AWS Automations Services

14 March 2024 15:12

1. Process Overview:
    - Write your infrastructure code using JSON or YAML templates.
    - Deploy your code as a stack or stack set in CloudFormation, where a stack represents a collection of resources managed as a single unit.
    - CloudFormation handles the creation, updating, or deletion of AWS resources based on the instructions provided in the template.
2. Benefits of CloudFormation:
    - Enables managing infrastructure changes, tracking versions, and collaborating with team members using version control systems like Git.
    - Provides automation for provisioning and managing resources, ensuring consistency and reducing the likelihood of human error.
    - Supports tagging of stacks and resources for better organization and cost allocation.
    - Offers flexibility to deploy portable stacks into multiple regions and preview changes before applying them using change sets.
3. Template Sections:
    - Understand the required sections of a CloudFormation template, including Template Format Version, Resources, and Outputs.
    - Optional sections like Parameters, Mappings, and Transform provide additional flexibility and customization options.
4. Console Demo:
    - Demonstrated the process of creating a CloudFormation stack using the AWS Management Console.
    - Explained the structure of a CloudFormation template and its visual representation in CloudFormation Designer.
    - Showed how CloudFormation automates resource creation and provides insights into the deployment process.
5. Exam Tips:
    - CloudFormation is suitable for creating immutable infrastructure and automating resource management.
    - Understand the layout of a CloudFormation template and the purpose of each section, including required sections.
    - Be aware of rollback behavior in case of errors during stack creation or updates.
    - Avoid hard-coding values and resource IDs in templates for better flexibility and reliability.
    - Know that CloudFormation uses the same API calls as manual resource provisioning but automates the process for repeatable deployments.

Overall, CloudFormation is a powerful tool for managing and automating AWS infrastructure, offering benefits such as consistency, repeatability, and scalability. Mastering CloudFormation is essential for efficient and reliable infrastructure management in AWS environments.

From <https://chat.openai.com/c/d6fe90a1-a00b-4374-9bfc-54c5891880f3>

6. Platform as a Service (PaaS):
    - PaaS is an application deployment model where developers supply their code, and the provider manages the underlying infrastructure, simplifying the

  deployment and management process.
  - With PaaS, developers can focus on app development while the platform takes care of building, deploying, and managing the infrastructure.
7. Elastic Beanstalk Features:
   - Elastic Beanstalk is AWS's PaaS offering, allowing developers to deploy and manage web applications without dealing with the underlying infrastructure.
   - It supports various programming languages and platforms like Java, .NET, PHP, Node.js, Ruby, and Docker.
   - Elastic Beanstalk automates the deployment process, enables easy application code upload and testing, and manages the internal setup of instances, eliminating the need for manual host management.
8. Exam Tips:
   - Understand that Elastic Beanstalk automates the deployment and scaling of web applications, supporting various programming languages and platforms.
   - Recognize that PaaS services like Elastic Beanstalk provision AWS resources such as instances, databases, S3, and load balancers automatically.
   - Note that Elastic Beanstalk handles automatic updates for operating systems and application servers, provides built-in monitoring, metrics, and health checks, and can manage instances fully or allow for manual administrative control.

Overall, Elastic Beanstalk is a valuable service for simplifying application deployment and management, and understanding its features and capabilities is essential for AWS certification

From <https://chat.openai.com/c/d6fe90a1-a00b-4374-9bfc-54c5891880f3>

9. AWS Systems Manager Overview:
   - AWS Systems Manager is a suite of tools designed to provide visibility, control, and automation capabilities for managing instances both inside and outside of AWS.
   - The Systems Manager agent (SSM Agent) is a critical component installed on managed instances, enabling communication with the Systems Manager service.
10. Capabilities of Systems Manager:
    - Automation: Streamlines resource management with predefined or custom playbooks (documents).
    - Run Command: Enables remote execution of commands and scripts on managed instances without SSH or RDP access.
    - Patch Manager: Automates patching of operating systems and applications to ensure security and compliance.
    - Parameter Store: Securely stores configuration data and secrets, allowing hierarchical organization and granular access control.
    - Maintenance Windows: Defines schedules for performing actions on managed instances, facilitating efficient maintenance and updates.
    - Session Manager: Provides secure connectivity to managed instances without direct SSH or RDP access.
11. Important Concepts:
    - Logging: Systems Manager logs session activities for auditing and compliance purposes.
    - SSM Agent: Facilitates secure and remote management of instances without opening unnecessary ports.
    - Parameter Store Types: Supports string, string list, and secure string parameter types, allowing storage of various data types securely.
12. Demonstration:
    - Showcased the use of Systems Manager to connect to an EC2 instance using Session Manager, create a parameter in Parameter Store, and execute a run

command script to pull the parameter value onto the local file system.

13. Exam Tips:
    - Systems Manager can manage both on-premises and AWS compute resources, offering centralized management across hybrid environments.
    - Understand the capabilities of Patch Manager for maintaining security and compliance.
    - Differentiate between Parameter Store and Secrets Manager for storing configuration data and secrets.
    - Utilize Session Manager for secure connectivity to compute resources without traditional SSH or RDP access.
    - Ensure proper installation and configuration of the SSM agent on managed instances for Systems Manager interactions.

    Overall, Systems Manager offers a comprehensive set of tools for managing and automating infrastructure tasks, and understanding its features and capabilities is crucial for AWS certification exams.

From <https://chat.openai.com/c/d6fe90a1-a00b-4374-9bfc-54c5891880f3>

14. Automation Considerations:
    - Always consider automation options when faced with manual steps in a scenario.
    - Ensure automation solutions are repeatable, preferably using AWS-provided frameworks.
    - Consider cross-region and cross-account implications when designing automation solutions to avoid hard-coding IDs and values.
15. CloudFormation Tips:
    - Understand the sections of a CloudFormation template, including parameters, mappings, and resources.
    - Favor immutable architecture and stateless environments for robust and disposable infrastructure.
    - Avoid hard-coding IDs and values directly into the template; store them in mappings or Parameter Store instead.
16. Elastic Beanstalk and Systems Manager Tips:
    - Elastic Beanstalk is ideal for quick and simple web server deployments, while CloudFormation offers more granular control for complex architectures.
    - Systems Manager automation documents are versatile tools for configuring both EC2 instances and AWS environment settings.
    - Familiarize yourself with storing values in Parameter Store and utilizing Session Manager for secure instance access.

    These exam tips provide a solid foundation for understanding and utilizing automation tools effectively in AWS environments. Remember to practice scenarios and familiarize yourself with these tools to confidently tackle automation-related questions in the exam

From <https://chat.openai.com/c/d6fe90a1-a00b-4374-9bfc-54c5891880f3>