



Thanh Tùng Vũ @hayquen Theo dõi

★ 515 👤 15 ✎ 13

Đã đăng vào thg 6 28, 2022 10:29 SA - 5 phút đọc

👁 967 💬 0 📌 1

Jenkins Agent Overview

1. Giới thiệu về Jenkins Agent

Bản thân Jenkins Controller có thể vừa quản trị các biến môi trường, vừa chạy các jobs bằng tài nguyên mà nó sở hữu. Tuy nhiên, khi số lượng jobs (hay cụ thể hơn là project pipeline) quá lớn, Jenkins Controller không thể cứ "vertical scale" (nâng RAM, CPUs, ...) lên mãi được. Ngoài ra, việc để tất cả các jobs cùng chạy trên Controller còn dẫn đến nhiều vấn đề về bảo mật, khi mà tất cả người dùng đều có quyền truy cập vào resource, enviroment của Jenkins Controller.

Giải pháp cho vấn đề nêu trên là Jenkins Agent.

Jenkins Agent là một machine (VM, docker container, máy chủ vật lý, AWS EC2 instance, ...) kết nối và nhận jobs từ Controller. Nhiều agent kết nối với controller sẽ tạo ra kiến trúc phân tán cho Jenkins, ngoài việc giảm workloads cho Controller, tăng cường bảo mật, còn nâng cao khả năng sẵn có cho hệ thống.

2. Kết nối Agent với Controller

Hiện tại có 2 cách kết nối Agent với Controller phổ biến: SSH connector và Inbound connector.

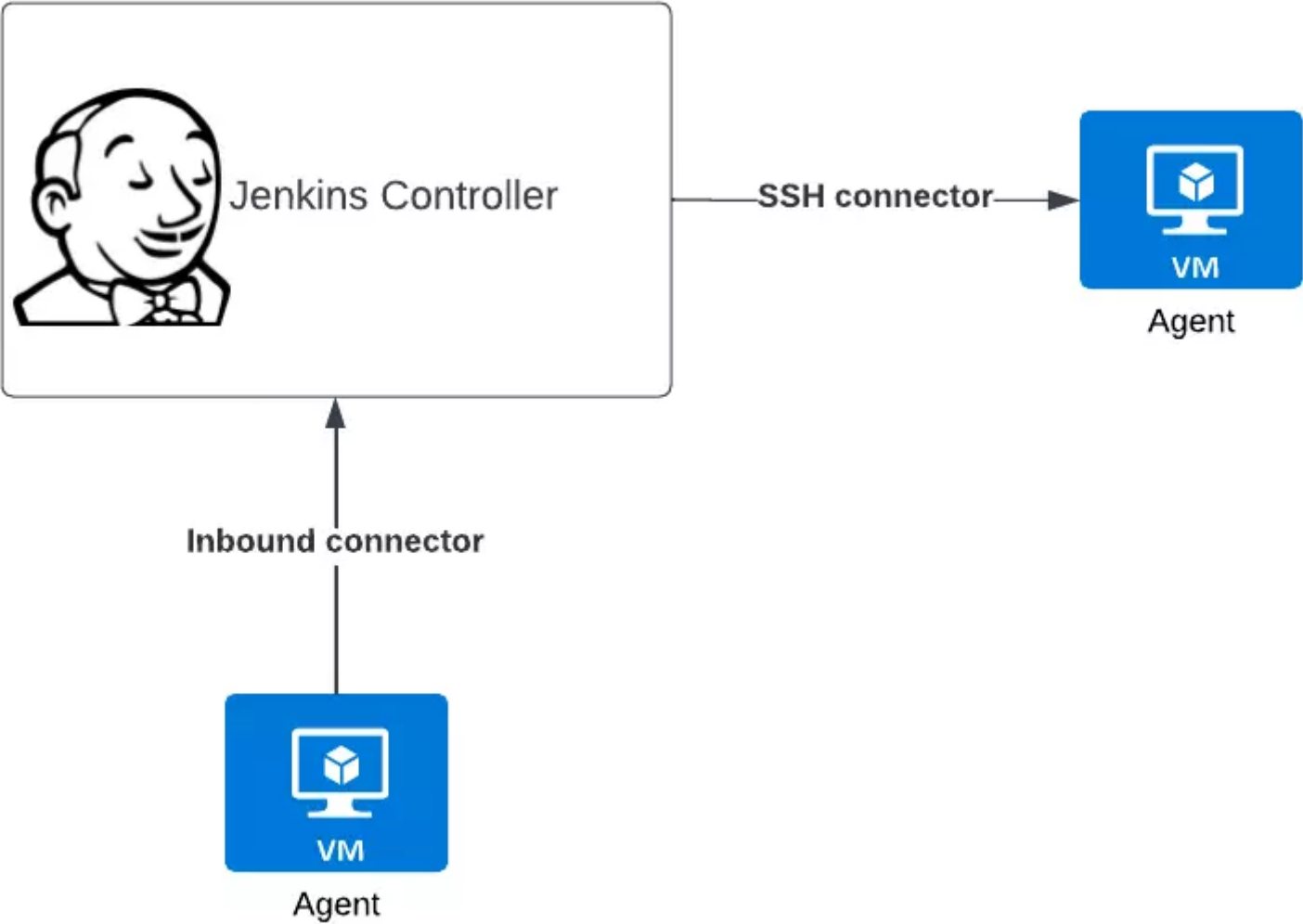


Hãy đăng ký một tài khoản Viblo để nhận được nhiều bài viết thú vị hơn.



↑ 0 ↓





2.1 SSH connector

Với cách thức kết nối này, Jenkins Controller sẽ có thông tin remote SSH, SSH key để chủ động call tới agent.

Demo cấu hình SSH connector cho agent (Docker container trên Linux)

Yêu cầu:

- Machine 1: đã cài đặt Jenkins (đóng vai trò là Jenkins Controller), cung cấp giao diện quản trị Jenkins.
- Machine 2: hệ điều hành Linux, đã cài đặt Docker (để chạy container Jenkins Agent)

#1 Trên machine 2, tạo SSH key với lệnh

```
ssh-keygen -f ~/.ssh/jenkins_agent_key
```

#2 Start agent trên machine 2:

Start agent container

```
docker run -d --rm --name=agent1 -p 22:22 \
-e "JENKINS_AGENT_SSH_PUBKEY=[your-public-key]" \
jenkins/ssh-agent:alpine
```

Trong đó, [your-public-key] là nội dung file ~/.ssh/jenkins_agent_key.pub (đã tạo ở B1)

Cập nhật container enviroment



Hãy đăng ký một tài khoản Viblo để nhận được nhiều bài viết thú vị hơn.



0

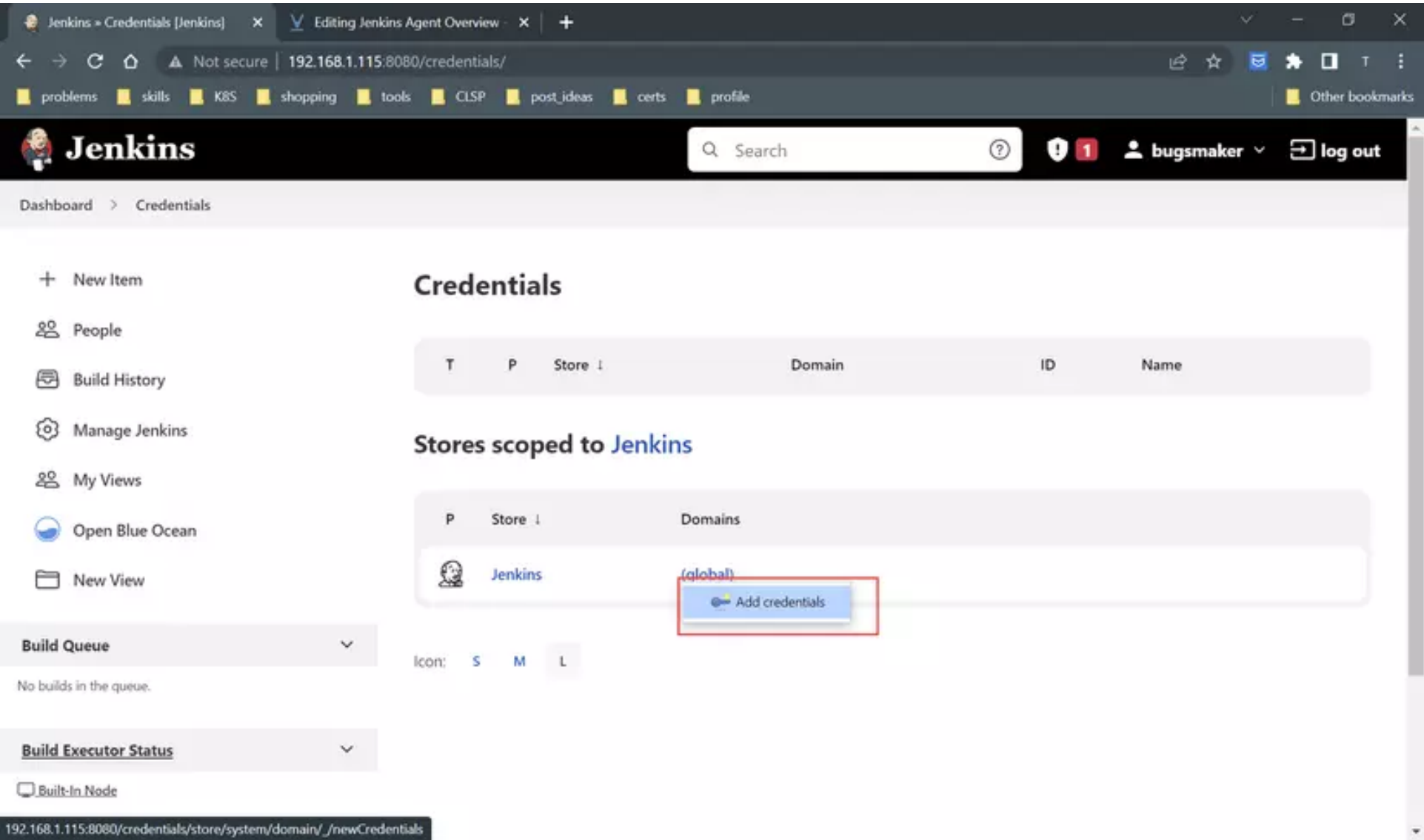


Lấy thông tin javaPath (để cấu hình cho bước sau):

```
docker exec agent1 which java
```

#3 Trên giao diện Jenkins Controller, tạo Jenkins SSH credential

Truy cập mục Manage Jenkins --> Manage Credentials --> Add Credentials (trong mục Global)



Trong mục New Credentials, điền các thông tin

- Kind: SSH Username with private key
- Scope: Global
- Username: agent1-credential (cái này bạn có thể tùy chỉnh theo ý muốn)
- Private key: tick chọn "Enter directly", click Add, điền nội dung file ~/.ssh/jenkins_agent_key (đã tạo ở B1)

Click Create để tạo Credential.

#4 Trên giao diện Jenkins Controller, khởi tạo Agent

Truy cập mục Manage Jenkins --> Manage Nodes and Clouds --> (+) New Node

Điền tên node, tick chọn Permanent Agent, click Create.

Trong mục New Agent, điền các thông tin:

- Remote directory: /home/directory



Hãy đăng ký một tài khoản Viblo để nhận được nhiều bài viết thú vị hơn.



0

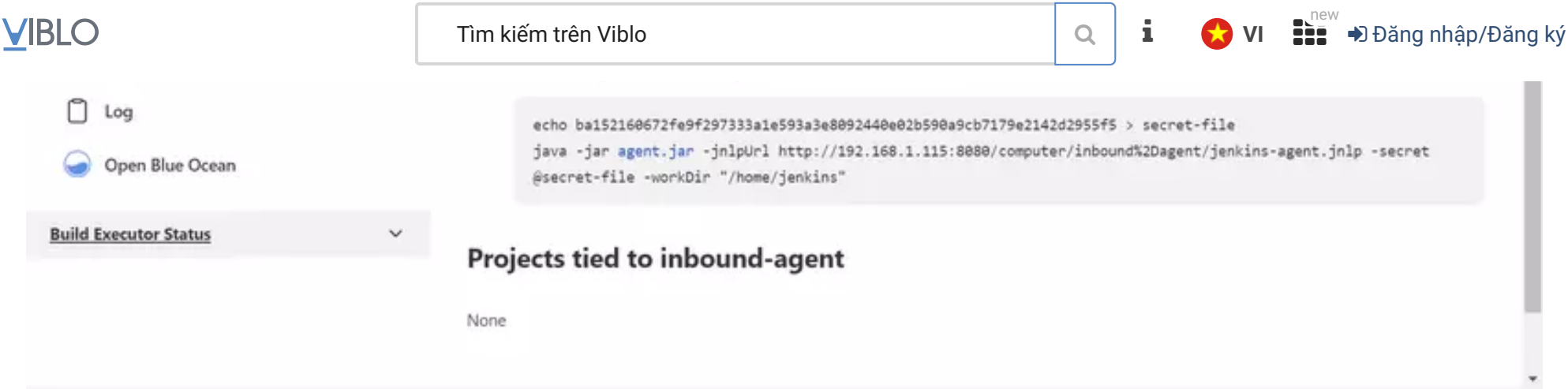
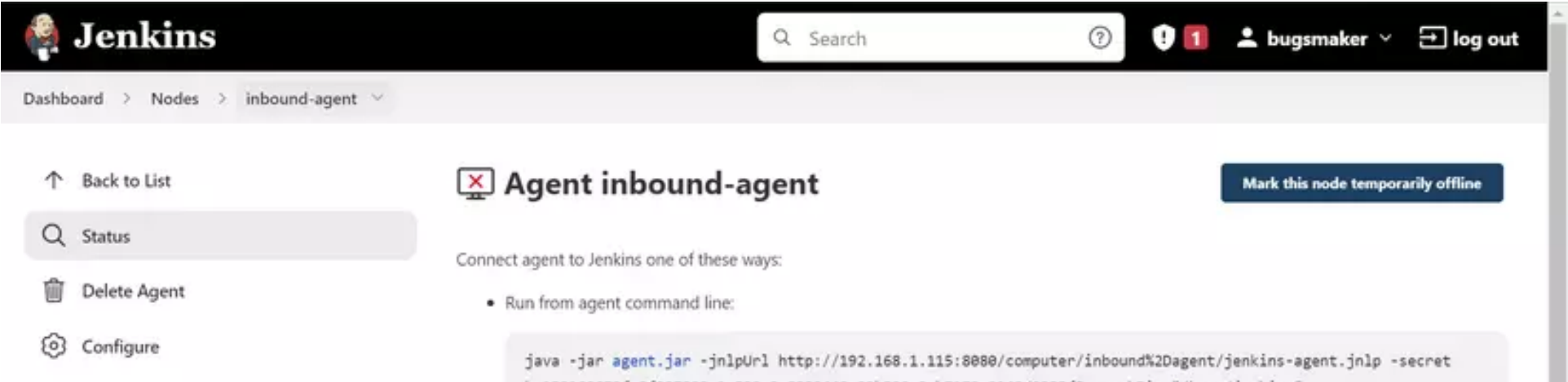


- Usage: Only build jobs with label expressions matching this node
- Launch method: Launch agent by connecting it to the controller

Click Save.

Trong mục Manage Jenkins --> Manage nodes and clouds, click chọn agent vừa tạo.

Jenkins sẽ sinh sẵn command để kết nối tới Jenkins Controller từ machine chạy agent.

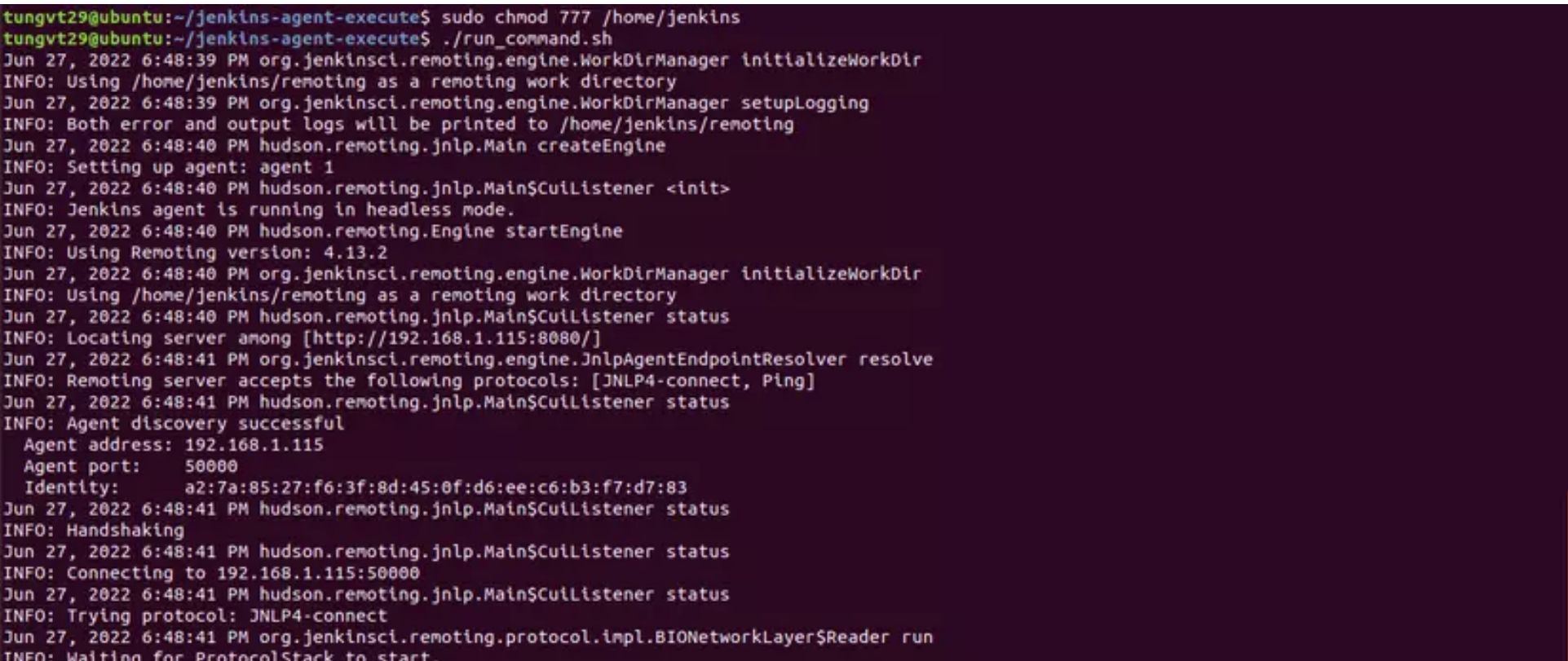


#2 Chạy lệnh trên machine 2

Download file agent.jar (tại đường dẫn http://<jenkins-host>/jnlpJars/agent.jar), copy lệnh phía trên và chạy

Lưu ý: cần đảm bảo machine 2 đã cài java.

Kết quả thu được khi agent kết nối thành công:



Viblo

Hãy đăng ký một tài khoản Viblo để nhận được nhiều bài viết thú vị hơn.

3. Link tham khảo

<https://www.jenkins.io/doc/book/using/using-agents/> <https://www.jenkins.io/doc/book/scaling/architecting-for-scale/>

- Jenkins Agent
- SSH connector
- Inbound connector

All rights reserved



Bài viết liên quan

[Làm quen với thẻ Polygon trong SVG](#)

[Tuan Vo](#)

0 phút đọc

222 0 0 2

[Vẽ hình ellipse trong SVG](#)

[Tuan Vo](#)

0 phút đọc

427 0 0 4

[Tìm hiểu từ khóa defined? trong Ruby](#)

[Nguyen Phi Viet](#)

7 phút đọc

169 0 0 1

[Hướng dẫn vẽ SVG](#)

[Tuan Vo](#)

0 phút đọc

419 0 0 0

[Machine Learning & Deep Learning Interview Questions \(P1\)](#)

[Tu Pham](#)

5 phút đọc

1.1K 8 4 16

[Part 4 - Jenkins CI : Cấu hình slave node machine và tích hợp hệ thống phân tán - phần cuối](#)

[Nguyễn Văn Mạnh](#)

13 phút đọc

6.4K 5 3 7

[Docker là gì? Khi nào nên dùng Docker?](#)

[Software Engineer Training](#)

10 phút đọc

2.7K 3 3 8

[10 kỹ năng mềm cần thiết cho lập trình viên thành công](#)

[Software Engineer Training](#)

15 phút đọc

889 3 1 6

Bài viết khác từ Thanh Tùng Vũ

[Dựng Oracle Database với Docker container](#)

[Thanh Tùng Vũ](#)

2 phút đọc

4.9K 3 1 4

[Oracle Redo log](#)

[Thanh Tùng Vũ](#)

2 phút đọc

960 1 0 1

[ReentrantLock trong java!](#)

[Thanh Tùng Vũ](#)

2 phút đọc

2.4K 0 1 4

[Lambda expression trong Java?](#)

[Thanh Tùng Vũ](#)

2 phút đọc

272 4 0 0

[Tất tật những điều bạn cần biết về Infrastructure as code !!!!](#)

[Thanh Tùng Vũ](#)

7 phút đọc

496 4 0 7

[Những điều bạn cần biết về Inode trong File System !!!](#)

[Thanh Tùng Vũ](#)

4 phút đọc

233 3 2 5



Hãy đăng ký một tài khoản Viblo để nhận được nhiều bài viết thú vị hơn.



↑ 0 ↓



Cloud Engineer với DevOps Engineer: Bạn nên chọn cái nào?

Thanh Tùng Vũ

8 phút đọc

👁 2.2K

📖 3

💬 0

📈 3

BLUE-GREEN DEPLOYMENT TRÊN KUBERNETES

Thanh Tùng Vũ

6 phút đọc

👁 834

📖 1

💬 0

📈 3

Bình luận

🗨 Đăng nhập để bình luận

TÀI NGUYÊN

- [Bài viết](#)[Tổ chức](#)
- [Câu hỏi](#)[Tags](#)
- [Videos](#)[Tác giả](#)
- [Thảo luận](#)[Đề xuất hệ thống](#)
- [Công cụ](#)[Machine Learning](#)
- [Trạng thái hệ thống](#)

DỊCH VỤ


-  [Viblo](#)
-  [Viblo Code](#)
-  [Viblo CTF](#)
-  [Viblo CV](#)
-  [Viblo Learning](#)
-  [Viblo Partner](#)
-  [Viblo Battle](#)
-  [Viblo Interview](#)

ỨNG DỤNG DI ĐỘNG




LIÊN KẾT





Hãy đăng ký một tài khoản Viblo để nhận được nhiều bài viết thú vị hơn.


Đăng ký




↑ 0 ↓

•

•



•



•

