

---

## Welcome to the AWS/Dynatrace Preview for AWS DevOps Agent

This document explains how to integrate AWS DevOps Agent with Dynatrace. After completing the steps in this guide, Dynatrace will initiate AWS DevOps Agent investigations for AWS-related Dynatrace problems (for example, issues running on AWS infrastructure). AWS DevOps Agent will analyze these and provide insights back to Dynatrace.

During the PoC phase, the Dynatrace AWS DevOps Agent integration consists of Dynatrace Workflows (for outbound and inbound communication) and a Dynatrace dashboard to explore AWS DevOps Agent investigation and mitigation results.

### Prerequisites

- Dynatrace SaaS
- DPS contract
- Dynatrace Workflows actor must have the following permissions:
  - `storage:events:write`
  - `storage:events:read`
  - `storage:buckets:read`
  - `storage:bizevents:read`
  - `storage:entities:read`
  - `app-engine:apps:run`
  - `app-engine:functions:run`
  - `automation:workflows:read`
  - `automation:workflows:run`
  - `environment:roles:viewer`
  - `document:documents:write`
  - `document:documents:read`

### Assets

Download relevant integration assets from [link](#).

### Remarks

- Workflows, bizevents, and events incur Dynatrace license costs. Price details are available in the Dynatrace rate cards.
- Additionally, AWS DevOps Agent will also query Dynatrace, incurring costs.

---

## Overview

1. When Dynatrace raises a problem, it triggers a workflow that queues the problem for further analysis.
2. A separate workflow runs every minute to check if there are any queued problems where Davis AI analysis has been completed.
3. Unprocessed problems are evaluated for their relevance to AWS. If deemed relevant, an AWS DevOps Agent investigation is initiated.
4. Based on the investigation, AWS DevOps Agent generates events. An investigation summary response is associated with the original problem.
5. Whenever AWS DevOps completes an investigation task, it can create mitigation action recommendations.
6. Similar to step 4, a mitigation summary response is associated with the problem.

### Step 1: Set up AWS DevOps Agent in your AWS account

Before proceeding, complete the AWS setup documentation. For the Dynatrace AWS DevOps Agent integration, you'll need the webhook details from the AWS setup:

#### // Example webhook details:

```
{
  "webhookId": "b12043af-86ce-4c1e-99e5-8130eb6bbe80",
  "webhookSecret": "aaaZdT1n16H/eeU+zXI7N97yXU+34KFitRAGyfvZBm1=",
  "webhookUrl":
  ↪ "https://app.aidevops.aws.com/d1432fe6-08b3-4fc2-b5f4-e75ad27bbb82/investigation/c5f
}
```

Store the `webhookSecret` in **Dynatrace Credential Vault**:

1. Open **Dynatrace Credential Vault** and select **Add new credential**.
2. Set the following values:
  - Credential type: Token
  - Credential scope: App Engine
  - Enable **Allow access without app context**
  - Token: value from `webhookSecret` property
3. Save and reopen the entry to retrieve its id (for example, `CREDENTIALS_VAULT-0C4FCF561A3AA13C`).  
You'll need this in step 2.1.

---

## Step 2: Configure Dynatrace integration Workflows

Dynatrace can trigger AWS DevOps Agent investigations and receive the results. It can also request mitigation steps for completed investigations.

### 2.1 Dynatrace triggers AWS DevOps Agent investigations

1. Open `::app-workflows::` and install `01_Queue potentially AWS-relevant problems.yaml`, `02_Dispatch pending analysis.yaml`, and `03_Check eligibility and invoke AWS DevOps Agent.yaml`.

These workflows prepare `::app-problems::` to trigger AWS DevOps Agent investigations by calling the AWS DevOps Agent webhook.

2. Update the `03_Check eligibility and invoke AWS DevOps Agent` workflow configuration:

- Open **Workflow options > Workflow input and result/Default Input**.
- Set:
  - `credentialVaultId` to the `credential id` created earlier.
  - `awsDevOpsAgentEndpoint` to the `webhookUrl` provided by AWS.

**//Example default input:**

```
{
  "credentialVaultId": "CREDENTIALS_VAULT-0C4FCF561A3AA13C",
  "awsDevOpsAgentEndpoint":
  ↪ "https://app.aidevops.aws.com/d1432fe6-08b3-4fc2-b5f4-e75ad27bbb82/investig
}
```

3. Deploy all workflows and make them all public.

### Further detailed information on the workflows you just installed

- `01_Queue potentially AWS-relevant problems` triggers on the arrival of a problem of type `Error`, `Slowdown`, or `Availability` and in active status. Queues the problem for further analysis.
- `02_Dispatch pending analysis` runs every 1 minute. The trigger interval can be adapted to your needs. It picks up the queued problems and forwards these for processing to `03_Check eligibility and invoke AWS DevOps Agent`.

- 
- 03\_Check eligibility and invoke AWS DevOps Agent is invoked by 02\_Dispatch pending analysis. The Dynatrace Workflow evaluates whether the issue is potentially related to AWS. If the issue is determined to be AWS-related, an AWS DevOps Agent investigation is initiated; otherwise, the workflow concludes.

## 2.2 Allow workflows to communicate with AWS DevOps Agent endpoint

By default, workflows cannot access the Internet. To enable this, navigate to **Settings > General > External requests** and add a **New host pattern** to allow communication with the AWS DevOps Agent webhook.

Exemplary allowlist entry, e.g., app.aidevops.aws.com

## 2.3 AWS DevOps Agent report investigation results to Dynatrace

1. In `::app-workflows::`, install `04_Handle AWS DevOps Agent responses.yaml`. This workflow processes AWS DevOps Agent results sent to the Dynatrace generic events ingest endpoint.
2. AWS DevOps Agent results (type: `investigation_summary`) will appear in Dynatrace within 5–15 minutes, but be aware that investigations may also take longer.

### Query example (DQL)

```
fetch dt.davis.events
| filter in(event.type,"CUSTOM_ANNOTATION")
| filter in(problem.id, "TODO Add your problem id here")
| sort event.end desc
| limit 100
```

## 2.4 Dynatrace asks AWS DevOps Agent for mitigation actions

1. Install the `05_Handle AWS DevOps Agent investigation completed responses.yaml` workflow. This workflow runs whenever an AWS DevOps Agent investigation concludes. It will trigger the mitigation action process in AWS DevOps Agent.
2. Make sure to configure **Workflow options > Workflow input and result/Default Input** as in step 2.1.

If you want to disable the workflow `05_Handle AWS DevOps Agent investigation completed responses` at a later stage, disable the trigger action.

---

## 2.5 AWS DevOps Agent report mitigation action results to Dynatrace

1. Install `06_Handle AWS DevOps Agent mitigation responses.yaml`. This workflow processes AWS DevOps Agent results. The functionality is similar to step 2.3.

### Step 3: Install the Dashboard

Use `::app-dashboards::` to upload `07_Display AWS DevOps Agent investigation results.json`. This dashboard provides an overview of AWS DevOps Agent investigation and mitigation results.

### Step 4: Explore AWS DevOps Agent response with dashboards or troubleshooting notebooks

Use the dashboard from step 3 to monitor investigation and mitigation results.

Next to the out-of-the-box dashboard, you can also navigate to the **Troubleshooting** section in `::app-problems::`. The feature is disabled by default because it may result in the creation of many notebooks. For every problem shared with AWS DevOps Agent, a notebook will be created.

You can enable this feature by modifying the default input of the workflow `04_Handle AWS DevOps Agent responses` installed in step 2.3.

```
//Example default input:
{
  "createTSGNotebook": true
}
```

Additionally, the default input of the workflow `06_Handle AWS DevOps Agent mitigation responses` requires changes.

```
//Example default input:
{
  "updateTSGNotebook": true
}
```

For every AWS DevOps Agent response, a troubleshooting notebook will be automatically created and attached to the problem that triggered the AWS DevOps Agent investigation. The notebook is updated with mitigation recommendations.

---

## Advanced usage of the 03\_Check eligibility and invoke AWS DevOps Agent workflow

The workflow 03\_Check eligibility and invoke AWS DevOps Agent you installed in Step 2.1 can be executed in different modes:

1. By default, it is automatically triggered when an AWS-related Dynatrace problems occur.
2. You can also run it on-demand to initiate AWS DevOps Agent investigations for existing problems.
3. If you are interested in what is sent to AWS DevOps Agent you can also run a workflow in dryRun mode.
4. For testing the Dynatrace to AWS DevOps Agent connectivity, run a simple connection test.

### Triggering an AWS DevOps Agent investigation for past problems

To manually trigger an AWS DevOps Agent investigation for a past problem, follow these steps:

1. Go to the 03\_Check eligibility and invoke AWS DevOps Agent workflow.
2. Select **Run**.
3. In the **Input** section, ensure you add or modify the following attributes as needed:

```
{
  "problemId": "4364692318491551794_1762551360000V2", // Replace this
  ↪ with the actual Problem ID
  "fromQuery": "now()-30d", // Adjust the query time range if the
  ↪ problem occurred more than 30 days ago
  "toQuery": "now()" // Optimize the query range for efficient querying
}
```

If a problem occurred more than 30 days ago, adjust the query timeframes.

{{#callout}} Note that AWS DevOps Agent does not support multiple investigations for the same problem. Once a problem has been investigated in an AgentSpace, it cannot be investigated again. {{/callout}}

### Trigger an AWS DevOps Agent investigation regardless of its AWS affinity

By default, the workflow checks if a problem relates to AWS entities, for example, it runs on AWS infrastructure or interacts with resources operated in AWS. By changing the workflow configuration, you can trigger an AWS DevOps Agent investigation even if no AWS associations are detected.

Change the **Workflow options > Workflow input and result/Default Input** configuration:

---

```
{
...
  "minAWSResources": 0,
...
}
```

### See what Dynatrace sends to AWS DevOps Agent

1. Go to the 03\_Check eligibility and invoke AWS DevOps Agent workflow.
2. Select **Run**.
3. In the **Input** section, ensure you add or modify the following attributes as needed:

```
{
  "dryRun": true,
  "problemId": "4364692318491551794_1762551360000V2", // Replace this
    ↳ with the actual Problem ID
  ...
}
```

Once you run the workflow, explore the logs to see what would have been sent to AWS DevOps Agent.

{{#callout}} In the dryRun mode, no data is exchanged. {{/callout}}

### Test Dynatrace to AWS DevOps Agent connectivity

1. Go to the 03\_Check eligibility and invoke AWS DevOps Agent workflow.
2. Select **Run**.
3. In the **Input** section, ensure you add or modify the following attributes as needed:

```
{
  "testConnectivity": true
  ...
}
```

Once you run the workflow, check the execution result of the workflow action **start\_aws\_aidevops\_investigation**  
> **Result**.

It should be similar to:

```
status: 200
eventId: test
```

---

```
success: true
awsdevopsagent: {
  "body": "{}",
  "status": 200,
  "statusText": "OK"
}
```

## Troubleshooting

### Not receiving AWS DevOps Agent responses

Verify that the user who created the Dynatrace OAuth Client has the `openpipeline:events:ingest` permission. To check this, navigate to **Identity & Access Management > Effective Policies**. If the user does not have this permission, add it to their user. Once the permission is granted, you will start receiving responses from the AWS DevOps Agent.