

Implementation Guide:

Integrating New Relic with AWS Control Tower



Table of Contents

Foreword	3
Solution Overview.....	4
Architecture Diagram.....	4
Pre-requisites.....	7
Deploying the Solution.....	7
Best Practices.....	14
New Relic and AWS.....	14
Pricing	14
Tearing it Down.....	14
Partner FAQs.....	14
Partner contact information	15

Foreword

[New Relic](#) is an [AWS Partner Network](#) (APN) [Advanced Technology Partner](#) focused on delivering observability as a platform, to help you create more perfect software. [New Relic One](#) is New Relic's observability platform that is open, connected, and programmable. For more information, see [New Relic in AWS Marketplace](#).

[AWS Control Tower](#) leverages AWS best practices to easily set up a scalable, secure, well-architected, multi-account environment, referred to as a [landing zone](#). The [Account factory](#) automates the provisioning of new accounts in your AWS environment. You can enable self-service for your teams to configure and provision new accounts using [AWS Service Catalog](#). After you have set up your landing zone, you need operational insights into your multi-account environment.

The purpose of this AWS Implementation Guide is to enable you to seamlessly activate, deploy and configure [New Relic's AWS integration](#) in your AWS Control Tower landing zone.

Solution Overview

[New Relic's integration with AWS Control Tower](#) solution allows you to streamline the observability of your AWS accounts with New Relic. With this solution, whenever a new AWS account is enrolled, it's automatically integrated with your New Relic account using the [New Relic Infrastructure](#) integration. You can then manage all your AWS operational data and insights from one place, with no need to hop back and forth between multiple AWS accounts or other set of observability tools.

[New Relic AWS integrations](#) is the New Relic Infrastructure integration for AWS and combines configuration monitoring and real-time health metrics with a dynamic, tag-driven approach to observability to give your DevOps teams full visibility of your diverse AWS infrastructure. At the time of this writing, New Relic AWS integrations support over 50 AWS services. It also supports seamless deployments of your workloads using [AWS Outposts](#). For a list of the supported services and the data that can be collected, visit [AWS integrations list](#).

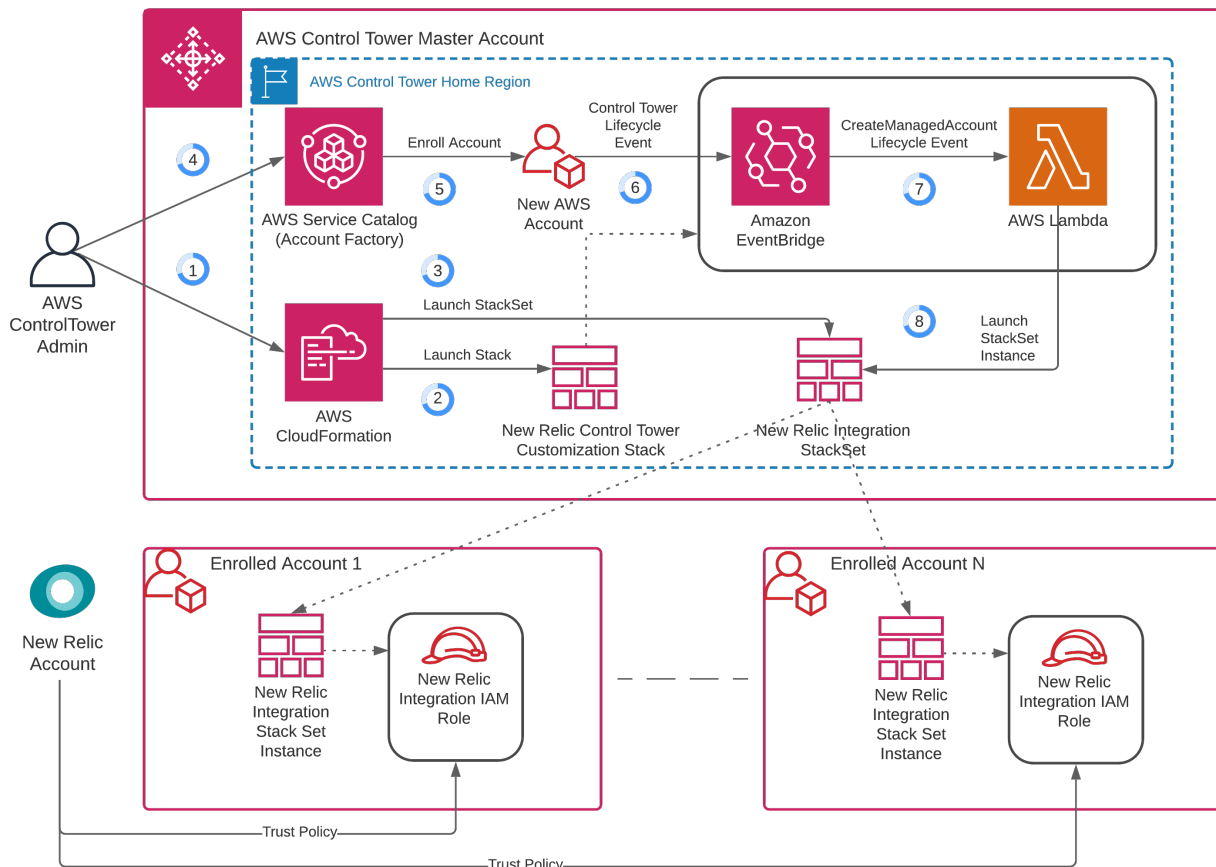
New Relic AWS integrations require you to grant New Relic permission to read operational telemetry data from your AWS account. This is achieved by using an AWS IAM role, referred to as “New Relic Integration”, that uses [AWS Identity and Access Management \(IAM\) cross-account access](#). New Relic AWS integrations uses the [Amazon CloudWatch API](#) to obtain telemetry data for the AWS services you monitor. The solution automates the deployment of this IAM role as you vend and enroll new accounts into your AWS landing zone, and hence makes it observable with New Relic, right from the very beginning.

The solution is open source, and the code is available in New Relic's [GitHub repository](#). You must deploy New Relic AWS integrations in your [AWS Control Tower management account](#) in the home [Region](#). This is the Region where your AWS Control Tower landing zone was set up. This solution uses [AWS CloudFormation](#) for its deployment, and it includes two AWS CloudFormation templates that you deploy in the following order:

1. The “New Relic Integration” template creates a [stack set](#) consisting of the New Relic Integration [Identity and Access Management \(IAM\)](#) role.
2. The “New Relic AWS Control Tower Customization” template creates a [stack](#) consisting of an [Amazon EventBridge](#) rule and an [AWS Lambda](#) function.

Architecture Diagram

The following architecture diagram illustrates the solution's deployment steps in an AWS Control Tower environment, its integration with New Relic and the enrollment of new AWS accounts.



Let's walk through the logical sequence of steps as you deploy the New Relic integration with AWS Control Tower:

1. First, you will use AWS CloudFormation to launch New Relic's AWS CloudFormation templates in your [AWS Control Tower management account](#). The templates are meant for launching a stack and a stack set. You will need to deploy these in your "home" region; this is the region where the AWS Control Tower Landing Zone is deployed.
2. The "New Relic Integration" stack set is created first. This stack set includes an AWS IAM role and policy. The role is necessary for trusting the New Relic's AWS account and the policy grants permissions to fetch data from your AWS accounts across the AWS services that are supported by New Relic.
3. The "New Relic Control Tower Customization" stack is created next. It's comprised of an Amazon EventBridge rule and an AWS Lambda function.
4. Once the stack set and stack are successfully created, you will navigate to AWS Service Catalog.
5. From AWS Service Catalog console, you will use Account Factory to vend and enroll one or more AWS accounts in your AWS Control Tower environment.
6. After each account is successfully enrolled, corresponding [AWS Control Tower Lifecycle event](#) is emitted and delivered to Amazon EventBridge and picked up by the Amazon EventBridge rule.

7. In response to the incoming event, the Amazon EventBridge rule will trigger the AWS Lambda function.
8. Based on the information in the event, including Account ID, and region, the AWS Lambda function launches the stack set instances of “New Relic Integration” stack set in the newly enrolled account. At this point, Amazon CloudWatch and other supported AWS Services are configured to accept requests for pulling observability data from your AWS accounts, and that data can now be accessed from your New Relic account.

Steps 6, 7, and 8 will be automatically triggered for every account that you enroll using Account Factory. Let’s look at how you can deploy the solution in your environment.

Pre-requisites

You need the following prerequisites to implement New Relic's integration with AWS Control Tower:

- Fully deployed AWS Control Tower. For information about setting up an AWS Control Tower landing zone, see [Getting Started with AWS Control Tower](#). You also need administrator privileges in the AWS Control Tower management account.
- An active New Relic account with Standard or higher pricing tier subscription, when using the [New Relic One pricing plan](#). Don't have an account yet? [Sign up](#) for a perpetually free access to New Relic, which includes 100 GB of ingest per month and one Standard User license. You can also contact [New Relic Sales](#) for more details.

Deploying the Solution

Now let's deploy the solution in your AWS Control Tower environment. At a high level, deployment involves following steps:

1. Make sure you have access to your New Relic account, or sign up for your free New Relic account. Take note of your New Relic Account ID and License key.
2. Using AWS CloudFormation, use the New Relic provided templates to create a stack set and stack in your AWS Control Tower management account in your "home" region, and then use Account factory to vend and enroll one or more new AWS accounts. These new accounts will be automatically integrated with New Relic.
3. Activate AWS integrations for the new accounts from your New Relic account. Take note of the Amazon Resource Name (ARN) of the "New Relic Integration" AWS IAM Role in each account.

Step 1.1: Make sure you have an active New Relic account or signup for free

Make sure you have an active New Relic account with Standard or higher pricing tier subscription, when using the [New Relic One pricing plan](#), or signup for a perpetually free access to New Relic.

Step 1.2: Take note of your New Relic Account Number and License key

Log into your [New Relic account](#), and navigate to [Account Settings](#). Take note of your New Relic Account ID and License key, as shown below. If you are unable to find your Account ID, please follow instructions under the [Account ID documentation](#).

Account information

Name

Account 

Subscription

[New Relic AI Incident Intelligence](#)

[New Relic AI Proactive Detection](#)

[New Relic One - Standard Users \(PAYG\)](#)

[New Relic One Data](#)

License key

 NRAL

Step 2.1: Create AWS CloudFormation stack set

You can create the AWS CloudFormation stack set via AWS Management Console or via AWS Command Line (CLI).

Create NewRelic-Integration AWS CloudFormation stack set using AWS Management Console:

1. Download the template file [newrelic-stack-set.yml](#) from solution GitHub repository. You will use this template to create a stack set. Make sure you name the stack set as NewRelic-Integration.
2. Log in to the AWS Control Tower management account. This is the AWS account where AWS Control Tower is deployed.
3. Verify that you are in your home region (region where AWS Control Tower is deployed).
4. Navigate to [AWS CloudFormation console](#).
5. On the left-hand side menu, select **StackSets**. If you are unable to view the left-hand side menu, make sure the menu is expanded by clicking on the three dashes that denote the minimized view of the menu.
6. Click **Create StackSet**.
7. In the “Choose a Template” step, select **Template is ready** and **Upload a template file**.
8. Select **Choose file** and browse to the downloaded newrelic-stack-set.yml file.
9. On the **Options** screen, click **Next**.
10. In the “Specify StackSet details” step, specify a name for your stack set in **StackSet name**.
11. Enter your **New Relic Account Number**. This is the same as your New Relic Account ID. Click **Next**.
12. In the “Configure StackSet options” step, navigate to **Permissions**. If you see it, select **Self-service permissions**. Select **AWSControlTowerStackSetRole** from the drop down menu under **IAM admin role ARN**. The **IAM execution role name** should be **AWSControlTowerExecution**. Click **Next**.

13. In the “Set deployment options” step, Enter the comma separated Account IDs of your AWS Control Tower [log archive account](#) and [audit account](#) in the **Account numbers** text area. For further details on how to find the Account ID, see [Your AWS account identifiers](#) in the *AWS General Reference*.
14. Select your home region in the drop down menu under **Specify regions**. You can optionally choose your **Deployment options** or leave as defaults depending on your preferences. Click **Next**.
15. In the “Review” step, review the stack set details. Select the acknowledgment check box under **Capabilities**. Click **Submit**.
16. You will be taken to “StackSet details” page, under the “Operations” tab, where you can monitor the progress of the stack set that you just attempted to create. Wait until you make sure, the **Status** reads **SUCCEEDED**.

Create NewRelic-Integration AWS CloudFormation stack set using AWS CLI:

Setup your AWS CLI profile to point to AWS Control Tower management account with Administrator permissions. Make sure you point to the AWS region where your AWS Control Tower is deployed.

Create an AWS CloudFormation stack set from `newrelic-stack-set.yml` template file. You can do so by running the below AWS CLI command from a terminal or shell.

- Ensure the stack set name is NewRelic-Integration.
- Replace `NEW_RELIC_ACCOUNT_ID` with your New Relic Account ID.
- Replace `AWS_CONTROL_TOWER_MANAGEMENT_ACCOUNT_ID` with your AWS Control Tower management account’s Account ID.

```
aws cloudformation create-stack-set \
--stack-set-name NewRelic-Integration \
--template-body https://raw.githubusercontent.com/newrelic-experimental/newrelic-control-tower-customization/master/templates/newrelic-stack-set.yml \
--description "Adds in New Relic integration to your AWS accounts" \
--parameters ParameterKey=NewRelicAccountNumber,ParameterValue=<NEW_RELIC_ACCOUNT_ID> \
--administration-role-arn arn:aws:iam::<AWS_CONTROL_TOWER_MANAGEMENT_ACCOUNT_ID>:role/service-role/AWSControlTowerStackSetRole \
--execution-role-name AWSControlTowerExecution \
--capabilities CAPABILITY_NAMED_IAM
```

After the command returns successfully, the stack set creation starts. You can check the status of the stack set creation after a few seconds. To verify the stack set was successfully created, run the below command:

```
aws cloudformation describe-stack-set --stack-set-name NewRelic-Integration --query "{Name: StackSet.StackSetName, Status: StackSet.Status}"
```

and make sure the following is returned.

```
{
  "Name": "NewRelic-Integration",
  "Status": "ACTIVE"
}
```

Step 2.2: Create AWS CloudFormation stack

Create an AWS CloudFormation stack from control-tower-customization.yml template file. This template does not require any parameters.

Create NewRelic-ControlTowerCustomization stack using AWS Management Console:

1. Download the template file [control-tower-customization.yml](https://raw.githubusercontent.com/newrelic-experimental/newrelic-control-tower-customization/master/templates/control-tower-customization.yml) from solution GitHub repository. You will use this template when creating the AWS CloudFormation stack.
2. Log in to your AWS Control Tower management account. This is the AWS account where AWS Control Tower is deployed.
3. Verify that you are in your home region (region where AWS Control Tower is deployed).
4. Navigate to [AWS CloudFormation console](#).
5. Click **Create Stack**. From the drop down menu select **With existing resources**. Click **Next**.
6. Select **Upload a template file** and click on **Choose file**. Browse to the downloaded control-tower-customization.yml file.
7. Select **Next** and create the stack.

Create NewRelic-ControlTowerCustomization stack using AWS CLI:

Setup your AWS CLI profile to point to AWS Control Tower management account with Administrator permissions. Make sure you point to the AWS region where your AWS Control Tower is deployed.

Create the AWS CloudFormation stack viz. NewRelic-ControlTowerCustomization from newrelic-stack-set.yml template file. You can do so by running the below AWS CLI command from a terminal or shell.

```
aws cloudformation create-stack \  
--stack-name NewRelic-ControlTowerCustomization \  
--template-body https://raw.githubusercontent.com/newrelic-experimental/newrelic-control-tower-customization/master/templates/control-tower-customization.yml \  
--capabilities CAPABILITY_NAMED_IAM
```

After the command returns successfully, the stack creation starts. You can check the status of the stack creation after a few seconds. To verify the stack was successfully created, run the below command:

```
aws cloudformation describe-stacks --stack-name NewRelic-ControlTowerCustomization \  
--query "Stacks[0].{Name:StackName,Status:StackStatus}"
```

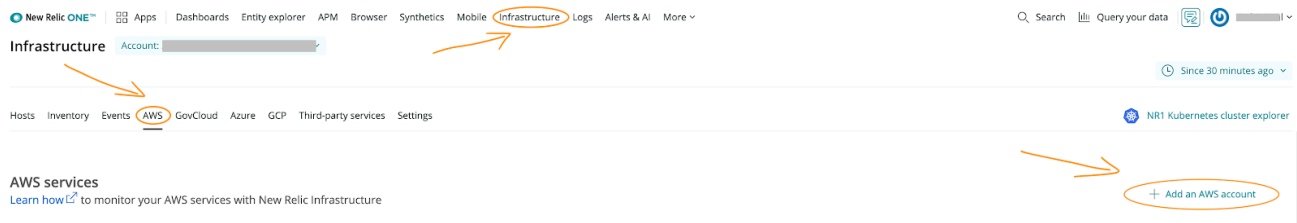
and make sure the following is returned.

```
{  
  "Name": "NewRelic-ControlTowerCustomization",  
  "Status": "CREATE_COMPLETE"  
}
```

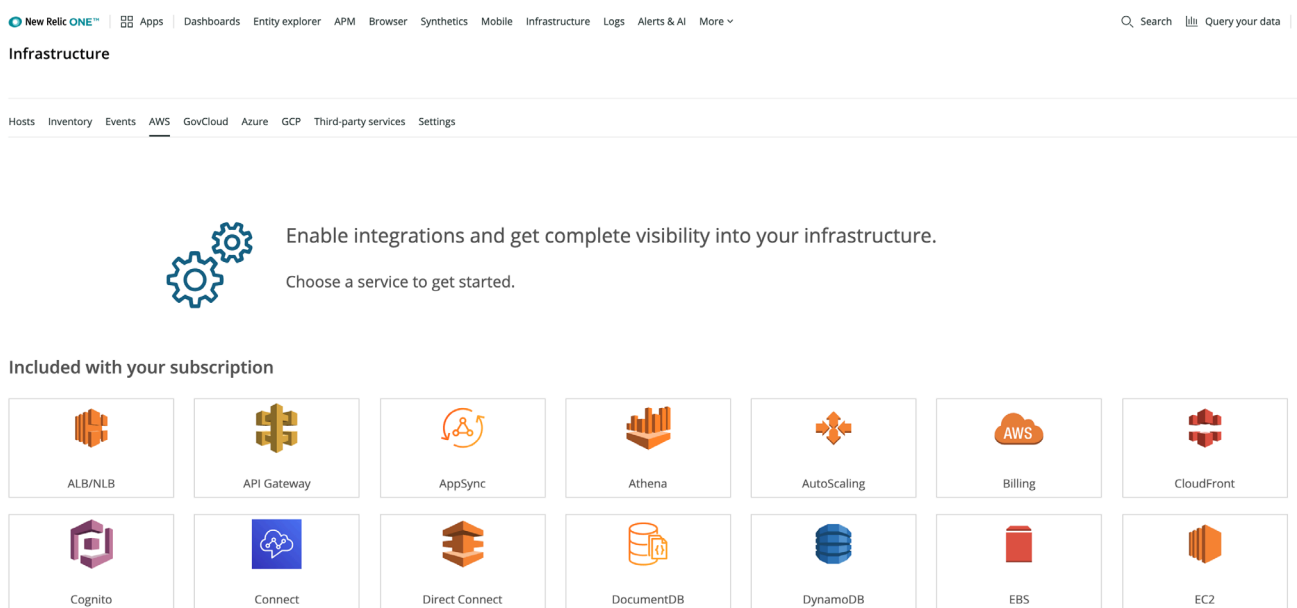
Step 3: Activate and Verify New Relic Integration

Once a new account is enrolled from AWS Control Tower, it is automatically set up to allow your New Relic account to ingest operational telemetry data via Amazon CloudWatch and other AWS services supported by New Relic integration. You do need to first activate new AWS account integrations from your New Relic account before you can see any telemetry data there.

3.1 Log into your [New Relic account](#), and then click the **Infrastructure** link on the top navigation bar. You will be taken to the **Infrastructure** page. If you happen to have access to multiple New Relic accounts, begin by choosing the New Relic account that you used for this implementation, from the dropdown list labeled **Infrastructure**. Otherwise, you should see your New Relic account show up next to the label. Make sure the account ID matches the one you used in this implementation. Next, select the **AWS** tab and finally click the **+ Add an AWS account** link as shown below.



In case you are adding an AWS account to your New Relic account for the first time, you may see the



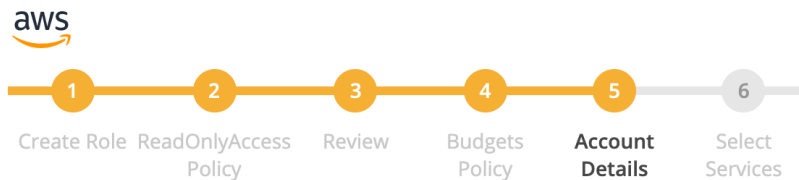
following screen instead. As instructed on the screen, click any one service tile to get started.

3.2 If you see the **Choose an integration mode** screen, click **Use API polling** button.

3.3 You will be presented with a multi-step account setup wizard. Since the solution automates the New Relic integration in your AWS accounts, you can move past the first few steps by clicking the **Next** button on each step until you get to **Step 5** named **Account Details**. You are skipping the steps since the solution automates the process of setting up the “New Relic Integration” IAM Role (done in Step 1 through Step 3). Step 4 named **Budgets Policy** is optional but recommended for you to keep track of your AWS cost.

Infrastructure

[Hosts](#) |
 [Inventory](#) |
 [Events](#) |
 [AWS](#) |
 [GovCloud](#) |
 [Azure](#) |
 [GCP](#) |
 [Third-party services](#) |
 [Settings](#)



[< Back to Step 4](#)

Step 5: Add Account Details

Add your AWS account name as it will show in the New Relic UI.

Changing the AWS account name is not recommended. Doing so may break New Relic Insights widgets, dashboards, and data apps created by your New Relic users.

Paste the ARN created in the previous step

Cancel Setup

Next

3.4 While you are in the **Account Details** step, type in the preferred name for your AWS account. This can be anything that helps you identify your AWS account from your New Relic account. Since you can integrate multiple AWS accounts, choose a name that's unique, or try matching it with the actual name of your AWS account. Enter the ARN of the "New Relic Integration" IAM role that was setup by the solution in your newly enrolled AWS account. Finally, Click **Next**.

You can find the ARN by logging into your recently enrolled AWS account using AWS Console, and then navigating to **IAM > Roles**. Search for the role named:

`NewRelicIntegrationRole_<NEW_RELIC_ACCOUNT_ID>`

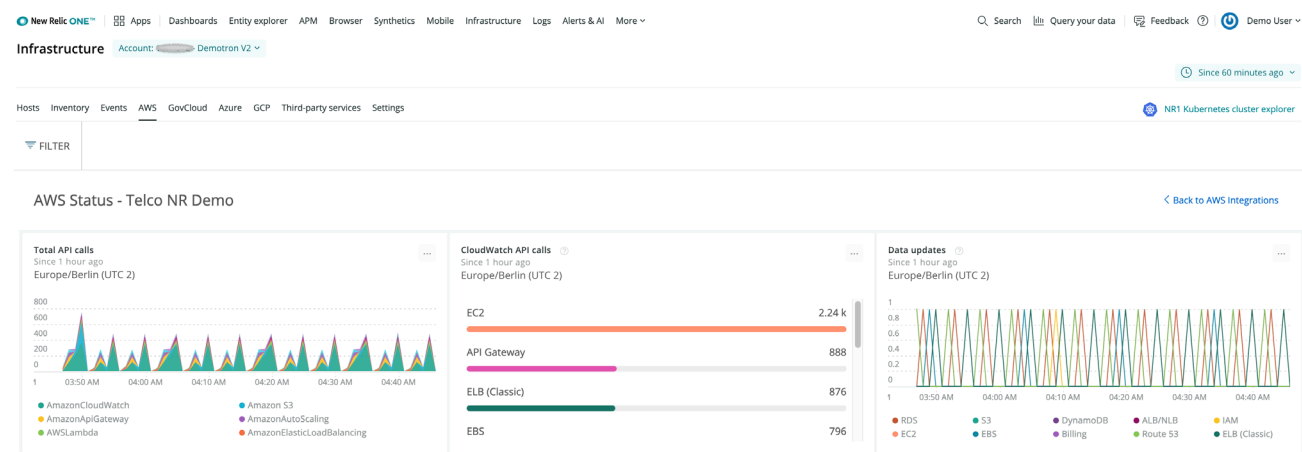
Click on the IAM role to be taken to the role details page, and finally copy the role's ARN

`arn:aws:iam::<AWS_ACCOUNT_ID>:role/NewRelicIntegrationRole_<NEW_RELIC_ACCOUNT_ID>`

3.5 In **Step 6**, named **Select Services**, select the AWS services you would like to monitor. Your new AWS account will be listed in the screen as shown below. Click on **Account status dashboard** link to view the account dashboard.

INTEGRATION NAME	DASHBOARDS	EXPLORE DATA	DOCUMENTATION
ALB/NLB	ALB dashboard NLB dashboard	ALB/NLB data	ALB/NLB Documentation
API Gateway	API Gateway (APIs) dashboard API Gateway (stages) dashboard	API Gateway data	API Gateway Documentation
AutoScaling	AutoScaling dashboard	AutoScaling data	AutoScaling Documentation
Billing	Billing (Costs) dashboard Billing (Budgets) dashboard	Billing data	Billing Documentation
DynamoDB	DynamoDB dashboard	DynamoDB data	DynamoDB Documentation
EBS	EBS dashboard	EBS data	EBS Documentation
EC2	EC2 dashboard	EC2 data	EC2 Documentation

Your AWS account status dashboard should look like this. Wait for a few minutes for the data to show up and adjust the time picker as necessary.



After waiting a few minutes, if you still do not see data for your Amazon Web Service (AWS) integrations in the account status dashboard, follow the suggestions in the [Troubleshooting AWS integrations](#) guide. Note that New Relic can obtain monitoring data from services and endpoints that are located in all AWS regions except from China regions.

Best Practices

Please refer to [Best practices for monitoring AWS with New Relic](#). See [AWS Cost Optimization](#) for best practices for cost optimization and billing management.

New Relic and AWS

Learn more about New Relic and AWS partnership at [New Relic and AWS](#). Read our [Data Sheet](#) and [Solution Sheet](#) for monitoring AWS.

Pricing

There are two components to the overall price you should expect to pay. First is the cost of data ingested into your New Relic account. With New Relic [Telemetry Data Platform](#), you get the first 100 GB of data for free, and then pay \$0.25 per GB of data ingested beyond the free limit. For more details on how New Relic is priced, see [New Relic Pricing](#). The second component of the price is your AWS bill largely owing to the Amazon CloudWatch API calls. New Relic AWS integrations uses the CloudWatch API to obtain telemetry data for the AWS services you monitor. The number of calls to the CloudWatch API increases as you enable more integrations, add AWS resources to those integrations, or scale those integrations across more regions. For details on how CloudWatch API calls are charged, see [Amazon CloudWatch pricing](#). There may be other components of your AWS bill including Network data transfer fee depending on your usage of New Relic Infrastructure agents, running on your EC2 instances or your containers running on ECS or EKS.

Please refer to [CloudWatch billing](#) for information on how to manage the cost associated with New Relic AWS integrations.

Tearing it Down

If you intend to deploy the solution for testing and demonstration purposes and you don't intend to use New Relic AWS integrations any longer, you can [Uninstall New Relic AWS integrations](#).

Partner FAQs

Can I purchase New Relic from the AWS Marketplace?

Yes! You can access and purchase New Relic One [directly from the AWS Marketplace](#) which offers pay-as-you-go pricing, including access to our perpetual free tier. For larger discounts, consider our [Annual Pool of Funds offering](#).

What data analytics capabilities does New Relic provide?

New Relic is built for [Full-Stack Observability](#). It links all relevant data so that you get the whole picture of everything that enables your systems to deliver value to your customers. To learn more, see [Introduction to New Relic](#).

Can I add additional, external data to New Relic?

Yes, New Relic's Telemetry Data Platform is the single source of truth for all your operational data, regardless of where it is coming from, empowering you to ask and answer any question. New Relic enables you to explore your operational data through easy-to-build charts and dashboards, including support for visualizing Prometheus data in Grafana.

New Relic provides over 350 out-of-the-box integrations for ingesting data from open source tools, such as Prometheus, Telegraf, FluentD, and Logstash, in addition to New Relic's instrumentation agents. Currently, New Relic provides exporters for OpenTelemetry's Go and Java SDKs, and we plan to add more in the future. New Relic's programmable platform enables you to build New Relic One apps to connect system performance to unique business needs, such as business KPIs and customer engagement. To learn more, see [Get Data into New Relic](#).

What is the data retention period for my data in New Relic?

Data retention varies by your Full-Stack Observability support tier. Standard tier accounts have between 8 and 395 days (about 13 months) of retention, depending on which capabilities of the New Relic platform you're using. This level of data retention ensures that you can explore your data and experience the value of New Relic. Beyond that, you can purchase additional retention to ensure it suits your business needs. For more details, see [Manage data retention](#).

Can I extract data from New Relic?

Exporting data from New Relic is available via New Relic's API or via JSON code provided with each chart, graph and dashboard. [Find out more](#) about exporting New Relic data.

How can I track my New Relic usage?

To create detailed queries of your usage, and get notifications when you are close to hitting certain usage levels, see [Query usage data](#).

How does New Relic ensure security and privacy of my data?

To learn about how New Relic ensures security, see [New Relic Security](#). For a list of compliance certifications, see [Compliance certifications](#)

How much does New Relic cost?

If you are an existing New Relic customer, see [New Relic One billing](#). If you are new to New Relic, refer to [New Relic pricing](#). For more details, contact [New Relic Sales](#) or see our [Pricing and billing FAQs](#). You can also [sign up](#) for a perpetually free account that gives 100 GB of data ingest per month. You'll get access to all of New Relic's features, including a free license for Full-Stack Observability and Applied Intelligence.

Partner contact information

AWS Alliances at New Relic, aws@newrelic.com