



Devops on *AWS* for beginner

Instructor – Linh Nguyen

Engineering Consultant, AWS Cloud Solution Architect

CICD Sử dụng Jenkins



Jenkins

“Không có việc gì khó, chỉ sợ không biết làm!”

Copyright@Linh Nguyen on Udemy
All right reserved

Target

- Giới thiệu về Jenkins.
- Cách cài đặt một Jenkins server.
- Thiết lập các setting & plugin cơ bản trên Jenkins.
- Làm quen với Jenkins thông qua các bài lab từ dễ đến khó.
- Cách tạo các Job Build/Deploy trên Jenkins cho từng loại repository.

Copyright@Linh Nguyen on Udemy
All right reserved

Giới thiệu Jenkins.

Jenkins là một công cụ mã nguồn mở được sử dụng để triển khai và tự động hóa quy trình phát triển phần mềm. Nó cung cấp một môi trường tích hợp liên tục (CI) cho việc xây dựng, kiểm tra và triển khai ứng dụng.

Jenkins cho phép bạn tạo các công việc (jobs) để thực hiện các tác vụ như **build source**, chạy các **automation test**, **triển khai** ứng dụng và thông báo kết quả. Bằng cách tích hợp với các công cụ khác và sử dụng các plugin có sẵn, Jenkins có thể được tùy chỉnh và mở rộng để phù hợp với nhu cầu phát triển phần mềm của bạn.



Jenkins

Copyright@Linh Nguyen on Udemy
All right reserved

Tại sao Jenkins được sử dụng nhiều?

- **Dễ sử dụng:** Jenkins có giao diện người dùng thân thiện và dễ sử dụng. Người dùng có thể tạo và quản lý các công việc (jobs) một cách dễ dàng thông qua giao diện web.
- **Đa nền tảng:** Jenkins hỗ trợ nhiều nền tảng và ngôn ngữ lập trình khác nhau, cho phép phát triển ứng dụng trên nhiều môi trường và hệ điều hành (Windows, Linux, Mac, Docker).
- **Mở rộng và linh hoạt:** Jenkins có một hệ sinh thái mở rộng phong phú với hàng ngàn plugin có sẵn. Điều này cho phép mở rộng chức năng của Jenkins và tùy chỉnh quy trình CI/CD theo nhu cầu cụ thể của dự án.
- **Hỗ trợ cộng đồng:** Jenkins có một cộng đồng lớn và tích cực, với nhiều nguồn tài liệu, hướng dẫn và diễn đàn để giúp đỡ người dùng giải quyết các vấn đề và chia sẻ kinh nghiệm.



Jenkins

Copyright@Linh Nguyen on Udemy
All right reserved

Hướng dẫn cài đặt Jenkins trên EC2 (Ubuntu)

- Tạo một EC2 instance với AMI Ubuntu.
- Tiến hành login vào EC2 instance.
- Cài đặt Jenkins theo guide sau:

<https://www.jenkins.io/doc/book/installing/linux/#debianubuntu>

- Login vào Jenkins theo password admin,
- Cài đặt các plugin theo recommend.
- Change password admin.

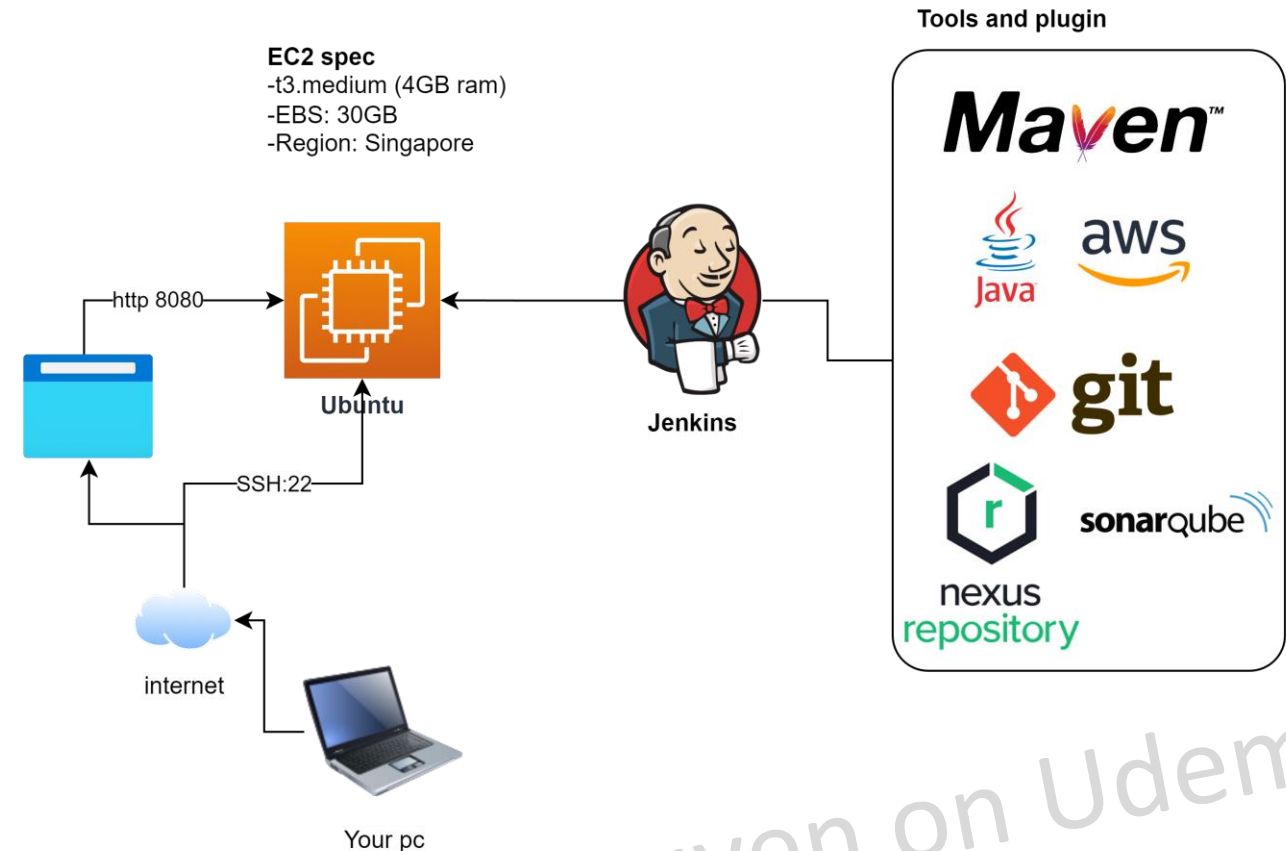
**Jenkins có thể hoạt động trên Windows, Mac OS, Docker,... các bạn có thể tự tìm hiểu thêm.*

Copyright@Linh Nguyen on Udemey
All right reserved

Hướng dẫn cài đặt Jenkins trên EC2 (Ubuntu)

Sơ đồ cho bài lab:

- Từ máy local kết nối tới jenkins dashboard port 8080
- Máy local có thể ssh vào jenkins (cho các tác vụ debug).
- Jenkins có thể cài sẵn các tool phục vụ CI/CD như Java, Maven, AWS, Git, Sonar, Nexus.



Copyright@Linh Nguyen on Udemy
All right reserved

Lab 1 - Jenkins basic – first hello job.

- Giới thiệu giao diện của Jenkins.
- Tạo một job Jenkins đầu tiên của bạn, chạy thử.
- Xem log & kết quả của job.
- Xem workplace của job.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 2 - Jenkins Variable

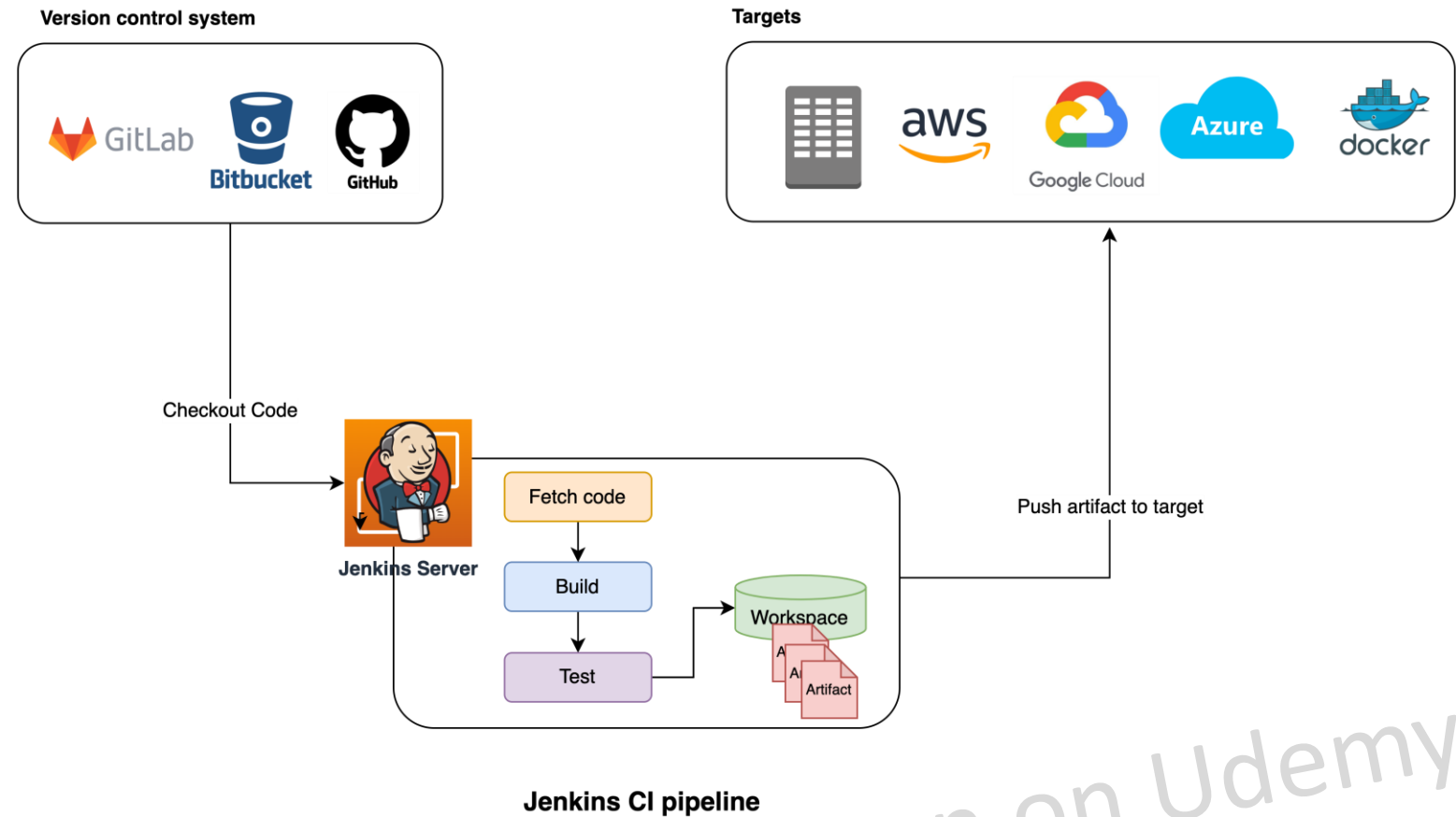
- Giới thiệu về Variable trong Jenkins.
- Built-in Variable.
- Job Variable.
- Credential variable

Copyright@Linh Nguyen on Udemy
All right reserved

Giới thiệu Jenkins CI/CD Concept

Về cơ bản một job CI của Jenkins sẽ có cấu trúc như sau:

1. Checkout source code
2. Build
3. Test
4. Push Artifact lên target

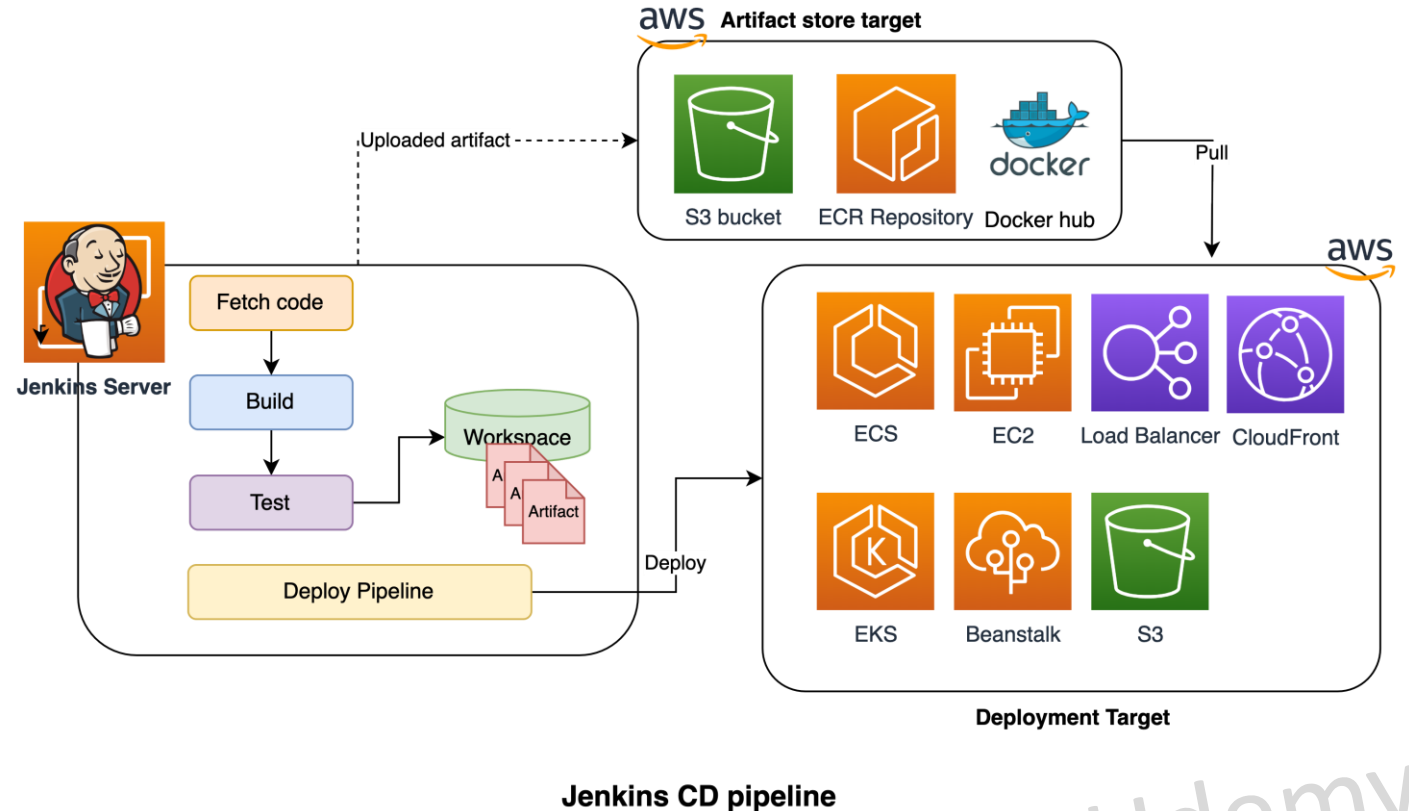


Copyright@Linh Nguyen on Udemy
All right reserved

Giới thiệu Jenkins CI/CD Concept

Về cơ bản một job CD của Jenkins sẽ có một hoặc kết hợp các action sau:

- Deploy package lên EC2, Beanstalk
- Deploy package lên ECS, EKS
- Deploy package lên S3, Clear cache trên CloudFront.
- Trigger rollout trên ECS, EKS Cluster.
- Switch version trên ALB (Blue/Green deployment).
- Trigger Canary deployment kết hợp với CodeDeploy.
- ...các action khác tùy platform.



Copyright@Linh Nguyen on Udemy
All right reserved

Lab 3 - Build Java Project (p1)

Yêu cầu:

- Cài đặt và cấu hình JDK cho server Jenkins.
- Cài đặt và cấu hình Maven cho server Jenkins.
- Tạo một project build với các step:
 - Checkout code từ một sample Github repository (code Java).
 - Tiến hành build sử dụng câu lệnh Maven.
- Build thử và kiểm tra kết quả.
- Kiểm tra artifact được tạo ra.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 3 - Build Java Project (p2)

Yêu cầu:

- Add thêm bước upload Artifact (gói war/jar) lên một S3 bucket.
 - Tiến hành build lại và kiểm tra gói war được upload lên S3.
- *Lưu ý: Jenkins server phải có IAM role cần thiết để upload lên s3.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 3 - Build Java Project (Declarative version)

Yêu cầu:

- Sử dụng lại repository của Buildjob3.
- Tạo một job mới với cú pháp Pipeline.
- Cấu hình 3 bước: checkout code, build, upload artifact to S3.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 4 - Build Nodejs Project (p1)

Yêu cầu:

- Cài đặt và cấu hình Nodejs cho server Jenkins.
- Tạo một project build với các step:
 - Checkout code từ một sample Github repository (code Vuejs+Webpack).
 - Tiến hành build sử dụng câu lệnh npm install + npm run build
- Build thử và kiểm tra kết quả.
- Kiểm tra artifact được tạo ra.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 4 - Build Nodejs Project (p2)

Yêu cầu:

- Add thêm bước upload Artifact (thư mục dist) lên một S3 bucket.
- Tiến hành build lại và kiểm tra **html, css, js** được upload lên S3.

*Lưu ý: Jenkins server phải có IAM role cần thiết để upload lên s3.

[Optional] Enable static web hosting cho S3 để xem code build ra có hoạt động không?

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 5 - Working with Docker in Jenkins.

Tiến hành cài đặt Docker & Docker compose cho Jenkins.

- <https://docs.docker.com/engine/install/ubuntu/>
- <https://docs.docker.com/compose/install/linux/>

- add step to add group of docker
- ```
>sudo usermod -a -G docker jenkins
```
- ```
>sudo systemctl restart jenkins
```

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 5 - Working with Docker in Jenkins.

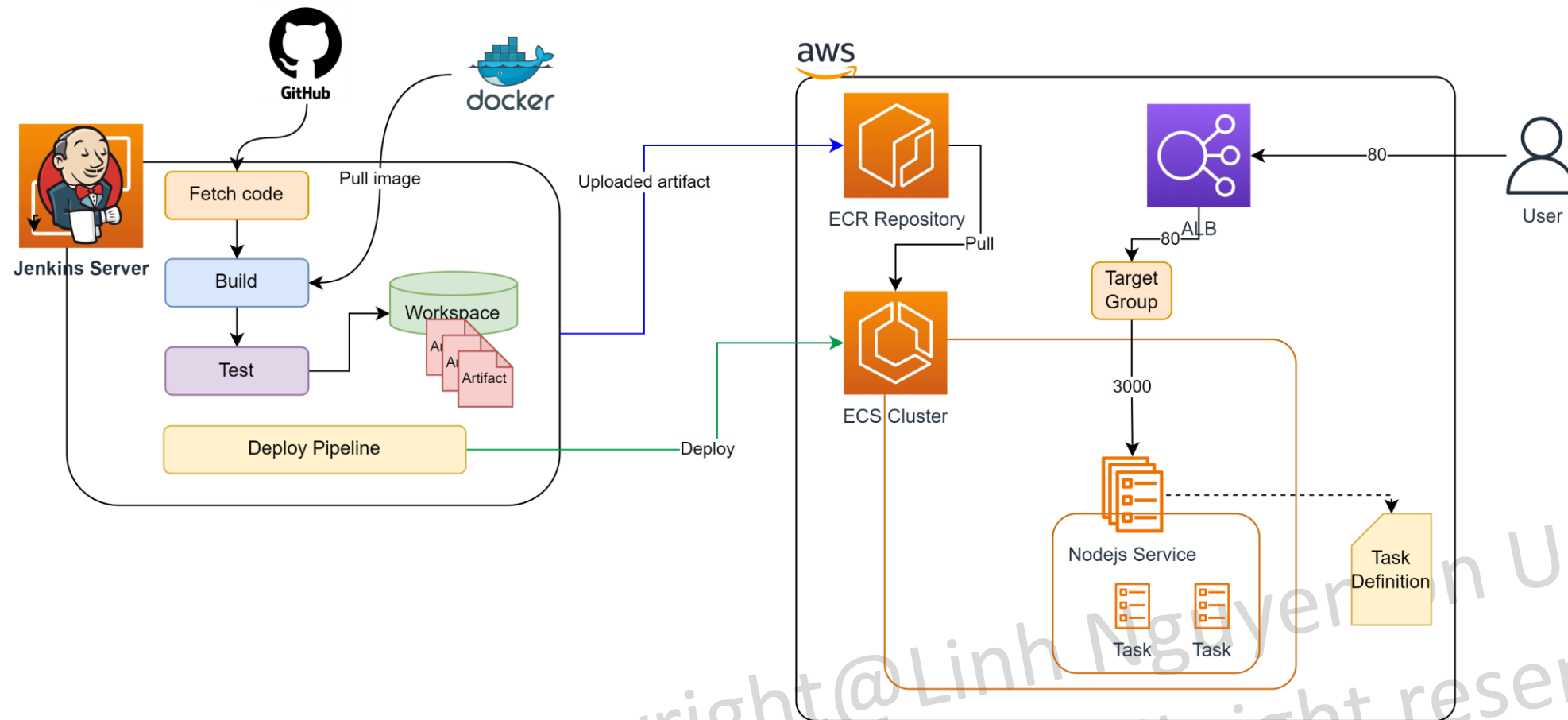
Yêu cầu:

- Tạo một ECR repository. Cấp quyền ECR cho Jenkins server.
- Tạo một job build có nội dung checkout code từ một Github repository.
- Build source code để tạo ra artifact.
- Sử dụng image nginx để build một custom image.
- Push image đã build lên ECR.
- Kiểm tra kết quả sau khi build.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 6 - Tạo một job deploy cho ECS service.

Sơ đồ hệ thống cho bài lab:



Copyright @ Linh Nguyen Udemey
All right reserved

Lab 6 - Tạo một job deploy cho ECS service.

Yêu cầu:

**Tái sử dụng resource của bài lab 5*

- Sử dụng terraform template đã cung cấp sẵn, tạo ra VPC, ECS, ALB... sử dụng ECR image từ bài lab5 để tạo ra một service **nodejs**.
- Test truy cập thông qua ALB.
- Tạo một job jenkins có nhiệm vụ deploy ECR image mới lên ECS Service (update task definition xong request force-deployment cho service). Tham số nhận vào là tag-name của ECR repository (vd: ver-1, ver-2)
- Update code trên Github repository.
- Chạy job build (của lab5), chạy job deploy, kiểm tra kết quả trên ALB.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab7-Cấu hình ssh key cho private repository.

Đối với các **private repository**, Jenkins server cần được add **ssh key** vào trong Github để có thể checkout code.

Steps:

1. Tạo ssh key bằng lệnh: ssh-keygen
2. Add public key vào Github account.
3. Chỉnh setting của Github repository về private.
4. Managed Jenkins -> Security -> Git host key... -> Accept first connection.
5. Tạo một Build job mới, trong "Pipeline", chọn "Pipeline script from SCM"
6. Add credential "SSH username with private key" (sử dụng private key từ step 1).
7. Tạo một Jenkins file và push lên repo.
8. Chạy job build và kiểm tra kết quả.

Copyright@Linh Nguyen on Udemy
All right reserved

Lab8-Trigger Build job tự động từ Github.

Yêu cầu:

1. Add file **Jenkinsfile-auto** vào repo, add phần build + deploy cho ECS (Job5+6).
2. Chỉnh sửa phần build tag thêm 1 tag "latest" cho ECR image.
3. Chỉnh sửa phần deploy deploy với version "latest".
4. Push **Jenkinsfile-auto** mới lên repo.
5. Tạo một **Buildjob8-CICD-nodejs-auto** với Pipeline from SCM. *Lưu ý set branch name về ***/develop**
6. Thiết lập trigger từ Github repository để tự động trigger một job build (CICD) mỗi khi có code merge vào nhánh **develop**.
7. Chỉnh sửa code ->push code lên nhánh **develop**.
8. Kiểm tra code được deploy lên ECS, truy cập ALB.

Copyright @ Linh Nguyen on Udemy
All right reserved

Lab9 - Job build & release theo tag trên nhánh master.

Thiết kế và setting một Job Jenkins cho production với yêu cầu sau:

- Job release cần được trigger thủ công (không phải auto từ github).
- Code checkout theo một tag chỉ định trên nhánh **master** thay vì lấy mặc định.
- Có thể chọn **BuildAndDeploy**, **OnlyBuild**, **OnlyDeploy** cho job.
- Trong step deploy, tạm thời sd chiến lược force-deployment (all-in-one) cho ECS.

**Mục đích chủ yếu làm quen với pipeline nên các bạn có thể sử dụng lại cluster ECS của bài lab trước.*

Copyright@Linh Nguyen on Udemy
All right reserved

Clear resource

- SD Terraform command của lab6 để destroy toàn bộ resource.
- Xóa bớt image trên ECR.
- Xóa Task Definition (nếu còn sót lại).
- Kiểm tra NAT, EIP.
- Stop Server Jenkins nếu không có nhu cầu sử dụng.

Copyright@Linh Nguyen on Udemy
All right reserved

Giới thiệu về Scripted Pipeline.

Jenkins Scripted Pipeline là một dạng cũ của Jenkins Pipeline và được viết bằng Groovy. Scripted Pipeline bắt đầu từ một nút **node** và thực thi từ đầu đến cuối trong một khối **node**.

```
node {  
    stage('Build') {  
        // Your build steps here  
    }  
  
    stage('Test') {  
        // Your test steps here  
    }  
  
    stage('Deploy') {  
        // Your deploy steps here  
    }  
}
```

Copyright@Linh Nguyen on
All right reserved

Compare Scripted Pipeline vs Declarative Pipeline

| Tiêu chí | Scripted Pipeline | Declarative Pipeline |
|-------------|---|--|
| Cú pháp | Linh hoạt hơn dựa trên ngôn ngữ Groovy | Cũ pháp cố định và phải tuân theo quy tắc: stage, steps, agent, post,... |
| Độ phức tạp | Cho phép custom nhiều tính toán logic phức tạp. | Hạn chế hơn. Tuy nhiên có ưu điểm là cú pháp dễ đọc. |
| Quản lý lỗi | Không cung cấp cơ chế quản lý lỗi. | Cung cấp khối "post" để handle lỗi và trạng thái cuối cùng. |
| Hỗ trợ | Version cũ của Jenkins và đang dần lỗi thời. | Dạng mới đang được khuyến khích sử dụng. Nhiều tính năng hơn. |

Copyright@Linh Nguyen
All right reserved

Lab 10 - Simple Scripted pipeline.

Trong bài lab này chúng ta sẽ thử thực hiện lại Job build số 5 (Docker) bằng một Scripted Pipeline.

*Chi tiết mời các bạn xem file: **lab10.groovy**

Copyright@Linh Nguyen on Udemy
All right reserved

Lab 11 - Job notification sử dụng Slack

Tính năng Slack notification cho phép notify kết quả chạy job sang một Slack channel.

Yêu cầu:

1. Tạo một slack channel, add Jenkins CI app vào channel.
2. Cài đặt plugin cần thiết để Jenkins có thể notify sang Slack.
3. Tạo một job có bước "post" sẽ notify kết quả sang Slack.
4. Chạy job trong trường hợp success.
5. Chạy job trong trường hợp fail.

*Chi tiết step-by-step mời các bạn xem file **lab11.sh**.

Copyright@Linh Nguyen on Udemy
All right reserved

Tổng kết

- Hiểu được cách thức hoạt động của Jenkins.
- Cách cài đặt Jenkins trên một EC2 & Các plugin cơ bản.
- Biết cách tạo các job CI/CD cho các loại project sử dụng Nodejs, Java, Docker.
- Vận dụng CI/CD trên môi trường AWS để deploy một số project cơ bản sử dụng S3, ALB, ECS, ECR.
- Kết hợp Jenkins với Github để release tự động.
- Kết hợp Jenkins với Slack để notify kết quả job.
- Kết hợp với chiến lược branch/tag để tạo một CI/CD flow hoàn chỉnh.

***Nhớ clear resource/stop server.**

Copyright @ Linh Nguyen on Udemy
All right reserved

Tổng kết & Clear resources

- Stop Jenkins EC2 instance.
- Terminate all resources using Terraform (if have).

Copyright@Linh Nguyen on Udemy
All right reserved

Thanks you and see you in the next chapter!

Copyright@Linh Nguyen on Udemy
All right reserved