

Amazon Web Services Instance Scheduler

Implementation Guide



Table of Contents

Home

Amazon Web Services Implementation Guide	6
--	---

Overview

Cost	7
------	---

Architecture	9
--------------	---

Components

Scheduler configuration table	11
-------------------------------	----

Schedules	11
-----------	----

Periods	11
---------	----

Time Zone	11
-----------	----

Hibernate Field	12
-----------------	----

Enforced field	12
----------------	----

Retain running field	12
----------------------	----

SSM maintenance window field	13
------------------------------	----

Override status field	13
-----------------------	----

Instance type	14
---------------	----

Schedule definitions	14
----------------------	----

Period rules	16
Start and stop times	16
Adjacent periods	17
Days of the week	18
Days of the month	18
Months	18
Period definitions	18
Cross-Account instance scheduling	20
Amazon Systems Manager Parameter Store	20
Automated tagging	20
Scheduler command line interface	21

Security

Amazon Key Management System	22
Amazon Identity Access Management	22

Design considerations

Partial automation	23
Instance shutdown behavior	23
Amazon EC2	23
Amazon RDS	23
Amazon RDS maintenance window	24
Global configuration settings	24

Performance	24
Encrypted Amazon EBS volumes	25
Regional deployments	25
Logging and notifications	25

CloudFormation templates

CloudFormation Template Resource	27
----------------------------------	----

Automated deployment

Update the stack	28
Deployment overview	29
Step 1. Launch the instance scheduler stack	29
Step 2. Configure periods	32
Step 3. Configure schedules	33
Step 4. Tag your instances	34
Setting the tag value	34
Step 5. Launch the remote stack in secondary accounts (Optional)	34

Amazon CloudWatch metrics

View Instance Scheduler Metrics	36
---------------------------------	----

Contributors

Revision

Notices

Home

Amazon Web Services Implementation Guide

Publication date: May 2020 (last update: March 2022)

This implementation guide discusses architectural considerations and configuration steps for deploying the Amazon Web Services Instance Scheduler in the Amazon Web Services Cloud. It includes links to Amazon CloudFormation templates that launch, configure, and run the services required to deploy this solution using Amazon Web Services best practices for security and availability.

The guide is intended for IT infrastructure architects, administrators, and DevOps professionals who have practical experience architecting in the Amazon Web Services Cloud.

The Amazon Web Service Instance Scheduler is a solution that automates the starting and stopping of Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Relational Database Service (Amazon RDS) instances.

This solution helps reduce operational costs by stopping resources that are not in use and starting resources when their capacity is needed. For example, a company can use Amazon Web Services Instance Scheduler in a production environment to automatically stop instances outside of business hours every day. If you leave all of your instances running at full utilization, this solution can result in up to 70% cost savings for those instances that are only necessary during regular business hours (weekly utilization reduced from 168 hours to 50 hours).

Amazon Web Services Instance Scheduler leverages resource tags and Amazon Lambda to automatically stop and restart instances across multiple Amazon Web Services Regions and accounts on a customer-defined schedule. This solution also allows you to use hibernation for stopped Amazon EC2 instances.

Cost

You are responsible for the cost of the services used while running Amazon Web Services Instance Scheduler. As of March 2022, the cost for running this solution with default settings in the Ningxia Region is approximately ¥39.06 per month in Amazon Lambda charges, or less if you have [Lambda free tier](#) monthly usage credit. This is independent of the number of Amazon EC2 instances you are running. The optional custom Amazon CloudWatch metric will cost an additional ¥8.00 per month per schedule or scheduled service. By default, this solution uses Auto Scaling for its Amazon DynamoDB tables to provide sufficient read and write capacity.

Amazon Web Services Instance Scheduler is designed to run different numbers of Amazon Lambda functions per run cycle. For example, if the solution is being used to manage both EC2 and RDS instances in one Region for two accounts (one account where the solution is deployed and the other account is a cross account), the solution will run five Lambda functions. One for the initial start of the process to handle CloudWatch Events, which is invoked based on the selected frequency (default: five minutes), and each service, account, and Region will be handled by an individual Lambda run (2 accounts x 2 services x 1 Region). The cost of the solution per run will depend on the number of instances being tagged and managed by the solution. As the number of EC2 and RDS instances increases, the Lambda run time also increases proportionately.

The costs in the following table are based on the following assumptions:

1. The solution is deployed in Ningxia Region.
2. The solution is managing both EC2 and RDS instances.
3. The solution is managing instances in an additional account.
4. The total number of executions per day is 1,440 runs (Lambda is scheduled to run every five minutes).
5. The average run time for each Lambda is assumed as eight seconds (this depends upon the number of instances being scheduled).
6. The memory selected for the Amazon Lambda 128 MB.

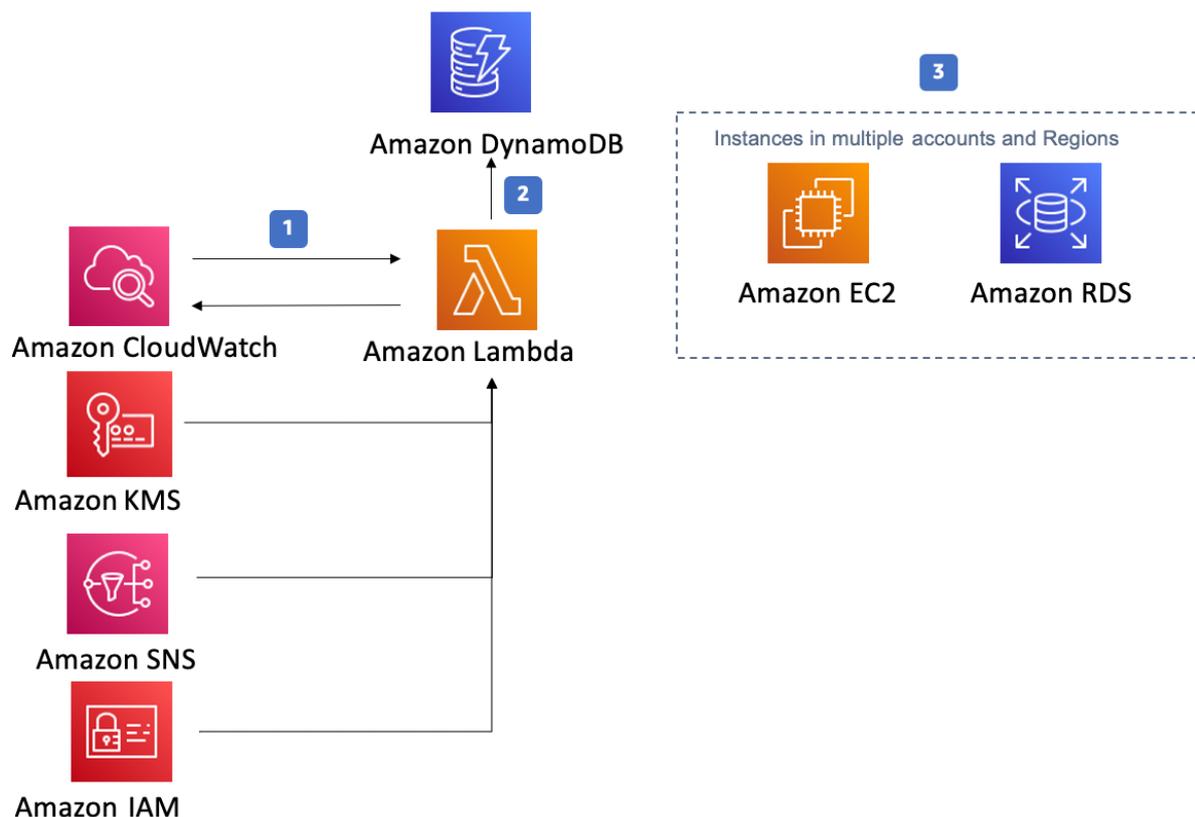
Service	Dimensions	Cost (per month)
Amazon Lambda	1,440 runs/24 hours	¥24.60
	40 seconds per run (8 seconds for each Lambda)	
	(¥0.0000142/second/ run)	
Amazon DynamoDB	1,080,000 (Write/ month)	¥12.05
	(¥11.1538 per million requests)	
Amazon DynamoDB	1,080,000 (Read/ month)	¥2.41
	(¥2.2308 per million requests)	
Amazon DynamoDB	<1GB	¥0.00
	(First 25 GB is free)	
	Total	¥39.06

The cost is independent of the number of Amazon EC2 and RDS instances you are running. By default, this solution uses on-demand scaling for its Amazon DynamoDB tables to provide sufficient read and write capacity. The services for this solution are listed under Additional Resources.

Prices are subject to change. For full details, refer to the pricing webpage for each service you will be using in this solution.

Architecture

Deploying this solution with the default parameters builds the following environment in the Amazon Web Services Cloud.



1. The Amazon CloudFormation template sets up an Amazon CloudWatch event at a customer-defined interval. This event invokes the Amazon Web Services Instance Scheduler Lambda function. During configuration, the user defines the Regions and accounts, as well as a custom tag that Amazon Web Services Instance Scheduler will use to associate schedules with applicable Amazon EC2, Amazon RDS instances, and clusters.
2. These values are stored in Amazon DynamoDB, and the Lambda function retrieves them each time it runs. You can then apply the custom tag to applicable instances.
3. During initial configuration of the Instance Scheduler, you define a tag key you will use to identify applicable Amazon EC2 and Amazon RDS instances. When you create a schedule, the name you specify is used as the tag value that identifies the schedule you want to apply to the tagged resource. For example, a user might use the solution's default tag name (tag key) Schedule and create a schedule called uk-office-hours. To identify an instance that will use the uk-office-hours schedule, the user adds the Schedule tag key with a value of uk-office-hours.

The Lambda function uses Amazon Identity Access Management (Amazon IAM) for permission requirements for your resources, and Amazon Key Management Service (Amazon KMS) for encryption of the Amazon Simple Notification Service (Amazon SNS topic) and Dynamo DB tables. Each time the solution's Lambda function runs, it checks the current state of each appropriately tagged instance against the targeted state (defined by one or more periods in a schedule in the instance tag) in the associated schedule, and then applies the appropriate start or stop action, as necessary.

For example, if the Lambda function is invoked on a Friday at 9 am (ET) and it identifies a stopped Amazon EC2 or Amazon RDS instance with a Schedule=office-hours tag, it will check Amazon DynamoDB for the office-hours schedule configuration details. If the office-hours schedule contains a period rule that indicates that the instance should run Monday through Friday from 9 am ET to 5 pm ET, the Lambda function will start that instance.

The Lambda function also records the name of the schedule, the number of instances associated with that schedule, and the number of running instances as an optional custom metric in Amazon CloudWatch (refer to Amazon CloudWatch Metrics).

Note

Stopping an Amazon EC2 instance is different from terminating an Amazon EC2 instance. By default, Amazon EC2 instances are configured to stop, not terminate, when shut down, but you can modify this behavior. Before using this solution, verify that instances are set to stop or terminate as appropriate.

Components

Scheduler configuration table

When deployed, the Amazon Web Services Instance Scheduler creates an Amazon DynamoDB table that contains global configuration settings. To modify these global configuration settings after the solution is deployed, update the CloudFormation stack. Do not modify these values in the DynamoDB table. If you modify the values in the DynamoDB table, you will create a conflict between the stored parameters in the stack and the values in the table.

Global configuration items contain a type attribute with a value of config in the configuration table. Schedules and periods contain type attributes with values of schedule and period, respectively. You can add, update, or remove schedules and periods from the configuration table using the DynamoDB console or the solution's command line interface.

Schedules

Schedules specify when Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Relational Database Service (Amazon RDS) instances should run. Each schedule must have a unique name, which is used as the tag value that identifies the schedule you want to apply to the tagged resource.

Periods

Each schedule must contain at least one period that defines the time(s) the instance should run. A schedule can contain more than one period. When more than one period is used in a schedule, the Instance Scheduler will apply the appropriate start action when at least one of the period rules is true. For more information, refer to [Period Rules](#).

Time Zone

You can also specify a time zone for the schedule. If you do not specify a time zone, the schedule will use the default time zone you specify when you launch the solution. For a list of acceptable time zone values, refer to the TZ column of the [List of TZ Database Time Zones](#).

Hibernate Field

The hibernate field allows you to use hibernation for stopped Amazon EC2 instances. Your EC2 instances will have an Amazon Machine Image (AMI) when the hibernate field is set to true. For more information, refer to [Hibernate your On-Demand or Reserved Linux instance in the Amazon Elastic Compute Cloud User Guide for Linux Instances](#). Hibernation saves the contents from the instance memory (RAM) to your Amazon Elastic Block Store (Amazon EBS) root volume. If this field is set to true, instances are hibernated when the solution stops them.

If you set the solution to use hibernation, but your instances are not enabled for hibernation or they do not meet the hibernation prerequisites, the solution logs a warning and the instances are stopped without hibernation. For more information, refer to [Hibernate your On-Demand or Reserved Linux instance in the Amazon Elastic Compute Cloud User Guide for Linux instances](#).

Enforced field

Schedules contain an enforced field that allows you to prevent an instance from being manually started outside of a running period, or manually stopped during a running period. If this field is set to true and a user manually starts an instance outside of a running period, the solution will stop the instance. If this field is set to true, it also restarts an instance if it was manually stopped during a running period.

Retain running field

The retain_running field prevents the solution from stopping an instance at the end of a running period if the instance was manually started before the beginning of the period. For example, if an instance with a period that runs from 9 am to 5 pm is manually started before 9 am, the solution will not stop the instance at 5 pm.

SSM maintenance window field

The `ssm-maintenance-window` field allows you to automatically add a Systems Manager maintenance window as a running period to a schedule. When you specify the name of a maintenance window that exists in the same account and Region as your deployed stack to schedule your Amazon EC2 instances, the solution will start the instance before the start of the maintenance window and stop the instance at the end of the maintenance window if no other running period specifies that the instance should run, and if the maintenance event is completed.

The solution uses the Lambda frequency you specified during initial configuration to determine how long before the maintenance window to start your instance. If you set the `Frequency CloudFormation` parameter to 10 minutes or less, the solution will start the instance 10 minutes before the maintenance window. If you set the frequency to greater than 10 minutes, the scheduler will start the instance the same number of minutes as the frequency you specified. For example, if you set the frequency to 30 minutes, the scheduler will start the instance 30 minutes before the maintenance window.

The CloudFormation parameter `Enable SSM Maintenance windows` in the solution stack should be set to `Yes` and the stack should be updated, so that the solution will start loading the SSM maintenance window into the DynamoDB table, which will be used when the Lambda function runs.

For more information, refer to [Systems Manager Maintenance Windows](#) in the Systems Manager user guide.

Override status field

Schedules also contain an `override_status` field that allows you to temporarily override the solution's start and stop actions. If you set the field to `running`, the solution will start but not stop the applicable instance. The instance will run until you stop it manually. If you set the field to `stopped`, the solution will stop but not start the applicable instance. The instance will not run until you manually start it.

Note that if you set the `override_status` field to `running` but use the `enforced` field to prevent an instance from being manually started outside of a running period, the solution will stop the instance. If you set the `override_status` field to `stopped` but use the `enforced` field to prevent an instance from being manually stopped during a running period, the solution will restart the instance.

Instance type

For Amazon EC2 instances only, a schedule allows you to specify an optional instance type for each period in a schedule. When you specify an instance type in the period, the solution will start the Amazon EC2 instance with the applicable instance type.

To specify an instance type, use the syntax `@`. For example, `weekends@t2.nano`. Note that if you specify an instance type for a period that schedules Amazon EC2 instances and Amazon RDS instances, the instance type will be ignored for Amazon RDS instances.

If the instance type of a running instance is different than the instance type specified for the period, the solution will stop the running instance and restart the instance with the specified instance type, if the specified instance type is compatible with the instance configuration of the running instance. For more information, refer to [Compatibility for Resizing Instances in the Amazon EC2 User Guide for Linux Instances](#).

Schedule definitions

The Instance Scheduler configuration table in Amazon DynamoDB contains schedule definitions. A schedule definition can contain the following fields:

Field	Description
<code>description</code>	An optional description of the schedule.
<code>hibernate</code>	Choose whether to hibernate Amazon EC2 instances running Amazon Linux. When this field is set to true, the scheduler will hibernate instances when it stops them. Note that your instances must turn on hibernation and must meet the hibernation prerequisites.
<code>enforced</code>	Choose whether to enforce the schedule. When this field is set to true, the scheduler will stop a running instance if it is manually started outside of the running period or it will start an instance if it is stopped manually during the running period.
<code>name</code>	The name used to identify the schedule. This name must be unique.

Field	Description
override_status	When this field is set to running, the instance will be started but not stopped until you stop it manually. When this field is set to stopped, the instance will be stopped but not started until you start it manually.
periods	The name of the periods that are used in this schedule. Enter the name(s) exactly as it appears in the period name field. You can also specify an instance type for the period using the syntax @. For example, weekdays@t2.large.
retain_running	Choose whether to prevent the solution from stopping an instance at the end of a running period if the instance was manually started before the beginning of the period.
ssm_maintenance_window	Choose whether to add an Systems Manager maintenance window as a running period. Enter the name of a maintenance window. To use this field, you must also set the use_maintenance_window parameter to true.
stop_new_instances	Choose whether to stop an instance the first time it is tagged if it is running outside of the running period. By default, this field is set to true.
timezone	The time zone the schedule will use. If no time zone is specified, the default time zone (UTC) is used. For a list of acceptable time zone values, refer to the TZ column of the List of TZ Database Time Zones.
use_maintenance_window	Choose whether to add an Amazon RDS maintenance window as a running period to an Amazon RDS instance schedule, or to add an Systems Manager maintenance window as a running period to an Amazon EC2 instance schedule. For more information, refer to Amazon RDS Maintenance Window and SSM Maintenance Window Field.

Field	Description
use_metrics	Choose whether to turn on CloudWatch metrics at the schedule level. This field overwrites the CloudWatch metrics setting you specified at deployment. Enabling this feature will incur charges of \$0.90/month per schedule or scheduled service.

Period rules

Period rules contain conditions that allow you to set the specific hours, days, and months an instance will run. A period rule can contain multiple conditions, but all conditions must be true for the Amazon Web Services Instance Scheduler to apply the appropriate start or stop action.

Start and stop times

The `begintime` and `endtime` fields define when the Instance Scheduler will start and stop instances. If you specify a start time only, the instance must be stopped manually. Note that if you specify a value in the `weekdays` field, the solution uses that value to determine when to stop the instance. For example, if you specify a `begintime` of 9 am with no `endtime` and a `weekdays` value of Monday through Friday, the instance will be stopped at 11:59 pm at the end of each day unless you have scheduled an adjacent period.

Similarly, if you specify a stop time only, the instance must be started manually. If you don't specify either time, the solution uses the days of the week, days of the month, or months rules to start and stop instances.

The `begintime` and `endtime` values for your period must be in the time zone specified in the schedule. If you do not specify a time zone in the schedule, the solution will use the time zone specified when you launch the solution.

If your schedule contains multiple periods, we recommend that you specify both a `begintime` and `endtime` in your period. If no time is specified, this solution will use the time specified in the other periods to determine when to start and stop your instances. For example, if in one period you specify a `begintime` of 9 am with no `endtime` because you want the instance to run until you manually stop it, Amazon Web Services Instance Scheduler will stop the instance at 00:00 am the following day. Amazon Web Services Instance Scheduler will evaluate whether to keep the instance started or stopped until it finds another period for that specific day.

If you start an instance before the specified start time, the instance will run until the end of the running period. For example, a user might define a period that starts an instance daily at 9 am and stops that instance at 5 pm.



If the user manually starts that instance at 5 am, the solution will stop the instance at 5 pm. Note that if you use the retain running field, the solution will not stop the instance at 5 pm.



If you stop an instance before the specified stop time, the instance will not run until the beginning of the next running period. Continuing from the previous example, if the user stops the instance at 1 pm on Wednesday, the solution will not start the instance until 9 am on Thursday.



Adjacent periods

The solution will not stop running instances if the schedule contains two adjacent running periods. For example, if you have a schedule with one period with an endtime of 11:59 pm and another period with a begintime of midnight the following day, the solution will not stop running instances, if there are no weekdays, monthdays, or months rules that stop the instances.

To implement a schedule that runs instances from 9 am Monday to 5 pm Friday, the solution requires three periods. The first period runs applicable instances from 9 am to 11:59 pm Monday. The second period runs the instances from midnight Tuesday to 11:59 pm Thursday. The third period runs the instances from midnight Friday to 5 pm Friday. For more information, refer to [Sample schedule](#).

Days of the week

The weekdays field defines which days during the week an instance will run. You can specify a list of days, a range of days, the nth occurrence of that day in a month, or the last occurrence of that day in a month. The solution supports abbreviated day names (Mon) and numbers (0). For more information, refer to Step 2.

Days of the month

The monthdays field defines which days during the month an instance will run. You can specify a list of days, a range of days, every nth day of the month, the last day of the month, or the nearest weekday to a specific date. For more information, refer to Step 2.

Months

The months field defines which months an instance will run. You can specify a list of months, a range of months, or every nth month. The solution supports abbreviated month names (Jan) and numbers (1). For more information, refer to Step 2.

Period definitions

The Instance Scheduler configuration table in Amazon DynamoDB contains period definitions. A period definition can contain the following fields. Note that some fields support Cron non-standard characters.

Important

You must specify at least one of the following items: begintime, endtime, weekdays, months, or monthdays.

Field	Description
begintime	The time, in HH:MM format, that the instance will start.
description	An optional description of the period rule
description	An optional description of the period rule
endtime	The time, in HH:MM format, that the instance will stop.

Field	Description
months	<p>Enter a comma-delimited list of months, or a hyphenated range of months, during which the instance will run. For example, enter jan, feb, mar or 1, 2, 3 to run an instance during those months. Or, you can enter jan-mar or 1-3.</p> <p>You can also schedule an instance to run every nth month or every nth month in a range. For example, enter Jan/3 or 1/3 to run an instance every third month starting in January. Enter Jan-Jul/2 to run every other month from January to July.</p>
monthdays	<p>Enter a comma-delimited list of days of the month, or a hyphenated range of days, during which the instance will run. For example, enter 1, 2, 3 or 1-3 to run an instance during the first three days of the month. You can also enter multiple ranges. For example, enter 1-3, 7-9 to run an instance from the 1st to the 3rd and the 7th through the 9th. You can also schedule an instance to run every nth day of the month or every nth day of the month in a range. For example, enter 1/7 to run an instance every seventh day starting on the 1st. Enter 1-15/2 to run an instance every other day from the 1st to the 15th. Enter L to run an instance on the last day of the month.</p> <p>Enter a date and W to run an instance on the nearest weekday to the specified date. For example, enter 15W to run an instance on the nearest weekday to the 15th.</p>
name	The name used to identify the period rule. This name must be unique.
weekdays	<p>Enter a comma-delimited list of days of the week, or a range of days of the week, during which the instance will run. For example, enter 0, 1, 2 or 0-2 to run an instance Monday through Wednesday. You can also enter multiple ranges. For example, enter 0-2, 4-6 to run an instance every day except Thursday. You can also schedule an instance to run every nth occurrence of a weekday in the month. For example, enter Mon#1 or 0#1 to run an instance the first Monday of the month. Enter a day and L to run an instance on the last occurrence of that weekday in the month. For example, enter friL or 4L to run an instance on the last Friday of the month.</p>

When a period rule contains multiple conditions, note that all conditions must be true for the Amazon Web Services Instance Scheduler to apply the appropriate action. For example, a period rule that contains a weekdays field with a value of Mon#1 and a months field with a value of Jan/3 will apply the action on the first Monday of the quarter.

Cross-Account instance scheduling

This solution includes a template (instance-scheduler-remote) that creates the Amazon Identity and Access Management (IAM) roles necessary to start and stop instances in secondary accounts. You can review and modify permissions in the remote template before you launch the stack.

To apply automated start-stop schedules to resources in secondary accounts, launch the main solution template (instance-scheduler) in the primary account. Then, launch the remote template (instance-scheduler-remote) in each applicable secondary account. When each remote stack is launched, it creates a cross-account role Amazon Resource Name (ARN). Update the main solution stack with each cross-account role ARN by entering the appropriate ARN(s) in the Cross-account roles parameter to allow the Amazon Web Services Instance Scheduler to perform start and stop actions on instances in the secondary accounts.

Amazon Systems Manager Parameter Store

You can use Systems Manager Parameter Store to store cross-account role ARNs. You can store cross-account ARNs as a list parameter where every item is an ARN, or as a string parameter that contains a comma-delimited list of ARNs. The parameter has the format {param:name} where the name is the name of the parameter in the parameter store.

To leverage this feature, you must launch the Instance Scheduler stack in the same account as your parameter store.

Automated tagging

The Instance Scheduler can automatically add tags to all instances it starts or stops. You can specify a list of tag names or tagname=tagvalue pairs in the Started tags and Stopped tags parameters. The solution also includes macros that allow you to add variable information to the tags:

- {scheduler}: The name of the scheduler stack
- {year}: The year (four digits)
- {month}: The month (two digits)
- {day}: The day (two digits)
- {hour}: The hour (two digits, 24-hour format)
- {minute}: The minute (two digits)
- {timezone}: The time zone

The following table gives examples of different inputs and the resulting tags.

Example Parameter Input	Instance Scheduler Tag
ScheduleMessage=Started by scheduler {scheduler}	ScheduleMessage=Started by scheduler MyScheduler
ScheduleMessage=Started on {year}/{month}/ {day}	ScheduleMessage=Started on 2017/07/06
ScheduleMessage=Started on {year}/{month}/ {day} at {hour}:{minute}	ScheduleMessage=Started on 2017/07/06 at 09:00
ScheduleMessage=Started on {year}/{month}/ {day} at {hour}:{minute} {timezone}	ScheduleMessage=Started on 2017/07/06 at 09:00 UTC

When you use the Started tags parameter, the tags are automatically deleted when the scheduler stops the instance. When you use the Stopped tags parameter, the tags are automatically deleted when the instance is started.

Scheduler command line interface

The Amazon Web Services Instance Scheduler includes a command line interface (CLI) that provides commands for configuring schedules and periods. The CLI allows customers to estimate cost savings for a given schedule. The cost estimates provided by the schedule CLI are for approximation purposes only. For more information about configuring and using the scheduler CLI, refer to Scheduler CLI.

Security

When you build systems on Amazon Web Services infrastructure, security responsibilities are shared between you and Amazon Web Services. This shared model reduces your operational burden because Amazon Web Services operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate.

Amazon Key Management System

The solution creates an Amazon managed Custom Master Key (CMK), which is used to configure server-side encryption for the SNS topic and the DynamoDB tables.

Amazon Identity Access Management

The Lambda function created by the solution requires permissions to start/stop both EC2 and RDS instances, modify instance attributes, update tags for the instances among other permissions. All the necessary permissions are provided by the solution to Lambda service role created as part of the solution template. Additionally, the Lambda service role also has access to get/put SSM parameters, access to CloudWatch log groups, KMS key encryption/decryption, and publish messages to SNS topic. For detailed information about each permission provided to the service role, refer to the CloudFormation templates.

Design considerations

Partial automation

Users have the option to implement a partially automated solution by default (i.e., configure start-only or stop-only actions). An example of this is a team that needs the flexibility to stop instances at varying times (manually) but must start those instances simultaneously each morning, or vice versa.

Instance shutdown behavior

Amazon EC2

This solution is designed to automatically stop Amazon Elastic Compute Cloud (Amazon EC2) instances and assumes that instance shutdown behavior is set to Stop, not Terminate. Note that you cannot restart an Amazon EC2 instance after it is terminated.

By default, Amazon EC2 instances are configured to stop, not terminate, when shut down, but you can modify this behavior. Therefore, make sure that the instances you control using the Amazon Web Services Instance Scheduler are configured with a Stop shutdown behavior, otherwise they will be terminated.

Amazon RDS

Note that this solution is designed to automatically stop, not delete, Amazon Relational Database Service (Amazon RDS) instances. You can use the Create RDS Instance Snapshot Amazon CloudFormation template parameter to create snapshots of RDS instances before the solution stops the instances. Snapshots are kept until the next time the instance is stopped and a new snapshot is created. Note that snapshots are not available for Amazon Aurora clusters.

You can use the Schedule Aurora Clusters template parameter to start and stop RDS instances that are part of an Aurora cluster or that manage Aurora databases. You must tag the cluster (not the individual instances) with the tag key you defined during initial configuration and the schedule name as the tag value to schedule that cluster.

For more information about limitations to starting and stopping an Amazon RDS instance, refer to [Stopping an Amazon RDS DB Instance Temporarily](#) in the Amazon RDS User Guide.

When an Amazon RDS instance is stopped, the cache is cleared which might lead to slower performance when the instance is restarted.

Amazon RDS maintenance window

Every Amazon RDS instance has a weekly maintenance window during which any system changes are applied. During the maintenance window, Amazon RDS will automatically start instances that have been stopped for more than seven days to apply maintenance. Note that Amazon RDS will not stop the instance once the maintenance event is complete.

The Instance Scheduler allows you to specify whether to add the preferred maintenance window of an Amazon RDS instance as a running period to its schedule. The solution will start the instance at the beginning of the maintenance window and stop the instance at the end of the maintenance window if no other running period specifies that the instance should run, and if the maintenance event is completed.

If the maintenance event is not completed by the end of the maintenance window, the instance will run until the scheduling interval after the maintenance event is completed. For more information about the Amazon RDS maintenance window, refer to [Maintaining a database instance](#) in the Amazon RDS User Guide.

Global configuration settings

When you deploy the Instance Scheduler's Amazon CloudFormation template, global configuration settings are stored in an Amazon DynamoDB table (ConfigTable). To modify these settings, update the solution stack using the Amazon CloudFormation template. Do not change these settings in the DynamoDB table.

Performance

If the Instance Scheduler Amazon Lambda function does not process all scheduled instances before its next invocation, the solution logs the error in Amazon CloudWatch Logs, and sends an Amazon Simple Notification Service (Amazon SNS) notification to the error SNS topic. To help ensure that all instances are processed before the next invocation, you can change the default interval at which the Lambda function runs or launch multiple deployments of the solution with different tag names.

If you increase the default interval, this might reduce the granularity of your schedules. For example, a Lambda function set to run on a fifteen-minute interval will only perform start and stop actions every 15 minutes.

To schedule a large number of instances, we recommend using an interval of at least five minutes and increasing the memory size of the Instance Scheduler's main Amazon Lambda function using the Memory Size parameter.

Encrypted Amazon EBS volumes

If your Amazon EC2 instances contain encrypted Amazon Elastic Block Store (Amazon EBS) volumes, you must grant the Instance Scheduler permission to use the customer master key (CMK) to start and stop instances. Add the kms:CreateGrant permission to the Instance Scheduler role (stackname-SchedulerRole-id).

Regional deployments

You can deploy Amazon Web Services Instance Scheduler in any Region. Once deployed, the Instance Scheduler applies the appropriate start or stop actions to tagged Amazon EC2 and Amazon RDS instances in all Regions of your account. If you use cross-account instance scheduling, the solution will apply actions to instances in all Regions in all accounts.

Important

Instance Scheduler actions will affect appropriately tagged instances in all Regions of your account, even though the Lambda function is running in a single Region.

You can use multiple deployments of the solution to schedule a large number of instances, or instances in many accounts and Regions. When you deploy multiple schedulers, use a different tag name for each stack, and configure a set of non-overlapping Regions for each deployment. Each deployment checks every instance in every Region in an account for the tag key that identifies resources it should schedule. If the Regions for multiple deployments overlap, each instance will be checked by multiple deployments.

Note

Amazon Web Services Instance Scheduler has been validated in the following Regions: - cn-north-1 (Beijing) - cn-northwest-1 (Ningxia)

Logging and notifications

Amazon Web Services Instance Scheduler leverages Amazon CloudWatch Logs for logging. This solution logs processing information for each tagged instance; the results of the period

rule evaluation for the instance; the desired state of the instance during that period; the applied action; and debugging messages. For more information, refer to [Solution resources](#).

Warning and error messages are also published to a solution-created Amazon SNS topic which sends messages to a subscribed email address (refer to [Subscribing to an Amazon SNS Topic](#) in the [Amazon SNS Developer Guide](#)). You can find the name of the Amazon SNS topic in the [Outputs](#) tab of the solution stack.

CloudFormation templates

CloudFormation Template Resource

This solution uses Amazon CloudFormation to automate the deployment of Amazon Web Services Instance Scheduler on the Amazon Web Services Cloud. It includes the following Amazon CloudFormation templates, which you can download before deployment:

[aws-instance-scheduler.template](#): Use this template to launch the Instance Scheduler and all associated components. The default configuration deploys an Amazon Lambda function, an Amazon DynamoDB table, an Amazon CloudWatch event, and CloudWatch custom metrics, but you can also customize the template based on your specific needs.

[aws-instance-scheduler-remote.template](#): Use this template to configure permissions for instances in secondary accounts. The default configuration creates the Amazon Identity and Access Management (IAM) roles necessary to start and stop instances in a secondary account.

Automated deployment

Before you launch the automated deployment, please review the architecture, configuration, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy Instance Scheduler into your account.

Time to deploy: Approximately five (5) minutes

Update the stack

If you have previously deployed the solution, follow this procedure to update the Amazon Web Services Instance Scheduler CloudFormation stack to get the latest version of the solution's framework.

1. Sign in to the [Amazon CloudFormation console](#), select your existing aws-instance-scheduler CloudFormation stack, and select Update.
2. Select Replace current template.
3. Under Specify template:
4. Select Amazon S3 URL.
5. Copy the link of the latest template.
6. Paste the link in the Amazon S3 URL box.
7. Verify that the correct template URL shows in the Amazon S3 URL text box, and choose Next. Choose Next again.
8. Under Parameters, review the parameters for the template and modify them as necessary. Refer to Step 1. Launch the Stack for details about the parameters.
9. Choose Next.
- 10 On the Configure stack options page, choose Next.
- 11 On the Review page, review and confirm the settings. Be sure to check the box acknowledging that the template might create Amazon Identity and Access Management (IAM) resources.
- 12 Choose View change set and verify the changes.
- 13 Choose Update stack to deploy the stack.

You can view the status of the stack in the CloudFormation console in the Status column. You should see a status of UPDATE_COMPLETE in approximately 15 minutes.

Deployment overview

The procedure for deploying this architecture on Amazon Web Services consists of the following steps. For detailed instructions, follow the links for each step.

Step 1. Launch the instance scheduler stack

- Launch the Amazon CloudFormation template into your Amazon Web Services account
- Enter values for the required parameter: Stack Name
- Review the other template parameters, and adjust if necessary

Step 2. Configure periods

- Create a period and set the applicable fields for the period

Step 3. Configure schedules

- Create a schedule and set the applicable fields for the schedule

Step 4. Tag your instances

- Apply the custom tag to applicable resources

Step 5. Launch the remote stack in secondary accounts (Optional)

- Launch the Amazon CloudFormation template into your Amazon Web Services account
- Enter values for the required parameter: Primary account

Step 1. Launch the instance scheduler stack

This automated Amazon CloudFormation template deploys the Instance Scheduler in Amazon Lambda, and configures related components. Please make sure that you've verified the settings for your instances before launching the stack.

Note

You are responsible for the cost of the services used while running this solution. For more details, visit to the Cost section in this guide, and refer to the pricing webpage for each service used in this solution.

1. Sign in to the Amazon Web Services Management Console and click [the link](#) to launch the aws-instance-scheduler Amazon CloudFormation template. You can also [download the template](#) as a starting point for your own implementation.
2. The template is launched in the US East (N. Virginia) Region by default. To launch the Instance Scheduler in a different Region, use the region selector in the console navigation bar.
3. On the Select Template page, verify that you selected the correct template and choose Next.
4. On the Specify Details page, assign a name to your solution stack.

Note

The stack name cannot contain more than 128 characters.

1. Under Parameters, review the parameters for the template, and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Instance Scheduler tag name	Schedule	This tag identifies instances to receive automated actions, and also allows for custom start-stop parameters. If you choose to modify the default value, make sure to assign a name that will be easy to apply consistently and correctly across all necessary instances.
Service(s) to schedule	EC2	The services to schedule. Select EC2, RDS, or Both.
Schedule Aurora Clusters	No	Choose whether to schedule Amazon Aurora clusters. To turn on Aurora cluster scheduling, you must select RDS or Both for the Service(s) to schedule parameter.
Create RDS instance snapshot	Yes	Choose whether to create a snapshot before stopping RDS instances. Note that snapshots are not available for Amazon Aurora clusters.

Parameter	Default	Description
Scheduling enabled	Yes	Select No to temporarily turn off scheduling.
Region(s)	"Optional input"	List of Regions where instances will be scheduled. For example, cn-north-1, cn-northwest-1. Note that if you leave this parameter blank, the solution will use the current Region.
Default time zone	UTC	Default time zone for schedules. For a list of acceptable time zone values, refer to the TZ column of the List of TZ Database Time Zones.
Cross-account roles	"Optional input"	Comma-delimited list of cross-account roles. For example, arn:aws-cn:iam::111122223333:role/stacknameSchedulerCrossAccountRole. If you store your cross-account ARNs in the Amazon Systems Manager Parameter Store, use the format {param:name}. For more information, refer to Amazon Systems Manager Parameter Store. Enter the secondary account CrossAccountRoleArn value(s) in this parameter.
This account	Yes	Select Yes to allow the task to select resources in this account. Note if you set this parameter to No, you must configure cross-account roles.
Frequency	5	The frequency in minutes at which the Amazon Lambda function runs. Select 1, 2, 5, 10, 15, 30, or 60.
Enable CloudWatch Metrics	No	Choose whether to collect data using CloudWatch Metrics for all schedules. You can override this default setting for an individual schedule when you configure it (refer to Step 3). Enabling this feature will incur charges of \$0.90/month per schedule or scheduled service.
Memory Size	128	The memory size of the solution's main Amazon Lambda function. Increase the default size to schedule a large number of Amazon EC2 and Amazon RDS instances.
Enable CloudWatch Logs	No	Choose whether to log detailed information in CloudWatch Logs.

Parameter	Default	Description
Enable SSM maintenance windows	No	Allow the solution to load SSM maintenance windows, so that they can be used for Amazon EC2 instance scheduling.
Log retention days	30	The log retention period in days
Started tags	"Optional input"	Tags to add to started instances. Use a list of tagname=tagvalue pairs.
Stopped tags	"Optional input"	Tags to add to stopped instances. Use a list of tagname=tagvalue pairs.
Send anonymous usage data	Yes	Send anonymous data to Amazon Web Services to help us understand solution usage and related cost savings across our customer base as a whole. To opt out of this feature, select No. For more information, refer to the Collection of operational metrics.

1. Choose Next.
2. On the Options page, choose Next.
3. On the Review page, review and confirm the settings. Check the box acknowledging that the template will create Amazon Identity and Access Management (IAM) resources.
4. Choose Create to deploy the stack.

You can view the status of the stack in the Amazon CloudFormation console in the Status column. You should see a status of CREATE_COMPLETE in roughly five (5) minutes.

Step 2. Configure periods

When you deploy the Amazon CloudFormation template, the solution creates an Amazon DynamoDB table that contains sample period rules and schedules that you can use as a reference to create your own custom period rules and schedules.

To create a period rule, you can use the Amazon DynamoDB console, the scheduler CLI, or the Amazon CloudFormation custom resource that you can create. For details about this stack resource type, refer to Custom resource.

Note

If you use the custom resource to create a period, you must not use the DynamoDB console or scheduler CLI to delete or modify that period. If you do, you will create a conflict between the stored parameters in the stack and the values in the table. Also, do not use periods configured using the custom resource in schedules created using the DynamoDB console or the scheduler CLI.

To create a period rule in DynamoDB, modify one of the periods in the configuration table (ConfigTable). To create a period in the scheduler CLI, use the applicable commands. To create a period using the custom resource, add the applicable fields to the solution's custom resource.

For an example period configuration, refer to [Sample schedule](#).

Step 3. Configure schedules

To create a schedule, you can use the Amazon DynamoDB console, the scheduler CLI, or the Amazon CloudFormation custom resource that you can create. For details about this stack resource type, refer to [Custom resource](#).

Note

If you use the custom resource to create a schedule, you must not use the DynamoDB console or scheduler CLI to delete or modify that schedule. If you do, you will create a conflict between the stored parameters in the stack and the values in the table.

To create a schedule in DynamoDB, modify one of the schedules in the configuration table (ConfigTable). To create a schedule in the scheduler CLI, use the applicable commands. To create a schedule using the custom resource, add the applicable fields to the solution's custom resource.

For an example schedule configuration, refer to [Sample schedule](#).

Step 4. Tag your instances

When you deployed the Amazon CloudFormation template, you defined the name (tag key) for the solution's custom tag. For the Instance Scheduler to recognize an Amazon EC2 or Amazon RDS instance, the tag key on that instance must match the custom tag name stored in the Amazon DynamoDB table. Therefore, it is important that you apply tags consistently and correctly to all applicable instances. You can continue to use existing tagging strategies for your instances while using this solution. For more information, refer to [Tagging Your Amazon EC2 Resources](#) and [Tagging Your Amazon RDS Resources](#).

On the Amazon Web Services Management Console, use the Tag Editor to apply or modify tags for multiple resources at a time. You can also apply and modify tags manually in the console.

Setting the tag value

When you apply a tag to a resource, use the tag key you defined during initial configuration. Set the tag key to Schedule and set the tag value to the same scheduler name in the Amazon DynamoDB table to schedule your resource. You can also update the scheduler name in the Amazon DynamoDB table.

Note

For Amazon RDS instances, the tag value can be from 1 to 256 Unicode characters in length and cannot be prefixed with "aws:". The string can contain only the set of Unicode letters, digits, white-space, ', ', '/', '=', '+', '-' (*Java regex*: `"^([\p{L}\p{Z}\p{N}:/=+\-]*)$"`). For more information, refer to [Tagging Your Amazon RDS Resources](#).

Step 5. Launch the remote stack in secondary accounts (Optional)

This automated Amazon CloudFormation template configures secondary account permissions.

1. Navigate to the Amazon Web Services Instance Scheduler stack Outputs tab and copy the Value of SchedulerRole.
2. Sign in to the Amazon Web Services Management Console of the applicable secondary account and click [the link](#) to below to launch the aws-instance-scheduler-remote Amazon CloudFormation template. You can also [download the template](#) as a starting point for your own implementation.

3. The template is launched in the US East (N. Virginia) Region by default. To launch the Instance Scheduler in a different Region, use the region selector in the console navigation bar.
4. On the Select Template page, verify that you selected the correct template and choose Next.
5. On the Specify Details page, assign a name to your remote stack.
6. Under Parameters, review the parameter for the template, and modify it.

Parameter	Default	Description
Primary account	"Requires Input"	Enter the account number of the account with the primary Instance Scheduler stack. This parameter gives the solution permission to schedule Amazon EC2 and Amazon RDS instances in this account.

1. Choose Next.
2. On the Options page, choose Next.
3. On the Review page, review and confirm the settings. Be sure to check the box acknowledging that the template will create Amazon Identity and Access Management (IAM) resources.
4. Choose Create to deploy the stack.

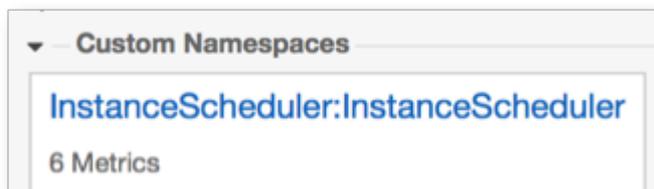
You can view the status of the stack in the Amazon CloudFormation console in the Status column. You should see a status of CREATE_COMPLETE in approximately five minutes.

Amazon CloudWatch metrics

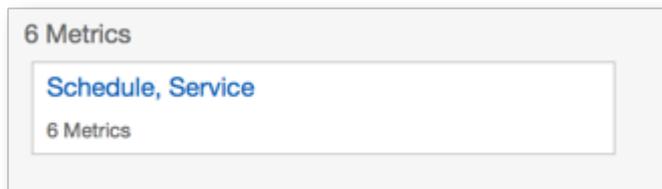
This solution creates a new custom Amazon CloudWatch metric (stackname:InstanceScheduler). Each time the Amazon Lambda function runs, it updates the metric data for each applicable instance and then applies the appropriate start or stop action. This data includes the name of the schedule, the number of instances associated with that schedule, and the number of running instances.

View Instance Scheduler Metrics

1. In the Amazon Web Services Management Console, open the Amazon CloudWatch console.
2. In the Custom Namespaces drop-down field, choose stackname:InstanceScheduler.



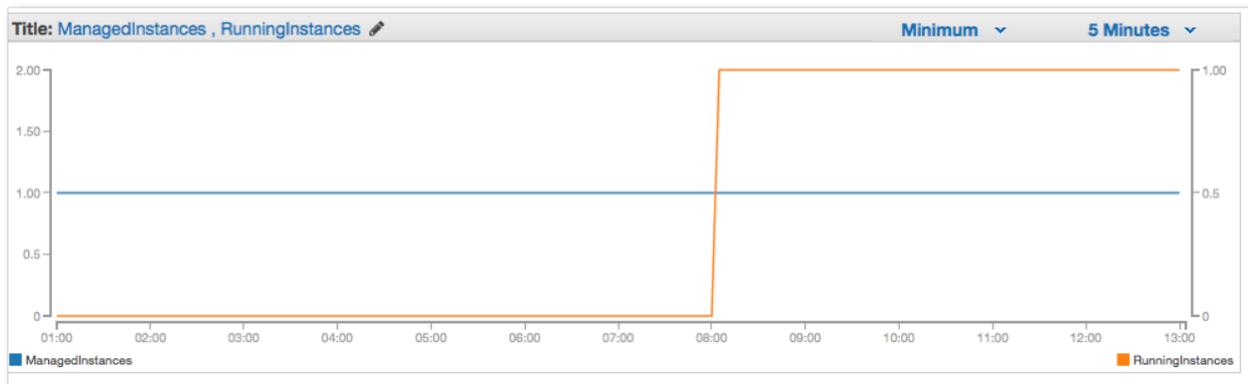
1. Select the schedule and service dimensions.



1. Select the schedule and service that you want to view the status of.

All metrics			Graphed metrics	Graph options
All > InstanceScheduler:InstanceScheduler > Schedule, Service <input type="text" value="Search for any metric, dimension or resource id"/>				
<input type="checkbox"/>	Schedule (6)	Service	Metric Name	
<input type="checkbox"/>	running	ec2	RunningInstances	
<input type="checkbox"/>	running	ec2	ManagedInstances	
<input type="checkbox"/>	stopped	ec2	ManagedInstances	
<input type="checkbox"/>	stopped	ec2	RunningInstances	
<input type="checkbox"/>	stopped	rds	ManagedInstances	
<input type="checkbox"/>	stopped	rds	RunningInstances	

At the bottom of the page, an individual graph will appear for each instance you selected, as shown in the following example. Note that a value of 0 is a stopped instance and a value of 1.00 is a running instance.



Contributors

The following individuals contributed to this document:

- Arie Leeuwesteijn
- Mahmoud ElZayet
- Ruald Andreae
- Nikhil Reddy
- Paul Li

Revision

Date	Change
May 2020	Initial release, version 1.3.0
March 2022	Update to version 1.4.0, support multiple deployment

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current Amazon Web Services product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from Amazon Web Services and its affiliates, suppliers or licensors. Amazon Web Services products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of Amazon Web Services to its customers are controlled by Amazon Web Services agreements, and this document is not part of, nor does it modify, any agreement between Amazon Web Services and its customers.

The Amazon Web Services Instance Scheduler solution is licensed under the terms of the Apache License Version 2.0 available at <https://www.apache.org/licenses/LICENSE-2.0>