Automation

Đà Nẵng, năm COVID thứ 2

CloudFormation:

- A provisioning automation service for setting up AWS services.
- Stacks:
  - you manage related resources as a single unit called a stack.
  - You create, update, and delete a collection of resources by creating, updating, and deleting stacks.
  - Stacks can be nested. E.g. An application stack referencing a network stack.
  - Stacks are created using CloudFromation templates.
  - Stack creation errors: by default the whole stack is rolled back.

- Templates:
    - You create a template that describes all the AWS resources that you want and AWS CloudFormation takes care of provisioning and configuring those resources for you.
    - A template is a JSON or YAML formatted text file used as a blueprint for building your AWS resources.
    - Templates include several major sections. The Resources section is the only required section.
    - Templates can be imported from S3 or uploaded from your computer.
- Parameters Section:
    - Values to pass to your template at runtime (when you create or update a stack).
    - You can refer to parameters from the Resources and Outputs sections of the template.
    - You can have a maximum of 200 parameters in a template.
    - Parameters can be marked as mandatory or optional. In the latter case, you provide a default value.

- Rules section:
    - Validates a parameter or a combination of parameters passed to a template during a stack creation or stack update.
- Mappings section:
    - a mappings section can include multiple maps.
    - Each map is a list of map keys.
    - Each map key points to a dictionary of key-value pairs.
    - Eg: map of Regions: map key = region name. Key-value pairs: AMI IDs in that region.
    - You can match a key to a corresponding value by using the Fn::FindInMap intrinsic function in the Resources and Outputs sections.
- Resources section:
    - Specifies the stack resources and their properties, such as an EC2 instance or an S3 bucket.
    - You can refer to resources in the Resources and Outputs sections of the template.
    - With the DependsOn attribute you can specify that the creation of a specific resource follows another.

- Outputs section:
    - A list of dictionaries with the following keys: OutputKey, OutputValue, Description.
    - You can declare a maximum of 200 outputs in a template.
- Stack modification:
    - you can make changes to a stack without having to tear it down and recreate it.
    - Direct Changes: used for quick changes.
    - Change Set: used for more complex changes and when you want to review the changes before applying them. A change set is saved.
- Stack drifting:
    - CloudFromation can detect drifted configurations: resources that have been changed from outside CloudFormation and therefore do not match the template anymore.
    - Not supported on all resources.
    - The execution of a stack drifting search is launched manually.

- StackSets:
    - StackSets extends the functionality of stacks by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation.
    - A stack instance is a reference to a stack in a target account within a Region.
    - 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 target account is the account into which you create, update, or delete one or more stacks in your stack set.
    - Stack sets can be created using either self-managed permissions or service-managed permissions.
    - With self-managed permissions, you create the IAM roles in each target account to enable the deployment of the stack instances.
    - With service-managed permissions, you can deploy stack instances to accounts managed by AWS Organizations. StackSets creates the necessary IAM roles on your behalf in current and future accounts of the Organization.

- Deletion Policy:
  - With the DeletionPolicy attribute you can preserve, and in some cases, backup a resource when its stack is deleted.
  - You specify a DeletionPolicy attribute for each resource that you want to control.
  - If a resource has no DeletionPolicy attribute, AWS CloudFormation deletes the resource by default. Exception: The default policy is Snapshot for RDS.
  - To keep a resource when its stack is deleted, specify "DeletionPolicy: Retain" for that resource.
  - You can choose also "DeletionPolicy: Snapshot" for the following resources: EC2 Volume, RDS, Redshift, ElastiCache, Neptune.
- Puppet and Chef Integration.
- You can define Bootstrap scripts.
- Supports WaitCondition

AWS Step Functions:

- A serverless orchestration service that lets you combine AWS Lambda functions and other AWS services to build business-critical applications.

AWS Systems Manager (formerly known as SSM):

- An AWS service that you can use to view and manage your instances’ OS on AWS.
- Capabilities:
  - Operations Management.
  - Application Management: create, manage, and deploy application configurations. Includes the « Parameter Store ».
  - Change Management: enterprise change management framework for requesting, approving, implementing, and reporting on operational changes to your application configuration and infrastructure.
  - Node Management: OS/software inventory, patch management, troubleshooting, …
  - Shared Resources (SSM Documents).

- SSM agent:
  - On EC2: installed by default in Amazon AMIs. Needs to be enabled and started.
  - Can also be installed on on-prem servers. This is called a Hybrid environment. Needs an Amazon provided certificate and activate code.
  - The EC2 instance needs to have an instance profile (IAM role) so that the agent can push data to SSM.
  - A managed instance is any EC2 instance or on-premises machine in your hybrid environment that has been configured for Systems Manager.

- SSM Parameter Store:
  - Provides secure, hierarchical storage for configuration data and secrets management.
  - You can store data such as passwords, database strings, EC2 instance IDs and Amazon Machine Image (AMI) IDs, and license codes as parameter values.
  - You can store values as plain text or encrypted data using a KMS CMK (SecureString parameter type).
  - Parameters are retrieved using the GetParameter API call to ssm.{region}.amazonaws.com.
  - The parameters hierarchy is similar to a file path.
  - You can choose to encrypt your SecureString parameters using either:
    - your own customer-managed KMS CMK: you need to add an IAM permission to the user to use KMS operations on this KMS key.
    - or the default AWS-managed KMS CMK that SSM create for you in your account: all users within the account have access to this key, unless explicitly denied.
  - You can use AWS CloudTrail to monitor SecureString parameter activities.

- SSM Inventory:
  - Collects metadata from your instances: OS, applications, files, network, ...etc.
  - You can also collect custom inventory metadata.
  - To assign custom inventory to an instance, you can either store this metadata in a JSON file on your instance or use the Systems Manager PutInventory API action.
  - The shortest period for inventory is 30mn.
  - The inventory configuration is called an SSM association. Creating and association gives you an association ID and code.
  - The collected data can be stored in S3.

- SSM Run Command
  - Enables you to automate common administrative tasks and perform ad hoc configuration changes at scale.
  - Examples: install or bootstrap applications, build a deployment pipeline, capture log files when an instance is terminated from an Auto Scaling group, and join instances to a Windows domain.

- SSM document:
  - Defines the actions that Systems Manager performs on your managed instances.
  - Systems Manager includes more than 100 pre-configured documents that you can use by specifying parameters at runtime.
  - Documents use JSON or YAML formats, and they include steps and parameters that you specify.
  - Command/Automation documents: Used by "Run Command" to run commands. Used by "State Manager" to apply a configuration. Used by "Maintenance Windows" to apply a configuration based on the specified schedule.
  - Package documents: Used by "Distributor" to install software on managed instances.
  - Session document: Session Manager uses session documents to determine which type of session to start.
  - CloudFormation document: you can choose to store your AWS CloudFormation templates in SSM.

- SSM Patch Manager:
  - You can use Patch Manager to apply patches for operating systems and for applications.
  - Patch Manager uses patch baselines, which include rules for auto-approving patches within days of their release, as well as a list of approved and rejected patches.
  - Can target individual instances or a group of instances. Can target instances having specific tags.
  - Can scan your instances and report compliance on a schedule, install available patches on a schedule, and patch or scan instances on demand whenever you need to.
  - You create Maintenance Windows to run patching.
  - Uses "Run Command" in the background.
- SSM Change Calendar: lets you set up date and time ranges when actions you specify (for example, in Systems Manager Automation documents) may or may not be performed in your AWS account.
- SSM Session Manager: lets you connect to your instances without using a bastion host.