**Jenkins Server Backup and Restoration**



Introduction: In the world of continuous integration and continuous deployment, Jenkins is a popular choice for automating software development processes. However, like any other critical component of your infrastructure, it's essential to have a robust backup and recovery plan in place. In this blog post, we'll walk you through the process of backing up your Jenkins server to an Amazon S3 bucket and creating a new Jenkins server using that backup.

**THEORY :**

**Why Backup Jenkins Servers:**

* **Data Protection:**
* *Pro:* Jenkins backups safeguard critical configurations, job definitions, and build histories, protecting against data loss due to hardware failures, accidental deletions, or system crashes.
* **Disaster Recovery:**
* *Pro:* Backups are essential for disaster recovery. In the event of a server failure or data corruption, having a recent backup allows for quick restoration, minimizing downtime and maintaining continuity in the development process.
* **Configuration Preservation:**
* *Pro:* Backups capture the entire Jenkins configuration, preserving settings, plugins, and custom configurations. This ensures consistency when migrating to a new server or recreating the Jenkins environment.
* **Job and Pipeline Integrity:**
* *Pro:* Continuous Integration (CI) and Continuous Deployment (CD) rely on consistent job configurations. Backups ensure the integrity of these jobs, preventing disruptions in the software development lifecycle.
* **Version Control Integration:**
* *Pro:* Jenkins backups help maintain version control system integration. Storing backup files in version-controlled repositories adds an extra layer of security and facilitates collaboration among development teams.

Cons:

* **Storage Overhead:**
* *Con:* Regular backups consume storage space, and this can become a concern for large Jenkins instances. Organizations need to manage storage efficiently and consider the cost implications of storing backups.
* **Backup Frequency Challenges:**
* *Con:* Determining the optimal backup frequency can be challenging. Too frequent backups may strain system resources, while infrequent backups risk losing recent changes. Striking the right balance is crucial.
* **Security Risks:**
* *Con:* Backup files, if not properly secured, can pose security risks. It's crucial to encrypt and restrict access to backup archives to prevent unauthorized users from gaining access to sensitive data.
* **Dependency on Backup Tools:**
* *Con:* Relying on backup tools introduces a dependency on their reliability. Incompatibility or issues with backup plugins could impact the ability to restore Jenkins instances successfully.
* **Complexity of Restoration:**
* *Con:* While backups are essential, the process of restoring a Jenkins server, especially in a different environment, can be complex. Ensuring a well-documented and tested restoration process is vital.

Title: **Jenkins Server Backup and Restoration: A Step-by-Step Guide**

Introduction: In the world of continuous integration and continuous deployment, Jenkins is a popular choice for automating software development processes. However, like any other critical component of your infrastructure, it's essential to have a robust backup and recovery plan in place. In this blog post, we'll walk you through the process of backing up your Jenkins server to an Amazon S3 bucket and creating a new Jenkins server using that backup.

* Disaster Recovery
* Schedule backup
* Recovery Backups Drills periodically.
* Prerequisite
* Disaster recovery of Jenkins server.
* AWS CLI
* IAM Roles
* S3 bucket stores

**Step 1: Set Up an S3 Bucket**

Before we start with the backup process, ensure you have an Amazon S3 bucket set up. This bucket will serve as the storage location for your Jenkins server backup. Make sure you have the necessary AWS credentials to access this bucket programmatically.

**Step 2: Install the AWS CLI**

To interact with your S3 bucket from your Jenkins server, you'll need to install the AWS Command Line Interface (CLI). You can find installation instructions on the official AWS CLI documentation.

**Step 3: Configure AWS CLI**

After installing the AWS CLI, configure it with your AWS credentials. Run the following command and provide the required information:

bashCopy code

aws configure

**Step 4: Perform a Manual Backup**

Before proceeding to create a new Jenkins server, perform a manual backup to ensure everything is configured correctly. Run the backup job you created in the previous step.

**Step 5: Create a New Jenkins Server**

In the event of a disaster or if you need to replicate your Jenkins setup, you can use your S3 backup to create a new Jenkins server. Follow these steps:

* Install a fresh Jenkins instance on a new server.
* Restore the backup by running the restore job created during the configuration.

**Conclusion**

By following these steps, you can ensure the safety and availability of your Jenkins server by regularly backing up critical configurations and automating the restoration process. This proactive approach minimizes downtime and ensures that your development pipeline remains robust and reliable. Remember to periodically test your backup and restoration processes to verify their effectiveness.

**LABS Practical :**

**Prerequisite**

* Install new Two EC2 instance

Jenkins server instance & Jenkins Backup server instance

* Disaster recovery of Jenkins server.
* AWS CLI
* IAM Roles with s3 full access policy
* S3 bucket stores

Jenkins credentials

plugin installed on jenkins server

**A . Create a Jenkins Server**

In the event of a disaster or if you need to replicate your Jenkins setup, you can use your S3 backup to create a new Jenkins server. Follow these steps:

* Install a fresh Jenkins instance on a new server.
* Restore the backup by running the restore job created during the configuration.

Create a two jobs for data existing -

S3 buckets creations & IAM roles creation attached to Jenkins’s server

Install Jenkins on Jenkins server instance & Jenkins Backup server instance

Create a Pipeline to test CI CD

Jenkins systems status with following commands below.

Systemctl stop Jenkins

systemctl status Jenkins

path to jenkins

/var/lib/Jenkins

data of all the Jenkins exists

tar -zcvf jenkinsbackup.tar.gz /var/lib/jenkins/

**Install CLI AWS**

* **Install Python 3 pip on Ubuntu:**

sudo apt update

sudo apt install python3-pip

* **Install AWS CLI on Ubuntu:**

This assumes you have Python 3 and pip3 (Python package installer for Python 3) already installed on your system.

pip3 install awscli

sudo apt update

Assign the Role of S3 with Jenkins server instance

aws s3 cp jenkinsbackup.tar.gz s3:// jenkinsbackupbucketname/jenkinsbackup.tar.gz

aws:s3:::jenkinsbackupbucketname

**Create a Jenkins Backup Server**

In the event of a disaster or if you need to replicate your Jenkins setup, you can use your S3 backup to create a new Jenkins server. Follow these steps:

* Install a fresh Jenkins instance on a new server.
* Restore the backup by running the restore job created during the configuration.

S3 buckets creations & IAM roles creation attached to Jenkins server

ip:/home/ubuntu :

aws s3 cp s3://jenkinsbackupbucketname/jenkinsbackup.tar.gz jenkinsbackup.tar.gz

copied to root folder

To backup a Jenkins server, use the command tar -zxvcf Jenkins-backup.tar.gz -C / to compress and archive the Jenkins home directory. This creates a backup file that can be stored or transferred, providing a snapshot of the Jenkins configuration and data. To root

Delete default jnksin directory create during jenkins installation.

rm -rf /var/lib/Jenkins

Check status of jenkins on Jenkins backup server instance.

This command packages the entire Jenkins home directory into a gzip-compressed archive named "jenkinsbackup.tar.gz." Regularly executing this command ensures the protection of critical configurations, job definitions, and build histories in Jenkins.

tar -zxvf jenkinsbackup.tar.gz -C /

tar -zxvf Jenkinsbackup.tar.gz -C /

Restart your Jenkins service

systemctl status Jenkins

system start Jenkins

system status jenkins

access the Jenkins with ip:8080

open Inbound rules for jenkins 8080 port in Ec2 security group.

Access using ip:8080 jenkins to validate the Pipilines/jobs restores , plugins exists etc.  
\*\*\*\* use old credentials to login new server

refresh the page   
review the jobs & plugins is it exists