|  |  |  |  |
| --- | --- | --- | --- |
| **Enterprise Business Process** | Optum Connect | **Last Updated Date** | 08/03/2022 |
| **Revision #** | 0.1 | **Process Owner** | Charles Colstrom |
| **Written/Revised By** | Naresh Malyala | | |

Table of Contents

[1. Purpose 3](#_Toc129213467)

[2. Scope 3](#_Toc129213468)

[3. Introduction 3](#_Toc129213469)

[4. Benefits of Turbonomic 3](#_Toc129213470)

[5. Benefits of Integrating Turbonomic with Splunk 4](#_Toc129213471)

[A. Pre-Requisites for Integrating Turbonomic with Splunk 4](#_Toc129213472)

[B. Steps for Integrating Turbonomic with Splunk 5](#_Toc129213473)

[6. Benefits of Integrating Turbonomic with New Relic 6](#_Toc129213474)

[A. Pre-Requisites for Integrating Turbonomics with New Relic 7](#_Toc129213475)

[B. Steps for Integrating Turbonomic with New Relic 8](#_Toc129213476)

[7. Benefits of Integrating Turbonomic with Service Now 9](#_Toc129213477)

[A. Pre-Requisites for Integrating Turbonomics with Service Now 10](#_Toc129213478)

[B. Steps for Integrating Turbonomic with Service Now 11](#_Toc129213479)

[8. Benefits of Integrating Turbonomic with Single Sign On (SSO) 12](#_Toc129213480)

[A. Pre-Requisites for Integrating Turbonomic with SSO SAML 13](#_Toc129213481)

[B. Steps for Integrating Turbonomic with SSO SAML 14](#_Toc129213482)

# **Purpose**

The Purpose of this document is to identify the scope/integration plans to incorporate Turbonomic as a service in Service Catalog for clients as a part of OCC Engineering.

# **Scope**

The Optum Connect Engineering (OCE) team will leverage Turbonomic to be provided as a cost Management tool in the package.

# **Introduction**

Turbonomic is a cloud optimization and management platform that can help organizations manage their cloud resources more efficiently and effectively, ultimately leading to cost saving.

Turbonomic uses artificial intelligence (AI) and machine learning (ML) algorithms to analyze application performance, resource utilization, and workload demand, and then automatically optimize the underlying infrastructure to ensure optimal performance and cost efficiency.

# **Benefits of Turbonomic**

* Turbonomic can help with cost management by identifying and eliminating wasteful spending on cloud resources. The platform can analyze usage patterns and recommend changes to resource allocation or configurations that can reduce unnecessary costs.
* Turbonomic can provide real-time visibility into cloud spending and usage, allowing organizations to monitor and manage costs proactively.

# **Benefits of Integrating Turbonomic with Splunk**

* Enhanced visibility: Integrating Turbonomic with Splunk allows us to collect and analyze data from both systems in a single location, providing enhanced visibility into Infrastructure performance and health.
* Improved troubleshooting: With Turbonomic intelligent workload automation capabilities and Splunk's log analysis and search functionality, user can quickly identify and resolve issues in Infrastructure.
* Optimized performance: By combining Turbonomic's automated workload placement and optimization with Splunk's analytics capabilities, user can ensure environment is always running at optimal performance.
* Better decision-making: Integrating Turbonomic with Splunk provides a holistic view of Infrastructure, which enables user to make informed decisions about workload placement, capacity planning, and resource allocation.
* Simplified management: With Turbonomic and Splunk integrated, user can manage Infrastructure from a single console, streamlining management tasks and reducing administrative overhead.

# **Pre-Requisites for Integrating Turbonomic with Splunk**

* A working Turbonomic installation: A working instance of Turbonomic installed and configured to collect data from cloud environment.
* A working Splunk installation: A working instance of Splunk installed and configured to receive data from Turbonomic.
* Access to the Turbonomic API: To integrate Turbonomic with Splunk, user must have access to the Turbonomic REST API.
* Permissions: User must have the necessary permissions to configure data inputs and alerts in Splunk.
* Network connectivity: The Turbonomic instance and Splunk instance must be able to communicate with each other over the network.
* Turbonomic Add-on for Splunk: Need to download and install the Turbonomic Add-on for Splunk from Splunkbase.
* Field mapping: User must understand the data fields used by both Turbonomic and Splunk to map these fields during the integration process.

# **Steps for Integrating Turbonomic with Splunk**

* Install the Splunk Add-on for Turbonomic: Download and install the Splunk Add-on for Turbonomic on Splunk instance. This add-on enables Splunk to collect and index data from Turbonomic.
* Configure the Turbonomic REST API: In the Turbonomic UI, create a new API token with the necessary permissions to retrieve the data needed to collect in Splunk. Note down the API token to use it later.
* Configure inputs: In the Splunk UI, configure the inputs to collect the data from Turbonomic. Go to the "Inputs" section of the Splunk Add-on for Turbonomic and add the Turbonomic REST API endpoint, specifying the authentication details and the type of data user want to collect.
* Test the inputs: Verify that the data is being collected by testing the inputs in the Splunk UI.
* Create dashboards and visualizations: Use Splunk's search and reporting functionality to create dashboards and visualizations to analyze the data collected from Turbonomic.
* Fine-tune the integration: Refine the integration to collect only the necessary data and optimize the performance.

# **Benefits of Integrating Turbonomic with New Relic**

* Turbonomic and New Relic are two powerful tools that can be integrated to provide a comprehensive monitoring and optimization solution for modern IT environments. Here are some benefits of integrating Turbonomic with New Relic.
* Enhanced visibility: By integrating Turbonomic with New Relic, user can gain enhanced visibility into entire IT infrastructure. user can monitor and analyze applications and network in real-time to identify any performance issues and their root causes.
* Improved performance: The integration of Turbonomic with New Relic can help optimize application performance by dynamically allocating resources based on workload demands. Turbonomic's AI-driven platform analyzes real-time data to automatically adjust resources and ensure optimal performance, while New Relic provides real-time insights into the performance of applications and infrastructure.
* Increased efficiency: The integration of Turbonomic and New Relic can help increase efficiency by automating resource allocation and workload management. With Turbonomic's AI-driven platform, user can automate resource allocation to ensure that the right resources are allocated to the right workloads, while New Relic provides real-time insights into the performance of applications and infrastructure, allowing user to make informed decisions.
* Better collaboration: By integrating Turbonomic with New Relic, user can enable better collaboration between their IT teams. Turbonomic's platform provides a single source of truth for all their IT resources, while New Relic's platform provides real-time insights into application performance. This allows teams to work together more effectively to resolve issues and optimize performance.
* Cost savings: By optimizing resource allocation and improving performance, the integration of Turbonomic with New Relic can help reduce costs, Allocation of resources more efficiently, reducing the need for overprovisioning, and identify and resolve performance issues quickly, reducing downtime and associated costs.

# **Pre-Requisites for Integrating Turbonomics with New Relic**

* Access to both Turbonomic and New Relic: Need access to both the Turbonomic and New Relic platforms to configure the integration. User would also need administrative access to both platforms to set up the integration.
* Credentials: Require the necessary credentials to access both Turbonomic and New Relic APIs. This typically involves creating an API key or OAuth token for each platform.
* Compatibility: Check the compatibility matrix to ensure that the versions of Turbonomic and New Relic that are been used are compatible with each other.
* Network connectivity: Ensure that network allows communication between the Turbonomic and New Relic platforms. If necessary, user may need to configure firewalls or other network settings to allow this communication.
* Understanding of integration requirements: User should have a clear understanding of the integration requirements and the data that needs to be exchanged between Turbonomic and New Relic. This will help in configure the integration correctly and ensure that it meets our needs.
* Configuration: Configure the integration between Turbonomic and New Relic. This typically involves setting up webhooks or configuring API endpoints to exchange data between the two platforms. user should also configure any necessary data mapping or transformations to ensure that the data is properly interpreted by each platform.

# **Steps for Integrating Turbonomic with New Relic**

* Ensure that User have administrative access to both Turbonomic and New Relic.
* Log in to Turbonomic instance and navigate to the Integrations tab.
* Click the New Relic tile and select "Install" to initiate the integration.
* Provide New Relic account ID and API key to Turbonomic, which will be used to authenticate the integration.
* Select the data that needed to be sent from Turbonomic to New Relic. User can choose to send data for CPU, memory, disk, network, and application performance metrics.
* Configure the data mapping between Turbonomic and New Relic. This step involves mapping the data fields from Turbonomic to the corresponding fields in New Relic.
* Test the integration to ensure that the data is being sent correctly from Turbonomic to New Relic.
* Once the integration is complete and tested, user can view the data in New Relic Insights or New Relic APM, depending on the type of data being sent.

# **Benefits of Integrating Turbonomic with Service Now**

* Increased efficiency: The integration enables automatic ticket creation in ServiceNow for Turbonomic alerts. This reduces the need for manual intervention, enabling IT teams to respond to alerts and incidents faster.
* Enhanced visibility: The integration enables IT teams to view Turbonomic data and ServiceNow data in a single console. This provides a unified view of the IT environment, helping teams to make informed decisions.
* Improved collaboration: The integration enables IT teams to collaborate more effectively, as they can share information and insights from both Turbonomic and ServiceNow. This helps to ensure that all stakeholders are on the same page and working towards a common goal.
* Better resource allocation: The integration enables IT teams to optimize resource allocation across the IT environment. Turbonomic provides insights into resource utilization and performance, while ServiceNow provides information on business impact and service levels. Together, these tools enable teams to make more informed decisions about resource allocation.
* Reduced downtime: The integration enables IT teams to identify and resolve issues faster, reducing the risk of downtime. The automatic ticket creation feature ensures that alerts are addressed promptly, before they escalate into major incidents.

# **Pre-Requisites for Integrating Turbonomics with Service Now**

* Access to Turbonomic and ServiceNow: User will need administrative access to both Turbonomic and ServiceNow to configure the integration.
* Network connectivity: Ensure that there is network connectivity between the Turbonomic and ServiceNow instances. user may need to configure firewall rules or VPN connections to enable communication between the two systems.
* Permissions: Ensure that the user accounts being used for the integration have the necessary permissions to access the data and perform the required actions in both Turbonomic and ServiceNow.
* Turbonomic version: Verify that the version of Turbonomic that you are using supports integration with ServiceNow. Check the Turbonomic documentation or support resources for details on compatibility.
* ServiceNow instance: Need to obtain the URL, user credentials, and API key for the ServiceNow instance that you want to integrate with Turbonomic.
* Configuration details: Before configuring the integration, user need determine which Turbonomic alerts they want to send to ServiceNow and how those alerts to be processed ,also need to determine which ServiceNow fields to populate with data from Turbonomic.

# **Steps for Integrating Turbonomic with Service Now**

* Log in to Turbonomic as an administrator and navigate to the Integrations tab.
* Click the ServiceNow tile and select "Install" to initiate the integration.
* Provide the URL, username, password, and API key for your ServiceNow instance to Turbonomic, which will be used to authenticate the integration.
* Select the alerts that need to be sent from Turbonomic to ServiceNow. user can choose from a variety of alert types, including VM placement, workload balancing, and resource utilization.
* Configure the alert mapping between Turbonomic and ServiceNow. This step involves mapping the data fields from Turbonomic to the corresponding fields in ServiceNow.
* Determine how the alerts to be processed in ServiceNow. Configure ServiceNow to create a new incident, update an existing incident, or ignore the alert based on specific criteria.
* Test the integration to ensure that the alerts are being sent correctly from Turbonomic to ServiceNow and are being processed as expected.
* Once the integration is complete and tested, user can view the alerts in ServiceNow and manage them alongside other incidents and tasks.

# **Benefits of Integrating Turbonomic with Single Sign On (SSO)**

* Enhanced Security: SSO can increase the security of Turbonomic by reducing the likelihood of password reuse and simplifying the management of user credentials. This can help reduce the risk of unauthorized access to Turbonomic and protect sensitive information.
* Simplified User Experience: SSO can simplify the user experience of accessing Turbonomic by allowing users to log in using their existing credentials from another system or application. This can reduce the number of login prompts users encounter and make it easier for users to access Turbonomic.
* Reduced Administrative Overhead: SSO can reduce the administrative overhead of managing user accounts in Turbonomic by allowing user accounts to be centrally managed through the identity provider (IDP). This can simplify the process of granting and revoking access to Turbonomic and reduce the risk of errors in managing user accounts.
* Improved Compliance: SSO can help improve compliance with industry regulations and standards, such as HIPAA and GDPR, by providing a centralized and auditable source of user authentication and authorization.
* Cost Savings: SSO can help organizations reduce costs associated with managing user accounts and passwords by simplifying the process of creating and managing user accounts, reducing the need for IT support, and reducing the risk of account lockouts and password resets.
* Overall, using SSO with Turbonomic can provide numerous benefits, including improved security, simplified user experience, reduced administrative overhead, improved compliance, and cost savings.

# **Pre-Requisites for Integrating Turbonomic with SSO SAML**

* Identity Provider (IDP): Need to have an IDP that supports the SAML 2.0 protocol, such as Microsoft Active Directory Federation Services (ADFS), Okta, or Ping Identity.
* Turbonomic Version: Need to have a version of Turbonomic that supports SSO using SAML.
* SSL Certificate: An SSL certificate installed on the Turbonomic server to ensure secure communication between the IDP and Turbonomic.
* Service Account: Create a service account in Turbonomic that has administrative privileges to configure SSO settings.
* User Mapping: Define how user accounts will be mapped between the IDP and Turbonomic. This includes mapping user attributes, such as email addresses or usernames, to the appropriate Turbonomic user accounts.
* SAML Configuration: Configure the SAML settings in both the IDP and Turbonomic. This includes setting up the SSL certificate, defining the user mapping, and configuring the SAML endpoints and attributes.
* Test Environment: It is recommended to set up a test environment before configuring SSO in a production environment. This allows us to verify that the SSO configuration is working correctly before applying it to a live environment.

# **Steps for Integrating Turbonomic with SSO SAML**

* Configure the Identity Provider (IDP): Set up the IDP to support SAML 2.0 and create a new application for Turbonomic. Obtain the SAML metadata file from the IDP.
* Configure Turbonomic: Log in to Turbonomic as an administrator and navigate to the SSO Configuration page. Upload the SAML metadata file from the IDP, configure the SSO settings, and map the IDP attributes to Turbonomic attributes.
* Test the SSO Configuration: Verify that the SSO configuration is working correctly by logging out of Turbonomic and then logging back in through the IDP. Verify that the user account is mapped correctly and that the user has the expected level of access.
* Configure Additional SSO Settings (Optional): Configure additional SSO settings, such as the SSO authentication type and session timeout.
* Deploy the SSO Configuration to Production: Once the SSO configuration is working correctly in the test environment, deploy the configuration to the production environment.