_root
8         self.api_token = api_token
9         self.session = requests.Session()
10        self.session.headers = HEADERSself.auth()
11
12    def auth(self):
13        endpoint = '{}/api/v1/access_token/'
14        params = {'secret_key': self.api_token}
15        response = self.session.post(endpoint.format(self.api_root), params=params)
16        self.validate_response(response)
17        access_token = response.json()['data']['access_token']
18        self.session.headers.update({'Authorization': access_token})
19        return True
20
21    def get_device_by_ip(self, device_ip):
22        endpoint = '{}/api/v1/devices/'
23        params = {'ip': device_ip}
24        response = self.session.get(endpoint.format(self.api_root), params=params)
25        self.validate_response(response)
26        return response.json()['data']['data']
27
28    @staticmethod
29    def validate_response(res, error_msg="An error occurred"):
30        """
31        Validate a response
32        :param error_msg: (str) The error message to display
33        :param res: (requests.Response) The response to validate
34        """
35        try:
36            res.raise_for_status()
37
38        except requests.HTTPError as error:
39            raise Exception(
40                "{error_msg}: {error} {text}".format(
41                    error_msg=error_msg,
42                    error=error,
43                    text=error.response.content)
44            )

```

✿ Remember, Managers are not mandatory. For simple actions that don't need an API wrapper a Manager is not a necessity. A best practice is to create a custom integration for a collection of small custom actions that can be utilized to parse and transform data for your needs.

Parameters, Marketplace Configuration and the Ping Action

The parameters defined in the Integration Settings will appear in the Marketplace Configuration for the Integration.



Armis - Configure ✕

Description
Agentless IOT security

API Root

API Token

Test Clear Save

After entering the correct credentials, click the Save button and then the Test button.

✿ The Save button must be clicked every time credentials are updated. The Test button runs the Ping action that every integration that interacts with a third party system **MUST** have in order to perform a connectivity test from the Marketplace Integration Configuration screen.

The Test button will fail and display a red X because no Ping action has been configured for this integration.

Armis - Configure

Description

API Root

API Token

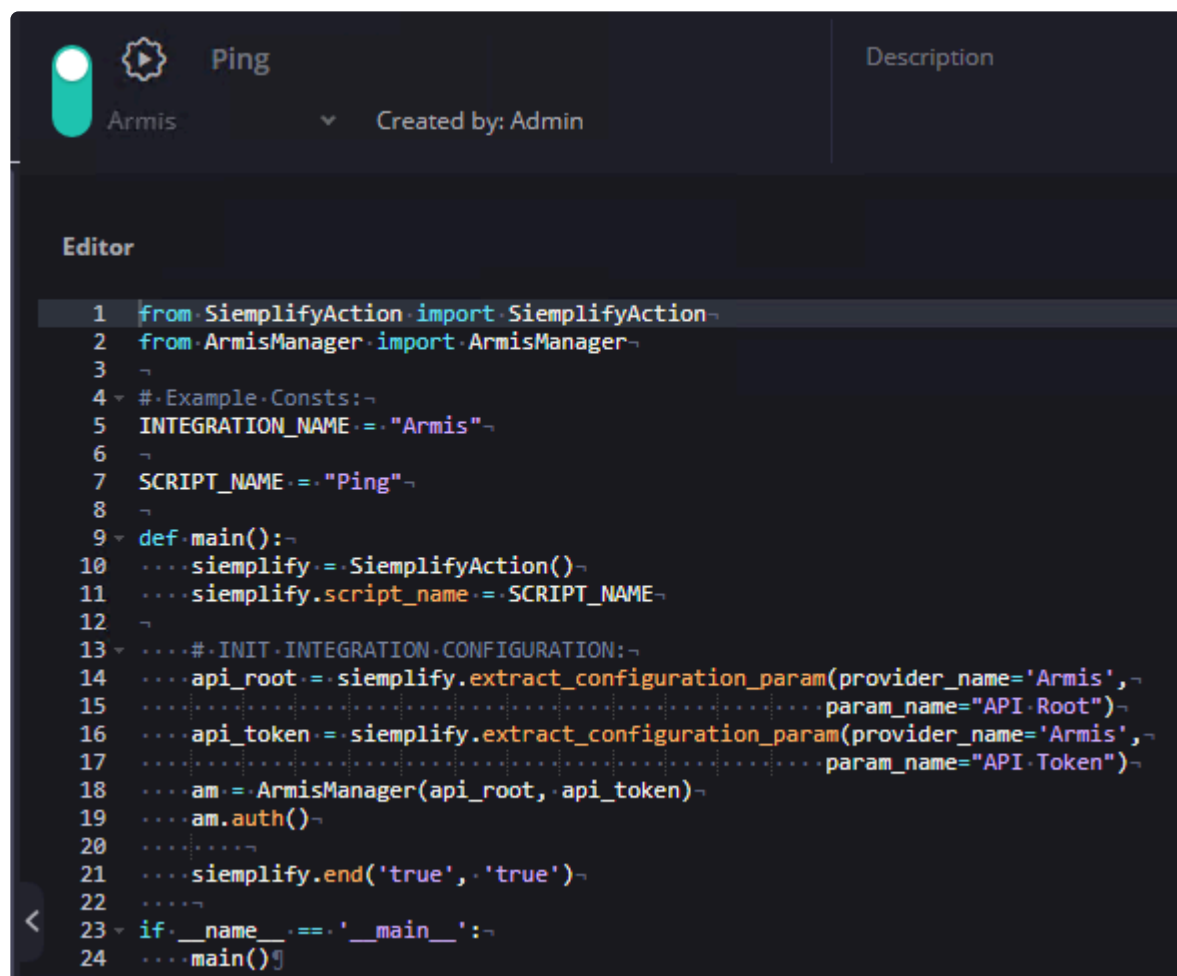
Test

Clear

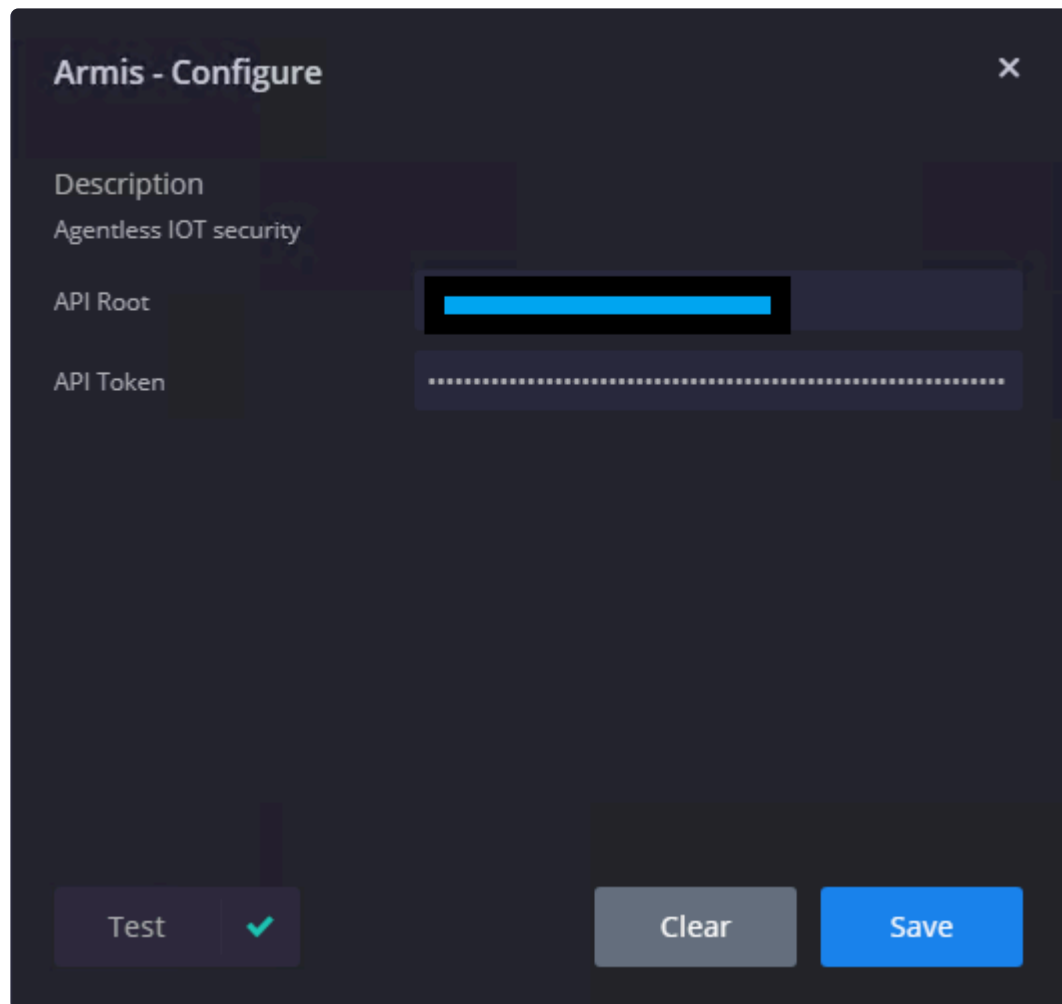
Save

Ping action is not implemented. Go to the IDE and create a Ping action which implements the testing logic.

The logic of the Ping action should be something like a successful authentication. In the Simplify IDE, create a new Action in the Armis Integration with the name Ping. Utilizing the ArmisManager and its auth method, the Action will look like the following:



Enable it by clicking the switch to the left of the integration name (it's already enabled in the screenshot above as shown by the green switch in the upper left hand corner) and Save it. In the Action above, the API Root and the API Token are being imported from the Marketplace Configuration and being passed to the `ArmisManager`. The Manager then calls its `auth` method and if there are no errors, the Test button will return a green checkmark.



Armis - Configure ×

Description
Agentless IOT security

API Root
[Redacted]

API Token
[Redacted]

Test ✓ Clear Save

Just to verify that the logic of the Action is sound, we can enter incorrect credentials, click Save and then run the Test again. The screenshot below shows an obvious credential error.

Armis - Configure [X]

Description
Agentless IOT security

API Root [REDACTED]

API Token [REDACTED]

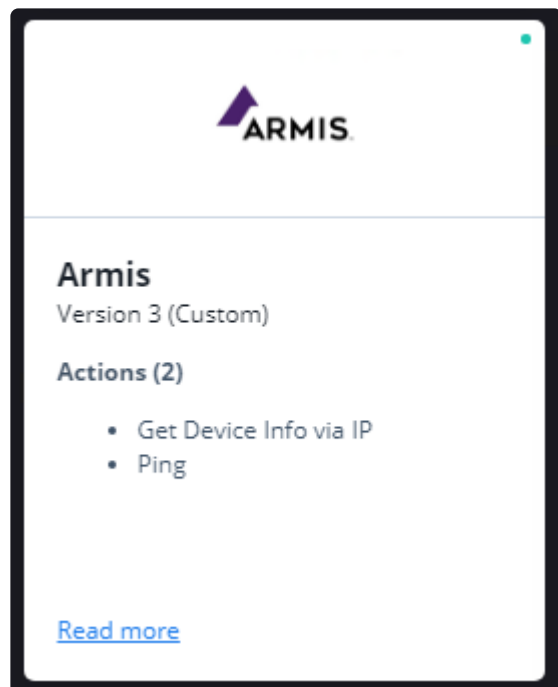
Test [X] Clear Save

! /access_token/?
secret_key=c [REDACTED]
[REDACTED] { "message":
"Invalid secret key.", "success": false
}

* The `extract_configuration_param` method is one of two ways to import Action parameters. Another way is to define the parameters in the Action itself and use the `extract_action_param` method but the Ping action will ALWAYS use the integration's configuration parameters because those are the parameters that must be tested.

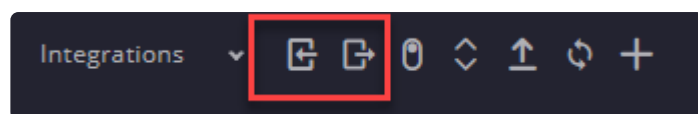
Viewing your Custom Integration in the Marketplace

Navigate to the Marketplace and search for the custom integration you created. If you didn't create an image during the initial configuration, then the default custom image will be assigned to it. Note that Marketplace updates will not override or delete any custom integrations. In the screenshot below, the integration has two Actions that have been created for the integration.



Exporting/Importing in the IDE

Use the Export / Import buttons to back up or share the integration with other parties.



Exporting the integration will download a ZIP file with the relevant Definition, Script, and other configuration files. Crucially, as of Simplify 5.2, the ZIP will not contain the Managers folder with the Manager, so this must be exported manually.

Importing integrations is a similar process. Simply upload a valid ZIP file that contains the proper folder structure and the integration will appear in the IDE and Marketplace.

Simplify SDK Documentation

The Simplify IDE contains a helpful link to the Simplify SDK documentation which can also be found [here](#).

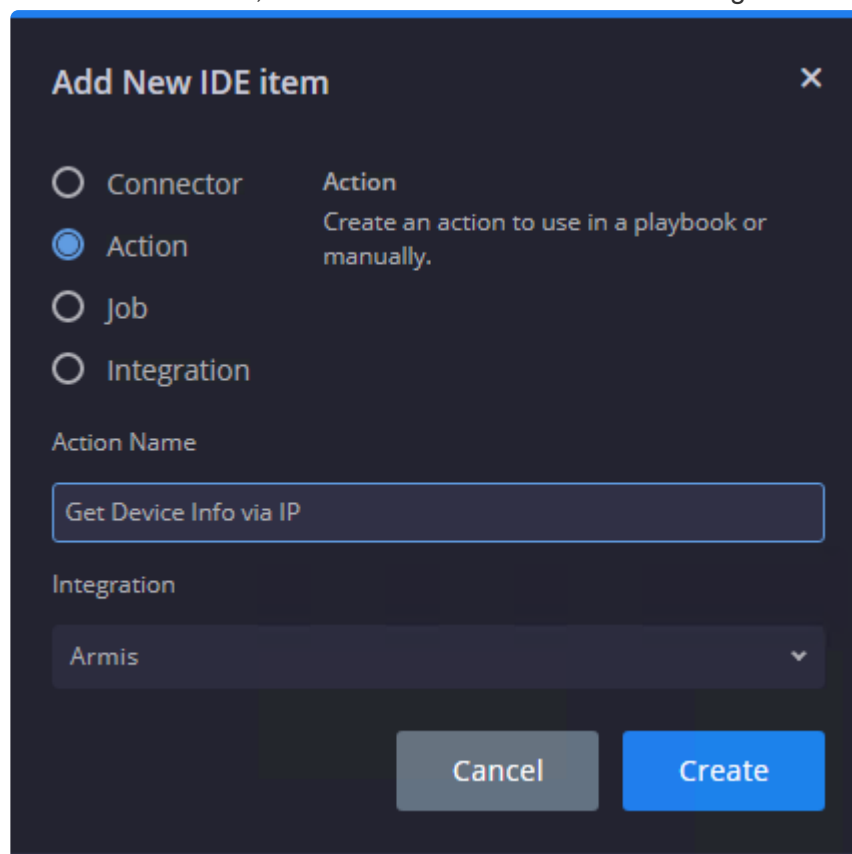
Creating a Custom Action

Overview

In the [Building a Custom Integration](#) How To, we created a Ping Action for the Armis integration. In this tutorial we will create an Action for the Armis integration that enriches entities. Knowledge of Python and object oriented programming is necessary for this tutorial. The SDK documentation and the Building a Custom Integration tutorial will be referenced frequently. Additionally, exploring the SDK modules themselves is highly recommended.

Creating a Custom Action

Navigate to the IDE and click the + sign in the upper left hand corner to Add New IDE Item. Select the Action radio button, name the Action and select the integration then click the Create button.



Add New IDE item ✕

☐ Connector **Action**
Create an action to use in a playbook or manually.

☐ Action

☐ Job

☐ Integration

Action Name

Get Device Info via IP

Integration

Armis ▼

Cancel Create

The IDE will create a new template that has some very useful code comments and explanations. Make sure to give this template a look over when possible.

The Siemplify Action Object

There are two things that must happen in a Siemplify Action. An object must be instantiated from the `Siemp`

`SimplifyAction` class and the object must utilize the class's `end` method to return an output message and a result value.

The Result Value

Every Action will have an Output Name that will represent the result value of what the Simplify Action returns in its end method. The Default Output Name is `ScriptResult` but it can be changed to anything from the IDE.

The screenshot shows the configuration interface for an action in the Simplify IDE. It has three tabs: 'Details' (selected), 'Testing', and 'Debug Output'. Under the 'Details' tab, there is a section for 'Output Name' with a text input field containing 'ScriptResult'. Below this is a 'Polling Configuration' section with a 'Polling Timeout' field. This field is composed of four spinners for 'Days' (0), 'Hours' (0), 'Minutes' (5), and 'Seconds' (0). Below the polling configuration is a 'Default Return Value' text input field containing 'Timeout'. At the bottom is a 'Parameters' section with a table header: 'Parameter', 'Type', 'Default Value', and 'Mandatory'. Below the header, it says 'No records found'. To the right of the 'Parameters' header are three icons: a trash can, a pencil, and a plus sign.

Additionally a Default Return Value can be set if the Action were to fail. For example, if the Action in the screenshot above were to time out after 5 minutes (or fail for any other reason), the `ScriptResult` would be set to the string 'Timeout'.

✳ Default Return Value is important for playbook logic. In this case you could utilize the Playbook Flow Control of Previous Action Conditions to build out playbook branches if the Action fails.

JSON Result Value

Arguably more important than the actual result value, is the JSON result that can be added via the Action. The JSON result is NOT mandatory, but it is extremely useful for pivoting on data within playbooks, or even for eyes on “manual” investigative analysis. The JSON result can be added via the `SimplifyAction` result

property's `add_result_json` method or via the `add_entity_json` method to attach a JSON result to the entity (not as useful but still covered here).

Imports and Constants

```
1 from SimplifyAction import SimplifyAction
2 from SimplifyUtils import output_handler, add_prefix_to_dict_keys, convert_dict_to_json_result_dict
3 from SimplifyDataModel import EntityTypes
4 from ArmisManager import ArmisManager
5 import json
6
7 # Example Consts:
8 INTEGRATION_NAME = "Armis"
9
10 SCRIPT_NAME = "Get Device Info via IP"
11
```

The `SimplifyAction` class from the `SimplifyAction` module will always be imported. The specific methods imported from `SimplifyUtils` are not mandatory but `output_handler` is very useful for debugging and `add_prefix_to_dict_keys` and `convert_dict_to_json_result_dict` will be utilized for data transformation. The `EntityTypes` class helps determine what type of entity the Action will run on. The `ArmisManager` was created in the Building a Custom Integration How To and it will be reused for the Armis API logic. The `json` library is also imported and a couple of constants are set.

The Action Logic

```

12 @output_handler-
13 def main():
14     ...simplify = SimplifyAction()
15     ...simplify.script_name = SCRIPT_NAME
16     ...simplify.LOGGER.info("===== Main -- Param Init =====")
17     ...
18     ...# INIT INTEGRATION CONFIGURATION:-
19     ...api_root = simplify.extract_configuration_param(provider_name='Armis',
20     ...                                              param_name="API Root")
21     ...api_token = simplify.extract_configuration_param(provider_name='Armis',
22     ...                                              param_name="API Token")
23     ...
24     ...simplify.LOGGER.info("----- Main -- Started -----")
25     ...
26     ...json_results = {}
27     ...enriched_entities = []
28     ...result_value = 'true'
29     ...
30     ...try:-
31     ...    am = ArmisManager(api_root, api_token)
32     ...    for entity in simplify.target_entities:-
33     ...        if entity.entity_type == EntityTypes.ADDRESS:-
34     ...            result = am.get_device_by_ip(entity.identifier)
35     ...            if result:-
36     ...                json_results[entity.identifier] = result
37     ...                enriched_entities.append(entity.identifier)
38     ...                simplify.result.add_entity_json(entity.identifier, json.dumps(result))
39     ...                # The result is a list that may have more than 1 JSON object... It appears the first one has better info but not sure how this works.-
40     ...                enrichment_properties = {}
41     ...                enrichment_properties['mac'] = result[0].get('macAddress', '')
42     ...                enrichment_properties['manufacturer'] = result[0].get('manufacturer', '')
43     ...                enrichment_properties['model'] = result[0].get('model', '')
44     ...                enrichment_properties['name'] = result[0].get('name', '')
45     ...                enrichment_properties['op_sys'] = result[0].get('operatingSystem', '')
46     ...                enrichment_properties['op_sys_ver'] = result[0].get('operatingSystemVersion', '')
47     ...                tags = result[0].get('tags')
48     ...                if tags:-
49     ...                    for tag in tags:-
50     ...                        enrichment_properties['tag_{}'.format(tags.index(tag))] = tag
51     ...                        enrichment_properties['riskLevel'] = result[0].get('riskLevel', '')
52     ...                        enrichment_properties['armis_type'] = result[0].get('type', '')
53     ...                        flat_report = add_prefix_to_dict_keys(enrichment_properties, 'armis')
54     ...                        entity.additional_properties.update(flat_report)
55     ...                        simplify.update_entities([entity])
56     ...                        entity.is_enriched = True
57     ...            except Exception as e:-
58     ...                simplify.LOGGER.error("General error performing action: {}".format(SCRIPT_NAME))
59     ...                simplify.LOGGER.exception(e)
60     ...                result_value = "Failed"
61     ...                output_message += e.message
62     ...            simplify.LOGGER.info("----- Main -- Finished -----")
63     ...
64     ...if enriched_entities:-
65     ...    output_message = 'The following entities were enriched: {}'.format(', '.join(enriched_entities))
66     ...else:-
67     ...    output_message = 'No assets with the IP address were found in Armis.'
68     ...simplify.LOGGER.info("result_value: {}\n output_message: {}".format(result_value, output_message))
69     ...
70     ...simplify.result.add_result_json(convert_dict_to_json_result_dict(json_results))
71     ...simplify.end(output_message, result_value)
72     ...
73     if __name__ == "__main__":-
74     ...    main()

```

This particular Action only has a main function that will be executed. In line 14 the `simplify` object is instantiated from the `SimplifyAction` class. The API Root and the API Token are imported from the Marketplace Configuration for this integration; this should look familiar from the Building a Custom Integration tutorial. In lines 26-27, an empty dictionary and an empty list are created for use later. A default result value is also set to the string 'true' in line 28.

In line 31, an object is instantiated from the `ArmisManager` class. In lines 32-33, we iterate through the entities that exist in the Alert that this Action is running on.



Remember that Actions run on Simplify Alerts! The `SimplifyAction` class allows you to retrieve all sorts of useful Case, Alert and Event data via the SDK or the Simplify API. Explore the SDK modules and the SDK documentation for more handy methods and

properties.

Notice that in line 33, every Entity has an `entity_type` property and it can be compared to the properties from the `EntityTypes` class. In this case, the entity will only be passed to the Manager's `get_device_by_ip` method if it's an IP Address (which is what the `ADDRESS` property represents). This prevents the Action from using useless cycles on Entities that will return errors from the Armis API endpoint. In line 34 notice that we are passing the entity identifier property. The identifier property is simply a string representation of the Entity, so an Entity object for the IP Address of 192.168.1.2 would have an identifier of '192.168.1.2'.

In line 36, we utilize the empty dict that we created earlier by setting the key to the entity identifier and the value to the result returned by the Armis API endpoint. In line 37, the identifier is appended to the empty list created earlier. In line 38, the `add_entity_json` method is used to create a JSON result for the entity. This method takes a mandatory two parameters: the identifier and the result in JSON format. In line 40, a new empty dict is instantiated for the properties to be used for enrichment. In lines 41-52, the result from Armis is parsed and added to this dict. In line 53, the dict is transformed by adding a string prefix of 'armis' to the keys. This is done to provide clarity in the application for what integration provided what enrichment properties within the entity.



Remember that the Entity itself is an object within the application that retains state and can be enriched by multiple integrations.

In lines 54-56, the entity is updated with the new enrichment properties. Notice in line 55 that the entity object is enriched, not the entity identifier. Also notice that it's in a list because this method expects a list of objects as input.

In lines 66-69, the output message is set based on whether there are any enriched entities in the list that was created earlier.

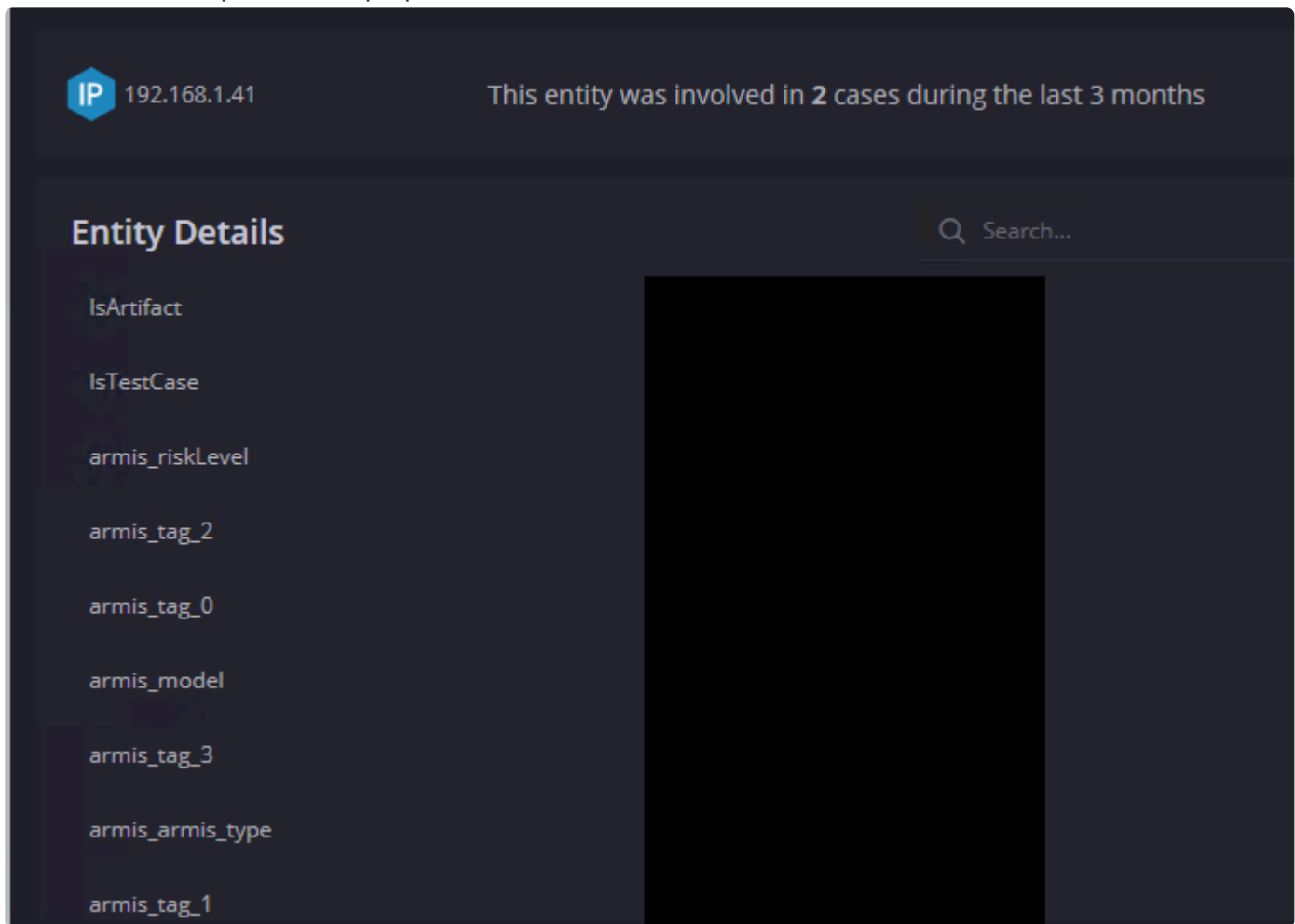
Line 72 may be the most important line of code in the Action. Here the `add_result_json` method is used to add an additional result to the Action. The `convert_dict_to_json_result_dict` method that was imported earlier is used to transform the `json_results` dictionary where the identifier is set as the key and the value is the results.

Line 73 ends the action with the output message and the result value. Lines 75 and 76 are mandatory for the Action to run.


Putting it All Together in the Application


Running this Action on an Alert with an Entity Identifier of 192.168.1.41 enriches the Entity with several new properties. Notice the 'armis' prefix on the properties that were added via this Enrichment Action. The

values that correspond to the properties are censored in this screenshot.



We can also look at the Action on the Case Wall. The Output Message and Entity JSON results are on the Results tab:


 **Armis_Get Device Info via IP**
Dec 2, 2019, 2:35:57 PM

 Peripheral Allowed: Microsoft Virtual Dvd-Rom

Results


Technical Details


Output message
The following entities were enriched: 192.168.1.41


 **192.168.1.41**


^


JSON Result


 0 {18}


 1 {18}


 2 {18}


 3 {18}


 4 {18}

 5 {18}


 6 {18}


 7 {18}

 8 {18}

 9 {18}

The Technical Details tab has the Script Result (the result value) and the JSON results. Again they are censored here.

 **Armis_Get Device Info via IP** Dec 2, 2019, 2:35:57 PM ×

 Peripheral Allowed: Microsoft Virtual Dvd-Rom

Results

Technical Details

Scope
All entities

Return Values
Script Result
ScriptResult true

JSON Result

```
▼ 0 {2}
  Entity 192.168.1.41
  ▼ EntityResult [10]
    ▼ 0 {18}
      category
      firstSeen
      id
      ipAddress 192.168.1.41
      lastSeen 2019-11-25T06:45:30.563284
      macAddress
      manufacturer
      model
      name
```

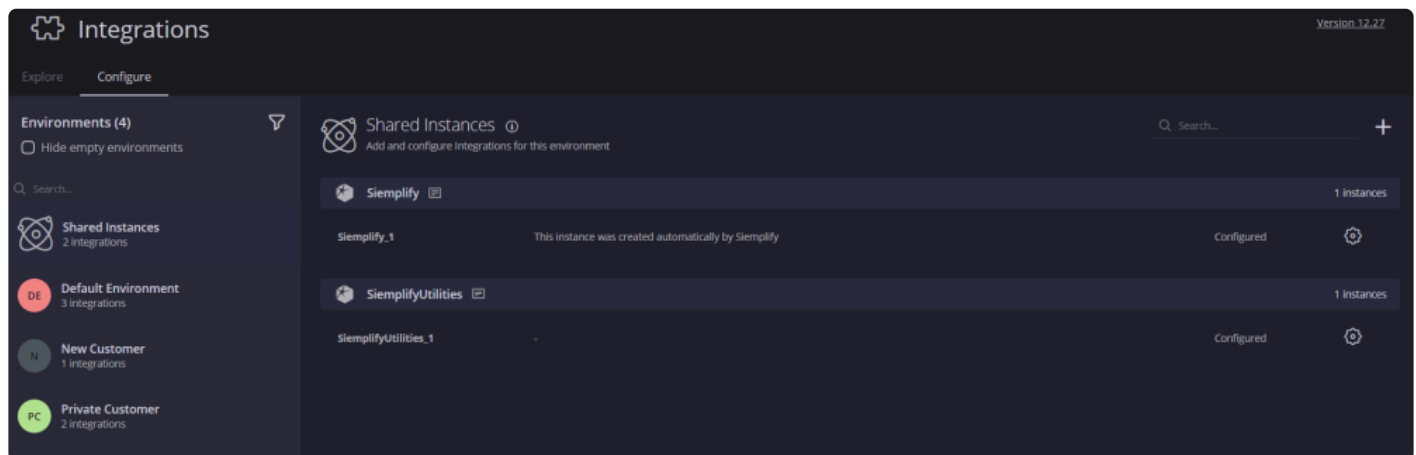

Supporting Multiple Instances

Introduction

Users can configure multiple instances of the same integration for the same environment. This feature provides users with greater flexibility and granularity when creating and running Playbooks. For example, when building a Playbook which caters to a customer with two sites, each site using its own Active Directory, you can now configure two instances of the same integration for the same environment and choose between them within the Playbook step.

This feature is configured in the Marketplace > Integrations > Configure Screen and supported by the Choose Instance field in the Playbook step, as well as the new multi-select environment option.

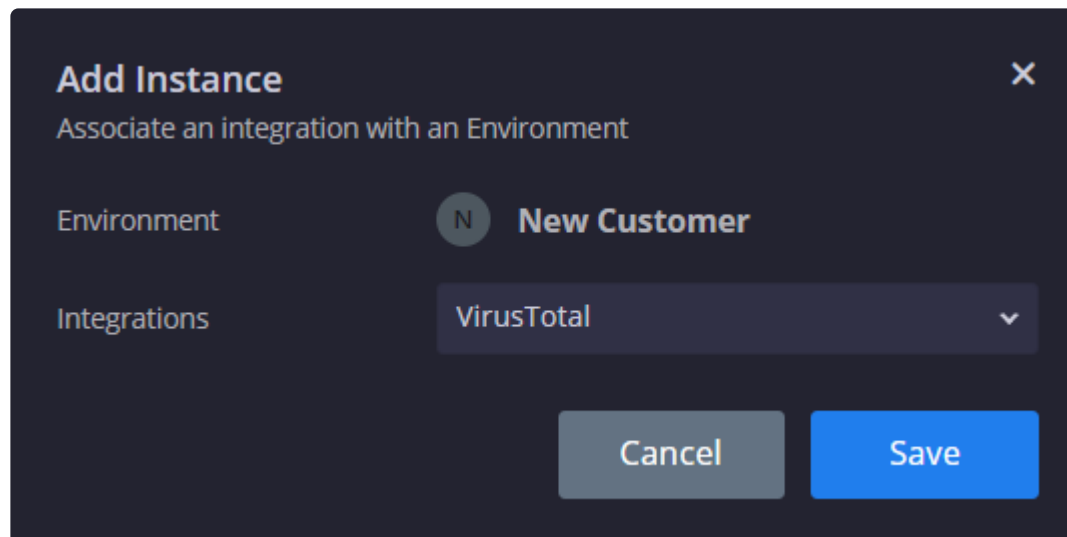
Configure Screen. Let's take a look at the Configure Screen. This screen comes with two predefined options on the left. One is called Shared instances and the other is the Default Environment (referred to as No Environment in releases prior to 5.5.0). In the screenshot below, we have defined two extra environments.



Shared Instances acts as a type of library for configured integrations that can be used for all environments that are created both now and in the future. The Shared instances repository also contains Simplify predefined Integrations out of the box.

Any environment that you create in the Settings > Environments tab will appear in the list on the left. You can choose to filter the display of environments and hide empty environments. Enterprise customers will primarily be working with the default environment.

Configure Instance: You add an instance by selecting an environment on the left side of the page and then on the plus sign on the top right. Select the Integration and then configure the parameters for the specific instance of that Integration. You must configure an instance of an Integration in order to use it in a Playbook. To reconfigure or edit this instance in the future, you can click on the Gear icon. To add two instances of the same Integration per environment, simply configure a second instance.

A dark-themed dialog box titled "Add Instance" with a close button (X) in the top right corner. Below the title is the subtitle "Associate an integration with an Environment". There are two main sections: "Environment" and "Integrations". The "Environment" section has a circular icon with the letter "N" and the text "New Customer". The "Integrations" section has a dropdown menu with "VirusTotal" selected and a downward arrow. At the bottom right are two buttons: "Cancel" (grey) and "Save" (blue).

Add Instance ×

Associate an integration with an Environment

Environment N **New Customer**

Integrations VirusTotal ▼

Cancel Save

Select Environment. Now, let's navigate to the Playbooks screen and take a look at the Multi-select environment option that appears when you create a new Playbook. You have two choices: one is to select All environments. This means that this Playbook will run on all current environments defined in the system as well as all environments that will be added in the future.

The second option is to select one or more environments for the Playbook to run on.

Note that selecting multiple or all environments will affect the type of Instance you can configure for the Playbook steps. Let's delve deeper into this.

Configure Instance. Now we will navigate to a Playbook step that contains an Integration. What will appear in the Configure Instance field depends both on what Instances you created and also on what environments you chose when creating the Playbook.

If you chose All Environments or Several environments: the first option in configure instance is “Dynamic Mode”.

Dynamic Mode: Dynamic mode means that when the playbook is attached to a case, Simplify will try to access the instance of the integration configured for the case environment

Fallback Instance: This is optional. Define a Fallback instance from the Shared Instances to be used if there is no environment associated with this Integration. If there is nothing defined in the Shared Instances then you will not be able to choose anything here.

Note: If you choose dynamic mode, and there are two instances defined in the associated environment, then the Playbook will stop and wait for the analyst to provide manual input.

If you choose a single environment, then the Configure Instance will allow you to choose the Integration that you have configured for that specific Action, or the Shared Instance integration.

Let's look at a few examples of this feature.

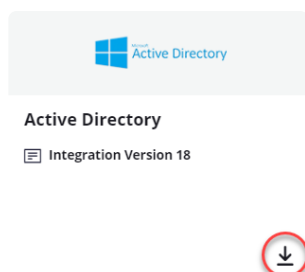
Use Case #1 Two Instances in a Default Environment

In this scenario, I have one enterprise network separated to two sites – US and UK. For each of the sites I want to have a separate Active Directory configuration.

Therefore, I need to configure two instances of ActiveDirectory integration for the same environment and then have the Playbook select the required one at runtime.

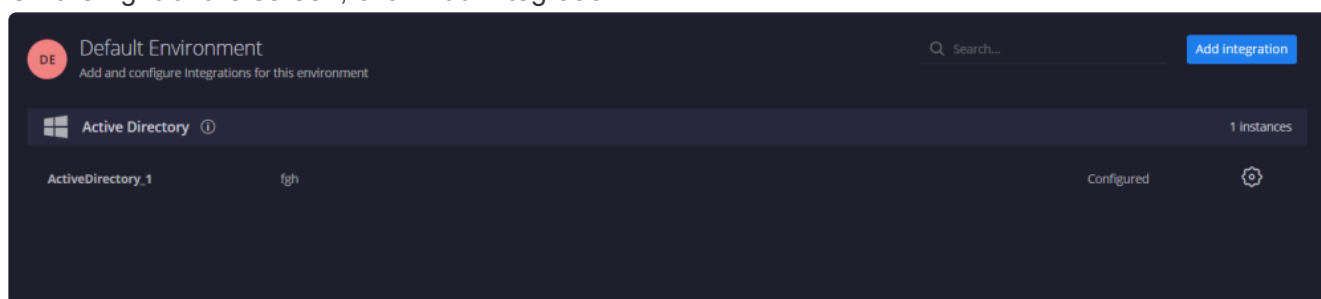
Install an Integration

1. Navigate to Marketplace > Integrations.
2. Search for the required Integration. For this example, we will be using Active Directory.
3. Install it.

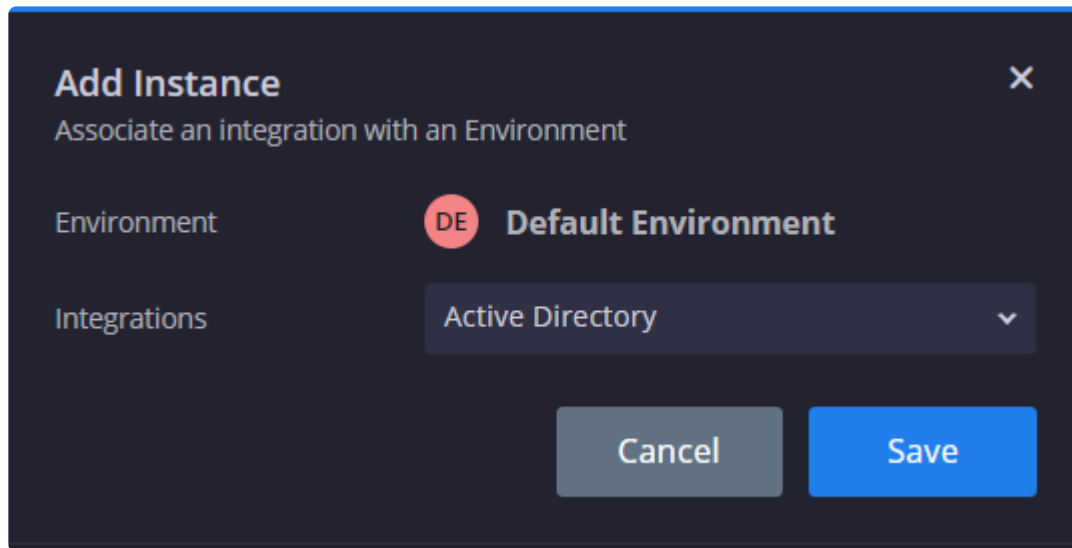


Configure an Instance

1. In the Marketplace > Integrations, click on the Configure tab.
2. In the Environments list on the left, click on the Environment you want to create an Instance for. For this example, we will use the Default environment.
3. On the right of the screen, click Add Integration.



4. Select the required Integration and click Save. In this example, we have selected Active Directory.

A dark-themed modal dialog box titled "Add Instance" with a close button (X) in the top right corner. Below the title is the subtitle "Associate an integration with an Environment". The dialog contains two rows of configuration options. The first row is labeled "Environment" and shows a red circular icon with "DE" next to the text "Default Environment". The second row is labeled "Integrations" and shows a dropdown menu with "Active Directory" selected and a downward arrow. At the bottom right of the dialog are two buttons: a grey "Cancel" button and a blue "Save" button.

Add Instance ×

Associate an integration with an Environment

Environment **DE** **Default Environment**

Integrations **Active Directory** ▼

Cancel **Save**

5. In the Configuration screen that displays, add in all the relevant information and parameters. We will configure it for users in the US site. When finished, click Save. You can also click Test to make sure that the configuration works.

Active Directory - Configure Instance

×

Configure all the necessary fields and parameters for this instance

Environment

DE **Default Environment**


Instance Name

ActiveDirectory_US Site

Description

Configured for US employee site

Parameters

 For more information on configuration and integration details, [click here](#)

Server

x.x.x.x

Username

user@domain.com

Domain

domain.com

Password

Custom Fields

customField1,customField2

Test

Cancel

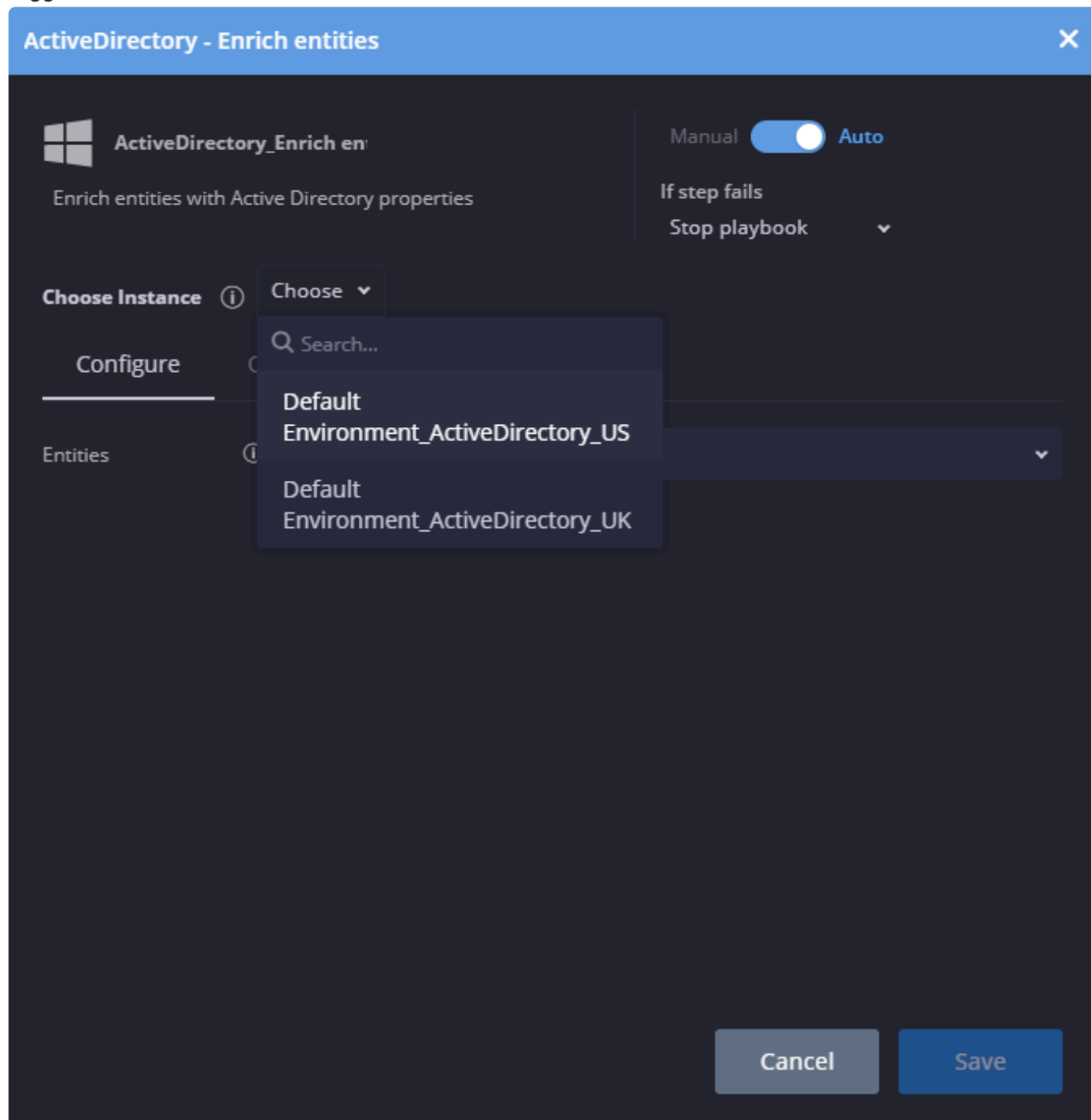
Save

- Now, let's add another instance of the Active Directory. And this time we will configure it for users in the UK site. Click Save when fully configured.
- Note that you can make changes at a later stage if needed. Once configured, the Instances can be used in Playbooks.

Use this Instance in Playbooks

- Navigate to Playbooks screen and click the plus icon to add a Playbook.

2. Make sure to select the relevant folder and for this example, to choose the Default Environment. We will talk in more detail about which Environment to choose later on in this How To guide.
3. In the Actions, under Active_Directory, let's choose Enrich Entities and drag it into a Step and then click on it.
4. In the Choose Instance field, select the Instance – either UK site or US site that this Playbook will be triggered for.



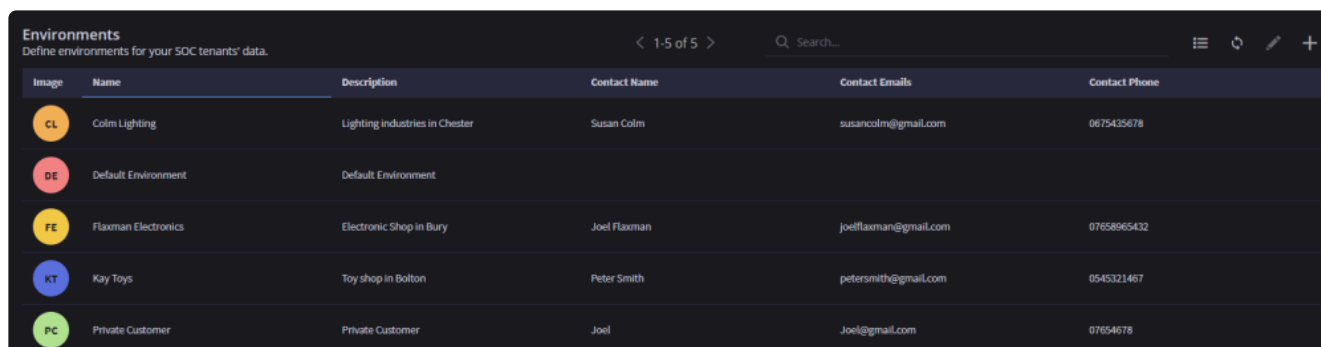
Use Case # 2 Dynamic Mode in Multi Environments

In this scenario, as an MSSP, I have several different customers with each one defined in a different environment. At runtime of the Playbook, I want the Playbook to choose the environment “dynamically”

based on which environment the case has come in from.

Define environments:

1. Navigate to Settings > Organization > Environments screen.
2. Click on the plus sign and define the required environment with the parameters.
3. Create several new environments.



Environments
Define environments for your SOC tenants' data.

< 1-5 of 5 > Search...

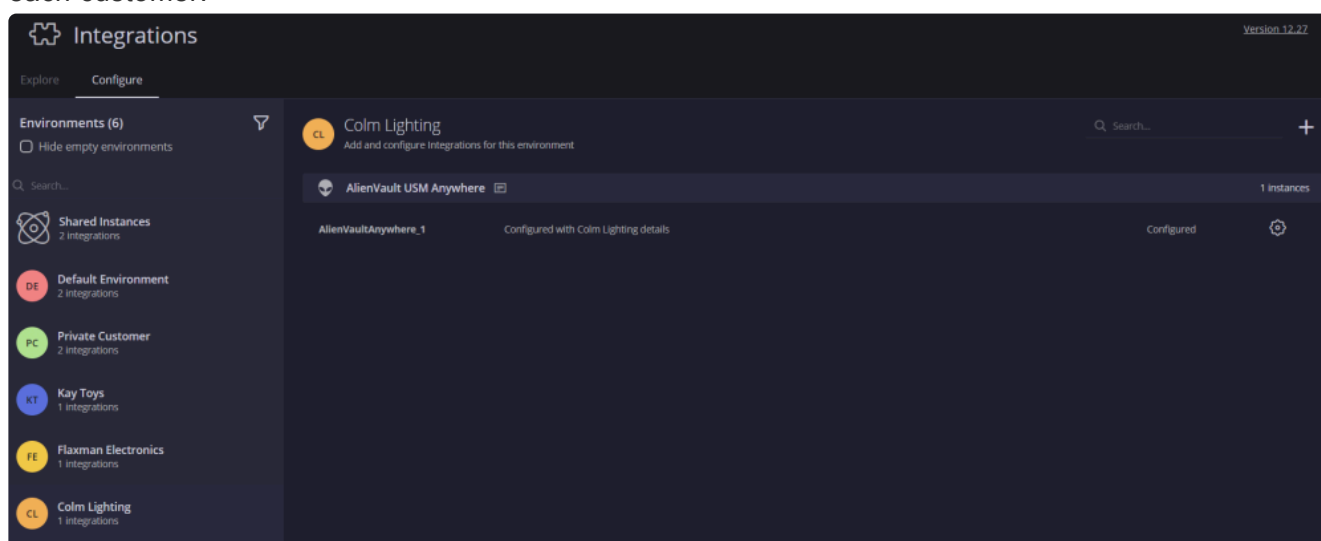
Image	Name	Description	Contact Name	Contact Emails	Contact Phone
CL	Colm Lighting	Lighting industries in Chester	Susan Colm	susancolm@gmail.com	0675435678
DE	Default Environment	Default Environment			
FE	Flaxman Electronics	Electronic Shop in Bury	Joel Flaxman	joelflaxman@gmail.com	07658965432
KT	Kay Toys	Toy shop in Bolton	Peter Smith	petersmith@gmail.com	0545321467
PC	Private Customer	Private Customer	Joel	Joel@gmail.com	07654678

Install an Integration

1. Navigate to Marketplace > Integrations.
2. Search for the required Integration. For this example, we will be using Alien Vault.
3. Install it.

Configure Instances

1. In the Marketplace > Integrations, select each customer and click on the Configure tab.
2. Configure each environment with the the Alien Vault integration instance according to the needs of each customer.



Integrations Version 12.27

Explore Configure

Environments (6) Hide empty environments

Search...

- Shared Instances 2 Integrations
- Default Environment 2 Integrations
- Private Customer 2 Integrations
- Kay Toys 1 Integrations
- Flaxman Electronics 1 Integrations
- Colm Lighting 1 Integrations

Colm Lighting Add and configure Integrations for this environment

Search...

AlienVault USM Anywhere 1 Instances

AlienVaultAnywhere_1 Configured with Colm Lighting details Configured ⚙

Set up Playbooks

1. Navigate to Playbooks tab.
2. Create a Playbook making sure to select the environments you created and configured above.

Create New ✕

Type ☒ Playbook ☐ Block

Folder Default ▼


Environment ☐ All Environments ☒ Select

Colm Lighting, Kay Toys, Flaxman Electronics ▼

Cancel Create

3. When using the Alien Vault ping action, select Dynamic Mode. This ensures that Simplify will check which environment the case comes from at run time and apply that specific instance to it.

AlienVaultAnywhere - Ping

 AlienVaultAnywhere_Ping_

Test connectivity

Choose Instance ⓘ

Dynamic mode ▼

Fallback Instance ⓘ

None ▼

Manual ☒ Auto

If step fails

Stop playbook ▼

Configure

Output

Entities ⓘ

All entities ▼

Cancel

Save

Playbooks

Understanding Playbook Metrics

The Playbook Metrics feature allows our customers to gain important insights into their Playbooks. Security engineers can use this vital information to tweak the Playbooks in order to get maximum optimal performance. Summary is to minimize the time that an analyst needs to get decisions when handling a case. Dashboards are to get general information about the quality of your playbooks.

The following places in the Simplify Platform can provide you with greater visibility into the Playbooks execution:

- **Playbook Monitoring:** The Monitoring feature allows customers to use automation to its full capacity. This interface is displayed for each individual Playbook.
- **Playbook Summary:** The Summary feature is to minimize the time that an analyst needs to get decisions when handling a case. This interface is displayed for each running Playbook on the Cases screen.

Playbook Monitoring

The Monitoring screen contains the following information:

- **Runs:** How many times the Playbook/Block ran during the defined time period. Thousands will be represented by a K. Millions will be represented by an M.
- **Redundant:** Number of times the Playbook/Block didn't run in the predefined time period (because it exceeded the maximum number of playbooks (3) that can be automatically added to an alert). If the number is larger than 1 – this could be a good indication to tweak the Playbook – maybe by using Blocks or other logical steps
- **Closed Alerts:** Percentage of alerts that were closed by this Playbook.
- **Average Run Time:** Average amount of time that this Playbook took to run. This statistic can prove useful in identifying identify weak points in playbooks – manual actions, frequently-errored steps etc.
- **Playbook Status Pie Chart:** Shows three options. Options are finished successfully, failed, or waiting for user action. This chart shows you playbook statuses according to the defined time period and is cumulative. Each option is clickable and will take you to a Search results page displaying the cases that this playbook with the specific status was attached to.
- **Playbook Trends Line Chart:** Shows completed runs, failed runs and a total of runs (both failed and successful). Hover your mouse over each dot on the line to see a pop-up showing more information. This chart can come in useful if a new playbook that you recently created is running as you've expected, or if an existing playbook that you recently improved was actually improved as you've expected or if more enhancements are needed in order to meet your expectations. For example, let's say you see that the Playbook didn't run twenty times over the last month, you might then tweak the trigger logic to make the Playbook more selective. You could then look at the Trends chart to check that the Playbook ran successfully from that time onwards.
- **Environments Bar Chart:** Displays all the environments that this Playbook ran in. Each section is

clickable and will take you to a Search results page.



In addition, hovering over the Actions will display popup showing success/failure rates of that Action during a Playbook run, and hovering over a Conditions branch will display a popup showing how many times that branch was selected.

Playbook Summary

When clicking on a Playbook, the Context Details appears as a Playbook Summary. This shows the following information:

- Playbook Name and Status
- Waiting for User Input: If the Playbook is waiting for the security engineer to do something, this will be displayed prominently at the top of the Playbook Summary. In addition, a Push notification will be sent to the relevant user letting them know that the Playbook is waiting for them.
- Time and Length of Playbook Run
- Integrations: list of Integrations being used by this Playbook. When clicking on an integration, the specific step will be marked in the playbook viewer so that the analyst can easily find the step that they want to focus on.
- Playbook Flow: each step that was run with its status and step result.
- Errors: any errors will be listed here. If an error caused the playbook to stop it will be highlighted at the top of the summary, but if it was skipped, it will be at the bottom. Each error is clickable and will direct you to the Kibana logs page. You can also choose to rerun the Action or Playbook from here.



Playbook - playbook

2020-03-15 10:03:17

🕒 Run Length: 1 min



Integrations



Simplify



Flow



ActiveDirectory



Playbook Flow



Simplify_Case Comment

Simplify_Case Comment_1



ActiveDirectory_Enable account

ActiveDirectory_Enable account_2



MultiChoiceQuestion

MultiChoiceQuestion_1



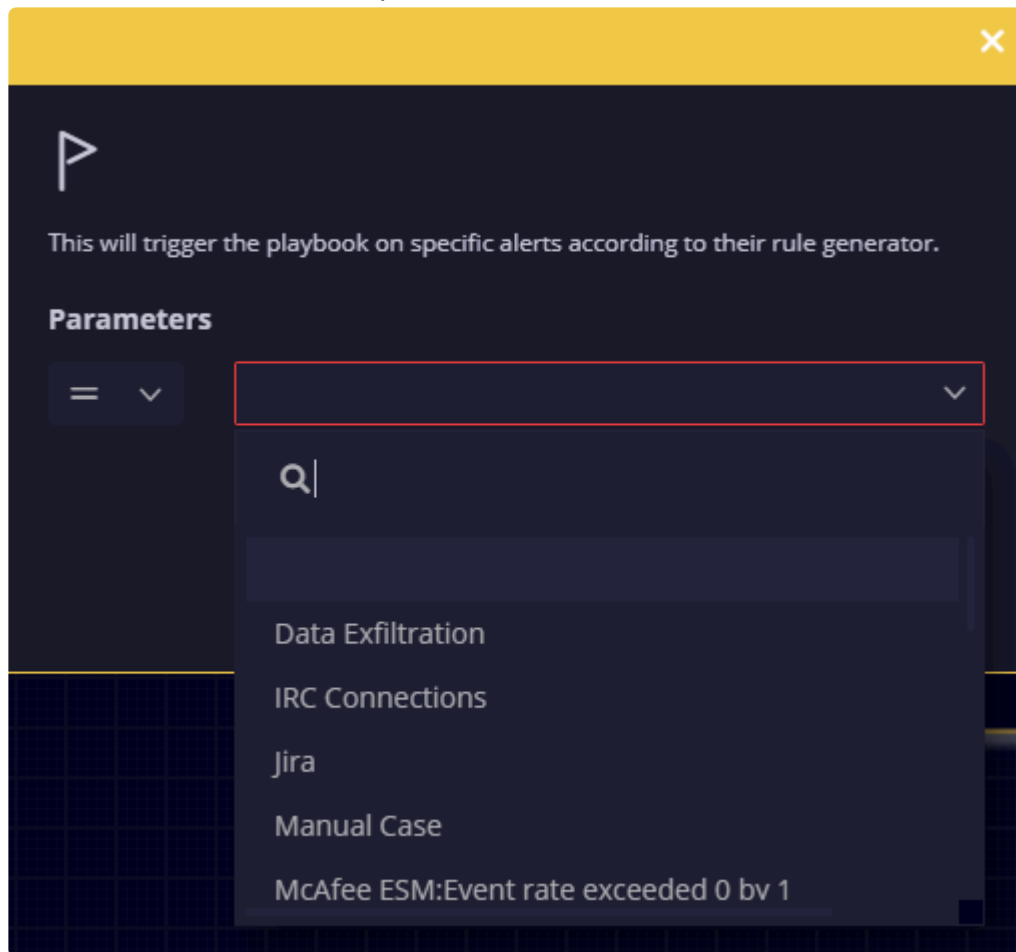
NestedAction

Inputs

Using Alert Type Trigger in a Playbook

The Alert Type trigger is used to trigger a playbook based on “Rule Generator Name”.

1. Navigate to the Playbooks screen and create a new Playbook.
2. Drag the Alert Type trigger to a Playbook step and click on it to open it. The parameters field requires the “Rule Generator Name” of each alert which will trigger this playbook. The Rule Generator name can be selected from the drop-down menu.



If you don't remember the Rule Generator Name of the Alert you want to trigger the Playbook for, you can navigate to Cases > Case Overview. Click on the Alert. Expand the Case window in the Context Details.

The screenshot displays the Simplify interface with the following components:

- Overview Tab:** Contains sections for Alerts, Insights, and Playbooks.
- Alerts (1):** Shows a card for 'TEST_ALARM_2' with 'Time not mapped' and '0 Events'.
- Insights (0):** Displays a timeline from April 2019 to September 2019 with the message 'No Insights'.
- Playbooks (1):** Shows a 'Test If' playbook with a flow diagram consisting of a yellow trigger box, a purple decision box with a green checkmark, and a blue action box.
- Context Details Panel:** Located on the right, it shows details for 'Alert Test Alarm 2'. It includes a search bar, a threat ID, and a table of fields and values.

FIELD NAME	VALUE
Priority	High
TIME	
CASE	
Ticket ID	101153
Alert ID	101153
Alert Name	TEST_ALARM_2
Rule Generator	McAfee ESM: Event rate exceeded 0 by 4

3. Either copy and paste this rule generator name or select it from the drop-down list.
4. Continue building the Playbook. Now, any alerts generated by this rule generator will have this playbook triggered.

Using the Expression Builder

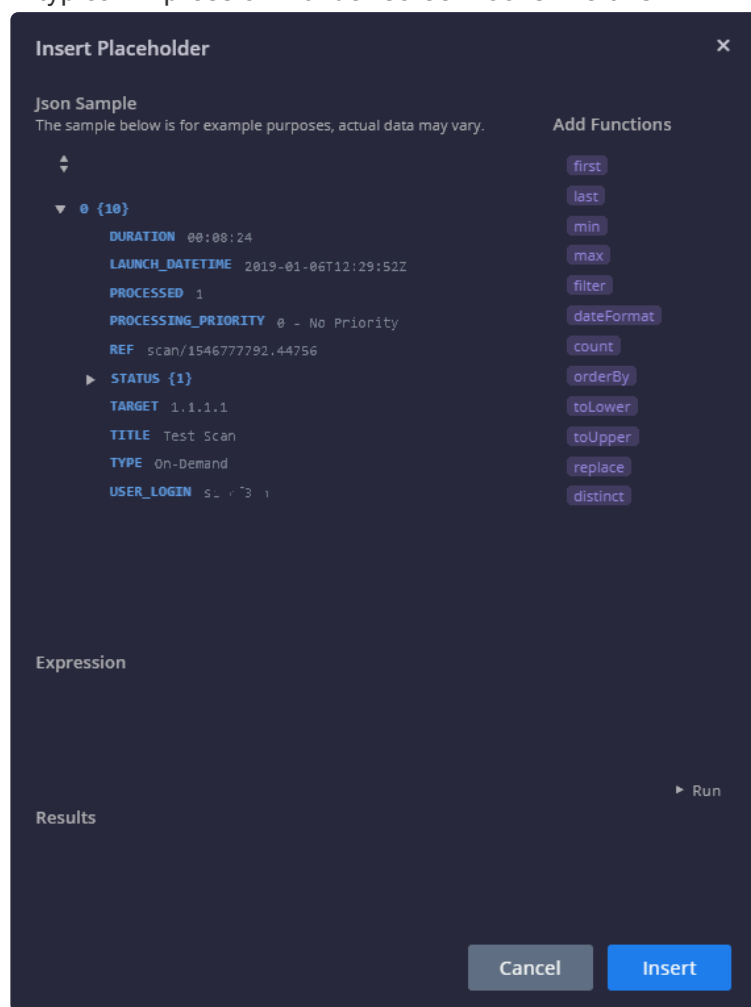
What is an Expression Builder?

In this article we are going to look at using the Expression Builder (introduced in Software Version 5.0) in Playbooks.

The Expression Builder allows the parsing and manipulating of JSON results and further utilizing them in subsequent actions in a simple and intuitive manner. The Expression Builder generates a variety of dynamic transformation functions that can be chained together and previewed and tested, thereby allowing for an interactive experience for the transformation and parsing of raw action results.

What does the Expression Builder screen look like?

A typical Expression Builder screen looks like this:



It contains the following information:

JSON Sample:

This is an example of potential data and is not based on real time results. The actual data may be different

and may contain more or less fields from the example. If the analyst knows of extra fields that will be returned in run time then they can type the relevant key path in the syntax textbox.

Functions

The following functions are supported:

- First (x) – Return first X elements of an array
- Last (x) – Return last X elements of an array
- Min (KeyPath) – return an element of an array by the minimum
- Max (KeyPath) – return an element of an array by the maximum
- Filter (ConditionKey, Operator, Value) – Filter objects by field
- DateFormat (“pattern”) – format date by given pattern (‘yyyy/dd/mm HH:mm:ss’) to supported format (“YYYY-MM-DDThh:mm:ssZ”)
- Count () – return the number of elements in expression
- OrderBy (KeyPath, “Direction”) – order array by given child field
- toLower () – convert expression to lower case letters
- toUpper () – convert expression to upper case letters
- Replace (“x”, “y”) – replace string in an expression
- Distinct () – remove duplicates from an array

Expression:

The Expression field is where you insert the JSON results together with the functions and pipes to add several functions together and build the expression. We will explore examples of building expressions later on in this article.

>Run / Results:

After filling in the Expression Builder, clicking Run will display the Results based on the JSON Sample Data shown above in the Expression Builder

Using the Expression Builder in Playbook Actions

Let’s look at three use cases building an Expression in an Action.

Use Case Number One: IPS

Let’s say we are building a Playbook which has found a malicious flow in a Network.

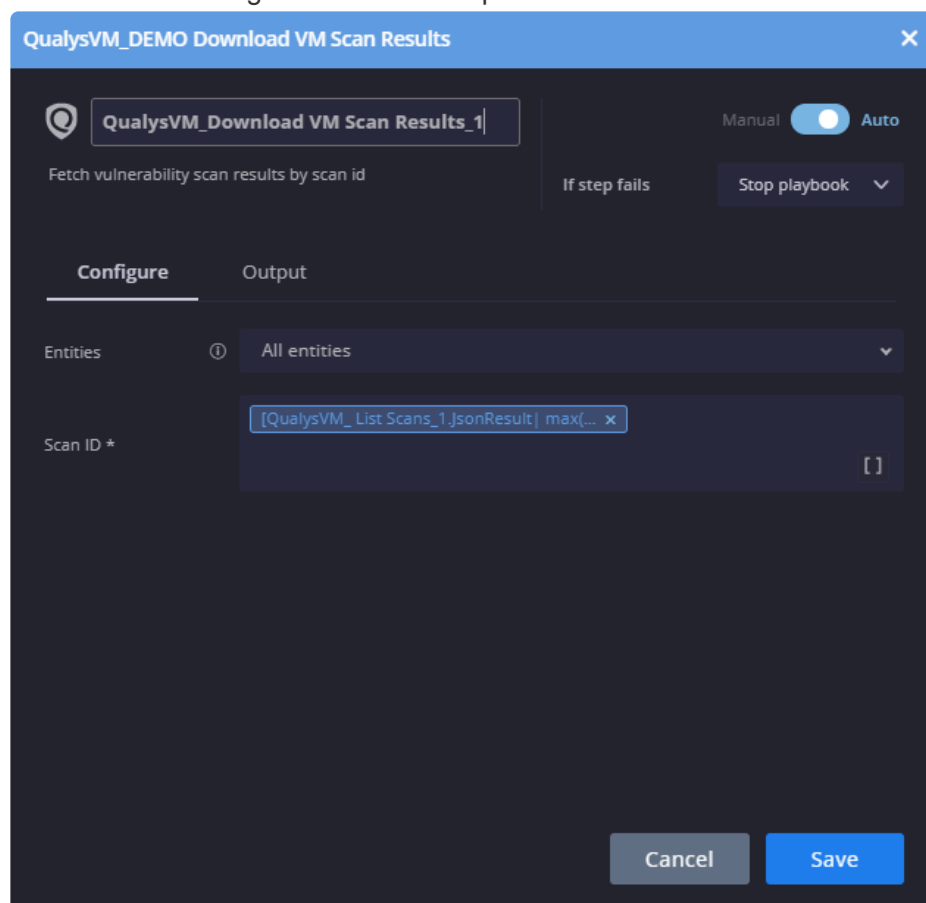
Imagine that a vulnerable management tool such as Qualys has scheduled scanning every day. In this example, we are using Qualys – List Scans to get all the latest scans from Qualys (30 days hard coded) We will be using the expression builder to extract the ID (REF) of the newest scan as placeholder for download VM scan results. VM scan results will download the relevant report.

Using the List Operations, we are going to extract the list of the vulnerabilities’ identifiers which was found on the network (CVE) from the report and compare it to the CVE from the case

We can use an IPS alert to trigger the Playbook. Start off with an Active Directory_Enrich Entities action so

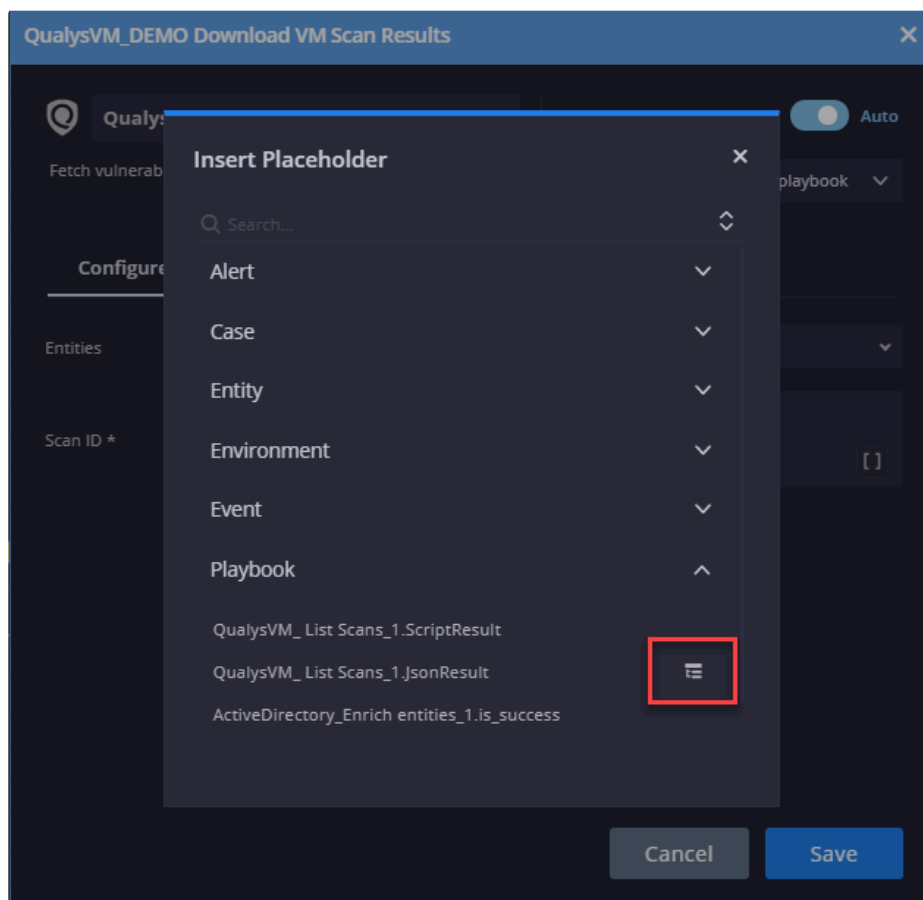
that we can enrich all the entities that are potentially affected. and then use Qualys VM – List Scans to retrieve the latest scan results for the network machines and determine if any of them are vulnerable to the detected flow.

Now let's take a look at the next action QualysVM_Download VM Scan Results_1. This screenshot shows the Placeholder together with the Expression Builder that has been added.



To add this placeholder:

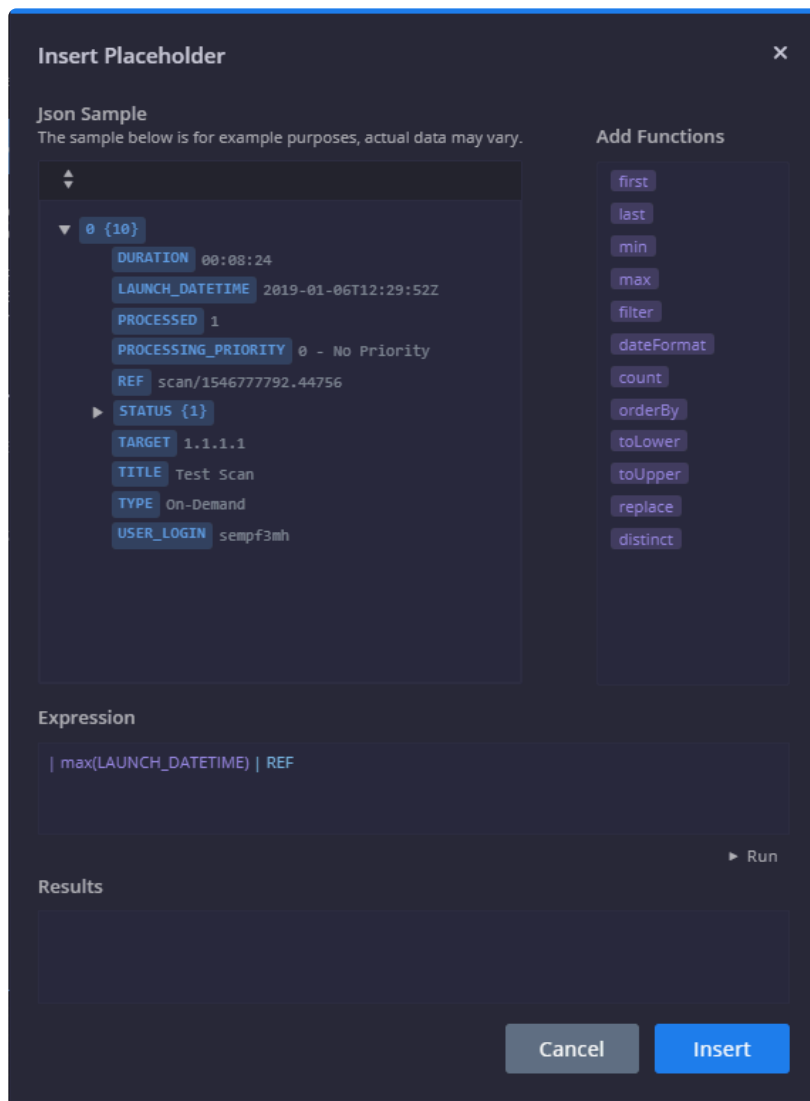
1. Click the Placeholder icon [].
2. Select Playbook > QualysVM_list_Scans_1_JSONResult.
3. Click on the Expression Builder icon as shown below.



The Expression Builder screen opens up.


4. Add the following in the Expression field. The expressions means that we use MAX to take the latest result by date (LAUNCH_DATETIME) and then extract the specific scan id of the relevant scan where REF means scan id.

bc. | max(LAUNCH_DATETIME) | REF



5. Click Run. The expected results will appear.
6. Click Insert to include the Expression Builder as part of the Placeholder.
7. Next action should be as follows: Action > List operations using CVEs from the cases + expression builder displays – see following screenshots.

SiemplifyUtilities_List Operations

 **SiemplifyUtilities_List Operations_1**

Manual ☒ Auto

Provide operations on lists.

If step fails

Stop playbook ▾

Configure

Output

Entities

ⓘ

All entities ▾

First List

ⓘ

[QualysVM_Download Vm Scan Results... x]

[]

Second List

ⓘ

[Alert.CVE] x

[]

Delimiter

,

[]

Operator

ⓘ

intersection

Cancel

Save

Simplify Private and Confidential

Page 59 of 218

Insert Placeholder

Json Sample

The sample below is for example purposes, actual data may vary.

```
{
  "0": {
    "severity": "High"
  },
  "1": {
    "severity": "Medium"
  },
  "2": {
    "associated_malware": "Trojan",
    "bugtraq_id": "12345",
    "category": "Web server",
    "cve_id": "CVE-2020-1234",
    "dns": "1dot1dot1dot1.cloudflare-dns.com",
    "exploitability": "Easy",
    "fqdn": "1.1.1.1",
    "impact": "N/A",
    "instance": "1",
    "ip": "1.1.1.1",
    "ip_status": "host scanned, found vuln",
    "netbios": "1"
  }
}
```

Add Functions

- first
- last
- min
- max
- filter
- dateFormat
- count
- orderBy
- toLower
- toUpper
- replace
- distinct

Expression

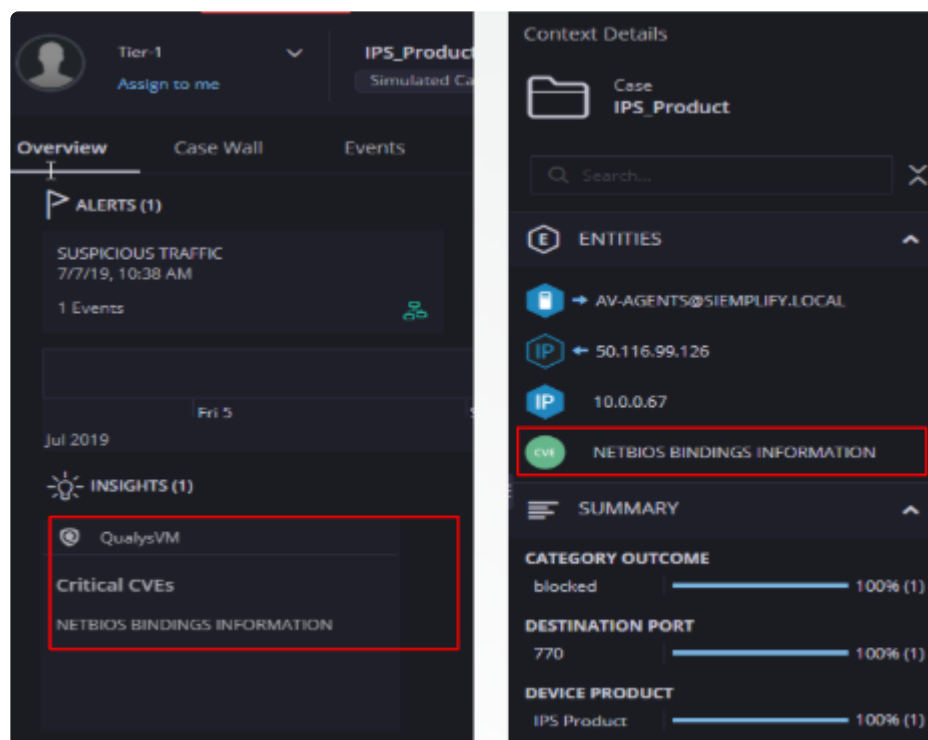
```
| filter(severity,">","2") | title | toUpper()
```

Results

Cancel

Insert

- Once the Playbook is triggered in real time, you can see the scan results in the Context Details pane, including the specific scan as pdf.



Use Case Number Two: Too Many Failed Login Attempts

For this use case let's say that we had failed login attempts and we want to figure out which department the user belongs to and when was the last time he changed his password in order to determine the severity of the alert. In this Playbook we are going to use Active Directory to get more information.

In the first action, we will use `ActiveDirectory_Enrich` entities to find out more information on all the internal entities. In this Insight message, we want to find out the user and the last time they logged in. Below is a screenshot of this action already with the necessary Placeholders with the Expression Builders in.

The screenshot shows the 'Siemplify_Add General Insight' configuration window. At the top, there's a title bar with a close button. Below it, a header section contains a star icon, a text input field with 'Siemplify_Add General Insight_1', a 'Manual' toggle switch (currently set to 'Auto'), and a 'Stop playbook' button with a dropdown arrow. A description 'Add a general insight configurable message to the case' is on the left, and 'If step fails' is on the right. The main area is divided into 'Configure' and 'Output' tabs. Under 'Configure', there are four fields: 'Entities' (set to 'All entities'), 'Title *' (set to 'Involved entities'), 'Message *' (containing HTML tags and two placeholders for 'ActiveDirectory_Enrich entities_1.JsonRes...'), and 'Triggered By' (set to 'ActiveDirectory'). Each field has an information icon and a placeholder icon. At the bottom right are 'Cancel' and 'Save' buttons.

Configure	Output
Entities	① All entities
Title *	① Involved entities
Message *	① <code><u>Usernames:</u></code> [ActiveDirectory_Enrich entities_1.JsonRes... x] <code>

</code> ① <code><u>Last Logons</u></code> [ActiveDirectory_Enrich entities_1.JsonRes... x]
Triggered By	ActiveDirectory

To add these placeholders:

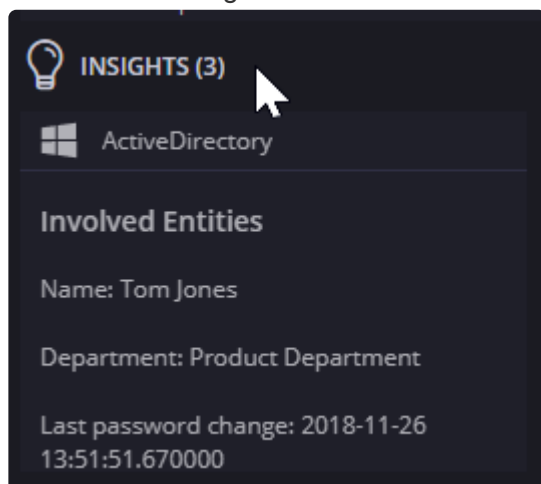
1. In the Message field, click the Placeholder icon [].
2. In the Insert Placeholder screen, click the Expression Builder icon next to the ActiveDirectory_Enrich entities_JSONResult
3. Add the following in the expression field: This will choose the entity identifier. Currently, if more than one entity returned results – we will get it as comma separated list.
bc. | Entity



4. Click Run and you will see the sample result. In this case, user@domain.com.
5. Click Insert to use this as part of your placeholder message. Add the relevant free text to your message as well.
6. Once again, click the Placeholder icon [] and then click the Expression Builder icon next to the ActiveDirectory_Enrich entities_JSONResult.
7. Add the following expression. This will capture the last logon time of the specified user. | EntityResult.lastLogon



8. Click Insert and then click Save.
9. Once the Playbook is triggered in real time, you will see a message on the Insight pane with the user name and last login time.



Use Case Number Three: Virus Total

The action checks the reputation of the file hash on VirusTotal. In this example, we are getting a report for a specific file hash. We are then extracting the reputation (i.e. is it known to be malicious) by a specific scan

engine. In this case, Kaspersky.

So we are going to check if Kaspersky marked the file hash as malicious and create an entity for that.

In the first action, we will use VirusTotal_Scan Hash.

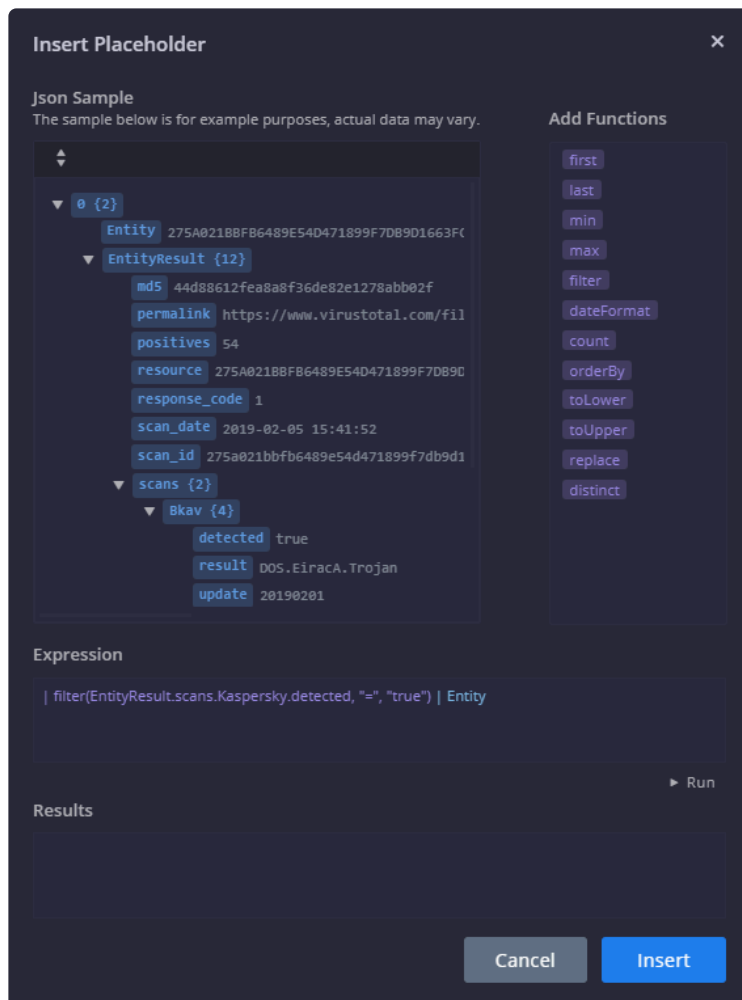
Now, let's take a look at the next action. Siemplify_Create Or Update Entity Properties. This creates or changes properties for an entity. Detected by Kaspersky.

Below is a screenshot of this action already with the necessary Placeholders with the Expression Builders in.

The screenshot shows the 'Siemplify_Create Or Update Entity Properties' configuration window. The window has a blue header bar with the title and a close button. Below the header, there is a section with a star icon and the title 'Siemplify_Create Or Update Entity Property'. To the right of this section, there is a toggle switch for 'Manual' (off) and 'Auto' (on). Below the title, there is a description: 'Create\Change properties for entities in an entity scope.' To the right of this, there is a section for 'If step fails' with a dropdown menu set to 'Stop playbook'. The main configuration area is divided into two tabs: 'Configure' (selected) and 'Output'. Under the 'Configure' tab, there are three rows of configuration fields: 1. 'Entities' with a value of 'All file hashes' and a dropdown arrow. 2. 'Entity Field *' with a value of 'DetectedByKaspersky' and a dropdown arrow. 3. 'Field Value *' with a value of '[VirusTotal_Scan Hash_1.JsonResult] filter...' and a dropdown arrow. At the bottom of the window, there are 'Cancel' and 'Save' buttons.

To add this placeholder:

1. In the Field Value field, click the Placeholder icon [].
2. In the Insert Placeholder screen, click the Expression Builder icon next to the VirusTotal_ScanHash_JSONResult.



3. Add the following expression: | filter(EntityResult.scans.Kaspersky.detected, "=", "true") | Entity
If we scanned more than one hash, it filters the results by all the entity objects that Kaspersky marked as malicious – and then returns just the entity name.

Insert Placeholder

Json Sample

The sample below is for example purposes, actual data may vary.

```
{
  "0": {
    "Entity": "275A021B8FB6489E54D471899F7DB9D1663FC",
    "EntityResult": {
      "md5": "44d88612fea8a8f36de82e1278abb02f",
      "permalink": "https://www.virustotal.com/fil",
      "positives": 54,
      "resource": "275A021B8FB6489E54D471899F7DB9D",
      "response_code": 1,
      "scan_date": "2019-02-05 15:41:52",
      "scan_id": "275a021bbfb6489e54d471899f7db9d1",
      "scans": {
        "Bkav": {
          "detected": true,
          "result": "DOS.EiracA.Trojan",
          "update": "20190201"
        }
      }
    }
  }
}
```

Add Functions

- first
- last
- min
- max
- filter
- dateFormat
- count
- orderBy
- toLower
- toUpper
- replace
- distinct

Expression

```
| filter(EntityResult.scans.Kaspersky.detected, "=", "true") | Entity
```

► Run

Results

Cancel Insert

- Click Insert and then click Save.
- Results will display at run time as follows.

ALERTS (1)

SUSPICIOUS TRAFFIC
7/7/19, 11:03 AM
1 Events

INSIGHTS (3)

VirusTotal

69630e4574ec6798239b091cda...

VirusTotal - Hash was marked as malicious by 52 of 57 engines. Threshold set to - 3

VirusTotal

275a021bbfb6489e54d471899f7...

VirusTotal - Hash was marked as malicious by 63 of 64 engines. Threshold set to - 3

VirusTotal

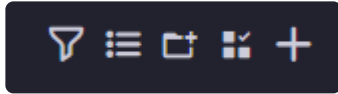
Malicious by KASPERSKY

69630e4574ec6798239b091cda43dca0.2
75a021bbfb6489e54d471899f7db9d1663
fc095ec2fe2a2c4538aab051fd0f





Bulk Actions and Filters in Playbooks

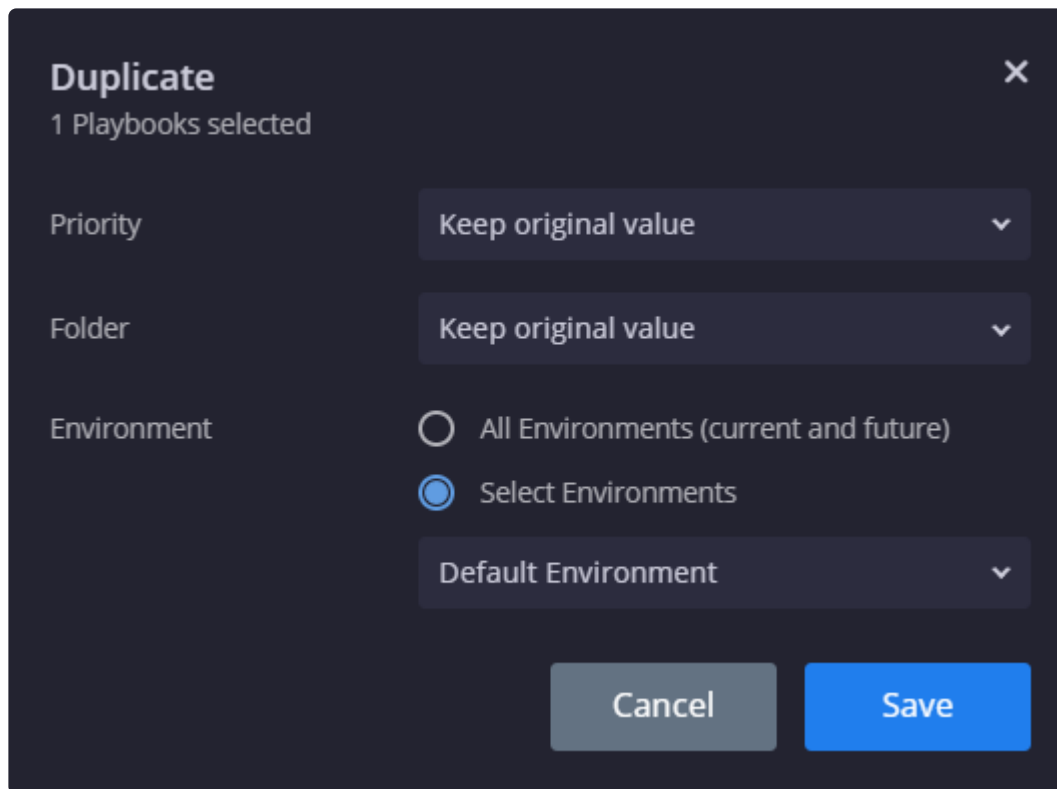
In Software Release 5.5.0, bulk actions were added, thereby allowing you to perform various actions on multiple Playbooks and Playbook Blocks at a time.

The following actions are available at the top left of the Playbooks screen.



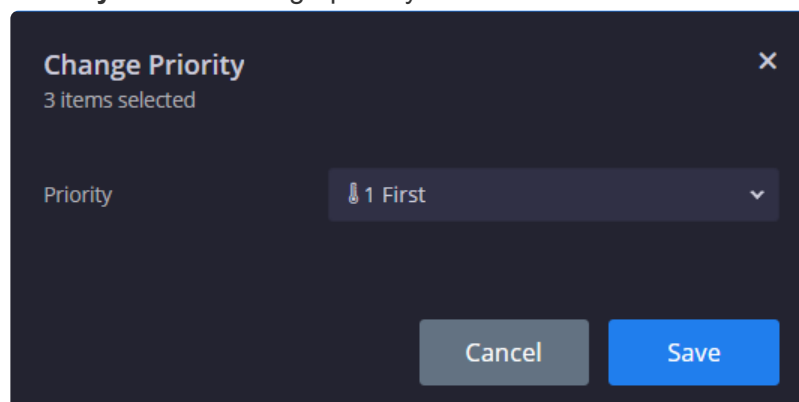
From right to left:

- **Plus**  : Add a new Playbook or Block. Here you can choose which folder and which environment the Playbook/Block belongs to.
- **Edit**  : Allows you to select single or multiple Playbooks and Blocks for use with the Action menu detailed below.
- **New Folder**  : Adds new folder. Note that to edit the folder name, you need to click on the Edit icon, and put your mouse on the name, click and then type new name.
- **Menu**  : Before clicking on the menu, make sure to click Edit and select the required Playbooks/Blocks. Clicking on the Menu opens up the following actions:
 - **Duplicate** Click to create a duplicate Playbook with the following options:
 - Keep or Change Priority – you have the option to keep the original priority value to change to a different one
 - Keep in same folder or move to a different folder
 - Choose environments it belongs to. Options include single or multiple environments or all environments where “all” means all currently defined environments as well as environments that will be defined in the future.




A dark-themed dialog box titled "Duplicate" with a close button (X) in the top right corner. Below the title, it says "1 Playbooks selected". The dialog contains three settings: "Priority" with a dropdown menu showing "Keep original value", "Folder" with a dropdown menu showing "Keep original value", and "Environment" with two radio button options: "All Environments (current and future)" and "Select Environments" (which is selected). Below the radio buttons is a dropdown menu showing "Default Environment". At the bottom right are two buttons: "Cancel" and "Save".

- **Priority** Click to change priority.



A dark-themed dialog box titled "Change Priority" with a close button (X) in the top right corner. Below the title, it says "3 items selected". The dialog contains a single setting: "Priority" with a dropdown menu showing "1 First". At the bottom right are two buttons: "Cancel" and "Save".

- **Export/Import** Useful for sending both Playbooks and Playbook blocks from staging to production server and vice versa. Note that the system only recognizes zip files for import.
- **Move To** Can move Playbooks/Blocks to another folder or even create a new folder from this option.
- **Delete** Delete Playbooks/Blocks.
- **Filter**  : Clicking on the Filter opens up the following dialog box where you can choose to filter the display based on:
 - Active/Inactive Playbooks – also referred to as Enabled/Disabled
 - Priority Level
 - Environments (multi-select option)

Playbooks Filters

×

Show Active Playbooks

☐

Priority

All

▼

Environments

Default Environment

▼

Reset

Cancel

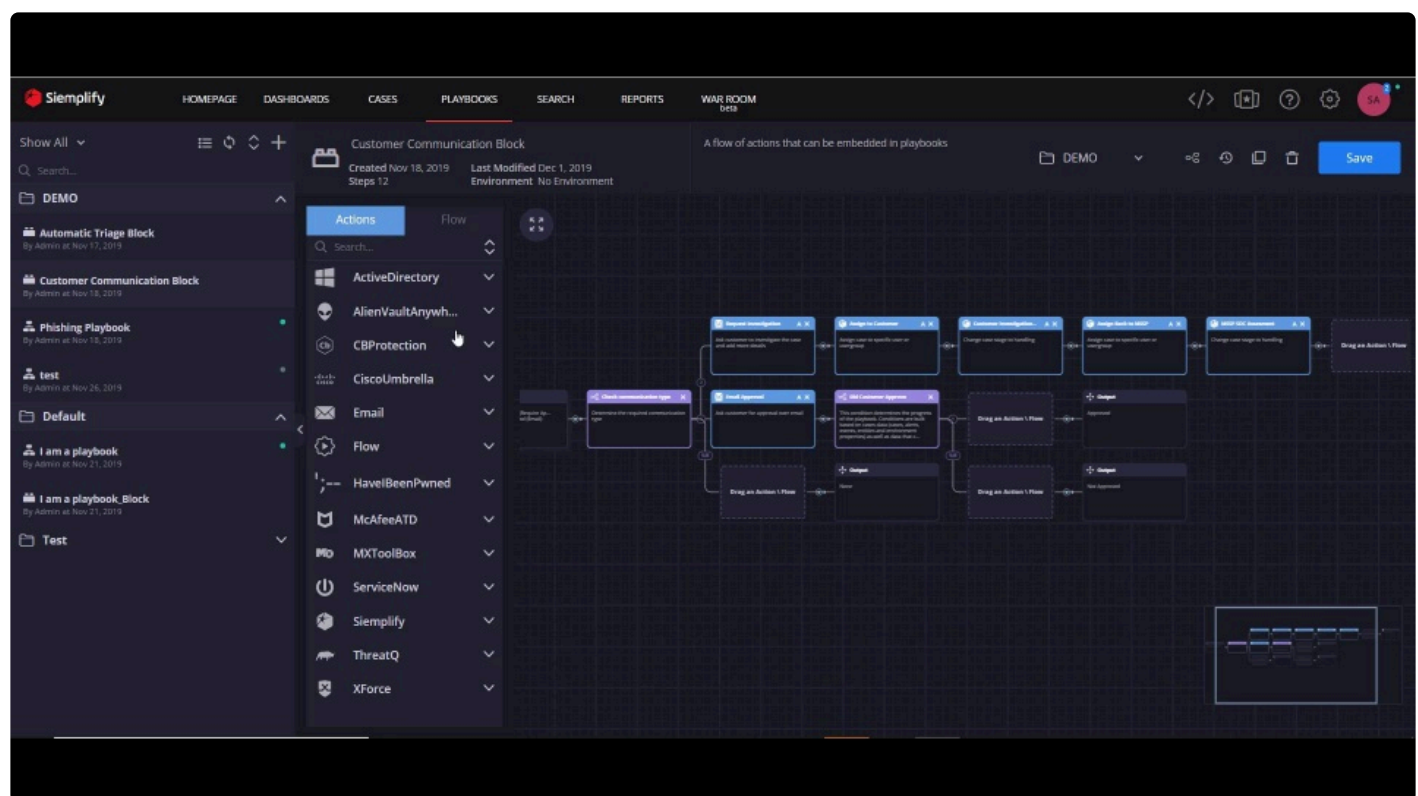
Save

Creating Playbook Blocks

Blocks are mini playbooks that users can create and reuse in other playbooks. The Blocks can implement workflows and logical decisions that might be useful in multiple playbooks. When you edit or change a Block, all playbooks using it will be affected which allows easy maintenance and playbooks improvement. When Blocks are used within other playbooks, users can configure Input parameter fields into the Block to alter its inner flow of actions.

The Block can also return an Output value into the parent playbook to allow interaction and conditioning between the two. Before you create these blocks, it's advisable to map out specific processes that you can easily reuse in parent playbooks, as well as giving thought to Inputs which can be configured per need.

The following video provides a more detailed explanation into creating and using Playbook Blocks.



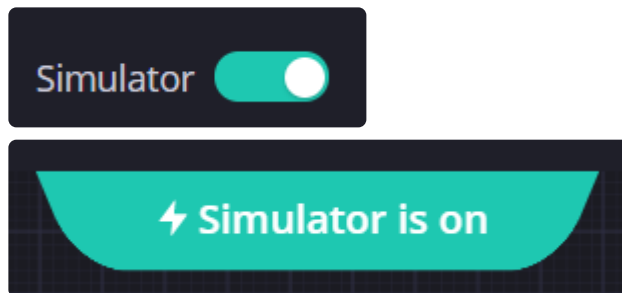
Boost playbook development efficiency with the Playbook Simulator

The Playbook Simulator provides you with a revolutionary way to develop Playbooks in less time and with less effort. Here are some of the standout features of the Simulator:

- Allows you to work in a pre-production environment where you can test your actions and play with your results without affecting production data (if the Playbook is turned off).
- Test each step of a Playbook or Playbook block to verify the required flow.
- Allows you to manipulate the results of your actions.
- Simulate your playbooks with different tools and integrations even if you don't have access to them.
- Test all branches of the Playbook conditions.

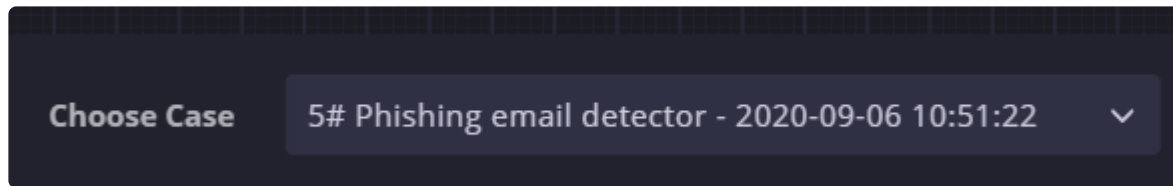
Using the Simulator on your playbook

1. Using an existing playbook or when creating a new one, your first step is to turn the Playbook Simulator on. The entire time the Simulator is on – you will have a green sign at the top of the screen.

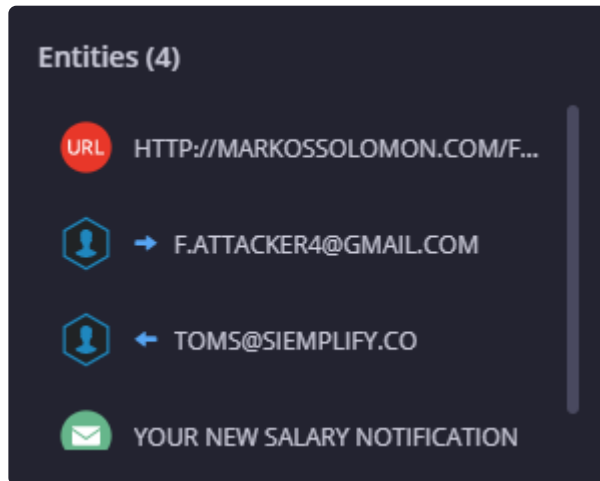


* Note that if you work with both the Playbook simulator on an enabled Playbook then all incoming alerts that trigger the playbook will be affected. This is because if you save a playbook with simulation data that you have inserted, this will be used on a live Case in production and may skew the results.

2. Navigate to the Cases screen, place your cursor on one of the alerts in the case and click Simulate Alert. This will create a test case in the system which you can then use to run your simulated Playbook on. Entities which are manipulated in test cases do not have any influence on entities in regular cases.
3. Move your cursor to the bottom of the screen and select a test Case.



- Note that the test case you select needs to match the Playbook you run. A good way of checking that the Case and Playbook match is to click on the Entities button on the right and see that the entities in the test case can be processed by the Playbook. The screenshot below shows the entities in the test Case we selected in the step above.

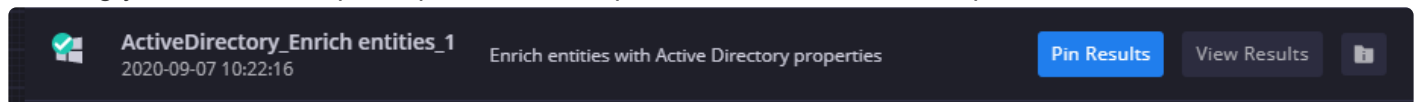


- Click Run. The Simulator starts processing the steps and running the Actions and providing the results.

How to understand the results of the Simulator for each step

After you click Run, the first row in the console will be the playbook trigger – and the simulator will show you success/failure of whether the specific trigger would be attached if it was a live Playbook in runtime.

For each step that is run in the Simulator, various options will be displayed, most commonly: Case Data, View Results, Pin Results (or Insert Results). For manual steps – a manual execution button will appear, allowing you enter the required parameters/responses and execute the step.





Case Data



On the far right, you have the Case Data icon. The case data dialog shows the case information at a specific stage (after this specific action finished running). Case data dialog will be updated at playbook runtime with the current step results. If this step added enrichment to the case you will see it here. Because it shows information for the current state of the case (what the case looks like when the step finished running) – this data will be different for each step of the simulation console. Inspecting the case data of different steps will explain what was changed in the case between the different steps execution. As you can see from the screenshot below, there are different tabs you can click through which provide more

information.

 **Case Data** ×

Action  ActiveDirectory_Enrich entities


Time 2020-09-07 10:22:16

Case

Alert

Events

Entity Details

 Search...

Field Name	Value
Case Name	Phishing email detector
Case ID	19
Case Description	
Priority	Informative
Assignee	@Tier1
Environment	Default Environment
Is Incident	false
Is Important	false
Tags	
Case Modification Time	2020-09-07 10:22:16
Case Creation Time	2020-09-07 10:22:16

Close

View Results

Next option displayed is “View Results” – this will show you the action results of the current step. The information displayed here is similar to case overview / case wall – view results and also includes information on enrichments. You can see output message, tables, links, attachments in the main tab, and you can see action result, and JSON result (where present) in the technical details tab.

As with the previous option, there is also an option to tab through data. Another action you can take from

the View Results screen is to click Set JSON result.

“Set JSON result” will replace the JSON sample of this action. The JSON sample can be changed from the IDE as well, and can be used within the Expression Builder to extract data from the JSON result.


Pin Results

This will be displayed if a step ran successfully.

Pin Results is a very useful option which allows you to take the result of the action and consider it as “fixed”. This can potentially save you time by not needing to wait for third party results and save credentials by reducing the number of queries to the 3rd party

In other words, when you rerun the Playbook again, this step is “passed over”, the code is not run, and the pinned results are taken as is. You also have the option here to manipulate the outcome and put in your own mock data. The screenshot below shows the Pin Results screen for this step. As you can see the Action has the simulate mode enabled and the step has changed from a blue background to a gray background to indicate that the Simulate mode is on. With regards to adding simulation data in to this screen, we will go into this in more detail in the next section. (Simplify actions will not have the option to pin results as there is no option to run them in simulate mode) Note that enrichments are not automatically added and need to be added manually in the Enrichment tabs.


ActiveDirectory - Enrich entities

ActiveDirectory_Enrich en

Simulate ☒

Enrich entities with Active Directory properties

Entities

All entities


Action Results

Enrichment

Script Results

Key	Value
is_success	

JSON Result



No JSON to present

Clear All

Cancel

Save

Insert Results

This will be displayed if a step failed.

Insert Results allows you to manually insert simulation data so that the next time you run this step, it will return the simulation data as the result. Once you click on this option, the action has the simulate mode enabled and the step changes from a blue background to a gray background to indicate that the Simulate mode is on. Note that “Script Result” is a mandatory field for all steps in simulation mode.

Why would I want to insert simulation (mock) data?

There are several reasons why you might want to insert mock data and we will discuss them briefly here.

- I want to build and test my playbook on the go. In this case I can run a step, view the results and understand how I can use it further along in the playbook.
- After a successful run, I want to pin the step results and save time by not running it again and again against third parties APIs
- I want to change my step results to test my playbook in different scenarios. In this case I'll set different simulation data to influence conditions and actions that are using previous results. For example, if you have a Playbook with a condition that branches off to two or more branches. You can play around with the simulation data in order to "force" the playbook to take a different branch each time and hence evaluate the entire Playbook.

How do I insert simulation data?

There are several ways of inserting simulation (mock) data: via Step configuration dialog in a playbook or via Pin Results (or Insert Results) after a Simulator run.

Playbook Step configuration dialog

1. Click on the step configuration dialog in the Playbook and toggle the Simulate mode to enable. The Action will be displayed as grey.
2. In the Action results, insert your simulation (mock) data – such as script result, JSON result by taking specific data from the JSON code. You can use enrichments from previous simulation runs or create custom enrichment keys of your own.
3. This data will now be returned every time the step runs.

Pin Results

1. Click on Pin Results next to the Action. The Step will open up already in Simulate mode.
2. In the Action results, insert your simulation (mock) data – such as script result, JSON result by taking specific data from the JSON code. You can use enrichments from previous simulation runs or create custom enrichment keys of your own.
3. This data will now be returned every time the step runs.

Insert Results

1. Click on Insert Results next to the Action.
2. In the Action results, insert your simulation (mock) data – such as script result, JSON result by taking specific data from the JSON code. You can use enrichments from previous simulation runs or create

custom enrichment keys of your own.

3. This data will now be returned every time the step runs.

Turning off the Playbook Simulator

Once you turn off the Playbook Simulator, the bottom Console is hidden and any step that might be in simulate mode will be reverted back to regular “live” mode. One exception to this rule is the Playbook Blocks – you need to turn off the block Simulator itself in order to close the simulation mode for it.

Note that any simulation mock data that you have inserted will be saved for you to use the next time you turn on the Simulator.

Working with Playbook blocks

You can also use the Playbook Simulator when building a new Playbook block. Note that when a block is in simulation mode – all the playbooks using this block will also be using the block’s mock data.

Playbooks FAQ

How do I scan multiple URLs in VirusTotal?

Answer

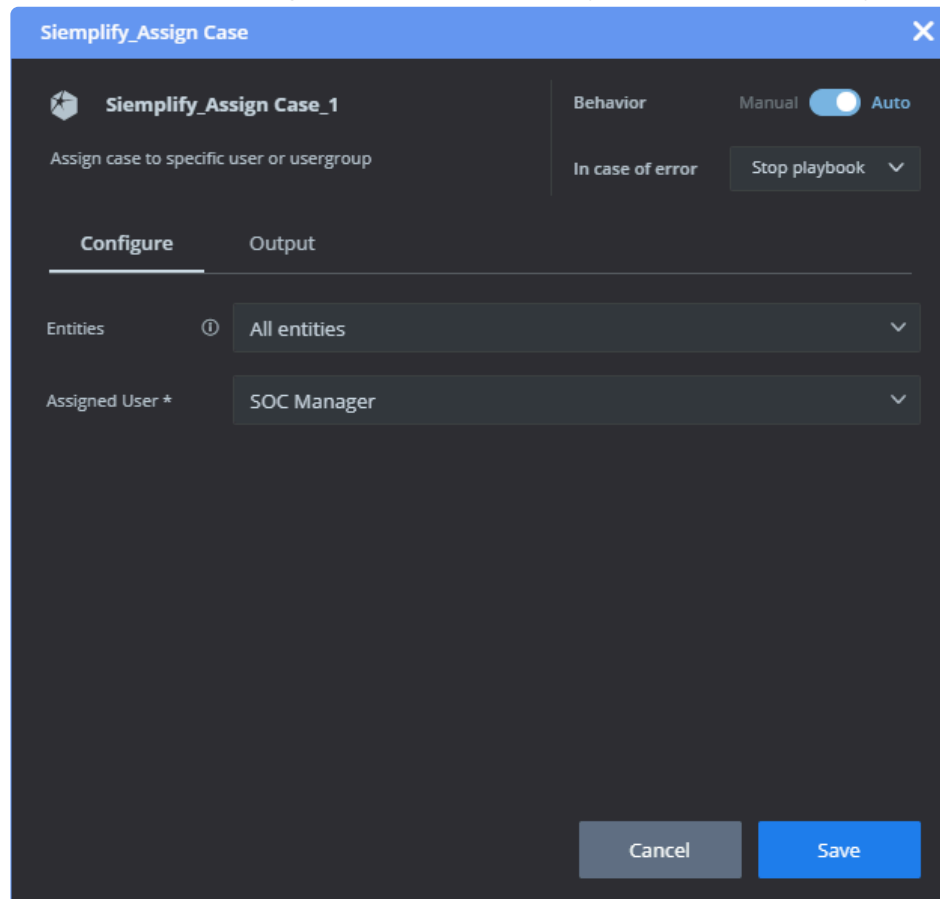
The VirusTotal – Scan URL action will iterate over the selected scope entities, and initiate a request to VirusTotal for each entity whose type is URL. When finished, the action will enrich the URL entities with a VirusTotal report and also post the result on the case wall.

An “is_risky” value will also be exposed so you can add further conditions to the playbook if some URLs were found risky.

Is there an auto-assignment feature that assigns cases to users?

Answer

You can use the Assign Case action in a Playbook to automatically assign a case to a user.



The screenshot shows a configuration window titled "Simplify_Assign Case" with a close button (X) in the top right corner. The window is divided into two main sections: "Configure" and "Output". The "Configure" section is active and contains the following elements:

- Header:** "Simplify_Assign Case_1" with a star icon and a description "Assign case to specific user or usergroup".
- Behavior:** A toggle switch set to "Auto" (with "Manual" as the alternative).
- In case of error:** A dropdown menu currently showing "Stop playbook".
- Entities:** A dropdown menu with a help icon (i) and the selected value "All entities".
- Assigned User *:** A dropdown menu with the selected value "SOC Manager".

At the bottom of the window, there are two buttons: "Cancel" (grey) and "Save" (blue).

How do I put elements of the case data into an email message?

Answer

This is done by using the Placeholders in the Playbook Actions. Placeholders are expressions that reference fields on events, alerts, cases entities and other elements. Placeholders will hold a place for the data that will be extracted when the action actually runs.

To use a placeholder, click on the “< >” button inside a text field (e.g. the message field in a “Send Email” action) and select the content you would like to extract when the action runs.

The screenshot shows the 'Email_Send Email' configuration window. The title bar is 'Email_Send Email'. The main area is divided into a header and a configuration section. The header includes an email icon, a text field 'Email_Send Email_1', a 'Behavior' section with 'Manual' and 'Auto' toggle (Auto is selected), and an 'In case of error' dropdown set to 'Stop playbook'. The configuration section has two tabs: 'Configure' and 'Output'. Under 'Configure', there are fields for 'Entities' (set to 'All users'), 'Recipients *' (containing '[Entity.AD_Email]'), 'Subject *' (containing 'Important Case'), 'Email HTML Template:' (a dropdown), and 'Content *' (containing '[Case.Name][Case.Id]'). There are 'Cancel' and 'Save' buttons at the bottom right.

You can string several placeholders in a row to create richer content.

Notice that a placeholder will always begin and end with a square bracket. Inside those brackets will always be the path of the required content. For example: The [Alert.Name] placeholder will check the alert name and put in the message before sending the email.

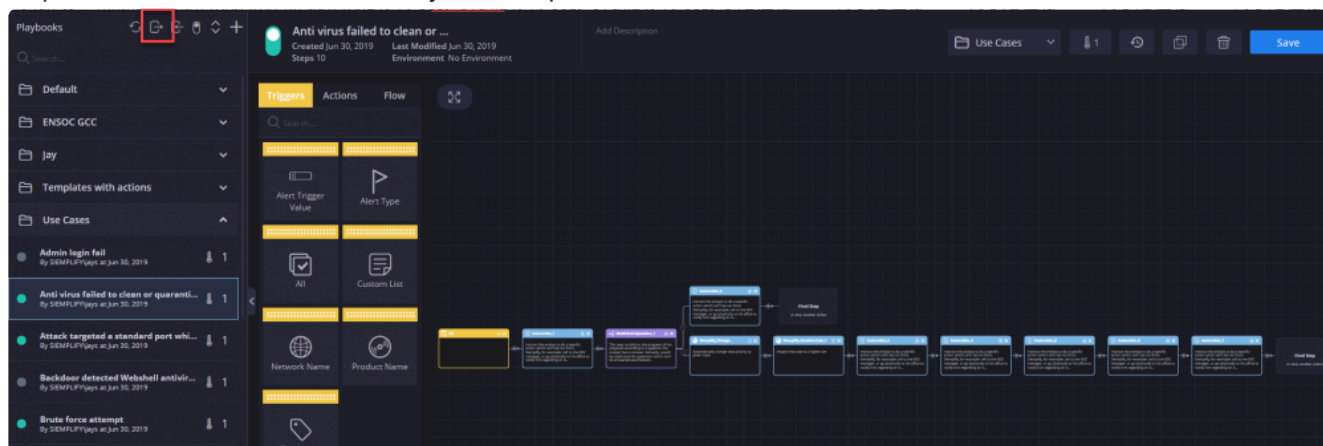
Remember that Placeholders can be used in email templates, allowing you to set a template once, with all the required placeholders and use it when needed (Setting > Environments > Email Templates).

How do I export and import Playbooks?

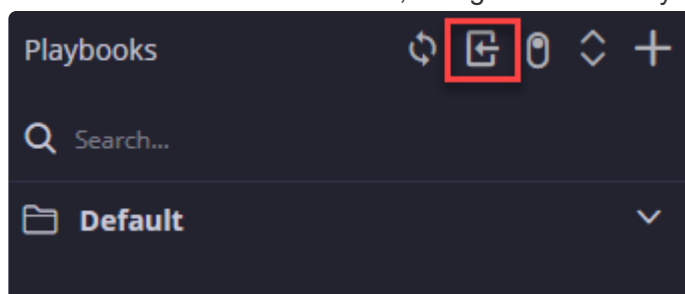
Answer

Exporting and importing Playbooks can be useful when creating information in a Staging environment and then pushing it into Production.

1. In the staging environment, in the Playbooks screen, highlight the Playbook that you created and click Export. The orch file is saved to your computer.



2. In the Production environment, navigate to the Playbooks screen and select Import.



3. Choose the orch file that you downloaded earlier and select it to import.
4. Assign a name and an environment to the newly imported playbook.

Can I scan URLs received by email?

Answer

The first step is to configure a connector that monitors an email box (with the Email or Exchange integrations). When email starts coming into Simplify, their content can be either parsed by the Mapping feature or extracted by the Create Entity playbook action (if playbooks are attached to the incoming emails). Once all URLs are extracted they can be used in manual actions and in playbooks.

Is it possible to send messages to a phone number?

Answer

Prerequisites

Twilio integration installed and an active account with Twilio.

Once configured in the Marketplace, Twilio actions will allow you to send SMS messages and even fork playbooks according to an SMS response.

How many playbooks can be assigned to a single alert?

Answer

Simplify allows attaching 3 playbooks automatically to a single alert, and attaching up to 7 playbooks manually.

How is an SLA calculated?

Answer

When an SLA is attached to an alert, it will look at the Start Time of this alert and then the defined SLA time to calculate the deadline.



SLAs are attached to alerts. When a case groups together several alerts, the case will present the shortest SLA out of all alerts.

What is the difference between Alert Grouping and Alert Overflow?

The Alerts Overflow mechanism was designed to prevent system overflow, when lots of alerts from the same environment, product and rule are occurring in a short period of time. The default configuration is more than 50 alerts in 10 minutes.

If configured, an Overflow case will be added to the case queue, with one alert indicating the environment, product and rule of the overflowing alert, and an Overflow tag.

The Alert Grouping mechanism was designed to intelligently group alerts into cases, by mutual entities and time proximity, and help the analyst to perform contextual analysis of multiple alerts in one case.

This means you would see multiple alerts in one case, and mutual entities marked in the entities list and the Explorer screen.

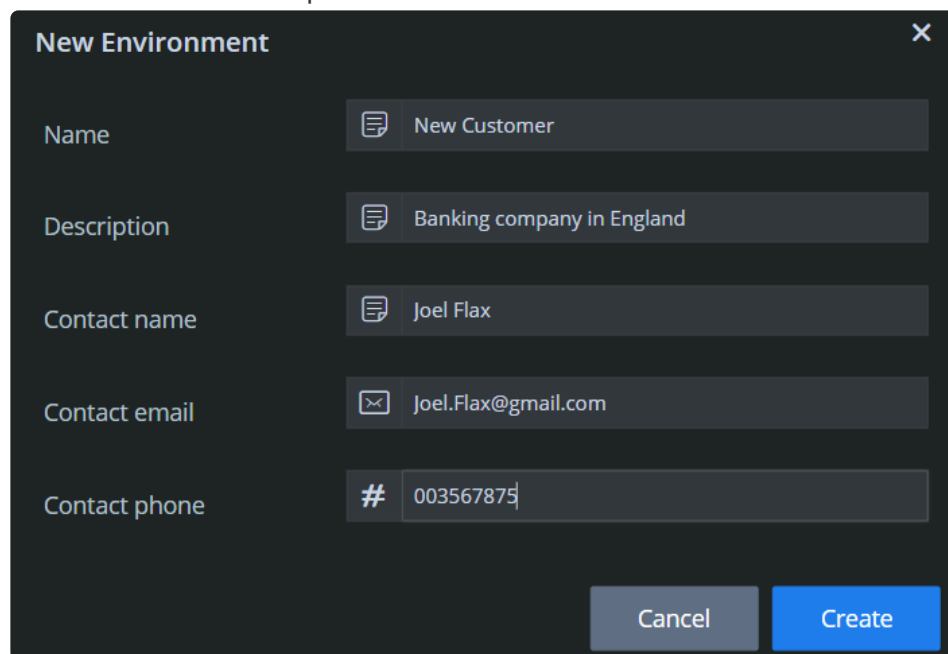
Multi-Tenancy Features

Supporting Multi-Tenancy (Environments)

Simplify allows MSSPs and SOCs to manage several different customers at a time. This is carried out using Environments. An environment is the Simplify term for a customer space.

To add new environments:

1. Click on the wheel icon the main page of the Simplify platform.
2. Select Settings > Organization > Environments.
3. Click on the + icon.
4. Fill out the fields as required.



The screenshot shows a 'New Environment' modal form with a dark background. It contains five input fields, each with a small icon on the left: 'Name' (document icon) with the value 'New Customer', 'Description' (document icon) with the value 'Banking company in England', 'Contact name' (document icon) with the value 'Joel Flax', 'Contact email' (envelope icon) with the value 'Joel.Flax@gmail.com', and 'Contact phone' (hash icon) with the value '003567875'. At the bottom right, there are two buttons: a grey 'Cancel' button and a blue 'Create' button.

5. Click Create. The new environment shows up on the screen. You can choose to add an image to this tenant from the main screen.

Management

Using Advanced Reports

Overview

Siimplify enables you to track and analyze every aspect of your security operation using Advanced Reports which synchronize with [Tableau](#), the world's leading data visualization software.

This integration with Tableau provides you with complete visibility and control over your security operation and a clear view of its business aspects.

✿ Using Advanced Reports requires installation and deployment of a dedicated Tableau server. Please contact your Siimplify CSM to discuss license and installation details.

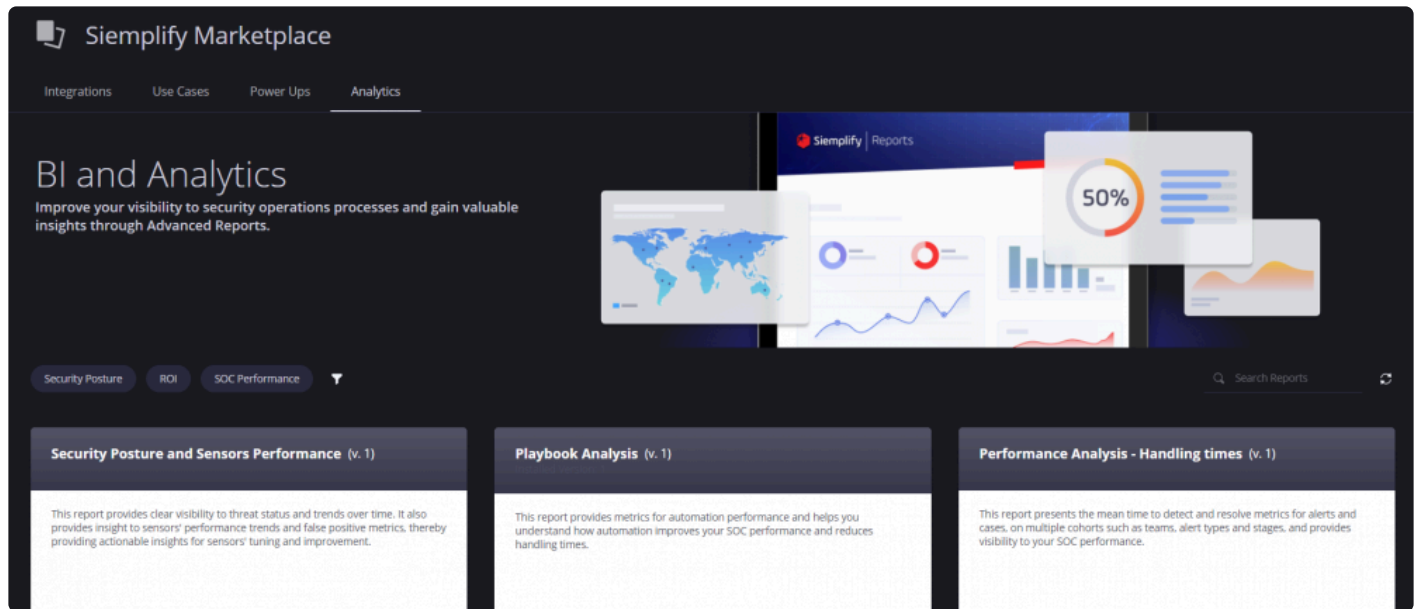
Creating Users to see Advanced Reports

✿ The amount of users that you can add will be defined according to your Advanced Reports license.

1. In the Siimplify platform, navigate to Settings > Organization > User Management
2. Create/Edit a user with the following levels of access to Advanced Reports according to your requirements:
 - No Access
 - Viewer – this level of user will only be able to see Tableau reports that were specifically shared with them from the Advanced Reports screen in the Siimplify platform.
 - Editor – this user can see all the Tableau reports. They can also upload reports and share reports.
3. Make sure to Save.

Marketplace Advanced Reports

Once Siimplify has been set up, you can navigate to the Marketplace > Analytics tab to download out of the box reports. Once you have downloaded them, you will be able to view them from the Reports tab.



Below are examples of some of the the out-of-the-box reports Simplify have defined for you:

Security Posture and Sensors Performance

This report provides clear visibility to threat status and trends over time. It also provides insights into sensors' performance trends and false-positive metrics, thereby providing actionable insights for sensors' tuning and improvement.

Playbook Analysis

This report provides metrics for automation performance and helps you understand how automation improves your SOC performance and reduces handling times.

Performance Analysis – Handling times

This report presents the mean time to detect and resolve metrics for alerts and cases, on multiple cohorts such as teams, alert types, and stages, and provides visibility to your SOC performance.

Performance Analysis – Analysts Workload

This report provides a clear view of your SOC's workload via alerts and events distributions, open vs closed cases' trends, alert grouping performance over time, and false positive trends.

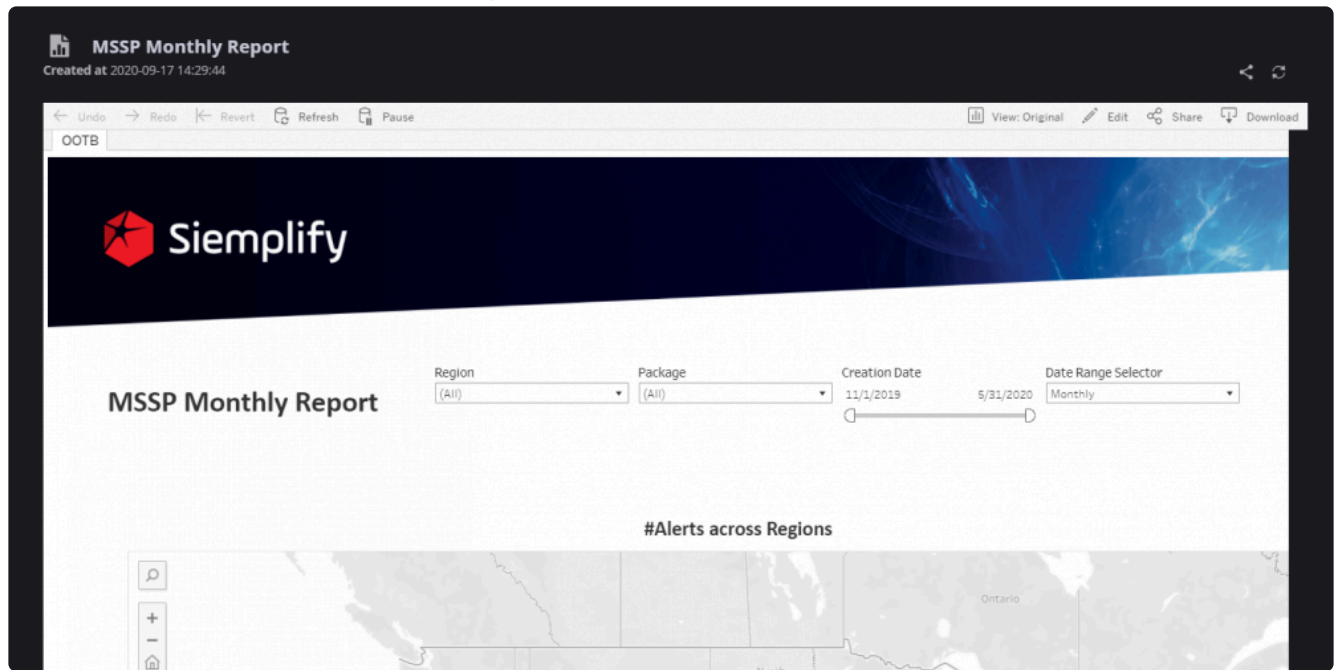
Customer Report

This report is aimed to give MSSP customers visibility to their security services status, security posture, and important events on a regular basis.

Viewing Reports


1. Navigate to the Reports tab and click on Advanced Reports.
2. On the left side you will see a list of available Reports that you have downloaded from the Marketplace.

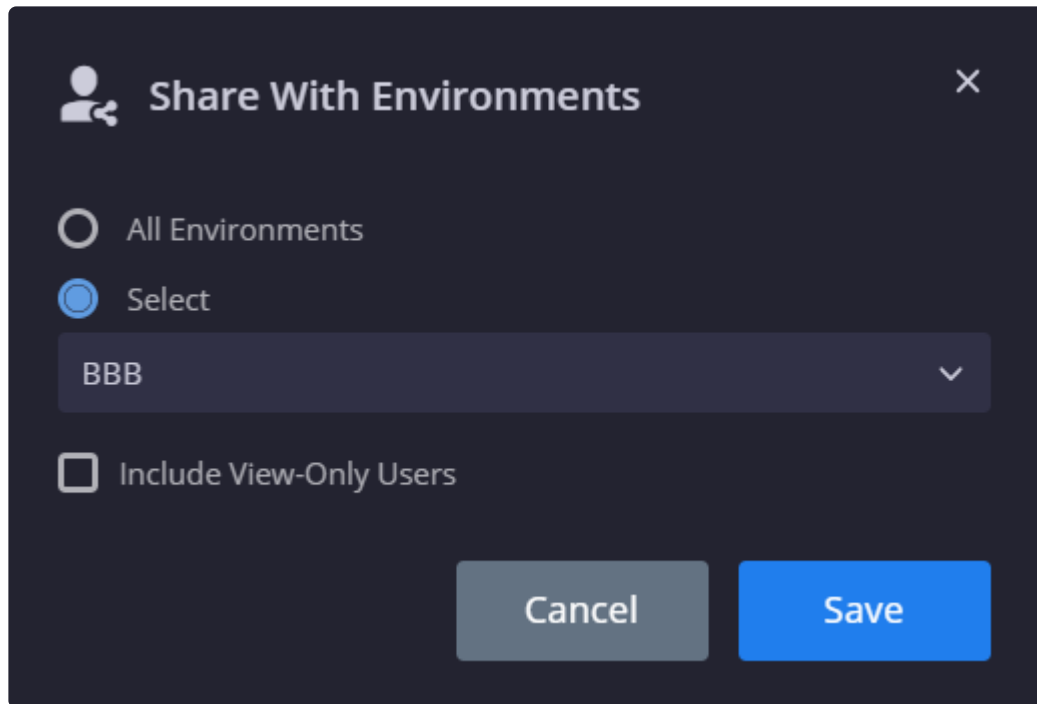
- Let's take a look at the MSSP Monthly Report as an example.



- At the top of the report several buttons are displayed which are embedded Tableau functionality and as such will only carry out actions seen in Tableau. As such, Siemplify only recommends using the embedded Download and Refresh buttons.

Share Reports

- At the top of each screen on the right is the option to share reports using the Share icon 
- Click Share
- Select the environments for the reports that you want to share with Siemplify users of those environments.
- If you want to share the reports with users who have view only access, make sure to select the checkbox.



Deleting Reports

Reports can only be deleted on the Tableau dashboard.

Getting Help

At the top of your dashboard is a question mark icon which leads you to Tableau documentation and support.

For more information please refer to [Simplify Tableau Deployment](#)

Administrative

Alert Grouping Rules

The Simplify Alert Grouping mechanism groups together alerts into cases in order for the Security Analyst to have a better context of issues they need to tackle.

The goal is to accentuate the importance of additional context to a security alert and avoid situations where the analysts will investigate the same security incident without the context and may lose precious time or even handle this incident in the wrong way.

The Alert Grouping mechanism allows you to create grouping rules controlling the exact type of alerts which will be grouped together into cases. The rules work on a hierarchical system whereby each incoming alert is matched against a rule in the following order:

1. Alert Type
2. Data Source
3. Product

Note that if an alert matches a rule, and there is no existing case it can be grouped into, then it will be added to a new case.

In addition, the platform has an out of the box rule which cannot be deleted. This fallback rule provides a general catchall for alerts to ensure that there is always grouping in Cases. However, two options (group by, grouping entities) can be edited.

The Grouping Mechanism configuration can be found in the Settings > Advanced > Alerts Grouping to configure the settings.

Alerts Grouping
Manage preferences for alert grouping into Cases.

General

Max. alerts grouped into a Case ⓘ
20

Timeframe for grouping alerts (in hours) ⓘ
22

Group entities and source grouping identifiers in the same case ⓘ
☒

Rules
Each entity will be checked according to a hierarchy in the following order: Alert Type > Product > Data Source, and will be grouped into the relevant Alert. Simplify comes with a fallback rule which will always display at the bottom of the list.

Category	Value	Group By	Grouping Entities
Data Source	Arcsight	Entities	Both Directions (Source and desti...
Data Source	Qradar	Source Grouping Identifier	
All	All alerts	Entities	Both Directions (Source and desti...

< 1-3 of 3 >

Overflow

Timeframe for Overflow Case grouping (in hours)
2

Max alerts grouped into an Overflow Case
50

Save

In the General part of the screen, you have the following cross-platform settings:

- **Max. alerts grouped into a Case:** Define the maximum number of alerts to group together into one

Case (20). After the maximum number is reached, a new Case is started.

- **Timeframe for grouping alerts (in hours):** Choose the number of hours with which to group the alerts for the Case (0.5-24 hours with half hour intervals supported). Note that this does NOT apply to the rules below which are grouped by Source Grouping Identifier.
- **Group entities and source grouping identifiers in the same case:** When enabled, an alert that should be grouped by “source grouping identifier” according to the grouping rule, will first look for alerts with the same source grouping identifier, and if it doesn’t find any, it will look for all cases in the system with mutual entities and group alerts accordingly (and according to the cross-platform timeframe)

Note: Whether disabled / enabled, if there is a rule with **group by: entities**, it will look for all cases in the system with common entities (even if they were originally grouped by source grouping identifier)

The rules section allows you to create specific rules targeting different options.

So as a basic example of grouping let’s say an alert “C&C traffic” with destination host 10.1.1.13 is added at 10:00 AM to a case called “Malware Found”.

And another alert “User account changed” with the same destination host is seen at 11:00 AM. Simplify identifies the same entity which is involved in both alerts and within the configured time frame, and groups the second alert to the “Malware Found” case.

Let’s look now at creating rules for specific use cases.

Use Case #1

An enterprise company works with 2 connectors – Arcsight and Cybereason EDR. They want to group Arcsight alerts according to both source and destination entities (Rule 1). In addition, they have two kinds of alerts ingested from the Cybereason EDR. They want to group Cybereason EDR Phishing alerts based on source entities only (Rule 2) and group Cybereason EDR Failed Login alerts based on destination entities only (Rule 3).

In order to capture this use case – they need to build the three rules as shown in the following screenshot. Note that the final rule is the fallback rule supplied by Simplify.

Rules
 Each entity will be checked according to a hierarchy in the following order: Alert Type > Product > Data Source, and will be grouped into the relevant Alert. Simplify comes with a fallback rule which will always display at the bottom of the list.

<div> <div><</div> <div>1-4 of 4</div> <div>></div> <div>✎</div> <div>+</div> </div>			
Category	Value	Group By	Grouping Entities
Alert Type	PHISHING	Entities	Sources Entities Only
Data Source	Arcsight	Entities	Both Directions (Source and desti...
Data Source	Qradar	Source Grouping Identifier	
All	All alerts	Entities	Both Directions (Source and desti...



Use Case #2

An MSSP has a customer who is using the Qradar connector. The MSSP wants to utilize the Qradar Grouping so that they can see cases in Simplify in the exact same way that the customer sees their cases in Qradar (Rule 1). The MSSP has another customer who is using Arcsight and they want to group alerts by common entities and for both directions (Rule 2)

But for Phishing alerts for this customer, they want to group by the destination entities, without taking the source entities into account (Rule 3)

In order to capture this use case – they need to build the following three rules:

Rules
Each entity will be checked according to a hierarchy in the following order: Alert Type > Product > Data Source, and will be grouped into the relevant Alert. Simplify comes with a fallback rule which will always display at the bottom of the list.

< 1-4 of 4 >  

Category	Value	Group By	Grouping Entities
Alert Type	PHISHING	Entities	Destinations Entities Only
Data Source	Arcsight	Entities	Both Directions (Source and desti...
Data Source	Qradar	Source Grouping Identifier	
All	All alerts	Entities	Both Directions (Source and desti...

Simplify Logs

✿ This article is relevant from 5.5.1 onwards

Simplify log files are now created on an hourly basis and kept for 7 days. You can access log files on the server and download them in one click. This article provides more information on the types of information that will be stored as log files, and how and where to access them.

Where are these log files stored?

At the top level, the log files are stored under* `/var/logs/simplify*`.

If we drill down, we can find the following logs under their names.

For example:

Simplify connectors: `"/var/logs/simplify/connector name"`

Simplify jobs: `"/var/logs/simplify/job name"`

Simplify integrations: `"/var/logs/simplify/Integrations/integration name"`

Simplify Services: `"/var/logs/simplify/service name"`

Simplify Services

The following services will have log files in a dedicated folder:

- Python Connector instances (each connector)
- Python Integrations (each integration)
- Python Jobs
- Indexer service
- Playbook actions service
- Connector service
- Python execution service
- App service
- ETL service

How often are these log files created?

Log files are created on an hourly basis per log component and stored in a daily folder. Each file stores the log records as created from HH:00:00 to HH:59:59. The daily folder name format is the date – according to server localization settings

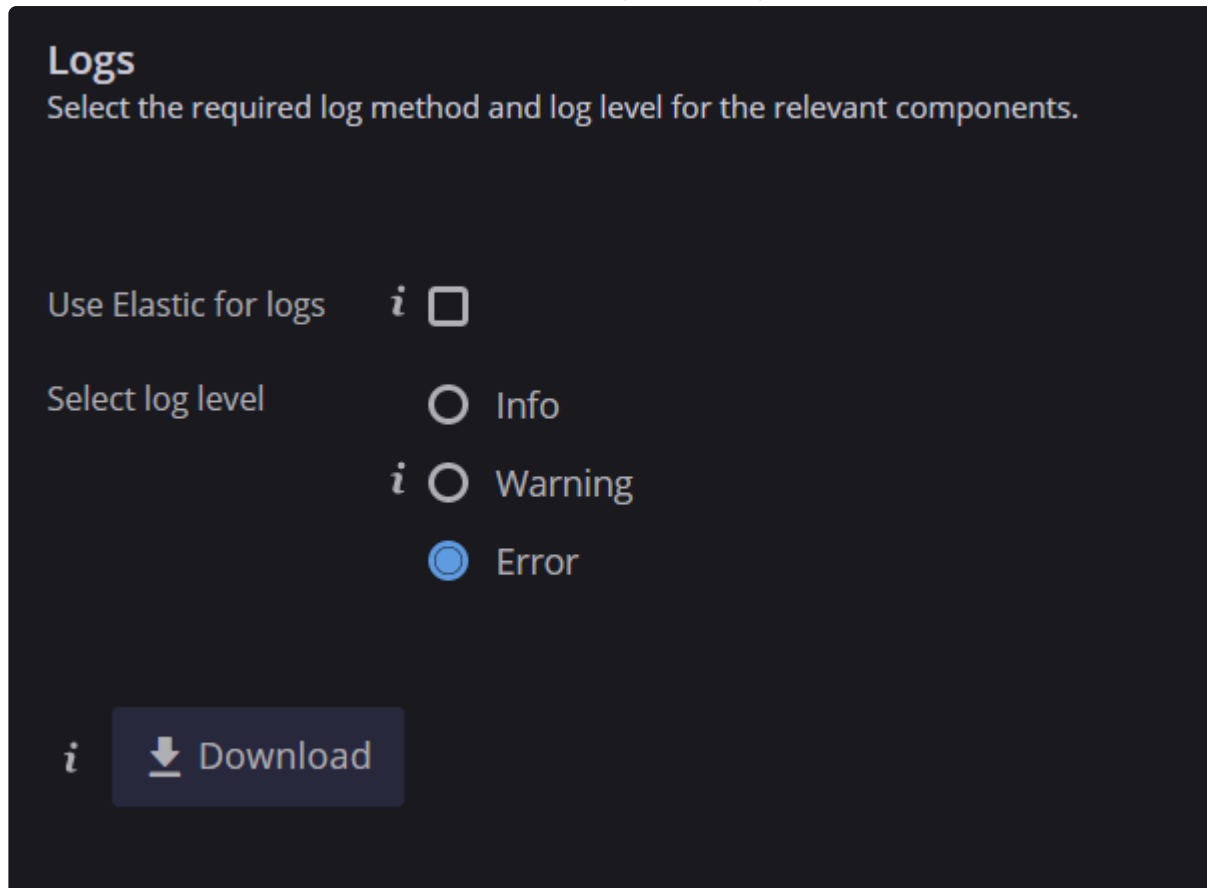
✿ Note that log files are only kept for 7 days. After this time, the log folders are deleted.

How can I change error level of logs?

1. Navigate to Settings > Advanced > Logs.
2. Choose from the following: Error (most critical), Warning (includes error), Info (Info includes any log record, including errors, warnings and many other system log records). Note that in this screen, you can also select to see Elastic logs. The level of logs that you choose here will apply to both Siemplify service logs and Elastic logs (if enabled). The change applies to log files from that moment on.

How can I see all the current logs?

1. Navigate to Settings > Advanced > Logs.
2. Click Download to download a zip file containing all the logs for the last ten hours.



The screenshot shows the 'Logs' configuration page in Siemplify. The title 'Logs' is at the top, followed by the instruction 'Select the required log method and log level for the relevant components.' Below this, there are two main sections. The first section, 'Use Elastic for logs', has a label 'i' and an unchecked checkbox. The second section, 'Select log level', has a label 'i' and three radio button options: 'Info', 'Warning', and 'Error'. The 'Error' option is selected, indicated by a blue dot. At the bottom left, there is a label 'i' and a 'Download' button with a download icon.

3. Alternatively, you can access the files directly from the server via the command line.

```
[root@localhost ~]# cd /var/log/simplify
[root@localhost simplify]# ll
total 0
drwxrwxr-x. 4 simplifyadmin simplifyadmin 48 May 24 09:49 agents
drwxrwxr-x. 9 simplifyadmin simplifyadmin 203 May 24 12:00 jobs
drwxrwxr-x. 3 simplifyadmin simplifyadmin 27 May 24 09:49 publishers
drwxrwxr-x. 8 simplifyadmin simplifyadmin 280 May 24 09:50 services
[root@localhost simplify]# cd jobs
[root@localhost jobs]# ll
total 0
drwxrwxr-x. 2 simplifyadmin simplifyadmin 28 May 24 12:00 Actions Monitor
drwxrwxr-x. 2 simplifyadmin simplifyadmin 138 May 24 14:00 Cases Collector_PublisherID_1
drwxrwxr-x. 2 simplifyadmin simplifyadmin 116 May 24 14:00 Connectors Monitor
drwxrwxr-x. 2 simplifyadmin simplifyadmin 116 May 24 14:00 ETL Monitor
drwxrwxr-x. 2 simplifyadmin simplifyadmin 28 May 24 12:00 Jobs Monitor
drwxrwxr-x. 2 simplifyadmin simplifyadmin 138 May 24 14:00 Logs Collector_PublisherID_1
drwxrwxr-x. 2 simplifyadmin simplifyadmin 28 May 24 12:00 Machine Resource Utilization
[root@localhost jobs]#
```

Set Shared Folder for Remote Backup (Linux)

[Install and Configure Samba Server on CENTOS](#)

Change SSL Certificates for Server and Client

Replace Certificate For Web Server (.crt and .key files)

1. Connect to a local server.
2. Go to `/etc/nginx/SSL/`
3. Replace the “certificate.crt” file and the “private.key” file with the new files with the same names.
4. Restart the nginx server with the following command:

```
systemctl restart nginx
```

Replace Certificate For API Access (.pfx certificate)

When installing Siemplify for the first time, the customer needs to provide the certificate and the passphrase on the command line, using `CERT_FILE` and `CERT_PASSPHRASE` parameters.

The installer will install the certificate over Siemplify server at `/opt/siemplify/siemplify_server/bin/cert.pfx`

Import and Export PostgreSQL Databases

You might want to import/export the PostgreSQL DB in the following situations:

- Moving the DB from one server to another (for example, staging to production).
- Moving from one architecture to another (HA, external DB).

Export Simplify databases

1. Change to postgres user by using this command:

```
su postgres
```

2. Execute the following commands by using the pg_dump command:

```
pg_dump simplify_agents_db>simplify_agents_db.bak
pg_dump simplify_dashboards_db>simplify_dashboards_db.bak
pg_dump simplify_entityexplorer_db>simplify_entityexplorer_db.bak
pg_dump simplify_homepage_db>simplify_homepage_db.bak
pg_dump simplify_integrations_db>simplify_integrations_db.bak
pg_dump simplify_jobs_db>simplify_jobs_db.bak
pg_dump simplify_metadata_db>simplify_metadata_db.bak
pg_dump simplify_monitoring_db>simplify_monitoring_db.bak
pg_dump simplify_notifications_db>simplify_notifications_db.bak
pg_dump simplify_ontology_db>simplify_ontology_db.bak
pg_dump simplify_orchestration_db>simplify_orchestration_db.bak
pg_dump simplify_report_system_db>simplify_report_system_db.bak
pg_dump simplify_search_everything_db>simplify_search_everything_db.bak
pg_dump simplify_system_db>simplify_system_db.bak
pg_dump simplify_war_room_db>simplify_war_room_db.bak
```

3. Verify that all the backup files are in the current folder:

```
-rw-r--r-- 1 postgres postgres 10835 Jan 28 08:39 simplify_agents_db.bak
-rw-r--r-- 1 postgres postgres 15624 Jan 28 08:39 simplify_dashboards_db.bak
-rw-r--r-- 1 postgres postgres 3463 Jan 28 08:39 simplify_entityexplorer_db.bak
-rw-r--r-- 1 postgres postgres 13062 Jan 28 08:39 simplify_homepage_db.bak
-rw-r--r-- 1 postgres postgres 359203 Jan 28 08:39 simplify_integrations_db.bak
-rw-r--r-- 1 postgres postgres 9804113 Jan 28 08:39 simplify_jobs_db.bak
-rw-r--r-- 1 postgres postgres 112664 Jan 28 08:39 simplify_metadata_db.bak
-rw-r--r-- 1 postgres postgres 81809 Jan 28 08:39 simplify_monitoring_db.bak
-rw-r--r-- 1 postgres postgres 10361 Jan 28 08:39 simplify_notifications_db.bak
-rw-r--r-- 1 postgres postgres 325009 Jan 28 08:39 simplify_ontology_db.bak
-rw-r--r-- 1 postgres postgres 18828 Jan 28 08:39 simplify_orchestration_db.bak
-rw-r--r-- 1 postgres postgres 93303 Jan 28 08:39 simplify_report_system_db.bak
-rw-r--r-- 1 postgres postgres 59844 Jan 28 08:39 simplify_search_everything_db.bak
-rw-r--r-- 1 postgres postgres 134974 Jan 28 08:39 simplify_system_db.bak
-rw-r--r-- 1 postgres postgres 30777 Jan 28 08:39 simplify_war_room_db.bak
```

4. Move these files into the postgres folder (in the target server) using this command:

```
/var/lib/pgsql
```

Import Backup Files

1. Connect to the DB via CLI as follows:

```
sudo -u postgres psql
```

2. Delete the existing databases (by using the DROP command):

```
DROP DATABASE siemply_dashboards_db;  
DROP DATABASE siemply_agents_db;  
DROP DATABASE siemply_entityexplorer_db;  
DROP DATABASE siemply_homepage_db;  
DROP DATABASE siemply_integrations_db;  
DROP DATABASE siemply_jobs_db;  
DROP DATABASE siemply_metadata_db;  
DROP DATABASE siemply_monitoring_db;  
DROP DATABASE siemply_notifications_db;  
DROP DATABASE siemply_ontology_db;  
DROP DATABASE siemply_orchestration_db;  
DROP DATABASE siemply_report_system_db;  
DROP DATABASE siemply_search_everything_db;  
DROP DATABASE siemply_system_db;  
DROP DATABASE siemply_war_room_db;
```

3. Create new (and empty) databases:

```
CREATE DATABASE siemply_agents_db;  
CREATE DATABASE siemply_dashboards_db;  
CREATE DATABASE siemply_entityexplorer_db;  
CREATE DATABASE siemply_homepage_db;  
CREATE DATABASE siemply_integrations_db;  
CREATE DATABASE siemply_jobs_db;  
CREATE DATABASE siemply_metadata_db;  
CREATE DATABASE siemply_monitoring_db;  
CREATE DATABASE siemply_notifications_db;  
CREATE DATABASE siemply_ontology_db;  
CREATE DATABASE siemply_orchestration_db;  
CREATE DATABASE siemply_report_system_db;  
CREATE DATABASE siemply_search_everything_db;  
CREATE DATABASE siemply_system_db;  
CREATE DATABASE siemply_war_room_db;
```

4. Change the owner of the DB to the local user using by siemply (default – sa)

```
ALTER DATABASE siemply_agents_db OWNER TO sa;  
ALTER DATABASE siemply_dashboards_db OWNER TO sa;  
ALTER DATABASE siemply_entityexplorer_db OWNER TO sa;  
ALTER DATABASE siemply_homepage_db OWNER TO sa;  
ALTER DATABASE siemply_integrations_db OWNER TO sa;  
ALTER DATABASE siemply_jobs_db OWNER TO sa;
```



```

ALTER DATABASE siemplify_metadata_db OWNER TO sa;
ALTER DATABASE siemplify_monitoring_db OWNER TO sa;
ALTER DATABASE siemplify_notifications_db OWNER TO sa;
ALTER DATABASE siemplify_ontology_db OWNER TO sa;
ALTER DATABASE siemplify_orchestration_db OWNER TO sa;
ALTER DATABASE siemplify_report_system_db OWNER TO sa;
ALTER DATABASE siemplify_search_everything_db OWNER TO sa;
ALTER DATABASE siemplify_system_db OWNER TO sa;
ALTER DATABASE siemplify_war_room_db OWNER TO sa;

```

5. Verify the databases. They should display as in the screenshot below:

```

postgres=# \list

```

Name	Owner	Encoding	Collate	Ctype	Access privileges
postgres	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_agents_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_dashboards_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_entityexplorer_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_homepage_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_integrations_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_jobs_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_metadata_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_monitoring_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_notifications_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_ontology_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_orchestration_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_report_system_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_search_everything_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_system_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
siemplify_war_room_db	sa	UTF8	en_US.UTF-8	en_US.UTF-8	
template0	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres +
					postgres=Ctc/postgres
template1	postgres	UTF8	en_US.UTF-8	en_US.UTF-8	=c/postgres +
					postgres=Ctc/postgres

```

(18 rows)
postgres=#

```

6. Exit postgres, and change the user to postgres by using this command:

```
su postgres
```

7. Make sure that all the .bak files are located in the postgres folder:

```
/var/lib/pgsql
```

```
[root@rhel-7-1 postgresql]# su postgres
bash-4.2$ ls -la
total 10868
drwx-----. 3 postgres postgres 4096 Feb 10 09:45 .
drwxr-xr-x. 31 root root 4096 Dec 18 14:00 ..
drwx-----. 4 postgres postgres 51 Dec 18 13:54 10
-rwxr-xr-x. 1 postgres postgres 977 Jan 28 08:34 backup_db.sh
-rw-----. 1 postgres postgres 1989 Feb 10 13:19 .bash_history
-rwx-----. 1 postgres postgres 266 Dec 18 13:54 .bash_profile
-rwxr-xr-x. 1 postgres postgres 83 Feb 10 09:25 drop_db.sh
-rw-----. 1 postgres postgres 312 Jan 28 10:35 .psql_history
-rwxr-xr-x. 1 postgres postgres 705 Jan 29 15:29 restore.sh
-rw-r--r--. 1 postgres postgres 10835 Jan 28 08:39 siemplyfy_agents_db.bak
-rw-r--r--. 1 postgres postgres 15624 Jan 28 08:39 siemplyfy_dashboards_db.bak
-rw-r--r--. 1 postgres postgres 3463 Jan 28 08:39 siemplyfy_entityexplorer_db.bak
-rw-r--r--. 1 postgres postgres 13062 Jan 28 08:39 siemplyfy_homepage_db.bak
-rw-r--r--. 1 postgres postgres 359203 Jan 28 08:39 siemplyfy_integrations_db.bak
-rw-r--r--. 1 postgres postgres 9804113 Jan 28 08:39 siemplyfy_jobs_db.bak
-rw-r--r--. 1 postgres postgres 112664 Jan 28 08:39 siemplyfy_metadata_db.bak
-rw-r--r--. 1 postgres postgres 81809 Jan 28 08:39 siemplyfy_monitoring_db.bak
-rw-r--r--. 1 postgres postgres 10361 Jan 28 08:39 siemplyfy_notifications_db.bak
-rw-r--r--. 1 postgres postgres 325009 Jan 28 08:39 siemplyfy_ontology_db.bak
-rw-r--r--. 1 postgres postgres 18828 Jan 28 08:39 siemplyfy_orchestration_db.bak
-rw-r--r--. 1 postgres postgres 93303 Jan 28 08:39 siemplyfy_report_system_db.bak
-rw-r--r--. 1 postgres postgres 59844 Jan 28 08:39 siemplyfy_search_everything_db.bak
-rw-r--r--. 1 postgres postgres 134974 Jan 28 08:39 siemplyfy_system_db.bak
-rw-r--r--. 1 postgres postgres 30777 Jan 28 08:39 siemplyfy_war_room_db.bak
bash-4.2$ pwd
/var/lib/pgsql
```

8. Run the following commands to load the .bak files into the empty databases:

```
psql -U postgres siemplyfy_dashboards_db < siemplyfy_dashboards_db.bak
psql -U postgres siemplyfy_agents_db<siemplyfy_agents_db.bak
psql -U postgres siemplyfy_entityexplorer_db<siemplyfy_entityexplorer_db.bak
psql -U postgres siemplyfy_homepage_db<siemplyfy_homepage_db.bak
psql -U postgres siemplyfy_integrations_db<siemplyfy_integrations_db.bak
psql -U postgres siemplyfy_jobs_db<siemplyfy_jobs_db.bak
psql -U postgres siemplyfy_metadata_db<siemplyfy_metadata_db.bak
psql -U postgres siemplyfy_monitoring_db<siemplyfy_monitoring_db.bak
psql -U postgres siemplyfy_notifications_db<siemplyfy_notifications_db.bak
psql -U postgres siemplyfy_ontology_db<siemplyfy_ontology_db.bak
psql -U postgres siemplyfy_orchestration_db<siemplyfy_orchestration_db.bak
psql -U postgres siemplyfy_report_system_db<siemplyfy_report_system_db.bak
psql -U postgres siemplyfy_search_everything_db<siemplyfy_search_everything_db.ba
k
psql -U postgres siemplyfy_system_db<siemplyfy_system_db.bak
psql -U postgres siemplyfy_war_room_db<siemplyfy_war_room_db.bak
```

9. Run Siemplyfy server and validate the database (cases, playbooks, users, etc.)

White List for HTML Templates

To make changes to the white list:

1. Navigate to /opt/simplify/simplify_server/Configs and open the Siimplify_Main_Config.xml file in an XML editor.
2. Within the config file, locate the Client > HtmlTemplatesWhiteListConfiguration tag.

```
<Client>
  <DashboardAndManagementDataPollingTimeInMs>300000</DashboardAndManagementDataPollingTimeInMs>
  <CasesAndNotificationsPollingTimeInMs>60000</CasesAndNotificationsPollingTimeInMs>
  <MaximumNotificationsToShow>30</MaximumNotificationsToShow>
  <EntityFieldNamesPlaceholderBlackList>
    <FieldName>IsVulnerable</FieldName>
    <FieldName>OriginalIdentifier</FieldName>
    <FieldName>IsFromLdapString</FieldName>
    <FieldName>OrigIdentifier</FieldName>
    <FieldName>IsEnriched</FieldName>
    <FieldName>IsTestCase</FieldName>
  </EntityFieldNamesPlaceholderBlackList>
  <HtmlTemplatesWhiteListConfiguration>
    <AllowedTags>
      <Name>style</Name>
    </AllowedTags>
    <AllowedAttributes>
      <Name>class</Name>
    </AllowedAttributes>
  </HtmlTemplatesWhiteListConfiguration>
</Client>
```

3. Edit the Allowed Tags and Allowed Attributes to include the elements you need.



Please be very careful when making changes to the White list so as not to expose yourself to vulnerabilities. Siimplify recommends as best practice using the list of allowed items as defined [here](#).

How to increase the timeout of Jobs/Connectors/Actions in Siemplify

1. Log in to the machine via SSH.
2. Navigate to the following file: /opt/siemplify/siemplify_server/Configs/Siemplify_Main_Config.xml
3. Open up this file in a text editor.
4. Locate these two tags in the config file: `<ScriptDefaultTimeOutInSeconds>Original Number</ScriptDefaultTimeOutInSeconds>` `<ScriptTimeoutTerminationGraceInSeconds>Original Number</ScriptTimeoutTerminationGraceInSeconds>`
5. Change the timeout of these two tags as follows: `<ScriptDefaultTimeOutInSeconds>New Number</ScriptDefaultTimeOutInSeconds>` `<ScriptTimeoutTerminationGraceInSeconds>New Number</ScriptTimeoutTerminationGraceInSeconds>`
6. Save the file back to the machine.
7. Restart the services as follows:

```
systemctl restart Siemplify.Server.PythonExecution.service
systemctl restart Siemplify.Server.Indexer.service
systemctl restart Siemplify.Connectors.service
systemctl restart Siemplify.Server.service
systemctl restart Siemplify.Server.PlaybookActions.service
systemctl restart Siemplify.Server.ETL.DataProcessingEngine.service
```

Administrative FAQ

How do I change a password for a DB user?

1. Log in to the database via SSH as a postgres user.
2. Enter the following command:

```
ALTER USER <UserName> WITH PASSWORD <'Password'>
```

3. Access your server via SSH.
4. Navigate to this path: /home/simplifyadmin
5. Open this file by entering the command:

```
.bash_profile
```

6. Change the password by entering this command:

```
DB_PASSWORD= <password>
```

7. Navigate to this file: /etc/systemd/system.conf
8. Change the password as follows:

```
@sudo systemctl set-environment DB_PASSWORD=
```

```
[root@siemplify siemplifyadmin]# sudo systemctl set-environment DB_PASSWORD=Yeswecan2020!
```

9. Restart all Simplify servers.

What are Networks used for?

Answer

Networks can be found under Settings > Environments > Networks.

Here you can set the internal networks in your organization. This will help Siemplify identify internal assets and consider the sensitivity of the network when running playbooks. To set a sensitivity rate for the network, just adjust its priority (1 being the highest). Network data can also be used to trigger playbooks.

Can I upgrade Java in Siemplify?

Answer

Siemplify does **not** recommend upgrading Java in the Siemplify server without first contacting Siemplify Support. This is because there is a high possibility that the the update will break the JAVA environment path which in turn will render the integrations depending on that path non functional. For example, Elastic Search integration will not work if the JAVA path is not in the environment.

What is Siemplify's Password Policy?

Answer

Siemplify recommends using strong passwords for all user logins to protect your information. It is also required to implement mandatory password changes periodically to safeguard your data.

As such Siemplify has implemented the following policy:

- The password is set for 90 days (3 months) after which it will expire.
- Notifications are sent to the Siemplify user letting them know that their password is about to expire and they need to change it.

Notifications will be sent:

- 14 days before password expiry
 - 7 days before password expiry
 - 2 days before password expiry
 - 1 day before password expiry
-
- If the user lets their password expire or tries to log in too many times with the wrong password, they will be locked out of their account.
 - For users who have been locked out of their account, the admin can enable their account again and the password will be valid for another 90 days.

Authentication, Permissions and Access

Configure SAML Provider in Simplify

This article describes how to configure a SAML provider.

If you are using Okta – look [here](#)

If you are using G suite – read here [first](#).

If you are using Azure – read here [first](#)

Currently, the platform supports G Suite, Okta, Azure and configuring your own custom SAML provider. This can be an existing solution like Centrify, or a company specific solution. The Simplify application uses the default sts of .NET core. Simplify uses their library for the token authentication against the identity provider; only using the nameID property from the tokens.

The following steps should be taken to configure the provider:

- Configure SAML Provider
- Configure Users and invite them to Simplify

For the purposes of this article, we will use **G Suite** as an example of a custom provider.

Configure SAML Provider

To configure the SAML Provider:

1. Navigate to Settings > Advanced > External Authentication.
2. Select G Suite.
3. Fill out the following fields.

Field	Description
Provider name	Add in the name of the provider. Note that the system will automatically have G Suite and Okta populated.
IDP Metadata	The IDP Metadata is SAML metadata and is used to share configuration information between the Identity Provider (IdP) and the Service Provider (SP). Note that if you use a certificate the following value WantAuthnRequestsSigned=" " in xml should be true. If you are not using a certificate then set it to false.
Identifier	URL of the provider.
Audience URI (SP Entity ID)	Simplify server name. Can be either an IP URL, Host Name URL or Local Host URL. Note that users have to connect to Simplify with the same URL pattern configured in this field in order to log in with SAML
Provider public certificate	The certificate is optional. It can be uploaded as necessary for custom custom providers.

4. Click Save in the top right corner.

5. Restart the Siemplify server for the configuration to take place.
6. Click Test in order to make sure the connection is working as expected.

External Authentication

SAML Provider
Configure SAML identity provider for user authentication

☐ None

☐ Okta

☒ G Suite

☐ Custom SAML provider

Provider name *

IDP Metadata *

Identifier *

ACS URL *

Provider Public Certificate

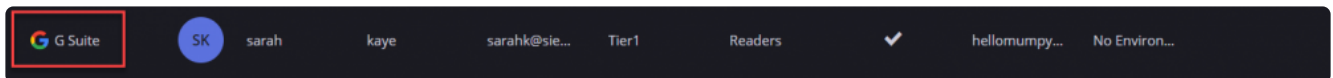
Test

Configure Users

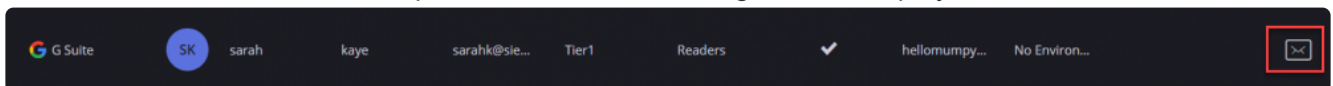
The next stage is to add users that can access Siemplify through the new SAML provider that you just created.

To add and configure users:

1. Navigate to Settings > Authentication > User Management.
2. Click the + icon on the top right.
3. Fill out the fields, making sure to choose G Suite Provider in the User Type field.
4. Click Add when done. The user will appear in the list of Users with the G Suite icon to the left.



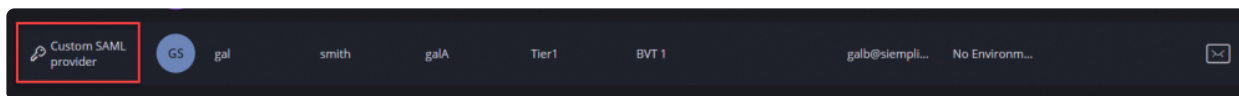
5. Repeat for any users you need.
6. Click the Send Invitation envelope to invite the user to sign into Siemplify.



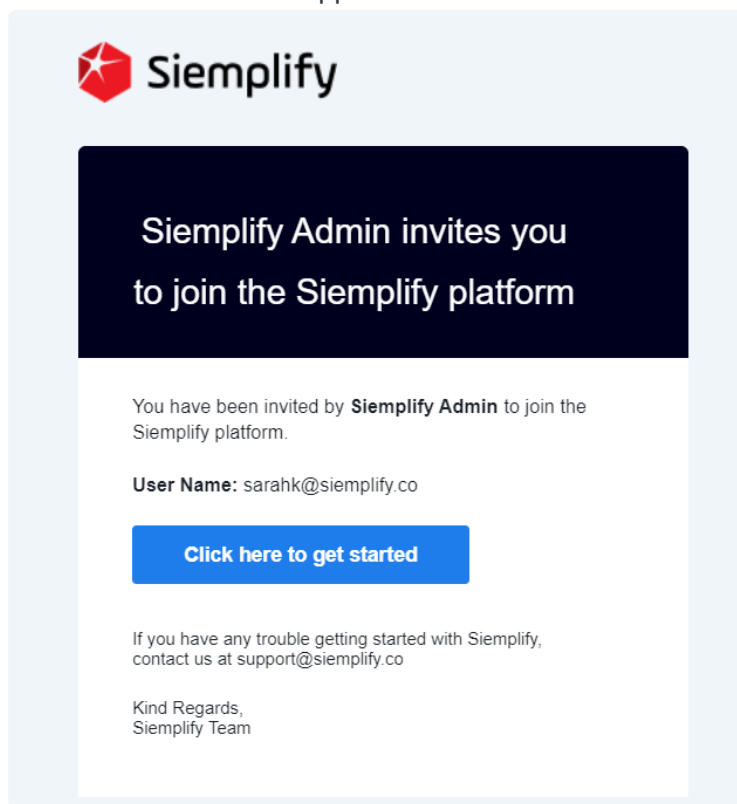
Change SAML Providers

To change SAML providers in the Siemplify platform (Admin only):

1. Disable users that are using the previous SAML Provider.
 - a. Navigate to Settings > Organization > User Management.
 - b. Select a user in the list that has the relevant SAML provider icon next to him.



- c. Click on the Edit icon and select the checkbox to Disable the user account.
- d. Repeat for all Users with custom SAML provider authentication.
2. Change SAML Provider in the Settings > Advanced > External Authentication.
3. Return to Settings > Organization > User Management.
4. Create new users, making sure to select the new SAML provider in the User Type drop-down field.
5. After creating a new user, make sure to send them an invitation to Siemplify with the Send Invitation link. The invitation will appear in the user's mailbox and will look as follows:

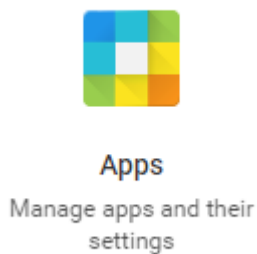


SAML Configuration for G Suite

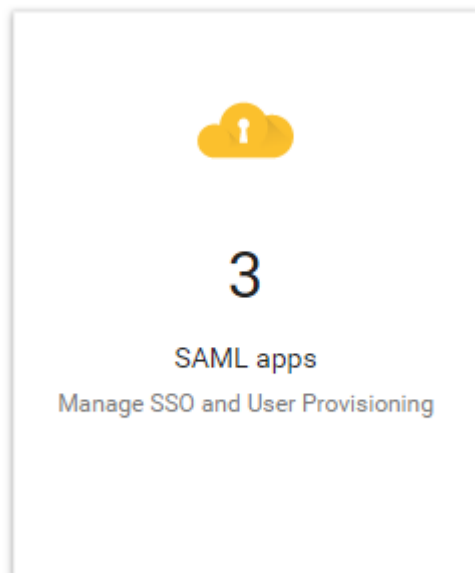
This article details both how to configure G Suite for authentication and how to configure the Simplify platform to support this.

To configure G Suite for Single Sign on (SSO):

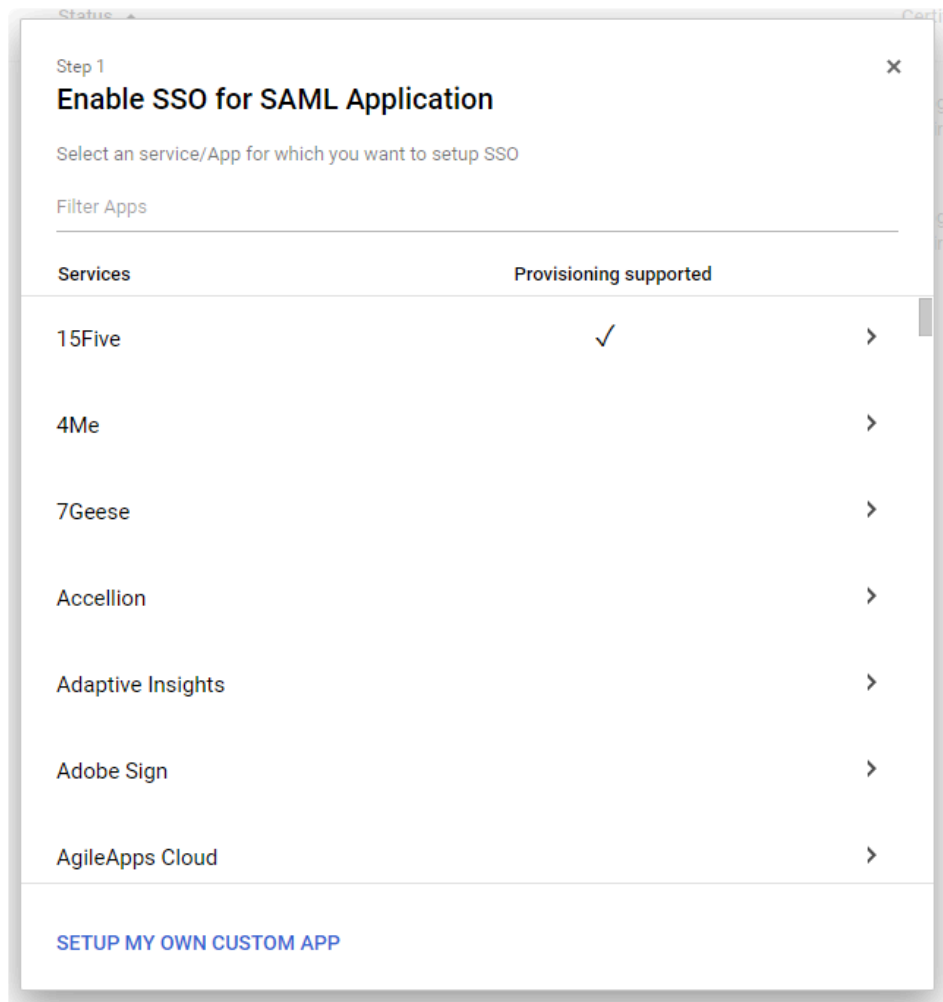
1. Navigate to the [Google Admin Portal](#).
2. Select Apps.



3. Select SAML apps.



4. Click on **Enable SSO for a SAML application**.
5. In the Enable SSO for a SAML application screen, click on **Setup my own Custom App**.



6. In the Google IdP Information screen, click Next.

Status

Certificate

Step 2 of 5

×

Google IdP Information

Choose from either option to setup Google as your identity provider. Please add details in the SSO config for the service provider. [Learn more](#)

Option 1

SSO URL

https://accounts.google.com/o/saml2/

Entity ID

https://accounts.google.com/o/saml2?i

Certificate

Google_2024-4-7-194023_SAML2.0

Expires Apr 07, 2024

↓ DOWNLOAD

----- OR -----

Option 2

IDP metadata

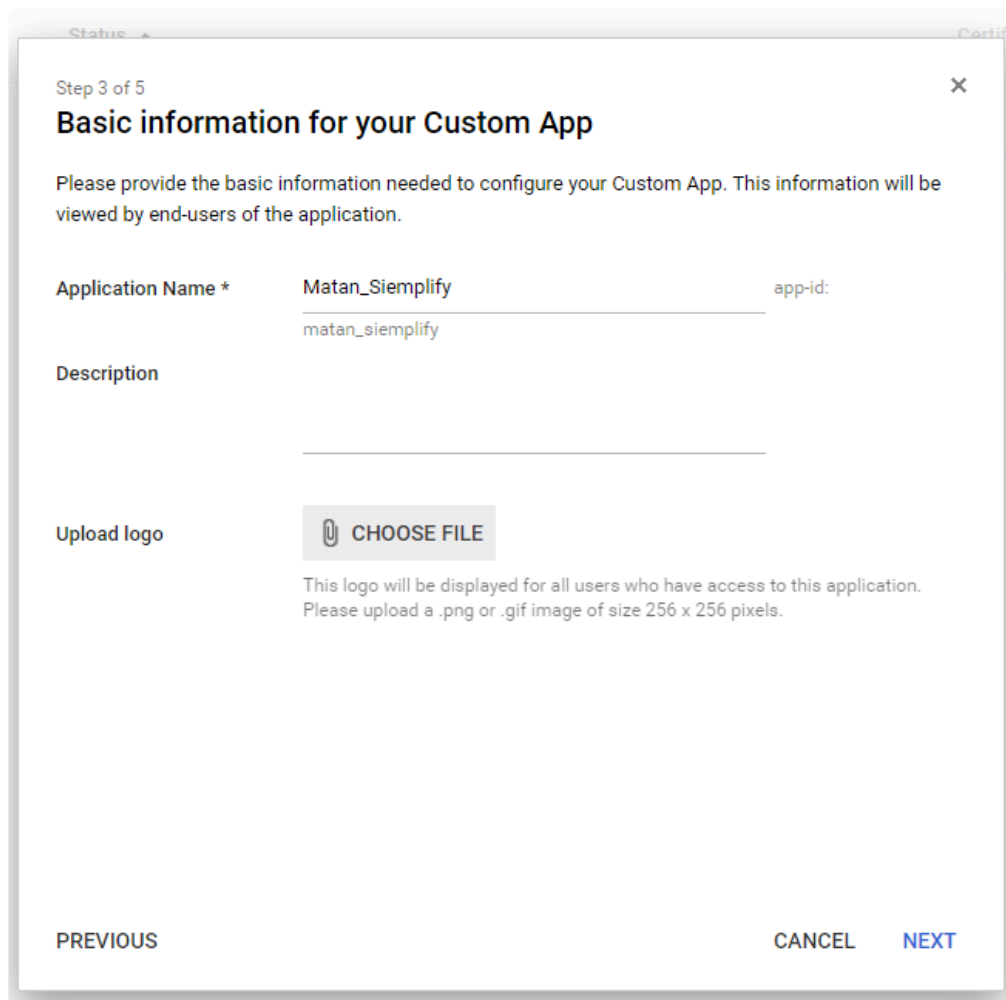
↓ DOWNLOAD

PREVIOUS

CANCEL

NEXT

7. In the basic information for your Custom App screen, enter your Application Name (user defined) and click Next.



The screenshot shows a dialog box titled "Step 3 of 5" with a close button (X) in the top right corner. The main heading is "Basic information for your Custom App". Below this, a paragraph states: "Please provide the basic information needed to configure your Custom App. This information will be viewed by end-users of the application."

The form contains three sections:

- Application Name ***: A text input field containing "Matan_Siemplify". To its right, the label "app-id:" is followed by a text input field containing "matan_siemplify".
- Description**: A text input field that is currently empty.
- Upload logo**: A button with a paperclip icon and the text "CHOOSE FILE". Below this button, a note reads: "This logo will be displayed for all users who have access to this application. Please upload a .png or .gif image of size 256 x 256 pixels."

At the bottom of the dialog, there are three buttons: "PREVIOUS" on the left, "CANCEL" in the center, and "NEXT" on the right.

8. In the Service Provider Details screen, enter the following information:
- ACS URL: `https://{your_siemplify_server_IP_address}/Saml2/Acs`
 - Entity ID: `https://{your_siemplify_server_IP_address}/Saml2`
 - Click Next.

Step 4 of 5

×

Service Provider Details

Please provide service provider details to configure SSO for your Custom App. The ACS url and Entity ID are mandatory.

ACS URL *

Entity ID *

Start URL

Signed Response

☐

Name ID

Basic Information ▼

Primary Email ▼

Name ID Format

UNSPECIFIED ▼

PREVIOUS

CANCEL

NEXT

9. In the Attribute Mapping screen, click Finish.

Step 5 of 5

×

Attribute Mapping

Provide mappings between service provider attributes to available user profile fields.

Some providers require you to map application attributes to user fields. You should check the application's documentation to see if this is required. You can always come back later to complete the mapping.

There are currently no mappings for this application

ADD NEW MAPPING

PREVIOUS

CANCEL

FINISH

✿ For customers using 5.11 and above, please continue with the procedure outlined in [Configure SAML Provider in Simplify](#).

SAML Configuration for Azure

Before getting started, please take a look at these two links below. The customer should have both of these set up in their Azure:

[How to create an non gallery app in Azure](#)

[How to configure the SAML in Azure](#)

To configure SAML on Simplify:

- Navigate to Simplify > Settings > Advanced > External Authentication.
- Select the Custom SAML Provider.
- Fill out the following.
 - Provider Name: Azure
 - IDP Metadata: Click Download next to the **Federation Metadata XML** in the SAML screen
 - Identifier: Copy what is written in the **Azure AD Identifier** field in the SAML screen
 - Audience URI (SP Entity ID): For example: `https://Simplify_Address/Saml2/`. You will add this later on to the SAML screen
 - Provider Public Certificate: download the Certificate (Base64) from Azure in the SAML screen

The screenshot shows the 'External Authentication' settings page in Simplify. On the left, there is a sidebar with options: 'None', 'Okta', 'G Suite', and 'Custom SAML provider' (which is selected). The main area is titled 'SAML Provider' and 'Configure SAML identity provider for user authentication'. It contains a 'Custom SAML provider' section with the following fields:

- Provider name ***: Azure
- IDP Metadata ***: simplify_url
- Identifier ***: https://ms.simplify.net/168P12ca-5d1-a-1700-8a0b-4b0f0a7a0a0a
- Audience URI (SP Entity ID) ***: https://SIMPPLY_ADDRESS/Saml2/
- Provider Public Certificate ***: simplify_url

Below these fields are two checkboxes: 'Auto redirect' and 'Enable Just-in-Time User Provisioning', both of which are currently unchecked. A 'Test' button is located next to the 'Provider Public Certificate' field. A 'Save' button is in the top right corner.

1

Basic SAML Configuration

Identifier (Entity ID)	https://SIEMPLIFY_ADDRESS/Saml2/
Reply URL (Assertion Consumer Service URL)	https://SIEMPLIFY_ADDRESS/Saml2/ACS
Sign on URL	Optional
Relay State	Optional
Logout Url	Optional



2

User Attributes & Claims

givenname	user.givenname
surname	user.surname
emailaddress	user.mail
name	user.userprincipalname
Unique User Identifier	user.userprincipalname

3




SAML Signing Certificate

Status	Active
Thumbprint	8AA83530B4A2CE8FD2F86ECE444E73570900A0B7
Expiration	3/18/2023, 3:51:10 PM
Notification Email	
App Federation Metadata Url	 
Certificate (Base64)	Download
Certificate (Raw)	Download
Federation Metadata XML	Download

4

Set up Siemplify

You'll need to configure the application to link with Azure AD.

Login URL	https://login.microsoftonline.com/d48f52ca-5b...	
Azure AD Identifier	https://sts.windows.net/d48f52ca-5b1a-4708-8...	
Logout URL	https://login.microsoftonline.com/common/wsf...	

[View step-by-step instructions](#)


To Configure SAML on Azure:

Navigate to Azure > Basic SAML Configuration and configure the following:


Identifier (Entity ID): https://Siemplify_Address/Saml2/ (take this from the Siemplify platform)

Reply URL (Assertion Consumer Service URL): https://Siemplify_Address/Saml2/ACS


- 1



Basic SAML Configuration 

Identifier (Entity ID)	https://SIEMPLIFY_ADDRESS/Saml2/
Reply URL (Assertion Consumer Service URL)	https://SIEMPLIFY_ADDRESS/Saml2/ACS
Sign on URL	Optional
Relay State	Optional
Logout Url	Optional
- 2

User Attributes & Claims 

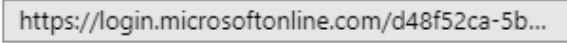

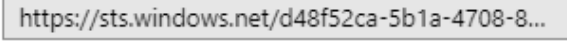

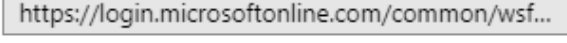

givenname	user.givenname
surname	user.surname
emailaddress	user.mail
name	user.userprincipalname
Unique User Identifier	user.userprincipalname
- 3

SAML Signing Certificate 

Status	Active
Thumbprint	8AA83530B4A2CE8FD2F86ECE444E73570900A0B7
Expiration	3/18/2023, 3:51:10 PM
Notification Email	siemplify@siemplify.com
App Federation Metadata Url	 
Certificate (Base64)	Download
Certificate (Raw)	Download
Federation Metadata XML	Download
- 4

Set up Siemplify

You'll need to configure the application to link with Azure AD.

Login URL	 
Azure AD Identifier	 
Logout URL	 

[View step-by-step instructions](#)

Using Permissions in Siemplify

Siemplify allows creating different sets of user groups and then assigning them different levels of permissions to different modules.

1. Navigate to Settings > Organizations > Permissions.
2. Click **Add Permission Group** and add a suitable name. For example, SOC Managers.
3. Select Edit/View/None for each of the modules for this new Group.

- Cases
- Connectors
- Dashboards
- Entities
- Homepage
- IDE
- Investigator
- Jobs
- Marketplace
- Ontology
- Playbooks
- Reports
- Search
- Selfbranding
- Settings
- Simulation
- Warroom

Permissions

Manage permissions and restrictions for different sets of users.

+ Add Permission Group

Admins

Basic

Readers

New Group

Soc Managers

Read/Write Permissions

Allow permissions for the following modules.

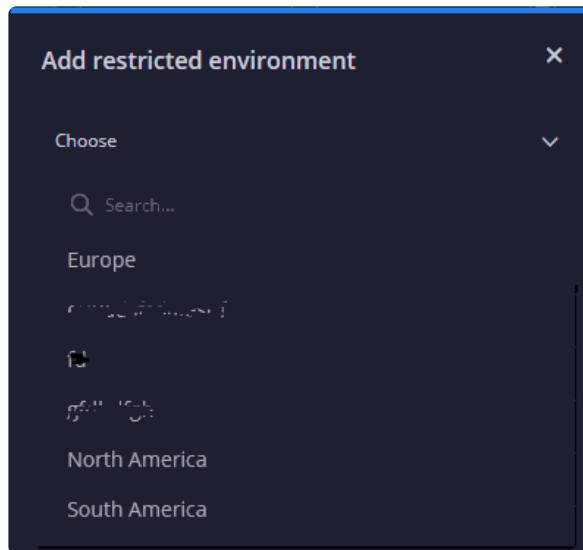
Module name	Edit	View	None
Cases	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Connectors	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dashboards	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Entities	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Homepage	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ide	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Investigator	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Jobs	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Marketplace	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Ontology	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Playbooks	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reports	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

- Make sure to click the blue Save button on the top right of the screen.

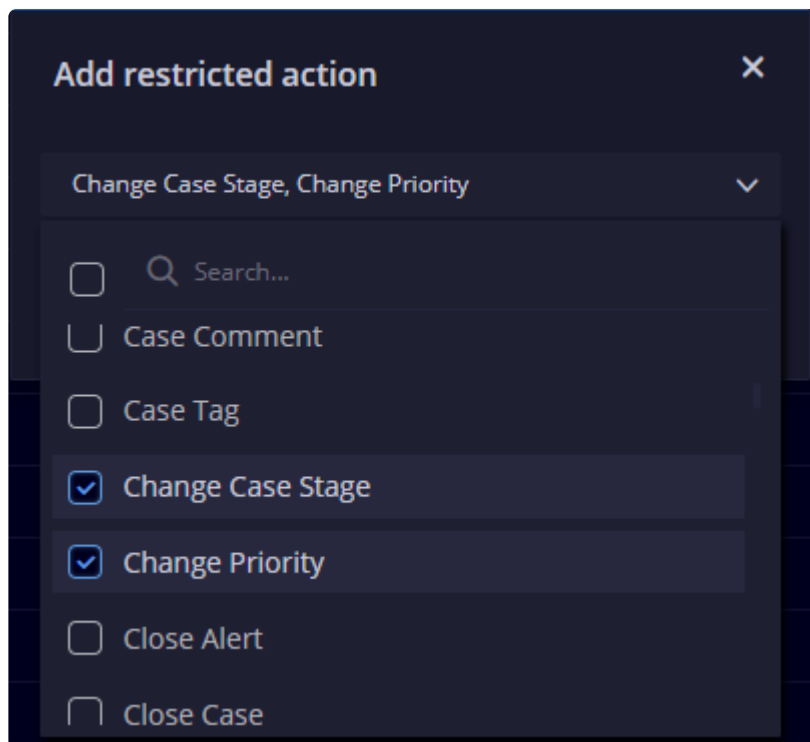
Restricting Environments and Actions

You can also use Restrictions as a way of preventing access to certain types of information from certain User Groups. For example, as an MSSP, you might have a separate SOC Manager groups for separate environments.

- Click on the required User Group.
- In the Restricted Environments section, click the plus icon and choose the Environment which this User Group should not be able to access.



3. In the Restricted Actions section, click the plus icon and choose the Action/s which this User Group should not be able to access.



How to Configure LDAP

Users can configure LDAP Authentication through the UI. Simplify allows you to configure granular permissions using Windows Active Directory and Simplify permission groups.

Configure LDAP Settings

The first stage is to configure the LDAP settings. Please note that the User you configure below (Admin DN) must have the appropriate permissions to query LDAP accounts. This user can be either an Admin or a User that has permissions to view the users who are allowed to login to Simplify. The suggested best practice is to use a service account in the built-in Windows Active Directory Group 'Account Operators'.

1. Navigate to Settings > Advanced > General.
2. In the LDAP Configuration area, fill out the following mandatory parameters:
 - **Host:** Active Directory address [AD_server_address]. Example: 10.0.0.1
 - **Port:** Note that port 389 is the default port if TLS is not checked. If TLS is checked, then use port 636
 - **Admin DN:** Admin user distinguished name
Example: CN=admin,DC=ldapserver,DC=com
 - **Admin Password:** Admin password
 - **User Base DN:** A point from which the server will search for users
OU=corporate,DC=ldapserver,DC=com
 - **User Attribute:** Attribute for the username

✿ Note that if not specified otherwise, the default attribute "sAMAccountName" will be used.


1. Optionally, you can fill out the following parameters (or keep the default values):
 - **Group Attribute:** Attribute for the group name
 - **First Name Attribute**
 - **Last Name Attribute**
 - **Email Attribute**
 - **TLS:** Relevant if LDAPS is used
 - **Trust Certificate:** Relevant if LDAPS is used and TLS is checked
- To find the following attributes, run the Powershell command: `Get-ADUser`
 - Admin DN
 - User Base DN
 - User Attribute
 - Group Attribute
 - First Name Attribute

- Email Attribute

2. Click Save when done.

LDAP Configuration

Configure the field to enable LDAP authentication

Host *	<i>i</i>	ldaphost.com	User Attribute	<i>i</i>	sAMAccountName
Port *		389	Group Attribute	<i>i</i>	memberOf
Admin DN *	<i>i</i>	cn=admin,dc=ldapserver,dc=com	First Name Attribute	<i>i</i>	givenName
Admin Password *	<i>i</i>	Admin account password	Last Name Attribute	<i>i</i>	sn
User Base DN *	<i>i</i>	dc=ldapserver,dc=com	Email Attribute	<i>i</i>	mail
<input type="checkbox"/> TLS					
<input type="checkbox"/> Trust Certificate					
		<div>Save</div>			

Configure Individual Users

Each user in the system that you want to log in with LDAP must be defined as such.

1. Navigate to Settings > Organization > User Management.
2. For each user, both new and existing, make sure to select LDAP from the User Type field drop-down.

Add User [X]

User Type: Internal (dropdown menu open showing: Internal, LDAP, Custom SAML Provider)

First Name: [Search...]

Last Name: [Custom SAML Provider]

User Name: [LDAP]

Email: []

Password: []

Confirm Password: []

License Type: Standard (dropdown menu)

SOC Role: Choose (dropdown menu)

Permission Group: Choose (dropdown menu)

Environments: Default Environment (dropdown menu)

User Account Is Disabled: ☐

[Cancel] [Add]

Configure User Authentication Groups

The next stage is to configure User Authentication groups for the specific AD group.

1. Navigate to Settings > Organization > Permissions.
2. Make sure you highlight the required Permission Group on the left (for example: Readers, Admins etc)
3. In the Active Directory Groups, make sure to add the name of the AD group that holds your users.
4. Don't forget to click Save.

Permissions
Manage permissions and restrictions for different sets of users.

+ Add Permission Group

Readers

Admins

Basic

Readers

Read/Write Permissions
Allow permissions for the following modules.

Module name	Edit	View	None
Agents	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Cases	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Connectors	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Dashboards	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Entities	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Environments	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Homepage	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Idc	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Investigator	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Jobs	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Marketplace	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Ontology	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Playbooks	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Reports	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Active Directory Groups
Specify the relevant Active Directory Groups for these permissions.

Simplify QA x


Restrict Actions
Users will not be able to perform any of the actions below.


Action Name


Save


Test the Configuration

1. Log out of the system.
2. Make sure the login screen now shows the Login with LDAP checkbox.
3. Use your Active Directory username and password to log in.
4. Make sure to select the Login with LDAP checkbox.

 **Simplify**

 10.0.0.68

 Username

 Password

☐ Login with LDAP

Sign In

Copyright © 2019 Simplify ALL RIGHTS RESERVED

Create View Only Users

✿ This article is relevant from Release 5.3.0 and up.

In addition to standard users, the admin can also add view only users to the Simplify platform. This can be useful when you want somebody in your organization or an MSSP customer to be aware of information pertaining to specific places in the platform but without having the ability to change them.

The procedure to add these users is the same as that of adding regular users.

Step One: Purchase the license for the required number of users with your Account Manager.

Step Two: Create a new View Only user group or use the predefined View Only group.

Step Three: Create new user.


Purchase View Only License

1. Arrange a license for the required number of View Only Users.
2. Navigate to Settings > Organizations > License Management to view the details.

License Management

Keep track of your system license status and anticipated expiration.

License Validity


 Your license is up to date
System license will expire in 22 days
[Paste New License](#)

System Version

Server Version	Client Version	Customer ID
5.3.0.301	5.3.0.301	2e0a364c-934c-47b8-be70-a9829f602957

System Limitations

	Limit	Usage
Standard Users	5	2
View-Only Users	1	0
Environments	1	0
Agents	1	0

View user agreement

System Modules

Module Name	Active
Playbooks	✓
Marketplace	✓
Jobs	✓
Settings	✓
Connectors	✓
Dashboards	✓
Search	✓
Cases	✓
Homepage	✓
Investigator	✓
Reports	✓
Simulation	✓

Set up a User Group

1. Navigate to Settings > Organization > Permissions.
2. Click on the predefined Permissions group called View Only. (Alternatively, you can create a new group and select the checkbox next to **This group is for View-Only License users.**)
3. Select the required categories that you want to be viewable.

Permissions

Manage permissions and restrictions for different sets of users.

+ Add Permission Group

Readers

Admins

Basic

View-Only

New Group

Dashboard View Only

☒ This group is for View Only License users

Read/Write Permissions

Allow permissions for the following modules.


Module name	Edit	View	None
Cases	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Dashboards	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Investigator	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Reports	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Search	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

- Click Save in the top right corner.

Create a View Only User

- Navigate to Settings > Organization > User Management.
- In the License Type field, make sure to select the View Only type.
- In the Permission Group drop-down, make sure to select the View Only group or any new group that you created which is for View Only users.

Add User ×

User Type	<div> Internal ▼</div>
First Name	<div>Lucy</div>
Last Name	<div>Smith</div>
User Name	<div>LSmith@mycompany.com</div>
SOC Role	<div>Tier3 ▼</div>
Permission Group	<div>Dashboard View Only ▼</div>
Email	<div>LSmith@mycompany.com</div>
Environments	<div>No Environment ▼</div>
<h3>Change Password</h3>	
New Password	<div>.....</div>
Confirm New Password	<div>.....</div>
<div><input type="checkbox"/> User Account is Disabled</div>	
<div><div>Cancel</div><div>Add</div></div>	

4. Fill out the rest of the fields as required.
5. Click Add. The user is added to the list.

Authentications and Permissions FAQ

How can I prevent users from changing playbooks?

Answer

When creating or editing users you can assign a permission group to them that will prevent them from changing playbooks.

Navigate to Settings > Organization > Permissions and click +.

The View permission will allow users to view playbooks in cases and in the case designer.

The None permission will only allow viewing (and running) playbooks that were attached to cases.

Permissions
Manage permissions and restrictions for different sets of users.

+ Add Permission Group

Readers

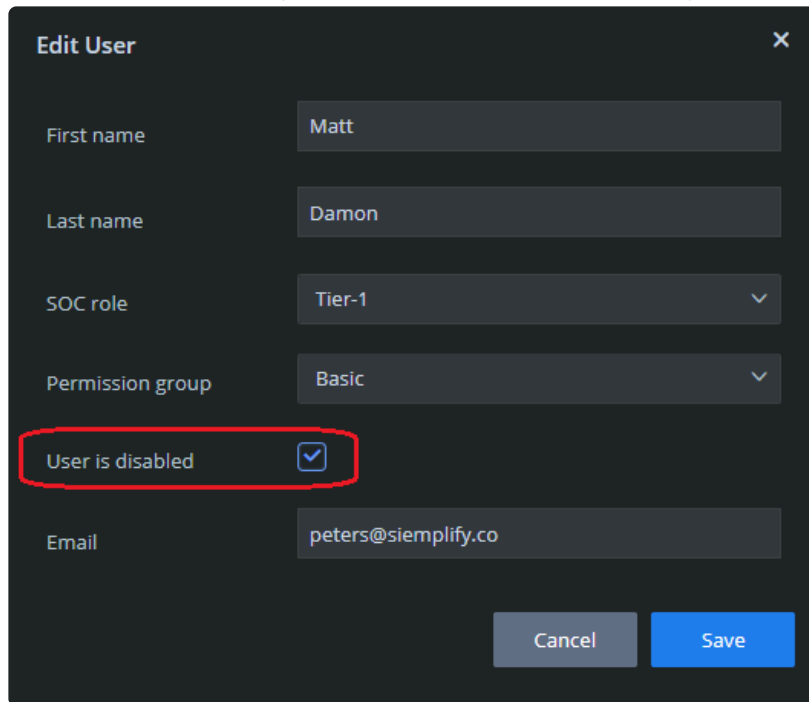
Read/Write Permissions
Allow permissions to the following modules.

Module name	Edit	View	None
Cases	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Connectors	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Dashboards	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Entities	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Environments	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Homepage	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Ide	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Investigator	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Jobs	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Marketplace	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Ontology	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Playbooks	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Reports	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Search	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Self-Reflection	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

How can I disable a user account in Simplify?

Answer

1. Navigate to Setting > Organization > User Management.
2. Select the required user, and click Edit.
3. In the Edit user dialog box, tick the checkbox that says User is disabled.



Edit User ×

First name

Last name

SOC role

Permission group

User is disabled ☒

Email

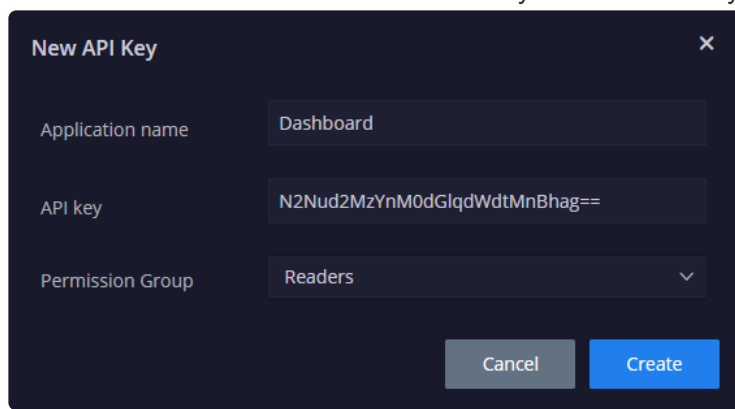
How do I generate an API key to access Siemplify's API?

Answer

You can access the Siemplify API over port 8443 of the Siemplify server. Note that from Release 5.1, you need to use port 443.

In order to be authenticated, you need to create an API key as follows:

1. In the Siemplify platform, click the wheel icon on the top right of the screen.
2. Navigate to Settings > Advanced > API Keys.
3. Click the plus icon on the top right of the screen.
4. Fill out the details. Note that the API key is automatically generated.



The screenshot shows a 'New API Key' modal window. It has a dark background. The title bar says 'New API Key' with a close button (X) on the right. There are three input fields: 'Application name' with the text 'Dashboard', 'API key' with the text 'N2Nud2MzYnM0dGlqdWdtMnBhag==', and 'Permission Group' with a dropdown menu showing 'Readers'. At the bottom right, there are two buttons: 'Cancel' and 'Create'.

5. Click Create. This API key can be edited or deleted at any time.

Daily Tasks

Using the Search page

In the Search bar, you can search using a key:phrase. For example: `AlertName:SUSPICIOUS PHISHING EMAIL`

You can also search according to Case or Entities. Switching between the two changes the list of Filters that appears below.

You can also search according to a specific time frame.

Let's look at some specific examples of searching by Cases:

- Query by **caseids:20481,20482** to return specific case data.

1-2 of 2

<input type="checkbox"/>	ID	Assigned User	Title	Time	Tags	Products				Ports	Outcomes	Status
<input type="checkbox"/>	20481	Tier-1	Phishing e...	1/31/19, 6:36 PM	Phishing Email						allowed	Open
<input type="checkbox"/>	20482	Bradyl	Virus Foun...	2/1/19, 10:33 PM	Malware Detect...					770,633	blocked,Allowed	Open

You can click on each ID to reach the Case Details screen.

- Query by **Ports:663,770** will return all the alerts that have port 80 and 443 involved.

1-18 of 18

<input type="checkbox"/>	ID	Assigned User	Title	Time	Tags	Products				Ports	Outcomes	Status
<input type="checkbox"/>	20540	Tier-1	Simulation...	2/7/19, 10:14 PM	Simulated Case,...					770	blocked	Open
<input type="checkbox"/>	20542	Tier-1	Data Exfilt...	2/7/19, 10:15 PM	Data Exfiltration					770	blocked	Open
<input type="checkbox"/>	20543	Admin	Out of wor...	2/7/19, 10:15 PM	Out of Working ...					770,633	blocked,Allowed	Open
<input type="checkbox"/>	20548	Tier-1	Virus Foun...	2/8/19, 6:11 PM	Malware Detect...					633,770	Allowed,blocked	Open
<input type="checkbox"/>	20551	Tier-1	Simulation...	2/8/19, 6:18 PM	Simulated Case,...					633	Allowed	Open
<input type="checkbox"/>	20552	Tier-1	Simulation...	2/8/19, 6:19 PM	Simulated Case,...					633	Allowed	Open

- Query by **Entity:10.210.1.13** will return all the cases with IP address 10.210.1.13 as an entity.

1-8 of 8

<input type="checkbox"/>	ID	Assigned User	Title	Time	Tags	Products				Ports	Outcomes	Status	Environment	Priority	Stage	Ticket IDs
<input type="checkbox"/>	20543	Admin	Out of wor...	2/7/19, 10:15 PM	Out of Working ...					770,633	blocked,Allowed	Open		High	Triage	928370a6-...
<input type="checkbox"/>	20548	Tier-1	Virus Foun...	2/8/19, 6:11 PM	Malware Detect...					633,770	Allowed,blocked	Open		High	Triage	8dfc6785-...
<input type="checkbox"/>	20551	Tier-1	Simulation...	2/8/19, 6:18 PM	Simulated Case,...					633	Allowed	Open		Informative	Triage	93a9d4da-...
<input type="checkbox"/>	20552	Tier-1	Simulation...	2/8/19, 6:19 PM	Simulated Case,...					633	Allowed	Open		Informative	Triage	0dcce825-...
<input type="checkbox"/>	20553	Tier-1	Simulation...	2/8/19, 6:21 PM	Simulated Case,...					633	Allowed	Open		Informative	Triage	9e240970-...
<input type="checkbox"/>	20554	Tier-1	Simulation...	2/8/19, 6:25 PM	Simulated Case,...					633	Allowed	Open		Informative	Triage	488c326d-...
<input type="checkbox"/>	20555	Tier-1	Simulation...	2/10/19, 1:13 PM	Simulated Case,...					633	Allowed	Open		Informative	Triage	5133d420-...
<input type="checkbox"/>	20556	Tier-1	Simulation...	2/10/19, 1:19 PM	Simulated Case,...					633	Allowed	Open		Informative	Triage	8858a91e-...

- Query by **AlertName:IRC Connections** will return all the cases with matching alert name.

1-11 of 11

<input type="checkbox"/>	ID	Assigned User	Title	Time	Tags	Products				Ports	Outcomes	Status
<input type="checkbox"/>	20540	Tier-1	Simulation...	2/7/19, 10:14 PM	Simulated Case,...					770	blocked	Open
<input type="checkbox"/>	20543	Admin	Out of wor...	2/7/19, 10:15 PM	Out of Working ...					770,633	blocked,Allowed	Open
<input type="checkbox"/>	20548	Tier-1	Virus Foun...	2/8/19, 6:11 PM	Malware Detect...					633,770	Allowed,blocked	Open
<input type="checkbox"/>	20557	Tier-1	Simulation...	2/10/19, 1:53 PM	Simulated Case,...					770	blocked	Open
<input type="checkbox"/>	20558	Tier-1	Simulation...	2/10/19, 1:58 PM	Simulated Case,...					770	blocked	Open
<input type="checkbox"/>	20559	Tier-1	Simulation...	2/10/19, 1:59 PM	Simulated Case,...					770	blocked	Open
<input type="checkbox"/>	20560	Tier-1	Simulation...	2/10/19, 2:01 PM	Simulated Case,...					770	blocked	Open
<input type="checkbox"/>	20561	Tier-1	Simulation...	2/10/19, 2:04 PM	Simulated Case,...					770	blocked	Open
<input type="checkbox"/>	20562	Tier-1	Simulation...	2/10/19, 2:05 PM	Simulated Case,...					770	blocked	Open
<input type="checkbox"/>	20563	Tier-1	Simulation...	2/10/19, 2:06 PM	Simulated Case,...					770	blocked	Open
<input type="checkbox"/>	20564	Tier-1	Simulation...	2/10/19, 2:47 PM	Simulated Case,...					770	blocked	Open

Let's look at some specific examples of searching by Entities:


- Search by Entities allows us free-text search. For example, a free-text search for “Simplify” returns all the entities that have the word Simplify in them.

1-50 of 60 >

Type	Risk	Location	Environment	Cases Count
ADAM@SIMPLIFY.CO	Not Suspicious	Internal		18
ADAM-WS@SIMPLIFY.LOCAL	Not Suspicious	Internal		15
TMP-WIN7-5@SIMPLIFY.CO	Not Suspicious	Internal		10
LAB@SIMPLIFY.LOCAL	Not Suspicious	Internal		13
GARRY-WS@SIMPLIFY.LOCAL	Not Suspicious	Internal		31
ITAIS@SIMPLIFY	Not Suspicious	Internal		27
VICKIE.B@SIMPLIFY.CO	Not Suspicious	Internal		28
JACKIE.L@SIMPLIFY.CO	Not Suspicious	Internal		19

The result contains the following information about the entity: Risk, Location, Environment, and Case count.

Clicking on the individual entity takes us to the Entity Details page where we can see more information.

 GARRY-WS@SIMPLIFY.LOCAL

This entity was involved in **26** cases during the last 3 months

0
Malicious Cases

ENTITY DETAILS


Q


DEFAULT


^


FIELD NAME	VALUE
Type	HOSTNAME
Environment	
IsInternalAsset	True
IsSuspicious	False
IsEnriched	False
IsVulnerable	False
IsArtifact	False


LINKED ENTITIES (7)


 10.0.0.28


 192.0.0.44

 C:\USERS\GAD\APPDATA\LOCAL\TEMP\SMONA130796192033095381194

 ITAIS@SIMPLIFY

 10.0.0.150

 10.1.210.13

 10.1.0.28

LAST CASES (26)

All

Virus Found Or security risk found

2/2/19, 1:16 AM [ID 20487](#)

Virus Found Or security risk found

2/1/19, 10:33 PM [ID 20482](#)

Out of working hours

1/31/19, 5:32 PM [ID 20450](#)

IRC Connections

1/31/19, 4:39 PM [ID 20445](#)

symantec:cep:risk:file


1/30/19, 6:33 PM [ID 20438](#)

IPS_Product


1/30/19, 10:10 AM [ID 20435](#)

CASE DISTRIBUTION


Product

 symantec:cep:risk:file


44

 IPS_Product


40

 Windows:OutOfWorkingHours

40

 Windows:FailedLogin

6

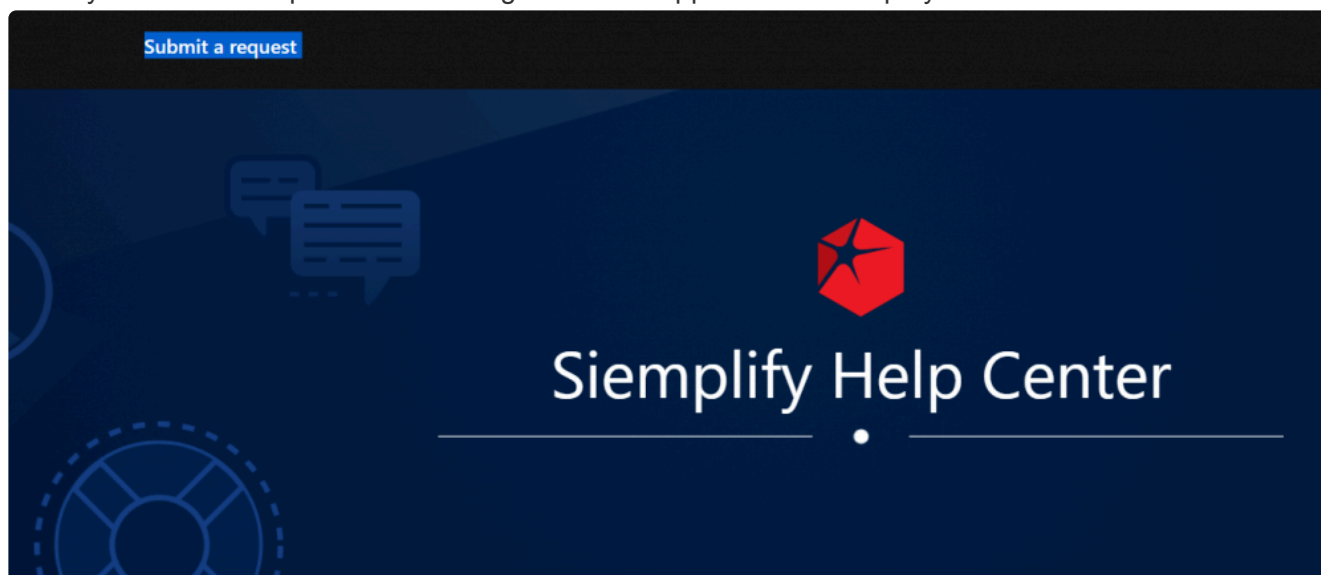
 DLP_Product

6

You can also use the Filters that appear on the left pane of the Search page to further refine your Search results.

Open a ticket for Siemplify Support

1. Navigate to [Siemplify Support Center](#).
2. Enter your name and password and sign in. The Support Center displays.



3. Click **Submit a Request** on the top left and then select the relevant form from the drop-down list in the main screen.

[Siemplify Support Center](#) > Submit a request

Submit a request

Please choose the relevant request form

- Question
- Product Issue - Integrations \ Connectors
- Product Issue - Siemplify Platform
- New Product Request - Integrations \ Connectors
- New Product Request - Siemplify Platform

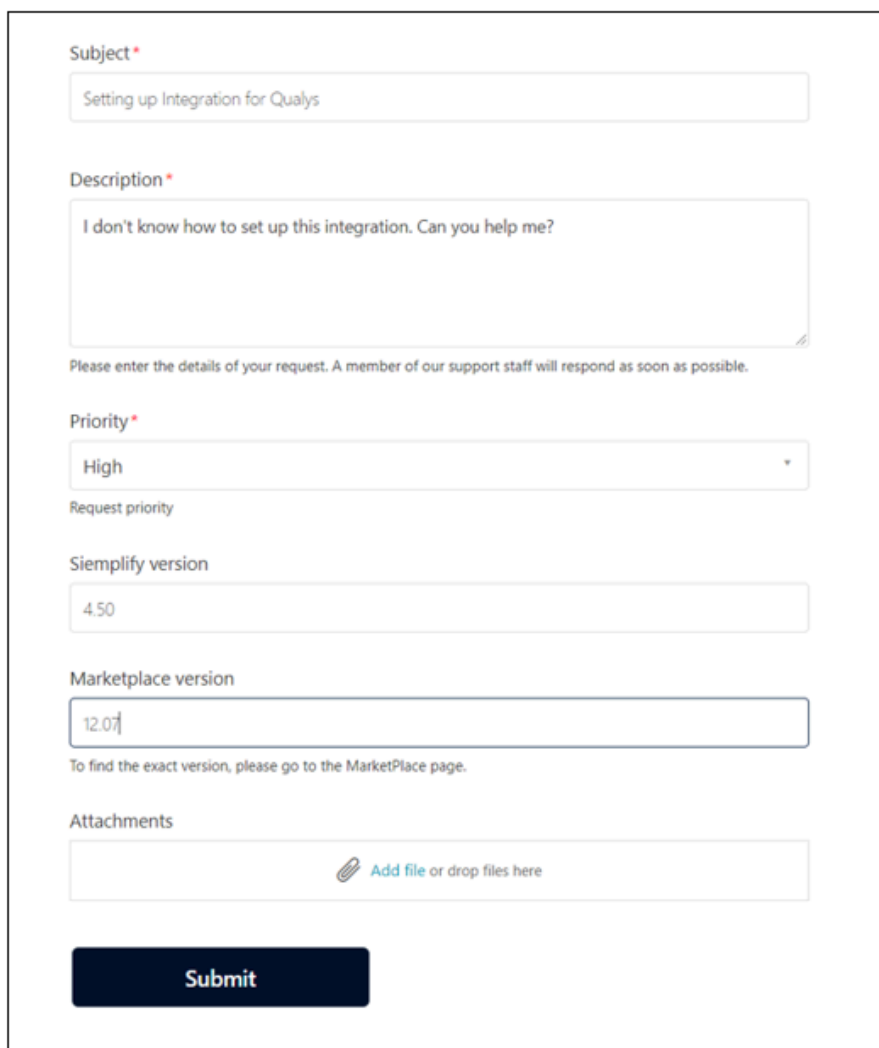
The following forms can be chosen:

- [Question](#)
- [Product Issue – Integrations / Connectors](#)
- [Product Issue – Siemplify Platform](#)
- [New Product Request – Integrations / Connectors](#)

- [New Product Request – Siemplify Platform](#)

Question

1. Select the Question option to display the Question form.
2. Fill out the following fields with as much information as you can:
 - Subject
 - Description
 - Priority
 - Siemplify version
 - Marketplace version (this is displayed in the top right of the Marketplace screen in the Siemplify platform)
 - Add any attachments that can help us with your query.
 - Click Submit when finished. The screenshot below is for illustrative purposes only.



The screenshot shows a web form for submitting a question. It includes the following fields and elements:

- Subject ***: A text input field containing "Setting up Integration for Qualys".
- Description ***: A larger text area containing "I don't know how to set up this integration. Can you help me?". Below this field is a note: "Please enter the details of your request. A member of our support staff will respond as soon as possible."
- Priority ***: A dropdown menu currently set to "High". Below it is the label "Request priority".
- Siemplify version**: A text input field containing "4.50".
- Marketplace version**: A text input field containing "12.07". Below it is a note: "To find the exact version, please go to the MarketPlace page."
- Attachments**: A section with a file upload icon and the text "Add file or drop files here".
- Submit**: A dark blue button at the bottom of the form.

Product Issue – Integrations / Connectors

1. Select the Product Issue – Integrations / Connectors option to display the form.
2. Fill out the following fields with as much information as you can:

- Subject
- Description
- Priority
- Siemplify version
- Marketplace version (this is displayed in the top right of the Marketplace screen in the Siemplify platform)
- Python Exception Error – Copy this message you see while running the Action either manually or through a playbook.

For example, a failed action in a playbook will show the error in the Context Details pane under Action Failed.

The screenshot shows the Siemplify platform interface. The top navigation bar includes a user profile (Tier-1), a case ID (DLP_Product ID 10197), and a status (Stages: Triage). The main workspace is divided into three sections: Alerts (1), Insights (0), and Playbooks (1). The Alerts section shows a 'DATA EXFILTRATION' alert from 4/29/19. The Playbooks section shows a 'TestSuraj' playbook. The right-hand 'Context Details' pane displays the 'Action Failed' status for the 'CiscoAMP_Get File List Items' script, with a detailed Python traceback error message.

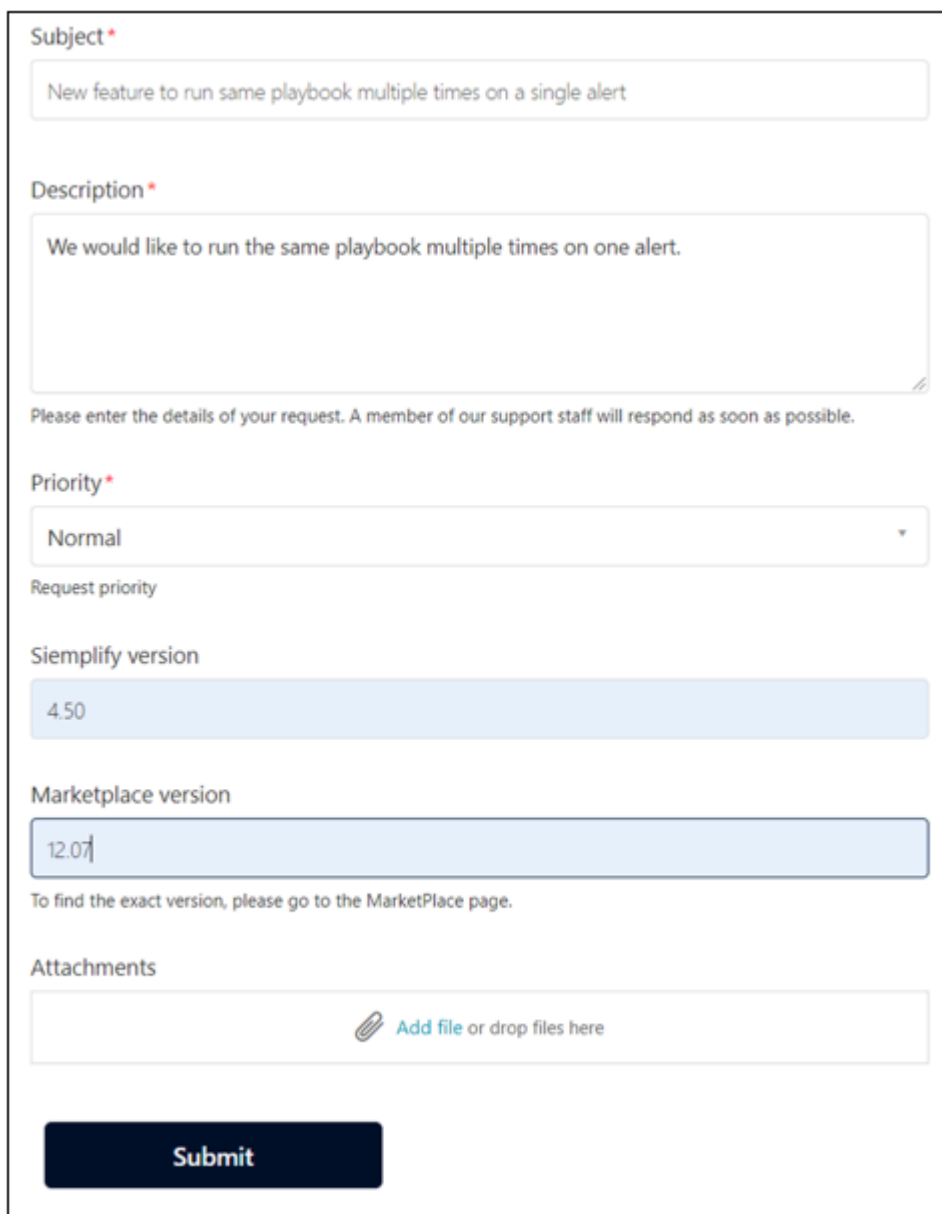
Python Exception Error

```
Execution Failed: Scripts - CiscoAMP_Get File List Items. Message: Error to Run Script
CiscoAMP_Get File List Items , script output Script did not return expected data. Did you
call build_result/end_script?
Check DebugOutput for details:Traceback (most recent call last):
File
File
```

Please copy the python exception error from the platform

* NOTE: the error log for connectors can be retrieved from the following directory in Siemplify: C:\Siemplify_Server\Scripting\SiemplifyConnectorExecution\{Integration Name}\logdata.log

1. Add any attachments that can help us with your product issue.
2. Click Submit when finished. The screenshot below is for illustrative purposes only.



The screenshot shows a web form for submitting a product issue. The form is titled "Subject *" and "Description *". It includes a "Priority *" dropdown menu set to "Normal", a "Request priority" label, a "Siemplify version" field with the value "4.50", a "Marketplace version" field with the value "12.07", and an "Attachments" section with a file upload button. A "Submit" button is at the bottom.

Subject *

New feature to run same playbook multiple times on a single alert

Description *

We would like to run the same playbook multiple times on one alert.

Please enter the details of your request. A member of our support staff will respond as soon as possible.

Priority *

Normal

Request priority

Siemplify version

4.50

Marketplace version

12.07

To find the exact version, please go to the MarketPlace page.

Attachments

Add file or drop files here

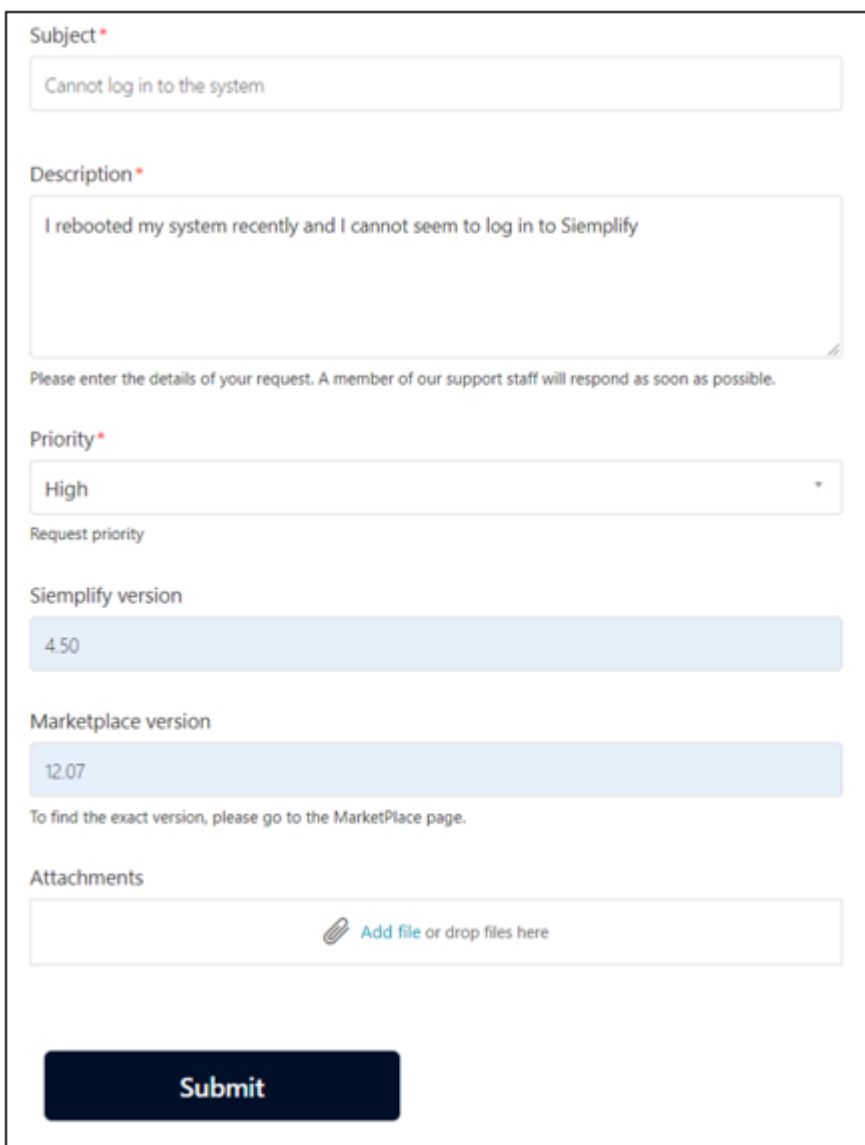
Submit

Product Issue – Siemplify Platform

1. Select the Product Issue – Siemplify Platform option to display the form.
2. Fill out the following fields with as much information as you can:

- Subject

- Description
- Priority
- Simplify version
- Marketplace version (this is displayed in the top right of the Marketplace screen in the Simplify platform)
- Add any attachments that can help us with your query.
- Click Submit when finished. The screenshot below is for illustrative purposes only.



The screenshot shows a web form for submitting a new product request. The form is titled 'Subject *' and contains a text input field with the placeholder text 'Cannot log in to the system'. Below this is a 'Description *' section with a text area containing the text 'I rebooted my system recently and I cannot seem to log in to Simplify'. A note below the description reads: 'Please enter the details of your request. A member of our support staff will respond as soon as possible.' The 'Priority *' section is a dropdown menu with 'High' selected. Below this is a 'Simplify version' text input field with '4.50' entered. The 'Marketplace version' section is a text input field with '12.07' entered. A note below this reads: 'To find the exact version, please go to the MarketPlace page.' The 'Attachments' section is a text input field with a paperclip icon and the text 'Add file or drop files here'. At the bottom of the form is a dark blue 'Submit' button.

Subject *

Cannot log in to the system

Description *

I rebooted my system recently and I cannot seem to log in to Simplify

Please enter the details of your request. A member of our support staff will respond as soon as possible.

Priority *

High

Request priority

Simplify version

4.50

Marketplace version

12.07

To find the exact version, please go to the MarketPlace page.

Attachments

Add file or drop files here

Submit

New Product Request – Integrations / Connectors

1. Select the New Product Request – Integrations / Connectors option to display the form.
2. Fill out the following fields with as much information as you can:

- Subject
- Description

- Priority
- Simplify version
- Integration name (if this is a brand new integration then select **Other**)
- API Documentation Link
- Marketplace version (this is displayed in the top right of the Marketplace screen in the Simplify platform)
- Add any attachments that can help us with your query.
- Click Submit when finished. The screenshot below is for illustrative purposes only.

Subject *

Integration for new product

Description *

The new product is a SIEM product which is one of our major systems used in SOC. We need to integrate the product with Simplify so we can ingest all the alerts in Simplify platform.

Please enter the details of your request. A member of our support staff will respond as soon as possible.

Priority *

Normal

Request priority

Integration name *

Other

Please write the Integration name

API Documentation Link

<https://new-product-documentation.com/api/docs>

Please add here the link for the API documentation of your new integration request

Deployment Model

Cloud

Marketplace version


12.07

To find the exact version, please go to the MarketPlace page.

Simplify version

5.0

Attachments

 Add file or drop files here

Submit

New Product Request – Simplify Platform

1. Fill out the following fields with as much information as you can:

- Subject
- Description
- Priority
- Simplify version
- Marketplace version (this is displayed in the top right of the Marketplace screen in the Simplify platform)
- Add any attachments that can help us with your query.
- Click Submit when finished. The screenshot below is for illustrative purposes only.

Subject *

Description *

We would like to run the same playbook multiple times on one alert.

Please enter the details of your request. A member of our support staff will respond as soon as possible.

Priority *

Normal


Request priority

Simplify version

Marketplace version

To find the exact version, please go to the MarketPlace page.

Attachments

 [Add file](#) or drop files here

Submit

Notifications

The screenshot displays the Siemplify user interface. The top navigation bar includes links for HOMEPAGE, DASHBOARDS, CASES, PLAYBOOKS, SEARCH, REPORTS, and WAR ROOM. The main content area is divided into several sections:

- Left Sidebar:** Lists 8 cases, including "DLP_Product" and "Phishing email detector", each with associated icons and status indicators.
- Case Overview:** Shows details for "DLP_Product" (ID 3), including a timeline of events and a list of alerts (16).
- Alerts (16):** A grid of alerts categorized by type: DATA EXFILTRATION, IRC CONNECTIONS, VIRUS FOUND OR SECURI..., OUT OF WORKING HOURS, and DATA EXFILTRATION. Each alert has a timestamp and a status icon.
- Insight (0):** A section for insights, currently showing "No Insights".
- Playbooks (0):** A section for playbooks, currently showing "Select an alert to present relevant playbooks".

On the right side, a **Notifications** panel is open, displaying:

- User Notifications:** A list of notifications from "Siemplify Admin" regarding case assignments and mentions.
- System Notifications:** A list of system alerts, including CPU utilization and component job errors.
- Device Product:** A section showing device status for "Windows..." and "IPS Product" with associated counts.

Daily Tasks FAQ

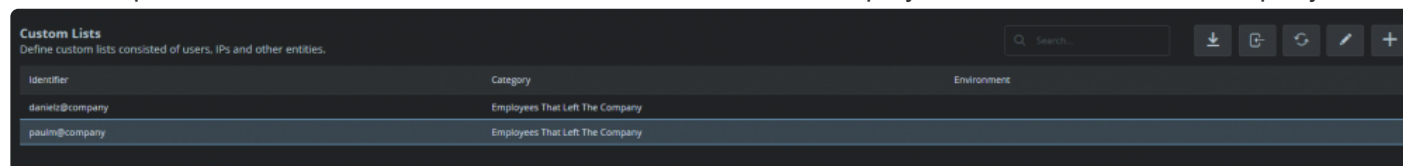
Does Simplify have custom lists?

Answer

Navigate to Settings > Environments to find the Custom Lists screen.

Custom lists are used to keep user names, asset names, IP addresses, artifact names and other items that might be referred from playbooks or incident response processes. Playbooks can be triggered by custom list items, and even manage the lists via automation.

For example, we can create a custom list to hold the names of employees who have left the company.



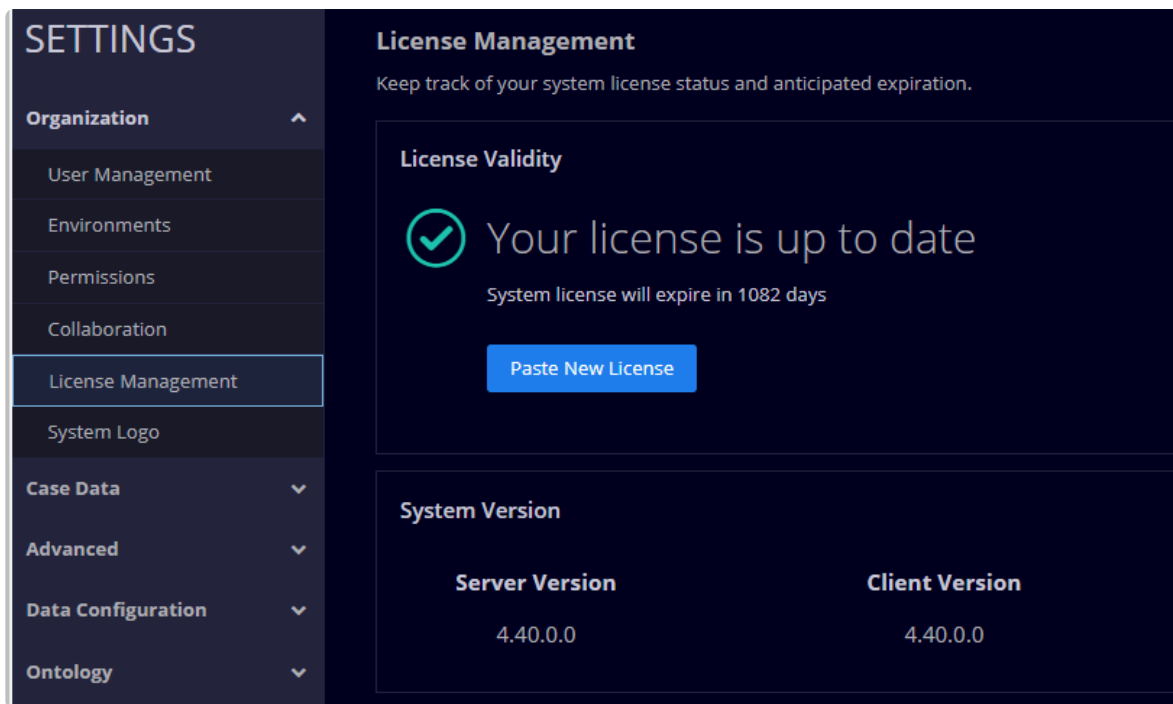
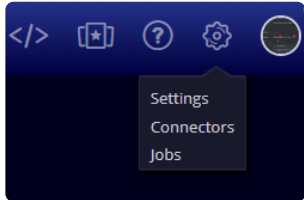
Identifier	Category	Environment
daniel@company	Employees That Left The Company	
paulm@company	Employees That Left The Company	

We can then use various Simplify actions in the Playbook such as Add to Custom List, Is in Custom List, Remove from Custom List to populate the Custom List as required.

Where can I see the Simplify Version number?

Answer

In the Simplify platform, navigate to Settings > Organization > License Management.



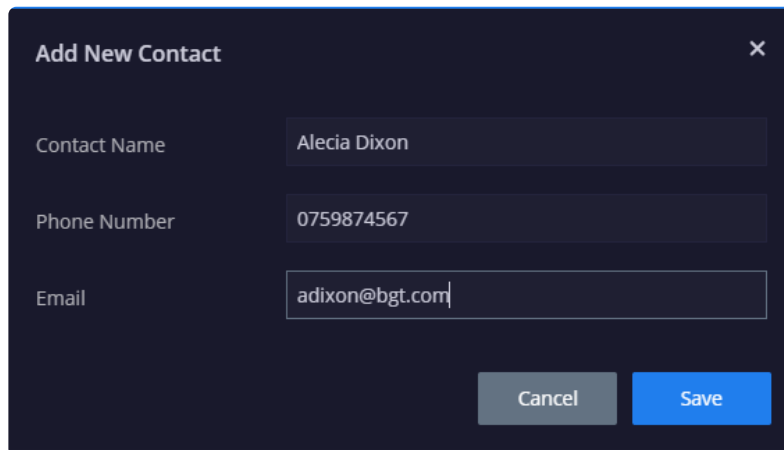
In this screen, you will see both the Server Version and the Client Version.

How can I manage contacts in Siemplify?

Answer

Siemplify allows you to add contacts and store them on the platform. This feature is accessible on the homepage.

1. Navigate to the Homepage in the Siemplify Platform.
2. In the My Contacts column on the right, click the plus icon.
3. Add in the required information.

A screenshot of a dark-themed modal window titled "Add New Contact" with a close button (X) in the top right corner. The form contains three input fields: "Contact Name" with the value "Alecia Dixon", "Phone Number" with the value "0759874567", and "Email" with the value "adixon@bgt.com". At the bottom right of the modal are two buttons: a grey "Cancel" button and a blue "Save" button.

Contact Name	Alecia Dixon
Phone Number	0759874567
Email	adixon@bgt.com

Cancel Save

4. Click Save.
You can also edit or delete contacts from the same screen.

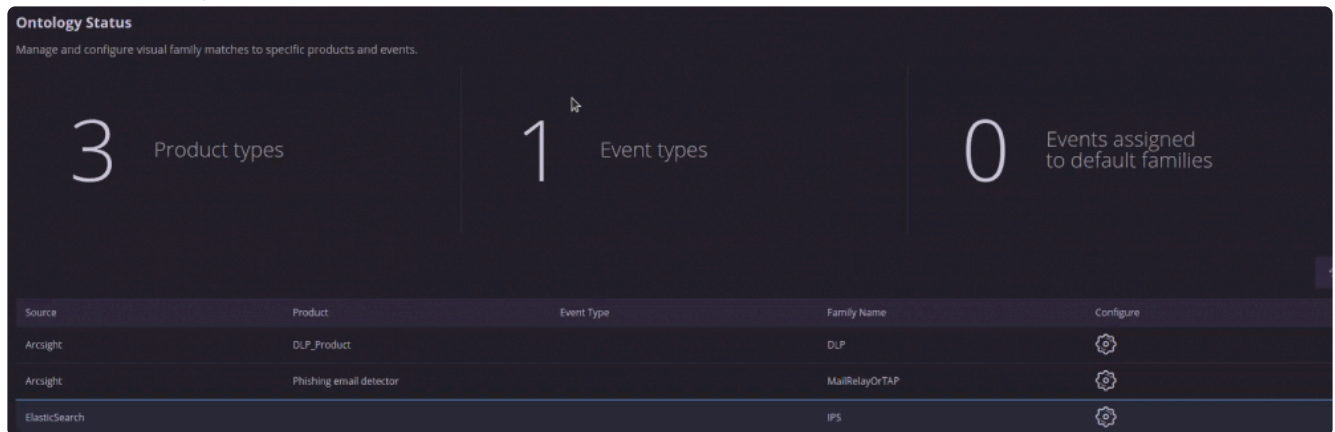
Connectors

ElasticSearch Connector: Mapping Custom DateTime

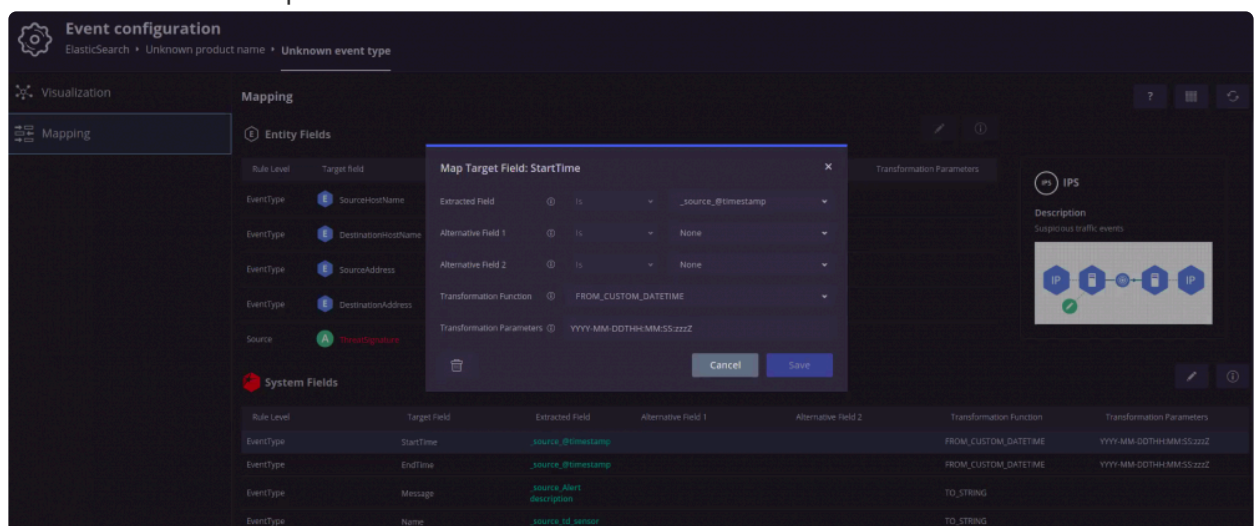
After installing and configuring an integration, you need to map their fields to Simplify fields in order to show the information in the Simplify platform.

When configuring the ElasticSearch Connector, we need to “convert” or map the custom date time such as `_source_@timestamp` field to `startTime` and `endTime` of Simplify cases.

1. Navigate to Settings > Ontology > Ontology Status.
2. Click the Configure icon in the same row as the ElasticSearch connector.



3. In the Event Configuration page, select the Mapping screen on the left hand side.
4. In the System Fields, select the Start Time row and click the Edit icon. The Map Target Field: StartTime screen opens.
5. In the Map Target Field: Start Time area:
 - a. Select the extracted Field to be the `_source_@timestamp` which is from EK stack.
 - b. In the Transformation Function field use `FROM_CUSTOM_DATETIME`
 - c. In the transformation parameters use `YYYY-MM-DDTHH:MM:SS:zzzZ`.



6. In the Map Target Field: End Time area:
 - a. Select the extracted Field to be the `_source_@timestamp` which is from EK stack.
 - b. In the Transformation Function field use `FROM_CUSTOM_DATETIME`

- c. In the transformation parameters use YYYY-MM-DDTHH:MM:SS:zzzZ. This is to generalize the time format.

7. Click Save.

The ElasticSearch timestamp fields are now converted to the Siemplify standardized time and date fields as can be seen in the screenshot below.



Defining Environments in Connectors

Note that there is no defined standard for defining environments as different Connectors have different configuration processes.

However, in general, the analyst needs to define the Environment Field Name in the Simplify platform in order for the connector to extract the environment name from the specified field.

Some Connectors environment input is a predefined field, so the the free text box of “Environment Field Name” is replaced with a checkbox. ie: McAfee ESM Connector’s “Use McAfee Enviroments”

The screenshot displays the configuration window for the 'Email Connector'. At the top, there's a header with a toggle for 'Email Connector', a 'Description' field, and a 'Definition name' set to 'Email Connector'. Below this are three tabs: 'Parameters' (selected), 'Testing', and 'Logs'. The 'Parameters' tab contains the following settings:

- Default Environment:** A dropdown menu currently showing 'Choose'.
- Run Every:** A frequency selector with four input fields: Days (0), Hours (0), Minutes (0), and Seconds (10).
- Product Field Name:** A text field containing 'device_product'.
- Event Field Name:** A text field containing 'event_name'.
- Script Timeout (Seconds) *:** A text field containing '60'.
- Sender's address *:** A text field containing 'testemail@testemail.com'.
- IMAP Server Address *:** A text field containing '192.168.0.1'.
- IMAP Port *:** A text field containing '143'.
- Username *:** A text field containing 'admin'.
- Password *:** A password field with masked characters '*****'.
- Server Time Zone:** A text field containing 'UTC'.
- Environment Field Name:** An empty text field.
- Environment Regex Pattern:** An empty text field.
- IMAP USE SSL:** A checkbox that is checked.
- Unread Emails Only:** A checkbox that is checked.
- Mark Emails as Read:** A checkbox that is checked.
- Attach Original EML:** An unchecked checkbox.

In certain connectors, in addition to defining the field name, you can also manipulate the field data using a Regex pattern to extract the specific string.

For example, Email Connector:

If the extracted environment name and the Simplify environment name do not match, you need to map the alias correctly in the map.JSON file.

This is located in the = C:\Simplify_Server\Scripting\SimplifyConnectorExecution\map.JSON


```
{  
  "Original environment name": "Desired environment name",  
  "Env1": "MyEnv1"  
}
```

If after the entire process, the connector has no environment \ empty environment (""), the default will override the empty result.

Creating a Custom Connector

The Simplify SDK allows you to easily build custom integrations for third party tools that may not exist in the marketplace. Connectors are used to ingest data from data sources that usually, but not always, have some sort of alert queue. Cases, Alerts, and Events are created in Simplify via the Connector's interaction with the Simplify application. A Connector will ingest base Event data, assign that data to an Alert and then send the Alert to the Simplify application and its Data Processing Pipeline.

This How To assumes you have a good understanding of Python and object oriented programming.

Instructions

To build a connector it's necessary to:

1. Build a Manager for the integration that contains the actual API logic for the third party tool.
2. Build the Connector utilizing the Simplify IDE.

Creating the Manager

The first step in creating the Connector is to actually build the Manager that will contain all of the API logic for the technology you are trying to integrate with. In this tutorial we will build a Connector for Netskope. The Netskope integration already exists in Simplify and the Manager is part of that integration. In this case the Manager has all the logic we already need to interact with the Netskope API.

Creating the Connector

In the Simplify IDE, create a new connector by following the instructions in [Building a Custom Integration](#). The IDE will populate with a generic template that explains the basic requirements for a Connector. This is a great starting point and provides some useful details in the code comments.

Connector logic varies but the basic steps can be broken down into the following:

Retrieve a list of alerts/detections/alerts/offenses from the third party tool. In this case, the Netskope Manager provides this capability with the `get_alerts()` method. Most tools with a queue of alerts have some way of querying alerts by time. Sending the query with the proper time fields ensures that alerts are not retrieved more than once and allows the Connector to operate in sequential order.

1. Build a Simplify Alert by instantiating a variable to the `AlertInfo` (formerly called `CaseInfo`) class and ensuring that the mandatory properties are assigned valid values.
2. Retrieve the Event data for each alert, flatten it to prevent issues with nested lists and dictionaries, and append it to the Simplify Alert as a Python list.
3. Sort the alerts by time and retrieve the latest timestamp to save it in order to send it in the next query

to retrieve the alerts. Of course if no alerts are found in this iteration of the Connector running, then use the last timestamp.

✿ Keep in mind that an Alert can have more than one Event! This is especially true for data sources such as SIEMs which use correlating logic to bundle events into an Alarm (such as McAfee ESM) or an Offense (i.e. QRadar). However other data sources such as EDR solutions don't typically perform this type of correlation so an Alert will only have one Event. The key takeaway here is to understand the data source that the Connector will be interacting with and allowing for the possibility that further logic will be needed to retrieve the base Events of a retrieved Alert.

Imports and the Simplify SDK

```
1 from SimplifyConnectors import SimplifyConnectorExecution
2 from SimplifyConnectorsDataModel import AlertInfo as SimplifyAlertInfo
3 from SimplifyUtils import output_handler, unix_now, dict_to_flat
4 from NetskopeManager import NetskopeManager
5
6 import uuid
7 import sys
```

Every connector will import the `SimplifyConnectorExecution` class from `SimplifyConnectors`. An object of this class will be instantiated, usually in the `main()` function of the Connector script. The Connector script will end when this object passes a list of Alerts to the Simplify application using the object's `return_package()` method.

Every connector will import `AlertInfo` class from `SimplifyConnectorsDataModel`. Instantiating an object of this class will actually create the Alert. In this case we it's renamed to `SimplifyAlertInfo` to avoid confusion with Netskope alerts; this is completely optional.

`SimplifyUtils` is a very useful module that contains some frequently used methods for handling logging, data formats, time, and a few other things. Always import `output_handler` and `dict_to_flat`. We'll also import `unix_now` because it's necessary for this Connector's time logic.

Connectors will almost always import the integration Manager. Keep in mind some integrations may have more than one Manager. A couple of standard libraries are also imported for this Connector.

✿ In older Connector code, you may see `CaseInfo` imported from `SimplifyConnectors` instead of `AlertInfo`. This is the same class with a deprecated naming convention.

Constant Variables

It's a good idea, although not mandatory to declare a few constant variables for use later. More on these later.

```
9 CONNECTOR_NAME := "Netskope Connector"
10 VENDOR := "Netskope"
11 PRODUCT := "Netskope"
12
13 SEVERITY_MAP := {"unknown": -1,
14                 "low": 40,
15                 "medium": 60,
16                 "high": 80,
17                 "critical": 100}
```

Creating the Simplify Alert

The Simplify Alert is created by instantiating an object of the `AlertInfo` class (as discussed previously it's renamed to `SimplifyAlertInfo()` in this Connector). The `alert_info` object has several properties that must be set in order for the Simplify application to process the Alert correctly. Here the `build_alert_info` function receives a Netskope alert (raw JSON that is received from the API) and the `Simplify` object (to be discussed below) as inputs and then parses the Netskope alert and sets the `alert_info` properties to the relevant values. This function also utilizes the constants set earlier. All of these object properties will become part of the Alert properties within the Simplify application.

Line 35 of the below screenshot is perhaps the most important line of code in the entire Connector. Utilizing the `dict_to_flat` method that was imported earlier, the alert is flattened and then appended to the `events` property, which is actually a list of the base events per alert. Remember that an alert can have more than one event, so additional logic must be added if that is the case. Here there is only one event per alert so this is sufficient. `dict_to_flat` is utilized to flatten the JSON due to nested lists and dictionaries within the JSON. The raw key and value fields are transformed into a flattened version that is slightly modified.

Going over the logic below:

`display_id` is set to a random value generated by the `uuid` library that was imported. Netskope alerts do not have a UUID field that can be used for this purpose, but if one existed it could be used.

`ticket_id` is simply set to `display_id`. `ticket_id` and `display_id` MUST BE UNIQUE in the system per alert; that's why a random value is generated with `uuid`.

`name` is the Alert Name and will be displayed in the GUI.

`rule_generator` is a field for the Rule that creates the alert in the original system. This field is not always present in the raw data and can be set to anything, but it must be set to something.

`start_time` and `end_time` are for timestamps from the alert. In this case the Netskope timestamp is in epoch time and it's multiplied by 1000 to convert it to millis time which is what Siemplify expects. If the timestamp is another format, conversion is necessary here. See the `SiemplifyUtils.py` module for some helpful time conversion methods.

`priority`: the Siemplify Application assigns a Priority to every Case based on the Alert priority in the case. The Siemplify API will map a numerical value to the displayed priority based on the following: {"Informative": -1, "Low": 40, "Medium": 60, "High": 80, "Critical": 100} where the integer value is what's passed in Connector. So for example, passing a value of 100 to the application will result in the Alert being prioritized as Critical in the GUI. In this Connector, there is an additional helper function that utilizes the `SEVERITY_MAP` constant to try and map the Siemplify priority to the severity field in the original alert. Unfortunately, the severity field is not consistent in the Netskope alerts and it requires some additional logic to check multiple fields.

`device_vendor` and `device_product` are set to the constants defined earlier.

`environment` is extremely important if environments are defined in Siemplify. Here the property is being set to a property of the `siemplify` object which is going to utilize the set environment for the Connector from the Siemplify Application.

Finally in line 37 we are modifying the base event for this alert by adding an additional key to the event (which is really a Python dictionary at this point). `product_name` is set to "Netskope" because there is not a consistent field in the raw data to set a Product that can be utilized for mapping and modeling in the Siemplify Ontology.

```

19 def build_alert_info(alert, siemplify):
20     """
21     Returns an alert, which is an aggregation of basic events. (ie: Arcsight's correlation, QRadar's Offense)
22     """
23     alert_info = SiemplifyAlertInfo()
24     alert_info.display_id = str(uuid.uuid4()) # May be Alert ID
25     alert_info.ticket_id = alert_info.display_id # In default, ticket_id = display_id. But, if for some reason the external alert id, is different
26     alert_info.name = alert.get('alert_name', 'Netskope Alert')
27     alert_info.rule_generator = "Netskope-{}".format(alert['type'])
28     alert_info.start_time = alert['timestamp'] * 1000
29     alert_info.end_time = alert['timestamp'] * 1000
30     alert_info.priority = map_priority(alert)
31     alert_info.device_vendor = VENDOR
32     alert_info.device_product = PRODUCT
33     alert_info.environment = siemplify.context.connector_info.environment
34
35     alert_info.events.append(dict_to_flat(alert))
36     for event in alert_info.events:
37         event['product_name'] = "Netskope"
38     return alert_info
39
40 def map_priority(alert):
41     """
42     Maps the Netskope's alert's severity field to Siemplify priority. The Netskope severity field varies by alert type, so this
43     may need some additional values
44     """
45     try:
46         if alert['sa_rule_severity']:
47             return SEVERITY_MAP.get(alert['sa_rule_severity'].lower(), -1)
48         if alert['severity']:
49             return SEVERITY_MAP.get(alert['severity'].lower(), -1)
50     except KeyError:
51         return -1

```

Running the Connector

The `main` function is defined for the actual execution of the Connector logic. The `output_handler` decorator is utilized for debugging and won't be covered in detail. The `main` function itself has an optional parameter of `is_test_run` which is set to `False` by default. As the parameter name suggests, it will determine if the Connector runs in production and actually ingests alerts or whether it will run from the Connector Testing tab in the application. Two empty lists are created in lines 56 and 57; more on them later. In line 58 the `siemplify` object is instantiated from the `SiemplifyConnectorExecution` class. This object will be utilized for the majority of the Connector execution. Line 61 instantiates the Connector whitelist which isn't used in this Connector and won't be covered in detail.

In lines 67-69, variables are defined for the parameters in the Connector. In line 71, an object is instantiated from the `NetskopeManager` and passes two of the parameter variables to ensure successful authentication (the code doing this in the Manager is not shown here). In line 73, the timestamp is fetched from a file that is created on the filesystem when the Connector executes. Line 74 and 75 perform some basic error handling for the first time the Connector runs since Netskope will not accept a timestamp of 0. Because Netskope expects epoch time, not millis time (remember the conversion that was done earlier the other way), `unix_now` retrieves the current time in millis format and this value must be divided by 1000 in order for Netskope to recognize it. After the start time and end time are defined, they are passed to the `get_alerts` method from the Manager. Usually end time is not a critical parameter to pass but the Netskope API requires an end time if a start time is used for querying.



Dealing with time and timestamps is one of the hardest things to do with a Connector. Different third party systems will return timestamps in various formats and some do not support querying by the format they return! Understanding the underlying API is critical.

A list of Netskope alerts are retrieved in line 77 and only the last one is selected in lines 79-80 if the Connector is executing a test run.

Overflow Logic

The Connector then iterates through the retrieved alerts and builds a Siemplify Alert out of each utilizing the `build_alert_info` function created earlier. It appends the Siemplify Alert to the empty list `all_alerts` defined earlier. The next piece of the Connector logic deals with Overflow; a very brief explanation of Overflow can be found [here](#):

Essentially Overflow is a threshold for alerts in a certain amount of time if an alert shares environment, product and rule generator. This is a built in mechanism to avoid system performance degradation. It's not mandatory but it is a good practice. In lines 104-105, if the `alert` is not determined to be overflow, it's appended to the empty alerts list defined earlier.

```

82 ~ .....for alert in nsalerts:-
83 ~ .....case = build_alert_info(alert, simplify)-
84 ~ .....all_alerts.append(case)-
85 ~ .....
86 ~ .....is_overflow = False-
87 ~ .....
88 ~ .....# Overflow logic-
89 ~ .....try:-
90 ~ .....    is_overflow = simplify.is_overflowed_alert(-
91 ~ .....        environment=case.environment,-
92 ~ .....        alert_identifier=str(case.ticket_id),-
93 ~ .....        alert_name=str(case.rule_generator),-
94 ~ .....        product=str(case.device_product)-
95 ~ .....    )-
96 ~ .....
97 ~ .....except Exception as e:-
98 ~ .....    simplify.LOGGER.error(-
99 ~ .....        "Overflow check failed for Alert {}".format(-
100 ~ .....            case.name)-
101 ~ .....    )-
102 ~ .....    simplify.LOGGER.exception(e)-
103 ~ .....
104 ~ .....if not is_overflow:-
105 ~ .....    alerts.append(case)-
106 ~ .....
107 ~ .....else:-
108 ~ .....    simplify.LOGGER.warn(-
109 ~ .....        "{alertname}-{alertid}-{environ}-{product} found as overflow alert, skipping this alert.".format(-
110 ~ .....            alertname=case.name,-
111 ~ .....            alertid=case.ticket_id,-
112 ~ .....            environ=case.environment,-
113 ~ .....            product=case.device_product)-
114 ~ .....    )-
115 ~ .....

```

Ending the Connector Execution

If the Connector is not a test run the timestamp must be updated to the last timestamp of retrieved alerts so that the Connector does not retrieve that data in its next iteration. Notice that the `all_alerts` list is sorted so even overflowed alerts will contribute to sorting by timestamp. In line 126, the list of non-overflowed alerts is submitted to the application which will create the Alerts in the GUI. Lines 128-131 define whether or not the Connector is a test run. Don't worry about the system arguments; these come from the application when the Run Connector Once button is pressed in the Testing tab.

```

116 ~ .....
117 ~ .....# update the timestamp-
118 ~ .....if not is_test_run:-
119 ~ .....    if all_alerts:-
120 ~ .....        newtimestamp = sorted(all_alerts, key=lambda alert: alert.end_time)[-1].end_time # sort ALL alerts (even overflow) by timestamp-
121 ~ .....    else:-
122 ~ .....        newtimestamp = timestamp-
123 ~ .....    simplify.save_timestamp(new_timestamp=newtimestamp)-
124 ~ .....
125 ~ .....simplify.LOGGER.info("-----Main--Finished-----")-
126 ~ .....simplify.return_package(alerts)-
127 ~ .....
128 ~ .....if __name__ == "__main__":-
129 ~ .....    # Connectors are run in iterations. The interval is configurable from the ConnectorsScreen UI.-
130 ~ .....    is_test_run = not (len(sys.argv) < 2 or sys.argv[1] == 'True')-
131 ~ .....    main(is_test_run=is_test_run)

```

Siemplify Installation Guide

Starting from version 5.2.0, Siemplify can be installed using an online Installer. Ask your Siemplify Account Manager to send you details on how to download the installer.

Installer Attributes

Simplify Installer provides advanced options to support alternative installation modes.

To change the mode, the user needs to add the relevant attribute when running the installer (this will install only the database module of Simplify):

```
sudo chmod +x simplify_installer.sh
sudo bash simplify_installer.sh --run_mode db
```

Short forms can also be used, for example:

```
sudo bash simplify_installer.sh -m db
```

Here is the full list of attributes that can be configured for the installation:

Attribute	Short Form	Default Value	Description
app_ip	ii	localhost	Application server address
cert_file	cf		Certificate file path
cert_passphrase	cp		Certificate password
customer_id	cid	(use for dpu mode only)	Set customer id for dpu
db_ip	di	localhost	Database server address
db_password	dpw		Database password
db_port	dp	5432	Database port
db_username	du	sa	Database username
deploy_demo	dd		Deploy with demo data
ha_cluster_password	hap		HA cluster user password
ha_cluster_vip	hav		HA cluster virtual IP
ha_host	hh		HA cluster host
help	h		Display help
hostname	ho	localhost.localdomain	Set machine hostname
htpasswd_pwd	hp		Set web authentication password
htpasswd_user	hu		Set web authentication user
localshare	lsf		Defines i as a local shared folder
run_mode	m	all_in_one	Installation Mode (options: all_in_one, app, db, dpu, ha)
shared_folder	sf		Shared folder for multi node deployment
shared_password	sp		Password for shared folder access

shared_username	su		Username for shared folder access
silent	s		Run in silent mode
uninstall	un		Uninstall Simplify
upgrade	upg		Upgrade Simplify
upgrade_backup_folder	up	/opt/ simplify_backup	Database backup path

Basic Installation (All-In-One)

Basic Installation (All in One)

The following procedure describes the basic process of running the Simplify Installer (deploying an AIO Node):

1. Prepare a clean CentOS or Red hat machine with the [Hardware Requirements of an AIO Deployment](#).
2. Copy the installer file to the machine.
3. SSH to the machine and run the following commands:

```
sudo chmod +x simplify_installer.sh
sudo bash simplify_installer.sh
```

```
(root@localhost ~)# sudo bash simplify_installer.sh
#####
Welcome to the Simplify Platform Installer <INSTALLER_VERSION>

The installer will install the Simplify SOAR platform in a variety of modes (see --help for further information)

Supported operating systems include:
Centos: Version: 7.5.1804 and above
#####

Loaded plugins: fastestmirror
Examining /var/tmp/yum-root-5SWET8/epel-release-latest-7.noarch.rpm: epel-release-7-12.noarch
Marking /var/tmp/yum-root-5SWET8/epel-release-latest-7.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package epel-release.noarch 0:7-12 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
epel-release noarch 7-12 /epel-release-latest-7.noarch 24 k
=====
Transaction Summary
=====
Install 1 Package

Total size: 24 k
Installed size: 24 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : epel-release-7-12.noarch 1/1
  Verifying : epel-release-7-12.noarch 1/1

Installed:
  epel-release.noarch 0:7-12

Complete!
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: centos.spd.co.il
 * epel: rep-epel-il.upress.io
 * extras: centos.spd.co.il
 * updates: centos.spd.co.il
```

4. Wait for the installation to finish (around 10-15 minutes).

Basic Installation (External Database)

Basic Installation with External Database

The following procedure describes how to install Simplify with an external database storage:

1. Prepare clean CentOS or RedHat machines with the [Hardware Requirements of an External Database Deployment](#).
2. Copy the installer file to the machines.
3. Install the DB node on the dedicated machine by executing the following commands:

```
sudo chmod +x simplify_installer.sh  
sudo bash simplify_installer.sh -m db
```
4. Wait for the installation to complete
5. Install the APP node on the dedicated machine by executing the following commands:

```
sudo chmod +x simplify_installer.sh  
sudo bash simplify_installer.sh -m app -di [db_ip] -sf //[shared_folder_ip]/[path] -su [username] -sp [password]
```
6. Wait for the installation to finish (around 10-15 minutes).

Scaled Installation (Multi-Node)

Scale Mode (Multi-Node)

The following procedure describes how to install Simplify in scale mode (multi-node deployment):

1. Decide on your preferred deployment type:
 - a. 2 Node Deployment – All-in-one node (with database on it) + DPU node
 - b. 3 Node Deployment – All-in-one node + DPU node + DB node (external database)
2. Prepare the machines required for the deployment following [this guide](#)
3. Copy the installer file to all the machines you prepared
4. Prepare a dedicated shared folder (address, username, password)
5. Follow the provided deployment steps:

2 Node Deployment

1. Install the AIO node on the dedicated machine by executing the following commands:

```
sudo chmod +x simplify_installer.sh
sudo bash simplify_installer.sh -sf //[shared_folder_ip]/[path] -su [username] -sp [password]
```
2. Wait for the installation to complete
3. Copy your customer ID from: `/opt/simplify/simplify_server/bin/Simplify.customer`
4. Now install the DPU node on the dedicated machine by executing the following commands:

```
sudo chmod +x simplify_installer.sh
sudo bash simplify_installer.sh -m dpu -ii [app_ip] -di [db_ip] -sf //[shared_folder_ip]/[path] -su [username] -sp [password] -cid [customer_id]
```
5. Wait for the installation to complete and access Simplify to start working

3 Node Deployment

1. Install the DB node on the dedicated machine by executing the following commands:

```
sudo chmod +x simplify_installer.sh
sudo bash simplify_installer.sh -m db
```
2. Wait for the installation to complete
3. Install the APP node on the dedicated machine by executing the following commands:

```
sudo chmod +x simplify_installer.sh
sudo bash simplify_installer.sh -m app -di [db_ip] -sf //[shared_folder_ip]/[path] -su [username] -sp [password]
```
4. Wait for the installation to complete
5. Copy your customer ID from: `/opt/simplify/simplify_server/bin/Simplify.customer`
6. Now install the DPU node on the dedicated machine by executing the following commands:

```
sudo chmod +x simplify_installer.sh
sudo bash simplify_installer.sh -m dpu -ii [app_ip] -di [db_ip] -sf //[shared_folder_ip]/[path] -su [username] -sp [password] -cid [customer_id]
```

7. Wait for the installation to complete and access Simplify to start working.

Upgrading Simplify

Running Upgrade

With every new release, the Simplify installer file can be used to upgrade your Simplify nodes (the AIO, APP and DPU nodes).

To upgrade a Simplify node, copy the installer to the machine and run the following command:

```
sudo bash simplify_installer.sh -upg
```

Database Maintenance

Databases require ongoing maintenance to prevent poor application performance, system downtime, and data loss. Here are Simplify guidelines for the maintenance of your Simplify database (PostgreSQL server).

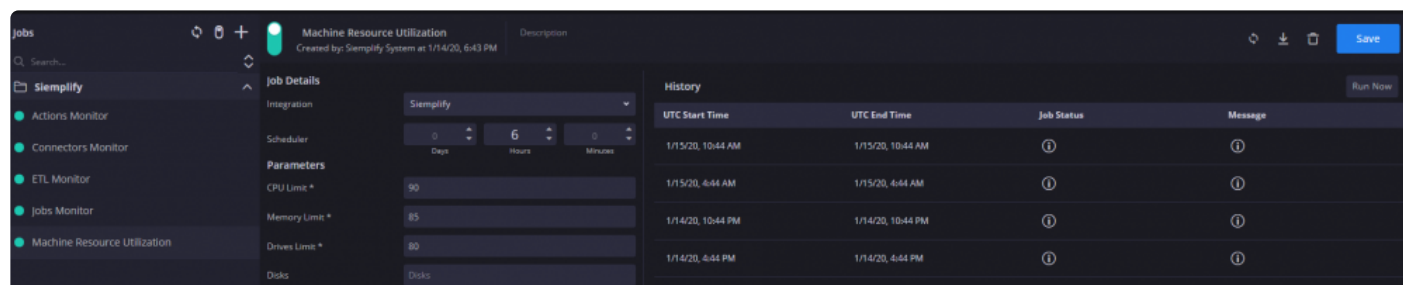
Monitoring

✿ Please note that the customer is responsible for Monitoring.

Please make sure to use your company's monitoring tools to monitor the following:

- Disk space and system load (800GB). If disk space exceeds 80%, please contact Simplify Support.
- CPU. If CPU exceeds 80% for more than 5 minutes in a row, please contact Simplify Support.
- Memory. If Memory exceeds 80% for more than 5 minutes in a row, please contact Simplify Support.

One option for monitoring is to use the Simplify Job “Machine Resource Utilization” in the Simplify Platform.



The screenshot displays the 'Machine Resource Utilization' job configuration in the Simplify Platform. The interface is divided into several sections: a left sidebar with navigation links, a main configuration area, and a history table.

Job Details:

- Integration:** Simplify
- Scheduler:** Days: 1, Hours: 6, Minutes: 0
- Parameters:**
 - CPU Limit: 90
 - Memory Limit: 85
 - Drives Limit: 80
 - Disks: Disks

History:

UTC Start Time	UTC End Time	Job Status	Message
1/15/20, 10:44 AM	1/15/20, 10:44 AM	①	①
1/15/20, 4:44 AM	1/15/20, 4:44 AM	①	①
1/14/20, 10:44 PM	1/14/20, 10:44 PM	①	①
1/14/20, 4:44 PM	1/14/20, 4:44 PM	①	①

Refer [here](#) for full information on data prerequisites.

Backup

✿ Please note that Simplify is responsible for the Backup procedure.

A database backup strategy is an important focus of any maintenance plan. While primarily meant to protect against data loss, database backups may also be necessary to address other significant maintenance requirements.

Simplify provides a built-in capability to run a daily full backup. You can use this option for an all-in-one deployment. However Simplify recommends running an external backup, with an incremental backup every day and a full backup once a week. Folder retention can handle two full backup files (i.e. the last two

weeks).

Simplify best practice is that for one year retention the backup folder should be 450 GB. The database backup files must be stored in an external share folder.

For information on using the Simplify Backup Settings, click [here](#).

PostgreSQL backup can be easily performed with pg_dump (or pgAdmin depending on your requirements). Note that making a copy of the database has no impact on it. However, this becomes impractical if the database is bigger than a couple of GB.

For more information on Backup and Restore a PostgreSQL database, click [here](#).

Import and Export

✿ Please note that Simplify is responsible for the Import and Export procedure.

In order to ensure smooth migration from one server to another, or to move from a single node to an HA mode, we need to export and import the database. For more details on Import and Export for PostgreSQL, click [here](#).

Routine Tasks

Vacuum Freeze

✿ Please note that Simplify is responsible for the Vacuum Freeze procedure.

Simplify will perform a vacuum freeze periodically depending on the customer's data load and volume. This is an important procedure which needs to be performed in order to recover or reuse disk space, update data statistics, update the visibility map, and protect against loss of very old data. The Vacuum Freeze procedure speeds up and optimizes database performance.

For more details on vacuum freeze, please click [here](#)

Reindex

✿ Please note that Simplify is responsible for the Reindex procedure.

From time to time Simplify periodically rebuilds indexes with the reindex command. The exact time period will depend on the customer's data and volume of their database changes.

For full details on Reindexing, please click [here](#).

Patching



Please note that Simplify is responsible for the Patching procedure.

Although Simplify relies on the publicly available CentOS repositories, we do not recommend that you apply untested patches on your production system.

All Simplify releases and updates, both major and minor, are certified to work on a fully patched CentOS system. In addition, all of the software components installed by Simplify (PostgreSQL, Elasticsearch) are updated with each release of Simplify.

Major upgrades in the OS, for example, from CentOS 7.5 to 8.0 will be addressed as part of a Simplify major release.

Should a critical vulnerability be exposed in the OS, our teams will test Simplify against the patch and release a corresponding minor update, which our engineers can install along with the OS patch.

Troubleshooting

Starting and Stopping Services

The following commands can be used if you are required to stop \ start \ restart any of Simplify services:

```
systemctl [command] Simplify.Server.service
systemctl [command] Simplify.Server.ETL.DataProcessingEngine.service
systemctl [command] Simplify.Server.Indexer.service
systemctl [command] Simplify.Server.PlaybookActions.service
systemctl [command] Simplify.Server.PythonExecution.service
systemctl [command] Simplify.Connectors.service
commands: start \ stop \ restart \ status
```

Simplify Operational Folders

1. **Simplify Server** – opt/simplify/simplify_server
2. **Simplify Client** – opt/simplify/simplify_web
3. **Simplify Web Server** – etc/nginx
4. **Simplify Channels** – etc/nginx
5. **Custom Integrations** – opt/simplify/simplify_server/customintegrations

Logs

Installer Logs:

The installer creates a text file (simplify_installer.log) in the same path with the installer file that logs the entire installation process.

System Logs:

Once Simplify is up and running, it will start generating logs that are collected in Elasticsearch.

The logs can be viewed in the Kibana dashboard accessible in: [http://\[simplify_server\]:5601](http://[simplify_server]:5601)

Hardening and Security Procedures

Application Level Security

Application

- The system user's passwords are securely stored in the database.
- Sensitive data such as integration passwords, usernames and/or app keys is encrypted and stored in the database.
- The system web APIs contain a built-in mechanism to prevent brute force attacks.
- System access to DB includes a built-in mechanism to prevent SQL injection attacks.
- Input validation is performed throughout the system for both client and server-side access.
- Playbook/integration are performed by a dedicated Sandbox server with limited access credentials.

Penetration Testing

- A full penetration test is performed on both appliance and application on a periodic basis.

OS Level Security

Network Access

- All communication is performed via HTTPS
- Network Access – Inbound & Outbound traffic is limited to all but necessary ports
- The SSL is provided with a valid, signed certificate
- Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides a mechanism for supporting access control security policies.

Additional Software

- Software installed on the appliance is limited to only required applications
- All open source software is scanned for Open Source License Compliance.

Operating System Updates

- The Appliances OS is kept up to date for every version release.

Vulnerability Scanning

- The appliance is thoroughly scanned for vulnerabilities on every release, utilizing leading Vulnerability Scanning solutions.

Access Control

- Strong user account credentials are enforced.
- Accounts are locked after exceeding maximum login attempts.

Remote Agent Infrastructure

Remote Agents

- All communication Remote Agents is performed via Job Publisher and limited to one-way communication.
- The Job Publisher data store is encrypted with a key that is not stored locally on the server.
- All data is deleted automatically after a set time period

High Availability Installation

[Overview](#)

[Prerequisites](#)

[Deployment Components](#)

[Deployment Process](#)

[Server Specifications](#)

[Upgrade Considerations](#)

Overview

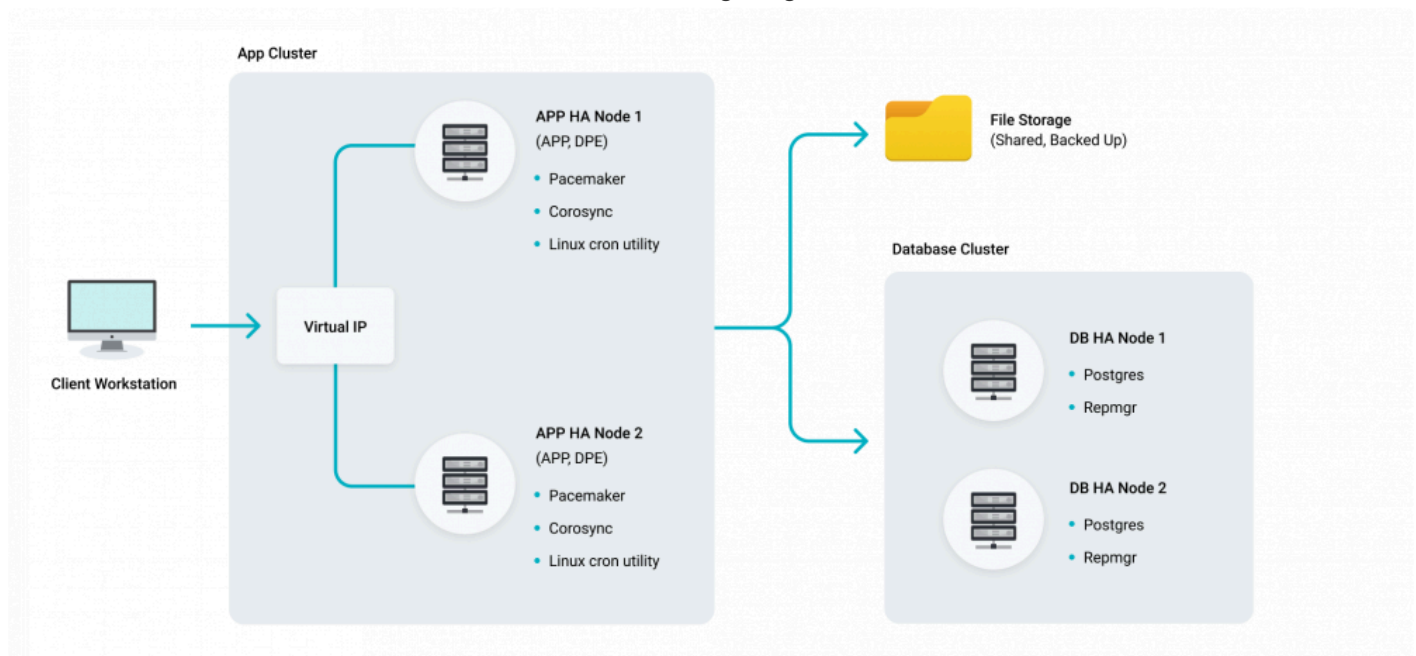
Overview

Siimplify provides multiple deployment modes with high-availability clusters to ensure the constant availability of services. There are three layers involved in the Siimplify High Availability mechanism:

- Application HA cluster
- Database HA cluster
- File storage

The two clusters work in a master/slave configuration, allowing automatic activation of Siimplify application and DB on another node if it failed for any reason (e.g hardware failure).

The overall architecture is demonstrated in the following diagram:



Siimplify High-Availability deployment contains the following components:

Application cluster

- Application Master Server
- Application Slave Server
- Virtual IP/Load Balancer
- Database cluster (based on PostgreSQL v10)*
- Database Master Server
- Database Slave Server

File Storage

The Simplify High Availability solution uses the following tools:

Database Cluster Tools

Repmgr (version 5.0) is an open-source tool suite for managing replication and failover in a cluster of PostgreSQL servers.

Application Cluster Tools

Pacemaker (version 1.1.19-8.el7_6.5) is an open-source high availability resource manager software used to manage resources, and ensure that they remain available in the event of a node failure

Corosync is an open source program that provides cluster membership and messaging capabilities, often referred to as the messaging layer.

Virtual IP as a load balancer – Cloud service or Pacemaker capability.

Linux cron utility is used to detect the active primary DB for the connection string.

Storage

Share folder with HA capabilities (RAID) and SMB protocol supporting

Prerequisites

Prerequisites

- Four servers (2 x APP servers, 2 x DB servers) of CentOS 7.5 / RedHat 7.6, In cloud deployment the RedHat should be with HA add-on.
- All servers should contain “simplify” user with permissions to use sudo command
- Files Storage – shared folder. Recommended to use storage service in the cloud.
- [Simplify HA DB rpms](#)
- [Simplify HA APP rpms](#)
- [Simplify DB availability detection script for CronJob](#) 4C2PNMwjLPh6HKFtHQPzXivP4-rs6wa/view?usp=sharing

Deployment Components

App Cluster: The application layer is composed of two Simplify application servers that do not store any persistent data and are able to switch between nodes. The process of switching between the application nodes can take up to a few minutes. In cloud environments, the entry point will be a load balancer, and on the on-prem topology a virtual IP will be used. On both cloud & On-prem, the switch between cluster nodes is controlled by CoroSync and Pacemaker.

DB Cluster: The database layer is composed of two database servers that are being replicated at all times using an open-source replication tool called Repmgr for PostgreSQL. The replication uses the same port that the application is using (e.g. 5432) and ssh (e.g 22). The application failover uses /etc/hosts file in order to pass the app the primary db connection IP. The database layer includes an automatic failover capability that will detect if the primary server is down for some reason – and it will promote the standby server to become the primary. Once this process is done (called promotion) the “fallen” primary can be returned to the cluster as a standby server and not as the primary again – this is done to prevent recurring errors on that server. The application uses a script (built into the application server) that will always detect which is the primary server and then connect to it

File Storage: Simplify HA solution uses shared storage. This storage should have its own HA capabilities in order to make sure that the HA architecture doesn't have any single point of failure (e.g. RAID/Cloud Replication).

The access to the app cluster is carried out through a balancer or Virtual IP (depends on cloud/on-prem). The master app node connects to both shared storage and DB and performs data operations. In case of an app failure – the second app server becomes master and begins to provide the service. In case of a DB failure, the slave DB will become the master and the main app node will re-establish a connection to the DB server and will continue to supply the service.

Deployment Process

This topic will outline the main steps for the offline installation of Simplify High-Availability system on Linux.

The deployment process consists of 3 main steps:

Step 1 – PostgreSQL & Repmgr Installation Procedure

Step 2 – Install Simplify Application system

Step 3 – Install Simplify DB availability detection script

Deployment process assumptions

- The customer should provide 4 servers with supported OS
- The servers should support Simplify servers specifications
- The “simplify” user will be created on all servers
- The deployment process required sudo permission for certain steps.
- IP Addresses – A customer should provide 6 static internal IP addresses
- 4 for APP and DB servers
- 1 for Virtual IP
- 1 for Shared Storage
- Network Access – All Simplify machines should have access to each other
- For connection between Simplify machine some Firewall rules will be updated
- The deployment process required dynamic configuration depending on servers parameters

Server Specifications

Siemplify Server (2 x APP Nodes)

- CPU 16 vCPU cores
- RAM 32 GB
- Storage 450 GB (preferably at "/" partition)
- Storage Disk Type: SSD / SAS 10k / Similar High-Speed Storage
- 1Gbps Network Adapter
- Ports 443, 80 (redirect), 5601 (Kibana), 9200 (Elastic), 22 (ssh)

Database Node (2 x DB Nodes)

- CPU 8 vCPU cores
- RAM 32 GB
- Storage 150 GB system + 450 GB data (/var/lib/pgsql OR wherever DB will be stored)
- Disk Type SSD / SAS 10k / Similar High-Speed Storage
- 1Gbps Network Adapter
- Ports 5432 (db PostgreSQL), 22 (ssh)

File Storage (with HA capabilities, e.g. RAID/Cloud replication)

- 150Gb of shared storage (SMB protocol)
- Disk Type SSD / SAS 10k / Similar High-Speed
- Ports 139, 445 (smb protocol)
- 1Gbps Network Adapter

Upgrade Considerations

Simplify recommends making a final decision about High-Availability(HA) support before first deployment. This is because HA deployment contains many components and complex processes.

Having said that, a customer can upgrade the system from single-node/all-in-one deployment to HA deployment. The upgrade process involves building a new environment with all components as a parallel environment according to the deployment guide and copying data and configurations from the existing all-in-one environment to the new HA environment. This process should be carried out by the Simplify Deployment Team. Note that this upgrade process to HA requires the down-time of the system.

Shutdown and Restart HA Servers

This article details the procedure to shut down and restart HA servers (including the Applications and the DB) without causing any damage to the data and to the ongoing work.

Prerequisites

- Master APP server
- Slave APP server
- Master DB server
- Slave DB server

Shut down HA App Servers with “Graceful shutdown”

A “graceful shutdown” involves shutting down each server separately as shown in the following procedure.

1. Connect via SSH to the applications – app1 + app2
2. Run the following command on the application you want to shutdown:
`pcs cluster stop [node] → pcs cluster stop app1`
3. Verify that the system is still working on the other node (App2) by entering:
`pcs status`

```
[root@app2 home]# pcs status
Cluster name: siimplify_cluster
Stack: corosync
Current DC: app2 (version 1.1.23-1.el7-9acf116022) - partition with quorum
Last updated: Sun Dec 13 16:47:34 2020
Last change: Sun Dec 13 16:43:15 2020 by hacluster via crmd on app2

2 nodes configured
8 resource instances configured

Online: [ app2 ]
Offline: [ app1 ]

Full list of resources:

Cluster_VIP      (ocf::heartbeat:IPaddr2):      Started app2
webserver        (ocf::heartbeat:nginx): Started app2
Server_service  (systemd:Siimplify.Server):    Started app2
Connectors_service (systemd:Siimplify.Connectors): Started app2
ETL_service      (systemd:Siimplify.Server.ETL.DataProcessingEngine): Started app2
Indexer_service  (systemd:Siimplify.Server.Indexer): Started app2
PlaybookActions_service (systemd:Siimplify.Server.PlaybookActions): Started app2
PythonExecution_service (systemd:Siimplify.Server.PythonExecution): Started app2

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

1. After maintenance work is carried out on this application you can turn it back by entering:
`pcs cluster start [node]--> pcs cluster start app1`

2. Repeat the same steps for the other application.

Shut down HA App Servers with “Kill shutdown”

A kill shutdown involves shutting both applications down at the same time.

1. Connect via SSH to the applications – app1 + app2
2. Run the following command on both applications:
`pcs cluster stop --all`
3. After maintenance is carried out, restart both application nodes by entering the following command on both nodes:
`pcs cluster start --all`
4. Verify using `pcs_status` on one of the nodes that everything is back to normal and that all services are working.

Shut down HA DB Servers with Full Shutdown (both DB and Apps)

A full shutdown of the HA Servers involves shutting down the DBs and the applications using the procedure below.

An HA cluster DB will have both the primary and standby DB nodes running as shown in the picture below.

ID	Name	Role	Status	Upstream	Location	Priority	Timeline
1	N1	primary	* running		default	100	1
2	N2	standby	running	N1	default	100	1

1. If you want to maintain both Databases in parallel before they start, you need to shut down both applications using `pcs cluster stop --all`
2. Connect to both DBs via SSH.
3. Stop postgres services on both nodes (N1+N2) by entering:
`systemctl stop postgresql-10`
4. After you finish maintenance, begin postgres services on N1
`systemctl start postgresql-10`
5. Rejoin N2 as standby node by entering:

```
sudo su -
systemctl stop postgresql-10
su postgres
cd ~
PGPASSWORD=<postgres user password> /usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf -h <N1 ip> -U repuser -d postgres standby clone --force-rewind --force
exit
systemctl start postgresql-10
su postgres
/usr/pgsql-10/bin/repmgr standby register --force
```

6. On both nodes, verify the DB has returned to the starting point with the following command:

```
/usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf cluster show and verify as in the picture below
```

ID	Name	Role	Status	Upstream	Location	Priority	Timeline
1	N1	primary	* running		default	100	1
2	N2	standby	running	N1	default	100	1

7. Run the following command to restart the services on any of the two application nodes:

```
pcs cluster start --all
```

8. Verify on one of the nodes that everything is back to normal and all services are working:

```
pcs_status
```

```
[root@hal ~]# pcs status
Cluster name: simplify cluster
Stack: corosync
Current DC: hal.simplify.com (version 1.1.20-5.el7_7.2-3c4c782f70) - partition with quorum
Last updated: Mon Dec 23 17:47:02 2019
Last change: Mon Dec 23 17:46:59 2019 by root via cibadmin on hal.simplify.com

2 nodes configured
8 resources configured

Online: [ hal.simplify.com ha2.simplify.com ]

Full list of resources:

Cluster_VIP (ocf::heartbeat:IPaddr2): Started hal.simplify.com
webserver (ocf::heartbeat:nginx): Started hal.simplify.com
Server_service (systemd:Siimplify.Server): Started hal.simplify.com
Connectors_service (systemd:Siimplify.Connectors): Started hal.simplify.com
ETL_service (systemd:Siimplify.Server.ETL.DataProcessingEngine): Started hal.simplify.com
Indexer_service (systemd:Siimplify.Server.Indexer): Started hal.simplify.com
PlaybookActions_service (systemd:Siimplify.Server.PlaybookActions): Started hal.simplify.com
PythonExecution_service (systemd:Siimplify.Server.PythonExecution): Started hal.simplify.com

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

Shut down HA DB Servers with Full Shutdown (DB only)

This full shutdown procedure of the HA Servers involves shutting down just DBs but leaving the applications running.

Shut down Replica Node N2

1. Connect with SSH
2. Run the command in order to stop the postgres services:

```
systemctl stop postgresql-10
```
3. Verify that applications are working as usual.
4. When maintenance work is done, restart the postgres services with

```
systemctl start postgresql-10
```


- Wait for the N2 to replicate N1 to cover for loss of data before shutting down N1

Shutdown Node N1

- Connect to N1 and N2 with SSH
- Stop postgres services on N1 using the following command:

```
systemctl stop postgresql-10
```
- On N2 switch to postgres user:

```
su - postgres
```
- Type the command

```
/usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf cluster show
```
- When N2 becomes the Primary server you can perform maintenance on N1. You should see the HA as failed as in the screenshot below:

ID	Name	Role	Status	Upstream	Location	Priority	Timeline
1	N1	primary	- failed		default	100	?
2	N2	primary	* running		default	100	2

Register N1 as standby

- In order for N1 to replicate the data created on N2, during maintenance of N1, enter the following:

```
sudo su -
systemctl stop postgresql-10
su postgres
cd ~
PGPASSWORD=<postgres user password> /usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf -h <N2 ip> -U repuser -d postgres standby clone --force-rewind --force
exit
systemctl start postgresql-10
su postgres
/usr/pgsql-10/bin/repmgr standby register --force
```

ID	Name	Role	Status	Upstream	Location	Priority	Timeline
1	N1	standby	running	N2	default	100	2
2	N2	primary	* running		default	100	2

Return to Starting Position

- On Node 2 (currently acting as Primary) stop postgres services:

```
systemctl stop postgresql-10
```
- Go to N1 and wait for it to become primary:

```
sudo - postgres
```

```
/usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf cluster show
```

3. Once N1 becomes primary, move to N2 and register N2 as standby:

```
sudo su -
su postgres
cd ~
PGPASSWORD=<postgres user password> /usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf -h <N1 ip> -U repuser -d postgres standby clone --force-rewind --force
exit
systemctl start postgresql-10
su postgres
/usr/pgsql-10/bin/repmgr standby register --force
```

4. Verify on both nodes that the cluster is back to the starting position

```
/usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf cluster show
```

and as shown in the screenshot below:

ID	Name	Role	Status	Upstream	Location	Priority	Timeline
1	N1	primary	* running		default	100	1
2	N2	standby	running	N1	default	100	1

5. Verify repmgr services are active on both nodes

```
systemctl status repmgr10
```

Disaster Recovery Installation



This guide is compatible with CentOS 7.8 + all-in-One deployment mode + using Online installer.

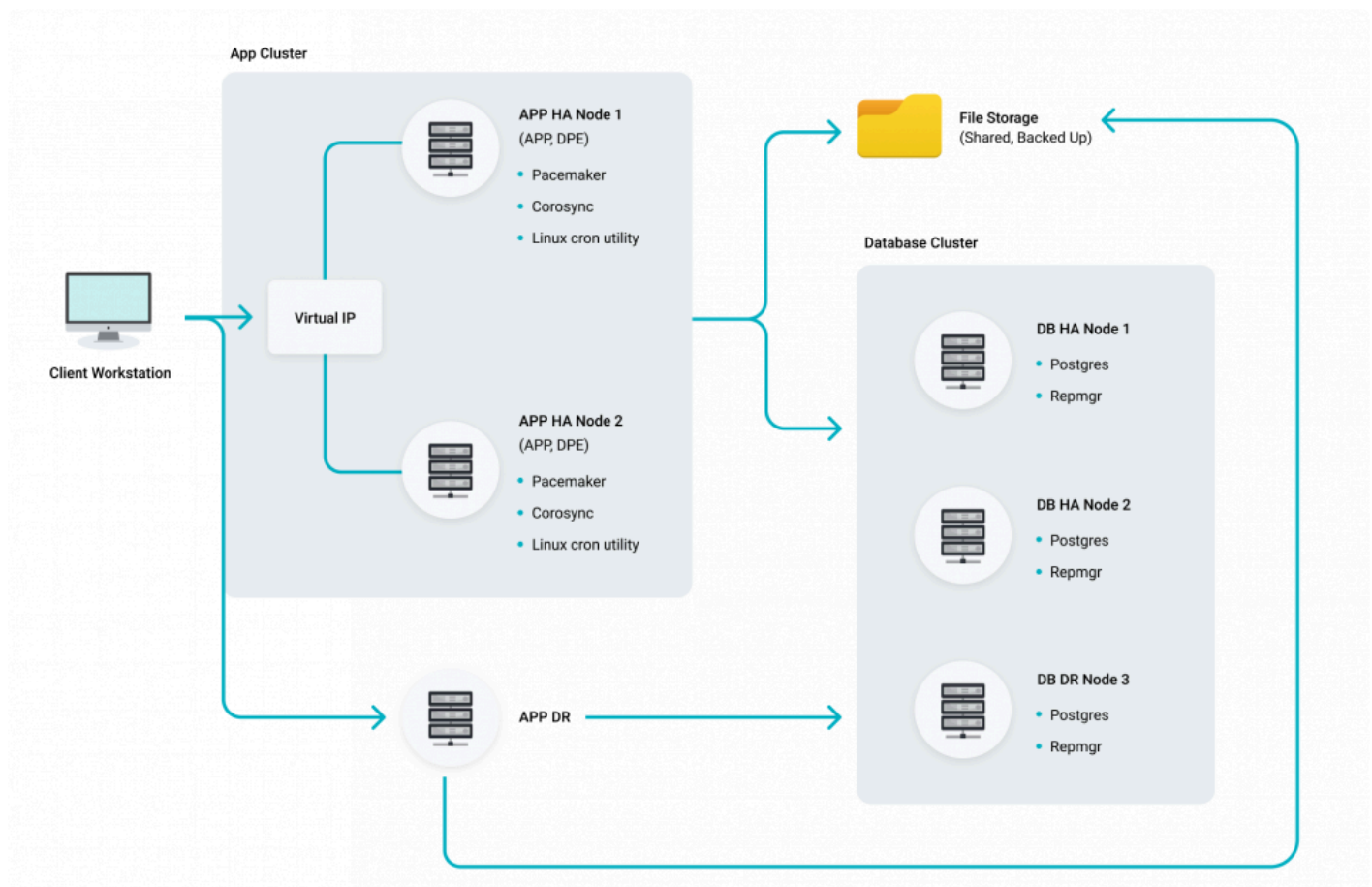
Overview

Simplify provides multiple deployment modes with high-availability clusters to ensure the high availability of services. There are three layers of Simplify High Availability mechanism:

- Application HA cluster
- Database HA cluster
- File storage

The two clusters work in an active-passive configuration, allowing automatic activation of Simplify application and DB on another node if it has failed for any reason (e.g hardware failure).

The overall architecture is demonstrated in the following diagram:



Database High-Availability

Simplify HA solution uses the following tools:

- Database High Availability
 - Repmgr (version 5.0) is an open-source tool suite for managing replication and failover in a

cluster of PostgreSQL servers.- <https://repmgr.org>

- Application High-Availability:
 - Pacemaker (version 1.1.19-8.el7_6.5) is an open-source high availability resource manager software used to manage resources, and ensure that they remain available in the event of a node failure – <https://wiki.clusterlabs.org/wiki/Pacemaker>
 - Corosync is an open source program that provides cluster membership and messaging capabilities, often referred to as the messaging layer. – <http://corosync.github.io/corosync>
 - Virtual IP as a load balancer – Cloud service or Pacemaker capability.
 - Linux cron utility – used to detect the active primary DB for the connection string. – <https://en.wikipedia.org/wiki/Cron>
- Storage
 - Share folder with HA capabilities(RAID) and SMB protocol supporting

Prerequisites

- Four servers of CentOS 7.5 / RedHat 7.6, In cloud deployment the RedHat should be with HA add-on.
 - Master APP server
 - Slave APP server
 - Master DB server
 - Slave DB server
- For DR solution – 2 additional servers: APP server and DB server.
- All servers should contain “simplify” user with permissions to use sudo command
- Files Storage – share folder. It is recommended to use storage service in the cloud.
- Simplify HA DB rpms: (ask Customer Success for link)
- Simplify HA APP rpms: (ask Customer Success for link)
- Simplify detect script for CronJob: (ask Customer Success for link)

Deployment Process

This topic will outline the main steps for the offline installation of Simplify High-Availability system on Linux.

The deployment process consists of 5 steps:

- Step 1 – [PostgreSQL & Repmgr Installation Procedure](#)
- Step 2 – [Install Simplify Application Nodes](#)
- Step 3 – [Install Simplify DB availability detection script](#)
- Step 4 – [Revert DR system to Primary](#)
- Step 5 – [Return to main Site](#)

PostgreSQL & Repmgr Installation Procedure

This section will provide explanations for the offline installation of PostgreSQL 10 on Linux – including replication configuration with automatic failover (using REPMGR).

Please make sure you have the “HA DB rpms.zip” file which contains all the necessary RPMs.

For this first step of the procedure, we will use two DB servers, one as the Master and the second as Slave.

Installation Flow High Level Summary

1. Extract & Install (on all DB servers)
2. Configure PostgreSQL & allow ports – on all DB servers
3. Restart PostgreSQL – on all DB servers
4. Configure REPMGR – On the master DB server
5. Configure REPMGR – On the slave DB server
6. Configure REPMGR – On the slave DB server (mandatory for DR)
7. Pre-failover Actions

Extract and Install

The following procedure should be executed with sudo permissions on all three servers – Master DB, Slave DB and the DR DB.

1. Download the HA DB rpms.zip file to the home folder of the Simplify user.
2. Enter the following command:
3. `sudo su -`
4. `cd /home` Alternatively, go to the folder that contains the zip with the RPMs
5. `unzip HA DB rpms.zip`
6. `yum remove libicu-50.2-4.el7_7.x86_64 -y`
7. `yum install *.rpm -y`
8. `systemctl enable postgresql-10`
9. `passwd postgres` Enter a password here for the postgres user
10. `su - postgres`
11. `ssh-keygen -t rsa`
12. `ssh-copy-id postgres @ <OTHER_IP_ADDRESS>` Add here the IP address of the other DB
13. Make sure all the .ssh key files have the right permissions
 - a. `chmod 700 ~/.ssh`
 - b. `chmod 600 ~/.ssh/authorized_keys`
 - c. `chmod 644 ~/.ssh/known_hosts`
 - d. `chmod 600 ~/.ssh/id_rsa`
 - e. `chmod 644 ~/.ssh/id_rsa.pub`
 - f. `restorecon -R -v ~/.ssh`
14. Validate passwordless connection between the servers using


```
ssh postgres !https://manula.r.sizr.io/large/user/14758/img/at sign.png! <remoteIP>
```
15. `exit`
16. PostgreSQL 10 will be installed with all the necessary packages of REPMGR 5.0.
To check the status of the postgres service use the following command:
`systemctl status postgresql-10`

Configure PostgreSQL

1. Initialize the database folder “/var/lib/pgsql/10/data” using the following command. Please keep in mind that the Linux user “postgres” has the data directory already configured as an environment variable called “\$PGDATA”:

```
sudo -u postgres /usr/pgsql-10/bin/initdb -D /var/lib/pgsql/10/data
```
2. Change the following configurations in the “/var/lib/pgsql/10/data/postgresql.conf” configuration file as follows: (Make sure to remove the comments #)

```
wal_level = replica
archive_mode = on
archive_command = '/bin/true'
max_wal_senders = 10
wal_sender_timeout = 3600s
max_replication_slots = 10
hot_standby = on
hot_standby_feedback = on
autovacuum = on
work_mem = 24MB
shared_buffers = 12GB
max_connections = 1000
listen_addresses = '*'
shared_preload_libraries = 'repmgr'
wal_log_hints = on
```

If you want more information on these parameters, please look here:

The parameter descriptions you can find in this link:

<https://www.postgresql.org/docs/10/runtime-config.html>

<https://www.postgresql.org/docs/10/runtime-config-replication.html> – (for Replication)

Note that the configuration above is for VM with 12 CPUs and 32GB RAM and is based on the following:

VM Type	shared_buffers	work_mem	max_worker_processes
2 CPU \ 8GB RAM	2GB	4MB	2
8 CPU \ 16GB RAM	3GB	6MB	8
12 CPU \ 32GB RAM	12GB	24MB	8

3. Next, we need to add more rule to the “/var/lib/pgsql/10/data/pg_hba.conf” file to accept all connections from all sources. Note that Simplify recommends typing it into the file and not copy/pasting it.

IPv4 local connections:

host all all 0.0.0.0/0 md5

host replication all 0.0.0.0/0 md5

4. Finally, we need to add “postgres” as a sudoer for the “system_start_command” in repmgr as follows.

Add a new text file “/etc/sudoers.d/postgres” which contains the following:

```
Defaults:postgres !requiretty
postgres ALL = NOPASSWD: /usr/bin/systemctl stop postgresql-10,/usr/bin/systemctl start postgresql-10,/usr/bin/systemctl restart postgresql-10,/usr/bin/systemctl reload postgresql-10
```

Restart PostgreSQL

After configuring the PostgreSQL, we need to restart PostgreSQL:

1. Enter the command

```
systemctl restart postgresql-10
```

2. After restarting we need to open the default port (5432) using firewall-cmd:

(root)

- a. `firewall-cmd --add-port=5432/tcp --permanent`
- b. `firewall-cmd --reload`

3. Enter the following commands on the primary node ONLY:

- a. `su - postgres`
- b. `psql -c "alter role postgres password '<postgres user password>';"`
- c. `exit`

4. On all nodes create a user (role) in PostgreSQL to perform the replication. The user has to have the replication role on all nodes:

- a. `su - postgres`
- b. `psql -c "CREATE ROLE repuser LOGIN SUPERUSER REPLICATION PASSWORD '<postgres user password>';"`
- c. `exit`

5. Create the REPMGR extension once for each PostgreSQL server:

- a. `su - postgres`
- b. `psql -c "CREATE EXTENSION repmgr;"`
- c. `exit`

6. Finally, we need to update the path in “~/.bash_profile” (user: root). Add the following line to the `bash_profile` file (Note the commands are case-sensitive):

- a. `vi /root/.bash_profile`
Replace “`export PATH`” with “`export PATH=/usr/pgsql-10/bin:$PATH`”
- b. `source /root/.bash_profile`
- c. Validate by running “`repmgr`”, you will get an error message “cannot be run as root

Configure REPMGR (primary node)

This needs to be carried out on the primary (master) node. Note that the commands executed are different in the primary node than the secondary.

1. Configure the REPMGR configuration file (default located on “/etc/repmgr/10/repmgr.conf”) by changing the configuration as follows:
(Remove comments # in the file)

```
node_id = 1
node_name = 'N1'
conninfo = 'host=<LOCAL IP> port=5432 user=repuser dbname=postgres password=<POSTGRES USER PASSWORD>'
replication_type = 'physical'
use_replication_slots = yes
pg_bindir = '/usr/pgsql-10/bin'
failover = 'automatic'
data_directory = '/var/lib/pgsql/10/data'
promote_command = '/usr/pgsql-10/bin/repmgr standby promote -f /etc/repmgr/10/repmgr.conf --log-to-file'
follow_command = '/usr/pgsql-10/bin/repmgr standby follow -f /etc/repmgr/10/repmgr.conf --log-to-file --upstream-node-id=%n'
monitoring_history = yes
monitor_interval_secs = 10
service_start_command = 'sudo systemctl start postgresql-10'
service_stop_command = 'sudo systemctl stop postgresql-10'
service_restart_command = 'sudo systemctl restart postgresql-10'
service_reload_command = 'sudo systemctl reload postgresql-10'
```

Note that you can find the the configuration parameters description here:

<https://raw.githubusercontent.com/2ndQuadrant/repmgr/master/repmgr.conf.sample>

2. After changing the above configurations, the linux postgres user needs to initialize the primary node.
 - a. `su - postgres`
 - b. `/usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf primary register`
 - c. Make sure that the primary was registered by running the following command:
`/usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf cluster show`

Configure REPMGR (replica node)

This needs to be carried out on the secondary (replica) node. Note that the commands executed here are different in the secondary node.

1. Configure the REPMGR configuration file (default located on “/etc/repmgr/10/repmgr.conf”) by changing the configuration as follows:

```
node_id = 2
node_name = 'N2'
conninfo = 'host=<LOCALIP> port=5432 user=repuser dbname=postgres password=<POSTGRES USER PASSWORD>'
replication_type = 'physical'
use_replication_slots = yes
pg_bindir = '/usr/pgsql-10/bin'.
failover = 'automatic'
data_directory = '/var/lib/pgsql/10/data'
promote_command = '/usr/pgsql-10/bin/repmgr standby promote -f /etc/repmgr/10/repmgr.conf --log-to-file'
follow_command = '/usr/pgsql-10/bin/repmgr standby follow -f /etc/repmgr/10/repmgr.conf --log-to-file --upstream-node-id=%n'
monitoring_history = yes
monitor_interval_secs = 10
service_start_command = 'sudo systemctl start postgresql-10'
service_stop_command = 'sudo systemctl start postgresql-10'
service_restart_command = 'sudo systemctl restart postgresql-10'
service_reload_command = 'sudo systemctl reload postgresql-10'
```

After changing the above configurations, we need to initialize the replica node. This should be done by the linux postgres user except for the systemctl commands:

2. Shutdown PostgreSQL (root):

```
systemctl stop postgresql-10
```

3. Delete the data directory (as it will be cloned later on):

```
cd /var/lib/pgsql/10/data
```

```
rm -rdf **
```

```
# Clone the data from the primary (postgres user):
```

```
## su - postgres@
```

- a. PGPASSWORD=<place postgres user password???*> /usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf -h <primary-ip> -U repuser -d postgres standby clone

- b. exit

4. Start PostgreSQL (root):

```
systemctl start postgresql-10
```

5. Register the standby (postgres user):

- a. `su - postgres`
- b. `/usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf standby register`
- c. `exit`

6. Ensure that the primary was registered by running the following commands:

- a. `su - postgres`
- b. `/usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf cluster show`
- c. `exit`

Configure REPMGR (replica node – DR)

This needs to be carried out on the secondary (replica) node on the DR database. Note that the commands executed here are different in the secondary node.

1. Configure the REPMGR configuration file (default located on “/etc/repmgr/10/repmgr.conf”) by changing the configuration as follows:

```
node_id = 3
node_name = 'DR'
conninfo = 'host=<local-ip> port=5432 user=repuser dbname=postgres password=<postgres user password>'
replication_type = 'physical'
use_replication_slots = yes
pg_bindir = '/usr/pgsql-10/bin'.
failover = 'automatic'
priority=0
data_directory = '/var/lib/pgsql/10/data'
promote_command = '/usr/pgsql-10/bin/repmgr standby promote -f /etc/repmgr/10/repmgr.conf --log-to-file'
follow_command = '/usr/pgsql-10/bin/repmgr standby follow -f /etc/repmgr/10/repmgr.conf --log-to-file --upstream-node-id=%n'
monitoring_history = yes
monitor_interval_secs = 10
service_start_command = 'sudo systemctl start postgresql-10'
service_stop_command = 'sudo systemctl stop postgresql-10'
service_restart_command = 'sudo systemctl restart postgresql-10'
service_reload_command = 'sudo systemctl reload postgresql-10'
```

2. After changing the above configurations, the postgres linux user needs to initialize the replica node. (except for the systemctl commands)
3. Shut down PostgreSQL (root) by entering the following command:
`systemctl stop postgresql-10`
4. Delete the data directory (as it will be cloned later on):
 - a. `cd /var/lib/pgsql/10/data`
 - b. `rm -rdf *`
5. Clone the data from the primary (postgres user):
 - a. `su - postgres`
 - b. `PGPASSWORD=<place postgres user password> /usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf -h <primary-ip> -U repuser -d postgres standby clone`
 - c. `exit`
6. Start PostgreSQL (root):
`systemctl start postgresql-10`
7. Register the standby (postgres user):
 - a. `su - postgres`

- b. `/usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf standby register`
- c. `exit`

8. Make sure that the secondary was registered by running the following command:

- a. `su - postgres`
- b. `/usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/repmgr.conf cluster show`
- c. `exit`

9. Start repmgr10 Service (root) – on all nodes!

- a. `systemctl enable repmgr10`
- b. `systemctl start repmgr10`
- c. **Validate the service is running by entering** `systemctl status repmgr10`

PreFailover Actions

REPMGR supports automatic failover. In a case of a failover – we need to “REJOIN” the fallen node to the cluster manually with the following commands:

```
sudo su -
systemctl stop postgresql-10
su postgres
cd ~
PGPASSWORD=<postgres user password> /usr/pgsql-10/bin/repmgr -f /etc/repmgr/10/re
pmgr.conf -h <new primary ip> -U repuser -d postgres standby clone --force-rewin
d --force
exit
systemctl start postgresql-10
su postgres
/usr/pgsql-10/bin/repmgr standby register --force
```

Troubleshooting:

If the failover does not work use the following command on the standby DB machine:

1. `su - postgres`
2. `/usr/pgsql-10/bin/repmgr standby promote`

Install Simplify application Nodes

The following procedure needs to be carried out twice – first for the Master Simplify node (primary) and then for the Slave Simplify node (replica).

1. Adjust the timezone on Simplify APP machine with the following command:

```
timedatectl set-timezone [selected_timezone] e.g. Asia/Jerusalem
```

or use the following command for a list of timezones: `timedatectl list-timezones`

2. Copy the installation file
3. Set execution permission for the installation file by running:

```
sudo chmod +x simplify_installer.sh
```

4. On the Master node, run the following:

```
sudo bash simplify_installer.sh --run_mode ha --db_ip [primary_db] --db_port [db_port] (default:5432) --db_username [db_username] --db_password [db_password] --hostname [master_machine_hostname] --ha_host [master_machine_ip],[master_machine_hostname] --ha_host [slave_machine_ip],[slave_machine_hostname] --ha_cluster_vip [app_server_vip] -sf //[shared_folder_ip]/i -su [shared_folder_username] -sp [shared_folder_password]
```

5. On the Slave node, run the following:

```
sudo bash simplify_installer.sh --run_mode ha --db_ip [primary_db] --db_port [db_port] (default:5432) --db_username [db_username] --db_password [db_password] --hostname [slave_machine_hostname] --ha_host [master_machine_ip],[master_machine_hostname] --ha_host [slave_machine_ip],[slave_machine_hostname] --ha_cluster_vip [app_server_vip] -sf //[shared_folder_ip]/i -su [shared_folder_username] -sp [shared_folder_password]
```

6. On the DR App node, run the following:

```
sudo bash simplify_installer.sh --run_mode dr --db_ip [db_dr_ip] --db_port [db_port, default:5432] --db_username [db_username] --db_password [db_password] -sf //[shared_folder_ip]/i -su [shared_folder_username] -sp [shared_folder_password]
```

Example Commands

```
HA_VIP = 10.0.1.99
10.0.0.59
node1: sudo bash simplify_installer.sh --run_mode ha --db_ip 10.0.1.98 --db_port 5432 --db_username simplifydb --db_password djksfjdkfsj --hostname ha1.simplify.com --ha_host 10.0.0.59,ha1.simplify.com --ha_host 10.0.0.76,ha2.simplify.com --ha_cluster_vip 10.0.1.99 -sf //172.22.14.3/y2 -su test -sp Aa123456
```

```
10.0.0.76
node2: sudo bash siemplify_installer.sh --run_mode ha --db_ip 10.0.1.98 --db_port 5432 --db_username siemplifydb --db_password Yeswecan2015 --hostname ha2.siemplify.com --ha_host 10.0.0.59,ha1.siemplify.com --ha_host 10.0.0.76,ha2.siemplify.com --ha_cluster_vip 10.0.1.99 -sf //172.22.14.3/y2 -su test -sp Aa123456
```

Check Status Cluster

- Check the status of the cluster using the following command:

```
pcs status
```

The result should look like this:

```
[root@hal ~]# pcs status
Cluster name: siemplify cluster
Stack: corosync
Current DC: hal.siemplify.com (version 1.1.20-5.e17_7.2-3c4c782f70) - partition with quorum
Last updated: Mon Dec 23 17:47:02 2019
Last change: Mon Dec 23 17:46:59 2019 by root via cibadmin on hal.siemplify.com

2 nodes configured
8 resources configured

Online: [ hal.siemplify.com ha2.siemplify.com ]

Full list of resources:

Cluster_VIP      (ocf::heartbeat:IPaddr2):      Started hal.siemplify.com
webserver        (ocf::heartbeat:nginx): Started hal.siemplify.com
Server_service   (systemd:Siemplify.Server):    Started hal.siemplify.com
Connectors_service (systemd:Siemplify.Connectors): Started hal.siemplify.com
ETL_service      (systemd:Siemplify.Server.ETL.DataProcessingEngine): Started hal.siemplify.com
Indexer_service  (systemd:Siemplify.Server.Indexer): Started hal.siemplify.com
PlaybookActions_service (systemd:Siemplify.Server.PlaybookActions): Started hal.siemplify.com
PythonExecution_service (systemd:Siemplify.Server.PythonExecution): Started hal.siemplify.com

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

Tips and Commands

After a successful installation we will have 2 nodes and 8 resources:

- Cluster_VIP
- webserver = nginx
- Server_service
- Connectors_service
- ETL_service
- Indexer_service
- PlaybookActions_service
- PythonExecution_service

If you connect to the VIP (virtual IP) it will connect to the active node.

If you connect to the DB VIP it will connect to the active DB.

If one or more resource is stuck you can use the following command to reset:

```
pcs resource failcount reset [RESOURCE]
```

How to manually switch nodes:

1. Make sure the node that you want to switch to is online:

```
Online: [ ha1.siemplify.com ha2.siemplify.com ]
```

2. If one of the nodes is on standby it will look like this:

```
Node ha2.siemplify.com: standby
Online: [ ha1.siemplify.com ]
```

3. We can “unstandby” the node with the following command:

```
pcs cluster unstandby ha2.siemplify.com
```

4. After we are sure that both nodes are online I can set the active node on standby with the following command:

```
pcs cluster standby ha1.siemplify.com
```

This command will cause switching nodes as follows:

```
[root@ha1 ~]# pcs status
Cluster name: siemplify_cluster
Stack: corosync
Current DC: ha1.siemplify.com (version 1.1.20-5.el7_7.2-3c4c782f70) - partition with quorum
Last updated: Mon Dec 23 18:23:11 2019
Last change: Mon Dec 23 18:22:14 2019 by root via cibadmin on ha1.siemplify.com

2 nodes configured
8 resources configured

Node ha1.siemplify.com: standby
Online: [ ha2.siemplify.com ]

Full list of resources:

Cluster_VIP      (ocf::heartbeat:IPaddr2):      Started ha2.siemplify.com
webserver        (ocf::heartbeat:nginx):        Started ha2.siemplify.com
Server_service   (systemd:Siemplify.Server):     Started ha2.siemplify.com
Connectors_service (systemd:Siemplify.Connectors): Started ha2.siemplify.com
ETL_service      (systemd:Siemplify.Server.ETL.DataProcessingEngine): Started ha2.siemplify.com
Indexer_service  (systemd:Siemplify.Server.Indexer): Started ha2.siemplify.com
PlaybookActions_service (systemd:Siemplify.Server.PlaybookActions): Started ha2.siemplify.com
PythonExecution_service (systemd:Siemplify.Server.PythonExecution): Started ha2.siemplify.com

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

Don't forget to “unstandby” if you want to go back to the previous node

Install Simplify DB availability detection script

1. Edit `siimplify_pg_detect.sh` and update
 - a. `V_SERVER_NODE_1="[primary_db_ip]"`
 - b. `V_SERVER_NODE_2="[slave_db_ip]"`
 - c. `PRIMARY_NODE_1=$(PGPASSWORD=[db_password]`
 - d. `PRIMARY_NODE_2=$(PGPASSWORD=[db_password]`
2. Upload HA APP rpms.zip and `siimplify_pg_detect.sh` to both app machines
3. Unzip HA APP rpms.zip and install with "sudo yum install" the rpms on both app machines.
4. Configure the crontab job on both app machines with the following commands:
 - a. `chmod +x /home/siimplify_pg_detect.sh`
 - b. `crontab -l > //tmpjobs`
 - c. `echo reboot /home/siimplify_pg_detect.sh >> /root/tmpjobs@` (make sure the path here is the same as the path in a above)
 - d. `crontab /root/tmpjobs`
 - e. `rm -f /root/tmpjobs`
 - f. `reboot - both nodes`
 - g. Use `ps -aux | grep siimplify` to make sure the process finished successfully.

Make DR system primary

1. In the DR DB system rename the recovery.conf file to recovery_backup.conf as follows:

```
mv /var/lib/pgsql/10/data/recovery.conf /var/lib/pgsql/10/data/recovery_backup.conf
```

2. Restart postgres service

```
systemctl restart postgresql-10
```

Return to main site

1. Perform a full db backup on the DR DB using postgres tool: <https://www.postgresql.org/docs/current/backup.html>.
2. Stop application services on the primary.
3. Unregister primary and standby db from the cluster.
4. Register primary the same way as in Configure REPMGR – Primary Node
5. Restore the DB using the backup from the DR.
6. Register standby the same way as in Configure REPMGR – Replica Node