- `ELEMENTS_TOO_MANY` – Occurs when a particular attribute in an effective policy exceeds the allowed limit, such as when more than 10 rules are given for a backup plan.
- `ELEMENTS_TOO_FEW` – Occurs when a particular attribute in an effective policy does not meet the minimum limit, such as when no region is defined for a backup plan.
- `KEY_REQUIRED` – Occurs when a required configuration is missing in the effective policy, such as when a backup plan is missing a backup rule.

AWS Organizations validates effective policies before applying them to the accounts in your organization. This auditing process is especially beneficial if you have a large organization structure, and if your organization's policies are managed by more than one team.

Declarative policies

Declarative policies allow you to centrally declare and enforce your desired configuration for a given AWS service at scale across an organization. Once attached, the configuration is always maintained when the service adds new features or APIs. Use declarative policies to prevent noncompliant actions. For example, you can block public internet access to Amazon VPC resources across your organization.

The key benefits of using declarative policies are:

- **Ease of use**: You can enforce the baseline configuration for an AWS service with a few selections in the AWS Organizations and AWS Control Tower consoles or with a few commands using the AWS CLI & AWS SDKs.
- **Set once and forget**: The baseline configuration for an AWS service is always maintained, even when the service introduces new features or APIs. The baseline configuration is also maintained when new accounts are added to an organization or when new principals and resources are created.
- **Transparency**: The account status report allows you to review the current status of all attributes supported by declarative policies for the accounts in scope. You can also create customizable error messages, which can help administrators redirect end users to internal wiki pages or provide a descriptive message that can help end users understand why an action failed.

For a full list of supported AWS services and attributes, see Supported AWS services and attributes.

Topics

- How declarative policies work
- Custom error messages for declarative policies
- Account status report for declarative policies
- Supported AWS services and attributes
- Getting started with declarative policies
- Best practices for using declarative policies
- Generating the account status report for declarative policies
- Declarative policy syntax and examples

How declarative policies work

Declarative policies are enforced in the service's control plane, which is an important distinction from authorization policies such as service control policies (SCPs) and resource control policies (RCPs). While authorization policies regulate access to APIs, declarative policies are applied directly at the service level to enforce durable intent. This ensures that the baseline configuration is always enforced, even when new features or APIs are introduced by the service.

The following table helps illustrate this distinction and provides some use cases.

	Service control policies	**Resource control policies**	**Declarative policies**		
Why?	To centrally define and enforce consisten t access controls on principals (such as IAM users and IAM roles) at scale.	To centrally define and enforce consisten t access controls on resources at scale	To centrally define and enforce the baseline configura tion for AWS services at scale.		

	Service control policies	Resource control policies	Declarative policies		
How?	By controlling the maximum available access permissions of principals at an API level.	By controlling the maximum available access permissions for resources at an API level.	By enforcing the desired configuration of an AWS service without using API actions.		
Governs service-linked roles?	No	No	Yes		
Feedback mechanism	Non-customizable access denied SCP error.	Non-customizable access denied RCP error.	Customizable error message. For more information, see Custom error messages for declarative policies.		
Example policy	Deny member accounts from leaving the organization	Restrict access to only HTTPS connections to your resources	Allowed Images Settings		

After you have created and attached a declarative policy, it is applied and enforced across your organization. Declarative policies can be applied to an entire organization, organizational units (OUs), or accounts. Accounts joining an organization will automatically inherit the declarative policy in the organization. For more information, see Understanding management policy inheritance.

The *effective policy* is the set of rules that are inherited from the organization root and OUs along with those directly attached to the account. The effective policy specifies the final set of rules that apply to the account. For more information, see Viewing effective management policies.

If a declarative policy is detached, the attribute state will roll back to its previous state before the declarative policy was attached.

Custom error messages for declarative policies

Declarative policies allow you to create custom error messages. For example, if an API operation fails due to a declarative policy, you can set the error message or provide a custom URL, such as a link to an internal wiki or a link to a message that describes the failure. If you do not specify a custom error message, AWS Organizations provides the following default error message: `Example: This action is denied due to an organizational policy in effect.`

You can also audit the process of creating declarative policies, updating declarative policies, and deleting declarative policies with AWS CloudTrail. CloudTrail can flag API operation failures due to declarative policies. For more information, see Logging and monitoring.

> **Important**
>
> Do not include *personally identifiable information (PII)* or other sensitive information in a custom error message. PII includes general information that can be used to identify or locate an individual. It covers records such as financial, medical, educational, or employment. PII examples include addresses, bank account numbers, and phone numbers.

Account status report for declarative policies

The *account status report* allows you to review the current status of all attributes supported by declarative policies for the accounts in scope. You can choose the accounts and organizational units (OUs) to include in the report scope, or choose an entire organization by selecting the root.

This report helps you assess readiness by providing a Region breakdown and if the current state of an attribute is *uniform across accounts* (through the `numberOfMatchedAccounts`) or *inconsistent* (through the `numberOfUnmatchedAccounts`). You can also see the *most frequent value*, which is the configuration value that is most frequently observed for the attribute.

In Figure 1, there is a generated account status report, which shows uniformity across accounts for the following attributes: VPC Block Public Access and Image Block Public Access. This means that, for each attribute, all the accounts in scope have the same configuration for that attribute.

The generated account status report shows inconsistent accounts for the following attributes: Allowed Images Settings, Instance Metadata defaults, Serial Console Access, and Snapshot Block Public Access. In this example, each attribute with an inconsistent account is due to there being one account with a different configuration value.

If there is a most frequent value, that is displayed in its respective column. For more detailed information of what each attribute controls, see Declarative policy syntax and example policies.

You can also expand an attribute to see a Region breakdown. In this example, Image Block Public Access is expanded and in each Region, you can see that there is also uniformity across accounts.

The choice to attach a declarative policy for enforcing a baseline configuration depends on your specific use case. Use the account status report to help you assess your readiness before attaching a declarative policy.

For more information, see Generating the account status report.

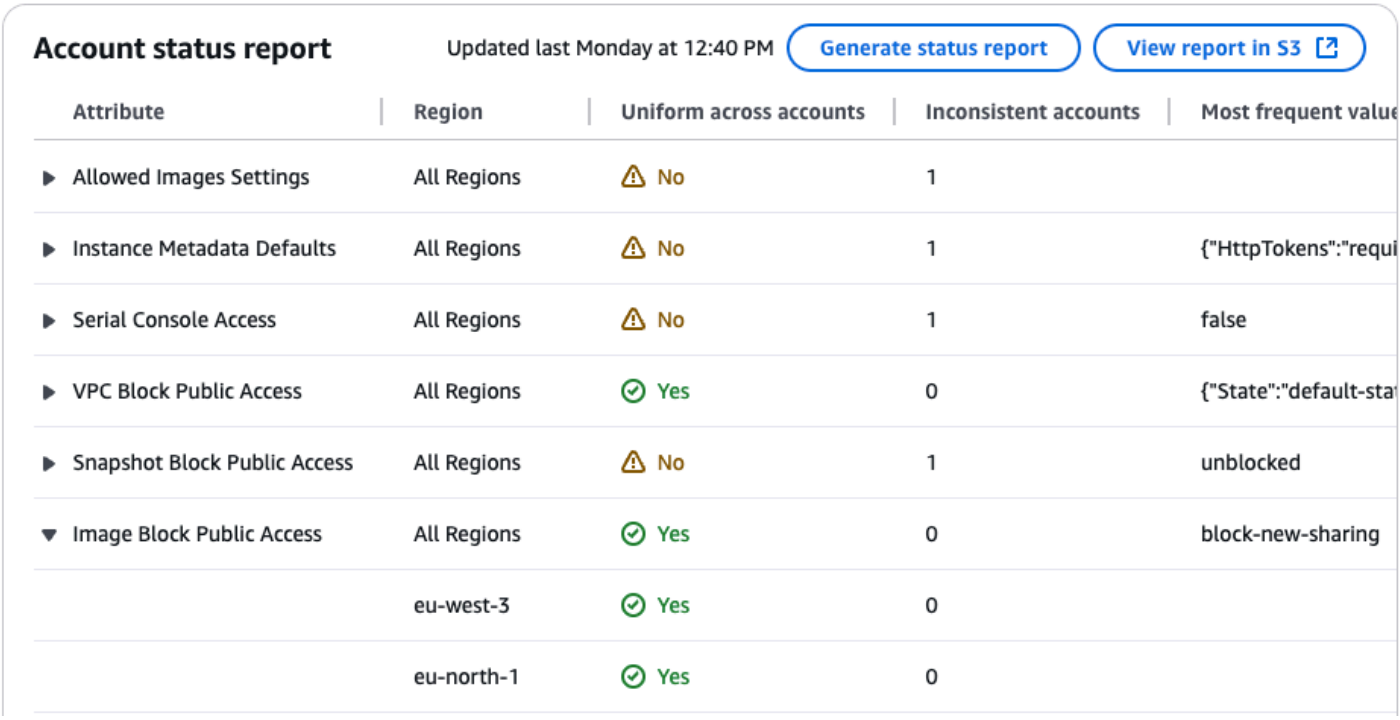

Figure 1: Example account status report with uniformity across accounts for VPC Block Public Access and Image Block Public Access.

Supported AWS services and attributes

Supported attributes for declarative policies for EC2

The following table displays the attributes supported for Amazon EC2 related services.

Declarative policies for EC2

AWS service	Attribute	Policy effect	Policy contents	More information
Amazon VPC	VPC Block Public Access	Controls if resources in Amazon VPCs and subnets can reach the internet through internet gateways (IGWs).	View policy	For more information, see Block public access to VPCs and subnets in the *Amazon VPC User Guide*.

AWS service	Attribute	Policy effect	Policy contents	More information
Amazon EC2	Serial Console Access	Controls if the EC2 serial console is accessible.	View policy	For more information, see Configure access to the EC2 Serial Console in the *Amazon Elastic Compute Cloud User Guide*.
	Image Block Public Access	Controls if Amazon Machine Images (AMIs) are publicly sharable.	View policy	For more information, see Understand block public access for AMIs in the *Amazon Elastic Compute Cloud User Guide*.
	Allowed Images Settings	Controls the discovery and use of Amazon Machine Images (AMI) in Amazon EC2 with Allowed AMIs.	View policy	For more information, see Amazon Machine Images (AMIs) in the *Amazon Elastic Compute Cloud User Guide*.

AWS service	Attribute	Policy effect	Policy contents	More information
	Instance Metadata Defaults	Controls IMDS defaults for all new EC2 instances launches.	View policy	For more information, see Configure instance metadata options for new instances in the *Amazon Elastic Compute Cloud User Guide*.
Amazon EBS	Snapshot Block Public Access	Controls if Amazon EBS snapshots are publicly accessible.	View policy	For more information, see Block public access for Amazon EBS snapshots in the *Amazon Elastic Block Store User Guide*.

Getting started with declarative policies

Follow these steps to get started using declarative policies.

1. Learn about the permissions you must have to perform declarative policy tasks.
2. Enable declarative policies for your organization.

> **Note**
>
> **Enabling trust access is required**
> You must enable trusted access for the service where the declarative policy will enforce a baseline configuration. This creates a read-only service-linked role that is used to

generate the account status report of what the existing configuration is for accounts across your organization.

Using the console

If you use the Organizations console, this step is a part of the process for enabling declarative policies.

Using the AWS CLI

If you use the AWS CLI, there are two separate APIs:

- EnablePolicyType, which you use to enable declarative policies.
- EnableAWSServiceAccess, which you use to enable trusted access.

For more information on how to enable trusted access for a specific service with the AWS CLI see, AWS services that you can use with AWS Organizations.

3. Run the account status report.
4. Create a declarative policy.
5. Attach the declarative policy to your organization's root, OU, or account.
6. View the combined effective declarative policy that applies to an account.

For all of these steps, you sign in as an IAM user, assume an IAM role, or sign in as the root user (not recommended) in the organization's management account.

Other information

- Learn declarative policy syntax and see example policies

Best practices for using declarative policies

AWS recommends the following best practices for using declarative policies.

Leverage readiness assessments

Use the declarative policy *account status report* to assess the current status of all attributes supported by declarative policies for the accounts in scope. You can choose the accounts and organizational units (OUs) to include in the report scope, or choose an entire organization by selecting the root.

This report helps you assess readiness by providing a Region breakdown and if the current state of an attribute is *uniform across accounts* (through the `numberOfMatchedAccounts`) or *inconsistent*

(through the `numberOfUnmatchedAccounts`). You can also see the *most frequent value*, which is the configuration value that is most frequently observed for the attribute.

The choice to attach a declarative policy for enforcing a baseline configuration depends on your specific use case.

For more information and an illustrative example, see Account status report for declarative policies.

Start small and then scale

To simplify debugging, start with a test policy. Validate the behavior and impact of each change before making the next change. This approach reduces the number of variables you have to account for when an error or unexpected result occurs.

For example, you can start with a test policy attached to a single account in a noncritical test environment. After you have confirmed that it works to your specifications, you can then incrementally move the policy up the organization structure to more accounts and more organizational units (OUs).

Establish review processes

Implement processes to monitor for new declarative attributes, evaluate policy exceptions, and make adjustments to maintain alignment with your organizational security and operational requirements.

Validate changes using `DescribeEffectivePolicy`

After you make a change to a declarative policy, check the effective policies for representative accounts below the level where you made the change. You can view the effective policy by using the AWS Management Console, or by using the DescribeEffectivePolicy API operation or one of its AWS CLI or AWS SDK variants. Ensure that the change you made had the intended impact on the effective policy.

Communicate and train

Ensure your organizations understand the purpose and impact of your declarative policies. Provide clear guidance on the expected behaviors and how to handle failures due to policy enforcement.

Generating the account status report for declarative policies

The *account status report* allows you to review the current status of all attributes supported by declarative policies for the accounts in scope. You can choose the accounts and organizational units (OUs) to include in the report scope, or choose an entire organization by selecting the root.

This report helps you assess readiness by providing a Region breakdown and if the current state of an attribute is *uniform across accounts* (through the `numberOfMatchedAccounts`) or *inconsistent* (through the `numberOfUnmatchedAccounts`). You can also see the *most frequent value*, which is the configuration value that is most frequently observed for the attribute.

The choice to attach a declarative policy for enforcing a baseline configuration depends on your specific use case.

For more information and an illustrative example, see Account status report for declarative policies.

Prerequisites

Before you can generate an account status report, you must perform the following steps

1. The `StartDeclarativePoliciesReport` API can only be called by the management account or delegated administrators for an organization.
2. You must have an S3 bucket before generating the report (create a new one or use an existing one), it must be in the same Region in which the request is made, and it must have an appropriate S3 bucket policy. For a sample S3 policy, see *Sample Amazon S3 policy* under Examples in the *Amazon EC2 API Reference*
3. You must enable trusted access for the service where the declarative policy will enforce a baseline configuration. This creates a read-only service-linked role that is used to generate the account status report of what the existing configuration is for accounts across your organization.

   Using the console

   For the Organizations console, this step is a part of the process for enabling declarative policies.

   Using the AWS CLI

   For the AWS CLI, use the EnableAWSServiceAccess API.

   For more information on how to enable trusted access for a specific service with the AWS CLI see, AWS services that you can use with AWS Organizations.

4. Only one report per organization can be generated at a time. Attempting to generate a report while another is in progress will result in an error.

Access the compliance status report

Minimum permissions

To generate a compliance status report, you need permission to run the following actions:

- `ec2:StartDeclarativePoliciesReport`
- `ec2:DescribeDeclarativePoliciesReports`
- `ec2:GetDeclarativePoliciesReportSummary`
- `ec2:CancelDeclarativePoliciesReport`
- `organizations:DescribeAccount`
- `organizations:DescribeOrganization`
- `organizations:DescribeOrganizationalUnit`
- `organizations:ListAccounts`
- `organizations:ListDelegatedAdministrators`
- `organizations:ListAWSServiceAccessForOrganization`
- `s3:PutObject`

Note

If your Amazon S3 bucket uses SSE-KMS encryption, you must also include the `kms:GenerateDataKey` permission in the policy.

AWS Management Console

Use the following procedure to generate an account status report.

To generate an account status report

1. Sign in to the AWS Organizations console. You must sign in as an IAM user, assume an IAM role, or sign in as the root user (not recommended) in the organization's management account.
2. On the **Policies** page, choose **Declarative policies for EC2**.
3. On the **Declarative policies for EC2** page, choose **View account status report** from the **Actions** dropdown menu.
4. On the **View account status report** page, choose **Generate status report**.
5. In the **Organizational structure** widget, specify which organizational units (OUs) you want to include in the report.
6. Choose **Submit**.

AWS CLI & AWS SDKs

To generate an account status report

Use the following operations to generate a compliance status report, check on its status, and view the report:

- `ec2:start-declarative-policies-report`: Generates an account status report. The report is generated asynchronously, and can take several hours to complete. For more information, see StartDeclarativePoliciesReport in the *Amazon EC2 API Reference*.
- `ec2:describe-declarative-policies-report`: Describes the metadata of an account status report, including the state of the report. For more information, see DescribeDeclarativePoliciesReports in the *Amazon EC2 API Reference*.
- `ec2:get-declarative-policies-report-summary`: Retrieves a summary of the account status report. For more information, see GetDeclarativePoliciesReportSummary in the *Amazon EC2 API Reference*.
- `ec2:cancel-declarative-policies-report`: Cancels the generation of an account status report. For more information, see CancelDeclarativePoliciesReport in the *Amazon EC2 API Reference*.

Before generating a report, grant the EC2 declarative policies principal access to the Amazon S3 bucket where the report will be stored. To do this, attach the following policy to the

bucket. Replace `amzn-s3-demo-bucket` with your actual Amazon S3 bucket name, and `identity_ARN` with the IAM identity used to call the `StartDeclarativePoliciesReport` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "DeclarativePoliciesReportDelivery",
            "Effect": "Allow",
            "Principal": {
                "AWS": "identity_ARN"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaLast": "organizations.amazonaws.com"
                }
            }
        }
    ]
}
```

Declarative policy syntax and examples

This page describes declarative policy syntax and provides examples.

Considerations

- When you configure a service attribute using a declarative policy, it might impact multiple APIs. Any noncompliant actions will fail.
- Account administrators will not be able to modify the value of the service attribute at the individual account level.

Syntax for declarative policies

A declarative policy is a plaintext file that is structured according to the rules of JSON. The syntax for declarative policies follows the syntax for all management policy types. For a complete discussion of that syntax, see Policy syntax and inheritance for management policy types. This topic focuses on applying that general syntax to the specific requirements of the declarative policy type.

The following example shows basic declarative policy syntax:

```
{
  "ec2_attributes": {
    "exception_message": {
      "@@assign": "Your custom error message.https://myURL"
    }
  }
}
```

- The `ec2_attributes` field key name. Declarative policies always start with a fixed key name for the given AWS service. It's the top line in the example policy above. Currently declarative policies only supported Amazon EC2 related services.
- Under `ec2_attributes`, you can use `exception_message` to set a custom error message. For more information, see Custom error messages for declarative policies.
- Under `ec2_attributes`, you can insert one or more of the supported declarative policies. For those schemas, see Supported declarative policies.

Supported declarative policies

The following are the AWS services and attributes that declarative policies support. In some of the following examples, the JSON whitespace formatting might be compressed to save space.

- VPC Block Public Access
- Serial Console Access
- Image Block Public Access
- Allowed Images Settings
- Instance Metadata Defaults
- Snapshot Block Public Access

VPC Block Public Access

Policy effect

Controls if resources in Amazon VPCs and subnets can reach the internet through internet gateways (IGWs). For more information, see Configuration for internet access in the *Amazon Virtual Private Cloud User Guide*.

Policy contents

```
{
  "ec2_attributes": {
    "vpc_block_public_access": {
      "internet_gateway_block": {
        "mode": {
          "@@assign": "block_ingress"
        },
        "exclusions_allowed": {
          "@@assign": "enabled"
        }
      }
    }
  }
}
```

The following are the available fields for this attribute:

- `"internet_gateway"`:
  - `"mode"`:
    - `"off"`: VPC BPA is not enabled.
    - `"block_ingress"`: All internet traffic to the VPCs (except for VPCs or subnets which are excluded) is blocked. Only traffic to and from NAT gateways and egress-only internet gateways is allowed because these gateways only allow outbound connections to be established.
    - `"block_bidirectional"`: All traffic to and from internet gateways and egress-only internet gateways (except for excluded VPCs and subnets) is blocked.
- `"exclusions_allowed"`: An exclusion is a mode that can be applied to a single VPC or subnet that exempts it from the account's VPC BPA mode and will allow bidirectional or egress-only access.

- `"enabled"`: Exclusions can be created by the account.
- `"disabled"`: Exclusions cannot be created by the account.

> **Note**
>
> You can use the attribute to configure if exclusions are allowed, but you cannot create exclusions with this attribute itself. To create exclusions, you must create them in the account that owns the VPC. For more information about creating VPC BPA exclusions, see Create and delete exclusions in the *Amazon VPC User Guide*.

Considerations

If you use this attribute in a declarative policy, you cannot use the following operations to modify the enforced configuration for the accounts in scope. This list is not exhaustive:

- `ModifyVpcBlockPublicAccessOptions`
- `CreateVpcBlockPublicAccessExclusion`
- `ModifyVpcBlockPublicAccessExclusion`

Serial Console Access

Policy effect

Controls if the EC2 serial console is accessible. For more information about the EC2 serial console, see EC2 Serial Console in the *Amazon Elastic Compute Cloud User Guide*.

Policy contents

```
{
  "ec2_attributes": {
    "serial_console_access": {
      "status": {
        "@@assign": "enabled"
      }
    }
  }
}
```

The following are the available fields for this attribute:

- `"status"`:
  - `"enabled"`: EC2 serial console access is allowed.
  - `"disabled"`: EC2 serial console access is blocked.

Considerations

If you use this attribute in a declarative policy, you cannot use the following operations to modify the enforced configuration for the accounts in scope. This list is not exhaustive:

- `EnableSerialConsoleAccess`
- `DisableSerialConsoleAccess`

Image Block Public Access

Policy effect

Controls if Amazon Machine Images (AMIs) are publicly sharable. For more information about AMIs, see Amazon Machine Images (AMIs) in the *Amazon Elastic Compute Cloud User Guide*.

Policy contents

```
{
  "ec2_attributes": {
    "image_block_public_access": {
      "state": {
        "@@assign": "block_new_sharing"
      }
    }
  }
}
```

The following are the available fields for this attribute:

- `"state"`:
  - `"unblocked"`: No restrictions on the public sharing of AMIs.
  - `"block_new_sharing"`: Blocks new public sharing of AMIs. AMIs that were already publicly shared remain publicly available.

Considerations

If you use this attribute in a declarative policy, you cannot use the following operations to modify the enforced configuration for the accounts in scope. This list is not exhaustive:

- `EnableImageBlockPublicAccess`
- `DisableImageBlockPublicAccess`

Allowed Images Settings

Policy effect

Controls the discovery and use of Amazon Machine Images (AMI) in Amazon EC2 with Allowed AMIs. For more information about AMIs, see Control the discovery and use of AMIs in Amazon EC2 with Allowed AMIs in the *Amazon Elastic Compute Cloud User Guide*.

Policy contents

The following are the available fields for this attribute:

```
{
  "ec2_attributes": {
    "allowed_images_settings": {
      "state": {
        "@@assign": "enabled"
      },
      "image_criteria": {
        "criteria_1": {
          "allowed_image_providers": {
            "@@append": [
              "amazon"
            ]
          }
        }
      }
    }
  }
}
```

- `"state"`:
  - `"enabled"`: The attribute is active and enforced.

- "disabled": The attribute is inactive and not enforced.
- "audit_mode": The attribute is in audit mode. This means it will identify noncompliant images but not block their use.

- "image_criteria": A list of criteria. Support up to 10 criteria with the name from criteria_1 to criteria_10
  - "allowed_image_providers": A comma-separated list of 12 digit account IDs or owner alias of amazon, aws_marketplace, aws_backup_vault.
  - "image_names": The names of the allowed images. Names can include wildcards (? and *). Length: 1–128 characters. With ?, the minimum is 3 characters.
  - "marketplace_product_codes": The AWS Marketplace product codes for allowed images. Length: 1-25 characters Valid characters: Letters (A–Z, a–z) and numbers (0–9)
  - "creation_date_condition": The maximum age for allowed images.
    - "maximum_days_since_created": The maximum number of days that have elapsed since the image was created. Valid Range: Minimum value of 0. Maximum value of 2147483647.
  - "deprecation_time_condition": The maximum period since deprecation for allowed images.
    - "maximum_days_since_deprecated": The maximum number of days that have elapsed since the image was deprecated. Valid Range: Minimum value of 0. Maximum value of 2147483647.

Considerations

If you use this attribute in a declarative policy, you cannot use the following operations to modify the enforced configuration for the accounts in scope. This list is not exhaustive:

- EnableAllowedImagesSettings
- ReplaceImageCriteriaInAllowedImagesSettings
- DisableAllowedImagesSettings

Instance Metadata Defaults

Policy effect

Controls IMDS defaults for all new EC2 instance launches. Note that this configuration sets defaults only and does not enforce IMDS version settings. For more information about IMDS defaults, see IMDS in the *Amazon Elastic Compute Cloud User Guide*.

Policy contents

The following are the available fields for this attribute:

```
{
  "ec2_attributes": {
    "instance_metadata_defaults": {
      "http_tokens": {
        "@@assign": "required"
      },
      "http_put_response_hop_limit": {
        "@@assign": "4"
      },
      "http_endpoint": {
        "@@assign": "enabled"
      },
      "instance_metadata_tags": {
        "@@assign": "enabled"
      }
    }
  }
}
```

- `"http_tokens"`:
  - `"no_preference"`: Other defaults apply. For example, AMI defaults if applicable.
  - `"required"`: IMDSv2 must be used. IMDSv1 is not allowed.
  - `"optional"`: Both IMDSv1 and IMDSv2 are allowed.

  > **Note**
  >
  > **Metadata version**
  > Before setting `http_tokens` to `required` (IMDSv2 must be used), make sure that none of your instances are making IMDSv1 calls.

- `"http_put_response_hop_limit"`:

- "*Integer*": Integer value from -1 to 64, representing the maximum number of hops the metadata token can travel. To indicate no preference, specify -1.

  > **Note**
  >
  > **Hop limit**
  > If `http_tokens` is set to `required`, it is recommended to set `http_put_response_hop_limit` to a minimum of 2. For more information, see Instance metadata access considerations in the *Amazon Elastic Compute Cloud User Guide*.

- `"http_endpoint"`:
  - `"no_preference"`: Other defaults apply. For example, AMI defaults if applicable.
  - `"enabled"`: The instance metadata service endpoint is accessible.
  - `"disabled"`: The instance metadata service endpoint is not accessible.
- `"instance_metadata_tags"`:
  - `"no_preference"`: Other defaults apply. For example, AMI defaults if applicable.
  - `"enabled"`: Instance tags can be accessed from instance metadata.
  - `"disabled"`: Instance tags cannot be accessed from instance metadata.

Snapshot Block Public Access

Policy effect

Controls if Amazon EBS snapshots are publicly accessible. For more information about EBS snapshots, see Amazon EBS snapshots in the *Amazon Elastic Block Store User Guide*.

Policy contents

```
{
  "ec2_attributes": {
    "snapshot_block_public_access": {
      "state": {
        "@@assign": "block_new_sharing"
      }
    }
  }
```

```
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
resource_control_policy/p-i9j8k7l6m5",
            "Name": "DenyIAMRCP",
            "Description": "Deny all IAM actions",
            "Type": "RESOURCE_CONTROL_POLICY",
            "AwsManaged": false
        },
         "Content": "{\"Version\":\"2012-10-17\",       \"Statement\":[{\"Sid\":
\"Statement1\",\"Effect\":\"Deny\",\"Action\":[\"iam:*\"],\"Resource\":[\"*\"]}]}"
    }
}
```

- AWS SDKs: CreatePolicy

> **Note**
>
> RCPs don't take effect on the management account and in a few other situations. For more information, see Resources and entities not restricted by RCPs.

Create a declarative policy

> **Minimum permissions**
>
> To create a declarative policy, you need permission to run the following action:
>
> - `organizations:CreatePolicy`

AWS Management Console

To create a declarative policy

1. Sign in to the AWS Organizations console. You must sign in as an IAM user, assume an IAM role, or sign in as the root user (not recommended) in the organization's management account.
2. On the **Declarative policies** page, choose **Create policy**.

3. On the **Create new declarative policy for EC2** page, enter a **Policy name** and an optional **Policy description**.
4. (Optional) You can add one or more tags to the policy by choosing **Add tag** and then entering a key and an optional value. Leaving the value blank sets it to an empty string; it isn't `null`. You can attach up to 50 tags to a policy. For more information, see Tagging AWS Organizations resources.
5. You can build the policy using the **Visual editor** as described in this procedure. You can also enter or paste policy text in the **JSON** tab. For information about declarative policy syntax, see Declarative policy syntax and examples.

   If you choose to use the **Visual editor**, select the service attribute you want to include in your declarative policy. For more information, see Supported AWS services and attributes.
6. Choose **Add service attribute**, and configure the attribute to your specifications. For more detailed information on the each effect, see Declarative policy syntax and examples.
7. When you're finished editing your policy, choose **Create policy** at the lower-right corner of the page.

AWS CLI & AWS SDKs

To create a declarative policy

You can use one of the following to create a declarative policy:

- AWS CLI: create-policy

  1. Create a declarative policy like the following, and store it in a text file.

```
{
    "ec2_attributes": {
        "image_block_public_access": {
            "state": {
                "@@assign": "block_new_sharing"
            }
        }
    }
}
```

The following example adds or changes the description for a declarative policy.

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --description "My new description"
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
declarative_policy_ec2/p-i9j8k7l6m5",
            "Name": "Renamed policy",
            "Description": "My new description",
            "Type": "DECLARATIVE_POLICY_EC2",
            "AwsManaged": false
        },
        "Content": "{"ec2_attributes":{"image_block_public_access":{"state":
{"@@assign":"block_new_sharing"}}}}".
    }
}
```

- AWS SDKs: UpdatePolicy

Update a backup policy

When you sign in to your organization's management account, you can edit a policy that requires changes in your organization.

Minimum permissions

To update a backup policy, you must have permission to run the following actions:

- `organizations:UpdatePolicy` with a `Resource` element in the same policy statement that includes the ARN of the policy to update (or "*")
- `organizations:DescribePolicy` with a `Resource` element in the same policy statement that includes the ARN of the policy to update (or "*")

AWS Management Console

To update a backup policy

1. Sign in to the AWS Organizations console. You must sign in as an IAM user, assume an IAM role, or sign in as the root user (not recommended) in the organization's management account.
2. On the **Backup policies** page, choose the name of the policy that you want to update.
3. Choose **Edit policy**.
4. You can enter a new **Policy name**, **Policy description**. You can change the policy content by using either the **Visual editor** or by directly editing the **JSON**.
5. When you're finished updating the policy, choose **Save changes**.

AWS CLI & AWS SDKs

To update a backup policy

You can use one of the following to update a backup policy:

- AWS CLI: update-policy

  The following example renames a backup policy.

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --name "Renamed policy"
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
backup_policy/p-i9j8k7l6m5",
            "Name": "Renamed policy",
            "Type": "BACKUP_POLICY",
            "AwsManaged": false
        },
         "Content": "{\"plans\":{\"TestBackupPlan\":{\"regions\":{\"@@assign\":
  ....TRUNCATED FOR BREVITY....   "@@assign\":[\"Yes\"]}}}}}}"
    }
}
```

The following example adds or changes the description for a backup policy.

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --description "My new description"
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
backup_policy/p-i9j8k7l6m5",
            "Name": "Renamed policy",
            "Description": "My new description",
            "Type": "BACKUP_POLICY",
            "AwsManaged": false
        },
       "Content": "{\"plans\":{\"TestBackupPlan\":{\"regions\":{\"@@assign\":
  ....TRUNCATED FOR BREVITY....   "@@assign\":[\"Yes\"]}}}}}}"
    }
}
```

The following example changes the JSON policy document attached to a backup policy. In this example, the content is taken from a file called `policy.json` with the following text:

```
{
    "plans": {
        "PII_Backup_Plan": {
            "regions": { "@@assign": [ "ap-northeast-2", "us-east-1", "eu-
north-1" ] },
            "rules": {
                "Hourly": {
                    "schedule_expression": { "@@assign": "cron(0 5/1 ? * * *)" },
                    "start_backup_window_minutes": { "@@assign": "480" },
                    "complete_backup_window_minutes": { "@@assign": "10080" },
                    "lifecycle": {
                        "move_to_cold_storage_after_days": { "@@assign": "180" },
                        "delete_after_days": { "@@assign": "270" },
                        "opt_in_to_archive_for_supported_resources": {"@@assign":
 false}
                    },
                    "target_backup_vault_name": { "@@assign": "FortKnox" },
                    "copy_actions": {
```

```
                        "arn:aws:backup:us-east-1:$account:backup-vault:secondary-
vault": {
                            "lifecycle": {
                                "move_to_cold_storage_after_days": { "@@assign":
 "10" },
                                "delete_after_days": { "@@assign": "100" },
                                "opt_in_to_archive_for_supported_resources":
 {"@@assign": false}
                            }
                        }
                    }
                }
            },
            "selections": {
                "tags": {
                    "datatype": {
                        "iam_role_arn": { "@@assign": "arn:aws:iam::$account:role/
MyIamRole" },
                        "tag_key": { "@@assign": "dataType" },
                        "tag_value": { "@@assign": [ "PII" ] }
                    }
                }
            }
        }
    }
}
```

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --content file://policy.json
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
backup_policy/p-i9j8k7l6m5",
            "Name": "Renamed policy",
            "Description": "My new description",
            "Type": "BACKUP_POLICY",
            "AwsManaged": false
        },
         "Content": "{\"plans\":{\"TestBackupPlan\":{\"regions\":{\"@@assign\":
  ....TRUNCATED FOR BREVITY....   "@@assign\":[\"Yes\"]}}}}}}"
```

AWS Management Console

To edit the tags attached to an RCP

1. Sign in to the AWS Organizations console. You must sign in as an IAM user, assume an IAM role, or sign in as the root user (not recommended) in the organization's management account.
2. On the **Resource control policy** page, choose the name of the policy with the tags that you want to edit.
3. On the policy details page, choose the **Tags** tab, and then choose **Manage tags**.
4. Make any or all of the following changes:
   - Change the value of a tag by entering a new value over the old one. You can't directly modify the tag key. To change a key, you must delete the tag with the old key and then add a tag with the new key.
   - Remove an existing tag by choosing **Remove**.
   - Add a new tag key and value pair. Choose **Add tag**, then enter the new key name and optional value in the provided boxes. If you leave the **Value** box empty, the value is an empty string; it isn't `null`.
5. When you're finished, choose **Save changes**.

AWS CLI & AWS SDKs

To edit the tags attached to an RCP

You can use one of the following commands to edit the tags attached to an RCP:

- AWS CLI: tag-resource and untag-resource
- AWS SDKs: TagResource and UntagResource

Edit tags attached to an declarative policy

When you sign in to your organization's management account, you can add or remove the tags attached to a declarative policy. For more information about tagging, see Tagging AWS Organizations resources.

Minimum permissions

To edit the tags attached to a declarative policy in your AWS organization, you must have the following permissions:

- `organizations:DescribeOrganization`– required only when using the Organizations console
- `organizations:DescribePolicy`– required only when using the Organizations console
- `organizations:TagResource`
- `organizations:UntagResource`

AWS Management Console

To edit the tags attached to a declarative policy

1. Sign in to the AWS Organizations console. You must sign in as an IAM user, assume an IAM role, or sign in as the root user (not recommended) in the organization's management account.
2. On the **Declarative policies** page, choose the name of the policy with the tags that you want to edit.
3. On the chosen policy's detail page, choose the **Tags** tab, and then choose **Manage tags**.
4. You can perform any of these actions on this page:
   - Edit the value for any tag by entering a new value over the old one. You can't modify the key. To change a key, you must delete the tag with the old key and add a tag with the new key.
   - Remove an existing tag by choosing **Remove**.
   - Add a new tag key and value pair. Choose **Add tag**, then enter the new key name and optional value in the provided boxes. If you leave the **Value** box empty, the value is an empty string; it isn't `null`.
5. Choose **Save changes** after you've made all the additions, removals, and edits you want to make.