Red Hat

Observe

Ansible Automation Platform 2.6

May 12, 2026

Contents

1 Observe

Ensure system health and efficiency through monitoring

Monitor your deployment to maintain system health and identify performance issues before they affect operations. Track application metrics and system resources to understand workload patterns and optimize capacity allocation.

Monitoring your deployment helps you to:

- **Track application performance**: Monitor job status, system performance, and event processing to identify bottlenecks and understand how jobs impact your deployment.
- **Manage system resources**: Monitor CPU, memory, and disk performance to prevent resource exhaustion and maintain optimal system responsiveness.
- **Optimize capacity allocation**: Use monitoring data to adjust instance capacity settings and balance workloads across available resources based on actual usage patterns.

How Ansible Automation Platform supports monitoring

Ansible Automation Platform provides metrics endpoints for job status and subsystem performance data. Access these endpoints to view job output processing, scheduling, and other operational metrics.

Metrics for monitoring automation controller application

For application level monitoring, automation controller provides Prometheus-style metrics on an API endpoint `/api/v2/metrics`. Use these metrics to check data about job status and subsystem performance, such as for job output processing or job scheduling.

The metrics endpoint includes descriptions of each metric. Metrics of particular interest for performance include:

- `awx_status_total`
    - Current total of jobs in each status. Helps correlate other events to activity in system.
    - Can check upticks in errored or failed jobs.
- awx_instance_remaining_capacity

    - Amount of capacity remaining for running additional jobs.

- `callback_receiver_event_processing_avg_seconds`
    - colloquially called “job events lag”.
    - Running average of the lag time between when a task occurred in ansible and when the user is able to see it. This indicates how far behind the callback receiver is in processing events. When this number is very high, users can consider scaling up the control plane or using the capacity adjustment feature to reduce the number of jobs a control node controls.
- `callback_receiver_events_insert_db`
    - Counter of events that have been inserted by a node. Can be used to calculate the job event insertion rate over a given time period.
- `callback_receiver_events_queue_size_redis`
    - Indicator of how far behind callback receiver is in processing events. If too high, Redis can cause the control node to run out of memory (OOM).

System level monitoring

Monitoring CPU and memory is vital since instance capacity management doesn't introspect host resource usage. Automation impact varies by playbook; cloud modules process on the execution node, while native modules like "yum" perform work on target hosts, leaving the node waiting for results.

If CPU or memory usage is very high, consider lowering the capacity adjustment (available on the instance detail page) on affected instances in the automation controller. This limits how many jobs are run on or controlled by this instance.

Monitor the disk I/O and use of your system. The manner in which an automation controller node runs Ansible and caches output on the file system, and eventually saves it in the database, creates high levels of disk reads and writes. Identifying poor disk performance early can help prevent poor user experience and system degradation.

Related information
Metrics
Automation analytics and Red Hat Lightspeed for Red Hat Ansible Automation Platform

Generate consumption-based billing reports with the `metrics-utility`

The Ansible Automation Platform metrics utility tool (`metrics-utility`) is a command-line utility that is installed on a system containing an instance of automation controller.

When installed and configured, `metrics-utility` gathers billing-related metrics from your system and creates a consumption-based billing report. The `metrics-utility` tool is especially suited for users who have multiple managed hosts and want to use consumption-based billing. After a report is generated, it is deposited in a target location that you specify in the configuration file.

`metrics-utility` collects two types of data from your system: configuration data and reporting data.

The configuration data includes the following information:

- Version information for automation controller and for the operating system
- Subscription information
- The base URL

The reporting data includes the following information:

- Job name and ID
- Hostname
- Inventory name
- Organization name
- Project name
- Success or failure information
- Report date and time

To ensure that `metrics-utility` continues to work as configured, clear your report directories of outdated reports regularly.

Configure the `metrics-utility`

Configure the `metrics-utility` to gather and report usage data for your Ansible Automation Platform, both on Red Hat Enterprise Linux and OpenShift Container Platform.

Configure the `metrics-utility` on Red Hat Enterprise Linux

You can configure the `metrics-utility` on a Red Hat Enterprise Linux system to gather and report usage metrics for automation controller.

Before you begin

- **Subscription:** An active Ansible Automation Platform subscription.
- **Installation:** The `metrics-utility` tool is included by default with the Ansible Automation Platform installation on the automation controller node. No separate installation is required.
- **User privileges:** You must be logged in as the `root` user or the `awx` user to run the `metrics-utility` tool.

IMPORTANT:
The `metrics-utility` requires read access to `/etc/tower/SECRET_KEY` to function correctly. Attempting to run this utility as a standard user (non-root or non-awx) results in a `PermissionError` and execution failure.

The following procedure gathers the relevant data and generate a CCSP report containing your usage metrics. You can configure these commands as `cron` jobs to ensure they run at the beginning of every month. See How to schedule jobs using the Linux `cron` utility for more on configuring using the cron syntax.

Procedure

1. Create two scripts in your user's home directory to set correct variables to ensure that `metrics-utility` gathers all relevant data.
   a. In `/home/my-user/cron-gather`:

```
#!/bin/sh


# Specify the following variables to indicate where the report is
deposited in your file system

export METRICS_UTILITY_SHIP_TARGET=directory

export METRICS_UTILITY_SHIP_PATH=/awx_devel/awx-dev/metrics-utility/
shipped_data/billing


# Run the following command to gather and store the data in the provided
SHIP_PATH directory:

metrics-utility gather_automation_controller_billing_data --ship --
until=10m
```

   b. In `/home/my-user/cron-report`:

```
#!/bin/sh


# Specify the following variables to indicate where the report is
deposited in your file system

export METRICS_UTILITY_SHIP_TARGET=directory

export METRICS_UTILITY_SHIP_PATH=/awx_devel/awx-dev/metrics-utility/
shipped_data/billing


# Set these variables to generate a report:

export METRICS_UTILITY_REPORT_TYPE=CCSPv2

export METRICS_UTILITY_PRICE_PER_NODE=11.55 # in USD

export METRICS_UTILITY_REPORT_SKU=MCT3752MO

export METRICS_UTILITY_REPORT_SKU_DESCRIPTION="EX: Red Hat Ansible
Automation Platform, Full Support (1 Managed Node, Dedicated, Monthly)"

export METRICS_UTILITY_REPORT_H1_HEADING="CCSP Reporting <Company>:
ANSIBLE Consumption"

export METRICS_UTILITY_REPORT_COMPANY_NAME="Company Name"

export METRICS_UTILITY_REPORT_EMAIL="email@email.com"

export METRICS_UTILITY_REPORT_RHN_LOGIN="test_login"

export METRICS_UTILITY_REPORT_COMPANY_BUSINESS_LEADER="BUSINESS LEADER"

export METRICS_UTILITY_REPORT_COMPANY_PROCUREMENT_LEADER="PROCUREMENT
LEADER"


# Build the report

metrics-utility build_report
```

2. To ensure that these files are executable, run:

   `chmod a+x /home/my-user/cron-gather /home/my-user/cron-report`

3. To open the `cron` file for editing, run:

   `crontab -e`

4. To configure the run schedule, add the following parameters to the end of the file and specify how often you want `metrics-utility` to gather information and build a report using cron syntax. In the following example, the `gather` command is configured to run every hour at 00 minutes. The `build_report` command is configured to run on the second day of each month at 4:00 AM.

   `0 */1 * * * /home/my-user/cron-gather`

   `0 4 2 * * /home/my-user/cron-report`

5. Save and close the file.

Result

Use the following verification steps to ensure correct configuration:

1. To confirm that your `cron` job entries have been saved correctly, run:

```
crontab -l
```

2. Inspect the `cron` log to verify that the `cron` daemon is executing the commands and that `metrics-utility` is producing output:

```
cat /var/log/cron
```

For reference, see the following example output:

```
May  8 09:45:03 ip-10-0-6-23 CROND[51623]: (root) CMDOUT (No billing data for
month: 2024-04)

May  8 09:45:03 ip-10-0-6-23 CROND[51623]: (root) CMDEND (metrics-utility
build_report)

May  8 09:45:19 ip-10-0-6-23 crontab[51619]: (root) END EDIT (root)

May  8 09:45:34 ip-10-0-6-23 crontab[51659]: (root) BEGIN EDIT (root)

May  8 09:46:01 ip-10-0-6-23 CROND[51688]: (root) CMD (metrics-utility
gather_automation_controller_billing_data --ship --until=10m)

May  8 09:46:03 ip-10-0-6-23 CROND[51669]: (root) CMDOUT (/tmp/9e3f86ee-
c92e-4b05-8217-72c496e6ffd9-2024-05-08-093402+0000-2024-05-08-093602+0000-0.ta
r.gz)

May  8 09:46:03 ip-10-0-6-23 CROND[51669]: (root) CMDEND (metrics-utility
gather_automation_controller_billing_data --ship --until=10m)

May  8 09:46:26 ip-10-0-6-23 crontab[51659]: (root) END EDIT (root)
```

The generated report will have the default name `CCSP-<YEAR>-<MONTH>.xlsx` and is saved in the ship path that you specified in step 1a.

NOTE:
Time and date might vary depending on how your configure the run schedule.

Configure the `metrics-utility` on OpenShift Container Platform from the Ansible Automation Platform operator

The `metrics-utility` is a command-line tool that collects and reports metrics from your OpenShift Container Platform cluster to your automation controller instance.

`metrics-utility` is included in the OpenShift Container Platform image beginning with version 4.12, 4.512, and 4.6. If your system does not have `metrics-utility` installed, update your OpenShift image to the latest version.

Complete the following steps to configure the run schedule for `metrics-utility` on OpenShift Container Platform using the Ansible Automation Platform operator:

Create a ConfigMap in the OpenShift UI YAML view

Learn how to create a ConfigMap in the OpenShift UI YAML view to inject configuration data for the `metrics-utility` cronjobs.

Before you begin

- A running OpenShift cluster
- An operator-based installation of Ansible Automation Platform on OpenShift Container Platform.

NOTE:
`metrics-utility` runs as indicated by the parameters you set in the configuration file. You cannot run the utility manually on OpenShift Container Platform.

To inject the `metrics-utility` cronjobs with configuration data, and create a ConfigMap in the OpenShift UI YAML view:

Procedure

1. From the navigation panel, select **ConfigMaps**.
2. Click **Create ConfigMap**.
3. On the next screen, select the YAML view tab.
4. In the YAML field, enter the following parameters with the appropriate variables set:

```
 apiVersion: v1

kind: ConfigMap

metadata:

  name: automationcontroller-metrics-utility-configmap

data:

  METRICS_UTILITY_SHIP_TARGET: directory

  METRICS_UTILITY_SHIP_PATH: /metrics-utility

  METRICS_UTILITY_REPORT_TYPE: CCSPv2

  METRICS_UTILITY_PRICE_PER_NODE: '11' # in USD

  METRICS_UTILITY_REPORT_SKU: MCT3752MO

  METRICS_UTILITY_REPORT_SKU_DESCRIPTION: "EX: Red Hat Ansible Automation
Platform, Full Support (1 Managed Node, Dedicated, Monthly)"

  METRICS_UTILITY_REPORT_H1_HEADING: "CCSP Reporting <Company>: ANSIBLE
Consumption"

  METRICS_UTILITY_REPORT_COMPANY_NAME: "Company Name"

  METRICS_UTILITY_REPORT_EMAIL: "email@email.com"

  METRICS_UTILITY_REPORT_RHN_LOGIN: "test_login"

  METRICS_UTILITY_REPORT_COMPANY_BUSINESS_LEADER: "BUSINESS LEADER"

  METRICS_UTILITY_REPORT_COMPANY_PROCUREMENT_LEADER: "PROCUREMENT LEADER"
```

5. Click **Create**.

Result

- To verify that the ConfigMap was created and `metrics-utility` is installed, select **ConfigMap** from the navigation panel and search for your ConfigMap in the list.

Deploy automation controller

automation controller includes a `metrics-utility` cronjob that gathers usage information and generates a report at specified intervals.

To deploy automation controller and specify variables for how often `metrics-utility` gathers usage information and generates a report, use the following procedure:

Procedure

1. From the navigation panel, select **Installed Operators**.
2. Select **Ansible Automation Platform**.
3. In the Operator details, select the automation controller tab.
4. Click **Create automation controller**.

5. Select the YAML view option. The YAML now shows the default parameters for automation controller. The relevant parameters for `metrics-utility` are the following:

Parameter	Variable
`metrics_utility_enabled`	True.
`metrics_utility_cronjob_gather_schedule`	`@hourly` or `@daily`.
`metrics_utility_cronjob_report_schedule`	`@daily` or `@monthly`.

6. Find the `metrics_utility_enabled` parameter and change the variable to true.
7. Find the `metrics_utility_cronjob_gather_schedule` parameter and enter a variable for how often the utility should gather usage information (for example, `@hourly` or `@daily`).
8. Find the `metrics_utility_cronjob_report_schedule` parameter and enter a variable for how often the utility generates a report (for example, `@daily` or `@monthly`).
9. Click **Create**.

Configure the `metrics-utility` on a manual containerized installation of Ansible Automation Platform

The `metrics-utility` tool generates performance metrics and reports for Ansible Automation Platform installations.

`metrics-utility` is included in the OpenShift Container Platform image beginning with version 4.12, 4.512, and 4.6. If your system does not have `metrics-utility` installed, update your OpenShift image to the latest version.

Use the following steps to configure `metrics-utility` on a manual containerized installation Ansible Automation Platform:

1. Enable and configure `metrics-utility` in the inventory file.
2. Apply your `metrics-utility` configuration.
3. Verify the `systemctl` timer.
4. Verify the data collection.
5. Locate the generated reports.

NOTE:
You must have an active Ansible Automation Platform subscription

Minimum resource requirements

Using the `metrics-utility` tool on a containerized installation of Ansible Automation Platform requires the following resources:

- CPU: 1 dedicated CPU core
  - 100% of 1 core used during execution
- Memory:
  - Minimum: 256 MB RAM (supports up to ~10,000 job host summaries)
  - Recommended: 512 MB RAM (standard deployments)
  - Large-scale: 1 GB RAM (supports up to ~100,000 job host summaries)

    NOTE:
    Memory requirements scale with the number of hosts and jobs processed.

- Execution time: Report generation typically completes within 10-30 seconds, depending on data volume

Enable and configure the `metrics-utility` in the inventory file

Modify your Ansible Automation Platform inventory file to enable and configure `metrics-utility`.

Procedure

1. Modify your inventory file to enable `metrics-utility` container deployment by adding the following line under the `[automationcontroller]` section:

   ```
   metrics_utility_enabled=true
   ```

   This setting instructs the installation program to create and configure two dedicated `automation-controller-metrics-utility` containers as part of your Ansible Automation Platform deployment. One of these containers is used to collect the data, and the other is used to build the report. If your Ansible Automation Platform deployment has already been configured, re-run the installation script to activate the container.
2. Configure the reporting parameters by adding the `metrics_utility_extra_settings` variable. This variable controls where reports are saved, what they contain, and other metadata.

```
metrics_utility_extra_settings=[
{"setting": "METRICS_UTILITY_SHIP_TARGET", "value": "directory"},
{"setting": "METRICS_UTILITY_SHIP_PATH", "value": "~/aap/controller/data/metrics/"},
{"setting": "METRICS_UTILITY_REPORT_TYPE", "value": "CCSPv2"},
{"setting": "METRICS_UTILITY_PRICE_PER_NODE", "value": "100"},
{"setting": "METRICS_UTILITY_REPORT_COMPANY_NAME", "value": "My Company Inc"},
{"setting": "METRICS_UTILITY_REPORT_EMAIL", "value": "admin@mycompany.com"},
{"setting": "METRICS_UTILITY_REPORT_SKU", "value": "MCT3752MO"}]
```

3. Optional: Override the default data gathering schedule by adding the following variables with your `systemd timer` expressions:

NOTE:
`systemd timer` expressions differ from `cron` expressions.

```
# Gathers data every 30 minutes
metrics_utility_cronjob_gather_schedule=*:0/30
# Generates the report at midnight on the 2nd of the month
metrics_utility_cronjob_report_schedule=*-*-02 00:00:00
```

Apply your `metrics-utility` configuration

If you are running `metrics-utility` on a new installation, you do not need to take any additional actions to apply your configuration.

If you are applying your `metrics-utility` configuration to an existing deployment, you must re-run the Ansible Automation Platform installer script. Re-running the script reads the updated inventory file, deploys the `automation-controller-metrics-utility` container, and creates the `systemd` user services and timers necessary to automate data collection and reporting. Use the following verification steps to ensure your `metrics-utility` configuration has been applied and is running correctly:

- Verify your `systemctl` timer.
- Verify data collection.
- Locate generated reports.

Result

1. Run the following command to verify that your `systemctl timer` job entries were saved correctly:

```
systemctl --user list-timers --no-pager | grep metrics-utility
```

Example output:

```
Wed 2025-08-13 10:45:00 IST 8min left Wed 2025-08-13 10:30:04 IST 6min ago
metrics-utility-build-report.timer metrics-utility-build-report.service

Wed 2025-08-13 10:45:00 IST 8min left Wed 2025-08-13 10:30:04 IST 6min ago
metrics-utility-gather.timer       metrics-utility-gather.service
```

2. Use the following command to verify data collection by inspecting the output logs of the services you are running:

```
systemctl --user status metrics-utility-gather.service
```

Example output:

```
metrics-utility-gather.service - Podman metrics-utility-gather.service

     Loaded: loaded (/home/aap/.config/systemd/user/metrics-utility-
gather.service; disabled; preset: disabled)

     Active: inactive (dead) since Wed 2025-08-13 10:00:06 IST; 5min ago

   Duration: 2.008s

TriggeredBy: ● metrics-utility-gather.timer

       Docs: man:podman-generate-systemd(1)

    Process: 1472847 ExecStart=/usr/bin/podman start metrics-utility-gather
(code=exited, status=0/SUCCESS)

    Process: 1472927 ExecStop=/usr/bin/podman stop -t 10 metrics-utility-
gather (code=exited, status=0/SUCCESS)

    Process: 1472937 ExecStopPost=/usr/bin/podman stop -t 10 metrics-utility-
gather (code=exited, status=0/SUCCESS)

   Main PID: 1472874 (code=exited, status=0/SUCCESS)

        CPU: 197ms



Aug 13 10:00:04 aap.example.org podman[1472847]: metrics-utility-gather

Aug 13 10:00:04 aap.example.org systemd[993]: Started Podman metrics-utility-
gather.service.

Aug 13 10:00:05 aap.example.org metrics-utility-gather[1472874]: 2025-08-13
09:00:05,806 INFO     [-] awx.main.analytics /tmp/3292ca44-3314-4f>

Aug 13 10:00:05 aap.example.org metrics-utility-gather[1472874]: /tmp/
3292ca44-3314-4f0b-b3f6-ba4a1e47a2b1-2025-08-13-083505+0000-2025-08-13-0>

Aug 13 10:00:05 aap.example.org metrics-utility-gather[1472874]: 2025-08-13
09:00:05,808 INFO     [-] awx.main.analytics /tmp/3292ca44-3314-4f>

Aug 13 10:00:05 aap.example.org metrics-utility-gather[1472874]: /tmp/
3292ca44-3314-4f0b-b3f6-ba4a1e47a2b1-2025-08-13-083505+0000-2025-08-13-0>

Aug 13 10:00:06 aap.example.org podman[1472912]: 2025-08-13 10:00:06.169271763
+0100 IST m=+0.019922418 container died 5dc8d5674f1d1745258530f>

Aug 13 10:00:06 aap.example.org podman[1472912]: 2025-08-13 10:00:06.187584135
+0100 IST m=+0.038234790 container cleanup 5dc8d5674f1d17452585>

Aug 13 10:00:06 aap.example.org podman[1472927]: metrics-utility-gather

Aug 13 10:00:06 aap.example.org podman[1472937]: metrics-utility-gather
```

3. Locate the generated reports. Reports are saved in the directory you specified in the `METRICS_UTILITY_SHIP_PATH` setting.
    - Path: Using the example provided in this document, the report path would be `/aap/controller/data/metrics/`.
    - Filename: The report name follows the format `CCSP-<YEAR>-<MONTH>.xlsx`. For example, a report generated for August, 2025 would be named `CCSP-2025-08.xlsx`.

Configure a monthly usage report

You can fetch a monthly report from Ansible Automation Platform to gather usage metrics and create a consumption-based billing report. To fetch a monthly report on Red Hat Enterprise Linux or on OpenShift Container Platform, use the following procedures:

Fetch a monthly report on Red Hat Enterprise Linux

Use the following procedure to fetch a monthly report on Red Hat Enterprise Linux:

- Run: `scp -r username@controller_host:$METRICS_UTILITY_SHIP_PATH/data/<YYYY>/<MM>/ /local/directory/`

Result

The system saves the generated report as `CCSP-<YEAR>-<MONTH>.xlsx` in the ship path that you specified.

Fetch a monthly report on OpenShift Container Platform from the Ansible Automation Platform Operator

Run a dedicated Ansible Playbook to fetch monthly usage reports directly from the OpenShift Container Platform persistent volume claim. This helps ensure accurate consumption-based billing and compliance.

Procedure

1. Use the following playbook to fetch a monthly consumption report for Ansible Automation Platform on OpenShift Container Platform:

> **NOTE:**
> To use this playbook, you must have the `kubernetes.core.k8s` collection installed on your machine.

```
# Requires Ansible and Kubernetes.core collection
- name: Copy directory from Kubernetes PVC to local machine
  hosts: "{{ host | default(omit) }}"

  vars:
    report_dir_path: "/mnt/metrics/reports/{{ year }}/{{ month }}/"
    data_files_dir_path: "/mnt/metrics/data/{{ year }}/{{ month }}/{{ day }}"

  tasks:
    - name: Create a temporary pod to access PVC data
      kubernetes.core.k8s:
        definition:
          apiVersion: v1
          kind: Pod
          metadata:
            name: temp-pod
            namespace: "{{ namespace_name }}"
          spec:
            containers:
              - name: busybox
                image: busybox
                command: ["/bin/sh"]
                args: ["-c", "sleep 3600"]  # Keeps the container alive for 1
hour
                volumeMounts:
                  - name: "{{ pvc }}"
                    mountPath: "/mnt/metrics"
            volumes:
              - name: "{{ pvc }}"
                persistentVolumeClaim:
                  claimName: automationcontroller-metrics-utility
            restartPolicy: Never
      register: pod_creation

    - name: Wait for both initContainer and main container to be ready
      kubernetes.core.k8s_info:
        kind: Pod
```

```
        namespace: "{{ namespace_name }}"
        name: temp-pod
      register: pod_status
      until: >
        pod_status.resources[0].status.containerStatuses[0].ready
      retries: 30
      delay: 10


    - name: Create a tarball of the directory of the report in the container
      kubernetes.core.k8s_exec:
        namespace: "{{ namespace_name }}"
        pod: temp-pod
        container: busybox
        command: tar czf /tmp/metrics.tar.gz -C "{{ report_dir_path }}" .
      register: tarball_creation


    - name: Create a tarball of the directory of the data files in the
container
      kubernetes.core.k8s_exec:
        namespace: "{{ namespace_name }}"
        pod: temp-pod
        container: busybox
        command: tar czf /tmp/data_files.tar.gz -C
"{{ data_files_dir_path }}" .
      register: tarball_creation_files


    - name: Copy the report tarball from the container to the local machine
      kubernetes.core.k8s_cp:
        namespace: "{{ namespace_name }}"
        pod: temp-pod
        container: busybox
        state: from_pod
        remote_path: /tmp/metrics.tar.gz
        local_path: "{{ local_dir }}/metrics.tar.gz"
      when: tarball_creation is succeeded


    - name: Copy the data files tarball from the container to the local
machine
```

```
      kubernetes.core.k8s_cp:
        namespace: "{{ namespace_name }}"
        pod: temp-pod
        container: busybox
        state: from_pod
        remote_path: /tmp/data_files.tar.gz
        local_path: "{{ local_dir }}/data_files.tar.gz"
      when: tarball_creation_files is succeeded

    - name: Ensure the local directory exists
      ansible.builtin.file:
        path: "{{ local_dir }}"
        state: directory
        mode: '0755'

    - name: Extract the report tarball on the local machine
      ansible.builtin.unarchive:
        src: "{{ local_dir }}/metrics.tar.gz"
        dest: "{{ local_dir }}"
        remote_src: true
        extra_opts: "--strip-components=1"
      when: tarball_creation is succeeded

    - name: Extract the data files tarball on the local machine
      ansible.builtin.unarchive:
        src: "{{ local_dir }}/data_files.tar.gz"
        dest: "{{ local_dir }}"
        remote_src: true
        extra_opts: "--strip-components=1"
        list_files: true
      register: unarchive_result
      when: tarball_creation_files is succeeded

    - name: Extract the extracted data files tarball on the local machine
      ansible.builtin.unarchive:
        src: "{{ local_dir }}/{{ item }}"
```

```
        dest: "{{ local_dir }}"
        remote_src: true
        extra_opts: "--strip-components=1"
      loop: "{{ unarchive_result.files }}"
      when: tarball_creation_files is succeeded
      ignore_errors: true  # noqa ignore-errors

    - name: Delete the temporary pod
      kubernetes.core.k8s:
        api_version: v1
        kind: Pod
        namespace: "{{ namespace_name }}"
        name: temp-pod
        state: absent
```

Configure the `metrics-utility` to run at specific times

You can configure `metrics-utility` to run at specified times and intervals. Run frequency is expressed in cronjobs. For more information on the cron utility, see How to schedule jobs using the Linux `Cron` utility.

To modify the run schedule on Red Hat Enterprise Linux and on OpenShift Container Platform, use one of the following procedures:

Procedure

1. From the command line, run:

   `crontab -e`

2. After the code editor has opened, update the `gather` and `build` parameters using cron syntax as shown below:

   `*/2 * * * * metrics-utility gather_automation_controller_billing_data --ship --until=10m`

   `*/5 * * * * metrics-utility build_report`

3. Save and close the file.

Modify the run schedule on OpenShift Container Platform from the Ansible Automation Platform operator

To adjust the execution schedule of the `metrics-utility` within your Ansible Automation Platform deployment running on OpenShift Container Platform, use the following procedure.

Procedure

1. From the navigation panel, select **Workloads › Deployments**.
2. On the next screen, select **automation-controller-operator-controller-manager**.
3. Beneath the heading **Deployment Details**, click the down arrow button to change the number of pods to zero. This pauses the deployment so you can update the running schedule.
4. From the navigation panel, select **Installed Operators**.
5. From the list of installed operators, select Ansible Automation Platform.
6. On the next screen, select the automation controller tab.
7. From the list displayed, select your automation controller instance.
8. On the next screen, select the `YAML` tab.
9. In the `YAML` file, find the following parameters and enter a variable representing how often `metrics-utility` should gather data and how often it should produce a report:

   `metrics_utility_cronjob_gather_schedule:`

   `metrics_utility_cronjob_report_schedule:`
10. Click **Save**.
11. From the navigation menu, select **Deployments** and then select **automation-controller-operator-controller-manager**.
12. Increase the number of pods to 1.
13. To verify that you have changed the `metrics-utility` running schedule successfully, you can take one or both of the following steps:
    a. Return to the `YAML` file and ensure that the previously described parameters reflect the correct variables.
    b. From the navigation menu, select **Workloads › Cronjobs** and ensure that your cronjobs show the updated schedule.

Specify where to store consumption-based reports

Supported storage is available for storing the raw data obtained by using the `metrics-utility gather_automation_controller_billing_data` command and storing the generated reports obtained by using the `metrics-utility build_report` command.

Apply the environment variables to this storage based on your Ansible Automation Platform installation.

Local disk

For an installation of Ansible Automation Platform on Red Hat Enterprise Linux, the default storage option is a local disk. Using an OpenShift deployment of OpenShift Container Platform, default storage is a path inside the attached Persistent Volume Claim.

- Set the environment variables for your target directory and your local disk path.

```
 # Set needed ENV VARs for gathering data and generating reports
export METRICS_UTILITY_SHIP_TARGET=directory
# Your path on the local disk
export METRICS_UTILITY_SHIP_PATH=/path_to_data_and_reports/...
```

Object storage with S3 interface

To use object storage with S3 interface, for example, with AWS S3, Ceph Object storage, or Minio, you must define environment variables for data gathering and report building commands and cronjobs.

- Set the environment variables for your S3 object storage path, name, endpoint, region, access key, and secret key.

```
################
export METRICS_UTILITY_SHIP_TARGET=s3
# Your path in the object storage
export METRICS_UTILITY_SHIP_PATH=path_to_data_and_reports/...

################
# Define S3 config
export METRICS_UTILITY_BUCKET_NAME=metricsutilitys3
export METRICS_UTILITY_BUCKET_ENDPOINT="https://s3.us-east-1.amazonaws.com"
# For AWS S3, define also a region
export METRICS_UTILITY_BUCKET_REGION="us-east-1"

################
# Define S3 credentials
export METRICS_UTILITY_BUCKET_ACCESS_KEY=<access_key>
export METRICS_UTILITY_BUCKET_SECRET_KEY=<secret_key>
```

Configure options for the Certified Cloud and Service Provider report (CCSPv2)

Additional configurations for data gathering and report building based on a report type. Apply the environment variables to each report type based on your Ansible Automation Platform installation.

CCSPv2

CCSPv2 is a report which shows the following information:

- Directly and indirectly managed node usage
- The content of all inventories
- Content usage

The primary use of this report is for partners under the CCSP program, but all customers can use it to obtain on-premise reporting showing managed nodes, jobs and content usage across their automation controller organizations.

Set the report type by using `METRICS_UTILITY_REPORT_TYPE=CCSPv2`.

Optional collectors for `gather` command

You can use the following optional collectors for the `gather` command:

- `main_jobhostsummary`
  - If present by default, this incrementally collects data from the `main_jobhostsummary` table in the automation controller database, containing information about jobs runs and managed nodes automated.
- `main_host`
  - This collects daily snapshots of the `main_host` table in the automation controller database and has managed nodes and hosts present across automation controller inventories.
- `main_jobevent`
  - This incrementally collects data from the `main_jobevent` table in the automation controller database and contains information about which modules, roles, and Ansible collections are being used.
- `main_indirectmanagednodeaudit`
  - This incrementally collects data from the `main_indirectmanagednodeaudit` table in the automation controller database and contains information about indirectly managed nodes.

```
# Example with all optional collectors

export
METRICS_UTILITY_OPTIONAL_COLLECTORS="main_host,main_jobevent,main_indirec
tmanagednodeaudit"
```

Optional sheets for `build_report` command

You can use the following optional sheets for the `build_report` command:

- `ccsp_summary`
  - This is a landing page specifically for partners under CCSP program. This report takes additional parameters to customize the summary page. For more information, see the following example:

```
export METRICS_UTILITY_PRICE_PER_NODE=11.55 # in USD

export METRICS_UTILITY_REPORT_SKU=MCT3752MO

export METRICS_UTILITY_REPORT_SKU_DESCRIPTION="EX: Red Hat Ansible
Automation Platform, Full Support (1 Managed Node, Dedicated, Monthly)"

export METRICS_UTILITY_REPORT_H1_HEADING="CCSP NA Direct Reporting
Template"

export METRICS_UTILITY_REPORT_COMPANY_NAME="Partner A"

export METRICS_UTILITY_REPORT_EMAIL="email@email.com"

export METRICS_UTILITY_REPORT_RHN_LOGIN="test_login"

export METRICS_UTILITY_REPORT_PO_NUMBER="123"

export METRICS_UTILITY_REPORT_END_USER_COMPANY_NAME="Customer A"

export METRICS_UTILITY_REPORT_END_USER_CITY="Springfield"

export METRICS_UTILITY_REPORT_END_USER_STATE="TX"

export METRICS_UTILITY_REPORT_END_USER_COUNTRY="US"
```

- `jobs`
  - This is a list of automation controller jobs launched. It is grouped by job template.
- `managed_nodes`
  - This is a deduplicated list of managed nodes automated by automation controller.
- `indirectly_managed_nodes`
  - This is a deduplicated list of indirect managed nodes automated by automation controller.
- `infrastructure_summary`
  - This additional tab summarizes the infrastructure taxonomy for indirect nodes in three levels:
    a. Infrastructure
    b. Device category
    c. Device type

> **NOTE:**
> Taxonomy mapping is dependent on the facts column in the raw data.

- `inventory_scope`
  - This is a deduplicated list of managed nodes present across all inventories of automation controller.
- `usage_by_organizations`

    - This is a list of all automation controller organizations with several metrics showing the organizations usage. This provides data suitable for doing internal chargeback.
- `usage_by_collections`
    - This is a list of Ansible collections used in a automation controller job runs.
- `usage_by_roles`
    - This is a list of roles used in automation controller job runs.
- `usage_by_modules`
    - This is a list of modules used in automation controller job runs.
- `managed_nodes_by_organization`
    - This generates a sheet per organization, listing managed nodes for every organization with the same content as the managed_nodes sheet.
- `data_collection_status`
    - This generates a sheet with the status of every data collection done by the `gather` command for the date range the report is built for.
    - To outline the quality of data collected, `data_collection_status` also lists:
        - Unusual gaps between collections (based on collection_start_timestamp)
        - Gaps in collected intervals (based on `since` compared to `until`)

```
# Example with all optional sheets

export
METRICS_UTILITY_OPTIONAL_CCSP_REPORT_SHEETS='ccsp_summary,jobs,mana
ged_nodes,indirectly_managed_nodes,inventory_scope,usage_by_organiz
ations,usage_by_collections,usage_by_roles,usage_by_modules,data_co
llection_status'
```

Filter reports by organization

When generating reports by using the metrics utility, you can filter the data by organization.

To filter your report so that only certain organizations are present, use this environment variable with a semicolon separated list of organization names.

`export METRICS_UTILITY_ORGANIZATION_FILTER="Organization 1;Organization 2"`

This renders only the data from these organizations in the built report. This filter currently does not have any effect on the following optional sheets:

- `usage_by_collections`
- `usage_by_roles`
- `usage_by_modules`

Select a date range for your CCSPv2 report

The default behavior of the CCSPv2 report is to build a report for the previous month. The following examples describe how to override this default behavior to select a specific date range for your report:

```
# Build report for a specific month
metrics-utility build_report --month=2025-03


# Build report for a specific date range, including the provided days
metrics-utility build_report --since=2025-03-01 --until=2025-03-31


# Build report for a last 6 months from a current date
metrics-utility build_report --since=6months


# Build report for a last 6 months from a current date overwriting an existing report
metrics-utility build_report --since=6months --force
```

Configure options in the `RENEWAL_GUIDANCE` report

The `RENEWAL_GUIDANCE` report provides historical usage from the HostMetric table, applying deduplication and showing real historical usage for renewal guidance purposes.

To generate this report, set the report type to `METRICS_UTILITY_REPORT_TYPE=RENEWAL_GUIDANCE`.

IMPORTANT:
This report is currently a tech preview solution. It is designed to provide more information than automation controller when built in the `awx-manage host_metric` command.

Storage and invocation

The `RENEWAL_GUIDANCE` report supports the use of only local disk storage to store the report results. This report does not have a gather data step. It reads directly from the controller HostMetric table, so it does not store any raw data under the `METRICS_UTILITY_SHIP_PATH`.

```
# All parameters the RENEWAL_GUIDANCE report needs
export METRICS_UTILITY_SHIP_TARGET=controller_db
export METRICS_UTILITY_REPORT_TYPE=RENEWAL_GUIDANCE
export METRICS_UTILITY_SHIP_PATH=/path_to_built_report/...

# Will generate report for 12 months back with ephemeral nodes being nodes
# automated for less than 1 month.
metrics-utility build_report --since=12months --ephemeral=1month
```

Generate reports to show ephemeral usage

The `metrics-utility` command-line tool can generate reports showing ephemeral usage of managed nodes.

The `RENEWAL_GUIDANCE` report has the capability to list additional sheets with ephemeral usage if the `-ephemeral` parameter is provided. Using the parameter `--ephemeral=1month`, you can define ephemeral nodes as any managed node that has been automated for a maximum of one month, then never automated again. Using this parameter, the total ephemeral usage of the 12-month period is computed as maximum ephemeral nodes used over all 1-month rolling date windows. This sheet is also added into the report.

```
# Will generate report for 12 months back with ephemeral nodes being nodes
# automated for less than 1 month.
metrics-utility build_report --since=12months --ephemeral=1month
```

Select a date range for your `RENEWAL_GUIDANCE` report

The default behavior of the `RENEWAL_GUIDANCE` report is to build a report for the current date. The following examples describe how to override this default behavior to select a specific date range for your report:

The `RENEWAL_GUIDANCE` report requires a `since` parameter as the parameter is not supported due to the nature of the `HostMetric` data and is always set to `now`. To override a report date

range that is already built, use parameter `-force` with the command. For more information, see the following examples:

```
# Build report for a specific date range, including the provided days

metrics-utility build_report --since=2025-03-01


# Build report for a last 12 months from a current date

metrics-utility build_report --since=12months


# Build report for a last 12 months from a current date overwriting an existing
report

metrics-utility build_report --since=12months --force
```

Original Certified Cloud and Service Provider (CCSP) report

`CCSP` is the original report format. It does not include many of the customization of CCSPv2, and the intention is only to use it for the CCSP partner program.

Optional collectors for `gather` command

You can use the following optional collectors for the `gather` command:

- `main_jobhostsummary`
    - If present by default, this collects the `main_jobhostsummary` table from the automation controller database, and has information about jobs runs and managed nodes automated.
- `main_host`
    - This collects daily snapshots of the `main_host` table from the automation controller database and has managed nodes/hosts present across automation controller inventories,
- `main_jobevent`
    - This collects the `main_jobevent` table from the automation controller database and has information about which modules, roles, and ansible collections are being used.
- `main_indirectmanagednodeaudit`

- This collects the `main_indirectmanagednodeaudit` table from the automation controller database and has information about indirectly managed nodes,

```
# Example with all optional collectors

export
METRICS_UTILITY_OPTIONAL_COLLECTORS="main_host,main_jobevent,main_indirectmanagednod
eaudit"
```

Optional sheets for `build_report` command

You can use the following optional sheets for the `build_report` command:

- `ccsp_summary`
  - This is a landing page specifically for partners under the CCSP program. It shows managed node usage by each automation controller organization.
  - This report takes additional parameters to customize the summary page. For more information, see the following example:

```
export METRICS_UTILITY_PRICE_PER_NODE=11.55 # in USD

export METRICS_UTILITY_REPORT_SKU=MCT3752MO

export METRICS_UTILITY_REPORT_SKU_DESCRIPTION="EX: Red Hat Ansible
Automation Platform, Full Support (1 Managed Node, Dedicated, Monthly)"

export METRICS_UTILITY_REPORT_H1_HEADING="CCSP Reporting <Company>:
ANSIBLE Consumption"

export METRICS_UTILITY_REPORT_COMPANY_NAME="Company Name"

export METRICS_UTILITY_REPORT_EMAIL="email@email.com"

export METRICS_UTILITY_REPORT_RHN_LOGIN="test_login"

export METRICS_UTILITY_REPORT_COMPANY_BUSINESS_LEADER="BUSINESS LEADER"

export METRICS_UTILITY_REPORT_COMPANY_PROCUREMENT_LEADER="PROCUREMENT
LEADER"
```

- `managed_nodes`
  - This is a deduplicated list of managed nodes automated by automation controller.
- `indirectly_managed_nodes`
  - This is a deduplicated list of indirect managed nodes automated by automation controller.
- `inventory_scope`
  - This is a deduplicated list of managed nodes present across all inventories of automation controller.

- `usage_by_collections`
  - This is a list of Ansible collections used in automation controller job runs.
- `usage_by_roles`
  - This is a list of roles used in automation controller job runs.
- `usage_by_modules`
  - This is a list of modules used in automation controller job runs.

```
# Example with all optional sheets

export
METRICS_UTILITY_OPTIONAL_CCSP_REPORT_SHEETS='ccsp_summary,managed_nodes,indirectly_m
anaged_nodes,inventory_scope,usage_by_collections,usage_by_roles,usage_by_modules'
```

Select a date range for your CCSP report

By default, the CCSPv2 report generates data for the previous calendar month. You can override this behavior by specifying a different month when you run the `metrics-utility build_report` command.

The default behavior of this report is to build a report for the previous month. The following examples describe how to override this default behavior to select a specific date range for your report:

```
# Builds report for a previous month

metrics-utility build_report


# Build report for a specific month

metrics-utility build_report --month=2025-03


# Build report for a specific month overwriting an existing report

metrics-utility build_report --month=2025-03 --force
```

How to deduplicate host data in reports

Deduplication changes how `metrics-utility` merges individual host records into countable managed nodes when building reports. Deduplication identifies identical hosts to ensure an accurate count of unique hosts.

`metrics-utility` tracks individual hosts based on their hostnames. It tracks any entries that use the same hostname as the host. Additional deduplication strategies are also available using the following environment variable: `METRICS_UTILITY_DEDUPLICATOR=` .

Deduplication in the `RENEWAL_GUIDANCE` report

The `RENEWAL_GUIDANCE` report uses deduplication to merge several entries for the same managed node into a single entry. This is important to provide accurate counts of managed nodes for subscription renewal.

The default value for `METRICS_UTILITY_DEDUPLICATOR=renewal` . This is the original method, which analyzes `host_name` , `ansible_host` , `ansible_product_serial` , and `ansible_machine_id` separately, and merges entries any duplicated items.

`METRICS_UTILITY_DEDUPLICATOR=renewal` applies deduplication in multiple iterations. It is limited by the `REPORT_RENEWAL_GUIDANCE_DEDUP_ITERATIONS` environment variable, which defaults to `3` .

You can also run `METRICS_UTILITY_DEDUPLICATOR` with the following environment variables:

- `METRICS_UTILITY_DEDUPLICATOR=renewal-hostname` . This is similar to `ccsp` , again preferring `ansible_host` over `host_name` when present. No other fields are considered.
- `METRICS_UTILITY_DEDUPLICATOR=renewal-experimental` . This is similar to `ccsp-experimental` , which first applies the hostname-based deduplication, then deduplicates again, merging when both of the serials match.

Deduplication in the CCSP or CCSPv2 reports

When generating the CCSP or CCSPv2 reports, you can set deduplication options to avoid counting the same managed node multiple times.

The default value for `METRICS_UTILITY_DEDUPLICATOR=ccsp` . This limits deduplication to hostnames only.

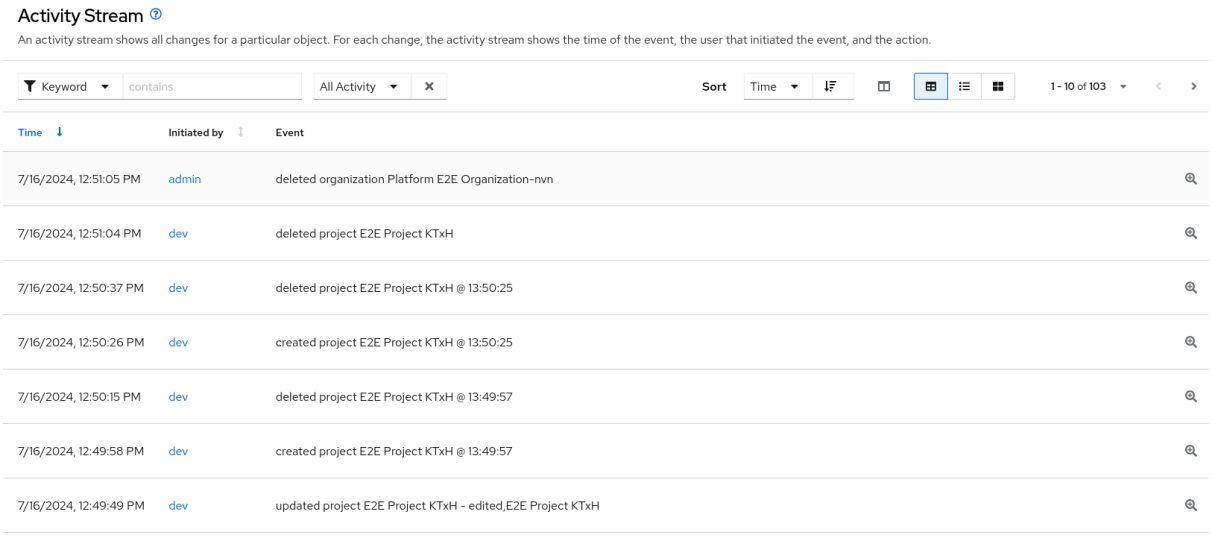
The `ansible_host` variable, from `main_host.variables` , is preferred over `host_name` , from `main_jobhostsummary` , when present.

You can also set `METRICS_UTILITY_DEDUPLICATOR=ccsp-experimental` . This setting merges entries when both their `ansible_product_serial` and `ansible_machine_id` facts are present and duplicated.

View activity streams for all resources

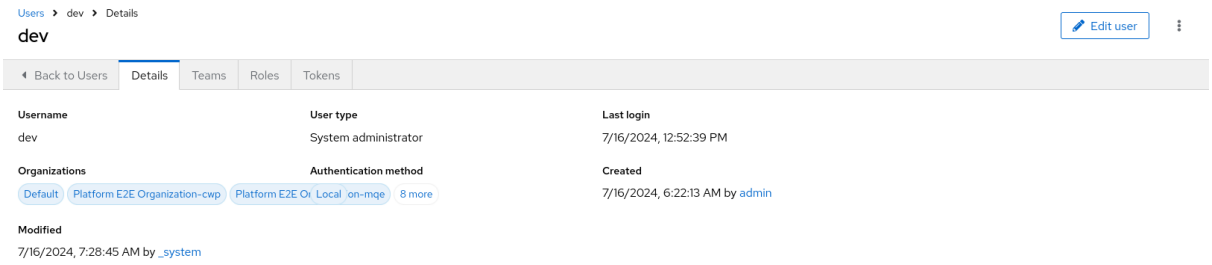
Track all changes made to any object in automation controller by reviewing the **Activity Stream**. You can audit the time, the user, and the specific action for every event.

- From the navigation panel, select **Automation Execution > Administration > Activity Stream**.



An Activity Stream shows all changes for a particular object. For each change, the Activity Stream shows the time of the event, the user that initiated the event, and the action. The information displayed varies depending on the type of event.

- Click the icon to display the event log for the change.



You can filter the Activity Stream by the initiating user, by system (if it was system initiated), or by any related object, such as a credential, job template, or schedule. The Activity Stream shows the Activity Stream for the entire instance. Most pages permit viewing an activity stream filtered for that specific object.

You can view the activity stream on any page by clicking the **Activity Stream** icon.

Understand and configure notifications

A Notification Type such as Email, Slack or a Webhook, is an instance of a Notification Template, and has a name, description and configuration defined in the Notification template.

The following include examples of details needed to add a notification template:

- A username, password, server, and recipients are needed for an Email notification template
- The token and a list of channels are needed for a Slack notification template
- The URL and Headers are needed for a Webhook notification template

When a job fails, a notification is sent using the configuration that you define in the notification template.

The following shows the typical flow for the notification system:

- You create a notification template to the `REST API` at the `/api/v2/notification_templates endpoint`, either through the API or through the UI.
- You assign the notification template to any of the various objects that support it (all variants of job templates, organizations and projects) and at the appropriate trigger level for which you want the notification (started, success, or error). For example, you might want to assign a particular notification template to trigger when Job Template 1 fails. In this case, you associate the notification template with the job template at `/api/v2/job_templates/n/notification_templates_error` API endpoint.
- You can set notifications on job start and job end. Users and teams are also able to define their own notifications that can be attached to arbitrary jobs.

Related information
Work with notifications

Notification hierarchy

Automation controller uses a hierarchical notification system where notification templates can be defined at various levels, and lower-level objects can inherit templates from their parent objects.

Notification templates inherit templates defined on parent objects, such as the following:

- Job templates use notification templates defined for them. Additionally, they can inherit notification templates from the project used by the job template, and from the organization that it is listed under.
- Project updates use notification templates defined on the project and inherit notification templates from the organization associated with it.
- Inventory updates use notification templates defined on the organization that it is listed under.
- Ad hoc commands use notification templates defined on the organization that the inventory is associated with.

Notification workflow

Automation controller can send notifications when jobs succeed or fail.

When a job succeeds or fails, the error or success handler pulls a list of relevant notifications. It then creates a notification object for each one, containing relevant details about the job and sends it to the destination. These include email addresses, slack channels, and SMS numbers.

These notification objects are available as related resources on job types (jobs, inventory updates, project updates), and also at `/api/v2/notifications`. You can also see what notifications have been sent from a notification template by examining its related resources.

If a notification fails, it does not impact the job associated with it or cause it to fail. The status of the notification can be viewed at its detail endpoint `/api/v2/notifications/<n>`.

Related information
Understand and configure notifications

Create a notification template

Use the following procedure to create a notification template.

Procedure

1. From the navigation panel, select **Automation Execution > Administration > Notifiers**.
2. Click **Add notifier**.
3. Complete the following fields:
    - **Name**: Enter the name of the notification.
    - **Description**: Enter a description for the notification. This field is optional.
    - **Organization**: Specify the organization that the notification belongs to.
    - **Type**: Choose a type of notification from the drop-down menu. For more information, see the Notification types section.
4. Click **Save notifier**.

Notification types

Automation controller supports multiple notification types that you can use to send notifications about job status and other events.

The following notification types are supported with automation controller:

- Email
- Grafana
- IRC
- Mattermost
- PagerDuty
- Rocket.Chat
- Slack
- Twilio
- Webhook

Each notification type has its own configuration and behavioral semantics. You might need to test them in different ways. Additionally, you can customize each type of notification down to a specific detail or a set of criteria to trigger a notification.

Related information
Create custom notifications
customize the text content of each Notification type on the notification form.
Email
Grafana
IRC
Mattermost
Pagerduty
PagerDuty system. This is the token that is given to automation controller. Then create a **Service** which provides an **Integration Key** that is also given to automation controller.
Rocket.Chat
Slack
Twilio
Webhook

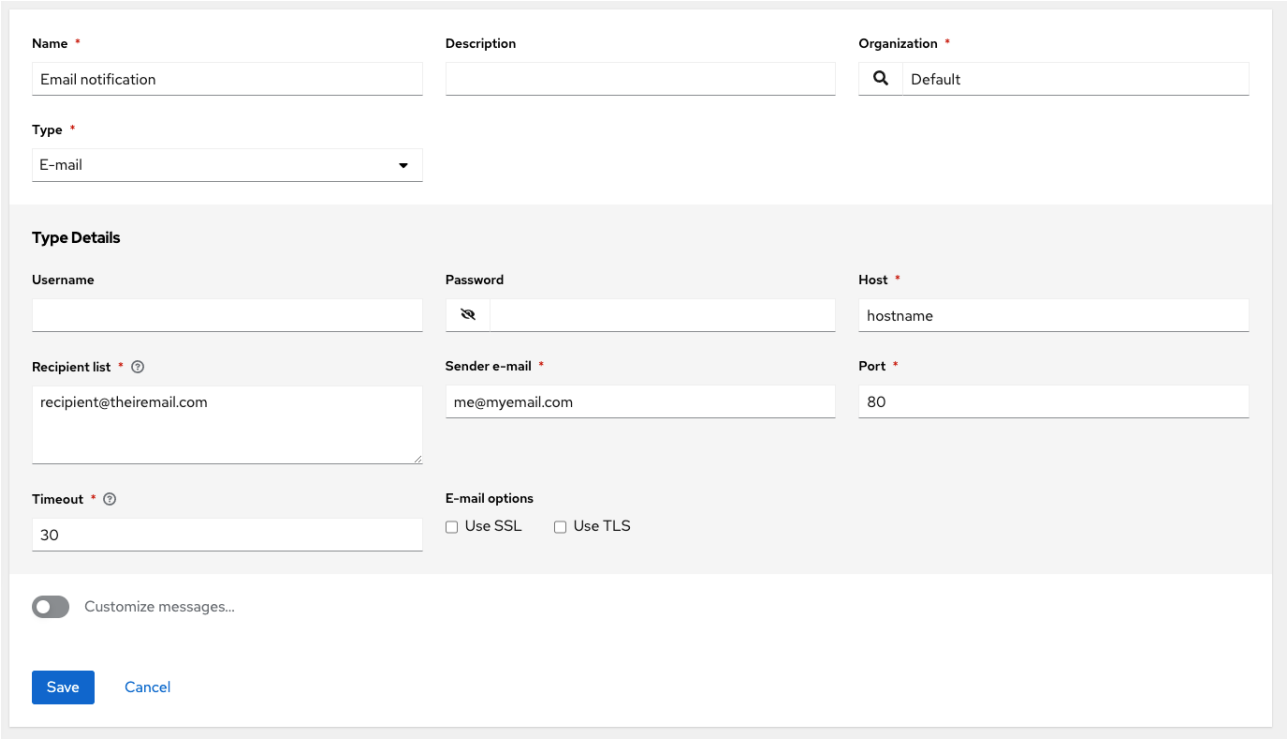
Email

The email notification type supports a wide variety of SMTP servers and has support for SSL/TLS connections.

Provide the following details to set up an email notification:

- **Host**
- **Recipient list**
- **Sender e-mail**
- **Port**

- **Timeout** (in seconds): You can set this up to 120 seconds. This is the length of time that automation controller tries to connect to the email server before failure.



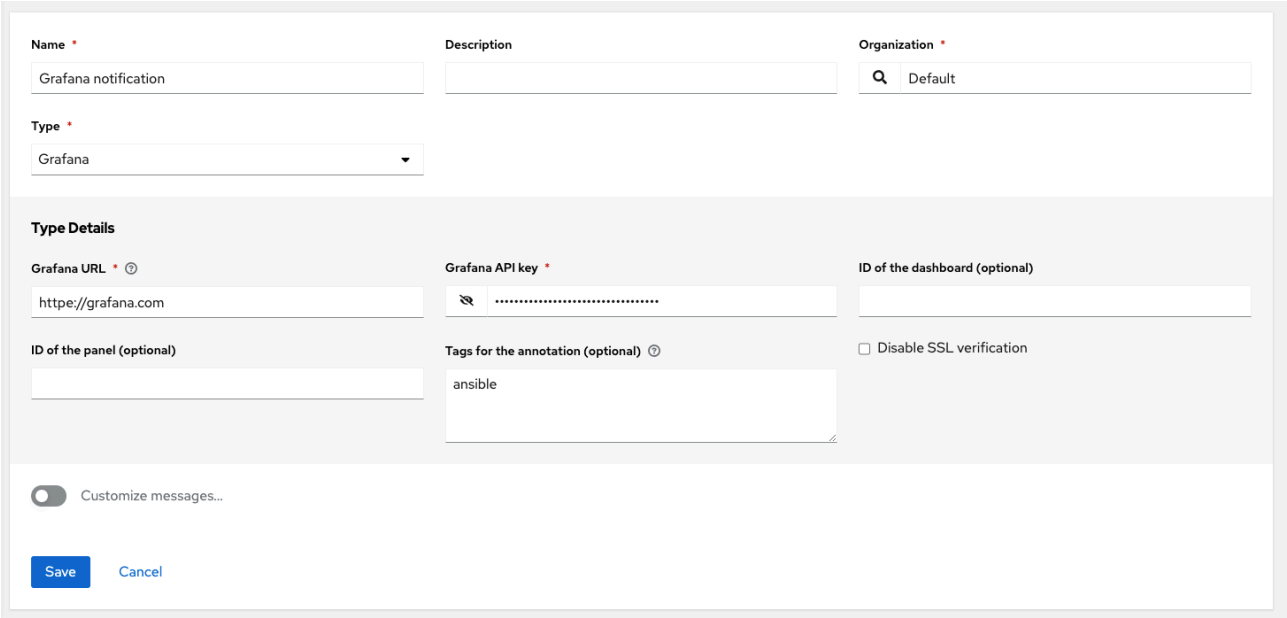
Grafana

You can configure automation controller to send notifications to Grafana.

To integrate Grafana, you must first create an API key in the Grafana system. This is the token that is given to automation controller.

Provide the following details to set up a Grafana notification:

- **Grafana URL**: The URL of the Grafana API service, such as: http://yourcompany.grafana.com.
- **Grafana API key**: You must first create an API key in the Grafana system.
- Optional: **ID of the dashboard**: When you create an API key for the Grafana account, you can set up a dashboard with a unique ID.
- Optional: **ID of the panel**: If you added panels and graphs to your Grafana interface, you can give its ID here.
- Optional: **Tags for the annotation**: Enter keywords to identify the types of events of the notification that you are configuring.
- **Disable SSL verification**: SSL/TLS verification is on by default, but you can turn off verification of the authenticity of the target's certificate. Select this option to disable verification for environments that use internal or private CA's.



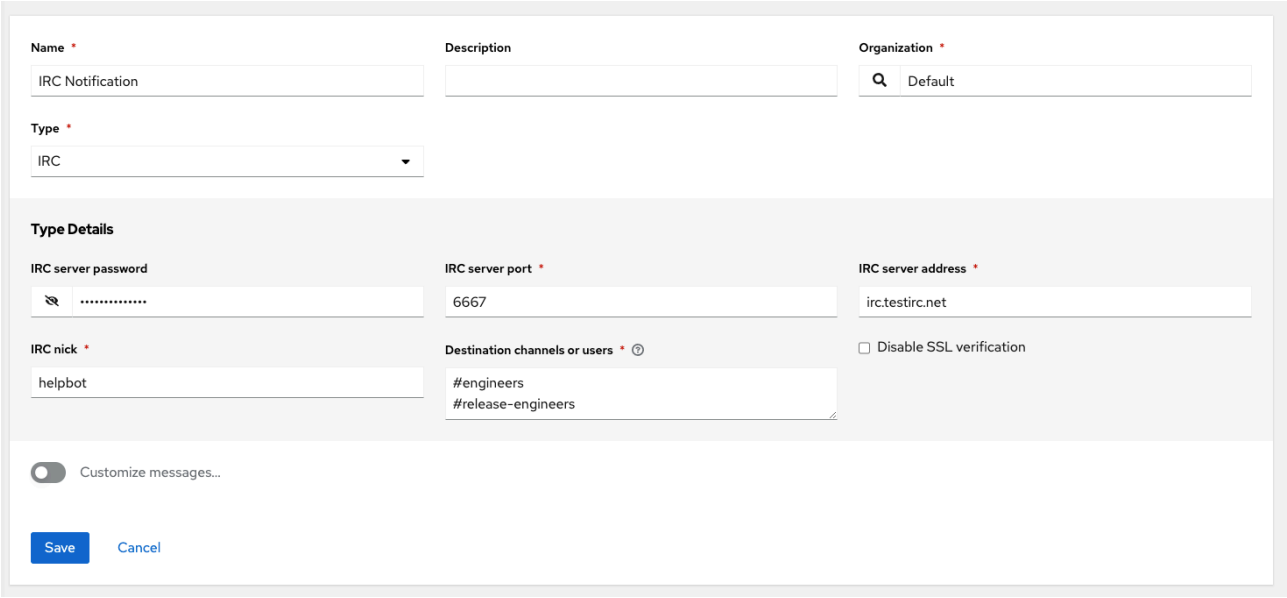
IRC

You can configure automation controller to send notifications by using IRC (Internet Relay Chat).

The IRC notification takes the form of an IRC bot that connects, delivers its messages to channels or individual users, and then disconnects. The notification bot also supports SSL/TLS authentication. The bot does not currently support Nickserv identification. If a channel or user does not exist or is not online then the notification fails. The failure scenario is reserved specifically for connectivity.

Provide the following details to set up an IRC notification:

- Optional: **IRC server password**: IRC servers can require a password to connect. If the server does not require one leave it blank. **IRC Server Port**: The IRC server port. **IRC Server Address**: The hostname or address of the IRC server. **IRC Nick**: The bot's nickname once it connects to the server. **Destination Channels or Users**: A list of users or channels to which the notification is sent.
- Optional: **Disable SSL verification**: Check if you want the bot to use SSL/TLS when connecting.

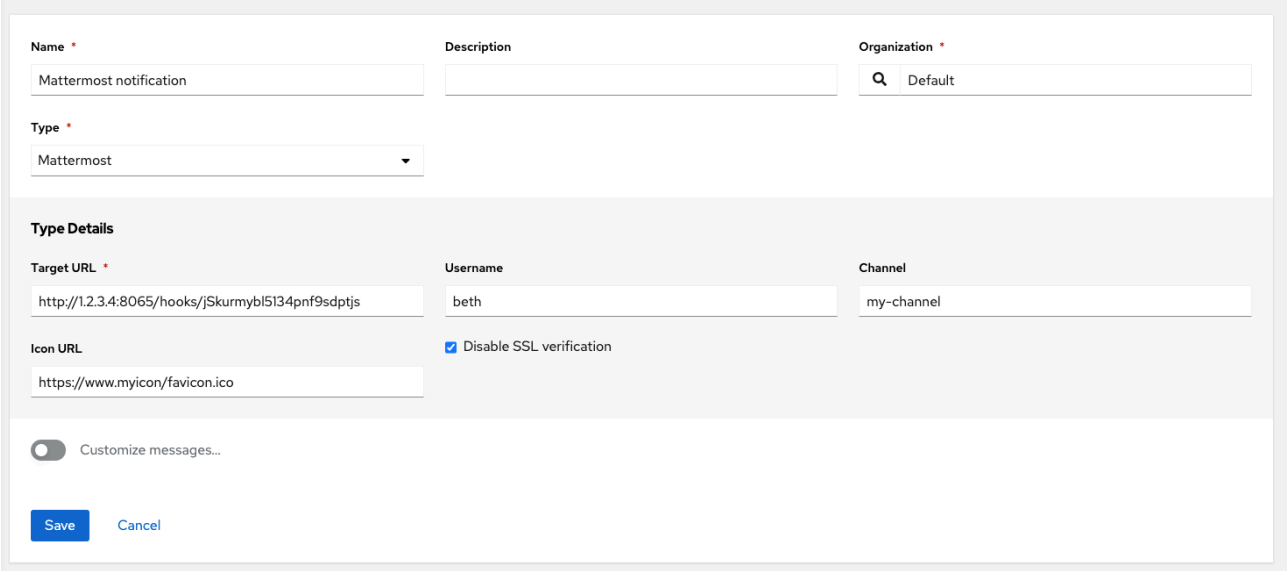


Mattermost

The Mattermost notification type provides a simple interface to Mattermost's messaging and collaboration workspace.

Provide the following details to set up a Mattermost notification:

- **Target URL**: The full URL that is posted to.
- Optional: **Username**: Enter a username for the notification.
- Optional: **Channel**: Enter a channel for the notification.
- **Icon URL**: Specifies the icon to display for this notification.
- **Disable SSL verification**: Turns off verification of the authenticity of the target's certificate. Select this option to disable verification for environments that use internal or private CA's.

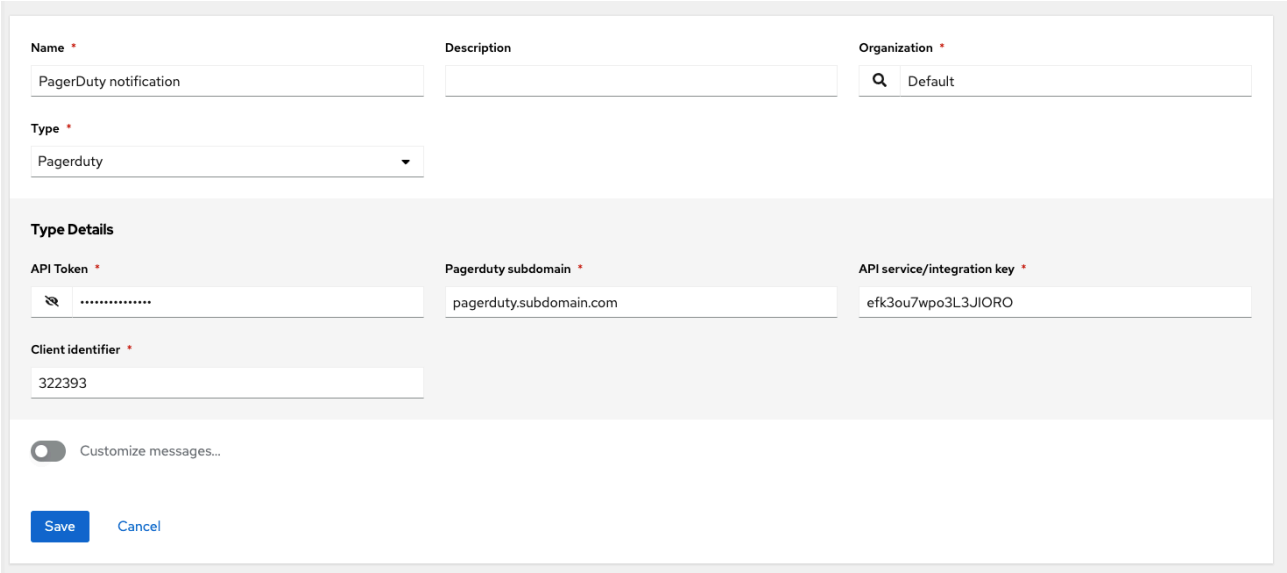


Pagerduty

To integrate Pagerduty, you must first create an API key in the PagerDuty system. This is the token that is given to automation controller. Then create a **Service** which provides an **Integration Key** that is also given to automation controller.

Provide the following details to set up a Pagerduty notification:

- **API Token**: You must first create an API key in the Pagerduty system. This is the token that is given to automation controller.
- **PagerDuty subdomain**: When you sign up for the Pagerduty account, you receive a unique subdomain to communicate with. For example, if you signed up as "testuser", the web dashboard is at `testuser.pagerduty.com` and you give the API `testuser` as the subdomain, not the full domain.
- **API service/Integration Key**: Enter the API service/integration key created in Pagerduty.
- **Client Identifier**: This is sent along with the alert content to the Pagerduty service to help identify the service that is using the API key and service. This is helpful if multiple integrations are using the same API key and service.



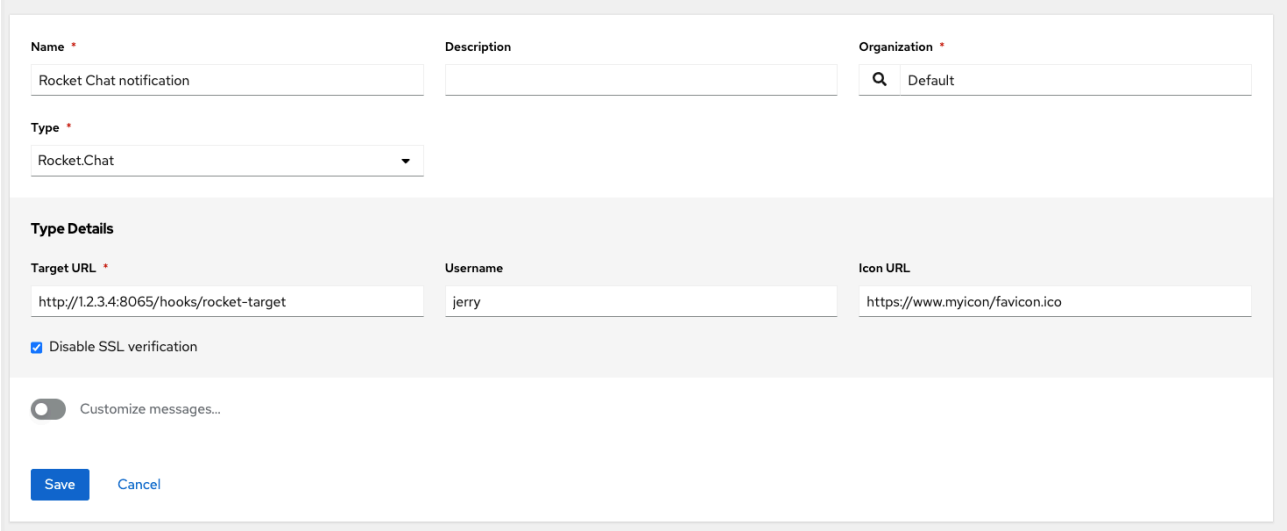
Rocket.Chat

The Rocket.Chat notification type provides an interface to Rocket.Chat's collaboration and communication platform.

Provide the following details to set up a Rocket.Chat notification:

- **Target URL**: The full URL that is `POSTed` to.
- Optional: **Username**: Enter a username.
- Optional: **Icon URL**: Specifies the icon to display for this notification

- **Disable SSL Verification**: Turns off verification of the authenticity of the target's certificate. Select this option to disable verification for environments that use internal or private CA's.

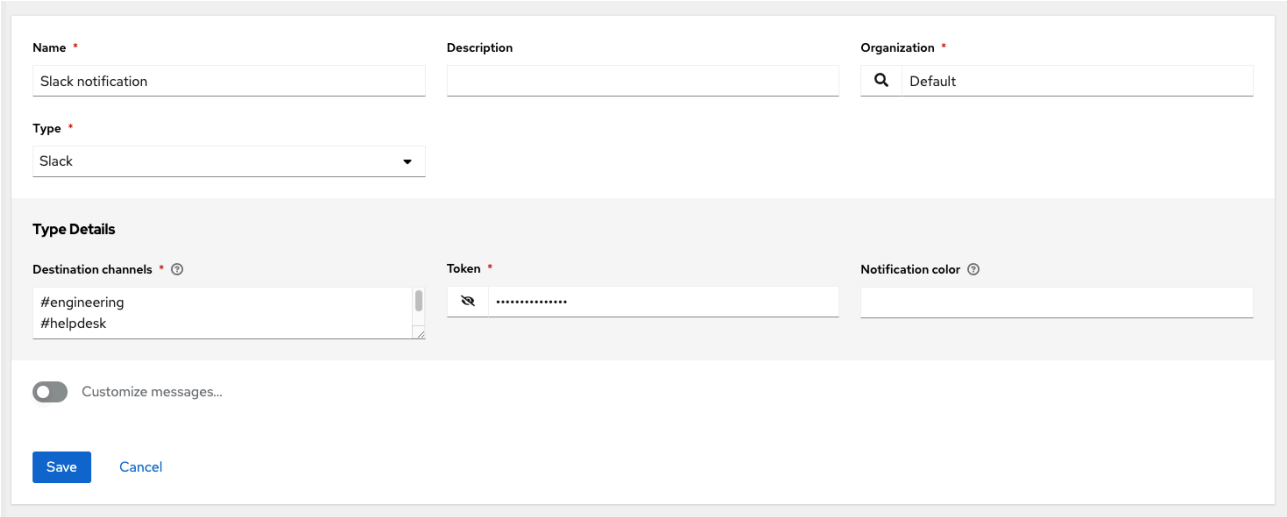


Slack

Slack is a collaborative team communication and messaging tool.

Give the following details to set up a Slack notification:

- A Slack application. For more information, see the Quickstart page of the Slack documentation on how to create one.
- **Token**: A token. For more information, see Legacy bots and specific details on bot tokens on the Current token types documentation page.
- **Destination Channel**: One Slack channel per line. The pound symbol (#) is required for channels. To respond to or start a thread to a specific message add the parent message Id to the channel where the parent message Id is 16 digits. A dot (.) must be manually inserted after the 10th digit. For example, :#destination-channel, 1231257890.006423.
- **Notification color**: Specify a notification color. Acceptable colors are hex color code, for example: #3af or #789abc. When you have a bot or app set up, you must complete the following steps:
    a. Go to **Apps**.
    b. Click the newly-created app and then go to **Add features and functionality**, which enables you to configure incoming webhooks, bots, and permissions, and **Install your app to your workspace**.



Twilio

Configure automation controller to send notifications by using Twilio.

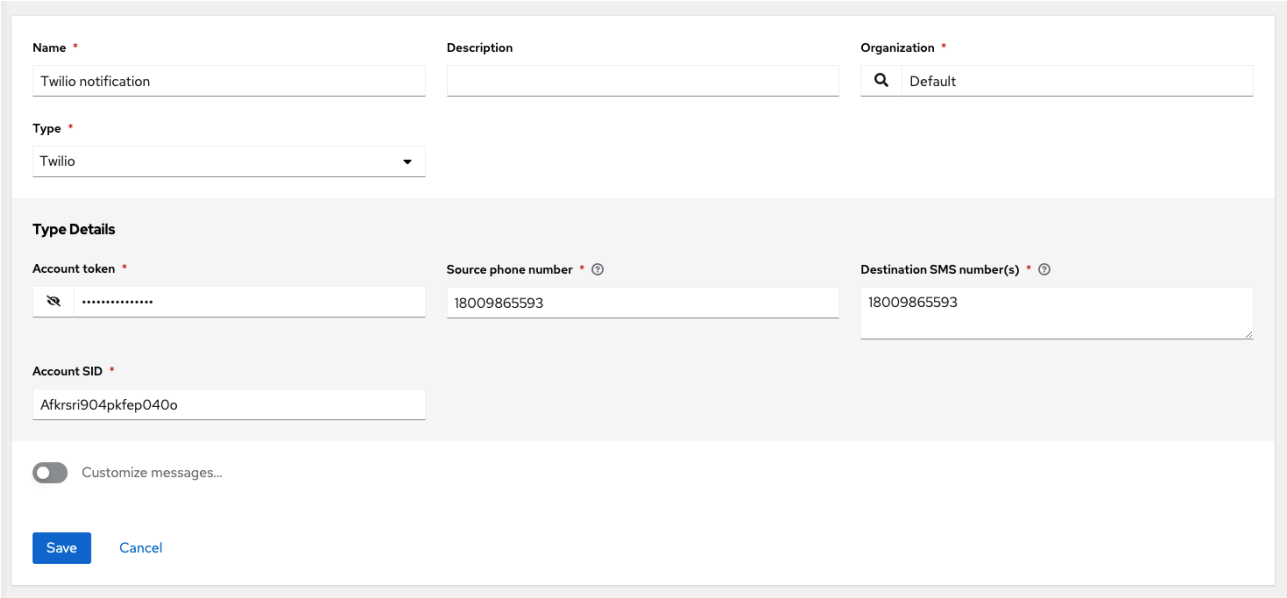
Twilio is a voice and SMS automation service. When you are signed in, you must create a telephone number from which the messages are sent. You can then define a **Messaging Service** under **Programmable SMS** and associate the number you previously created with it.

You might need to verify this number or some other information before you are permitted to use it to send to any numbers. The **Messaging Service** does not require a status callback URL and it does not need the ability to process inbound messages.

Under your individual (or sub) account settings, you have API credentials. Twilio uses two credentials to determine which account an API request is coming from. The **Account SID**, which acts as a username, and the **Auth Token** which acts as a password.

Provide the following details to set up a Twilio notification:

- **Account SID**: Enter the account SID.
- **Account Token**: Enter the account token.
- **Source Phone Number**: Enter the number associated with the messaging service in the form of "+15556667777".
- **Destination SMS Numbers**: Enter the list of numbers you want to receive the SMS. It must be a 10 digit telephone number.



Webhook

The webhook notification type provides a simple interface for sending `POSTs` to a predefined web service.

Automation controller `POSTs` to this address by using application and JSON content type with the data payload containing the relevant details in JSON format. Some web service APIs expect HTTP requests to be in a certain format with certain fields.

Configure the webhook notification with the following:

- Configure the HTTP method, usingBasic authentication `PUT`.
- The body of the outgoing request.
- Configure authentication, using Basic authentication.

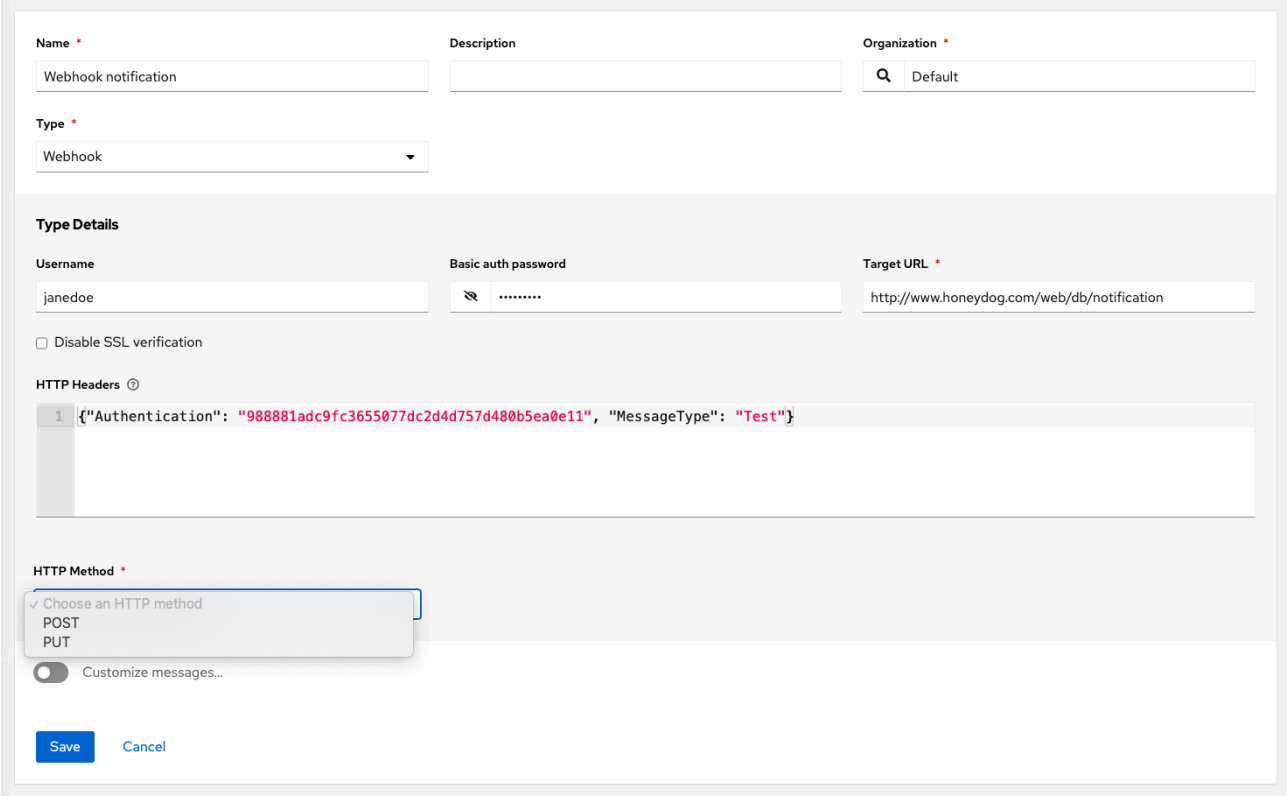
Provide the following details to set up a webhook notification:

- Optional: **Username**: Enter a username.
- Optional: **Basic auth password**:
- **Target URL**: Enter the full URL to which the webhook notification is `PUT` or `POSTed`.
- **HTTP Headers**: Enter Headers in JSON format where the keys and values are strings. For example:

```
 {"Authentication": "988881adc9fc3655077dc2d4d757d480b5ea0e11", "MessageType":
"Test"}`.
```

- **Disable SSL Verification**: SSL/TLS verification is on by default, but you can choose to turn off verification of the authenticity of the target's certificate. Select this option to disable verification for environments that use internal or private CA's.

- **HTTP Method**: Select the method for your webhook:
- **POST**: Creates a new resource. It also acts as a catch-all for operations that do not fit into the other categories. It is likely that you need to **POST** unless you know your webhook service expects a **PUT**.
- **PUT**: Updates a specific resource (by an identifier) or a collection of resources. You can also use **PUT** to create a specific resource if the resource identifier is known beforehand.



Advanced notification settings

Automation controller uses the system hostname for notifications by default.

In Settings > Automation Execution > System, you can replace the default value in the **Base URL of the service** field with your preferred hostname to change the notification hostname.

Refreshing your license also changes the notification hostname. New installations of automation controller do not have to set the hostname for notifications.

Create custom notifications

You can customize the text content of each Notification type on the notification form.

Procedure

1. From the navigation panel, select **Automation Execution > Administration > Notifiers**.
2. Click **Create notifier**.
3. Choose a notification type from the **Type** list.
4. Enable **Customize messages** by using the toggle.

OBSERVE

Customize messages...

Use custom messages to change the content of notifications sent when a job starts, succeeds, or fails. Use curly braces to access information about the job: `{{ job_friendly_name }}`, `{{ url }}`, `{{ job.status }}`. You may apply a number of possible variables in the message. For more information, refer to the Ansible Tower Documentation.

Start message

```
{{ job_friendly_name }} #{{ job.id }} '{{ job.name }}' {{ job.status }}: {{ url }}
```

Start message body

```
{{ job_friendly_name }} #{{ job.id }} had status {{ job.status }}, view details at {{ url }}

{{ job_metadata }}
```

Success message

```
{{ job_friendly_name }} #{{ job.id }} '{{ job.name }}' {{ job.status }}: {{ url }}
```

Success message body

```
{{ job_friendly_name }} #{{ job.id }} had status {{ job.status }}, view details at {{ url }}

{{ job_metadata }}
```

Error message

```
{{ job_friendly_name }} #{{ job.id }} '{{ job.name }}' {{ job.status }}: {{ url }}
```

Error message body

```
{{ job_friendly_name }} #{{ job.id }} had status {{ job.status }}, view details at {{ url }}

{{ job_metadata }}
```

Workflow approved message

```
The approval node "{{ approval_node_name }}" was approved. {{ workflow_url }}
```

Workflow approved message body

```
The approval node "{{ approval_node_name }}" was approved. {{ workflow_url }}

{{ job_metadata }}
```

Workflow denied message

```
The approval node "{{ approval_node_name }}" was denied. {{ workflow_url }}
```

Workflow denied message body

```
The approval node "{{ approval_node_name }}" was denied. {{ workflow_url }}

{{ job_metadata }}
```

Workflow pending message

```
The approval node "{{ approval_node_name }}" needs review. This node can be viewed at: {{ workflow_url }}
```

Workflow pending message body

```
The approval node "{{ approval_node_name }}" needs review. This approval node can be viewed at: {{ workflow_url }}

{{ job_metadata }}
```

Workflow timed out message

```
The approval node "{{ approval_node_name }}" has timed out. {{ workflow_url }}
```

5. You can provide a custom message for various job events, such as the following:

- **Start message body**
- **success message body**
- **Error message body**
- **Workflow approved body**
- **Workflow denied message body**
- **Workflow pending message body**
- **Workflow timed out message body**
  The message forms vary depending on the type of notification that you are configuring. For example, messages for Email and PagerDuty notifications appear to be a typical email, with a body and a subject, in which case, automation controller displays the fields as **Message** and **Message Body**. Other notification types only expect a **Message** for each type of event.

  The **Message** fields are pre-populated with a template containing a top-level variable, `job` coupled with an attribute, such as `id` or `name`. Templates are enclosed in curly brackets and can draw from a fixed set of fields provided by automation controller, shown in the pre-populated message fields:

  This pre-populated field suggests commonly displayed messages to a recipient who is notified of an event. You can customize these messages with different criteria by adding your own attributes for the job as needed. Custom notification messages are rendered using Jinja; the same templating engine used by Ansible playbooks.

  Messages and message bodies have different types of content, as the following points outline:

- Messages are always just strings, one-liners only. New lines are not supported.
- Message bodies are either a dictionary or a block of text:
  - The message body for Webhooks and PagerDuty uses dictionary definitions. The default message body for these is `{{ job_metadata }}`, you can either leave that as it is or provide your own dictionary.
  - The message body for email uses a block of text or a multi-line string. The default message body is:

```
{{ job_friendly_name }} #{{ job.id }} had status {{ job.status }},
view details at {{ url }} {{ job_metadata }}
```

  You can edit this text leaving `{{ job_metadata }}` in, or drop `{{ job_metadata }}`. Since the body is a block of text, it can be any string you want. `{{ job_metadata }}` is rendered as a dictionary containing fields that describe the job being executed. In all cases, `{{ job_metadata }}` includes the following fields:

- `id`
- `name`
- `url`
- `created_by`
- `started`
- `finished`
- `status`
- `traceback`

  You cannot query individual fields within `{{ job_metadata }}`. When you use `{{ job_metadata }}` in a notification template, all data is returned.

  The resulting dictionary looks like the following:

```
{"id": 18,
 "name": "Project - Space Procedures",
 "url": "https://host/#/jobs/project/18",
 "created_by": "admin",
 "started": "2019-10-26T00:20:45.139356+00:00",
 "finished": "2019-10-26T00:20:55.769713+00:00",
 "status": "successful",
 "traceback": ""
}
```

  If `{{ job_metadata }}` is rendered in a job, it includes the following additional fields:

- `inventory`
- `project`
- `playbook`
- `credential`
- `limit`
- `extra_vars`
- `hosts`

  The resulting dictionary is similar to the following:

```
{"id": 12,
 "name": "JobTemplate - Launch Rockets",
 "url": "https://host/#/jobs/playbook/12",
 "created_by": "admin",
 "started": "2019-10-26T00:02:07.943774+00:00",
 "finished": null,
 "status": "running",
 "traceback": "",
 "inventory": "Inventory - Fleet",
 "project": "Project - Space Procedures",
 "playbook": "launch.yml",
 "credential": "Credential - Mission Control",
 "limit": "",
 "extra_vars": "{}",
 "hosts": {}
}
```


If `{{ job_metadata }}` is rendered in a workflow job, it includes the following additional field:

- `body` (This enumerates the nodes in the workflow job and includes a description of the job associated with each node)

The resulting dictionary is similar to the following:

```
{"id": 14,
 "name": "Workflow Job Template - Launch Mars Mission",
 "url": "https://host/#/workflows/14",
 "created_by": "admin",
 "started": "2019-10-26T00:11:04.554468+00:00",
 "finished": "2019-10-26T00:11:24.249899+00:00",
 "status": "successful",
 "traceback": "",
 "body": "Workflow job summary:


         node #1 spawns job #15, \"Assemble Fleet JT\", which
finished with status successful.

         node #2 spawns job #16, \"Mission Start approval
node\", which finished with status successful.\n

         node #3 spawns job #17, \"Deploy Fleet\", which
finished with status successful."

}
```


If you create a notification template that uses invalid syntax or references unusable fields, an error message displays indicating the nature of the error. If you delete a notification's custom message, the default message is shown in its place.

IMPORTANT:
If you save the notifications template without editing the custom message (or edit and revert back to the default values), the **Details** screen assumes the defaults and does not display the custom message tables. If you edit and save any of the values, the entire table displays in the **Details** screen.

Related information
Supported attributes for custom notifications

Enable and disable notifications

You can set up notifications to notify you when a specific job starts, and on the success or failure at the end of the job run. Note the following behaviors:

- If a workflow job template has notification on start enabled, and a job template within that workflow also has notification on start enabled, you receive notifications for both.

- You can enable notifications to run on many job templates within a workflow job template.
- You can enable notifications to run on a sliced job template start and each slice generates a notification.
- When you enable a notification to run on job start, and that notification gets deleted, the job template continues to run, but results in an error message.

You can enable notifications on job start, job success, and job failure, or a combination of these, from the **Notifications** tab of the **Details** page for the following resources:

- Job Templates
- Workflow Templates
- Projects

For workflow templates that have approval nodes, in addition to **Start**, **Success**, and **Failure**, you can enable or disable certain approval-related events:

Related information
Approval nodes

Reset TOWER_URL_BASE

Automation controller determines how the base URL (`TOWER_URL_BASE`) is defined by looking at an incoming request and setting the server address based on that incoming request.

Automation controller takes settings values from the database first. If no settings values are found, it uses the values from the settings files. If you post a license by navigating to the automation controller host's IP address, the posted license is written to the settings entry in the database.

Use the following procedure to reset `TOWER_URL_BASE` if the wrong address has been picked up:

Procedure

1. From the navigation panel, select **Settings › System**.
2. Click **Edit**.
3. Enter the address in the **Base URL of the service** field for the DNS entry you want to appear in notifications.

Notifications API

The Notifications API enables you to trigger notifications for various events in Automation controller. The Notifications API provides endpoints to trigger notifications for job events, such as when a job starts, succeeds, or fails.

Use the `started`, `success`, or `error` endpoints:

```
/api/v2/organizations/N/notification_templates_started/
/api/v2/organizations/N/notification_templates_success/
/api/v2/organizations/N/notification_templates_error/
```

Additionally, the `../../../N/notification_templates_started` endpoints have `GET` and `POST` actions for:

- Organizations
- Projects
- Inventory Sources
- Job Templates
- System Job Templates
- Workflow Job Templates

Custom notification attributes

Learn about the list of supported job attributes and the proper syntax for constructing the message text for notifications.

Related information
Controller notification templates

Access log information

Automation controller logs are accessed in different ways depending on whether you have an RPM-based or containerized installation of Ansible Automation Platform.

Access automation controller logs for containerized Ansible Automation Platform

Logs for containerized Ansible Automation Platform are not saved to specific files. The application logs are sent to the container `stdout` and handled by Podman with `journald`.

The three containers associated with automation controller are:

- `automation-controller-rsyslog`
- `automation-controller-task`
- `automation-controller-web`

For more information about the purpose of each of these containers and how to inspect the logs, see Diagnosing the problem in *Containerized installation*.

Access automation controller logs for RPM-based Ansible Automation Platform

Automation controller logfiles can be accessed from two centralized locations:

- `/var/log/tower/`
- `/var/log/supervisor/`

In the `/var/log/tower/` directory, you can view logfiles captured by:

- **tower.log:** Captures the log messages such as runtime errors that occur when the job is executed.
- **callback_receiver.log:** Captures callback receiver logs that handles callback events when running ansible jobs.
- **dispatcher.log:** Captures log messages for the automation controller dispatcher worker service.
- **job_lifecycle.log:** Captures details of the job run, whether it is blocked, and what condition is blocking it.
- **management_playbooks.log:** Captures the logs of management playbook runs, and isolated job runs such as copying the metadata.
- **rsyslog.err:** Captures rsyslog errors authenticating with external logging services when sending logs to them.

- **task_system.log:** Captures the logs of tasks that automation controller is running in the background, such as adding cluster instances and logs related to information gathering or processing for analytics.
- **tower_rbac_migrations.log:** Captures the logs for rbac database migration or upgrade.
- **tower_system_tracking_migrations.log:** Captures the logs of the controller system tracking migration or upgrade.
- **wsbroadcast.log:** Captures the logs of WebSocket connections in the controller nodes.

In the `/var/log/supervisor/` directory, you can view logfiles captured by:

- **awx-callback-receiver.log:** Captures the log of callback receiver that handles callback events when running ansible jobs, managed by `supervisord`.
- **awx-daphne.log:** Captures the logs of WebSocket communication of WebUI.
- **awx-dispatcher.log:** Captures the logs that occur when dispatching a task to an automation controller instance, such as when running a job.
- **awx-rsyslog.log:** Captures the logs for the `rsyslog` service.
- **awx-uwsgi.log:** Captures the logs related to uWSGI, which is an application server.
- **awx-wsbroadcast.log:** Captures the logs of the WebSocket service that is used by automation controller.
- **failure-event-handler.stderr.log:** Captures the standard errors for `/usr/bin/failure-event-handler` supervisord's subprocess.
- **supervisord.log:** Captures the logs related to `supervisord` itself.
- **wsrelay.log:** Captures the communication logs within the WebSocket relay server.
- **ws_heartbeat.log:** Captures the periodic checks on the health of services running on the host.
- **rsyslog_configurer.log:** Captures rsyslog configuration activity associated with authenticating with external logging services.

The `/var/log/supervisor/` directory includes `stdout` files for all services as well.

You can expect the following log paths to be generated by services used by automation controller (and Ansible Automation Platform):

- **/var/log/nginx/**
- **/var/lib/pgsql/data/pg_log/**
- **/var/log/redis/**

Troubleshooting

Error logs can be found in the following locations:

- Automation controller server errors are logged in `/var/log/tower`.

- Supervisors logs can be found in `/var/log/supervisor/`.
- Nginx web server errors are logged in the httpd error log.
- Configure other automation controller logging needs in `/etc/tower/conf.d/`.

Explore client-side issues by using the JavaScript console built into most browsers and report any errors to Ansible through the Red Hat Customer portal at: https://access.redhat.com/.

Send log files to third-party aggregation services

Logging enables sending detailed logs to third-party aggregation services. These feeds provide insights into Ansible Automation Platform controller use and technical trends. Data can be used to analyze infrastructure events, monitor for anomalies, and correlate events across multiple services.

The types of data that are most useful to automation controller are job fact data, job events or job runs, activity stream data, and log messages. The data is sent in JSON format over a HTTP connection by using minimal service-specific adjustments engineered in a custom handler or through an imported library.

The version of `rsyslog` that is installed by automation controller does not include the following `rsyslog` modules:

- rsyslog-udpspoof.x86_64
- rsyslog-libdbi.x86_64

After installing automation controller, you must only use the automation controller provided `rsyslog` package for any logging outside of automation controller that might have previously been done with the RHEL provided `rsyslog` package.

If you already use `rsyslog` for logging system logs on the automation controller instances, you can continue to use `rsyslog` to handle logs from outside of automation controller by running a separate `rsyslog` process (using the same version of rsyslog that automation controller uses), and pointing it to a separate `/etc/rsyslog.conf` file.

Use the `/api/v2/settings/logging/` endpoint to configure how the automation controller `rsyslog` process handles messages that have not yet been sent if your external logger goes offline:

- `LOG_AGGREGATOR_ACTION_MAX_DISK_USAGE_GB`: Maximum disk persistence for rsyslogd action queuing in GB.
  Specifies the amount of data to store (in gigabytes) during an outage of the external log aggregator (defaults to 1).

  Equivalent to the `rsyslogd queue.maxDiskSpace` setting.
- `LOG_AGGREGATOR_ACTION_QUEUE_SIZE`: Maximum number of messages that can be stored in the log action queue.
  Defines how large the rsyslog action queue can grow in number of messages stored. This can have an impact on memory use. When the queue reaches 75% of this number, the queue starts writing to disk (`queue.highWatermark` in `rsyslog`). When it reaches 90%, `NOTICE`,

`INFO`, and `DEBUG` messages start to be discarded (`queue.discardMark` with 'queue.discardSeverity=5`).

Equivalent to the `rsyslogd queue.size` setting on the action.

It stores files in the directory specified by `LOG_AGGREGATOR_MAX_DISK_USAGE_PATH`.

- `LOG_AGGREGATOR_MAX_DISK_USAGE_PATH`: Specifies the location to store logs that should be retried after an outage of the external log aggregator (defaults to `/var/lib/awx`). Equivalent to the `rsyslogd queue.spoolDirectory` setting.

For example, if `Splunk` goes offline, `rsyslogd` stores a queue on the disk until `Splunk` comes back online. By default, it stores up to 1GB of events (while Splunk is offline) but you can increase that to more than 1GB if necessary, or change the path where you save the queue.

Configure logging components

Automation controller provides several loggers that can be configured to deliver structured log data for analysis and monitoring.

The following are special loggers (except for `awx`, which constitutes generic server logs) that provide large amounts of information in a predictable structured or semi-structured format, using the same structure as if obtaining the data from the API:

- `job_events`: Provides data returned from the Ansible callback module.
- `job_lifecycle`: Provides data about the lifecycle of jobs, including when they are created, started, and finished.
- `broadcast_websocket`: Provides data about broadcast websocket messages sent to the clients.
- `activity_stream`: Displays the record of changes to the objects within the application.
- `system_tracking`: Provides fact data gathered by Ansible `setup` module, that is, `gather_facts: true` when job templates are run with **Enable Fact Cache** selected.
- `awx`: Provides generic server logs, which include logs that would normally be written to a file. It contains the standard metadata that all logs have, except it only has the message from the log statement.

These loggers only use the log-level of `INFO`, except for the `awx` logger, which can be any given level.

Additionally, the standard automation controller logs are deliverable through this same mechanism. It should be apparent how to enable or disable each of these five sources of data without manipulating a complex dictionary in your local settings file, and how to adjust the log-level consumed from the standard automation controller logs.

From the navigation panel, select **Settings > Automation Execution > Logging** to configure the logging components in automation controller.

Log message schema

This section describes the common schema for log messages generated by various automation controller components. Understanding the log message schema can help in effectively monitoring and troubleshooting the system.

Common schema for all loggers:

- `cluster_host_id`: Unique identifier of the host within the automation controller cluster.
- `level`: Standard python log level, roughly reflecting the significance of the event. All of the data loggers as a part of 'level' use `INFO` level, but the other automation controller logs use different levels as appropriate.
- `logger_name`: Name of the logger we use in the settings, for example, "activity_stream".
- `@timestamp`: Time of log.
- `path`: File path in code where the log was generated.

Activity stream schema

Automation controller includes an `activity_stream` logger that records changes to objects in the system, such as job templates, inventories, and credentials.

This uses the fields common to all loggers listed in Log message schema.

It has the following additional fields:

- `actor`: Username of the user who took the action documented in the log.
- `changes`: JSON summary of what fields changed, and their old or new values.
- `operation`: The category of the changes logged in the activity stream, for example, "associate".
- `object1`: Information about the object being operated on, consistent with what is shown in the activity stream.
- `object2`: If applicable, the second object involved in the action.

This logger reflects the data being saved into job events, except when they would otherwise conflict with expected standard fields from the logger, in which case the fields are nested. Note that the field host on the `job_event` model is given as `event_host`. There is also a sub-dictionary field, `event_data` within the payload, which has different fields depending on the specifics of the Ansible event.

This logger also includes the common fields in Log message schema.

Scan / fact / system tracking data schema

This section describes the schema for log messages produced by the scan/fact/system tracking logger in automation controller.

These contain detailed dictionary-type fields that are either services, packages, or files.

- `services`: For services scans, this field is included and has keys based on the name of the service.

> **NOTE:**
> Periods are not allowed by elastic search in names, and are replaced with "_" by the log formatter.

- `package`: Included for log messages from package scans.
- `files`: Included for log messages from file scans.
- `host`: Name of the host the scan applies to.
- `inventory_id`: The inventory id the host is inside of.

This logger also includes the common fields in Log message schema.

Job status changes

The job status changes logger captures changes in the status of jobs as they occur.

This is a lower-volume source of information about changes in job states compared to job events, and captures changes to types of unified jobs other than job template based jobs.

This logger also includes the common fields in Log message schema and fields present on the job model.

Automation controller logs

Automation controller uses the standard Python logging module to log messages from various parts of the system.

This logger also includes the common fields in Log message schema.

In addition, this contains a `msg` field with the log message. Errors contain a separate `traceback` field. From the navigation panel, select **Settings > Automation Execution > Logging**. On the

Logging Settings page click **Edit** and use the **ENABLE EXTERNAL LOGGING** option to enable or disable the logging components.

Configure third-party services

The logging aggregator service works with the following monitoring and data analysis systems:

- Splunk
- Loggly
- Sumologic
- Elastic Stack (formerly ELK stack)

Set up logging

To set up logging to any of the aggregator types for centralized logging follow these steps:

- From the navigation panel, select **Settings › Automation Execution › Logging**.
- On the **Logging settings** page, click **Edit**.
- You can configure the following options:
    - **Logging Aggregator**: Enter the hostname or IP address that you want to send logs to.
    - **Logging Aggregator Port**: Specify the port for the aggregator if it requires one.

      NOTE:
      When the connection type is HTTPS, you can enter the hostname as a URL with a port number, after which, you are not required to enter the port again. However, TCP and UDP connections are determined by the hostname and port number combination, rather than URL. Therefore, in the case of a TCP or UDP connection, supply the port in the specified field. If a URL is entered in the **Logging Aggregator** field instead, its hostname portion is extracted as the hostname.

    - **Logging Aggregator Type**: Click to select the aggregator service from the list:
    - **Logging Aggregator Username**: Enter the username of the logging aggregator if required.
    - **Logging Aggregator Password/Token**: Enter the password of the logging aggregator if required.

- **Loggers to Send Data to the Log Aggregator Form**: All four types of data are pre-populated by default. Click the tooltip icon next to the field for additional information on each data type. Delete the data types you do not want.
- **Cluster wide unique identifier**: Use this to uniquely identify instances.
- **Logging Aggregator Protocol**: Click to select a connection type (protocol) to communicate with the log aggregator. Subsequent options vary depending on the selected protocol.
- **TCP Connection Timeout**: Specify the connection timeout in seconds. This option is only applicable to HTTPS and TCP log aggregator protocols.
- **Logging Aggregator Level Threshold**: Select the level of severity you want the log handler to report.
- **Maximum number of messages that can be stored in the log action queue**:Defines how large the `rsyslog` action queue can grow in number of messages stored. This can have an impact on memory use. When the queue reaches 75% of this number, the queue starts writing to disk (`queue.highWatermark` in `rsyslog`). When it reaches 90%, `NOTICE` , `INFO` , and `DEBUG` messages start to be discarded (`queue.discardMark` with `queue.discardSeverity=5`).
- **Maximum disk persistence for rsyslogd action queuing (in GB)**: The amount of data to store (in gigabytes) if an `rsyslog` action takes time to process an incoming message (defaults to 1). Equivalent to the `rsyslogd queue.maxdiskspace` setting on the action (e.g. `omhttp`). It stores files in the directory specified by `LOG_AGGREGATOR_MAX_DISK_USAGE_PATH` .
- **File system location for rsyslogd disk persistence**: Location to persist logs that should be retried after an outage of the external log aggregator (defaults to `/var/lib/awx`). Equivalent to the `rsyslogd queue.spoolDirectory` setting.: Configure a specific error message. When the API encounters an issue with a request, it typically returns an HTTP error code in the 400 range along with an error. When this happens, an error message is generated in the log that follows the following pattern:
- **Log Format For API 4XX Errors:** When the API encounters an issue with a request, it typically returns an HTTP error code in the 400 range along with an error. When this happens, an error message is generated in the log that follows the following pattern: `' status {status_code} received by user {user_name} attempting to access {url_path} from {remote_addr} '`

- You can set the following options:
  - **Log System Tracking Facts Individually**: Click the tooltip icon for additional information, such as whether or not you want to turn it on, or leave it off by default.
- Review your entries for your chosen logging aggregation.
  - **Enable External Logging**: Select this checkbox if you want to send logs to an external log aggregator.
  - **Enable/disable HTTPS certificate verification**: Certificate verification is enabled by default for the HTTPS log protocol. Select this checkbox if you want the log

handler to verify the HTTPS certificate sent by the external log aggregator before establishing a connection.

- **Enable rsyslogd debugging**: Select this checkbox to enable high verbosity debugging for `rsyslogd`. Useful for debugging connection issues for external log aggregation.

- Click **Save** or **Cancel** to abandon the changes.

Related information
API 4XX Error Configuration

Troubleshoot logging

This section provides information to help troubleshoot logging issues in automation controller.

Logging Aggregation

If you have sent a message with the test button to your configured logging service through http or https, but did not receive the message, check the `/var/log/tower/rsyslog.err` log file. This is where errors are stored if they occurred when authenticating rsyslog with an http or https external logging service. Note that if there are no errors, this file does not exist.

API 4XX Errors

You can include the API error message for 4XX errors by modifying the log format for those messages. Refer to the Set up logging.

LDAP

You can enable logging messages for the LDAP adapter. For more information, see Set up logging.

SAML

You can enable logging messages for the SAML adapter the same way you can enable logging for LDAP.

Related information
API 4XX Error Configuration

Send metrics to system monitoring software

Automation controller provides metrics that can be used to monitor the health and performance of the system.

A metrics endpoint, `/api/controller/v2/metrics/` is available in the API that produces instantaneous metrics about automation controller, which can be consumed by system monitoring software such as the open source project Prometheus.

The types of data shown at the `metrics/` endpoint are `Content-type: text/plain` and `application/json`.

This endpoint has useful information, such as counts of how many active user sessions there are, or how many jobs are actively running on each automation controller node.

You can configure Prometheus to scrape these metrics from automation controller by hitting the automation controller metrics endpoint and storing this data in a time-series database.

Clients can later use Prometheus in conjunction with other software such as Grafana or Metricbeat to visualize that data and set up alerts.

Set up Prometheus

Learn how to use Prometheus to collect and visualize metrics from automation controller.

You must install Prometheus on a virtual machine or container.

For more information, see the First steps with Prometheus documentation.

Procedure

1. In the Prometheus configuration file (typically `prometheus.yml`), specify a `<token_value>`, a valid username and password for an automation controller user that you have created, and a `<controller_host>`.

> **NOTE:**
> Alternatively, you can provide an OAuth2 token (which can be generated at `/api/v2/users/N/personal_tokens/`). By default, the configuration assumes a user with username= `admin` and password= `password`.

Using an OAuth2 Token, created at the `/api/v2/tokens` endpoint to authenticate Prometheus with automation controller, the following example provides a valid scrape configuration if the URL for your automation controller's metrics endpoint is `/https://controller_host:443/metrics`.

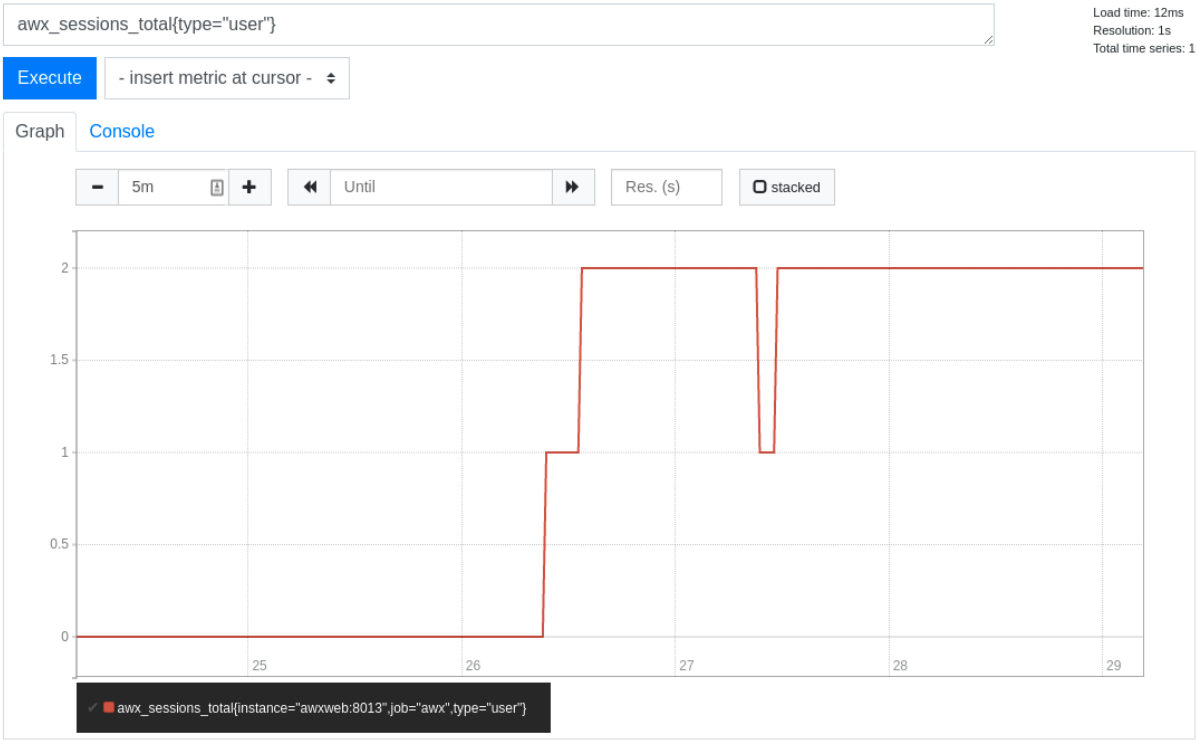
```
scrape_configs


  - job_name: 'controller'
    tls_config:
        insecure_skip_verify: True
    metrics_path: /api/v2/metrics
    scrape_interval: 5s
    scheme: https
    bearer_token: <token_value>
    # basic_auth:
    #   username: admin
    #   password: password
    static_configs:
        - targets:
            - <controller_host>
```

For help configuring other aspects of Prometheus, such as alerts and service discovery configurations, see the Prometheus configuration documentation.

If Prometheus is already running, you must restart it to apply the configuration changes by making a **POST** to the reload endpoint, or by killing the Prometheus process or service.

2. Use a browser to navigate to your graph in the Prometheus UI at `/http://<your_prometheus>:9090/graph` and test out some queries. For example, you can query the current number of active automation controller user sessions by executing: `awx_sessions_total{type="user"}`.



Next steps

Refer to the metrics endpoint in the automation controller API for your instance (`api/v2/metrics`) for more ways to query.

Configure logging for Event-Driven Ansible

Event-Driven Ansible offers an audit logging solution over its resources. Each supported create, read, update and delete (CRUD) operation is logged against rulebook activations, event streams, decision environments, projects, and activations.

Some of these resources support further operations, such as sync, enable, disable, restart, start, and stop; for these operations, logging is supported as well. These logs are only retained for the life cycle of its associated container.

See the following sample logs for each supported logging operation.

Log samples

Review logging samples for various API operations (CRUD, sync, and the like) to understand the expected audit format and efficiently monitor resource changes.

Rulebook activation

```
1. Create

    1. 2024-08-15 14:13:20,384 aap_eda.api.views.activation INFO   Action:
Create / ResourceType: RulebookActivation / ResourceName: quick_start_project /
ResourceID: 53 / Organization: Default

2. Read

    1. 2024-08-15 14:21:26,844 aap_eda.api.views.activation INFO   Action: Read /
ResourceType: RulebookActivation / ResourceName: quick_start_activation /
ResourceID: 1 / Organization: Default

3. Disable

    1. 2024-08-15 14:23:57,798 aap_eda.api.views.activation INFO   Action:
Disable / ResourceType: RulebookActivation / ResourceName:
quick_start_activation / ResourceID: 1 / Organization: Default

4. Enable

    1. 2024-08-15 14:24:16,472 aap_eda.api.views.activation INFO   Action:
Enable / ResourceType: RulebookActivation / ResourceName:
quick_start_activation / ResourceID: 1 / Organization: Default

5. Delete

    1. 2024-08-15 14:24:53,847 aap_eda.api.views.activation INFO   Action:
Delete / ResourceType: RulebookActivation / ResourceName:
quick_start_activation / ResourceID: 1 / Organization: Default

6. Restart

    2024-08-15 14:24:34,169 aap_eda.api.views.activation INFO      Action:
Restart / ResourceType: RulebookActivation / ResourceName:
quick_start_activation / ResourceID: 1 / Organization: Default
```

EventStream Logs

```
1. Create

    1. 2024-08-15 13:46:26,903 aap_eda.api.views.webhook INFO     Action:
Create / ResourceType: EventStream / ResourceName: ZackTest / ResourceID: 1 /
Organization: Default

2. Update

    1. 2024-08-15 13:56:17,440 aap_eda.api.views.webhook INFO     Action:
Update / ResourceType: EventStream / ResourceName: ZackTest / ResourceID: 1 /
Organization: Default

3. Read

    1. 2024-08-15 13:56:56,271 aap_eda.api.views.webhook INFO     Action: Read /
ResourceType: EventStream / ResourceName: ZackTest / ResourceID: 1 /
Organization: Default

4. List

    1. 2024-08-15 13:56:17,492 aap_eda.api.views.webhook INFO     Action: List /
ResourceType: EventStream / ResourceName: * / ResourceID: * / Organization: *

5. Delete

    1. 2024-08-15 13:57:13,124 aap_eda.api.views.webhook INFO     Action:
Delete / ResourceType: EventStream / ResourceName: ZackTest / ResourceID: None /
Organization: Default
```

Decision Environment

```
1. Create

    1. 2024-08-15 14:10:53,311 aap_eda.api.views.decision_environment INFO
Action: Create / ResourceType: DecisionEnvironment / ResourceName:
quick_start_de / ResourceID: 86 / Organization: Default

2. Read

    1. 2024-08-15 14:10:53,349 aap_eda.api.views.decision_environment INFO
Action: Read / ResourceType: DecisionEnvironment / ResourceName: quick_start_de /
ResourceID: 86 / Organization: Default

3. Update

    2024-08-15 14:11:20,970 aap_eda.api.views.decision_environment INFO
Action: Update / ResourceType: DecisionEnvironment / ResourceName:
quick_start_de / ResourceID: 86 / Organization: Default

4. Delete

2024-08-15 14:11:42,369 aap_eda.api.views.decision_environment INFO     Action:
Delete / ResourceType: DecisionEnvironment / ResourceName: quick_start_de /
ResourceID: None / Organization: Default
```

Project

```
1. Create

    1. 2024-08-15 14:05:26,874 aap_eda.api.views.project INFO     Action:
Create / ResourceType: Project / ResourceName: quick_start_project / ResourceID:
86 / Organization: Default

2. Read

    1. 2024-08-15 14:05:26,913 aap_eda.api.views.project INFO     Action: Read /
ResourceType: Project / ResourceName: quick_start_project / ResourceID: 86 /
Organization: Default

3. Update

    1. 2024-08-15 14:06:08,255 aap_eda.api.views.project INFO     Action:
Update / ResourceType: Project / ResourceName: quick_start_project / ResourceID:
86 / Organization: Default

4. Sync

    1. 2024-08-15 14:06:30,580 aap_eda.api.views.project INFO     Action: Sync /
ResourceType: Project / ResourceName: quick_start_project / ResourceID: 86 /
Organization: Default

5. Delete

    1. 2024-08-15 14:06:49,481 aap_eda.api.views.project INFO     Action:
Delete / ResourceType: Project / ResourceName: quick_start_project / ResourceID:
86 / Organization: Default
```

Activation Start/Stop

```
1. Start

    1. 2024-08-15 14:21:29,076 aap_eda.services.activation.activation_manager
INFO     Requested to start activation 1, starting.

    2024-08-15 14:21:29,093 aap_eda.services.activation.activation_manager INFO
  Creating a new activation instance for activation: 1

    2024-08-15 14:21:29,104 aap_eda.services.activation.activation_manager INFO
  Starting container for activation instance: 1

2. Stop

    1. eda-activation-worker-1  | 2024-08-15 14:40:52,547
aap_eda.services.activation.activation_manager INFO     Stop operation requested
for activation id: 2 Stopping activation.

    eda-activation-worker-1  | 2024-08-15 14:40:52,550
aap_eda.services.activation.activation_manager INFO     Activation 2 is already
stopped.

    eda-activation-worker-1  | 2024-08-15 14:40:52,550
aap_eda.services.activation.activation_manager INFO     Activation manager
activation id: 2 Activation restart scheduled for 1 second.

    eda-activation-worker-1  | 2024-08-15 14:40:52,562 rq.worker INFO
activation: Job OK (activation-2)
```

Capture telemetry data for Red Hat Developer Hub

Red Hat Developer Hub (RHDH) sends telemetry data to Red Hat using the `backstage-plugin-analytics-provider-segment` plug-in, which is enabled by default. This includes telemetry data from the Ansible plug-ins.

Red Hat collects and analyzes the following data to improve your experience with Red Hat Developer Hub:

- Events of page visits and clicks on links or buttons.
- System-related information, for example, locale, timezone, user agent including browser and OS details.
- Page-related information, for example, title, category, extension name, URL, path, referrer, and search parameters.
- Anonymized IP addresses, recorded as 0.0.0.0.
- Anonymized username hashes, which are unique identifiers used solely to identify the number of unique users of the RHDH application.
- Feedback and sentiment submitted through the Ansible plug-ins for Red Hat Developer Hub feedback form, including a 1-5 star rating and feedback text. Users must acknowledge that they share the feedback with Red Hat before submitting. The feedback form is disabled by default.

With Red Hat Developer Hub, you can disable or customize the telemetry data collection feature. For more information, refer to *Telemetry data collection and analysis* .

Related information
Telemetry data collection and analysis

Capture telemetry data for the Ansible self-service portal

The telemetry data collection feature helps in collecting and analyzing the telemetry data to improve your experience with self-service automation portal. This feature is enabled by default.

Telemetry data collected by Red Hat

Red Hat collects and analyses the following data:

- Events of page visits and clicks on links or buttons.
- System-related information, for example, locale, timezone, user agent including browser and OS details.
- Page-related information, for example, title, category, extension name, URL, path, referrer, and search parameters.
- Anonymized IP addresses, recorded as `0.0.0.0`.
- Anonymized username hashes, which are unique identifiers used solely to identify the number of unique users of the RHDH application.
- Feedback and sentiment submitted through the self-service automation portal feedback form, including a 1-5 star rating and feedback text. Users must acknowledge that they share the feedback with Red Hat before submitting.

NOTE:
The feedback form is optional and disabled by default. You can enable for your users if you choose.

Related information
Enable feedback to Red Hat

Disable telemetry data collection

You can disable and enable the telemetry data collection feature for Ansible automation portal by updating the Helm chart for your OpenShift Container Platform project.

Procedure

1. Log in to the OpenShift Container Platform console and open the project for Ansible automation portal in the **Developer** perspective.
2. Navigate to **Helm**.
3. Click the **More actions** ⋮ icon for your Ansible automation portal Helm chart and select **Upgrade**.
4. Select **YAML view**.
5. Locate the `redhat-developer-hub.global.dynamic.plugins` section of the Helm chart.
6. To disable telemetry data collection, add the following lines to the `redhat-developer-hub.global.dynamic.plugins` section.

```
redhat-developer-hub:
  global:
    # ...
    dynamic:
      plugins:
        - disabled: true
          package: >-
            ./dynamic-plugins/dist/backstage-community-plugin-analytics-
provider-segment
```

   To re-enable telemetry data collection, delete these lines.
7. Click **Upgrade** to apply the changes to the Helm chart and restart the pod.

Red Hat product documentation legal notices

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If

such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions. "This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code. The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component,

or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions. All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law. No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work’s users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies. You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions. You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so. A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms. You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms. "Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination. You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies. You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients. Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents. A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom. If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License. Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License. The Free Software Foundation may publish revised and/ or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty. THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES

SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16. If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see link:https://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read link:https://www.gnu.org/licenses/why-not-lgpl.html.

Apache license

Version 2.0, January 2004

http://www.apache.org/licenses/

Terms and Conditions for use, reproduction, and distribution

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, **"control"** means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or **"Your"**) shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, **"submitted"** means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as **"Not a Contribution."**

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a **"NOTICE"** text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications,

or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

www.redhat.com