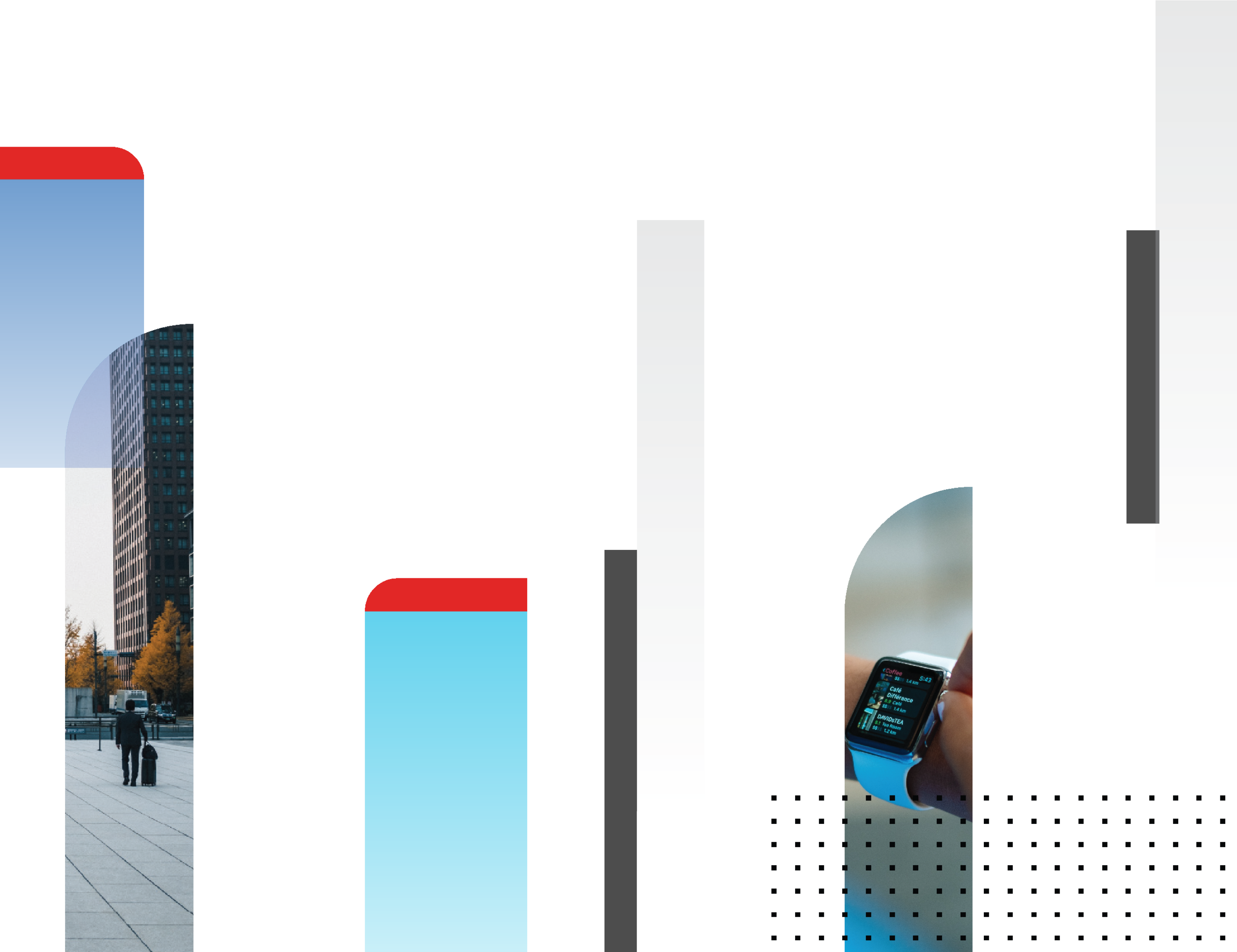
Multi-Tenancy Support

FortiSOAR 7.0.1



FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO GUIDE
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/support-and-training/training.html

NSE INSTITUTE
https://training.fortinet.com

FORTIGUARD CENTER
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

July, 2021
FortiSOAR 7.0.1 Multi-Tenancy Support
00-400-000000-20210107

TABLE OF CONTENTS

Change Log

Date	Change Description
2021-07-09	Initial release of 7.0.1

Overview of Multi-tenancy support in FortiSOAR

FortiSOAR provides additional features and out-of-the box configuration for a multi-tenant environment, enhancing its native support for multi-tenancy for managed security services providers (MSSPs). The FortiSOAR platform supports multi-tenancy in either of the following forms as well as a hybrid of both forms:

- Distributed Tenancy support: Used by MSSPs to provide services to remote tenants. In this case, each tenant will have its own FortiSOAR instance that will also automatically communicate to a master FortiSOAR instance having visibility across all the tenants. For more information, see the Distributed Tenancy Support chapter.
- Shared Tenancy support: Used by master (teams like the SOC team) to provide services to local tenants. In this case, you have only a single FortiSOAR instance, wherein multiple tenants will access a single FortiSOAR instance. For more information, see the Shared Tenancy Support chapter.

When multi-tenancy is enabled for the FortiSOAR platform, all alerts, incidents, and other forms of records created are associated with a tenant. RBAC and ownership rules applied to each tenant govern who has visibility to which records, thereby restricting visibility of one tenant's records to the other, but at the same time, providing the service provider a single consolidated view across tenants.

From version 6.4.1 onwards, tenants act like a wrapper that can contain multiple agents, which can connect to various disparate networks and remotely execute action. In the case of dedicated tenants, a default agent is automatically created and added to the dedicated tenant as part of the tenant creation process. In the case of shared tenancy, you do not require to create an agent for that tenant, since everything in case of a shared tenant runs on the master instance only using the master agent. If you require to run some actions on a separate network, then you can add and configure another agent, associate that with the shared tenant. The RBAC of this agent will be controlled by correct team assignment. Further, you can also add multiple agents to both shared and dedicated tenants. For more information on agents and how to run remote actions using agents, see the *Segmented Network support in FortiSOAR* chapter in the "Administration Guide."

Licensing for Multi-Tenancy

For multi-tenancy support, you require a FortiSOAR license that has been enabled for multi-tenancy. Therefore, when you are generating a license in FortiCare, ensure that you generate a license that has multi-tenancy support, i.e., a license with edition set to "MT | MT_Tenant". For more information about licensing, see the *Licensing FortiSOAR* chapter in the "Deployment Guide."

For a master node, you will be issued a license with type "MT" (`master`), and for a tenant node, you will be issued a license with type "MT_Tenant" (`tenant`). You can deploy the FortiSOAR license using the FortiSOAR Admin CLI (`csadm`) as follows:

SSH to your FortiSOAR VM and login as a *root* user and run the `# csadm license --deploy-multi-tenant-license <License File Path>` command, for a license that is enabled for multi-tenancy. For more information on `csadm`, see the *FortiSOAR Admin CLI* chapter in the "Administration Guide."

You will see a **Multi Tenancy** section in your `System` page by clicking the **Settings** icon on the top-right corner of FortiSOAR, only if your FortiSOAR license has been enabled for multi-tenancy.

After your license is deployed, FortiSOAR runs a **Publish** on the instance. If during this time, you try to open the FortiSOAR UI, you might see a "Publish In Progress" message. Also, you cannot revert the environment to a non-multitenant environment once you have deployed the license.

Notes for upgraded multi-tenant configurations

- You can upgrade a FortiSOAR distributed multi-tenant configuration to 7.0.0 from 6.4.3 or 6.4.4 only. For the upgrade procedure, see the *Upgrading a FortiSOAR Distributed Multi-tenancy Configuration to 7.0.0* section in the "Upgrade Guide."
- In case of a distributed deployment, both the master and the tenant nodes must be upgraded. A version mismatch will not work if either of them upgrades to 7.0.0.
- From version 6.4.1 onwards, each dedicated tenant automatically creates an agent that can be used to remotely execute actions. You can add multiple agents to a tenant, therefore, tenants become a wrapper that can contain various agents that can connect to various disparate networks. For more information on agents, see the *Segmented Network support in FortiSOAR* chapter in the "Administration Guide."

Recommended Resource Requirements for Virtual Machines (VM)

For Master and Tenant

Minimum Specifications

- 8 available vCPUs
- 22 GB available RAM
- 500 GB available disk space: Recommended to have high-performance storage, preferably SSDs.
- 1 vNIC

Recommended Specifications

- 8 available vCPUs
- 32 GB available RAM
- 1 TB available disk space: Recommended to have high-performance storage, preferably SSDs.
- 1 vNIC

For Secure Message Exchange

Minimum Specifications

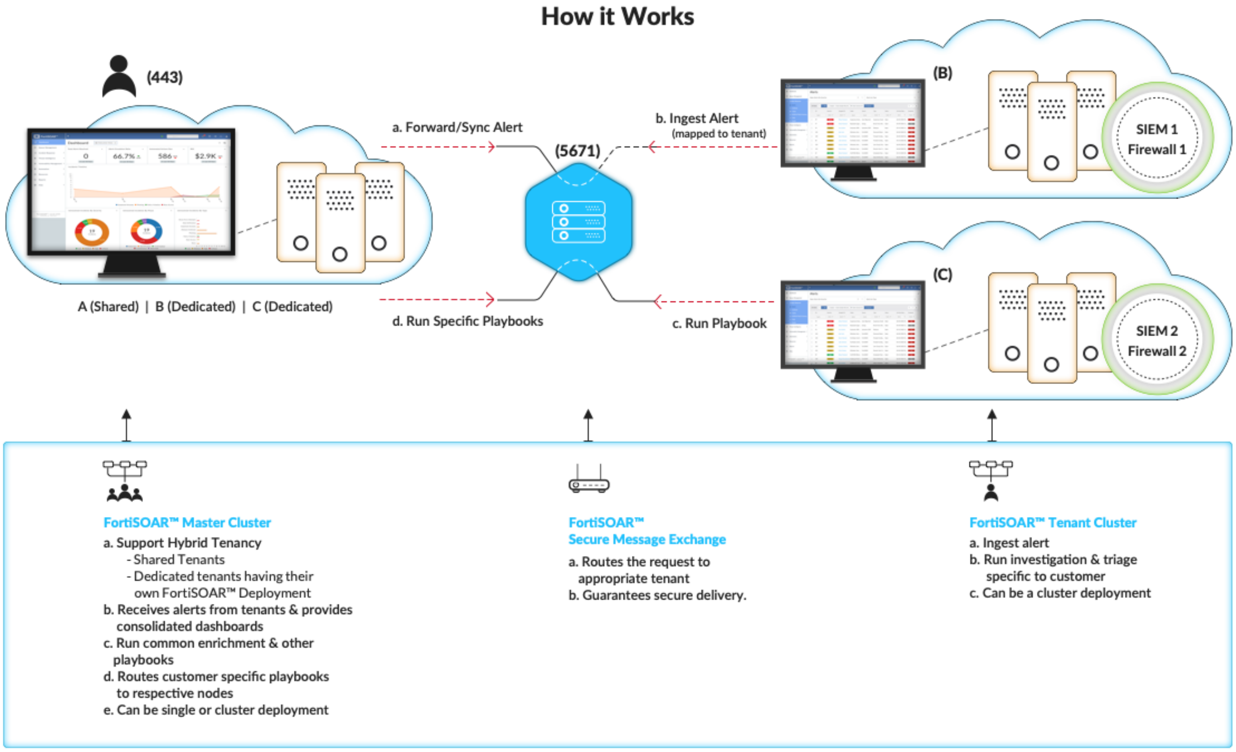
- 4 available vCPUs
- 8 GB available RAM
- 100 GB available disk space: Recommended to have high-performance storage, preferably SSDs.
- 1 vNIC

Recommended Specifications

- 8 available vCPUs
- 16 GB available RAM
- 100 GB available disk space: Recommended to have high-performance storage, preferably SSDs.
- 1 vNIC

Architecture

The overall architecture of the FortiSOAR Distributed Multi-Tenancy Model:


How it Works
(443)
A (Shared) | B (Dedicated) | C (Dedicated)
a. Forward/Sync Alert
(5671)
d. Run Specific Playbooks
b. Ingest Alert
(mapped to tenant)
c. Run Playbook
(B)
SIEM 1
Firewall 1
(C)
SIEM 2
Firewall 2
FortiSOAR™ Master Cluster
FortiSOAR™ Secure Message Exchange
FortiSOAR™ Tenant Cluster

FortiSOAR™ Master Cluster

a. Support Hybrid Tenancy
  - Shared Tenants
  - Dedicated tenants having their own FortiSOAR™ Deployment

b. Receives alerts from tenants & provides consolidated dashboards

c. Run common enrichment & other playbooks

d. Routes customer specific playbooks to respective nodes

e. Can be single or cluster deployment

FortiSOAR™ Secure Message Exchange

a. Routes the request to appropriate tenant

b. Guarantees secure delivery.

FortiSOAR™ Tenant Cluster

a. Ingest alert

b. Run investigation & triage specific to customer

c. Can be a cluster deployment

A brief description of how FortiSOAR supports the Distributed Multi-Tenancy model follows:

- A master FortiSOAR node is installed and configured at the MSSP (Service Provider) location.
- A FortiSOAR tenant node is installed and configured for each tenant (customer) location, same as configured for the master node.
  Note: Each of these FortiSOAR installations (both master as well as tenants) are fully functional standalone instances. They are additionally enabled for automatic replication of data between them, and also for remotely executing workflows at the tenant node from the master node itself.
  - Communication between the master and tenant nodes are done over a secure channel using the Secure Message Exchange.
    Each tenant node communicates with the secure message exchange on a dedicated space that is access controlled with credentials that are unique to each tenant.
    The secure message exchange ensures a guaranteed message delivery irrespective of whether the target node is up or not at the time the message is sent. The messages are queued and delivered to the target node once it comes online. This prevents message loss during network disruptions.

An example: The master FortiSOAR node is monitoring the tenants' FortiSOAR instance, and if for example, the investigation requires to run a playbook to get the reputation of a particular IP address from the tenants' end, then this facility allows the master node to run the playbook remotely at the tenant's end. The playbook will be run at the tenant node, using the tenant's credentials, for the 3rd-party connector for checking the IP reputation, and the remediation actions (if any), specified in the playbook, will also be done on the tenants' end.

Shared Tenancy Support

FortiSOAR supports a shared tenancy model for multi-tenancy and MSSP support. A managed security service provider (MSSP) is an IT service provider that provides organizations with some amount of cybersecurity monitoring and management, which may include virus and spam blocking, intrusion detection, and firewalls and virtual private network (VPN) management.

In case of shared tenancy, tenants share the same system as that of the master, i.e., tenants are local, but with restricted access on the system. The master (teams like the SOC team) provides cybersecurity monitoring and management to various tenants (customers/teams) in a single FortiSOAR instance.

You can choose between the Distributed and Shared tenancy models depending on your requirements based on the following points:

- In case of distributed tenancy support, tenants have greater control over the data that they share with the master.
- In case of distributed tenancy, you can avoid additional infrastructure overheads such as, directly connecting the tenant's SIEM to the MSSPs SOAR platform etc.
- Scaling considerations: If you have a large number of tenants then it is easier to scale in case of the distributed tenancy support model, as data primarily resides at the tenant nodes, and the computations also happen at the tenant, a single console at the master node can easily handle multiple customers.

The shared tenancy model ensures that data of different tenants are segregated, and data access is controlled using RBAC (role-based access control). Therefore, a tenant can view only their own data or record and not the data for other tenants.

Each tenant can be given their own login, using which they can view their dashboards, report, check the actions taken on their records, check their SLA management, etc.

The master can extend the `Tenants` module to include fields that they require for various tenant, such as address of the tenant, tenant SLA, etc. For more information, see the *Extending the Tenants module* section in the Distributed Tenancy Support chapter.

Steps required for deploying and configuring shared tenancy support

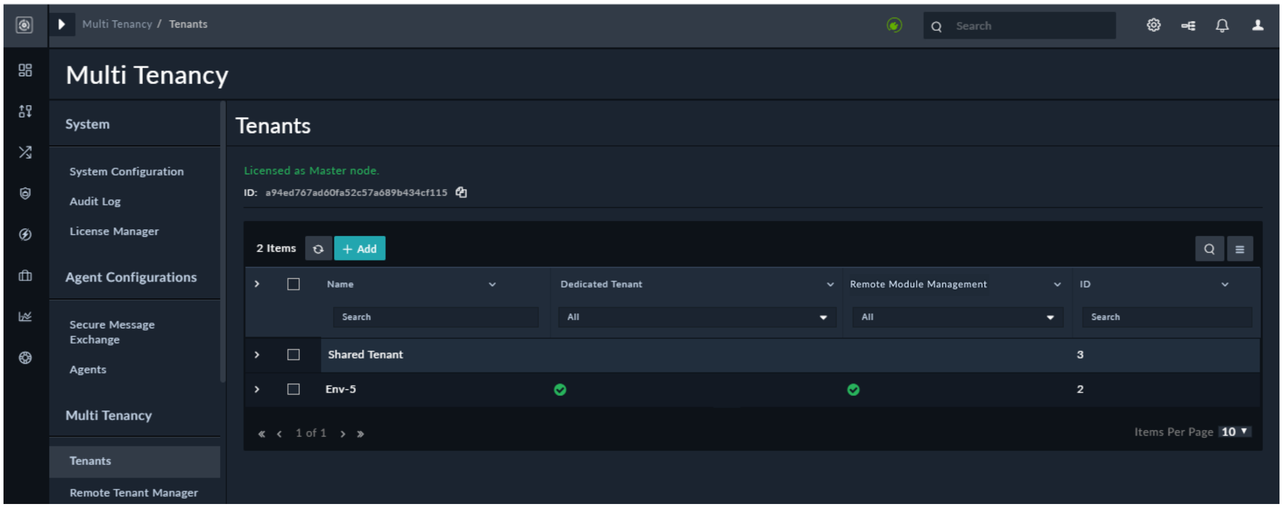
1. **Licensing FortiSOAR**
   For shared tenancy support, you require a FortiSOAR license that has been enabled for multi-tenancy. Therefore, when you are generating a license in FortiCare, ensure that you generate a license for multi-tenancy support, i.e., a license with edition set to "MT". For more information, see *Licensing for Multi-Tenancy* section in the Overview chapter.
2. **Deploying FortiSOAR**
   See the *Deploying FortiSOAR* chapter in the "Deployment Guide" for detailed information on how to deploy the FortiSOAR Virtual Appliance. For more information, see the *Deploying the Master Node* section in the Distributed Tenancy Support chapter.
3. **Configuring Shared Tenancy**
   See the Configuring Shared Tenancy section for configuring and onboarding local tenants.

Configuring Shared Tenancy

Only if your FortiSOAR license has been enabled for multi-tenancy you will see a **Multi Tenancy** section on your `System` page.

To see whether your system has been enabled for shared tenancy, you can do the following:

1. Log on to FortiSOAR as an administrator.
2. Click the **Settings** (⚙) icon to open the `System` page.
3. On the `System` page, you will see the `Multi Tenancy` section. Click the **Tenants** item in the left menu, to view details of the master.
   On the top of the `Tenants` page, you will see the tenant ID of the master node, and also see `Licensed as Master node` text.



Licensed as Master node.

ID: a94ed767ad60fa52c57a689b434cf115

Name	Dedicated Tenant	Remote Module Management	ID
Shared Tenant			3
Env-5	✓	✓	2

Onboarding tenants

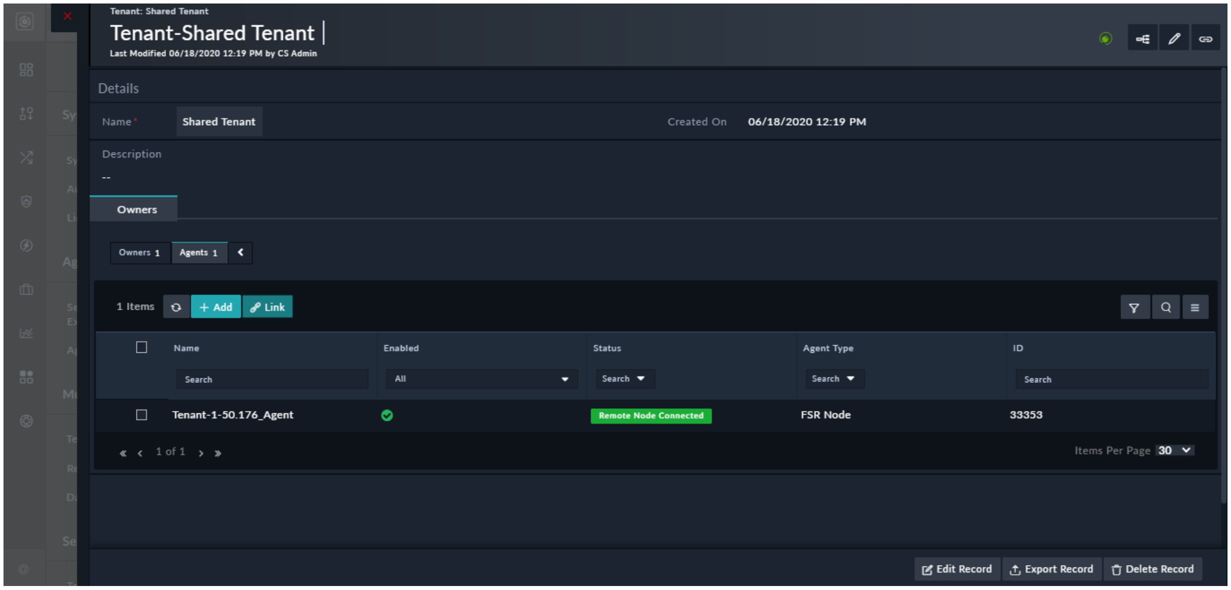
Adding tenants

1. Log on to FortiSOAR as an administrator and click the **Settings** icon to open the `System` page.
2. To add tenants, click **Tenants** in the left menu and on the `Tenants` page, click **Add**.
   To edit the configuration of an existing tenant, click the tenant whose configuration you want to update, which opens the agent record in the detail view. Update the configuration parameters as required, and click **Save**.
   If you no longer require an existing tenant you can deboard that tenant. Deboarding a tenant is an irreversible operation which also deletes all data related to that tenant from the master node. For more information see Deboarding Tenants.
   If you want to deactivate a shared tenant, then you can deactivate the user login for that shared tenant.
3. In the `Add Tenant` dialog, configure the following parameters:
   a. From the **Choose Tenant Type** field choose the type of tenant you want to add.
      In this case, select **Shared** to add this tenant as a local tenant of the master, i.e., in the same FortiSOAR instance for shared tenancy. Local or Shared tenants, based on their permissions, share the current FortiSOAR instance with the Master.
   b. In the **Name** field, enter the name of the tenant.
   c. (Optional) In the **Description** field, enter the description of the tenant.

d. From the **Owners** drop-down list, select the teams that you want to add as an owner to the records coming from this tenant and click **Link Team**.
 The teams that you have selected and the SOC Admin team (considering that the tenant has been created using the SOC Admin team) will be associated as owners of any record created for that tenant. If no team is selected, then the teams that the user who adds the tenant belongs to will be associated as owners of any record created for that tenant.

e. To complete adding a new tenant associated with the master node, click **Save**.
 As tenants are wrappers, no configuration state is associated with a tenant and you also do not need to create an agent, since everything in case of a shared tenant runs only on the master instance using the master agent. However, if you require to run some actions on a separate network, then you can add and configure another agent, and then associate that with the shared tenant.

f. To link an agent with a shared tenant, open the record of the shared tenant and in the `Owners` section, click the **Agents** tab, and click **Link** to open the `Link Agents` dialog that contains a list of configured agents. Select the agent that you want to associate with the shared tenant and click **Save Relationship**:



Tenant: Shared Tenant

Tenant-Shared Tenant

Last Modified 06/18/2020 12:19 PM by CS Admin

Details

Name * Shared Tenant

Created On 06/18/2020 12:19 PM

Description

--

Owners

Owners 1 | Agents 1

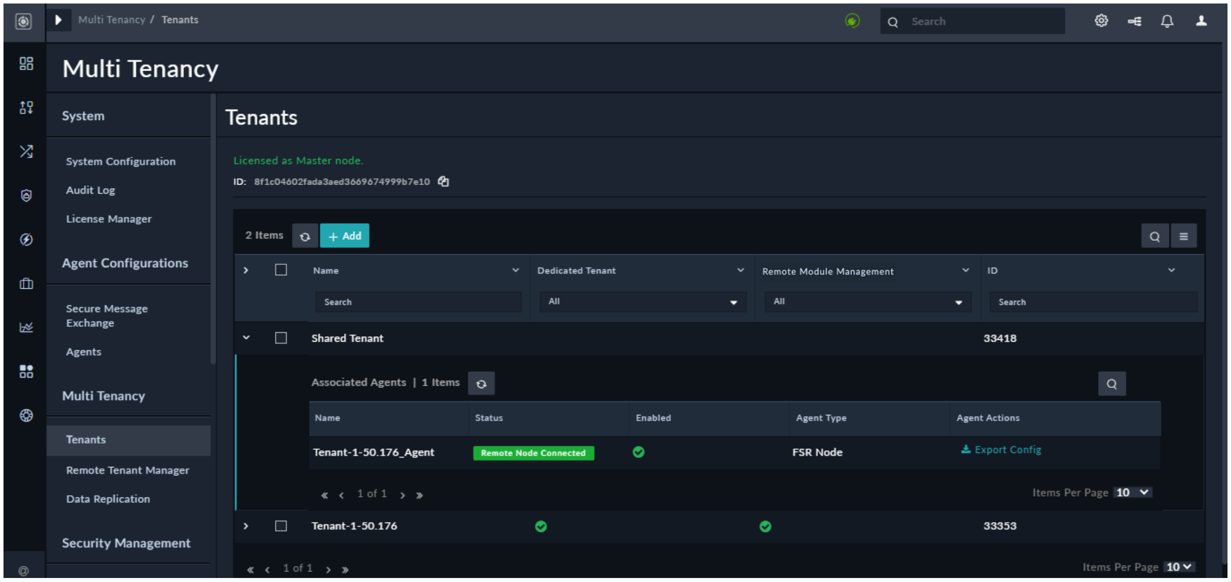
1 Items

□	Name	Enabled	Status	Agent Type	ID
	Search	All	Search	Search	Search
□	Tenant-1-50.176_Agent	✓	Remote Node Connected	FSR Node	33353

1 of 1

Items Per Page 30

 Once you have linked an agent you can expand the row of the shared tenant and check the connection between the agent and the master node using secure message exchange. Once the connection is established from the master node to the agent, the **Status** field displays "Remote Node Connected".



Tenants

Licensed as Master node.

ID: 8f1c04602fada3aed3669674999b7e10

Name	Dedicated Tenant	Remote Module Management	ID
Search	All	All	Search
Shared Tenant			33418
Tenant-1-50.176	✓	✓	33353

Associated Agents | 1 Items

Name	Status	Enabled	Agent Type	Agent Actions
Tenant-1-50.176_Agent	Remote Node Connected	✓	FSR Node	Export Config

1 of 1 Items Per Page 10

1 of 1 Items Per Page 10

List of Statuses

Following is the list of statuses that can be displayed:

- **Configuration In Progress**: Process of configuring the agent has begun.
- **Awaiting Remote Node Connection**: Connection between the FortiSOAR node and secure message exchange is established and awaiting the connection to the agent.
- **Remote Note Connected**: Agent has been connected to the FortiSOAR node using secure message exchange.
- **Configuration Failed**: Agent failed to be added on the secure message exchange.
- **Message Exchange Unreachable**: Secure message exchange is unreachable.
- **Remote Node Unreachable**: Agent is unreachable from the FortiSOAR node.

Associating the tenant with teams and users

If you have not associated the tenant with teams at the time of adding tenants (step 3e of the previous procedure), then once you have created a tenant, you must associate the tenant to team(s). Tenants must be associated with teams so that all records created for a given tenant are automatically owned by the same teams who are associated with the tenant. For example, if alerts are created whose tenant is set as T01, then only the teams associated with team T01 and their ancestors in the team hierarchy will be able to view the alert record.

Steps mentioned in the following sections describe how to add a team and then associate the tenant with that team.

Once you associate the tenant with teams you must also assign users to those teams, as mentioned in the following *Adding a user to a team* section.

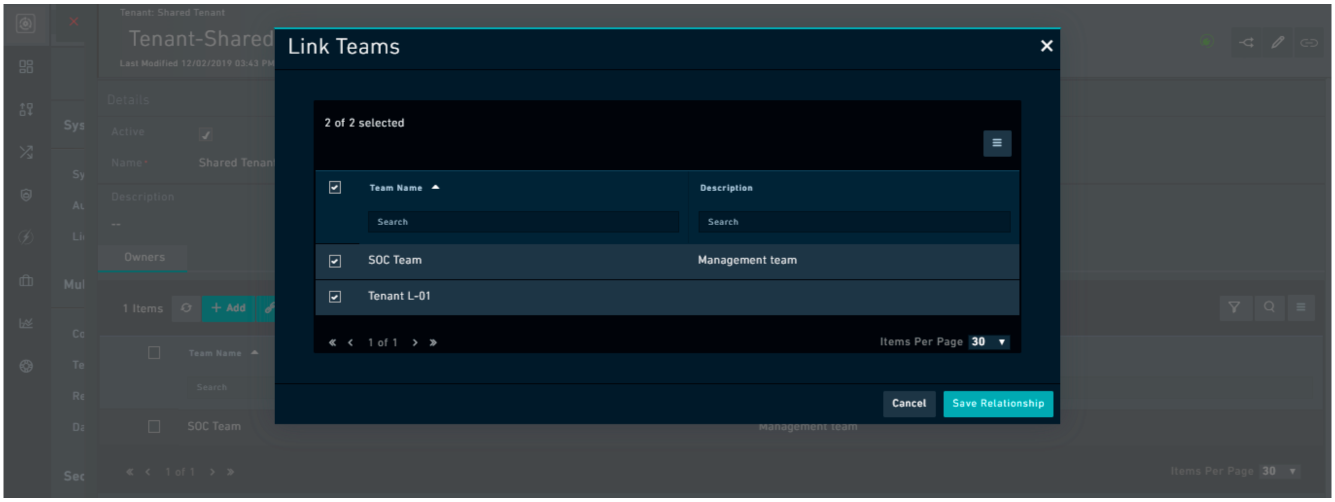
Adding a Team

To add a team, click the **Settings** icon and in the `Security Management` section click **Teams**. Click **Add Team** to display the `Add Team` dialog and enter the name of the team that you want to create, for example, `Tenant-L01` and optionally add a description for the team and click **Create** to add the new team.

Associating the tenant with the teams

You should now associate the tenant with this newly created team, so that records created in this tenant will be owned by the team. You must also link a management or master team, such as SOC Manager with the tenant, or ensure that the master team is at the top of the team hierarchy, so that record will be visible to the master.

To assign the tenant to a team, click the **Settings** icon and in the `Multi Tenancy` section click **Tenants**. Click the tenant that you want to assign to the team, for example, `Tenant-L01`, which opens the tenant record. In the tenant record, click **Link**, and in the `Link Teams` dialog, select the team that you want to associate with the tenant. In our example, select the `Tenant-L01` and `SOC Manager` team and click **Save Relationship**:



Link Teams

2 of 2 selected

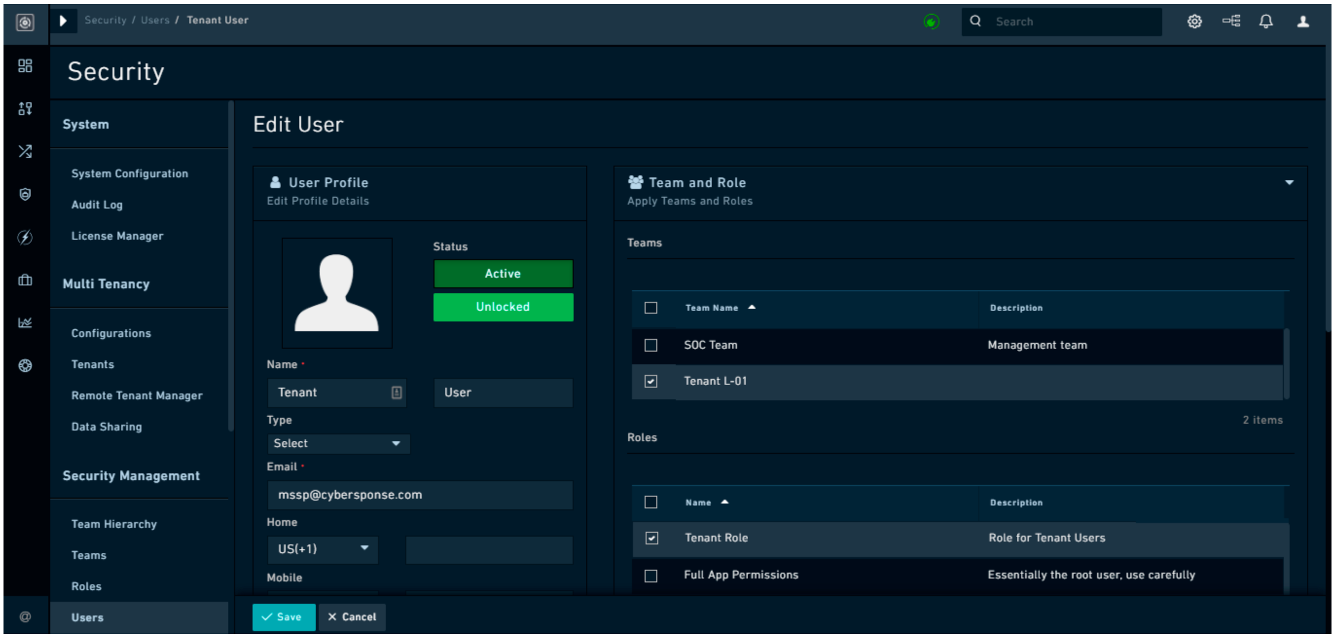
☑	Team Name	Description
☑	SOC Team	Management team
☑	Tenant L-01	

Adding a user to a team

The users in the team that you have created should be assigned to a role with a minimum of `Read` permission on the `Application`, `Tenants`, and `People` modules, apart from required permissions on the case management modules, like `Alerts`, `Incidents`, `Tasks`, etc. You must not assign these users any `Security` permissions. It is also recommended that you should not assign any `Playbook` or `Connector` permissions to these users, if you do not want them to run any playbooks.

You can create a new role (for example `Tenant` Role) with the recommended permissions and assign them to the users that you want to assign to the Tenant team.

For example, a user named **Tenant User**, who you want to create as a tenant user, should be added or linked to the team selected as `Tenant-L01` and role assigned as `Tenant Role`:



Status: Active, Unlocked

Name: Tenant User

Type: Select

Email: mssp@cybersponse.com

Home: US(+1)

Mobile

Teams

☐	Team Name	Description
☐	SOC Team	Management team
☑	Tenant L-01	

2 items

Roles

☐	Name	Description
☑	Tenant Role	Role for Tenant Users
☐	Full App Permissions	Essentially the root user, use carefully

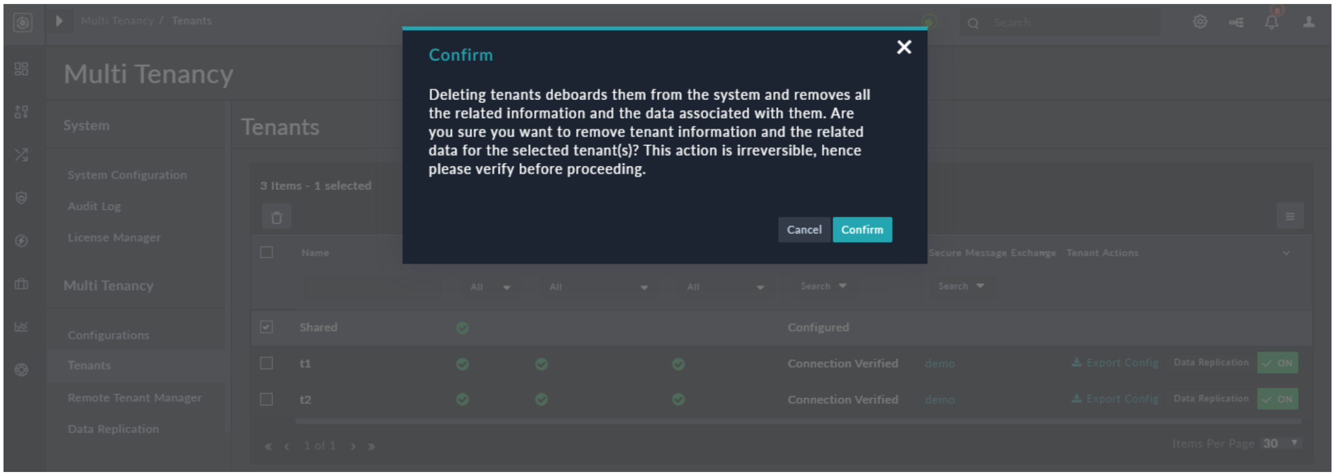
Deboarding tenants

You might want to deboard existing tenants due to the following reasons:

- Tenant has not been configured correctly and you are required to remove the tenant and then add the tenant back.
- You (service provider) no longer manages the tenant.

To deboard existing tenants, you require to have `Read` and `Delete` permissions on the `Tenants` module. Deboarding tenants not only deletes the tenant from the master node, but also removes all information and data associated with that tenant from the master node. Once you delete a tenant, you cannot retrieve any information related to that tenant, therefore you must be careful while performing this operation.

To deboard a tenant, log on to FortiSOAR as an administrator and click the **Settings** icon to open the `System` page. Click **Tenants** in the left menu and on the `Tenants` page, click **Delete**. FortiSOAR will display the following warning dialog:



Deleting tenants deboards them from the system and removes all the related information and the data associated with them. Are you sure you want to remove tenant information and the related data for the selected tenant(s)? This action is irreversible, hence please verify before proceeding.

To deboard the tenant, click **Confirm** on the warning dialog.

Once you deboard a dedicated tenant, the associated dedicated agent is also removed and other agents unlink themselves from the tenant.

Example of editing the templates to add the Tenant field

In the ingestion playbooks, you must ensure that the tenant field is correctly set on the created records. While creating records manually, you must take care that the tenant field is defined. If no tenant is defined, the 'self' tenant gets associated with the record.

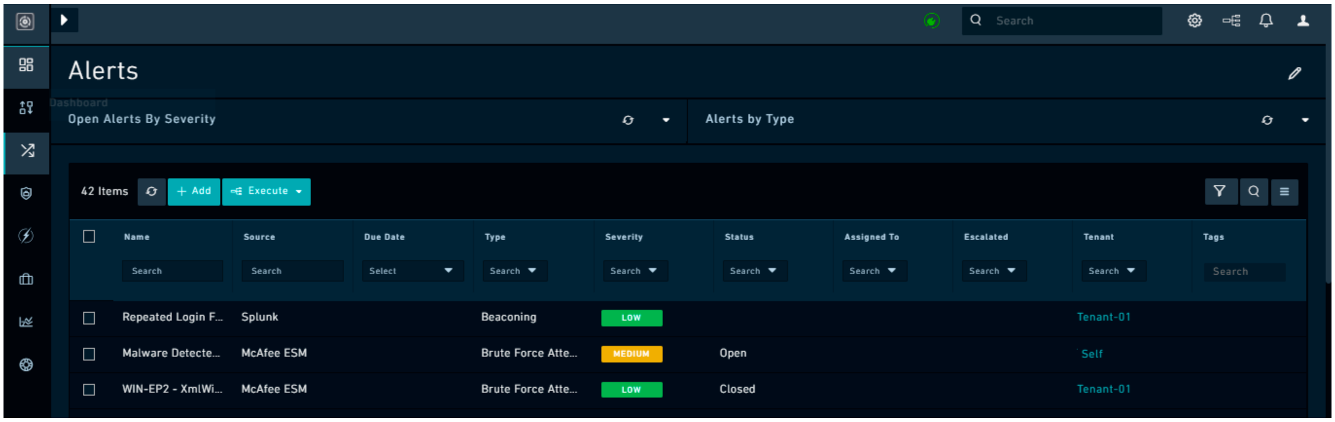
You (administrators of the master team) should edit the templates (SVT), for the modules in which tenants will create records, for example, **Alerts**, to include the **Tenant** field when a user is creating a record.

To edit the SVT, navigate to the module for which you want to update the SVT, for example, `Alerts`, open a record and click **Edit Template.** On the `Template Editing Mode Enabled` page, in the appropriate widget, such as `Form Group Details` click **Edit**, add the `Tenant` field, and then click **Add**. Click **Save** to save your changes to the SVT.

When an alert from a tenant is added, for example **Repeated Login Failures**, then the **Tenant** field the name of tenant associated with this record gets populated, thereby enabling the master to easily identify which records belong to which tenant. If no tenant is assigned while adding the alert, i.e., if the alert is created on the master, then that record is created as a `Self` record, i.e., **Self** will appear in the Tenant column.

Once a record is created, you cannot change the tenant that is associated with the record, i.e., tenant becomes a read-only field once the record is created.

The following image displays a view on the master with alert records from different tenants:



□	Name	Source	Due Date	Type	Severity	Status	Assigned To	Escalated	Tenant	Tags
	Search	Search	Select	Search	Search	Search	Search	Search	Search	Search
□	Repeated Login F...	Splunk		Beaconing	LOW				Tenant-01	
□	Malware Detecte...	McAfee ESM		Brute Force Atte...	MEDIUM	Open			Self	
□	WIN-EP2 - XmlWi...	McAfee ESM		Brute Force Atte...	LOW	Closed			Tenant-01	

Now the master can perform all required actions such as running investigative playbooks, on this alert record and provide cybersecurity management to the tenant.

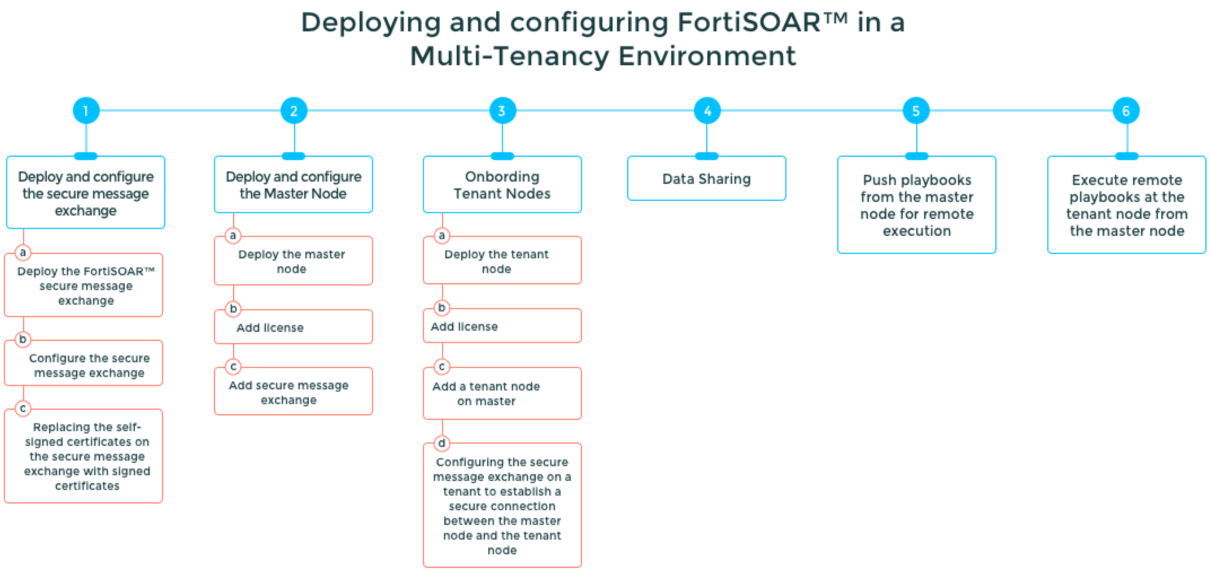
Distributed Tenancy Support

FortiSOAR introduced a distributed managed security service provider model for multi-tenancy and MSSP support. In the case of the distributed tenancy model, tenants are remote, i.e., the master and each of the tenants have their own FortiSOAR instance.

Benefits of the FortiSOAR Distributed Multi-Tenancy Model

- **Autonomous Tenant Nodes**: Tenant nodes work independently.
- **Secured Communication**: Data and actions are exchanged securely.
- **Privacy and Integrity of Data**: Tenant's data remains in their environment, and they control how much data they want to share with the master node. All sensitive information stays with the tenant node. Since the actual workflow execution happens at the tenant node itself, the master node requires only the summary of information to help identify what investigations are to be run.
- **Diversity Handling**: Multiple tenants' having their unique procedures, and SLAs can be handled.
- **Simplified Remediation**: Use standard playbooks to triage or remediate incidents on the tenant node from the master node, and the procedures look simplified and manageable at the master node.
  For example, if you have a Phishing Email Alert, the analyst at the master node just invokes a generic “Investigate Phishing” workflow. This automatically resolves and executes the procedure at the tenant node the alert came from. Hence, the workflow run aligns with the tools and procedures on every tenant.
- **Scalability**: As data primarily resides at the tenant nodes, and the computations also happen at the tenant, a single console at the master node can easily handle multiple customers.
- **No overhead on Network Infrastructure**: The FortiSOAR distributed managed security service provider model does not need a VPN setup between the customer and service provider environment. Both tenants, as well as master node only, need an outbound TCP connectivity to the secure message exchange. Hence the extra network setup overhead and switching of VPNs is avoided.

Deploying and configuring FortiSOAR in a multi-tenancy environment


Deploying and configuring FortiSOAR™ in a Multi-Tenancy Environment
1
Deploy and configure the secure message exchange
a
Deploy the FortiSOAR™ secure message exchange
b
Configure the secure message exchange
c
Replacing the self-signed certificates on the secure message exchange with signed certificates
2
Deploy and configure the Master Node
a
Deploy the master node
b
Add license
c
Add secure message exchange
3
Onbording Tenant Nodes
a
Deploy the tenant node
b
Add license
c
Add a tenant node on master
d
Configuring the secure message exchange on a tenant to establish a secure connection between the master node and the tenant node
4
Data Sharing
5
Push playbooks from the master node for remote execution
6
Execute remote playbooks at the tenant node from the master node

1. Deploy and configure the Secure Message Exchange.
   a. Deploy the FortiSOAR Secure Message Exchange.
   b. Configure the Secure Message Exchange.
   c. Replacing the self-signed certificates on the secure message exchange with signed certificates.
2. Deploy and configure the Master Node.
   a. Deploy the master node.
   b. Add license: See the *Licensing for Multi-Tenancy* section in the Overview chapter.
   c. Add Secure Message Exchange.
3. Onboarding Tenant Nodes.
   a. Deploy the tenant node.
   b. Add license: See the *Licensing for Multi-Tenancy* section in the Overview chapter.
   c. Add a tenant node on master.
   d. Configure the secure message exchange on a tenant to establish a secure connection between the master node and the tenant node.
4. Data Replication.
5. Push playbooks from the master node for remote execution.
6. Execute remote playbooks at the tenant node from the master node.

Deploying and configuring the FortiSOAR Secure Message Exchange

A secure message exchange establishes a secure channel using which you can relay information to your agent or tenant nodes.

Contact FortiSOAR Support to provide you with secure message exchange appliance that must be installed in addition to the FortiSOAR OVA on a different Virtual Machine (VM) from your FortiSOAR VM.

You do not require any additional licensing for the FortiSOAR secure message exchange.

Deploying the FortiSOAR Secure Message Exchange

1. Deploy the FortiSOAR Secure Message Exchange, which is the same as deploying a FortiSOAR Virtual Appliance. See the *Deploying FortiSOAR* chapter in the "Deployment Guide" for detailed information on how to deploy the FortiSOAR Virtual Appliance. Steps are as follows:
   a. Review and ensure that you meet the recommended resources required to deploy the FortiSOAR secure message exchange (same as FortiSOAR Virtual Appliance).
   b. Import the FortiSOAR secure message exchange (same as FortiSOAR Virtual Appliance).
   c. Deploy the FortiSOAR secure message exchange using vSphere/vCenter or AWS (same as FortiSOAR Virtual Appliance).

Configuring the Secure Message Exchange

After you have completed deploying the secure message exchange appliance, and when you log in to the secure message exchange appliance using `ssh` for the first time, the **FortiSOAR Secure Message Exchange Configuration Wizard** is displayed. The wizard guides you through the RabbitMQ configuration process with appropriate instructions so that you can efficiently perform the configuration required for RabbitMQ.

The wizard performs the following configuration steps:

1. **Hostname Change (Optional)**: You can change the hostname for your RabbitMQ VM. The wizard checks if the hostname is valid or not; and throws an error in case of an invalid hostname.
   You must change the hostname to a resolvable hostname using which the master and tenant nodes can connect to the secure message exchange.
2. **Add DNS Name or IP Address (Optional)**: You can change the DNS Name or IP address for your RabbitMQ VM.
3. **Add Username (Optional)**: Username that you will use to connect to the RabbitMQ management console. If you do not specify any username, then by default, the username is set as `admin`.
4. **Add Password**: Password that you will use to connect to the RabbitMQ management console. You must mandatorily specify the password.
   Note: Ensure you retain a copy of this password. FortiSOAR does not save the password in any file on the secure message exchange for security reasons. Therefore, this password cannot be recovered.
5. **Add SSL port (Optional)**: Specify an SSL port between `49152` and `65535` for RabbitMQ management console. By default, this port is set as `15671`.
6. **Add TCP port (Optional)**: Specify the management TCP port between `49152` and `65535`. By default, this port is set as `5671`.
   Once you specify the above parameters, the FortiSOAR Secure Message Exchange Configuration wizard gets ready to configure your secure message exchange, which includes generating Self-Signed Certificates for RabbitMQ.
   Important: You get logged out after the FortiSOAR Secure Message Exchange is configured so that the changes can take effect. Therefore, you are required to `ssh` again to the FortiSOAR Secure Message Exchange VM.
   Note: The configuration log for FortiSOAR Secure Message Exchange is located at `/var/log/cyops/install/config_vm_<timestamp>`.
   For example, `/var/log/cyops/install/config-vm-30_Oct_2018_12h_50m_38s.log``

The FortiSOAR Secure Message Exchange Configuration Wizard also displays the path of the Secure Message Exchange configuration log.

Replacing the self-signed certificates on the secure message exchange with signed certificates

It is highly recommended that the certificates used for encrypted communication between the FortiSOAR nodes and the secure message exchange should be signed from the Certificate Authority.

To replace the self-signed certificates on the secure message exchange with signed certificates, do the following:

1. Replace the following files on your FortiSOAR secure message exchange with the corresponding signed files from the Certificate Authority:
   - CA Certificate: `/opt/cyops/configs/rabbitmq/ssl/cyopsca/cacert.pem`
   - Server Certificate: `/opt/cyops/configs/rabbitmq/ssl/server/cert.pem`
   - Service Private Key: `/opt/cyops/configs/rabbitmq/ssl/server/key.pem`
     Note: A `.key` file has the path to a PEM encoded file containing the private key. A `.pem` file has the path to a PEM encoded file containing the certificate (or certificate chain) that will be presented when requested.
2. Restart the RabbitMQ server and all its related services using the following command:
   `# systemctl restart rabbitmq-server`.

Deploying the Master Node

See the *Deploying FortiSOAR* chapter in the "Deployment Guide" for detailed information on how to deploy the FortiSOAR Virtual Appliance. Steps included in deploying the master node:

1. Review and ensure that you meet the recommended resources required to deploy the FortiSOAR Virtual Appliance.
2. Import and deploy the FortiSOAR Virtual Appliance using vSphere/vCenter or AWS.
3. FortiSOAR Configuration Wizard
   FortiSOAR Configuration Wizard runs automatically on the first `ssh` login by the `csadmin` user and performs the initial configuration steps that are required for FortiSOAR and performs tasks as is the case with the FortiSOAR enterprise OVA. It generates a Device UUID for your FortiSOAR instance. Use this Device UUID to generate the FortiSOAR license in FortiCare so that you can begin using FortiSOAR. Ensure that you generate a license that has its edition set to "MT".

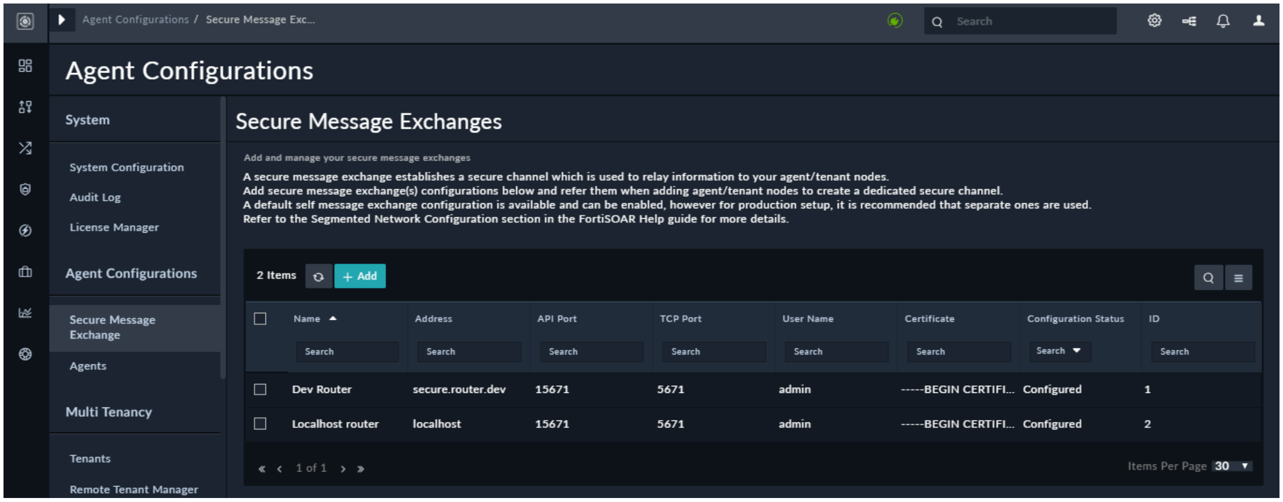
Adding the secure message exchange on the master node

Only if your FortiSOAR license has been enabled for multi-tenancy you will see a **Multi Tenancy** section on your `System` page.

To add the secure message exchange on the master node, do the following:

1. Log on to FortiSOAR as an administrator.
2. Click the **Settings** (⚙) icon to open the `System` page.

3. On the `System` page, you will see the `Agent Configurations` section. Click the **Secure Message Exchange** item in the left menu, to configure the secure message exchange on the master node.



Secure Message Exchanges

Add and manage your secure message exchanges

A secure message exchange establishes a secure channel which is used to relay information to your agent/tenant nodes.
Add secure message exchange(s) configurations below and refer them when adding agent/tenant nodes to create a dedicated secure channel.
A default self message exchange configuration is available and can be enabled, however for production setup, it is recommended that separate ones are used.
Refer to the Segmented Network Configuration section in the FortiSOAR Help guide for more details.

□	Name	Address	API Port	TCP Port	User Name	Certificate	Configuration Status	ID
	Search	Search	Search	Search	Search	Search	Search	Search
□	Dev Router	secure.router.dev	15671	5671	admin	-----BEGIN CERTIFI...	Configured	1
□	Localhost router	localhost	15671	5671	admin	-----BEGIN CERTIFI...	Configured	2

4. Add a secure message exchange to the master node and add the reference of this secure message exchange in the tenant or agent node(s) to create a dedicated secure channel of communication.
 You can have more than secure message exchange in the configuration. You can distribute tenants across secure message exchanges based on the geographical locations, scale, or compliance policies of the respective customers.
 Note: You should have already configured the secure message exchange using the steps mentioned in Configuring the Secure Message Exchange.
 To add a secure message exchange, click **Add** on the `Secure Message Exchanges` page.
 Important: To add a secure message exchange and configure tenants, you must have a role that has a minimum of `Create`, `Read`, and `Update` permissions on the **Secure Message Exchange** and **Tenants** modules.
 To edit the configuration of an existing secure message exchange, click the secure message exchange row whose configuration you want to update. This displays the `Edit Secure Message Exchange` dialog. Update the configuration parameters, as required, in the dialog and click **Update**.
5. In the `Add New Secure Message Exchange` dialog, configure the following parameters:
 a. In the **Name** field, enter the name of the secure message exchange that you have configured to act as a secure channel of data replication between the master and tenant nodes.
 b. In the **Address** field, enter the FQHN (Fully Qualified Host Name) of the secure message exchange.
 Important: Ensure that the FQHN matches the Certificate Name (CN) or the Subject Alternative Name (SAN) provided in the SSL certificate used to configure the secure message exchange.
 c. In the **API Port** field, enter the RabbitMQ Management port number that you had specified while configuring the secure message exchange, and ensure that the master node has outbound connectivity to the secure message exchange at this port.
 By default, it is set as `15671`.
 d. In the **TCP Port** field, enter the TCP port number that you had specified while configuring the secure message exchange, and ensure that the master node has outbound connectivity to the secure message exchange at this port.
 By default, it is set as `5671`.
 e. In the **User Name** field, enter the username you will use to login to your secure message exchange as an administrator.
 By default, it is set as `admin`.
 f. In the **Password** field, enter the password you will use to login to your secure message exchange as an administrator.
 g. In the **Certificate** field, copy-paste the certificate text of the Certificate Authority (CA) that has signed the secure message exchange certificate in the `pem` format. If it is a chain, then the complete chain must be

provided.
By default, the CA certificate for the FortiSOAR self-signed certificate is present at the following location: `/opt/cyops/configs/rabbitmq/ssl/cyopsca/cacert.pem`.
Important: If in the future, your secure message exchange certificate expires, and you need to deploy a new certificate, then the new certificate must be copied back to the master node as well as the tenant's router entry.

h. To save the configuration for the secure message exchange on the master node, click **Create**.

A default "Self" entry is created for every master node and you can directly link agents to the master node. For more information on agents, see the *Segmented Network support in FortiSOAR* chapter in the "Administration Guide."

Onboarding tenant nodes

From version 6.4.1 onwards, a default agent is automatically created and added to a dedicated tenant as part of the tenant creation process. For more information on agents, see the *Segmented Network support in FortiSOAR* chapter in the "Administration Guide." You can add multiple agents to a tenant, therefore, tenants become a wrapper that can contain various agents that can connect to various disparate networks and execute actions remotely.

Deploying tenant nodes

Deploy tenants as required. See the *Deploying FortiSOAR* chapter in the "Deployment Guide" for detailed information on how to deploy the FortiSOAR Virtual Appliance. Steps included in deploying a tenant node:

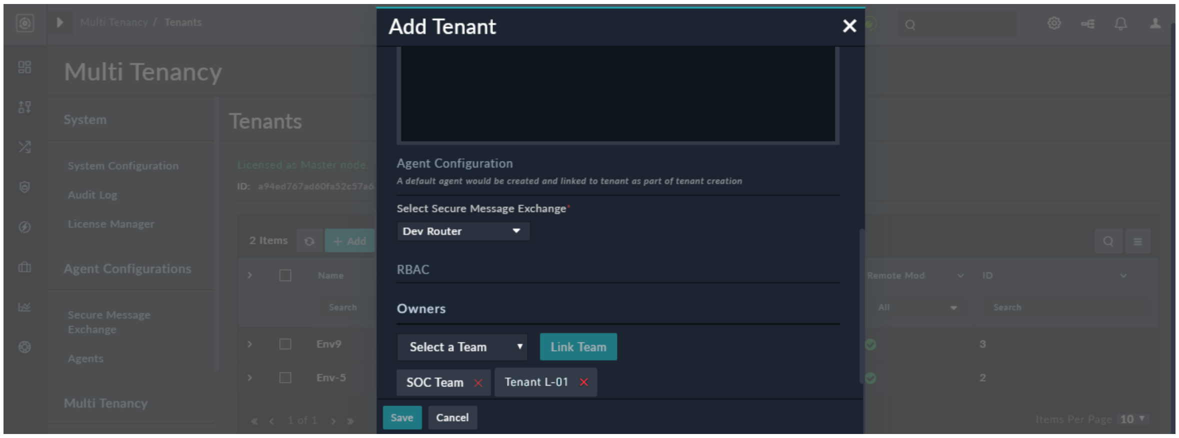
1. Review and ensure that you meet the recommended resources required to deploy the FortiSOAR Virtual Appliance.
2. Import and deploy the FortiSOAR Virtual Appliance using vSphere/vCenter or AWS.
3. FortiSOAR Configuration Wizard
   FortiSOAR Configuration Wizard runs automatically on the first `ssh` login by the `csadmin` user and performs the initial configuration steps that are required for FortiSOAR and performs tasks as is the case with the FortiSOAR enterprise OVA. It generates a Device UUID for your FortiSOAR instance. Use this Device UUID to generate the FortiSOAR license in FortiCare so that you can begin using FortiSOAR. Ensure that you generate a license that has its edition set to "MT_Tenant" or "MT_RegionalSOC".

Adding a tenant node on the master

On the top of the Tenants page, you will see the tenant ID of the master node, and also see Licensed as Master node text. You must use the tenant ID of the master when you are adding details of the master node on the tenants' node.

1. Log on to your FortiSOAR master node as an administrator and click the **Settings** icon to open the `System` page.
2. To add tenants associated with a master node, in the `Multi Tenancy` section, click **Tenants** in the left menu and on the `Tenants` page, click **Add**.
   To edit the configuration of an existing tenant, click the tenant whose configuration you want to update, which opens the agent record in the detail view. Update the configuration parameters as required, and click **Save**.
   If you no longer require an existing tenant you can deboard that tenant. Deboarding a tenant is an irreversible operation which also deletes all data related to that tenant from the master node. For more information see the *Deboarding Tenants* section in the Shared Tenancy chapter.
   You can deactivate agents associated with the tenants. To deactivate an agent, open the agent record, and clear the **Enabled** checkbox in that record.

3. In the `Add Tenant` dialog, configure the following parameters:
   a. From the **Choose Tenant Type** field choose the type of tenant you want to add.
      In this case, select **Dedicated** to add this tenant as a remote or dedicated tenant. Remote or Dedicated tenants have their own FortiSOAR instance and do not share the current FortiSOAR instance with the Master.
   b. In the **Tenant ID** field, enter the ID of the tenant, which is the Device UUID of the tenant's FortiSOAR instance.
   c. In the **Tenant Name** field, enter the name of the tenant.
   d. (Optional) In the **Description** field, enter the description of the tenant.
   e. From version 6.4.1 onwards, a default agent is automatically created and added to a dedicated tenant as part of the tenant creation process. For more information on agents, see the *Segmented Network support in FortiSOAR* chapter in the "Administration Guide."
      i. From the **Secure Message Exchange** drop-down list, choose the secure message exchange that you have configured as the secure channel using which you can relay information to your agent or tenant nodes.
      ii. From the **Owners** drop-down list, select the teams that you want to add as owners of the agent and of the records coming from this tenant and click **Link Team**. The teams that you have selected will be associated as owners of any record created for that tenant.
      As shown in the following image, the SOC Team and the Tenant L-01 team will be assigned to any record that comes from this tenant.



Add Tenant

Agent Configuration

A default agent would be created and linked to tenant as part of tenant creation

Select Secure Message Exchange*: Dev Router

RBAC

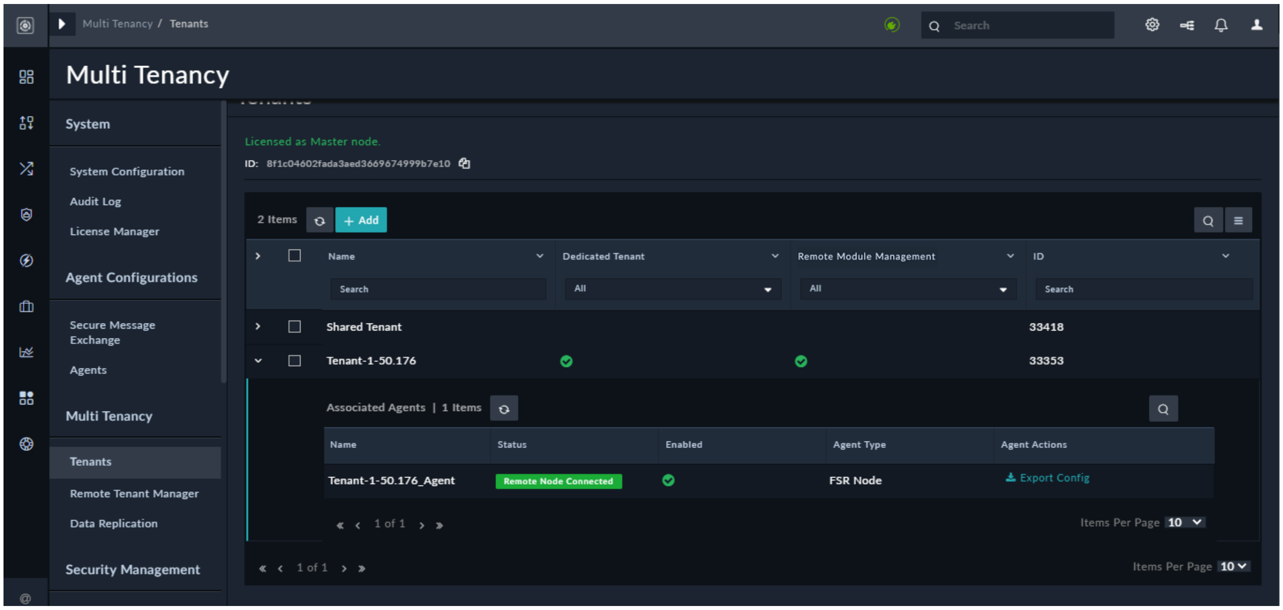
Owners

Select a Team | Link Team

SOC Team ✕ | Tenant L-01 ✕

Save | Cancel

4. To complete adding a new tenant associated with the master node, click **Save**.
   As tenants as just a wrapper, there will be no configuration state associated with a tenant. However, in case of dedicated tenants, a default agent is automatically created and added to a dedicated tenant as part of the tenant creation process. Therefore, you can expand the row of the tenant and check the connection between the agent and the master node using secure message exchange. Once the connection is established from the master node to the agent, the **Status** field displays "Remote Node Connected". To know more about the statuses, see the *List of Statuses* section in the Shared Tenancy chapter.



Licensed as Master node.

ID: 8f1c04602fada3aed3669674999b7e10

Name	Dedicated Tenant	Remote Module Management	ID
Shared Tenant			33418
Tenant-1-50.176	✓	✓	33353

Name	Status	Enabled	Agent Type	Agent Actions
Tenant-1-50.176_Agent	Remote Node Connected	✓	FSR Node	Export Config

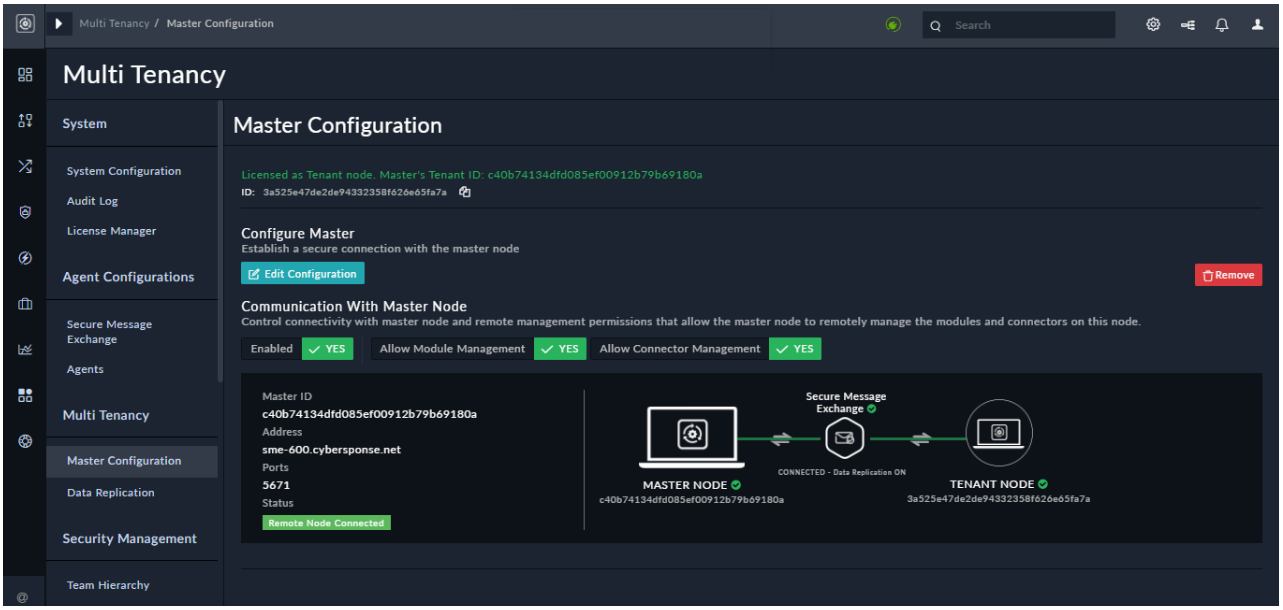
You can export the configuration of an agent in the JSON format using the **Export Config** link, in the agent row that represents the dedicated node and therefore contains the configuration of the tenant. This JSON file containing the configuration of the tenant can then be imported on the Master Configuration page of the tenant node on which you are configuring the secure message exchange, by clicking the **Import Master Config** link.

From FortiSOAR version 6.4.4 onwards, you might also see a **Warning** symbol in the Agents Action column, if the master node cannot remotely execute or manage connector actions on the agent node.

You can add multiple agents to a tenant by linking the respective agents to the tenant by opening the detail view record of the tenant and in the Owners section, click the **Agents** tab, and click **Link** to open the Link Agents dialog that contains a list of configured agents. Select the agent that you want to associate with the tenant and click **Save Relationship**. Note that the linking of agents to tenants is just representational, and RBAC is governed by the "Owners" that you specify for the agent.

Configuring the secure message exchange on a tenant to establish a secure connection between the master node and the tenant node

1. Log on to your FortiSOAR tenant node as an administrator and click the **Settings** icon to open the System page.
2. On the System page, you will see the Multi Tenancy section. Click the **Master Configuration** item on the left menu, to configure your tenant node.
   On the Master Configuration page, you will see the tenant ID of the tenant. You must use the Tenant ID of the tenant when you are adding details of the tenant node on the master node. You will also see the text such as:
   `Licensed as Tenant node: Master's Tenant ID <ID value>.`
   Tenant ID is the Device UUID of the tenant.
   Master's Tenant ID is the Device UUID of the master's FortiSOAR instance.
   Important: To configure the master node, you must have a role that has a minimum of Create, Read, and Update permissions on the **Secure Message Exchange** and **Tenants** modules.



Licensed as Tenant node. Master's Tenant ID: c40b74134dfd085ef00912b79b69180a

ID: 3a525e47de2de94332358f626e65fa7a

Configure Master

Establish a secure connection with the master node

Communication With Master Node

Control connectivity with master node and remote management permissions that allow the master node to remotely manage the modules and connectors on this node.

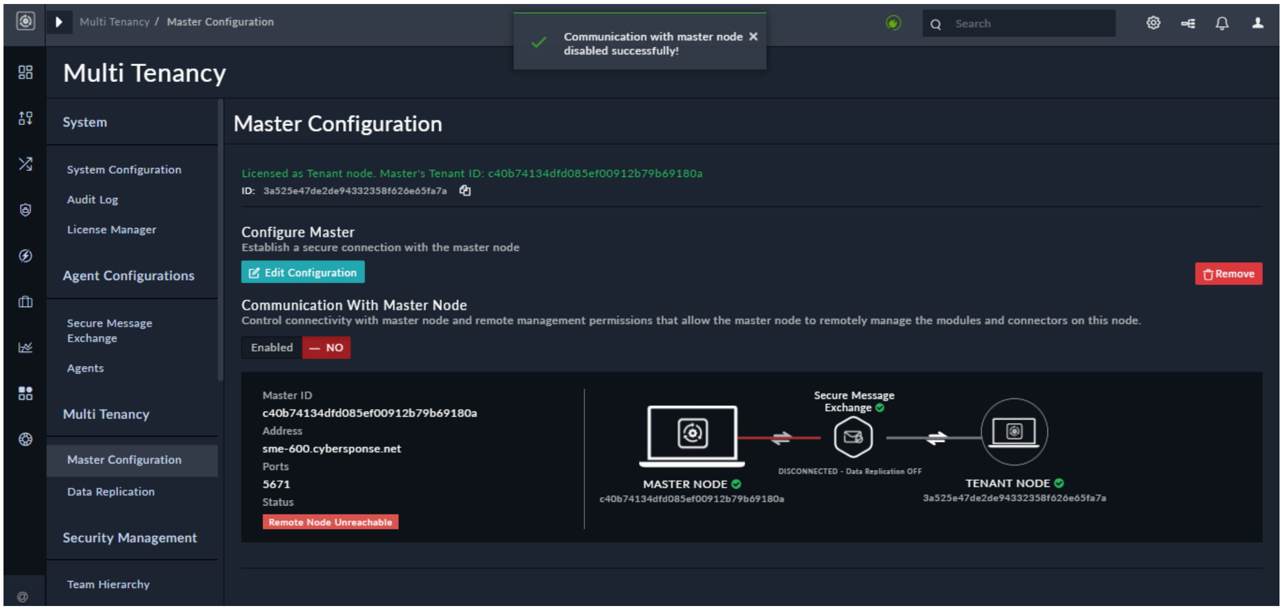
Enabled ✓ YES | Allow Module Management ✓ YES | Allow Connector Management ✓ YES

Master ID
c40b74134dfd085ef00912b79b69180a
Address
sme-600.cybersponse.net
Ports
5671
Status
Remote Node Connected

The `Configure Master` section displays the master and secure message exchange configuration details, and the status of configuration. You can see the following labels, in the `Status` field and in the image that displays the connection between the master and the secure message exchange. These labels signify the connection status between the master and secure message exchange:

- **Not Configured**: Secure Message Exchange has not been configured.
  This will display **Inactive** in the image displaying the connection between master and secure message exchange.
- **Configuration In Progress**: Configuration of the secure message exchange is in process.
  This will display **Inactive** in the image displaying the connection between master and secure message exchange.
- **Configured**: Connection between the master and secure message exchange is established without errors.
  This will display as **Connected - Data Replication ON** in the image if data replication is on from this tenant node to the master node.
  This will display as **Connected - Data Replication OFF** in the image if data replication is turned off from this tenant node to the master node.
- **Configuration Failed**: Connection between the master and secure message exchange is not established, due to either, wrong configuration or network failure.
  This will display **Disconnected** in the image displaying the connection between master and secure message exchange.

3. (Optional) To stop communication and replication of data, from this tenant to the master, click the **YES** button appearing beside **Enabled**. Disabling stops the master from receiving any data from this tenant and now the master cannot remotely manage modules and connectors of this tenant and you will see **NO** in the **Enabled** field, as shown in the following image:



Communication with master node disabled successfully!

Master Configuration

Licensed as Tenant node. Master's Tenant ID: c40b74134dfd085ef00912b79b69180a

ID: 3a525e47de2de94332358f626e65fa7a

Configure Master

Establish a secure connection with the master node

Communication With Master Node

Control connectivity with master node and remote management permissions that allow the master node to remotely manage the modules and connectors on this node.

Master ID: c40b74134dfd085ef00912b79b69180a

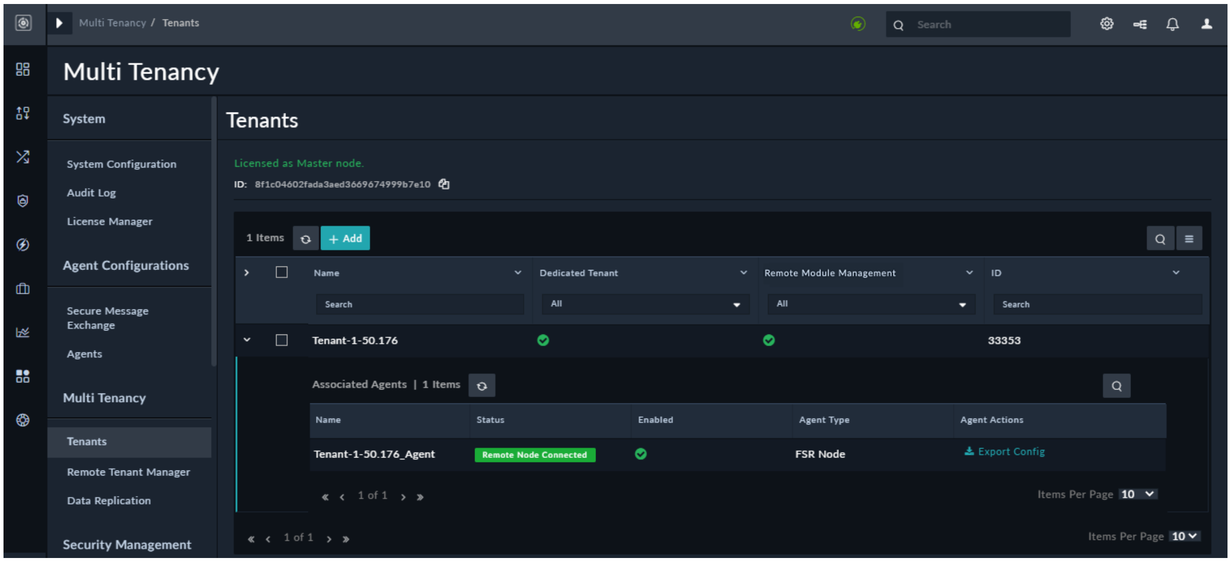
Address: sme-600.cybersponse.net

Ports: 5671

Status: Remote Node Unreachable

This global enable/disable button stops the communication from the tenant to the secure message exchange, and the tenant can no longer send data. However, data coming from the master is yet available at the secure message exchange, and once the tenant node resumes communication, the data from the master is available for consumption by the tenant. The tenant node continues to receive data from the master if sharing of data is enabled from the master node to the tenant node. You can use this option to suspend external communication from the tenant to the secure message exchange/master in cases such as, a planned downtime for regular maintenance. Once such activity is completed the tenant node can enable the communication again and replay the stored messages.

4. (Optional) To disallow the master to push picklists and make changes to its (the tenant's) model metadata (MMD) and push those changes to the tenant node, i.e., to disallow the master to make changes to the tenant's mmd, click the **YES** button appearing beside **Allow Module Management**, which turns this button to **NO**.
5. (Optional) To disallow the master the ability and permissions to remotely execute and manage the connector actions at the tenant node, click the **YES** button appearing beside **Allow Connector Management**, which turns this button to **NO**. For more information, see the Managing connectors of distributed tenants section.
6. (Optional) To delete the configuration of the master node from a tenant node, click **Remove**.
   Note: To remove configurations of the master from the tenant node, you must have `Read` and `Delete` permissions on the `Secure Message Exchange` module.
7. To configure the secure message exchange on the tenant and establish a secure connection with the master, click **Edit Configuration** in the `Configure Master` section, and choose one of the following methods to complete the configuration:
   a. Export the configuration of the tenant in the `JSON` format, from your master node, by clicking the **Export Config** link in the agent row. The agent that has the **Export Config** link represents the dedicated node, and therefore contains the configuration of the tenant and its `Agent Type` will be set as "FSR Node". You can view the agents added to the tenant by expanding the tenant row. If the tenant has multiple agents associated with it, then only the dedicated node will have the **Export Config** link, all remaining agents will have the **Download Installer** link and their `Agent Type` will be set as "FSR Agent".
   You can then import this JSON file on the `Master Configuration` page of the tenant node on which you are configuring the secure message exchange, by clicking the **Import Master Config** link.



Tenants

Licensed as Master node.

ID: 8f1c04602fada3aed3669674999b7e10

		Name	Dedicated Tenant	Remote Module Management	ID
	□	Search	All	All	Search
⌄	□	Tenant-1-50.176	✓	✓	33353

Name	Status	Enabled	Agent Type	Agent Actions
Tenant-1-50.176_Agent	Remote Node Connected	✓	FSR Node	Export Config

OR

b. Add the details manually:

i. To add a secure message exchange, click **Add Master** on the `Master Configuration` page.
To edit an existing secure message exchange configuration, click **Edit Configuration** in the `Configure Master` section.

ii. In the **Configure Master** dialog, configure the following parameters:
Note: The **Master ID** is a read-only field that displays the Device UUID of the master's FortiSOAR instance.

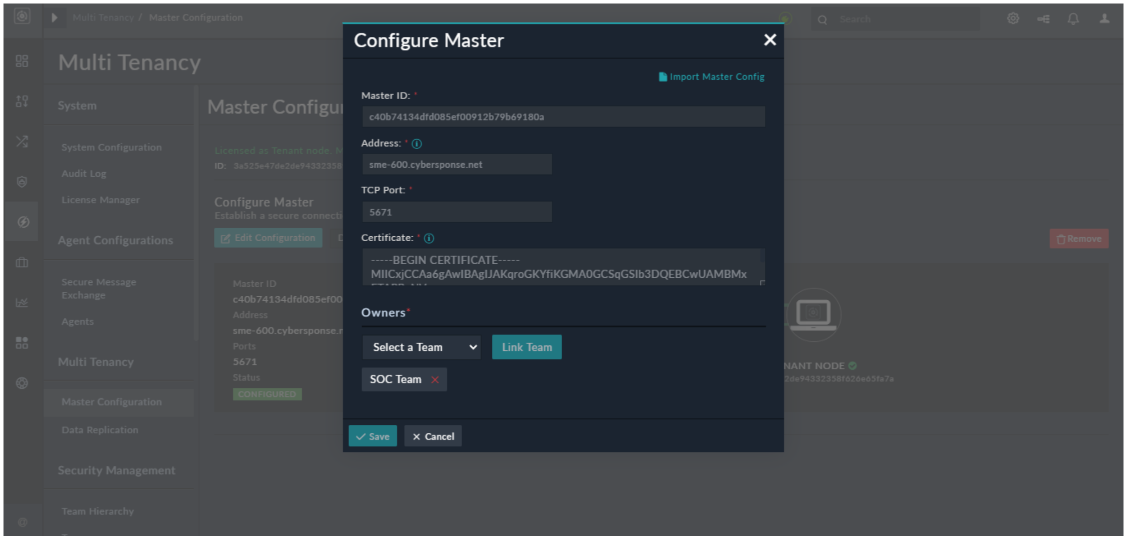
i. In the **Address** field, enter the IP address of the secure message exchange, which should match the IP address that you have specified when configuring the secure message exchange.

ii. In the **TCP Port** field, enter the TCP port number that you had specified when configuring the secure message exchange for this tenant. By default, it is set as `5671`.

iii. In the **Certificate** field, copy-paste the certificate text of the Certificate Authority (CA) that has signed the secure message exchange certificate in the `pem` format. If it is a chain, then the complete chain must be provided.
By default, the CA certificate for the FortiSOAR self-signed certificate is present at the following location on the secure message exchange:
`/opt/cyops/configs/rabbitmq/ssl/cyopsca/cacert.pem`.
Important: If in the future, your secure message exchange certificate expires, and you need to deploy a new certificate, then the new certificate must be copied back to the master node as well as the tenant's router entry.

iv. From the **Owners** drop-down list, select the teams that you want to add as an owner to the records coming from this tenant, and click **Link Team**. The teams that you have selected will be associated as owners of any record created for that tenant on the master, either manually or using an Custom API Endpoint trigger.
Note: If a record is created on the master manually, then the teams that are selected in the **Owners** drop-down list will be associated as owners of that record. Similarly, if any record is created using an *Custom API Endpoint trigger*, then that record will also be owned by the teams selected in this **Owners** drop-down list, irrespective of whether these are the teams assigned to the appliance

(**Settings** > **Appliances** > **Team and Role**).



Configure Master

Import Master Config

Master ID: * c40b74134dfd085ef00912b79b69180a

Address: * sme-600.cybersponse.net

TCP Port: * 5671

Certificate: *

```
-----BEGIN CERTIFICATE-----
MIICxjCCAa6gAwIBAgIJAKqroGKYfiKGMA0GCSqGSIb3DQEBCwUAMBMx
```

Owners*

Select a Team | Link Team

SOC Team ✕

✓ Save | ✕ Cancel

v. To save the configuration for the master node on a tenant node, click **Save**.
Once you click save, FortiSOAR checks the connection between the master and secure message exchange, and if a connection can be established without errors, the status is displayed as **Connected**.

Replicating data between the master and the tenant nodes

To leverage bidirectional synchronization of data for updates between the master and the tenant node, you must enable data replication of modules on both the master and the tenant nodes. Tenants can configure and control the data that is shared with the master node.

From version 7.0.0 support has been added for file replication between the master and tenant nodes. Therefore, records containing the "file" type fields or records with correlations that map to modules containing "file" type fields also get replicated. For example, now you can replicate 'Alert' records that contain 'Attachments' correlations.

In case a conflict is observed between the master copy and the tenant copy for record(s), the tenant copy of the data will be retained.

On the `Data Replication` page, you will see the list of modules. You can control the data that you want to share at the module level and also at the field level, enabling you to maintain data security and ensuring that you do not share sensitive data.

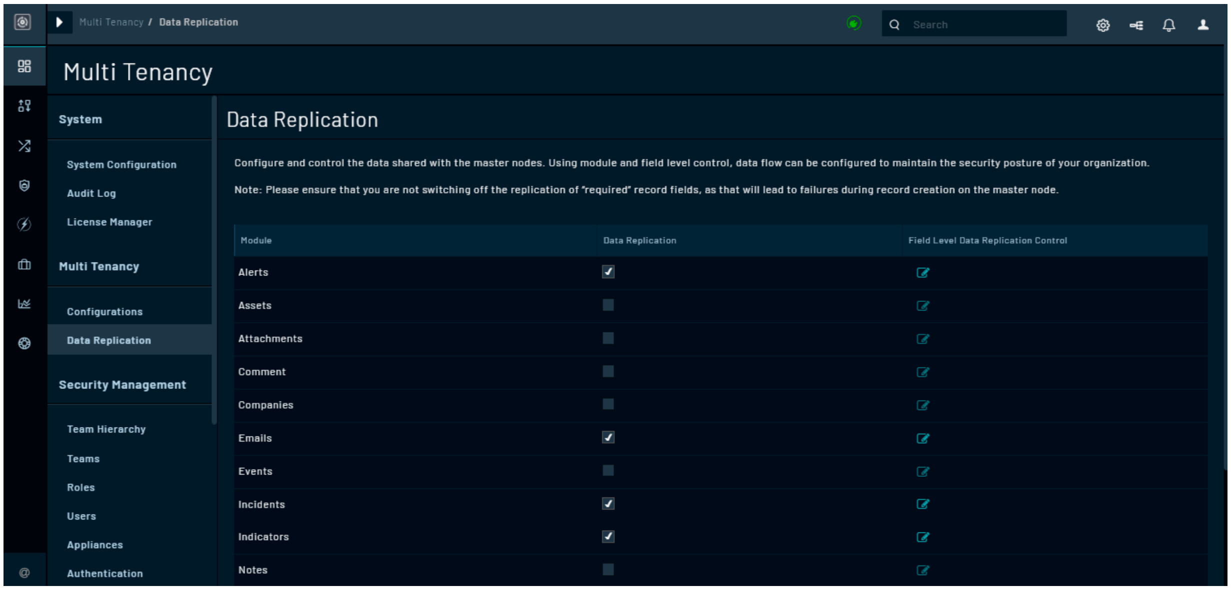
Notes on data replication:

- On the tenant node, you can configure and control what data should be shared with the master node. Using field controls, you can choose the fields that you want to share with the master node, thereby ensuring the security of your sensitive data. On the Tenant node, by default, the `Alerts`, `Emails`, `Incidents`, and `Indicators` modules and all its fields are marked as replicable.
- On the master node, you can configure and control what data should flow from the tenant nodes. Records from only those modules that are marked for data replication are replicated from the tenant nodes. On the Master node, post-initial license deployment, all existing modules along with all their fields are marked as replicable, except for the `People` module.

- You can mark any module as replicable on their peers by selecting the module and clicking the **Data Replication** checkbox on the `Modules` page (**Application Editor** > **Modules**).

On the tenant node

1. Log on to your FortiSOAR tenant node as an administrator and click the **Settings** icon to open the `System` page.
2. In the `Multi Tenancy` section, click **Data Replication** in the left menu to display the `Data Replication` page.
   Use the `Data Replication` page to configure the following:
   a. To share records of a module, in that particular module's row, select the checkbox in the **Data Replication** column.
      This sets up data replication between the tenant node and the master node for the particular module.



Multi Tenancy

Data Replication

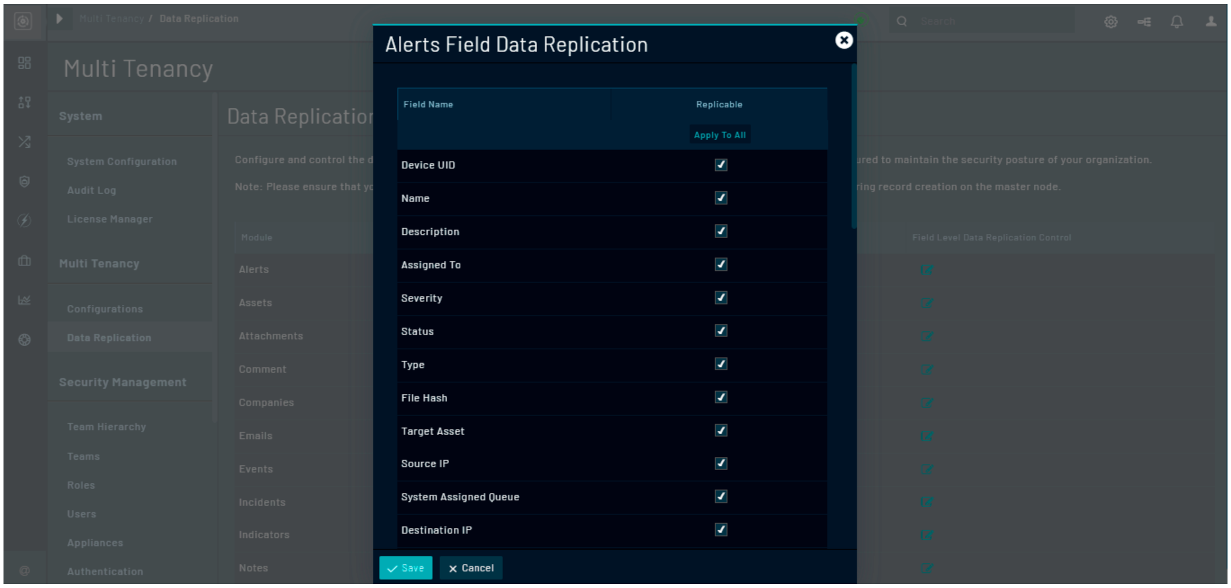
Configure and control the data shared with the master nodes. Using module and field level control, data flow can be configured to maintain the security posture of your organization.

Note: Please ensure that you are not switching off the replication of "required" record fields, as that will lead to failures during record creation on the master node.

Module	Data Replication	Field Level Data Replication Control
Alerts	☑	✎
Assets	□	✎
Attachments	□	✎
Comment	□	✎
Companies	□	✎
Emails	☑	✎
Events	□	✎
Incidents	☑	✎
Indicators	☑	✎
Notes	□	✎

   b. To specify the fields that should be replicated between the tenant node and the master node, click the **Edit** icon in the **Field Level Data Replication Control** column to display the **Field Data Replication** dialog.
      Note: By default, all fields of a module are set up as replicable.
      The `Field Data Replication` dialog contains a list of fields that are part of the selected module. To remove data replication for a particular field, clear the checkbox that appears in the **Replicable** column in the field's row.
      Important: Ensure that you do not switch off the replication of the “required” record fields, as that will lead to the

failure of record creation on the master node.



Alerts Field Data Replication

Field Name	Replicable Apply To All
Device UID	☑
Name	☑
Description	☑
Assigned To	☑
Severity	☑
Status	☑
Type	☑
File Hash	☑
Target Asset	☑
Source IP	☑
System Assigned Queue	☑
Destination IP	☑

3. To save your changes, click **Save**.

Custom Module Considerations

When you create a custom module on a tenant node, you must consider the following points:

On the Tenant Node

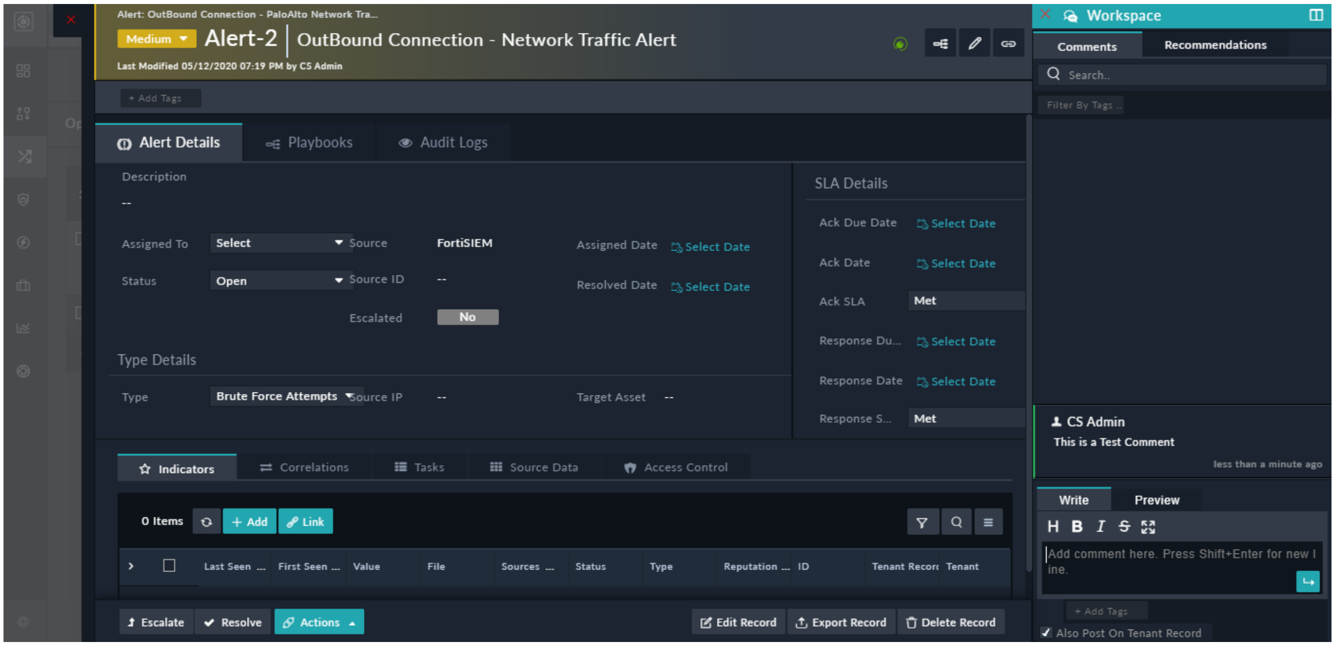
- By default, replication is turned off (for both master and tenant) for any custom module you create on the tenant node. Therefore, if you want to replicate the fields of the tenant node to the master node, you can enable replication of that module by setting data replication to true at the module level. You can set data replication to true by selecting the module and clicking the **Data Replication** checkbox on the `Modules` page (**Application Editor** > **Modules**).

On the Master node

- Ensure that the custom module that has been created on the tenant node is present on the master node, with data replication enabled for all the required fields. You can achieve this by exporting the custom module from the tenant node and importing the same into the master node. You must ensure that all the relationships of the custom module are kept intact during the export-import process.
- Ensure that the custom picklists are exported and imported correctly into the master node.

Posting of comments from the master node to the tenant node

Comments get pushed from the tenant node to the master node by default. However, if you add a comment on the master node in a tenant record, which you want to push onto the tenant node, then click the **Also Post On Tenant Record** checkbox in the collaboration panel of the record on the master node. Once you click the **Also Post On Tenant Record** checkbox, then the comment appears with a shared icon on the master node:



Description: --

Assigned To: Select | Source: FortiSIEM | Assigned Date: Select Date

Status: Open | Source ID: -- | Resolved Date: Select Date

Escalated: No

Type Details

Type: Brute Force Attempts | Source IP: -- | Target Asset: --

SLA Details

Ack Due Date: Select Date

Ack Date: Select Date

Ack SLA: Met

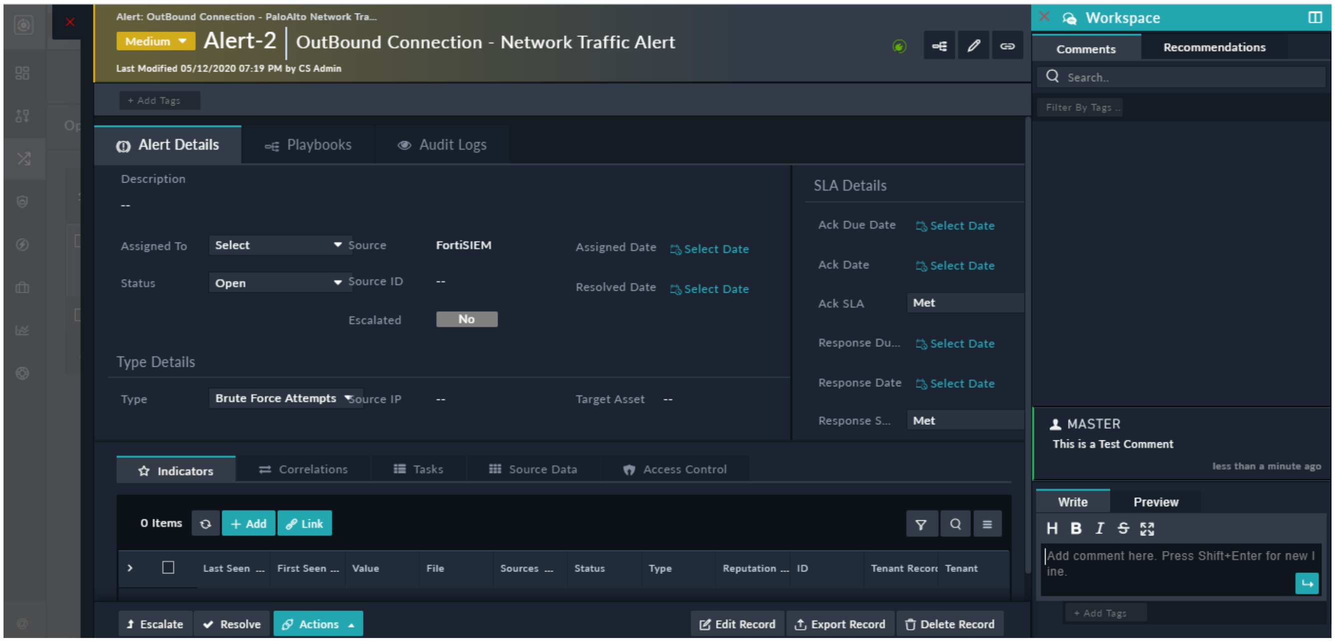
Response Du...: Select Date

Response Date: Select Date

Response S...: Met

CS Admin
This is a Test Comment
less than a minute ago

On the tenant node the pushed comment appears with the "Master" label:



Description: --

Assigned To: Select | Source: FortiSIEM | Assigned Date: Select Date

Status: Open | Source ID: -- | Resolved Date: Select Date

Escalated: No

Type Details

Type: Brute Force Attempts | Source IP: -- | Target Asset: --

SLA Details

Ack Due Date: Select Date

Ack Date: Select Date

Ack SLA: Met

Response Du...: Select Date

Response Date: Select Date

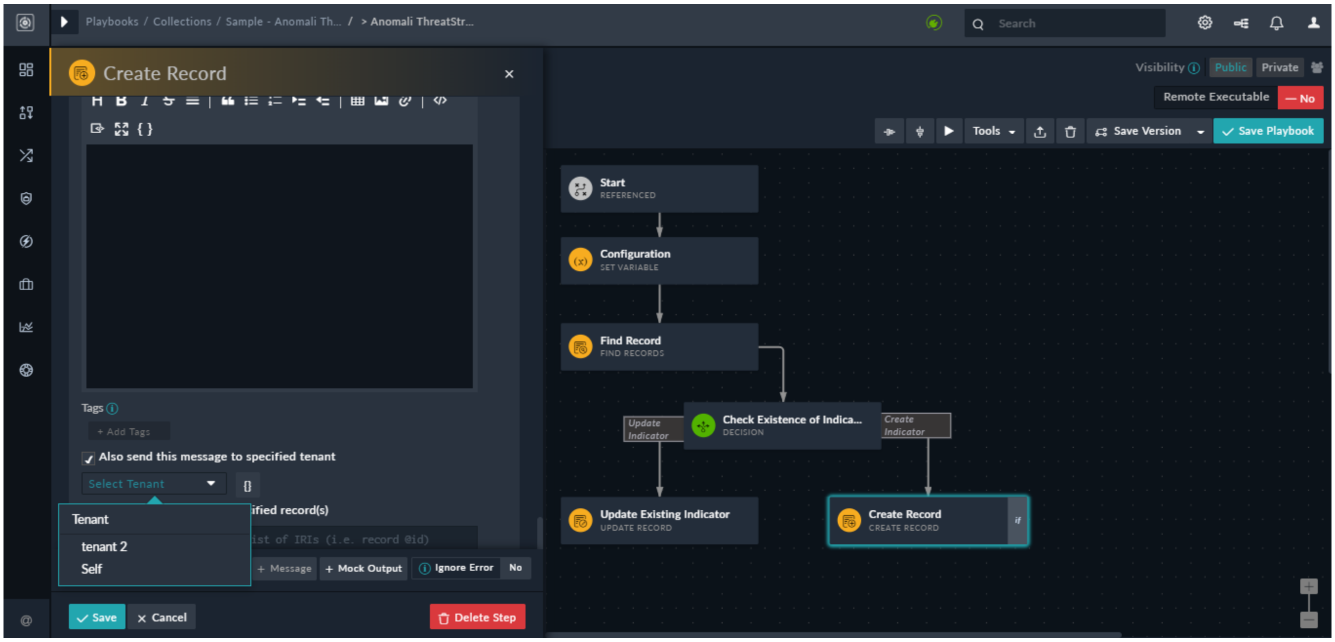
Response S...: Met

MASTER
This is a Test Comment
less than a minute ago

From version 7.0.0 onwards, you can add attachments (files) to comments and those comments along with the associated files get replicated between the respective master and tenant nodes.

Users with appropriate permissions can edit or delete comments and the same will be reflected on the respective master or tenant nodes, i.e., if a comment is edited on a master node, the same will be reflected on the tenant node. For more information on editing comments, see the *Working with Modules - Alerts & Incidents* chapter in the "User Guide."

If you have a playbook containing steps with "Messages" being added to the record that triggers the playbook on the master node, you can choose to replicate the comments that are linked to the record on the tenant node, so that a user on the tenant node can follow the investigation that is being conducted on the record. To replicate comments on the tenant node, click the **Also send this message to specified tenant** checkbox, and from the **Select Tenant** drop-down list, select the tenant node on which you want to replicate the comments or click **{}** to specify tenant IRIs in this field.



Create Record

Tags

☑ Also send this message to specified tenant

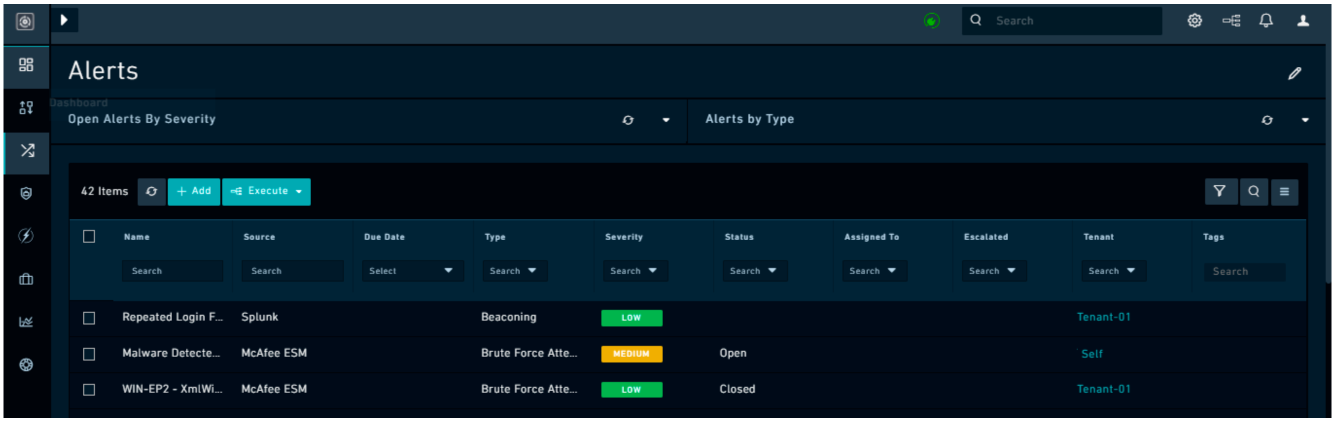
Select Tenant ▾ {}

Tenant
tenant 2
Self

On the Master node:

Ensure that the modules that require to be replicated and synchronized bidirectionally are enabled for replication on the master node. By default, all modules are marked as replicable on the master node. If you want to change this configuration follow the same steps on the master node, as mentioned above for the tenant node.

On the master node, users can identify records using the *UUID* of the record, which is common for both master and tenants' records. You can get the UUID of the record by clicking the record, and the UUID will be shown on the URL bar in the format: `/view-panel/<moduleName>/UUID?`. Also, on the master node, in the **Grid** or **Listing** view, each module that is replicable will have **Tenant** as a column, by default, as shown in the following image:



Alerts

42 Items

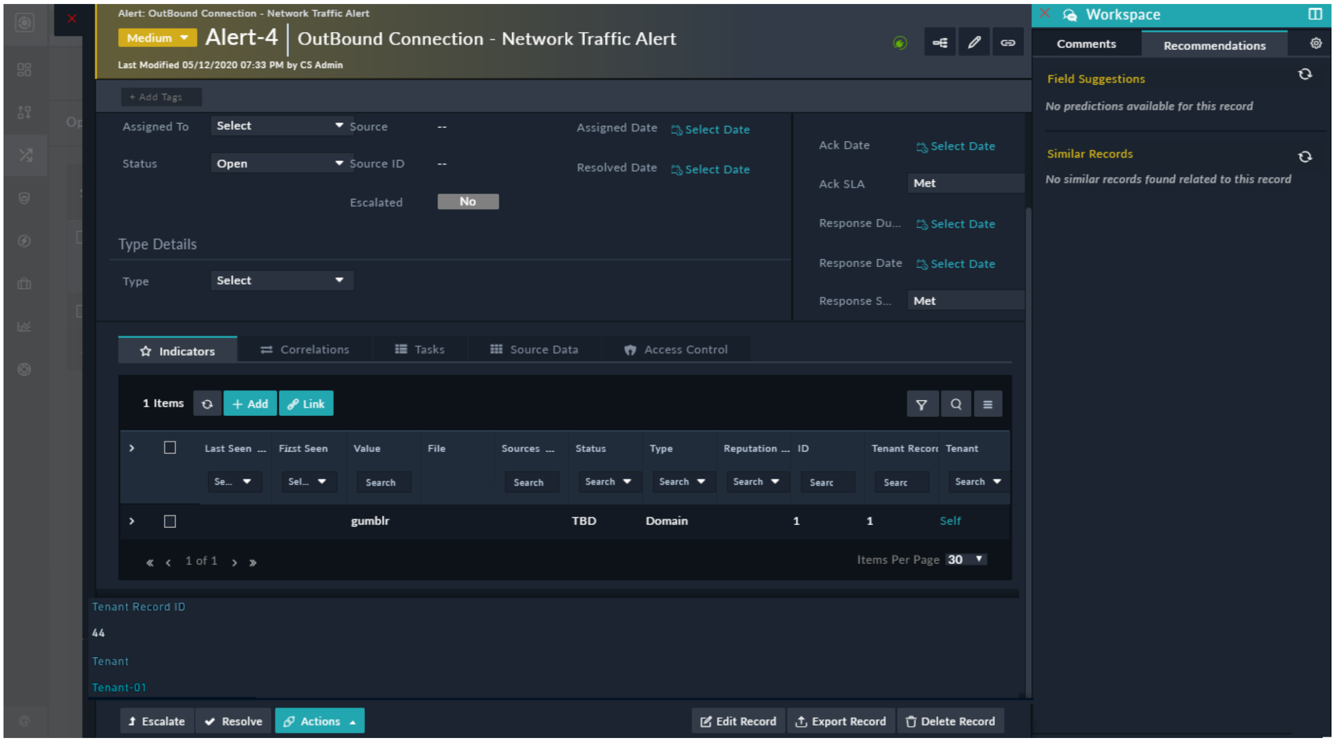
Name	Source	Due Date	Type	Severity	Status	Assigned To	Escalated	Tenant	Tags
Repeated Login F...	Splunk		Beaconing	LOW				Tenant-01	
Malware Detecte...	McAfee ESM		Brute Force Atte...	MEDIUM	Open			Self	
WIN-EP2 - XmlWi...	McAfee ESM		Brute Force Atte...	LOW	Closed			Tenant-01	

If no tenant is assigned while adding the alert, i.e., if the alert is created on the master, then that record is created as a "`Self`" record, i.e., **Self** will appear in the Tenant column.

For modules that are being replicated, it is important that assignments, SLA computations, and other On Create or On Update playbooks are active only on one of the nodes (either master or tenant). Not doing so can lead to unnecessary data traffic and conflict on the records due to simultaneous updates on both the nodes, since updates on each record are synced bidirectionally.

When a record is replicated from a tenant node onto the master node, in the detail view of the record, a new field named **Tenant Record ID** is added and populated with the ID of the record on the tenant node. When a record is replicated from a tenant node to the master node, its ID differs on both the nodes. Therefore, knowing the ID of the record on the tenant node makes it easier to identify the replicated record.

For example, in the following image, the ID of the alert on the master node is Alert-44, and the ID of the record on the tenant node is 44 and the tenant is Tenant-01, thus making it easier to identify the alert record:



Alert: OutBound Connection - Network Traffic Alert

Medium Alert-4 | OutBound Connection - Network Traffic Alert

Last Modified 05/12/2020 07:33 PM by CS Admin

Assigned To: Select

Source: --

Assigned Date: Select Date

Status: Open

Source ID: --

Resolved Date: Select Date

Escalated: No

Ack Date: Select Date

Ack SLA: Met

Response Du...: Select Date

Response Date: Select Date

Response S...: Met

Type Details

Type: Select

1 Items

Last Seen ...	First Seen	Value	File	Sources ...	Status	Type	Reputation ...	ID	Tenant Record	Tenant
		gumblr			TBD	Domain		1	1	Self

1 of 1

Items Per Page 30

Tenant Record ID

44

Tenant

Tenant-01

Workspace

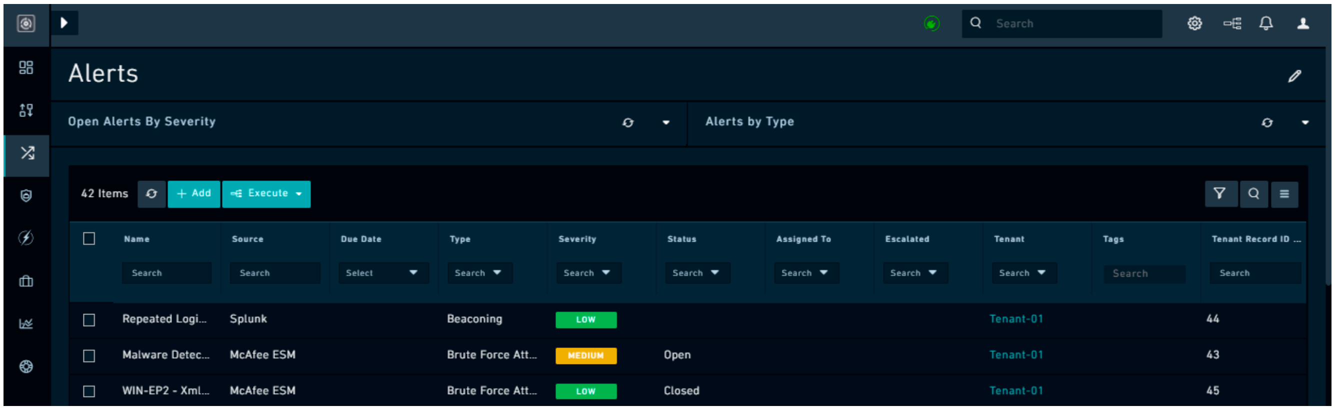
Field Suggestions

No predictions available for this record

Similar Records

No similar records found related to this record

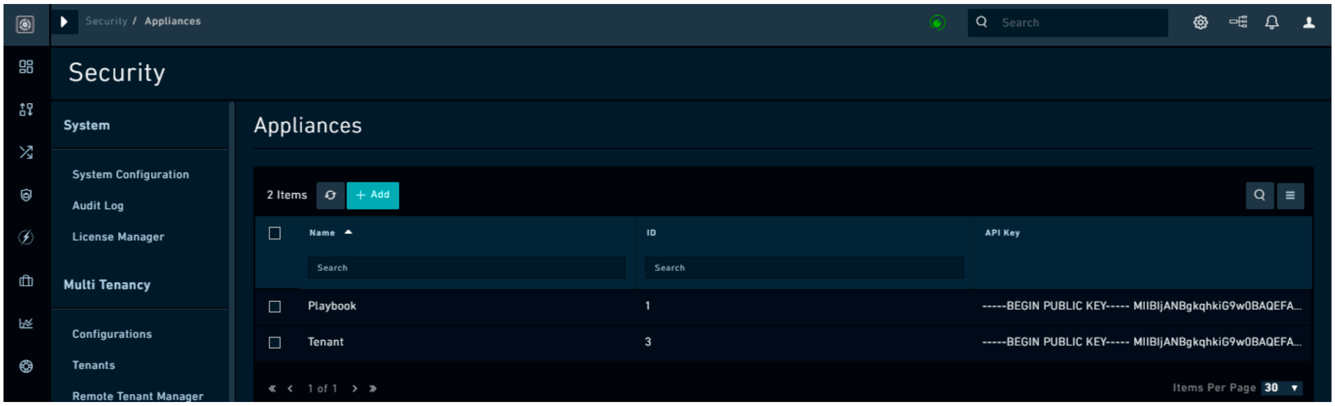
Similarly, in the grid view of the alert module on the master node, you will see a **Tenant Record ID** column that lists the IDs of the tenant records:



☐	Name	Source	Due Date	Type	Severity	Status	Assigned To	Escalated	Tenant	Tags	Tenant Record ID ...
	Search	Search	Select	Search	Search	Search	Search	Search	Search	Search	Search
☐	Repeated Logi...	Splunk		Beaconing	LOW				Tenant-01		44
☐	Malware Detec...	McAfee ESM		Brute Force Att...	MEDIUM	Open			Tenant-01		43
☐	WIN-EP2 - Xml...	McAfee ESM		Brute Force Att...	LOW	Closed			Tenant-01		45

Required permissions for data replication

The FortiSOAR Routing Service hosted at each of the nodes uses an `Appliance` User for applying updates coming from the remote node. This Appliance User is displayed as the `Master` User on a tenant node and as the `Tenant` user on the master node. For all records updated from one node to the other, this appliance user should have the necessary permission on the replicated modules to create and update the records.



☐	Name	ID	API Key
	Search	Search	
☐	Playbook	1	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFA...
☐	Tenant	3	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFA...

Default Ownership of records replicated from the tenant nodes to the master

For records replicated from a tenant node to the master node, the record would have the same ownership as that for the tenant by default. Therefore, to define which team should have visibility of alerts coming in from a particular tenant, assign the specific team as the owner of the particular tenant. In addition, the records will be visible to the `agent` appliance teams. You can assign owners of the records belonging to the tenant directly when you are adding tenants on the master node, see the Adding a tenant node on the master section.

Resolving of conflict of data between the master and tenant nodes

Post creation of a record, only the changed or updated attributes of the record are synced bidirectionally. However, when a record is updated simultaneously at both the nodes, it could lead to conflicts and override the updates at the respective nodes. If there is such a conflict, then the tenant copy of the data will be retained.

To address such scenarios, when an update to a record, which is updated at the master node, arrives at the corresponding tenant node, the routing service compares the modified time of the record at the remote node against the last modified time of the record at the local node. If the local modified time is found to be greater, the update from the master node is rejected, and the record is marked for conflict. Subsequently, a complete sync of the latest copy of the record at the tenant node is automatically sent to the master node.

Since the modified time of the records at the distributed nodes are compared, it is important that the time at the nodes are in sync with a common NTP server. For example, time.apple.com.

Handling of network outages or loss of connectivity between a node and the secure message exchange

When a node loses connectivity to the secure message exchange, all outbound messages are stored locally by the routing service hosted on the node. This replicator log is automatically replayed once the connectivity is restored. All inbound messages from other nodes to the said node are persisted at the secure message exchange. These messages get automatically delivered once the connectivity is restored.
For more information, see the Troubleshooting chapter.

Handling a mismatch in the module metadata of replicated modules

When a module, say the `Alert` module, is configured for replication from the tenant node to the master node, the module replication would work seamlessly, if the metadata at both the nodes is identical. However, if there is an MMD (module metadata) difference or in case of custom picklists, the following must be ensured:

- If there is a difference in the module metadata at the two nodes, or if some fields are excluded from replication, these must be fields that are "not-required" for record creation. Else, record replication to the target node will fail due to missing required fields.
- For custom picklists added to a module being replicated, the picklist item IDs should be the same at both the nodes. You can ensure this by exporting picklists from one node and importing them to the other node using the `Configuration Manager`. For more information, see the *Application Editor* chapter in the "Administration Guide."
  Creating the picklists manually at each node would result in a difference in the IDs. Since the record replication works on record IDs, the picklist values would not be correctly replicated, if there is a mismatch in the IDs.

Whenever a record replication fails to get applied at the target node due to the reasons mentioned above, an entry will be created in the Audit Logs with the type `Replication Failure`. These audit entries can also be seen on the respective records if any update from a remote node fails to be applied due to any metadata changes done later.

Managing connectors of distributed tenants

Prior to version 6.4.4., all the connector installations or configurations done at the distributed tenant node replicated back to master, by default. To disallow the master from remotely executing connector actions on the tenant, the `ENABLE_REMOTE_CONNECTOR_OPERATION` parameter required to be set to `false` in the agent's (associated with the tenant) rpm `config.ini` file (`/opt/cyops-integrations/integrations/configs/config.ini`). The `ENABLE_REMOTE_CONNECTOR_OPERATION` parameter is set to `true` by default. However, in this case, the master node did not get notified that its tenant had turned off this remote operation setting. Due to which if the master triggers any remote request then that would get ignored by the tenant since the remote operation had been disallowed. Therefore, from version 6.4.4 onwards, you can allow or disallow the master from managing and executing connector actions, including actions executed using remotely executable playbooks ("Reference Remote Playbooks") on the tenant node, using the FortiSOAR UI. Remote connector management is achieved using the **Allow Connector Management** button on the on the `Master Configuration` page of the tenant node. To remove the master's ability and permissions to remotely manage and execute connectors on the tenant node, toggle the **Allow Connector Management** button to **NO**.

To disallow the master from remotely executing connector actions on an agent, ensure that the agent's version must be 6.4.4 and later

Once you set the **Allow Connector Management** button to **NO**, the master node is notified and the master node can no longer send remote connector requests to such tenants.

If you have upgraded a tenant node to version 6.4.4 and later from a version prior to 6.4.4, in which you had changed the value of the `ENABLE_REMOTE_CONNECTOR_OPERATION` parameter in the `config.ini` file from `true` to `false`, then the value (`false`) will not be honored, i.e., after the upgrade the master will yet be able to remotely manage connectors on the tenant node. This is because in the case of tenant nodes, the value of the `ENABLE_REMOTE_CONNECTOR_OPERATION` parameter in the `config.ini` file has no effect from version 6.4.4 onwards, as post version 6.4.4 , remote connector management is handled by the **Allow Connector Management** button.

Managing Remote Tenants

The "Remote Tenant Manager" helps the master remotely manage playbooks, pushing and mapping playbooks and tenants' data, including picklist and modules. The master node can make changes to the tenants' model metadata (MMD) and push those changes on the tenant node.

Pushing playbooks from the master node for remote execution

The playbook mapping page displays a list of playbook aliases and their mappings to various tenants. A playbook alias makes it easier to reference tenant playbooks in the **Remote Reference Playbook** step in the playbook designer. For example, you might have a collection of "Remediation" playbooks, such as playbooks to block an IP address and a playbook to check the reputation of an IP address using a 3rd party threat intelligence tool such as Anomali ThreatStream. Or you might also have a number of "Investigate BFA" playbooks such as, Investigate BFA using Splunk, Investigate BFA using QRadar, or Investigate BFA using LogRhythm, which could map to different tenants having or using different SIEM products. You can use the Playbook Mapping page to push the playbook collection to any or all of your tenant nodes.

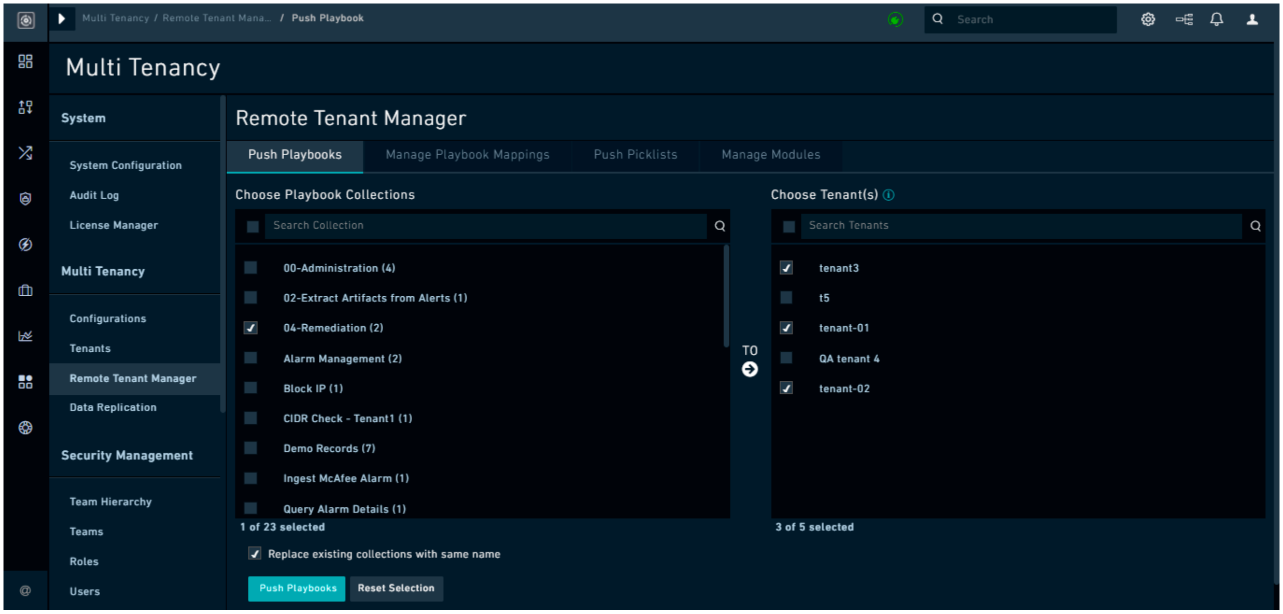
To create an alias, you must ensure that the **Remote Executable** setting for the playbook is enabled in the playbook designer. For more information, see the Executing remote playbooks at the tenant node from the master node section.

When a playbook is pushed from the master node to the tenant node, then if you have assigned a playbook step, for example, manual input step, to a specific user or team, then the ownership of that playbook step gets reset.

To push playbook collections on the tenant nodes, do the following:

1. Log on to your FortiSOAR master node as an administrator and click the **Settings** icon to open the `System` page.
2. In the `Multi Tenancy` section, click **Remote Tenant Manager** in the left menu.
3. Click the **Push Playbooks** tab.
   The `Push Playbooks` page displays a list of playbook alias and their mappings to various tenants. A playbook alias makes it easier to reference tenant playbooks in the **Remote Reference Playbook** step in the playbook designer. To create an alias, you must ensure that the **Remote Executable** setting for the playbook is enabled in the playbook designer. For more information, see the Executing remote playbooks at the tenant node from the master node section.
   For example, you might have a collection of Remediation playbooks that you want to push to any or all of your tenants.

4. On the `Push Playbooks` page, from **Choose Playbook Collections**, select the playbook collection(s) that you want to push on tenant nodes from the master node.
5. From **Choose Tenant(s)**, select the tenant(s) to whom you want to push the playbooks and click the **TO** arrow.
   The **Choose Tenant(s)** list lists only those tenants that are active and whose connections are verified.
   You can also search for playbook collections and tenants using the search functionality.



Choose Playbook Collections

- □ 00-Administration (4)
- □ 02-Extract Artifacts from Alerts (1)
- ☑ 04-Remediation (2)
- □ Alarm Management (2)
- □ Block IP (1)
- □ CIDR Check - Tenant1 (1)
- □ Demo Records (7)
- □ Ingest McAfee Alarm (1)
- □ Query Alarm Details (1)

1 of 23 selected

Choose Tenant(s)

- ☑ tenant3
- □ t5
- ☑ tenant-01
- □ QA tenant 4
- ☑ tenant-02

3 of 5 selected

☑ Replace existing collections with same name

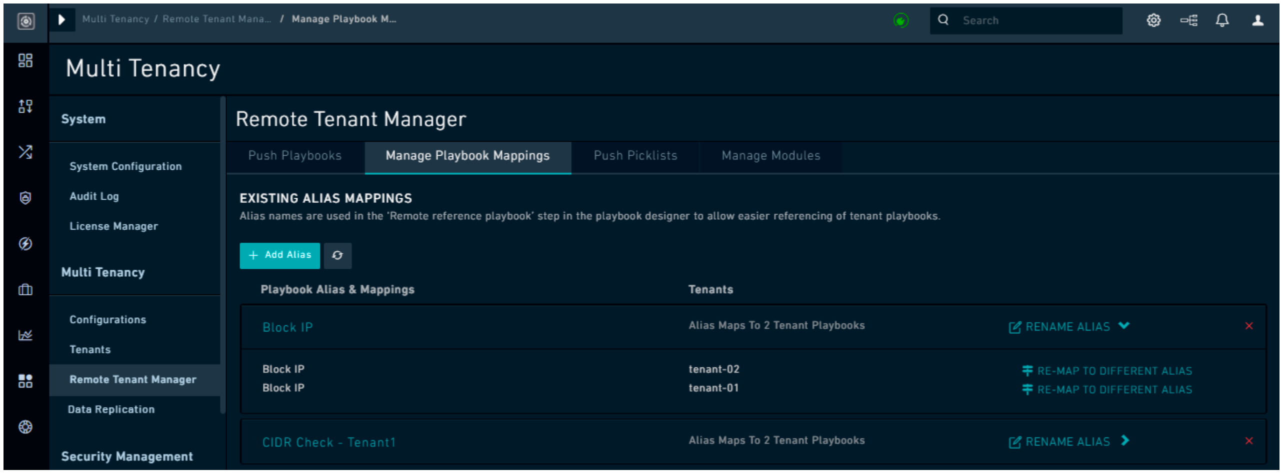
6. By default, the **Replace existing collections with same name** checkbox is selected, which signifies that if there is an existing playbook collection with the same name (ID) on the tenant node, then that playbook collection will get replaced on the tenant node.
   In case you want to keep the existing playbook collection on the tenant node, then you must clear the **Replace existing collections with same name** checkbox. In this case, the existing playbook collection is not replaced on the tenant node.
7. To push the mapped playbooks on the tenant nodes, click **Push Playbooks**.
   The pushed playbook collections are maintained on the master node, and they are cloned and mapped on the tenant node, i.e., the Push action creates a copy of playbook on the tenant node. Therefore, the playbook on the master and the tenant copy is maintained separately on the respective nodes.
   If any changes are required to the playbooks, then playbook(s) can be updated in the following ways:
   Changes can be made on the master node and then pushed again to the tenant node. Ensure that you select the **Replace existing collections with same name** checkbox in this case.
   Changes can be made on the tenant node itself. The updated playbook on the tenant node will not replace the playbook on the master node.

Managing Playbook Mappings

You can manage your playbook mappings by adding alias names that facilitates referencing of a tenant playbook or you can also remap an existing alias.

1. Log on to your FortiSOAR master node as an administrator and click the **Settings** icon to open the `System` page.
2. In the `Multi Tenancy` section, click **Remote Tenant Manager** in the left menu.
3. Click the **Manage Playbook Mappings** tab.

4. To add a new alias name, click **Add Alias** and click **Create**.
 Alias names are used in the **Reference Remote Playbook** step.

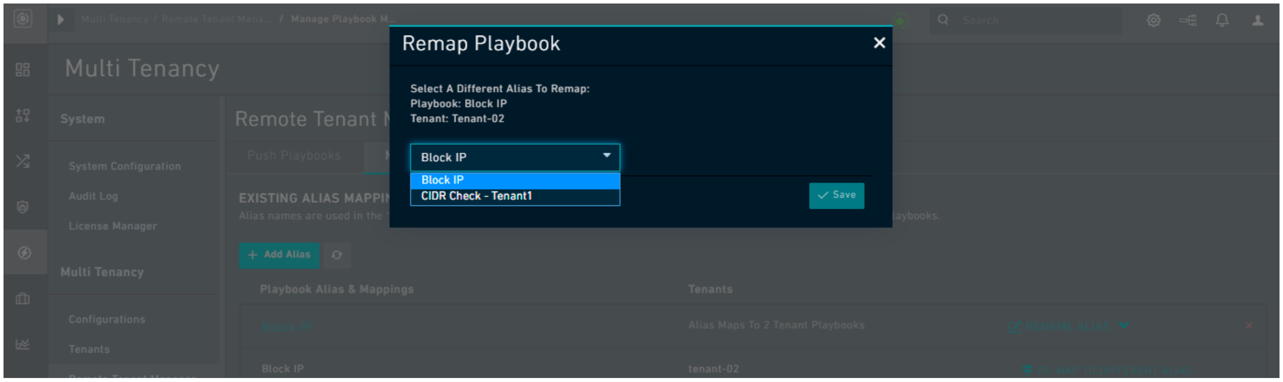


EXISTING ALIAS MAPPINGS

Alias names are used in the 'Remote reference playbook' step in the playbook designer to allow easier referencing of tenant playbooks.

Playbook Alias & Mappings	Tenants
Block IP	Alias Maps To 2 Tenant Playbooks
Block IP Block IP	tenant-02 tenant-01
CIDR Check - Tenant1	Alias Maps To 2 Tenant Playbooks

5. If you want to remap an existing alias to a different alias, expand **Rename Alias**, and then click the **Re-map To Different Alias** link. This displays a `Remap Playbook` dialog, using which you can select (map) the appropriate playbook:



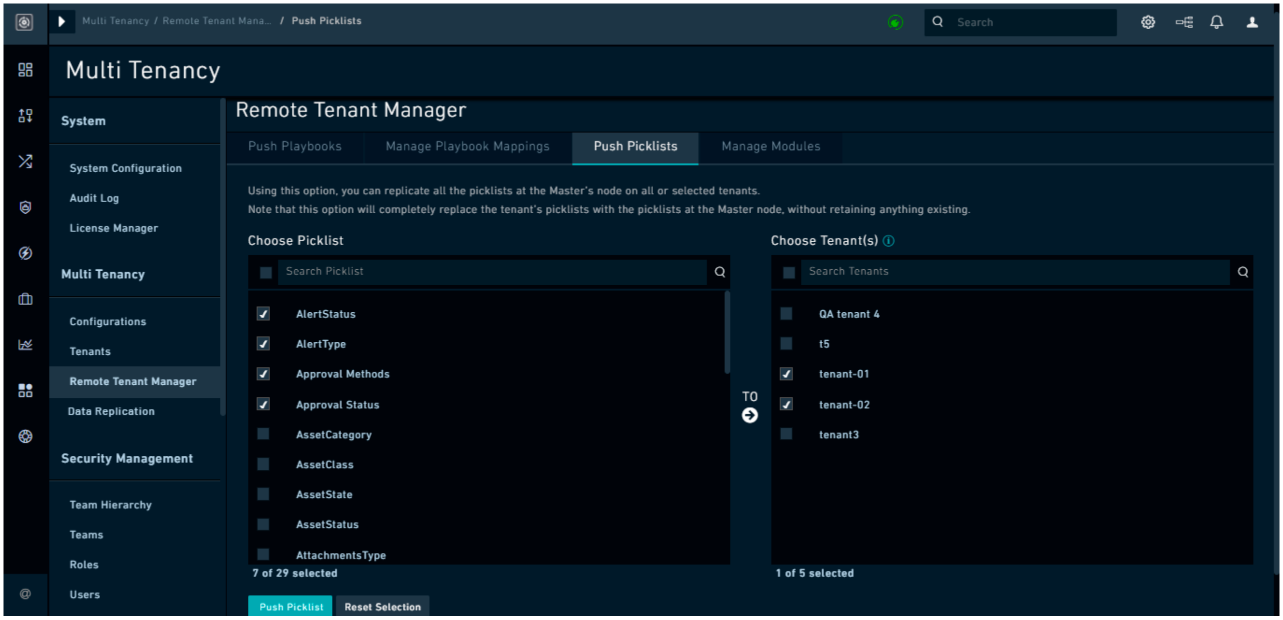
Select A Different Alias To Remap:
Playbook: Block IP
Tenant: Tenant-02

Pushing picklists from the master node to the tenant node(s)

You can push picklists from the master node to the selected tenant node(s). This operation will replace the picklists that are present in the selected tenant node(s) with the picklists from the master node.

1. Log on to your FortiSOAR master node as an administrator and click the **Settings** icon to open the `System` page.
2. In the `Multi Tenancy` section, click **Remote Tenant Manager** in the left menu.
3. Click the **Push Picklists** tab.
4. From the **Choose Picklist** list select the picklists that you want to push from the master node to the tenant node(s).
5. From the **Choose Tenant(s)** list select the tenant(s) to which you want to push the picklists and click the **TO** arrow.
 The **Choose Tenant(s)** list lists only those tenants that are active and whose connections are verified, and those who granted permissions to the master to modify their mmd.

You can also search for picklists and tenants using the search functionality.



Using this option, you can replicate all the picklists at the Master's node on all or selected tenants.
Note that this option will completely replace the tenant's picklists with the picklists at the Master node, without retaining anything existing.

Choose Picklist

□ Search Picklist

- ☑ AlertStatus
- ☑ AlertType
- ☑ Approval Methods
- ☑ Approval Status
- □ AssetCategory
- □ AssetClass
- □ AssetState
- □ AssetStatus
- □ AttachmentsType

7 of 29 selected

Choose Tenant(s)

□ Search Tenants

- □ QA tenant 4
- □ t5
- ☑ tenant-01
- ☑ tenant-02
- □ tenant3

1 of 5 selected

6. Click **Push Picklists**.
   Important: Clicking **Push Picklists** completely replaces the picklists at the tenants' picklists (none of the existing picklists will be retained at the tenants' node) with the master's picklists.

Editing the tenant's model metadata at the master and then pushing the model metadata to remote tenants

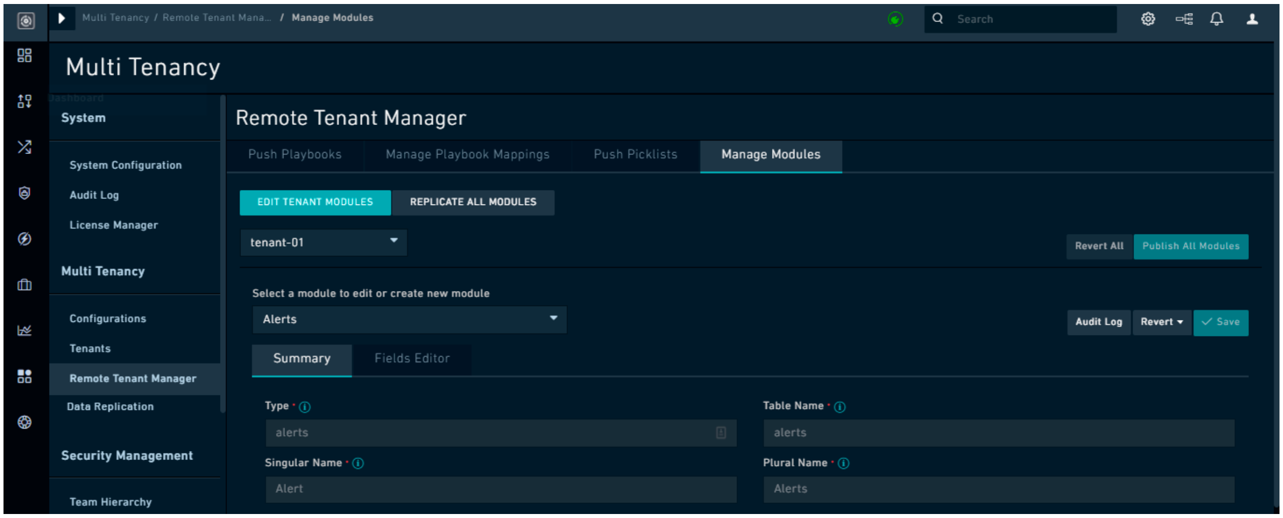
The master node can make changes to the tenants' model metadata (MMD) and push those module changes, and can also push picklists, to the tenant node. Similarly, if tenants make any changes to their mmd, those changes will also reflect on the master.

To allow the master node to remotely edit the tenant's modules and push picklists, the tenant requires to allow the master to control its mmd. This is managed by the **Allow Module Management** button on the on the `Master Configuration` page of the tenant node. To allow the master node to control the mmd of the tenant's node, ensure that the **Allow Module Management** button is set to **YES**.

On the master node do the following to update tenants' mmd:

1. Log on to your FortiSOAR master node as an administrator and click the **Settings** icon to open the `System` page.
2. In the `Multi Tenancy` section, click **Remote Tenant Manager** in the left menu.
3. Click the **Manage Modules** tab.
   By default, the **Edit Tenant Modules** button will be highlighted.
4. At the top-right of the `Edit Tenant Modules` page, from the **Select a Tenant** drop-down list, choose the tenant's whose mmd you want to modify.
5. From the **Select a module to edit or create a new module** drop-down list, either create a new module or choose the module that you want to modify for the selected tenant.
   Note: The picklists that appear in the Fields Editor tab for the selected tenant are the picklists of the master node. Therefore, you must ensure that the picklists between the master and tenant nodes are synchronized for successful modification of a tenant MMD.
   For example, if you are adding a new module, then you must ensure that you have added any picklist that is associated with the new module at the master node, then using the **Push Picklist** tab you should push the picklist to the tenant nodes that require that picklist. This ensures that the picklists are synchronized between the master and

the tenant nodes.



tenant-01

Select a module to edit or create new module: Alerts

Type: alerts

Table Name: alerts

Singular Name: Alert

Plural Name: Alerts

6. Add or modify fields or attributes that you want to change in the tenant's MMD and click **Save**.
   This puts the changes in the *Staging* state for both the master and the tenant.
7. To update the database and make the changes permanent in both the master and tenant environments, click **Publish All Modules**.
   Important: Once you click Publish, the publishing operation begins on the tenant environment. At this time FortiSOAR displays a message stating the same and until the publishing operation is completed, users are unable to work in FortiSOAR. Therefore, it is recommended that you should send a prior notification to all users of a publish.

Tenants might be required to edit their system view templates (SVTs) to view the fields that are added to records based on the changes made in the MMD. An example of how to modify an SVT is given in the Setting up a customer who has multiple sites topic, in *Step 2: Extend the Tenants Module and then edit the required SVTs* section. Step 2 demonstrates adding a custom field.

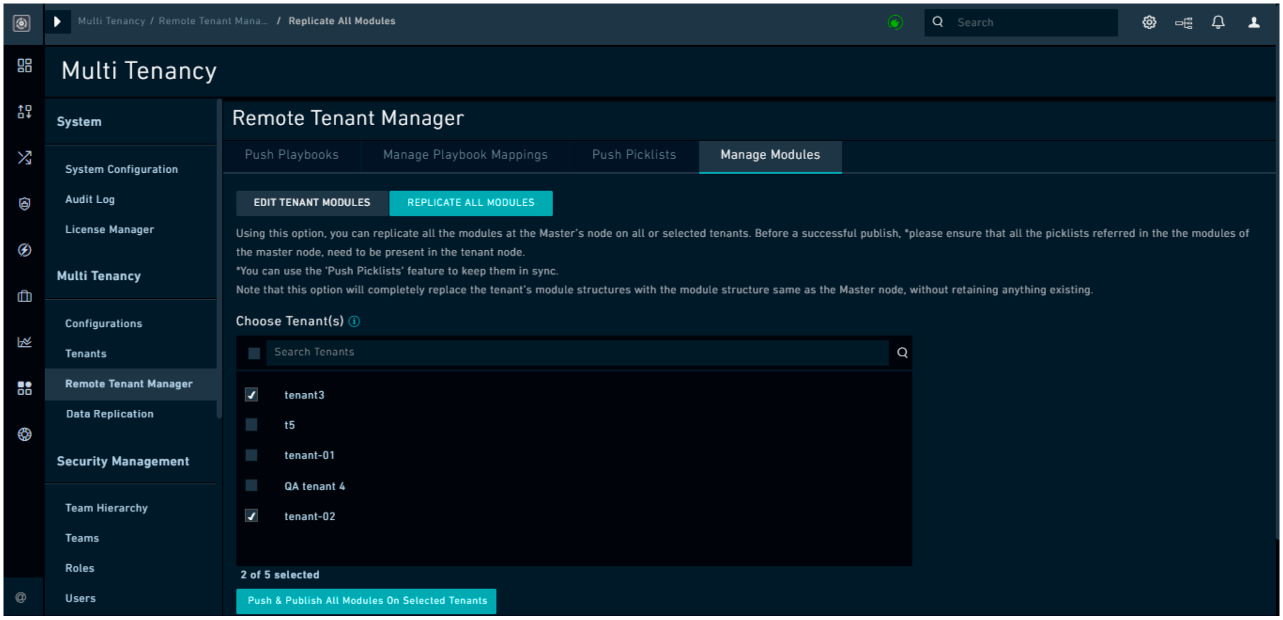
You can also push all modules from the master node, i.e., replicates the module structure of the master on the selected tenant node(s). The ability to replicate the module structure makes onboarding new tenants from the master node effective.

This operation will completely replace the tenant's module structure with the module structure (including all the picklists) of the master node.

In case you want to replace the complete tenant MMD with the master's MMD, do the following on the `Manage Tenant Modules` page:

1. Click the **Replicate All Modules** button.
2. From the **Choose Tenant(s)** list, select the tenant(s) whose MMD you want to replace with the master's MMD.
   The **Choose Tenant(s)** list lists only those tenants that are active and whose connections are verified, and those who granted permissions to the master to modify their mmd.

You can also search for tenants using the search functionality.



Using this option, you can replicate all the modules at the Master's node on all or selected tenants. Before a successful publish, *please ensure that all the picklists referred in the the modules of the master node, need to be present in the tenant node.
*You can use the 'Push Picklists' feature to keep them in sync.
Note that this option will completely replace the tenant's module structures with the module structure same as the Master node, without retaining anything existing.

Choose Tenant(s)

☐ Search Tenants

☑ tenant3
☐ t5
☐ tenant-01
☐ QA tenant 4
☑ tenant-02

2 of 5 selected

3. Click **Push & Publish All Modules on Selected Tenants**.
   Warning: Clicking **Push & Publish All Modules on Selected Tenants** completely replaces the selected tenants' module structures (none of the existing module structures will be retained at the tenants' node) with the master's module structures.

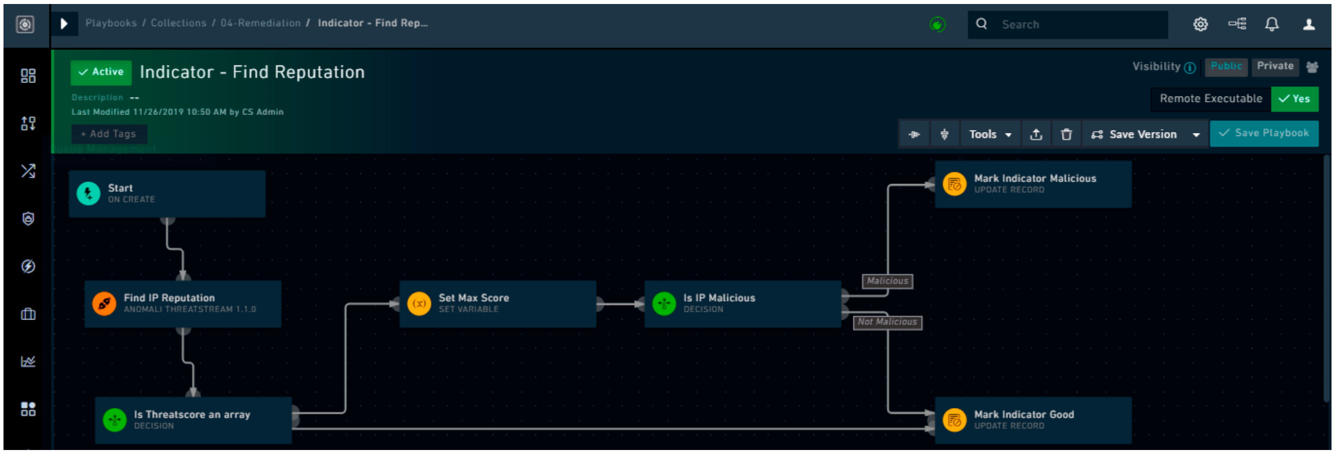
Executing remote playbooks at the tenant node from the master node

To allow remote execution of a playbook, i.e., to allow a FortiSOAR master node to execute a playbook on FortiSOAR tenant nodes, do the following on the master node:

Set the Remote Executable flag:

Enable the **Remote Executable flag** on the playbooks that you want to enable for remote execution from the master node.

In the Playbook Designer, open the playbook that you want to execute remotely, and click to enable the **Remote Executable** box.

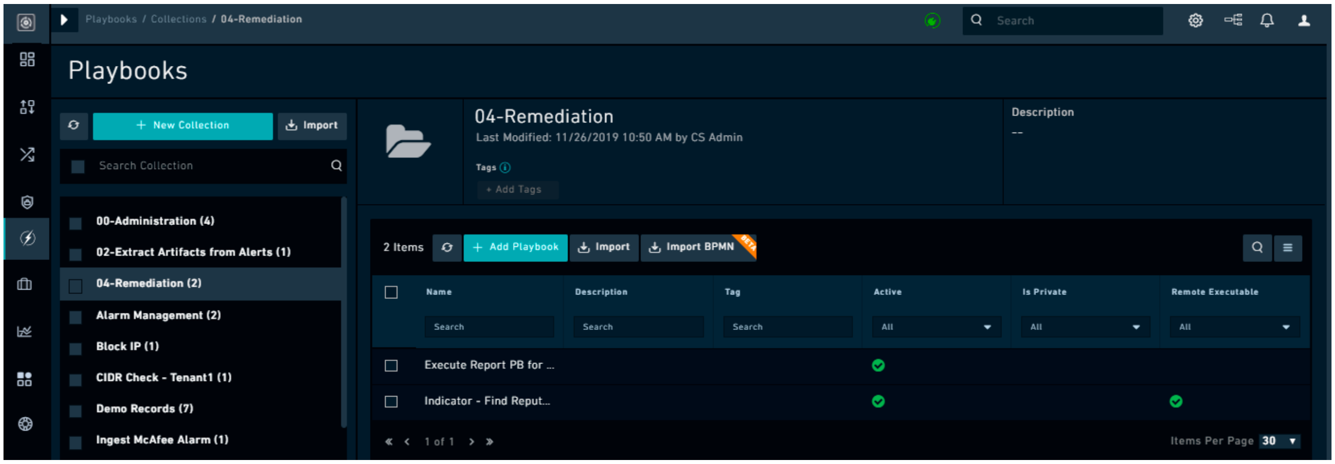


Description --
Last Modified 11/26/2019 10:50 AM by CS Admin

Remote Executable ✓ Yes

Alternatively, you can also enable the **Remote Executable** box in the playbook on the tenant node.

A **Remote Executable** column is also added to the `Grid` view of the `Playbooks Collection` page. This column displays whether the playbook is enabled for remote execution from the master node. If the playbook can be remotely executed, then the **Remote Executable** column displays a tick in a green circle. If the playbook cannot be remotely executed, then the **Remote Executable** column is blank.



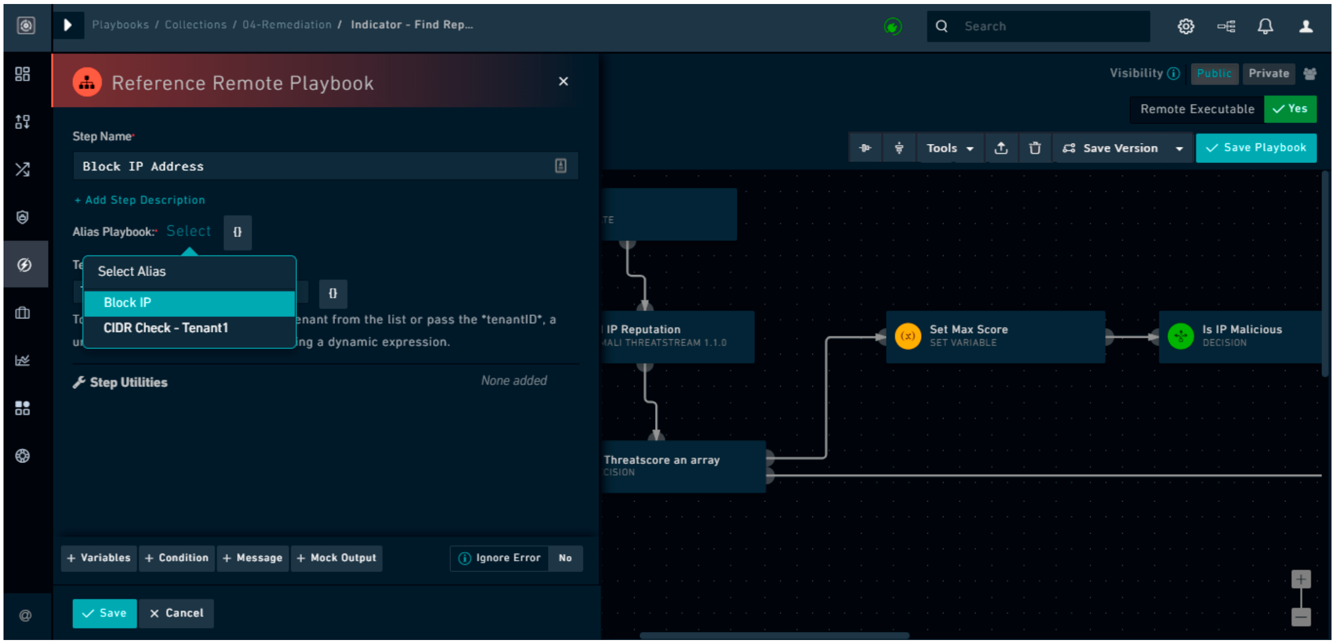
- 00-Administration (4)
- 02-Extract Artifacts from Alerts (1)
- 04-Remediation (2)
- Alarm Management (2)
- Block IP (1)
- CIDR Check - Tenant1 (1)
- Demo Records (7)
- Ingest McAfee Alarm (1)

Name	Description	Tag	Active	Is Private	Remote Executable
Execute Report PB for ...			✓		
Indicator - Find Reput...			✓		✓

Add a Reference Remote Playbook step

Add a **Reference Remote Playbook** step in the playbook that you want to trigger from the master node and remotely execute and retrieve details from the tenant node.

In the Playbook Designer, open the playbook to which you want to add the Reference Remote Playbook step, so that the master node can remotely execute/trigger the playbook on the tenant nodes.
To add a remote reference to a playbook, click the **Reference Remote Playbook** step and in the **Step Name** field, type the name of the step, then in the `Step Description` field, optionally add a description for this step. Click **Select** in the **Alias Playbook** field, which displays the `Select Alias` list. The `Select Alias` list displays a list of all the available playbooks for which you have defined alias names. From the `Select Alias` list, select the appropriate alias name for this playbook. This playbook will be mapped to the alias name, and this mapping will be displayed in the `Playbook Mapping` page in the master node.



Step Name: Block IP Address

Alias Playbook: Select

- Block IP
- CIDR Check - Tenant1

...enant from the list or pass the "tenantID", a ...ng a dynamic expression.

Remote Executable: ✓ Yes

In the `Alias Playbook` field, you can click the **Add Custom Expression** (⧉) button to specify jinja variable that contains the alias value of the remote reference playbook.

The **Tenant** drop-down list displays the tenants on which you can execute this playbook. By default, the selected tenant is the one who was assigned to the record that triggered the playbook, i.e., the **Tenant Assigned To Input Record** option. You can use the **Tenant** drop-down list to select a different tenant on which to run the playbook. You can also add a `tenantId`, which is a unique property of a tenant node, using a dynamic expression.

The **Loop** variable has been removed from the Remote Reference Playbook step. If you want to loop through records, see Use case of how to loop through records.

Ensure that you push the appropriate playbooks to the tenant node. See the Pushing playbooks from the master node for remote execution section.

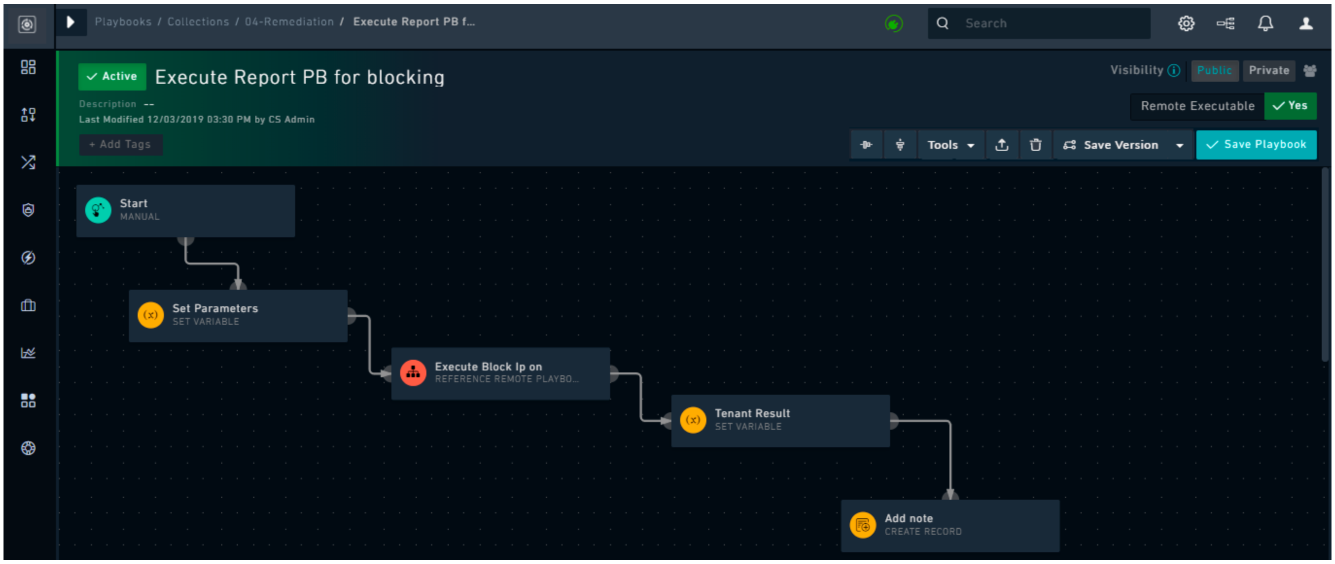
Use case of executing remote playbooks

Consider a scenario where there is a master node and two tenant nodes, Tenant 1 has VirusTotal as its threat intelligence tool, and Tenant 2 has IBM xForce as its threat intelligence tool.

This use case demonstrates how a remote playbook is executed from the master node to a tenant node, and the result of the playbook is passed back to the master node. In this case, the master node executes a playbook, using the Tenant 1 record, that has a reference remote playbook step, with an alias mapped to Tenant 1, which in turn executes the alias playbook on tenant 1, and returns the result of the playbook to the master node.

On the master node create a playbook as follows:

1. **Trigger step**: Manual Trigger. This playbook will be triggered when you select records and execute this playbook. Also, ensure that you select the Run Separately For Each Selected Record option in this step
2. **Set Variable step**: Set the variables that should be passed from the master node to the tenant node. You should add this step just before the "Reference Remote Playbook" step.
   For example, to extract the IP address from the record, add `ipAddr` as a variable, whose value will be `{{vars.input.params.ipAddress}}`.
3. **Reference Remote Playbook step**: In this step ensure that you have mapped the Alias playbook correctly and select the appropriate tenant on which you want to execute the playbook. By default, the selected tenant is the one who was assigned to the record that triggered the playbook, i.e., the **Tenant Assigned To Input Record** option. You can use the **Tenant** drop-down list to select a different tenant on which to run the playbook. You can also add a `tenantId`, which is a unique property of a tenant node, using a dynamic expression.
4. **Set Variable step**: Set a variable to consume the result from the tenant node. For example, add `execution_result` as a variable, whose value will be `{{vars.result}}`.
5. Further based on the execution result, you can create additional steps, such as adding a note or comment to the input record.



Playbooks / Collections / 04-Remediation / Execute Report PB f...

✓ Active Execute Report PB for blocking

Description --

Last Modified 12/03/2019 03:30 PM by CS Admin

Visibility: Public Private

Remote Executable ✓ Yes

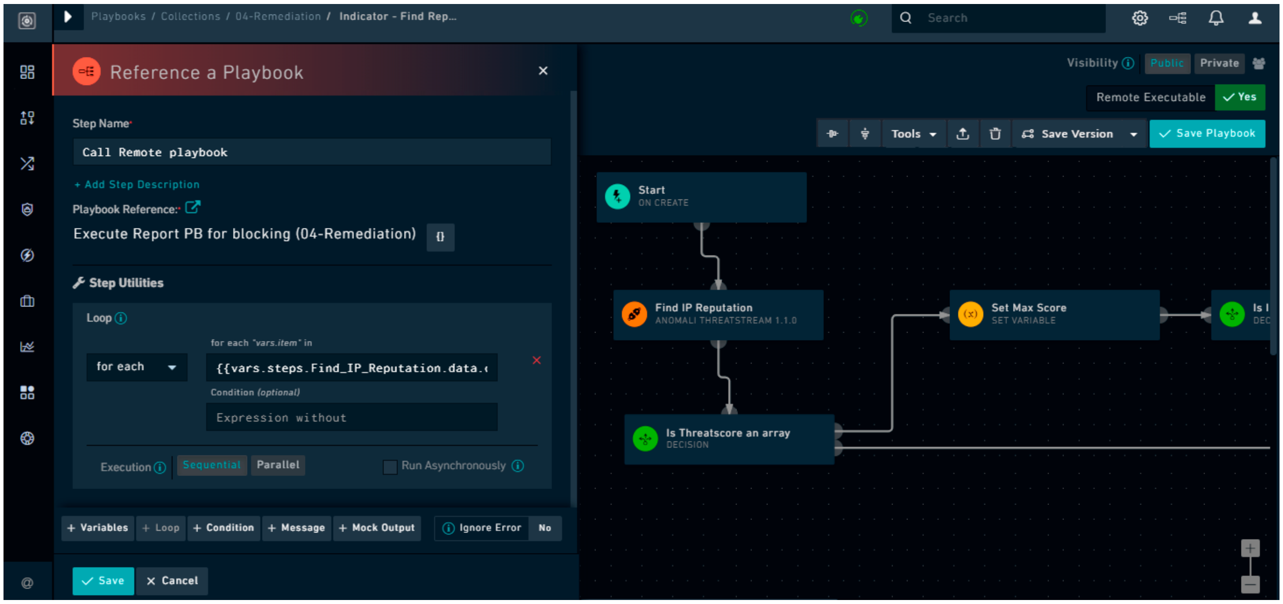
On the Tenant node, you can either create a playbook using the following steps or push the created (using the following steps) playbook from the master node to the tenant node. You must ensure that the **Remote Executable flag** is enabled for this playbook.

1. **Trigger step**: Referenced, since the master node will trigger this playbook.
2. **Set Variable step**: Set this variable to consume the input sent from the master node. For example to consume the IP address sent from the master node, add `ipAddr` as a variable, whose value will be `{{vars.ipAddr}}`.
3. **Connector step**: Add the VirusTotal connector step (since Tenant 1 has the VirusTotal threat intelligence tool) to analyze the IP address received from the previous Set Variable step and send its result to the next step.
4. **Set Variable step**: Set a variable so that the master node can consume the result from the tenant node. For example, add `execution_result` as a variable, whose value will be `{{vars.result}}`.

Use case of how to loop through records

Consider a scenario where on the master node there are 5 records, belonging to different tenants, on which you want to run an *Investigative* playbook. To achieve looping through records, do the following:

1. In the *Investigate* playbook add a "Reference A Playbook" step that contains the looping logic and, which calls another playbook that contains the Reference Remote Playbook step, as shown in the following image:


Playbooks / Collections / 04-Remediation / Indicator - Find Rep...

Reference a Playbook

Step Name: Call Remote playbook

Playbook Reference: Execute Report PB for blocking (04-Remediation)

Loop: for each

for each "vars.item" in: {{vars.steps.Find_IP_Reputation.data.(

Condition (optional): Expression without

Execution: Sequential

Visibility: Public

Remote Executable: Yes

2. In the called playbook, i.e., the playbook with the "Reference Remote Playbook" step, you must specify the `tenantId` in the **Tenant** field.

Extending the Tenants module

The `Tenants` module (included by default in FortiSOAR) is now extendable, i.e., you can add fields that you require to the tenant's module. The steps you can follow to extend the `Tenants` module are mentioned in *Step 2: Extend the Tenants Module and then edit the required SVTs* section that is part of the Setting up a customer who has multiple sites section.

Additional Configurations

Changing the hostname

The FortiSOAR Configuration Wizard is available only on the first `ssh` login. If at a later stage, you require to change the hostname of your FortiSOAR secure message exchange, then you can use the FortiSOAR Admin CLI (`csadm`). For more information on `csadm`, see the *FortiSOAR Admin CLI* chapter in the "Administration Guide."

You must run `csadm` immediately after deployment of the FortiSOAR Virtual Appliance. Running this after the FortiSOAR system is active and online might lead to data loss.

To change the hostname, ensure that the hostname is resolvable and then do the following:

1. SSH to your FortiSOAR VM and login as a *root* user.
2. To change your hostname of your FortiSOAR secure message exchange, type the following command:
```
# csadm hostname --set [<hostname>]
```
 This command changes your current hostname to the new hostname that you have specified, sets up the message broker, regenerates certificates, and restarts FortiSOAR services.

When you run the `# csadm hostname --set[<hostname>]` command on the secure message exchange, then all previously configured tenant user accounts and exchanges will be deleted. Also, if there are any pending messages, then they will be lost. After you have completed changing the hostname, you will have to reconfigure the tenants from the master node.

Important: When you change the hostname for your ***Secure Message Exchange VM***, you must ensure that the certificate the secure message exchange uses for communication with the FortiSOAR nodes has the same Common Name (aka CN) or is a wildcard certificate for the domain.

If you get the following error, when you are changing the hostname on your secure message exchange:

```
unable to connect to epmd (port 4369) on rabbitmqserver: nxdomain (non-existing domain)
```

Then you must terminate the current session, open a new session, and then run the `# rabbitmqctl status` command.

Monitoring and administering your multi-tenancy environment

Setting up High Availability in case of a multi-tenancy environment

If you are planning to configure high availability for your master or tenant nodes, you must first configure high availability then add the master or tenant configurations. For information about High Availability in FortiSOAR, see the *High Availability support in FortiSOAR* chapter in "Administration Guide."

Setting up High Availability of the Secure Message Exchange

In case of a failure of the secure message exchange, an alternate secure message exchange can be set up, and the secure message exchange details at the master and tenant nodes can be updated to point to the IP address, Certificate, etc., of the new secure message exchange. The master node would handle reconfiguring the queues and exchanges at the new secure message exchange seamlessly. This, however, would mean downtime of the secure message exchange till the new secure message exchange is provisioned. Also, any messages that were yet residing on the secure message exchange and which were not delivered yet to the tenant would require to be re-sent. If a downtime of the secure message exchange is not desired, the secure message exchange can be set up for an Active-Passive or an Active-Active High Availability configuration.

The secure message exchange internally uses RabbitMQ for the persistence and routing of messages across the FortiSOAR nodes. Therefore, you can achieve High Availability for the secure message exchange by using any of the standard RabbitMQ high availability methods. The secure message exchange can be made to work in an active/passive setup by using a reliable shared storage such as NAS or SAN for the RabbitMQ data directory, `/var/lib/rabbitmq`, between the two nodes and using monitoring and management tools such as Pacemaker to perform the failover as explained in the High availability with Pacemaker and DRBD article.

RabbitMQ also supports clustering, that in conjunction with Queue Mirroring can be used for an Active-Active configuration as explained in the Clustering Guide and in the Highly Available (Mirrored) Queues article, which includes steps on how to set up the clusters and monitor queues. The clustered instances should be fronted by a TCP Load Balancer such as HAProxy, and clients should connect to the cluster using the address of the proxy. For information about High Availability in FortiSOAR, see the *High Availability support in FortiSOAR* chapter in "Administration Guide."

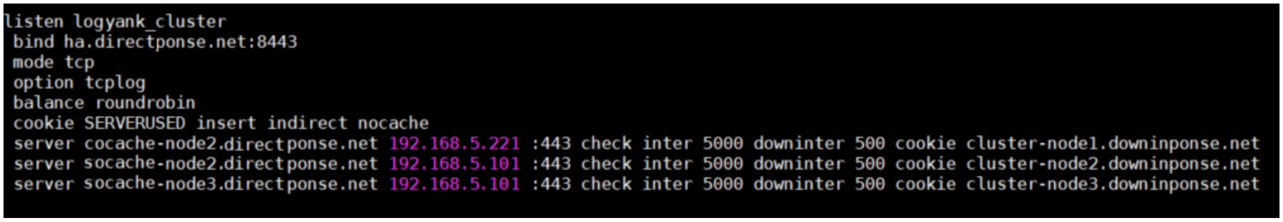
Using the RabbitMQ documentation mentioned above, the following sequence of steps and commands can help you set up clustering of two secure message exchanges with queue mirroring:

1. Deploy and configure the first secure message exchange. See the Deploy and configure the secure message exchange section.
2. Deploy and configure the second secure message exchange. See the Deploy and configure the secure message exchange section.
3. Ensure that the two nodes are resolvable from each other using DNS entries or using the `/etc/hosts` entry.
4. Copy the `erlang` cookie from the following file on the first secure message exchange:
   `/var/lib/rabbitmq/.erlang.cookie` onto the same file on the second secure message exchange.
5. Run `# firewall-cmd --zone=public --add-port=4369/tcp --permanent` on both the secure message exchanges and ensure that these ports are open for each other from any security groups, etc.
6. Run `# firewall-cmd --zone=public --add-port=25672/tcp --permanent` on both the secure message exchanges.
7. Run `# firewall-cmd --reload` on both the secure message exchanges.
   Read https://www.rabbitmq.com/clustering.html#ports for using alternate ports for steps 5 and 6.
8. On the second secure message exchange, run the following commands:
   a. `# rabbitmqctl stop_app`
   b. `# rabbitmqctl join_cluster rabbit@<nodenamesecuremessageexchange1>`
   c. `# rabbitmqctl start_app`
9. `rabbitmqctl cluster_status` on both the nodes should show two nodes in the cluster. For example
   ```
   Cluster status of node rabbit@awsrouter2...
   [{nodes,[{disc,['rabbit@awsrouter2',rabbit@awsrouter1]}]},
   {running_nodes,[rabbit@awsrouter1,'rabbit@awsrouter2']},
   {cluster_name,<<"rabbit@awsrouter1">>},
   {partitions,[]},
   {alarms,[{rabbit@awsrouter1,[]},{'rabbit@awsrouter2',[]}]}]
   ```
10. Enabling Queue Mirroring by setting the following policy on secure message exchange 1:
   ```
   # rabbitmqctl set_policy tenants-qa-ha "^queue.postman\."
   '{"ha-mode":"exactly","ha-params":2,"ha-sync-mode":"automatic"}'
   ```

Setting up HAProxy as a TCP load balancer fronting the two clustered nodes

The following steps list out the steps to install "HAProxy" as a load balancer on a CentOS Virtual Machine:

1. `# yum install haproxy`
2. In the `/etc/haproxy/haproxy.cfg` file and add the policy as shown in the following image:



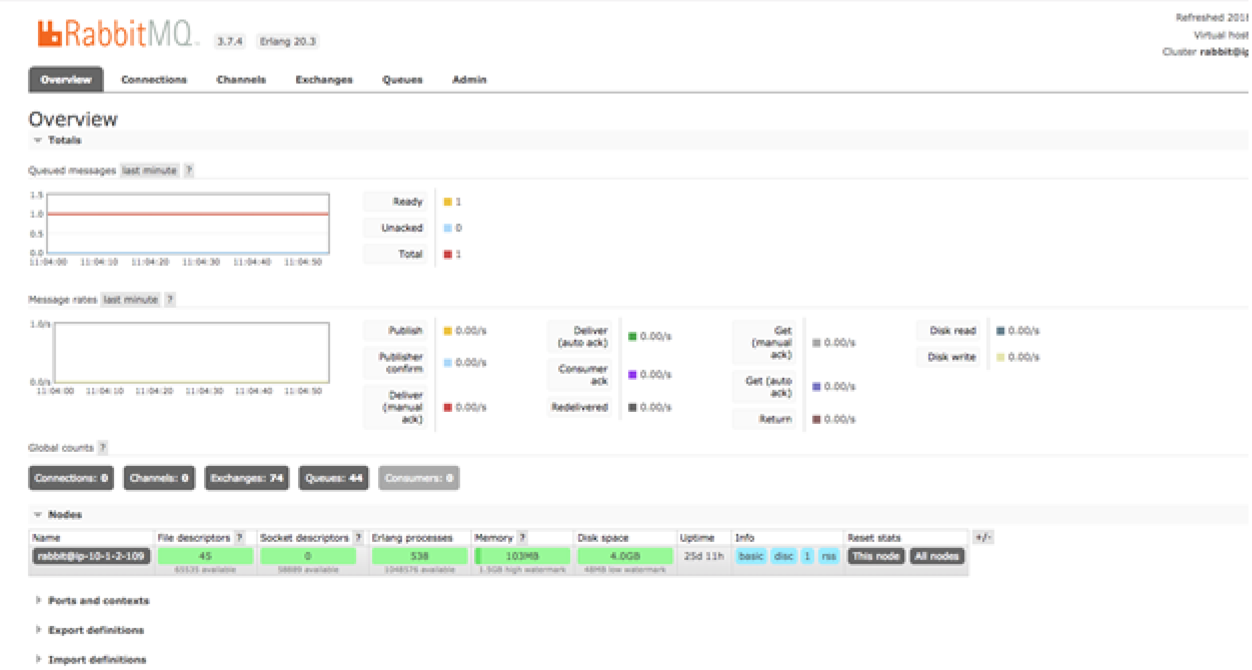
```
listen logyank_cluster
 bind ha.directponse.net:8443
 mode tcp
 option tcplog
 balance roundrobin
 cookie SERVERUSED insert indirect nocache
 server cocache-node2.directponse.net 192.168.5.221 :443 check inter 5000 downinter 500 cookie cluster-node1.downinponse.net
 server socache-node2.directponse.net 192.168.5.101 :443 check inter 5000 downinter 500 cookie cluster-node2.downinponse.net
 server socache-node3.directponse.net 192.168.5.101 :443 check inter 5000 downinter 500 cookie cluster-node3.downinponse.net
```

3. Restart `haproxy` using the following command:
   `# systemctl restart haproxy`
4. Use the bind address (instead of the IP address of the node in the cluster) for accessing the FortiSOAR UI.
   You must also ensure that the CA certificates on both the nodes are the certificates that were provided for secure message exchange configuration.

Monitoring the connectivity of the different nodes at the secure message exchange

Once you deploy and configure a secure message exchange, its remote management UI is enabled. The connectivity of the different nodes, messages pending on the data, action and instructions queues, and the message flow rate can be monitored using the remote management UI.

Open `https://<secure message exchange_address>:<management_port>` and connect using your administration credentials. The management port and administration credentials are configurable and specified at the time of configuring the secure message exchange. See the Configuring the Secure Message Exchange section.



Overview

▾ Totals

Queued messages last minute ?

Ready	🟨 1
Unacked	🟦 0
Total	🟥 1

Message rates last minute ?

Publish	🟨 0.00/s	Deliver (auto ack)	🟩 0.00/s	Get (manual ack)	■ 0.00/s	Disk read	■ 0.00/s
Publisher confirm	🟦 0.00/s	Consumer ack	🟪 0.00/s	Get (auto ack)	🟪 0.00/s	Disk write	🟨 0.00/s
Deliver (manual ack)	🟥 0.00/s	Redelivered	■ 0.00/s	Return	■ 0.00/s		

Global counts ?

Connections: 0 Channels: 0 Exchanges: 74 Queues: 44 Consumers: 0

▾ Nodes

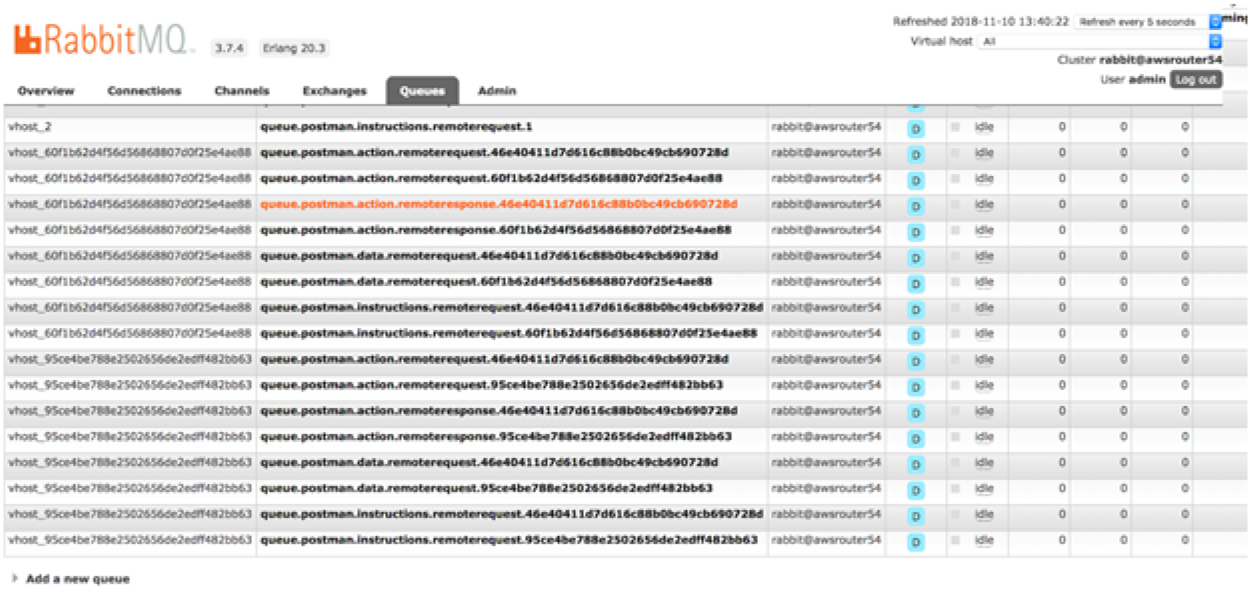
Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-10-1-2-109	45 65535 available	0 58889 available	538 1048576 available	103MB 1.5GB high watermark	4.0GB 48MB low watermark	25d 11h	basic disc 1 rss	This node All nodes	

▸ Ports and contexts

▸ Export definitions

▸ Import definitions

All vhosts, exchanges, and queues associated with the node are created with the respective tenant ID.



vhost_2	queue.postman.instructions.remoterequest.1	rabbit@awsrouter54	D	idle	0	0	0
vhost_60f1b62d4f56d56868807d0f25e4ae88	queue.postman.action.remoterequest.46e40411d7d616c88b0bc49cb690728d	rabbit@awsrouter54	D	idle	0	0	0
vhost_60f1b62d4f56d56868807d0f25e4ae88	queue.postman.action.remoterequest.60f1b62d4f56d56868807d0f25e4ae88	rabbit@awsrouter54	D	idle	0	0	0
vhost_60f1b62d4f56d56868807d0f25e4ae88	queue.postman.action.remoteresponse.46e40411d7d616c88b0bc49cb690728d	rabbit@awsrouter54	D	idle	0	0	0
vhost_60f1b62d4f56d56868807d0f25e4ae88	queue.postman.action.remoteresponse.60f1b62d4f56d56868807d0f25e4ae88	rabbit@awsrouter54	D	idle	0	0	0
vhost_60f1b62d4f56d56868807d0f25e4ae88	queue.postman.data.remoterequest.46e40411d7d616c88b0bc49cb690728d	rabbit@awsrouter54	D	idle	0	0	0
vhost_60f1b62d4f56d56868807d0f25e4ae88	queue.postman.data.remoterequest.60f1b62d4f56d56868807d0f25e4ae88	rabbit@awsrouter54	D	idle	0	0	0
vhost_60f1b62d4f56d56868807d0f25e4ae88	queue.postman.instructions.remoterequest.46e40411d7d616c88b0bc49cb690728d	rabbit@awsrouter54	D	idle	0	0	0
vhost_60f1b62d4f56d56868807d0f25e4ae88	queue.postman.instructions.remoterequest.60f1b62d4f56d56868807d0f25e4ae88	rabbit@awsrouter54	D	idle	0	0	0
vhost_95ce4be788e2502656de2edff482bb63	queue.postman.action.remoterequest.46e40411d7d616c88b0bc49cb690728d	rabbit@awsrouter54	D	idle	0	0	0
vhost_95ce4be788e2502656de2edff482bb63	queue.postman.action.remoterequest.95ce4be788e2502656de2edff482bb63	rabbit@awsrouter54	D	idle	0	0	0
vhost_95ce4be788e2502656de2edff482bb63	queue.postman.action.remoteresponse.46e40411d7d616c88b0bc49cb690728d	rabbit@awsrouter54	D	idle	0	0	0
vhost_95ce4be788e2502656de2edff482bb63	queue.postman.action.remoteresponse.95ce4be788e2502656de2edff482bb63	rabbit@awsrouter54	D	idle	0	0	0
vhost_95ce4be788e2502656de2edff482bb63	queue.postman.data.remoterequest.46e40411d7d616c88b0bc49cb690728d	rabbit@awsrouter54	D	idle	0	0	0
vhost_95ce4be788e2502656de2edff482bb63	queue.postman.data.remoterequest.95ce4be788e2502656de2edff482bb63	rabbit@awsrouter54	D	idle	0	0	0
vhost_95ce4be788e2502656de2edff482bb63	queue.postman.instructions.remoterequest.46e40411d7d616c88b0bc49cb690728d	rabbit@awsrouter54	D	idle	0	0	0
vhost_95ce4be788e2502656de2edff482bb63	queue.postman.instructions.remoterequest.95ce4be788e2502656de2edff482bb63	rabbit@awsrouter54	D	idle	0	0	0

▸ Add a new queue

Backing up and Restoring your FortiSOAR systems

Use the FortiSOAR Admin CLI (`csadm`) to regularly perform backups on master and tenant nodes and to restore the data seamlessly to a new FortiSOAR environment. To perform backup and restore, you must have *root access* on your FortiSOAR system.

For detailed information on backing up and restoring your FortiSOAR system, see the *Backing up and Restoring your FortiSOAR system* chapter in the "Administration Guide."

In case of a multi-tenant configuration, after you have completed restoring your master node or tenant node, you must restart the `postman` service on the new restored system using the following command:

```
# systemctl restart cyops-postman
```

Use Cases

Setting up a customer who has multiple sites

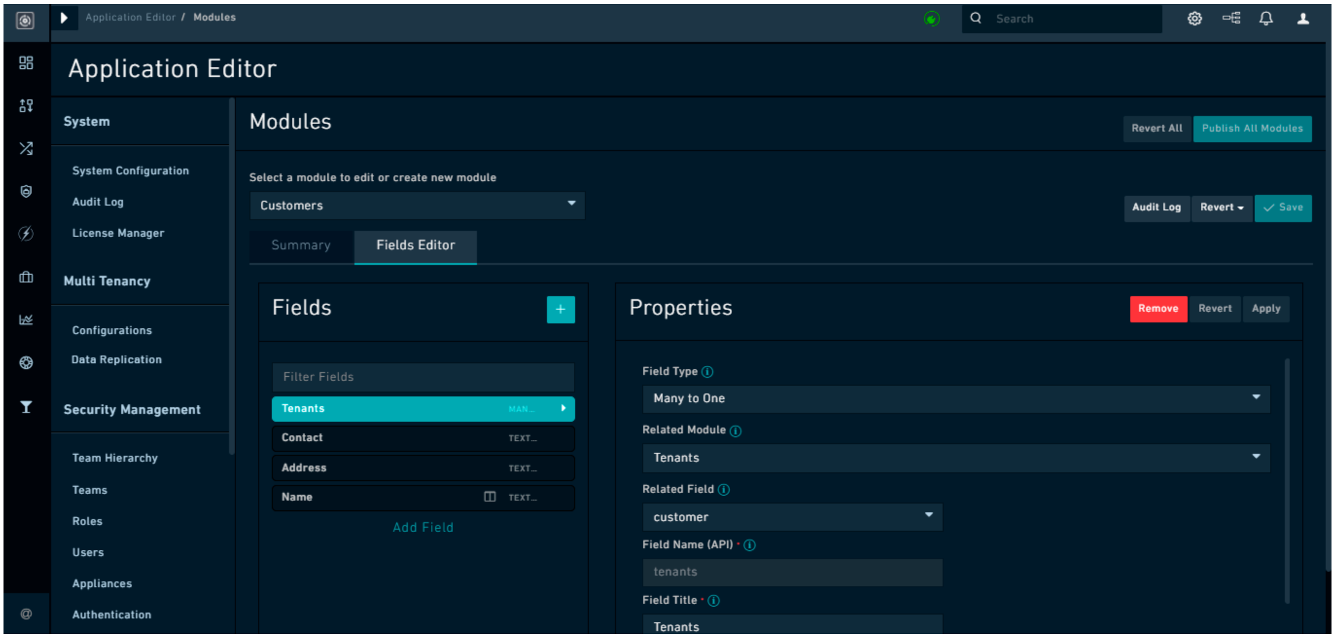
This use case describes the additional steps you need to perform, if the multi-tenant configuration has a customer with multiple sites (tenant). Each site (tenant) in this case has their own FortiSOAR instance.

All these steps must be performed on the Master node.

Step 1: Create a custom "Customers" Module

You must create a new "Customers" module in FortiSOAR by clicking the Setting icon on the top-right of the FortiSOAR UI. In the Application Editor section, click **Modules** and then click **+Create new module** to create a new module.

You can add whatever fields you require to this module, however, you must add a **Tenants** field that has a **Many to One** to `customer`.



Select a module to edit or create new module: Customers

Fields:
- Tenants MAN...
- Contact TEXT...
- Address TEXT...
- Name TEXT...

Field Type: Many to One

Related Module: Tenants

Related Field: customer

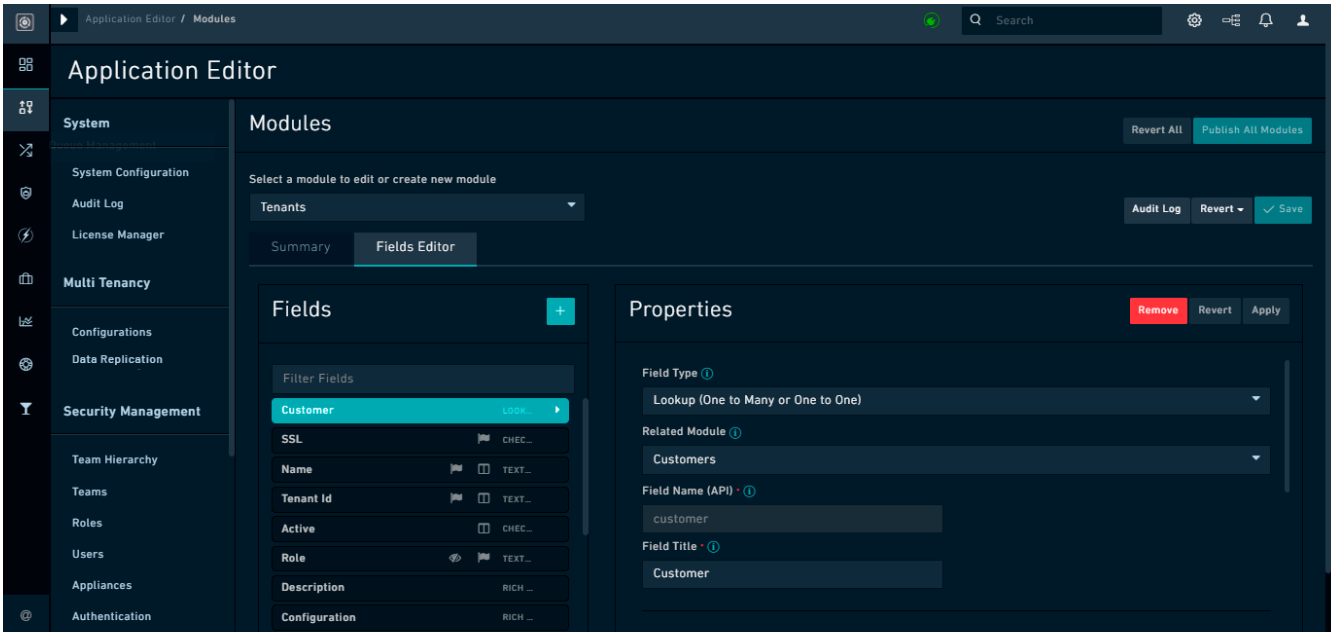
Field Name (API) *: tenants

Field Title *: Tenants

Once you create the Customers module, you must ensure that you assign the administrator's role appropriate permissions on this newly-created Customers module, so that you can add customers and associate tenants with customers.

Step 2: Extend the Tenants Module and then edit the required SVTs

Open the `Tenants` module (included by default in FortiSOAR) in the **Application Editor** and extend this module by adding the **Customer** field. Add the **Customer** field which has the field type set as **Lookup (One to Many or One to one).**



Select a module to edit or create new module: Tenants

Fields	
Customer	LOOK...
SSL	CHEC...
Name	TEXT...
Tenant Id	TEXT...
Active	CHEC...
Role	TEXT...
Description	RICH ...
Configuration	RICH ...

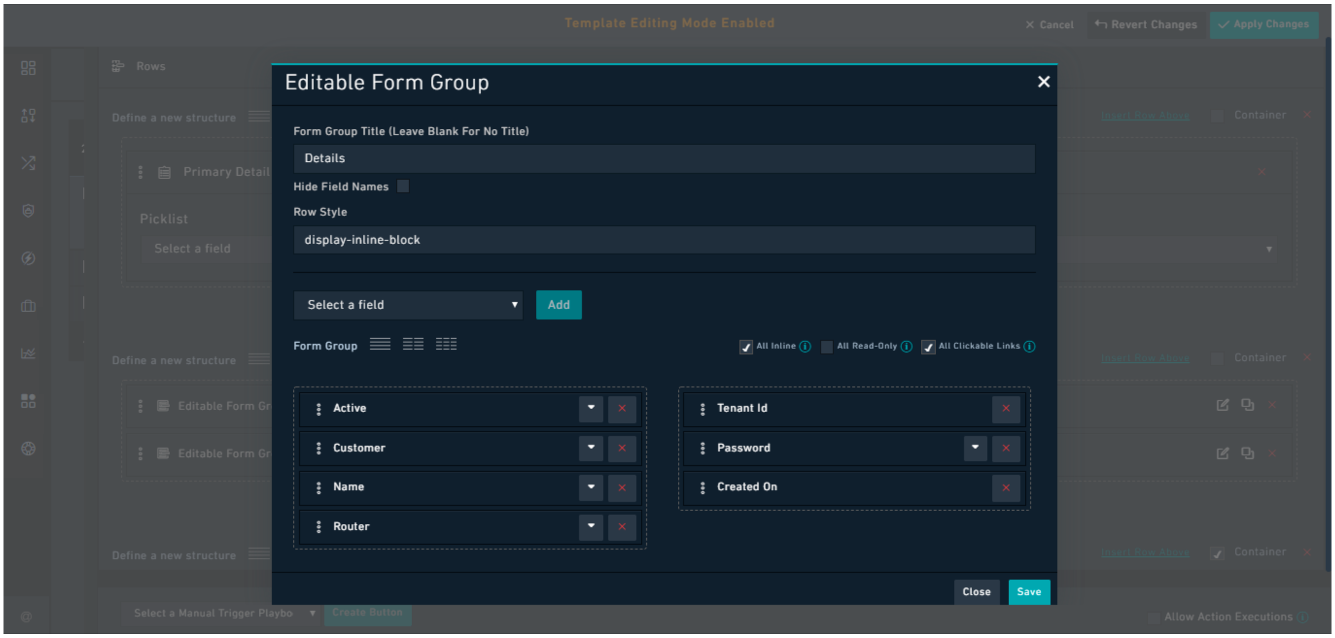
Field Type: Lookup (One to Many or One to One)

Related Module: Customers

Field Name (API) *: customer

Field Title *: Customer

Once you have created the `Customers` module and set up the relationship between the Customers and Tenants modules, you must now edit the system view template (SVT) for the `Tenants` module. Edit the SVT for the `Tenants` module to include the **Customer** field:



Form Group Title (Leave Blank For No Title): Details

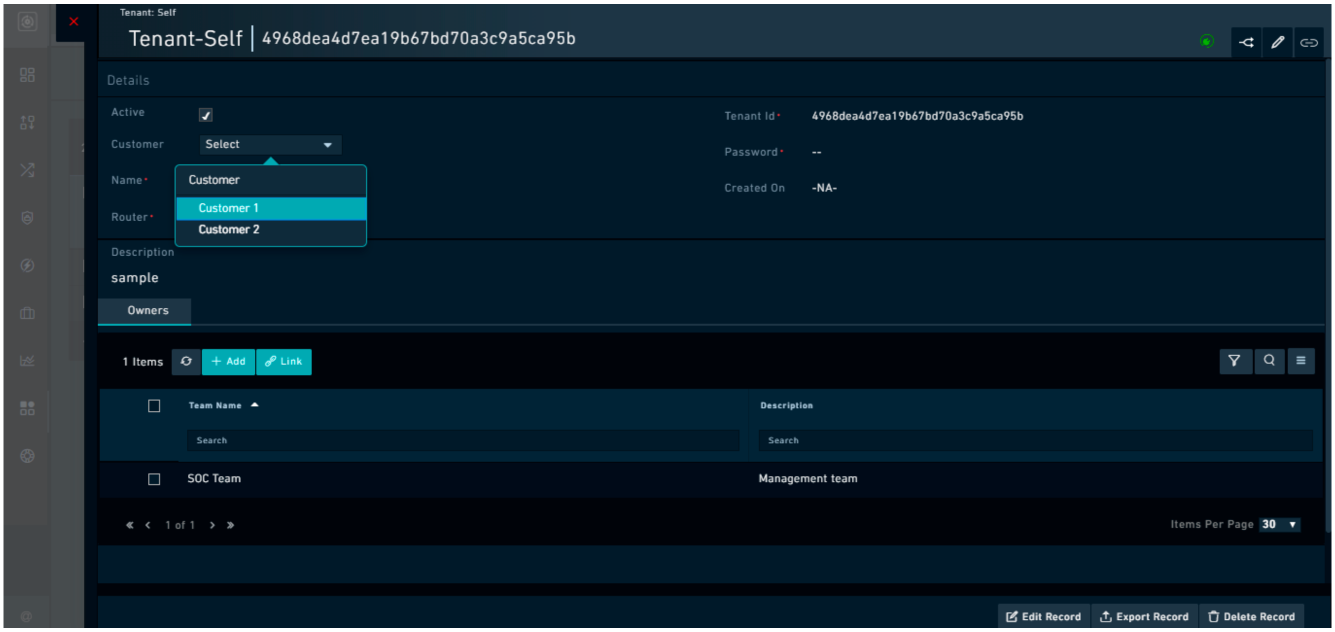
Hide Field Names ☐

Row Style: display-inline-block

Form Group ☑ All Inline ☐ All Read-Only ☑ All Clickable Links

Active	Tenant Id
Customer	Password
Name	Created On
Router	

The `Tenants` module will now appear as shown in the following image:



Tenant-Self | 4968dea4d7ea19b67bd70a3c9a5ca95b

Active ☑

Customer: Select

Customer
- Customer 1
- Customer 2

Name*

Router*

Tenant Id*: 4968dea4d7ea19b67bd70a3c9a5ca95b

Password*: --

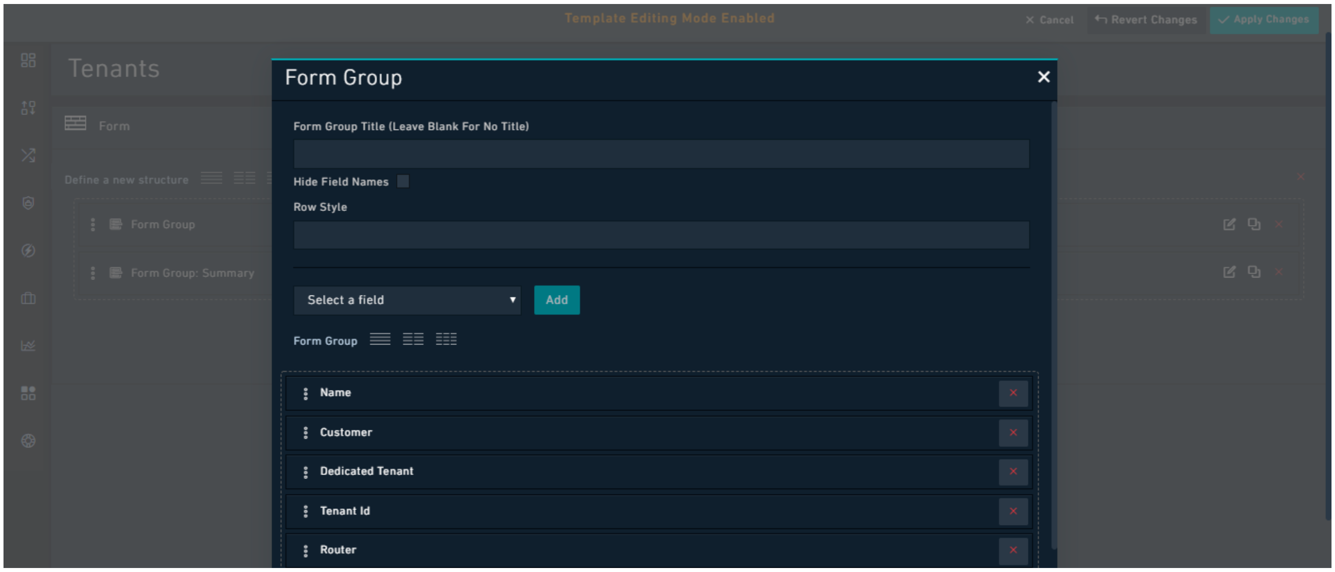
Created On: -NA-

Description: sample

Team Name	Description
SOC Team	Management team

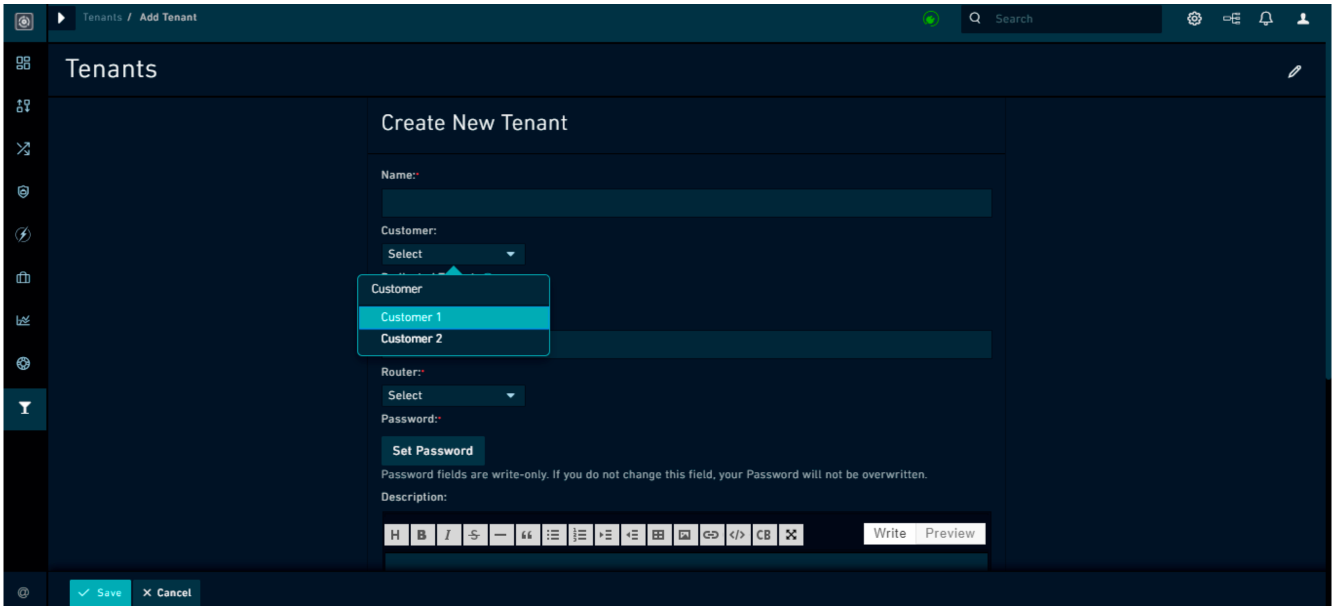
1 of 1

You should also edit the SVT of the *Add Tenant* form to include the **Customer** field, so that when you add a new tenant, you can select the **Customer** to whom the tenant belongs.



Step 3: Configuring the tenants on the Master Node

Now, you can configure the tenants on the master node (see Deploying tenant nodes). However, in this case you must ensure that you select the **Customer** to whom the tenant belongs to create the association between tenants and the customer.



Create New Tenant

Name:*

Customer: Select

Customer
Customer 1
Customer 2

Router:* Select

Password:*

Set Password

Password fields are write-only. If you do not change this field, your Password will not be overwritten.

Description:

Best practices

Following are the best practices that you should follow while configuring the distributed managed service provider model for multi-tenancy in your environment

- On the master node, you should edit your template (SVT) in modules that are being replicated from the tenant nodes so that you include the Tenant field on `Add Record` Dialog.
  To edit the SVT, navigate to the module for which you want to update the SVT, for example, `Alerts`, open a record and click **Edit Template**. On the `Template Editing Mode Enabled` page, in the appropriate widget, such as `Form Group Details` click **Edit**, and add the `Tenant` field ensuring that this field has been marked as a **Required** field and then click **Add**. Click **Save** to save your changes to the SVT.
- SOC analysts at the master node must at the minimum have `Read` access for the `Tenants` module so that they can investigate any incidents that might occur at the tenant node.
- To assign a team to a tenant, on the master node, edit the tenant record on the tenant's page. In the detail view of the tenant record, add or link a team. Once a team is assigned, the records from that tenant will automatically be assigned to that team.
- If you are creating a record on the master node, ensure that you assign that record to the appropriate tenant.
  Important: Once a tenant is assigned to a record, the assignment of that record cannot be changed. If you require to change the assignment of that record, then you must delete the record and re-assign the record correctly.
- Once playbooks that contain On Update, On Create, or Custom API Endpoint triggers are pushed to the tenant they should be deactivated at the master node. Otherwise, the playbook will be triggered at both the nodes.
- On the **tenant** node, disable the FortiSOAR system playbooks in the *System Notification and Escalation Playbooks* collection, except for the **Notify Playbooks**, and disable all the playbooks in the *SLA Management Playbook* collection.

Troubleshooting

For troubleshooting the distributed managed service provider model for multi-tenancy issues that you might face, you can use the `postman.log` located at:

```
/var/log/cyops/cyops-routing-agent/postman.log
```

You can change the logging levels for the postman.log by using the following command:

`# vi /opt/cyops-routing-agent/postman/config.ini` and set the required logging level. By default, the logging level is set to **WARN**.

You can set the following logging levels in the log files:

- **DEBUG**: Low-level system information for debugging purposes.
- **INFO**: General system information.
- **WARN**: Information describing a minor problem that has occurred.
- **ERROR**: Information describing a major problem that has occurred.
- **CRITICAL**: Information describing a critical problem that has occurred.

Deployment Troubleshooting

While connecting to a secure message exchange, retries never stop if the DNS is not resolved

You can continue to retry to connect to the secure message exchange for 10 minutes. A long retry has been specified to ensure that you do not have to restart the service every time the network breaks since all consumers stop after a long network break.

Resolution

You can configure the time for which you will continue retrying to establish a connection with the secure message exchange using the `config` file, located at `/opt/cyops-routing-agent/postman/config.ini`:

```
heartbeat = 120
connection_attempts = 50
retry_delay = 10
```

After adding a tenant on the master, you see a Retry button on the tenant node

This issue can occur due to two reasons:

- Incorrect secure message exchange configuration.
- Incorrect selection of secure message exchange while configuring the tenant on the master node.

Resolution

Correct the secure message exchange configuration or selection and click the **Retry** button.

Configuration Troubleshooting

Failure while configuring master on a tenant node

This issue can occur if you have specified the wrong password or port while configuring the master on the tenant node.

Resolution

On the tenant node, click **Master Configuration** > **Edit Configuration**. On the `Configure Master` dialog and correct the password or TCP port number that you have specified.

If you yet see that the **Enabled** button is in the **NO** state, and you want to enable data replication, toggle the **Enabled** button to **YES**.

Records created at the tenant node do not replicate to the master node even when data replication is turned on

In this case, the FortiSOAR logs contain the following message: `Error message on master "Error: Bad Request for url: <URL> when a field is non-mandatory on tenant and mandatory on master`

This issue occurs if you have switched off replication of a required record field on a tenant node, or the field does not exist on the same module defined at the tenant node. This leads to a failure of record creation on the master node.

Resolution

You must ensure that replication is switched on for all required record fields, and you should also note that if you have done any schema changes in a module at the master node, then you must ensure that the required fields are marked as required across all tenants that are replicating that module.

A user is not able to view records and FortiSOAR displays a 500 error

Resolution

Any user who is created in a multi-tenant environment and who requires to view records must have a minimum of read-only access for the `Tenants` module in their respective role.

Therefore, ensure that you have assigned the user a role that has a minimum of read-only access for the `Tenants` module.

Tenant modules not getting displayed at the master node after initial configuration

You might not see any modules of a tenant node at the master node in **Remote Tenant Manager** > **Manage Modules** after the initial configuration (addition) of the tenant node if there are errors while configuring the tenant.

Resolution

This issue occurs due to the tenant being in the "Verification Failed" state at the time of configuration. To make the MMD of the tenant visible at the master node, restart the "cyops-postman" service at the tenant node.

In case of clustering of secure message exchanges, FortiSOAR is unable to connect to primary secure message exchange even after the primary node has come back online after a failure

If you have your secure message exchanges setup as a cluster for high availability, and you face a failure, and the primary secure message exchange node comes back online and yet the FortiSOAR node is still not able to connect to the secure message exchange. You will see the following error in the postman logs:

```
2018-11-20 10:36:21,022 140437372753664 59b195ab94e35ef70e28ed129c58c804 ERROR pika.callback
callback process(): Calling <bound method BlockingChannel._on_channel_closed of
<BlockingChannel impl=<Channel number=1 CLOSED conn=<SelectConnection OPEN socket=
('<xxx.xxx.xx.xxx', 53378)->('xxx.xxx.xx.xxx', 52011) params=>>>> for "1:Channel.Close"
failed
Traceback (most recent call last):
File "/opt/cyops-routing-agent/.env/lib/python3.4/site-packages/pika/callback.py", line 236,
in process
callback(*args, **keywords)
File "/opt/cyops-routing-agent/.env/lib/python3.4/site-packages/pika/adapters/blocking_
connection.py", line 1358, in _on_channel_closed
method.reply_text)
pika.exceptions.ChannelClosed: (404, "NOT_FOUND - home node 'rabbit@' of durable queue
'queue.postman.data.remoterequest.805553bfece2f6a5895c8db8c54b9ae0' in vhost 'vhost_
59b195ab94e35ef70e28ed129c58c804' is down or inaccessible")
```

Resolution

Stop and start the rabbitmq app on the secure message exchange node using the following commands:

```
# rabbitmqctl stop_app
```

```
# rabbitmqctl start_app
```

Shifting the Secure Message Exchange of tenants leads to MMDs not being pushed to the tenants that have been shifted to the new Secure Message Exchange

If you have shifted the Secure Message Exchange of any tenant in your multi-tenant configuration to a new Secure Message Exchange, then the remote mmd management does not work since the `uwsgi` service keeps the Secure Message Exchange details in memory and uses it to publish to the Secure Message Exchange. However, when the Secure Message Exchange is changed for any tenant the change gets notified using "rabbitmq publish" and the updated Secure Message Exchange setting is available only to the `postman` service; the `uwsgi` still has old Secure Message Exchange details.

Resolution

After you have shifted any tenant to a new Secure Message Exchange, then you must update the router settings in the `uwsgi` service by restarting the `postman` and `uwsgi` services on the master node using the following commands:

```
# systemctl restart uwsgi
# systemctl restart cyops-postman
```

You also need to restart the `postman` service on the tenant node.

Known Issues and Workarounds

- **Issue #45664**: Playbooks that are already pushed on the Tenant node will not be replaced if these playbooks have been moved to a different collection on the Master node.
  To replace the playbooks on the tenant node, you must delete the existing collection, which contains the playbook that you want to replace, on the Tenant node.
- **Issue #46368**: Parent playbooks whose child playbooks contain the "Remote Reference Playbook" step, will remain in the **Awaiting** stage indefinitely if the parent playbook requires some input or response from the child playbook. For example, if you have a parent playbook that has called a child playbook that requires some response from a tenant or if that tenant cannot be reached, then the parent playbook will remain in the Awaiting stage indefinitely till there is some response from the child playbooks.
  Currently, there is no workaround for this issue.
- **Issue #78595**: The master node gets stuck during the "Publish" operation and displays the "Publish initiated. Awaiting response from tenant" after changing the MMD at a tenant node.
  This issue occurs when the "Publish" operation from the master node to a tenant node fails due to any reason such as, changing the type of a field from "string" to "integer" that will result in failure of the "Publish" operation.
  To resolve this issue, refresh the browser window which enables the **Revert All** button. Click Revert All to revert all the changes made after which you will be able to use the system as before the "Publish" operation.

www.fortinet.com