Policies

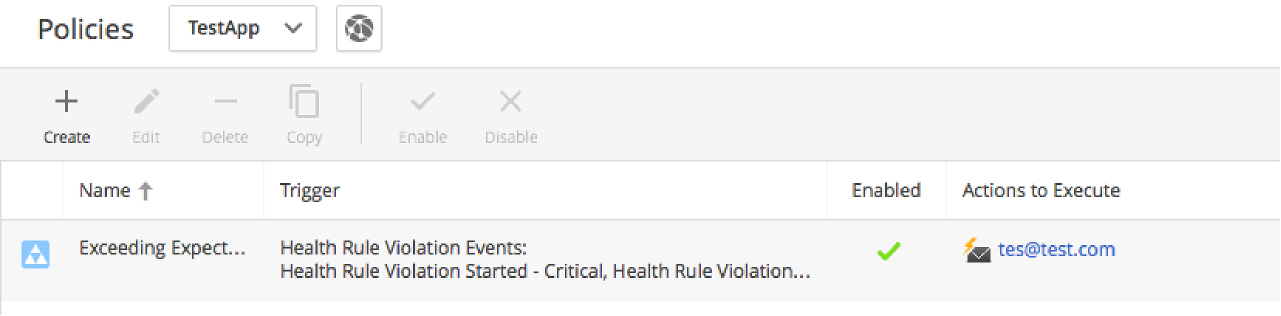
On this page:

This topic describes policies, the configuration element that associates an event in AppDynamics to the actions taken in response that event.

View Policies

To view and create policies, select **Alert & Respond > Policies**.

The policy list displays all the policies created for your application, with its triggers and actions taken. You can modify a policy by double-clicking it in the policy list.



	Name ↑	Trigger	Enabled	Actions to Execute
	Exceeding Expect...	Health Rule Violation Events: Health Rule Violation Started - Critical, Health Rule Violation...	✓	tes@test.com

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 Monitor Events.

The triggering events can be broadly defined as affecting any object in the application or 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 tune policies specifically for different entities and situations.

For example, you can have a broadly-defined policy that fires whenever a resource pool limit (such as > 80% usage of EJB pools, connection pools and/or thread pools) is reached for any object in the application.

Or a narrowly-defined JVMViolationInWebTier policy that fires only when existing health rules on memory utilization or JVM garbage collection time are violated.

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

Policy Actions

You can assign one or more actions to be automatically taken in response to the policy trigger.

Perhaps the most common action is a notification: sending an email or SMS.

Other types of actions do more than just notify.

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

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 a Custom Action 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 precisely defined, you can take different actions for breaching the same thresholds based on which tier or node the violation occurred in.

Policy Actions in Batch

You can configure a policy:

- To execute its actions immediately for every triggering event. This is the default.
  For example, if in a two-second period a policy matched 100 events, it would start its actions 100 times as soon as each event occurred.
- To execute its actions once a minute for all the events that triggered over the past minute. This is the batch option.
  For example, if in a two-second period, a policy matched 100 events and then no triggering events occurred for the next 58 seconds, the policy would start each action just once. The context for the actions would be all 100 events.

Whether or not to batch the actions depends primarily on the type of action. For a notification action it probably doesn't make sense to send 100 emails or SMSs in two seconds. In this case, it makes sense to batch the actions with a summary of the last one minute's events. This can be easily accomplished using an email template that iterates through the event list. See the example in Predefined Templating Variables.

However, if the actions are thread dumps, there is no reason to expect that all 100 events are on the same node. They might be on different nodes. For that kind of action, you would probably want the thread dump to be taken for each event and also, not to wait another 58 seconds before taking the thread dump.