Automate The Servers Using Cloud Watch

Final Report

Department of Computer Science & Application

Institute of Engineering & Technology

SUBMITTED TO: -

Mr. Amir Khan

(Technical Trainer)

SUBMITTED BY: -

Yash Gupta(201500817)

Acknowledgement

It gives us a great sense of pleasure to present the synopsis of the B.Tech mini project undertaken during B.Tech III Year. This project is going to be an acknowledgement to the inspiration, drive and technical assistance will be contributed to it by many individuals. We owe special debt of gratitude to Mr Amir Khan, Technical Trainer , for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal and for his constant support and guidance to our work.

His sincerity, thoroughness and perseverance has been a constant source of inspiration for us. We believe that he will shower us with all his extensively experienced ideas and insightful comments at different stages of the project & also taught us about the latest industry-oriented technologies. We also do not like miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and co-operation.

YASH GUPTA (201500817)

ABSTRACT

The study described in this research report focused on variables which were posited to capture students' experiences of the online tutoring service, e-Learning, and relationships with the students' perceptions of their academic capabilities and academic performance. A theoretical model incorporating variables from the Technology Acceptance Model, the Theory of Planned Behaviour, and Social Cognitive Theory was developed and tested. A total of 506 undergraduate students from a university located in Sydney, Australia, completed an online survey. Data were analysed using confirmatory factor analysis (CFA) and structural equation modelling (SEM). The results suggested that the perceived usefulness of E-Learning had a direct positive relationship with academic self-efficacy, and an indirect positive association with the students' academic grades through academic self-efficacy. There was a direct positive relationship between academic self-efficacy and students' academic grades. The implications of these results and directions for future research are discussed in this report.

Contents

INTRODUCTION

Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track metrics, which are variables you can measure for your resources and applications.

The CloudWatch home page automatically displays metrics about every AWS service you use. You can additionally create custom dashboards to display metrics about your custom applications, and display custom collections of metrics that you choose.

You can create alarms that watch metrics and send notifications or automatically make changes to the resources you are monitoring when a threshold is breached. For example, you can monitor the CPU usage and disk reads and writes of your Amazon EC2 instances and then use that data to determine whether you should launch additional instances to handle increased load. You can also use this data to stop under-used instances to save money.

With CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health.

SOFTWARE AND HARDWARE REQUIREMENTS

- Amazon Web Services Log in Credentials
- Good Connectivity of Internet
- One Active Server
- Lamda Function's
- Created IAM Role's
- Cloud Watch

PROJECT DESCRIPTION

Automation, a capability of AWS Systems Manager, simplifies common maintenance, deployment, and remediation tasks for AWS services like Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS), Amazon Redshift, Amazon Simple Storage Service (Amazon S3), and many more. To get started with Automation, open the Systems Manager console. In the navigation pane, choose **Automation**.

Automation helps you to build automated solutions to deploy, configure, and manage AWS resources at scale. With Automation, you have granular control over the concurrency of your automations. This means you can specify how many resources to target concurrently, and how many errors can occur before an automation is stopped.

The project is divided into 3 modules – student, course expert and administrator. The roles of the modules are as follows:

- **Student :**

  The student selects from various courses available. The student **t**akes a test on a course. There might be courses, which has only test modules. Each question has multiple choices with only one correct answer. The test will be time bound. Student can see the test schedule. New Users will be able to register themselves in the system as students. All students will be able to modify their own profile. Student views previous test reports, receives feedback for a test taken Student can go to the discussion board and browse through questions and answers and discussing solutions of questions asked in test. Student can chat with course expert. Student can also send messages to the course expert.

- **Course Expert :**

  Creating test questions for the course**,** test questions will reside in the Draft area if either it is saved while creating/modifying or it has been rejected by admin. Modifying test questions, deleting the entire test, browse through the tests that students have submitted, just as a student would., view the results of those students that have taken test for his courses. Replying back to the messages from students.

- **Administrator:**

  Publish tests submitted by Course Experts. Before publishing test questions it is customary to get it reviewed by admin. After going through its content either it gets approved or gets rejected**.** Modify the profile of other users registered in the system. Change user status from inactive to active.

WORKING

A student has to register his profile for a course, by authentication and authorization and chat with others. A student can join discussion forums,

send mail to instructor(s) of the course and provide feedback about the test

given. A student can view test schedule, take test to assess his knowledge, view test report and edit his/her profile.

A course expert creates a test for the course, test questions will reside in the Draft area if either it is saved while creating/modifying or it has been rejected by admin. Modifying test questions, deleting the entire test, browse through the tests that students have submitted, just as a student would., view the results of those students that have taken test for his courses. Replying back to the messages from students.

Publish tests submitted by Course Experts. Before publishing test

questions it is customary to get it reviewed by admin. After going through its content either it gets approved or gets rejected. Modify the profile of other users registered in the system. Change user status from inactive to active

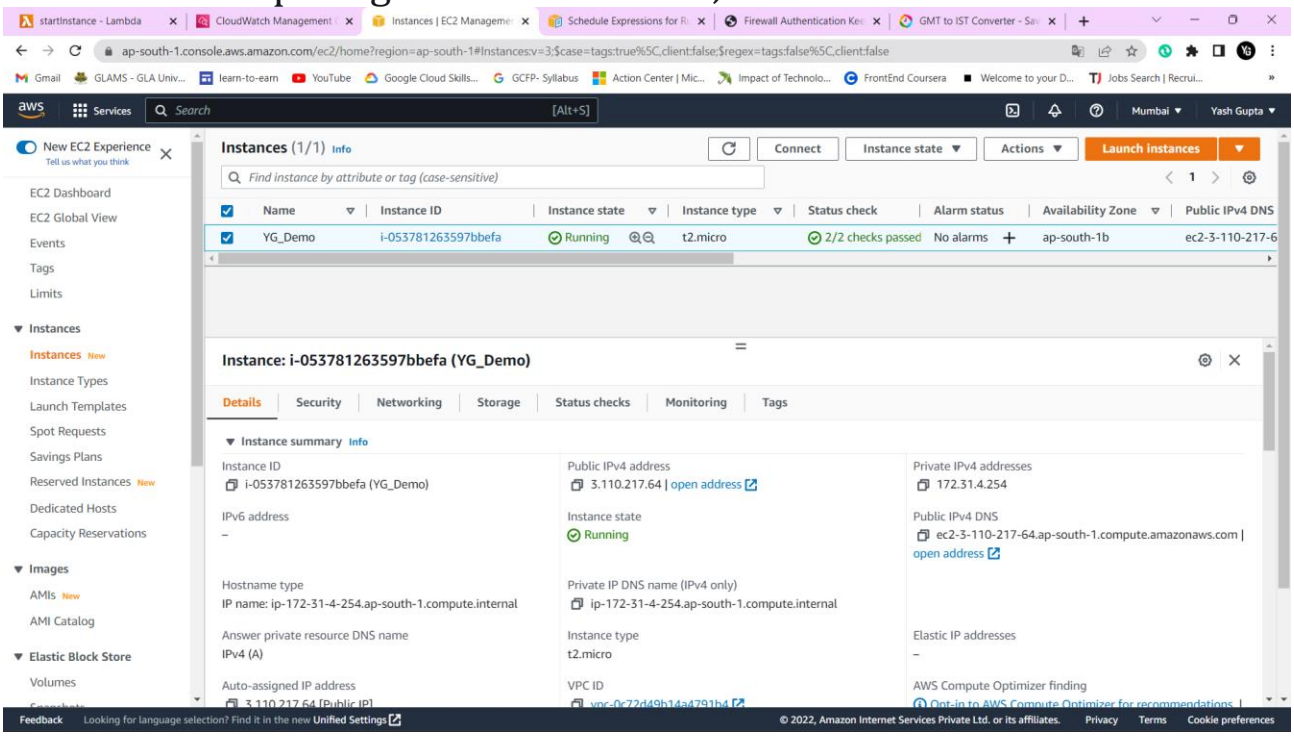
AWS Instance

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security

Amazon EC2 provides the following features:

- Virtual computing environments, known as *instances*



- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop, hibernate, or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *Regions* and *Availability Zones*
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using *security groups*

- Static IPv4 addresses for dynamic cloud computing, known as *Elastic IP addresses*
- Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources
- Virtual networks you can create that are logically isolated from the rest of the AWS Cloud, and that you can optionally connect to your own network, known as *virtual private clouds* (VPCs)

Related services

You can provision Amazon EC2 resources, such as instances and volumes, directly using Amazon EC2. You can also provision Amazon EC2 resources using other services in AWS. For more information,

- Amazon EC2 Auto Scaling User Guide
- AWS CloudFormation User Guide
- AWS Elastic Beanstalk Developer Guide
- AWS OpsWorks User Guide

- To automatically distribute incoming application traffic across multiple instances, use Elastic Load Balancing. For more information, see the Elastic Load Balancing User Guide.

- To get a managed relational database in the cloud, use Amazon Relational Database Service (Amazon RDS) to launch a database instance. Although you can set up a database on an EC2 instance, Amazon RDS offers the advantage of handling your database management tasks, such as patching the software, backing up, and storing the backups. For more information, see the Amazon Relational Database Service Developer Guide.

- To make it easier to manage Docker containers on a cluster of EC2 instances, use Amazon Elastic Container Service (Amazon ECS). For more information, see the Amazon Elastic Container Service Developer Guide or the Amazon Elastic Container Service User Guide for AWS Fargate.

- To monitor basic statistics for your instances and Amazon EBS volumes, use Amazon CloudWatch. For more information, see the Amazon CloudWatch User Guide.
- To detect potentially unauthorized or malicious use of your EC2 instances, use Amazon GuardDuty. For more information see the Amazon GuardDuty User Guide.

Access Amazon EC2

Amazon EC2 provides a web-based user interface, the Amazon EC2 console. If you've signed up for an AWS account, you can access the Amazon EC2 console by signing into the AWS Management Console and selecting **EC2** from the console home page.

If you prefer to use a command line interface, you have the following options:

AWS Command Line Interface (CLI)

Provides commands for a broad set of AWS products, and is supported on Windows, Mac, and Linux. To get started, see AWS Command Line Interface User Guide. For more information about the commands for Amazon EC2, see ec2 in the *AWS CLI Command Reference*.

AWS Tools for Windows PowerShell

Provides commands for a broad set of AWS products for those who script in the PowerShell environment. To get started, see the AWS Tools for Windows PowerShell User Guide. For more information about the cmdlets for Amazon EC2, see the AWS Tools for PowerShell Cmdlet Reference.

Amazon EC2 supports creating resources using AWS CloudFormation. You create a template, in JSON or YAML, that describes your AWS resources, and AWS CloudFormation provisions and configures those resources for you. You can reuse your CloudFormation templates to provision the same resources multiple times, whether in the same Region and account or in multiple Regions and accounts. For more information about the resource types and properties for Amazon EC2,

see EC2 resource type reference in the *AWS CloudFormation User Guide*.

Amazon EC2 provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named `Action`. For more information about the API actions for Amazon EC2, see Actions in the *Amazon EC2 API Reference*.

If you prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS, AWS provides libraries, sample code, tutorials, and other resources for software developers. These libraries provide basic functions that automate tasks such as cryptographically signing your requests, retrying requests, and handling error responses, making it is easier for you to get started. For more information, see Tools to Build on AWS.

What is IAM?

PDFRSS

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *AWS General Reference*.

What is AWS Lambda?

Lambda is a compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging. With Lambda, you can run code for virtually any type of application or backend service. All you need to do is supply your code in one of the languages that Lambda supports.

Note

In the AWS Lambda Developer Guide, we assume that you have experience with coding, compiling, and deploying programs using one of the supported languages.

You organize your code into Lambda functions. Lambda runs your function only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time that you consume—there is no charge when your code is not running.

You can invoke your Lambda functions using the Lambda API, or Lambda can run your functions in response to events from other AWS services. For example, you can use Lambda to:

- Build data-processing triggers for AWS services such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.
- Process streaming data stored in Amazon Kinesis.
- Create your own backend that operates at AWS scale, performance, and security.

Lambda is a highly available service. For more information, see the AWS Lambda Service Level Agreement.

Sections

- Lambda features
- Getting started with Lambda
- Related services
- Accessing Lambda
- Pricing for Lambda

When should I use Lambda?

Lambda is an ideal compute service for many application scenarios, as long as you can run your application code using the Lambda standard runtime environment and within the resources that Lambda provides.

When using Lambda, you are responsible only for your code. Lambda manages the compute fleet that offers a balance of memory, CPU, network, and other resources to run your code. Because Lambda manages these resources, you cannot log in to compute instances or customize the operating system on provided runtimes. Lambda performs operational and administrative activities on your behalf, including managing capacity, monitoring, and logging your Lambda functions.

If you need to manage your own compute resources, AWS has other compute services to meet your needs. For example:

- Amazon Elastic Compute Cloud (Amazon EC2) offers a wide range of EC2 instance types to choose from. It lets you customize operating systems, network and security settings, and the entire software stack. You are responsible for provisioning capacity, monitoring fleet health and performance, and using Availability Zones for fault tolerance.
- AWS Elastic Beanstalk enables you to deploy and scale applications onto Amazon EC2. You retain ownership and full control over the underlying EC2 instances.

Lambda features

The following key features help you develop Lambda applications that are scalable, secure, and easily extensible:

Concurrency and scaling controls

Concurrency and scaling controls such as concurrency limits and provisioned concurrency give you fine-grained control over the scaling and responsiveness of your production applications.

Functions defined as container images

Use your preferred container image tooling, workflows, and dependencies to build, test, and deploy your Lambda functions.

Code signing

Code signing for Lambda provides trust and integrity controls that let you verify that only unaltered code that approved developers have published is deployed in your Lambda functions.

Lambda extensions

You can use Lambda extensions to augment your Lambda functions. For example, use extensions to more easily integrate Lambda with your favorite tools for monitoring, observability, security, and governance.

Function blueprints

A function blueprint provides sample code that shows how to use Lambda with other AWS services or third-party applications. Blueprints include sample code and function configuration presets for Node.js and Python runtimes.

Database access

A database proxy manages a pool of database connections and relays queries from a function. This enables a function to reach high concurrency levels without exhausting database connections.

File systems access

You can configure a function to mount an Amazon Elastic File System (Amazon EFS) file system to a local directory. With Amazon EFS, your function code can access and

Cloud Watch

Amazon CloudWatch is a monitoring and management service that provides data and actionable insights for AWS, hybrid, and on-premises applications and infrastructure resources. You can collect and access all your performance and operational data in the form of logs and metrics from a single platform rather than monitoring them in silos (server, network, or database). CloudWatch enables you to monitor your complete stack (applications, infrastructure, and services) and use alarms, logs, and events data to take automated actions and reduce mean time to resolution (MTTR). This frees up important resources and allows you to focus on building applications and business value.

CloudWatch gives you actionable insights that help you optimize application performance, manage resource utilization, and understand system-wide operational health. CloudWatch provides up to one-second visibility of metrics and logs data, 15 months of data retention (metrics), and the ability to perform calculations on metrics. This allows you to perform historical analysis for cost optimization and derive real-time insights into optimizing applications and infrastructure resources. You can use CloudWatch Container Insights to monitor, troubleshoot, and alert your containerized applications and microservices. CloudWatch collects, aggregates, and summarizes compute utilization information such as CPU, memory, disk, and network data, as well as diagnostic information such as container restart failures, to help DevOps engineers isolate issues and resolve them quickly. Container Insights gives you insights from container management services such as Amazon ECS for Kubernetes (EKS), Amazon Elastic Container Service (ECS), AWS Fargate, and standalone Kubernetes (k8s).

Collect and aggregate infrastructure and application metrics

Amazon CloudWatch allows you to collect infrastructure metrics from more than 70 AWS services, such as Amazon Elastic Compute Cloud (Amazon EC2), Amazon DynamoDB, Amazon Simple Storage Service (Amazon S3), Amazon ECS, AWS Lambda, and Amazon API Gateway, with no action on your part. For example, Amazon EC2 instances automatically publish CPU utilization, data transfer, and disk usage metrics to help you understand changes in state. You can use built-in metrics for API Gateway to detect latency, or use built-in metrics for AWS Lambda to detect errors or throttles. Likewise, Amazon CloudWatch also allows you to collect application metrics (such as user activity, error metrics or memory used) from your own applications to monitor operational performance, troubleshoot issues, and spot trends. You can use CloudWatch Agent or the PutMetricData API service call to publish these metrics to CloudWatch. If you need more detailed metrics beyond the default infrastructure metrics for example, such as shard-level Amazon Kinesis Data Streams metrics, you can simply opt in per resource. Similarly application metrics are available at up to one-second frequency and can be used in statistics, graphs, and alarms with high resolution.

Collect and aggregate container metrics and logs

Container Insights simplifies the collection and aggregation of curated metrics and container ecosystem logs. It collects compute performance metrics such as CPU, memory, network, and disk information from each container as performance events and automatically generates custom metrics used for monitoring and alarming. The performance events are ingested as CloudWatch Logs with metadata about the running environment, such as the Amazon EC2 instance ID, Service, and Amazon Elastic Block Store (Amazon EBS) volume mount and ID, to simplify monitoring and troubleshooting. CloudWatch custom metrics are automatically

extracted from these ingested logs and can be further analyzed using CloudWatch Logs Insights' advanced query language. Container Insights also provides an option to collect application logs (stdout/stderr), custom logs, predefined Amazon EC2 instance logs, Amazon EKS/k8s data plane logs, and Amazon EKS control plane logs. For Amazon EKS and k8s clusters, a preconfigured FluentD agent can be used to collect your logs. See the Container Insights logs setup documentation for more details. For Amazon ECS, the Amazon CloudWatch Logs logging driver or Fluent Bit can be used to collect application logs.

Collect and aggregate Lambda metrics and logs

CloudWatch Lambda Insights simplifies the collection and aggregation of curated metrics and logs from AWS Lambda functions. It collects compute performance metrics such as CPU, memory, and network from each Lambda function as performance events, while automatically generating custom metrics used for monitoring and alarming. The performance events are ingested as CloudWatch logs to simplify monitoring and troubleshooting. CloudWatch custom metrics are automatically extracted from these ingested logs and can be further analyzed using CloudWatch Logs Insights' advanced query language. See the Lambda Insights getting started documentation for more details.

Stream Metrics

Amazon CloudWatch Metric Streams enables you to create continuous, near-real-time streams of metrics to a destination of your choice. This makes it easier to send CloudWatch metrics to popular third-party service providers using an Amazon Kinesis Data Firehose HTTP endpoint. You can create a continuous, scalable stream including the most up-to-date CloudWatch metrics data to power dashboards, alarms, and other tools that rely on accurate and timely metric data. Easily direct your metrics to your data lake on AWS (such as on Amazon S3) and start analyzing usage or performance with tools such as Amazon Athena.

Monitor

Unified operational view with dashboards

Amazon CloudWatch dashboards enable you to create reusable graphs and visualize your cloud resources and applications in a unified view. You can graph metrics and logs data side by side in a single dashboard to quickly get the context and move from diagnosing the problem to understanding the root cause. For example, you can visualize key metrics, such as CPU utilization and memory, and compare them to capacity. You can also correlate the log pattern of a specific metric and set alarms to alert you to performance and operational issues. This gives you system-wide visibility into operational health and the ability to quickly troubleshoot issues, reducing MTTR.