Application Instrumentation and Monitoring

New Relic is our primary instrumentation and monitoring tool to help replace SCOM and enable previously un-obtainable metrics about our application performance. We are also using New Relic to instrument our infrastructure on-prem and in AWS.

Instrumenting an Application

To instrument an application using New Relic, the correct framework agent must first be installed on the server. Shared cookbooks exist for .NET Framework and .NET Core agents on Windows Server. Details on other agent installs can be found on the New Relic documentation.

Once the agent is installed, an application-specific configuration must be created to encapsulate it into a unique New Relic application. Details about the configuration can be found on the New Relic documentation. If using Octopus Deploy for your application, there is a step template (New Relic - Create Application APM Config) and shared variable sets (New Relic APM, Data Tier) which can be used for to create this config.

Naming Standards

To keep consistency with our application names inside New Relic, please config the application-specific name to follow the following pattern:

`<platform> - <environment> - <application>`

`<platform>` - This value should describe the Direct Supply platform. i.e. **eCommerce**, **TELS**, etc.

`<environment>` - This value should contain the same value for environment described in the cloud resource naming standard.

`<application>` - This value should contain the same value for application described in the cloud resource naming standard.

Enhancing Instrumentation

By default, New Relic will only instrument entry point methods of code, database calls and external service calls. It will also not include any parameters or data from the request (besides the URL.) To include additional method instrumentation, or to supply custom attributes (such as username, custcode, etc.) you can reference the New Relic API within code.

Infrastructure Agent Installation

In preparation of a patching process refactor, we will no longer be installing the New Relic infrastructure agent into our AMIs as of version 3 of our linux-base cookbook or version 5 of our windows-base cookbook. Our shared base Linux and Windows images will likewise be versioned to remove New Relic. With these version updates (and updates to any relevant Terraform modules), New Relic installation will migrate to AWS Systems Manager (SSM).

Running the SSM document to install the New Relic Infrastructure agent can be accomplished by running the `install-new-relic-infrastructure` document either as a step in an existing SSM document, or by tagging your instances.

> **Note**
>
> Your ec2 instances will need to have, at least, the following IAM policy attached to their IAM roles for AWS SSM to be able to associate documents: `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore`

New Relic Infrastructure Agent Installation as an SSM Step

If your infrastructure bootstrap automation triggers a reboot (e.g. by joining a domain), you must run the New Relic installation document as a step in your existing SSM automation. The windows-standalone module is an example of this process.

In the parameters section of your SSM document, include:

```
"parameters": {
    "InstallNewRelic": {
        "type": "String",
        "default": "${install_new_relic}",
        "allowedValues": [
            "true",
            "false"
        ]
    },
    "EnableNewRelicProcessMetrics": {
        "type": "String",
        "default": "${enable_new_relic_process_metrics}",
        "allowedValues": [
            "true",
            "false"
        ]
    }
}
```

Then, in the mainsteps portion of your ssm document, include the following:

```
{
    "name": "InstallAndConfigureNewRelic",
    "action": "aws:runDocument",
    "maxAttempts": 2,
    "precondition": {
        "StringEquals": [
            "{{ InstallNewRelic }}",
            "true"
        ]
    },
    "inputs": {
        "documentType": "SSMDocument",
        "documentPath": "install-new-relic-infrastructure",
        "documentParameters": {
            "EnableNewRelicProcessMetrics": "{{ EnableNewRelicProcessMetrics }}"
        }
    }
}
```

Note

The version of your SSM document must be 2.2. If it is not, you'll need to update your json to be SSM v 2.2 compliant. You will have to either rename your SSM document when you deploy your updates or first destroy the existing version of your document as version 0.3 documents cannot be updated in place. An example of a 2.2 document that runs the New Relic Infrastructure Agent install document can be found here.

In your Terraform, when creating the ssm document from your .json file, you'll need to include values for `install_new_relic` and `enable_new_relic_process_metrics`. Your code should look something like this:

```
resource "aws_ssm_document" "configure_server" {
    name          = "${module.base_tags.name_prefix}-${var.role}-configure-new"
    document_type = "Command"

    content = templatefile("${path.module}/configure-server.json", {
        install_new_relic                = var.install_new_relic
        enable_new_relic_process_metrics = var.enable_new_relic_process_metrics
    }
}
```

New Relic Infrastructure Agent Installation via Tags

If your infrastructure bootstrap automation doesn't trigger a reboot, you may simply tag your instances to associate with the New Relic installation SSM document: `InstallNewRelic = true` and `EnableNewRelicProcessMetrics = [true or false]` Note that Process Metrics should be enabled in production, but should be disabled for sandbox and testing unless needed for troubleshooting purposes.

Additional Configuration

For on-prem servers, the following custom attributes should be configured so we can correctly group servers as we would with AWS instances:

`label.Application` - Follows the cloud resource tagging standard for "application"

`label.Environment` - Follows the cloud resource tagging standard for "environment" or the Octopus Deploy environment name.

`label.SoftwareApplication` - Describes the software running on the server (e.g. "SQL Server", "SSRS", "IIS", etc.)

`on-premise` - Should be set to "true" if the server is hosted on-premises, otherwise excluded or set to "false"

`maintenance.mode` - Should be set to "true" while a server/instance is undergoing maintenance to prevent alerts from being generated. Otherwise exclude or set to "false"

Custom Instrumentation

Using Infrastructure agent integrations or the New Relic Insights API, we are able to record any number of custom events or metrics for reporting or alerting. Please reach out to SRE if you have any interest in performing custom instrumentation of this level.

Alerting

Alerts around any metric or event data can be configured within New Relic. Only SRE currently has access to create alerts as we ask that we are involved with the creation of any alerts to ensure we are knowledgeable on their cause and possible resolution. In most cases these alerts will be routed to PagerDuty for notifying the on-call SRE, however additional communication methods can be configured such as email or SMS.

If you would like to have alerts created, please reach out to SRE.

New Relic Usage

To help control costs, by default New Relic infrastructure instrumentation should be configured as follows in our AWS accounts:

- Sandbox - Disabled by default
- Testing - Enabled by default with process-level metrics disabled by default
- Production - Enabled by default with process-level metrics enabled by default

Any uses of APM or On-Host-Integrations should likewise be disabled by default in our pre-production accounts.