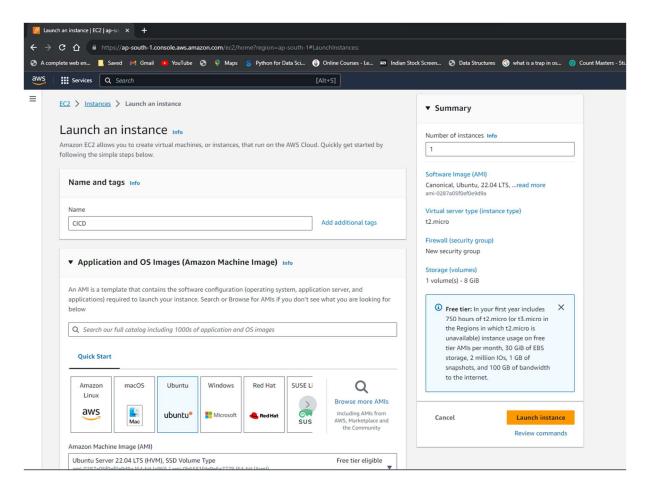# CI/CD Pipeline using Jenkins, AWS & Docker
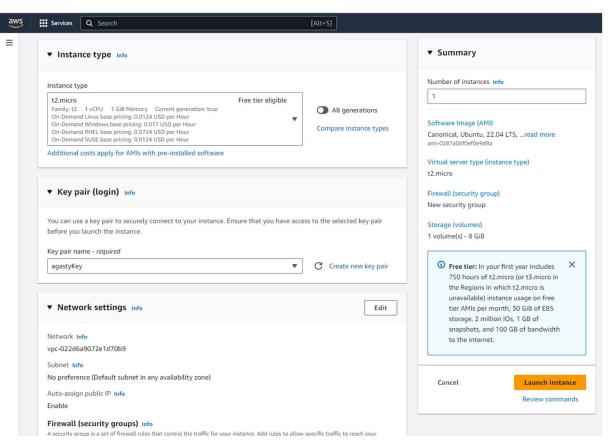
Brief Overview of the Project:

- Creating EC2 Instance using Ubuntu as the platform

- Installing Java on the instance

- Installing Jenkins on the instance

- Setting Inbound rules for EC2 to enable accessibility of Jenkins through port 8080

- Connecting to Jenkins

- Dockerfile is added to the GitHub repository for the project code.

- Linking GitHub repository to Jenkins and enabling webhooks.

- Building Docker Image for the project using Dockerfile and Running it in container.

Any change made further on from here in the code (GitHub repo will trigger automatic build of the Jenkins pipeline)

- Creating EC2 Instance:

aws   Services   Search   [Alt+S]

EC2 > Instances > Launch an instance

Launch an instance Info

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

| CICD |

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

| Search our full catalog including 1000s of application and OS images |

Quick Start

| Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li | Q |
| aws | Mac | ubuntu | Microsoft | Red Hat | SUS | Browse more AMIs |

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type   Free tier eligible
ami-0287a05f0ef0e9d9a (64-bit (x86)) / ami-0b6581fde9e6e7779 (64-bit (Arm))

▼ Summary

Number of instances Info

| 1 |

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0287a05f0ef0e9d9a

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, 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

---

aws   Services   Search   [Alt+S]

▼ Instance type Info

Instance type

t2.micro   Free tier eligible
Family: t2   1 vCPU   1 GiB Memory   Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0724 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

⬤ All generations

Compare instance types

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*

| agastyKey | ↻ Create new key pair

▼ Network settings Info   Edit

Network Info
vpc-022d6a9072e1d70b9

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your

▼ Summary

Number of instances Info

| 1 |

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0287a05f0ef0e9d9a

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, 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

- Installing Java and Jenkins on the instance once it is set up and running:

- Changing Inbound Rules(adding port 8080-custom TCP) for the instance :

- Checking Status of Jenkins:

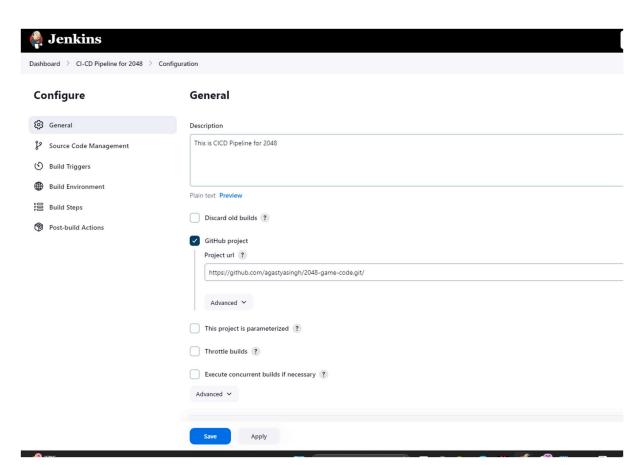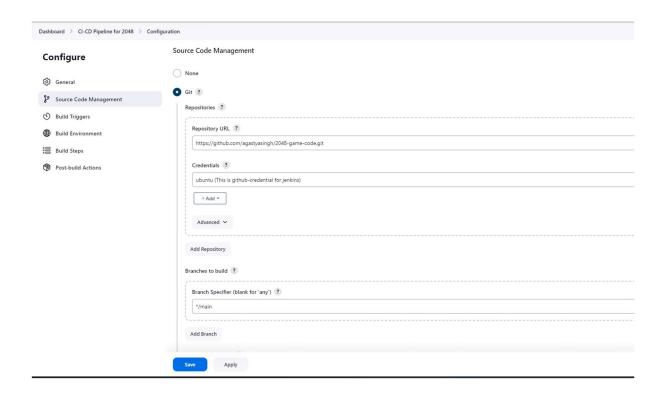- Connecting to Jenkins through IP address of the instance and port 8080 (in this case: 3.111.52.182:8080)

- Setting up the pipeline:

Configure

- ⚙ General
- ⌥ Source Code Management
- ⏱ Build Triggers
- 🌐 Build Environment
- ☰ Build Steps
- 📦 Post-build Actions

Source Code Management

○ None

● Git ?

Repositories ?

Repository URL ?

https://github.com/agastyasingh/2048-game-code.git

Credentials ?

ubuntu (This is github-credential for jenkins)

+ Add ▾

Advanced ⌄

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Save Apply

Configure

- ⚙ General
- ⌥ Source Code Management
- ⏱ Build Triggers
- 🌐 Build Environment
- ☰ Build Steps
- 📦 Post-build Actions

Source Code Management

○ None

● Git ?

Repositories ?

Repository URL ?

https://github.com/agastyasingh/2048-game-code.git

Credentials ?

ubuntu (This is github-credential for jenkins)

+ Add ▾

Advanced ⌄

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Save Apply
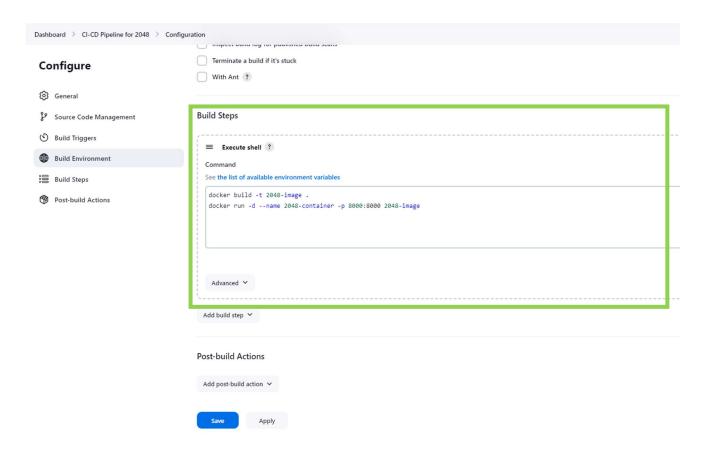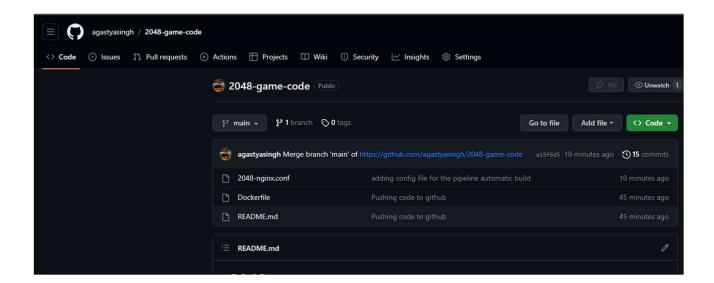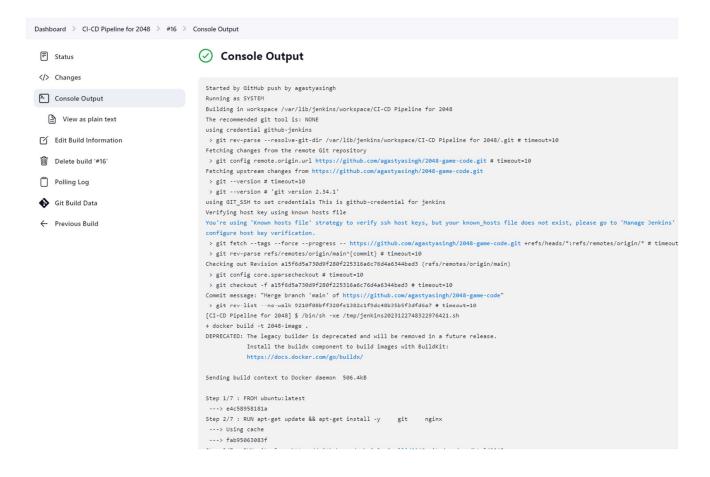
- Build Steps to build docker image when pipeline is triggered:

- Adding conf file will automatically trigger the build of pipeline:

✅ **Console Output**

```
Started by GitHub push by agastyasingh
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/CI-CD Pipeline for 2048
The recommended git tool is: NONE
using credential github-jenkins
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/CI-CD Pipeline for 2048/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/agastyasingh/2048-game-code.git # timeout=10
Fetching upstream changes from https://github.com/agastyasingh/2048-game-code.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
using GIT_SSH to set credentials This is github-credential for jenkins
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please go to 'Manage Jenkins'
configure host key verification.
 > git fetch --tags --force --progress -- https://github.com/agastyasingh/2048-game-code.git +refs/heads/*:refs/remotes/origin/* # timeout
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision a15f6d5a730d9f280f225316a6c76d4a6344bed3 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f a15f6d5a730d9f280f225316a6c76d4a6344bed3 # timeout=10
Commit message: "Merge branch 'main' of https://github.com/agastyasingh/2048-game-code"
 > git rev-list --no-walk 9210f08bff320fe1382c1f9dc48b35b5f3dfd6a7 # timeout=10
[CI-CD Pipeline for 2048] $ /bin/sh -xe /tmp/jenkins20231227483322976421.sh
+ docker build -t 2048-image .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  506.4kB

Step 1/7 : FROM ubuntu:latest
 ---> e4c58958181a
Step 2/7 : RUN apt-get update && apt-get install -y    git    nginx
 ---> Using cache
 ---> fab95063083f
```
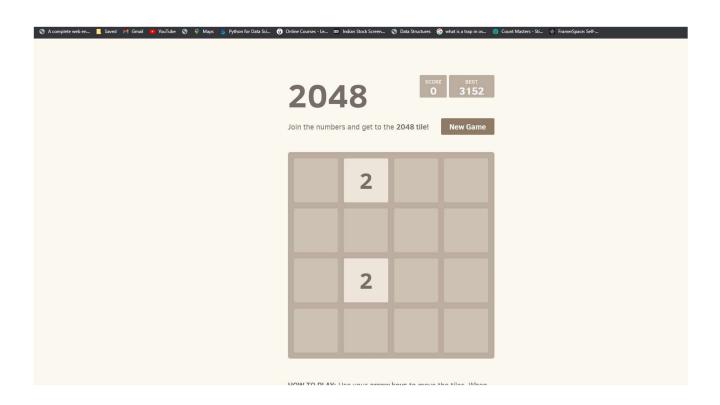
Build number 16:

So the build is successfully completed, hence the docker image is built and our project is up and running.

History of all the commands executed in the EC2 Shell:

2048-Game:

All the project goals have been completed and CI/CD Pipeline was successfully set-up and running.