



APPDYNAMICS

Alert and Respond

AppDynamics Pro Documentation

Version 3.8.x

1. Alert and Respond	3
1.1 Policies	5
1.1.1 Configure Policies	10
1.2 Health Rules	14
1.2.1 Configure Health Rules	25
1.2.2 Import and Export Health Rule Configurations	36
1.2.3 Troubleshoot Health Rule Violations	39
1.3 Actions	44
1.3.1 Notification Actions	47
1.3.2 Diagnostic Actions	49
1.3.3 Cloud Auto-Scaling Actions	52
1.3.4 Custom Actions	53
1.3.5 Remediation Actions	53
1.4 Action Suppression	59
1.4.1 Configure Action Suppression	63
1.5 Email Digests	66
1.5.1 Configure Email Digests	67
1.6 Getting Started Wizard for Alerts	70
2. Alerting for Business Transaction Health Problems	72
3. Best Practices for Alerting and Integrating with Third Party Alerting Systems	73

Alert and Respond

Alerts let you know when problems exist and help you anticipate problems that might be developing. Responses let you automate preventative actions to address those problems before they cause a severe slowdown or outage. Think of alert and respond as the automation of your runbooks.

The alert and respond system is made up of three parts:

- **Health rules:** Use these rules to define key performance metric thresholds for your application, across the stack.
- **Policies:** Use policies to link health rule violations, and other performance-based events, with appropriate actions.
- **Actions:** Use actions to specify what should be done in a wide variety of situations, including sending alerts and performing diagnostic and remedial tasks.

Out of the box, AppDynamics recognizes some broad-based health issues commonly experienced by applications, such as "Business Transaction response time is much higher than normal" or "Memory utilization is too high". These are configured as default health rules, which define how high is "much higher than normal" or "too high". Use policies to attach your these rules to alerts (whom to notify) and responses (what to do) when these problems exist. You can use these rules "as is" or modify them for your environment. See [Default Health Rules](#).

In addition to the broad-based rules, you can customize precise automatic alerts and responses for very narrowly circumscribed situations. This lets you finely tune your system, ensuring that the right alert goes to the right person, the right action is taken for the right problem on the right cluster or server.

Notifications

Alerts

For example:

- You do not want to alert your team if performance in a few clusters is lagging, but if more than 20% of the clusters are unhealthy, or if servers in particular clusters or servers that meet certain criteria are performing poorly, you do want to trigger an alert. You can define health rules that apply to specific tiers or nodes. If these rules violate the system knows exactly which entity is experiencing problems and therefore whom to alert. This rule affects only one node: the order processing server.

What does this Health Rule affect?

☐ Tiers

☒ Nodes

Select what Nodes this Health Rule affects

Type of Nodes **All Nodes**

These specified Nodes:

Selected Nodes (1)

Name	Tier
Node_8001	Order Processing Server

< ADD

REMOVE >

Tip: You can drag items between these lists

Other Nodes (3)

Name	Tier
Node_8000	ECommerce Server
Node_8002	Inventory Server
Node_8003	ECommerce Server

Refresh List

- Performance is deteriorating in one business transaction so you want to view snapshots for that one transaction. You create a diagnostic action.

Create Diagnostic Session Action

Name

Duration minutes

Snapshot rate snapshots per minute

Business Transactions to run on

☐ Affected Business Transactions

☒ These Business Transactions:

Selected Business Transactions (1)

Name
ViewCart.addToCart

< ADD

REMOVE >

Tip: You can drag items between these lists

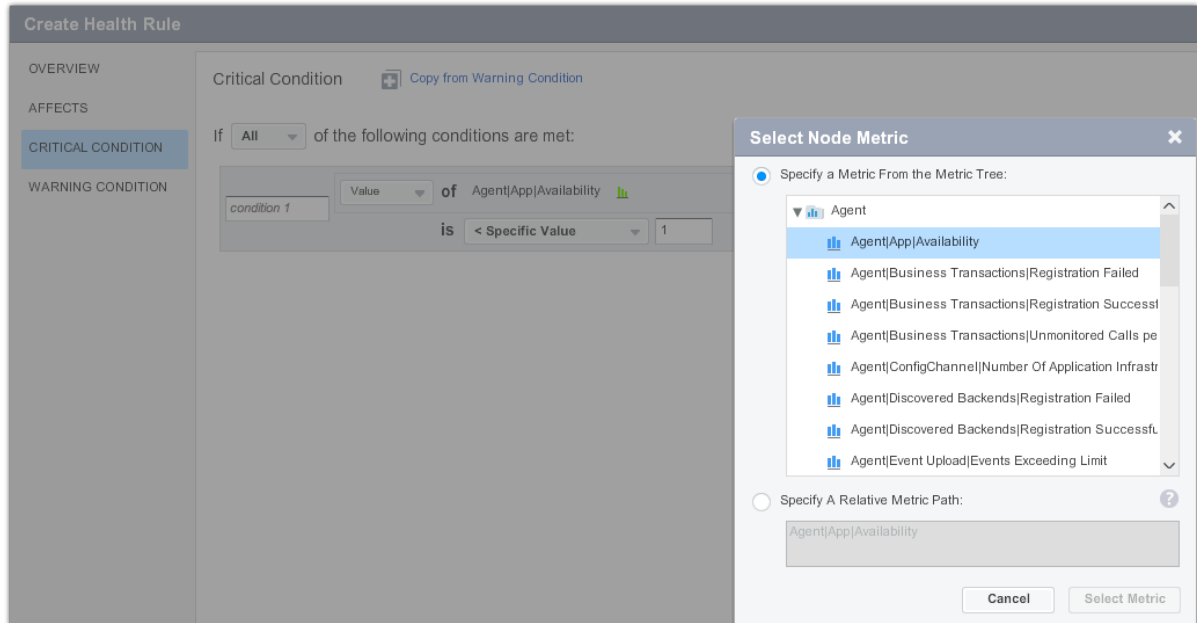
Available Business Transactions (6)

Name
ViewItems.getAllItems
UserLogin.memberLogin
/appdynamicspilot/
ViewCart.sendItems
UserLogout.memberLogout
/appdynamicspilot/WEB-INF

Refresh List

- You want to send an alert whenever an app agent stops reporting to the Controller. Create a node health rule based on the value of the Availability metric reported by the agent. If

Availability is less than 1, the agent is not reporting.



- You have a large operation with several development teams, each responsible for a different service. You create a health rule for one service and then copy it. Then you create different policies in which you can pair each copy of the health rule to an alert addressed to the appropriate team.
- You have an application that performs well for normal load. However, peak loads can cause the application to slow. During peak load, AppDynamics not only detects the connection pool contention, but also allows you to create a remediation script that can automate increasing or decreasing the size of connection pool. You can require human approval to run this script or simply configure it to execute automatically when it is triggered. Create a Runbook and associate it with a policy so that it will fire when the connection pool is exhausted.

To learn more about using policies, health rules, and actions:

- See these topics:
- Watch [this walk through the process](#).
- [Notifications](#)
- [Actions](#)

Policies

- [Policy Structure](#)
 - [Policy Triggers](#)
 - [Policy Actions](#)
- [Policy List](#)
- [Learn More](#)

Policies let you anticipate problems and take actions to address those problems before they cause a severe slowdown or outage.

Policies provide a mechanism for automating monitoring and problem remediation. Instead of continually scanning metrics and events for the many conditions that could suggest problems, you can proactively define the events that are of greatest concern in keeping your applications running smoothly and then create policies that specify actions to start automatically when those events occur.

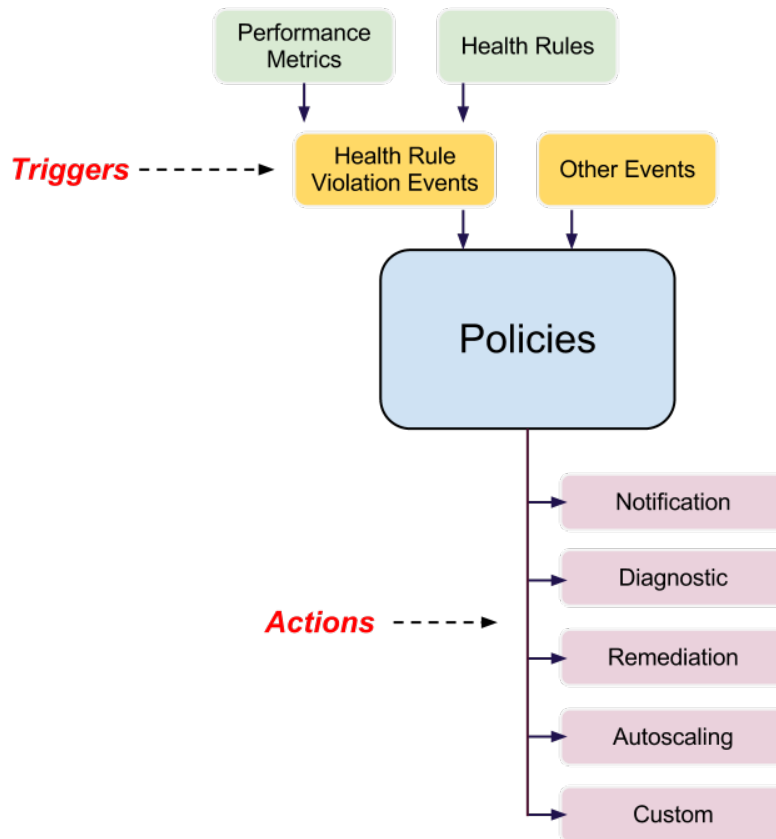


*Health Rules and Policies
from AppDynamics 6 minutes*

Policy Structure

A policy connects two things:

- evaluated event triggers
- actions to be taken in response to those triggers



Policy Triggers

Policy triggers are events that cause the policy to fire. The events can be health-rule violation events or other types of events, such as hitting a slow transaction threshold or surpassing a resource pool limit. See [Health Rules](#), [Troubleshoot Health Rule Violations](#) and [Events](#).

The triggering events can be broadly defined as affecting any object in the application or very narrowly defined as affecting only specific objects. You can create a policy that fires when an event involving all the tiers in the application occurs, or one involving only specific tiers. You can create a policy that fires on events affecting only certain nodes, or only certain business transactions or certain errors. You can very finely tune policies for different entities and situations.

For example, this very broadly-defined policy would fire whenever a resource pool limit (> 80% usage of EJB pools, connection pools, and/or thread pools) is reached for any object in the application.

Create Policy

Name: Enabled: ☒

TRIGGER

This Policy will fire when **any of these Events occur** on **any object**

ACTIONS

Health Rule Violation Events

- ☐ Health Rule Violation Started - Warning
- ☐ Health Rule Violation Started - Critical
- ☐ Health Rule Violation Upgraded - Warning to Critical
- ☐ Health Rule Violation Downgraded - Critical to Warning
- ☐ Health Rule Violation Ended

Other Events

- ☐ Slow Transactions
- ☐ Errors
- ☒ Code Problems
 - ☐ Code Deadlock
 - ☒ Resource Pool Limit Reached
- ☐ Application Changes
- ☐ AppDynamics Config Warnings

On the other hand, this narrowly-defined JVMViolationInWebTier policy fires only when existing health rules on JVM heap utilization or JVM garbage collection time are violated.

Here the triggering events for this policy are configured:

Create Policy

Name: Enabled: ☒

TRIGGER

This Policy will fire when **any of these Events occur** on **any object**

ACTIONS

Health Rule Violation Events

- ☒ Health Rule Violation Started - Warning
- ☒ Health Rule Violation Started - Critical
- ☐ Health Rule Violation Upgraded - Warning to Critical
- ☐ Health Rule Violation Downgraded - Critical to Warning
- ☐ Health Rule Violation Ended

What Health Rules?

☐ Any Health Rule

☒ These Health Rules:

- ☒ JVM Garbage Collection Time is too high
- ☒ CLR Garbage Collection Time is too high

+ Select Health Rule 2 of 7 Selected [Create New Health Rule](#)

Other Events

- ☐ Slow Transactions
- ☐ Errors
- ☐ Code Problems
- ☐ Application Changes
- ☐ AppDynamics Config Warnings

and here the affected object is limited to a specific tier - the ECommerce Server.

Create Policy

Name: JVMViolationInWebTier Enabled: ☒

TRIGGER This Policy will fire when **any of these Events occur** on **these specified objects**

ACTIONS

☐ Any objects
☒ Any of these specified objects:
☐ Business Transactions
☒ Tiers and Nodes
☐ Errors

Events that are associated to ANY of the specified objects will make this Policy fire

Select Tiers or Nodes

☒ Tiers
☐ Nodes

Select Tiers

These specific Tiers

Selected Tiers (1)

Name	Type
ECommerce Server	Application Server

Other Tiers (2)

Name
Inventory Server
Order Processing Server

< ADD REMOVE >

A policy is triggered when at least one of the specified triggering events occurs on at least one of the specified objects.

Policy Actions

The second part of creating a policy is assigning one or more actions to be automatically taken in response to the policy trigger.

For example, for the resource pool violation, you want to take a thread dump and then run a script to increase the pool size.

Create Policy

Name: ResourcePoolPolicy Enabled: ☒

TRIGGER This Policy will fire when **any of these Events occur** on **these specified objects**

ACTIONS

Actions to Execute

Name	Type
ThreadDump	Thread dump
IncreasePool	Run local script

Selected Tiers (1)

Name	Type
ECommerce Server	Application Server

Other Tiers (2)

Name
Inventory Server
Order Processing Server

< ADD REMOVE >

How to see when Actions are executed?

If this Policy fires in response to an Event, the Actions that are executed will be visible in the 'Actions' column on the Events screen.

[View Events](#)

Other common actions include restarting an application server if it crashes, purging a message queue that is blocked, or triggering the collection of transaction snapshots. You can also trigger a custom action to invoke third party systems. See [Build an Alerting Extension](#) for information about custom actions.

See [Actions](#) for more information about the different types of actions.

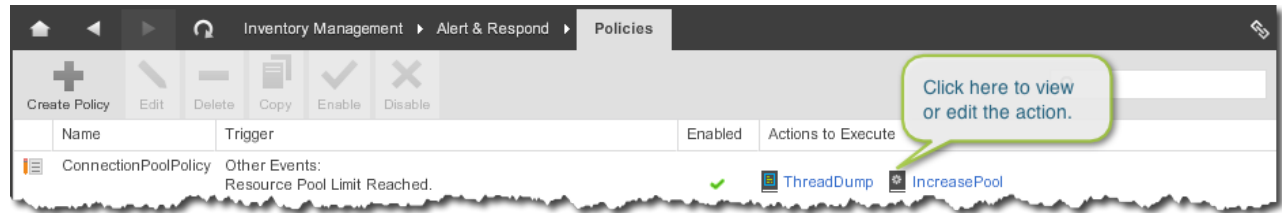
See [Actions Limits](#) for information about limits on the number of actions that the Controller will process.

Because the definition of health rules is separate from the definition of actions, and both health rules and actions can be very precisely defined, you can take different actions for breaching the same thresholds based on context, for example, which tier or node the violation occurred in.

Policy List

To access the list of policies in an application, select **Alert & Respond ->Policies**.

The policy list lists all the policies created for your application, with its triggers and actions taken. You can view and edit an action assigned to a specific policy by clicking the action in the policy list.



Learn More

- [Actions](#)
- [Configure Policies](#)
- [Custom Actions](#)
- [Diagnostic Sessions](#)
- [Events](#)
- [Health Rules](#)
- [Workflow Overview](#)
- [Build an Alerting Extension](#)

Configure Policies

- [Using the Policy Wizard](#)
 - [To access the Policy Wizard](#)
- [Configuring the Policy Trigger](#)
 - [To configure policy triggers](#)
- [Configuring the Policy Actions](#)
 - [To configure policy actions](#)
- [Learn More](#)

Using the Policy Wizard

The Policy Wizard contains two panels:

- **Trigger:** Sets the policy name, enabled status, events that trigger the policy, entities that are affected by the policy
- **Actions:** Sets the actions to take when the policy is triggered.

To access the Policy Wizard

1. Click **Alert & Respond -> Policies** in the left navigation pane.
2. Do one of the following:
 - To create a new policy, click the plus icon.
 - To edit an existing policy, select the policy and click the pencil icon.
 - To remove an existing policy, select the policy and click the minus icon.

Configuring the Policy Trigger

The policy trigger panel defines the events and objects generating the events that cause the policy to fire and invoke its actions.

For policy triggers that depend on health violation events, the health rules must be created before you can create a policy on them. See [Health Rules](#) and [Troubleshoot Health Rule Violations](#).

To configure policy triggers

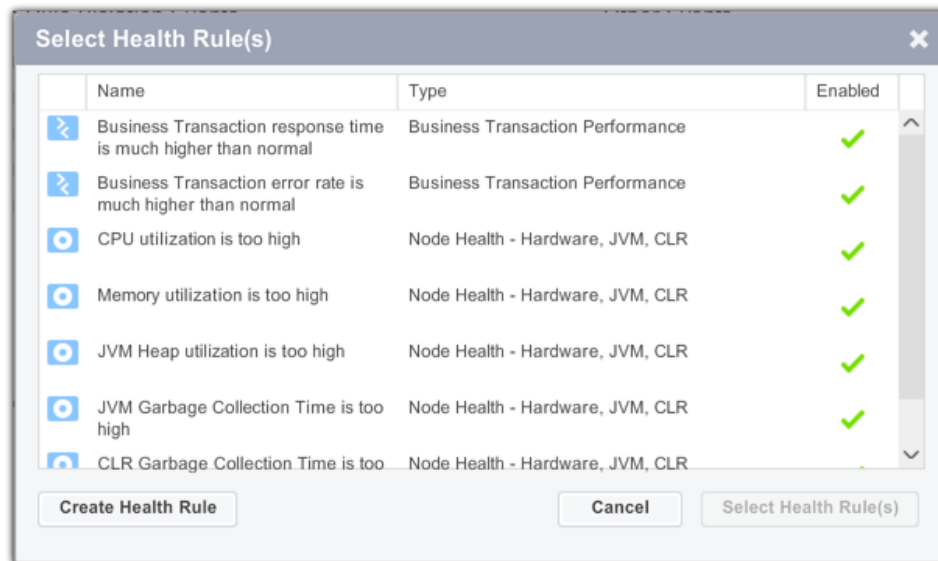
1. Click the plus icon to create a new policy or select an existing policy and click the pencil icon. The Policy Wizard opens.
2. Enter a name for the policy in the Name field.
3. To enable the policy, check the Enabled check box. To disable the policy, clear the Enabled check box.
4. On the left, click **Trigger** if it is not already selected.
5. Check the type of event that should trigger the policy. You may need to click the arrow to expose specific events within an event category.

If you check at least one health rule violation event, you can choose whether any (that is, all) health rule violations or only specific health rule violations will trigger the policy.

This Policy will fire when ▼ **any of these Events occur** on ► **any object**

Health Rule Violation Events	Other Events
<input checked="" type="checkbox"/> Health Rule Violation Started - Warning	▼ <input type="checkbox"/> Slow Transactions
<input checked="" type="checkbox"/> Health Rule Violation Started - Critical	<input type="checkbox"/> Slow Transactions
<input type="checkbox"/> Health Rule Violation Upgraded - Warning to Critical	<input type="checkbox"/> Very Slow Transactions
<input type="checkbox"/> Health Rule Violation Downgraded - Critical to Warning	<input type="checkbox"/> Stalled Transactions
<input type="checkbox"/> Health Rule Violation Ended	<input type="checkbox"/> Errors
What Health Rules?	▼ <input type="checkbox"/> Code Problems
<input type="radio"/> Any Health Rule	<input type="checkbox"/> Code Deadlock
<input checked="" type="radio"/> These Health Rules:	<input type="checkbox"/> Resource Pool Limit Reached
+ Select Health Rule 0 of 7 Selected Create New Health Rule	▼ <input type="checkbox"/> Application Changes
	<input type="checkbox"/> Application Deployment
	<input type="checkbox"/> App Server Restart
	<input type="checkbox"/> Application Configuration Change
	▼ <input type="checkbox"/> Server Crashes
	<input type="checkbox"/> JVM Crash
	▼ <input type="checkbox"/> AppDynamics Config Warnings
	<input type="checkbox"/> Agent Configuration Error

To designate specific health rule violations to trigger the policy, select These Health Rules, click the "+" icon, and then choose the health rules from the embedded health rule browser.



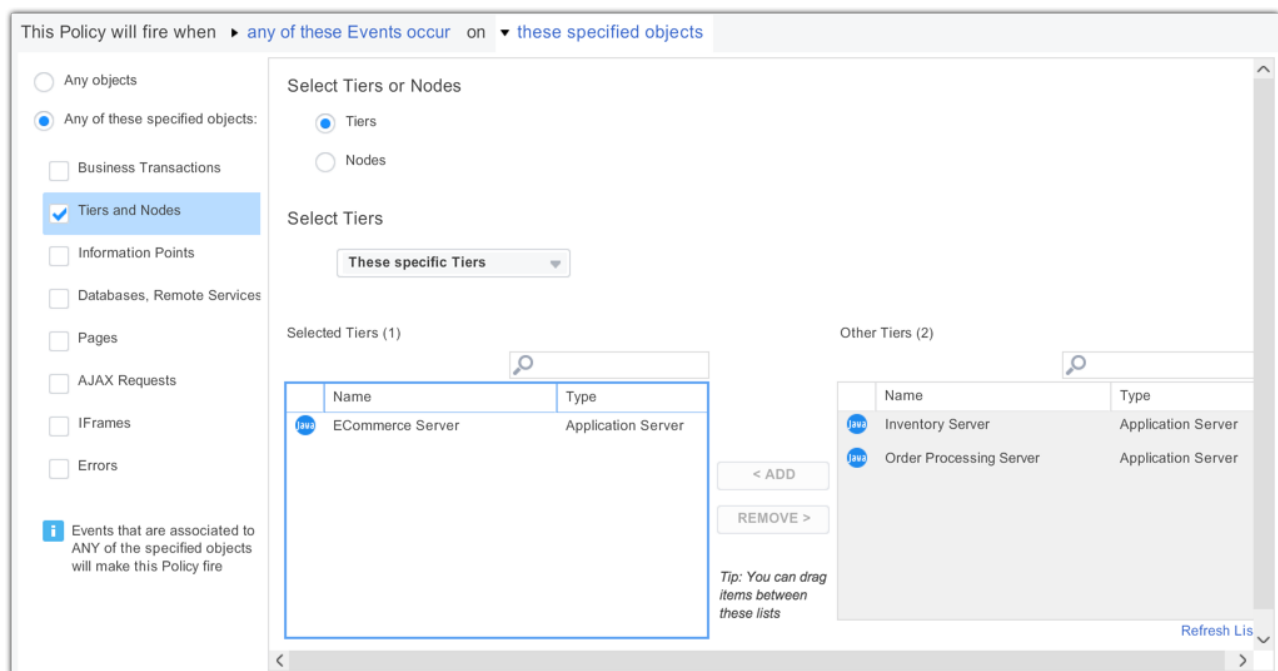
6. When you have finished selecting the events that trigger the policy, click "any object" to configure which objects to monitor for those events in order to trigger the policy.

This Policy will fire when ▼ any of these Events occur on ► any object

If you select **Any Objects** the policy will be triggered by the configured events when they occur on any object in your application.

To restrict the policy to specific objects, select **Any of these specified objects** and then choose the objects.

For example, the following policy fires when selected events occur on the the ECommerce Server tier. You can similarly restrict the objects to specific tiers, business transactions, and so forth.



⚠ If you wish to have policies triggered by Health Rule Violation Events, you should leave the

selection at **Any object**. Selecting **Any of these specified objects** means that only non-health rule events - slow transactions, errors, and so forth - will trigger the policy.

7. Click **Save** to save the policy configuration.

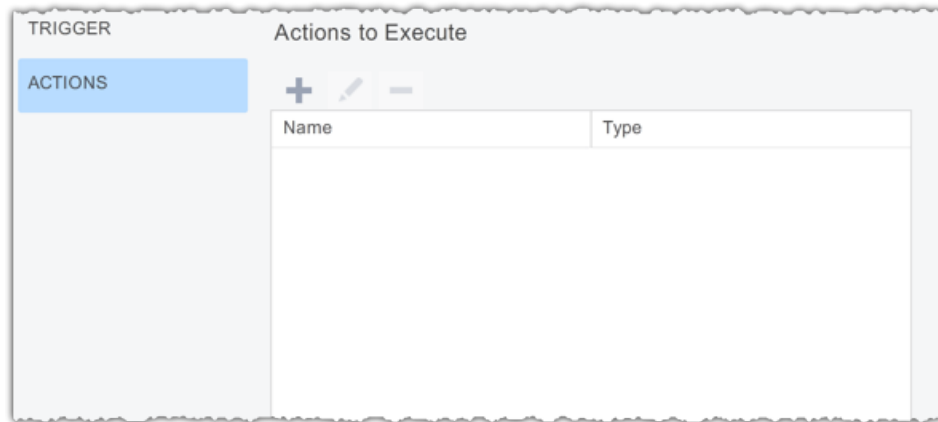
Configuring the Policy Actions

The policy actions panel defines the actions that the policy automatically initiates when the trigger causes the policy to fire.

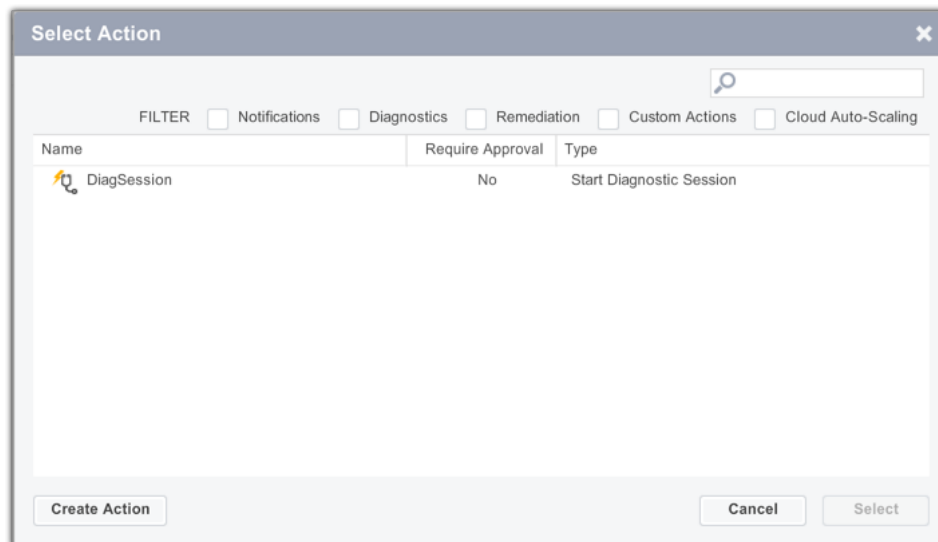
The actions must be created before you can create a policy that fires them. See [Actions](#) and the documentation for individual types of actions (notification actions, remediation actions, etc.) for information on creating an action.

To configure policy actions

1. If you have not already done so, open the policy wizard and edit the policy to which you want to add actions. See [To access the Policy Wizard](#).
2. On the left, click **Actions** if it is not already selected.




3. Click "+" icon. The list of defined actions appears.



You can filter the list by checking the check boxes for the types of actions you want to see.

4. In the list of actions, select the action that you want this policy to execute and click **Select**.

 If you do not see an appropriate action for your needs, click **Create Action** to create an action. For more information on creating actions, see [Actions](#). After you have created the action select it here to assign it to the policy that you are configuring.

5. Click **Save** in the Policy Wizard.

Learn More

- [Policies](#)
- [Events](#)
- [Health Rules](#)
- [Troubleshoot Health Rule Violations](#)
- [Actions](#)
- [Notification Actions](#)
- [Diagnostic Actions](#)
- [Remediation Actions \(Java only\)](#)
- [Auto-Scaling Actions](#)
- [Custom Actions](#)
- [Create a Workflow and Workflow Steps](#)
- [Workflow Overview](#)

Health Rules

- [Understanding the Health Rule Wizard](#)
 - [Health Rule Types](#)
 - [Health Rule Schedules](#)
 - [Health Rule Enabled Schedule](#)
 - [Health Rule Evaluation Window](#)
 - [Health Rule Wait Time After Violation](#)
 - [Health Rule Entities](#)
 - [Entities Affected by a Health Rule](#)
 - [Health Rule Evaluation Scope](#)
 - [Health Rule Conditions](#)
 - [Critical and Warning Conditions](#)
- [Default Health Rules](#)
- [Suggested Metrics for Additional Health Rules](#)
 - [Metrics for Business Transactions](#)
 - [Metrics for Tiers](#)
 - [Metrics for Nodes](#)
 - [Metrics for Backends](#)
- [Preparing to Set Up Health Rules](#)
- [Health Rule Management](#)
- [Learn More](#)

AppDynamics collects a wide range of metric information covering your entire application. Some of those metrics are key indicators of the overall state of the system. Being able to proactively track the status of those indicators gives you a window into the health of your system and can inform you when various important entities—nodes, business transactions, databases, etc.—may be in trouble.

Health rules allow you to define acceptable values for key metrics associated with specific entities, and to monitor those essential metrics automatically. Should metric values exceed the ranges you have specified, the health rule is said to violate, and a health rule violation event occurs and is surfaced in the controller user interface.

The violation event can also be used to trigger a policy, which can initiate pre-defined actions to respond to the situation, from sending alerting emails to running remedial scripts. To understand how to create policies and actions, see [Policies](#) and [Actions](#).

The simplest way to create health rules is to use the basic health rule wizard. The wizard groups commonly used system entities and related metrics to ease the process of setting up your particular system's health rules. Should you need to create health rules that connect less commonly used entities and/or metrics, you can use one of the custom methods in the wizard, the [Custom Health Rule Types](#) or the [Hybrid](#) methods.

For specifics on using the health rule wizard, see [Configure Health Rules](#).

Understanding the Health Rule Wizard


To understand how the health rule wizard works, you need to understand four basic concepts:

- [How the wizard groups entities and metrics](#)
- [How the wizard schedules health rule evaluations](#)
- [How the wizard defines which entities are affected](#)
- [How the wizard defines the metric conditions that are to be evaluated on those entities](#)

Health Rule Types

To simplify creating health rules, the basic health rule wizard groups entities, like nodes or business transactions, with metrics that are commonly associated them into health rule types.

Doing so allows the wizard to automatically show you relevant information during the health rule creation process.

 If your needs are not covered by these types, you can use one of the custom methods in the wizard, [Custom Health Rule Types](#) or the [Hybrid](#) methods.

The health rule types are:

- **Transaction Performance**
 - **Overall Application Performance:** groups metrics related to load, response time, slow calls, stalls, with applications
 - **Business Transaction Performance:** groups metrics related to load, response time, slow calls, stalls, etc. with business transactions
- **Error Rates:** groups metrics related to exceptions, return codes, and other errors with applications or tiers
- **Node Health**

- **Node Health-Hardware, JVM, CLR:** groups metrics like CPU and heap usage, disk I/O, etc. with nodes
- **Node Health-Transaction Performance:** groups metrics related to load, response time, slow calls, stalls, etc. with nodes
- **Node Health-JMX:** groups metrics related to connection pools, thread pools, etc with nodes
- **Databases & Remote Services:** groups metrics related to response time, load, or errors with databases and other backends
- **End User Experience**
 - **Pages:** groups metrics like DOM building time, JavaScript errors, etc. with the performance of application pages for the end user
 - **Iframes:** groups metrics like first byte time, requests per minute, etc. with the performance of iframes for the end user
 - **Ajax Requests:** groups metrics like Ajax callback execution time, errors per minute, etc. with the performance of Ajax requests for the end user
- **Information Points:** groups metrics like response time, load, or errors with information points

If you select one of these health rule types, AppDynamics automatically presents you with a list of the commonly associated metrics, simplifying the health rule creation process by giving you a manageable number of relevant options.

If the types do not cover the entities and/or metrics you wish to use, you can use one of the custom options:

- Select the Custom health rule type, which allows you to create a rule based on any metric AppDynamics collects on any single entity that AppDynamics monitors
- Use the Hybrid method, which allows you to create a rule based on any metric AppDynamics collects across multiple entities. If you wish to create health rules based on custom metrics that cover multiple entities, see [Using the Hybrid Method](#).

Health Rule Schedules

The metrics associated with a health rule are evaluated according to a schedule that you control. You can configure:

- [when a health rule is in effect](#)
- [which data set should be used, based on time](#)
- [what special rules should be in place during a violation event](#)

Health Rule Enabled Schedule

By default, health rules are always enabled. But you can also configure your own schedules during which the rule is in effect.

Built-in schedules are:

- End of business hour
- Weekday lunch
- Weekday mornings
- Weekdays
- Weekends

You can also create a new schedule based on UNIX cron expressions using your custom values.

Health Rule Evaluation Window

Different kinds of metrics may provide better results using different sets of data. You can manage how much data AppDynamics uses when it evaluates a particular health rule by setting the data collection time period. The default value is 30 minutes.

For metrics based on an average calculation, such as average response time, AppDynamics averages the response time over the evaluation window - a five minute window means that the last five minutes of data is used to evaluate if the health rule is in range. For metrics based on a sum calculation, such as number of calls, AppDynamics uses the total number of calls counted during the evaluation window. And so forth. Use values that work best with the kinds of metrics you are interested in.

Health Rule Wait Time After Violation

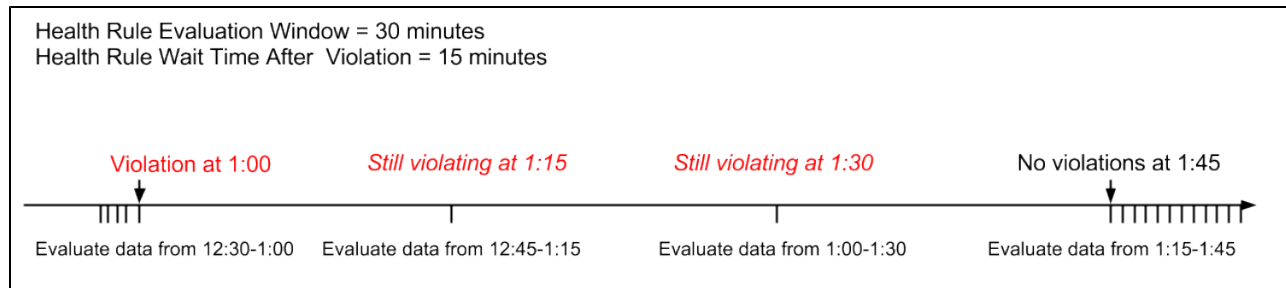
Normally health rules are evaluated every minute. But once a violation has been detected, continuing to evaluate the rule by the minute can generate a large number of policy triggers, alerts, and actions, which may not be helpful in the situation. You can set an after-violation wait period to sleep the evaluation process for a specific number of minutes, to allow remedial action to be taken before the next evaluation occurs.

The default for the wait time after violation is 30 minutes.

After the wait time has elapsed, the violated health rule is once again evaluated. By now, the evaluation window has moved, so the data that is evaluated is for a later time period than before. At this point, one of three situations can arise:

1. The health rule is no longer being violated. In this case the evaluation returns to its normal schedule which is to evaluate to health rule every minute.
2. The health rule is still being violated, but there has been no state change. In other words, if there was a critical violation, there is still a critical violation. In this case AppDynamics waits another wait time period before re-evaluating the health rule.
3. The health rule is still being violated, but there has been a state change; for example the violation has escalated from warning to critical or de-escalated from critical to warning. In this case AppDynamics waits another wait time period before re-evaluating the health rule.

The timeline below illustrates how evaluation works with a 30-minute evaluation window and a 15-minute wait period after violation. In this scenario, AppDynamics evaluates the rule every minute until 1:00, when it detects a violation. It then switches to a 15-minute evaluation frequency, which it maintains until the violation is no longer detected, at which time it switches back to a 1-minute evaluation frequency. Note that the evaluation window is still moving during this time, so that only the last 30 minutes of data is evaluated. The slice of data that was evaluated when the violation was first detected is not the same set of data that was evaluated when the violation ceased.

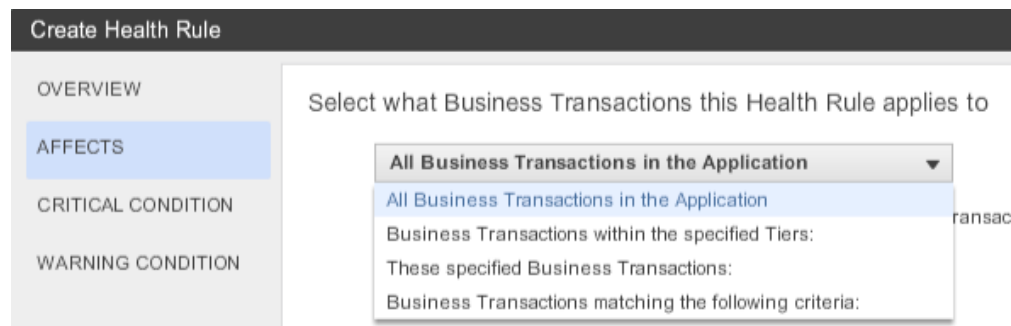


Health Rule Entities

A health rule can evaluate metrics associated with an entire application or a very limited set of entities. For example, you can create business transaction performance health rules that evaluate certain metrics for all business transactions in the application or node health rules that cover all the nodes in the application or all the nodes in specified tiers. The default health rules are in this category.

You can also create health rules that are very narrowly applied to a limited set of entities in the application, or even a single entity such as a node or a JMX object or an error. For example, you can create a JMX health rule that evaluates the initial pool size and number of active connections for specific connection pools in nodes that share certain system properties.

The health rule wizard lets you specify precisely which entities the health rule affects, enabling the creation of very specific health rules. For example, for a business transaction you can limit the tiers that the health rule applies to or specific business transactions by name or by names that match certain criteria.



For the node health rules, you can specify that a health rule applies only to nodes that meet certain criteria:

What does this Health Rule affect?

☐ Tiers
☒ Nodes

Select what Nodes this Health Rule affects

Type of Nodes All Nodes

Nodes matching the following Criteria:

☐ Specify Node Name
☒ Specify Node Properties / Variables

Nodes with the following Meta-Info Properties ?

+ Add Meta Info Criteria

Nodes with the following Environment Variables ?

+ Add Environment Variable Match Criteria

Nodes with the following JVM System Environment Properties (java nodes only) ?

+ Add JVM System Property Match Criteria

Entities Affected by a Health Rule

If you are using the basic health rule wizard and health rule types, AppDynamics provides a list of commonly used sets of entities on which metrics can be evaluated.

For an Overall Application Performance health rule type, the health rule applies the entire application, regardless of business transaction, tier, or node.

For a Business Transaction Performance health rule type, you can apply the health rule to:

- All Business Transactions in the application
- All Business Transactions within tiers that you select
- Individual Business Transactions that you select
- Business Transactions with names that have patterns matching criteria that you specify (such as all Business Transactions with names that start with "INV")

For an Error Rates health rule type, you can apply the health rule to:

- All Errors in the application
- Specific error types that you select
- Errors with the specified tiers
- Errors with names that have patterns matching criteria that you specify

For a Node Health – Transaction Performance or Node Health – Hardware, JVM, CLR health rule types, you can apply the health rule to:

- All tiers in the application
- Individual tiers that you specify
- All nodes in the application
- Nodes types, such as Java nodes, PHP nodes, etc.
- Nodes within specified tiers

- Individual nodes that you specify
- Nodes with names, meta-data, environment variables or JVM system environment properties with matching criteria that you specify

For a Node Health – JMX health rule type, you select

- the JMX objects on which the health rule is evaluated
- and apply the health rule to:
- All nodes in the application
 - Nodes within tiers that you specify
 - Individual nodes that you specify
 - Nodes with names matching criteria that you specify

For a Databases & Remote Services health rule type, you can apply the health rule to:

- All databases and remote services in the application
- Individual databases and remote services that you specify
- Databases and remote services with name matching criteria that you specify

For End User Experience – Pages, iframes, and Ajax Requests health rule types, you can apply the health rule to:

- All such entities
- Entities that you specify
- Entities with names matching criteria that you specify


For Information Points health rule types, you can apply the health rule to:

- All information points
- Information points that you specify
- Information points with names matching criteria that you specify

Using the Custom health rule type limits you to a single entity hard-coded in the metrics themselves. If you want to use custom metrics but associate them with multiple entities, use the hybrid method. See [Using the Hybrid Method](#).

Health Rule Evaluation Scope

The health rule evaluation scope defines how many nodes in the affected entities must violate the condition before the health rule is considered violated.

 Evaluation scope applies only to business transaction performance type health rules and node health health rules in which the affected entities are defined at the tier level.

For example, you may have a critical condition in which the condition is unacceptable for any node, or you may want to trigger the violation only if the condition is true for 50% or more of the nodes in a tier.

Options for this evaluation scope are:

- any node – If any node exceeds the threshold(s), the violation fires.
- percentage of the nodes – If x% of the nodes exceed the threshold(s), the violation fires.
- number of nodes – If x nodes exceed the threshold(s), the violation fires.

- the tier average – Evaluation is performed on the tier average instead of the individual nodes.

Health Rule Conditions

You define the acceptable range for a metric by establishing health rule conditions. A health rule condition defines what metric levels constitute a Warning status and what metric levels constitute a Critical status.

A condition consists of a Boolean statement that compares the current state of a metric against one or more static or dynamic thresholds based on a selected baseline. If the condition is true, the health rule violates. The rules for evaluating a condition using multiple thresholds depend on configuration.

Static thresholds are straightforward. For example, is a business transaction's average response time greater than 200 ms?

Dynamic thresholds are based on a percentage in relation to, or a standard deviation from, a baseline built on a rolled-up baseline trend pattern. For example, a daily trend baseline rolls up values for a particular hour of the day during the last thirty days, whereas a weekly trend baseline rolls up values for a particular hour of the day, for a particular day of the week, for the last 90 days. For more information about baselines, see [Behavior Learning and Anomaly Detection](#).

You can define a threshold for a health rule based on a single metric value or on a mathematical expression built from multiple metric values.

The following are typical conditions:

- IF the value of the Average Response Time is greater than the default baseline by 3 X the Baseline Standard Deviation . . .
- IF the count of the Errors Per Minute is greater than 1000 . . .
- IF the number of MB of Free Memory is less than 2 X the Default Baseline . . .
- IF the value of Errors per Minute/Calls per Minute over the last 15 days > 0.2 . . .

The last example combines two metrics in a single condition. You can use the expression builder in the health rules wizard to create conditions based on a complex expression comprising multiple interdependent metrics

Often a condition consists of multiple statements that evaluate multiple metrics. A health rule is violated either when one of its condition evaluates to true or when all of its conditions evaluate to true, depending on how it is configured. You can correlate multiple metrics to focus the health rules for your environment.

For example, a health rule that measures response time (average response time greater than some baseline value) makes more business sense if it is correlated with the application load (for example, 50 concurrent users or 10,000 calls per minute) on the system. You may not want to use the response time condition alone to trigger a policy that initiates a remedial action if the load is low, even if the response time threshold is reached. To configure this correlation, the first part of the condition would evaluate the actual performance measurement and the second part would ensure that the health rule is violated only when there is sufficient load.

Critical Condition [+ Copy from Warning Condition](#)If **All** of the following conditions are met:[+ Add Condition](#)

Slow	Value	of	Average Response Time (ms)	is	> Baseline	Weekly Trend - Last 3 months	by	4	Baseline Standard Deviation(s)
Busy	Value	of	Calls per Minute	is	> Specific Value			10000	

Critical and Warning Conditions

Conditions are classified as either critical or warning conditions.

Critical conditions are evaluated before warning conditions. If you have defined a critical condition and a warning condition in the same health rule, the warning condition is evaluated only if the critical condition is not true.

The configuration procedures for critical and warning conditions are identical, but you configure these two types of conditions in separate panels. You can copy a critical condition configuration to a warning configuration and vice-versa and then adjust the metrics in the copy to differentiate them. For example, in the Critical Condition panel you can create a critical condition based on the rule:

- IF the Average Response Time is greater than 1000

Then from the Warning Condition panel, copy that condition and edit it to be:

- IF the Average Response Time is greater than 500

As performance changes, a health rule violation can be upgraded from warning to critical if performance deteriorates to the higher threshold or downgraded from critical to warning if performance improves to the warning threshold.

Default Health Rules

Out of the box, AppDynamics provides a default set of health rules:

Health Rule Name	Health Rule Type
Business Transaction response time is much higher than normal	Business Transaction Performance
Business Transaction error rate is much higher than normal	Business Transaction Performance
CPU utilization is too high	Node Health – Hardware, JVM, CLR Performance
Memory utilization is too high	Node Health – Hardware, JVM, CLR Performance

JVM Memory Heap is too high	Node Health – Hardware, JVM, CLR Performance
JVM Garbage Collection Time is too high	Node Health – Hardware, JVM, CLR Performance
CLR Garbage Collection Time is too high	Node Health – Hardware, JVM, CLR Performance

If any of these predefined health rules are violated, the affected items are marked in the UI as yellow-orange, if it is a Warning violation and red, if it is a Critical violation.

In many cases the default health rules may be the only health rules that you need. If the conditions are not configured appropriately for your application, you can edit them. You can also disable the default health rules.

Suggested Metrics for Additional Health Rules

The default health rules provide the basics for monitoring the health of your business applications. To extend the monitoring capabilities for a particular application, you can configure additional health rules. The following are suggested metrics you might want to monitor with health rules, based on what many customers have found useful.

Metrics for Business Transactions

- Calls Per Minute
- Slow Call Rate
- Stalls

Metrics for Tiers

- Average Response Time
- Calls Per Minute
- Error Rate
- Slow Call Rate
- Stalls

Metrics for Nodes

 Some of these metrics might apply only to certain types of nodes, such as those running JVMs.

- Availability
- Thread Pool – Utilization Rate
- Thread Pool - Average Wait Time
- Thread Pool – Queue Size
- Connection Pool – Utilization Rate
- Connection Pool – Wait Time to Acquire a Connection
- Thread Contention

Metrics for Backends

- Average Response Time
- Calls Per Minute
- Error Rate

Preparing to Set Up Health Rules

AppDynamics recommends the following process to set up health rules for your application:

1. Identify the key metrics on the key entities that you need to monitor for your application. These metrics should be representative of the overall health of your application.
2. Click **Alert & Respond -> Health Rules** to examine the default health rules that are provided by AppDynamics.
 - Compare your list of metrics with the metrics configured in these rules.
 - If the default health rules cover all the key metrics you need, determine whether the pre-configured conditions are applicable to your environment. If necessary, modify the conditions for your needs.
 - You can also view the list of affected entities for each of the default health rules and modify the entities.
3. If the health rules do not cover all your needs or if you need very finely-applied health rules to cover specific use cases, create new health rules.
 - First, identify the type of the health rule that you want to create. See [Health Rule Types](#).
 - Then decide which entities should be affected by the new rule. See [Entities Affected by a Health Rule](#).
 - Then define the conditions to monitor.
4. Create schedules for health rules, if needed.

In some situations a health rules is more useful if it runs at a particular time. See [Health Rule Schedules](#).
5. If desired, configure policies and actions that should come into play when health rules are violated. See [Policies](#) and [Actions](#).

Health Rule Management

To view current health rules in an application, including the default health rules, and to access the health rule wizard, click **Alert & Respond -> Health Rules**.

Current health rules are listed in the left panel. If you click one of these rules, a list appears in the right panel showing what entities this selected health rule affects and what the status of the latest evaluation is. You can also select the Evaluation Events tab to see a detailed list of evaluation events.

In the left panel you can directly delete or duplicate a health rule. From here you can also access the health rule wizard to add a new rule or edit an existing one.

To delete an existing health rule:

1. Select the health rule in the left panel.
2. Click the minus icon.

The health rule is removed.

To duplicate an existing health rule:

1. Select the health rule in the left panel.
2. Click the copy icon.

The health rule is duplicated under the name you assign.

To edit an existing health rule:

1. Select the health rule in the left panel.
2. Click the pencil icon.

The health rule wizard appears, with the rule's current values configured. You can modify these values in the wizard.

To create a new health rule:

1. Click the plus icon.
2. The health rule wizard appears with some default values configured. You define the health rule in the wizard.

See [Configure Health Rules](#) for details on using the health rule wizard.

Learn More

- [Notification Actions](#)
- [Configure Health Rules](#)
- [Configure Baselines](#)
- [Events](#)
- [Policies](#)
- [Actions](#)

Configure Health Rules

- [Accessing the Health Rule Wizard](#)
 - [The Structure of the Health Rule Wizard](#)
- [Using the Basic Health Rules Wizard](#)
 - [Configure Generic Health Rule Settings for the Basic Health Rule Wizard](#)
 - [To Create a New Health Rule Schedule](#)
 - [Configure Affected Entities for the Basic Health Rules Wizard](#)
 - [Configure Health Rule Conditions for the Basic Health Rules Wizard](#)
 - [To Create a Condition](#)
 - [To Configure a Condition Component](#)
 - [To Remove a Condition Component](#)
 - [To Build an Expression](#)
- [Using the Custom Health Rule Types Method](#)
- [Using the Hybrid Method - Custom Health Rules for Multiple Entities](#)
- [Learn More](#)

This topic describes the detailed steps for configuring health rules using the health rule wizard.

Accessing the Health Rule Wizard

Click **Alert & Respond->Health Rules**.

1. To edit an existing health rule, in the left panel of the health rule list, select the health rule and click the pencil icon.
2. To create a new health rule, click the plus icon.

The Structure of the Health Rule Wizard

The health rule wizard contains four panels:

- **Overview:** Sets the health rule name, enabled status, health rule type, health rule enabled period, and health rule evaluation time.
- **Affects:** Sets the entities evaluated by the health rule. The options presented vary according to the health rule type set in the Overview panel.
- **Critical Condition:** Sets the conditions, whether all or any of the conditions need to be true for a health rule violation to exist, and the evaluation scope (BT and node health policies defined at the tier level only); it also includes an expression builder to create complex expressions containing multiple metrics.
- **Warning Condition:** Settings are identical to Critical Condition, but configured separately.

You can navigate among these panels using the **Back** and **Next** buttons at the bottom of each panel or by clicking their entries in the left panel of the wizard. You should configure the panels in order, because the configuration of the health rule type in the Overview panel determines the available affected entities in the Affects panel as well as the available metrics in the Condition panels.

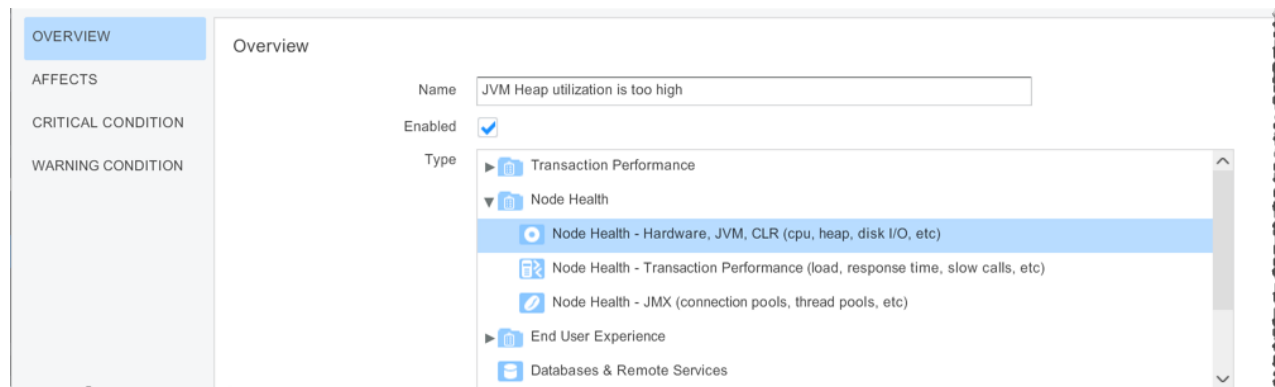
The basic health rule wizard uses some short cuts to define the most commonly use health rules. If you need to set up rules that are outside of that set you should use the [Custom Health Rule Types](#) method or the [Hybrid](#) method.

Using the Basic Health Rules Wizard

Most of the time you should use the basic health rules wizard to set up your health rules.

Configure Generic Heath Rule Settings for the Basic Health Rule Wizard

You configure generic settings for your health rules in the Overview panel.



1. Set a name. If a name already exists, you can change it if you like.
2. If you want the health rule in an enabled state (the default), check Enabled. Clear to disable.

3. Select a health rule type by clicking on the name in the list.

This setting affects metrics offered for configuration in subsequent panels in the basic wizard, so you must select a health rule type before continuing to other panels. If none of the predefined health rule types are applicable for your needs, you need to use the Custom method or the Hybrid method.

4. If the health rule is always (24/7) enabled, check the Always check box.

When is this Health Rule Enabled? Always ☒

During these times: End of Business Hour: 5pm-6pm, Mon-Fri Create New Schedule

Use the last 30 minutes of data when evaluating the Health Rule

Wait Time after Violation Health Rules are evaluated every minute.

If a Health Rule violation occurs, wait 30 minutes before evaluating it again for that affected entity (Business Transaction, Tier, Node, etc).

For example, if a Health Rule violates for Business Transaction 'Checkout' at 1:00pm, and this wait time is set to 30 min, then the Health Rule will not be evaluated again for 'Checkout' until 1:30pm.

? Cancel < Back Next > Save

If the health rule is enabled only at certain times, clear the Always check box and either:

- Select a predefined time interval from the drop-down menu.
- or
- Click **Create New Schedule**.

The Create Schedule window opens. See [To Create a New Health Rule Schedule](#).

i The time is the time at the site of the controller, not the app agent. For example, if the enabled time is set to 5pm-6pm, Mon-Fri and the controller is in San Francisco but the app agent is in Dubai, the health rule engine uses San Francisco time.

5. Click the dropdown menu **Use the last <> minutes of data** and select a value between 1 and 360 minutes for the evaluation window. This is the amount of recent data to use to determine whether a health rule violation exists. This value applies to both critical and warning conditions. See [Health Rule Evaluation Window](#).

6. In the Wait Time after Violation section, enter the number of minutes to wait before evaluating the rule again for the same affected entity in which the violation occurred. See [Health Rule Wait Time After Violation](#).

7. Click **Save**.

8. Click **Next**.

To Create a New Health Rule Schedule

1. In the Overview window of the Health Wizard, clear the Always check box if it is checked.

2. Click **Create New Schedule**.
2. Enter a name for the schedule.
3. Enter an optional description of the schedule.
4. Enter the start and end times for the schedule as cron expressions. See <http://www.quartz-scheduler.org/documentation/quartz-2.1.x/tutorials/crontrigger> for details about and examples of cron syntax.
5. Click **Next**.

Configure Affected Entities for the Basic Health Rules Wizard

The Affects panel lets you define which thing your health rule affects. These things are called *entities*. For example, a business transaction, a node or set of nodes, or an information point are all entities. The choices you are offered depends on the health rule type you chose in the Overview panel. In this case, the health rule type selected affects Tiers or Nodes.

1. Use the dropdown menu to select the the entities affected by this health rule.
The entity affected and the choices presented in the menu depend on the health rule type configured in the Overview window.
See [Entities Affected by a Health Rule](#) for information about the types of entities that can be affected by the various health rule types.
2. If you select entities based on matching criteria, specify the matching criteria.
3. If you are configuring a JMX health rule, select the JMX objects that the health rule is evaluated on.
4. Click **Next**.

Configure Health Rule Conditions for the Basic Health Rules Wizard

The high-level process for configuring conditions is:

1. Determine the number and kind of metrics the health rule should evaluate. For each performance metric you want to use, create a condition.

- You can use a single condition component or multiple condition components for a single condition state.
- You can use values based on complex mathematical expressions.

2. Decide whether the health rule is violated if all of the tests are true or if any single test is true.

3. For business transaction performance health rules and node health rule types that specify affected entities at the tier level, decide how many of the nodes must be violating the health rule to produce a violation event. See [Health Rule Evaluation Scope](#).

4. To configure a critical condition use the Critical Condition window. To configure a warning condition use the Warning Condition window.

i The configuration processes for critical and warning conditions are identical.

You can copy the settings between Critical and Warning condition panels and just edit the fields you desire. For example, if you have already defined a critical condition and you want to create a warning condition that is similar, in the Warning Condition window click **Copy from Critical Condition** to populate the fields with settings from the Critical condition.

To Create a Condition

1. In the Critical Condition or Warning Condition window, click **+ Add Condition** to add a new condition component.

The row defining the component opens. See [To Configure a Condition Component](#). Continue to add components to the condition as needed.

2. From the drop-down menu above the components, select All if **all** of the components must evaluate to true to constitute violation of the rule. Select Any if a health rule violation exists if **any** single component is true.

3. For health rules based on the following health rule types:

- business transaction
- node health-hardware
- node health-transaction performance

you must specify evaluation scope.

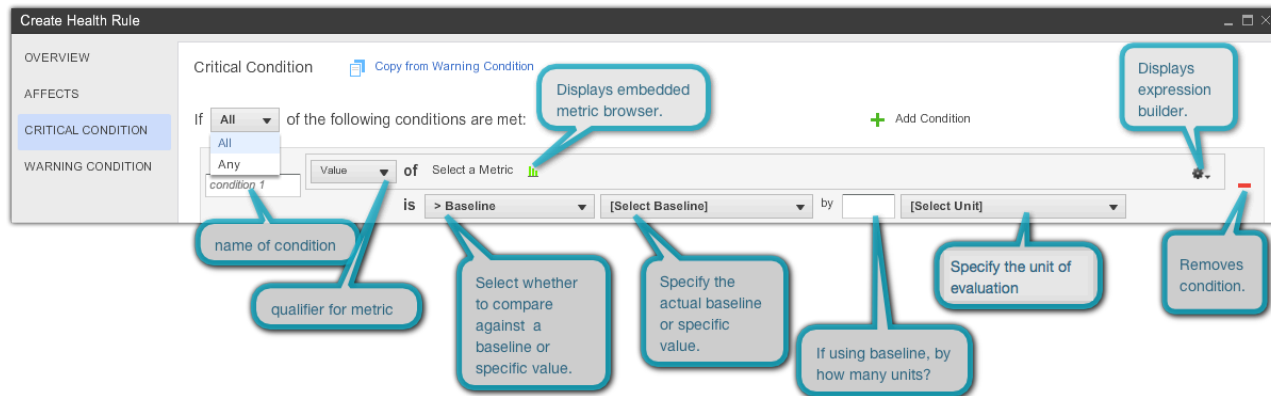
Health Rule will violate if the conditions above evaluate to true for:

- ☐ the Tier Average (the aggregate of all Nodes)
- ☒ Any Node
- ☐ % of the Nodes
- ☐ of the Nodes

If the **Health Rule will violate if the conditions above evaluate to true** section is visible, click the appropriate radio button to set the evaluation scope.

If you select percentage of nodes, enter the percentage. If you select number of nodes, enter the absolute number of nodes.

To Configure a Condition Component



1. In the first field of the condition row, name the condition.

This name is used in the generated notification text and in the AppDynamics console to identify the violation.

2. To select the metric on which the condition is based, do one of the following:

- To specify a simple metric, click the metric icon to open a small metric browser. The browser displays metrics appropriate to the health rule type. Select the metric to monitor and click **Select Metric**. The selected metric appears in the configuration.

or

- To build an expression using multiple metric values, click the gear icon at the end of the row and select **Use a mathematical expression of 2 or more metric values**. This opens the mathematical expression builder where you can construct the expression to use as the metric. See [To Build an Expression](#).

3. From the Value drop-down menu before the metric, select the qualifier to apply to the metric from the following options:

Qualifier Type	What This Means
Minimum	The minimum value reported across the configured evaluation time length. Not all metrics have this type.
Maximum	The maximum value reported across the configured evaluation time length. Not all metrics have this type.
Value	The arithmetic average of all metric values reported across the configured evaluation time length.
Sum	The sum of all the metric values reported across the configured evaluation time length.

Count	The number of times the metric value has been measured across the configured evaluation time length.
Current	The value for the current minute.

4. From the drop-down menu after the metric, select the type of comparison by which the metric is evaluated.

- To limit the effect of the health rule to conditions during which the metric is *within* a defined distance (standard deviations or percentages) from the baseline, select **Within Baseline** from the menu. To limit the effect of the health rule to when the metric is *NOT within* that defined distance, select **Not Within Baseline**. Then select the baseline to use, the numeric qualifier of the unit of evaluation and the unit of evaluation. For example:

Within Baseline of the Default Baseline by 3 Baseline Standard Deviations

- To compare the metric with a static literal value, select **< Specific value** or **> Specific Value** from the menu, then enter the specific value in the text field. For example:

Value of Errors per Minute > 100

- To compare the metric with a baseline, select **< Baseline** or **> Baseline** from the drop-down menu, and then select the baseline to use, the numeric qualifier of the unit of evaluation and the unit of evaluation. For example:

Maximum of Average Response Time is > Baseline of the Daily Trend by 3 Baseline Standard Deviations

See [Baselines and Periodic Trends](#) for information about the baseline options.

Baseline Percentages

The "baseline percentage" is the percentage above or below the established baseline at which the condition will be triggered. If, for example, you have a baseline value of 850 and you have defined a baseline percentage of "> 1%", then the condition should trigger if the value is > [850+(850x0.01)] or 859. In addition, in order to prevent too small sample sets from triggering health rules violations, these rules are **not** evaluated if the load (the number of times the value has been measured) is less than 1000. So if, for example, a very brief time slice is specified, the rule may not violate even if the conditions are met, simply because the load is not large enough.

5. Click **Save**.

To Remove a Condition Component

You can remove a component condition by clicking the delete icon.

To Build an Expression

To access the expression builder to create a complex expression as the basis of a condition, click the gear icon at the end of the row and select **Use a mathematical expression of 2 or more metric values**.

In the expression builder, use the Expression pane to construct the expression.

Use the Variable Declaration pane to define variables based on metrics to use in the expression.

1. In Variable Declaration pane of the Mathematical Expression builder, click **+ Add variable** to add a variable.
2. In the Variable Name field enter a name for the variable.
3. Click **Select a metric** to open a small metric browser
4. From the drop-down menu select the qualifier for the metric.
5. Repeat steps 1 through 4 for each metric that you will use in the expression.
You can remove a variable by clicking the delete icon.
6. Build the expression by typing the expression in the Expression pane. Click the **Insert Variable** button to insert variables created in the Variable Declaration pane.

Mathematical Expression

Expression

Type in a mathematical expression below using a combination of mathematical operators (eg. +, -, *, /, and ()), constants (eg. 1, 2, etc.), and variables declared above enclosed in curly brackets (eg. {numSlows}). An example of a valid Expression would be 2+{x} or {variableName1} + (4 * (5 - {variableName2})) .

{numSlows} + {numVerySlows} +

Variable Declaration

First declare variables to represent your desired Metric Expressions. For example, you can declare numSlows = Value of Number of Slow Calls.

Variable Name	Variable Definition
numStalls	Value of Stall Count
numVerySlows	Value of Number of Slow Calls
numSlows	Value of Number of Slow Calls

+ Add Variable

Insert Variable...

numStalls - Value of Stall Count
numVerySlows - Value of Number of Slow Calls
numSlows - Value of Number of Slow Calls

Cancel

Use Expression

7. When the expression is built, click **Use Expression**.

The expression appears as the metric in the condition configuration window.

condition 1

{numSlows} + {numVerySlows} + {numStalls}

Edit Expression

is

> Specific Value

100

Using the Custom Health Rule Types Method

The procedure for using the Custom health rule type option in the health rules wizard is very similar to using the basic health rules wizard. Use this procedure if you want to create a health rule for a single entity based on a custom metric.

1. In the Overview section, select Custom (use any metrics) in the Type field.
2. In the Affects section, select the kind of entity that the health rule affects: Business Transaction Performance, Node Performance, or Application Performance.

- If the health rule affects Business Transaction Performance:
 - Select the Business Transaction performance radio button
 - Select the Business Transaction affected from the drop-down menu. You can use the search field in the Select a Business Transaction browser that opens to locate the specific Business Transaction.
- If the health rule affects Node Performance:
 - Select the Node Performance radio button.
 - Click **Node (for new configuration) or Change** (to modify an existing configuration).
 - In the Node browser that opens, navigate to the node which you want the health rule to affect.
 - Click **Select** .
- If the health rule is not specific to a Business Transaction or Node, select Application Performance.

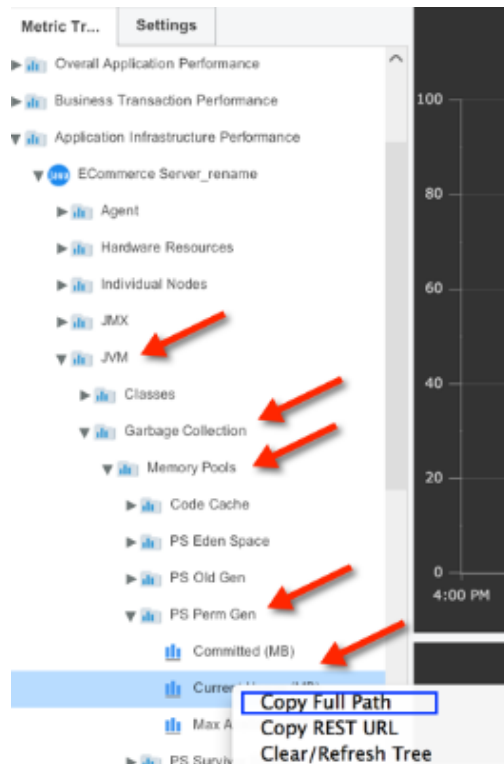
3. When you go on to [Configure Health Rule Conditions for the Basic Health Rules Wizard](#) you will have access to the entire metric browser when you are choosing metrics.

Using the Hybrid Method - Custom Health Rules for Multiple Entities

If you use the [Custom Health Rule Types](#) method, you must set up your evaluation conditions on an entity by entity basis. If you want to use the same custom metrics for evaluation across multiple entities, you should use the **Specify a Relative Metric Path** method when you are setting up your conditions instead.

1. In the Overview section, choose the health rule type that covers the kind of entity for which you wish to set up custom rules. So, for example, if you wish to set up a custom rule that applies to all nodes based on Perm Gen usage, select Node Health - Hardware, JVM, CLR as your health rule type.

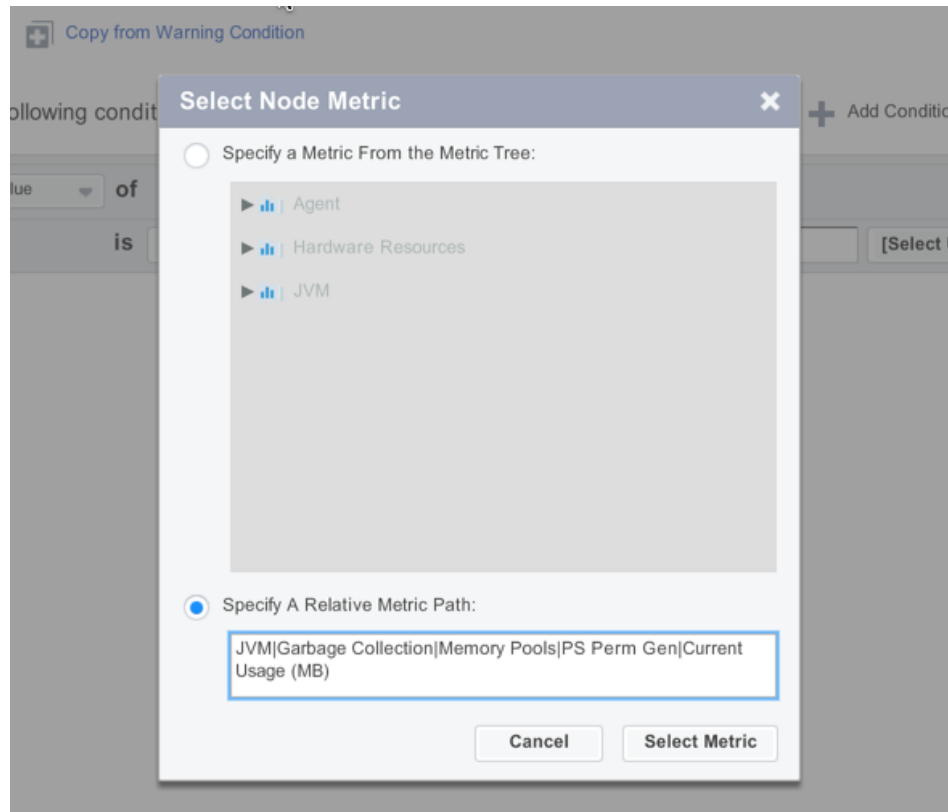
2. Go to the metric browser (**Analyze->Metric Browser**) and locate the relative path of the metric you wish to add. Right click on the metric and select Copy Full Path.



3. In the Affects section, select the correct type of entity. For example, select All Nodes in the Application.

4. Go on to [Configure Health Rule Conditions](#) in the normal way. When you click to bring up the metric list to use for evaluation, select Specify a Relative Metric Path instead of Specify a Metric from the Metric Tree. Use a cropped value for the metric you located in the metric browser and complete configuring your conditions as usual.

- For all health rule types **except** Node Health-Hardware, JVM, CLR or Custom, use the metric name alone - for example, Average Wait Time (ms))
- For Node Health-Hardware, JVM, CLR and Custom health rule types, use everything after the entity, for example, after the Node name. In the example the path would look like this.



Finish configuring your condition as you normally would.

Learn More

- [Import and Export Health Rule Configurations](#)

Import and Export Health Rule Configurations

- [Exporting Health Rules from an Application](#)
 - [To export the configurations for all health rules in an application](#)
 - [To export the configuration for a single health rule](#)
- [Importing Health Rules to an Application](#)
 - [To import the configurations for health rules in an application](#)
- [Learn More](#)

You can export your health configurations from one application to another using a special AppDynamics REST API. This capability allows you to re-use health rule configurations in different applications instead of re-configuring each application manually from the AppDynamics console.

Exporting Health Rules from an Application

Exports are HTTP GET operations.

To export the configurations for all health rules in an application

`http://<controller-host>:<controller-port>/controller/healthrules/<application-name|application-id>`

Example

```
http://pml.appdynamics.com:80/controller/healthrules/3
```

produces the output in [all_health_rules](#).

To export the configuration for a single health rule

```
http://<controller-host>:<controller-port>/controller/healthrules/<application-name|application-id>?name=<health_rule_name>
```

For example:

```
http://pml.appdynamics.com/controller/healthrules/3?name=Business  
Transaction response time is much higher than normal
```

produces the output in [one_health_rule](#).

Importing Health Rules to an Application

Exports are HTTP POST operations.

After you have exported health rules you can import them to a different application passing the xml file created by the export operation as payload to the POST. You can modify the exported file before you import it. You might want to do this to add or remove one or more health rule configurations or to change their names.

Use UTF-8 URL encoding of the URI before posting; for example, do not replace a space (" ") with "%20" in the URI.

The default behavior is not to overwrite an existing health rule of the same name. If you want to overwrite an existing health rule of the same name, specify the **overwrite=true** parameter. since the default is **false**.

The syntax is the same for importing one health rule configurations or several. All the health rule configurations in the xml files are imported.

To import the configurations for health rules in an application

```
http://<controller-host>:<controller-port>/controller/healthrules/<application-name|application-id>?overwrite=true|false
```

This example imports the health rule in the uploaded file, overwriting any health rules of the same name.

▶
http://pm1.appdynamics.com:80/controller/healthrules/3?overwrite=true

☐ GET
☒ POST
☐ PUT
☐ PATCH
☐ DELETE
☐ HEAD
☐ OPTIONS
☐ Other

Raw
Form
Headers

Raw
Form
Files (1)
Payload

Add new file field

Choose Files
all_health_rules
fileUpload
x
all_health_rules (30.8 KB)

application/xml
Set "Content-Type" header to overwrite this value.

The next example imports the health rules without overwriting. In this case, any health rules in the destination controller that have the same names as health rules in the all_health_rules file are not overwritten.

▶
http://pm1.appdynamics.com:80/controller/healthrules/3

☐ GET
☒ POST
☐ PUT
☐ PATCH
☐ DELETE
☐ HEAD
☐ OPTIONS
☐ Other

Raw
Form
Headers

Raw
Form
Files (1)
Payload

Add new file field

Choose Files
all_health_rules
fileUpload
x
all_health_rules (30.8 KB)

application/xml
Set "Content-Type" header to overwrite this value.

Learn More

- [Configure Health Rules](#)

- [Import and Export Transaction Detection Configuration for Java](#)

Troubleshoot Health Rule Violations

- [Health and Health Rules](#)
- [Troubleshoot Health Rule Violations](#)
 - [To find all health rule violations](#)
 - [To troubleshoot a health rule violation](#)
 - [To see health rule status in the UI](#)
- [Learn More](#)

Health and Health Rules

"Health" throughout the AppDynamics UI refers to the extent to which the component being monitored is operating within the acceptable limits defined by health rules. Health rules allow you to automate pro-active monitoring and problem mediation in your managed environment. By default, AppDynamics provides a set of basic health rules, which you can extend, add to, or remove as your needs dictate.

A health rule violation exists when the conditions that define the rule are true. For example, you might have defined a health rule condition that states that a CPU%Busy rate of more than 90% on any node is a critical condition. If the rate on a node then goes over 90%, the health rule is said to "violate" and the AppDynamics UI displays a notification of that violation.

Because there is a set of default health rules, you may see health rule violations reported for your application even if you have not set up your own health rules. If you see violations reported for the APPDYNAMICS_DEFAULT_TXT business transaction, these are for default health rule violations in the All Other Traffic business transaction. If you are not interested in monitoring these business transactions, you may want to examine your business transaction setup. See [Organizing Traffic as Business Transactions](#).

For general information about health rules, see [Health Rules](#). For information on setting up your own health rules, see [Configure Health Rules](#).

Troubleshoot Health Rule Violations

To start troubleshooting health rule violations, you can:

- [Get a list of all the health rule violations](#) by clicking **Troubleshoot -> Health Rule Violations**.
- Click on a particular health rule violation you see [displayed in the UI](#).

You can access the list of health rule violations in your application for the selected time range.

To find all health rule violations

1. In the left navigation pane, click **Troubleshoot -> Health Rule Violations**.
The list of health violations displays.

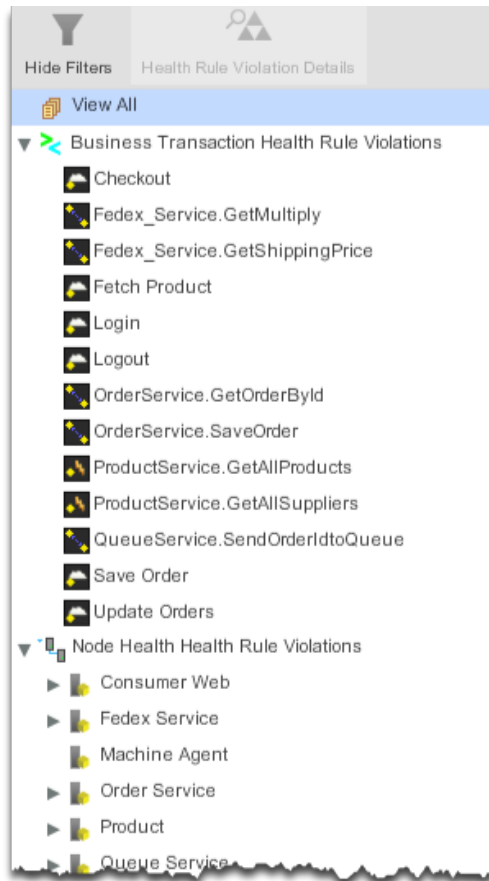
Status	Health Rule	Description	Start Time	End Time	Duration	Affects
Open	Business Transaction response time Business Transaction Performance (load, response time, slow calls, etc)	Appdynamics has detected a problem with Business Transaction UserLogin.memberLogin . Business Transaction response time is much higher than normal continues to violate with critical . All of the following conditions were found to be violating For Business Transaction UserLogin.memberLogin	03/20/14 1:59:55 PM	-	Ongoing (1 hour, 5 minutes)	UserLogin.memberLogin
Open	Business Transaction response time Business Transaction Performance (load, response time, slow calls, etc)	Appdynamics has detected a problem with Business Transaction ViewItems.getAllItems . Business Transaction response time is much higher than normal continues to violate with critical . All of the following conditions were found to be violating For Business Transaction ViewItems.getAllItems :	03/20/14 1:59:55 PM	-	Ongoing (1 hour, 5 minutes)	ViewItems.getAllItems
Open	Error-All Error Rates (exceptions, return codes, etc)	Appdynamics has detected a problem with Error Rates com.singularity.inconsistency.ResourceConsumptionHandler - com.singularity.ee.agent.appagent.services.transactionmonitor.error.a . Error-All continues to violate with warning .	03/20/14 9:18:55 AM	-	Ongoing (5 hours, 46 minutes)	com.singularity.inconsistency.
Open	Node-Jvm Node Health - Hardware, JVM, CLR (cpu, heap, disk I/O, etc)	Appdynamics has detected a problem with Tier ECommerce Server_rename . Node-Jvm continues to violate with critical . All of the following conditions were found to be violating For Node Node_8000 : 1) condition 1	03/19/14 6:10:55 PM	-	Ongoing (20 hours, 54 minutes)	ECommerce Server_rename
Open	test Business Transaction Performance (load, response time, slow calls, etc)	Appdynamics has detected a problem with Business Transaction /appdynamicspilot/ . test continues to violate with critical . All of the following conditions were found to be violating For Business Transaction /appdynamicspilot/ :	03/19/14 6:02:55 PM	-	Ongoing (21 hours, 2 minutes)	/appdynamicspilot/

2. Select **View All Health Rule Violations in the Time Range** or **View Only Health Rule Violations Open Now**.

It is possible that health rule violations that were reported are no longer open because remedial action has been taken or performance has improved on its own.

3. To see the filters click **Show Filters**. To hide them click **Hide Filters**.

With the filters showing in the left filters panel you can select the health rule violations that you want to troubleshoot. You can view all health rule violations or expand the nodes in the tree to select by health rule type (such as business transaction health rules or node health rules) or affected entity (such as business transaction, tier or node).



You can filter health rule violations by entering the name of the health rule in the search field on the upper right.

The health rule violations are displayed in the right panel, with their status, description, start time, end time and duration (if ended), and the affected entity.

To troubleshoot a health rule violation

Once you have located the violation you are interested in, you can get more information in three ways:

- To see the health rule definition that was violated for a specific violation, find the health rule violation in the list and in the Health Rule column, click the link to the health rule configuration. The Edit Health Rule window for the specific definition appears.
- To see the dashboard for the entity, such as a business transaction or a node, affected by the violation, click the link to the entity in the Affects column. The Transaction Flow Map appears.
- To troubleshoot a specific health rule violation, select the health rule violation row from the list and click **Health Rule Violation Details** in the top bar.

The Health Rule Violation Event window displays a summary of the violation and any actions that were executed to respond to it.

Health Rule Violation Event

Summary

Actions Executed

Type

Health Rule

BT-1

Health Rule Type

Business Transaction Performance (load, response time, slow calls, etc)

Affects

/BasicSample/1.pojoservlet

View Dashboard During Health Rule Violation

Violation State

Resolved

Duration

9 minutes

From: 01/24/13 10:05:55 PM To: 01/24/13 10:15:55 PM

No Actions Executed

What does this mean?

Actions Executed

Description

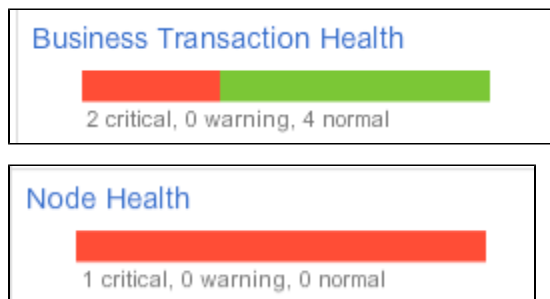
Health rule violation status changed from None to Open (Critical level) for health rule 'BT-1' of type Business Transaction Performance. All of the following conditions were evaluated for each node on tier8086 for business transaction performance /BasicSample/1.pojoservlet and 1 nodes were found to be violating thresholds. For Node node8086:1) condition 1The condition 1 observed value 450 was greater than the threshold 10 for the last 3 minutes.

You can click the **View Dashboard During Health Rule Violation** in the details window to view the dashboard at the time the violation occurred. The time range in this and all other dashboards is set to the time range of the health rule violation. From the dashboard you can get an overall picture of the application at the time of the violation. If you select the Transaction Snapshots tab you get a list of relevant snapshots which allows you to drill down to the root cause of the problem. See [Transaction Snapshots](#) for more information.

To see health rule status in the UI

Across the UI, health rule status is color-coded: green is healthy; yellow/orange is a warning condition; and red is a critical condition. If you see a health rule violation reported in the UI, you can click it to get more information about the violation.

Here are the health summary bars on the dashboards:



There is a health column in the business transaction list:

View Dashboard More Actions View Options		
Name	Health	Server Time (ms)
ViewItems.getAllItems	✓	33
ViewCart.sendItems	⚠	756
UserLogout.memberLogout	✓	10
UserLogin.memberLogin	✓	16

In the Events panel on the dashboards, health violations are displayed as events.

Events

Health Rule Violations Started	2	⚠
Business Transaction Health	3	✗
Node Health	1	⚠
Error Rates	1	⚠
Code Problems	74	✗
Application Changes	4	!

To see a summary of the violation, click a health rule violation from the Events list, then select the violation you are interested in from the list that appears. Click **View Event Details** in the top bar and the Health Rule Violation Started window appears. It displays detailed information and a link to the appropriate dashboard at the time of the violation. If any Policy actions were executed in response to the violation, they are also displayed.

Health Rule Violation Started - Critical

Summary
Actions Executed (0)

Event Type
✗ Health Rule Violation Started - Critical

Health Rule
⚡ Slow Requests

Health Rule Type
Business Transaction Performance (load, response time, slow calls, etc)

Affects
🖨 Supplier Search

View Dashboard During Health Rule Violation

Violation State
Open

Duration
Ongoing (6 minutes)
From: 05/06/13 4:48:55 PM To: -

No Actions Executed
What does this mean?

Description

Health rule violation status changed from None to Open (Critical level) for health rule 'Slow Requests' of type Business Transaction Performance.

All of the following conditions were evaluated for each node on E-Commerce for business transaction performance Supplier Search and 1 nodes were found to be violating thresholds.

For Node E-Commerce-Node-8004:

Learn More

- [Health Rules](#)
- [Configure Health Rules](#)

- [Organizing Traffic as Business Transactions](#)
- [Policies](#)
- [Alert and Respond](#)
- [Transaction Snapshots](#)

Actions

- [Types of Actions](#)
- [Actions Limits](#)
- [Actions Requiring Approval](#)
- [Viewing and Creating Actions](#)
 - [To view and edit existing actions](#)
 - [To create an action](#)
- [Action Suppression](#)
- [Learn More](#)

An action is a predefined, reusable, automated response to an event. You can use actions to automate your runbooks.

A policy can trigger an action in response to any event. You configure which actions are triggered by which events when you configure policies. See [Policies](#).

Types of Actions

You can create the following types of actions:

- [Notifications](#)
- [Diagnostics](#)
- [Remediation](#)
- [Custom Actions](#)
- [Cloud Auto-Scaling](#)

Not all actions are applicable to all application environments or to all situations. Below are some general guidelines concerning different types of actions. For more details, see the pages on the specific actions before you assign an action to a policy.

- The diagnostic thread dump actions can be performed only on nodes running a Java agent.
- The diagnostic session actions can be triggered only by violations of business transaction performance health rules or slow or stalled transaction events, since these are the events that produce a view into transaction snapshots.
- Remediation actions run a local script in a node and are available on nodes running on machines that have an installed machine agent. See [Install the Standalone Machine Agent](#).
- Custom Actions require a dedicated controller, deployed using either the on-premise or SaaS option. This feature is not supported for accounts on multi-tenant SaaS controllers.
- Cloud Auto-Scaling actions require a previously created workflow. See [Workflow Overview](#).

Actions Limits

The Controller limits the actions invoked based on the number of triggering events per event type. There is a maximum of ten events for any single event type that can trigger actions in a given

minute. If the number of triggering events per type exceeds the limit, the actions that would have been triggered by the excess events are not started. You will not see a visual indication that these actions are not being started.

For example, your application can have up to ten Health Violation Started events triggering actions and up to ten Resource Pool Limit Reached events triggering actions within the same minute. But if you have eleven Health Violation Started events firing, the action that would be triggered by the eleventh event is not started.

To reduce unnecessary actions, there is a limit on the number of diagnostic and remediation actions that AppDynamics will invoke. The default limit is five actions per minute per machine for each type of action.

If, for example, a policy is configured on all the nodes where there are 100 nodes triggering actions, AppDynamics randomly selects five of the actions to execute.

To avoid exceeding the limits, design your policies so that they do not trigger an excessive number of actions for any particular event. You can generate fewer events by configuring the affected entities of your health rules at the tier level. See [Entities Affected by a Health Rule](#).

Actions Requiring Approval

For actions that take thread dumps or run a local script, you can optionally require email approval to run the action whenever it is triggered. If you configure this option, human intervention is required before the "automated" action actually starts.

Create Thread Dump

Thread Dumps will be taken by the Java App Server Agent, and uploaded to the controller when completed. You can download them on the Events screen (for events that trigger Policies).

Name: Adjudicated Thread Dump

Number of thread dumps: 2 (Maximum 50)

Thread dump session duration in ms: 500 (Maximum 500 ms)

Require approval before executing this Action: ☒

E-mail address for approver: javaGuy@ourcompany.com

[Configure Email / SMS settings](#)

Cancel OK

If you specify the approval required option when you configure the action, when the action is triggered an email containing a link is sent to the configured email address. The link presents a login screen (if the user is not already logged in to AppDynamics) and after the user logs in, a dialog requesting approval to take the thread dump or run the script. The user can click in this dialog to approve and start the action or cancel the action.

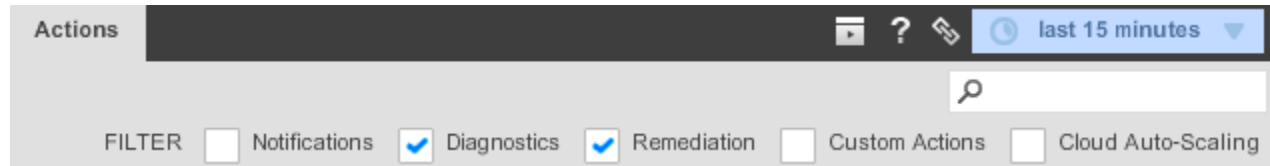
If you do not check the Require approval option before executing the Action check box, the action will start automatically with no human intervention.

Viewing and Creating Actions

To view and edit existing actions

1. Click **Alert & Respond -> Actions** in the left navigation pane.
The list of actions in the application appears.

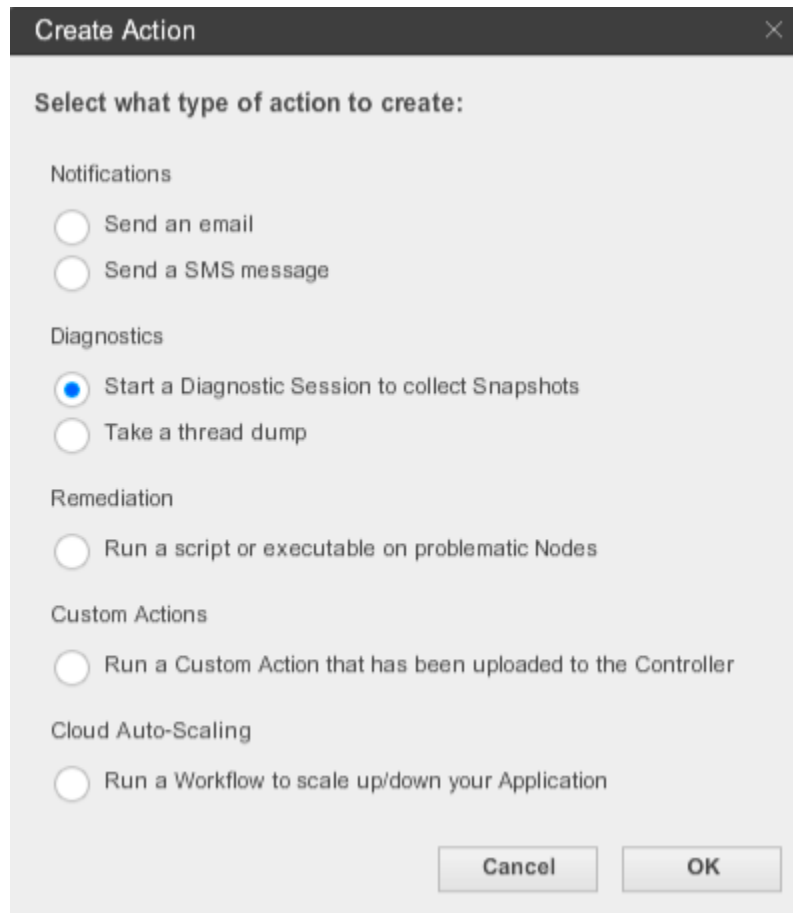
To filter the types of actions displayed in the list, select the type at the top of the list. For example, to see only diagnostic and remediation actions, check Diagnostics and Remediation and clear the other check boxes.



2. To examine or modify an action, select the action in the list and double-click or click **Edit**.
3. To delete an action, select the action in the list and click **Delete**.

To create an action

1. Click **Alert & Respond -> Actions** in the left navigation pane.
2. Click the **Create Action** button.



3. Select the type of action that you want to create.
4. Click **OK**.

The instructions beyond this point vary depending on the type of action you are creating. See the topics for the action type you have selected.

Action Suppression

You can prevent policies from firing actions for a configurable time period. See [Action Suppression](#).

Learn More

- [Policies](#)
- [Notification Actions](#)
- [Diagnostic Actions](#)
- [Remediation Actions](#)
- [Cloud Auto-Scaling Actions](#)
- [Custom Actions](#)
- [Action Suppression](#)
- [Install the Standalone Machine Agent](#)

Notification Actions


- [Email Notifications](#)
 - [To create an email notification](#)
- [SMS Notifications](#)
 - [To create an SMS notification](#)
- [Learn More](#)

A notification action sends an email or SMS to a recipient list. The text of the notification is automatically generated by the event that triggered the action.

Note that if you are using a SaaS Controller, all notification timestamps are in Pacific time (PTZ).

Email Notifications


An email notification contains a deep link to the details of the event that triggered it. Clicking this deep link takes you directly to the place to start troubleshooting the problem.


AppDynamics Notification
ACME Book Store Application

My Policy

Summary of events occurring during the 1+ minute(s) prior to Wed Oct 16 17:40:55 EDT 2013:

Count	Event Type
1	Health Rule

 [New Warning Health Rule Violation](#)

Wed Oct 16 17:39:55 EDT 2013

Appdynamics has detected a problem with Business Transaction **ViewCart.sendItems**. **Business Transaction error rate is much higher than normal** started violating and is now **warning**. All of the following conditions were found to be violating

For Business Transaction **ViewCart.sendItems**:

- 1) Errors per Minute Baseline Condition
Errors per Minute's value 1.0 was **greater than** baseline-based calculated value by 2 standard deviation(s). Baseline used here is **'Daily Trend - Last 30 days'** for the last 30 minutes
- 2) Errors per Minute Condition
Errors per Minute's value 18.0 was **greater than** the threshold 5 for the last 30 minutes
- 3) Calls per Minute Condition
Calls per Minute's value 71.0 was **greater than** the threshold 50 for the last 30 minutes


Notes:
Policy: My Policy

To create an email notification

1. Follow the instructions in [To create an action](#), selecting **Notifications->Send an email** in the Create Action window.
2. Enter the email address to which to send the notification.
3. Click **Save**.

When you configure a policy to fire an email notification action, you have an opportunity to add a note to the email. This note is applied only when the action is invoked by the particular policy. By adding an optional note, you can customize email notifications for the policies that invoke them.

Configure Action

Action  **lld@appd.com**

Notes to include in emails:

Cancel
Save

If email and SMS settings have not been configured for AppDynamics, configure them now. See [Configure the SMTP Server](#).

SMS Notifications

The content of the SMS is automatically generated. It contains:

- the notification header
- the application name
- the triggered time

Notifications of health rule violations also include:

- name of health rule violated

Event notifications also include:

- event notification configuration name
- map of event types to number of these events

An SMS notification configuration specifies the phone number of the recipient.

To create an SMS notification

1. Follow the instructions in [To create an action](#), selecting **Notifications->Send an SMS message** in the Create Action window.
2. Enter the phone number to which to send the notification.
3. Click **Save**.

If email and SMS settings have not been configured for AppDynamics, configure them now. See [Configure the SMTP Server](#).

Learn More

- [Actions](#)
- [Email Digests](#)
- [Policies](#)
- [Health Rules](#)

Diagnostic Actions

- [Diagnostic Action Results](#)
 - [To get the details of a diagnostic session or a thread dump that has been initiated by an action](#)
- [Diagnostic Session Actions](#)
 - [To create a diagnostic session action](#)
- [Thread Dump Actions \(Java only\)](#)
 - [Agent Limit on Thread Dumps](#)
 - [To create a thread dump action](#)
- [Learn More](#)

A diagnostic action can:

- start a diagnostic session to collect snapshots
- take a thread dump (Java only)

When performance is slow or your application is experiencing a lot of errors you can start a diagnostic action to get to the root cause.

A diagnostic session gives you a view into captured transaction snapshots with full call graphs. These snapshots help you diagnose violations of business transaction performance health rules or

slow or stalled transaction events. The affected entity of the event triggering a diagnostic session must be a business transaction.

A thread dump is a general-purpose snapshot of the state of all threads that are part of a JVM process. The state of each thread is presented with a stack trace that shows the contents of each thread's stack. Thread dumps are used for diagnosing JVM performance problems, such as code deadlocks.

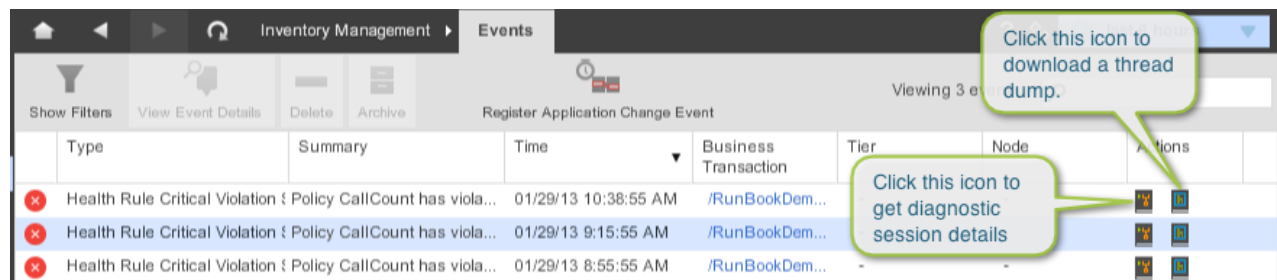
Thread dumps are not supported for .NET or PHP agents.

Diagnostic Action Results

The results of a diagnostic action that has executed are available in the events list for the event that triggered the action.

To get the details of a diagnostic session or a thread dump that has been initiated by an action

1. Click **Events** in the left navigation pane.
2. Locate the row for the event that triggered the action for which you want to see the results.
3. In the Actions Executed column, click the diagnostic sessions or thread dump icon for the event that you want to troubleshoot.



On disk, the thread dumps are stored in the `app_agent_operation_logs` directory in the controller installation folder. The files are named based on the id in the `app_agent_operation` table.

Diagnostic Session Actions

A diagnostic session is always associated with a business transaction. It shows transaction snapshots with full call graphs to help you drill down to the root cause of a problem.

To create a diagnostic session action

1. Follow the instructions in [To create an action](#), selecting **Diagnostics->Start a diagnostic sessions to collect snapshots** in the Create Action window.
 2. Enter a name for the action, the duration of the diagnostic session in minutes, and the number of snapshots to take per minute.
 3. Select whether a diagnostic session will be started for any business transaction affected by an event or specific business transactions.
- If you choose specific business transactions, specify the business transactions that will trigger the diagnostic session by moving them from the "available" list to the "selected" list. The business transactions that you can specify are not limited to those that triggered the action.

4. Click **OK**.

Thread Dump Actions (Java only)

You can direct the agent to take a thread dump for a specified number of samples (maximum of 50) with each sample lasting for a specified number of milliseconds (maximum of 500 ms). The thread dump is executed on the node.

Thread dump actions are not supported on .NET or PHP.

Agent Limit on Thread Dumps

One thread dump operation is executed at a time. They are not executed in parallel. If additional thread dump requests are received while one is being executed, they are queued with a limit of five per agent.

If the five thread dumps per agent limit is exceeded, the console shows an event with a thread dump operation that was skipped because of the limit and the associated action dialog for the executed policy links to this event.

To create a thread dump action

1. Follow the instructions in [To create an action](#), selecting **Diagnostics->Take a thread dump** in the Create Action window.
2. Enter a name for the action, the number of samples to take, and the interval between the thread dumps in milliseconds.
3. If you want to require approval before the thread dump action can be started, check the Require approval before this Action check box and enter the email address of the individual or group that is authorized to approve the action. See [Actions Requiring Approval](#) for more information.
4. Click **OK**.

Create Thread Dump

Thread Dumps will be taken by the Java App Server Agent, and uploaded to the controller when completed. You can download them on the Events screen (for events that trigger Policies).

Name

Number of thread dumps Maximum 50 ?

Interval in ms Maximum 500 ms ?

Require approval before executing this Action ☐ ?

E-mail address for approver

[Configure Email / SMS settings](#)

Cancel OK

Learn More

- [Actions](#)
- [Call Graphs](#)
- [Diagnostic Sessions](#)
- [Policies](#)
- [Transaction Snapshots](#)
- [Install the Standalone Machine Agent](#)

Cloud Auto-Scaling Actions

- [To Create a Cloud Auto-Scaling Action](#)
- [Learn More](#)

A cloud auto-scaling action allows you to move instances of your application into a cloud provider, automatically, in response to load or any other criteria. To use this action, you must first set up your cloud provider and create workflows to manage the steps. See [Automation](#), [Cloud Computing Workflows](#), and [Workflow Overview](#) for more information on preparing this action.

To Create a Cloud Auto-Scaling Action

1. Follow the instructions in [To create an action](#), selecting **Cloud Auto-Scaling->Run a workflow to scale up/down your application** in the Create Action window.
2. Give the workflow action a Name.
3. Select the name of the previously created Workflow from the dropdown list.
4. Click **OK**.

Learn More

- [Actions](#)
- [Automation](#)
- [Cloud Computing Workflows](#)
- [Workflow Overview](#)

- [Create a Workflow and Workflow Steps](#)
- [Policies](#)

Custom Actions

- [To Create a Custom Action](#)
- [Learn More](#)

A custom action is typically used to integrate third party alerting and ticketing systems with an on-premise controller. Custom action scripts are not supported on SaaS controller. A custom action is different from other actions in that it executes just once on the on-premise controller instance.

The custom action is made up of a custom action script and a custom.xml file, which you must create before you can create an action that uses them. The custom action scripts include parameters for specifying the affected entity, for example the tier, node, business transaction, etc. See [Build an Alerting Extension](#) for details on how to create the custom action script and xml file.

Custom actions are commonly used when you want to trigger a human work flow or leverage an existing alerting system that is external to AppDynamics. For example, you could use a custom action to file a JIRA ticket when AppDynamics reports that a connection pool is near saturation.

If you are using a SAAS controller, you may be able to use a remediation action on a local node to accomplish similar results. See [Remediation Actions](#) for more information.

To Create a Custom Action

After the custom action script and custom.xml files have been tested manually and installed on the Controller and you have restarted the Controller, you can create the custom action.

1. Follow the instructions in [To create an action](#), selecting **Automation->Run any custom action that has been uploaded to the controller** in the Create Action window.
2. Enter a name for the action.
3. Select the custom action from the dropdown list.
4. Click **OK**.

The custom action is now available for assignment to a policy.

Learn More

- [Build an Alerting Extension](#)

Remediation Actions

- [Prerequisites for Local Script Actions](#)
- [Remediation Scripts](#)
 - [Guidelines for Remediation Scripts](#)
 - [Troubleshooting Remediation Scripts](#)
- [Remediation Example](#)
- [Creating a Local Script \(Remediation\) Action](#)
 - [To create a local Script Action](#)
 - [To specify the nodes on which the action will run](#)
 - [To see the output of the local script](#)
- [Learn More](#)

A remediation action runs a local script in a node. The script executes on the machine from which it was invoked or on the node specified by the remediation action configuration. You can use this type of action to automate your runbook procedures.

By default the script is a shell script in `/bin/sh` invoked with the `-ex` option, unless the script has a header, in which case the interpreter in the header is used. For example, if the script header is `#!/bin/perl`, the PERL interpreter is invoked.

You can optionally configure the remediation action to require human approval before the script is started. See [Actions Requiring Approval](#).

Remediation actions can be performed only on machines that are running the machine agent. See [Install the Standalone Machine Agent](#).

Prerequisites for Local Script Actions

- The Standalone (Java-based) Machine Agent must be installed running on the host on which the script executes.
To see a list of installed machine agents for your application, click **View machines with machine-agent installed** in the bottom left corner of the remediation script configuration window.
See [Install the Standalone Machine Agent](#) if you need to install a machine agent.
- The machine agent OS user must have full permissions to the script file and the log files generated by the script and/or its associated child processes.
- The script must be placed in a sub-directory of the machine agent installation directory that is named "local-scripts".
- The script must be available on the host on which it executes.
- Processes spawned from the scripts must be daemon processes

Remediation Scripts

A remediation script is run on the machines that you specify in the remediation script configuration. You can run the script from the machine affected by the violation that triggered the action or from a central management server. It is not necessary for an app agent to be running on the machine on which the script executes, just a machine agent.

Guidelines for Remediation Scripts

A process exit code of zero indicates that the script execution succeeded. A non-zero exit code indicates that it failed.

The script should be written as generically as possible to allow it run on any of the nodes for which it is invoked. AppDynamics exports the following environment variables to the script runtime to provide context regarding the environment and the event that triggered the action.

Environment Variable	Cardinality (1 or M)	Notes
APP_ID	1	Name of the Application
EVENT_TIME	1	Timestamp of the event

EVENT_ID	1	Event Id
EVENT_TYPE	1	type of event, such as: ERROR, APPLICATION_ERROR, APPLICATION_INFO, STALL, BT_SLA_VIOLATION, DEADLOCK, MEMORY_LEAK, MEMORY_LEAK_DIAGNOSTICS, LOW_HEAP_MEMORY, ALERT, CUSTOM, APP_SERVER_RESTART, BT_SLOW, SYSTEM_LOG, INFO_INSTRUMENTATION_VISIBILITY, AGENT_EVENT, INFO_BT_SNAPSHOT, AGENT_STATUS, SERIES_SLOW, SERIES_ERROR, ACTIVITY_TRACE, OBJECT_CONTENT_SUMMARY, DIAGNOSTIC_SESSION, HIGH_END_TO_END_LATENCY, APPLICATION_CONFIG_CHANGE, APPLICATION_DEPLOYMENT, AGENT_DIAGNOSTICS, MEMORY, LICENSE
ENV_STARTUP_ARGS	1	Process args
ENV_SYSTEM_PROPERTIES	1	JVM System Props (When Java)
AFFECTED_ENTITY	1	Affected Entity that triggered the event

Troubleshooting Remediation Scripts

To troubleshoot your remediation script, look for the process in the machine agent log. The log is located at

<Machine_Agent_Installation_Directory>/logs/machine-agent.log

The snippet below from the machine agent log shows both error and success messages from running a local script named "script.sh".

```
[Agent-Scheduler-1] 07 May 2013 18:20:24,580 ERROR RunLocalScriptEventHandler - Script is not in correct directory. while executing run local script operation
[Agent-Scheduler-1] 07 May 2013 18:20:24,580 INFO RunLocalScriptRequestHandler - Received run local script request: opId=27,
actionGuid=f117d181-a3a0-407e-b0ac-0e3878547f2f
[Agent-Scheduler-1] 07 May 2013 18:20:24,581 ERROR ScriptExecutor - Script '/Users/akilman/script.sh' must reside in
'/Users/akilman/Work/cart-tmp/machineagent/local-scripts'
[Agent-Scheduler-1] 07 May 2013 18:20:24,593 ERROR RunLocalScriptEventHandler - Error occurred while executing run local script operation
[Agent-Scheduler-1] 07 May 2013 18:20:24,594 INFO RunLocalScriptRequestHandler - Received run local script request: opId=28,
actionGuid=45202a5b-af43-4f2a-9308-1bce7f2408b2
[Agent-Scheduler-1] 07 May 2013 18:20:24,594 ERROR ScriptExecutor - Script '/Users/akilman/script.sh' must reside in
'/Users/akilman/Work/cart-tmp/machineagent/local-scripts'
[Agent-Scheduler-1] 07 May 2013 18:20:24,606 ERROR RunLocalScriptEventHandler - Error occurred while executing run local script operation
[Agent-Scheduler-1] 07 May 2013 18:26:24,689 INFO RunLocalScriptRequestHandler - Received run local script request: opId=29,
actionGuid=b5b80d3e-7859-400f-927c-d42c1b495d0c
[Agent-Scheduler-1] 07 May 2013 18:26:24,693 INFO ScriptExecutor - Executing: [/Users/akilman/Work/cart-tmp/machineagent/local-
scripts/script.sh]
[Agent-Scheduler-1] 07 May 2013 18:26:24,693 INFO ScriptExecutor - Using working directory: /Users/akilman/Work/cart-tmp/machineagent
[Agent-Scheduler-1] 07 May 2013 18:26:26,582 INFO RunLocalScriptEventHandler - Run local script request completed successfully
[Agent-Scheduler-1] 07 May 2013 18:26:26,582 INFO RunLocalScriptRequestHandler - Received run local script request: opId=30,
actionGuid=29169ecf-fdf4-4a59-9a95-987d992a7610
[Agent-Scheduler-1] 07 May 2013 18:26:26,584 INFO ScriptExecutor - Executing: [/Users/akilman/Work/cart-tmp/machineagent/local-
scripts/script.sh]
[Agent-Scheduler-1] 07 May 2013 18:26:26,584 INFO ScriptExecutor - Using working directory: /Users/akilman/Work/cart-tmp/machineagent
[Agent-Scheduler-1] 07 May 2013 18:26:28,248 INFO RunLocalScriptEventHandler - Run local script request completed successfully
[Agent-Scheduler-1] 07 May 2013 18:26:28,249 INFO RunLocalScriptRequestHandler - Received run local script request: opId=31,
actionGuid=8c2e6530-184a-40ed-b672-1cf28aa7120d
[Agent-Scheduler-1] 07 May 2013 18:26:28,250 INFO ScriptExecutor - Executing: [/Users/akilman/Work/cart-tmp/machineagent/local-
scripts/script.sh]
[Agent-Scheduler-1] 07 May 2013 18:26:28,250 INFO ScriptExecutor - Using working directory: /Users/akilman/Work/cart-tmp/machineagent
[Agent-Scheduler-1] 07 May 2013 18:26:29,913 INFO RunLocalScriptEventHandler - Run local script request completed successfully
```

Remediation Example

The following remediation action, named `increasePool`, executes a local script named `runbook.sh`, which increases the size of the connection pool on the JVM.

Create Remediation Script Action

You can specify any script or executable and the Machine Agent will execute it, and upload the results to the controller. You can download the script output on the Events screen (for events that trigger Policies).

Name

IncreasePool

Relative path to script

\$(machine.agent.directory)/local-scripts/ runBook.sh

Absolute paths to log files

+

-

?

Path

/tmp/script.out

Script timeout in minutes

2

Require approval before executing this Action

☐

E-mail address for approver

[Configure Email / SMS settings](#)

[View machines with machine-agent installed](#)

Cancel

OK

A policy named `ConnectionPoolPolicy` triggers this action when the Resource Pool Limit Event

fires:

Edit Policy - ConnectionPoolPolicy

Name: Enabled: ☒

TRIGGER This Policy will fire when **any of these Events occur** on **any object**

ACTIONS

Health Rule Violation Events

- ☐ Health Rule Violation Started - Warning
- ☐ Health Rule Violation Started - Critical
- ☐ Health Rule Violation Upgraded - Warning to Critical
- ☐ Health Rule Violation Downgraded - Critical to Warning
- ☐ Health Rule Violation Ended

Other Events

- ☐ Slow Transactions
- ☒ Code Problems
 - ☐ Code Deadlock
 - ☒ Resource Pool Limit Reached

Creating a Local Script (Remediation) Action

To create a local Script Action

1. Follow the instructions in [To create an action](#), selecting **Remediation->Run a script or executable on problematic Nodes** in the Create Action window.
2. Enter a name for the action.
3. In the field that terminates the Relative path to script entry, enter the rest of the path to the executable script.

Remediation scripts must be stored in a sub-directory of the machine agent installation. The sub-directory must be named "local-scripts". The following paths are all valid:

```

${machine.agent.directory}/local-scripts/runMe.sh
${machine.agent.directory}/local-scripts/johns_scripts/runMe.sh
${machine.agent.directory}/local-scripts/ops/johns_scripts/runMe.sh

```

4. Click the **+** to enter the absolute paths of any log files that the script writes to that you want included in the script output.
5. Enter the timeout period for the script process in minutes.
6. If you want to require approval before the script action can be started, check the Require approval before this Action check box and enter the email address of the individual or group that is authorized to approve the action. See [Actions Requiring Approval](#) for more information.
7. Click **OK**.

To specify the nodes on which the action will run

When you bind the action to a policy, you specify the nodes on which the script should execute. You can configure the number of nodes or the percentage of nodes or you can configure a specific node. This flexibility allows you to configure scripts to run from a central management server, not just the node on which the violation occurred.

In the Configure Action window, do one of the following:

1. Select **Execute Action on Affected Nodes**.
2. Enter the percentage of the nodes or the number of nodes on which to run the script.

or

1. To designate the specific node on which to run the script, select **Execute Action on Specified Node**.

2. Click **Select Node**.

3. From the popup node browser select the node on which the script should run.

4. Click **Select**.

The selected node is displayed in the Configure Action window.

5. Click **Save** to save the configuration.

Click **Change** if you want to designate a different node.

Configure Action

Action increasePool

☐ Execute Action on Affected Nodes

☒ Execute Action on % of the Nodes

☐ Execute Action on Nodes

☒ Execute Action on Specified Node:

Node_8001 **Change**

Cancel **Save**

To see the output of the local script

1. Click **Events** in the left navigation pane to navigate to the Events list.

2. Locate the row for the event that triggered the action for which you want to see the results.

3. In the Actions column, click the remediation script icon.

Type	Summary	Time	Business Transaction	Tier	Node	Actions
	Health Rule Violat Policy TooSlow has violated	01/31/13 2:00	/RunB...	-	Admin...	-
	Health Rule Violat Policy TooSlow has violated	01/31/13 12:5	/RunB...	-	Admin...	-
	Health Rule Violat Policy TooSlow has violated	01/31/13 11:3	/RunB...	-	Admin...	-
	Health Rule Violat Policy TooSlow has violated	01/31/13 9:13	/RunB...	-	Admin...	-
	Health Rule Violat Policy TooSlow has violated	01/31/13 9:07	/RunB...	-	Admin...	

2 Remediation Script(s) were executed.

4. In the script result list, select the script output that you want and click **Download Local Script Result**.

Select Local Script Result	
Download Local Script Result	
Time	Summary
01/31/13 9:09:55 AM	Run local script request completed successfully
01/31/13 9:09:48 AM	Run local script request completed successfully
01/31/13 9:09:55 AM	Run local script request completed successfully
01/31/13 9:09:48 AM	Run local script request completed successfully

Learn More

- [Actions](#)
- [Policies](#)
- [Install the Standalone Machine Agent](#)

Action Suppression

- [Create a New Action Suppression](#)
- [Objects Affected by Action Suppression](#)
 - [Application-Level Action Suppression](#)
 - [Business Transaction Action Suppression](#)
 - [Tier-Level Action Suppression](#)
 - [Node-Level Action Suppression](#)
 - [Machine-Level Action Suppression](#)
- [Health Rules Affected by Action Suppression](#)
- [Cancelling Action Suppression](#)
 - [To delete/cancel an action suppression configuration](#)
- [Learn More](#)

You can temporarily suppress a policy's automatic invocation of actions and alerts. You may want to do this while you are performing maintenance on or troubleshooting a component.

To see action suppression configurations created for an application:

1. Click **Alert & Respond -> Actions**.
2. Click the Action Suppression tab.

The list of action suppression configurations displays in the left panel with the objects affected by a selected configuration displayed in the right panel.

Actions					
Action Suppression					
Action Suppression Configurations			MyActSuppression		
			Object Scope		
Name	Start Time	End Time	Type	Name	
MyActSuppression	02/28/14 1:39:57 PM	02/28/14 2:39:57 PM	Tier	ECommerce Server	

Create a New Action Suppression

1. Click the plus icon. The Create Action Suppression popup appears.

Create Action Suppression

OVERVIEW

OBJECT SCOPE

HEALTH RULE SCOPE

Overview

Name

Scope

Disable reporting for associated Agents

Start Time

End Time

Application

Business Transactions

Tiers and Nodes

Machines

Now

At Date / Time

After 60 minutes

At Date / Time

AppDynamics can be configured to stop executing Actions for a time period.

For example, while a Node is under maintenance, you can suppress notifications about Health Rule violations on that Node.

2. In the Overview section, give the Action Suppression a name, define its scope, and set the times it should cover. See [Objects Affected by Action Suppression](#) for more information on scope.

Create Action Suppression

OVERVIEW

OBJECT SCOPE

HEALTH RULE SCOPE

Select Tiers or Nodes

☒ Tiers

☐ Nodes

Select Tiers

All Tiers in the Application

All Tiers in the Application

These specific Tiers

3. In the Object Scope section, drill down to the set of entities that it should cover.

Create Action Suppression

OVERVIEW

OBJECT SCOPE

HEALTH RULE SCOPE

Health Rule Scope (Optional) ?

☒ Only suppress Action execution for these specified Health Rules:

+ Select Health Rule(s)

Select Health Rule(s)

	Name	Type	Enabled
<input checked="" type="checkbox"/>	Business Transaction response time is much higher than normal	Business Transaction Performance	✓
<input checked="" type="checkbox"/>	Business Transaction error rate is much higher than normal	Business Transaction Performance	✓
<input checked="" type="checkbox"/>	CPU utilization is too high	Node Health - Hardware, JVM, CLR	✓
<input checked="" type="checkbox"/>	Memory utilization is too high	Node Health - Hardware, JVM, CLR	✓
<input checked="" type="checkbox"/>	JVM Heap utilization is too high	Node Health - Hardware, JVM, CLR	✓
<input checked="" type="checkbox"/>	JVM Garbage Collection Time is too high	Node Health - Hardware, JVM, CLR	✓
<input checked="" type="checkbox"/>	CLR Garbage Collection Time is too high	Node Health - Hardware, JVM, CLR	✓

Create Health Rule

Cancel

Select Health Rule(s)

4. If you wish to limit the particular health rules that this action suppression should effect, select Only suppress Action for these specified Health Rules. A list of possible rules pops up. Select the appropriate rules. See [Health Rules Affected by Action Suppression](#) for more information.

Objects Affected by Action Suppression

You configure action suppression for a specific time period to apply to a specific object or several objects. The following entities can be the objects of action suppression:

- Application
- Business Transaction

- Tier
- Node
- Machine

Within the time period configured for the action suppression, no policy actions are fired for health rule violation events that occur on the specified object(s).

You can also optionally disable reporting of metrics for an object for which actions are suppressed. Using this option can cause reported metrics for those objects to change without notice. If you see a sudden unexpected change in reported metrics for an object, check the action suppression configurations list to see whether action suppression with reporting disabled is currently active for that object.

If the object scope of an action suppression is at the node level, the suppression affects only node health rules.

If the object scope of an action suppression is at the tier level, the suppression affects individual node health rules as well as tier-level health rules.

For example, if a tier-level health rule is configured to fire an action when a percentage of the nodes violates the condition, and then action suppression is configured on certain nodes in that tier, those nodes are still evaluated by the tier-level health rule.

Application-Level Action Suppression

In an application-level configuration, all entities in the application are affected.

Business Transaction Action Suppression

In a business-transaction-level configuration, you can suppress actions in:

- All business transactions in the application
- All business transactions within specific tiers
- Specific business transactions
- Business transactions with names having patterns that match certain criteria (such as all business transactions with names that start with "XYZ")

Tier-Level Action Suppression

In a tier-level configuration, all the nodes in the specified tier(s) are affected. You can suppress actions for:

- All tiers in the application
- Specific tiers

Node-Level Action Suppression

In a node-level configuration, you can specify the types of nodes for which to suppress actions:

- All nodes
- Java nodes
- .NET nodes
- PHP nodes

and within those types you can suppress actions for:

- All nodes
- Nodes in specific tiers
- Specific nodes
- Nodes with names, meta-data, environment variables or JVM system environment properties with matching criteria that you specify

Machine-Level Action Suppression

You can suppress actions run on specific machines. Actions run on all the nodes on the specified machine (s) are suppressed.

Health Rules Affected by Action Suppression

By default, an action suppression configuration applies to actions triggered by all events that are generated by the configured objects.

You can refine the configuration to apply only to actions triggered by specific health rule violations. For example, if an application contains HealthRuleA, HealthRuleB and HealthRuleC, but only HealthRuleC is configured for action suppression, actions will continue to fire for violations of HealthRuleA and HealthRuleB during the configured time period.

Cancelling Action Suppression

If action suppression is no longer needed (for example, where the estimated time to fix a problem was longer than the fix actually required) you can delete it. Policies on the objects affected by the action suppression configuration will start firing a few minutes after the suppression configuration is cancelled.

To delete/cancel an action suppression configuration

1. From the action suppression configurations list, select the configuration to delete.
2. Click the Delete icon.

Learn More

- [Policies](#)
- [Health Rules](#)
- [Actions](#)
- [Configure Action Suppression](#)

Configure Action Suppression

- [To access action suppression configuration](#)
- [Structure of the Action Suppression Wizard](#)
- [Configuring General Action Suppression Settings](#)
- [To configure general action suppression settings](#)
- [Configuring Object Scope for Action Suppression](#)
- [To configure object scope](#)
- [Configuring Health Rules for Action Suppression](#)

- To suppress actions for all health rule violations
- To suppress actions triggered by only certain health rules

[Learn More](#)

Configure action suppression using the Action Suppression Wizard.

To access action suppression configuration

1. Click **Alert & Respond -> Actions**.
2. Click the Action Suppression tab.
3. To edit an existing action suppression configuration, select the configuration in the list and click the Edit icon.
4. To delete an existing action suppression configuration, select the configuration in the list and click the Delete icon.
5. To create a new action suppression configuration click the Add icon.

The Action Suppression Wizard appears when you edit or add a configuration.

Structure of the Action Suppression Wizard

The Action Suppression Wizard contains three panels:

1. **Overview:** Sets:

- configuration name
- scope
- option to suppress metrics reporting by agents associated with the affected objects
- schedule

2. **Object Scope:** Sets the objects affected by the configuration.

The options presented vary according to the scope set in the Overview panel.

3. **Health Rule Scope:** Sets the events that trigger action suppression.

By default, all events on the affected objects trigger action suppression. You can restrict action suppression to apply only to violations of specific health rules

You can navigate among these panels using the **Back** and **Next** buttons at the bottom of each panel or by clicking their entries in the left panel of the wizard. When you create a new configuration, configure the panels in order because the configuration of the scope in the Overview panel determines the available affected objects presented in the object scope panel.

Configuring General Action Suppression Settings

Configure general settings in the Overview panel.

To configure general action suppression settings

1. In the Overview panel of the Action Suppression Wizard:
 - If you are creating a new configuration, enter the configuration name in the Name field.
 - If you are editing an existing configuration, the name will already be there.
2. Click one of the object types in the Scope list to select the type of entity affected by the configuration.

If you choose Application, policy actions are suppressed for all entities in the application.

3. If you want to suppress metric collection and reporting while actions are being suppressed, check the **Disable reporting for associated agents** checkbox.

If you want the agent to continue to report metrics while actions are being suppressed, this checkbox should be clear.

4. Enter the time period that action suppression is in effect.

- **Start Time:** Choose **Now**, or configure a starting Date/Time in the future.
Now is the time at which the configuration is enabled.
- **End Time:** Configure **After** and the number of hours (as an integer) that the configuration is in effect after the start time, or configure an ending Date/Time in the future.

5. Click **Next**.

Configuring Object Scope for Action Suppression

In the Object Scope panel of the Action Suppression Wizard you can define the object scope of the configuration broadly or fine-tune it very precisely to suppress actions for specific objects in your application.

To configure object scope

1. In the Object Scope panel, select the objects affected by this configuration.

The choices presented in this panel depend on the scope configured in the Overview panel. See [Objects Affected by Action Suppression](#) for information about the types of objects that you can select.

If the object scope is Application, there is no configuration to be done in this panel, so click **Next** and skip to [Configuring Health Rules for Action Suppression](#).

2. For types other than Application, configure the object scope using the tools in the panel. The display varies depending on the type of object.

If your configuration includes selecting specific objects, move those objects from the "other" list to the "selected" list.

Select Tiers

These specific Tiers ▼

Selected Tiers (1)

Name	Type
ECommerce Server	Application Server

< ADD

REMOVE >

Tip: You can drag items between these lists

Other Tiers (2)

Name	Type
Inventory Server	Application Server
Order Processing Server	Application Server

Refresh List

You can either drag and drop the objects from one list to the other or select them and click **Add** or

Remove as appropriate.

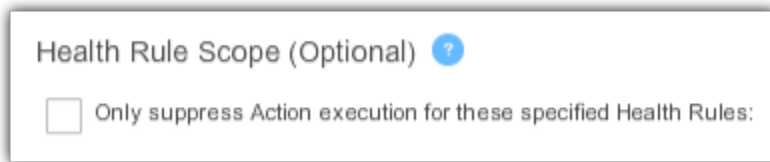
3. Click **Next**.

Configuring Health Rules for Action Suppression

In the Health Rules panel of the Action Suppression Wizard you can specify the health rules affected by the configuration.

To suppress actions for all health rule violations

1. Leave the **Only suppress Action execution for these specified health rules** checkbox clear.



Health Rule Scope (Optional) ?

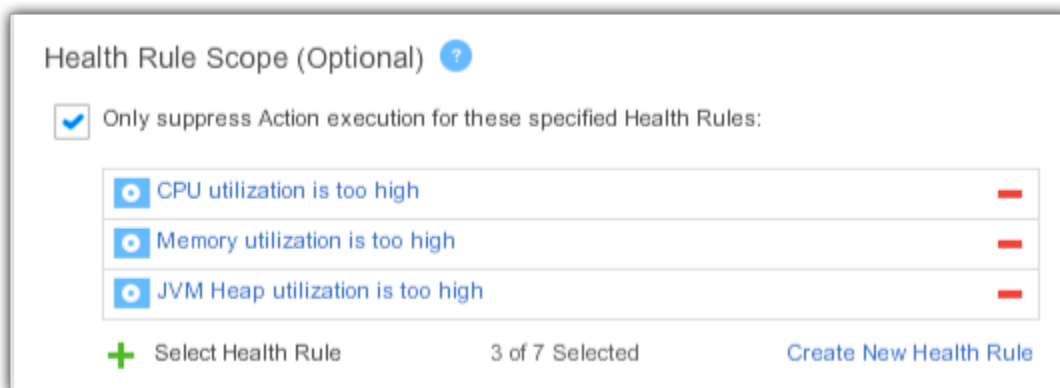
☐ Only suppress Action execution for these specified Health Rules:

2. Click **Save** to save the configuration.

To suppress actions triggered by only certain health rules

1. Check the **Only suppress Action execution for these specified health rules** checkbox.

2. For each health rule to be affected by the configuration, click **+** to select the health rule from a list.



Health Rule Scope (Optional) ?

☒ Only suppress Action execution for these specified Health Rules:

<input checked="" type="radio"/> CPU utilization is too high	—
<input checked="" type="radio"/> Memory utilization is too high	—
<input checked="" type="radio"/> JVM Heap utilization is too high	—

+ Select Health Rule 3 of 7 Selected [Create New Health Rule](#)

To delete a health rule from the suppression list select it and click the Delete icon.

3. Click **Save**.

Learn More

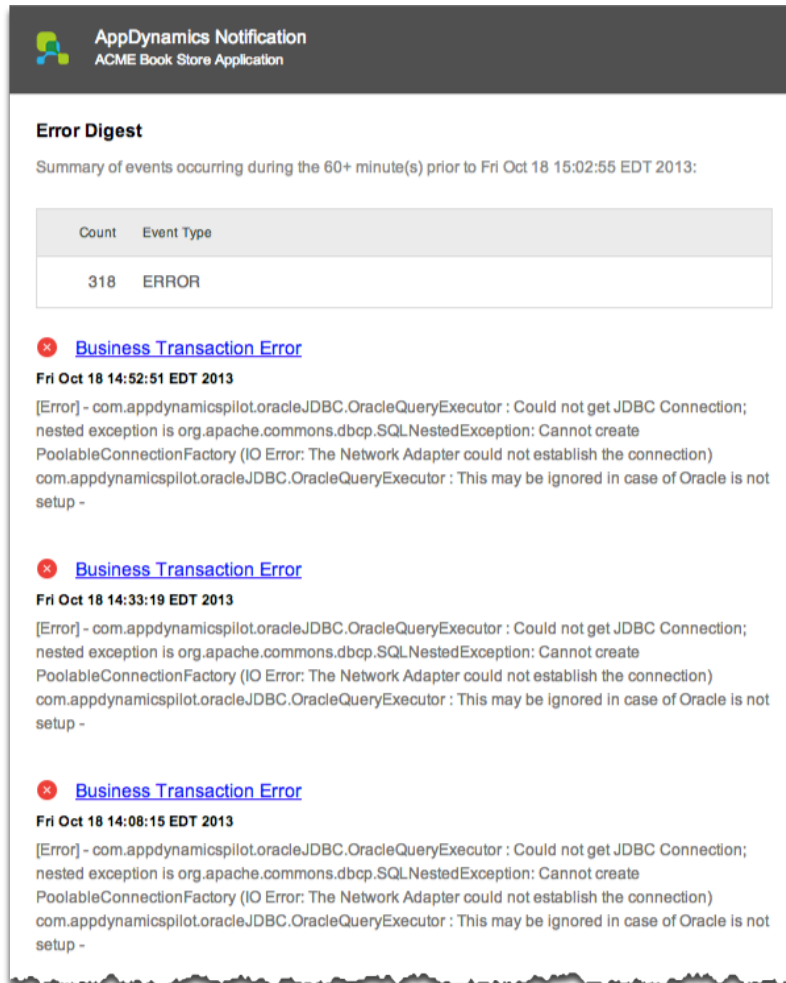
- [Action Suppression](#)
- [Health Rules](#)

Email Digests

An email digest is a compilation of messages sent to a recipient list by email at a configured time interval.

The purpose of the digest is to notify the recipients of events that have occurred in the application. The contents of the digest are automatically generated based on the context of the events that the digest reports.

See [Configure Email Digests](#) for configuration details.



Configure Email Digests

- [Structure of the Email Digest Wizard](#)
- [Configuring the Email Digest](#)
 - [To configure content settings](#)
 - [To configure digest recipients](#)
 - [To configure the digest interval](#)
- [Learn More](#)

Structure of the Email Digest Wizard

To access the Email Digest Wizard:

1. Click **Alert & Respond -> Email Digests**.
2. To edit an existing email digest, select the digest and click the Edit icon.

3. To remove an existing digest click the delete icon.

4. To create a new digest, click "+" .

The Email Digest Wizard opens. It contains four panels:

1. Contents: Sets the digest name, enabled status, health rule type, events and objects that trigger the sending of the digest
2. Recipients Adds the email addresses of the digest recipients.
4. How Often: Sets how often the digest is sent, in hours.

You can navigate among these panels using the Back and Next buttons at the bottom of each panel or by clicking their entries in the left panel of the wizard.

Configuring the Email Digest

To configure content settings

1. In the Content panel of the Email Digest Wizard, if you are creating a new digest, enter the digest name in the Name field.
If you are editing an existing digest, the name will already be there. You can change the name of the digest in the Name field.
2. To enable sending the digest check the Enabled check box. To disable sending the digest, clear the Enabled check box.
3. Check the check boxes for the events that will be included in the digest. You may need to click the down arrow to expose specific events within an event category.

4. When you have finished selecting the events in the digest, you can click "any object" to refine the contents by specifying only events that affect certain objects in the application. If you select

Any Objects the digest will include configured events when they occur on any object in your application.

To restrict the policy to specific objects, select Any of these specified objects and then choose the objects from the embedded object browser.

5. Click **Save** to save the digest configuration.

To configure digest recipients

Configuration of digest recipients involves selecting or creating an email notification action for every recipient of the digest. You can create these notification actions from this Email Digest Wizard as well as from the **Actions** menu.

1. In the Recipients panel of the Email Digest Wizard, if you are creating a new digest, enter the digest name in the Name field. If you are modifying an existing digest, double-click the digest in the list.
2. To edit an existing recipient, select the recipient from the list and double-click or click the Edit icon.
3. To remove an existing recipient from the email digest, select the recipient from the list and click the delete icon.
4. To add a recipient, click the "+" sign.
5. Do one of the following:

- To add a recipient who has already been configured (i.e. an existing email notification action), select the email notification from the list and click **Select**.

or

- Click **Enter Email Address** and enter the email address of the recipient in the text field.

6. Optional: In the Configure Action screen, you can also add an optional note to include in the email.
7. Click **Save** to add the recipient.
8. Click **Save** in the Email Digest Wizard to save the digest configuration.

To configure the digest interval

1. Click How Often to access the How Often panel.
- 2 In the text field enter an integer to indicate the number of hours between digests.
3. Click **Save** to save the digest configuration.

Learn More

- [Email Digests](#)
- [Policies](#)
- [Health Rules](#)
- [Notification Actions](#)

Getting Started Wizard for Alerts

- [To access the Getting Started wizard](#)
- [To edit or delete the generated policy](#)

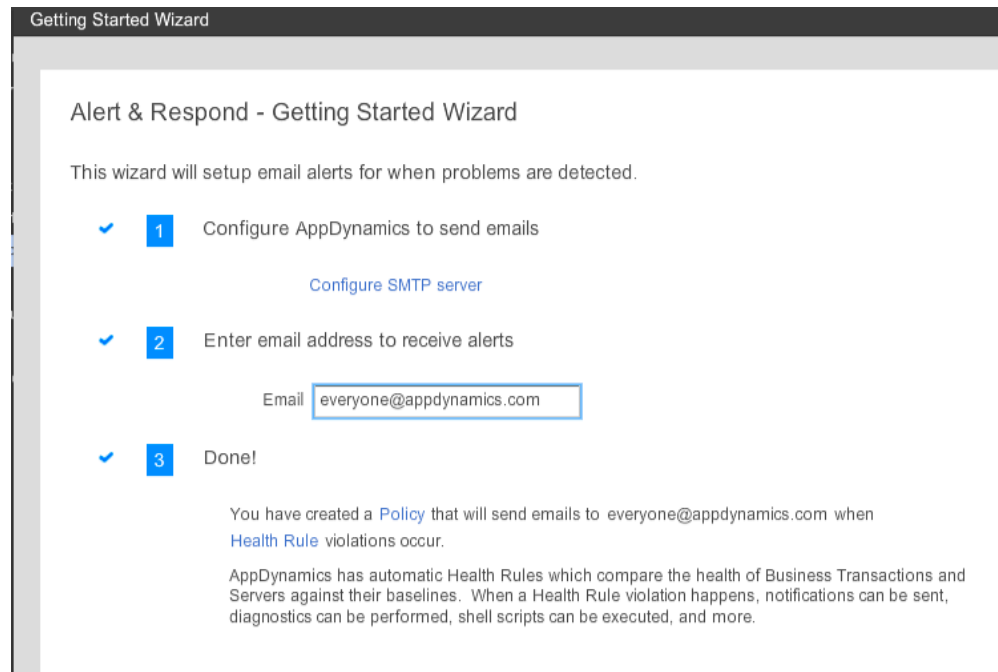
Learn More

When you are starting out, you can quickly configure an email notification to be sent when any health rule violates, using the Getting Started Wizard. This wizard creates an automatic policy that sends an alert to a single email address.

To access the Getting Started wizard

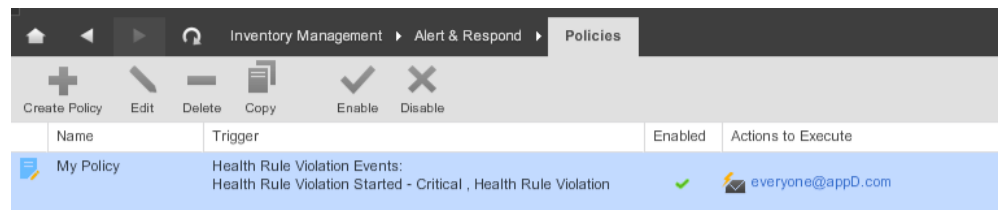
1. Click **Alert & Respond** in the left navigation pane. The first time you do this, the **Getting Started Wizard** opens automatically.
2. If this is **not** the first time you have opened **Alert & Respond**, click the **Getting Started Wizard** at the bottom of the Alert & Respond screen.
3. If your SMTP server is not set up, click **Configure SMTP Server**.
See [Configure the SMTP Server](#) for instructions on configuring the SMTP server for email and SMS notifications.
4. Enter the email address to which alerts will be sent.
5. Click **Save**.

This wizard creates a policy named My Policy or My Policy *n* that sends email to the configured address whenever any health rule violation is started in the application.



To edit or delete the generated policy

1. Click **Alert & Respond -> Policies** to access the policy list.



2. Do one of the following:

- To delete the automatically generated policy, select the policy and click the delete icon.
- To edit the automatically generated policy, select the in the list and click the edit icon. You can fine-tune the types of violations and events that trigger the alert by editing the policy manually. For example, you can change the name of the policy, add or remove events that trigger the notification, add additional notification email addresses or other actions to be triggered by the policy.

You can also create entirely new health rules, policies and actions. See [Policies](#).

3. Click **Save**.

Learn More

- [Health Rules](#)
- [Policies](#)
- [Actions](#)
- [Notification Actions](#)
- [Email Digests](#)

- [Configure Policies](#)
- [Configure Email Digests](#)

Alerting for Business Transaction Health Problems

- [Default Health Rules for Business Transactions](#)
- [Alerting with Notification Actions](#)
- [Creating Policies to Match Health Rule Violations with Alerts](#)
- [Learn More](#)

Business transaction health refers to the extent to which a business transaction is experiencing critical and warning health rule violations.

Use health rules, notification actions and policies to alert staff of business transaction performance problems.

Default Health Rules for Business Transactions

AppDynamics provides the following default health rules for business transaction performance:

- **Business Transaction response time is much higher than normal**
This rule defines a critical condition as the combination of an average response time greater than the default baseline by 3 standard deviations and a load greater than 50 calls per minute.
This rule defines a warning condition as the combination of an average response time greater than the default baseline by 2 standard deviations and a load greater than 100 calls per minute.
- **Business Transaction error rate is much higher than normal**
This rule defines a critical condition as the combination of an error rate greater than the default baseline by 3 standard deviations and an error rate greater than 10 errors per minute and a load greater than 50 calls per minute.
This rule defines a warning condition as the combination of an error rate greater than the default baseline by 2 standard deviations and an error rate greater than 5 errors per minute and a load greater than 50 calls per minute.

You can use these health rules as they are or you can modify them.

For information on how AppDynamics determines normal performance, see [Behavior Learning and Anomaly Detection](#) and [Configure Baselines](#).

For information about modifying health rules see [Health Rules](#) and [Configure Health Rules](#).

Alerting with Notification Actions

You can create notification actions to set up email and SMS addresses to receive notifications when the health rules are violated. See [Notification Actions](#).

You can also create email digests that are sent on a predefined schedule to email recipients summaries of these (and other) health rule violations. See [Email Digests](#).

Alerts in email notifications and digests provide a deep link to help the recipient start analyzing the root cause of the problem.

Creating Policies to Match Health Rule Violations with Alerts

Create one or more policies to assign a notification action to a specific health rule violation. You can optionally create different policies for warning and critical health rule violations. See [Policies](#) and [Configure Policies](#).

If you want simply to send an email notification to a single email address whenever any health rule violation is started, you can use the Alerting Wizard.

Learn More

- [Behavior Learning and Anomaly Detection](#)
- [Configure Baselines](#)
- [Health Rules](#)
- [Configure Health Rules](#)
- [Policies](#)
- [Configure Policies](#)
- [Notification Actions](#)
- [Email Digests](#)
- [Getting Started Wizard for Alerts](#)

Best Practices for Alerting and Integrating with Third Party Alerting Systems

- [Adopt Proactive Monitoring with AppDynamics Alerts](#)
- [Determine What Should Trigger an Alert](#)
- [Integrate AppDynamics into an Alerting Workflow](#)
- [Create, Test, and Debug Email Alerts](#)
- [Resolve Issues Using Remediation and Workflow Automation](#)
- [Integrate with Third-Party Alerting and Ticketing Applications](#)
 - [Pull Alerts Into Other Systems Using REST](#)
- [Develop a Playbook](#)
- [Learn More](#)

Need a PDF?

Download: [Best Practices for Alerting and Integrating with Third Party Alerting Systems](#)

Adopt Proactive Monitoring with AppDynamics Alerts

When operations professionals monitor AppDynamics dashboards, they can easily see diagnostic information on a variety of system conditions. However you cannot always be sure that someone is watching dashboards at all times. In addition to relying on dashboards to let you know when problems occur, you can implement AppDynamics alerts that directly inform operations staff when an issue needs attention.

By using AppDynamics alerts effectively, you can find and fix problems before they get serious enough to affect your customers. For example, instead of alerting when a system crashes, AppDynamics can alert you when a crash may be imminent, such as when CPU usage is too high or when memory is near its limit. This type of alert gives you time to diagnose and resolve the problem, and prevent a crash from occurring.

Effective alerts can help you shift your support practices from purely reactive, responding to a customer or internal complaint about an issue only after it occurs, to proactive, by preventing issues from affecting users in the first place.

Adopting proactive monitoring practices has many benefits, including:

- Reduction in support calls
- Increased customer satisfaction
- Increased revenue due to more uptime and lower response times
- Increased hours of sleep enjoyed by on-call support staff

When adopting proactive monitoring practices, it is important to develop a complete alerting strategy. The steps for doing this are:

- [Determine what should trigger an alert](#)
- [Integrate AppDynamics into an alerting workflow](#)
- [Create, test, and debug AppDynamics alerts](#)
- [Resolve issues directly using remediation actions and workflow automation](#)
- [Integrate AppDynamics with third-party alerting and ticketing applications](#)
- [Develop a playbook for responding to AppDynamics alerts](#)

Determine What Should Trigger an Alert

An effective proactive alerting strategy focusses on issues that are mission-critical. Too few alerts may result in important issues not being reported. If people get too many alerts, they may start ignoring them. A production issue that has no impact on the underlying business should not be a candidate for alerting, as it will create noise. Alert on issues that are worth waking someone up for.

The process for determining what alerts to create involves working with the relevant teams to determine their application's KPIs (key performance indicators). Based on the KPIs, define what conditions indicate a critical issue in the production application that needs immediate attention, and identify metrics that should trigger alerts, such as memory usage. You can also review historical events in the application lifecycle, and consult all application owners about what they consider critical to their success.

For example, suppose you have a "checkout" business transaction that has an average response time (ART) of 1000 ms. Based on historical information, you know that if the ART goes over 3000 ms, customers start abandoning their carts, and if it goes over 9000 ms, customers start contacting support or complaining on social media sites. In this example, you would want someone to be notified if the ART approaches 3000 ms, and a larger or different group to be notified if it approaches 9000 ms. By proactively notifying the staff who can then use AppDynamics to diagnose the problem and get the ART down to its normal time, you increase the likelihood that the problem will never reach a level that affects your customers and, by extension, your revenue.

Consider creating alerts for the following conditions:

- The response time of a key website operation (purchase, search) is too slow.
- Availability has fallen below your SLA threshold.
- The error rate for a critical operation, such as Business Transaction error rate, is over 10%.
- A database or remote service has stopped responding or is too slow, for example:
 - when a travel site needs pricing info from a hotel chain, and the call to the hotel chain becomes very slow, or
 - when an LDAP server is not responding.
- A JVM has crashed (see [JVM Crash Guard](#)).

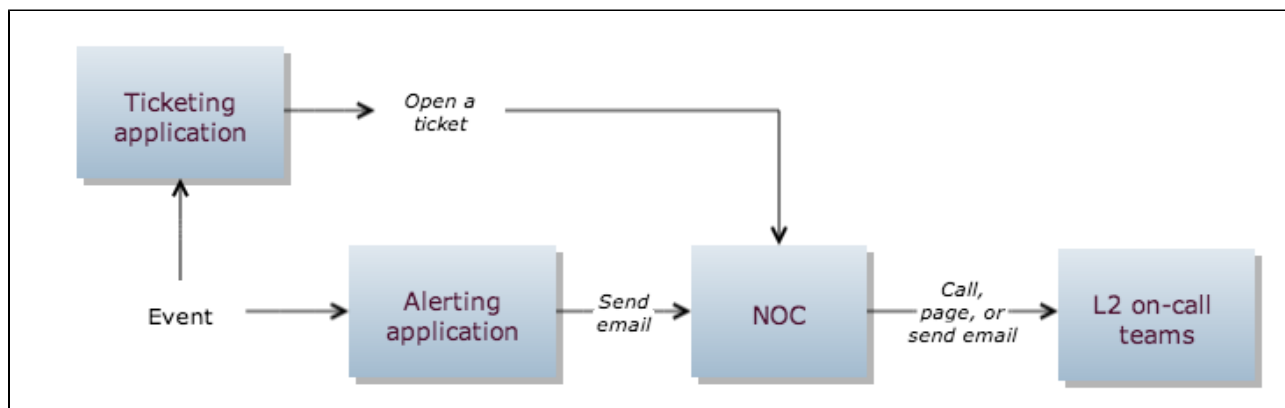
You can email an alert to any person or team who needs to address the issue. For example, you may want AppDynamics alerts to route through a centralized alerting system to your network operations center and also a ticketing system.

You can also use alerting techniques to send what you might call "warnings" to specific people or teams. For example, your development team may like to know when a heap is approaching its maximum capacity. This problem might not be serious enough to warrant an immediate call to action, but having the information would give the team a "heads-up" that they might need to review something in their environment.

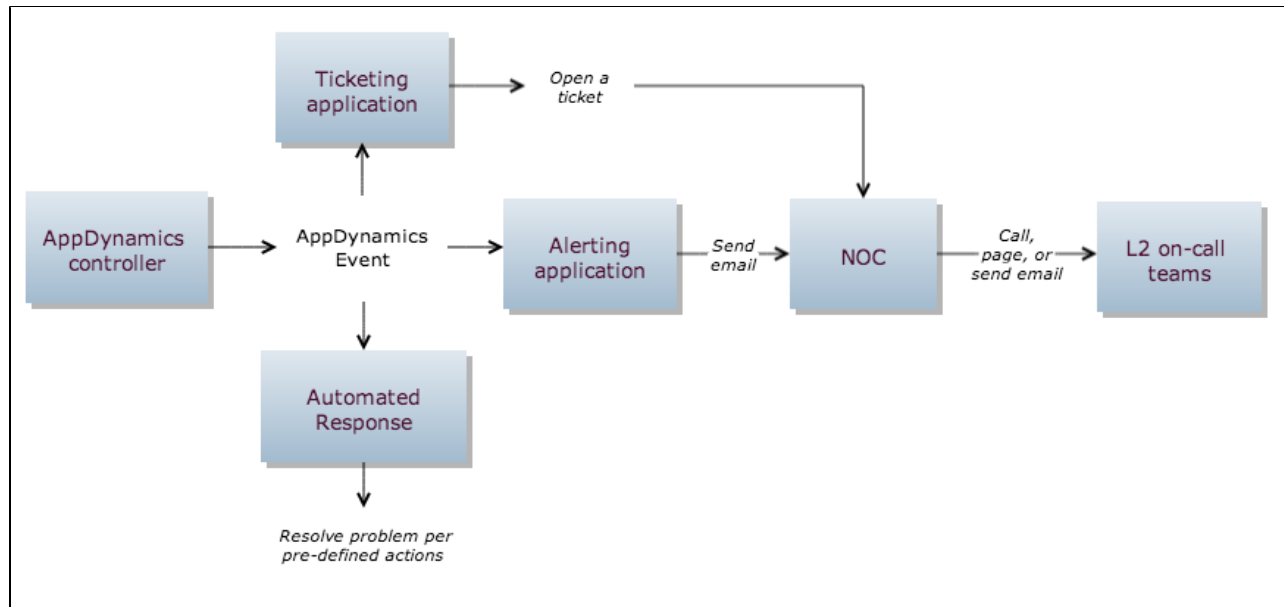
You can also create alerts specific to AppDynamics for Databases. For more information, see [Alert on Performance Metrics](#).

Integrate AppDynamics into an Alerting Workflow

Many organizations using AppDynamics already have an alerting workflow in place to notify the appropriate staff about a problem that needs immediate attention. For example, assume that your current alerting process looks something like this diagram:



After integrating AppDynamics alerts into your workflow, the AppDynamics piece of the process could look like this diagram:



After you have determined what alerts are needed, train team members how to interpret AppDynamics alerts and how to use AppDynamics to find and resolve the problem.

Depending on the alert, a notification email might contain a node name, a tier name, a business transaction name, and/or an application name. Teams are usually familiar with node names in alerts; they may need to learn about other AppDynamics terminology. AppDynamics recommends that you train NOC and other personnel by reviewing the AppDynamics interface and demonstrating flow maps, business transactions, the app/tier/node hierarchy, and various diagnostic techniques such as "drilling down" to find the root cause of a problem.

Remember to set up accounts and permissions

Make sure that everyone who will be receiving alerts has an AppDynamics account and the correct permissions to follow up on issues.

Create, Test, and Debug Email Alerts

After your teams know how to respond to AppDynamics alerts, you're ready to start setting them up.

To implement alerting in AppDynamics, you:

- Define health rules thresholds and other events that monitor critical status in your application
- Create policies that trigger when that status indicates a problem is emerging
- Configure actions to respond to those policy triggers.

For more information and instructions see [Alert and Respond](#).

Customers often use the "Email" option as the action. An email notification contains a deep link to the details of the event that triggered it. Clicking this deep link takes you directly to the place to start troubleshooting the problem. For more information, see [Email Notifications](#).

AppDynamics recommends that you use a staging system when you first develop AppDynamics alerts, to verify that you are sending the optimal type and number of alerts. Create alerts that operations staff will respond to because they know the alert reflects a serious issue.

Here are a few suggestions for setting up and verifying alerts on the staging system:

- For each policy, set up an action to send an email to only yourself, or to a few core team members.
- Tune the alerts:
 - If you get a lot of false positives, tweak your enabling conditions to trigger fewer alerts. Some conditions you might want to look at include: setting different threshold or baseline values, using a different evaluation period, including more than one condition in a health rule to narrow down when the alert will be triggered, alerting on a different metric (such as "max" instead of "value").
 - Compare alerts being sent with system health information displayed in AppDynamics. If there are issues for which alerts are not triggered when they should be, tweak your policy conditions to send more alerts.
- Route a single email alert to your central monitoring system or the NOC.
 - Check that it arrives; that the system routes it correctly; that NOC knows how to resolve the issue, etc.
 - Once you've confirmed the alert is getting routed and addressed correctly, route all alerts through your standard process.
- When you are confident that you are sending the correct alerts from AppDynamics, explore options other than sending emails.
 - Create actions related to diagnostics, running remediation scripts, etc. See the upcoming section, [Resolve Issues Using Remediation and Workflow Automation](#).
 - If you use a centralized routing platform, see the next section, [Integrate with Third-Party Alerting and Ticketing Applications](#).

Resolve Issues Using Remediation and Workflow Automation

Once you gain confidence in the validity of your alerts in AppDynamics, you can identify situations appropriate for using policy-based actions that run remediation scripts. For example, you can restart a JVM when its CPU processing time hits 90%, instead of sending an email to a team for them to restart it manually. You can also send them email so that they know the remediation occurred.

Remediation can accelerate mean time to resolution (MTTR), and reduce costs by allowing AppDynamics to take action to remediate the problem without waiting for the issue to be handled by staff. When you are confident about what actions to take based on the type of alert, you can configure a policy so the actions happen automatically when an appropriate alert fires. Depending on your agent (Java, .NET, etc.) you can automatically perform diagnostics such as thread dumps or snapshots. Performing diagnostics and also sending an email can reduce the MTTR, as the diagnostic data is already available when they respond to the email. For more information, see [Remediation Actions](#).

An additional automated resolution option is workflow automation. Workflow automation lets you spin up or down JVMs, configure JVMs, and so on. If you are running AppDynamics in a cloud environment, you can use the cloud auto-scaling feature to perform these same tasks in your compute cloud. As with diagnostics and remediation, automating these tasks leaves your staff free

to concentrate on other issues. Cloud auto-scaling can also directly reduce costs; by keeping up only as many instances are required for a particular load, you are not paying for instances that are not necessary. For more information, see [Workflow Overview](#) and [Cloud Computing Workflows](#).

Note: Workflow Automation and some remediation actions require the Standalone Machine Agent.

Integrate with Third-Party Alerting and Ticketing Applications

After you have built and tested an alerting workflow that incorporates the specific alerts generated by AppDynamics, you are ready to integrate those alerts with any third-party alerting/notification system you might already have in place. By using your existing system, your staff has a much shorter learning curve when it comes to reacting to AppDynamics alerts. Their process changes very little as they start to receive new types of alerts.

If you use a ticketing system to track problems, AppDynamics recommends integrating alerts into that system as well. Linking your existing ticketing system to AppDynamics helps make it easy for everyone to stay current with issues and their resolution, while avoiding duplication or extra data entry work.

AppDynamics has developed a number of product extensions to integrate with third party alerting and ticketing systems. To see a list of currently available extensions, go to [the AppDynamics Exchange](#) and filter on the Alerting category. If you are using an alerting system that is not available on the AppDynamics Exchange, you can build your own (see [Build an Alerting Extension](#)) or request that AppDynamics build one (see [Request an extension](#)). For information on integrating an extension you have built with AppDynamics, see [Custom Actions](#).

Pull Alerts Into Other Systems Using REST

If you have a program that pulls alerts from other systems, you can use the AppDynamics REST API to pull alerts and other events out of AppDynamics. You can do this by polling every few minutes. Once the alerts are extracted from AppDynamics, the program can then route the alert using information such as tier, node, or business transaction names. See [Use the AppDynamics REST API](#).

Develop a Playbook

Many NOC organizations use playbooks that specify the steps for staff to take to resolve certain types of alerts, when to escalate to a higher level of support, etc. Add the steps for responding to AppDynamics alerts to a playbook.

For example, you might include a section on general steps to take if the issue is related to business transactions, or to a node, tier, or application. An entry for business transactions might say, "Display the Business Transaction List in AppDynamics and search for the business transaction named in the email, then drill down to start looking for the root cause of the problem." Include a link to [Rapid Troubleshooting](#) in your playbook, for quick examples of how to troubleshoot different types of issues.

If your system connects to external services, include instructions for addressing issues. For example, an alert is received when an external hotel system (shown as a backend remote system

in AppDynamics) is not responding. Include information about the standard process for contacting the external team in the playbook. The playbook might read "Send an email with the subject 'System not responding' to support@hotelSupport.com" or "Send a summary of the issue to external_support@yourCompany.com; they will contact hotel support personnel."

Learn More

- [Alert and Respond](#)
- [Best Practices for NOC and Other Front-Line Support Staff](#)

AppDynamics Blogs:

- [Deploying APM in the Enterprise Part 5: Alerts](#)