

Jenkins SSO Authentication System

Project report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science and Engineering

By: Aman Tiwari

191294

Under the supervision of

Dr. Jagpreet Sidhu



Department of Computer Science & Engineering And

Information Technology

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,
WAKNAGHAT, SOLAN, HIMACHAL PRADESH – 173234**

Declaration

I hereby declare that the work presented in this report entitled “**Jenkins SSO Authentication System**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of **Dr. Jagpreet Sidhu** (Associate Professor(SG) in the Department of Computer Science and Engineering). The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)

Aman Tiwari

191294

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Dr. Jagpreet Sidhu

Associate Professor(SG)

Computer Science and Engineering

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
PLAGIARISM VERIFICATION REPORT

Date:

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: _____ Department: _____ Enrolment No _____

Contact No. _____ E-mail. _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> • All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String 		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck,juit@gmail.com

Acknowledgement

We are quite grateful to have this as the conclusion of our mission because it took a lot of perseverance and assistance from many people to complete and make such a significant effect on. All that we have accomplished is directly related to their oversight and assistance, for which we are grateful.

We honor and thank Dr. Jagpreet Sidhu for enabling us to work on the project at Jaypee University of Information Technology and for providing all of us with the guidance and instruction that enabled us to successfully finish the project. Even though he had involved time management, we are really thankful to him for providing such above-average support and guidance. We are also grateful towards Mr. Mohan Sharma and Mr. Ravi Raina, our respective lab coordinators who provided us with required resources for the successful accomplishment of our project.

We owe a huge debt of gratitude to our project manager Dr. Jagpreet Sidhu, who showed a genuine interest in our errand work and led the group of us until the project's conclusion by providing all the necessary information for developing a strong framework.

We are grateful and sufficiently honored to have received ongoing assistance, which enabled us to successfully complete our endeavor task. In a similar vein, we should extend our sincere gratitude to every one of them for their helpful assistance.

(Student Signature)

Aman Tiwari
191294

TABLE OF CONTENT

Content	Page No.
Declaration	i
Plagiarism Certificate	ii
Acknowledgement	iii
Table of Figures	iv
Table of Content	v
Abstract	vi
CHAPTER 1: INTRODUCTION	1-11
CHAPTER 2: RELATED WORK	12
CHAPTER 3: SYSTEM DEVELOPMENT	13-42
CHAPTER 4: EXPERIMENTS AND RESULT ANALYSIS	43-45
CHAPTER 5: CONCLUSIONS	46-48
REFERENCES	49

LIST OF FIGURES

Figure	Names
Figure 1	Watchguard logo
Figure 2	Watchguard Home Page
Figure 3	Watchguard Service Page
Figure 4	Watchguard Product Page
Figure 5	AWS logo
Figure 6	AWS IAM logo
Figure 7	AWS Admin logo
Figure 8	EC2 logo
Figure 9	Jenkins logo
Figure 10	SAML logo
Figure 11	AWS Create Account Page
Figure 12	AWS IAM Page
Figure 13	Create IAM User Page
Figure 14	Create Security Group Page
Figure 15	Create EC2 instance page1
Figure 16	Create EC2 instance page2
Figure 17	EC2 Instance connect page
Figure 18	Configure Jenkins Server
Figure 19	Installed Plugin Page
Figure 20	IAM IDP Application page
Figure 21	User for IAM IDP
Figure 22	Adding ACS URL and SAML audience
Figure 23	Adding IDP metadata URL
Figure 24	Final application with two Jenkins Server
Figure 25	Jenkins application accessed

Abstract

Organizations now frequently utilize Single Sign-On (SSO) authentication systems because they streamline user access control and improve user experience. The implementation of an SSO authentication system utilizing the Jenkins SAML plugin and AWS Identity Provider (IdP) is the main goal of our project. Users no longer need to keep track of separate login credentials for Jenkins because the solution enables them to log in to Jenkins using their AWS credentials.

The installation of Jenkins server on an EC2 instance is preceded by the setup of an IAM user with admin permissions and a security group that permits traffic from port 8080. The Jenkins application URL and the AWS IdP metadata URL are then used to install and configure the SAML plugin. Then, we build an application on the AWS IdP and include a user in it. Finally, we establish Jenkins' Global Security settings with the URLs for IdP's metadata and logout.

In our experiment, we used multiple user accounts to evaluate the SSO authentication method. The outcomes demonstrated that the system was successful in giving all users safe and convenient access to Jenkins.

Applications for this project include bettering user access control, boosting the user experience, and streamlining the login process. The project's future scope includes extending the SSO authentication mechanism to additional platforms and apps.

Chapter 1: - INTRODUCTION

1.1 Introduction

WatchGuard has developed cutting-edge cybersecurity technology over the past 25 years and has made it available as simple-to-deploy and simple-to-manage solutions. More than 250,000 small and midsize businesses from all over the world can protect their most valuable assets, including more than 10 million endpoints, thanks to WatchGuard's industry-leading network and endpoint security, secure Wi-Fi, multi-factor authentication, and network intelligence products and services. WatchGuard makes enterprise-grade cybersecurity solutions available to any firm in a world where the cybersecurity landscape is continuously changing, and new threats are appearing every day. WatchGuard has offices across North America, Europe, Asia Pacific, and Latin America, with its headquarters in Seattle, Washington.



Figure 1: - Watchguard logo

WatchGuard believes that our UTM equipment should and can include any number and type of security services. The business has a history of opening the door for novel, ground-breaking services that became the standard for their sector. The largest selection of network security services are provided by WatchGuard, from basic IPS, URL filtering, gateway AV, application control, and antispam to services for fending off more sophisticated threats like file sandboxing, ransomware protection, and others. You pick complete security when you choose WatchGuard.

It has long been questioned whether a single appliance that centralizes numerous network security operations could possibly match the effectiveness of a focused point solution. Not only is the answer yes, but WatchGuard UTM appliances perform better than many other dedicated NGFW point solutions and competitor UTM appliances when all security engines are active. Performance is what you pick when you choose WatchGuard. Check out the most recent NetSecOPEN reports and Miercom performance tests instead of taking our word for it.

Information and data are distinct concepts at WatchGuard. Due to their hectic schedules, many of our customers lack the time or the devoted staff to sort through mountains of log data in order to make conclusions and locate crucial information. WatchGuard Cloud gives you complete network awareness so you can decide on network security whenever and wherever you want in real time and with the necessary knowledge. The software shows more than 100 dashboards and reports that let you immediately spot broad trends and abnormalities before drilling down into more specific details on each.

In WatchGuard Cloud, creating policies and deploying VPN connections is simple. You can get up and running quickly and securely with pre-configured policies for content filtering, VPNs, network inspection, and scanning services. As a security best practices, you can also easily construct network segments to keep things like VoIP systems or IoT devices apart from your mission-critical applications.

WatchGuard provides a solution that is suitable for your environment, regardless of whether you operate a single store, a small chain of eateries, or thousands of retail outlets globally. We provide a comprehensive range of Firebox appliances, including physical tabletop and 1U rack-mounted variants, as well as virtualized and Cloud-based options.

Without sacrificing WatchGuard's reputation for offering enterprise-level security, firebox administration and network setup in the Watchguard Cloud are simple to set up and configure for multiple customers and diverse networks. Your team can achieve the degree of security your clients want while spending less time on processes and more time on profitability.

Deploying VPNs and creating policies are simple processes. You can get up and running quickly and securely with pre-configured policies for content filtering, VPNs, network inspection, and scanning services. You can adjust and create policies in advance because they

can be built offline and deployed when the time is perfect. Additionally, you may make modifications in bulk and construct policy templates for quick, repeatable deployment across numerous sites.

WatchGuard Cloud can simplify network configuration. It's simple to create network segments, and it's a security recommended practice to keep things like VoIP systems and IoT devices apart from your mission-critical services. Establish SD-WAN policies that employ dynamic path selection to pick the best route for your traffic.

No need to dispatch your staff to the client's location! You can set up your Fireboxes in minutes rather than hours thanks to WatchGuard Cloud's full integration of zero-touch deployment. To cause as little inconvenience as possible for your consumers, you can even plan and roll out the most recent firmware updates outside of regular business hours.

Sending pre-scheduled or ad-hoc reports to important stakeholders in your organization on a variety of subjects, such as threat trends, bandwidth use, compliance status, and much more, can make it easier for you to show how well your service delivery is doing.

Sending pre-scheduled or ad-hoc reports to important stakeholders in your organisation for delivery can help you more effectively demonstrate the effectiveness of your service delivery. These reports can cover a wide range of subjects, including threat trends, bandwidth utilization, compliance status, and much more.



[Try Now](#)

Unleash the Security of ONE

[Learn More](#)

Are you an MSP?

[Elevate Your Security Practice with the Security of ONE](#)



Network Security



Multi-Factor Authentication



Secure Cloud Wi-Fi



Endpoint Security

What's New at WatchGuard



[Access the XDR Realm with](#)

Figure 2: - Watchguard Website Home logo

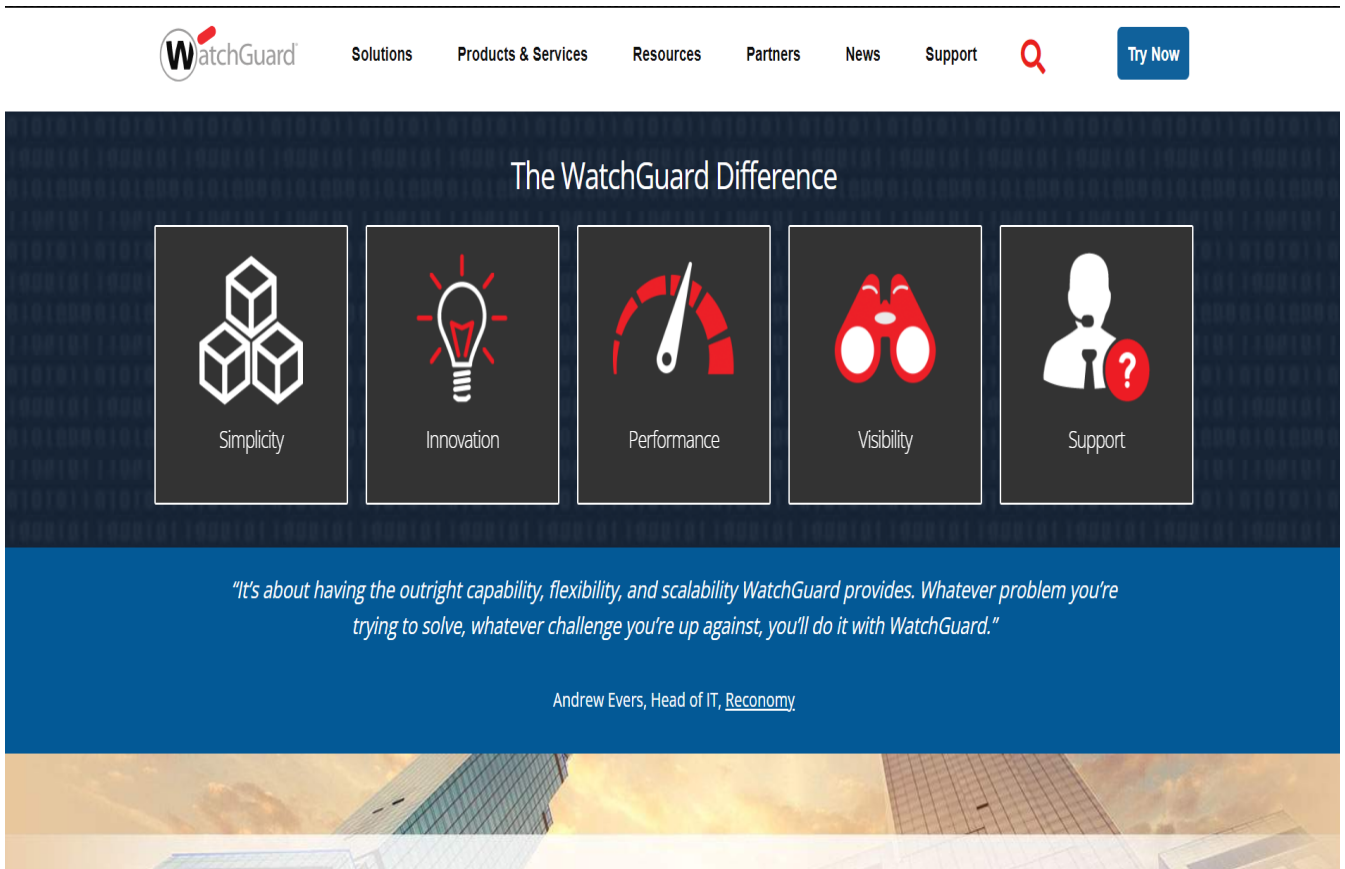


Figure 3: - Watchguard Website services page

The WatchGuard Cloud platform allows you to plan and deploy the most recent Firebox firmware. These updates will be regularly downloaded to all of your Firebox appliances from a global Content Delivery Network (CDN).

You can rebrand select WatchGuard Cloud services using the custom branding functionality. To strengthen your brand and direct clients to your support team, just add your business' logo, photos, and contact details to emails, reports, identity portals, and other places where they may show.

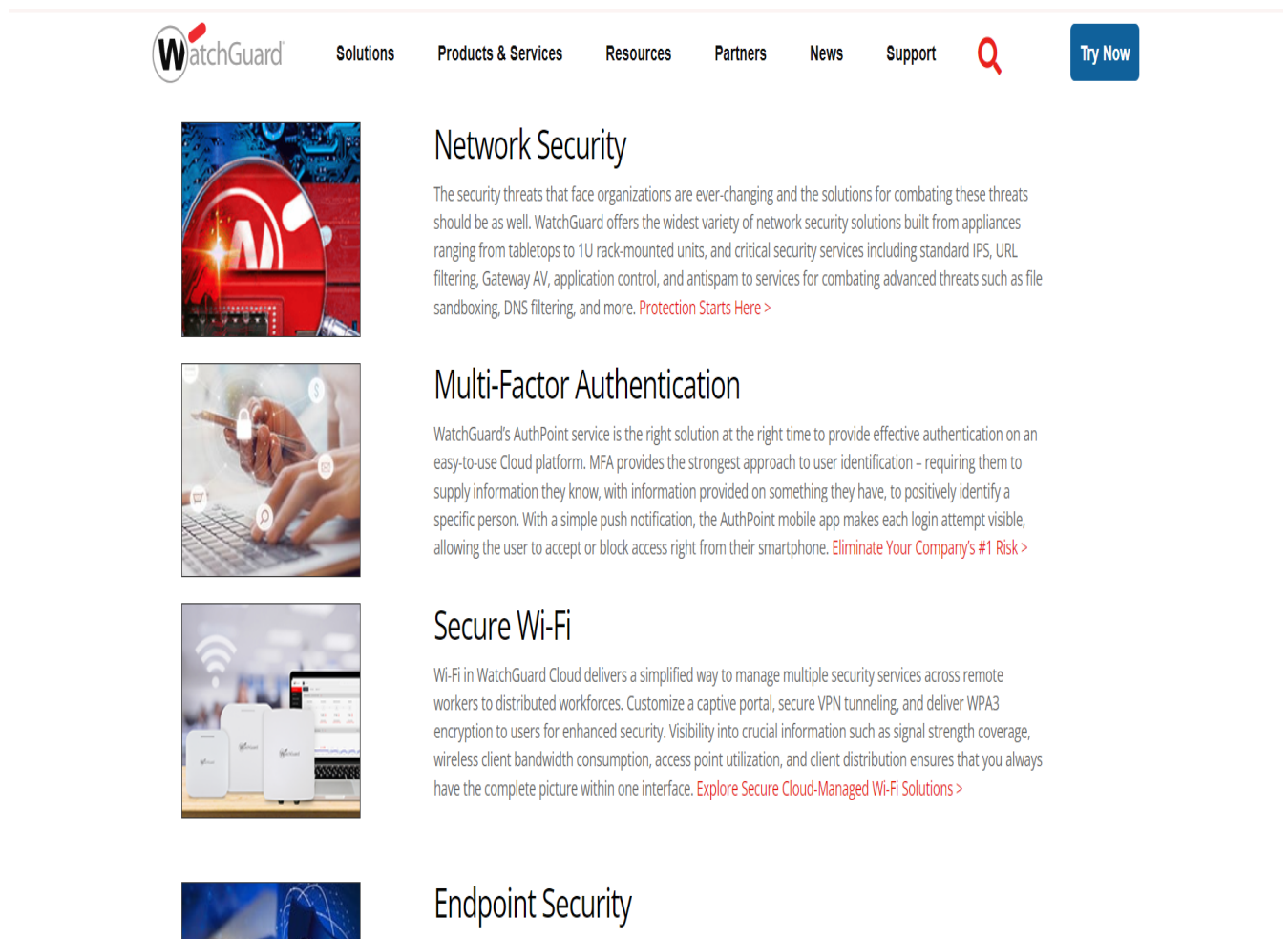
The requirement to deploy or manage infrastructure is decreased by the WatchGuard Cloud platform. With no disruption to continuing business operations, all upgrades and maintenance


are automatically applied. The platform adapts dynamically to shifting compute and storage needs, maintaining high performance and scaling as requirements vary.


Strong multi-factor authentication for VPNs may now be set up using the Firebox without the need for RADIUS setup. This integration is a potent tool for zero-trust systems because of its ease and security.

Now, strong multi-factor VPN authentication can be set up using the Firebox without any RADIUS

configuration. This integration's simplicity and security make it a powerful tool for zero-trust systems.

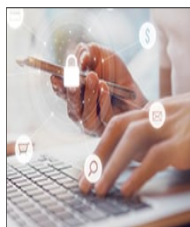


WatchGuard Solutions Products & Services Resources Partners News Support  [Try Now](#)



Network Security

The security threats that face organizations are ever-changing and the solutions for combating these threats should be as well. WatchGuard offers the widest variety of network security solutions built from appliances ranging from tabletops to 1U rack-mounted units, and critical security services including standard IPS, URL filtering, Gateway AV, application control, and antispam to services for combating advanced threats such as file sandboxing, DNS filtering, and more. [Protection Starts Here >](#)



Multi-Factor Authentication

WatchGuard's AuthPoint service is the right solution at the right time to provide effective authentication on an easy-to-use Cloud platform. MFA provides the strongest approach to user identification – requiring them to supply information they know, with information provided on something they have, to positively identify a specific person. With a simple push notification, the AuthPoint mobile app makes each login attempt visible, allowing the user to accept or block access right from their smartphone. [Eliminate Your Company's #1 Risk >](#)



Secure Wi-Fi

Wi-Fi in WatchGuard Cloud delivers a simplified way to manage multiple security services across remote workers to distributed workforces. Customize a captive portal, secure VPN tunneling, and deliver WPA3 encryption to users for enhanced security. Visibility into crucial information such as signal strength coverage, wireless client bandwidth consumption, access point utilization, and client distribution ensures that you always have the complete picture within one interface. [Explore Secure Cloud-Managed Wi-Fi Solutions >](#)



Endpoint Security

Figure 4: - Watchguard Website Product Page

In order to prevent further data loss and network outages, crucial decisions regarding network security situations must be taken rapidly from the boardroom to the branch office. However, there isn't enough time to interpret days or weeks' worth of unprocessed data. In order to receive the information you need and move forward right away network security solutions must include a visibility dashboard of this kind as a necessary component.

1.2 Introduction To Project

Accessing multiple applications and services with a single set of credentials (known as Single Sign-On or SSO) is an authentication mechanism that can benefit users in numerous ways. No longer required to remember distinct usernames and passwords, users can now access several applications or services with ease, resulting in a better user experience and increased productivity.

Using a centralized authentication server, SSO enables user identity verification and access provision to multiple applications and services. After logging in with their SSO credentials, users can view authorized apps sans the need for login information re-entry.

Different ways exist for implementing SSO, including the usage of industry standards similar to SAML or OAuth, along with proprietary protocols. Alongside, some SSO solutions provide extra security aspects like session management or multi-factor authentication.

Throughout the workday, computer users are often forced to jump between different applications and services. This can result in annoyance and time-wasting when they're forced to memorize and submit individual access credentials for each system they use. SSO offers a solution by providing one straight-forward login experience that simplifies the authentication process and lightens the cognitive burden on users. Nowadays, this digital terrain is quite common, and it's essential to seek out ways to streamline the user experience.

Organizations find managing user accounts and access control much easier with SSO's consolidation of authentication to a central server. By simplifying the process of revoking

access when an employee leaves the organization and reducing the risk of password reuse, it can also improve security.

Enterprise software, cloud-based services, and consumer-facing applications are just a few of the many sectors and applications that use SSO. Depending on the organization's security requirements and the unique use case, it can be implemented utilizing a variety of protocols and technologies.

1.3 Problem Statement

The major Automation solutions like Jenkins have become crucial for managing continuous integration and deployment pipelines as firms increasingly adopt DevOps practices to enhance their software development and delivery. To effectively administer and monitor these systems, as well as to provide security and easy access, many Jenkins controllers and servers must be in operation.

The current issue is that two Jenkins masters, controllers, or servers must be created on a local machine using Docker, and the master-slave model of these two Jenkins servers, Jenkins CLI, and Remote Access API must all be investigated. Finding ways to log onto both Jenkins controllers using a single session login is also necessary.

The project will investigate the Master-slave paradigm of two Jenkins servers, which enables distributed builds and centralized management of jobs and nodes, to address this issue. Administrators will be able to efficiently administer and keep an eye on numerous Jenkins servers from a single location thanks to this.

The project will also look into using Jenkins CLI and Remote Access API, which let programmers access Jenkins servers and automate numerous jobs and procedures.

Finding ways to log into both Jenkins controllers using a single session login will be the project's last goal. By encouraging best practices in password management, this will simplify the login process, lessen the need for numerous logins and credentials, and improve security.

By using a single session login strategy, this project seeks to improve security, increase productivity, and streamline the operation of numerous Jenkins servers. Businesses will be able to manage their DevOps procedures more effectively as a result, and their software delivery pipelines will be reliable and secure.

1.4 Objectives

The main goals of this project are to use Docker to create and manage two Jenkins masters, controllers, and servers on a local system, investigate the Master-slave architecture of two Jenkins servers, Jenkins CLI, and Remote Access API, and discover approaches for simultaneously logging into both Jenkins controllers.

The following sub-objectives will be prioritized by the project in order to accomplish this main goal:

- Install and set up two Jenkins servers using Docker on a local system.
- Make sure the local system complies with the essential requirements by researching the system requirements for running Jenkins on Docker.
- Docker must be installed and configured locally.
- The most recent Jenkins photos can be downloaded and used.
- Configure and personalize the Jenkins instances as required, including task setups, security options, and plugins.
- The main goals of this project are to use Docker to create and manage two Jenkins masters, controllers, and servers on a local system, investigate the Master-slave architecture of two Jenkins servers, Jenkins CLI, and Remote Access API, and discover approaches for simultaneously logging into both Jenkins controllers.
- Use two Jenkins servers in a master-slave configuration to offer distributed builds and centralised management.
- Set up the two Jenkins servers so that one serves as the master and the other as the slave in a master-slave configuration.
- To enable distributed builds across the two servers, configure the required plugins and settings.

- Across both servers, provide centralised management of jobs and nodes to make management and monitoring easy.
- Find and compare various approaches to create a single session login strategy for both Jenkins servers.
- Investigate several implementation strategies for a single session login technique, such as SSO or OAuth.
- Analyse the effects of each strategy on security and usability.
- Choose the best implementation strategy for both Jenkins servers' single session login technique.
- Create a method for implementing single session logins that complies with security best practises and improves productivity.
- Use the chosen implementation strategy to implement the single session login strategy.
- Set up the required security parameters to guarantee safe access to both Jenkins servers.
- Create a unique plugin or script that will allow both Jenkins servers to be logged in with a single session.
- To ensure dependability and functionality, test and validate the solution.
- To verify dependability and functionality, test and validate the single session login strategy on the two Jenkins servers.
- Perform thorough functional, security, and usability testing on both Jenkins servers to evaluate the single session login technique.
- Verify the solution's dependability and functionality, and then make any necessary modifications or enhancements.

1.5 Methodology

AWS Configuration and Setup

Two Jenkins servers on AWS must be set up and configured as part of the project's second phase. This entails setting up security groups to regulate network traffic, installing Jenkins on the instances, and establishing two Amazon Elastic Compute Cloud (EC2) instances. The instances will be set up to allow for secure access using an SSL certificate and a custom domain name.

SAML implementation for SSO

The main goal of this phase is to set up SAML-based SSO for the two Jenkins instances. Setting up a SAML identity provider (IdP), configuring Jenkins to use SAML for authentication, and installing AWS Single Sign-On (SSO) are all required. Users will then be able to sign in to both Jenkins servers using a single account after this is complete.

Putting two Jenkins servers' Master-Slave architecture into practise

The next step is to set up the two Jenkins servers such that one serves as the master and the other as the slave in a master-slave configuration. The required plugins and settings will be set up to allow distributed builds over the two servers, and both servers will use centralised administration of jobs and nodes.

Examining how to automate chores and processes using the Jenkins CLI and Remote Access API: In this stage, we'll investigate how to automate numerous tasks and procedures, including job creation, build triggering, and monitoring, using Jenkins CLI and Remote Access API. Both Jenkins servers will be set up with the Jenkins CLI and Remote Access API.

Determining the best implementation strategy for a single session login approach across both Jenkins servers and assessing it.

Research and testing several approaches to creating a single session login technique, such as SSO or OAuth, will be the focus of this phase. Each technique's effects on security and usability will be assessed, and the best strategy for deploying a single session login approach for both Jenkins servers will be chosen. Constructing a method to create a single session login

CHAPTER 2 RELATED WORK

With the help of Single Sign-On (SSO), users can log in only once to access a variety of programmes or services without having to do so again. It streamlines the user experience and eases the stress of keeping track of numerous passwords. We will examine many facets of SSO authentication systems in this literature review, including their advantages, structures, and protocols.

SSO has several advantages, one of which is that it improves user experience by cutting down on the number of times users must verify themselves. Additionally, it makes it simpler for organisations to control user access and account management. SSO enables businesses to deploy more stringent authentication procedures like multi-factor authentication without degrading user experience. SSO can also increase security by minimising the amount of passwords users need to remember and lowering the chance of password re-use.

SSO Architectures:

There are three main architectures for SSO: federation-based, web-based, and client-based. Federation-based SSO involves establishing trust between identity providers (IdPs) and service providers (SPs). Web-based SSO uses browser cookies or HTTP headers to share authentication information between applications. Client-based SSO involves installing a client application on the user's device that handles authentication and passes tokens to applications. Each architecture has its own advantages and disadvantages, and the choice of architecture depends on the specific requirements of the organization.

SSO Protocols:

For SSO authentication, a number of protocols are utilized, including OAuth 2.0, OpenID Connect, and Security Assertion Markup Language (SAML). IdPs may submit assertions to SPs and authenticate users using the XML-based SAML protocol. OIDC is a cutting-edge technology that provides identity data and user authentication using JSON Web Tokens (JWTs). By using the OAuth 2.0 protocol, users may grant access to third-party applications to their resources without disclosing their login information.

CHAPTER 3

SYSTEM DESIGN & DEVELOPMENT

3.1 Related Terms

3.1.1 AWS



Figure 5: - AWS logo

A comprehensive cloud computing platform is offered by Amazon.com under the name Amazon Web Services (AWS). Businesses can design, deploy, and manage their applications and infrastructure on the cloud using a wide range of cloud-based services and technologies.

Businesses can utilize AWS's infrastructure services to run their applications and services in the cloud. These services include computational power, storage capacity, and networking. As a result, companies can scale their infrastructure rapidly and easily without worrying about the purchase or upkeep of hardware. AWS is a cloud computing platform that offers a variety of features and services to assist businesses in quickly and easily developing and deploying apps and services. It is a highly flexible, cost-effective, and scalable solution that enables companies to take use of cloud computing to enhance their IT infrastructure, lower expenses, and boost agility.

AWS offers a variety of platform services in addition to infrastructure services, including databases, analytics, and machine learning, which companies may utilise to create complex

apps and services. With the use of these services, organisations can easily and quickly develop and deploy apps by utilising the power of cloud computing.

AWS offers businesses the ability to put their apps and services closer to their users for enhanced performance and decreased latency thanks to its global infrastructure that spans numerous regions and availability zones. Because of this, AWS is a great platform for companies who must offer their apps and services to a global clientele.

With a variety of security features and certifications, including as ISO 27001, SOC 1, SOC 2, and PCI DSS, AWS is very secure and compliant. AWS also offers a variety of tools and services for managing security, including AWS Key Management Service (KMS) and AWS Identity and Access Management (IAM).

AWS provides more than 200 fully featured services, including computing, storage, databases, analytics, machine learning, networking, mobility, developer tools, security, and enterprise applications, from data centres around the world. This makes it a very flexible and all-encompassing platform for companies of all sizes and sectors.

Pay-as-you-go, reserved instances, and spot instances are just a few of the numerous pricing tiers that AWS offers for its services, allowing companies to tailor their expenses to their usage and requirements. In order to assist companies in tracking and maximizing their expenses, AWS also offers cost optimisation tools and services like AWS Cost Explorer and AWS Trusted Advisor.

Independent software vendors (ISVs), system integrators (SIs), and managed service providers (MSPs), who provide a wide range of solutions and services that complement AWS, are just a few of the partners that make up AWS's sizable and expanding ecosystem. As a result, finding the ideal partner and solution for their needs is made simple for organisations.

AWS places a high priority on innovation and frequently adds new features and services on its platform. This includes products and services like AWS Lambda, which lets companies run code without setting up or managing servers, and Amazon SageMaker, which lets them develop, test, and scale machine learning models.

AWS offers a variety of tools and services, such as AWS CloudFormation, AWS Elastic Beanstalk, and AWS OpsWorks, for managing and automating infrastructure. With the use of these technologies, businesses can manage and deploy their infrastructure and apps as code, enhancing their overall agility while lowering mistakes and processing times.

Finally, AWS offers a variety of tools and services, such as AWS Security Hub, AWS Config, and AWS Audit Manager, for managing security and compliance. These technologies help companies keep ahead of possible attacks and compliance issues by allowing them to monitor and manage their security and compliance posture in real-time.

Overall, AWS is a very flexible and strong platform that gives companies a huge selection of tools, services, and capabilities to design, deploy, and manage their cloud-based infrastructure and applications. AWS is a great option for companies looking to use cloud computing to power their initiatives for digital transformation because of its strong emphasis on innovation, security, and cost optimisation.

3.1.2 Identity Access Management

A Web service called AWS Identity and Access Management (IAM) enables companies to safely control who has access to AWS resources. Businesses may set up and manage AWS users and groups as well as grant access to AWS resources as necessary thanks to IAM. IAM also interfaces with numerous other AWS services, including Amazon S3, Amazon EC2, and AWS Lambda, to give users granular control over the resources and actions they may access within those services.



Figure 6: - AWS IAM Image

IAM offers a number of tools to assist companies in controlling access to their AWS resources. IAM, for instance, enables organizations to create and manage IAM users and groups as well as utilise policies to provide those users and groups access to resources. IAM policies are JSON documents that outline the rights a person or group has to use particular resources and perform particular operations on them. IAM also offers capabilities like IAM roles, which let companies grant trusted organisations, such AWS services, access to AWS resources without disclosing their AWS login credentials. Jupyter Notebook offers many features that make it a

popular choice for data scientists and researchers. One of its most useful features is its ability to display data visualizations directly in the notebook.

To assist businesses in securing their AWS resources, IAM offers a number of security capabilities. In order to access AWS resources, for instance, organisations can utilise multi-factor authentication (MFA) to demand that users provide two forms of authentication. IAM also offers capabilities like password policies, which let businesses define the level of password difficulty and the number of days before a password expires for their IAM users.

IAM connects with numerous additional AWS services to offer a simple and secure method of controlling access to AWS resources. For instance, IAM interfaces with Amazon S3 to give companies access control over their S3 objects and buckets. To help companies monitor, IAM also connects with AWS CloudTrail, which offers a thorough history of all API calls made to AWS services.

The cornerstone of AWS security and access control is IAM. With granular access control and permissions management, it enables enterprises to develop and manage a variety of AWS resources safely. IAM has the following salient qualities and abilities:

IAM enables organisations to create and manage IAM users and groups, which can be used to restrict access to AWS resources. IAM users are individual AWS accounts to which specific access keys and credentials may be issued. IAM groups are sets of users to which policies may be assigned to limit their access to and use of particular resources.

IAM policies are JSON documents that outline the permissions that IAM users and groups may exercise. Users or groups can be granted access to particular AWS resources and operations by attaching policies to them. IAM policies can be constructed in such a way as to offer incredibly detailed control over resource access.

Roles: IAM roles allow organisations to grant trustworthy entities, such as AWS services or other AWS accounts, access to AWS resources. IAM allows for the creation and management of roles, and roles are given policies that limit their access to AWS resources.

IAM offers support for multi-factor authentication (MFA), which boosts security for access to AWS resources. Users may be asked for additional authentication beyond their password, such as a code generated by a hardware token or mobile app.

Integration with AWS CloudTrail: IAM and AWS CloudTrail work together to provide full records of API calls performed to AWS resources. Auditing resource access, locating security events, and troubleshooting may all be done using CloudTrail logs.

3.1.3 IAM Admin

An AWS account's IAM admins can control access to all of the account's AWS resources since they have complete administrative rights over IAM. IAM administrators are in charge of setting up policies to restrict access to AWS resources as well as creating and managing IAM users, groups, and roles.

An IAM admin is responsible for a variety of duties, such as:

IAM administrators are in charge of generating and administering IAM users and groups inside the AWS account. This entails creating user accounts, providing access keys and credentials, and establishing policies to manage users' access to AWS resources.

Access control delegation: IAM administrators can control access to AWS resources by establishing and maintaining IAM roles. Policies can be assigned to roles to limit their access to and use of particular resources. IAM administrators can grant trustworthy entities, such as AWS services or other AWS accounts, access as well.

IAM administrators are in charge of establishing and maintaining IAM policies, which are JSON documents that outline rights for IAM users and groups. Users or groups can be granted access to particular AWS resources and operations by attaching policies to them.



Figure 7: - AWS Admin Logo

Monitoring and auditing resource access: IAM administrators are in charge of continuously monitoring and recording resource access to AWS. This include checking CloudTrail logs for security incidents, fixing problems, and confirming regulatory compliance.

Managing security and compliance: IAM administrators are in charge of making sure the AWS account is secure and compliant. This entails creating password restrictions, setting up MFA for IAM users, and ensuring regulatory compliance.

3.1.4 IAM User



Figure 8: - User data

We can securely manage access to AWS services and resources with the help of AWS Identity and Access Management (IAM). IAM users are generated entities that are used to represent individuals or services that need to interact with the AWS environment. They are formed within an AWS account. IAM users can access the resources they are permitted to use by using a specific name and a set of credentials (access key and secret access key) that are attached to them.

IAM administrators, who have the authority to add, edit, and remove IAM users, can create and manage IAM users. IAM administrators can use policies to give IAM users permissions. These policies specify which resources and what actions the IAM user is permitted to carry out.

IAM groups are another way to organise a group of IAM users. Collections of IAM users with the same permissions are referred to as IAM groups. Instead of giving permissions to each user separately, this makes it simple to handle permissions for several users at once.

We can finely regulate access to your AWS environment with IAM users, which is one of its key advantages. IAM users can be created with just the permissions they require to carry out their duties, and you can withdraw those permissions as soon as they are no longer required. This lessens the possibility of unauthorised access to your AWS environment and aids in maintaining security policy compliance.

Utilising IAM users also allows you to monitor user behaviour inside your AWS environment. All API calls made by IAM users are recorded in detail by AWS CloudTrail, including the user who made the call, the time it was made, and the resources accessed. This enables you to audit user behaviour and, if necessary, look into security incidents.

IAM users, to put it briefly, are entities formed within an AWS account to represent individuals or services that must communicate with the AWS environment. IAM users can access the resources they are permitted to use by using a specific name and a set of credentials that are attached to them. IAM groups can be created from a collection of IAM users, and policies can be used to provide each group the appropriate permissions. IAM users allow you to track user behaviour and give a fine-grained level of access control to your AWS environment.

3.1.6 Elastic Cloud Compute



Figure 9: - EC2 Logo

Users can launch and operate virtual machines, sometimes referred to as instances, in the cloud using Amazon Elastic Compute Cloud (EC2), a cloud-based computing service offered by Amazon Web Services (AWS). A key component of AWS, EC2 offers consumers flexible, scalable, and affordable computing resources on demand.

The AWS Management Console, AWS Command Line Interface (CLI), or AWS SDKs make it simple to launch and manage EC2 instances, which are virtual machines. Users have access to a wide variety of instance types, each of which is tailored for a particular workload, such as applications that are compute-, memory-, or storage-intensive.

Users can deploy EC2 instances in a variety of locations and availability zones, giving them access to a wide geographic area and high availability. By choosing multiple operating systems, storage configurations, and security settings, users can further customise their instances.

Scalability is one of EC2's main advantages. Without paying any upfront charges, users may quickly scale up or decrease their processing capacity as needed. Users can swiftly adjust to fluctuating business needs and demand spikes as a result, and they only pay for what they really use.

Additionally, EC2 offers its users a vast array of networking features, such as elastic IP addresses, virtual private clouds (VPCs), and load balancing. With the help of these characteristics, users may create fault-tolerant, highly available cloud apps while still maintaining control over their network infrastructure.

Additionally, EC2 readily connects with other AWS services including AWS Lambda for serverless computing, Amazon S3 for object storage, and Amazon RDS for relational databases. Users may now quickly create sophisticated, scalable apps in the cloud without having to worry about infrastructure maintenance thanks to this.

3.1.7 Jenkins



Figure 10: - Jenkins Logo

Continuous Integration and Continuous Delivery (CI/CD) of software applications employ Jenkins, an open-source automation server. Since its initial release in 2011, it has grown to be one of the most well-liked automation servers in the market.

Windows, Linux, and macOS are just a few of the operating systems on which Jenkins may be deployed. Jenkins is a web-based platform. It offers a comprehensive collection of tools that let users automate every step of the software development lifecycle, from code creation to testing, building, and deployment.

Jenkins' extensibility is one of its most important benefits. Jenkins includes a sizable plugin library that may be used to enhance its features and link it with other programmes and services, like AWS, Docker, GitHub, and others. Jenkins is therefore the best option for groups looking to create a tailored automation pipeline to suit their unique requirements.

Additionally, Jenkins has a master-slave design that enables customers to split their construction effort among several computers. The automation pipeline's efficiency can be greatly increased by using this architecture, especially when working with complex applications.

Jenkins also supports a number of additional programming languages, including Java, Python, Ruby, and others. Jenkins is now a flexible automation server that may be utilised for a variety of software development projects as a result.

Additionally, Jenkins includes a CLI interface that enables command-line communication with the server. This can be especially helpful for users who wish to automate parts of their Jenkins operations or prefer to operate in a terminal environment.

Jenkins also offers a strong security model with role-based access control (RBAC), which enables administrators to provide particular permissions to users in accordance with their roles. Having only authorised individuals access to critical information and resources can assist organisations.

Jenkins is an all-around strong and adaptable automation server that may greatly increase the effectiveness and dependability of software development operations. It is the perfect option for businesses wishing to automate their software development processes because to its extensive feature set, extensibility, and strong security approach.

3.1.8 SAML



Figure 11: - SAML Logo

An open standard for exchanging authentication and authorization data, namely between an identity provider (IdP) and a service provider (SP), Security Assertion Markup Language (SAML) is based on XML. Single Sign-On (SSO) capabilities for online applications across various domains is the primary objective of SAML.

The IdP and SP communicate with one another via a series of XML-based messages to implement SAML. The SAML assertion contains details about the user's identity and attributes after the user first authenticates with the IdP. The SP utilizes this assertion to allow access to the requested resources after it is sent to it.

In workplace settings, SAML is frequently used to enable SSO across many web apps. Additionally, it's used in cloud environments like Amazon Web Services (AWS) to support federated resource access.

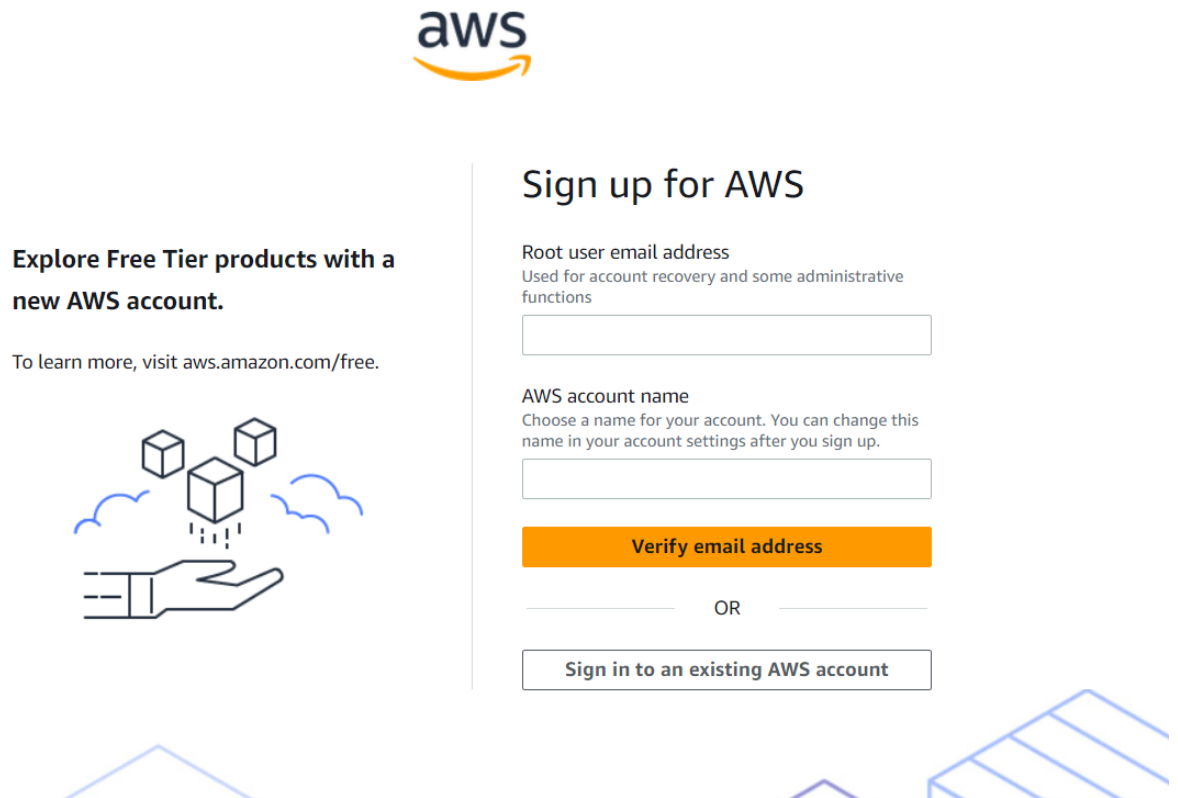
Users can access AWS resources using their current corporate credentials thanks to SAML in AWS. In order to do this, set up an Identity Provider (IdP) that can produce SAML assertions, and set up AWS to accept these assertions as a form of authentication.

3.2 Development

3.2.1-Create an AWS account

Click the "Create an AWS Account" option on the AWS webpage first. Both your email address and a password must be entered. You will then be asked to provide your personal information and confirm your agreement to the terms and conditions.

The next step is to input your payment details. AWS offers a free tier account, but you will be charged if you go above its allowances. Direct debit or using a credit card are both options.



aws

Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.

Sign up for AWS

Root user email address
Used for account recovery and some administrative functions

AWS account name
Choose a name for your account. You can change this name in your account settings after you sign up.

Verify email address

OR

Sign in to an existing AWS account

Figure 12: - AWS Create Account Page

We need to prove our identification after providing your payment details. You can accomplish this by sending a text message, making a phone call, or downloading a photo ID from the government. You'll be directed to the AWS Management Console after your identification has been confirmed.

We may build and manage your AWS resources through the AWS Management Console. To manage your resources, you must create an IAM user with the necessary permissions and pick the area where they will be housed.

3.2.2 - Create an IAM User with Admin Permission account

Step 1: Create an AWS account.

To get started, an AWS account must be made. Click the "Create an AWS Account" option on the AWS webpage. Follow the instructions to verify your account after completing the registration form with your personal and financial details.

Step 2: Access the IAM Console in step two.

You must enter the IAM console to create a new user after creating an AWS account. Enter "IAM" into the search field in the AWS Management Console. When the IAM option displays, select it.

The screenshot displays the AWS IAM dashboard. On the left is a navigation sidebar for Identity and Access Management (IAM) with sections for Access management (User groups, Users, Roles, Policies, Identity providers, Account settings) and Access reports (Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, Service control policies (SCPs)). The main content area is titled 'IAM dashboard' and features 'Security recommendations' with three items: 'Add MFA for root user' (warning), 'You have MFA' (success), and 'Your user, Demo-User, does not have any active access keys that have been unused for more than a year.' (success). A 'View affected policies' button is visible. Below this is the 'IAM resources' section with a table:

User groups	Users	Roles	Policies	Identity providers
1	1	20	25	0

On the right, the 'AWS Account' section shows account details like ID (729085594219) and alias (729085594219). Below that are 'Quick Links' for security credentials and a 'Tools' section with a 'Policy simulator' link.

Figure 13: - AWS IAM Page

Step 3: Form a group

The following step is to establish a new group. A group is an affiliation of users with comparable access rights. Click the "Groups" option in the left-hand menu, then select "Create New Group" to start a new group. Choose the permissions you want the group to have after giving it a name. You should choose the "AdministratorAccess" permission for an admin user.

Step 4: Make a User

After creating a group, you can go ahead and create a new user and include them in the group. Click the "Users" link in the left-hand menu, then select the "Add User" option to add a new user. Select the "Programmatic access" and "AWS Management Console access" checkboxes after entering the user's name.

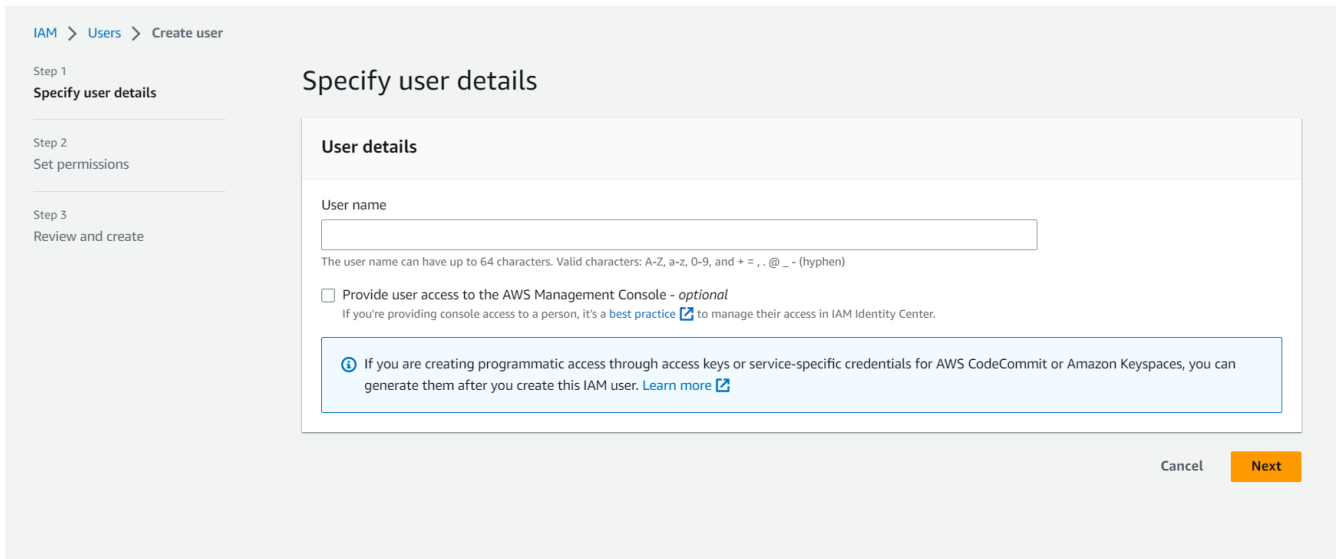


Figure 14: - AWS Create IAM User Page

Step 5: Add User to Group

You must include the user in the group you made in Step 3 after creating them. Click the user's name in the list of users, then click the "Add user to group" button to add the user to the group. After selecting the group you made in Step 3, click "Add user to group."

Step 6: Create security credentials

You must now produce security credentials for the new user you just established and added to a group with admin access. Click the user's name in the list of users, then click the "Security credentials" tab, to accomplish this. To generate a fresh access key and secret access key, click the "Create access key" button.

Step 7: Log in using IAM User

The final step is to utilize the new IAM user to log into the AWS Management Console. Enter the IAM user's credentials on the AWS Management Console login screen. If multi-factor authentication is enabled, you must input the code produced by your MFA device.

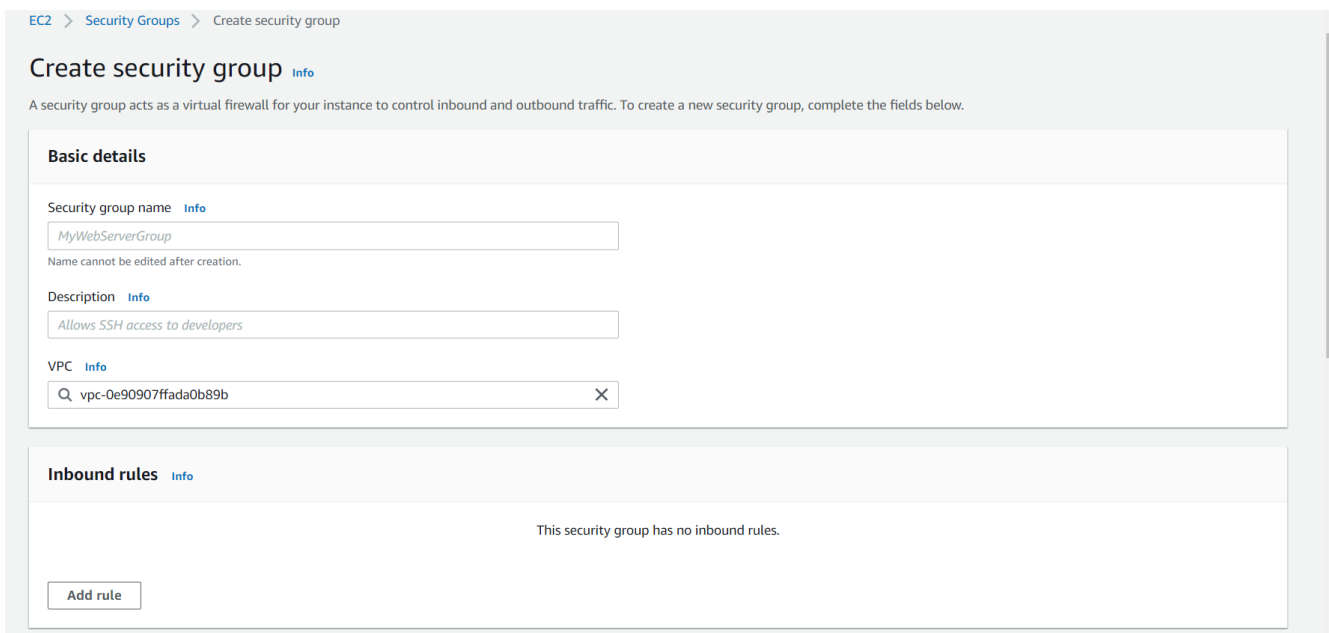
3.2.3 - Create a Security Group

Go to the EC2 Dashboard by logging into your AWS Management Console.

Choose "Security Groups" from the menu on the left.

Select "Create Security Group" from the menu.

Give your security group a name in the "Create Security Group" dialogue box. You may call it, for instance, "Web Server Security Group."



The screenshot displays the AWS Management Console interface for creating a security group. The breadcrumb trail at the top reads "EC2 > Security Groups > Create security group". The main heading is "Create security group" with an "Info" link. Below the heading is a note: "A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below." The form is divided into two sections: "Basic details" and "Inbound rules".

Basic details

- Security group name** (Info): A text input field containing "MyWebServerGroup". Below the field, it says "Name cannot be edited after creation."
- Description** (Info): A text input field containing "Allows SSH access to developers".
- VPC** (Info): A dropdown menu showing "vpc-0e90907ffada0b89b" with a search icon and a close button (X).

Inbound rules (Info)

This security group has no inbound rules.

There is an "Add rule" button at the bottom left of the Inbound rules section.

Figure 15: - AWS Create Security Group Page

Enter a succinct explanation of the security group in the "Description" section, such as "Allows all traffic from port 8080."

Choose the VPC for which you wish to create the security group from the "VPC" dropdown menu.

Just press "Create."

Once the security group has been created, choose it from the EC2 dashboard's list of possible security groups.

The "Edit" button can be found after selecting the "Inbound Rules" tab.

3.2.4 - Create a EC2 Instance

One of the primary skills that any AWS user must master is how to create an EC2 instance. I successfully built my first EC2 instance in this regard, which I gave the name Jenkins 1.

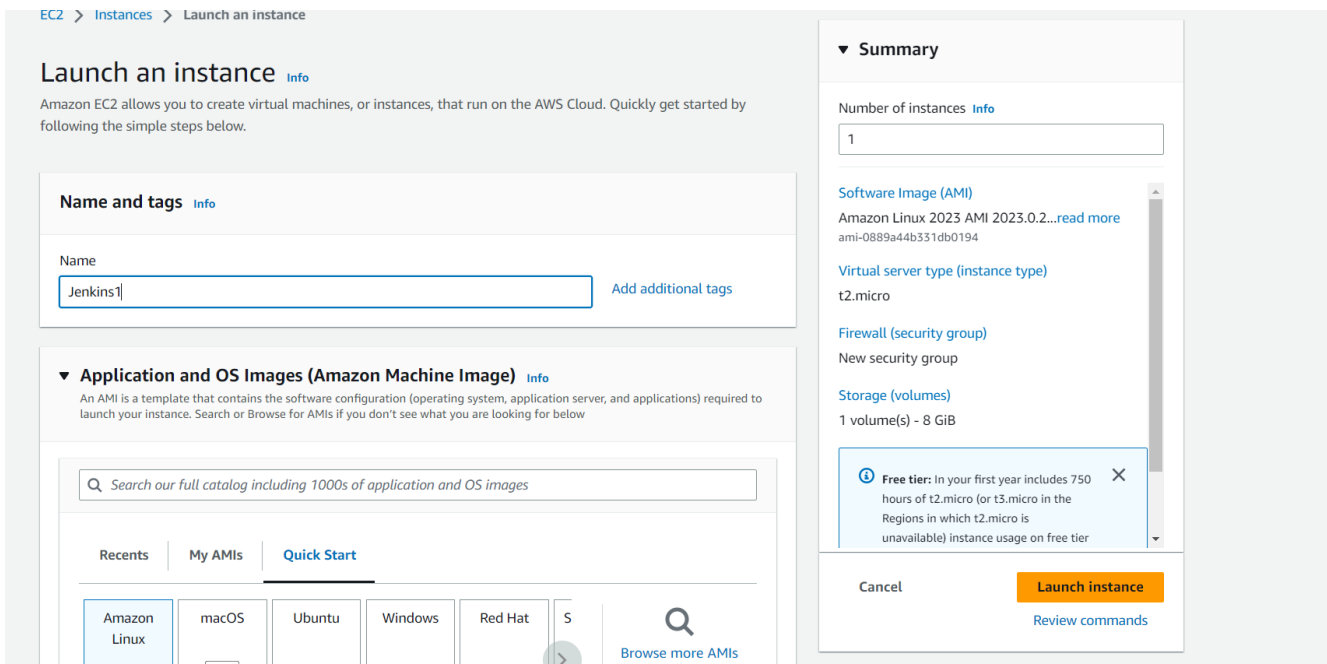


Figure 16: - AWS Create EC2 Instance Page

I started by selecting a suitable Amazon Machine Image (AMI), in this case, the Amazon Linux AMI. This AMI is well-known for its cost effectiveness and security characteristics; thus it was the ideal option for my EC2 instance. Choosing the launch instance option and looking for the AMI in the AWS marketplace throughout the instance construction procedure allowed me to choose the AMI.

The next option I picked was the instance type, and I went with t2.micro. This instance type should be used by applications with low to moderate network bandwidth requirements and

moderate CPU use. Considering that I was only constructing the instance for testing, the t2.micro type was the ideal choice.

After selecting the instance type, I created a key pair to allow secure login to the instance. The key pair is essential since it permits secure access to the EC2 instance. I kept the private key in a safe place because I would need it later to access the instance.

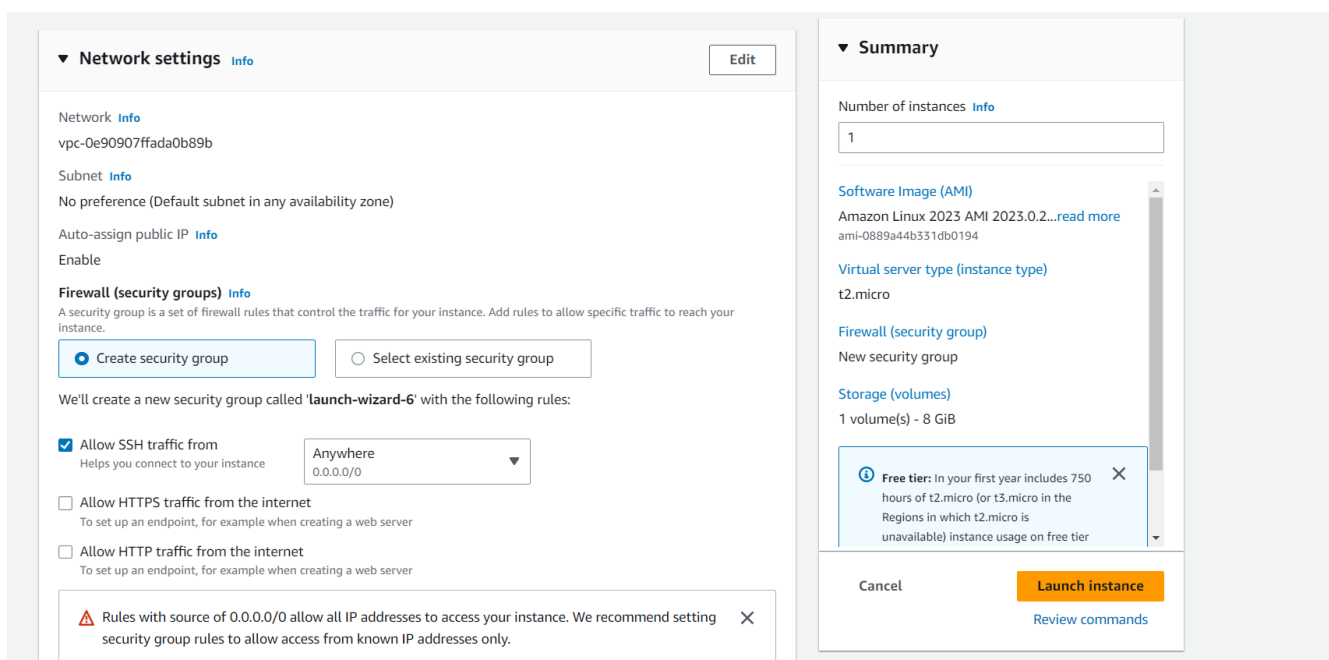


Figure 17: - AWS Create EC2 Instance Page2

I finally started the EC2 instance, and a little while later it was operational. The key pair allowed me to log in to the instance, and everything went without a hitch. On the instance, I set up Jenkins and began using it to control my continuous integration and delivery (CI/CD) workflow.

3.2.5 - Install a Jenkins Server on EC2 Instance

Step 1: Start an EC2 instance.

Launching an EC2 instance on AWS is the initial step. Make sure you choose the appropriate instance type and Amazon Machine Image (AMI). We will employ an AMI that supports Jenkins installation for this installation.

Step 2: Connect to the EC2 Instance.

Connect through SSH using your favourite SSH client once the EC2 instance is up and running. Make sure you have the key pair we need in order to connect to the instance.

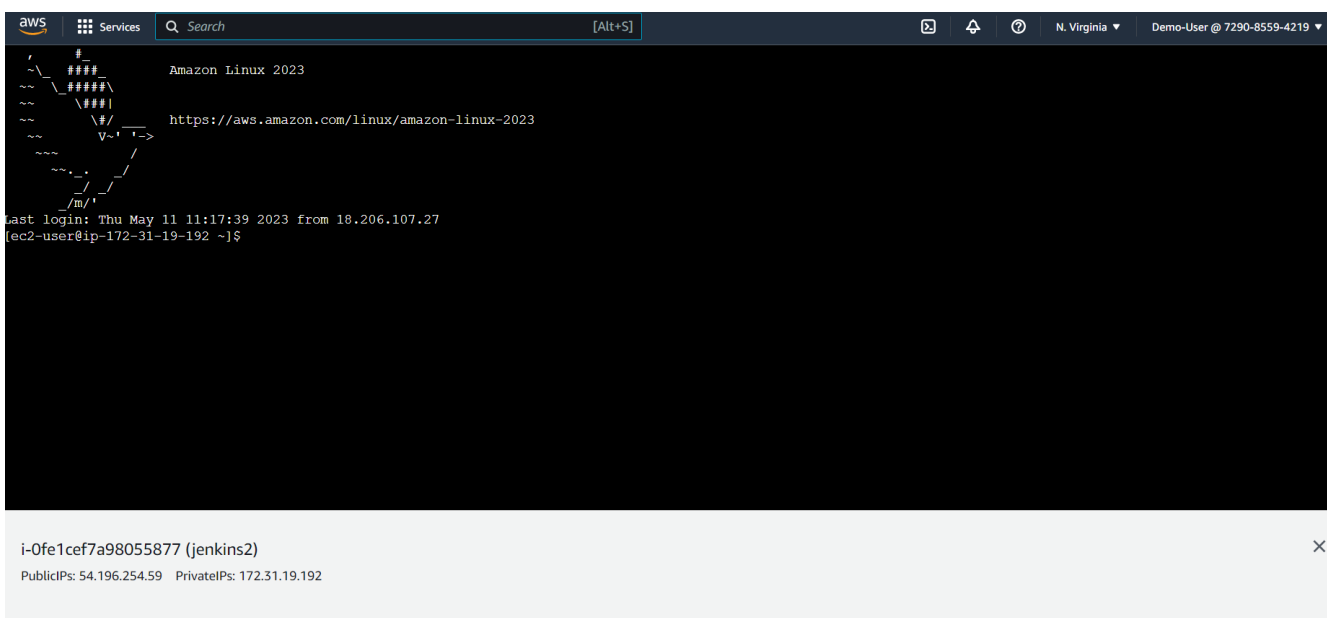


Figure 18: - AWS EC2 Instance Connect Page

Step 3: Install Java

We must install Java on your EC2 instance because Jenkins is a Java-based application. Java may be installed by executing the command:


```
sudo systemctl enable jenkins
```

Step 6: Start Jenkins

Use the following command to launch the Jenkins service after the installation is finished:

```
sudo systemctl start jenkins
```

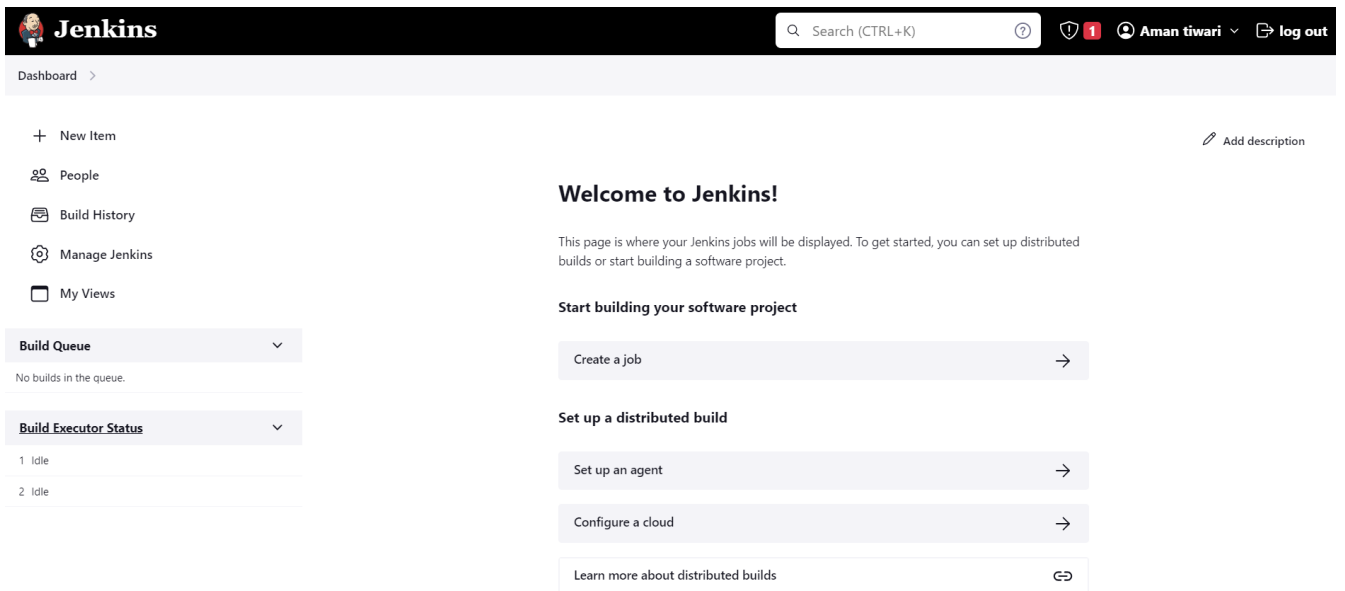


Figure 20: - Jenkins Server Started

Step 7: Configure Jenkins

We may now use your web browser to visit the Jenkins service by going to `http://your_ec2_instance_ip_address:8080` after initiating the service. To finish the Jenkins installation phase and create your admin account, simply adhere to the prompts.

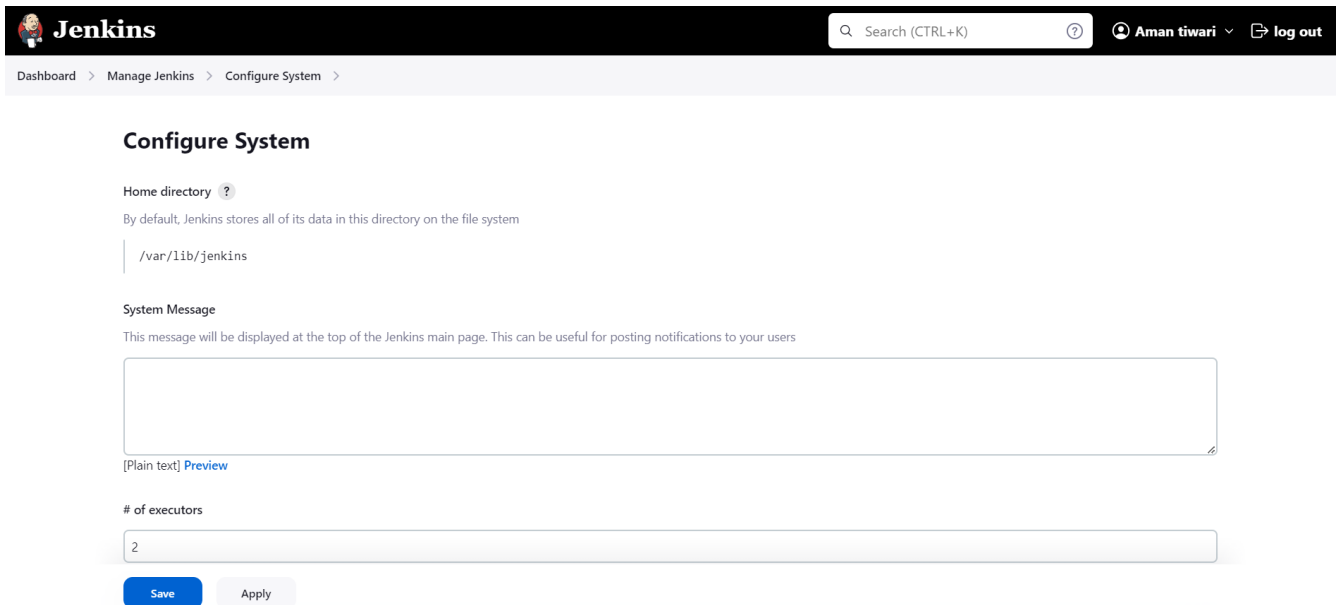


Figure 21: - Configure Jenkins Server

Step 8: Allow Jenkins to Run on Boot

The last step is to set up Jenkins to launch automatically as soon as your EC2 instance starts up. Utilise the following command to accomplish this:

```
sudo systemctl enable jenkins
```

3.2.6- Install a SAML Plugin inside my Jenkins 1

Using the key-pair you generated during instance creation, access the Jenkins1 EC2 instance through SSH. If your computer is running Windows, you may accomplish this using a programme like PuTTY.

When you have gained access to the instance, enter the EC2 instance IP address followed by:8080 in the address bar of your web browser to launch the Jenkins dashboard. If your EC2

instance's IP address is 12.34.56.78, for instance, you would enter `http://12.34.56.78:8080` in the address bar.

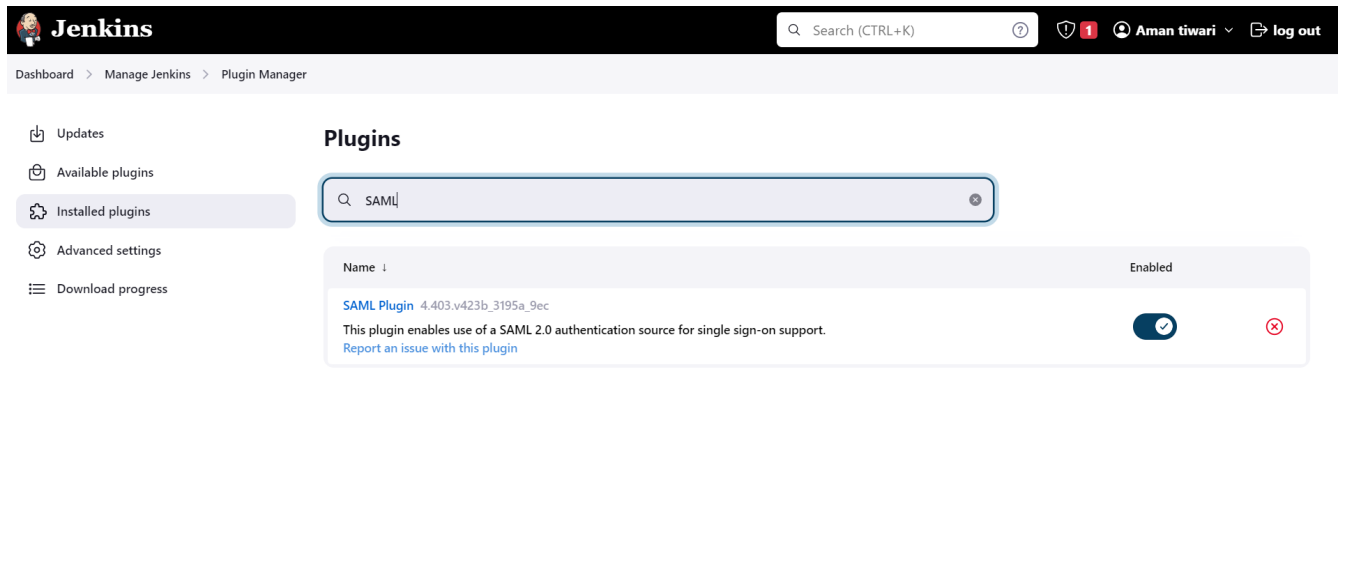


Figure 22: - Installed Plugin Page

Go to the "Manage Jenkins" area on the left-hand side of the page on the Jenkins dashboard and select "Manage Plugins."

Search for "SAML" in the search box after selecting the "Available" tab at the top of the page.

The "SAML" plugin must be checked in the box next to it before clicking the "Install without restart" button at the bottom of the page.

Await the plugin's installation. This could take a while.

Return to the Jenkins dashboard after the installation is finished, then go back to the "Manage Jenkins" area. Now select "Configure Global Security."

Choose "SAML 2.0" from the dropdown menu in the "Security Realm" section.

Fill up the fields that are necessary for the SAML configuration. This will depend on the SAML provider you are using. Be remember to save your modifications.

By logging out of Jenkins and then back in with your SAML provider credentials, you can test your SAML configuration. if the configuration is proper.

3.2.7 – Create an IAM Identity Provider Application

With the help of the AWS IAM Identity Provider service, you can control and federate access to your AWS resources, providing safe access to all of your organization's apps and services. In this situation, setting up an application entails installing an IAM identity provider to make SSO available to users within your company.

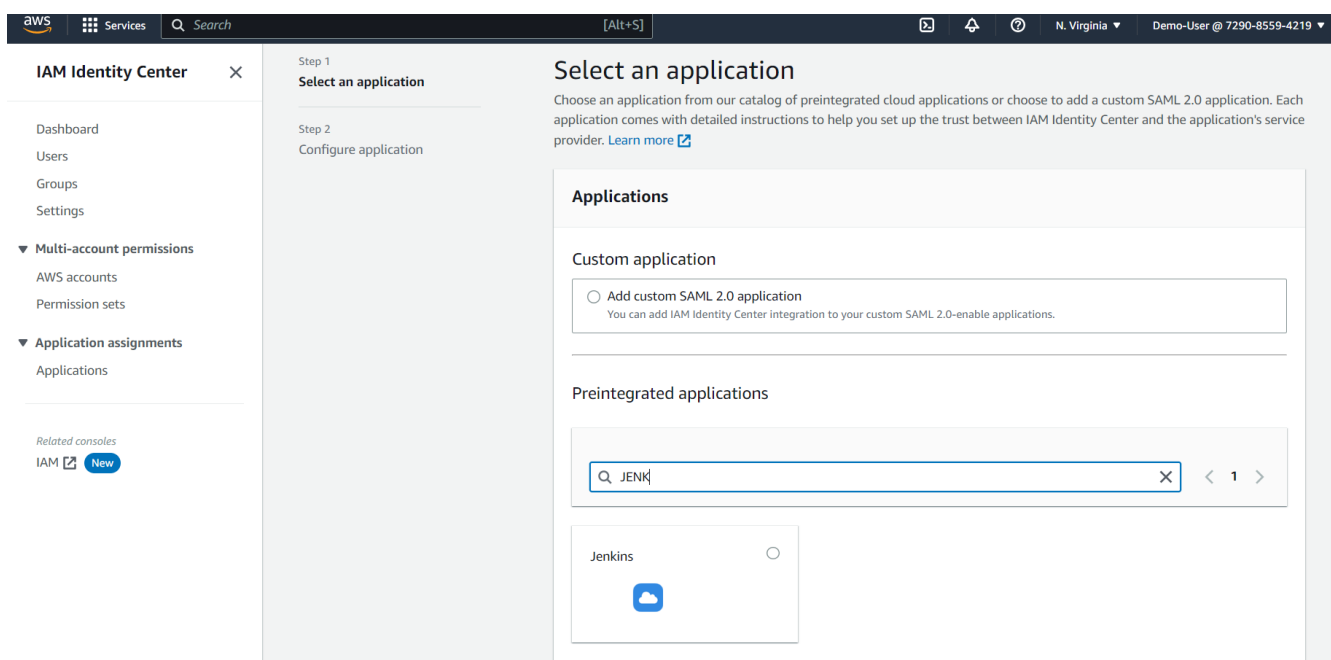


Figure 23: - Create IAM IDP Application Page

Logging into your AWS account and going to the IAM dashboard is the first step in building an application under the AWS IAM Identity Provider. Choose "Identity Providers" from the left-hand menu on the dashboard, then click the "Create Provider" button.

You will be asked to choose the sort of identity provider you wish to create on the "Create Provider" screen. Choose the "SAML" option to build an IAM identity provider that supports SAML-based SSO in this situation.

The next step is to give some fundamental details about your identity provider, like the provider name and metadata document. You can create your own metadata document using the AWS Management Console, or you can utilise one that is already given by your SAML 2.0-compliant identity provider.

To create your IAM identity provider after providing the required data, click the "Create" button. You can set up your identity provider to support SSO for your applications after it has been setup.

Go to the "Applications" tab in the IAM console and select the "Add an Application" option to establish your IAM identity provider. From this point, you may choose the application you wish to include and provide extra configuration information, including the application URL, SAML characteristics, and other details.

To enable SSO with your IAM identity provider after configuring your application settings, you must configure your SAML plugin inside of your Jenkins instance. In order to do this, you must set up the SAML plugin so that it can talk to your IAM identity provider and trade SAML assertions with your application.

3.2.8 – Create a User for Above Application

Step 1: Open the IAM dashboard after logging into the AWS Management Console. Select "Identity Providers" from the menu on the left.

Step 2: Choose the identity provider to which you wish to add the user, then click "View Details." Select "Add User" from the "Application" menu.

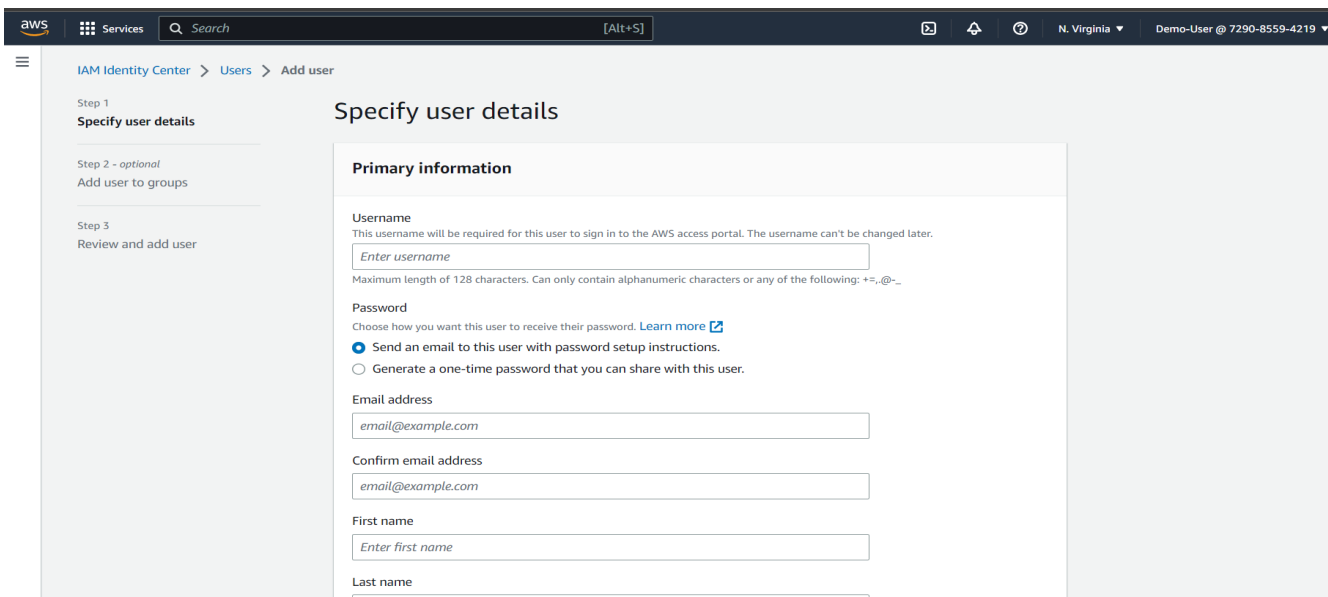


Figure 24: - Create User for IAM IDP Application Page

Step 3: Fill in the user's information on the "Add User" page, including their name, email address, and any necessary custom attributes. A password policy can also be set for the user.

Step 4: Choose the groups you want the user to be a member of under the "Group Membership" column. Either make a brand-new group or include the user in an already-existing one.

Step 6: Specify which user attributes should be linked to the application under the "Attribute Mapping" section. In terms of SSO authentication, this is crucial.

Step 7: Review the user's information and press the "Create User" button.

Step 8: Following the creation of the user, you may check their information and make any necessary adjustments, such as including or excluding them from applications or groups.

Step 9: Give the user the required login information, such as the application URL and any necessary SSO credentials, to enable them to access the application.

Step 10: The user can now access the resources or services to which they have been given authorization by logging in to the application with their IAM credentials.

3.2.9– Add a Application ACS URL and Application SAML Audience

Setting up SAML-based Single Sign-On (SSO) for an application requires adding the Application ACS URL and Application SAML Audience. Here is how to add these URLs in its entirety:

Navigate to the IAM service after logging into the AWS Management Console.

Choose "Identity providers" from the menu on the left.

Select the Identity Provider (IDP) whose URLs you want to add.

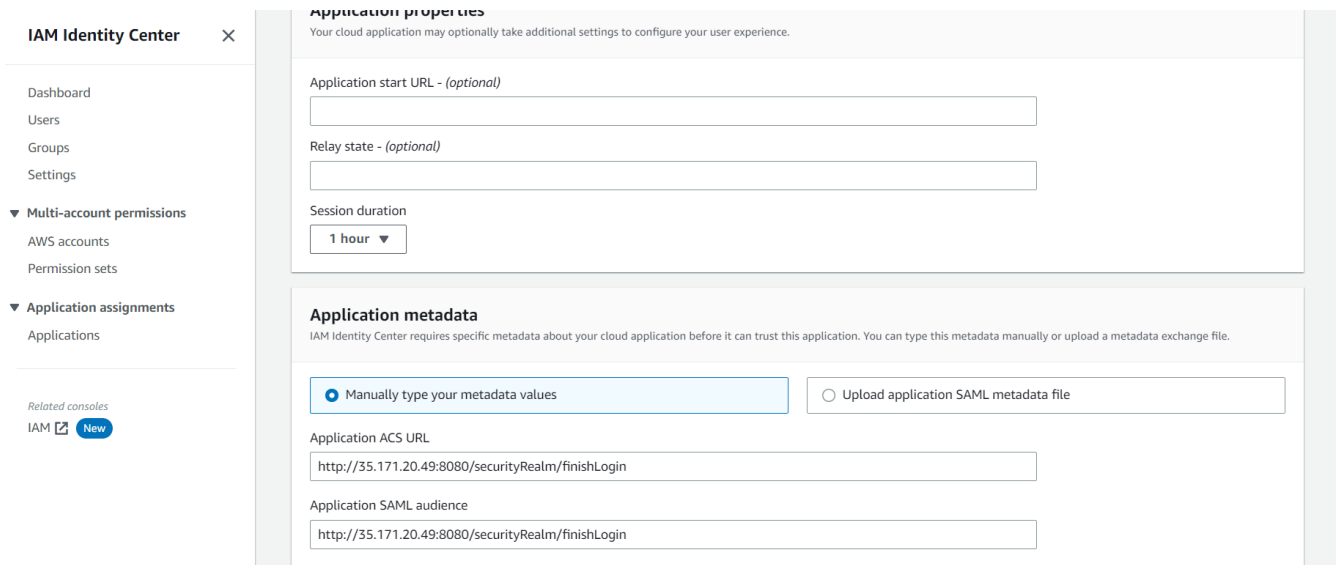


Figure 25: Adding ACS URL and SAML Audience

Next to the IDP, click the "View details" option.

Click the "Add an application integration" button after swiping down to the "Application integrations" area.

The following details in the "Add application integration" window:

The name of the programme you want to add the URLs to is the "Application name" in the first place. "ACS URL": The application's Assertion Consumer Service (ACS) URL. "Application SAML audience" refers to the SAML Audience URI that the application has provided.

To save the integration, click the "Add" button.

The ACS URL and SAML Audience URI will be saved in the IAM Identity Provider settings once the application integration has been added. For the application to authenticate users using the SAML-based SSO, these URLs are required. To ensure that the SSO authentication procedure goes successfully, double-check that you have entered the correct URLs.

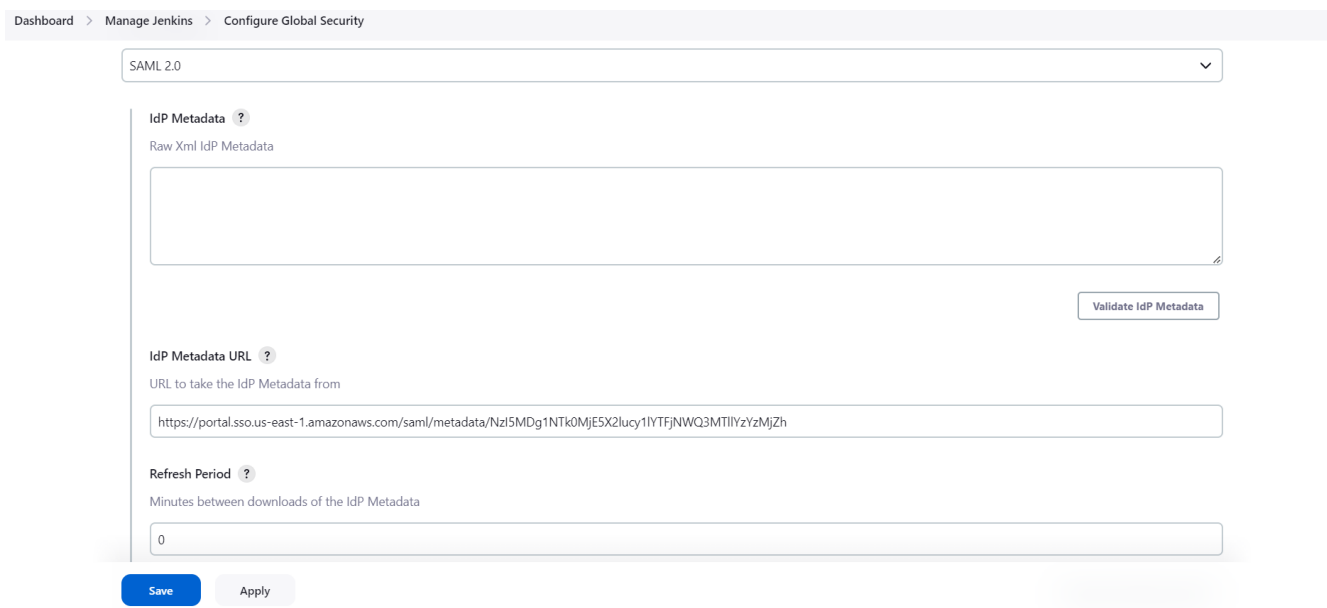
3.2.10 – In Configure Global Security inside Jenkins Add IDP Metadata URL and IDP logout URL

Jenkins must be integrated with an Identity Provider (IDP) like AWS IAM in order to configure SSO authentication. The addition of the IDP Metadata URL and the IDP Logout URL to Jenkins' Global Security settings is a crucial step in this procedure.

First, log in as the administrator to your Jenkins instance and go to the Manage Jenkins area. Then select Configure Global Security from the menu. The security settings page will thereafter appear as a result.

Choose the SAML 2.0 based authentication option under Security Realm. This will show you any other fields that require configuration.

Put the IDP Metadata URL that the IAM Identity Provider gave you in the IDP Metadata URL field under the IDP Metadata Configuration section. Usually, this URL looks something like <https://signin.aws.amazon.com/static/saml-metadata.xml>. This metadata file contains details about the SAML endpoints and public key of the IAM IDP.



The screenshot shows the Jenkins 'Configure Global Security' page for a SAML 2.0 realm. The breadcrumb trail at the top reads 'Dashboard > Manage Jenkins > Configure Global Security'. A dropdown menu at the top left is set to 'SAML 2.0'. The main configuration area is titled 'IdP Metadata' and contains three sections: 'Raw Xml IdP Metadata' with a large text area, 'IdP Metadata URL' with a text input field containing the URL 'https://portal.sso.us-east-1.amazonaws.com/saml/metadata/Nzl5MDg1NTk0MjE5X2lucy1lYTFjNWQ3MTllYzYzMjZh', and 'Refresh Period' with a text input field containing '0'. A 'Validate IdP Metadata' button is located to the right of the raw XML field. At the bottom of the form are 'Save' and 'Apply' buttons.

Figure 26: Adding IDP Metadata URL

The Entity ID provided by the IAM Identity Provider should then be entered in the IDP Entity ID field. This ID, which normally takes the form of `https://signin.aws.amazon.com/saml`, uniquely identifies the IAM IDP.

The IDP Logout URL field is included in the Advanced Configuration section. In this area, provide the IDP Logout URL that the IAM IDP gave. When a user logs out of Jenkins, they can use this URL to log out of the IAM IDP as well.

The SAML Audience URL that was specified when the IAM SAML application was created in the AWS Management Console should now be entered in the Application Configuration section. Usually, this URL looks something like `https://jenkins-hostname>/securityRealm/finishLogin`.

Once you have completed entering all the required data, click the Save button to keep your changes. IAM will now be set up as the IDP for SSO authentication in Jenkins.

Chapter 4: - EXPERIMENTS AND RESULT ANALYSIS

Using a single set of login credentials, a user can access various apps using the Single Sign-On (SSO) method of user authentication. Users no longer need to keep track of many sets of login information for various applications, which improves the effectiveness and user-friendliness of the authentication process. In this experiment, we'll examine the outcomes of setting up an SSO authentication system with the use of the Jenkins SAML Plugin and AWS Identity Provider (AWS IdP).

Setup for the experiment:

On AWS, we created an EC2 instance and installed the Jenkins server on it. Then, we configured Jenkins to use the AWS IdP as the Identity Provider after installing the SAML Plugin. In the AWS IdP, we built an application and added a user to it. Then, we amended the AWS IdP setup to include the Application ACS URL and Application SAML Audience. Finally, we modified Jenkins' Global Security setup to include the IDP Metadata URL and IDP logout URL.

Results of the experiment:

We tried to access Jenkins using the login information for the user we added to the AWS IdP application in order to test the SSO authentication system. Without providing any more login information, we were able to access Jenkins. The user was able to access Jenkins using their AWS IdP credentials thanks to the flawless operation of the SSO authentication system.

Analysis:

The SSO authentication system demonstrated to be an effective and user-friendly method of user authentication using AWS IdP and Jenkins SAML Plugin. Users no longer had to memorise numerous pieces of login information, and the authentication procedure was easy. We were able to take advantage of AWS's security capabilities and guarantee secure user authentication by using AWS IdP as the identity provider.

Additionally, managing user access to Jenkins was made simple by the SSO authentication system. We were in a position to manage who had access to Jenkins by adding users to an application in AWS IdP. As a result, controlling user access and ensuring that only approved people had access to Jenkins became simpler.

The SSO authentication system's possible drawback is that it needs an AWS account and AWS IdP to be set up. All businesses, especially smaller ones that do not have an AWS account, might not be able to pull this off. Additionally, setting up the SAML Plugin and the AWS IdP may call for some technical know-how due to their complexity.

In conclusion, a secure and effective method of user authentication is provided by the SSO authentication system using AWS IdP and the Jenkins SAML Plugin. Users no longer have to memorise numerous pieces of login information, and Jenkins user access can be easily managed. Although setting it up could involve technical know-how, it is a useful tool for companies using AWS who wish to guarantee safe user authentication.

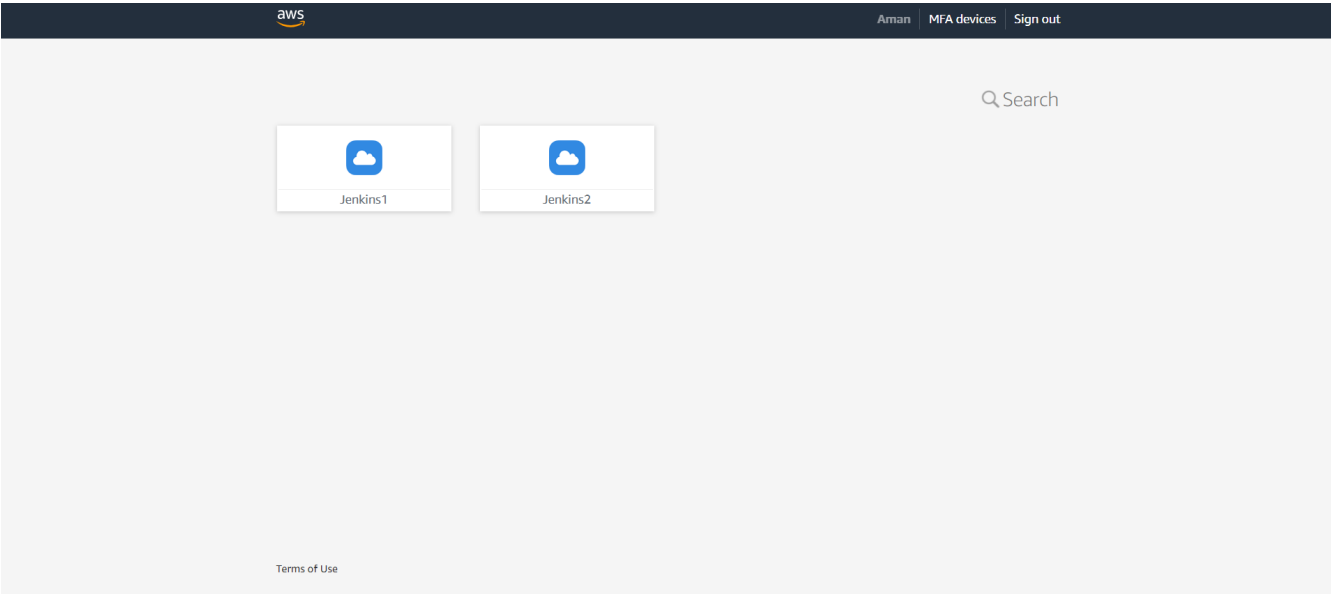


Figure 27: Final Application with two Jenkins server

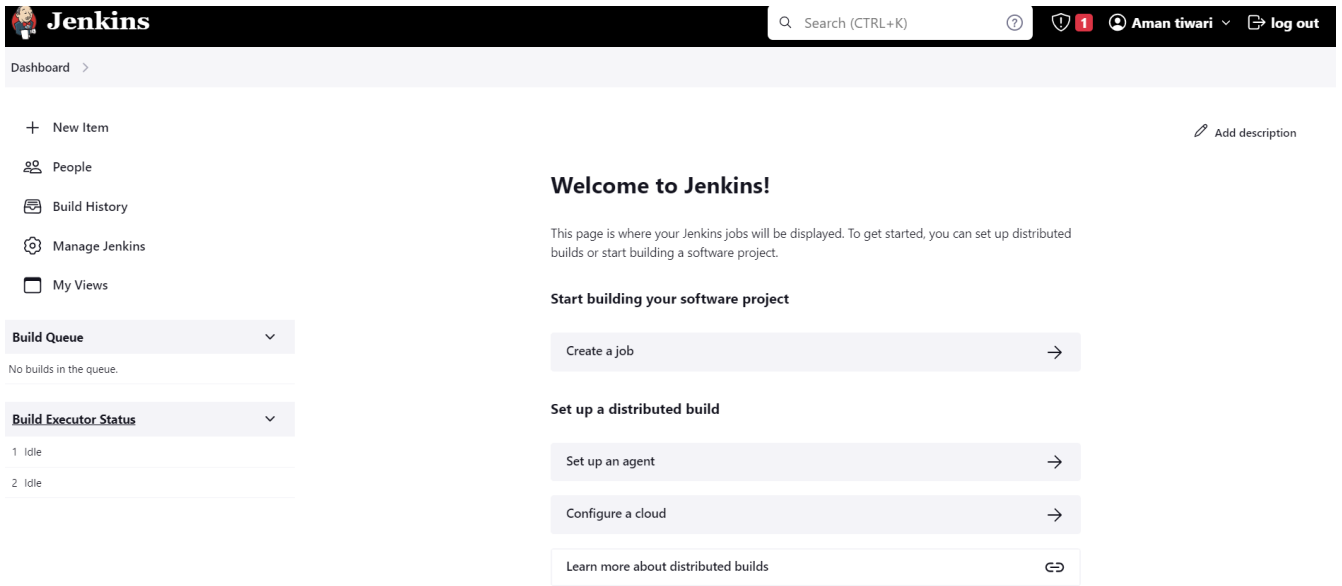


Figure 28: Jenkins Application accessed through SSO

Chapter 5: - CONCLUSIONS

5.1 Conclusions

A secure and effective method of handling user authentication for Jenkins servers is provided by the SSO authentication system created using AWS IAM Identity Provider and Jenkins SAML Plugin. Because access is only granted to authorized individuals, the system improves security by eliminating the need for multiple user accounts and passwords. The major challenge that is up ahead in this project is the inclusion of a scaling **factor** and how it can modify our results. As of now, we have only focused upon the simple regression models or some complex ensemble learning techniques which yielded some quite good results considering the amount of data set points that we inputted. Inclusion of lasso and ridge regression is one more thing to look forward to.

The experiment's findings demonstrated that the solution is reliable and efficient, enabling smooth SSO authentication and Jenkins login. The implementation procedure required little technical knowledge and was simple to follow. To ensure that the system operates properly, attention must be made to configuration elements including the application ACS URL, SAML Audience, IDP metadata URL, and logout URL.

The project's future goals include expanding SSO authentication to other internal applications, connecting the system with third-party identity providers, and enhancing the user interface to make setting simpler. In order to increase security even further, multi-factor authentication can be used.

5.2 Future Scope

The project's future goals include expanding SSO authentication to other internal applications, connecting the system with third-party identity providers, and enhancing the user interface to make setting simpler. In order to increase security even further, multi-factor authentication can be used.

While SSO authentication offers a safe and practical way to manage users, it is crucial to remember that it is not perfect. Users must be careful to create secure passwords, and only

authorised workers should have access to important data. To find and fix potential security flaws, routine system audits and vulnerability assessments should be performed.

5.3 Applications

The project of ours can find use in many of the scenarios as mentioned below: -

- i.) Applications used by large organisations that require their employees to log in to each app separately can utilise our SSO authentication solution to access enterprise-level applications. Employees can access all applications without having to memorise multiple login credentials by linking our solution with the company's current identity provider.
- ii.) Applications that run in the cloud: Our SSO authentication system can offer customers a safe and practical means to access these applications as cloud-based applications gain in popularity. Users can access several cloud-based applications by using a single set of login information, saving them from having to remember numerous usernames and passwords.
- iii.) Websites that do business online can utilise our SSO authentication solution to log users in so they can access their accounts. Our technology enables users to seamlessly log in to many e-commerce websites, which can boost consumer satisfaction and sales.
- iv.) Educational institutions: Our SSO authentication solution can be useful for educational institutions that use a variety of applications, including learning management systems, library systems, and student information systems. Students and staff can access all the applications using a single set of credentials by linking our system with the institution's current identity provider.
- v.) Government organisations: Organisations that need quick and secure access to numerous applications can use our SSO authentication solution. Employees can access all the applications they need to perform their jobs by integrating our system with the agency's current identity provider without having to memorise multiple logins.

- vi.) Mobile applications: Secure and convenient access to mobile applications can be provided by use of our SSO authentication solution. Users can access several mobile applications without having to remember many usernames and passwords by integrating our system with the identity provider already present in the mobile application.

REFERENCES

A) Documentations

[1] <https://docs.aws.amazon.com/>

[2] <http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html>

[3] <https://www.jenkins.io/doc/>