Siemplify

System Architecture

Table of Contents

Architecture Guide

Architecture Overview

The Siemplify platform is used to manage the entire security operations process in an MSSP or enterprise SOC. The platform provides solutions for two major fields:

Handling and solving security threats

- Data ingestion & transformation
- Data enrichment
- Data fusion
- Alert grouping
- Alert prioritization
- Visualization & dashboards
- Orchestration
- Automation
- Response

SOC management

- Case management and auditing
- Collaboration and escalation
- End customers management
- KPIs and ROI measurements
- Reporting

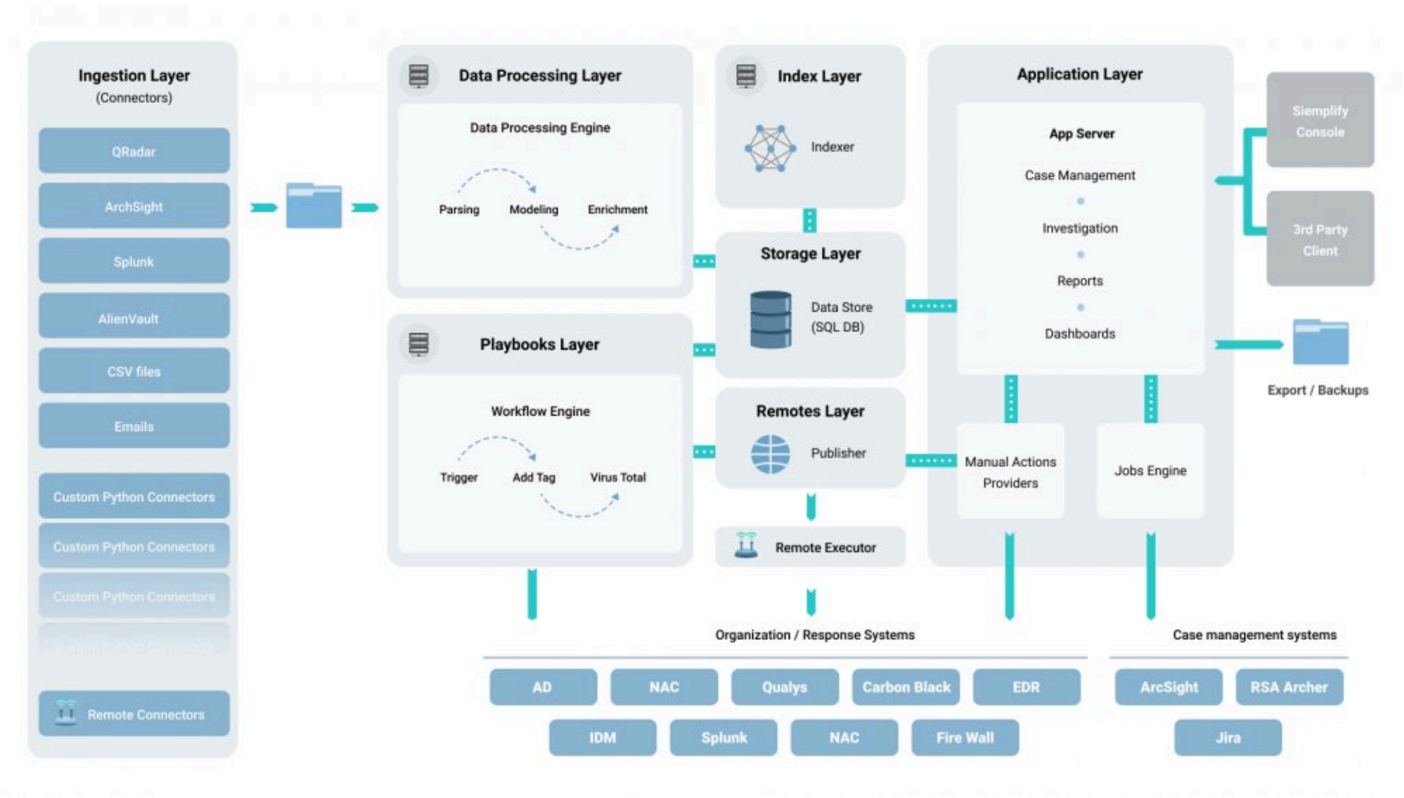
- Maintaining the knowledge base of the SOC

Siemplify offers multiple deployment modes to support scaled solutions. In addition, the system provides full multi-tenancy to support MMSP/MSSP requirements and use cases.

System Components

The Siemplify system architecture is multi layered and contains several major components:

- Connectors (Ingestion Layer)
- Data Processing
- Playbooks Engine
- Storage & Indexing
- Application Server & Client Console
- Remote Agents
- Logs & Error Management



Connectors (Ingestion Layer)

The connectors are the entry point for alerts into Siemplify. Their goal is to translate raw input data coming from multiple sources into Siemplify data. The connectors get alerts (or equivalent data – e.g. alarms, correlation events, TI hit-lists etc) from 3rd party tools and forward normalized data into the Data Processing layer. Siemplify platform provides out-of-the-box connectors for most popular security systems used today.

The component is based on in-house development framework that provides Python SDK to develop new connectors in a quick and easy way. The framework supports a variety of input data formats (CSV, JSON, XML, etc) and connection protocols (Files, RESTfull services, SysLog, etc).

The connector framework also provides a mechanism to filter noise data withing a time period (the Overflow Mechanism). This allows users to manage overflow alerts in an easier way.

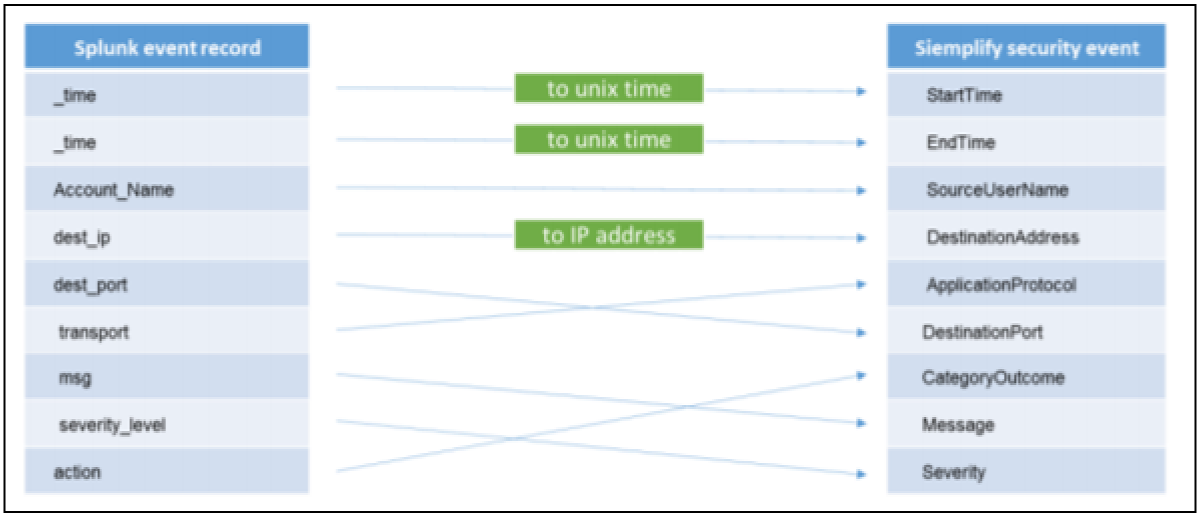
Key Points

- The connectors framework supports a variety of input formats (CSV, JSON, XML, etc) and connection protocols (Files, RESTfull services, SysLog, etc)
- Multiple connector instances can run in parallel to allow scaling out
- The framework and connector types can be extended with custom Python scripts
- The Overflow Mechanism – Helps manage noisy data with rule-based configuration
- The Connectors are managed directly from Siemplify console

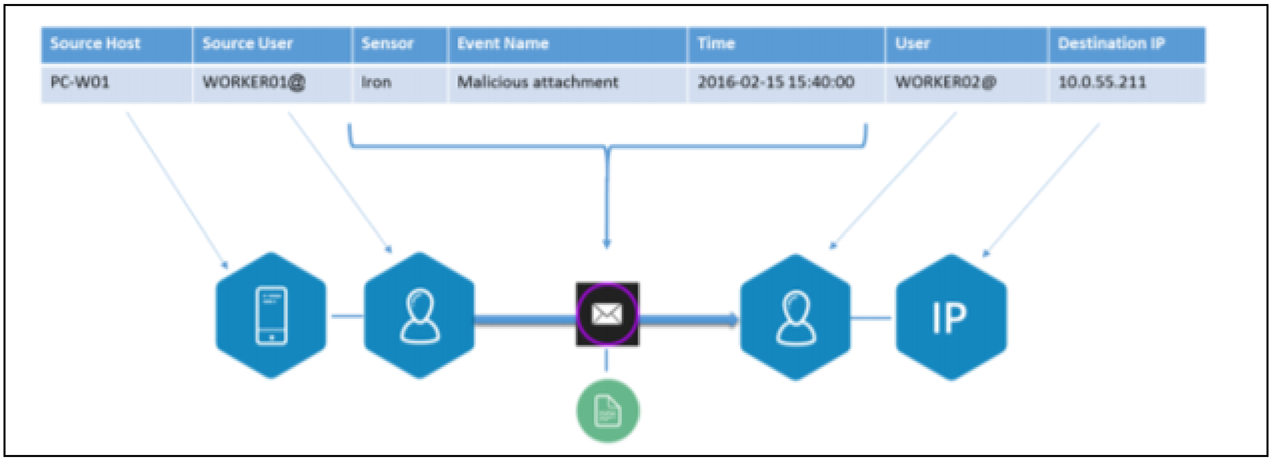
Data Processing

The magic behind Siemplify is the Data Processing Engine (DPE). This component analyzes, processes and aggregates data automatically to create Siemplify Cases. On a technical side – the Data Processing engine is designed as a parallel massive framework that parses raw data, enriches it, maps and models it into a graph structure, groups related Alerts into a Case and stores it in the database.
The DPE can be configured to suit any data type (Alerts, Enrichment etc.) or data source (Splunk, QRadar, ELK etc.) and provide flexibility into the pre-processing and processing stage.

The following example show how Splunk record fields are mapped into specific fields of the Siemplify data model.



Following that, the DPE will model the mapped fields into a graph representation that is stored in the database.



Source Host	Source User	Sensor	Event Name	Time	User	Destination IP
PC-W01	WORKER01@	Iron	Malicious attachment	2016-02-15 15:40:00	WORKER02@	10.0.55.211

Both data mapping and data modeling rules can be updated using the Siemplify Console.

An SDK is also available to develop new data processing actions. The SDK contains the data model, common methods for working with different data formats, transformation functions, etc. The DPE supports an internal dynamic data mapper that performs configurable mapping levels from a raw data model to Siemplify data models.

Multiple instances of DPE can run in parallel, either on a single or multiple nodes, to allow scaling out.

Playbooks Engine

The engine that powers Siemplify Orchestration and Automation was designed to automate tasks and playbooks on alerts or grouped cases.

The Playbooks engine runs in parallel, triggering playbooks according to user defined logic. Playbooks are attached to alerts, meaning that a case with 4 alerts might have 4 different playbooks running (one or more for each alert). All automation parts are executed at this stage and the results are pushed to the next module for storage. The steps of the playbook are executed in an isolated context to prevent unwanted or harmful actions to the system.

Multiple instances of the playbooks engine can run in parallel, either on a single or multiple Siemplify nodes, to allow scaling out.

Storage & Indexing

The Storage Layer is responsible for persistent storing of system data. It is based on the PostgreSQL server for operational data and Elastic Search Index for logging data. The PostgreSQL server holds metadata, operational and management data as follows:

- System settings
- Users details
- Environments details (multi-tenancy)
- Integrations details
- Orchestrations definitions
- Cases details
- Enrichment data
- Reports metadata
- Jobs metadata
- Dashboards metadata

The ELK stack will allow you to troubleshoot ingestion errors, data processing and playbook failures, remote agents and other system capabilities.

The storage layer can be scaled up by adding additional resources.

Application Server & Client Console

The Application Layer is responsible for providing all application logic for the Siemplify Console and performing analytics for ingested cases.
This layer contains the following modules:

- Analysis engine
- Case management
- Orchestration & automation
- Ad-Hoc query provider
- Response system provider
- Enrichment provider
- External case management sync provider
- Business intelligence dashboard
- Siemplify API for 3rd party clients
- Reporting tools

The application layer was designed with modern development methods like N-Tier architecture, SOA, low coupling patterns, etc. Users can choose to include or leave out components to meet deployment requirements.

Remote Agents

The Remote Agents module provides a secure way to connect a local Siemplify instance to remote sites. This provides MSSP and enterprise security operations centers with a variety of capabilities:

- Executing actions and playbooks on remote sites directly from Siemplify
- Pulling alerts and security data from remote sites with remote connectors
- Connecting to separate networks to pull data for incident response purposes

The Remote Agents infrastructure consists of 3 main components:

Siemplify Platform
Deployment of Siemplify platform to consolidate all security alerts in one place, and orchestrate security and network products with automated workflows.

Siemplify Publisher

A proxy component that receives and holds commands from Siemplify Platform. The publisher accepts only incoming communication from Siemplify platform and Siemplify Agents. The Publisher is used to transfer data in a secure way without any direct access to the remote site.

Siemplify Agent

A lite agent deployed on the remote site. The agent pulls new tasks from the Publisher, executes locally (on the remote\separate network) and updates the Publisher with the results.
The agent is easily distributed, which allows MSSP end customers deploy it by themselves.
The agent uses only outgoing communication to the publisher.

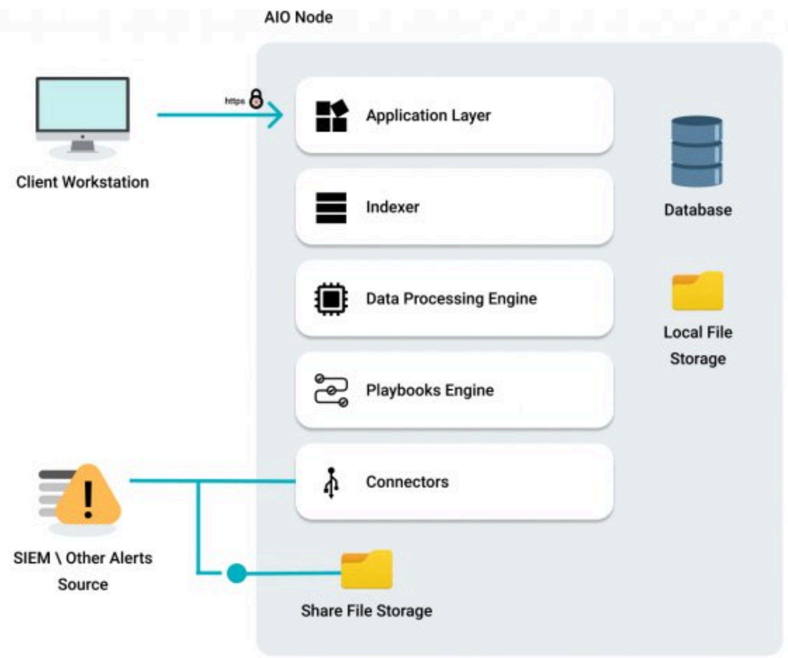
Logs & Error Management

Siemplify is deployed with the ELK stack to aggregate and manage the logs from all modules.
The ELK indexing engine and dashboards provide an easy way to troubleshoot modules, research errors and obtain visibility into module operations.

Deployment Options

Single Node Deployment

Siemplify’s standard system deployment uses a single All-In-One (AIO) node, which has all system modules on one machine.
With this deployment method, the system can ingest up to 15k alerts per day (using the recommended hardware requirements).

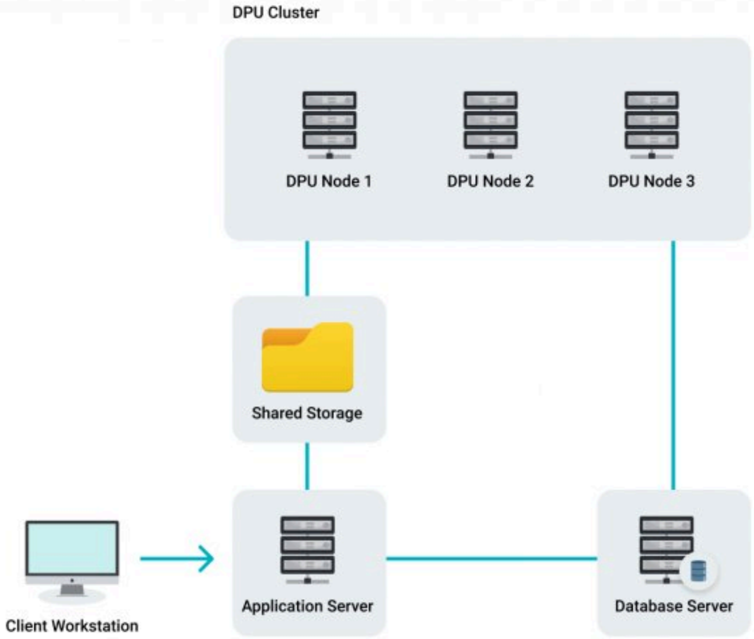


Multi-Node Deployment

The AIO deployment supports ingestion of up to 15K Alerts per day. If a higher ratio is required (alerts \ day) the system can be deployed with additional Data Processing nodes. Each data processing node add up to

7.5K Alerts per day.

When the system is deployed with multiple Data Processing nodes, the database should be installed on a separate server. The maximum amount of Data Processing nodes that can be added to one main node instance is 3. This deployment requires using a shared folder that is accessible for all system servers.

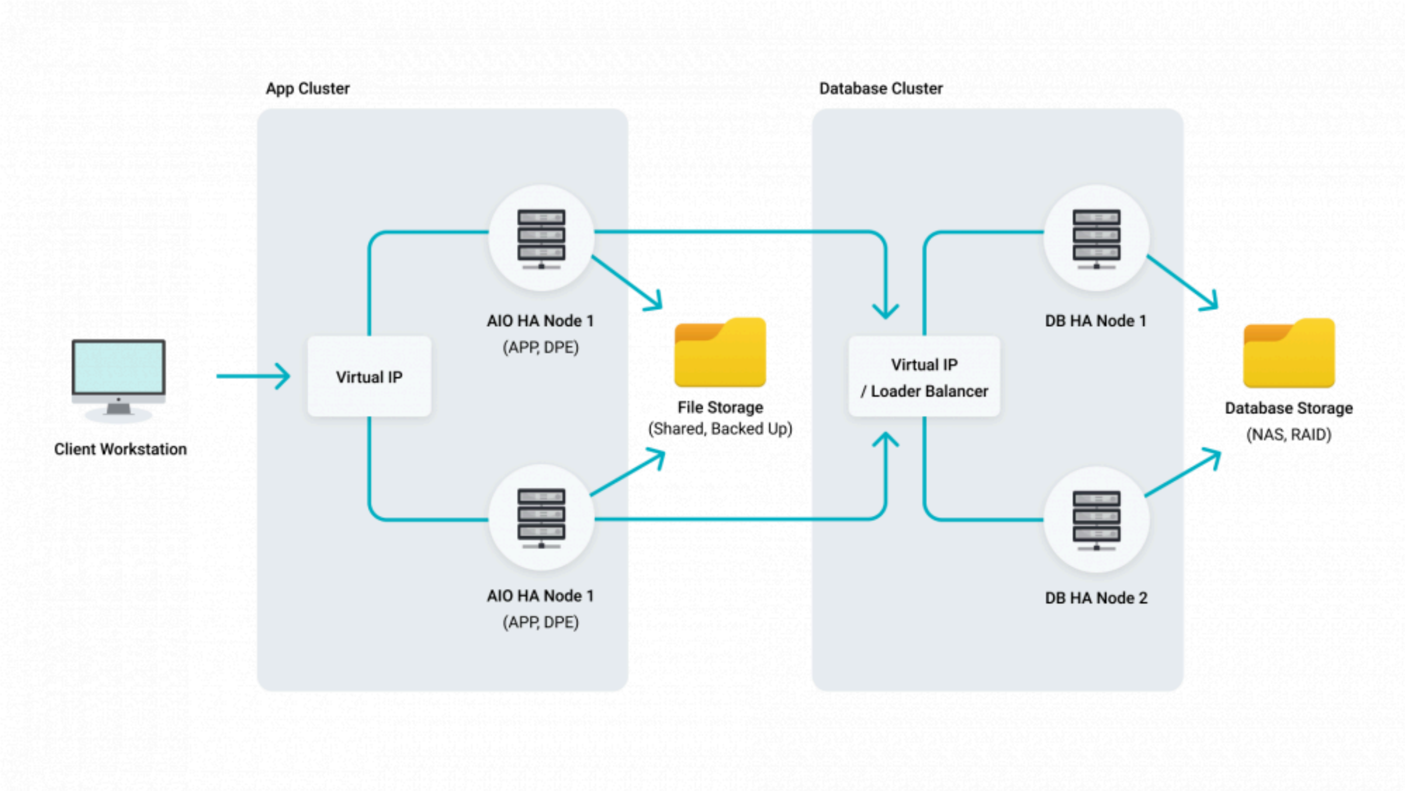


High Availability Deployment

Siemplify provides multiple deployment modes with high-availability cluster to provide the continued availability of the services. The cluster works in an active-passive configuration allowing automatic activation of Siemplify on another node If it has failed for some reason (e.g hardware failure).

The high availability solutions is based on the official guidelines of CentOS 7.5 and PostgreSQL:

- PostgreSQK Guidelines Link
- CentOS 7.5 Guidelines Link1 Link2 Link3



Scaling Strategy

In order to satisfy the scaling needs of SOCs who deal with exceptional amounts of data, Siemplify supports scale out architecture by deploying system modules on separate servers. This makes it possible to combine system components in different deployment combinations and customize deployment by customer requirements.

The system can be scaled by adding additional nodes allowing Siemplify to distribute the ingestion process between multiple Data Processing nodes.

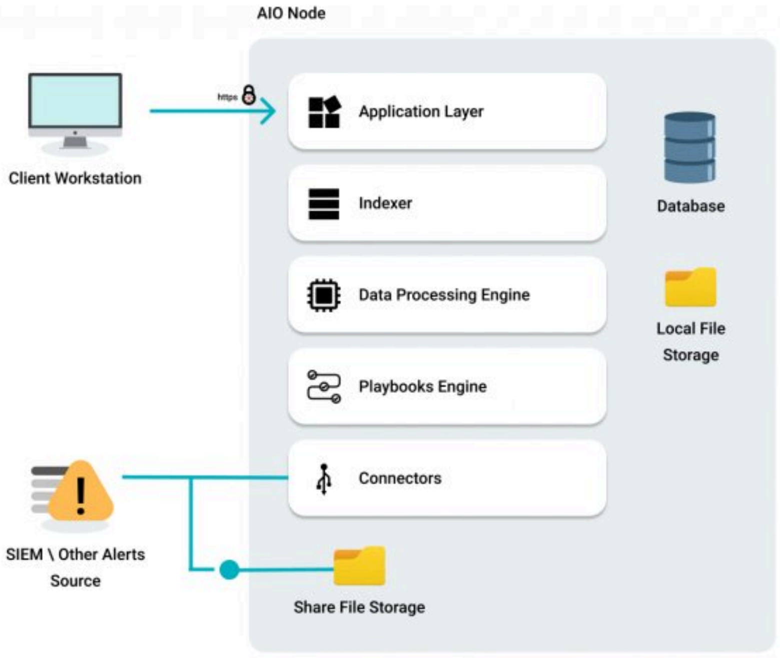
Siemplify Requirements

Hardware Requirements

The following describes the requirements needed to deploy Siemplify system. The requirements apply to Single-Node (AIO), Multi-Node (Scale) and High Availability Siemplify deployments.

Single Node (AIO)

Single Node Diagram



Siemplify Server (AIO Node)

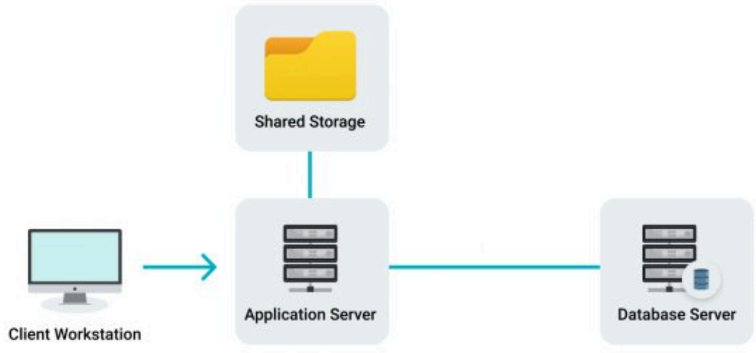
CPU	2 virtual sockets with 6 cores each (12 cores)
RAM	16 GB (minimum) / 32 GB (recommended)
Storage	800 GB
Storage	Disk Type: SSD / SAS 10k / Similar High-Speed Storage
Virtual NIC	E1000 Adapter
Supported ESX Version	6.0 and higher (for OVA deployment)
Supported Virtual Machine Version	11 and higher (for OVA deployment)
Open Ports	443, 80 (redirect), 5432 (db PostgreSQL), 5601 (Kibana), 9200 (Elastic)

Client Workstation

Network	100 MBPS Ethernet (or higher)
Monitor Resolution	1920×1080 (Full HD) or 1366×768
Browser	Google Chrome 66.0.3359 or higher
Open ports	443

Multi-Node (External Database)

Multi Node Diagram



Siemplify Server (App Server)

CPU	2 virtual sockets with 6 cores each (12 cores)
RAM	32 GB
Storage	450 GB
Storage	Disk Type: SSD / SAS 10k / Similar High-Speed Storage
Virtual NIC	E1000 Adapter
Open Ports	443, 80 (redirect), 5601 (Kibana), 9200 (Elastic)

Database Server

used when the database is external

CPU	2 virtual sockets with 4 cores each (8 cores)
RAM	32 GB
Storage	150 GB system + 350 GB data
Storage	Disk Type: SSD / SAS 10k / Similar High-Speed Storage
Virtual NIC	E1000 Adapter
Open Ports	5432 (db PostgreSQL)

Shared Storage

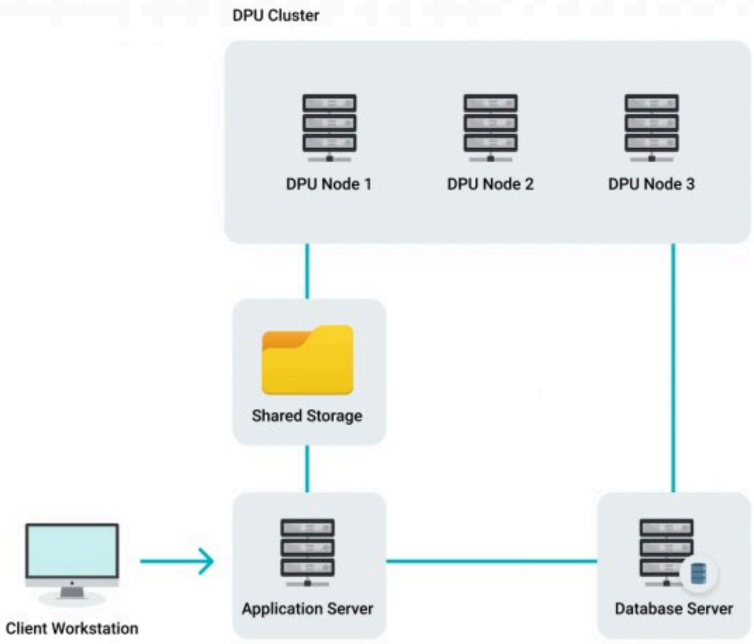
Storage	150 GB
Disk Type	SSD / SAS 10k / Similar High-Speed Storage
Configuration	shared, backed up by customer
Connectivity	SMB
Open Ports	Storage

Client Workstation

Network	100 MBPS Ethernet (or higher)
Monitor Resolution	1920×1080 (Full HD) or 1366×768
Browser	Google Chrome 66.0.3359 or higher
Open ports	443

Multi-Node (Scale)

Multi Node Diagram



Siemplify Server (App Server)

CPU	2 virtual sockets with 6 cores each (12 cores)
RAM	32 GB
Storage	450 GB
Storage	Disk Type: SSD / SAS 10k / Similar High-Speed Storage
Virtual NIC	E1000 Adapter
Open Ports	443, 80 (redirect), 5601 (Kibana), 9200 (Elastic)

Data Processing (DPU Node)

CPU	2 virtual sockets with 2 cores each (4 cores)
RAM	16 GB

Storage	250 GB
Storage	Disk Type: SSD / SAS 10k / Similar High-Speed Storage
Virtual NIC	E1000 Adapter
Open Ports	None

Database Server

used when the database is external

CPU	2 virtual sockets with 4 cores each (8 cores)
RAM	32 GB
Storage	150 GB system + 350 GB data + 100 GB for each additional DPU node
Storage	Disk Type: SSD / SAS 10k / Similar High-Speed Storage
Virtual NIC	E1000 Adapter
Open Ports	5432 (db PostgreSQL)

Shared Storage

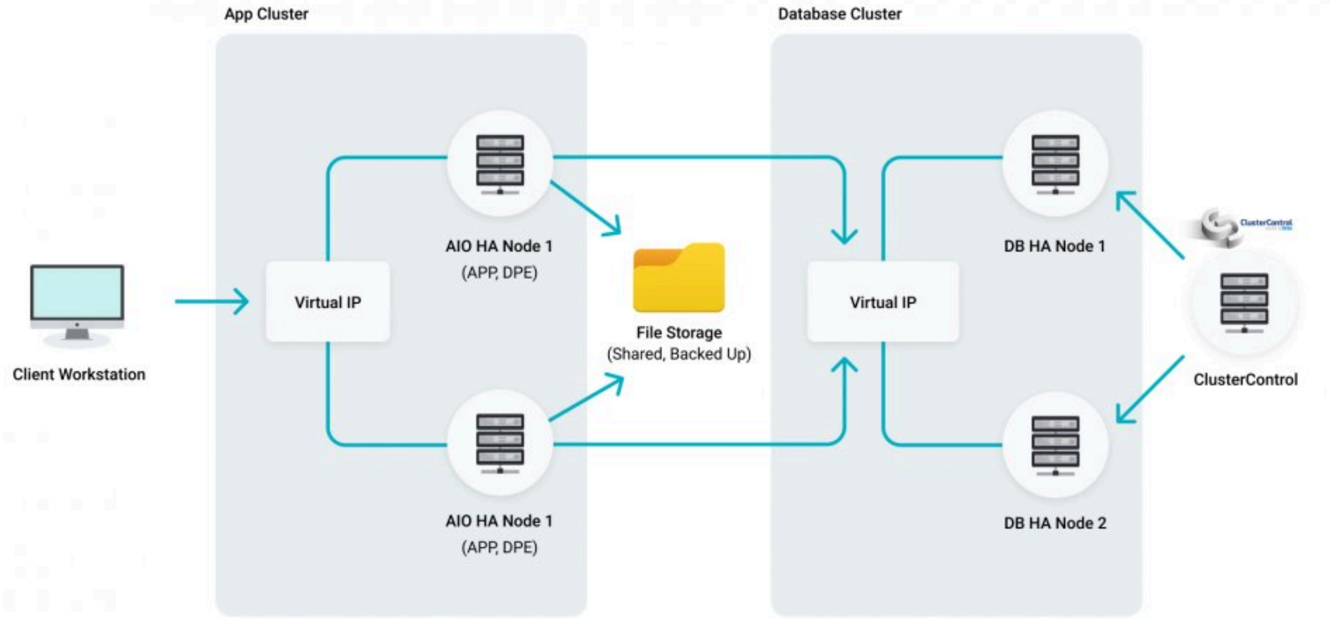
Storage	150 GB
Disk Type	SSD / SAS 10k / Similar High-Speed Storage
Configuration	shared, backed up by customer
Connectivity	SMB
Open Ports	Storage

Client Workstation

Network	100 MBPS Ethernet (or higher)
Monitor Resolution	1920×1080 (Full HD) or 1366×768
Browser	Google Chrome 66.0.3359 or higher
Open ports	443

High Availability

HA App Server & DB Server Diagram



Siemplify Server (2 x AIO Nodes)

CPU	2 virtual sockets with 8 cores each (16 cores)
RAM	32 GB
Storage	450 GB
Storage	Disk Type: SSD / SAS 10k / Similar High-Speed Storage
Virtual NIC	E1000 Adapter
Open Ports	443, 80 (redirect), 5601 (Kibana), 9200 (Elastic)

Database Node (2 x DB Nodes)

CPU	2 virtual sockets with 4 cores each (8 cores)
RAM	32 GB

Storage	150 GB system + 350 GB data
Disk Type	SSD / SAS 10k / Similar High-Speed Storage
Connectivity	Shared storage access
Open Ports	5432 (db PostgreSQL), Storage

ClusterControl Node

CPU	2 cores
RAM	2 GB
Storage	20 GB system
Disk Type	SSD / SAS 10k / Similar High-Speed Storage
Connectivity	Database access
Open Ports	443, 80 (client access)

File Storage

Storage	150 GB
Disk Type	SSD / SAS 10k / Similar High-Speed Storage
Configuration	shared, backed up by customer
Connectivity	SMB
Open Ports	Storage

Publisher

CPU	4 cores (minimum) / 8 cores (recommended)
RAM	8 GB / 16 GB
Storage	100 / 200 GB for disk
Network	E1000 Adapter
Connectivity	Web access
Ports	443

Remote Agent

CPU	4 cores (minimum) / 8 cores (recommended)
RAM	8 GB / 16 GB
Storage	100 GB for disk
Connectivity	Web access \ Publisher access
Network	E1000 Adapter

Prerequisites

Integrations

To properly configure the integrations between Siemplify and your security products, please be sure to provide the credentials \ API keys required to access them.
In the Siemplify console, navigate to Marketplace to see what information is required for each integration.
In addition, network \ web access should be tested (from Siemplify machine to the security product) prior to the configuration step.

Proxy

To use a web proxy, first make sure Siemplify machine has network access to the proxy.
Then, in the Siemplify Console, navigate to Settings > Advanced > General to set up the proxy.

Web Access

Internet access from Siemplify machine is required to allow Siemplify Installer download packages from the online repository.

Shared storage access

Make sure to prepare accessible shared storage in case your mode of deployment requires that.

Multi-Tenancy

The Siemplify platform enables MSSPs to seamlessly manage disparate technologies, permissions, reporting and playbooks across their entire client base from a single pane of glass. Using multi-tenant deployment, security teams are able to:

- Consolidate customer alerts into a single queue
- Establish incident response for generic and customer specific use cases
- Run commands and playbooks in client environments via remote execution
- Manage customer integrations
- Investigate threats across the entire customer base
- Measure and broadcast customer metrics with Reports and Dashboards
- Configure customer specific SLA, custom lists, email templates, logos and more
- Manage views and data separation
- Run multi-site deployments

> ✻ In Siemplify, tenants are defined by Environments. These appear across the entire platform to help users easily focus the Console on a specific client.

Multi-Tenancy Features

Siemplify uses Environments to manage tenants. Each environment that represents a tenant \ customer is created with a set of metadata fields – customer image, customer name, description, contact name, phone and email and Siemplify Remote Agents configuration. In addition, the following capabilities are provided by Siemplify for additional value in a multi-tenant deployment:

Environment Operational Settings
The following settings are configured per environment to help with customer specific use-cases in daily operation: SLAs, custom lists, customer domains and networks, email templates, blacklisted items.

Connectors
Connectors are applications that ingest alerts from different types of sources (SIEM, Database, Email box etc.) into Siemplify. Multiple connectors can run in parallel collecting alerts from local or remote products, and assigning them automatically to the relevant environment.
Connectors can also take into consideration the multi-tenancy defined in the source product (e.g. multi-tenant QRadar SIEM).

Data Separation

Ingested and collected data (Cases, Alerts, Events, Playbook Results etc.) is separated into environments. Each environment will contain data relevant to the customer, without any possibility for data moving to another environment. Data assigned to an environment will be visible to permitted users only.

Data Consolidation

All data is consolidated in a single queue with the same language for the SOC team (analyzed processes) – regardless of the source product -

Easier to onboard new customers (just switching the connector) and new security analysts (they don't need to be experts in products).

Support more customers with different types of technologies (EK, Splunk, AlienVault etc.)

Entity Explorer

Security teams can view entities across the entire customer base or within the context of a specific environment (e.g. see if a malicious hash found on "Customer A" also appeared on "Customer B" site.)

User Permissions

Along with module permissions, users can also be assigned to the environments they can view or handle. Customers can also get limited user access to Siemplify to review dashboards, reports, playbooks etc. with their relevant information alone.

Marketplace

Integrations are defined per environment.

Playbooks

Extend the playbooks to customer remote sites, to allow security analysts (who have sufficient privileges) collect information and run IR processes on customers environment. Security teams can create generic playbooks (which can automatically pick the integration credentials relevant to the customer) and customer specific playbooks as-well.

Remote Agents

Siemplify platform has the ability to orchestrate and automate workflows on remote \ separated networks. This ability allows MSSPs to extend the use cases between thier SOC and the customer.

Dashboards

Dashboards can be customer specific or generic.

In any case, it is always possible to filter a dashboard by environment.

Reports

Reports can be customer specific or generic. Siemplify provides periodic reports that can be generated automatically for different customers and purposes (e.g. weekly SLA, attacks statistics etc.) It is also possible to add customer logos to reports.