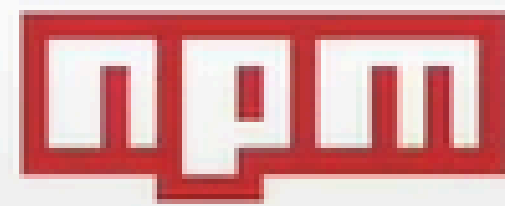
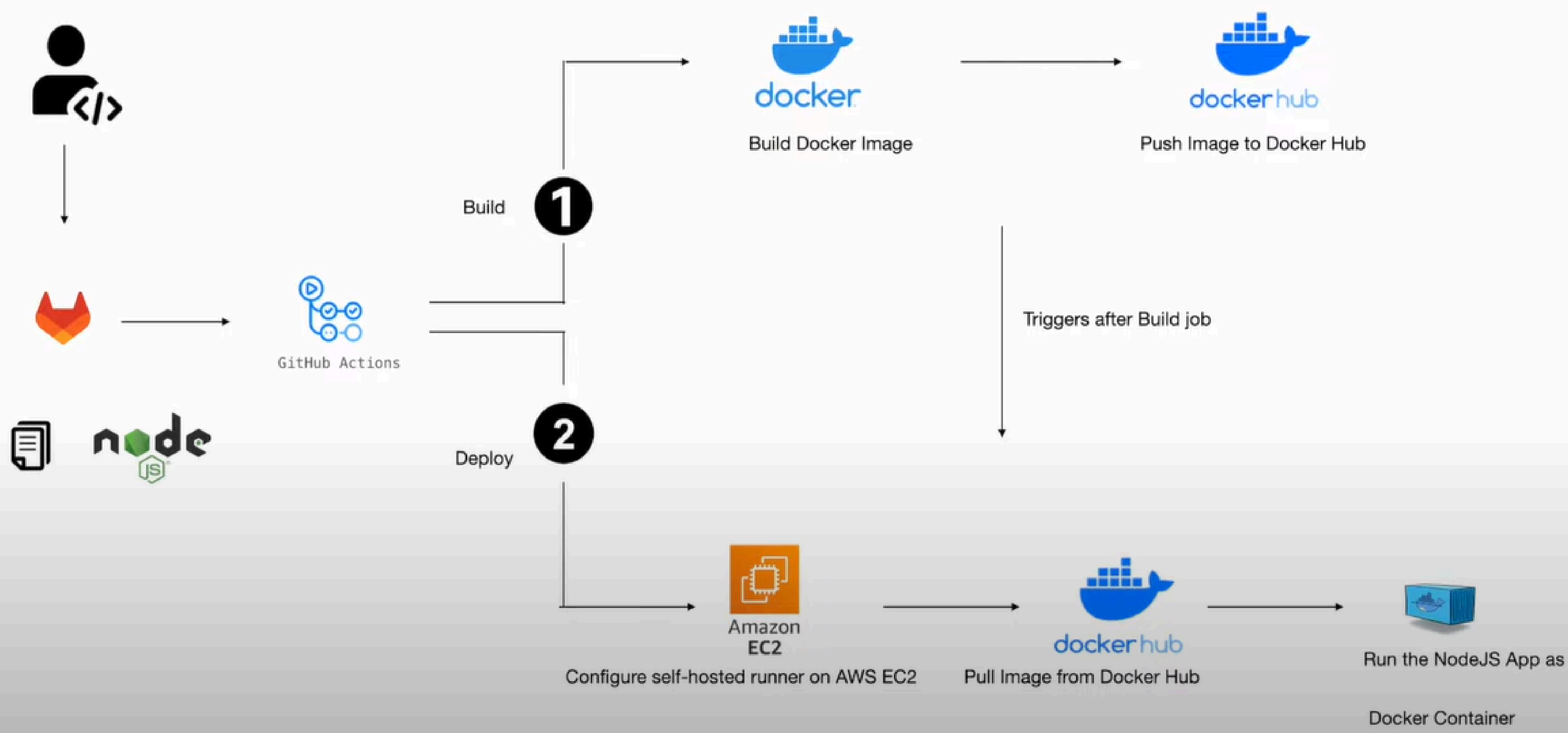


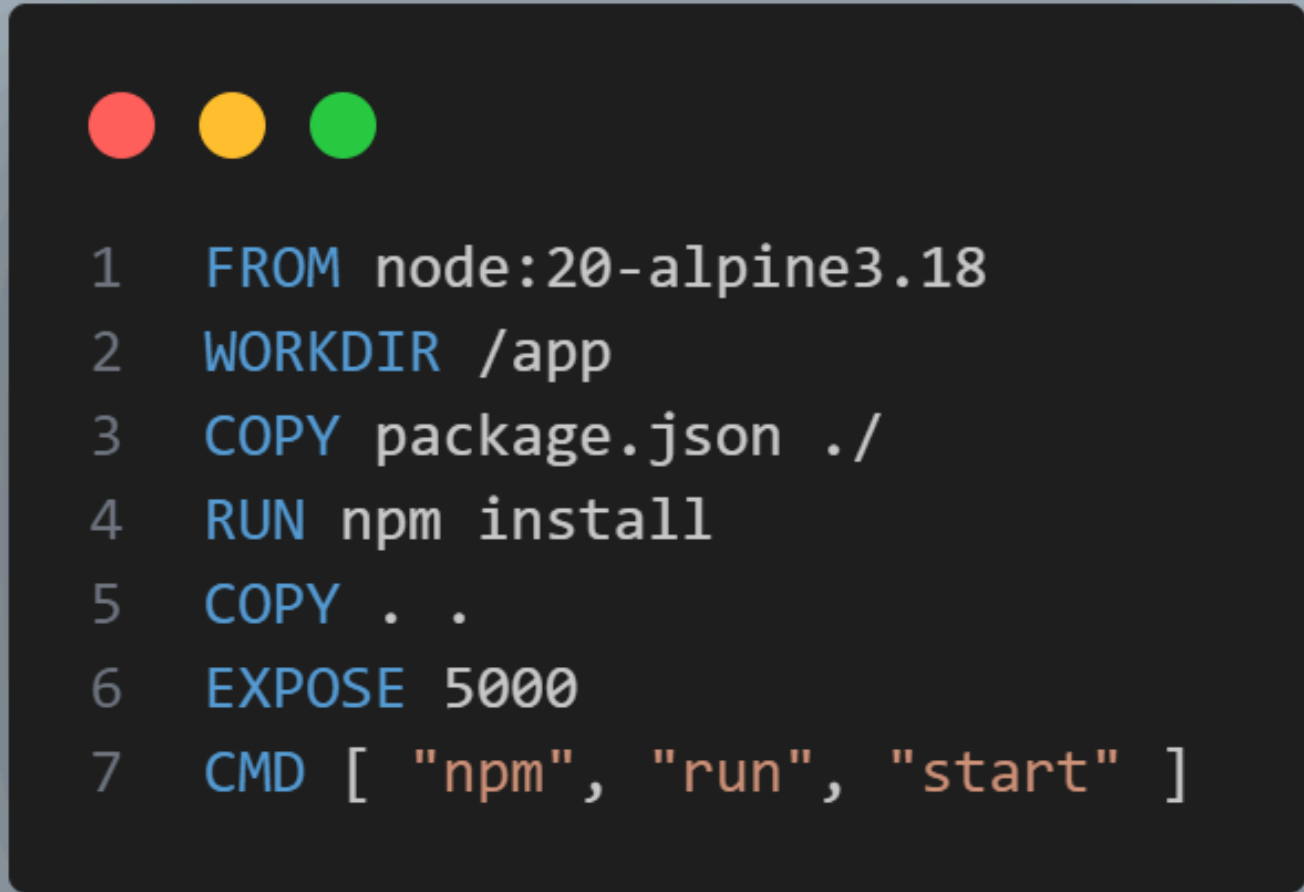
Deploy NodeJS Application to AWS EC2 Using GitLab CI





ให้เตรียมไฟล์ Dockerfile ใน Project

```
FROM node:20-alpine3.18
WORKDIR /app
COPY package.json ./
RUN npm install
COPY . .
EXPOSE 5000
CMD [ "npm", "run", "start" ]
```

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The terminal displays a Dockerfile with seven lines of code, each preceded by a line number from 1 to 7. The code is color-coded: 'FROM' is blue, 'WORKDIR' is blue, 'COPY' is blue, 'RUN' is blue, 'EXPOSE' is blue, and 'CMD' is blue. The arguments in the 'CMD' line are in orange.

```
1 FROM node:20-alpine3.18
2 WORKDIR /app
3 COPY package.json ./
4 RUN npm install
5 COPY . .
6 EXPOSE 5000
7 CMD [ "npm", "run", "start" ]
```

ให้เตรียมไฟล์ .gitlab-ci.yml ใน Project

default:

```
image: docker:24.0.5
services:
  - docker:24.0.5-dind
before_script:
  - docker info
```

stages:

```
- build
- deploy
```

build_job:

```
stage: build
script: |
  echo "Building the application.."
  docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
  docker build -t tongchaiiii/sbtvc-register-student-api:$CI_COMMIT_SHA .
  docker push tongchaiiii/sbtvc-register-student-api:$CI_COMMIT_SHA
```

deploy_job:

```
stage: deploy
script: |
  docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
  docker rm -f sbtvc-register-student-api-container
  docker image prune -a -f
  docker pull tongchaiiii/sbtvc-register-student-api:$CI_COMMIT_SHA
  docker run -d -p 5000:5000 --name sbtvc-register-student-api-container tongchaiiii/sbtvc-register-student-api:$CI_COMMIT_SHA
```

tags:

```
- ec2-runner
```

only:

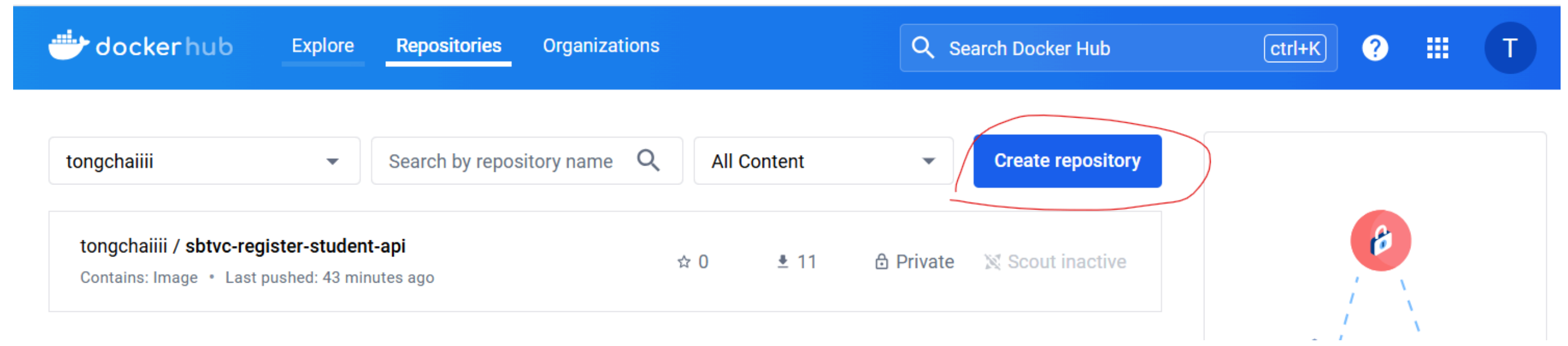
```
- main #จะ deploy เฉพาะ branch main
```

```
1 default:
2   image: docker:24.0.5
3   services:
4     - docker:24.0.5-dind
5   before_script:
6     - docker info
7
8   stages:
9     - build
10    - deploy
11
12   build_job:
13     stage: build
14     script: |
15       echo "Building the application.."
16       docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
17       docker build -t tongchaiiii/sbtvc-register-student-api:$CI_COMMIT_SHA .
18       docker push tongchaiiii/sbtvc-register-student-api:$CI_COMMIT_SHA
19
20   deploy_job:
21     stage: deploy
22     script: |
23       docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
24       docker rm -f sbtvc-register-student-api-container
25       docker image prune -a -f
26       docker pull tongchaiiii/sbtvc-register-student-api:$CI_COMMIT_SHA
27       docker run -d -p 5000:5000 --name sbtvc-register-student-api-container tongchaiiii/sbtvc-register-student-api:$CI_COMMIT_SHA
28   tags:
29     - ec2-runner
30   only:
31     - main
```

การตั้งค่า \$DOCKER_USERNAME \$DOCKER_PASSWORD

```
docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
```

1. เข้าไปที่ docker hub สมัครสมาชิก <https://hub.docker.com/>
2. สร้าง repository



การตั้งค่า \$DOCKER_USERNAME \$DOCKER_PASSWORD

```
docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
```

3. เมื่อสร้างสำเร็จก็จะมี username และ password ให้ไปตั้งค่าที่ส่วนของ gitlab

The screenshot shows the GitLab CI/CD Settings page for a project named 'sbtvc-register-student-api'. The left sidebar is open to the 'Settings' menu, with 'CI/CD' selected. The main content area displays several settings sections, each with an 'Expand' button. Red annotations highlight the 'Settings' menu item and the 'Expand' button for the 'Variables' section.

- General pipelines
- Auto DevOps
- Runners
- Artifacts
- Variables
- Pipeline trigger tokens
- Deploy freezes
- Job token permissions

การตั้งค่า \$DOCKER_USERNAME \$DOCKER_PASSWORD

```
docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
```

4. เมื่อสร้างสำเร็จก็จะมี username และ password ให้ไปตั้งค่าที่ส่วนของ gitlab

The screenshot shows the GitLab CI/CD Variables configuration page. It includes a 'Variables' section with explanatory text and a list of attributes. Below this is a table of existing variables. Handwritten red annotations include a '1' pointing to the 'Collapse' button, a '2' in a circle pointing to the 'Add variable' button, and an arrow pointing from the 'Add variable' button to the table.

Variables

Variables store information that you can use in job scripts. Each project can define a maximum of 8000 variables. [Learn more.](#)

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?](#)

Variables can have several attributes. [Learn more.](#)

- Protected:** Only exposed to protected branches or protected tags.
- Masked:** Hidden in job logs. Must match masking requirements.
- Expanded:** Variables with `$` will be treated as the start of a reference to another variable.

CI/CD Variables </> 2 Reveal values Add variable

Key ↑	Value	Environments	Actions
DOCKER_PASSWORD Expanded	*****	All (default)	
DOCKER_USERNAME Masked Expanded	*****	All (default)	

การตั้งค่า \$DOCKER_USERNAME \$DOCKER_PASSWORD

```
docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
```

5. เมื่อสร้างสำเร็จก็จะมี username และ password ให้ไปตั้งค่าที่ส่วนของ gitlab

5.1 เพิ่มชื่อ key เป็น DOCKER_USERNAME

5.2 เพิ่มชื่อ value เป็น username ของ docker ที่สมัคร

5.3 เพิ่มชื่อ key เป็น DOCKER_PASSWORD

5.4 เพิ่มชื่อ value เป็น password ของ docker ที่สมัคร

Add variable [X]

Type: Variable (default)

Environments: All (default)

Visibility: Visible (Can be seen in job logs.)
 Masked (Masked in job logs but value can be revealed in CI/CD settings. Requires values to meet regular expressions requirements.)

Flags: Protect variable (Export variable to pipelines running on protected branches and tags only.)
 Expand variable reference (\$ will be treated as the start of a reference to another variable.)

Description (optional): [Empty text box]

The description of the variable's value or usage.

Key: 1

You can use CI/CD variables with the same name in different places, but the variables might overwrite each other. [What is the order of precedence for variables?](#)

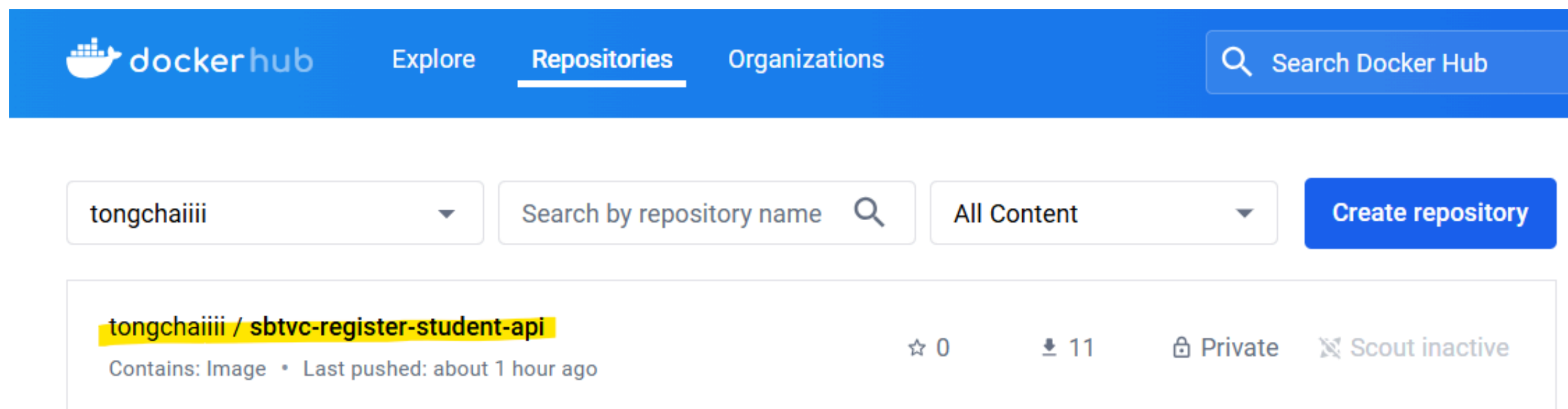
Value: 2

[Add variable] [Cancel]

การนำข้อมูลจาก Docker hub เข้ามาใช้งานในการตั้งค่าไฟล์

การนำค่าข้อมูลมาใช้งาน tongchaiiii/sbtvc-register-student-api

6. ไปนำข้อมูลชื่อ space และ ชื่อของ repository



The screenshot shows the Docker Hub interface. At the top, there is a blue navigation bar with the Docker Hub logo, 'Explore', 'Repositories' (underlined), and 'Organizations'. A search bar on the right contains the text 'Search Docker Hub'. Below the navigation bar, there is a search filter section with a dropdown menu set to 'tongchaiiii', a search input field containing 'Search by repository name', a dropdown menu set to 'All Content', and a blue 'Create repository' button. Below this, a search result is displayed for the repository 'tongchaiiii / sbtvc-register-student-api'. The repository name is highlighted in yellow. Below the name, it says 'Contains: Image • Last pushed: about 1 hour ago'. To the right of the repository name, there are icons for stars (0), downloads (11), a lock icon for 'Private', and a crossed-out flag icon for 'Scout inactive'.

การตั้งค่าใน server ของ AWS

7. login ใน AWS https://aws.amazon.com/marketplace/management/signin?ref_=footer_nav_management_portal

8. เข้าที่ EC2

The screenshot displays the AWS Management Console Home page. At the top, there is a navigation bar with the AWS logo, a search bar, and user information (Tokyo, tongchaiiii). Below the navigation bar, the main content area is divided into two columns. The left column features a 'Recently visited' widget with a list of services: EC2, S3, RDS, CloudFront, Support, VPC, and AWS Health Dashboard. A yellow arrow points to the EC2 icon in this list. The right column features an 'Applications (0)' widget with a 'Create application' button and a message: 'No applications. Get started by creating an application.' with another 'Create application' button below it. The page also includes a 'Reset to default layout' button and an '+ Add widgets' button in the top right corner.

การตั้งค่าใน server ของ AWS

9. คือ ข้อมูลที่ server เรามีอยู่ใน instances (running)

10. สร้าง instances

The screenshot displays the AWS Management Console interface for the EC2 service. The top navigation bar includes the AWS logo, a search bar, and the current region (Tokyo) and user (tongchaiiii). The left sidebar shows the navigation menu with categories like EC2 Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area is divided into several panels:

- Resources:** A summary of EC2 resources in the Asia Pacific (Tokyo) Region. A table shows the following counts:

Instances (running)	1	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1	Key pairs	1
Load balancers	0	Placement groups	0	Security groups	3
Snapshots	0	Volumes	1		
- Launch instance:** A panel with a prominent orange "Launch instance" button, which is highlighted by a yellow arrow. Below it is a "Migrate a server" button and a note: "Note: Your instances will launch in the Asia Pacific (Tokyo) Region".
- Service health:** Shows the AWS Health Dashboard for the Asia Pacific (Tokyo) region, with a status of "This service is operating normally."
- Zones:** A table listing available zones in the region:

Zone name	Zone ID
ap-northeast-1a	apne1-az4
- Account attributes:** Displays account information such as the Default VPC (vpc-045bbe2fd496797c7) and various settings like Data protection and security, Zones, and EC2 Serial Console.
- Instance alarms:** A panel with a "View in CloudWatch" button.
- Explore AWS:** Promotional banners for better price performance, Amazon GuardDuty Malware Protection, and saving up to 90% on EC2 with Spot Instances.

การตั้งค่าใน server ของ AWS

9. คือ ข้อมูลที่ server เรามีอยู่ใน instances (running)

10. สร้าง instances

1. ตั้งชื่อ โพรเจค
2. เลือก Server เป็น ubuntu
3. Key pair เลือกเป็นของเรา ถ้าไม่มีก็ create
4. Launch instance เพื่อสร้าง

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name
 [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents | **Quick Start**

Amazon Linux | macOS | **Ubuntu** | Windows | Red Hat | S

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Architecture: AMI ID: Verified provider

Instance type [Info](#) | [Get advice](#)

Instance type: Free tier eligible All generations [Compare instance types](#)

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0198 USD per Hour
On-Demand SUSE base pricing: 0.0152 USD per Hour
On-Demand RHEL base pricing: 0.0296 USD per Hour
On-Demand Linux base pricing: 0.0152 USD per Hour

[Additional costs apply for AMIs with pre-installed software](#)

Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required: [Create new key pair](#)

Network settings [Info](#) [Edit](#)

Summary

Number of instances: [Info](#)

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Review commands](#)

การตั้งค่าใน server ของ AWS

11. .กดเข้าที่ ID ของมัน

The screenshot shows the AWS Management Console interface for the EC2 service. The top navigation bar includes the AWS logo, a search bar, and the user's profile. The left sidebar contains navigation options for EC2, including the EC2 Dashboard, Global View, Events, and Instances. The main content area displays the 'Instances (1)' page, which includes a search bar, a filter for 'Instance state = running', and a table of instances. A red circle highlights the search bar, and a red arrow points to the Instance ID 'i-050f338805a225d3e' in the table.

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
<input type="checkbox"/>	sbtvc-register-...	i-050f338805a225d3e	Running	t2.micro	2/2 checks passed	View alarms +	ap-northeast-1a	ec2-54-25...

การตั้งค่าใน server ของ AWS

12. เข้าไปตั้งค่าความปลอดภัย

EC2 > Instances > i-050f338805a225d3e

Instance summary for i-050f338805a225d3e (sbtvc-register-student-api) Info

Updated less than a minute ago

Connect Instance state Actions

Instance ID i-050f338805a225d3e (sbtvc-register-student-api)	Public IPv4 address 54.250.238.113 open address	Private IPv4 addresses 172.31.34.145
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-250-238-113.ap-northeast-1.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-34-145.ap-northeast-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-34-145.ap-northeast-1.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 54.250.238.113 [Public IP]	VPC ID vpc-045bbe2fd496797c7	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-0f36f375a118f5ea6	
IMDSv2 Required	Instance ARN arn:aws:ec2:ap-northeast-1:212105134622:instance/i-050f338805a225d3e	

Details | Status and alarms | Monitoring | **Security** | Networking | Storage | Tags

▼ Security details

IAM Role -	Owner ID 212105134622	Launch time Fri Jul 05 2024 21:55:56 GMT+0700 (Indochina Time)
Security groups sg-09acc89f60d2ed92d (launch-wizard-2)		

▼ Inbound rules

การตั้งค่าใน server ของ AWS

13. เข้าไปที่ Edit Inbound rules

The screenshot shows the AWS Management Console interface for a Security Group. The breadcrumb navigation is [EC2](#) > [Security Groups](#) > [sg-09acc89f60d2ed92d - launch-wizard-2](#). The main title is **sg-09acc89f60d2ed92d - launch-wizard-2** with an **Actions** dropdown menu.

Details

Security group name launch-wizard-2	Security group ID sg-09acc89f60d2ed92d	Description launch-wizard-2 created 2024-07-05T14:54:27.458Z	VPC ID vpc-045bbe2fd496797c7
Owner 212105134622	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules (2) Manage tags Edit inbound rules

Search

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-09b7c7049359adfca	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-09508142042c5d...	IPv4	Custom TCP	TCP	5000	0.0.0.0/0	-

A red circle highlights the VPC ID `vpc-045bbe2fd496797c7` in the details section, and a red arrow points from it to the **Edit inbound rules** button in the Inbound rules section.

การตั้งค่าใน server ของ AWS

14. เข้าไปที่ Edit Inbound rules

1. กดที่ Add rule
2. Type เลือกเป็น Custom TCP
3. Port range 5000
4. ตัวของ IP เป็น 0.0.0.0-0
5. กด Save rules

EC2 > Security Groups > sg-09acc89f60d2ed92d - launch-wizard-2 > Edit inbound rules

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-09b7c7049359adfca	SSH	TCP	22	Custom	<input type="text"/>	<input type="button" value="Delete"/>
sgr-09508142042c5d5a9	Custom TCP	TCP	5000	Custom	<input type="text"/>	<input type="button" value="Delete"/>

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

การเชื่อมต่อไปที่ใช้ในการใช้ Docker server ของ AWS

16. ย้อนกลับมาที่หน้าหลักเข้าไปกด ที่ connect และ กด connect อีกทีหนึ่ง และจะได้หน้าจอ command

The screenshot displays the AWS Management Console interface. At the top, the navigation bar includes the AWS logo, a search bar, and the user's profile. The left sidebar shows the navigation menu with categories like 'Instances' and 'Images'. The main content area shows the 'Instances (1/1)' page. A table lists the instance details, with the 'Status check' column showing '2/2 checks passed'. A red circle highlights this status, and a red arrow points to the 'Connect' button in the table's action column. Below the table, the 'Connect to instance' dialog is open, showing options for connection type, public IP address, and username. A red circle highlights the 'Connect' button at the bottom right of the dialog.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
sbtvc-register-...	i-050f338805a225d3e	Running	t2.micro	2/2 checks passed	View alarms +	ap-northeast-1a	ec2-54-250-238-113.ap...

Connect to instance

Instance ID: i-050f338805a225d3e (sbtvc-register-student-api)

Connection Type:

- Connect using EC2 Instance Connect (selected)
- Connect using EC2 Instance Connect Endpoint

Public IP address: 54.250.238.113

Username: ubuntu

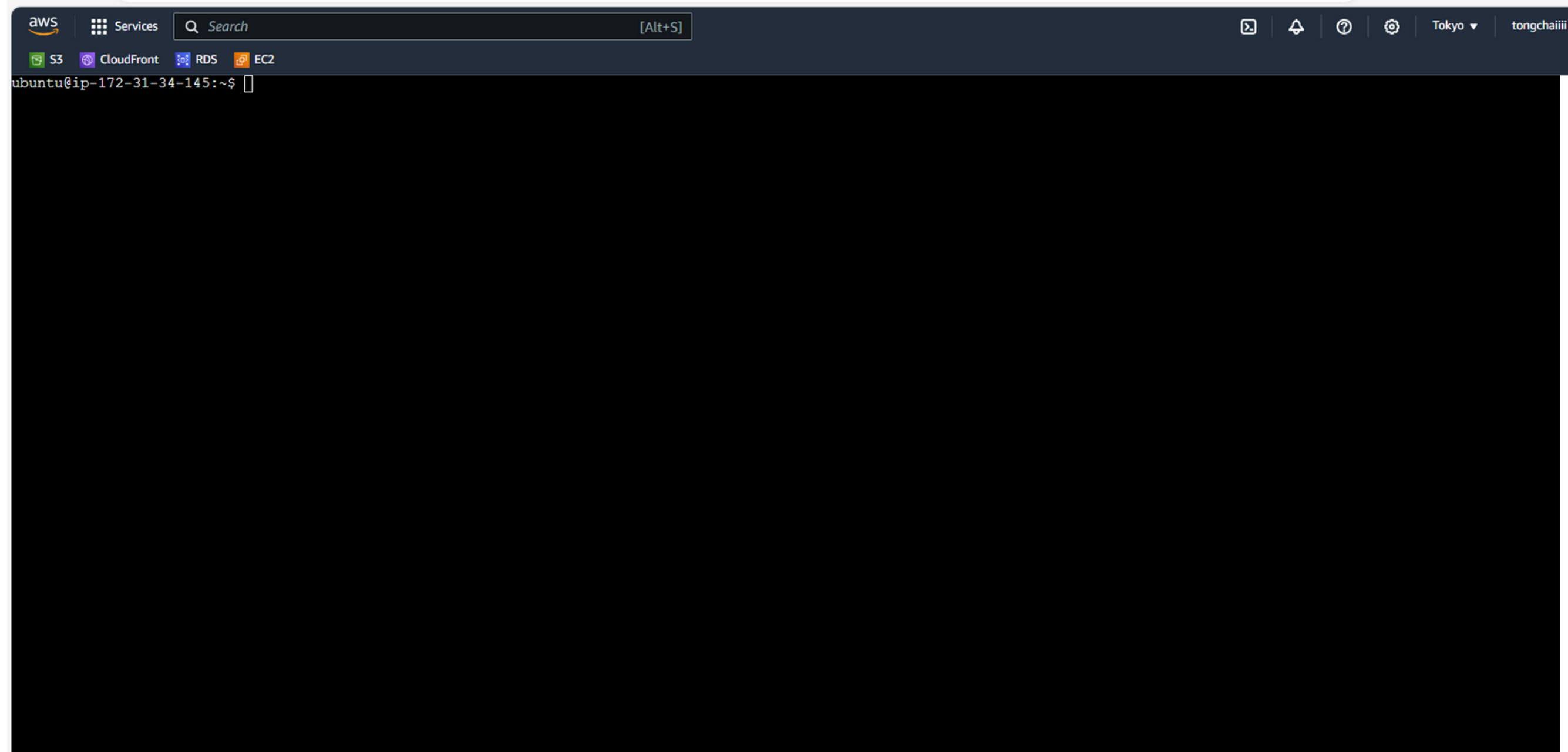
Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Buttons: Cancel, Connect

การเชื่อมต่อไปที่ในการใช้ Docker server ของ AWS

17. ให้ใช้คำสั่งต่อไปนี้ใน command เพื่อติดตั้งการที่เราจะใช้งาน docker

```
sudo apt-get update && sudo apt-get install docker.io -y && sudo systemctl start docker && sudo chmod 666 /var/run/docker.sock && sudo systemctl enable docker && docker --version
```



การสร้าง CI/CD เข้าไปยัง Docker

18. ไปตั้งค่า runner CI/CD กดที่คำว่า new project runner

The screenshot shows the GitLab CI/CD Settings page for a project named 'sbtvc-register-student-api'. The left sidebar contains a navigation menu with 'CI/CD' highlighted. A red circle labeled '1' is drawn around the 'CI/CD' menu item. The main content area is titled 'CI/CD Settings' and includes a search bar. Below the search bar are three sections: 'General pipelines' (with an 'Expand' button), 'Auto DevOps' (with an 'Expand' button), and 'Runners' (with a 'Collapse' button). The 'Runners' section contains the following text: 'Runners are processes that pick up and execute CI/CD jobs for GitLab. [What is GitLab Runner?](#) Register as many runners as you want. You can register runners as separate users, on separate servers, and on your local machine. How do runners pick up jobs? Runners are either:

- **active** - Available to run jobs.
- **paused** - Not available to run jobs.

Tags control which type of jobs a runner can handle. By tagging a runner, you make sure runners only handle the jobs they are equipped to run. [Learn more.](#)

Below this text are two columns: 'Project runners' and 'Instance runners'. The 'Project runners' column has a blue box with the text 'These runners are assigned to this project.' and a button labeled 'New project runner' with a three-dot menu icon. A red circle labeled '2' is drawn around this button, with a red arrow pointing to it. The 'Instance runners' column has a blue box with the text 'These runners are available to all groups and projects. Each CI/CD job runs on a separate, isolated virtual machine.' Below this is a toggle switch for 'Enable instance runners for this project' which is turned on. Underneath, it says 'Available instance runners: 100' and lists three runners with their IDs and hostnames: '#11573930 (KzYhZxBvi) 1-blue.shared-gitlab-org.runners-manager.gitlab.com', '#11573990 (NL4gfoBem) 2-blue.shared-gitlab-org.runners-manager.gitlab.com', and '#11574038 (sUrYYgEG8)'.

การสร้าง CI/CD เข้าไปยัง Docker

19. ตั้งชื่อ runner และทำการ create runner

New project runner

Create a project runner to generate a command that registers the runner with all its configurations.

Tags

Tags

Add tags to specify jobs that the runner can run. [Learn more.](#)

Separate multiple tags with a comma. For example, `macos, shared`.

Run untagged jobs

Use the runner for jobs without tags in addition to tagged jobs.

Configuration (optional)

Runner description

Paused

Stop the runner from accepting new jobs.

Protected

Use the runner on pipelines for protected branches only.

Lock to current projects 

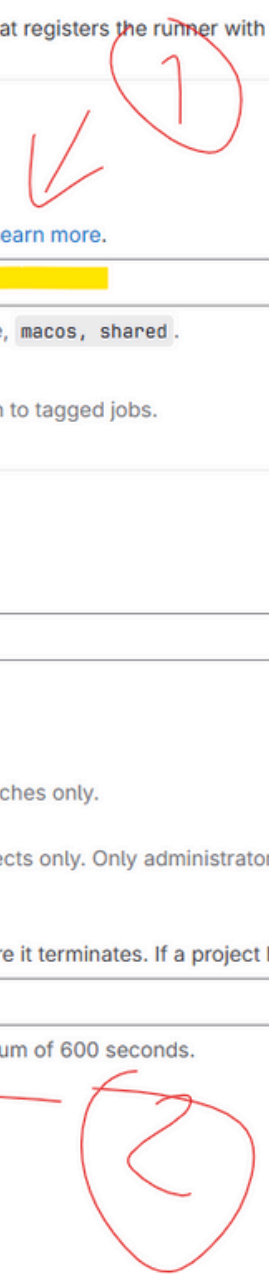
Use the runner for the currently assigned projects only. Only administrators can change the assigned projects.

Maximum job timeout

Maximum amount of time the runner can run before it terminates. If a project has a shorter job timeout period, the job timeout period of the instance runner is used instead.

Enter the job timeout in seconds. Must be a minimum of 600 seconds.

Create runner



การสร้าง CI/CD เข้าไปยัง Docker

20. กดเลือกเป็น linux และทำการ กด ที่ How do i install Gitlab Runner?

ต่อมา ก็อป install ส่วนที่ 2 ไปที่หน้า command AWS

Runner created.

Register runner

Platform

Operating systems

Linux macOS Windows

Cloud

Google Cloud

Containers

Docker Kubernetes

GitLab Runner must be installed before you can register a runner. [How do I install GitLab Runner?](#)

Step 1

Copy and paste the following command into your command line to register the runner.

```
$ gitlab-runner register
--url https://gitlab.com
--token glrt-a-sDExybuZiW2Tday18c
```

Install GitLab Runner

Select platform specifications to install GitLab Runner.

Architecture

amd64

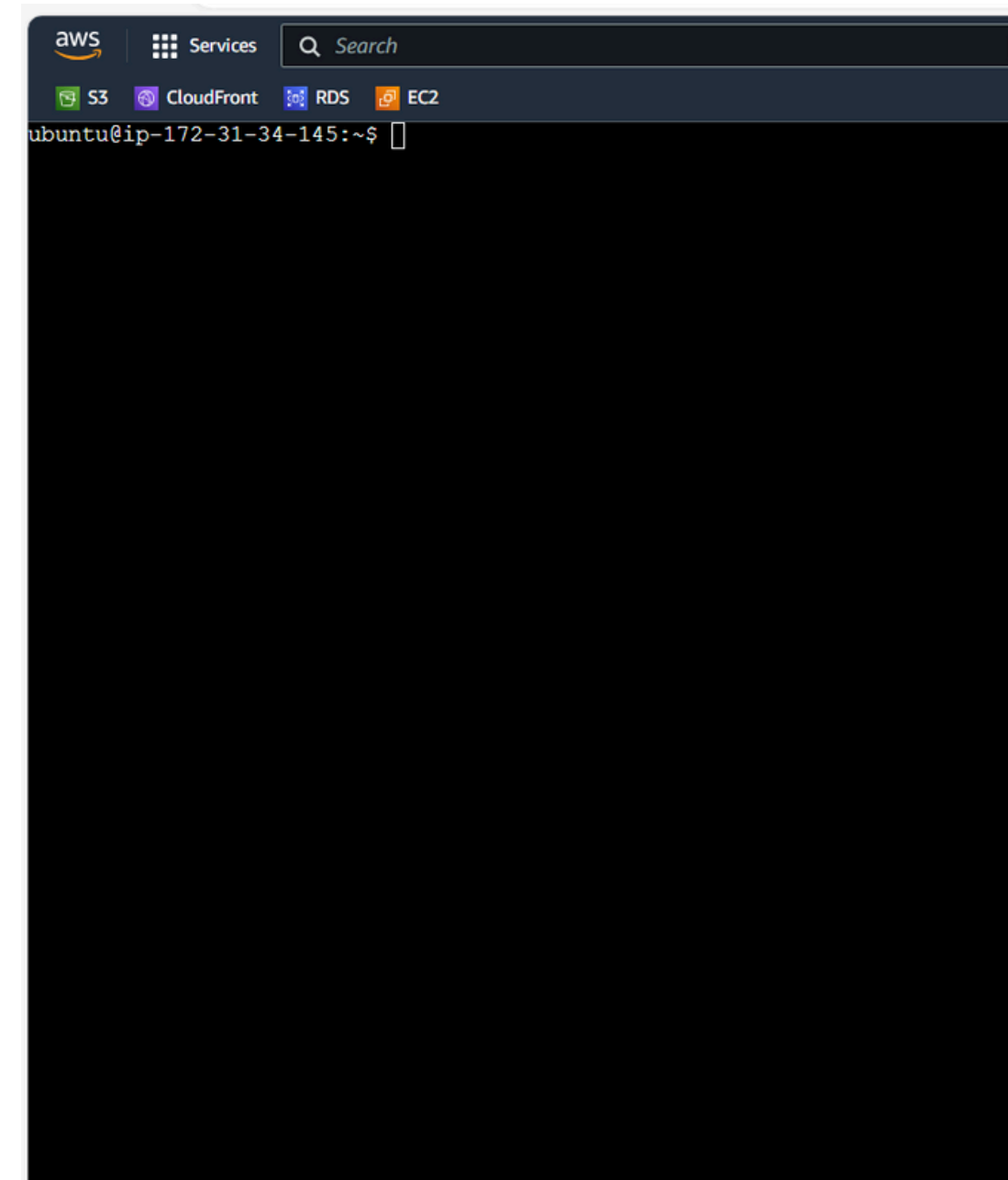
```
# Download the binary for your system
sudo curl -L --output /usr/local/bin/g

# Give it permission to execute
sudo chmod +x /usr/local/bin/gitlab-ru

# Create a GitLab Runner user
sudo useradd --comment 'GitLab Runner

# Install and run as a service
sudo gitlab-runner install --user=gitl
sudo gitlab-runner start
```

See more [installation methods and architectures](#).



การสร้าง CI/CD เข้าไปยัง Docker

21. กดเลือกเป็น linux และทำการ ก๊อปปี้ code step 1 นั้นไปวางหน้า ของ aws ให้ก๊อปปี้ เลข 1 เลข และไปทำข้อที่ 21 ก่อนที่จะกลับมา view runners

Register runner

Platform

Operating systems

Linux macOS Windows

Cloud

Google Cloud

Containers

Docker Kubernetes

GitLab Runner must be installed before you can register a runner. [How do I install GitLab Runner?](#)

Step 1

Copy and paste the following command into your command line to register the runner.

```
$ gitlab-runner register
--url https://gitlab.com
--token glrt-xxWPyfQaiAYgUm61m_zs
```

The runner authentication token `glrt-xxWPyfQaiAYgUm61m_zs` displays here for a short time only. After you register the runner, this token is stored in the `config.toml` and cannot be accessed again from the UI.

Step 2

Choose an executor when prompted by the command line. Executors run builds in different environments. [Not sure which one to select?](#)

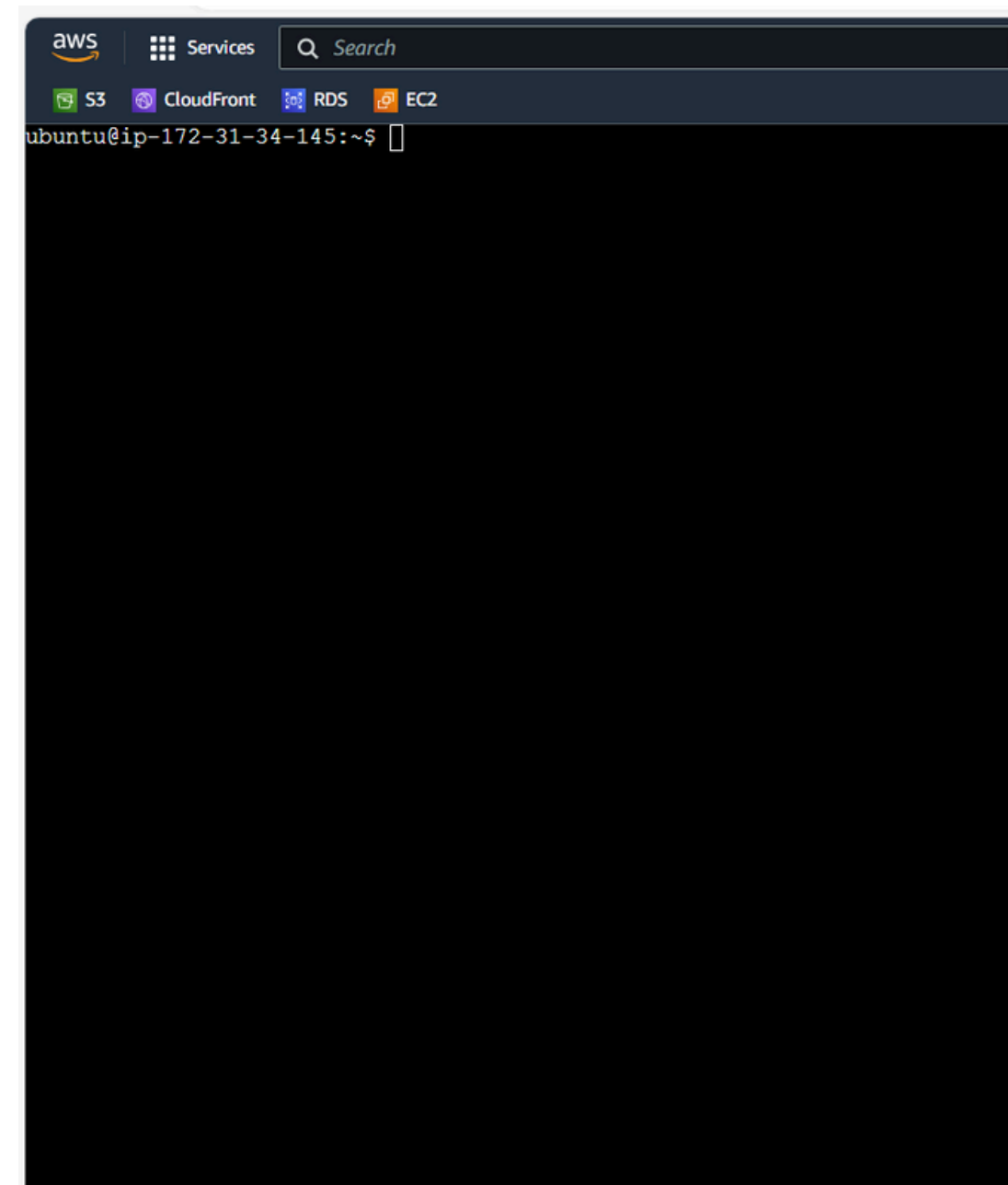
Step 3 (optional)

Manually verify that the runner is available to pick up jobs.

```
$ gitlab-runner run
```

This may not be needed if you manage your runner as a system or user service.

[View runners](#)



การสร้าง CI/CD เข้าไปยัง Docker

22. ต่อมา ใช้คำสั่ง sudo su เพื่อเข้าไปที่ home/ubuntu

ต่อมา copy จากข้อที่ 21 ที่หมายเลข 1 แล้วทำการวาง แล้วก็ enter และ ทำการ enter อีกครั้ง

ต่อมาใช้คำสั่ง เรียกหาที่เราสร้างก็คือ ec2-runner ecnter และกด shell อีกที enter

ต่อมาใช้คำสั่ง sudo chmod u+r /home/gitlab-runner/.bashrc

sudo systemctl restart gitlab-runner.service

ต่อมาย้อนกลับไปข้อที่ 20 แล้วกด view runner

```
WARNING: $ gitlab-runner run
WARNING: Use sudo for system-mode:
WARNING: $ sudo gitlab-runner...

Enter the GitLab instance URL (for example, https://gitlab.com/):
[https://gitlab.com]:
Verifying runner... is valid runner=F7M3sLj-6
Enter a name for the runner. This is stored only in the local config.toml file:
[ip-172-31-34-145]: sudo su
Enter an executor: ssh, docker-windows, docker+machine, instance, kubernetes, docker-autoscaler, custom, shell, parallels, virtualbox, docker:
s^C^CUnregistering runner from GitLab succeeded runner=F7M3sLj-6
FATAL: RECEIVED SIGNAL: interrupt
ubuntu@ip-172-31-34-145:~$ ^C
ubuntu@ip-172-31-34-145:~$ sudo
usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABkNnS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABkNnS] [-g group] [-h host] [-p prompt] [-U user]
[-u user] [command [arg ...]]
usage: sudo [-ABbEHkNnPS] [-r role] [-t type] [-C num] [-D directory]
[-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
[-u user] [VAR=value] [-i | -s] [command [arg ...]]
usage: sudo -e [-ABkNnS] [-r role] [-t type] [-C num] [-D directory]
[-g group] [-h host] [-p prompt] [-R directory] [-T timeout]
[-u user] file ...
ubuntu@ip-172-31-34-145:~$ sudo su
root@ip-172-31-34-145:/home/ubuntu# gitlab-runner register --url https://gitlab.com --token glrt-F7M3sLj-6-HiXw4HAqyy
Runtime platform arch=amd64 os=linux pid=11996 revision=fe451d5a version=17.1.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
[https://gitlab.com]:
Verifying runner... is valid runner=F7M3sLj-6
Enter a name for the runner. This is stored only in the local config.toml file:
[ip-172-31-34-145]: ec2-runner
Enter an executor: shell, ssh, parallels, docker, docker+machine, instance, custom, docker-windows, kubernetes, docker-autoscaler, virtualbox:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!

Configuration (with the authentication token) was saved in "/etc/gitlab-runner/config.toml"
root@ip-172-31-34-145:/home/ubuntu#
```

ตรวจสอบ view runner

23. ต่อมาย้อนกลับไปข้อที่ 21 แล้วกด view runner

The screenshot shows the GitLab Runners configuration page. It is divided into two main sections: Project runners and Instance runners. The Project runners section includes a 'New project runner' button and a list of 'Assigned project runners'. One runner is listed with ID #39844742 (F7M3sLj-6) and tag ec2-runner. A red hand-drawn circle highlights this runner, and a red arrow points from the circle to the 'Remove runner' button. The Instance runners section shows a toggle switch for 'Enable instance runners for this project' which is turned on, and a list of 'Available instance runners' with ID #11573930 (KzYhZxBvi) and tag gitlab-org.

Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. [What is GitLab Runner?](#)

Register as many runners as you want. You can register runners as separate users, on separate servers, and on your local machine.

How do runners pick up jobs?

Runners are either:

- **active** - Available to run jobs.
- **paused** - Not available to run jobs.

Tags control which type of jobs a runner can handle. By tagging a runner, you make sure runners only handle the jobs they are equipped to run. [Learn more.](#)

Project runners

These runners are assigned to this project.

[New project runner](#) ⋮

Assigned project runners

- #39844742 (F7M3sLj-6)
ec2-runner [Remove runner](#)

Instance runners

These runners are available to all groups and projects.

Each CI/CD job runs on a separate, isolated virtual machine.

Enable instance runners for this project

Available instance runners: 100

- 🔗 #11573930 (KzYhZxBvi)
1-blue.shared-gitlab-org.runners-manager.gitlab.com
gitlab-org

ถ้าเกิดปัญหาให้ลบ

```
sudo rm /etc/gitlab-runner/config.toml
```

```
sudo rm /etc/systemd/system/gitlab-runner.service
```

และไปเริ่มข้อที่ 20 ใหม่อีกครั้ง

```
gitlab-runner status
```

```
gitlab-runner stop
```

```
gitlab-runner start
```

```
gitlab-runner restart
```

Deploy NodeJS Application to AWS EC2 Using GitLab CI (youtube.com)

<https://www.youtube.com/watch?v=hmbeksdTAls>

ทำรูป CI/CD ทีละขั้นตอนให้เข้าใจง่ายที่สุด

