**Made By – Nitanshu Tak**

**Synopsis-1**

**Project Report on** [**AWS InfraControl CLI Tool**](https://github.com/Nitanshu715)

**1. Synopsis of the Problem Statement**

In this project, we aim to automate the management of EC2 instances using AWS CLI and Python scripts, integrating security groups and key pair management as well. The problem is to efficiently and cost-effectively manage cloud resources, including the creation, termination, and configuration of instances and other essential elements like security groups, key pairs, and volumes, while minimizing costs.

**2. Literature Review**

Cloud computing has emerged as a pivotal technological advancement, enabling businesses to scale their operations quickly and efficiently. AWS (Amazon Web Services) provides several tools to manage and automate cloud resources. Many studies have explored the efficient management of resources like EC2 instances, storage, and networking, which have led to the development of frameworks and tools for cloud automation.

In recent years, cloud computing has become a driving force in the IT industry due to its flexibility, scalability, and efficiency. A major part of cloud computing is the management of virtual machines and resources. Amazon EC2, a key service offered by AWS, enables the deployment of virtual servers that can be easily scaled based on demand. In the past decade, many solutions have been developed to simplify and automate the management of EC2 instances and other cloud resources.

Key Literature and Findings:

1. Amazon EC2 and Cloud Computing: Amazon EC2 enables the scalable deployment of instances in the cloud. It's a popular IaaS (Infrastructure as a Service) platform that allows users to provision virtual servers.  
Reference: ‘Cloud Computing: Principles, Systems and Applications’ by Nikos Antonopoulos and Lee Gillam (Springer, 2010).

2. Automation and Cloud Infrastructure Management: Many studies focus on automating cloud infrastructure using scripting languages like Python and AWS CLI to increase efficiency and reduce human error.  
Reference: ‘Automated Cloud Infrastructure Management: Towards Cloud Automation’ by M. Moniruzzaman et al. (Journal of Cloud Computing, 2020).

3. Security Groups and Key Management in AWS: AWS security groups are used to control inbound and outbound traffic for instances, ensuring the security of resources.  
Reference: ‘Security Challenges in Cloud Computing’ by R. Buyya et al. (IEEE International Conference on Cloud Computing, 2018).

**3. Objective**

The primary objective of this project is to automate EC2 instance management through AWS CLI and Python scripts. The project aims to address the challenges of efficient resource management, automation of routine tasks, and cost optimization.

We used AWS CLI for infrastructure management and Python (Lambda) for automation. Our project follows an Infrastructure-as-Code (IaC) approach, which is a best practice.

The key objectives are:

1. Automate EC2 Instance Management: The system should enable users to create, start, stop, and terminate EC2 instances through automation.

2. Automate Security Group Configuration: The system should allow users to create security groups, configure inbound/outbound rules, and assign them to EC2 instances.

3. Automate Key Pair Management: The system should automate the creation and deletion of key pairs, ensuring secure access to EC2 instances.

4. Minimize Cloud Resource Costs: The system should automatically identify unused resources, such as EC2 instances, S3 buckets, and volumes, and terminate them to reduce unnecessary costs.

5. Provide User-Friendly Interfaces: The system should provide a user-friendly interface for interacting with cloud resources via AWS CLI commands and Python scripts.

**4. Reference**

1. Amazon EC2 Documentation – AWS EC2, Available at: https://aws.amazon.com/ec2/

2. AWS CLI Command Reference – AWS CLI Documentation, Available at: https://docs.aws.amazon.com/cli/latest/reference/

3. Cloud Computing: Principles, Systems and Applications by Nikos Antonopoulos, Lee Gillam – Springer (2018). DOI: 10.1007/978-1-84996-291-1

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://download.e-bookshelf.de/download/0009/9634/13/L-G-0009963413-0020076340.pdf

4. AWS Security Best Practices – AWS Security Documentation, Available at: https://aws.amazon.com/security/

5. Security Challenges in Cloud Computing by R. Buyya et al. (2018) – IEEE International Conference on Cloud Computing. DOI: 10.1109/CloudCom.2018.00016

<https://www.researchgate.net/publication/354788317_CLOUD_COMPUTING_SECURITY_CHALLENGES>

**5. Software Requirements Specification (SRS)**

# Functional Requirements:

1. Instance Management: The system should allow users to create, stop, and terminate EC2 instances.

2. Security Group Management : The system should create and delete security groups. It should also allow users to specify inbound and outbound rules for security groups.

3. Key Pair Management: The system should create and delete key pairs for EC2 instances.

4. Cost Management: The system should ensure that unused instances, volumes, and S3 buckets are terminated to reduce costs.

# Non-Functional Requirements:

1. Performance: The system should respond to user commands within 5 seconds.

2. Reliability: The system should ensure that there are no errors while terminating or creating resources.

**6. Bibliography of IEEE Research Papers (2018-2024)**

1. Buyya, R., et al. (2018). ‘Security Challenges in Cloud Computing.’ IEEE International Conference on Cloud Computing. DOI: 10.1109/CloudCom.2018.00016

2. Moniruzzaman, M., et al. (2020). ‘Automated Cloud Infrastructure Management: Towards Cloud Automation.’ Journal of Cloud Computing, 9(1), 101–114. DOI: 10.1007/s10207-019-0485-5

3. Antonopoulos, N., Gillam, L. (2019). ‘Cloud Computing: Principles, Systems, and Applications.’ Springer. DOI: 10.1007/978-1-84996-291-1

4. Aziz, M., et al. (2019). ‘Cost Optimization in Cloud Computing: A Survey.’ IEEE Access, 7, 103234-103249. DOI: 10.1109/ACCESS.2019.2928384

5. Chhabra, M., et al. (2021). ‘Secure and Cost-efficient Cloud Resource Management for Enterprises.’ IEEE Transactions on Cloud Computing. DOI: 10.1109/TCC.2021.3053521

**7. Implementation**

The implementation involves automating the management of AWS EC2 instances using AWS CLI and Python scripts. The flow of execution includes:

1. Configuration of AWS CLI with `aws configure`.

2. Creating EC2 instances and security groups via AWS CLI commands.

3. Monitoring the status of EC2 instances and terminating instances that are no longer needed.

4. Automating key pair creation and security group assignment.

The Python script implementation for terminating EC2 instances can be found in the project directory. Python libraries like `boto3` were used to interact with AWS services programmatically.