How-To on Product Deployment and Operation for Cloud Providers

Kaspersky Security for Virtualization Light Agent

Table of Contents

1 Kaspersky Security for Virtualization Light Agent

Kaspersky Security for Virtualization Light Agent (hereinafter also referred to as “the solution”, “the application”, and “KSVLA”) is an integrated solution that provides comprehensive protection of virtual machines against various types of information security threats, network attacks, and fraud. The solution contains the latest technologies and provides multi-layered protection of virtual and hybrid infrastructures. Kaspersky Security for Virtualization Light Agent makes it possible to achieve level of protection and consolidation of virtual machines in a virtual infrastructure, and supports the latest technologies from major vendors of virtualization and cloud solutions: VMware vSphere, Microsoft Hyper-V, Citrix Hypervisor, KVM, OpenStack, Clouds, etc.

KSVLA includes a dedicated virtual machine (SVM) where malware databases are located. The SVM scans fragments of files sent by Light Agents installed on virtual machines for viruses and other malware. Thanks to smart scan optimization, for example, by using a shared cache and eliminating superfluous information, the SVM reduces the volume of data processed and the number of actions performed, significantly decreasing the number of I/O operations per second, CPU instructions, and requirements for RAM and disk space. This makes it possible to achieve high consolidation coefficients, making investments in virtualization projects more profitable.

The application is managed through Kaspersky Security Center (KSC), using management plug-ins.

Integration Server, an application component, is used for virtual machines to get information about the virtual infrastructure and the SVM's address.

KSC is used as a proxy server for interacting with Kaspersky services related to licensing, using KSN, updating application databases and modules. This architecture makes it possible to create a virtual infrastructure that is isolated from external networks.

Learn more:

https://www.kaspersky.com/small-to-medium-business-security/virtualization-hybrid-cloud

https://www.kaspersky.com/small-to-medium-business-security/virtualization-light-agent

https://support.kaspersky.com/KSVLA/5.2/en-US/145134.htm

2 Glossary

KSC (Kaspersky Security Center) – A tool for centralized management of a comprehensive protection system.

SVM – Secure virtual machine. A virtual machine on a hypervisor where the Protection Server, a component of KSVLA, is installed.

Light Agent – A component of KSVLA. It is installed on each virtual machine that needs to be protected.

Integration Server – A component of KSVLA. It facilitates interaction between KSVLA components and the virtual infrastructure.

Integration Server Console – A component of KSVLA. A graphical interface for managing the Integration Server.

MMC plug-in – A KSVLA component that provides an interface for managing KSVLA through KSC Administration Console.

Web plug-in – A KSVLA component that provides an interface for managing KSVLA through KSC Web Console.

Multitenancy – A mode in which one provider-installed instance of the application serves multiple tenants.

vKSC (Virtual Administration Server) – A KSC component for managing the network of the tenant organization.

3 Application architecture

3.1 Distribution kit

The distribution kit includes:

- ksvla-components_X.X.X.X_mlg.exe for starting the installation wizard for MMC plug-ins (for managing KSVLA through KSC Administration Console), Integration Server, and Integration Server Console. The wizard is also used to unpack the files necessary to install Light Agent for Windows and Light Agent for Linux.
- SVM images (secure virtual machine images) with an installed Protection Server and other software necessary for running SVMs. These are provided as several archives for various types of hypervisors. Each archive contains a ksvla-svm_manifest_X.X.X.X.xml image description file for the SVM Management Wizard as well as the image itself in the format required by the hypervisor:
    - ksvla-svm_microsoft-hyper-v_X.X.X.X_mlg.vhdx for deployment on Microsoft Windows Server (Hyper-V).
    - ksvla-svm_citrix-hypervisor_X.X.X.X_mlg.xva for deployment on Citrix Hypervisor.
    - ksvla-svm_vmware-vsphere_X.X.X.X_mlg.ova for deployment on VMware ESXi.
    - ksvla-svm_kvm_based_X.X.X.X_mlg.qcow2 for deployment on KVM, Proxmox VE, R-Virtualization, HUAWEI FusionCompute CNA, and Nutanix AHV.
- KSVLA-MIB.txt, a MIB file which you can use to get SVM status using an SNMP-based monitoring system.
- Archives for installing web plug-ins (for managing KSVLA through KSC Web Console):
    - ksvla-web_plugin_wla_X.X.X.X_mlg.zip for installing the web plug-in for Light Agent for Windows.
    - ksvla-web_plugin_lla_X.X.X.X_mlg.zip for installing the web plug-in for Light Agent for Linux.
    - ksvla-web_plugin_svm_X.X.X.X_mlg.zip for installing the web plug-in for the Protection Server.

3.2 Application components

SVM

Protection Server, a KSVLA component, is provided as an SVM image (secure virtual machine image). Select the image that corresponds to the virtualization platform being used. The Protection Server installed on an SVM performs the following functions:

- Checks for threats in the files sent by Light Agents;
- Receives application database and module updates from the storage of the Kaspersky Security Center Administration Server;
- Manages license keys and licensing restrictions.

Light Agent

Light Agent, a KSVLA component, is installed on each virtual machine that needs to be protected. The Light Agent performs the following functions:

- Protects the host virtual machine from viruses and other threats.
- Controls operation of applications and devices on the protected virtual machine, and monitors changes in the virtual machine's operating system.

Integration Server

Integration Server, a KSVLA component, facilitates interaction between KSVLA components and the virtual infrastructure. The Integration Server is used for performing the following tasks:

- Deploys, removes, and reconfigures SVMs.
- Receives information about the protected infrastructure and sends this information to SVMs.
- Sends Light Agents the list of SVMs available to connect.
- Deploys and uses KSVLA in multitenancy mode.

Depending on the scenario, you can use the REST API or Integration Server Console to manage the Integration Server.

Management plug-ins (MMC and web plug-ins)

Management plug-ins provide an interface for managing KSVLA through Kaspersky Security Center. You can use KSC Administration Console or KSC Web Console.

Learn more:

https://support.kaspersky.com/ksvla/5.2/en-US/67380.htm

3.3 Hardware and software requirements of KSVLA

KSVLA's hardware and software requirements are described in detail in the help: https://support.kaspersky.com/KSVLA/5.2/en-US/64743.htm

KSC's hardware and software requirements are described in detail in the help: https://support.kaspersky.com/KSC/13.2/en-US/96255.htm

In most scenarios, Light Agent for Windows requires 200–300 MB of RAM. In most scenarios, Light Agent for Linux requires 150–200 MB of RAM.

When starting on-demand scan tasks (including on a schedule), more RAM may be required, but it will be freed when the scan is complete. The amount of required RAM may also increase if malware activity or malicious attempts to encrypt data are detected.

KSVLA algorithms optimize when scans and updates are started, automatically distributing the load across Light Agents to prevent task storms.

Learn more:

https://support.kaspersky.com/KSVLA/5.2/en-US/67380.htm

https://support.kaspersky.com/KSVLA/5.2/en-US/145134.htm (KSVLA)

https://support.kaspersky.com/KSC/13.2/en-US/3396.htm (KSC)

3.4 Network interaction

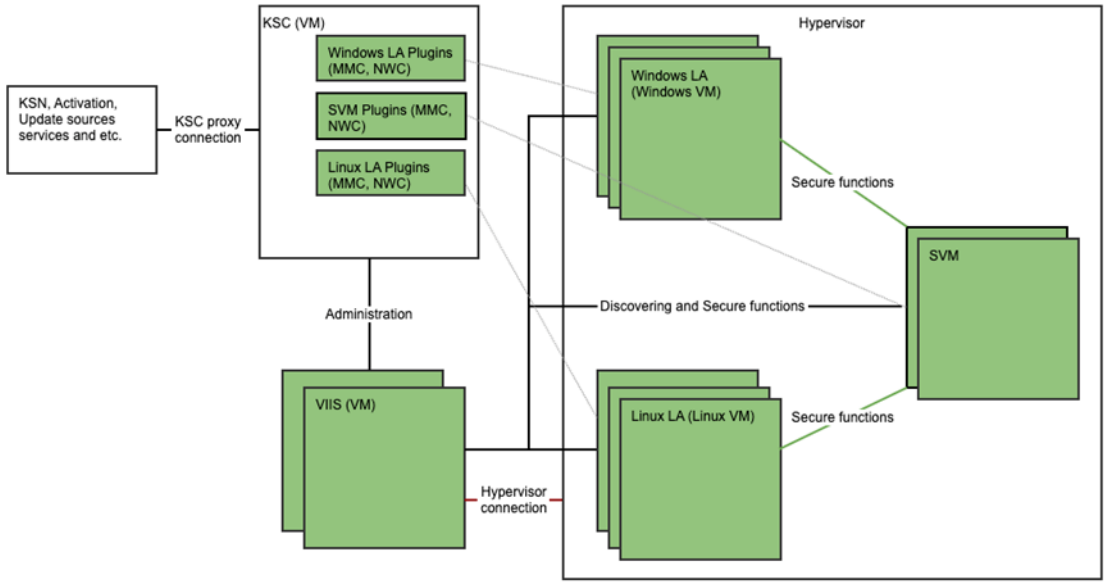
For the solution to fully work, you need to provide the following interactions:

- Interaction of an SVM with the virtual machines it protects.
- Interaction of SVMs and Light Agents with the Integration Server.
- Interaction of SVMs and Light Agents with KSC.
- To use the solution in multitenancy mode, interaction of the Integration Server with KSC.
- Interaction of the Integration Server with hypervisors or virtualization tools.

Configuring ports for KSVLA is described in detail in the help:
https://support.kaspersky.com/KSVLA/5.2/en-US/133882.htm

Configuring ports for KSC is described in detail in the help:
https://support.kaspersky.com/KSC/13.2/en-US/158830.htm

Accounts for the installation and operation of the solution are described in detail in the help:
https://support.kaspersky.com/KSVLA/5.2/en-US/85889.htm



Isolating the virtual infrastructure from external networks:

KSC is used as a proxy server for interacting with Kaspersky services related to licensing, using KSN, updating application databases and modules. This architecture makes it possible to isolate the virtual infrastructure from external networks:

- To prevent KSC from contacting activation servers, KSVLA needs to be activated using key files, not activation codes.
- To prevent KSC from contacting KSN servers (for more details, see https://www.kaspersky.com/ksn), you need to use Kaspersky Private Security Network (KPSN) (for more details, see https://www.kaspersky.com/enterprise-security/private-security-network). You can also disable use of KSN.
- To prevent KSC from contacting update servers, you need to use Kaspersky Updater Utility (KUU) (for more details, see https://support.kaspersky.com/updater4). The utility, which downloads database and module updates from a specified update source, can be used on an isolated node, and KSC can be configured to be used as an update source on this node.

Encrypting connections between Agents and SVMs:

If you need to secure the channels used by KSVLA to transfer data between SVMs and Light Agents, you can configure a secure connection using SSL encryption. Keep in mind that using encryption to secure connections may reduce the performance of KSVLA.

Learn more:

https://support.kaspersky.com/KSVLA/5.2/en-US/102192.htm

4 Recommended procedure for deploying protection

1. Install Kaspersky Security Center (for more details, see https://support.kaspersky.com/KSC/13.2/en-US/171268.htm). Administration Server must have access to all protected virtual machines in order to deploy and manage Light Agent. You can install Administration Server on a separate virtual machine or on a physical server. For more details, see the KSC help: https://support.kaspersky.com/KSC/13.2/en-US/3396.htm
2. Install KSVLA management components: MMC plug-ins (plug-in for Protection Server, plug-in for Light Agent for Windows, plug-in for Light Agent for Linux), Integration Server, and Integration Server Console. Integration Server can be installed separately from KSC (on a physical or virtual machine). The wizard is used to perform the installation. The installation wizard for installing management components also creates Light Agent installation packages for the Windows and Linux operating systems and deploys them to the protected virtual machines.
3. Create KSC policies for Light Agents (https://support.kaspersky.com/KSVLA/5.2/en-US/74263.htm, https://support.kaspersky.com/KSVLA/5.2/en-US/103975.htm) and SVMs (https://support.kaspersky.com/KSVLA/5.2/en-US/129817.htm).
4. Deploy the required number of SVMs (deployment options, which depend on the basic scenarios for the infrastructure, are described below). As part of the deployment process, you need to configure network settings for the deployment. The SVMs' network connection must meet the basic requirements of the Protection Server:
    - a connection with Kaspersky Security Center (for getting updates, policies, and tasks, and for event forwarding);
    - a connection with the Integration Server;
    - a connection with all protected virtual machines.
5. Create an activation task for SVMs (https://support.kaspersky.com/KSVLA/5.2/en-US/86947.htm).
6. Install Light Agents on the protected virtual machines. If you configured automatic movement of virtual machines with installed Light Agents, then after the Light Agents are installed the virtual machines will be moved to the KSC administration group with the KSC policy created in the previous steps. The KSC policy propagates the required settings for a Light Agent's connection with an SVM. As a result, upon connecting to an SVM, a Light Agent automatically receives updates, license information, and certain protection settings.

5 Scenarios for using KSVLA in multitenancy mode

In multitenancy scenarios, the provider organization provides virtualization services for tenant organizations. The provide wants to ensure that its virtual machines are protected, but cannot install a separate KSC, Integration Server, and SVM for each tenant due to excessive resource consumption. KSVLA offers its own approaches to automated deployment and tenant protection management in multitenancy mode.

The following scenarios are supported for using the application in multitenancy mode.

5.1 "Complete tenant"

"Complete tenant" is the main scenario for multitenancy mode. In this scenario, tenant protection is structured using REST API requests to the Integration Server. This scenario relies on providing a virtual KSC Administration Server (hereinafter referred to as "vKSC"; for more details, see https://support.kaspersky.com/KSC/13.2/en-US/92246.htm) to each tenant.

Thanks to the use of a vKSC, the "Complete tenant" scenario possesses several advantages:

- It saves tenant resources, because there is no need to install a new KSC and Integration Server for each tenant.
- A tenant gains the ability to use functionality similar to an ordinary KSC. Each vKSC has its own group of unassigned devices, its own selection of reports, selection of devices, events, installation packages, rules for moving virtual machines, etc.
- Tenants are isolated from one another. They can only work as part of their own vKSC.

Under this scenario:

- For each tenant, a provider creates its own isolated vKSC.
- A provider can enable and disable protection of its tenants with a single request to the Integration Server's REST API. For example, a provider can disable protection if a tenant fails to pay for protection services.
- A tenant works with its virtual machines as part of its own vKSC, and can manage protection of its virtual machines using a Light Agent policy.
- A provider can get reports to monitor its tenants' use of virtual machine protection. These reports can be used to bill tenants.

5.2 "Simple tenant"

This scenario is used if a provider has already configured multitenancy mode and wants to add the ability to get reports on its tenants' virtual machine protection without changing the application's configuration. The provider's infrastructure can be set up based on a vKSC or administration groups.

Under this scenario:

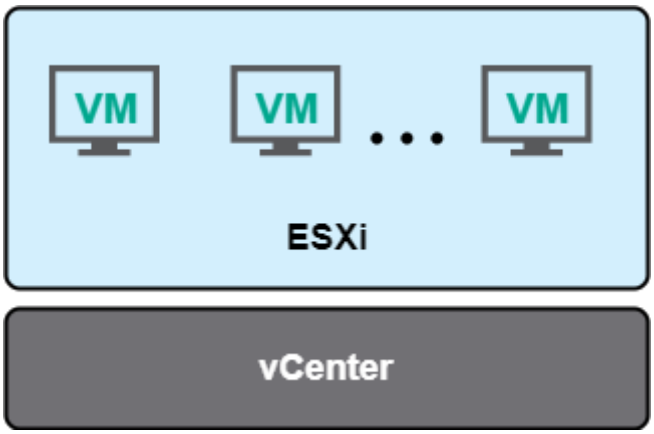
- The provider registers tenants and their virtual machines on the Integration Server**.**
- If protection services are not paid for, the provider can disable a tenant's protection, but this does not happen automatically in this scenario:
    - The provider disables a tenant's protection without using REST API requests, for example, through a KSC policy.
    - The provider sends a REST API request to the Integration Server to communicate that the tenant's protection is disabled.
    - The Integration Server stops monitoring protection of all of this tenant's registered machines and adds information about the tenant's protection to reports.
- A provider can get reports to monitor its tenants' use of virtual machine protection. These reports can be used to bill tenants.

6 Scenarios for deploying KSVLA

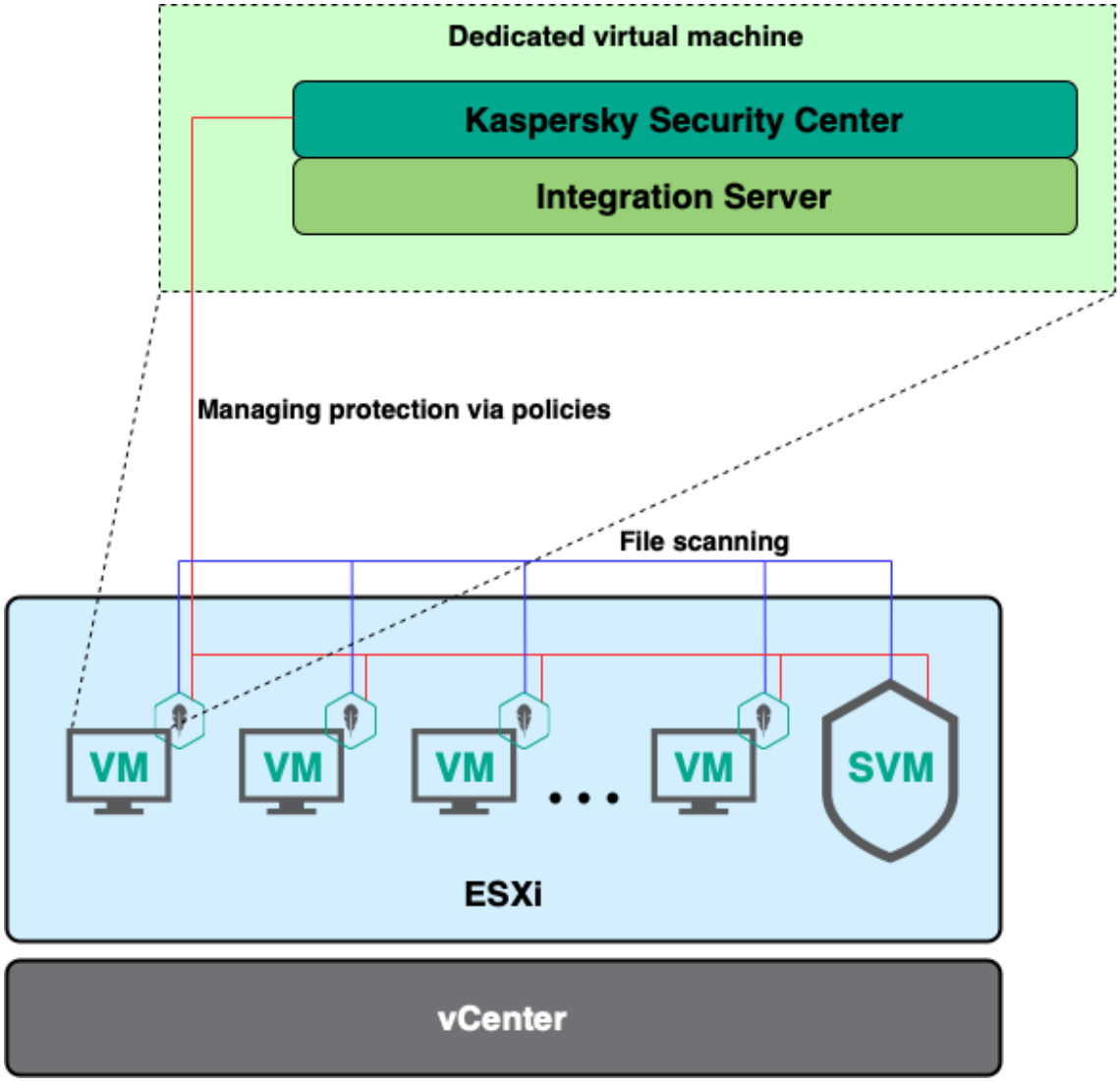
Let’s consider deployment scenarios by looking at various user infrastructures. As an example, we’ll take an infrastructure built on VMware vSphere with a vCenter Server for managing the virtual infrastructure.

Basic scenario

The basic scenario is when the user has a single ESXi hypervisor in vCenter.



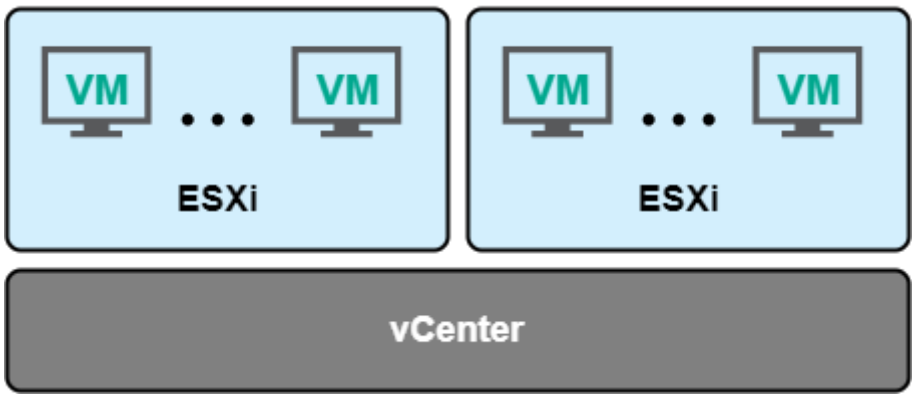
In this case, the user allocates a single virtual machine for managing protection, and installs KSC and Integration Server on this machine. The user deploys SVMs on the hypervisor according to the basic estimate of 80 Light Agents per 1 SVM (*see the note below). On all machines that need to be protected, the user installs a Light Agent.



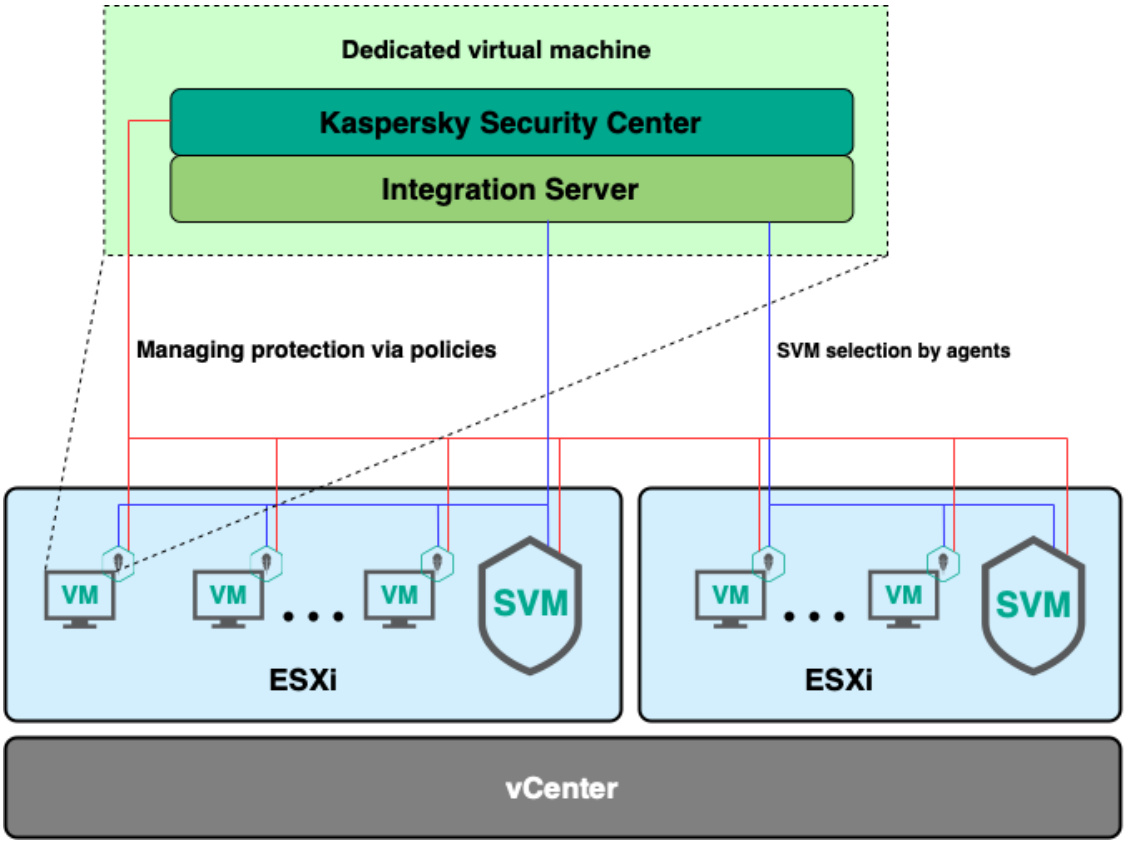
*Note: It is important to understand that the "basic estimate" of the number of Light Agents on an SVM can vary from "160 to 1" to "40 to 1", depending on the activity and homogeneity of the virtual machines protected by any given SVM. The estimate of "80 to 1" is suitable for the overwhelming majority of tenants, although there are also tenants for which "200 to 1" creates no apparent problems based on available monitoring artifacts.

Standard scenario

The most common example of user infrastructure is when the user has several hypervisors and fewer than 10,000 virtual machines.

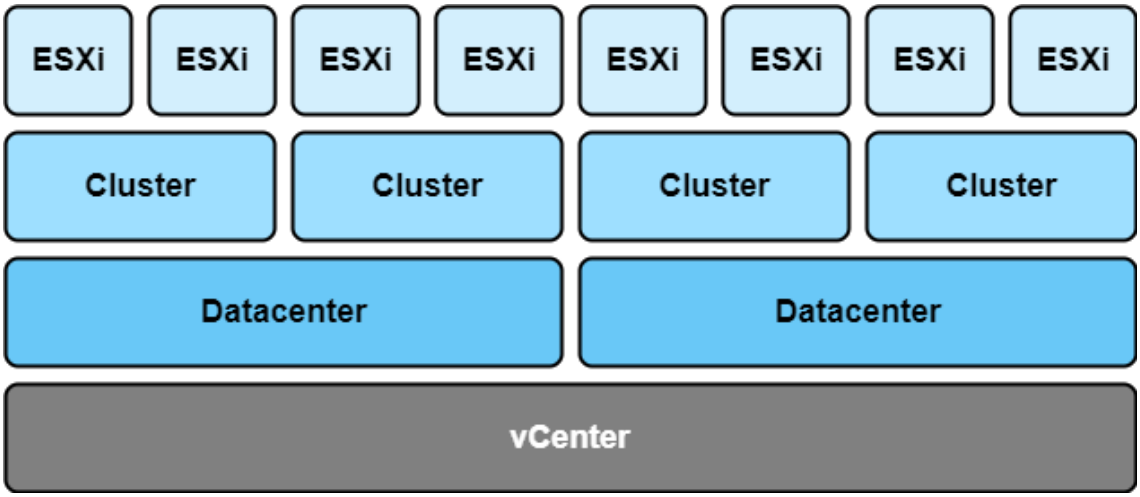


In this case, the user also allocates a single virtual machine on one of the hypervisors and installs KSC and Integration Server on it. The user deploys an SVM on each hypervisor and installs Light Agent on the virtual machines. In this scenario, it makes sense to use the standard SVM selection algorithm. With the standard algorithm, the Light Agent chooses to connect to the SVM deployed on the same hypervisor. If there are no SVMs available to connect to on the hypervisor, the Light Agent selects the SVM with the lowest number of connected Light Agents, regardless of where the SVM is located.



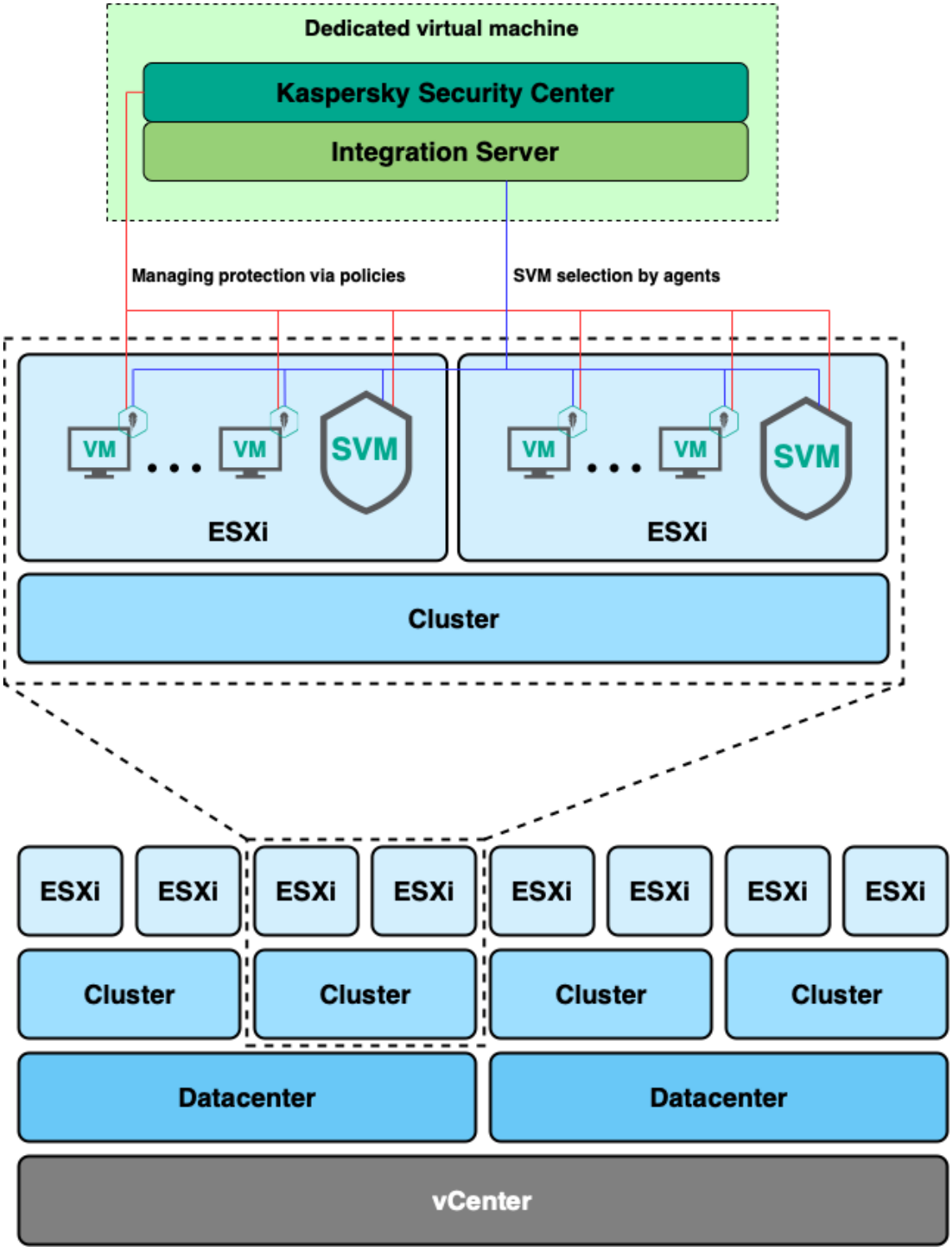
Scenario for large infrastructures

Let's consider a more complex clusterization scenario, or a scenario for large infrastructures (10,000+ virtual machines). In this case, the user has a large number of hypervisors joined into clusters, which in turn may be joined into a data center.



Given such a infrastructure, the user should use the advanced SVM selection algorithm. If this algorithm is applied, the user can specify in the Light Agent policy how Light Agents should

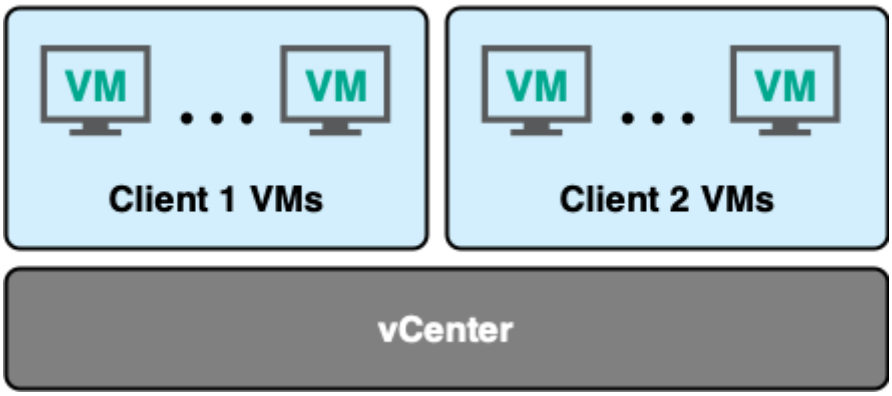
account for the SVM path. For example, a Light Agent may connect to an SVM deployed in the same hypervisor cluster.



The advanced SVM selection algorithm can balance the load on SVMs. For example, if one of the ESXi hypervisors has few virtual machines and an SVM's capacity is standing idle, the SVM can protect virtual machines on a different ESXi located in the same cluster, reducing the load on a different SVM.

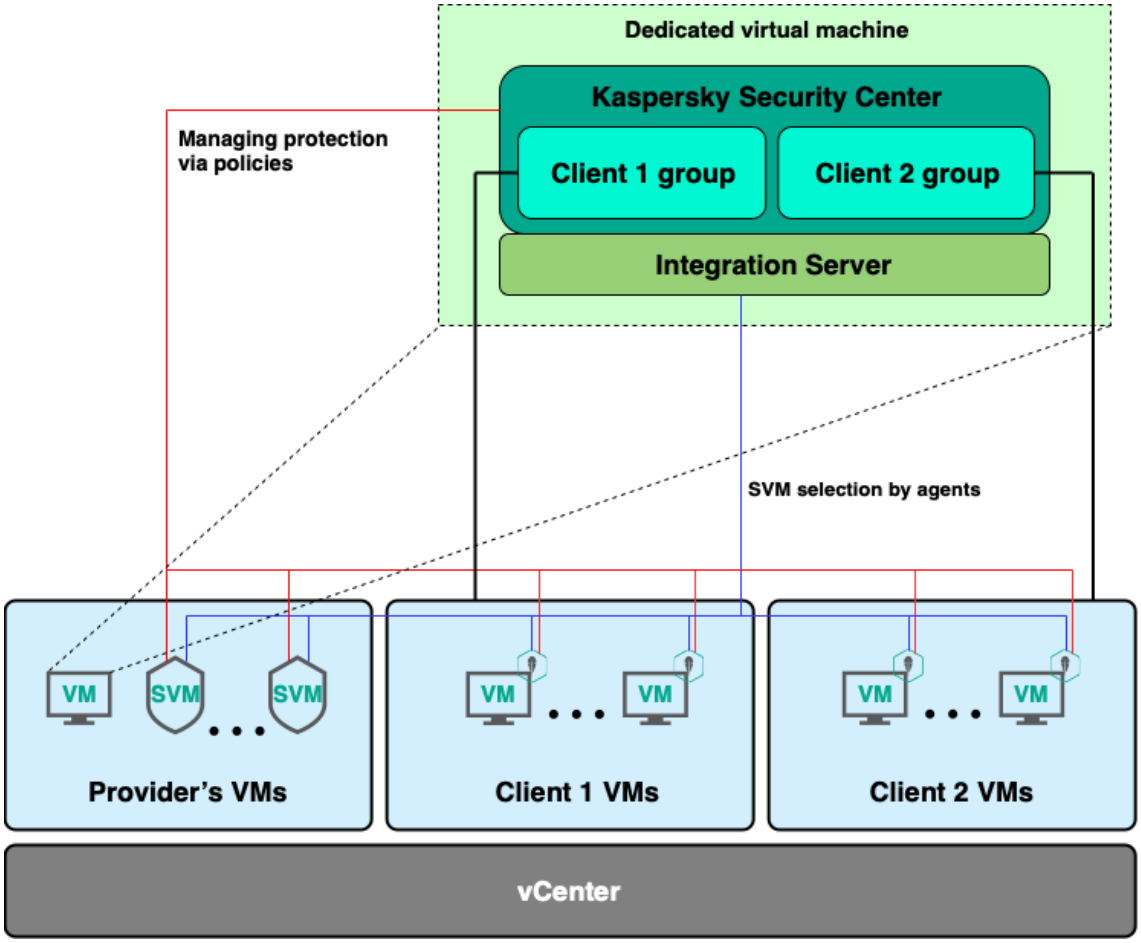
Scenario for a SaaS provider

Let’s consider the scenario where the user provides virtualization services on its own vCenter (software as a service). The provider sells access to virtual machines deployed on its vCenter to multiple tenants. Moreover, a tenant's virtual machines can be deployed on different ESXi hypervisors. The tenant may know nothing about the vCenter infrastructure.

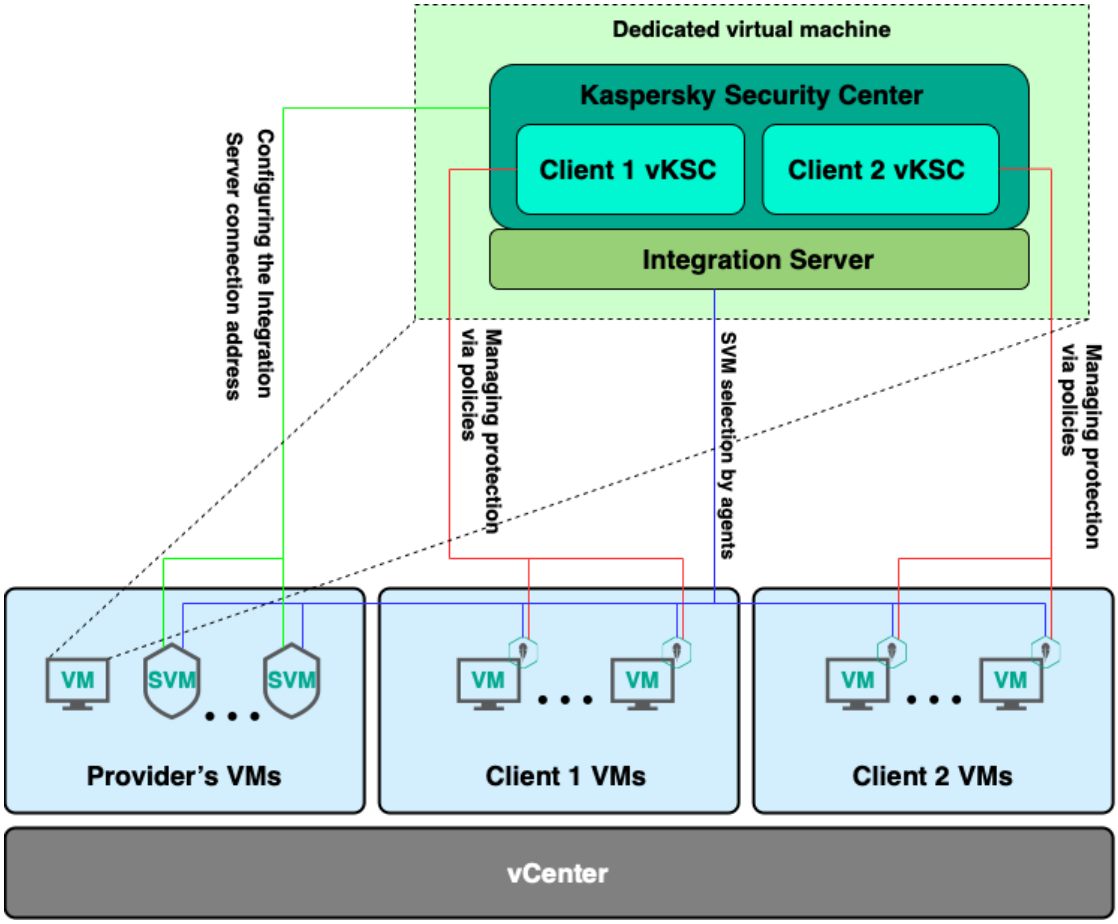


The provider wants to use KSVLA to protect its tenants’ virtual machines. Depending on the tenants’ specific characteristics and requirements, there are various ways to do this.

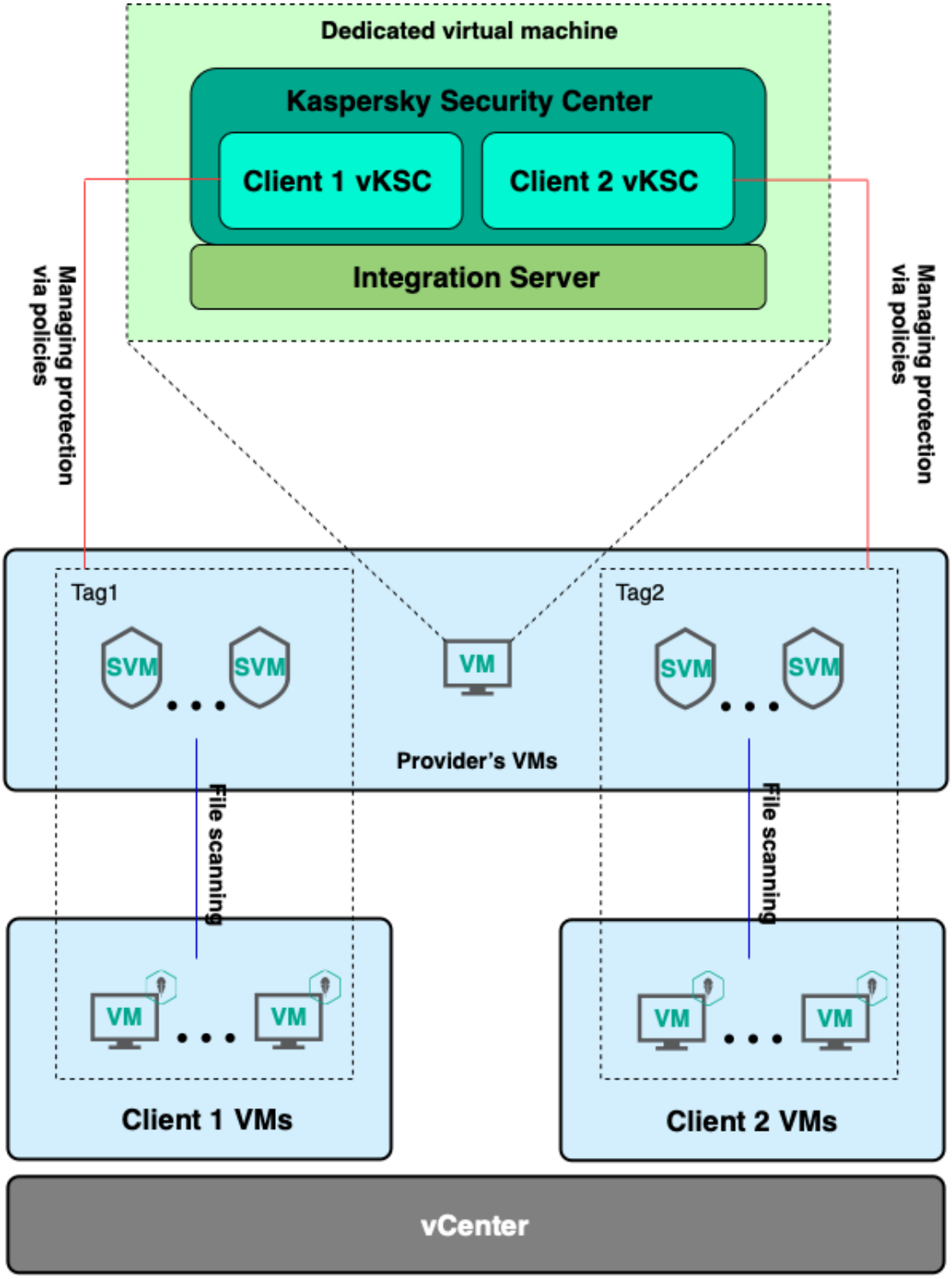
- The simplest option is that the provider manages its tenants’ protection entirely on its own. The provider deploys KSC and Integration Server on a machine that only the provider has access to. In KSC, a separate service group is created for each tenant, where its virtual machines will be registered. The provider installs Light Agents on the tenants’ virtual machines and deploys the appropriate number of SVMs. SVMs can be deployed on each hypervisor where the virtual machines are, or on a separate hypervisor. If the provider charges for protecting tenants’ virtual machines and wants to get reports on how long each of a tenant's virtual machines was protected, the provider can use KSVLA in multitenancy mode under the "Simple tenant" scenario.



- If the tenant wants to be able to configure the protection of its own machines, the provider can provide each tenant with its own vKSC. To automate creation of vKSCs for each tenant, the provider can use KSVLA in multitenancy mode under the "Complete tenant" scenario. In this case, the provider also installs KSC and Integration Server, and deploys SVMs, while the tenant itself can install Light Agents through its vKSC. As in the previous option, the provider can get reports on the protection of its tenants.



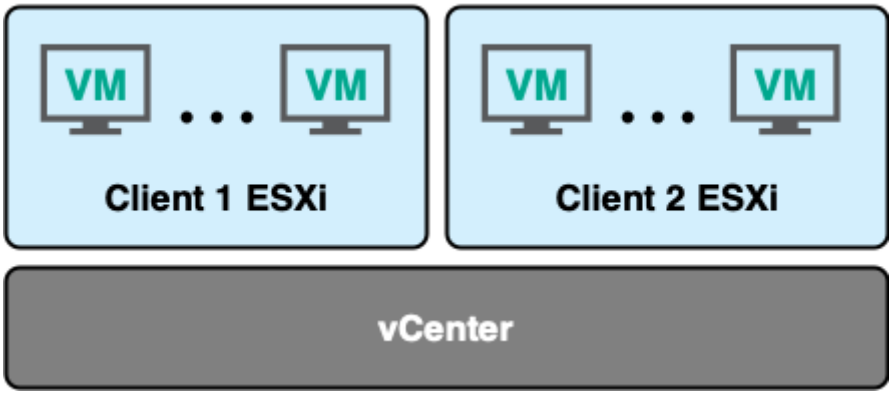
In vKSC option, the provider can also provide a selection of SVMs that protect only that tenant's virtual machines. To do this, the provider should use tags assigned in policies for SVMs and Light Agents. A Light Agent will only connect to SVMs that share the same tag as the Light Agent.



It is important to differentiate the tags used in KSVLA from KSC device tags (for more details, see https://support.kaspersky.com/KSC/13.2/en-US/166115.htm). In KSVLA, tags are assigned only in policies for SVMs and Light Agents.

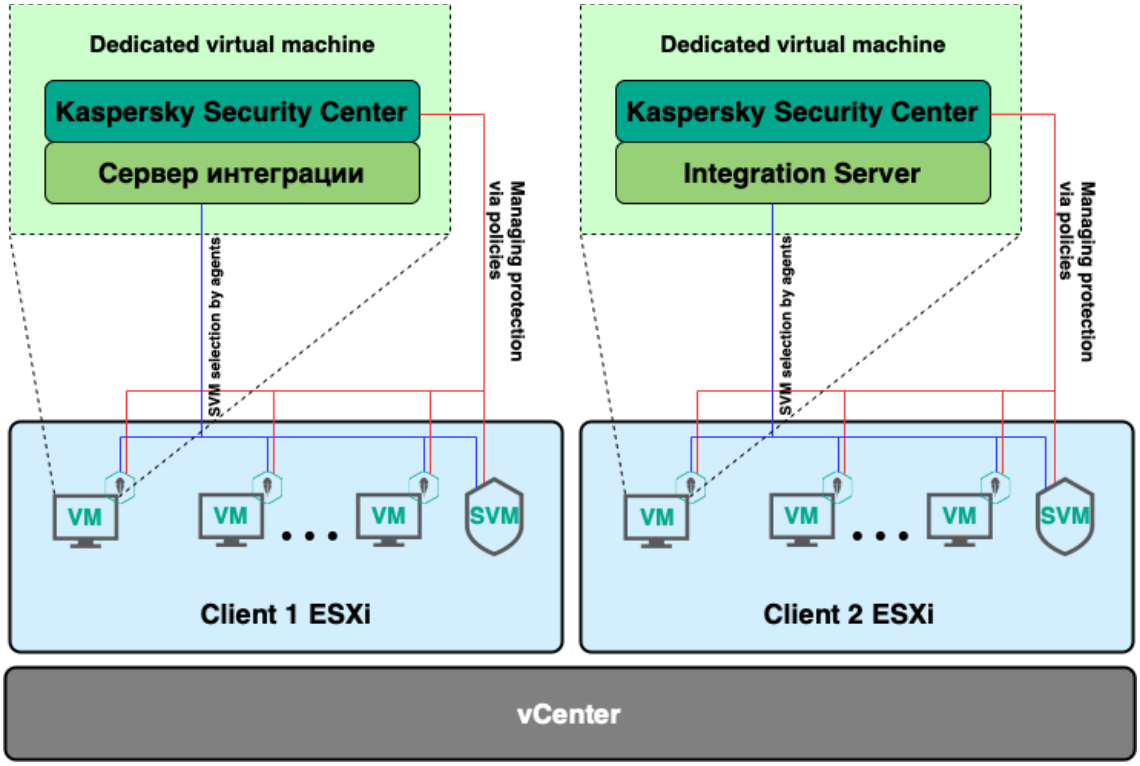
Scenario for a service provider (IaaS provider)

The user may also be virtualization service provider who provides infrastructure as a service (Iaas). In this case, it does not simply provide a tenant with access to virtual machines, but provides the tenant with one or more ESXi hypervisors on which the tenant itself can deploy virtual machines on its own.

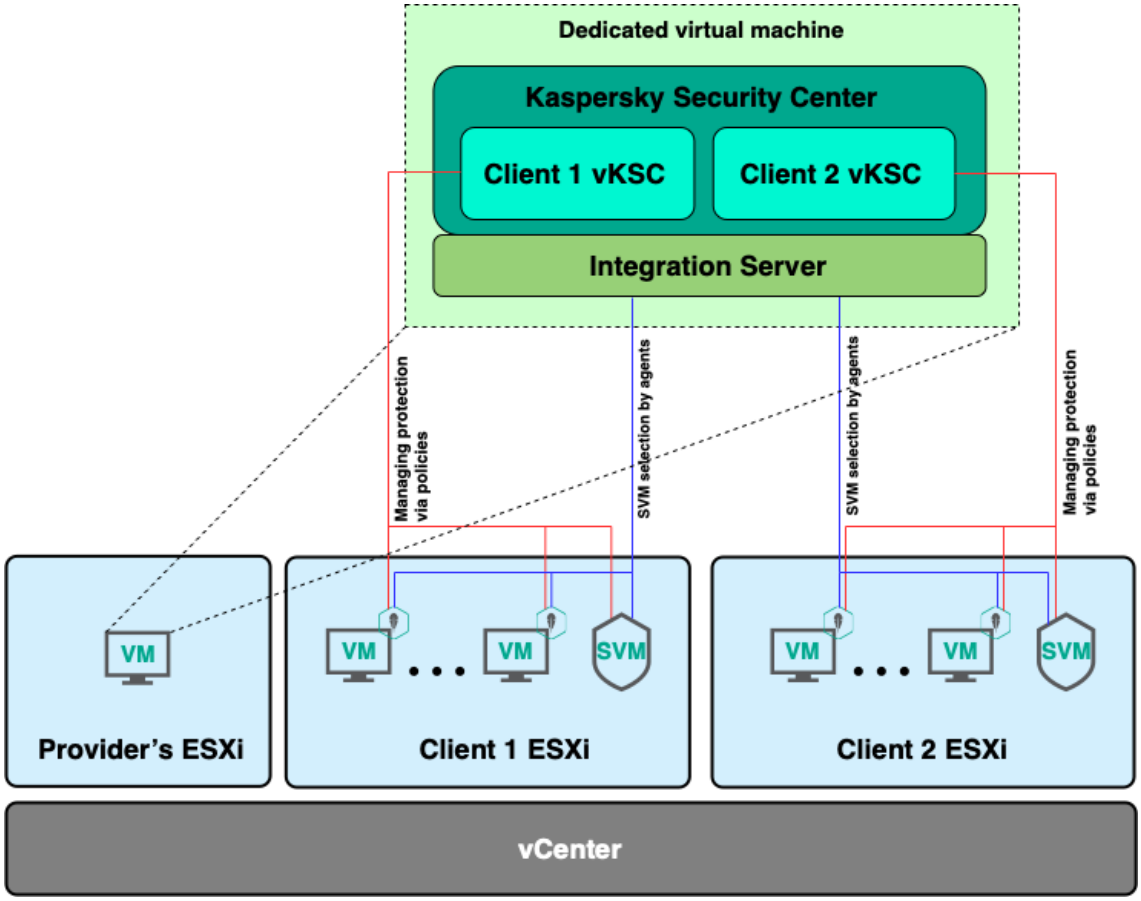


In this scenario, the provider also has several ways to protect its tenants.

Because tenants fully control their ESXi hypervisors, they can independently deploy their own KSC and Integration Server and deploy their own SVMs without using KSVLA's multitenancy mode.



If the provider wants to sell virtual machine protection services to tenants, it can use KSVLA in multitenancy mode under the "Complete tenant" scenario. In this scenario, it is simpler to provide each tenant with its own selection of SVMs. It is enough to configure the advanced SVM selection algorithm. This saves the tenant's resources, because the tenant does not install its own KSC. Additionally, all of the tenant's protected machines interact with SVMs only within the tenant's network, making it possible to not encrypt the connections between Light Agents and SVMs.


Dedicated virtual machine
Kaspersky Security Center
Client 1 vKSC
Client 2 vKSC
Integration Server
Managing protection via policies
SVM selection by agents
SVM selection by agents
Managing protection via policies
VM
Provider's ESXi
VM
VM
SVM
Client 1 ESXi
VM
VM
SVM
Client 2 ESXi
vCenter

7 "Complete tenant" scenario

Using KSVLA in multitenancy mode under the "Complete tenant" scenario assumes that the Integration Server's REST API will be used to structure tenants' protection. When the provider creates a complete tenant, the Integration Server creates a service group with the indicated tenant's name. In this service group, for each tenant the Integration Server creates a virtual Administration Server (vKSC) as well as an account that the tenant can use to connect to its own vKSC.

A tenant's protection is managed by different methods depending on the version of the application.

- In version 5.2, the Integration Server manages a tenant's protection. When Light Agents installed on a tenant's virtual machines contact the Integration Server for a list of available SVMs, the Integration Server checks the tenant's protection status:
  - If the tenant's protection status is enabled, then the Light Agent returns a list of SVMs available to connect. The Light Agent connects to one of them, and the protection of the virtual machine begins.
  - If the tenant's protection is disabled, then the Light Agent returns a non-existent, fictitious SVM. The Light Agent cannot connect to it, and the virtual machine is not protected.
- In version 5.1 of the application, protection is managed by policies that enable and disable protection. For more details, see https://support.kaspersky.com/ksvla/5.1/en-US/199684.htm.

Let's consider in greater detail the requests to the Integration Server's REST API, which are used to structure tenants' protection.

7.1 Creating a new tenant

Use the following request to create a new complete tenant:

POST /api/2.0/virtualization/tenants

```
<tenant>
    <!-- The tenant name is used for vKSC and the group -->
    <name>{name}</name>
    <description>{description}</description>
    <userData>
        <!-- more detailed description of the tenant -->
        <![CDATA[{additional information}]]>
    </userData>
    <preferredViisAddress>{IP address of the Integration
Server}</preferredViisAddress>
    <type>Complete</type>
    <vKsc>
        <!-- tenant's user for connecting to vKSC -->
        <user>
            <name>{administrator name}</name>
            <!-- the password must be encoded in Base64 -->
            <password>{administrator password}</password>
        </user>
    </vKsc>
</tenant>
```

Creating a tenant is a relatively long operation, so it is performed asynchronously and the user receives an XML response with the created task.

```
<task id={ID} created={date and time of creation} stateChanged={dated and
time of status change} changed={date and time of progress change}>
    <!-- for example, Created, Running, Completed, Failed -->
    <state>{state}</state>
    <!-- in this case, CreateTenant -->
    <type>{type}</type>
    <!-- for example, CreateVKsc, CreateVKscUser, CreateDbEntries -->
    <stage>{stage}</stage>
    <progress>{execution progress}</progress>
    <!-- contains the task result if the task completed successfully -->
    <result>{result}</result>
    <!-- contains an error description if the task completed with an error
-->
    <error>{error message}</error>
</task>
```

The task ID can be used in a request to get the current status of a task and to wait for it to complete

GET /api/2.0/virtualization/tasks/{task ID}

After the task is complete, the result field contains information about the created tenant:

```
<task id={ID} created={date and time of creation} stateChanged={dated and
time of status change} changed={date and time of progress change}>
    <state>Completed</state>
    <type>CreateTenant</type>
    <progress>100</progress>
    <!-- contains the task result if the task completed successfully -->
    <result>
        <tenant id="{tenant ID}" created="{date and time of creation}"
updated="{date and time of update}">
            <name>{tenant name}</name>
            <vKsc id="{vKSC ID}">
                <user>
                    <name>{administrator name}</name>
                </user>
            </vKsc>
            <status>Inactive</status>
        </tenant>
    </result>
</task>
```

7.2 Getting tenant information

Use the following requests to get information about a single tenant based on its tenant ID, or about all tenants all at once:

GET /api/2.0/virtualization/tenants/{tenant ID}

GET /api/2.0/virtualization/tenants

The response contains the information indicated by the provider when creating the tenant:

```
<tenant id="{ID}" created="{date and time of creation}" updated="{date and
time of update}">
    <name>{tenant name}</name>
    <description>{description}</description>
    <userData>
        <![CDATA[{additional information}]]>
    </userData>
    <type>Complete</type>
    <status>{status (Active/Inactive)}</status>
    <vKsc id="{vKSC ID}">
        <user>
            <name>{administrator name}</name>
        </user>
    </vKsc>
</tenant>
```

A tenant is created with the “Inactive” status, so the status field initially indicates “Inactive”.

7.3 Removing a tenant

Use the following request to remove a tenant:

DELETE /api/2.0/virtualization/tenants/{tenant ID}?removeTenantArtifacts={true|false}

The removeTenantArtifacts parameter indicates whether various tenant artifacts in KSC must be removed: tenant group, vKSC, account for connecting to vKSC.

7.4 Registering a virtual machine

To get reports on the protection of its tenants, a provider needs to register the tenants’ virtual machines on the Integration Server using the following request, which must contain the machine's BIOS ID:

POST /api/2.0/virtualization/tenants/{tenant ID}/vms/register

```
<vm biosId="{BIOS ID}">
     <name>{name}</name>
     <userData><![CDATA[{additional information}]]></userData>
</vm>
```

7.5 Getting information about protected virtual machines

To get information about the virtual machines registered for a tenant, use the following request.

GET /api/2.0/virtualization/tenants/{tenant ID}/vms

The response contains a list of the tenant's machines:

```
<vms>
    <vm id="{identifier in the database}" biosId="{BIOS ID}">
        <name>{name}</name>
        <userData><![CDATA[{additional information}]]></userData>
    </vm>
<vms>
```

7.6 Unregistering a virtual machine

To unregister a tenant's virtual machine, use the following request, which must contain the tenant ID and machine ID:

POST /api/2.0/virtualization/tenants/{tenant ID}/vms/unregister?vmId={ID}

7.7 Enabling protection of a tenant

To enable protection, activate the tenant using the following request:

POST /api/2.0/virtualization/tenants/{tenant ID}/activate

The response contains a ChangeTenantActivation task.

When a tenant's protection is enabled, the following actions take place:

- The protection-enabling policies applied to Light Agents version 5.1 are activated.
- The tenant's status in the Integration Server database changes to "Active".
- The Integration Server starts returning to Light Agents version 5.2 a list of SVMs available to connect.

7.8 Disabling protection of a tenant

To disable protection, a tenant must be deactivated using the following request:

POST /api/2.0/virtualization/tenants/{tenant ID}/deactivate

The response contains a ChangeTenantActivation task.

When a tenant's protection is disabled, the following actions take place:

- The protection-disabling policies applied to Light Agents version 5.1 are activated.
- The tenant's status in the Integration Server database changes to "Inactive".
- The Integration Server starts returning a fictitious SVM to Light Agent version 5.2.

After protection is disabled, we recommend that the provider isolate the tenants' machines from SVMs by configuring network settings to prevent the tenant from connecting Light Agents to SVMs without authorization.

7.9 Working with reports

To get a report, use the following request:

POST /api/2.0/virtualization/reports/tenants?tenantId={tenant ID}&from={date and time}&to={date and time}

The query string may include the following optional parameters:

- tenantId – the identifier of the tenant for which a report will be generated.
- from, to – the time interval for which a report will be generated. Times need to be indicated in UTC format, e.g. 2022-08-11T11:54:20.

After the task returned in the response is completed, the result field will contain a report ID. Use this ID in the following request to get the report itself:

GET /api/2.0/virtualization/reports/tenants/{report ID}

The response is in CSV format with the following columns:

- Tenant identifier
- Tenant name
- virtual machine identifier;
- virtual machine name;
- date and time when protection was enabled;
- date and time when protection was disabled.

The report indicates the periods in which tenants' virtual machines were protected. The first two columns indicate information about the tenant. The next two indicate information about the virtual machine. The last two indicate the time period in which the machine was protected. All time periods are indicated in UTC format.

8 "Simple tenant" scenario

Using KSVLA in multitenancy mode under the "Simple tenant" scenario assumes that the provider has already configured the structure of the tenant's protection in multitenancy mode. The protection can be structured using administration groups, into which the tenants' machines are placed, as well as vKSCs. This scenario is applied for the provider to automatically get reports on the protection of tenants' virtual machines.

As in the "Complete tenant" scenario, the provider needs to register its tenants and their virtual machines on the Integration Serve. However, in this scenario, no entities are created in KSC. Information about tenants is only added to the Integration Server database.

Use the following requests to apply this scenario.

8.1 Creating a new tenant

A simple tenant is created using the same request that is used to create a complete tenant. The difference is that the body of the request indicates a "Simple" tenant type, and no vKSC user is indicated.

POST /api/2.0/virtualization/tenants

```
<tenant>
    <!-- tenant name that will be stored in the database -->
    <name>{name}</name>
    <description>{description}</description>
    <userData>
        <![CDATA[{additional information}]]>
    </userData>
    <type>Simple</type>
</tenant>
```

8.2 Enabling protection of a tenant

Use the following request to enable protection of a tenant:

POST /api/2.0/virtualization/tenants/{tenant ID}/activate

The response contains a ChangeTenantActivation task.

When the protection of a simple tenant is enabled, only the tenant's protection status in the Integration Server database changes (to "Active"). After the status changes, information about the tenant's protection begins to be added to reports on tenants' protection.

8.3 Other requests

Other requests for working with a simple tenant, e.g. registering a virtual machine, getting tenant information, getting reports, etc. are not different than the requests described above for working with a complete tenant.

9 Examples of deploying KSVLA in multitenancy mode on user infrastructures

Let's consider a KSVLA deployment in multitenancy mode on infrastructure built on VMware vSphere.

9.1 "Complete tenant" scenario

Let's consider the example of a provider's user infrastructure that does not have KSVLA.

The provider provides virtualization services on its vCenter Server, on which virtual machine are deployed for its tenants. The provider wants to use KSLVA to protect tenants' virtual machines and also give tenants the ability to manage protection.

1. The provider's administrator begins the deployment by installing Kaspersky Security Center, which will be shared by all tenants.

The KSC Administration Server must have access to all protected virtual machines for deployment and management of Light Agents. You can install Administration Server on a separate virtual machine or on a physical server. For more details, see the KSC help: https://support.kaspersky.com/KSC/13.2/en-US/3396.htm

2. The provider's administrator installs the KSVLA management components that are shared by all.

The management components are installed using the component installation wizard (ksvla-components_X.X.X.X_mlg.exe, which is included in the distribution kit). When the wizard is done, Integration Server, Integration Server Console, and MMC plug-ins for Integration Server and Light Agent are installed:

- Kaspersky Security for Virtualization <version number> Light Agent – Protection Server
- Kaspersky Security for Virtualization <version number> Light Agent for Linux
- Kaspersky Security for Virtualization <version> Light Agent for Windows

3. If the provider wants to get reports on tenant protection, on the Integration Server the provider's administrator enables collection of data on tenant protection. To do this, the administrator sets the EnableTenantsProtectionReports parameter to "true" in the Integration Server configuration file (%ProgramFiles(x86)%\Kaspersky Lab\Kaspersky VIISLA\viislaservice.exe.config) and restarts the Integration Server after restarting the VIISLA service.

4. The provider's administrator connects the Integration Server to KSC using Integration Server Console. This is done in the "Kaspersky Security Center connection settings" section by specifying the address, login, and password for connecting to KSC.

5. The provider's administrator adds its tenants to the Integration Server database.

To create tenants, the administrator sends the REST API request described above (POST /api/2.0/virtualization/tenants) to the Integration Server. After the tenants are added, the provider's administrator transfers access to the created vKSCs (address and account login/password) to the tenants' administrators. A vKSC's address is KSC's address + "/" + the vKSC's name.

6. Tenant administrators install KSC Network Agent on their virtual machines (for more details, see https://support.kaspersky.com/KSC/13.2/en-US/3305.htm).

A Network Agent is required to connect a virtual machine to KSC (in this case, the tenant's vKSC). To allow a Network Agent to connect to the correct vKSC, the vKSC's address needs to be indicated in the "Server address" field when the Network Agent is installed. After the Network Agents are installed, the virtual machines appear in the list of managed devices in the vKSC.

7. A provider's administrator sends installation packages for installing Light Agents to vKSC tenants using an installation package distribution task (for more details, see https://support.kaspersky.com/KSC/13.2/en-US/6383.htm). In the “Managed devices” folder in their own vKSCs, the tenants’ administrators select the machines where a Light Agent should be installed, and start the received Installation packages on them by selecting the “Install application” option.

8. The provider's administrator uses the REST API request described above to add the machines where Light Agents were installed to the Integration Server database.

9. The provider’s administrator installs the required number of SVMs on ESXi hypervisors, for example, relying on the basic estimate of 80 Light Agents per 1 SVM. The SVM component is provided as an image. SVMs are deployed using the SVM installation wizard, which is started in Integration Server Console. During the deployment process, the administrator connects vCenter to the Integration Server, selects the hypervisors where SVMs will be deployed, configures virtual machine settings (network adapters and storages), as well as the settings for connection SVMs to the Integration Server and Administration Server.

10. The provider's administrator creates policies for SVMs and Light Agents. The Light Agent policy determines the algorithm by which Light Agents select SVMs:

- Standard SVM selection algorithm: a Light Agent attempts to connect to an SVM deployed on the same hypervisor on which the Light Agent is running.
- Advanced SVM selection algorithm: the user can use the “SVM path” slider to choose how Light Agents should account for the SVM path when selecting which SVM to connect to. For example, a Light Agent can connect to an SVM deployed on the same hypervisor cluster or in the same data center as the Light Agent.

11. The provider's administrator enables protection of a tenant by sending the REST API described above to the Integration Server. The provider can disable protection of the tenant, if necessary, e.g. if a tenant fails to pay for protection services.

This scheme should be scaled up once the provider's virtual infrastructure reaches approximately 50,000 virtual machines, as there is a limitation on how many tenant requests the Integration Server can reliably handle. At present, there are also KSC limitations on the number of virtual Administration Servers to 500, and the number of protected machines in KSC is limited to 15,000. Thus, we recommend scaling up the entire scheme once tenants in a group reach 15,000 machines.

9.2 "Simple tenant" scenario

Now let's consider an example of a user infrastructure where the structure of tenants’ protection is already configured in multitenancy mode.

The provider provides virtualization services on its vCenter and already has KSVLA installed: KSC and Integration Server are installed, and SVMs are deployed, and Light Agent is installed on the tenant's machines. In KSC, the provider has grouped the tenant's machines into service groups and independently manages tenants’ protection. The provider wants to automate billing of tenants for protection services. To do this, the provide needs to know the periods during which each of the tenant's machines were protected.

1. On the Integration Server, the provider's administrator enables the collection of data on tenants’ protection. To do this, the administrator sets the EnableTenantsProtectionReports parameter to "true" in the Integration Server configuration file (%ProgramFiles(x86)%\Kaspersky Lab\Kaspersky VIISLA\viislaservice.exe.config) and restarts the Integration Server after restarting the VIISLA service.

2. The provider’s administrator uses the REST API request described above to add information about its tenants to the Integration Server database. If tenants unsubscribe from protection services or if new tenants buy protection services, the provider's administrator also uses REST API requests to remove and add tenants.

3. The provider's administrator uses the REST API request described above to add information about tenants' virtual machines to the Integration Server database. If a tenant stops using a machine or if tenants get new machines, the provider's administrator also uses REST API requests to register and unregister the tenant's machines.

4. The provider's administrator manages tenants' protection. If protection services are not paid for, the provider can disable a tenant's protection, but this does not happen automatically in this scenario:

- The provider disables a tenant's protection without using REST API requests, for example, through a KSC policy.
- The provider sends a REST API request to the Integration Server to communicate that the tenant's protection is disabled.
- The Integration Server stops monitoring protection of all of this tenant's registered machines and adds information about the tenant's protection to reports.

5. When preparing to bill tenants for protection services, the provider's administrator requests a report from the Integration Server, specifying the tenant ID and time period, such as one month, for the report. The generated report lets the provider know how much time each of the tenant's machines was protected, and makes it possible to calculate the tenant's bill based on the provider's rate.