**Systems Operations Overview**

*The Deployment/Administration/Monitoring of Systems & Network Resources in an Automatable & Reusable Manner.*

|  |  |
| --- | --- |
| *SysOps Tasks & Responsibilities* | |
| *Build* | *Create Separate Environments for Deployment/Test/Production.* |
| *Test* | *Test Backup & Disaster Recover Procedures.* |
| *Deploy* | *Deploy Applications & Workloads into RTE.* |
| *Monitor* | *Monitor Health/Performance of Infrastructure Resources.* |
| *Maintain* | *Apply Patches/Upgrades Consistently/Regularly.* |
| *Safeguard* | *Apply/Enforce Security Measures in all Infrastructure Levels.* |

Cloud Computing allows Organizations to Automate Complex IT Operation Development/Testing/Deployment.

**Automation** allows Repeatable Deployment, Creation of Self-Describing Systems and the Ability to Build Well-Tested Secure Systems, it can take place through differing platforms (*Linux Bash Shell, Python, Cloud Formation IaC Deployment*).

**Building a Troubleshooting Knowledge Base** is done through all Labs we do moving forward with a template detailing question:

|  |  |  |
| --- | --- | --- |
| *What Happened?* | *What Service?* | *Symptoms?* |
| *RCA? (Root Cause Analysis).* | *Resolution/Solution Procedures.* |  |

**AWS Identity & Access Management**

|  |  |
| --- | --- |
| **Types of Credentials** | |
| **Email & Password** | Associate with AWS Account (Root User) |
| **IAM User & Password** | Used for Accessing AWS Management Console. |
| **Access Keys** | Typically with AWS CLI & Programmatic Requests (APIs/SDKs) |
| **MFA** | Extra Layer of Security, Can be turned on for Account Root User/ IAM Users. |
| **Key Pairs** | Only Specific AWS Services (i.e. – EC2) |

**A screenshot of a computer

Description automatically generated**

|  |  |
| --- | --- |
| **Policy Types** | |
| **Identity-Based** | Allow Users to attach Manged/Inline Policies to IAM Identities (users/groups) or Roles. *Defined & Stored as JSON Documents.* |
| **Resource-Based** | Attach Inline Policies to Resources (i.e. – S3 Bucket Policies/IAM Role-Trust Policies), *Defined & Stored as JSON Documents*. Only available for certain AWS Services/Resources. |
| **AWS Organization Service Control Policies (SCPs)** | Apply Permission Boundaries to AWS Organizations/Organizational Units (OUSs)/Accounts. *Uses JSON format.* |
| **Access Control Lists (ACLs)** | Controls which Principals (Users/Resources) can access Resource. Like Resource-Based but *don’t use JSON Policy Document Structure.* |

**A screenshot of a computer

Description automatically generated**

|  |  |
| --- | --- |
| **IAM Best Practices** |  |
| *Avoid Using Root User Credentials for Daily Administration* | *Delegate Admin Functions through Principle of Least Privilege.* |
| *Use IAM Roles to provide Cross-Account Access.* | *Implement MFA as Additional Level of Account Security.* |

**AWS Command Line Interface**

*Can be accessed through either the AWS Management Console, the AWS CLI or SDKs (which call AWS services API from major programming languages).*

|  |  |  |
| --- | --- | --- |
| *AWS CLI Output Format Options* | | |
| **Table**   * Human-Readable. * Tabular Form. | **JSON**   * Default Format * Useful Output for Programmatic Usage. | **Text**   * Tab-Delimited Lines. * Useful in Bash/PowerShell Scripts. |

***To Install the AWS CLI, follow these steps:***

1. Use the Curl Command, specifying the file name the downloaded package is written to.

*$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"*

1. Use the Unzip Command to extract the Installer Package, *use an Equivalent Program if Unzip isn’t available.*

*$ unzip awscliv2.zip*

1. Run the Install Program *which are installed to /usr/local/aws-cli by default.*

*$ sudo ./aws/install*

1. Use the Version Command to confirm the Installation.

*$ aws –version*

1. Use the Command Line

*A black and white text on a black background

Description automatically generated*

*A black background with white text

Description automatically generated*

*A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated*

*A black and white text on a white background

Description automatically generatedA screenshot of a computer screen

Description automatically generated*

**Tooling & Automation**

*Concerns the AWS Systems Manager (SSM Inventory, Patching, Config), CloudFormation and Open Fleets.*

The **Run** command controls/schedules tasks through predefined (Install Windows Update) or Linux Update. It’s capable of running custom commands.

***The following illustrate Patch Manager Responsibilities:***

A diagram of patch manager

Description automatically generated

**State Managers** help Automate Fleet Setup/Build/Monitor Configuration Drift. They create an Automation Document and link/associate it with you Instances, *Specifying a Schedule and Outputting Data/Logs to an S3 Bucket.*

The **Parameter Store Stores Set-Up Credentials Securely, using AWS KMS to Encrypt Parameters.**

The **Inventory** includes *App Data, Files, Network Config, Server Roles, Updates, General System Properties and Windows Services.*

**Administration & Development Tools**

**Software Development Kits (SDKs)** are used to access AWS Services Programmatically & Write Administrative Scripts in Different Programming Languages (i.e. - *.NET, C++, Java/script, Ruby, Python, etc*)

**AWS Cloud Formation** creates a Stack from a Template (*Written in JSON/YAML*) where you can Preview Stack Changes, Detect Config/Built Drift and can be used with Lambda Functions. *It works through the following steps*:

1. Define Resources in Template/Prebuilt.
2. Upload Template to CloudFormation/Point to Template stored in S3 Bucket.
3. Run Create Stack Creating Resources across Multiple AWS Services.
4. Stack Retains Resource Control (*Update/Delete Stack, Detect Drift*).

|  |  |  |
| --- | --- | --- |
| **Cloud Formation Benefits** | | |
| *Deploy Complex Environments Rapidly.* | *Duplicate the Same Environment.* | *Ensure Configuration Consistency.* |
| *Delete Resources in Single Action (Delete Stack)* | *Propagate Same Change to all Stacks (Update Stacks)* |  |

**AWS OpsWorks** can be used to Automate when you Configure/Deploy/Manage Servers, *based of Chef/Puppet Open-Source Automation Platforms.*