

Môn học

DevOps

Giảng viên

Tan Do

0868880797



Nội dung buổi 10

- Cài đặt SSH server
- SSH vào remote server với plugin
 - SSH Agent
 - Publish Over SSH
- Cấu hình master-slave cho Jenkins
- Thực hành CI/CD cho dự án từng bước
 - Chuẩn bị source code
 - Thiết lập file Jenkinsfile
 - Cài đặt server Jenkins
 - Cài đặt plugin cần thiết
 - Cấu hình công cụ
 - Cấu hình hệ thống
 - Tạo mới Jenkins project pipeline for build
 - Deploy lên server

Cài đặt SSH server

- Dockerfile-ubuntu

```
FROM ubuntu
```

```
RUN apt update && apt install openssh-server sudo -y
```

```
RUN useradd -rm -d /home/ubuntu -s /bin/bash -g root -G sudo -u  
1000 admin
```

```
RUN echo 'admin:admin' | chpasswd
```

```
RUN service ssh start
```

```
EXPOSE 22
```

```
CMD ["/usr/sbin/sshd", "-D"]
```

Cài đặt SSH server

- Dockerfile-centos7

```
FROM centos:7
```

```
RUN yum install openssh-server -y
```

```
RUN echo 'root' | passwd --stdin root
```

```
RUN sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin  
yes/' /etc/ssh/sshd_config
```

```
RUN ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key -N ''
```

```
EXPOSE 22
```

```
CMD ["/usr/sbin/sshd", "-D"]
```

Cài đặt SSH server

- Docker compose

services:

ubuntu:

container_name: ssh-ubuntu

build:

context: .

dockerfile: ./Dockerfile-ubuntu

command: sleep infinity

ports:

- 2222:22

centos7:

container_name: ssh-centos7

build:

context: .

dockerfile: ./Dockerfile-centos7

command: sleep infinity

ports:

- 2227:22

Cài đặt SSH server

- ssh admin@localhost -p 2222
- ssh root@localhost -p 2227
- docker exec -it ssh-ubuntu bash
 - ssh -V
 - apt update install -y
 - apt install iputils-ping -y
 - ping ssh-centos7
 - apt install vim -y
 - ssh root@ssh-centos7 -p 22
- docker exec -it ssh-centos7 bash
 - yum install update -y
 - yum install openssh-server openssh-clients -y
 - ssh -V
 - ssh admin@ssh-ubuntu -p 22

Cài đặt SSH server bằng bash Docker

- Tạo Docker
 - `docker run --rm -it -p 2222:22 --name sshd ubuntu bash` # sleep infinity
- Set up trong server
 - `apt update -y`
 - `apt -y install openssh-server openssh-clients vim ssh ufw` # ufw: ubuntu firewall # openssh-clients: optional
 - `vi /etc/ssh/sshd_config` # bỏ comment Port 22
 - `ufw allow 22`
 - `systemctl enable ssh`
 - **Bật ssh server lên**
 - Cách 1: `systemctl start ssh` # nếu không work thì chạy lệnh dưới
 - Cách 2: `/etc/init.d/ssh start`
 - **Tạo keygen**
 - `ssh-keygen -t rsa`
 - `cd /root/.ssh`
 - `ls`
 - `Pwd`
 - **Tạo authorized_keys**
 - Cách 1: `cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys`
 - Cách 2:
 - `touch authorized_keys`
 - `cat id_rsa.pub`
 - `vi authorized_keys` # update key
 - **Check user tạo keygen: whoami**
 - **Lưu thông tin private key trên localhost**
 - **Có thể dung PuTTYgen để chuyển thành file .ppk nếu muốn**
 - **Login: `ssh -i private_key root@localhost -p 2222`**

Lỗi Failed to download metadata for repo 'AppStream' trên CentOS 8

- Tạo app_server centos

- docker run --name server -d centos sleep infinity
 - cd /etc/yum.repos.d/
 - sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*
 - sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-*
 - yum update -y

- Tham khảo

- <https://kb.confusedit.org/books/centos/page/error-failed-to-download-metadata-for-repo-appstream>

SSH vào remote server với plugin SSH Agent

- Tham khảo
 - <https://xuanthulab.net/tao-ssh-key-va-xac-thuc-ket-noi-ssh-bang-public-private-key.html>
- Cài đặt plugin SSH Agent
- Tạo public/private key ở remote server
 - docker exec -it ubuntu_server bash
 - apt update -y
 - apt install openssh-server -y
 - ssh -V
 - apt install vim -y
 - whoami
 - #
 - ssh-keygen -t rsa
 - cd /root/.ssh
 - ls
 - pwd
 - touch authorized_keys
 - cat id_rsa.pub
 - vi authorized_keys # update key

SSH vào remote server với plugin SSH Agent

- Cài đặt plugin SSH Agent
 - Restart jenkins server
- Generate pipeline script
 - Sample step: sshagent: SSH Agent
 - Add Jenkins Credentials
 - Kind: SSH username with private key
 - Username: root (user khi tạo key - whoami)
 - Private key: Click Enter directly > Add > Nhập value (cat /root/.ssh/id_rsa on remote server)

SSH vào remote server với plugin SSH Agent

- Jenkinsfile

```
pipeline {
  agent any
  stages{
    // stage('Clone') {
    //   steps {
    //     git branch: 'main', credentialsId: 'github', url: 'https://github.com/letdoitnow/mediplus-
    lite.git'
    //   }
    // }
    stage('SSH server') {
      steps {
        sshagent(['ssh']) {
          // some block
          sh 'ssh -o StrictHostKeyChecking=no -l root ssh-ubuntu touch test.txt' // 172.17.0.2
        }
      }
    }
  }
}
```

SSH vào remote server với plugin Publish Over SSH

- Cấu hình

- Manage Jenkins

- System

- Publish over SSH: Cập nhật private key đến ô thông tin Key
- Nhập thông tin SSH Server

- Generate pipeline script

- Sample: sshPublisher: Send build artifacts over SSH

- Cập nhật:

- Source files: Vd package.json
- Exec command: Vd cp package.json package-dev.json

SSH vào remote server với plugin SSH Agent

- Jenkinsfile

```
pipeline {
    agent any
    stages{
        stage('SSH server') {
            // steps {
            //     sshagent(['ssh']) {
            //         // some block
            //         sh 'ssh -o StrictHostKeyChecking=no -l root ssh-ubuntu touch test.txt' // 172.17.0.2
            //     }
            // }
            steps {
                sshPublisher(publishers: [sshPublisherDesc(configName: 'ssh-ubuntu', transfers:
                [sshTransfer(cleanRemote: false, excludes: '', execCommand: 'cp package.json package-dev.json', execTimeout:
                120000, flatten: false, makeEmptyDirs: false, noDefaultExcludes: false, patternSeparator: '[,]+' ,
                remoteDirectory: '', remoteDirectorySDF: false, removePrefix: '', sourceFiles: 'package.json')],
                usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false)])
            }
        }
    }
}
```


Cấu hình master-slave cho Jenkins

- Cấu hình

- Manage Jenkins > Nodes > New Node
 - Nhập tên node
 - Check Type Permanent Agent
 - Nhập Remote root directory: Vd: /home/dev/Jenkins
 - Launch method: Launch agent via SSH
 - Host Key Verification Strategy: Manually trusted key Verification Strategy

Thực hành CI/CD cho dự án Nodejs

- Chuẩn bị source code
 - <https://github.com/vercel/>
 - Clone 1 project
- Upload lên repo GitHub

```
> .github
> public
> src
.env
.env.production
.env.staging
.eslintrc.json
.gitignore
Jenkinsfile
README.md
next.config.js
package.json
postcss.config.js
project.production.json
tailwind.config.js
tsconfig.json
```

```
1 pipeline {
2   agent any
3
4   tools {
5     nodejs "nodejs"
6   }
7
8   stages {
9     stage("install") {
10      steps {
11        sh 'npm install'
12      }
13    }
14    stage("build") {
15      steps {
16        sh 'npm run build'
17      }
18    }
19  }
20
21  post {
22    success {
23      echo "SUCCESSFUL"
24    }
25    failure {
26      echo "FAILED"
27    }
28  }
29 }
```

Thực hành CI/CD cho dự án Nodejs

```
- Thiết lập file Jenkinsfile
pipeline {
  agent any

  tools {
    nodejs "nodejs"
  }

  stages {
    stage("install") {
      steps {
        sh 'npm install'
      }
    }
    stage("build") {
      steps {
        sh 'npm run build'
      }
    }
  }

  post {
    success {
      echo "SUCCESSFUL"
    }
    failure {
      echo "FAILED"
    }
  }
}
```

Thực hành CI/CD cho dự án Nodejs

- Cài đặt Jenkins server
- Cài đặt ngrok

Thực hành CI/CD cho dự án Nodejs

- Cài đặt plugin cần thiết
 - Nodejs
 - GitHub Pull Request Builder
- Cấu hình công cụ
 - Manage Jenkins > Tools > NodeJS
 - Nhập tên và check Install automatically
- Cấu hình hệ thống
 - GitHub Pull Request Builder
 - Check use comments to report results when updating commit status fails
 - Check use comments to report intermediate phases, triggered et al

Thực hành CI/CD cho dự án Nodejs

- Tạo mới Jenkins project pipeline for build
 - Nhập GitHub project
 - Build Trigger
 - Check GitHub pull request builder
 - Check use github hooks for build triggering. Click (?) để xem hướng dẫn
 - Truy cập vào github > settings > webhooks >
 - Tạo webhook1
 - Payload URL: Nhập địa chỉ Jenkins ngrok, thêm đuôi **/ghprbhook/**
 - Check Let me select individual events
 - Check Pull requests & Push
 - Tạo webhook2
 - Payload URL: Nhập địa chỉ Jenkins ngrok, thêm đuôi **/github-webhook/**
 - Trở lại Jenkins, mở Advance
 - Check Build every pull request automatically without asking
 - Click Trigger Setup > Add > Update commit status during build
 - Nhập commit status context. Vd Jenkins CI/CD Builder
 - Commit Status Build Result > Add > Nhập message theo status
 - Pipeline
 - Chọn pipeline script from SCM > Git > Nhập repo và credentials
 - Thông tin Branch Specifier có thể update từ */master về master hoặc \${sha1}
 - Trong Advance nhập Refspec: +refs/pull/*:refs/remote/origin/pr/*
 - Uncheck lightweight checkout

Thực hành CI/CD cho dự án Nodejs

- Build test
 - Build now
 - Check Console Output
 - Kiểm tra và xác nhận kết quả

THANK YOU