

# TASK 19

## 1. GitLab CI/CD :

GitLab CI/CD is an integrated software development tool within the **GitLab DevSecOps platform** that automates the entire software delivery process, from the initial code commit to deployment in production environments. It helps teams implement continuous methodologies including Continuous Integration (CI), Continuous Delivery (CD), and Continuous Deployment (CD).

- **.gitlab-ci.yml file:** The YAML file where pipeline logic, stages, and jobs are defined.
- **Pipelines:** The top-level components that orchestrate the sequence of building, testing, and deploying the code.
- **Jobs:** The fundamental tasks executed by a runner.
- **Stages:** Define the order in which jobs run.
- **Runners:** The execution agents that run the jobs (can be GitLab-hosted or self-managed).
- **CI/CD Variables:** Key-value pairs used to store and pass configuration or sensitive information (like API keys) securely to jobs.
- **Artifacts and Cache:** Mechanisms to store intermediate build results and speed up subsequent jobs by saving dependencies.

GitLab CI/CD persetup:

💡 **Ssh-keygen -t rsa** (create key pair in git bash - public key and private key)

```
sarav@hp MINGW64 ~
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/users/Sarav/.ssh/id_rsa):
Enter passphrase for "/c/Users/Sarav/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/users/Sarav/.ssh/id_rsa
Your public key has been saved in /c/users/Sarav/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vqYga7V1LKKR1kwwqx33ymcJJj2jjXScnT0Ibt1fzYZk sarav@hp
The key's randomart image is:
+---[RSA 3072]---+
|   o.
|   o.
|   o+.
|  ooo.    o
| . . . .OS  o E
|  oo+Oo=  . + .
|  . ==+O& =  .
|  . =O+ +. @
|   .  oo. .
+---[SHA256]---+
```

## ⊕ Private key

```
sarav@hp MINGW64 ~
$ cat ~/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzac1rZXktdjEAAAAABG5vbmuAAAAEb9uZQAAAAAAAAABAAABTwAAAAdzc2gtcn
NhAAAAAAwEAAQAAAYEAZB0quH/kk1K+1VwdtFM6+x5eBmfefx001/BqOPE45U7KiK+UhrRM
YP35Drwhh9fQ2GMbVoAcNcGnu7oRXBe8pBwcqnqzset600/V7xmfc1PguwKuIdl2FS92/
7wkjkDBVSGzMSQg0NcEK4LVo7SyzPrabeMGiasJQ2ks/Nco4HYqHsJv/Vv1UNnoluPNr05
2pwCDSsqChUvcv2U20v/yEl100zBga1JyGIB1YAWfrLyoZKwpS7j0tGP+TMSwu01D+v5Na
ezdhs9ZBpDiiE3t2U9Dqz3nY1bqWE1d++KnAqUs/DFwKha83Fj0t0YCsRT3WZiFDB8UhM
u3Ecj42ABDVJQisHZowRbtcm1pbXB9G76kk25RS/1ioEni1FA47HAnIVbyP0LXIauoSG5G
32bMhs1wiUmMcIAqywh70zqcid7uwOr4ELYph5PO5HiBpv/w7fv8LtJvmktSOEK+shnTYG
T5t1VTyOuzUssu2uOTrw49Ex7iTy586YXAA3S9FjAAAFgFhEnuNYRJ7jAAAAB3NzaC1yc2
EAAAGBAMwTqrh/5JJsvtVvnBRT0vseXgZn3n19DtfwajjxooVoyoi1B60TGD9+Q61oR/X
0NhjG1aAHDXBp106EvwXvKQCHJ6p87hreqNP1e8ZnwpT4LsCriInZdhUvdv+8JCZAwUhs
ZEKINDXBCuC1a0osz62m3jbomrCUNpLPZXDuBKh7Cb/1b5VDZ6jbjzazudqvgg0rKgov
LwldlNjr/8hC4tDswRmtSchAdwAFn6y8qgs1qUu49LRj/kzEsLqNQ/leTwns3YUvWQaQ4
ohN7d1PQ6md52NYgalhNXfvipwK1LPwxccowvNxYzrdGArEU91mYhQwfFITLtxHI+NgAQ7
yuIrB2TsEQbXJpaW1wfRu+pJNuUUv9YqbJ4pRQOOxwJyL28j9c1yALqEhuRt9mzIUpViLp
jHIgEM1h+9M6nIne7sDq+BC2KR+TzuRyG6Vf8031fc7sb5pLutBCvrlZ02Bk+bdVu8jrmv
LLLtrjk0VuPRMeyE8uf0mFwan0vRYwAAAAMBAEAAAGAF8LidID6TXSHoQdk9B1iF1hof0
PIgAv47SoIt+b7n0zDoD1Gocc/GBFPgmrwy72xJMDFWhj92onQaxXrf4dGq5pVABRxD4w/
cexwba/UrhB+G3LvxkbtZigCdFmxYOSv+VrnfHLBRhyoXON80859X144Ns2w6SSU6UDhay
jsHBxOQa+xvuwGMKgchhxAXI996v16yasQ8K5EFRm++ag2K/cTDwhso19jNor9c60U8w5H
LzuGLCLS2/pn17geoxs/G1dgGu7181sSYCUQo9Pb4LcwLGgA9CP8VpvMh28VjuGuWnM8YZ
43FEh+Xmmn9hxkocuI+FpViw4BND2BBka7myzzSkwfJAT0rpmbq2YYVaY1JZYS2FZ41KB
R1YnFGaGFswW8uB7Lyob+yJ8E9Jro2JeOhvEjv8N6wdnD06LPj0Dhw3k20ardM4pswIeHg
9mDT3Q20mnXU2i8xkQAIqrxaDLa0s00qIQQEQi8kiafwqHSCd2W6sUOYiGQG960hAAAA
```

## ⊕ Public key

```
sarav@hp MINGW64 ~
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQDME6q4f+SSUr7VVZ20Uzr7H14GZ959fQ7X8Go48Tj1Tsqr5QetExg/f
kotaEf19DYYxtWgBw1wadTuhFcF7ykHBByeqfox63qjt9XvgZ8Ku+C7Aq4iJ2XYVL3b/vCQmQMVFVibMxJCDQ1wQrgtwjtLL
M+tpt4waJqwlDaSz81w7gdiowem/9W+vQ2eiw482s7na1YINKyokFS8JXZTY6//IQuLQ7MEzrUnIYgHVgBZ+svKhkpallLu
PS0Y/5MxLC6jUP5Xk1p7N2FL1kgkOKITE3ZT0opnedjWIGpYTV374qcCpSz8MXAqFrzcWM63RgKxFPdZmIUMHxSEy7cRyP
jYAE081CKwdk7BEG1yawltcH0bvqSTb1FL/WKgSeKUUDjsscCc19vI/QtcgC6hIbkbfZsyFKVYi6YxyIBDJYfvTopyJ3u7A
6vgQtikfk87kchulX/Dt9Xwu0m+aS1LQQr6yGdNgZPm3VVPI651syy7a45NFbj0THshPLnzphcAddL0WM= Sarav@hp
```

## ⊕ Update the ssh key into your gitlab account (past your public key)

The screenshot shows the GitLab user settings interface. On the left, there's a sidebar with various settings like Profile, Account, Billing, Applications, Integration accounts, Personal access tokens, Emails, Password, Notifications, SSH Keys (which is currently selected), GPG keys, Preferences, Comment templates, Active sessions, Authentication log, and Usage quotas. The main content area is titled "User Settings / SSH Keys / Sarav@hp". It displays a table for "Key details" with columns for Usage type (Authentication & Signing), Created (Dec 10, 2025 9:08am), Last used (Never), and Expires (Dec 10, 2026 12:00am). Below this is a section for "SSH Key" containing the public key text: "ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQDME6q4f+SSUr7VVZ20Uzr7H14GZ959fQ7X8Go48Tj1Tsqr5QetExg/fk0taEf19DYYxtWgBw1wadTuhFcF7ykHBByeqfox63qjt9XvgZ8Ku+C7Aq4iJ2XYVL3b/vCQmQMVFVibMxJCDQ1wQrgtwjtLLM+tpt4waJqwlDaSz81w7gdiowem/9W+vQ2eiw482s7na1YINKyokFS8JXZTY6//IQuLQ7MEzrUnIYgHVgBZ+svKhkpallLuPS0Y/5MxLC6jUP5Xk1p7N2FL1kgkOKITE3ZT0opnedjWIGpYTV374qcCpSz8MXAqFrzcWM63RgKxFPdZmIUMHxSEy7cRyPjYAE081CKwdk7BEG1yawltcH0bvqSTb1FL/WKgSeKUUDjsscCc19vI/QtcgC6hIbkbfZsyFKVYi6YxyIBDJYfvTopyJ3u7A6vgQtikfk87kchulX/Dt9Xwu0m+aS1LQQr6yGdNgZPm3VVPI651syy7a45NFbj0THshPLnzphcAddL0WM= Sarav@hp". There's also a "Delete" button. At the bottom, there are sections for "Fingerprints" (MD5: dc:f3:7a:cf:9d:a6:21:09:0d:9e:c3:3f:ce:8a:c7:47, SHA256: vqYga7V1LkkR1kwqx33ymcJj2jjXSct01btlfzYzk) and a "Copilot" section.

## Result :

User Settings / SSH Keys

Search settings

### SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab. SSH fingerprints verify that the client is connecting to the correct host. Check the current instance configuration.

Your SSH keys  1						<a href="#">Add new key</a>
Title	Key	Usage type	Created	Last used	Expires	Actions
Sarav@hp	 dc:f3:7a:cf:9d:a6:21:09:0d:9e:c3:3f:ce:8a:c7:47	Authentication & Signing	in 10 seconds	Never	2026-12-10	<a href="#">Revoke</a> 

## Create IAM user in your aws account

### Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.



[Delete](#)

[Create user](#)

<input type="checkbox"/>	User name	 Path	 Group:	 Last activity	 MFA	 Password age	 Console last sign-in	
<input type="checkbox"/>	<a href="#">saran</a>	/	0	-	-	-	-	

## Generate the access key and secret access key in your IAM user

### Retrieve access keys Info

#### Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key

Secret access key

 AKIA4O3W5DSTTKKGVI70

 \*\*\*\*\* Show

#### Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

## Update the gitlab project variables into IAM access key and secret access key

### Project variables

Variables can be accidentally exposed in a job log, or maliciously sent to a third party server. The masked variable feature can help reduce the risk of accidentally exposing variable values, but is not a guaranteed method to prevent malicious users from accessing variables. How can I make my variables more secure?

CI/CD Variables </> 2		Reveal values	Add variable
Key ↑	Value	Environments	Actions
AWS_ACCESS_KEY_ID 	..... 	All (default) 	 
AWS_SECRET_ACCESS_KEY 	..... 	All (default) 	 