AWS Control Tower

User Guide

AWS Control Tower: User Guide

Table of Contents

What Is AWS Control Tower?

AWS Control Tower provides the easiest way to set up and govern a secure, compliant, multi-account AWS environment based on best practices established by working with thousands of enterprises. With AWS Control Tower, end users on your distributed teams can provision new AWS accounts quickly. Meanwhile your central cloud administrators will know that all accounts are aligned with centrally established, company-wide compliance policies.

Features

AWS Control Tower has the following features:

- **Landing zone** – A landing zone is a well-architected, multi-account AWS environment that's based on security and compliance best practices. This is the enterprise-wide container that holds all of your organizational units (OUs), accounts, users, and other resources that you want to be subject to compliance regulation. A landing zone can scale to fit the needs of an enterprise of any size.
- **Guardrails** – A guardrail is a high-level rule that provides ongoing governance for your overall AWS environment. It's expressed in plain language. Two kinds of guardrails exist: *preventive* and *detective*. Three categories of guidance apply to the two kinds of guardrails: *mandatory*, *strongly recommended*, or *elective*. For more information about guardrails, see How Guardrails Work (p. 7).
- **Account Factory** – An Account Factory is a configurable account template that helps to standardize the provisioning of new accounts with pre-approved account configurations. AWS Control Tower offers a built-in Account Factory that helps automate the account provisioning workflow in your organization. For more information, see Account Factory (p. 52).
- **Dashboard** – The dashboard offers continuous oversight of your landing zone to your team of central cloud administrators. Use the dashboard to see provisioned accounts across your enterprise, guardrails enabled for policy enforcement, guardrails enabled for continuous detection of policy non-conformance, and noncompliant resources organized by accounts and OUs.

Related Services

AWS Control Tower is built on top of trusted and reliable AWS services including AWS Service Catalog, AWS Single Sign-On, and AWS Organizations. For more information, see Integrated Services (p. 45).

Pricing

No additional charge exists for using AWS Control Tower. You only pay for the AWS services enabled by AWS Control Tower, and the services you use in your landing zone. For example, you pay for AWS Service Catalog for provisioning accounts with Account Factory, and AWS CloudTrail for events tracked in your landing zone. For information about the pricing and fees associated with AWS Control Tower, see AWS Control Tower pricing.

Are You a First-Time User of AWS Control Tower?

If you’re a first-time user of this service, we recommend that you read the following:

1. If you’re ready to create your first landing zone, see Getting Started with AWS Control Tower (p. 14).

2. For information on drift detection and prevention, see Detecting and Resolving Drift in AWS Control Tower (p. 58).
3. For security details, see Security in AWS Control Tower (p. 63).
4. For information on updating your landing zone and member accounts, see Configuration Update Management in AWS Control Tower (p. 75).

How AWS Control Tower Works

The following describes at a high level how AWS Control Tower works. Your landing zone is a well-architected multi-account environment for all of your AWS resources. You can use this environment to enforce compliance regulations on all of your AWS accounts.

What Are the Shared Accounts?

In AWS Control Tower, three shared accounts in your landing zone are not provisioned in Account Factory: the master account, the log archive account, and the audit account.

When you create your landing zone a number of AWS resources are created. To use AWS Control Tower, you must not modify or delete these AWS Control Tower managed resources outside of the supported methods described in this guide. Deleting or modifying these resources will cause your landing zone to enter an unknown state.

> **Important**
> When you enable guardrails with strongly recommended guidance, AWS Control Tower managed AWS resources are created in your accounts. Do not modify or delete resources created by AWS Control Tower. Doing so could result in the guardrails entering an unknown state. For more information, see Guardrail Reference (p. 22).

What Is the Master Account?

This is the account that you created specifically for your landing zone. This account is used for billing for everything in your landing zone. It's also used for Account Factory provisioning and accounts, as well as to manage OUs and guardrails.

When you set up your landing zone, the following AWS resources are created within your master account.

AWS service	Resource type	Resource name
AWS Organizations	Accounts	audit log archive
AWS Organizations	OUs	Core Custom
AWS Organizations	Service Control Policies	aws-guardrails-*
AWS CloudFormation	Stacks	AWSControlTowerBP-BASELINE-CLOUDTRAIL-MASTER
AWS CloudFormation	StackSets	AWSControlTowerBP-BASELINE-CLOUDTRAIL AWSControlTowerBP-BASELINE-CLOUDWATCH

AWS service	Resource type	Resource name
		AWSControlTowerBP-BASELINE-CONFIG AWSControlTowerBP-BASELINE-ROLES AWSControlTowerBP-BASELINE-SERVICE-ROLES AWSControlTowerBP-SECURITY-TOPICS AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-READ-PROHIBITED AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-WRITE-PROHIBITED AWSControlTowerLoggingResources AWSControlTowerSecurityResources
AWS Service Catalog	Product	AWS Control Tower Account Factory
AWS CloudTrail	Trail	aws-controltower-BaselineCloudTrail
Amazon CloudWatch	CloudWatch Logs	aws-controltower/CloudTrailLogs
AWS Identity and Access Management	Roles	AWSControlTowerAdmin AWSControlTowerStackSetRole AWSControlTowerCloudTrailRolePolicy
AWS Identity and Access Management	Policies	AWSControlTowerServiceRolePolicy AWSControlTowerAdminPolicy AWSControlTowerCloudTrailRolePolicy AWSControlTowerStackSetRolePolicy

AWS service	Resource type	Resource name
AWS Single Sign-On	Directory groups	AWSAccountFactory AWSAuditAccountAdmins AWSControlTowerAdmins AWSLogArchiveAdmins AWSLogArchiveViewers AWSSecurityAuditors AWSSecurityAuditPowerUsers AWSServiceCatalogAdmins
AWS Single Sign-On	Permission Sets	AWSAdministratorAccess AWSPowerUserAccess AWSServiceCatalogAdminFullAccess AWSServiceCatalogEndUserAccess AWSReadOnlyAccess AWSOrganizationsFullAccess

What Is the Log Archive Account?

This account works as a repository for logs of API activities and resource configurations from all accounts in the landing zone.

When you set up your landing zone, the following AWS resources are created within your log archive account.

AWS service	Resource type	Resource Name
AWS CloudFormation	Stacks	StackSet-AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-READ-PROHIBITED- StackSet-AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-WRITE-PROHIBITED StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH- StackSet-AWSControlTowerBP-BASELINE-CONFIG- StackSet-AWSControlTowerBP-BASELINE-CLOUDTRAIL-

AWS service	Resource type	Resource Name
		StackSet-AWSControlTowerBP-BASELINE-SERVICE-ROLES- StackSet-AWSControlTowerBP-BASELINE-ROLES- StackSet-AWSControlTowerLoggingResources-
AWS Config	AWS Config Rules	AWSControlTower_AWS-GR_AUDIT_BUCKET_PUBLIC_READ_PROHIBITE AWSControlTower_AWS-GR_AUDIT_BUCKET_PUBLIC_WRITE_PROHIBIT
AWS CloudTrail	Trails	aws-controltower-BaselineCloudTrail
Amazon CloudWatch	CloudWatch Event Rules	aws-controltower-ConfigComplianceChangeEventRule
Amazon CloudWatch	CloudWatch Logs	aws-controltower/CloudTrailLogs /aws/lambda/aws-controltower-NotificationForwarder
AWS Identity and Access Management	Roles	aws-controltower-AdministratorExecutionRole aws-controltower-CloudWatchLogsRole aws-controltower-ConfigRecorderRole aws-controltower-ForwardSnsNotificationRole aws-controltower-ReadOnlyExecutionRole AWSControlTowerExecution
AWS Identity and Access Management	Policies	AWSControlTowerServiceRolePolicy
Amazon Simple Notification Service	Topics	aws-controltower-SecurityNotifications
AWS Lambda	Applications	StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-*
AWS Lambda	Functions	aws-controltower-NotificationForwarder

AWS service	Resource type	Resource Name
Amazon Simple Storage Service	Buckets	aws-controltower-logs-* aws-controltower-s3-access-logs-*

What Is the Audit Account?

The audit account is a restricted account that's designed to give your security and compliance teams read and write access to all accounts in your landing zone. From the audit account, you have programmatic access to review accounts, by means of a role that is granted to Lambda functions only. The audit account does not allow you to log in to other accounts manually. For more information about Lambda functions and roles, see Configure a Lambda function to assume a role from another AWS account.

When you set up your landing zone, the following AWS resources are created within your audit account.

AWS service	Resource type	Resource name
AWS CloudFormation	Stacks	StackSet-AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-READ-PROHIBITED- StackSet-AWSControlTowerGuardrailAWS-GR-AUDIT-BUCKET-PUBLIC-WRITE-PROHIBITED- StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH- StackSet-AWSControlTowerBP-BASELINE-CONFIG- StackSet-AWSControlTowerBP-BASELINE-CLOUDTRAIL- StackSet-AWSControlTowerBP-BASELINE-SERVICE-ROLES- StackSet-AWSControlTowerBP-SECURITY-TOPICS- StackSet-AWSControlTowerBP-BASELINE-ROLES- StackSet-AWSControlTowerSecurityResources-*
AWS Config	Aggregator	aws-controltower-GuardrailsComplianceAggregator
AWS Config	AWS Config Rules	AWSControlTower_AWS-GR_AUDIT_BUCKET_PUBLIC_READ_PROHIBITE

AWS service	Resource type	Resource name
		AWSControlTower_AWS-GR_AUDIT_BUCKET_PUBLIC_WRITE_PROHIBITE
AWS CloudTrail	Trail	aws-controltower-BaselineCloudTrail
Amazon CloudWatch	CloudWatch Event Rules	aws-controltower-ConfigComplianceChangeEventRule
Amazon CloudWatch	CloudWatch Logs	aws-controltower/CloudTrailLogs /aws/lambda/aws-controltower-NotificationForwarder
AWS Identity and Access Management	Roles	aws-controltower-AdministratorExecutionRole aws-controltower-CloudWatchLogsRole aws-controltower-ConfigRecorderRole aws-controltower-ForwardSnsNotificationRole aws-controltower-ReadOnlyExecutionRole aws-controltower-SecurityAdministratorRole aws-controltower-SecurityReadOnlyRole AWSControlTowerExecution
AWS Identity and Access Management	Policies	AWSControlTowerServiceRolePolicy
Amazon Simple Notification Service	Topics	aws-controltower-AggregateSecurityNotifications aws-controltower-AllConfigNotifications aws-controltower-SecurityNotifications
AWS Lambda	Functions	aws-controltower-NotificationForwarder

How Guardrails Work

A guardrail is a high-level rule that provides ongoing governance for your overall AWS environment. Each guardrail enforces a single rule, and it's expressed in plain language. Compliance needs evolve, and you can change the elective or strongly recommended guardrails that are in force, at any time, from the AWS Control Tower console. Mandatory guardrails are always applied, and they can't be changed.

Preventive guardrails prevent actions from occurring. For example, the **Disallow policy changes to log archive** guardrail prevents any IAM policy changes within the log archive shared account. Any attempt to perform a prevented action is denied and logged in CloudTrail. The resource is also logged in AWS Config.

Detective guardrails detect specific events when they occur and log the action in CloudTrail. For example, the **Enable encryption for EBS volumes attached to EC2 instances** detects if an unencrypted Amazon EBS volume is attached to an EC2 instance in your landing zone.

Related Topics

- Guardrails in AWS Control Tower (p. 20)
- Detecting and Resolving Drift in AWS Control Tower (p. 58)

How AWS Regions Work With AWS Control Tower

Currently, AWS Control Tower is supported in the following AWS Regions:

- US East (N. Virginia)
- US East (Ohio)
- US West (Oregon)
- Europe (Ireland)

When you create a landing zone, the region that you're using for access to the AWS Management Console becomes your home AWS Region for AWS Control Tower. During the creation process, some resources are provisioned in the home AWS Region. Other resources, such as OUs and AWS accounts, are global.

Currently, all preventive guardrails work globally. Detective guardrails, however, only work in regions where AWS Control Tower is supported.

How AWS Control Tower Works With Roles to Create and Manage Accounts

AWS Control Tower creates a customer's account by calling the `CreateAccount` API of AWS Organizations. When AWS Organizations creates this account, it creates a role within that account, which AWS Control Tower names by passing in a parameter to the API. The name is `AWSControlTowerExecution`.

AWS Control Tower takes over the `AWSControlTowerExecution` role for all accounts created by Account Factory. Using this role, AWS Control Tower baselines the account and applies mandatory (and any other enabled) guardrails, which results in creation of other roles. These roles in turn are used by other services, such as AWS Config.

> **Note**
> To baseline an account is to set up its blueprints and guardrails. The baselining process also sets up the centralized logging and security audit roles on the account, as part of deploying the blueprints.

AWS Control Tower and VPCs

This section is intended primarily for network administrators. Your company's network administrator usually is the person who selects the overall CIDR range for your AWS Control Tower organization. The network administrator then allocates subnets from within that range for specific purposes.

Here are some essential facts about AWS Control Tower and VPCs:

- Each AWS Control Tower account is allowed one VPC.
- Every VPC has three Availability Zones. By default, each Availability Zone contains one public subnet and two private subnets. Therefore, each VPC contains nine subnets by default.
- Each of the nine subnets in your VPC is assigned a unique range, of equal size.
- Because the IP addresses do not overlap, the nine subnets within your VPC can communicate with each other in an unrestricted manner.

The best practice for controlling communication among your VPC subnets, if needed, is to set up access control lists with rules that define the permitted traffic flow. Use security groups for control of traffic among specific instances.

The number of subnets in a VPC is configurable. For more information about how to change your VPC subnet configuration, see the Account Factory topic.

CIDR and Peering for VPC and AWS Control Tower

When you choose a CIDR range for your VPC, AWS Control Tower applies *Carrier Grade NAT (CGN)* and Account Factory validates the IP address ranges according to the RFC 1918 specification. Account Factory allows the IP ranges of `10.0.0.0/16`, `172.16.0.0/16`, and `192.168.0.0/16`. If the range you specify is outside of that, AWS Control Tower provides an error message.

When AWS Control Tower creates a VPC using the CIDR range you select, it assigns the identical CIDR range to your master account and to *every VPC* for every account you create within the organizational unit (OU). This implementation does not permit peering from any of your VPCs to any other VPC within your AWS Control Tower OU.

Within each VPC, AWS Control Tower divides your specified CIDR range evenly into nine subnets. None of the subnets within a VPC overlap. Therefore, they all can communicate with each other.

In summary, by default, subnet communication within the VPC is unrestricted, and VPC-to-VPC peering is not possible.

The default CIDR range is `172.31.0.0/16`.

Options for VPC Peering in AWS Control Tower

Instead of peering, AWS Control Tower offers VPC endpoint services through AWS PrivateLink as the recommended solution for VPC peering among AWS Control Tower VPCs. Packets can be sent directly from a specific IP address in one VPC to another specific IP address within another VPC.

However, another option is available. Within AWS Control Tower, if you change the CIDR range in the settings of Account Factory, all new accounts that are subsequently created by AWS Control Tower (using Account Factory) are assigned the new CIDR range. The old accounts are not updated. For example, you can create an account, then change the CIDR range and create a new account, and the VPCs allocated to those two accounts can be peered. Peering is possible because their IP address ranges are not identical. For information about how to change account settings, see the Account Factory documentation on updating an account.

Notes on VPC and CIDR

Some network administrators may realize that it is possible to peer two subnets in two different VPCs (that is, in two different accounts) without changing the CIDR settings for accounts. Because the nine subnets in a VPC do not overlap, peering technically is possible from *[VPC1, subnet 1]* to *[VPC2, subnet 2]*, for example. However, this approach depends on an implementation detail of how subnet ranges are allocated within a VPC. We don't recommend this method of peering, because it could fail at any time.

When working with VPCs, AWS Control Tower makes no distinction at the Region level. Every subnet is allocated from the exact CIDR range that you specify. The VPC subnets can exist in any Region.

Setting Up

Before you use AWS Control Tower for the first time, complete the following tasks:

1. Sign up for AWS (p. 11)
2. Create an IAM User (p. 11)

These tasks create an AWS account and an IAM user with administrator privileges for the account. For information on additional setup tasks specifically for AWS Control Tower, see Getting Started with AWS Control Tower (p. 14).

Sign up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including AWS Control Tower. If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Note your AWS account number, because you need it for the next task.

Create an IAM User

Services in AWS, such as AWS Control Tower, require that you provide credentials when you access them, so that the service can determine whether you have permissions to access its resources. AWS recommends that you don't use the AWS account root user credentials of your AWS account to make requests. Instead, create an AWS Identity and Access Management (IAM) user and grant that user full access. We call these users administrators.

You can use the administrator credentials, instead of AWS account root user credentials of your account, to interact with AWS and perform tasks, such as create users and grant them permissions. For more information, see Root Account Credentials vs. IAM User Credentials in the *AWS General Reference* and IAM Best Practices in the *IAM User Guide*.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM Management Console.

To create an administrator user for yourself and add the user to an administrators group (console)

1. Use your AWS account email address and password to sign in as the *AWS account root user* to the IAM console at https://console.aws.amazon.com/iam/.

> **Note**
> We strongly recommend that you adhere to the best practice of using the **`Administrator`** IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

2. In the navigation pane, choose **Users** and then choose **Add user**.
3. For **User name**, enter **`Administrator`**.
4. Select the check box next to **AWS Management Console access**. Then select **Custom password**, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter **`Administrators`**.
10. Choose **Filter policies**, and then select **AWS managed -job function** to filter the table contents.
11. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.

    > **Note**
    > You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM Entities in the *IAM User Guide*.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access Management and Example Policies.

To sign in as this new IAM user, first sign out of the AWS Management Console. Then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is `1234-5678-9012`, your AWS account ID is `123456789012`).

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays ***your_user_name@your_aws_account_id***.

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. To do so, from the IAM dashboard, choose **Create Account Alias** and enter an alias, such as your company name. To sign in after you create an account alias, use the following URL.

```
https://your_account_alias.signin.aws.amazon.com/console/
```

To verify the sign-in link for IAM users for your account, open the IAM console and check under **AWS Account Alias** on the dashboard.

Set up MFA

Because of the nature of AWS Control Tower, we strongly recommend that you enable multi-factor authentication (MFA) for your master account. For more information, see Enable MFA on the AWS Account Root User in the *IAM User Guide*.

Next Step

Getting Started with AWS Control Tower (p. 14)

Getting Started with AWS Control Tower

This is the AWS Control Tower getting started procedure for central cloud administrators. Use this procedure when you're ready to set up your landing zone. From start to finish, it should take about an hour. This procedure has two steps.

Guidance for Using AWS Control Tower

We recommend the following when you use AWS Control Tower. This guidance might change as the service is updated.

General Guidance

- Do not modify or delete resources created by AWS Control Tower in the master account or in the shared accounts. Modification of these resources can require an update to your landing zone.
- Do not modify or delete the AWS Identity and Access Management (IAM) roles created within the shared accounts in the core organizational unit (OU). Modification of these roles can require an update to your landing zone.
- For more information about the resources created by AWS Control Tower, see What Are the Shared Accounts? (p. 2)

AWS Organizations Guidance

- Do not use AWS Organizations to update service control policies (SCPs) that are attached by AWS Control Tower to an AWS Control Tower managed OU. Doing so could result in the guardrails entering an unknown state, which will require you to re-enable affected guardrails.
- If you use Organizations to create, invite, or move accounts within an organization created by AWS Control Tower, those outside accounts will not be managed by AWS Control Tower and will not appear in the **Accounts** table.
- If you use Organizations to create or move OUs within an organization created by AWS Control Tower, those outside OUs will not be managed by AWS Control Tower and will not appear in the **Organizational Units** table.
- If you use Organizations to rename or delete an OU that was created by AWS Control Tower, then this OU will continue to be displayed by AWS Control Tower using its original name. You will not be able to provision a new account to this OU using Account Factory.

AWS Single Sign-On Guidance

- If you reconfigure your directory in AWS Single Sign-On to Active Directory, all preconfigured users and groups in AWS SSO will be deleted.

Account Factory Guidance

- When you use Account Factory to provision new accounts in AWS Service Catalog, do not define `TagOptions`, enable notifications, or create a provisioned product plan. Doing so can result in a failure to provision a new account.

Pre-Launch Checks for Your Master Account

Before AWS Control Tower does any work in your account to set up the landing zone, it runs a series of pre-launch checks. These ensure that your master account is ready for the changes that establish your landing zone. The checks that run before setting up a landing zone are as follows:

- The existing service limits for the AWS account must be sufficient for AWS Control Tower to launch. For more information, see Quotas in AWS Control Tower (p. 77).
- The AWS account cannot be a member of an existing AWS Organizations OU (regardless of whether it's set up with all features enabled or for consolidated billing).
- The AWS account must be subscribed to the following AWS services:
  - Amazon Simple Storage Service (Amazon S3)
  - Amazon SNS
  - Amazon Virtual Private Cloud (Amazon VPC)
  - AWS CloudFormation
  - AWS CloudTrail
  - Amazon CloudWatch
  - AWS Config
  - AWS Identity and Access Management (IAM)
  - AWS Lambda

    Note
    By default, all accounts are subscribed to these services.
- The AWS account must not have an AWS Config aggregator already configured.
- The AWS account must not have AWS Single Sign-On (AWS SSO) already set up.

When you set up a landing zone, AWS Control Tower performs the following actions in your master account on your behalf:

- Creates three Organizations organizational units (OUs): Root, Core, and Custom.
- Creates two shared accounts: the log archive account and audit account.
- Creates a cloud-native directory in AWS SSO, with preconfigured groups and single sign-on access.
- Applies 17 preventive guardrails to enforce policies.
- Applies three detective guardrails to detect configuration violations.

Step One: Create Your Shared Account Email Addresses

This guide assumes that you're setting up your landing zone in a new AWS account. For information on creating your account and your IAM administrator, see Setting Up (p. 11).

To set up your landing zone, AWS Control Tower requires two unique email addresses that aren't already associated with an AWS account. These email addresses should each be a collaborative inbox, a shared email account for the different users in your enterprise that will do specific work related to AWS Control Tower. The email addresses are:

- **Audit account** – This is for your team of users that need access to the audit information made available by AWS Control Tower. You can also use this account as the access point for third-party tools that will perform programmatic auditing of your environment to help you audit for compliance purposes.

- **Log archive account** – This is for your team of users that need access to all the logging information for all of your managed accounts within managed OUs in your landing zone.

Step Two: Set Up Your Landing Zone

AWS Control Tower has no APIs or programmatic access. To set up your landing zone, perform the following procedure.

To set up your landing zone

1. Open a web browser, and navigate to the AWS Control Tower console at https://console.aws.amazon.com/controltower.
2. Choose **Set up your landing zone**.
3. Provide the email addresses for your log archive and audit accounts. Note that the email addresses must not already have associated AWS accounts.
4. Review the **Service permissions**, and when you're ready, choose **I understand the permissions AWS Control Tower will use to administer AWS resources and enforce rules on my behalf**.
5. Choose **Launch your AWS Control Tower**.

This starts the process of setting up your landing zone, which can take about an hour to complete. During setup, your core accounts are created, your root and Core OUs are created, and AWS resources are created, modified, or deleted.

> **Important**
> The email address you provided for the audit account will receive **AWS Notification - Subscription Confirmation** emails from every AWS Region supported by AWS Control Tower. To receive compliance emails in your audit account, you must choose the **Confirm subscription** link within each email from each AWS Region supported by AWS Control Tower.

Next Steps

Now that your landing zone is set up, it's ready for use.

When you set up a landing zone, AWS Control Tower performs the following actions in your master account on your behalf:

- Creates two organizational units (OUs), **Core** and **Custom**.
- Creates two accounts: the log archive account and the security audit account within the Core OU.
- Creates a cloud-native directory in AWS SSO, with preconfigured groups, permission sets, and single sign-on access.
- Applies 25 preventive guardrails, to enforce policies.
- Applies two detective guardrails, to detect configuration violations.
- Creates the Account Factory product in AWS Service Catalog so your end-users can provision new AWS accounts within your landing zone.

To learn more about how you can use AWS Control Tower, see the following topics:

- Your end users can provision their own AWS accounts in your landing zone using Account Factory. For more information, see Configuring and Provisioning Accounts Through AWS Service Catalog (p. 52).

- From time to time, you may need to update your landing zone to get the latest backend updates, the latest guardrails, and to keep your landing zone up-to-date. For more information, see Configuration Update Management in AWS Control Tower (p. 75).
- If you encounter issues while using AWS Control Tower, see Troubleshooting (p. 84).

Best Practices for Account Administrators

This topic is intended primarily for master account administrators.

Master account administrators are responsible for explaining some tasks that AWS Control Tower guardrails prevent their member account administrators from doing. This topic describes some best practices and procedures for transferring this knowledge, and it gives other tips for setting up your AWS Control Tower environment efficiently.

Explaining Access to Users

The AWS Control Tower console is available only to users with the master account administrator permissions. Only these users can perform administrative work within your landing zone. In accordance with best practices, this means that the majority of your users and member account administrators will never see the AWS Control Tower console. As a member of the master account administrator group, it's your responsibility to explain the following information to the users and administrators of your member accounts, as appropriate.

- Explain which AWS resources that users and administrators have access to within the landing zone.
- List the preventive guardrails that apply to each Organizational Unit (OU) so that the other administrators can plan and execute their AWS workloads accordingly.

Explaining Resource Access

Some administrators and other users may need an explanation of the AWS resources to which they have access to within your landing zone. This access can include programmatic access and console-based access. Generally speaking, read access and write access for AWS resources is allowed. To perform work within AWS, your users require some level of access to the specific services they need to do their jobs.

Some users, such as your AWS developers, may need to know about the resources to which they have access, so they can create engineering solutions. Other users, such as the end users of the applications that run on AWS services, do not need to know about AWS resources within your landing zone.

AWS offers tools to identify the scope of a user's AWS resource access. After you identify the scope of a user's access, you can share that information with the user, in accordance with your organization's information management policies. For more information about these tools, see the links that follow.

- **AWS access advisor** – The AWS Identity and Access Management (IAM) access advisor tool lets you determine the permissions that your developers have by analyzing the last timestamp when an IAM entity, such as a user, role, or group, called an AWS service. You can audit service access and remove unnecessary permissions, and you can automate the process if needed. For more information, see our AWS Security blog post.
- **IAM policy simulator** – With the IAM policy simulator, you can test and troubleshoot IAM-based and resource-based policies. For more information, see Testing IAM Policies with the IAM Policy Simulator.
- **AWS CloudTrail logs** – You can review AWS CloudTrail logs to see actions taken by a user, role, or AWS service. For more information about CloudTrail, see the AWS CloudTrail User Guide.

Actions taken by CloudTrail landing zone administrators are logged in the landing zone master account. Actions taken by member account administrators and users are logged in the shared log archive account.

Explaining Preventive Guardrails

A preventive guardrail ensures that your organization's accounts maintain compliance with your corporate policies. The status of a preventive guardrail is either **enforced** or **not-enabled**. A preventive guardrail prevents policy violations by using service control policies and AWS Lambda functions. In comparison, a detective guardrail only informs you of various events or states that exist.

Some of your users, such as AWS developers, may need to know about the preventive guardrails that apply to any accounts and OUs they use, so they can create engineering solutions. The following procedure offers some guidance on how to provide this information for the right users, according to your organization's information management policies.

> **Note**
> This procedure assumes you've already created at least one child OU within your landing zone, as well as at least one AWS Single Sign-On user.

To show preventive guardrails for users with a need to know

1. Sign in to the AWS Control Tower console at https://console.aws.amazon.com/controltower/.
2. From the left navigation, choose **Organizational units**.
3. From the table, choose the **name** of one of the OUs for which your user needs information about the applicable guardrails.
4. Note the name of the OU and the guardrails that apply to this OU.
5. Repeat the previous two steps for each OU about which your user needs information.

For detailed information about the guardrails and their functions, see Guardrails in AWS Control Tower (p. 20).

Tips for Landing Zone Setup

- The AWS Region where you do the most work should be your home Region.
- Set up your landing zone and deploy your Account Factory accounts from within your home Region.
- If you’re investing in several AWS Regions, be sure that your cloud resources are in the Region where you’ll do most of your cloud administrative work and run your workloads.
- The audit and other buckets are created in the same AWS Region from which you launch AWS Control Tower. We recommend that you do not move these buckets.
- You can make your own log buckets in the log archive account. Just be sure to leave the buckets created by AWS Control Tower. Note that your Amazon S3 access logs must be in the same Region as the source buckets.
- By keeping your workloads and logs in the same AWS Region, you reduce the cost that would be associated with moving and retrieving log information across Regions.
- The VPC created by AWS Control Tower is limited to the AWS Regions in which AWS Control Tower is available. Some customers whose workloads run in non-supported Regions may want to disable the VPC that is created with your Account Factory account. They may want to create a new VPC using the AWS Service Catalog portfolio, or create a custom VPC that runs in only the required Regions.
- If you delete your default VPC in your home Region, it's best to delete it in all other unnecessary Regions.

Guardrails in AWS Control Tower

A guardrail is a high-level rule that provides ongoing governance for your overall AWS environment. It's expressed in plain language. When users perform work in an AWS account in your landing zone, they're subject to guardrails.

The behavior of each guardrail is either preventive or detective.

- **Prevention** – A preventive guardrail ensures that your accounts maintain compliance, because it disallows actions that lead to policy violations. The status of a preventive guardrail is either **enforced** or **not enabled**. Preventive guardrails are supported in all AWS Regions.
- **Detection** – A detective guardrail detects noncompliance of resources within your accounts, such as policy violations, and provides alerts through the dashboard. The status of a detective guardrail is either **clear**, **in violation**, or **not enabled**. Detective guardrails apply only in those AWS Regions supported by AWS Control Tower.

AWS Control Tower provides mandatory, strongly recommended, and elective guardrails. When you create a new landing zone, all mandatory guardrails are enforced by default. Strongly recommended and elective guardrails are not enabled.

Guardrails enable you to express your policy intentions. AWS Control Tower implements preventive or detective controls to govern and monitor compliance of your resources across AWS accounts. For example, enable the **Disallow public read access to S3 buckets** guardrail to deny public read access to all S3 buckets for all accounts under an OU. When you enable guardrails on organizational units, they are applied to all child accounts under the OU.

Implementation of guardrails:

- The preventive guardrails are implemented using Service Control Policies (SCPs), which are part of AWS Organizations.
- The detective guardrails are implemented using AWS Config rules and AWS Lambda functions.

Considerations

When working with guardrails, consider the following:

- After creating your landing zone, all resources in your landing zone are subject to guardrails.
- OUs created through AWS Control Tower have guardrails applied to them. OUs created outside of a landing zone can't have guardrails applied to them, and they do not display in the AWS Control Tower console.
- Accounts created through Account Factory inherit their parent OU's guardrails. Accounts created outside of a landing zone do not, and they are not displayed in the AWS Control Tower console.
- The root user and any IAM administrators in the master account can perform work that guardrails would otherwise deny. This exception is intentional. It prevents the master account from entering into an unusable state. All actions taken within the master account continue to be tracked in the logs contained within the log archive account, for purposes of accountability and auditing.

Optional Guardrails

Three kinds of guidance apply to guardrails: mandatory, strongly recommended, and elective. Mandatory guardrails are always enforced. Strongly recommended guardrails are based on best practices for well-architected multi-account environments. Elective guardrails enable you to track or lock down actions that are commonly restricted in an AWS enterprise environment.

Strongly recommended and elective guardrails are optional, which means that you can customize the level of enforcement for your landing zone by choosing which ones to enable. Optional guardrails are not enabled by default. For more information, see the following guardrail references:

- Strongly Recommended Guardrails (p. 33)
- Elective Guardrails (p. 41)

The guidance of a guardrail is independent of whether it is preventive or detective.

Guardrail Details

In the guardrail details page of the console, you can find the following details for each guardrail:

- **Name** – The name of the guardrail.
- **Description** – A description of the guardrail.
- **Guidance** – The guidance is either mandatory, strongly recommended, or elective.
- **Category** – The category can be Audit Logs, Monitoring, Data Security, Network, IAM, or Control Tower Setup.
- **Behavior** – A guardrail's behavior is set to either preventive or detective.
- **Compliance Status** – A guardrail's compliance status can be clear, compliant, enforced, unknown, or in violation.

On the guardrail details page, you can also see guardrail artifacts. The guardrail is implemented by one or more artifacts. These artifacts can include a baseline AWS CloudFormation template, a service control policy to prevent account-level configuration changes or activity that may create configuration drift, and AWS Config Rules to detect account-level policy violations.

Enabling Guardrails

Most guardrails are enabled automatically according to an OU's configuration, and some guardrails can be enabled manually on your OUs. The following procedure describes the steps for enabling guardrails on an OU.

> **Important**
> When you enable guardrails with strongly recommended guidance, AWS Control Tower managed AWS resources are created in your accounts. Do not modify or delete resources created by AWS Control Tower. Doing so could result in the guardrails entering an unknown state.

To enable guardrails in an OU

1. Using a web browser, navigate to the AWS Control Tower console at https://console.aws.amazon.com/controltower.
2. From the left navigation, choose **Guardrails**.
3. Choose a guardrail that you want to enable; for example, **Guardrail: Enable encryption for EBS volumes attached to EC2 instances**. This choice opens the guardrail's details page.
4. From **Organizational units enabled**, choose **Enable guardrail on OU**.
5. A new page is displayed that lists the names of your OUs. Identify the OU on which you want to enable this guardrail.

6. Choose **Enable guardrail on OU**.
7. Your guardrail is now enabled. It may take several minutes for the change to complete. When it does, you'll see that this guardrail is enabled on the OU you selected. You can enable only one guardrail at a time.

Guardrail Reference

The following sections include a reference for each of the guardrails available in AWS Control Tower. Each guardrail reference includes the details, artifacts, additional information, and considerations to keep in mind when enabling a specific guardrail on a OU in your landing zone.

Topics

- Mandatory Guardrails (p. 22)
- Strongly Recommended Guardrails (p. 33)
- Elective Guardrails (p. 41)

Mandatory Guardrails

Mandatory guardrails are enabled by default when you set up your landing zone and can't be disabled. Following, you'll find a reference for each of the mandatory guardrails available in AWS Control Tower.

Enable Encryption at Rest for Log Archive

This guardrail enables encryption at rest for the Amazon S3 buckets in the log archive account. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on the **Core** OU.

The artifact for this guardrail is the following service control policy (SCP).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRAUDITBUCKETENCRYPTIONENABLED",
            "Effect": "Deny",
            "Action": [
                "s3:PutEncryptionConfiguration"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Enable Access Logging for Log Archive

This guardrail enables access logging in the log archive shared account. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on the **Core** OU.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRAUDITBUCKETLOGGINGENABLED",
            "Effect": "Deny",
            "Action": [
                "s3:PutBucketLogging"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Disallow Changes to CloudWatch Logs Log Groups

This guardrail prevents changes to CloudWatch Logs log groups that AWS Control Tower created in the log archive account when you set up your landing zone. It also prevents modifying retention policy in customer accounts. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on all OUs.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRLOGGROUPPOLICY",
            "Effect": "Deny",
            "Action": [
                "logs:DeleteLogGroup",
                "logs:PutRetentionPolicy"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:*aws-controltower*"
            ],
            "Condition": {
                "StringNotLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::*:role/AWSControlTowerExecution"
                    ]
                }
            }
        }
    ]
}
```

Disallow Deletion of AWS Config Aggregation Authorization

This guardrail prevents deletion of AWS Config aggregation authorizations that AWS Control Tower created in the audit account when you set up your landing zone. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRCONFIGAGGREGATIONAUTHORIZATIONPOLICY",
      "Effect": "Deny",
      "Action": [
        "config:DeleteAggregationAuthorization"
      ],
      "Resource": [
        "arn:aws:config:*:*:aggregation-authorization*"
      ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalArn": "arn:aws:iam::*:role/AWSControlTowerExecution"
        },
        "StringLike": {
          "aws:ResourceTag/aws-control-tower": "managed-by-control-tower"
        }
      }
    }
  ]
}
```

Disallow Deletion of Log Archive

This guardrail prevents deletion of Amazon S3 buckets created by AWS Control Tower in the log archive account. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on the **Core** OU.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRAUDITBUCKETDELETIONPROHIBITED",
      "Effect": "Deny",
      "Action": [
        "s3:DeleteBucket"
        ],
      "Resource": [
        "arn:aws:s3:::aws-controltower*"
        ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
          }
      }
    }
  ]
}
```

Disallow Policy Changes to Log Archive

This guardrail disallows any policy changes from occurring in the log archive shared account. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on the **Core** OU.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRAUDITBUCKETPOLICYCHANGESPROHIBITED",
            "Effect": "Deny",
            "Action": [
                "s3:PutBucketPolicy"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Disallow Public Read Access to Log Archive

This guardrail detects whether public read access is enabled to the Amazon S3 buckets in the log archive shared account. This guardrail does not change the status of the account. This is a detective guardrail with mandatory guidance. By default, this guardrail is enabled on the **Core** OU.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check that your S3 buckets do not allow public
 access
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForS3PublicRead:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks that your S3 buckets do not allow public read access. If an S3
 bucket policy or bucket ACL allows public read access, the bucket is noncompliant.
      Source:
        Owner: AWS
        SourceIdentifier: S3_BUCKET_PUBLIC_READ_PROHIBITED
      Scope:
        ComplianceResourceTypes:
          - AWS::S3::Bucket
```

Disallow Public Write Access to Log Archive

This guardrail detects whether public write access is enabled to the Amazon S3 buckets in the log archive shared account. This guardrail does not change the status of the account. This is a detective guardrail with mandatory guidance. By default, this guardrail is enabled on the **Core** OU.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check that your S3 buckets do not allow public
 access
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForS3PublicWrite:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks that your S3 buckets do not allow public write access. If an S3
 bucket policy or bucket ACL allows public write access, the bucket is noncompliant.
      Source:
        Owner: AWS
        SourceIdentifier: S3_BUCKET_PUBLIC_WRITE_PROHIBITED
      Scope:
        ComplianceResourceTypes:
          - AWS::S3::Bucket
```

Set a Retention Policy for Log Archive

This guardrail sets a retention policy of 365 days on the logs in the log archive shared account. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on the **Core** OU.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRAUDITBUCKETRETENTIONPOLICY",
            "Effect": "Deny",
            "Action": [
                "s3:PutLifecycleConfiguration"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Disallow Configuration Changes to CloudTrail

This guardrail prevents configuration changes to CloudTrail in your landing zone. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on all OUs.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCLOUDTRAILENABLED",
            "Effect": "Deny",
            "Action": [
                "cloudtrail:DeleteTrail",
                "cloudtrail:PutEventSelectors",
                "cloudtrail:StopLogging",
                "cloudtrail:UpdateTrail"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Integrate CloudTrail Events with CloudWatch Logs

This guardrail performs real-time analysis of activity data by sending CloudTrail events to CloudWatch Logs log files. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled on all OUs.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCLOUDTRAILENABLED",
            "Effect": "Deny",
            "Action": [
                "cloudtrail:DeleteTrail",
                "cloudtrail:PutEventSelectors",
                "cloudtrail:StopLogging",
                "cloudtrail:UpdateTrail"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Enable CloudTrail in All Available Regions

This guardrail enables CloudTrail in all available AWS Regions. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCLOUDTRAILENABLED",
            "Effect": "Deny",
            "Action": [
                "cloudtrail:DeleteTrail",
                "cloudtrail:PutEventSelectors",
                "cloudtrail:StopLogging",
                "cloudtrail:UpdateTrail"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Enable Integrity Validation for CloudTrail Log File

This guardrail enables integrity validation for the CloudTrail log file in all accounts and OUs. It protects the integrity of account activity logs using CloudTrail log file validation, which creates a digitally signed digest file that contains a hash of each log that CloudTrail writes to Amazon S3. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCLOUDTRAILENABLED",
            "Effect": "Deny",
            "Action": [
                "cloudtrail:DeleteTrail",
                "cloudtrail:PutEventSelectors",
                "cloudtrail:StopLogging",
                "cloudtrail:UpdateTrail"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Disallow Changes to CloudWatch Set Up by AWS Control Tower

This guardrail disallows changes to CloudWatch as it was configured by AWS Control Tower when you set up your landing zone. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRCLOUDWATCHEVENTPOLICY",
      "Effect": "Deny",
      "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "events:DisableRule",
        "events:DeleteRule"
      ],
      "Resource": [
        "arn:aws:events:*:*:rule/aws-controltower-*"
      ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
        }
        }
      }
    ]
}
```

Disallow Changes to AWS Config Aggregation Set Up by AWS Control Tower

This guardrail disallows changes to the AWS Config aggregation settings made by AWS Control Tower to collect configuration and compliance data when you set up your landing zone. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRCONFIGRULETAGSPOLICY",
      "Effect": "Deny",
      "Action": [
        "config:TagResource",
        "config:UntagResource"
      ],
      "Resource": ["*"],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": "aws-control-tower"
        }
      }
    }
  ]
}
```

Disallow Configuration Changes to AWS Config

This guardrail disallows configuration changes to AWS Config. It ensures that AWS Config records resource configurations in a consistent manner by disallowing AWS Config settings changes. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCONFIGENABLED",
            "Effect": "Deny",
            "Action": [
                "config:DeleteConfigurationRecorder",
                "config:DeleteDeliveryChannel",
                "config:DeleteRetentionConfiguration",
                "config:PutConfigurationRecorder",
                "config:PutDeliveryChannel",
                "config:PutRetentionConfiguration",
                "config:StopConfigurationRecorder"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Enable AWS Config in All Available Regions

This guardrail enables AWS Config in all available AWS Regions. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRCONFIGENABLED",
            "Effect": "Deny",
            "Action": [
                "config:DeleteConfigurationRecorder",
                "config:DeleteDeliveryChannel",
                "config:DeleteRetentionConfiguration",
                "config:PutConfigurationRecorder",
                "config:PutDeliveryChannel",
                "config:PutRetentionConfiguration",
                "config:StopConfigurationRecorder"
            ],
            "Resource": ["*"],
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
                }
            }
        }
    ]
}
```

Disallow Changes to AWS Config Rules Set Up by AWS Control Tower

This guardrail disallows changes to AWS Config Rules that were implemented by AWS Control Tower when the landing zone was set up. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRCONFIGRULEPOLICY",
      "Effect": "Deny",
      "Action": [
        "config:PutConfigRule",
        "config:DeleteConfigRule",
        "config:DeleteEvaluationResults",
        "config:DeleteConfigurationAggregator",
        "config:PutConfigurationAggregator"
      ],
      "Resource": ["*"],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
        },
        "StringEquals": {
          "aws:ResourceTag/aws-control-tower": "managed-by-control-tower"
        }
      }
    }
  ]
}
```

Disallow Changes to IAM Roles Set Up by AWS Control Tower

This guardrail disallows changes to the IAM roles that were created by AWS Control Tower when the landing zone was set up. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRIAMROLEPOLICY",
      "Effect": "Deny",
      "Action": [
        "iam:AttachRolePolicy",
        "iam:CreateRole",
        "iam:DeleteRole",
        "iam:DeleteRolePermissionsBoundary",
        "iam:DeleteRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePermissionsBoundary",
        "iam:PutRolePolicy",
        "iam:UpdateAssumeRolePolicy",
        "iam:UpdateRole",
        "iam:UpdateRoleDescription"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-controltower-*",
        "arn:aws:iam::*:role/*AWSControlTower*"
      ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

Disallow Changes to Lambda Functions Set Up by AWS Control Tower

This guardrail disallows changes to Lambda functions set up by AWS Control Tower. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRLAMBDAFUNCTIONPOLICY",
      "Effect": "Deny",
      "Action": [
        "lambda:AddPermission",
        "lambda:CreateEventSourceMapping",
        "lambda:CreateFunction",
        "lambda:DeleteEventSourceMapping",
        "lambda:DeleteFunction",
        "lambda:DeleteFunctionConcurrency",
        "lambda:PutFunctionConcurrency",
        "lambda:RemovePermission",
        "lambda:UpdateEventSourceMapping",
        "lambda:UpdateFunctionCode",
        "lambda:UpdateFunctionConfiguration"
      ],
      "Resource": [
        "arn:aws:lambda:*:*:function:aws-controltower-*"
      ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

Disallow Changes to Amazon SNS Set Up by AWS Control Tower

This guardrail disallows changes to Amazon SNS set up by AWS Control Tower. It protects the integrity of Amazon SNS notification settings for your landing zone. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRSNSTOPICPOLICY",
      "Effect": "Deny",
      "Action": [
        "sns:AddPermission",
        "sns:CreateTopic",
        "sns:DeleteTopic",
        "sns:RemovePermission",
        "sns:SetTopicAttributes"
      ],
      "Resource": [
        "arn:aws:sns:*:*:aws-controltower-*"
      ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
    ]
}
```

Disallow Changes to Amazon SNS Subscriptions Set Up by AWS Control Tower

This guardrail disallows changes to Amazon SNS subscriptions set up by AWS Control Tower. It protects the integrity of Amazon SNS subscriptions settings for your landing zone. This is a preventive guardrail with mandatory guidance. By default, this guardrail is enabled in all OUs.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRSNSSUBSCRIPTIONPOLICY",
      "Effect": "Deny",
      "Action": [
        "sns:Subscribe",
        "sns:Unsubscribe"
      ],
      "Resource": [
        "arn:aws:sns:*:*:aws-controltower-SecurityNotifications"
      ],
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/AWSControlTowerExecution"
        }
      }
    }
  ]
}
```

Strongly Recommended Guardrails

Strongly recommended guardrails are based on best practices for well-architected multi-account environments. These guardrails are not enabled by default, and can be disabled. Following, you'll find a reference for each of the strongly recommended guardrails available in AWS Control Tower.

Disallow Creation of Access Keys for the Root User

Secures your AWS accounts by disallowing creation of access keys for the root user. We recommend that you instead create access keys for the IAM users with limited permissions to interact with your AWS account. This is a preventive guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRRESTRICTROOTUSERACCESSKEYS",
            "Effect": "Deny",
            "Action": "iam:CreateAccessKey",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::*:root"
                    ]
                }
            }
        }
    ]
}
```

Disallow Actions as a Root User

Secures your AWS accounts by disallowing account access with root user credentials, which are credentials of the account owner that allow unrestricted access to all resources in the account. Instead, we recommend that you create AWS Identity and Access Management (IAM) users for everyday interaction with your AWS account. This is a preventive guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GRRESTRICTROOTUSER",
      "Effect": "Deny",
      "Action": "*",
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringLike": {
          "aws:PrincipalArn": [
            "arn:aws:iam::*:root"
          ]
        }
      }
    }
  ]
}
```

Enable Encryption for Amazon EBS Volumes Attached to Amazon EC2 Instances

This guardrail detects whether encryption is enabled for Amazon EBS volumes attached to Amazon EC2 instances in your landing zone. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail isn't enabled on any OUs.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check for encryption of all storage volumes
 attached to compute
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
```

```
Resources:
  CheckForEncryptedVolumes:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether EBS volumes that are in an attached state are encrypted.
      Source:
        Owner: AWS
        SourceIdentifier: ENCRYPTED_VOLUMES
      Scope:
        ComplianceResourceTypes:
          - AWS::EC2::Volume
```

Disallow Internet Connection Through RDP

This guardrail detects whether internet connections are enabled to Amazon EC2 instances through services like Remote Desktop Protocol (RDP). This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether security groups that are in use
 disallow unrestricted incoming TCP traffic to the specified ports.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
  blockedPort1:
    Type: String
    Default: '20'
    Description: Blocked TCP port number.
  blockedPort2:
    Type: String
    Default: '21'
    Description: Blocked TCP port number.
  blockedPort3:
    Type: String
    Default: '3389'
    Description: Blocked TCP port number.
  blockedPort4:
    Type: String
    Default: '3306'
    Description: Blocked TCP port number.
  blockedPort5:
    Type: String
    Default: '4333'
    Description: Blocked TCP port number.
Conditions:
  blockedPort1:
    Fn::Not:
    - Fn::Equals:
      - ''
      - Ref: blockedPort1
  blockedPort2:
    Fn::Not:
    - Fn::Equals:
      - ''
      - Ref: blockedPort2
  blockedPort3:
    Fn::Not:
    - Fn::Equals:
      - ''
      - Ref: blockedPort3
  blockedPort4:
    Fn::Not:
    - Fn::Equals:
      - ''
      - Ref: blockedPort4
  blockedPort5:
    Fn::Not:
    - Fn::Equals:
      - ''
      - Ref: blockedPort5
Resources:
  CheckForRestrictedCommonPortsPolicy:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether security groups that are in use disallow unrestricted
 incoming TCP traffic to the specified ports.
      InputParameters:
        blockedPort1:
          Fn::If:
          - blockedPort1
          - Ref: blockedPort1
          - Ref: AWS::NoValue
        blockedPort2:
          Fn::If:
          - blockedPort2
          - Ref: blockedPort2
          - Ref: AWS::NoValue
        blockedPort3:
          Fn::If:
          - blockedPort3
          - Ref: blockedPort3
          - Ref: AWS::NoValue
        blockedPort4:
          Fn::If:
          - blockedPort4
          - Ref: blockedPort4
          - Ref: AWS::NoValue
        blockedPort5:
          Fn::If:
          - blockedPort5
          - Ref: blockedPort5
          - Ref: AWS::NoValue
      Scope:
        ComplianceResourceTypes:
        - AWS::EC2::SecurityGroup
      Source:
        Owner: AWS
        SourceIdentifier: RESTRICTED_INCOMING_TRAFFIC
```

Disallow Internet Connection Through SSH

This guardrail detects whether any internet connections are allowed through remote services like the Secure Shell (SSH) protocol. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether security groups that are in use
 disallow SSH
```

```
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForRestrictedSshPolicy:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether security groups that are in use disallow unrestricted
 incoming SSH traffic.
      Scope:
        ComplianceResourceTypes:
        - AWS::EC2::SecurityGroup
      Source:
        Owner: AWS
        SourceIdentifier: INCOMING_SSH_DISABLED
```

Enable MFA for the Root User

This guardrail detects whether multi-factor authentication (MFA) is enabled for the root user of the master account. MFA reduces vulnerability risks from weak authentication by adding an extra authentication code on top of a user name and password. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to require MFA for root access to accounts
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
  MaximumExecutionFrequency:
    Type: String
    Default: 24hours
    Description: The frequency that you want AWS Config to run evaluations for the rule.
    AllowedValues:
    - 1hour
    - 3hours
    - 6hours
    - 12hours
    - 24hours
Mappings:
  Settings:
    FrequencyMap:
      1hour    : One_Hour
      3hours   : Three_Hours
      6hours   : Six_Hours
      12hours : Twelve_Hours
      24hours : TwentyFour_Hours
Resources:
  CheckForRootMfa:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether the root user of your AWS account requires multi-factor
 authentication for console sign-in.
      Source:
        Owner: AWS
        SourceIdentifier: ROOT_ACCOUNT_MFA_ENABLED
      MaximumExecutionFrequency:
        !FindInMap
          - Settings
          - FrequencyMap
          - !Ref MaximumExecutionFrequency
```

Disallow Public Read Access to Amazon S3 Buckets

This guardrail detects whether public read access is allowed to Amazon S3 buckets. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check that your S3 buckets do not allow public
 access
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForS3PublicRead:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks that your S3 buckets do not allow public read access. If an S3
 bucket policy or bucket ACL allows public read access, the bucket is noncompliant.
      Source:
        Owner: AWS
        SourceIdentifier: S3_BUCKET_PUBLIC_READ_PROHIBITED
      Scope:
        ComplianceResourceTypes:
          - AWS::S3::Bucket
```

Disallow Public Write Access to Amazon S3 Buckets

This guardrail detects whether public write access is allowed to Amazon S3 buckets. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check that your S3 buckets do not allow public
 access
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForS3PublicWrite:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks that your S3 buckets do not allow public write access. If an S3
 bucket policy or bucket ACL allows public write access, the bucket is noncompliant.
      Source:
        Owner: AWS
        SourceIdentifier: S3_BUCKET_PUBLIC_WRITE_PROHIBITED
      Scope:
        ComplianceResourceTypes:
          - AWS::S3::Bucket
```

Disallow Amazon EBS Volumes That Are Unattached to An Amazon EC2 Instance

Detects whether an Amazon EBS volume persists independently from an Amazon EC2 instance. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether EBS volumes are attached to EC2
 instances
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
  deleteOnTermination:
    Type: 'String'
    Default: 'None'
    Description: 'Check for Delete on termination'
Conditions:
  deleteOnTermination:
    Fn::Not:
    - Fn::Equals:
      - 'None'
      - Ref: deleteOnTermination
Resources:
  CheckForEc2VolumesInUse:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether EBS volumes are attached to EC2 instances
      InputParameters:
        deleteOnTermination:
          Fn::If:
          - deleteOnTermination
          - Ref: deleteOnTermination
          - Ref: AWS::NoValue
      Source:
        Owner: AWS
        SourceIdentifier: EC2_VOLUME_INUSE_CHECK
      Scope:
        ComplianceResourceTypes:
          - AWS::EC2::Volume
```

Disallow Amazon EC2Instance Types That Are Not Amazon EBS-Optimized

Detects whether Amazon EC2 instances are launched without an Amazon EBS volume that is performance optimized. Amazon EBS-optimized volumes minimize contention between Amazon EBS I/O and other traffic from your instance. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Description: Configure AWS Config rules to check whether EBS optimization is enabled for
 your EC2 instances that can be EBS-optimized
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForEbsOptimizedInstance:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether EBS optimization is enabled for your EC2 instances that
 can be EBS-optimized
      Source:
        Owner: AWS
        SourceIdentifier: EBS_OPTIMIZED_INSTANCE
      Scope:
        ComplianceResourceTypes:
          - AWS::EC2::Instance
```

Disallow Public Access to Amazon RDS Database Instances

Detects whether your Amazon RDS database instances have public access enabled. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether Amazon RDS instances are not
 publicly accessible.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForRdsPublicAccess:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether the Amazon Relational Database Service (RDS) instances
 are not publicly accessible. The rule is non-compliant if the publiclyAccessible field is
 true in the instance configuration item.
      Source:
        Owner: AWS
        SourceIdentifier: RDS_INSTANCE_PUBLIC_ACCESS_CHECK
      Scope:
        ComplianceResourceTypes:
          - AWS::RDS::DBInstance
```

Disallow Public Access to Amazon RDS Database Snapshots

Detects whether your Amazon RDS database snapshots have public access enabled. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Checks if Amazon Relational Database Service (Amazon RDS) snapshots are
 public.
```

```
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForRdsStorageEncryption:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks if Amazon Relational Database Service (Amazon RDS) snapshots are
 public. The rule is non-compliant if any existing and new Amazon RDS snapshots are public.
      Source:
        Owner: AWS
        SourceIdentifier: RDS_SNAPSHOTS_PUBLIC_PROHIBITED
      Scope:
        ComplianceResourceTypes:
          - AWS::RDS::DBSnapshot
```

Disallow Amazon RDS Database Instances That Are Not Storage Encrypted

Detects whether your Amazon RDS database instances are not encrypted at rest, along with their automated backups, Read Replicas, and snapshots. This guardrail does not change the status of the account. This is a detective guardrail with strongly recommended guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether storage encryption is enabled for
 your RDS DB instances
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForRdsStorageEncryption:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether storage encryption is enabled for your RDS DB instances.
      Source:
        Owner: AWS
        SourceIdentifier: RDS_STORAGE_ENCRYPTED
      Scope:
        ComplianceResourceTypes:
          - AWS::RDS::DBInstance
```

Elective Guardrails

Elective guardrails enable you to lock down or track attempts at performing commonly restricted actions in an AWS enterprise environment. These guardrails are not enabled by default, and can be disabled. Following, you'll find a reference for each of the elective guardrails available in AWS Control Tower.

Disallow Cross-Region Replication for Amazon S3 Buckets

Restricts the location of your Amazon S3 data to a single AWS Region by disabling any automatic, asynchronous copying of objects across buckets to other AWS Regions. This is a preventive guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRRESTRICTS3CROSSREGIONREPLICATION",
            "Effect": "Deny",
            "Action": [
                "s3:PutReplicationConfiguration"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Disallow Delete Actions on Amazon S3 Buckets Without MFA

Protects your Amazon S3 buckets by requiring MFA for delete actions. MFA adds an extra authentication code on top of a user name and password. This is a preventive guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following SCP.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GRRESTRICTS3DELETEWITHOUTMFA",
            "Effect": "Deny",
            "Action": [
                "s3:DeleteObject",
                "s3:DeleteBucket"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "BoolIfExists": {
                    "aws:MultiFactorAuthPresent": [
                        "false"
                    ]
                }
            }
        }
    ]
}
```

Disallow Access to IAM Users Without MFA

Protects your account by requiring MFA for all IAM users in the account. MFA adds an extra authentication code on top of a username and password. This guardrail detects whether MFA is enabled. This guardrail does not change the status of the account. This is a detective guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
```

```
Description: Configure AWS Config rules to check whether the IAM users have MFA enabled
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
  MaximumExecutionFrequency:
    Type: String
    Default: 1hour
    Description: The frequency that you want AWS Config to run evaluations for the rule.
    AllowedValues:
    - 1hour
    - 3hours
    - 6hours
    - 12hours
    - 24hours
Mappings:
  Settings:
    FrequencyMap:
      1hour   : One_Hour
      3hours  : Three_Hours
      6hours  : Six_Hours
      12hours : Twelve_Hours
      24hours : TwentyFour_Hours
Resources:
  CheckForIAMUserMFA:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether the AWS Identity and Access Management users have multi-
factor authentication (MFA) enabled. The rule is COMPLIANT if MFA is enabled.
      Source:
        Owner: AWS
        SourceIdentifier: IAM_USER_MFA_ENABLED
      MaximumExecutionFrequency:
        !FindInMap
          - Settings
          - FrequencyMap
          - !Ref MaximumExecutionFrequency
```

Disallow Console Access to IAM Users Without MFA

Protects your account by requiring MFA for all IAM users in the console. MFA adds an extra authentication code on top of a username and password. This guardrail detects whether MFA is enabled. This guardrail does not change the status of the account. This is a detective guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether MFA is enabled for all AWS IAM
 users that use a console password.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
  MaximumExecutionFrequency:
    Type: String
    Default: 1hour
    Description: The frequency that you want AWS Config to run evaluations for the rule.
    AllowedValues:
    - 1hour
    - 3hours
    - 6hours
    - 12hours
    - 24hours
Mappings:
  Settings:
    FrequencyMap:
      1hour   : One_Hour
      3hours  : Three_Hours
      6hours  : Six_Hours
      12hours : Twelve_Hours
      24hours : TwentyFour_Hours
Resources:
  CheckForIAMUserConsoleMFA:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether AWS Multi-Factor Authentication (MFA) is enabled for all
 AWS Identity and Access Management (IAM) users that use a console password. The rule is
 COMPLIANT if MFA is enabled.
      Source:
        Owner: AWS
        SourceIdentifier: MFA_ENABLED_FOR_IAM_CONSOLE_ACCESS
      MaximumExecutionFrequency:
        !FindInMap
          - Settings
          - FrequencyMap
          - !Ref MaximumExecutionFrequency
```

Disallow Amazon S3 Buckets That Are Not Versioning Enabled

Detects whether your Amazon S3 buckets are not versioning enabled. Versioning allows you to recover objects from accidental deletion or overwrite. This guardrail does not change the status of the account. This is a detective guardrail with elective guidance. By default, this guardrail is not enabled.

The artifact for this guardrail is the following AWS Config rule.

```
AWSTemplateFormatVersion: 2010-09-09
Description: Configure AWS Config rules to check whether versioning is enabled for your S3
 buckets.
Parameters:
  ConfigRuleName:
    Type: 'String'
    Description: 'Name for the Config rule'
Resources:
  CheckForS3VersioningEnabled:
    Type: AWS::Config::ConfigRule
    Properties:
      ConfigRuleName: !Sub ${ConfigRuleName}
      Description: Checks whether versioning is enabled for your S3 buckets.
      Source:
        Owner: AWS
        SourceIdentifier: S3_BUCKET_VERSIONING_ENABLED
      Scope:
        ComplianceResourceTypes:
          - AWS::S3::Bucket
```

Integrated Services

AWS Control Tower is a well-architected service that's built on top of other AWS services. This chapter provides a brief overview of these services, including configuration information about the following services and how they work in AWS Control Tower.

Topics

Scripting Environments with AWS CloudFormation

AWS CloudFormation enables you to create and provision AWS infrastructure deployments predictably and repeatedly. It helps you leverage AWS products to build highly reliable, highly scalable, cost-effective applications in the cloud without worrying about creating and configuring the underlying AWS infrastructure. AWS CloudFormation enables you to use a template file to create and delete a collection of resources together as a single unit (a stack). For more information, see *AWS CloudFormation User Guide*.

AWS Control Tower uses AWS CloudFormation stacksets to apply guardrails on accounts.

Monitoring Events with CloudTrail

With AWS CloudTrail, you can monitor your AWS deployments in the cloud by getting a history of AWS API calls for your accounts. You can also identify which users and accounts called AWS APIs for services that support CloudTrail, the source IP address the calls were made from, and when the calls occurred. You can integrate CloudTrail into applications using the API, automate trail creation for your organization, check the status of your trails, and control how administrators turn CloudTrail logging on and off. For more information, see *AWS CloudTrail User Guide*.

Monitoring Resources and Services with CloudWatch

Amazon CloudWatch provides a reliable, scalable, and flexible monitoring solution that you can start using within minutes. You no longer need to set up, manage, and scale your own monitoring systems and infrastructure. For more information, see *Amazon CloudWatch User Guide*.

Govern Resource Configurations with AWS Config

AWS Config provides a detailed view of the resources associated with your AWS account, including how they are configured, how they are related to one another, and how the configurations and their relationships have changed over time. For more information, see *AWS Config Developer Guide*.

AWS Control Tower uses AWS Config Rules with some guardrails. For more information, see Guardrails in AWS Control Tower (p. 20).

Manage Permissions for Entities with IAM

AWS Identity and Access Management (IAM) is a web service for securely controlling access to AWS services. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users and applications can access.

When you set up your landing zone, a number of groups are created for AWS SSO. These groups have permission sets that are pre-defined permissions policies from IAM. Your end users can also use IAM to define the scope of permissions for IAM users and other entities within member accounts.

Run Serverless Compute Functions with Lambda

With AWS Lambda, you can run code without provisioning or managing servers. You can run code for virtually any type of application or backend service—all with zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

Manage Accounts Through AWS Organizations

AWS Organizations is an account management service that lets you consolidate multiple AWS accounts into an organization that you create and centrally manage. With Organizations, you can create member accounts and invite existing accounts to join your organization. You can organize those accounts into groups and attach policy-based controls. For more information, see *AWS Organizations User Guide*.

In AWS Control Tower, Organizations helps centrally manage billing; control access, compliance, and security; and share resources across your member AWS accounts. Accounts are grouped into logical groups, called organizational units (OUs). For more information on Organizations, see *AWS Organizations User Guide*.

AWS Control Tower uses the following OUs:

- **Root** – The parent container for all accounts and all other OUs in your landing zone.
- **Core** – This OU contains the log archive account, the audit account, and the resources they own.
- **Custom OU** – This OU is created when you set up your landing zone. It and other child OUs in your landing zone contain your member accounts. These are the accounts that your end users access to perform work on AWS resources.

> **Note**
> You can add additional OUs in your landing zone through the AWS Control Tower console on the **Organizational units** page.

Considerations

OUs created through AWS Control Tower can have guardrails applied to them. OUs created outside of AWS Control Tower cannot, and they are not displayed in AWS Control Tower.

Store Objects with Amazon S3

Amazon Simple Storage Service (Amazon S3) is storage for the internet. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web. You can accomplish these tasks using the simple and intuitive web interface of the AWS Management Console. For more information, see *Amazon Simple Storage Service Console User Guide*.

When you set up your landing zone, an Amazon S3 bucket is created in your log archive account to contain all logs across all accounts in your landing zone.

Provisioning Accounts Through AWS Service Catalog

AWS Service Catalog enables IT administrators to create, manage, and distribute portfolios of approved products to end users, who can then access the products they need in a personalized portal. Typical products include servers, databases, websites, or applications that are deployed using AWS resources. You can control which users have access to specific products to enforce compliance with organizational business standards, manage product lifecycles, and help users find and launch products with confidence. For more information, see *AWS Service Catalog Administrator Guide*.

In AWS Control Tower, your central cloud administrators and your end users can provision accounts in your landing zone using Account Factory, a product in AWS Service Catalog. For more information, see Account Factory (p. 52).

Managing Users and Access Through AWS Single Sign-On

AWS Single Sign-On is a cloud-based service that simplifies managing SSO access to AWS accounts and business applications. You can control SSO access and user permissions across all your AWS accounts in AWS Organizations. You can also administer access to popular business applications and custom applications that support Security Assertion Markup Language (SAML) 2.0. In addition, AWS SSO offers a user portal where your users can find all their assigned AWS accounts, business applications, and custom applications in one place. For more information, see *AWS Single Sign-On User Guide*.

In AWS Control Tower, AWS Single Sign-On allows both central cloud administrators and end users to manage SSO access to multiple AWS accounts and business applications. AWS Control Tower uses this service for the creation and management of user access to the accounts created in AWS Service Catalog.

> **Note**
> When you first set up AWS Control Tower, only the root user and any IAM users with the correct permissions can add AWS SSO users. However, once end users have been added in the **AWSAccountFactory** group, they can create new SSO users from the Account Factory wizard. For more information, see Account Factory (p. 52).

Your landing zone is set up with a directory to manage user identities and single sign-on to provide your users with federated access across accounts. When you set up your landing zone, you have a default directory. This directory is preconfigured with user groups and permission sets.

The groups are designed for you to easily manage specialized roles within your shared accounts. You can create new groups for your end users in your member accounts. The permission sets available cover a broad range of distinct user permission use cases like read-only access, AWS Control Tower admin access, and AWS Service Catalog access. These permission sets enable your end users to quickly provision their own AWS accounts in your landing zone.

For more information on how to use this service in the context of AWS Control Tower, see the following topics in the *AWS Single Sign-On User Guide*.

- To add users, see Add Users.
- To add users to groups, see Add Users to Groups.
- To edit user properties, see Edit User Properties.
- To add group, see Add Groups.

> **Warning**
> When using AWS Control Tower, your AWS SSO directory is in US East (N. Virginia). If you set up your landing zone in another Region and then navigate to the AWS SSO console, you must change the Region to the US East (N. Virginia). Do not delete your AWS SSO configuration in US East (N. Virginia).

AWS SSO Groups for AWS Control Tower

AWS Control Tower offers preconfigured groups to organize users that perform specific tasks in your accounts. You can add users and assign them to these groups directly in AWS SSO. Doing so matches permission sets to users in groups within your accounts. The groups created when you set up your landing zone are as follows.

AWSAccountFactory

Account	Permission sets	Description
Master account	AWSServiceCatalogEndUserAccess	This group is only used in this account to provision new accounts using Account Factory.

AWSServiceCatalogAdmins

Account	Permission sets	Description
Master account	AWSServiceCatalogAdminFullAccess	This group is only used in this account to make administrative changes to Account Factory. Users in this group can't provision new accounts unless they're also in the **AWSAccountFactory** group.

AWSControlTowerAdmins

Account	Permission sets	Description
Master account	AWSAdministratorAccess	Users of this group in this account are the only ones that can access the AWS Control Tower console.
Log archive account	AWSAdministratorAccess	Users will have administrator access in this account.
Audit account	AWSAdministratorAccess	Users will have administrator access in this account.
Member accounts	AWSOrganizationsFullAccess	Users will have full access to Organizations in this account.

AWSSecurityAuditPowerUsers

Account	Permission sets	Description
Master account	AWSPowerUserAccess	Users can perform application development tasks and can create and configure resources and services that support AWS aware application development.
Log archive account	AWSPowerUserAccess	Users can perform application development tasks and can create and configure resources and services that support AWS aware application development.
Audit account	AWSPowerUserAccess	Users can perform application development tasks and can create and configure resources and services that support AWS aware application development.
Member accounts	AWSPowerUserAccess	Users can perform application development tasks and can create and configure resources and services that support AWS aware application development.

AWSSecurityAuditors

Account	Permission sets	Description
Master account	AWSReadOnlyAccess	Users have read-only access to all AWS services and resources in this account.
Log archive account	AWSReadOnlyAccess	Users have read-only access to all AWS services and resources in this account.

Account	Permission sets	Description
Audit account	AWSReadOnlyAccess	Users have read-only access to all AWS services and resources in this account.
Member accounts	AWSReadOnlyAccess	Users have read-only access to all AWS services and resources in this account.

AWSLogArchiveAdmins

Account	Permission sets	Description
Log archive account	AWSAdministratorAccess	Users will have administrator access in this account.

AWSLogArchiveViewers

Account	Permission sets	Description
Log archive account	AWSReadOnlyAccess	Users have read-only access to all AWS services and resources in this account.

AWSAuditAccountAdmins

Account	Permission sets	Description
Audit account	AWSAdministratorAccess	Users will have administrator access in this account.

Tracking Alerts Through Amazon Simple Notification Service

Amazon Simple Notification Service (Amazon SNS) is a web service that enables applications, end-users, and devices to instantly send and receive notifications from the cloud. For more information, see *Amazon Simple Notification Service Developer Guide*.

AWS Control Tower uses Amazon SNS to send programmatic alerts to your master account and audit account email addresses. These alerts help you prevent drift within your landing zone. For more information, see Detecting and Resolving Drift in AWS Control Tower (p. 58).

Build Distributed Applications with AWS Step Functions

AWS Step Functions makes it easy to coordinate the components of distributed applications as a series of steps in a visual workflow. You can quickly build and run state machines to execute the steps of your application in a reliable and scalable fashion. For more information, see *AWS Step Functions Developer Guide*.

Account Factory

This chapter includes an overview and procedures for Account Factory, the AWS Service Catalog console-based product used to provision new accounts in your landing zone.

Configuring and Provisioning Accounts Through AWS Service Catalog

With Account Factory, central cloud administrators and AWS Single Sign-On end users can provision accounts in your landing zone. By default, AWS SSO users that provision accounts must be in the **AWSAccountFactory** group or the master group. Exercise caution when working from the master account, as you would when using any account that has generous permissions across your organization.

However, if you're provisioning accounts programmatically, the identity that will perform this work must have the following IAM permissions policy, in addition to `AWSServiceCatalogEndUserFullAccess`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSControlTowerAccountFactoryAccess",
            "Effect": "Allow",
            "Action": [
                "sso:GetProfile",
                "sso:CreateProfile",
                "sso:UpdateProfile",
                "sso:AssociateProfile",
                "sso:CreateApplicationInstance",
                "sso:GetSSOStatus",
                "sso:GetTrust",
                "sso:CreateTrust",
                "sso:UpdateTrust",
                "sso:GetPeregrineStatus",
                "sso:GetApplicationInstance",
                "sso:ListDirectoryAssociations",
                "sso:ListPermissionSets",
                "sso:GetPermissionSet",
                "sso:ProvisionApplicationInstanceForAWSAccount",
                "sso:ProvisionApplicationProfileForAWSAccountInstance",
                "sso:ProvisionSAMLProvider",
                "sso:ListProfileAssociations",
                "sso-directory:ListMembersInGroup",
                "sso-directory:AddMemberToGroup",
                "sso-directory:SearchGroups",
                "sso-directory:SearchGroupsWithGroupName",
                "sso-directory:SearchUsers",
                "sso-directory:CreateUser",
                "sso-directory:DescribeGroups",
                "sso-directory:DescribeDirectory",
                "sso-directory:GetUserPoolInfo",
                "controltower:CreateManagedAccount",
                "controltower:DescribeManagedAccount",
                "controltower:DeregisterManagedAccount",
                "s3:GetObject"
            ],
            "Resource": "*"
        }
    ]
}
```

For more information on using AWS SSO with AWS Control Tower, see Managing Users and Access Through AWS Single Sign-On (p. 47). The following procedure describes how to provision accounts as an AWS SSO end user.

To provision accounts in Account Factory as an end user

1. Sign in your user portal URL.
2. From **Your applications**, choose **AWS Account**.
3. From the list of accounts, choose the account ID for your master account. It also has the label, **(Master)**. Ensure that you've selected the correct AWS Region for provisioning accounts, which should be your AWS Control Tower home region.
4. From **AWSServiceCatalogEndUserAccess**, choose **Management console**. This opens the AWS Management Console for this user in this account.
5. Search for and choose **Service Catalog** to open the AWS Service Catalog console.
6. From the navigation pane, choose **Products list**.
7. From **AWS Control Tower Account Factory**, choose the drop-down menu and select **Launch product**. This starts the wizard to provision a new account.
8. Fill in the information, and keep the following in mind:
   - The **SSOUserEmail** can be a new email address, or the email address associated with an existing AWS SSO user. Whichever you choose, this user will have administrative access to the account you're provisioning.
   - The **AccountEmail** must be an email address that isn't already associated with an AWS account. If you used a new email address in **SSOUserEmail**, you can use that email address here.
9. When you're finished, choose **NEXT** until you get to the **Review** page of the wizard. Do not define **TagOptions** and do not enable **Notifications**, otherwise the account can fail to be provisioned.
10. Review your account settings, and then choose **LAUNCH**. Do not create a resource plan, otherwise the account will fail to be provisioned.
11. Your account is now being provisioned. It can take a few minutes to complete. You can refresh the page to update the displayed status information. Only one account can be provisioned at a time.

Accounts that you provision can be closed or they can be changed to unmanaged accounts. Alternatively, you can repurpose accounts for other workloads and other users by updating the email addresses and user parameters for the account. You can change the Organizational Unit for the account by following the update procedures. For more information on unmanaging an account, see Unmanaging a Member Account (p. 55).

Updating Account Factory Accounts

The following procedure guides you through how to update or migrate your Account Factory accounts.

> **Important**
> Use this procedure to migrate an account's OU, as for any other update.

To update an Account Factory account

1. Sign in to the AWS Management Console and open the AWS Single Sign-On console at https://console.aws.amazon.com/singlesignon/.

   > **Note**
   > You must be signed in as a user with the permissions to provision new products in AWS Service Catalog; for example, an AWS SSO user in either the **AWSAccountFactory** or **AWSServiceCatalogAdmins** groups.

2. Choose **Provision new account** to open the AWS Service Catalog console and the Account Factory product.
3. From the navigation pane, choose **Provisioned products list**.
4. For each account listed, perform the following steps to update all your member accounts:

   a. From the drop-down menu for the account, choose **Provisioned product details**.

   b. Make a note of the following parameters:

      - **SSOUserEmail**
      - **AccountEmail**
      - **SSOUserFirstName**
      - **SSOUSerLastName**
      - **AccountName**

   c. From **ACTIONS**, choose **Update**.

   d. Choose the button next to the **Version** of the product you want to update, and choose **NEXT**.

   e. Provide the parameter values that were mentioned previously. For **ManagedOrganizationlUnit**, choose the OU that the account was already in, or to migrate to a new OU, choose the new OU for the account. A central cloud administrator can find this information in the AWS Control Tower console, under **Accounts**.

   f. Choose **NEXT**.

   g. Review your changes, and then choose **UPDATE**. This process can take a few minutes per account.

Configuring Account Factory with Amazon Virtual Private Cloud Settings

Account Factory enables you to create pre-approved baselines and configuration options for accounts in your organization. You can configure and provision new accounts through AWS Service Catalog.

On the Account Factory page, you can view the Amazon VPC configuration options available to your end users when they provision new accounts. You can see a list of organizational units (OUs) and their **allow list** status. By default, all OUs are on the allow list, which means that accounts can be provisioned under them. You can disable certain OUs for account provisioning through AWS Service Catalog.

To configure Amazon VPC settings in Account Factory

1. As a central cloud administrator, sign into the AWS Control Tower console with administrator permissions in the master account.
2. From the left side of the dashboard, select **Account Factory** to navigate to the Account Factory network configuration page. There you can see the default network settings displayed. To edit, select **Edit** and view the editable version of your Account Factory network configuration settings.
3. You can modify the each field of the default settings as needed. Choose the VPC configuration options you'd like to establish for all new Account Factory accounts that your end users may create, and enter these settings into the fields:

- Choose **disabled** or **enabled** to create a public subnet in Amazon VPC. By default, the internet-accessible subnet is disallowed.
- Choose the maximum number of private subnets in Amazon VPC from the list. By default, 1 is selected. The maximum number of private subnets allowed is 2.
- Enter the range of IP addresses for creating your account VPCs. The value must be in the form of a classless inter-domain routing (CIDR) block (for example, the default is `172.31.0.0/16`). This CIDR

block provides the overall range of subnet IP addresses for the VPC that Account Factory creates for your account. Within your VPC, subnets are assigned automatically from the range you specify, and they are equal in size. By default, subnets within your VPC do not overlap. However, subnet IP address ranges in the VPCs of all your provisioned accounts could overlap.

- Choose a region or all the regions for creating a VPC when an account is provisioned. By default all available regions are selected.
- From the list, choose the number of Availability Zones to configure subnets for in each VPC. The default and recommended number is 3.
- Choose **Save**.

Unmanaging a Member Account

If you created an account in Account Factory that you no longer want to be managed by AWS Control Tower in a landing zone, you can unmanage the account. This can be done in the AWS Service Catalog console by an AWS SSO user in either the **AWSAccountFactory** or **AWSServiceCatalogAdmins** groups. For more information on AWS SSO users or groups, see Managing Users and Access Through AWS Single Sign-On (p. 47). The following procedure describes how to unmanage a member account.

To unmanage a member account

1. Open the AWS Service Catalog console in your web browser at https://console.aws.amazon.com/servicecatalog.
2. From the left navigation pane, choose **Provisioned products list**.
3. From the list of provisioned accounts, choose the name of the account that you want AWS Control Tower to no longer manage.
4. On the **Provisioned product details** page, from the **ACTIONS** menu, choose **Terminate**.
5. From the dialog box that appears, choose **TERMINATE**.

   Important
   The word *terminate* is specific to AWS Service Catalog. When you terminate an Account Factory account in AWS Service Catalog, the account is not closed. This action removes the account from its OU and your landing zone.
6. The **Deregistering Managed Account** message displays.
7. To update the displayed account status, refresh the page. When the account has been unmanaged, its status changes to **terminated**.
8. If you no longer need the terminated account, close it. For information about closing AWS accounts, see Closing an Account in the *AWS Billing and Cost Management User Guide*

   Note
   An unmanaged (terminated) account is not closed or deleted. When the account has been unmanaged, the AWS SSO user that you selected when you created the account in Account Factory still has administrative access to the account. If you do not want this user to have administrative access, you must change this setting in AWS SSO by updating the account in Account Factory and changing the AWS SSO user email address for the account. For more information, see Updating Account Factory Accounts (p. 53).

Closing an Account Created in Account Factory

Accounts created in Account Factory are AWS accounts. For information about closing AWS accounts, see Closing an Account in the *AWS Billing and Cost Management User Guide*.

Considerations for Account Factory

Accounts created through the Account Factory in AWS Control Tower inherit the guardrails of the parent OU. Accounts created outside of AWS Control Tower won't inherit these guardrails. These accounts also do not display in AWS Control Tower.

When an account is provisioned with Account Factory, the following AWS resources are created within the account.

AWS service	Resource type	Resource name
AWS CloudFormation	Stacks	StackSet-AWSControlTowerBP-BASELINE-CLOUDTRAIL-* StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-* StackSet-AWSControlTowerBP-BASELINE-CONFIG-* StackSet-AWSControlTowerBP-BASELINE-ROLES-* StackSet-AWSControlTowerBP-BASELINE-SERVICE-ROLES-*
AWS CloudTrail	Trail	aws-controltower-BaselineCloudTrail
Amazon CloudWatch	CloudWatch Event Rules	aws-controltower-ConfigComplianceChangeEventRule
Amazon CloudWatch	CloudWatch Logs	aws-controltower/CloudTrailLogs /aws/lambda/aws-controltower-NotificationForwarder
AWS Identity and Access Management	Roles	aws-controltower-AdministratorExecutionRole aws-controltower-CloudWatchLogsRole aws-controltower-ConfigRecorderRole aws-controltower-ForwardSnsNotificationRole aws-controltower-ReadOnlyExecutionRole AWSControlTowerExecution
AWS Identity and Access Management	Policies	AWSControlTowerServiceRolePolicy

AWS service	Resource type	Resource name
Amazon Simple Notification Service	Topics	aws-controltower-SecurityNotifications
AWS Lambda	Applications	StackSet-AWSControlTowerBP-BASELINE-CLOUDWATCH-*
AWS Lambda	Functions	aws-controltower-NotificationForwarder

When you enable guardrails with strongly recommended guidance, AWS Control Tower managed AWS resources are created in your accounts. Do not modify or delete resources created by AWS Control Tower. Doing so could result in the guardrails entering an unknown state. For more information, see Guardrail Reference (p. 22).

Detecting and Resolving Drift in AWS Control Tower

When you create your landing zone, the landing zone and all the OUs, accounts, and resources are compliant with all the governance rules enforced by your chosen guardrails. As you and your users use the landing zone, changes in this compliance status may occur. Some changes may be accidental, and some may be made intentionally to respond to time-sensitive operational events.

Regardless of the reason, changes can complicate your compliance story. You can use drift detection to identify resources that need changes or configuration updates to resolve the drift. Resolving drift helps to ensure your compliance with governance regulations, and it is a regular operations task for your master account administrators.

Detecting Drift

Drift is detected automatically by AWS Control Tower. It is surfaced in the Amazon SNS notifications that are aggregated in the audit account. Notifications in each member account send alerts to a local Amazon SNS topic, and to a Lambda function.

Member account administrators can (and as a best practice, they should) subscribe to the drift notifications for specific accounts. The AWS Control Tower console displays banners that indicate to master account administrators when drift has occurred.

Resolving Drift

Although detection is automatic, the steps to resolve drift must be done through the console. Many types of drift can be resolved through the **Settings** page. If the **Repair** button in the **Versions** section of the page is selectable, you can choose **Repair** to repair some types of drift. A banner shows the types of drift that have occurred, and it may list the accounts or organizations that are affected by the drift. If no drift has occurred, the **Repair** button appears greyed-out.

Types of Drift

The types of governance drift that can be detected in AWS Control Tower are as follows:

Topics

Moved Member Account

This kind of drift can occur when a managed account, the audit, or the log archive account is moved from one OU to another OU. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
{
  "Message" : "AWS Control Tower has detected that your managed account 'account-
email@amazon.com (012345678909)' has been moved from organizational unit 'Custom (ou-0123-
eEXAMPLE)' to 'Core (ou-3210-1EXAMPLE)'. For more information, including steps to resolve
 this issue, see 'https://docs.aws.amazon.com/console/controltower/move-account'",
  "MasterAccountId" : "012345678912",
  "OrganizationId" : "o-123EXAMPLE",
  "DriftType" : "AccountMovedBetweenOrganizationalUnits",
  "RemediationStep" : "Update Account Factory Provisioned Product",
  "AccountId" : "012345678909",
  "SourceId" : "012345678909",
  "DestinationId" : "ou-3210-1EXAMPLE"
}
```

Resolutions

When this kind of drift occurs, you can resolve it as follows:

- **Account Factory Provisioned Account** – You can resolve the drift by updating the account in Account Factory. For more information, see Updating Account Factory Accounts (p. 53).
- **Shared account** – You can resolve the drift from moving the audit or log archive account by updating your landing zone. For more information, see Updating Your Landing Zone (p. 75).

Added Member Account

This kind of drift can occur when a managed account is added to a managed OU. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
"{
  ""Message"" : ""AWS Control Tower has detected that the managed account 'account-
email@amazon.com (012345678909)' has been added to organization o-123EXAMPLE. For more
 information, including steps to resolve this issue, see 'https://docs.aws.amazon.com/
console/controltower/add-account'"",
  ""MasterAccountId"" : ""012345678912"",
  ""OrganizationId"" : ""o-123EXAMPLE"",
  ""DriftType"" : ""AccountAddedToOrganization"",
  ""RemediationStep"" : ""Update Account Factory Provisioned Product"",
  ""AccountId"" : ""012345678909""
}"
```

Resolution

When this kind of drift occurs, you can resolve it by updating the account in Account Factory. For more information, see Updating Account Factory Accounts (p. 53).

Removed Member Account

This kind of drift can occur when a managed account is removed from a managed OU. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
"{
  ""Message"" : ""AWS Control Tower has detected that the managed account 012345678909
 has been removed from organization o-123EXAMPLE. For more information, including steps
 to resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/remove-
account'"",
  ""MasterAccountId"" : ""012345678912"",
  ""OrganizationId"" : ""o-123EXAMPLE"",
  ""DriftType"" : ""AccountRemovedFromOrganization"",
  ""RemediationStep"" : ""Add account to Organization and update Account Factory
 provisioned product"",
  ""AccountId"" : ""012345678909""
}"
```

Resolution

When this kind of drift occurs, you can resolve it by updating the account in Account Factory, and adding the account to a managed OU from the Account Factory update wizard. For more information, see Updating Account Factory Accounts (p. 53).

Unplanned Update to Managed SCP

This kind of drift can occur when an SCP for a guardrail is updated in the Organizations console or programmatically using the AWS CLI or one of the AWS SDKs. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
"{
  ""Message"" : ""AWS Control Tower has detected that the managed service control policy
 'aws-guardrails-012345 (p-tEXAMPLE)', attached to the managed organizational unit 'Core
 (ou-0123-1EXAMPLE)', has been modified. For more information, including steps to resolve
 this issue, see 'https://docs.aws.amazon.com/console/controltower/update-scp'"",
  ""MasterAccountId"" : ""012345678912"",
  ""OrganizationId"" : ""o-123EXAMPLE"",
  ""DriftType"" : ""ServiceControlPolicyUpdated"",
  ""RemediationStep"" : ""Update Control Tower Setup"",
  ""OrganizationalUnitId"" : ""ou-0123-1EXAMPLE"",
  ""PolicyId"" : ""p-tEXAMPLE""
}"
```

Resolution

When this kind of drift occurs, you can resolve it by updating your landing zone. For more information, see Updating Your Landing Zone (p. 75).

SCP Attached to Managed OU

This kind of drift can occur when an SCP is attached to an OU outside of the AWS Control Tower console. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
"{
  ""Message"" : ""AWS Control Tower has detected that the managed service control policy
 'aws-guardrails-012345 (p-tEXAMPLE)' has been attached to the managed organizational unit
 'Custom (ou-0123-1EXAMPLE)'. For more information, including steps to resolve this issue,
 see 'https://docs.aws.amazon.com/console/controltower/scp-detached-ou'"",
  ""MasterAccountId"" : ""012345678912"",
  ""OrganizationId"" : ""o-123EXAMPLE"",
  ""DriftType"" : ""ServiceControlPolicyAttachedToOrganizationalUnit"",
  ""RemediationStep"" : ""Update Control Tower Setup"",
  ""OrganizationalUnitId"" : ""ou-0123-1EXAMPLE"",
  ""PolicyId"" : ""p-tEXAMPLE""
}"
```

Resolution

When this kind of drift occurs, you can resolve it by updating your landing zone. For more information, see Updating Your Landing Zone (p. 75).

SCP Detached from Managed OU

This kind of drift can occur when an SCP has been detached from an OU outside of the AWS Control Tower console. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
"{
  ""Message"" : ""AWS Control Tower has detected that the managed service control policy
 'aws-guardrails-012345 (p-tEXAMPLE)' has been detached from the managed organizational
 unit 'Custom (ou-0123-1EXAMPLE)'. For more information, including steps to resolve this
 issue, see 'https://docs.aws.amazon.com/console/controltower/scp-detached'"",
  ""MasterAccountId"" : ""012345678912"",
  ""OrganizationId"" : ""o-123EXAMPLE"",
  ""DriftType"" : ""ServiceControlPolicyDetachedFromOrganizationalUnit"",
  ""RemediationStep"" : ""Update Control Tower Setup"",
  ""OrganizationalUnitId"" : ""ou-0123-1EXAMPLE"",
  ""PolicyId"" : ""p-tEXAMPLE""
}"
```

Resolution

When this kind of drift occurs, you can resolve it by updating your landing zone. For more information, see Updating Your Landing Zone (p. 75).

SCP Attached to Member Account

This kind of drift can occur when an SCP is attached to an account in the Organizations console. Guardrails and their SCPs can be enabled on OUs and all of an OU's member accounts through the AWS Control Tower console. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
"{
  ""Message"" : ""AWS Control Tower has detected that the managed service control policy
 'aws-guardrails-012345 (p-tEXAMPLE)' has been attached to the managed account 'account-
email@amazon.com (012345678909)'. For more information, including steps to resolve this
 issue, see 'https://docs.aws.amazon.com/console/controltower/scp-detached-account'"",
  ""MasterAccountId"" : ""012345678912"",
  ""OrganizationId"" : ""o-123EXAMPLE"",
  ""DriftType"" : ""ServiceControlPolicyAttachedToAccount"",
  ""RemediationStep"" : ""Update Control Tower Setup"",
  ""AccountId"" : ""012345678909"",
  ""PolicyId"" : ""p-tEXAMPLE""
}"
```

Resolution

When this kind of drift occurs, you can resolve it by updating your landing zone. For more information, see Updating Your Landing Zone (p. 75).

Deleted Managed OU

This kind of drift can occur if a managed OU is deleted outside of the AWS Control Tower console. The following is an example of the Amazon SNS notification when this type of drift is detected.

```
"{
  ""Message"" : ""AWS Control Tower has detected that the managed organizational unit
 'Custom (ou-0123-1EXAMPLE)' has been deleted. For more information, including steps to
 resolve this issue, see 'https://docs.aws.amazon.com/console/controltower/delete-ou'"",
  ""MasterAccountId"" : ""012345678912"",
  ""OrganizationId"" : ""o-123EXAMPLE"",
  ""DriftType"" : ""OrganizationalUnitDeleted"",
  ""RemediationStep"" : ""Delete managed organizational unit in Control Tower"",
  ""OrganizationalUnitId"" : ""ou-0123-1EXAMPLE""
}"
```

Resolution

When this kind of drift occurs, a central cloud administrator must sign in to the AWS Control Tower console and delete the managed OU from your list of **Organizational units**.

Security in AWS Control Tower

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Control Tower, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS services that you use. You are also responsible for other factors including the sensitivity of your data, your organization's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Control Tower. The following topics show you how to configure AWS Control Tower to meet your security and compliance objectives. You also learn how to use other AWS services that help you monitor and secure your AWS Control Tower resources.

Data Protection in AWS Control Tower

AWS Control Tower conforms to the AWS shared responsibility model, which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all the AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM), so that each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources.
- Set up API and user activity logging with CloudTrail. This is handled automatically in AWS Control Tower when you set up your landing zone.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields such as a **Name** field. This includes when you work with AWS Control Tower or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into AWS Control Tower or other services might get picked up for inclusion in diagnostic logs.

When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

For more information about data protection, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*. AWS Control Tower provides the following options that you can use to help secure the content that exists in your landing zone:

Topics

Encryption at Rest

AWS Control Tower uses Amazon S3 buckets and Amazon DynamoDB databases that are encrypted at rest by using Amazon S3-Managed Keys (SSE-S3) in support of your landing zone. This encryption is configured by default when you set up your landing zone. You can also establish encryption at rest for the services you use in your landing zone for the services that support it. For more information, see the security chapter of that service's online documentation.

Encryption in Transit

AWS Control Tower uses Transport Layer Security (TLS) and client-side encryption for encryption in transit in support of your landing zone. In addition, accessing AWS Control Tower requires using the console, which can only be accessed through an HTTPS endpoint. This encryption is configured by default when you set up your landing zone.

Restrict Access to Content

As a best practice, you should restrict access to the appropriate subset of users. With AWS Control Tower, you can do this by ensuring your central cloud administrators and end users have the right IAM permissions or, in the case of AWS SSO users, are in the correct groups.

- For more information about roles and policies for IAM entities, see *IAM User Guide*.
- For more information about the AWS SSO groups that are created when you set up your landing zone, see AWS SSO Groups for AWS Control Tower (p. 48).

Identity and Access Management in AWS Control Tower

To perform any operation in your landing zone, such as provisioning accounts in Account Factory or creating new organizational units (OUs) in the AWS Control Tower console, either AWS Identity and Access Management (IAM) or AWS Single Sign-On (AWS SSO) require that you to authenticate that you're an approved AWS user. For example, if you're using the AWS Control Tower console, you authenticate your identity by providing your AWS user name and a password.

After you authenticate your identity, IAM controls your access to AWS with a defined set of permissions on a specific set of operations and resources. If you are an account administrator, you can use IAM to control the access of other IAM users to the resources that are associated with your account.

Topics

Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with an identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.
- **IAM user** – An IAM user is an identity within your AWS account that has specific custom permissions. You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

  In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. AWS Control Tower supports Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.
- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user in that it is an AWS identity with permissions policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session. IAM roles with temporary credentials are useful in the following situations:
  - **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the *IAM User Guide*.
  - **AWS service access** – A service role is an IAM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within IAM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the *IAM User Guide*.
  - **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an Amazon EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the Amazon EC2 instance. To assign an AWS role to an Amazon EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the Amazon EC2 instance to get temporary credentials. For more

information, see Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances in the *IAM User Guide*.

- **AWS SSO user** Authentication to the AWS SSO user portal is controlled by the directory that you have connected to AWS SSO. However, authorization to the AWS accounts that are available to end users from within the user portal is determined by two factors:
  - Who has been assigned access to those AWS accounts in the AWS SSO console. For more information, see Single Sign-On Access in the *AWS Single Sign-On User Guide*.
  - What level of permissions have been granted to the end users in the AWS SSO console to allow them the appropriate access to those AWS accounts. For more information, see Permission Sets in the *AWS Single Sign-On User Guide*.

Access Control

To create, update, delete, or list AWS Control Tower resources, or other AWS resources in your landing zone you need permissions to perform the operation, and you need permissions to access the corresponding resources. In addition, to perform the operation programmatically, you need valid access keys.

The following sections describe how to manage permissions for AWS Control Tower:

Topics

Overview of Managing Access Permissions to Your AWS Control Tower Resources

Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles), and some services (such as AWS Lambda) also support attaching permissions policies to resources.

> **Note**
> An *account administrator* (or administrator) is a user with administrator privileges. For more information, see IAM Best Practices in the *IAM User Guide*.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific actions that you want to allow on those resources.

Topics

AWS Control Tower Resources and Operations

In AWS Control Tower, the primary resource is a *landing zone*. AWS Control Tower also supports an additional resource type, *guardrails*. However, for AWS Control Tower, you can manage guardrails only in the context of an existing landing zone. Guardrails are referred to as a *subresource*.

Understanding Resource Ownership

The AWS account owns the resources that are created in the account, regardless of who created the resources. Specifically, the resource owner is the AWS account of the principal entity (that is, the AWS account root user, an IAM user, or an IAM role) that authenticates the resource creation request. The following examples illustrate how this works:

- If you use the AWS account root user credentials of your AWS account to set up a landing zone, your AWS account is the owner of the resource.
- If you create an IAM user in your AWS account and grant permissions to set up a landing zone to that user, the user can set up a landing zone as long as their account meets the prerequisites. However, your AWS account, to which the user belongs, owns the landing zone resource.
- If you create an IAM role in your AWS account with permissions to set up a landing zone, anyone who can assume the role can set up a landing zone. Your AWS account, to which the role belongs, owns the landing zone resource.

Managing Access to Resources

A *permissions policy* describes who has access to what. The following section explains the available options for creating permissions policies.

> **Note**
> This section discusses using IAM in the context of AWS Control Tower. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

Policies attached to an IAM identity are referred to as *identity-based* policies (IAM polices). Policies attached to a resource are referred to as *resource-based* policies. AWS Control Tower supports only identity-based policies (IAM policies).

Topics

Identity-Based Policies (IAM Policies)

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – To grant a user permissions to create an AWS Control Tower resource, such as setting up a landing zone, you can attach a permissions policy to a user or group that the user belongs to.
- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in Account A can create a role to grant cross-account permissions to another AWS account (for example, Account B) or an AWS service as follows:
  1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in Account A.
  2. Account A administrator attaches a trust policy to the role identifying Account B as the principal who can assume the role.
  3. Account B administrator can then delegate permissions to assume the role to any users in Account B. Doing this allows users in Account B to create or access resources in Account A. The principal in the trust policy can also be an AWS service principal if you want to grant an AWS service permissions to assume the role.

For more information about using IAM to delegate permissions, see Access Management in the *IAM User Guide*.

The following is an example policy that allows a user to set up a landing zone in your AWS account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

For more information about using identity-based policies with AWS Control Tower, see Using Identity-Based Policies (IAM Policies) for AWS Control Tower (p. 69). For more information about users, groups, roles, and permissions, see Identities (Users, Groups, and Roles) in the *IAM User Guide*.

Resource-Based Policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket.

Specifying Policy Elements: Actions, Effects, and Principals

Currently, AWS Control Tower doesn't have an API. You can set up and manage your landing zone through the AWS Control Tower console. To set up your landing zone, you must be an IAM user with administrative permissions as defined in a IAM policy.

The following are the most basic policy elements:

- **Resource** – In a policy, you use an Amazon Resource Name (ARN) to identify the resource to which the policy applies. For more information, see AWS Control Tower Resources and Operations (p. 66).
- **Action** – You use action keywords to identify resource operations that you want to allow or deny.
- **Effect** – You specify the effect when the user requests the specific action—this can be either allow or deny. If you don't explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). AWS Control Tower doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

Specifying Conditions in a Policy

When you grant permissions, you can use the IAM policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see Condition in the *IAM User Guide*.

To express conditions, you use predefined condition keys. There are no condition keys specific to AWS Control Tower. However, there are AWS-wide condition keys that you can use as appropriate. For a complete list of AWS-wide keys, see Available Keys for Conditions in the *IAM User Guide*.

Using Identity-Based Policies (IAM Policies) for AWS Control Tower

This topic provides examples of identity-based policies that demonstrate how an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles) and thereby grant permissions to perform operations on AWS Control Tower resources.

> **Important**
> We recommend that you first review the introductory topics that explain the basic concepts and options available for you to manage access to your AWS Control Tower resources. For more information, see Overview of Managing Access Permissions to Your AWS Control Tower Resources (p. 66).

The following shows an example of a permissions policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

The policy has one statement that grants permissions for all AWS actions on all resources in the account. This is the permissions policy for administrator access in an AWS account. This is the necessary level of permissions for an IAM entity that will set up a landing zone.

The policy doesn't specify the `Principal` element because in an identity-based policy you don't specify the principal who gets the permission. When you attach policy to a user, the user is the implicit principal. When you attach a permissions policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

Permissions Required to Use the AWS Control Tower Console

AWS Control Tower requires creation of three roles to set up a landing zone. AWS Control Tower splits permissions into three roles as a best practice to restrict access to the minimal sets of actions and resources.

AWSControlTowerAdmin

This role provides AWS Control Tower with access to infrastructure critical to maintaining the landing zone. Inline Policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeAvailabilityZones",
            "Resource": "*"
        }
    ]
}
```

AWSControlTowerServiceRolePolicy

AWS CloudFormation assumes this role to deploy stack sets in accounts created by AWS Control Tower. Inline Policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/AWSControlTowerExecution"
            ],
            "Effect": "Allow"
        }
    ]
}
```

AWSControlTowerCloudTrailRole

AWS Control Tower enables CloudTrail as a best practice and provides this role to CloudTrail. CloudTrail assumes this role to create and publish CloudTrail logs. Inline Policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "logs:CreateLogStream",
            "Resource": "arn:aws:logs:*:*:log-group:aws-controltower/CloudTrailLogs:*",
            "Effect": "Allow"
        },
        {
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:*:*:log-group:aws-controltower/CloudTrailLogs:*",
            "Effect": "Allow"
        }
    ]
}
```

Logging and Monitoring in AWS Control Tower

Monitoring is an important part of the well-architected nature of AWS Control Tower. When you set up your landing zone, cross-account monitoring is set up as well. Of the shared accounts created, one is the log archive account, dedicated to collecting all logs centrally, including all of your other accounts. The health and status of your guardrails are monitored constantly. You can see their status at a glance in the AWS Control Tower console. This is also true for the health and status of the accounts you provisioned in Account Factory.

You should collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your resources and activity in your landing zone, allowing you to plan for and respond to potential incidents.

Logging of actions and events in AWS Control Tower is accomplished automatically through its integration with CloudWatch.

The Activities Page

The **Activities** page provides an overview of AWS Control Tower master account actions. To navigate to the AWS Control Tower **Activities** page, select **Activities** from the left navigation.

The **Activities** page shows all AWS Control Tower actions initiated from the master account. It includes actions that are logged automatically when you navigate through the AWS Control Tower console. Here are the fields that the **Activities** page shows you:

- Date and time: The timestamp for the activity.
- User: The person or account that initiated the activity.
- Action: The activity that occurred.
- Resources: The resources affected by the activity.
- Status: Success, failure, or other state of the activity.
- Description: More details about the activity.

The activities shown in the **Activities** page are the same ones reported in the AWS CloudTrail events log for AWS Control Tower, but they're shown in a table format. To learn more about a specific activity, select the activity from the table and then choose **View details**.

The following sections describe monitoring and logging in AWS Control Tower with more detail:

Topics

Monitoring

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Control Tower and your other AWS solutions. AWS provides the following monitoring tools to watch AWS Control Tower, report when something is wrong, and take automatic actions when appropriate:

- *Amazon CloudWatch* monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the Amazon CloudWatch User Guide.
- *Amazon CloudWatch Events* delivers a near real-time stream of system events that describe changes in AWS resources. CloudWatch Events enables automated event-driven computing, as you can write rules that watch for certain events and trigger automated actions in other AWS services when these events happen. For more information, see the Amazon CloudWatch Events User Guide.
- *Amazon CloudWatch Logs* enables you to monitor, store, and access your log files from Amazon EC2 instances, CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the Amazon CloudWatch Logs User Guide.
- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred.

For more information, see Logging AWS Control Tower Actions with AWS CloudTrail (p. 72).

Logging AWS Control Tower Actions with AWS CloudTrail

AWS Control Tower is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Control Tower. CloudTrail captures actions for AWS Control Tower as events. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Control Tower. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS Control Tower, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and enable it, see the AWS CloudTrail User Guide.

AWS Control Tower Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in AWS Control Tower, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for AWS Control Tower, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

AWS Control Tower logs the following actions as events in CloudTrail log files:

- `SetupLandingZone`
- `UpdateAccountFactoryConfig`
- `ManageOrganizationalUnit`
- `CreateManagedAccount`
- `EnableGuardrail`
- `GetLandingZoneStatus`
- `GetHomeRegion`
- `ListManagedAccounts`
- `DescribeManagedAccount`
- `DescribeAccountFactoryConfig`
- `DescribeGuardrailForTarget`
- `DescribeManagedOrganizationalUnit`

- `ListEnabledGuardrails`
- `ListGuardrailViolations`
- `ListGuardrails`
- `ListGuardrailsForTarget`
- `ListManagedAccountsForGuardrail`
- `ListManagedAccountsForParent`
- `ListManagedOrganizationalUnits`
- `ListManagedOrganizationalUnitsForGuardrail`
- `GetGuardrailComplianceStatus`
- `DescribeGuardrail`
- `ListDirectoryGroups`
- `DescribeSingleSignOn`
- `DescribeCoreService`
- `GetAvailableUpdates`

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Example: AWS Control Tower Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail events don't appear in any specific order in the log files.

The following example shows a CloudTrail log entry that shows the structure of a typical log file entry for a `SetupLandingZone` AWS Control Tower event, including a record of the identity of the user who initiated the action.

```
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:backend-test-assume-role-session",
    "arn": "arn:aws:sts::76543EXAMPLE;:assumed-role/AWSControlTowerTestAdmin/backend-test-
assume-role-session",
    "accountId": "76543EXAMPLE",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-11-20T19:36:11Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::AKIAIOSFODNN7EXAMPLE:role/AWSControlTowerTestAdmin",
        "accountId": "AIDACKCEVSQ6C2EXAMPLE",
        "userName": "AWSControlTowerTestAdmin"
      }
    }
  },
  "eventTime": "2018-11-20T19:36:15Z",
  "eventSource": "controltower.amazonaws.com",
  "eventName": "SetupLandingZone",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "Coral/Netty4",
  "errorCode": "InvalidParametersException",
  "errorMessage": "Home region EU_CENTRAL_1 is unsupported",
  "requestParameters": {
    "homeRegion": "EU_CENTRAL_1",
    "logAccountEmail": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "sharedServiceAccountEmail": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "securityAccountEmail": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "securityNotificationEmail": "HIDDEN_DUE_TO_SECURITY_REASONS"
  },
  "responseElements": null,
  "requestID": "96f47b68-ed5f-4268-931c-807cd1f89a96",
  "eventID": "4ef5cf08-39e5-4fdf-9ea2-b07ced506851",
  "eventType": "AwsApiCall",
  "recipientAccountId": "76543EXAMPLE"
}
```

Compliance Validation for AWS Control Tower

AWS Control Tower is a well-architected service that can help your organization meet your compliance needs with guardrails and best practices. Additionally, third-party auditors assess the security and compliance of a number of the services you can use in your landing zone as a part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact in the *AWS Artifact User Guide*.

Your compliance responsibility when using AWS Control Tower is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Config – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in AWS Control Tower

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability Zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

AWS Control Tower is available in four AWS Regions (US East (N. Virginia), US East (Ohio), US West (Oregon), and Europe (Ireland)) with a home Region defined as the one your landing zone was set up within.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

Infrastructure Security in AWS Control Tower

AWS Control Tower is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access AWS services and resources within your landing zone through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Configuration Update Management in AWS Control Tower

It is the responsibility of the members of central cloud administrators to keep your landing zone updated. Updating your landing zone ensures that AWS Control Tower is patched and updated. In addition, to protect your landing zone from potential compliance issues, the members of the central cloud administrator team should resolve drift issues as soon as they're detected and reported.

> **Note**
> The AWS Control Tower console indicates when your landing zone needs to be updated. If you don't see an option to update, your landing zone is already up to date.

Updating Your Landing Zone

The easiest way to update your AWS Control Tower landing zone is through the **Settings** page. Navigate to the **Settings** page by choosing **Settings** in the left navigation.

The **Settings** page shows you the current version of your landing zone, and it lists any updated versions that may be available. You can choose the **Update** button if you need to update your version. If the **Update** button appears greyed-out, you do not need to update.

Alternatively, you can update your landing zone manually. The update takes approximately the same amount of time, whether you use the **Update** button or the manual process.

The following procedure walks you through the steps of updating your landing zone manually.

To update your landing zone

1. Open a web browser, and navigate to the AWS Control Tower console at https://us-west-2.console.aws.amazon.com/controltower/home/update.
2. Review the information in the wizard and choose **Update**. This updates the backend of the landing zone as well as your shared accounts. This process can take a little more than an hour.
3. Update your member accounts. From the navigation pane, choose **Accounts**.
4. Choose **Provision new account** to open the AWS Service Catalog console and the Account Factory product.
5. From the navigation pane, choose **Provisioned products list**.
6. For each account listed, perform the following steps to update all your member accounts:

   a. From the menu for the account, choose **Provisioned product details**.
   b. Make a note of the following parameters:

      - SSOUserEmail
      - AccountEmail
      - SSOUserFirstName
      - SSOUSerLastName
      - AccountName

   c. From **ACTIONS**, choose **Update**.
   d. Choose the radio button next to the **Version** of the product you want to update, and choose **NEXT**.
   e. Provide the parameter values that were mentioned previously. For **ManagedOrganizationalUnit** choose the OU that the account is in. You can find this information in the AWS Control Tower console, under **Accounts**.
   f. Choose **NEXT**.
   g. Review your changes, and then choose **UPDATE**. This process can take a few minutes per account.

Resolve Drift

When you create your AWS Control Tower landing zone, the landing zone and all the OUs, accounts, and resources are compliant with all of the governance rules enforced by your guardrails, whether mandatory or elective. As you and your organization members use the landing zone, changes in compliance status may occur. Some changes may be accidental, and some may be made intentionally to respond to time-sensitive operational events. Regardless, changes can complicate your compliance story.

Resolving drift helps to ensure your organization's compliance with governance regulations. Drift resolution is a regular operations task for your master account administrators.

Drift detection is automatic in AWS Control Tower. It helps you identify resources that need changes or configuration updates that must be made to resolve the drift.

To repair most types of drift, choose **Repair** on the **Settings** page. The **Repair** button becomes selectable when drift has occurred. For more information, see Detecting and Resolving Drift in AWS Control Tower (p. 58).

Quotas in AWS Control Tower

This chapter covers the AWS service quotas that you should keep in mind as you use AWS Control Tower. If you're unable to set up your landing zone due to a service quota issue, contact AWS Support.

Quotas for Integrated Services

Each AWS service has its own quotas and limits. You can find the quotas for each service in its documentation. For more information, see the related links:

- **AWS CloudFormation** – AWS CloudFormation Quotas
- **AWS CloudTrail** – Quotas in AWS CloudTrail
- **Amazon CloudWatch** – CloudWatch Quotas
- **AWS Config** – AWS Config Quotas
- **AWS Identity and Access Management** – Quotas for IAM Entities and Objects
- **AWS Lambda** – AWS Lambda Quotas
- **AWS Organizations** – Quotas for AWS Organizations
- **Amazon Simple Storage Service** – Bucket Restrictions and Quotas
- **AWS Service Catalog** – AWS Service Catalog Default Service Quotas
- **AWS Single Sign-On** – Quotas in AWS SSO
- **Amazon Simple Notification Service** – Amazon Simple Notification Service (Amazon SNS) Quotas
- **AWS Step Functions** – Quotas

Walkthroughs

This chapter contains walkthrough procedures that can help you in your use of AWS Control Tower.

Topics

Walkthrough: Cleaning up AWS Control Tower Managed Resources

When your landing zone was set up, AWS Control Tower provisioned resources and services in your landing zone on your behalf. For example, an AWS Organizations organization with multiple accounts and organizational units (OUs) were provisioned. Additionally, guardrails were deployed in your accounts using AWS CloudFormation stacks, stack sets, and AWS Organizations policies.

After going through the getting started procedure, or if you're no longer evaluating AWS Control Tower for your enterprise, you may want to clean up the resources created when you set up your landing zone. The following procedures guide you through cleaning up these resources for lifecycle purposes.

Before performing these procedures, unless it's otherwise indicated, you must be signed in to the AWS Management Console in the home Region for your landing zone, and you must be signed in as an IAM user with administrative permissions for the master account that contains your landing zone.

> **Warning**
> These are destructive actions that can introduce governance drift into your AWS Control Tower setup. We strongly recommend that you only perform these procedures if you intend to stop using your landing zone.

Topics

Delete SCPs

AWS Control Tower uses service control policies (SCPs) for its guardrails. This procedure walks through how to delete the SCPs specifically related to AWS Control Tower.

To delete AWS Organizations SCPs

1. Open the Organizations console at https://console.aws.amazon.com/organizations/.
2. Open the **Policies** tab, and find the Service Control Policies (SCPs) that have the prefix **aws-guardrails-** and do the following for each SCP:

   a. Detach the SCP from the associated OU.

   b. Delete the SCP.

Delete StackSets and Stacks

AWS Control Tower uses StackSets and stacks to deploy AWS Config Rules related to guardrails in your landing zone. The following procedures walk through how to delete these specific resources.

To delete AWS CloudFormation StackSets

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation.
2. From the left navigation menu, choose **StackSets**.
3. For each StackSet with the prefix **AWSControlTower**, do the following. If you have many accounts in a StackSet, this can take some time.

   a. Choose the specific StackSet from the table in the dashboard. This opens the properties page for that StackSet.
   b. At the bottom of the page, in the **Stacks** table, make a record of the AWS account IDs for all the accounts in the table. Copy the list of all accounts.
   c. Choose **Manage StackSet** to open the management wizard.
   d. From **Select action**, choose **Delete stacks**, and choose **Next**.
   e. On **Set deployment options**, from **Specify accounts**, choose **Delete stacks from account**.
   f. In the text field, enter the AWS account IDs you made a record of in step 3.b, separated by commas. For example: *123456789012*, *098765431098*, and so on.
   g. From **Specify regions**, choose **Add all**, leave the rest of the parameters on the page set to their defaults, and choose **Next**.
   h. On the **Review** page, review your choices, and then choose **Delete stacks**.
   i. On the **StackSet properties** page, you can begin this procedure again for your other StackSets.
4. The process is complete when the records in the **Stacks** table of the different **StackSets properties** pages are empty.
5. When the records in the **Stacks** table are empty, choose **Delete StackSet**.

To delete AWS CloudFormation stacks

1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation.
2. From the **Stacks** dashboard, search for all of the stacks with the prefix **AWSControlTower**.
3. For each stack in the table, do the following:

   a. Choose the check box next to the name of the stack.
   b. From the **Actions** menu, choose **Delete Stack**.
   c. In the dialog box that opens, review the information to make sure it's accurate, and choose **Yes, Delete**.

Delete Amazon S3 Buckets in the Log Archive Account

The following procedures guide you through how to sign in to the log archive account as an AWS SSO user in the **AWSControlTowerExecution** group and then delete the Amazon S3 buckets in your log archive account.

To sign in to your log archive account with the right permissions

1. Open the Organizations console at https://console.aws.amazon.com/organizations/.
2. From the **Accounts** tab, find the **Log archive** account.
3. From the right pane that opens, make a record of the log archive account number.
4. From the navigation bar, choose your account name to open your account menu.
5. Choose **Switch Role**.
6. On the page that opens, provide the account number for the log archive account in **Account**.
7. For **Role**, enter **AWSControlTowerExecution**.
8. The **Display Name** populates with text.
9. Choose your favorite **Color**.
10. Choose **Switch Role**.

To delete Amazon S3 buckets

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Search for bucket names that contain **aws-controltower**.
3. For each bucket in the table, do the following:

   a. Choose the check box for the bucket in the table.

   b. Choose **Delete**.

   c. In the dialog box that opens, review the information to make sure it's accurate, enter the name of the bucket to confirm, and then choose **Confirm**.

Clean Up Account Factory

The following procedure guides you through how to sign in as an AWS SSO user in the **AWSServiceCatalogAdmins** group and then clean up your Account Factory accounts.

To sign in to your master account with the right permissions

1. Go to your user portal URL at *`directory-id`*.awsapps.com/start
2. From **AWS Account**, find the **Master** account.
3. From **AWSServiceCatalogAdminFullAccess**, choose **Management console** to sign in to the AWS Management Console as this role.

To clean up Account Factory

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/.
2. From the left navigation menu, choose **Portfolios list**.
3. In the **Local Portfolios** table, search for a portfolio named **AWS Control Tower Account Factory Portfolio**.
4. Choose the name of that portfolio to go to its details page.
5. Expand the **Constraints** section of the page, and choose the radio button for the constraint with the product name **AWS Control Tower Account Factory**.
6. Choose **REMOVE CONSTRAINTS**.
7. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.

8. From the **Products** section of the page, choose the radio button for the product named **AWS Control Tower Account Factory**.
9. Choose **REMOVE PRODUCT**.
10. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.
11. Expand the **Users, Groups, and Roles** section of the page, and choose the check boxes for all the records in this table.
12. Choose **REMOVE USERS, GROUP OR ROLE**.
13. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.
14. From the left navigation menu, choose **Portfolios list**.
15. In the **Local Portfolios** table, search for a portfolio named **AWS Control Tower Account Factory Portfolio**.
16. Choose the radio button for that portfolio, and then choose **DELETE PORTFOLIO**.
17. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.
18. From the left navigation menu, choose **Product list**.
19. On the **Admin products** page, search for the product named **AWS Control Tower Account Factory**.
20. Choose the product to open the **Admin product details** page.
21. From **Actions**, choose **Delete product**.
22. In the dialog box that opens, review the information to make sure it's accurate, and then choose **CONTINUE**.

Clean Up Roles and Polices

These procedures walk you through how to clean up the roles and policies that were created when your landing zone was set up.

To delete the AWS SSO AWSServiceCatalogEndUserAccess role

1. Open the AWS Single Sign-On console at https://console.aws.amazon.com/singlesignon/.
2. Change your AWS Region to US East (N. Virginia).
3. From the left navigation menu, choose **AWS accounts**.
4. Choose your master account link.
5. Choose the dropdown for **Permission sets**, select **AWSServiceCatalogEndUserAccess**, and then choose **Remove**.
6. Choose **AWS accounts** from the left panel.
7. Open the **Permission sets** tab.
8. Select **AWSServiceCatalogEndUserAccess** and delete it.

To delete IAM roles

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. From the left navigation menu, choose **Roles**.
3. From the table, search for roles with the name **AWSControlTower**.
4. For each role in the table, do the following:

    a. Choose the check box for the role.

    b. Choose **Delete role**.

   c. In the dialog box that opens, review the information to make sure it's accurate, and then choose **Yes, delete**.

To delete IAM policies

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. From the left navigation menu, choose **Policies**.
3. From the table, search for policies with the name **AWSControlTower**.
4. For each policy in the table, do the following:

   a. Choose the check box for the policy.
   b. Choose **Policy actions**, and **Delete** from the dropdown menu.
   c. In the dialog box that opens, review the information to make sure it's accurate, and then choose **Delete**.

AWS Control Tower Clean Up Help

If you encounter any issues that you can't resolve during this clean up process, contact AWS Support.

Walkthrough: Configuring AWS Control Tower Without a VPC

This topic walks through how to configure your AWS Control Tower accounts without a VPC.

If your workload does not require a VPC, you can do the following:

- You can delete the AWS Control Tower master account virtual private cloud (VPC). This VPC was created when you set up your landing zone.
- You can change your Account Factory settings so that new AWS Control Tower accounts are created without an associated VPC.

Delete the AWS Control Tower Master Account VPC

Outside of AWS Control Tower, every AWS customer has a default VPC, which you can view on the Amazon Virtual Private Cloud (Amazon VPC) console at https://console.aws.amazon.com/vpc/. You’ll recognize the default VPC, because its name always includes the word *(default)* at the end of the name.

When you set up a AWS Control Tower landing zone, AWS Control Tower deletes your AWS default VPC and creates a new AWS Control Tower default VPC. The new VPC is associated with your AWS Control Tower master account. This topic refers to that new VPC as the *Control Tower master account VPC*.

When you view your AWS Control Tower master account VPC in the Amazon VPC console, you will *not* see the word *(default)* at the end of the name. If you have more than one VPC, you must use the assigned CIDR range to identify the correct AWS Control Tower master account VPC.

You can delete the AWS Control Tower master account VPC, but if you later need a VPC in AWS Control Tower, you must create it yourself.

To delete the AWS Control Tower master account VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

2. Search for `VPC` or select **VPC** from the AWS Service Catalog options. You then see the **VPC Dashboard**.
3. From the menu on the left, choose **Your VPCs**. You then see a list of all your VPCs.
4. Identify the AWS Control Tower master account VPC by its CIDR range.
5. To delete the VPC, choose **Actions** and then choose **Delete VPC**.

An AWS *(default)* VPC already exists in every region for the AWS Control Tower master account. To follow security best practices, if you choose to delete the AWS Control Tower master account VPC, it's best also to delete the AWS default VPC associated with the master account from all AWS Regions. Therefore, to secure the master account, remove the default VPC from each Region, as well as removing the VPC created by Control Tower in your AWS Control Tower home region.

Create an Account in AWS Control Tower Without a VPC

If your end user workloads do not require VPCs, you can use this method to set up user accounts that don't have VPCs created for them automatically.

From the AWS Control Tower dashboard, you can view and edit your network configurations settings. After you change the settings so that AWS Control Tower accounts are created without an associated VPC, all new accounts are created without a VPC until you change the settings again.

To configure Account Factory for creating accounts without VPCs

1. Open a web browser, and navigate to the AWS Control Tower console at https://console.aws.amazon.com/controltower.
2. Choose **Account Factory** from the menu on the left.
3. You then see the Account Factory page with the **Network Configuration** section.
4. Note the current settings if you intend to restore them later.
5. Choose the **Edit** button in the **Network Configuration** section.
6. In the **Edit account factory network configuration** page, go to the **VPC Configuration options for new accounts** section.

    a. Turn off the **Internet-accessible subnet** toggle switch.

    b. Set the **Maximum number of private subnets** value to 0.

    c. Change the **Address range (CIDR) restriction for account VPCs** value to `10.0.0.0/16`

    d. Clear every checkbox in the **Regions for VPC creation** column.
7. Choose **Save**.

Possible Errors

Be aware of these possible errors that could occur when you delete your AWS Control Tower master account VPC or reconfigure Account Factory to create accounts without VPCs.

- Your existing master account may have dependencies or resources in the AWS Control Tower master account VPC, which can cause a *deletion failure* error.
- If you leave the default CIDR in place when setting up to launch new accounts without a VPC, your request fails with an error that *the CIDR is not valid*.

Troubleshooting

If you encounter issues while using AWS Control Tower, you can use the following information to resolve them according to our best practices. If the issues you encounter are outside the scope of the following information, or if they persist after you've tried to resolve them, contact AWS Support.

Landing Zone Launch Failed

If your master account is less than an hour old, you may encounter issues when the additional accounts are created.

Action to take

If you encounter this issue, check your email. You might have been sent confirmation email that is awaiting response. Alternatively, we recommend that you wait an hour, and then try again. If the issue persists, contact AWS Support.

Don't Change Email Addresses Outside of AWS Control Tower

The email addresses for your shared service accounts (the master account, the auditing account, and the log archive account) should never be changed. If you've changed one of these email addresses, contact AWS Support.

The email addresses for your member accounts created in Account Factory can be changed, but only by updating the account in Account Factory. For more information, see Updating Account Factory Accounts (p. 53).

Don't Migrate Your Account's Organizational Unit Outside of AWS Control Tower

To migrate an account's organizational unit in AWS Control Tower, use the instructions for updating an account in Account Factory. In step 4(e), choose the name of the new Organizational Unit for the account, instead of the name of the current Organizational Unit.

For more information, see Updating Account Factory Accounts (p. 53).

Received an Insufficient Permissions Error

It's possible that your account may not have the necessary permissions to perform certain work in certain AWS Organizations. If you encounter the following type of error, check all the permissions areas, such as IAM or SSO permissions, to make sure your permission is not being denied from those places:

"You have insufficient permissions to perform AWS Organizations API actions."

If you believe your work requires the action you're attempting, and you can't locate any relevant restriction, contact your system administrator or AWS Support.

AWS Support

If you want to move your existing member accounts into a different support plan, you can sign in to each account with root account credentials, compare plans, and set the support level that you prefer.

We recommend that you update the MFA and account security contacts when you make changes to your support plan.

Document History

- **Latest documentation update:** November 26, 2019

The following table describes important changes to the *AWS Control Tower User Guide*. For notifications about documentation updates, you can subscribe to the RSS feed.

update-history-change	update-history-description	update-history-date
Settings and Activities pages are available for AWS Control Tower (p. 86)	The Settings and Activities pages make it easier to update your landing zone and to view logged events.	November 26, 2019
Additional preventive guardrails are available for AWS Control Tower (p. 86)	Preventive guardrails in AWS Control Tower keep your organization and resources aligned with your environment.	September 6, 2019
Additional detective guardrails are available for AWS Control Tower (p. 86)	Detective guardrails in AWS Control Tower give information about the state of your organization and resources.	August 27, 2019
AWS Control Tower is now generally available (p. 86)	AWS Control Tower is a service that offers the easiest way to set up and govern your multi-account AWS environment at scale.	June 24, 2019

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.