AWS Management Tools

AWS Config

General

AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance.

With AWS Config you can discover existing AWS resources, export a complete inventory of your AWS resources with all configuration details, and determine how a resource was configured at any point in time.

These capabilities enable compliance auditing, security analysis, resource change tracking, and troubleshooting.

Allow you to assess, audit and evaluate configurations of your AWS resources.

Very useful for Configuration Management as part of an ITIL program.

Creates a baseline of various configuration settings and files and can then track variations against that baseline.

AWS Config vs CloudTrail

AWS CloudTrail records user API activity on your account and allows you to access information about this activity.

AWS Config records point-in-time configuration details for your AWS resources as Configuration Items (CIs).

You can use an AWS Config CI to answer “What did my AWS resource look like?” at a point in time.

You can use AWS CloudTrail to answer “Who made an API call to modify this resource?”.

Config Rules

A Config Rule represents desired configurations for a resource and is evaluated against configuration changes on the relevant resources, as recorded by AWS Config.

AWS Config Rules can check resources for certain desired conditions and if violations are found the resources are flagged as “noncompliant”.

Examples of Config Rules: .

* Is backup enabled on RDS?
* Is CloudTrail enabled on the AWS account?
* Are EBS volumes encrypted.

Configuration Items

A Configuration Item (CI) is the configuration of a resource at a given point-in-time. A CI consists of 5 sections:

1. Basic information about the resource that is common across different resource types (e.g., Amazon Resource Names, tags).
2. Configuration data specific to the resource (e.g., EC2 instance type).
3. Map of relationships with other resources (e.g., EC2::Volume vol-3434df43 is “attached to instance” EC2 Instance i-3432ee3a).
4. AWS CloudTrail event IDs that are related to this state.
5. Metadata that helps you identify information about the CI, such as the version of this CI, and when this CI was captured.

Charges

With AWS Config, you are charged based on the number configuration items (CIs) recorded for supported resources in your AWS account.

AWS Config creates a configuration item whenever it detects a change to a resource type that it is recording.

# AWS Organizations

AWS Organizations helps you centrally govern your environment as you grow and scale your workloads on AWS.

Organizations helps you to centrally manage billing; control access, compliance, and security; and share resources across your AWS accounts.

Using AWS Organizations, you can automate account creation, create groups of accounts to reflect your business needs, and apply policies for these groups for governance.

You can also simplify billing by setting up a single payment method for all of your AWS accounts.

Through integrations with other AWS services, you can use Organizations to define central configurations and resource sharing across accounts in your organization.

AWS Organizations is available to all AWS customers at no additional charge.

The [**AWS Organizations API**](https://docs.aws.amazon.com/organizations/latest/APIReference/Welcome.html) enables automation for account creation and management.

Available in two feature sets:

* Consolidated billing.
* All features.

By default, organizations support consolidated billing features.

Consolidated billing separates paying accounts and linked accounts.

You can use AWS Organizations to set up a single payment method for all the AWS accounts in your organization through consolidated billing.

With consolidated billing, you can see a combined view of charges incurred by all your accounts.

Can also take advantage of pricing benefits from aggregated usage, such as volume discounts for Amazon EC2 and Amazon S3.

Limit of 20 linked accounts for consolidated billing (default).

Policies can be assigned at different points in the hierarchy.

Can help with cost control through volume discounts.

Unused reserved EC2 instances are applied across the group.

Paying accounts should be used for billing purposes only.

Billing alerts can be setup at the paying account which shows billing for all linked accounts.

#### **Core concepts**

Some of the core concepts you need to understand are listed here:

**AWS Organization –**An organization is a collection of AWS accounts that you can organize into a hierarchy and manage centrally.

**AWS Account –**An AWS account is a container for your AWS resources.

**Master Account –**A master account is the AWS account you use to create your organization.

**Member Account –**A member account is an AWS account, other than the master account, that is part of an organization.

**Administrative Root –**An administrative root is the starting point for organizing your AWS accounts. The administrative root is the top-most container in your organization’s hierarchy.

**Organizational Unit (OU) –**An organizational unit (OU) is a group of AWS accounts within an organization. An OU can also contain other OUs enabling you to create a hierarchy.

**Policy –**A policy is a “document” with one or more statements that define the controls that you want to apply to a group of AWS accounts. AWS Organizations supports a specific type of policy called a Service Control Policy (SCP). An SCP defines the AWS service actions, such as Amazon EC2 RunInstances, that are available for use in different accounts within an organization.

#### **Migrating accounts between organizations**

Accounts can be migrated between organizations.

You must have root or IAM access to both the member and master accounts.

Use the AWS Organizations console for just a few accounts.

Use the AWS Organizations API or AWS Command Line Interface (AWS CLI) if there are many accounts to migrate.

Billing history and billing reports for all accounts stay with the master account in an organization.

Before migration download any billing or report history for any member accounts that you want to keep.

When a member account leaves an organization, all charges incurred by the account are charged directly to the standalone account.

Even if the account move only takes a minute to process, it is likely that some charges will be incurred by the member account.

## Resource Groups

You can use resource groups to organize your AWS resources.

In AWS, a resource is an entity that you can work with.

Resource groups make it easier to manage and automate tasks on large numbers of resources at one time.

Resource groups allow you to group resources and then tag them.

The Tag Editor assists with finding resources and adding tags.

**You can access Resource Groups through any of the following entry points:**

* On the navigation bar of the AWS Management Console.
* In the AWS Systems Manager console, from the left navigation pane entry for Resource Groups.
* By using the Resource Groups API, in AWS CLI commands or AWS SDK programming languages.

A resource group is a collection of AWS resources that are all in the same AWS region, and that match criteria provided in a query.

In Resource Groups, there are two types of queries on which you can build a group.

Both query types include resources that are specified in the format AWS::service::resource.

* **Tag-based –**Tag-based queries include lists of resources and tags. Tags are keys that help identify and sort your resources within your organization. Optionally, tags include values for keys.
* **AWS CloudFormation stack-based –**In an AWS CloudFormation stack-based query, you choose an AWS CloudFormation stack in your account in the current region, and then choose resource types within the stack that you want to be in the group. You can base your query on only one AWS CloudFormation stack.

Resource groups can be nested; a resource group can contain existing resource groups in the same region.

AWS Resource Access Manager

AWS Resource Access Manager (RAM) is a service that enables you to easily and securely share AWS resources with any AWS account or within your AWS Organization.

You can share AWS Transit Gateways, Subnets, AWS License Manager configurations, and Amazon Route 53 Resolver rules resources with RAM.

RAM eliminates the need to create duplicate resources in multiple accounts, reducing the operational overhead of managing those resources in every single account you own.

You can create resources centrally in a multi-account environment, and use RAM to share those resources across accounts in three simple steps:

1. Create a Resource Share.
2. Specify resources.
3. Specify accounts.

RAM is available at no additional charge.

**Key benefits:**

* **Reduce Operational Overhead –** Procure AWS resources centrally, and use RAM to share resources such as subnets or License Manager configurations with other accounts. This eliminates the need to provision duplicate resources in every account in a multi-account environment.
* **Improve Security and Visibility –** RAM leverages existing policies and permissions set in AWS Identity and Access Management (IAM) to govern the consumption of shared resources. RAM also provides comprehensive visibility into shared resources to set alarms and visualize logs through integration with Amazon CloudWatch and AWS CloudTrail.
* **Optimize Costs –**Sharing resources such as AWS License Manager configurations across accounts allows you to leverage licenses in multiple parts of your company to increase utilization and optimize costs.

AWS CloudFormation

AWS CloudFormation is a service that gives developers and businesses an easy way to create a collection of related AWS resources and provision them in an orderly and predictable fashion.

AWS CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment.

CloudFormation can be used to provision a broad range of AWS resources.

Think of CloudFormation as deploying infrastructure as code.

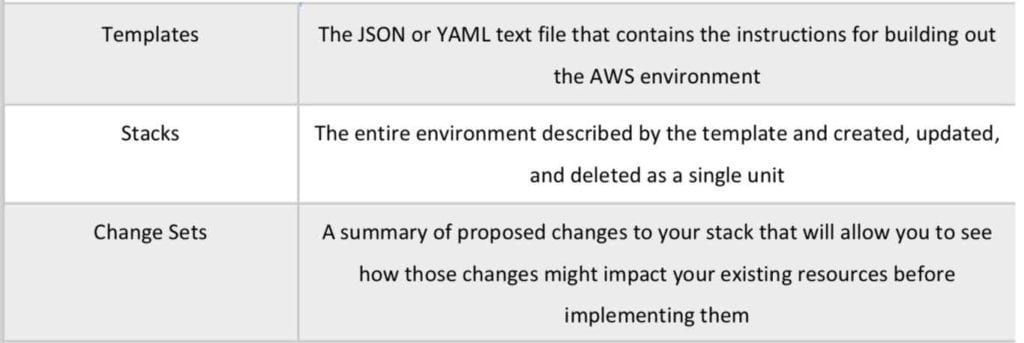
Elastic Beanstalk is more focussed on deploying applications on EC2 (PaaS).

CloudFormation can deploy Elastic Beanstalk-hosted applications however the reverse is not possible.

Logical IDs are used to reference resources within the template.

Physical IDs identify resources outside of AWS CloudFormation templates, but only after the resources have been created.

Concept of templates, stacks and change sets:



Templates:

* Architectural designs.
* Create, update and delete templates.
* Written in JSON or YAML.
* CloudFormation determines the order of provisioning.
* Don’t need to worry about dependencies.
* Modifies and updates templates in a controlled way (version control).
* Designer allows you to visualize using a drag and drop interface.

Stacks:

* Deployed resources based on templates.
* Create, update and delete stacks using templates.
* Deployed through the Management Console, CLI or APIs.

Template elements:

* Mandatory:
  + File format and version.
  + List of resources and associated configuration values.
* Not mandatory:
  + Template parameters (limited to 60).
  + Output values (limited to 60).
  + List of data tables.

Puppet and Chef integration is supported.

Can use bootstrap scripts.

Can define deletion policies.

Provides WaitCondition function.

Can create roles in IAM.

VPCs can be created and customized.

VPC peering in the same AWS account can be performed.

Route 53 is supported.

Stack creation errors:

* Automatic rollback on error is enabled by default.
* You will be charged for resources provisioned even if there is an error.

Updating stacks:

* AWS CloudFormation provides two methods for updating stacks: direct update or creating and executing change sets.
* When you directly update a stack, you submit changes and AWS CloudFormation immediately deploys them.
* Use direct updates when you want to quickly deploy your updates.
* With change sets, you can preview the changes AWS CloudFormation will make to your stack, and then decide whether to apply those changes.

StackSets.

* AWS CloudFormation StackSets extends the functionality of stacks by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation.
* Using an administrator account, you define and manage an AWS CloudFormation template, and use the template as the basis for provisioning stacks into selected target accounts across specified regions.
* An administrator account is the AWS account in which you create stack sets.
* A stack set is managed by signing in to the AWS administrator account in which it was created.
* A target account is the account into which you create, update, or delete one or more stacks in your stack set.

Before you can use a stack set to create stacks in a target account, you must set up a trust relationship between the administrator and target accounts.

Best Practices.

* AWS provides Python “helper scripts” which can help you install software and start services on your EC2 instances.
* Use CloudFormation to make changes to your landscape rather than going directly into the resources.
* Make use of Change Sets to identify potential trouble spots in your updates.
* Use Stack Policies to explicitly protect sensitive portions of your stack.
* Use a version control system such as CodeCommit or GitHub to track changes to templates.

Charges:

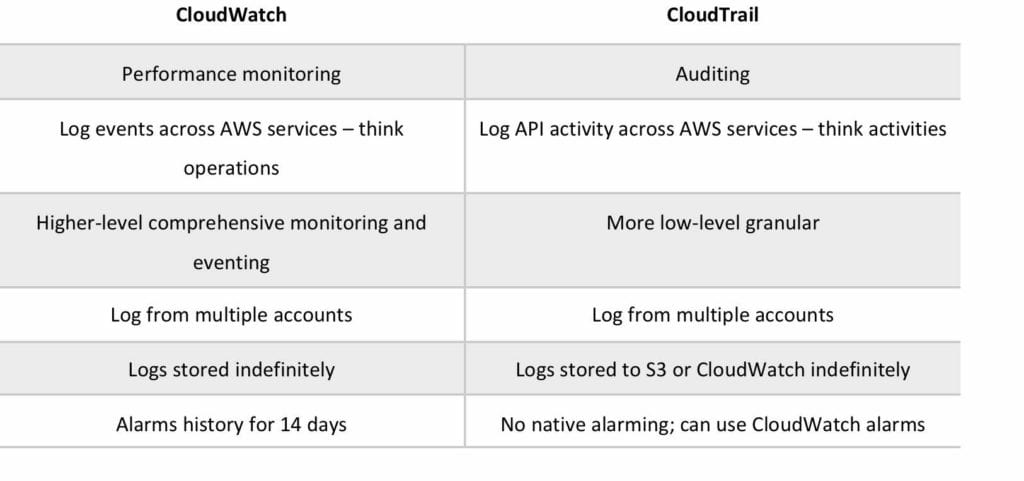
* There is no additional charge for AWS CloudFormation.
* You pay for AWS resources (such as Amazon EC2 instances, Elastic Load Balancing load balancers, etc.) created using AWS CloudFormation in the same manner as if you created them manually.
* You only pay for what you use, as you use it; there are no minimum fees and no required upfront commitments.

AWS CloudTrail

AWS CloudTrail is a web service that records activity made on your account

A CloudTrail trail can be created which delivers log files to an Amazon S3 bucket.

CloudWatch vs CloudTrail:



CloudTrail is about logging and saves a history of API calls for your AWS account.

Provides visibility into user activity by recording actions taken on your account.

API history enables security analysis, resource change tracking, and compliance auditing.

Logs API calls made via:

* AWS Management Console.
* AWS SDKs.
* Command line tools.
* Higher-level AWS services (such as CloudFormation).

CloudTrail records account activity and service events from most AWS services and logs the following records:

* The identity of the API caller.
* The time of the API call.
* The source IP address of the API caller.
* The request parameters.
* The response elements returned by the AWS service.

CloudTrail is per AWS account.

Trails can be enabled per region or a trail can be applied to all regions.

Trails can be configured to log data events and management events:

* **Data events:** These events provide insight into the resource operations performed on or within a resource. These are also known as data plane operations.
* **Management events:** Management events provide insight into management operations that are performed on resources in your AWS account. These are also known as control plane operations. Management events can also include non-API events that occur in your account.

CloudTrail log files are encrypted using S3 Server Side Encryption (SSE).

You can also enable encryption using SSE KMS for additional security.

A single KMS key can be used to encrypt log files for trails applied to all regions.

You can consolidate logs from multiple accounts using an S3 bucket:

1. Turn on CloudTrail in the paying account.
2. Create a bucket policy that allows cross-account access.
3. Turn on CloudTrail in the other accounts and use the bucket in the paying account.

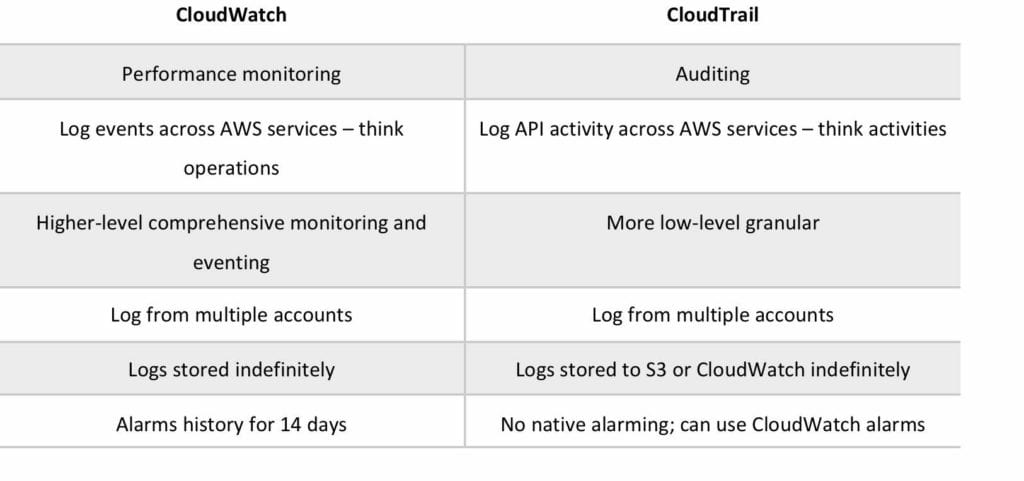
You can integrate CloudTrail with CloudWatch Logs to deliver data events captured by CloudTrail to a CloudWatch Logs log stream.

CloudTrail log file integrity validation feature allows you to determine whether a CloudTrail log file was unchanged, deleted, or modified since CloudTrail delivered it to the specified Amazon S3 bucket.

Amazon CloudWatch

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS.

CloudWatch vs CloudTrail:



Used to collect and track metrics, collect and monitor log files, and set alarms.

Automatically react to changes in your AWS resources.

With CloudWatch you can monitor resources such as:

* EC2 instances.
* DynamoDB tables.
* RDS DB instances.
* Custom metrics generated by applications and services.
* Any log files generated by your applications.

Gain system-wide visibility into resource utilization.

Monitor application performance.

Monitor operational health.

CloudWatch is accessed via API, command-line interface, AWS SDKs, and the AWS Management Console.

CloudWatch integrates with IAM.

CloudWatch Logs:

* Amazon CloudWatch Logs lets you monitor and troubleshoot your systems and applications using your existing system, application and custom log files.
* You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Route 53, and other sources.
* CloudWatch Logs can be used for real time application and system monitoring as well as long term log retention.
* CloudWatch Logs keeps logs indefinitely by default.
* CloudTrail logs can be sent to CloudWatch Logs for real-time monitoring.
* CloudWatch Logs metric filters can evaluate CloudTrail logs for specific terms, phrases or values.

CloudWatch retains metric data as follows:

* Data points with a period of less than 60 seconds are available for 3 hours. These data points are high-resolution custom metrics.
* Data points with a period of 60 seconds (1 minute) are available for 15 days.
* Data points with a period of 300 seconds (5 minute) are available for 63 days.
* Data points with a period of 3600 seconds (1 hour) are available for 455 days (15 months).