

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THÔNG THÔNG TIN

BÀI TẬP LỚN
QUẢN LÝ DỰ ÁN WEB BOOKING HOTEL
TRÊN MÔI TRƯỜNG CLOUD

Web booking hotel project management in the cloud environment

ĐIỆN TOÁN ĐÁM MÂY

GIẢNG VIÊN HƯỚNG DẪN

ThS. HÀ LÊ HOÀI TRUNG

SINH VIÊN THỰC HIỆN:

PHẠM TRỌNG TUẤN - 21521636

NGUYỄN DƯƠNG CHÍ TÂM - 21520439

TRẦN THỊ LUYỄN - 21521107

TP. HỒ CHÍ MINH, 2024

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THÔNG THÔNG TIN

BÀI TẬP LỚN
QUẢN LÝ DỰ ÁN WEB BOOKING HOTEL
TRÊN MÔI TRƯỜNG CLOUD

Web booking hotel project management on the cloud environment

ĐIỆN TOÁN ĐÁM MÂY

GIẢNG VIÊN HƯỚNG DẪN

ThS. HÀ LÊ HOÀI TRUNG

SINH VIÊN THỰC HIỆN:

PHẠM TRỌNG TUẤN - 21521636

NGUYỄN DƯƠNG CHÍ TÂM - 21520439

TRẦN THỊ LUYỀN - 21521107

TP. HỒ CHÍ MINH, 2024

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

LỜI CẢM ƠN

Lời đầu tiên nhóm chúng em xin gửi lời cảm ơn đến thầy Hà Lê Hoài Trung – giảng viên môn Điện toán đám mây thầy đã nhiệt tình giảng dạy, tận tình hướng dẫn cho nhóm trong suốt quá trình thực hiện dự án. Nhờ đó, chúng em đã tiếp thu được nhiều kiến thức, kỹ năng ứng dụng vào dự án. Một lần nữa nhóm xin chân thành cảm ơn thầy.

Nhóm thực hiện nghiên cứu về đề tài quản lý dự án web trên môi trường trường cloud tìm hiểu về cách thức hoạt động cũng như các bước tiến hành cài đặt, lập trình, kiểm thử trên môi trường cloud. Cùng với kiến thức Thầy cung cấp tại lớp, nhóm cũng tìm hiểu thêm các công cụ và kiến thức mới và từ đó nhóm cố gắng vận dụng tối đa những kiến thức đã thu thập được để hoàn thiện đồ án một cách tốt nhất. Tuy nhiên, trong quá trình thực hiện, nhóm chúng em không tránh khỏi những thiếu sót vì hạn chế về hiểu biết cũng như kinh nghiệm. Chính vì vậy, nhóm chúng em rất mong nhận được những sự góp ý từ phía Thầy nhằm hoàn thiện những kiến thức mà nhóm chúng em đã học tập và là hành trang để nhóm chúng em thực hiện tiếp các đề tài khác trong tương lai.

Cuối cùng, chúng em xin kính chúc mọi người sức khỏe, thành công trong những dự án tương lai.

MỤC LỤC

Chương 1. TỔNG QUAN ĐỀ TÀI	1
1.1. Giới thiệu đề tài	1
1.2. Mục tiêu đề tài	2
1.3. Mô tả bài toán	2
1.3.1. Lưu trữ	2
1.3.2. Trực quan	3
1.3.3. Xử lý	3
1.4. So sánh môi trường truyền thống và hệ thống cloud	4
Chương 2. CƠ SỞ LÝ THUYẾT.....	5
2.1. Mô hình dịch vụ	5
2.1.1. Infrastructure as a service (IaaS).....	6
2.1.2. Platform as a service (PaaS)	7
2.1.3. Software as a service (SaaS)	8
2.2. Azure Virtual Machines (VMs).....	8
2.2.1. Khái niệm Azure Virtual Machines (VMs)	8
2.2.2. Các thành phần trong VMs:	9
2.2.3. Lợi ích của Azure Virtual Machine	10
2.3. Azure Container Registry.....	10
2.3.1. Khái niệm Azure Container Registry.....	10
2.3.2. Chức năng chính Azure Container Registry	11
2.4. Azure Kubernetes Service (AKS)	11
2.4.1. Khái niệm Azure Kubernetes Service (AKS)	11
2.4.2. Kiến trúc Azure Kubernetes Service	12

2.5.	Azure Monitor	13
2.6.	Jenkins	14
2.6.1.	CI là gì?	15
2.6.2.	CD là gì?	15
2.7.	Azure Key Vault.....	16
2.7.1.	Khái niệm Azure Key Vault	16
2.7.2.	Lý do nên sử dụng Azure Key Vault	17
2.8.	Grafana	18
Chương 3.	MÔ HÌNH DỮ LIỆU	19
Chương 4.	TRIỀN KHAI QUẢN LÝ DỰ ÁN WEB TRÊN MÔI TRƯỜNG CLOUD	21
4.1.	Mô hình kiến trúc	21
4.2.	Các bước tiến hành	22
Chương 5.	TỔNG KẾT.....	67
5.1.	Kết quả đạt được	67
5.2.	Ưu điểm	67
5.3.	Nhược điểm	67
5.4.	Hướng phát triển.....	68

DANH MỤC HÌNH

Hình 2.1: Các lớp riêng biệt của dịch vụ kim tự tháp Điện toán đám mây.	5
Hình 2.2: Dịch vụ cơ sở hạ tầng (IaaS). (Nguồn ảnh: INDA Academy)	6
Hình 2.3: Dịch vụ nền tảng (PaaS). (Nguồn ảnh: Medium)	7
Hình 2.4: Software as a service (SaaS). (Nguồn ảnh: wallarm)	8
Hình 2.5: Mô hình kiến trúc Azure Kubernetes Service. (Nguồn ảnh: Medium)....	12
Hình 2.6: Azure Monitor. (Nguồn ảnh: Soft365).....	13
Hình 2.7: Mô hình Jenkins. (Nguồn ảnh: hostingviet).....	15
Hình 2.8: Tiến trình CI/CD. (Nguồn ảnh: VNPro)	16
Hình 2.9: Sử dụng Sử dụng Azure Key Vault trong Pipeline. (Nguồn ảnh: chuvash)	17
Hình 4.1: Mô hình CI/CD using Jenkins on Azure Container Service (AKS).	21
Hình 4.2: Thực hiện tạo máy ảo	22
Hình 4.3: Tạo mới resource group cho máy ảo.....	22
Hình 4.4: Cấu hình máy ảo	23
Hình 4.5: Tạo tài khoản Adminstrator	23
Hình 4.6: Chọn inbound port	24
Hình 4.7: Hoàn thành tạo máy ảo.....	24
Hình 4.8: Tải private key và create resource để kết nối máy ảo	25
Hình 4.9: Cài đặt Jenkins bằng “Remote-SSH” của Visual Code	25
Hình 4.10: Nhập thông tin máy ảo vừa tạo	25
Hình 4.11: Thực hiện kết nối với máy ảo	26
Hình 4.12: Thực thi câu lệnh "sudo su-"	26
Hình 4.13: Cài đặt docker-compose.....	26
Hình 4.14:Cài đặt docker	27
Hình 4.15: Cấu hình docker container	27
Hình 4.16: Vào máy ảo và mở port 8080 cho web Jenkins	28
Hình 4.17: Sau khi truy cập được web Jenkins.....	28
Hình 4.18: Thực thi lấy mật khẩu Jenkins	29

Hình 4.19: Truy cập vào Jenkins.....	29
Hình 4.20: Thực hiện caid đặt với "Install plug in"	30
Hình 4.21: Cài đặt thông tin và mật khẩu cho Jenkins	30
Hình 4.22: URL truy cập Jenkins.....	31
Hình 4.23: Trang web của Jenkins.....	31
Hình 4.24: Tạo Container Registry trên Azure Portal	32
Hình 4.25: Cấu hình Container Registry.....	32
Hình 4.26: Hoàn thành tạo Container Registry	33
Hình 4.27: Tạo Kubernetes services	33
Hình 4.28: Chọn Resoure group giống máy ảo cho Kubernetes services.....	34
Hình 4.29: Cấu hình node pools.....	35
Hình 4.30: Hoàn thành tạo Kubernetes services	35
Hình 4.31: Màn hình jenkins.....	36
Hình 4.32: Cài đặt các plugin cần thiết	38
Hình 4.33: Reset Jenkins sau khi cài đặt plugin	38
Hình 4.34; Reset Jenkins sau khi cài đặt plugin (1).....	39
Hình 4.35: Start lại Jenkins và đăng nhập lại.....	39
Hình 4.36: Cài đặt môi trường NodeJS	40
Hình 4.37: Cài đặt Git Hooks ở Jenkins	41
Hình 4.38: Cấu hình trên github.....	41
Hình 4.39: Cấu hình cho Token trên github.....	42
Hình 4.40: Hoàn thành tạo Token	42
Hình 4.41: Tạo webhook.....	43
Hình 4.42: Điền url web của Jenkins vào webhook.....	43
Hình 4.43: Tạo Credential GitHubPassword	44
Hình 4.44: Access keys và lấy Password của Container Registry	44
Hình 4.45: Điền mật khẩu cho CREDENTIAL_ACR.....	45
Hình 4.46: Lấy thông tin config cho CREDENTIAL_AKS	45
Hình 4.47: Điền thông tin config vào CREDENTIAL_AKS	46

Hình 4.48: Thực hiện lấy các thông tin AppID , Password, Tenant	47
Hình 4.49:Cấu hình Credential AZURE_SP_APP_ID	47
Hình 4.50: Credential AZURE_SP_PASSWORD	48
Hình 4.51: Cấu hình Credential AZURE_SP_TENANT.....	48
Hình 4.52:Thông tin tất cả Credential được lưu	49
Hình 4.53: Đặt tên pipeline	49
Hình 4.54: Cấu hình Build Triggers.....	50
Hình 4.55: Tạo checkout link Github của dự án web	50
Hình 4.56: Lấy Generate Pipeline Script	51
Hình 4.57: Tạo pipeline.....	51
Hình 4.58: Kiểm tra pipeline.....	56
Hình 4.59: Kiểm tra pipeline sau khi commit code	56
Hình 4.60: Lấy địa chỉ website dự án	57
Hình 4.61: Mở port 3306.....	58
Hình 4.62: Tạo docker compose trên terminal.....	58
Hình 4.63: Viết file docker.....	59
Hình 4.64: Đóng gói và chạy file docker	59
Hình 4.65: Tạo kết nối với database dựa trên IP máy ảo	60
Hình 4.66: Kết nối với database trên MySQL Workbench.....	60
Hình 4.67: MySQL Connections.....	61
Hình 4.68: Kết nối database từ máy khác	61
Hình 4.69: Giao diện trang chủ	62
Hình 4.70: Giao diện đăng nhập	62
Hình 4.71: Giao diện sau khi đăng nhập	63
Hình 4.72: Tiến hành đặt phòng.....	63
Hình 4.73: Đơn hàng đã được lưu vào database	64
Hình 4.74: Xem lại quá trình thực thi của CPU ở Insights (Monitoring) (1)	64
Hình 4.75: Xem lại quá trình thực thi của CPU ở Insights (Monitoring) (2)	65
Hình 4.76: View Grafana	65

Hình 4.77: Xem chi tiết hoạt động của CPU tại view Grafana.....66

DANH MỤC BẢNG

Bảng 2.1: Bảng so sánh môi trường truyền thống và hệ thống cloud.....4

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Từ đầy đủ
1	CI	Continuous Integration
2	CD	Continuous Integration
3	IaaS	Infrastructure as a service
4	PaaS	Platform as a service
5	SaaS	Software as a service
6	VNets	Virtual Networks
7	AKS	Azure Kubernetes Service
8	ACR	Azure Container Registry
9	VMs	Virtual Machines
10	CPU	Central Processing Unit
11	IP	Internet Protocol
12	NICs	Network Interface Cards

Chương 1. TỔNG QUAN ĐỀ TÀI

1.1. Giới thiệu đề tài

Cloud computing đã và đang trở thành một xu hướng không thể thiếu trong lĩnh vực công nghệ thông tin. Việt Nam được đánh giá là 1 trong 10 thị trường mới nổi trên thị trường trung tâm dữ liệu, điện toán đám mây với tốc độ tăng trưởng ấn tượng, trung bình 20-30% năm trong giai đoạn 2020-2026. Ngày càng nhiều các doanh nghiệp chuyển dịch hạ tầng và ứng dụng của mình lên cloud nhằm tận dụng các lợi thế về khả năng khả năng mở rộng, tính linh hoạt và hiệu quả chi phí. Các nhà cung cấp dịch vụ cloud lớn như: Amazon Web Services, Google Cloud Platform, và Microsoft Azure liên tục cải tiến và mở rộng dịch vụ của mình, giúp doanh nghiệp dễ dàng tiếp cận và triển khai các giải pháp công nghệ tiên tiến.

Lựa chọn cloud để quản lý dự án web mang lại nhiều lợi ích vượt trội. Cloud computing cho phép các dự án dễ dàng mở rộng tài nguyên theo nhu cầu thực tế, giúp tối ưu hóa hiệu suất và chi phí. Sử dụng cloud giúp giảm đáng kể chi phí đầu tư ban đầu cho hạ tầng phần cứng. Doanh nghiệp chỉ phải trả tiền cho những tài nguyên mà họ thực sự sử dụng, giúp tối ưu hóa chi phí vận hành. Các nhà cung cấp dịch vụ cloud đảm bảo tính sẵn sàng cao với các biện pháp bảo vệ dữ liệu và phục hồi sau thảm họa, giúp duy trì hoạt động liên tục của các ứng dụng web. Các nền tảng cloud lớn đều cung cấp các giải pháp bảo mật toàn diện và tuân thủ các tiêu chuẩn quốc tế về bảo mật dữ liệu, giúp bảo vệ thông tin khách hàng và doanh nghiệp.

Trong lĩnh vực du lịch và khách sạn, hệ thống booking online giữ vai trò quan trọng trong việc thu hút khách hàng và tối ưu hóa quy trình đặt phòng. Việc quản lý và triển khai dự án web booking khách sạn trên môi trường cloud không chỉ mang lại nhiều lợi ích về mặt kỹ thuật và chi phí mà còn giúp doanh nghiệp nâng cao năng lực cạnh tranh và đáp ứng nhanh chóng các yêu cầu của khách hàng. Xu hướng này chắc chắn sẽ tiếp tục phát triển mạnh mẽ trong tương lai, tạo ra những cơ hội lớn cho các doanh nghiệp và nhà phát triển.

1.2. Mục tiêu đề tài

Thực hiện nghiên cứu và phát triển mô hình quản lý dự án web hiệu quả trên môi trường cloud. Mô hình cần phù hợp với thực tế của doanh nghiệp và đảm bảo tính khả thi khi ứng dụng. Mô hình cần đáp ứng được các yếu tố đặc thù của cloud computing như tính sẵn sàng, khả năng mở rộng, tính bảo mật và linh hoạt. Sau khi thực hiện dự án nhóm phải:

- Hiểu rõ quy trình quản lý dự án web booking khách sạn trên môi trường cloud.
- Khám phá các bước và giai đoạn quan trọng: Tìm hiểu chi tiết các giai đoạn từ lập kế hoạch, phát triển, triển khai, đến bảo trì và nâng cấp hệ thống booking khách sạn.
- Tìm hiểu và áp dụng các nền tảng cloud cho việc triển khai và duy trì ứng dụng
- Sử dụng các dịch vụ cloud: Áp dụng các dịch vụ cloud như cơ sở dữ liệu, lưu trữ, máy chủ ảo, và công cụ CI/CD để tối ưu hóa quy trình phát triển và triển khai.
- Đảm bảo hệ thống booking khách sạn hoạt động mượt mà, nhanh chóng và an toàn, từ đó nâng cao trải nghiệm người dùng.

Đề tài này sẽ cung cấp những kiến thức và giải pháp cần thiết để thực hiện các dự án thành công, tạo ra giá trị bền vững cho doanh nghiệp trong thời đại số hóa.

1.3. Mô tả bài toán

Dự án phát triển và triển khai một ứng dụng web cho dịch vụ đặt phòng khách sạn. Ứng dụng này sẽ cung cấp các chức năng như tìm kiếm khách sạn, xem chi tiết phòng, đặt phòng, thanh toán và quản lý đặt phòng. Thuộc nhóm bài toán: lưu trữ, trực quan và xử lý.

1.3.1. Lưu trữ

Azure Container Registry: Lưu trữ và quản lý các hình ảnh Docker container cho ứng dụng. Azure Container Registry được sử dụng để lưu trữ các hình ảnh Docker

của ứng dụng web, giúp quản lý phiên bản, bảo mật và tích hợp dễ dàng với các dịch vụ khác của Azure như Kubernetes.

1.3.2. Trực quan

Azure Monitor: Giám sát hiệu suất và tình trạng của ứng dụng, nhận cảnh báo về các sự cố. Azure Monitor cung cấp khả năng giám sát toàn diện cho ứng dụng và cơ sở hạ tầng và thu thập, phân tích và hành động dựa trên các dữ liệu telemetry để tối ưu hóa hiệu suất và bảo đảm tính sẵn sàng của ứng dụng.

Grafana: Trực quan hóa dữ liệu giám sát và hiệu suất từ Azure Monitor. Grafana sẽ được tích hợp với Azure Monitor để cung cấp bảng điều khiển trực quan và các biểu đồ chi tiết về hiệu suất, giúp đội ngũ phát triển và vận hành theo dõi tình trạng hệ thống một cách hiệu quả.

1.3.3. Xử lý

Azure Kubernetes Services (AKS): quản lý và triển khai các container ứng dụng ở quy mô lớn, cung cấp môi trường để triển khai, quản lý và mở rộng các ứng dụng container hóa bằng Kubernetes. AKS giúp tự động hóa việc triển khai, quản lý và duy trì ứng dụng.

Docker: Container hóa các thành phần ứng dụng để đảm bảo tính nhất quán và dễ triển khai. Docker sẽ được sử dụng để container hóa các thành phần của ứng dụng web, từ frontend, backend đến các dịch vụ cơ sở dữ liệu. Điều này giúp đảm bảo rằng các ứng dụng chạy một cách nhất quán trên mọi môi trường.

Jenkins (Tự động hóa cho CI/CD): tự động hóa quy trình xây dựng, kiểm thử và triển khai ứng dụng, được sử dụng để thiết lập các pipeline CI/CD, tự động hóa quá trình xây dựng, kiểm thử và triển khai ứng dụng lên Azure. Điều này giúp cải thiện tốc độ phát triển, giảm thiểu lỗi và đảm bảo rằng ứng dụng luôn được triển khai với phiên bản mới nhất.

1.4. So sánh môi trường truyền thống và hệ thống cloud

Tiêu chí	Môi trường truyền thống	Hệ thống cloud
Chi phí	Đầu tư cho thiết bị phần cứng (máy chủ, thiết bị mạng,...)	Không cần mua sắm phần cứng. Chỉ cần trả phí dịch vụ cho nhà cung cấp cloud.
Linh hoạt	Hạn chế do phải tuân theo năng lực hiện có của hạ tầng.	Có thể thay đổi cấu hình, tài nguyên dựa trên nhu cầu sử dụng thực tế.
Mở rộng	Khó khăn do phải mua thêm phần cứng và cấu hình hệ thống.	Dễ dàng và nhanh chóng, có thể mở rộng hoặc thu hẹp tài nguyên theo nhu cầu chỉ bằng vài thao tác trên giao diện quản lý.
Phục hồi và sao lưu	Cần thiết lập và quản lý hệ thống sao lưu riêng.	Nhà cung cấp dịch vụ cloud thường cung cấp các giải pháp sao lưu và phục hồi tự động.
Tính dễ sử dụng	Mua sắm phần cứng truyền thống.	Tự phục vụ.
Độ sẵn sàng	Sửa chữa các sự cố một cách thủ công.	Tự động khôi phục nhờ tích hợp, tương tác.

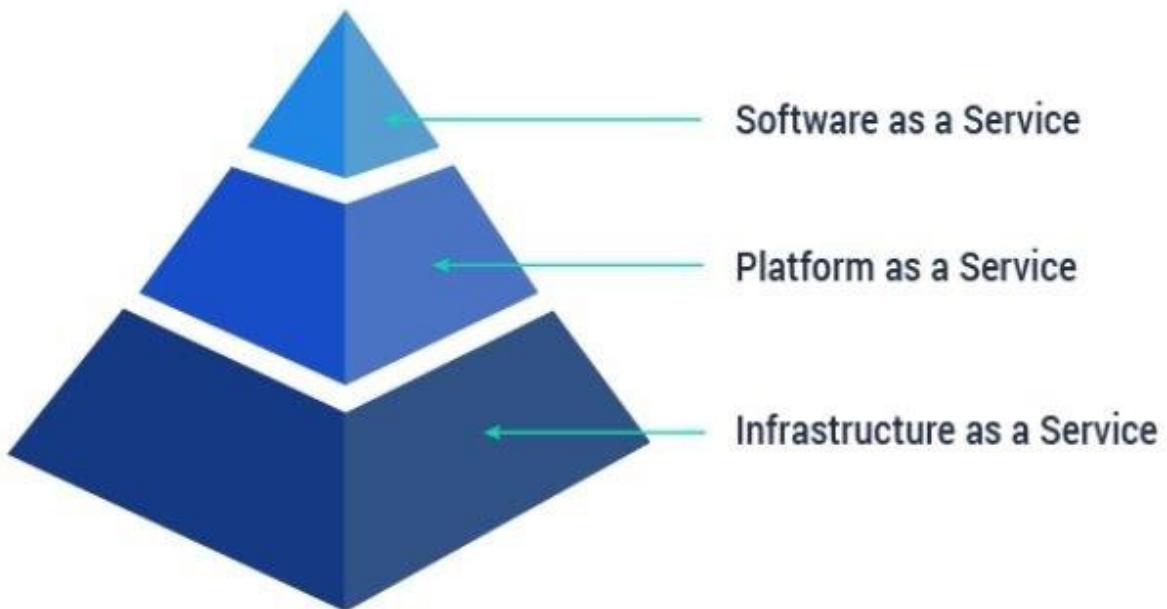
Bảng 1.1: Bảng so sánh môi trường truyền thống và hệ thống clou

Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Mô hình dịch vụ

Azure là nền tảng điện toán đám mây của Microsoft cung cấp một loạt các dịch vụ để xây dựng, triển khai và quản lý các ứng dụng thông qua mạng lưới trung tâm dữ liệu toàn cầu của Microsoft.

Azure cung cấp infrastructure as a service (IaaS), platform as a service (PaaS) và software as a service (SaaS), hỗ trợ nhiều ngôn ngữ lập trình, công cụ và framework khác nhau, bao gồm cả phần mềm dành riêng cho Microsoft và mã nguồn mở.



Hình 2.1: Các lớp riêng biệt của dịch vụ kim tự tháp Điện toán đám mây.
(Nguồn ảnh: viettelidc.com.vn)

2.1.1. Infrastructure as a service (IaaS)



Hình 2.2: Dịch vụ cơ sở hạ tầng (IaaS). (Nguồn ảnh: INDA Academy)

Dịch vụ hạ tầng (IaaS): dịch vụ cung cấp các nguồn tài nguyên bao gồm máy chủ, các thiết bị lưu trữ, hạ tầng mạng và không gian trung tâm dữ liệu trên cơ sở trả tiền trong phạm vi sử dụng (pay-per-use).

Có thể kể đến các nhà cung cấp IaaS nổi tiếng như Amazon Web Service, RackSpace, IBM Bluemix, Google Cloud Storage hay các nhà mạng cung cấp tại Việt Nam cũng bắt đầu cung cấp dịch vụ này như VDC, Viettel...

2.1.2. Platform as a service (PaaS)

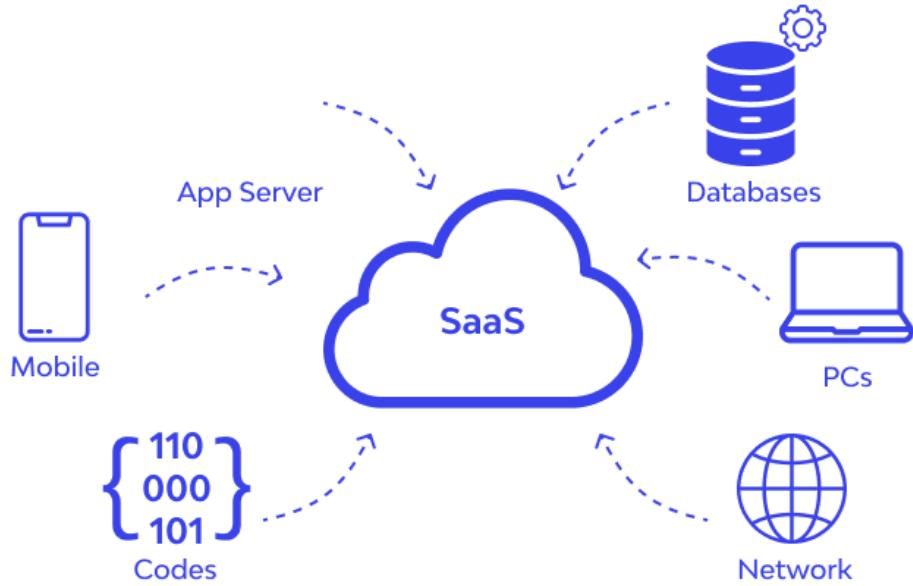


Hình 2.3: Dịch vụ nền tảng (PaaS). (Nguồn ảnh: Medium)

Dịch vụ nền tảng (PaaS): dịch vụ cung cấp nền tảng môi trường cho vòng đời hoàn và cung cấp ứng dụng web dựa trên cloud mà không phải quan tâm quá nhiều về đầu tư chi phí ban đầu, hay sự phức tạp của việc trang bị và quản lý các phần cứng và phần mềm bên dưới.

PaaS cung cấp các điều kiện cần thiết cho quá trình thiết kế, phát triển, kiểm thử, triển khai và lưu trữ ứng dụng. Các dịch vụ bao gồm công cụ cho cộng tác nhóm, quản lý và tích hợp dịch vụ Web, tích hợp cơ sở dữ liệu, bảo mật, khả năng mở rộng, quản lý trạng thái và phiên bản ứng dụng. Những dịch vụ này mang lại nhiều lợi ích cho cộng đồng phát triển và nghiên cứu ứng dụng, được cung cấp như một giải pháp tích hợp trên nền Web.

2.1.3. Software as a service (SaaS)



Hình 2.4: Software as a service (SaaS). (Nguồn ảnh: wallarm)

Dịch vụ phần mềm (SaaS): Cung cấp quyền truy cập vào phần mềm của ứng dụng, không cần phải quan tâm về việc cài đặt, thiết lập hay thử nghiệm ứng dụng. Tất cả những việc đó sẽ được dịch vụ phần mềm thực hiện và người thuê chỉ cần tìm kiếm khách hàng để trải nghiệm ứng dụng.

2.2. Azure Virtual Machines (VMs)

2.2.1. Khái niệm Azure Virtual Machines (VMs)

Azure Virtual Machines (VMs) là dịch vụ điện toán đám mây thuộc nền tảng Microsoft Azure, cung cấp khả năng tạo và quản lý các máy ảo để chạy các ứng dụng và dịch vụ khác nhau. Đây là một dịch vụ cơ sở hạ tầng dưới dạng dịch vụ (Infrastructure as a Service - IaaS), cho phép người dùng triển khai các hệ thống máy chủ ảo hóa với tính linh hoạt, khả năng mở rộng và bảo mật cao.

Ảo Hóa (Virtualization): Ảo hóa là công nghệ tạo ra các phiên bản ảo của các tài nguyên vật lý như máy chủ, lưu trữ và mạng. Mục tiêu của ảo hóa là tối ưu hóa việc sử dụng tài nguyên, tăng cường tính linh hoạt và giảm chi phí.

Máy Ảo (Virtual Machine): Là một phiên bản ảo của máy tính vật lý, có thể chạy các hệ điều hành và ứng dụng giống như một máy tính vật lý thực sự. Một máy chủ vật lý có thể chạy nhiều máy ảo đồng thời.

Hypervisor: Là phần mềm hoặc phần cứng cho phép tạo và quản lý các máy ảo. Azure sử dụng Hyper-V làm hypervisor để quản lý các VM, đảm bảo các VM chạy độc lập và hiệu quả trên cùng một máy chủ vật lý.

2.2.2. Các thành phần trong VMs:

Images: Images là các mẫu hệ điều hành và phần mềm cơ bản dùng để tạo máy ảo. Azure cung cấp một thư viện phong phú các images chuẩn, bao gồm cả Windows và Linux, hoặc người dùng có thể tạo và sử dụng các custom images riêng.

Disks: Azure VMs sử dụng ba loại ổ đĩa chính:

- OS Disk: Chứa hệ điều hành của máy ảo, thường là đĩa khởi động.
- Data Disk: Sử dụng để lưu trữ dữ liệu của ứng dụng và người dùng.
- Temporary Disk: Cung cấp không gian lưu trữ tạm thời, không đảm bảo tính bền vững dữ liệu sau khi VM khởi động lại.

Virtual Networks (VNets): VNets cho phép kết nối các VM và các tài nguyên Azure khác lại với nhau, cung cấp môi trường mạng riêng biệt và an toàn.

Network Interface Cards (NICs): NICs là các giao diện mạng cho phép các VM kết nối với VNets và các mạng khác. Một VM có thể có nhiều NIC để hỗ trợ các cấu hình mạng phức tạp.

Public IP Addresses: Địa chỉ IP công cộng được gán cho VM để truy cập từ internet. Đây là cách thức để người dùng có thể kết nối đến VM từ bên ngoài.

Security Groups: Các nhóm bảo mật giúp quản lý lưu lượng mạng đến và đi của VM thông qua các quy tắc bảo mật, đảm bảo chỉ những lưu lượng hợp lệ mới được phép.

2.2.3. Lợi ích của Azure Virtual Machine

- **Khả năng mở rộng:** Azure VMs cho phép mở rộng hoặc thu hẹp tài nguyên một cách linh hoạt theo nhu cầu thực tế, giúp tối ưu hóa hiệu suất và chi phí.
- **Tính linh hoạt:** Hỗ trợ nhiều hệ điều hành và cấu hình phần cứng khác nhau, phù hợp với mọi loại ứng dụng từ nhỏ đến lớn.
- **Thanh toán theo nhu cầu:** Người dùng chỉ phải trả phí cho tài nguyên đã sử dụng, giúp tiết kiệm chi phí và tránh lãng phí tài nguyên.
- **Bảo mật cao:** Azure cung cấp nhiều công cụ và dịch vụ bảo mật tiên tiến để bảo vệ dữ liệu và ứng dụng khỏi các mối đe dọa bảo mật.

2.3. Azure Container Registry

2.3.1. Khái niệm Azure Container Registry

Azure Container Registry (ACR) là một dịch vụ được quản lý hoàn toàn bởi Microsoft Azure, cho phép người dùng lưu trữ và quản lý các container images. ACR tích hợp chặt chẽ với các dịch vụ Azure khác, cung cấp một giải pháp toàn diện cho việc phát triển, vận hành và triển khai các ứng dụng container hóa.

- **Container:** Là một đơn vị phần mềm đóng gói mọi thứ cần thiết để chạy một ứng dụng, bao gồm mã nguồn, các thư viện và phụ thuộc. Containers đảm bảo rằng ứng dụng chạy nhất quán trên mọi môi trường.
- **Container Image:** Là một mẫu bất biến của container, chứa mọi thứ cần thiết để chạy ứng dụng. Images có thể được lưu trữ và phân phối qua các container registry.
- **Container Registry:** Là một kho lưu trữ các container images, cho phép người dùng push (đẩy) và pull (kéo) images để triển khai các container.

2.3.2. Chức năng chính Azure Container Registry

Push và Pull Images: Cho phép người dùng đẩy các container images từ máy phát triển lên registry và kéo chúng xuống các môi trường triển khai.

Webhooks: Kích hoạt các sự kiện khi có sự thay đổi trong registry, hỗ trợ tự động hóa các quy trình CI/CD.

Quản Lý Vùng (Geo-replication): Cho phép đồng bộ các images qua nhiều vùng địa lý, đảm bảo hiệu suất và sẵn sàng cao.

Quét Lỗ Hổng (Vulnerability Scanning): Kiểm tra các container images để phát hiện và cảnh báo về các lỗ hổng bảo mật tiềm ẩn.

2.4. Azure Kubernetes Service (AKS)

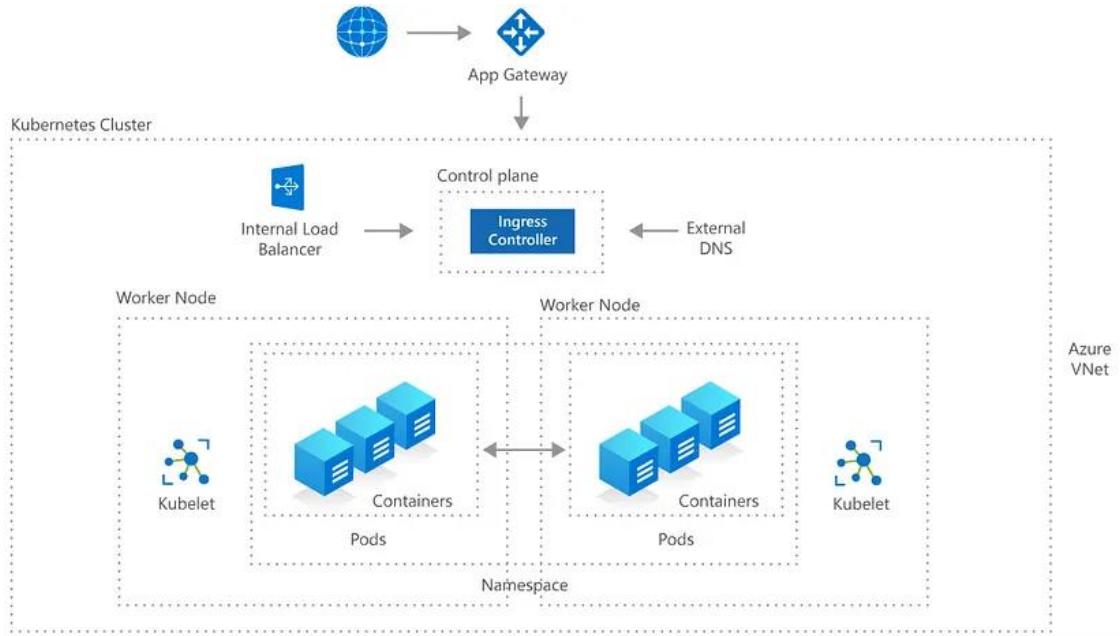
2.4.1. Khái niệm Azure Kubernetes Service (AKS)

Azure Kubernetes Service (AKS) là dịch vụ điều phối container được quản lý dựa trên hệ thống Kubernetes nguồn mở, có sẵn trên đám mây công cộng Microsoft Azure. Mặc dù Kubernetes rất hữu ích trong việc quản lý khối lượng công việc và dịch vụ được đóng gói nhưng việc cài đặt và bảo trì có thể phức tạp. AKS giúp giảm bớt sự phức tạp, cho phép các tổ chức xử lý các chức năng quan trọng, chẳng hạn như triển khai, mở rộng quy mô và quản lý các bộ chứa Docker cũng như các ứng dụng dựa trên bộ chứa. [1]

Kubernetes là nền tảng mã nguồn mở cho việc triển khai container, hoạt động dựa trên việc tạo ra một cluster – hay một cụm liên kết các máy tính với nhau (có thể là máy thật hay máy ảo) như một tài nguyên duy nhất, rồi triển khai các ứng dụng thành các deployment. Như vậy một ứng dụng được triển khai thành một deployment tương ứng, mỗi deployment có thể có một hay nhiều pod để chạy ứng dụng đó.

Kubernetes giúp lên lịch để deploy và theo dõi hoạt động của các container, khi một container nào đó gặp trục trặc, nó sẽ tự động thay thế một container mới.

2.4.2. Kiến trúc Azure Kubernetes Service

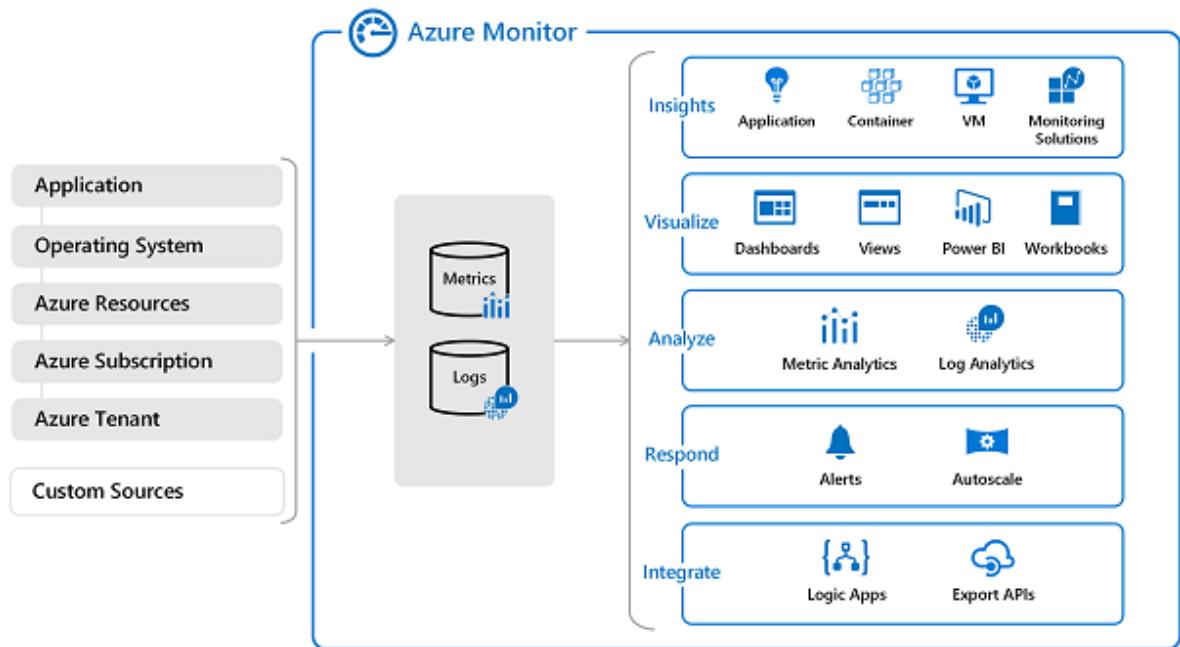


Hình 2.5: Mô hình kiến trúc Azure Kubernetes Service. (Nguồn ảnh: Medium)

Virtual nodes: cho phép mở rộng khi tải tăng đột biến (burstable scaling) trong cụm cluster.

VM scale sets: Khi kích hoạt tính năng này sẽ tạo ra cluster mà sử dụng VM scale sets thay vì chỉ trên một node VM cho các node trong cluster. Thông thường tính năng này được sử dụng cho các trường hợp autoscaling và Multiple node pools.

2.5. Azure Monitor



Hình 2.6: Azure Monitor. (Nguồn ảnh: Soft365)

Azure Monitor là một giải pháp toàn diện để thu thập, phân tích và đưa ra các hành động dựa trên dữ liệu trên môi trường đám mây và hệ thống on-premise của bạn. Qua đó, giúp bạn tối đa hóa tính khả dụng, hiệu suất của các ứng dụng và dịch vụ của bạn. Các tính năng chính của Azure Monitor gồm:

- Phát hiện và chẩn đoán sự cố trên các ứng dụng và phần phụ thuộc với Application Insights.
- Tương quan các vấn đề hạ tầng với VM insights và Container insights.
- Đào sâu vào các dữ liệu giám sát của bạn với Log Analytics để khắc phục sự cố và chẩn đoán sâu.
- Hỗ trợ các hoạt động theo quy mô với các cảnh báo thông minh và các hành động tự động.
- Tạo các bảng biểu trực quan với Azure dashboards và Azure workbooks.
- Thu thập dữ liệu từ các tài nguyên được giám sát bằng Azure Monitor Metrics.

Azure Monitor có khả năng thu thập dữ liệu từ nhiều nguồn khác nhau, bao gồm ứng dụng của bạn, hệ điều hành và dịch vụ mà ứng dụng phụ thuộc, cũng như chính nền tảng Azure. Cụ thể, Azure Monitor thu thập dữ liệu từ các cấp độ sau:

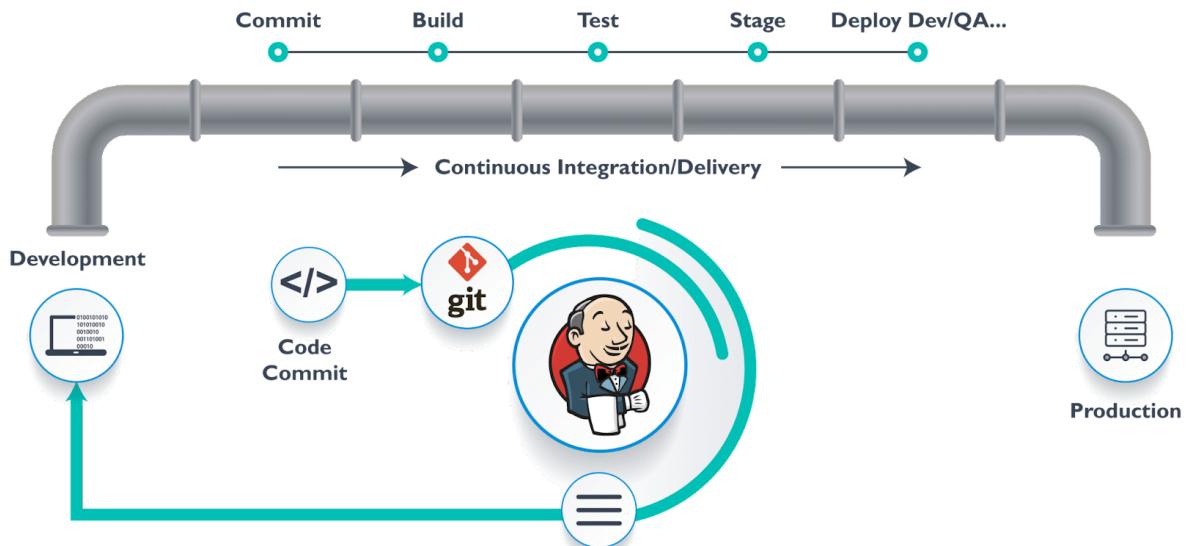
- **Giám sát dữ liệu ứng dụng:** Dữ liệu về hiệu suất và chức năng của mã bạn đã viết, bất kể nền tảng mà nó chạy trên.
- **Giám sát dữ liệu hệ điều hành máy khách:** Dữ liệu về hệ điều hành mà ứng dụng của bạn đang chạy, dù là trên Azure, các đám mây khác hay tại chỗ.
- **Giám sát dữ liệu tài nguyên Azure:** Dữ liệu về hoạt động của các tài nguyên Azure mà bạn sử dụng.
- **Giám sát dữ liệu đăng ký Azure:** Dữ liệu về hoạt động và quản lý của các đăng ký Azure, bao gồm cả sức khỏe và hoạt động của chính nền tảng Azure.
- **Giám sát dữ liệu tenant Azure:** Dữ liệu về hoạt động của các dịch vụ Azure cung cấp cho người thuê, chẳng hạn như Azure Active Directory.

Nhờ việc thu thập dữ liệu từ nhiều cấp độ này, Azure Monitor giúp cung cấp cái nhìn toàn diện về hiệu suất và sức khỏe của các ứng dụng và dịch vụ trong hệ thống.

2.6. Jenkins

Jenkins là một mã nguồn mở có khả năng thực hiện tích hợp liên tục (Continuous Integration, viết tắt là CI), và xây dựng những tác vụ tự động hóa.

Nó có thể tích hợp nhiều mã nguồn của tất cả các thành viên trong nhóm một cách nhanh chóng, liên tục, theo dõi quá trình thực thi, trạng thái triển khai bằng các bước kiểm tra (như Integration test, và units test). Điều này góp phần hoàn thiện sản phẩm, giúp chúng hoạt động ổn định.



Hình 2.7: Mô hình Jenkins. (Nguồn ảnh: hostingviet)

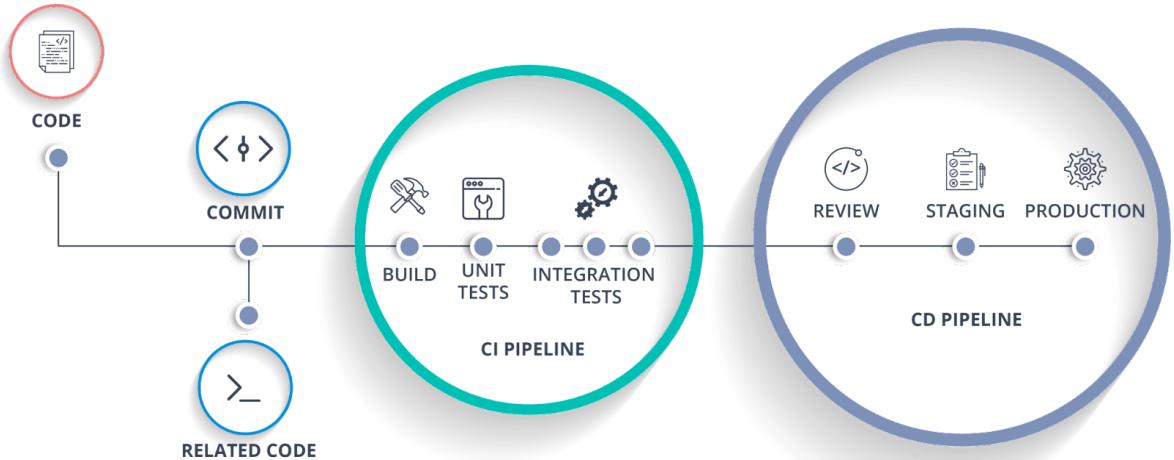
2.6.1. CI là gì?

CI hay Continuous Integration, là phương pháp phát triển phần mềm yêu cầu các thành viên của team tích hợp công việc của họ thường xuyên, mỗi ngày ít nhất một lần. Mỗi tích hợp được build tự động (bao gồm cả test) nhằm phát hiện lỗi nhanh nhất có thể. Và tất nhiên giúp cho việc tích hợp và phát triển phần mềm trở nên nhanh và hiệu quả hơn.

2.6.2. CD là gì?

CD hay Continuous Delivery là quá trình chuyển giao liên tục, là quá tập hợp nhiều kỹ thuật triển khai tích hợp mã nguồn trong môi trường staging (môi trường này tương tự như môi trường production).

Qua cách này, bạn luôn đảm bảo mã nguồn được review, tiến trình kiểm thử diễn ra tỉ mỉ trước khi đưa lên môi trường production.



Hình 2.8: Tiến trình CI/CD. (Nguồn ảnh: VNPro)

2.7. Azure Key Vault

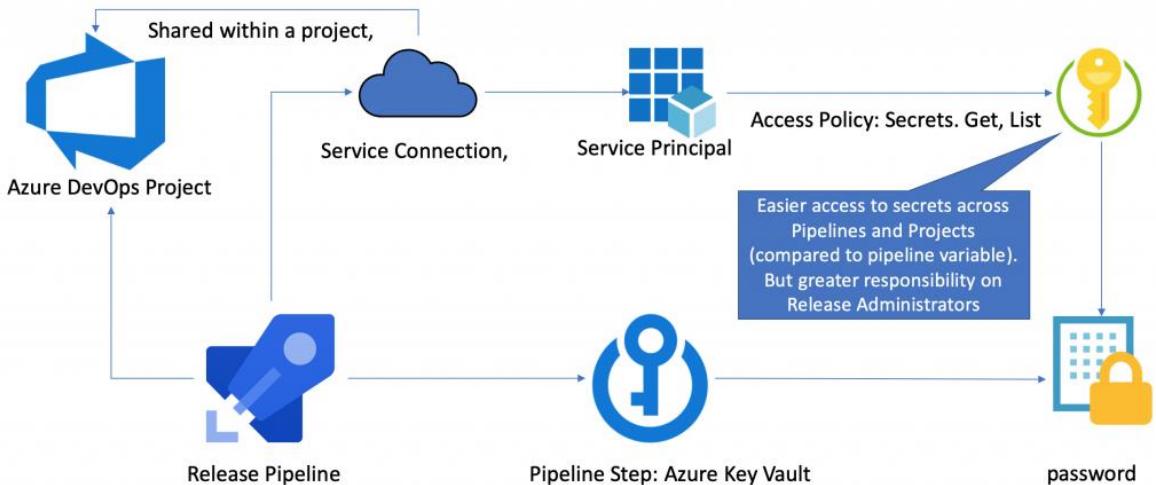
2.7.1. Khái niệm Azure Key Vault

Azure Key Vault là một giải pháp quản lý và bảo mật thông tin được cung cấp bởi Microsoft Azure. Công cụ này cho phép người dùng lưu trữ các thông tin quan trọng như mật khẩu, chứng chỉ SSL, và khóa mã hóa một cách an toàn.

Bên cạnh đó, Azure Key Vault còn hỗ trợ quản lý và phân quyền truy cập cho các ứng dụng, giúp doanh nghiệp kiểm soát tốt hơn các thông tin bảo mật. Đây là biện pháp mà nhiều doanh nghiệp áp dụng để ngăn chặn các mối đe dọa bảo mật dữ liệu như tin tặc, virus, và truy cập trái phép từ nhân viên.

Hiện nay, việc sử dụng Azure Key Vault được xem là một trong những xu hướng quan trọng trong lĩnh vực bảo mật thông tin và quản lý khóa mã hóa. Các doanh nghiệp và tổ chức ngày càng nhận thức được giá trị của việc sử dụng một nền tảng quản lý khóa mã hóa và bảo mật thông tin như Azure Key Vault.

2.7.2. Lý do nên sử dụng Azure Key Vault



Hình 2.9: Sử dụng Sử dụng Azure Key Vault trong Pipeline. (Nguồn ảnh: chuvash)

Bảo vệ thông tin quan trọng: kiểm soát ai có thể xem hoặc sửa đổi thông tin, đảm bảo chỉ những người được ủy quyền mới có thể truy cập. Bảo vệ thông tin quan trọng khỏi truy cập trái phép.

Quản lý khóa mã hóa: quản lý các khóa mã hóa riêng một cách an toàn và bảo vệ tài liệu quan trọng với các khóa của Azure.

Tính năng bảo mật khóa: lưu trữ khóa trong vùng an toàn, đặt mật khẩu và tự động quản lý chu kỳ đổi khóa. Vì thế tránh mất mật khẩu hoặc quên khóa mã hóa bằng cách tự động hóa quản lý.

Đơn giản hóa quy trình: đơn giản hóa và nhất quán, nâng cao hiệu quả vận hành và tính chuyên nghiệp. Đảm bảo chỉ người đúng quyền mới truy cập được. Phát hiện sớm các hoạt động đáng ngờ và lỗ hổng, bảo vệ doanh nghiệp khỏi các mối đe dọa bảo mật.

Tích hợp với các ứng dụng: tích hợp với các dịch vụ Azure dễ dàng triển khai và quản lý khóa mã hóa và chứng chỉ SSL. Bên cạnh đó tích hợp với các ứng dụng doanh nghiệp và ứng dụng bên thứ ba.

2.8. Grafana

Grafana là một nền tảng mã nguồn mở được thiết kế để giám sát và trực quan hóa dữ liệu. Nó cho phép người dùng tạo các bảng điều khiển (dashboards) tùy chỉnh để theo dõi và phân tích dữ liệu từ nhiều nguồn khác nhau. Grafana hỗ trợ nhiều nguồn dữ liệu, bao gồm Prometheus, Graphite, InfluxDB, Elasticsearch, và đặc biệt là Azure Monitor.

Lợi ích của việc sử dụng Grafana với Azure Monitor và Kubernetes:

- **Trực quan hóa dữ liệu:** Grafana cho phép tạo các bảng điều khiển tùy chỉnh để hiển thị dữ liệu từ Azure Monitor và Kubernetes, cung cấp cái nhìn toàn diện về hệ thống.
- **Cảnh báo và thông báo kịp thời:** Grafana hỗ trợ thiết lập cảnh báo dựa trên dữ liệu từ Azure Monitor và Kubernetes, giúp người dùng phản ứng nhanh chóng với các sự cố.
- **Quản lý dễ dàng:** tích hợp với Azure Active Directory giúp quản lý quyền truy cập và bảo mật dễ dàng hơn.
- **Phân tích chi tiết và tối ưu hóa hiệu suất:** cho phép người dùng phân tích dữ liệu chi tiết, xác định các điểm nghẽn và tối ưu hóa hiệu suất hệ thống.

Chương 3. MÔ HÌNH DỮ LIỆU

Dữ liệu trong dự án bao gồm mã nguồn, artifacts, file cấu hình và dữ liệu giám sát. Mỗi loại dữ liệu có vai trò quan trọng trong phát triển, triển khai, quản lý và giám sát ứng dụng.

Mã nguồn

- Node.js: mã nguồn viết bằng JavaScript để phát triển backend của ứng dụng, bao gồm các tệp “.js” và “.json”.
- Dockerfiles: tệp định nghĩa cách tạo Docker image, chứa các lệnh để cài đặt các dependency và thiết lập môi trường chạy ứng dụng.
- YAML: tệp cấu hình định dạng YAML, dùng để cấu hình các dịch vụ và tài nguyên trong Kubernetes, với đuôi “.yaml” hoặc “.yml”.

Artifact

- Docker image: mã nhị phân tạo từ Dockerfile, chứa toàn bộ môi trường cần thiết để chạy ứng dụng, bao gồm mã nguồn, các thư viện và cấu hình.
- File JAR: tệp Java ARchive chứa các class Java đã biên dịch, metadata và tài nguyên, định dạng “.jar”.

File cấu hình

- Manifest Kubernetes: tệp YAML định nghĩa các thành phần trong hệ thống Kubernetes như pod, service, configMap, và secret.
- Jenkinsfile: tệp cấu hình cho Jenkins pipeline, định dạng “.groovy” hoặc “Jenkinsfile”, xác định các bước trong quá trình CI/CD.

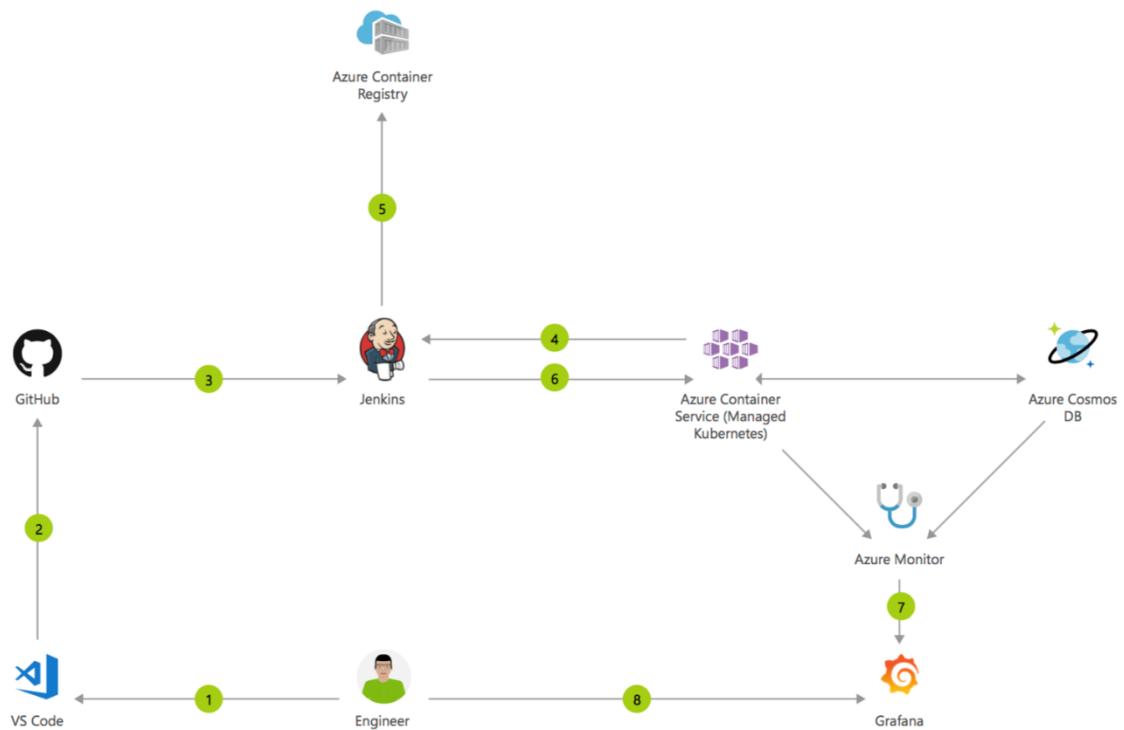
Dữ liệu giám sát

- Log: nhật ký ghi lại các sự kiện và hoạt động của hệ thống, giúp theo dõi hành vi ứng dụng và phát hiện lỗi.
- Application logs: ghi lại hoạt động và sự kiện của ứng dụng.
- System logs: ghi lại hoạt động của hệ điều hành và các dịch vụ hệ thống.

- Metrics: dữ liệu đo lường hiệu suất và trạng thái của hệ thống, cung cấp thông tin về tải hệ thống, hiệu suất ứng dụng, và sử dụng tài nguyên.
- CPU usage: mức độ sử dụng CPU.
- Memory usage: mức độ sử dụng bộ nhớ.
- Request count: số lượng yêu cầu được xử lý bởi ứng dụng.
- Alert: cảnh báo dựa trên các ngưỡng giá trị của metrics hoặc sự kiện từ logs.
- Threshold-based alerts: cảnh báo khi giá trị đo lường vượt ngưỡng định sẵn.
Ví dụ: CPU usage > 80%, ...
- Anomaly detection alerts: cảnh báo khi phát hiện hành vi bất thường trong hệ thống. Ví dụ: số lượng lỗi tăng đột biến, ...

Chương 4. TRIỂN KHAI QUẢN LÝ DỰ ÁN WEB TRÊN MÔI TRƯỜNG CLOUD

4.1. Mô hình kiến trúc



Hình 4.1: Mô hình CI/CD using Jenkins on Azure Container Service (AKS).

(Nguồn ảnh: Microsoft)

1. Thay đổi application source code.
2. Commit code lên github.
3. Kích hoạt Continuous Integration trên Jenkins.
4. Jenkins kích hoạt một công việc xây dựng bằng Azure Container Service (AKS) để sử dụng một đối tác xây dựng động.
5. Jenkins triển khai ứng dụng container mới lên Kubernetes trên Azure Container Service (AKS), được hỗ trợ bởi Azure Cosmos DB.
6. Grafana hiển thị trực quan hóa về cơ sở hạ tầng và các chỉ số ứng dụng thông qua Azure Monitor.
7. Jenkins xây dựng và đẩy Docker container lên Azure Container Registry.

8. Giám sát ứng dụng và thực hiện cài tiến.

4.2. Các bước tiến hành

Bước 1: Thực hiện tạo máy ảo bằng Azure virtual machine.

The screenshot shows the Microsoft Azure Virtual Machines dashboard. At the top, there's a search bar and various navigation links. Below it, a sidebar offers options like 'Create' (with sub-options for 'Azure virtual machine', 'Azure virtual machine with preset configuration', and 'More VMs and related solutions'), 'Reservations', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Assign tags', 'Start', 'Restart', 'Stop', 'Delete', 'Services', and 'Maintenance'. The main area displays a large icon of a computer monitor with a hexagon on it. Below the icon, the text 'No virtual machines to display' is centered. Underneath, there's a note: 'Create a virtual machine that runs Linux or Windows. Select an image from the marketplace or use your own customized image.' A prominent blue 'Create' button is at the bottom. At the very bottom right, there's a 'Give feedback' link.

Hình 4.2: Thực hiện tạo máy ảo

Bước 1.1 : Thực hiện tạo resource group cho máy ảo.

This screenshot shows the 'Create a new virtual machine' wizard in the Azure portal. The first step is 'Project details'. It includes fields for 'Subscription' (set to 'Azure for Students') and 'Resource group' (set to '(New) BTL'). There's also a 'Create new' link. The second step is 'Instance details', which includes fields for 'Virtual machine name' (empty), 'Region' (set to '(Asia Pacific) East Asia'), 'Availability options' (set to 'Availability zone'), and 'Availability zone' (set to 'Zone 1').

Hình 4.3: Tạo mới resource group cho máy ảo

Bước 1.2: Đặt tên máy chủ và tùy chỉnh cấu hình máy ảo.

Instance details

Virtual machine name *	VMBTL	✓
Region *	(Asia Pacific) East Asia	✓
Availability options	Availability zone	✓
Availability zone *	Zone 1	✓
You can now select multiple zones. Selecting multiple zones will create one VM per zone. Learn more		
Security type	Trusted launch virtual machines	✓
Configure security features		
Image *	Ubuntu Server 22.04 LTS - x64 Gen2	✓
See all images Configure VM generation		
VM architecture	<input type="radio"/> Arm64 <input checked="" type="radio"/> x64	

Hình 4.4: Cấu hình máy ảo

Bước 1.3: Thực hiện việc tạo tài khoản Administrator .

Administrator account

Authentication type	<input checked="" type="radio"/> SSH public key <input type="radio"/> Password	
Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.		
Username *	adminBTL	✓
SSH public key source	Generate new key pair	✓
Key pair name *	VMBTL_key	✓

Inbound port rules

Hình 4.5: Tạo tài khoản Adminstrator

Bước 1.4 : Thực hiện chọn thêm các cổng SSH,HTTP sau đó chọn Review + Create để tạo máy ảo .

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports *

None

Allow selected ports

Select inbound ports *

HTTP (80), HTTPS (443), SSH (22)

HTTP (80)

HTTPS (443)

SSH (22)

< Previous

Next : Disks >

Review + create

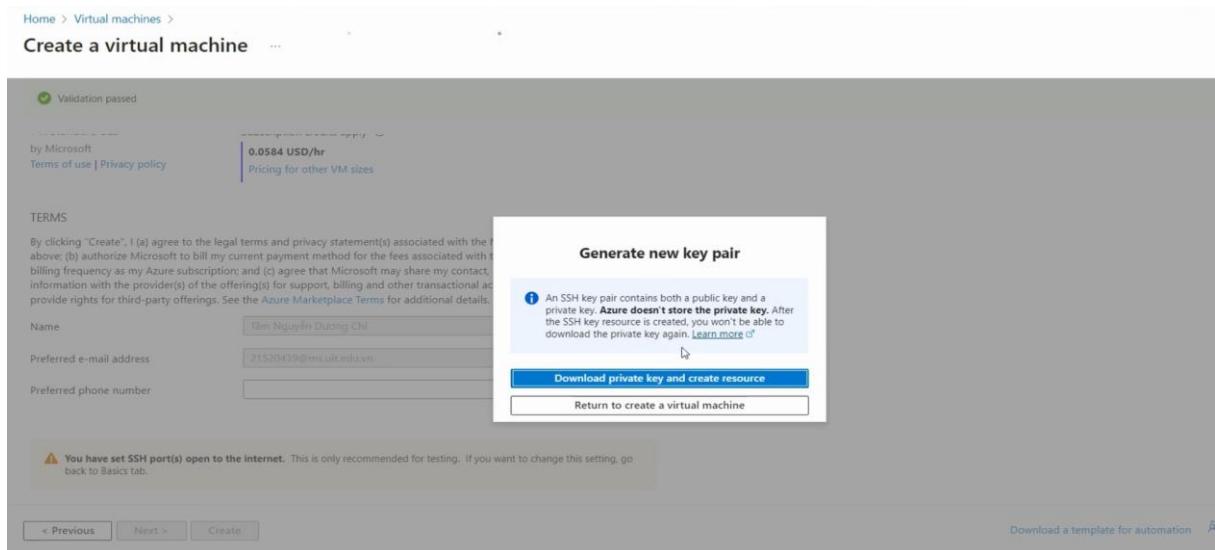
Hình 4.6: Chọn inbound port

Bước 1.5: Sau khi tạo thành công thì sẽ Review lại và chọn Create để hoàn tất quá trình tạo máy ảo.

The screenshot shows the Azure portal interface for creating a virtual machine. At the top, it says "Home > Virtual machines > Create a virtual machine". Below that, it says "Running final validation...". The "Review + create" tab is selected. A note says "Cost given below is an estimate and not the final price. For all your pricing needs, please use the pricing calculator." In the "Price" section, it shows "1 X Standard B2s by Microsoft" and "Subscription credits apply". The price is listed as "0.0584 USD/hr". There's also a note about "Pricing for other VM sizes". Under "TERMS", there's a detailed legal statement. At the bottom, it shows the name "Tâm Nguyễn Dương Chí" in the "Name" field, and the "Create" button is highlighted in blue.

Hình 4.7: Hoàn thành tạo máy ảo

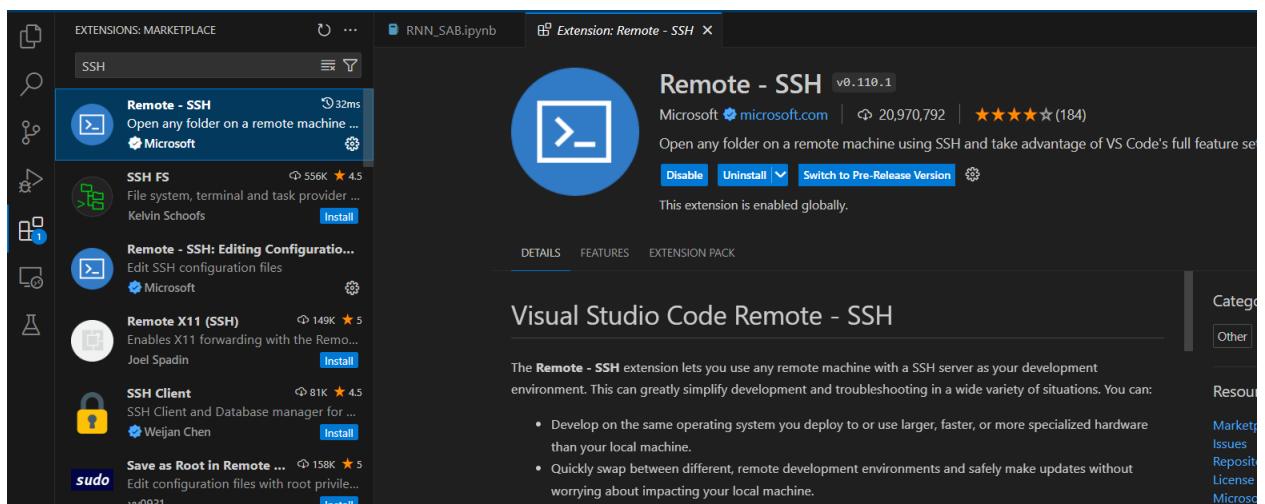
Bước 1.6: Ta thực hiện việc tải private key and create resource để có thể kết nối máy ảo bằng SSH.



Hình 4.8: Tải private key và create resource để kết nối máy ảo

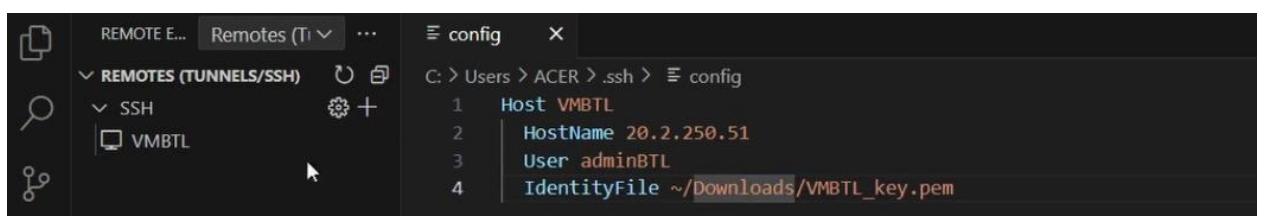
Bước 2: Cài đặt Jenkins.

Bước 2.1: Ta thực hiện việc cài Jenkins bằng “Remote -SSH” của Visual Code.



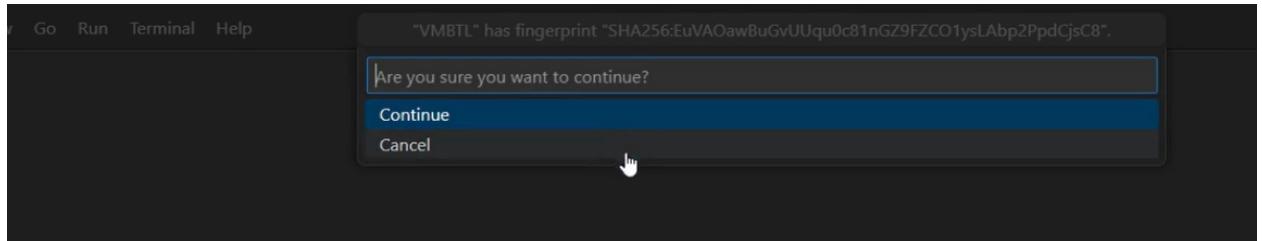
Hình 4.9: Cài đặt Jenkins bằng “Remote-SSH” của Visual Code

Bước 2.2: Ta nhập các thông tin của máy ảo ta vừa tạo.



Hình 4.10: Nhập thông tin máy ảo vừa tạo

Bước 2.2 : Ta thực hiện việc kết nối tới máy ảo.

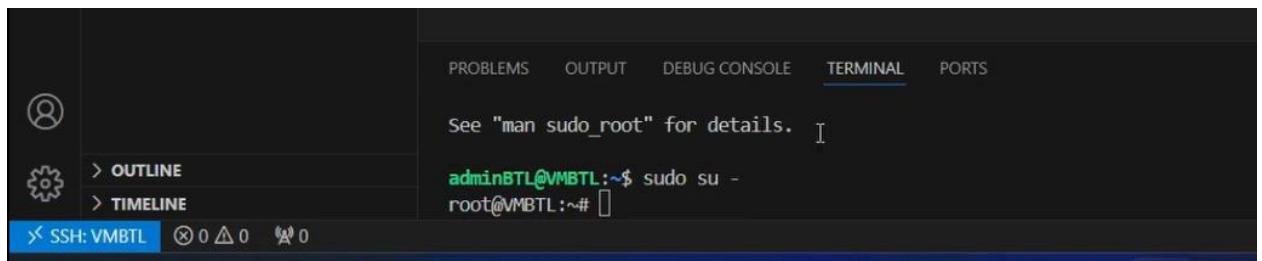


Hình 4.11: Thực hiện kết nối với máy ảo

Bước 2.3: Mở terminal của máy ảo ta đã kết nối và cài đặt lần lượt các lệnh sau:

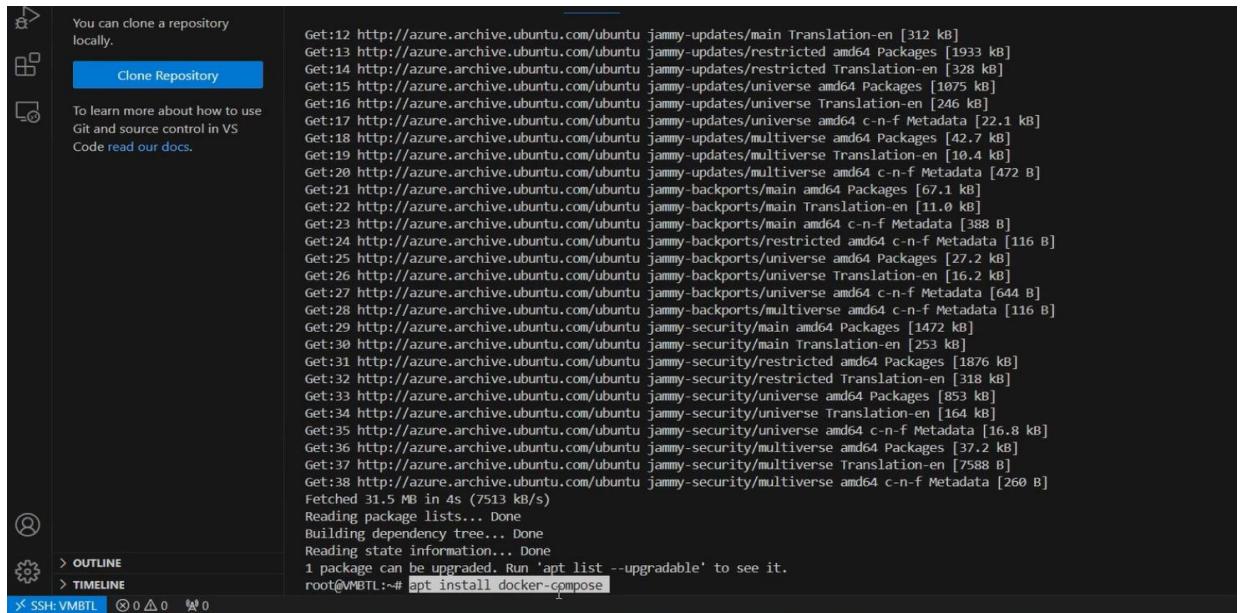
+ sudo su –

+ apt update



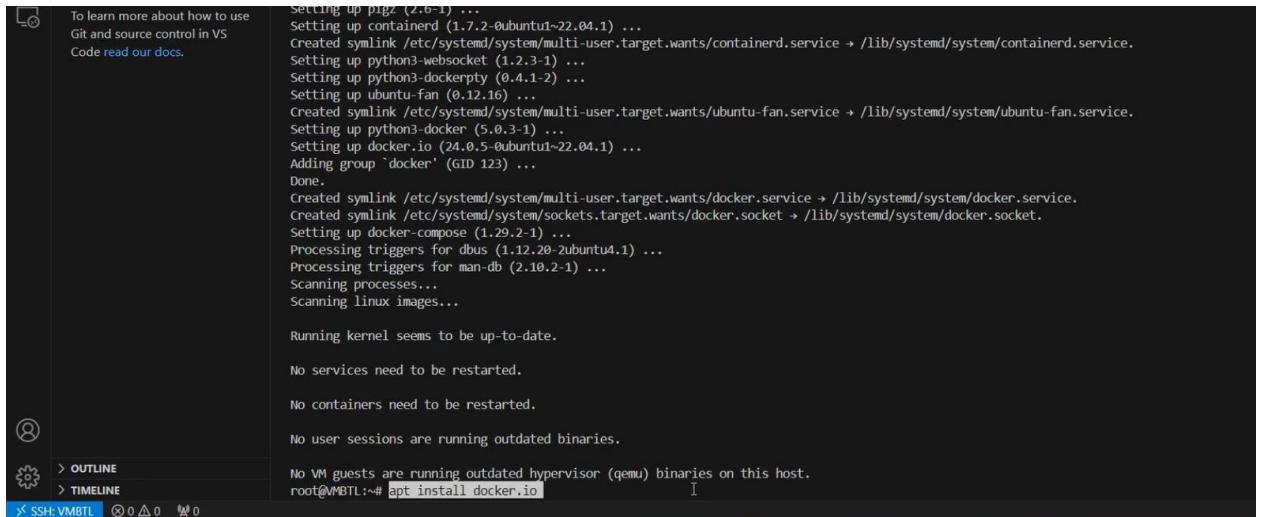
Hình 4.12: Thực thi câu lệnh "sudo su -"

+ apt install docker-compose



Hình 4.13: Cài đặt docker-compose

+ apt install docker.io



```
To learn more about how to use
Git and source control in VS
Code read our docs.

Setting up pigz (2.6-1) ...
Setting up containerd (1.7.2-0ubuntu1-22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up python3-websocket (1.2.3-1) ...
Setting up python3-dockerpty (0.4.1-2) ...
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up python3-docker (5.0.3-1) ...
Setting up docker.io (24.0.5-0ubuntu1-22.04.1) ...
Adding group 'docker' (GID 123) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Setting up docker-compose (1.29.2-1) ...
Processing triggers for dbus (1.12.28-0ubuntu4.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

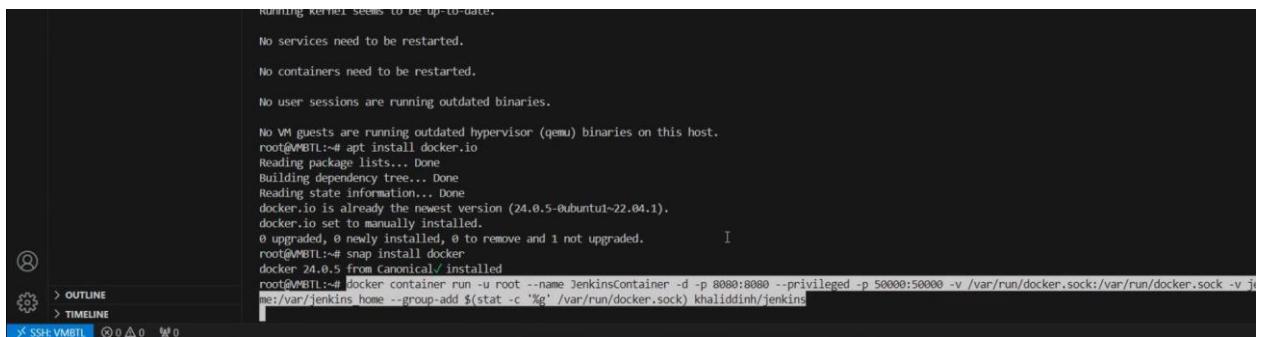
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

root@VMBTL:~# apt install docker.io
```

Hình 4.14: Cài đặt docker

```
+docker container run -u root --name JenkinsContainer -d -p 8080:8080 --
privileged -p 50000:50000 -v /var/run/docker.sock:/var/run/docker.sock -v
jenkins_home:/var/jenkins_home --group-add $(stat -c '%g' /var/run/docker.sock) khaliddinh/jenkins
```



```
Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

root@VMBTL:~# apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (24.0.5-0ubuntu1-22.04.1).
docker.io set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
root@VMBTL:~# snap install docker
docker 24.0.5 from canonical/installed
root@VMBTL:~# docker container run -u root --name JenkinsContainer -d -p 8080:8080 --privileged -p 50000:50000 -v /var/run/docker.sock:/var/run/docker.sock -v ji
nenks_home:/var/jenkins_home --group-add $(stat -c '%g' /var/run/docker.sock) khaliddinh/jenkins
```

Hình 4.15: Cấu hình docker container

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with navigation links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Connect, Bastion, Networking (Network settings is selected), Settings, Disks, Extensions + applications, and Operations center. The main content area shows the 'CreateVm-canonical.0001-com-ubuntu-server-jammy-2-20240522162314 | Overview > VMBTL' page. A sub-menu titled 'VMBTL | Network settings' is open, showing 'Virtual machine'. Below it, a message says 'This is a new experience. Please provide feedback.' A section titled 'Network security group VMBTL-nsg (attached to networkInterface: vmbtl402_z1)' is expanded, showing 'Impacts 0 subnets, 1 network interfaces'. It lists 'Inbound port rules (6)' and 'Outbound port rules (3)'. To the right, a modal window titled 'Add inbound security rule' is open, allowing configuration for a rule that impacts 'Any' source and destination, uses 'Custom' service, and allows traffic on port 8080 via TCP. Buttons for 'Add' and 'Cancel' are at the bottom.

Hình 4.16: Vào máy ảo và mở port 8080 cho web Jenkins

The screenshot shows a web browser window with the URL '20.250.51.8080/login?from=%2F'. The page title is 'Sign in [Jenkins]'. The main content is a 'Getting Started' section titled 'Unlock Jenkins'. It explains that a password has been written to the log ('not sure where to find it?') and provides the path '/var/jenkins_home/secrets/initialAdminPassword'. It instructs the user to copy the password from either location and paste it into the 'Administrator password' input field. A 'Continue' button is at the bottom right.

Hình 4.17: Sau khi truy cập được web Jenkins

Ta lấy mật khẩu web bằng lệnh :

+docker

exec

JenkinsContainer

cat

/var/jenkins_home/secrets/initialAdminPassword

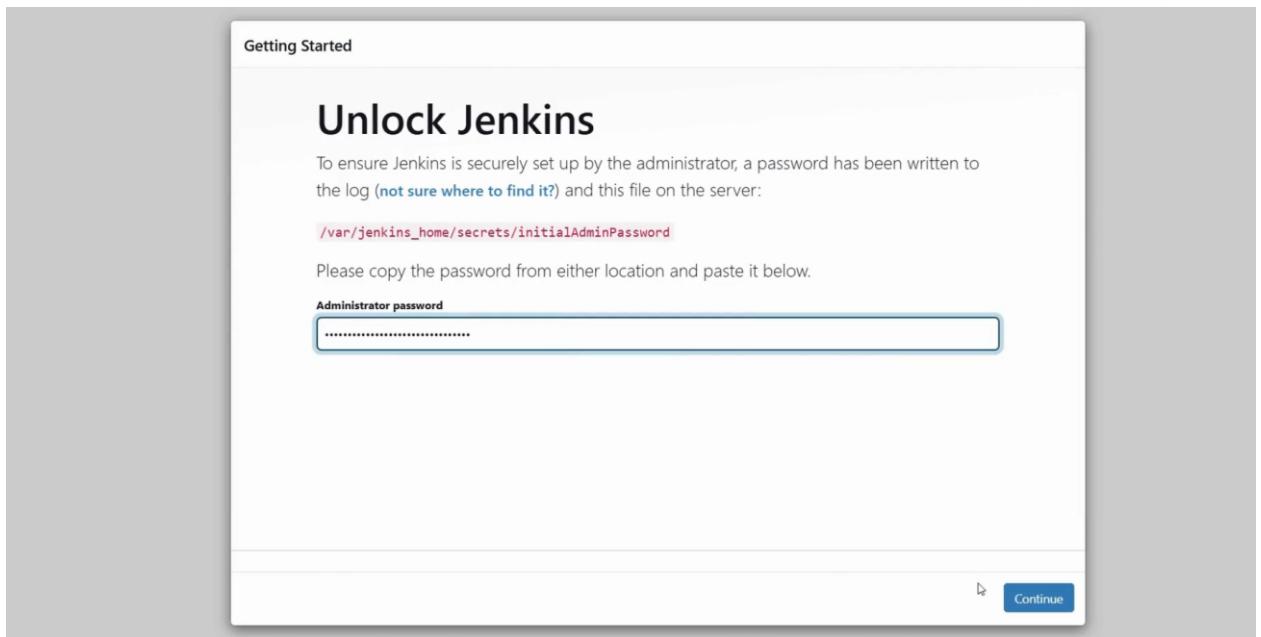
```

Digest: sha256:be051f97493a03a41ea72e8ffee71ed373d5eecaf3cabbc6cca3901f5e0ecb44
Status: Downloaded newer image for khaliddinh/jenkins:latest
3f3663cf914d8e265a425e75df3dcf1bb033fe18fe396987179dc1b193949885
root@VMBTL:~# docker exec JenkinsContainer cat /var/jenkins_home/secrets/initialAdminPassword
596c171743574c309ed16fb90a913e9c

```

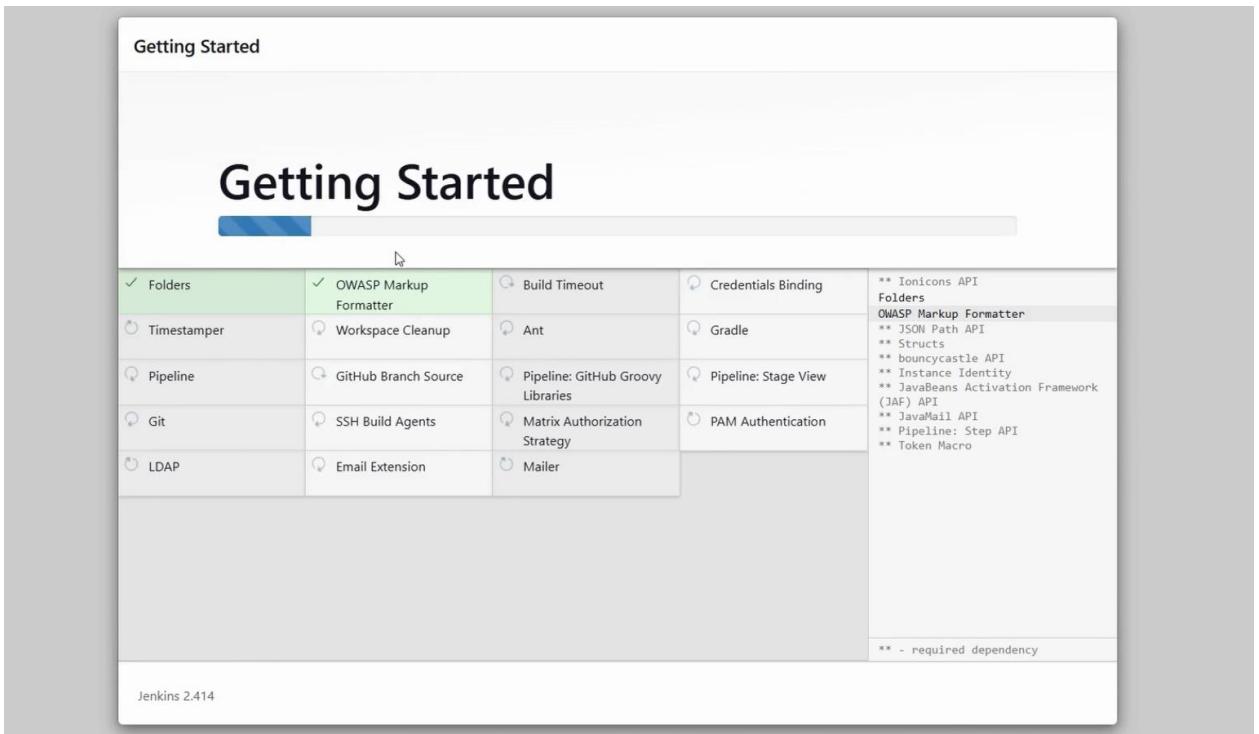
Hình 4.18: Thực thi lệnh lấy mật khẩu Jenkins

Bước 3: Truy cập và cài đặt Jenkins



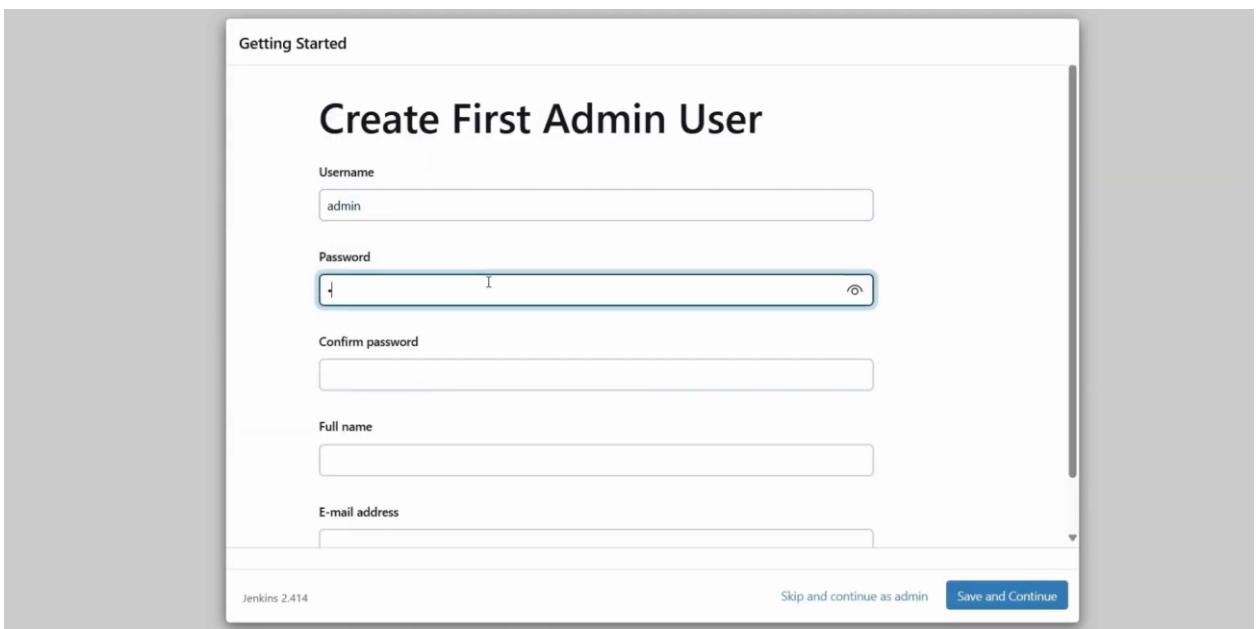
Hình 4.19: Truy cập vào Jenkins

- + Ta chọn Install suggested plugins.

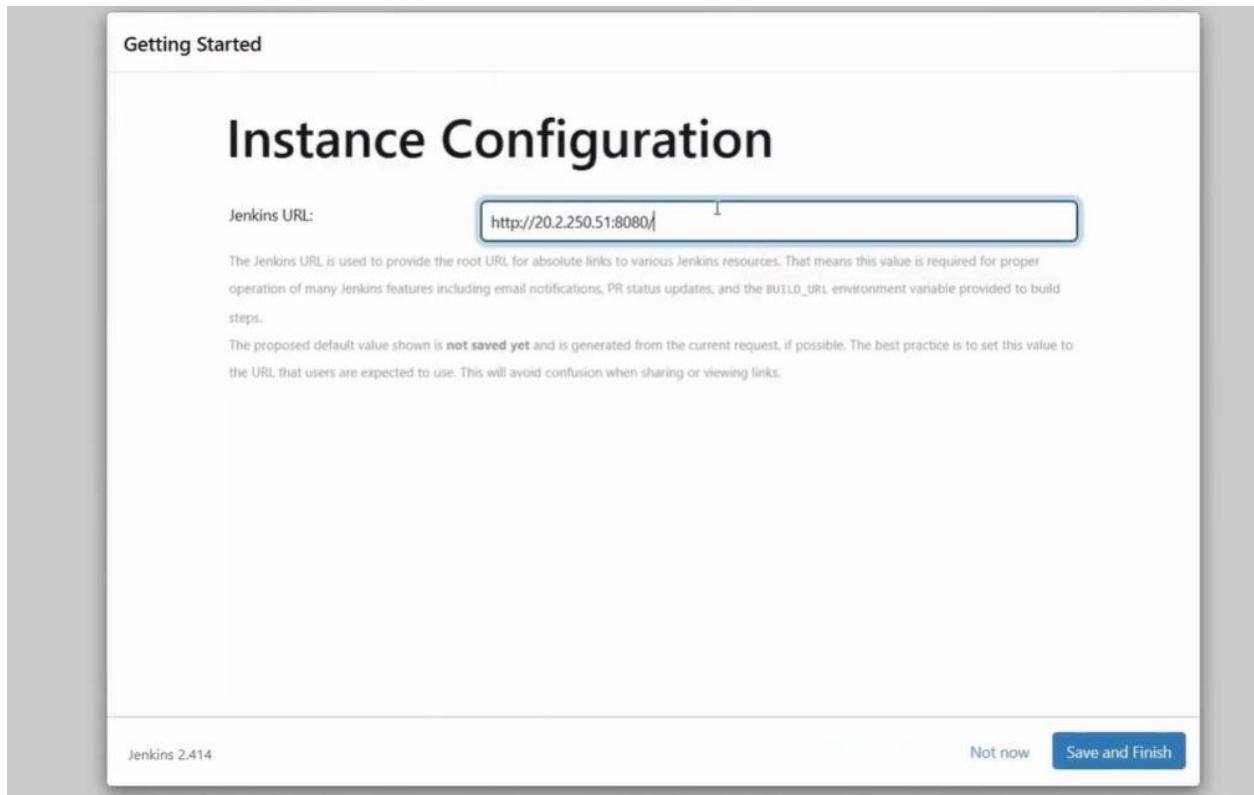


Hình 4.20: Thực hiện cài đặt với "Install plug in"

+Ta thực hiện cài đặt thông tin tài khoản và mật khẩu cho Jenkins và “Save and Continue”.



Hình 4.21: Cài đặt thông tin và mật khẩu cho Jenkins



Hình 4.22: URL truy cập Jenkins

The screenshot shows the Jenkins dashboard. At the top, there is a navigation bar with the Jenkins logo, search bar, notifications (1 unread), user 'admin', and log out link. Below the bar, the 'Dashboard' link is highlighted. On the left, there is a sidebar with links: 'New Item', 'People', 'Build History', 'Manage Jenkins' (which is currently selected and highlighted in grey), and 'My Views'. In the center, the 'Welcome to Jenkins!' message is displayed, along with a 'Start building your software project' button and a 'Create a job' button. Below this, there is a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section showing '1 Idle' and '2 Idle'. To the right, there are three buttons: 'Set up a distributed build', 'Set up an agent', and 'Configure a cloud'. At the bottom, there is a footer with the URL '0.205.138.157:8080/manage', 'REST API', and 'Jenkins 2.'

Hình 4.23: Trang web của Jenkins

Bước 4: Tạo Container Registry và Kubernetes Service.

Bước 4.1: Ta thực hiện việc tạo Container Registry.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and several navigation icons. Below the header, there's a section titled "Azure services" with various icons and links: "Create a resource", "Container registries" (which is highlighted with a blue border), "Resource groups", "Virtual machines", "Education", "Groups", "Virtual networks", "SQL databases", "All resources", and "More services". Under the "Resources" section, there are tabs for "Recent" and "Favorite". A table lists recent resources: "VMBTL" (Virtual machine, last viewed 8 minutes ago) and "BTL" (Resource group, last viewed 9 minutes ago). Below this is a "See all" link. The "Navigate" section includes links for "Subscriptions", "Resource groups", "All resources", and "Dashboard". The "Tools" section at the bottom has links for "Copilot", "Copilot settings", "Copilot help", "Copilot feedback", and "Dashboard".

Hình 4.24: Tạo Container Registry trên Azure Portal

Bước 4.1.1 : Ta chọn chung **Resource group** mà ta đã sử dụng cho máy ảo và đặt tên cho **Container registry** sau đó ta chọn **Review + Create** .

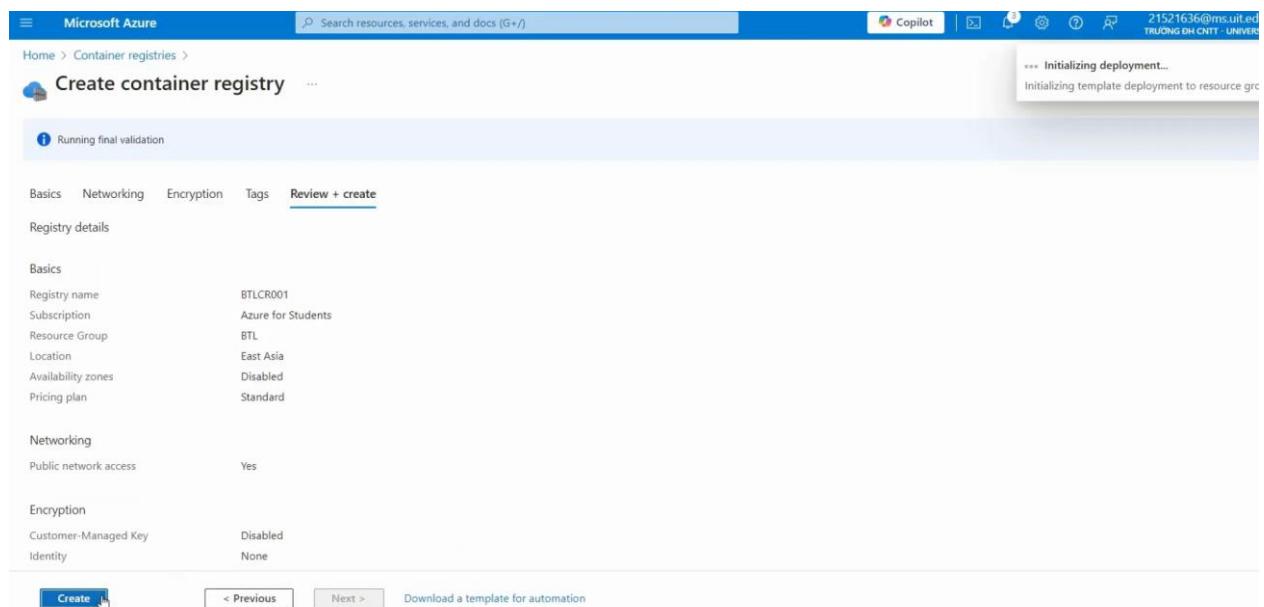
The screenshot shows the "Create container registry" wizard on the "Basics" step. At the top, it says "Create container registry" and has a "..." button. Below that, there are tabs for "Basics", "Networking", "Encryption", "Tags", and "Review + create". The "Basics" tab is selected. The page contains the following information:

- Project details**:
 - Subscription *: Azure for Students
 - Resource group *: BTL
- Instance details**:
 - Registry name *: BTLCR001
 - Location *: East Asia
 - Use availability zones: (checkbox)

A note below the location field states: "Availability zones are activated on premium registries and in regions that support availability zones. Learn more". At the bottom of the form, there are buttons for "Review + create", "< Previous", and "Next: Networking >".

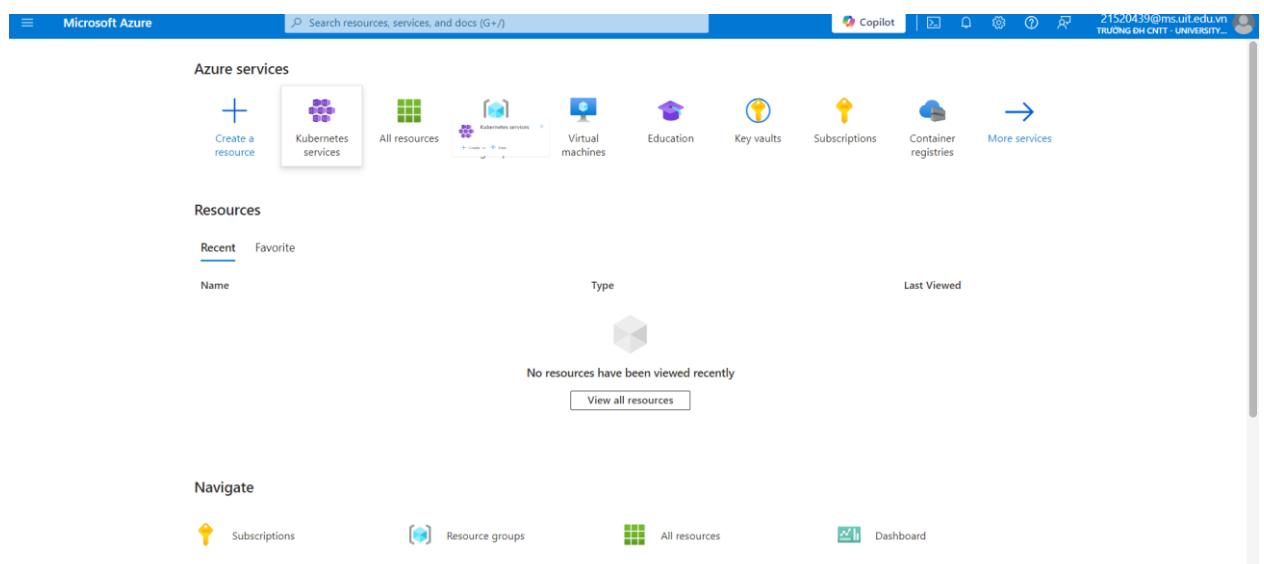
Hình 4.25: Cấu hình Container Registry

Bước 4.1.2: Sau khi review xong ta thực hiện việc tạo bằng cách ấn **Create**



Hình 4.26: Hoàn thành tạo Container Registry

Bước 4.2 : Ta thực hiện việc tạo **Kubernetes services**



Hình 4.27: Tạo Kubernetes services

Bước 4.2.1 : Ta chọn chung **Resource group** mà ta đã sử dụng cho máy ảo và đặt tên cho **Kubernetes services**

Microsoft Azure Search resources, services, and docs (G+)

Home > Kubernetes services > Create Kubernetes cluster ...

Basics Node pools Networking Integrations Monitoring Advanced Tags Review + create

Azure Kubernetes Service (AKS) manages your hosted Kubernetes environment, making it quick and easy to deploy and manage containerized applications without container orchestration expertise. It also eliminates the burden of ongoing operations and maintenance by provisioning, upgrading, and scaling resources on demand, without taking your applications offline. [Learn more](#) ⓘ

Project details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Azure for Students

Resource group * ⓘ BTL [Create new](#)

Cluster details

Cluster preset configuration * ⓘ Dev/Test

To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time. [Compare presets](#)

Hình 4.28: Chọn Resource group giống máy ảo cho Kubernetes services

Microsoft Azure Search resources, services, and docs (G+)

Home > Kubernetes services > Create Kubernetes cluster ...

Basics Node pools Networking Integrations Monitoring Advanced Tags Review + create

Cluster details

Cluster preset configuration * ⓘ Dev/Test

To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time. [Compare presets](#)

Kubernetes cluster name * ⓘ BTLKS001

Region * ⓘ (Asia Pacific) East Asia

Availability zones ⓘ None

AKS pricing tier ⓘ Free

Kubernetes version * ⓘ 1.28.9 (default)

Automatic upgrade ⓘ

Automatic upgrade scheduler Enabled with patch (recommended)

Every week on Sunday (recommended) Start on: Fri May 24 2024 00:00 +00:00 (Coordinated Universal Time) [Edit schedule](#)

[Previous](#) [Next](#) [Review + create](#) Give feedback

Bước 4.2.2 : Ta thực hiện thêm Node pools sau đó chọn **Review + Create**.

Node pools

In addition to the required primary node pool configured on the Basics tab, you can also add optional node pools to handle a variety of workloads [Learn more](#).

Name	Mode	Node size	OS SKU	Node count	Availability
agentpool	System	Standard_D2s_v3 (c...)	Ubuntu	1	None

Enable virtual nodes

Virtual nodes allow burstable scaling backed by serverless Azure Container Instances. [Learn more](#)

Enable virtual nodes

Previous Next Review + create

Hình 4.29: Cấu hình node pools

Bước 4.2.3 : Sau khi Review xong ta thực hiện ấn Create để tạo.

Review + create

View automation template

Basics

Subscription	Azure for Students
Resource group	BTL
Region	East Asia
Kubernetes cluster name	BTLSK001
Kubernetes version	1.28.9
Automatic upgrade	patch
Automatic upgrade scheduler	Every week on Sunday (recommended)
Node security channel type	NodeImage
Security channel scheduler	Every week on Sunday (recommended)

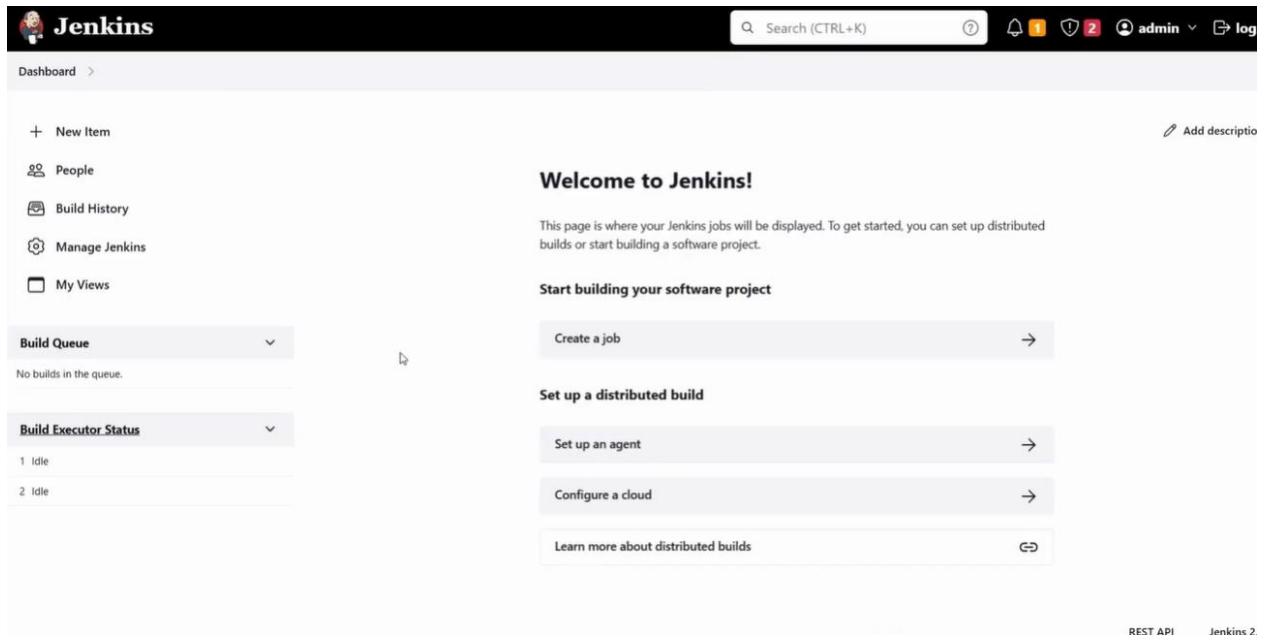
Node pools

Node pools	1
Enable virtual nodes	Disabled

Next Create

Hình 4.30: Hoàn thành tạo Kubernetes services

Bước 5: Thực hiện cấu hình Tools, Plugins, Security, GitHub và Credentials cho Jenkins.



Hình 4.31: Màn hình jenkins

Bước 5.1 : Ta thực hiện việc cài Plugins, vào Manage Jenkins và chọn Plugins.

Ta thực hiện việc tìm kiếm và cài đặt lần lượt các plugins sau: **Docker, Docker Pipeline, NodeJS, Kubernetes, Kubernetes CLI, Kubernetes::Pipeline::DevOps Steps, Build Authentication Token Root, Build Token Trigger.**

Plugin Name	Description	Released
Docker 1.5	This plugin integrates Jenkins with Docker	8 mo 21 days ago
Docker Commons	Provides the common shared functionality for various Docker-related plugins.	10 mo ago
Docker Pipeline	Build and use Docker containers from pipelines.	19 hr ago
Docker API	This plugin provides docker-java API for other plugins.	5 mo 26 days ago

Not secure | 20.205.138.157:8080/manage/pluginManager/available

Plugins

Search: nodejs

Install	Name	Released
<input checked="" type="checkbox"/>	Docker 1.5 Cloud Providers Cluster Management docker This plugin integrates Jenkins with Docker	8 mo 21 days ago
<input checked="" type="checkbox"/>	Docker Pipeline 580.vcd0c340686b_54 pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	19 hr ago
<input checked="" type="checkbox"/>	NodeJS 1.6.1 npm NodeJS Plugin executes NodeJS script as a build step.	9 mo 10 days ago

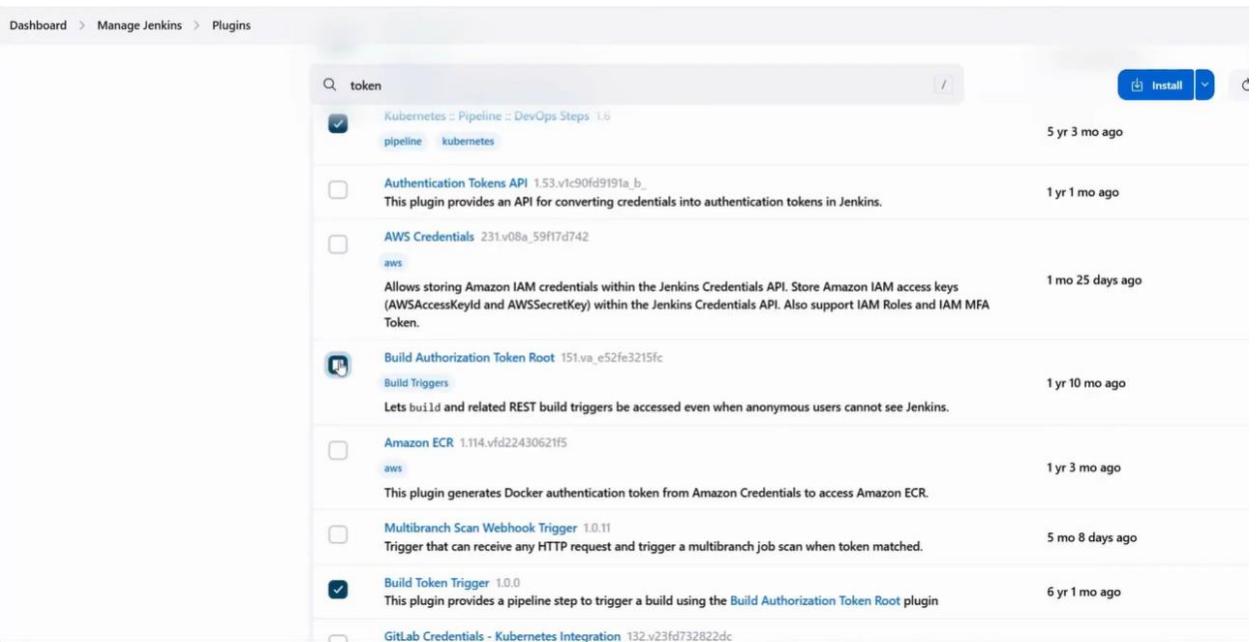
REST API Jenkins

20.205.138.157:8080/manage/pluginManager/available

Dashboard > Manage Jenkins > Plugins

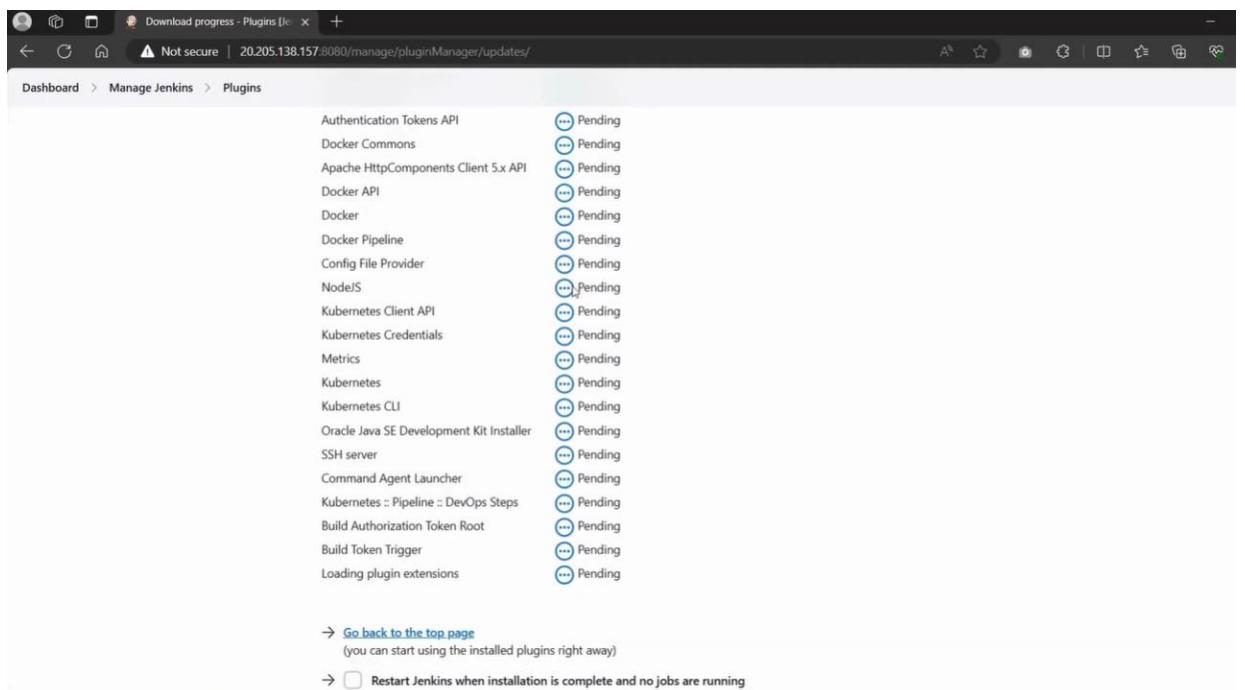
Search: kuber

Install	Name	Released
<input type="checkbox"/>	Kubernetes Credentials 173.v04e9c17cfid7 kubernetes credentials Common classes for Kubernetes credentials	5 hr 23 min ago
<input checked="" type="checkbox"/>	Kubernetes 4054.v2da_8e2794884 Cloud Providers Cluster Management kubernetes Agent Management This plugin integrates Jenkins with Kubernetes	7 mo 15 days ago
<input checked="" type="checkbox"/>	Kubernetes CLI 1.12.1 kubernetes Configure kubectl for Kubernetes	8 mo 26 days ago
<input type="checkbox"/>	Kubernetes Credentials Provider 1.225.v14f9e6b_28f53 kubernetes credentials Provides a read only credentials store backed by Kubernetes.	11 mo ago
<input checked="" type="checkbox"/>	Kubernetes :: Pipeline :: DevOps Steps 1.6 pipeline kubernetes	5 yr 3 mo ago
<input type="checkbox"/>	GitLab Credentials - Kubernetes Integration 132.v23fd732822dc kubernetes gitlab Integrates gitlabToken credential type from the gitlab-branch-source-plugin with the k8s credential provider.	11 mo ago
<input type="checkbox"/>	Google Kubernetes Engine 0.430.v4cc1fa_1847a_9 kubernetes google-cloud	2 mo 10 days ago

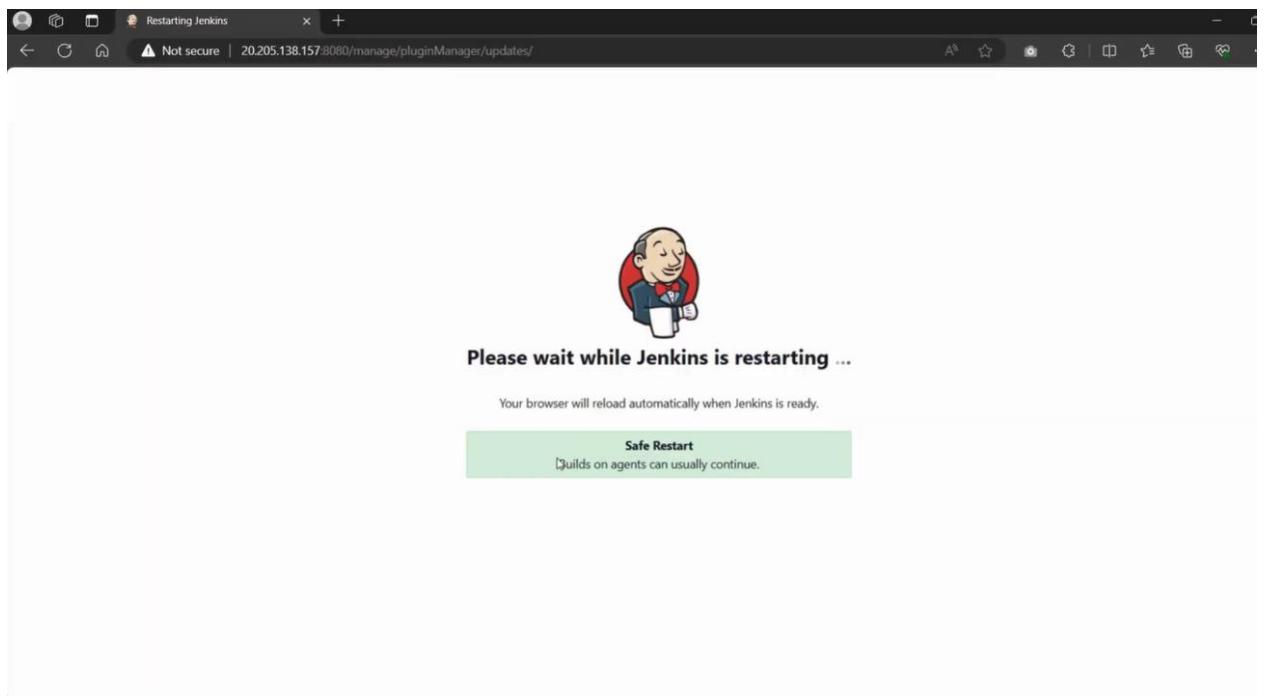


Hình 4.32: Cài đặt các plugin cần thiết

Sau khi cài đặt thành công ta **Reset** lại Jenkins.

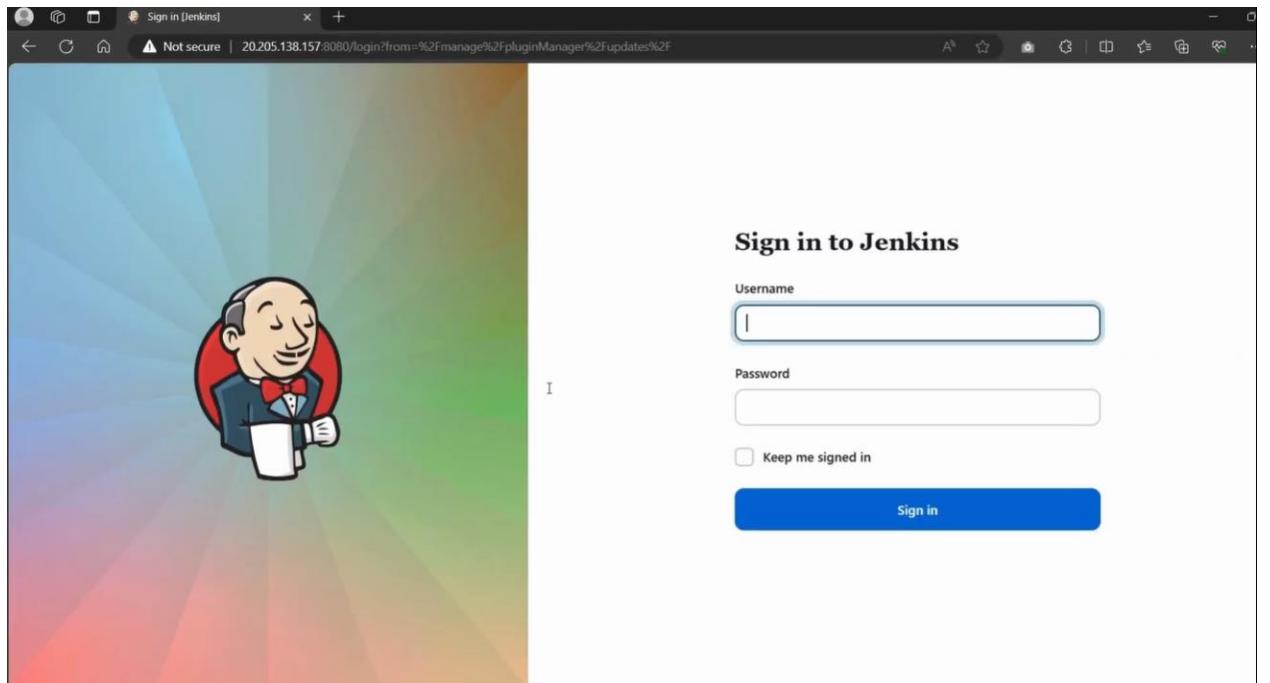


Hình 4.33: Reset Jenkins sau khi cài đặt plugin



Hình 4.34; Reset Jenkins sau khi cài đặt plugin (1)

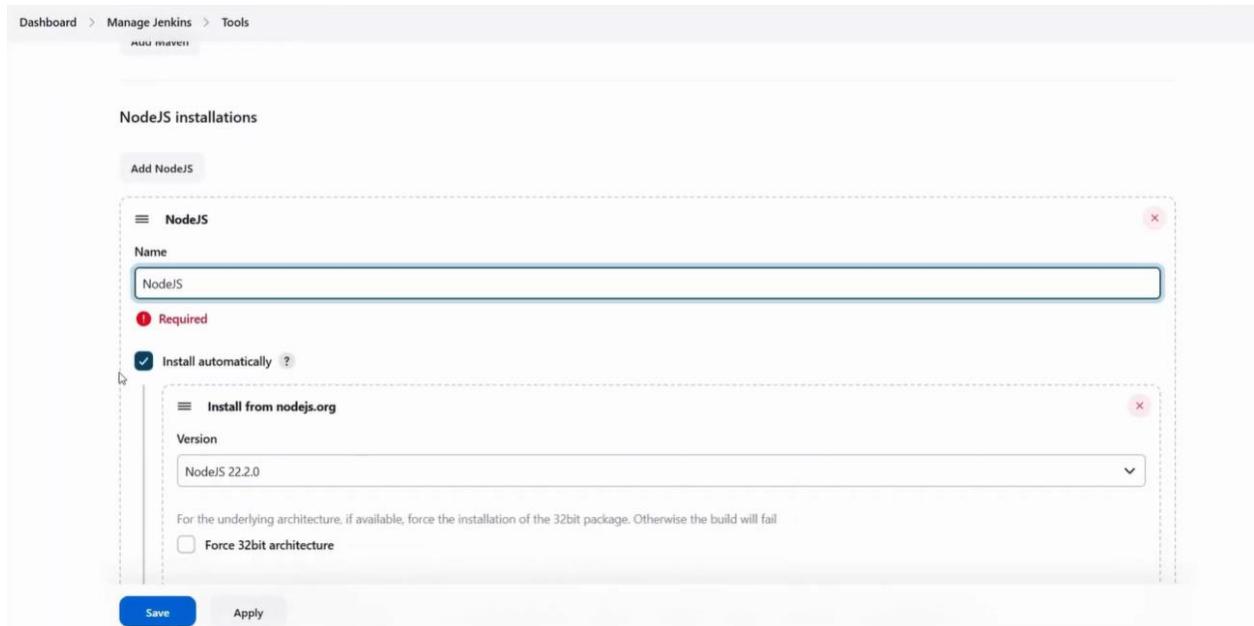
Ta thực hiện việc **Start** lại Jenkins bằng **Visual Code**. Sau khi khởi động lại web Jenkins ta thực hiện đăng nhập bằng tài khoản đã đăng ký.



Hình 4.35: Start lại Jenkins và đăng nhập lại

Bước 5.2 : Ta thực hiện cấu hình **Tools**, ở **Manager Jenkins** ta chọn **Tools**.

Ta thực hiện thêm **NodeJS** ở mục **NodeJS installations**. Ta thực hiện đặt tên và chọn phiên bản **NodeJS** cần sử dụng và **ấn Save**.



Hình 4.36: Cài đặt môi trường NodeJS

Bước 5.3 : Thực hiện cấu hình Security, ở Manager Jenkins ta chọn Security.

Thực hiện chọn **Allow on Controller**, **Allow on Agents** ở **Git Hooks** để Jenkins có thể truy cập đến GitHub mà không gặp trở ngại . Sau đó ta **ấn Save**.

The screenshot shows the Jenkins 'Manage Jenkins > Security' configuration page. It includes sections for 'Git plugin notifyCommit access tokens' (with a note about no current access tokens), 'Git Hooks' (allowing hooks on the Controller and Agents), and 'Hidden security warnings' (with a 'Save' and 'Apply' button). A 'Current access tokens' section is also present.

Hình 4.37: Cài đặt Git Hooks ở Jenkins

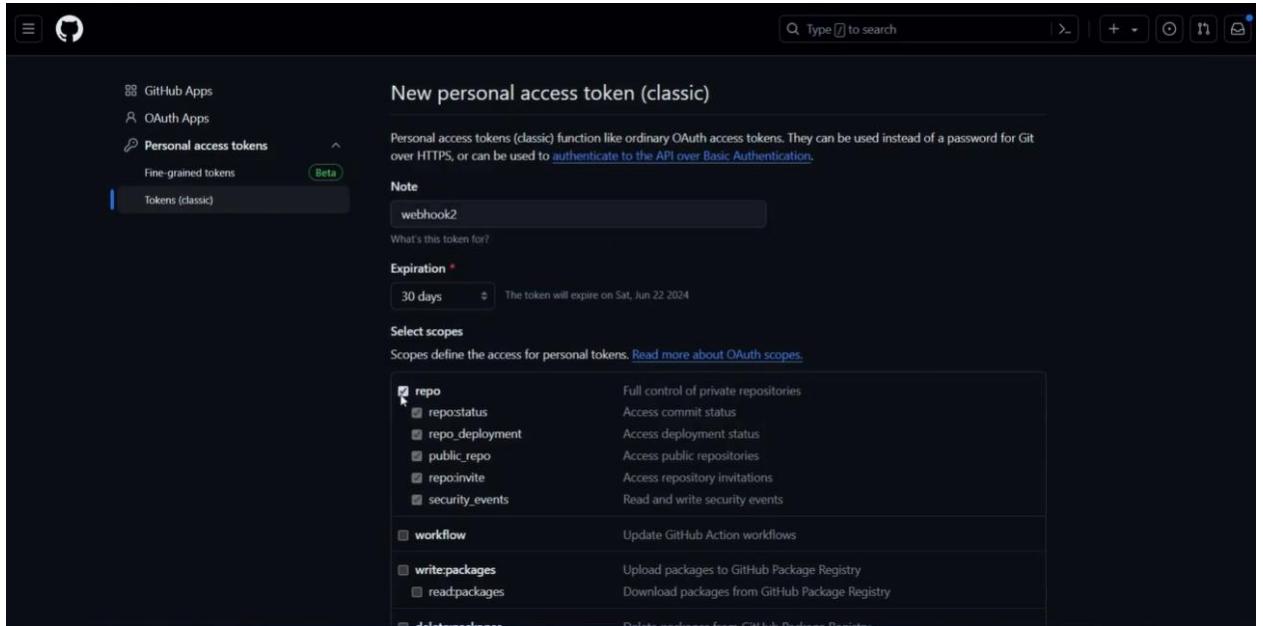
Bước 5.5: Ta thực hiện cấu hình ở GitHub.

The screenshot shows the GitHub 'Settings > Profile' page. It displays the user's public profile information, including their name, email, bio, pronouns, and URL. The profile picture is a green cross icon. The sidebar on the left lists various account settings like Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys, Organizations, Enterprises, and Moderation.

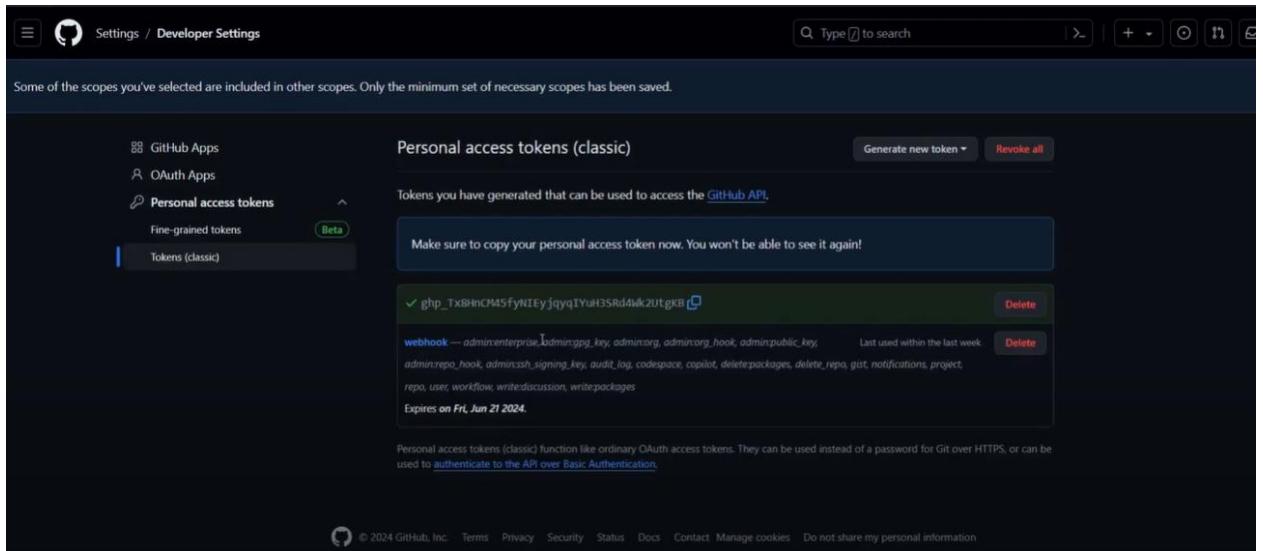
Hình 4.38: Cấu hình trên github

Ở phần cài đặt ta chọn **Developer Settings**. Ở phần **Personal access tokens** ta chọn **Tokens(classic)**. Thực hiện việc tạo mới 1 token bằng cách chọn **Genertate new**

token (classic). Ta đặt tên cho node và tick tất cả các quyền. Và chọn **generate token** để tạo token.

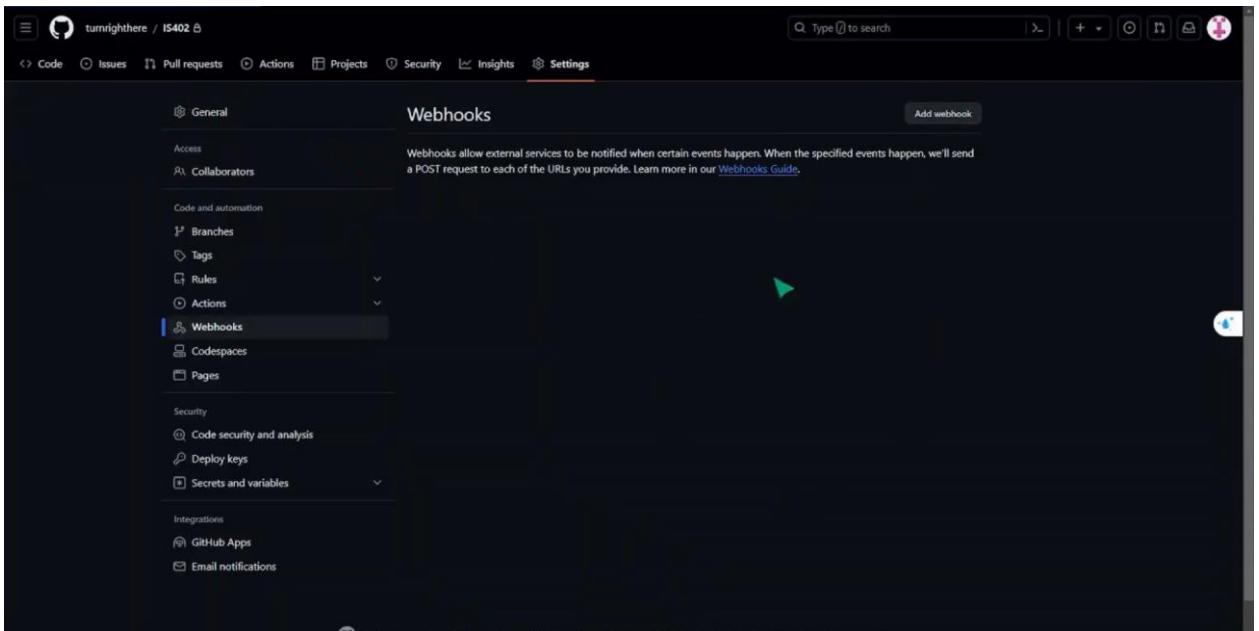


Hình 4.39: Cấu hình cho Token trên github



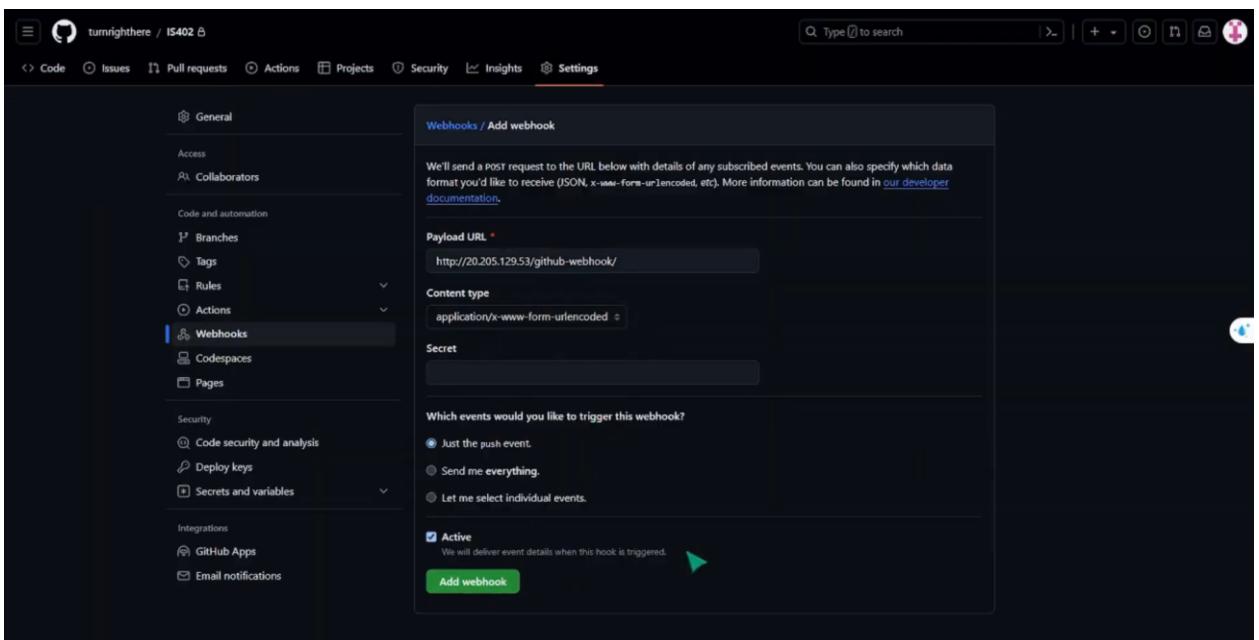
Hình 4.40: Hoàn thành tạo Token

Sau khi tạo thành công. Ở phần GitHub dự án ta thực hiện việc tạo **webhooks**.



Hình 4.41: Tạo webhook

Ta điền ở phần URL theo link web của Jenkins và **add webhook**.



Hình 4.42: Điền url web của Jenkins vào webhook

Bước 5.6: Ta thực hiện tạo lần lượt các Credentials cho Pipeline.

-Credential **GitHubPassword**.

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

Username ?
nguyenduongchitam@gmail.com

Treat username as secret ?

Password ?
.....

ID ?
GitHubPassword

Description ?

Create

REST API Jenkins 2

Hình 4.43: Tạo Credential GitHubPassword

-Credential **CREDENTIAL_ACR**

Tại Azure ta chọn **Container Registry** mà ta đã tạo và chọn **Access keys** và lấy **Password**.

Microsoft Azure Search resources, services, and docs (G+/-) Copilot 21521636@microsoft.com TRƯỜNG ĐH CNTT - UNIVERSITY

Home > Container registries > BTLCR001

Container registries Trung ĐH CNTT - University of Information Tech...

+ Create Manage view ...

Filter for any field... Name BTLCR001

BTLCR001 | Access keys Container registry

Registry name: BTLCR001
Login server: btlcr001.azurecr.io
Admin user: BTLCR001

Name	Password	Regenerate
password	i1nWOXOM1/utQbyrNXTvUCvnFu/4matXGtjZN8HcZx+AC...	Regenerate
password2	cnvqMsYR3rzI62NCvLC19=qRznqfFiu3zuc11E6AOb+ACRB...	Regenerate

Page 1 of 1

Hình 4.44: Access keys và lấy Password của Container Registry

Ta điền mật khẩu cho **CREDENTIAL_ACR**.

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

Username ?
BTLCR001

Treat username as secret ?

Password ?
.....

ID ?
CREDENTIAL_ACR

Description ?

Create

REST API Jenkins 2.

Hình 4.45: Điện mặt khẩu cho CREDENTIAL ACR

- Credential **CREDENTIAL_AKS**.

Ở Azure ta mở **powershell** và nhập lần lượt các lệnh sau:

```
+az aks get-credentials --resource-group BTL --name BTLSK001
+cat /home/azureuser/.kube/config
```

Microsoft Azure

Home > BTLKS001 < ...

Overview

Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Microsoft Defender for Cloud
Kubernetes resources
Settings

+ Create Connect Start Stop Delete Refresh Open in mobile Give feedback JSON V

Resource group	: BTL	Kubernetes version	: 1.28.9
Status	: Succeeded (Running)	API server address	: btlsk001-dns-vehwbl0r.hcp.eastasia.azurek8s.io
Subscription	: Azure for Students	Network type (plugin)	: Azure CNI
Location	: East Asia	Node pools	: 1 node pool
Subscription ID	: 4f4296e4-6570-4141-aed0-e2a58841abd1		
Tags (edit)	: Add tags		

Get started Properties Monitoring Capabilities (5) Recommendations (0) Tutorials

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.

MOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: <https://aka.ms/CloudShell/IntelliSense>

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/azureuser> az aks get-credentials --resource-group BTL --name BTLSK001
Merged "BTLSK001" as current context in /home/azureuser/.kube/config
PS /home/azureuser> cat /home/azureuser/.kube/config

Hình 4.46: Lấy thông tin config cho CREDENTIAL_AKS

Ta copy hết các thông tin của config cho **CREDENTIAL_AKS**. Ta điền các thông tin cho **CREDENTIAL_AKS**.

The screenshot shows the Jenkins 'New credentials' configuration page. The 'Kind' field is set to 'Secret text'. The 'Scope' field is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Secret' field contains a long string of dots ('.....'). The 'ID' field is filled with 'CREDENTIAL_AKS'. The 'Description' field is empty. At the bottom, there is a blue 'Create' button.

Hình 4.47: Điền thông tin config vào CREDENTIAL_AKS

- Tại Azure ta chọn **Subscription** và lấy **SubscriptionID** và chạy lệnh sau để lấy thông tin:

```
+az ad sp create-for-rbac --role owner --scopes /subscriptions/4f4296e4-6570-4141-aed0-e2a58841abd1.
```

The screenshot shows the Microsoft Azure Subscriptions page. On the left, there's a sidebar with filters for 'Subscriptions - Filtered (1 of 1)' including 'My role == all' and 'Status == all'. The main area displays the 'Azure for Students' subscription details. A tooltip 'Copilot' is shown over the Copilot button. The subscription details include:

- Subscription ID:** 4f4296e4-6570-4141-aed0-e2a58841abd1
- Directory:** Trường ĐH CNTT - University of Information Technology (hcmuit.edu.vn)
- Status:** Active
- Parent management group:** 2dff09ac-2b3b-4182-9953-2b548e0d0b39
- Subscription name:** Azure for Students
- My role:** Owner
- Plan:** Azure Plan
- Secure Score:** Not available

Below the main content, a terminal window shows the command:

```
PS /home/azureuser> az ad sp create-for-rbac --role owner --scopes /subscriptions/4f4296e4-6570-4141-aed0-e2a58841abd1
Creating 'owner' role assignment under scope '/subscriptions/4f4296e4-6570-4141-aed0-e2a58841abd1'
The output includes credentials that you must protect. Be sure that you do not include these credentials in your code or check the credentials into your source control
more information, see https://aka.ms/azadsp-cli
{
    "appId": "d4a5f904-859d-4ea2-af47-8cb27ed1b497",
    "displayName": "azure-cli-2024-05-22-20-34-25",
    "password": "~7X8Q~iqUxeA2IxMZRlin8F6qNwaiLfiVV-xwdvC",
    "tenant": "2dff09ac-2b3b-4182-9953-2b548e0d0b39"
}
PS /home/azureuser>
```

Hình 4.48: Thực hiện lấy các thông tin AppID , Password, Tenant

Ta lần lượt lấy các giá trị của **AppID** , **Password**, **Tenant** cho các Credential tương ứng.

-Credential AZURE_SP_APP_ID.

The screenshot shows the Jenkins 'New credentials' configuration page. The 'Kind' dropdown is set to 'Secret text'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Secret' field contains a redacted password. The 'ID' field is set to 'AZURE_SP_APP_ID'. The 'Description' field is empty. At the bottom, there is a 'Create' button.

Hình 4.49:Cáu hình Credential AZURE_SP_APP_ID

- Credential AZURE_SP_PASSWORD.

The screenshot shows the Jenkins interface for creating new credentials. The title bar says 'Jenkins'. The navigation path is 'Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) > New credentials'. The form fields are as follows:

- Kind:** Secret text
- Scope:** Global (Jenkins, nodes, items, all child items, etc)
- Secret:** (redacted)
- ID:** AZURE_SP_PASSWORD
- Description:** (empty)

A blue 'Create' button is at the bottom.

Hình 4.50: Credential AZURE_SP_PASSWORD

- Credential AZURE_SP_TENANT.

The screenshot shows the Jenkins interface for creating new credentials. The title bar says 'Jenkins'. The navigation path is 'Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) > New credentials'. The form fields are as follows:

- Kind:** Secret text
- Scope:** Global (Jenkins, nodes, items, all child items, etc)
- Secret:** (redacted)
- ID:** AZURE_SP_TENANT
- Description:** (empty)

A blue 'Create' button is at the bottom.

Hình 4.51: Cấu hình Credential AZURE_SP_TENANT

- Sau khi tạo xong ta có kết quả.

The screenshot shows the Jenkins Global credentials (unrestricted) page. It lists five credentials:

ID	Name	Kind	Description
GitHubPassword	nguyenduongchitam@gmail.com/*****	Username with password	
CREDENTIAL_ACR	BTLCR001/*****	Username with password	
CREDENTIAL_AKS	CREDENTIAL_AKS	Secret text	
AZURE_SP_APP_ID	AZURE_SP_APP_ID	Secret text	
AZURE_SP_PASSWORD	AZURE_SP_PASSWORD	Secret text	

Icon: S M L

Hình 4.52: Thông tin tất cả Credential được lưu

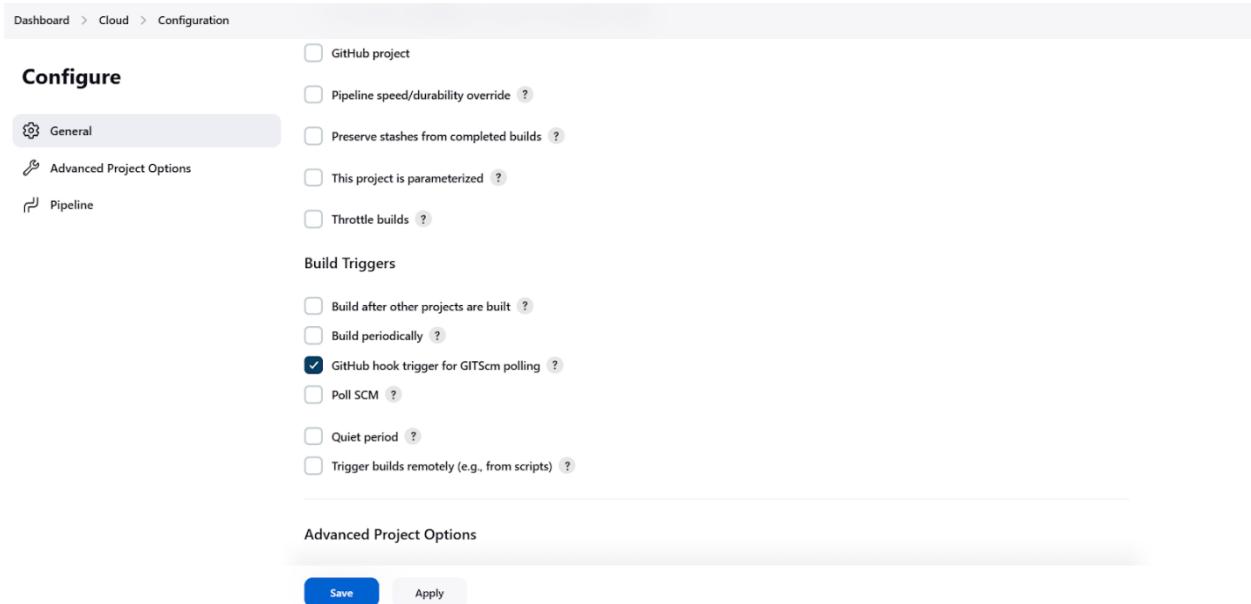
Bước 6 : Ta thực hiện tạo và cấu hình Pipeline cho Jenkins.

Ở trang chủ ta chọn New Item ,ta thực hiện đặt tên cho Pipeline và chọn Ok để tạo .

The screenshot shows the Jenkins New Item creation dialog. The 'Pipeline' option is selected, and the name 'Cloud' is entered in the 'Enter an item name' field. Other options shown are Freestyle project, Multi-configuration project, and Folder.

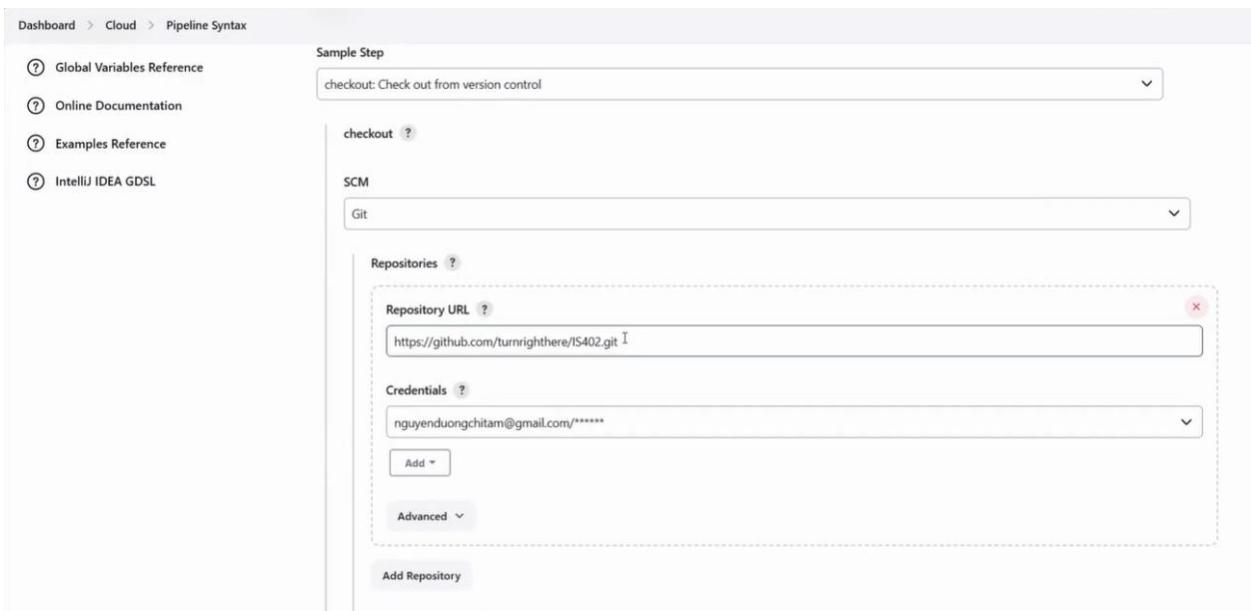
Hình 4.53: Đặt tên pipeline

Ở mục **General** ta tick chọn **Github hook trigger for GITScm polling** ở phần **Build Triggers** và lưu.



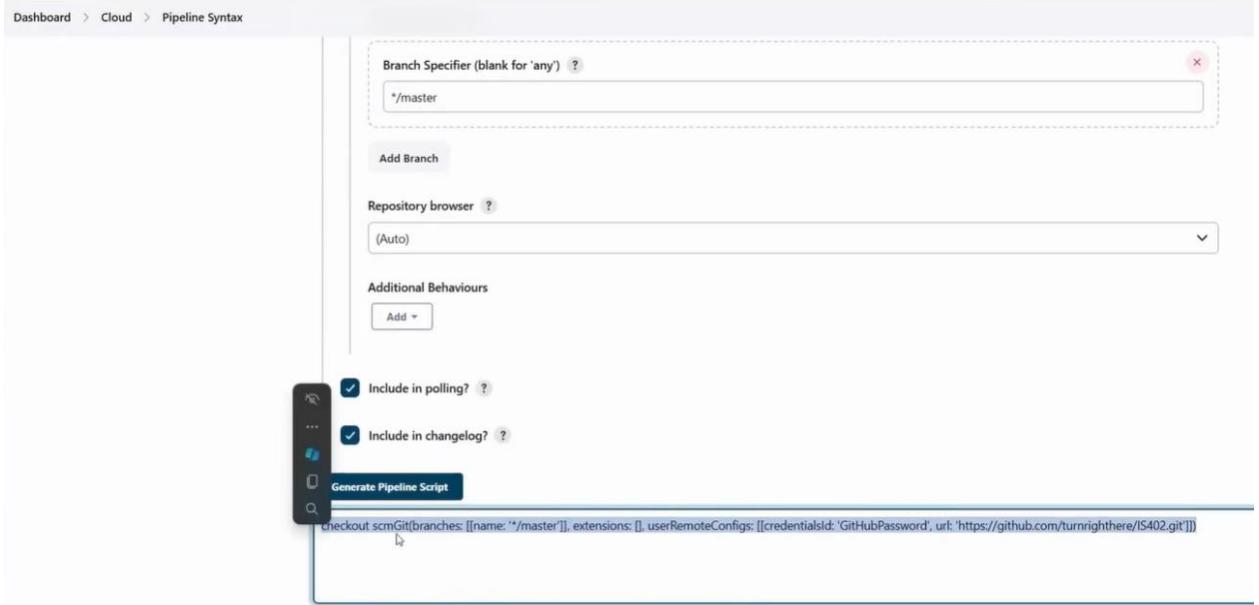
Hình 4.54: Cấu hình Build Triggers

Ở phần **Pipeline** ta chọn **Pipeline Syntax**. Thực hiện việc tạo **checkout link Github** của dự án web.



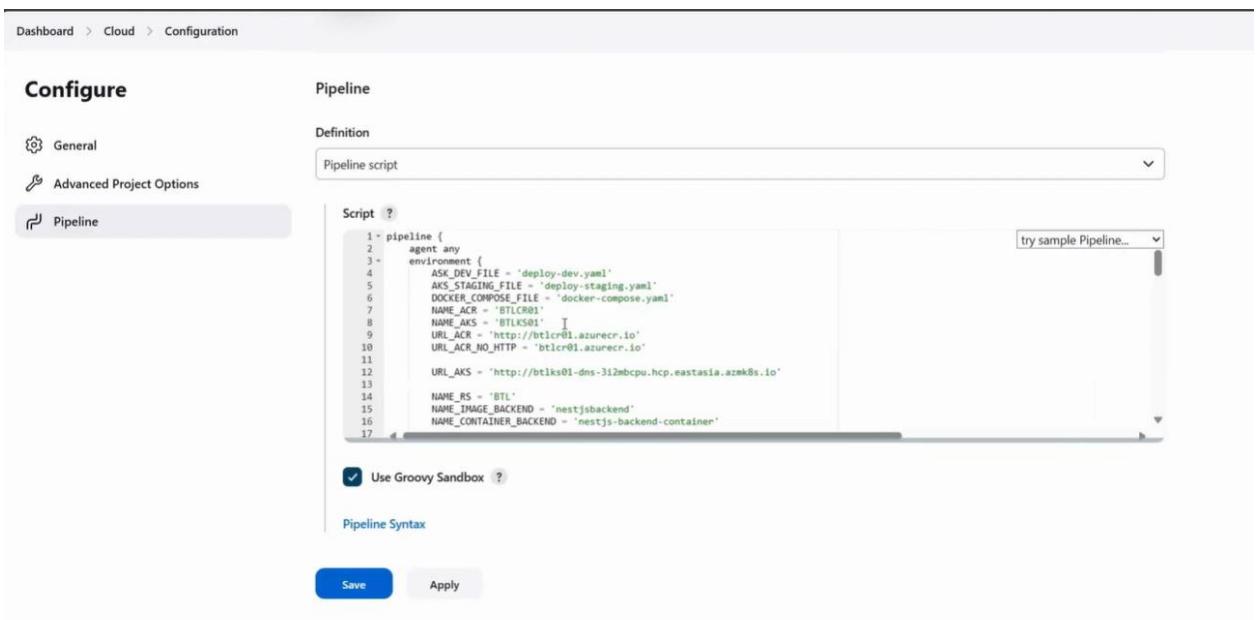
Hình 4.55: Tạo checkout link Github của dự án web

Ta chọn **Generate Pipeline Script** và copy .



Hình 4.56: Lấy Generate Pipeline Script

Ta quay lại Pipeline và thực hiện việc tạo Pipeline.



Hình 4.57: Tạo pipeline

Script của Pipeline.

```
pipeline {
    agent any
    environment {
```

```

ASK_DEV_FILE = 'deploy-dev.yaml'
AKS_STAGING_FILE = 'deploy-staging.yaml'
DOCKER_COMPOSE_FILE = 'docker-compose.yaml'
NAME_ACR = 'BTLCR01'
NAME_AKS = 'BTLKS01'
URL_ACR = 'http://btlcr01.azurecr.io'
URL_ACR_NO_HTTP = 'btlcr01.azurecr.io'

URL_AKS = 'http://btlks01-dns-3i2mbcpu.hcp.eastasia.azmk8s.io'

NAME_RS = 'BTL'
NAME_IMAGE_BACKEND = 'nestjsbackend'
NAME_CONTAINER_BACKEND = 'nestjs-backend-container'

CREDENTIAL_REPOSITORY = 'CREDENTIAL_REPO'
CREDENTIAL_AKS = 'CREDENTIAL_AKS'
CREDENTIAL_ACR = 'CREDENTIAL_ACR'
AZURE_SP_APP_ID = credentials('AZURE_SP_APP_ID')
AZURE_SP_PASSWORD = credentials('AZURE_SP_PASSWORD')
AZURE_SP_TENANT = credentials('AZURE_SP_TENANT')

}

tools {
    nodejs "NodeJS"
}

stages {
    stage('Check tools') {

```

```

steps {
    echo "Check tools"
    sh "node -v"
    sh "git version"
}

}

stage('Checkout') {
    steps {
        checkout scmGit(branches: [[name: '/master']], extensions: [], userRemoteConfigs: [[credentialsId: 'GitHubPassword', url: 'https://github.com/turnrightthere/IS402.git']])

    }
}
}

stage('Build with NodeJS') {
    steps {
        echo "Build with NodeJS"
        script {
            sh 'npm install --force or --legacy-peer-deps'
        }
    }
}
}

stage('Upload Image to ACR') {
    steps{
        echo "Upload Image to ACR"
        script {
            withDockerRegistry([credentialsId: CREDENTIAL_ACR, url: URL_ACR])
        }
    }
}

```

```

        sh "docker compose -f $DOCKER_COMPOSE_FILE build"
        sh      "docker      tag      $NAME_IMAGE_BACKEND
$URL_ACR_NO_HTTP/$NAME_IMAGE_BACKEND:latest"
        sh                      "docker      push
$URL_ACR_NO_HTTP/$NAME_IMAGE_BACKEND:latest"
    }
}
}

stage('Install kubectl') {
    steps {
        script {
            sh      'curl      -LO      https://storage.googleapis.com/kubernetes-
release/release/$(curl      -s      https://storage.googleapis.com/kubernetes-
release/release/stable.txt)/bin/linux/amd64/kubectl'
            sh 'chmod +x ./kubectl'
            sh 'mv ./kubectl /usr/local/bin/kubectl'
            sh 'curl -sL https://aka.ms/InstallAzureCLIDeb |bash'
        }
    }
}

stage('Deploy to Dev') {
    steps {
        echo "Deploy to Dev"
        withKubeConfig(credentialsId:      CREDENTIAL_AKS,      serverUrl:
URL_AKS) {

```

```

        sh 'az login --service-principal --username $AZURE_SP_APP_ID --
password $AZURE_SP_PASSWORD --tenant $AZURE_SP_TENANT --output
none'

        sh "az aks get-credentials --resource-group $NAME_RS --name
$NAME_AKS"

        sh "kubectl apply -f $ASK_DEV_FILE"

    }

}

}

stage('Clean') {

steps {

echo "Clean"

script {

sh 'docker compose down --volumes --rmi all'

}

}

}

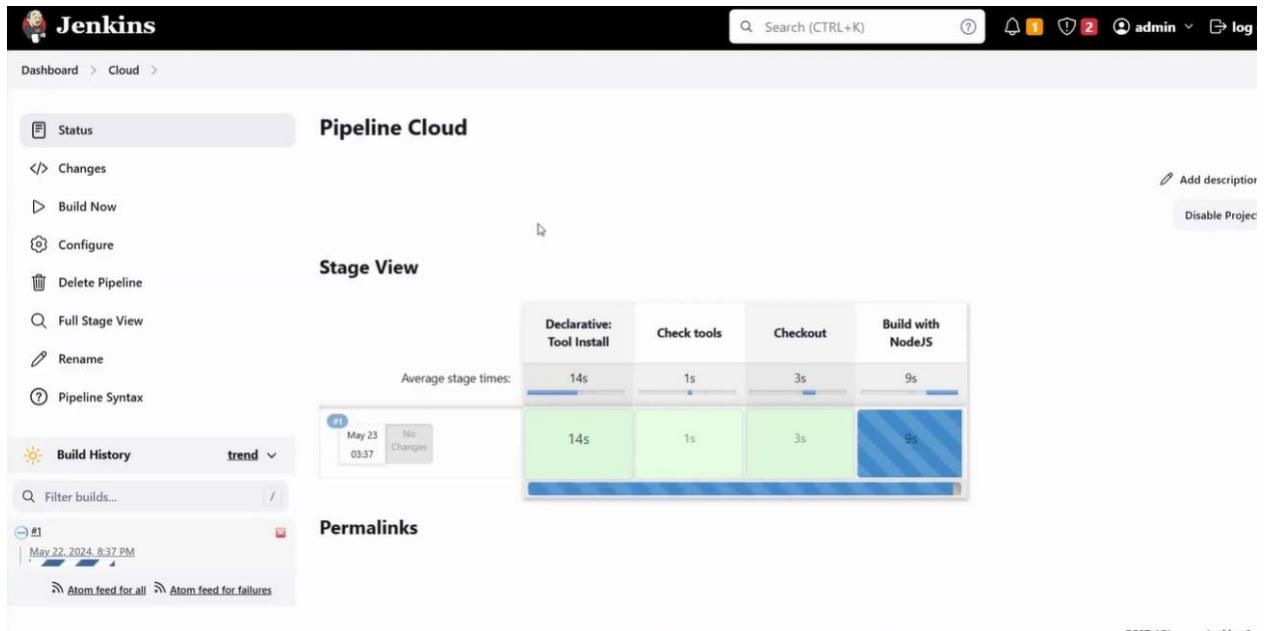
}

}

```

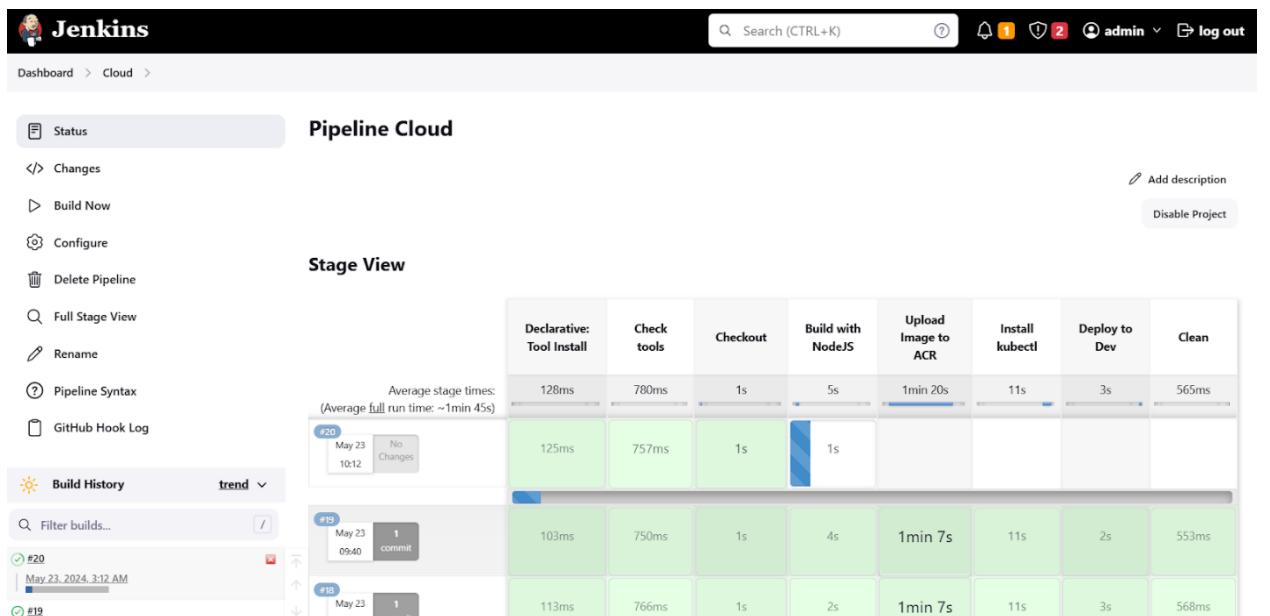
Bước 7: Thực hiện kiểm thử và khởi chạy Pipeline.

Ta chọn **Build Now** để kiểm tra **Pipeline**.



Hình 4.58: Kiểm tra pipeline

Sau khi chạy thử thành công ta có thẻ Commit trên dự án GitHub để chạy thử Pipeline nếu đã thành công .



Hình 4.59: Kiểm tra pipeline sau khi commit code

Bước 8 : Lấy địa chỉ website.

Ở trên Azure ta sử dụng **powershell** và chạy lần lượt các lệnh sau để lấy địa chỉ website . **-az aks get-credentials --resource-group BTL --name BTLKS01**

-kubectl get pod

-kubectl get svc

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with links like 'Create a resource', 'Virtual machines', 'Subscriptions', 'Kubernetes services', 'Container registries', 'Education', 'Resource groups', 'Azure DevOps organizations', 'All resources', and 'More services'. Below this is a 'Resources' section with tabs for 'Recent' and 'Favorite'. It lists two resources: 'VMBTL' (Virtual machine) and 'grafana-20244322955-ea' (Azure Managed Grafana). The main area is a terminal window titled 'Switch to Bash' showing the output of several kubectl commands:

```
OTD: SqlServer has been updated to Version 22!
ERBOSE: Authenticating to Azure ...
ERBOSE: Building your Azure drive ...
S /home/azureuser> az aks get-credentials --resource-group BTL --name BTLKS01
erged "BTLKS01" as current context in /home/azureuser/.kube/config
S /home/azureuser> kubectl get pod
NAME                  READY   STATUS    RESTARTS   AGE
estjs-backend-development-bd4df9587-tzv59  1/1     Running   0          4h53m
S /home/azureuser> kubectl get svc
NAME         TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
ubernetes   ClusterIP  10.0.0.1    <none>        443/TCP   14h
estjs-backend-dévelopment   LoadBalancer  10.0.253.48  4.191.77.230  80:30665/TCP  12h
S /home/azureuser>
```

Hình 4.60: Lấy địa chỉ website dự án

Ta đã lấy được địa chỉ website dự án : <http://4.191.77.230/>

Bước 9: Sử dụng máy ảo được tạo trên azure và chạy docker chứa database của website.

Mở port 3306 để có thể lấy database chung port với docker.

The screenshot shows the Microsoft Azure portal interface for a virtual machine named VM-BTL. The left sidebar has a 'Networking' section with 'Network settings' selected. The main area is titled 'Network security group VM-BTLnsg168' and shows a table of port rules:

Priority ↑	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
320	HTTP	80	TCP	Any	Any	Allow
340	HTTPS	443	TCP	Any	Any	Allow
350	AllowAnyCustom8080Inbound	8080	Any	Any	Any	Allow
360	AllowAnyCustom3306Inbound	3306	Any	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Hình 4.61: Mở port 3306

Tạo thư mục chứa docker-compose.

The terminal window shows the following commands being run:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

root@VM-BTL:~# cd MYSQL01
-bash: cd: too many arguments
root@VM-BTL:~# mkdir MYSQL01
root@VM-BTL:~# cd MYSQL01
root@VM-BTL:~/MYSQL01# touch docker-compose.yaml
root@VM-BTL:~/MYSQL01# █

```

Hình 4.62: Tạo docker compose trên terminal

Viết file docker với thông tin của database.

```

[1/2]
version: '3.8'

services:
  mysql:
    image: mysql:latest
    container_name: mysql_container
    environment:
      MYSQL_ROOT_PASSWORD: Thao@112
      MYSQL_DATABASE: vexere_db
    ports:
      - "3306:3306"
    volumes:
      - mysql_data:/var/lib/mysql

volumes:
  mysql_data:

```

Hình 4.63: Viết file docker

Đóng gói và chạy file docker.

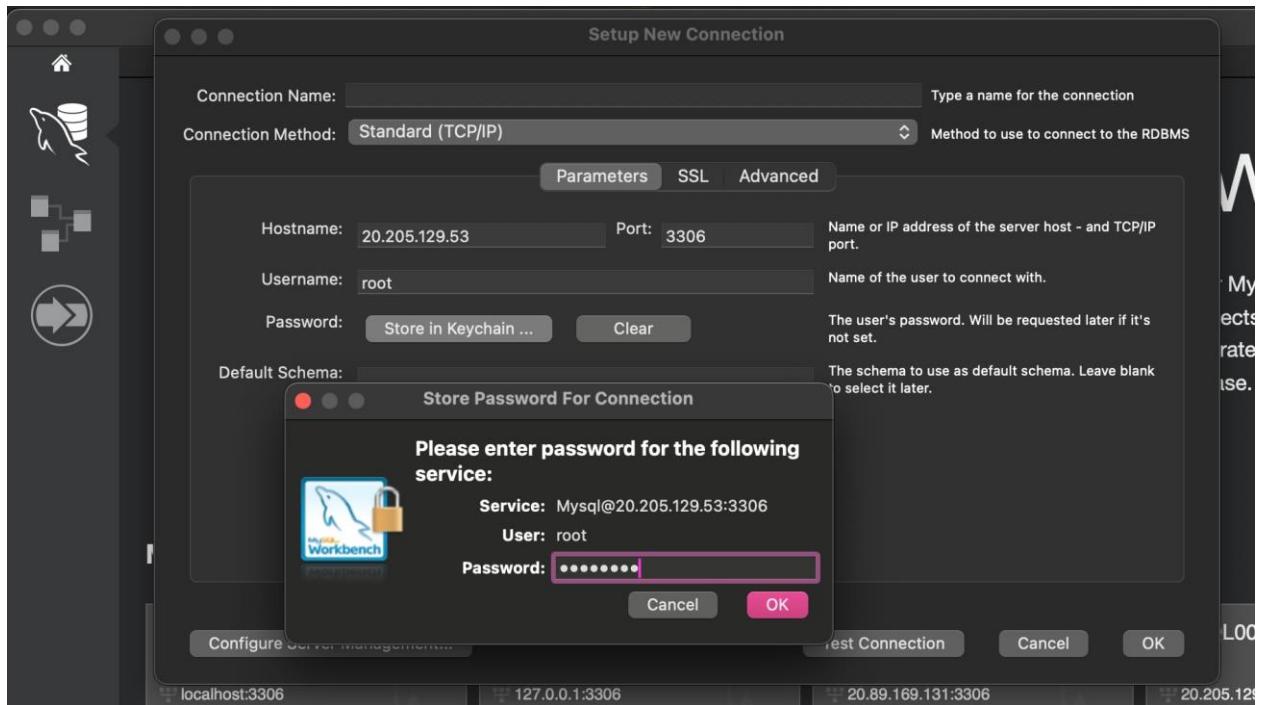
```

root@VMCTL:~/MySQL01# nano docker-compose.yaml
root@VMCTL:~/MySQL01# docker-compose up -d
Creating volume "mysql01_mysql_data" with default driver
Creating mysql_container@1 ... done
root@VMCTL:~/MySQL01# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
874d74ab122 mysql:latest "docker-entrypoint.s..." 13 seconds ago Up 8 seconds 3306/tcp, 33060/tcp, 0.0.0.0:3307->3307/tcp, :::3307->3307/tcp NA
a0f8ade3b25a mysql:latest "docker-entrypoint.s..." 13 hours ago Up 13 hours 0.0.0.0:3306->3306/tcp, 33060/tcp my
sql_container
bd61c7bb75 khaliddinh/jenkins "/usr/bin/tini -- /u..." 13 hours ago Up 13 hours 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp Je
nkinsContainer
root@VMCTL:~/MySQL01#

```

Hình 4.64: Đóng gói và chạy file docker

Vào MySQL Workbench để tạo kết nối tới database dựa trên IP của máy ảo.



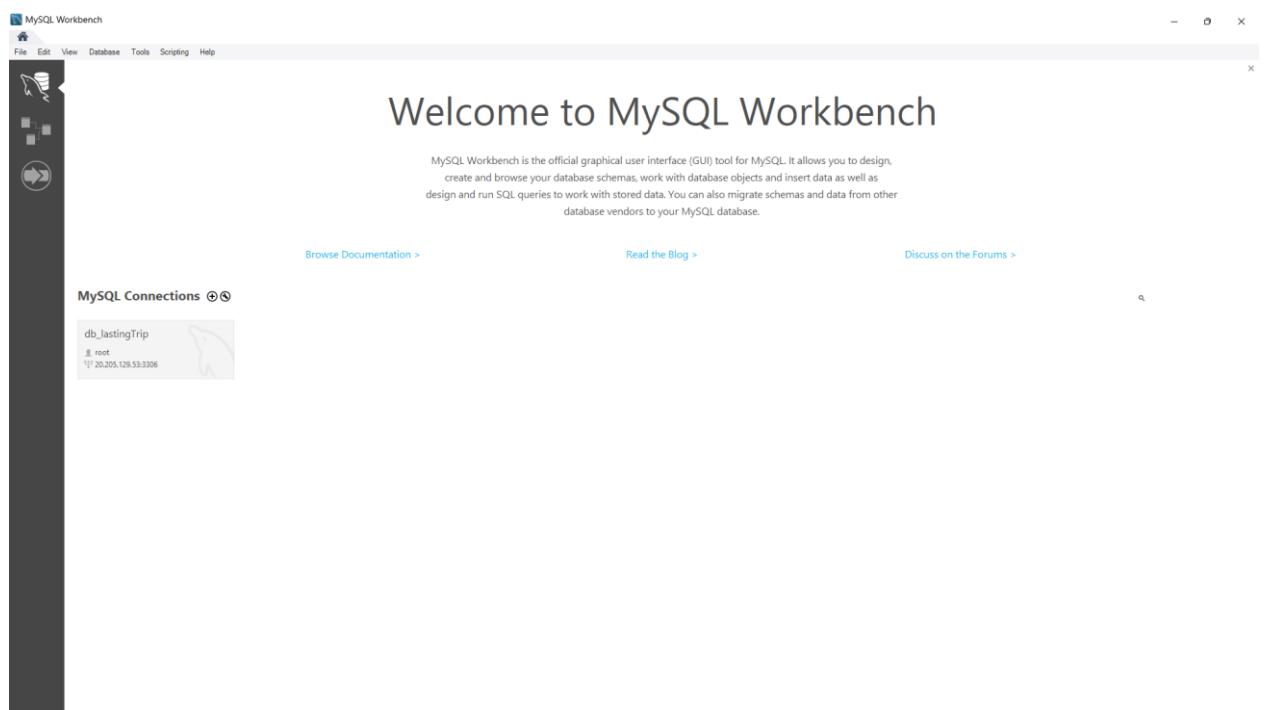
Hình 4.65: Tạo kết nối với database dựa trên IP máy ảo

Kết nối tới database đã được khởi tạo. Có thể vào MySQL Workbench để kiểm tra và chỉnh sửa database.

ID	Name	Email	Password	NumberPhone	BirthDate	Gender	Type	CCCD	Address	CreatedAt	UpdatedAt	
1	John Doe	john.doe@example.com	p4OCH76D%	(65)1386-7441	2004-12-03 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
2	Jane Smith	jane.smith@example.com	oKgRQGzFQ	+1-800-416-9932#x701	1936-06-05 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
3	Alan Scott	weberfelicia@example.net	%p0OH76D%	(65)1386-7441	2004-12-03 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
4	Patrick Serrano	teresa20@example.net	%_0pRQGzFQ	@b2VhU%9x0	2001-09-07-1649#x143	1946-02-17 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18
5	James Powell	cantujesicu@example.com	%b2VhU%9x0	(436)366-5012#x1223	2006-04-03 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
6	Todd Doyle	jacksonleslie@example.net	BxVbNfFu_f	g4ZnFOAb0x0	941-787-6142#x27369	1957-02-18 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18
7	Jason Padilla	edward77@example.org	%kwgGPAY@7	+1-387-906-6384#x98829	1985-10-13 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
8	Joshua Cox	timothybruce@example.net	%kwgGPAY@7	+1-387-906-6384#x98829	1985-10-13 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
9	Harold Gonzalez	imullins@example.com	\$7wJb1y8	(263)785-2213	1993-06-10 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
10	Daniel Williams	john.carter@example.org	4L7QOEzIV	621-422-8807#x3049	1960-12-24 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
11	Shane Sheppard	brittanycastillo@example...	7EBPGH4W+	245-891-8208#x74866	1994-07-11 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
12	Justin Manning	pgonzales@example.com	F0C9B0A8%	202-555-1234#x5003	1983-03-15 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
13	Jessica Hayes	jenemay.york@example.com	7E0K7_0	224-954-8512#x2223	1957-01-01 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
14	Kenneth Clegg	christopher.c...	RPhJbRWj6	16-503-4536#x582	1936-01-05 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
15	Tony Escobar	fletcherdennis@example.c...	5jKCkWv7%2	+1-759-331-5138#x26500	1958-10-21 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
16	Madison Fuentes	tbarney@example.net	7eNB0EWR@	(251)245-2927#x5440	2002-11-10 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
17	Larry Hernandez	jonathan62@example.com	u5J3nNVWei	807-338-6812	1986-05-06 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
18	Deborah Diaz	tammyjhinson@example...	MJB8Qw70	908.801.0787	1982-12-19 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
19	Ashley Cole	hsharp@example.org	Oq3vCaqu/u%	001-643-398-0509#x18484	2000-12-17 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
20	Michelle Henson	carrollamy@example.com	B8nFOFn_3	4704747434	1973-08-24 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
21	Katherine Lawrence	darellmalone@example.net	17BpLnV%	001-461-964-8887#x944	1993-10-11 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
22	Robert Park	hannah07@example.org	okYh7Wfa!	5412212325	1950-12-24 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
23	Paul Sanders	freemanelizabeth@example...	HoK3Qx7%	967-947-5242	1969-04-25 00:00:00	1	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
24	Charles Hebert	ericalhall@example.com	Z4MSVQJ9	001-863-965-2821#x35476	1974-10-05 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	
25	Lee Smith	kenneth hernandez@example...	yuXctf!Qd+	463-508-9943#x15553	1997-06-20 00:00:00	0	owner	HNL	HNL	2024-04-01 04:26:18	2024-04-01 04:26:18	

Hình 4.66: Kết nối với database trên MySQL Workbench

Có thể kết nối với database từ máy khác.

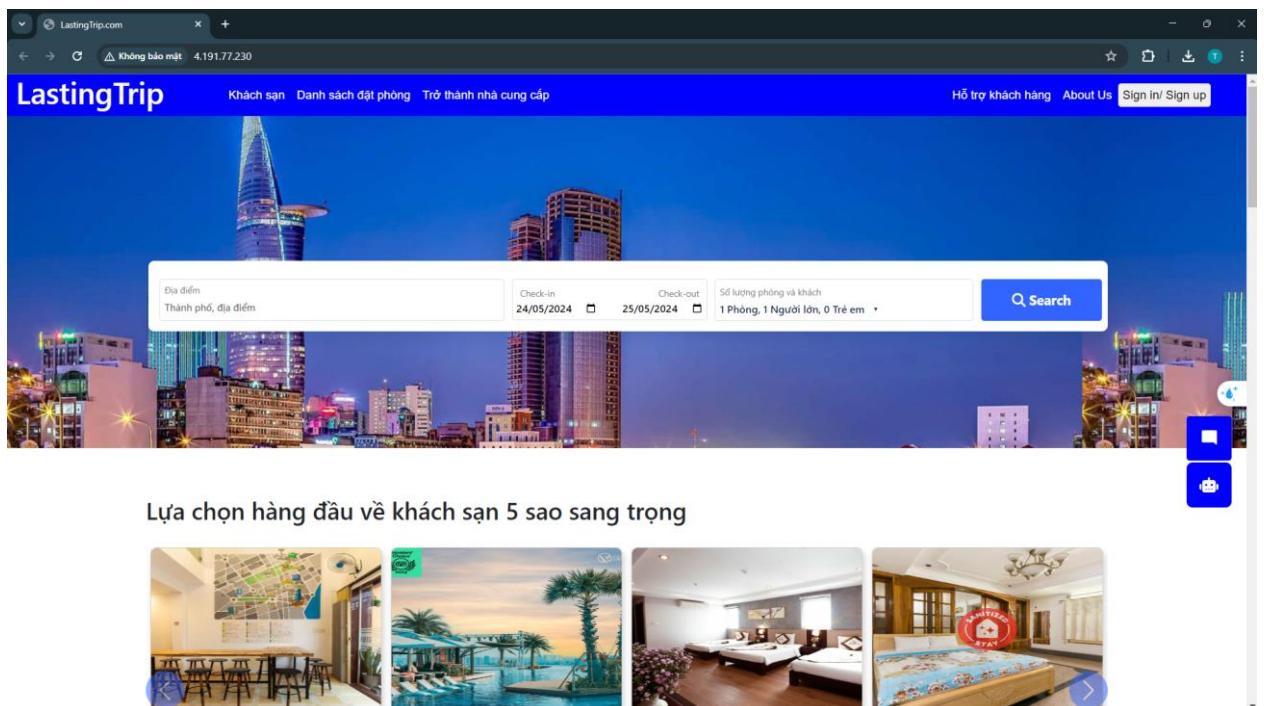


Hình 4.67: MySQL Connections

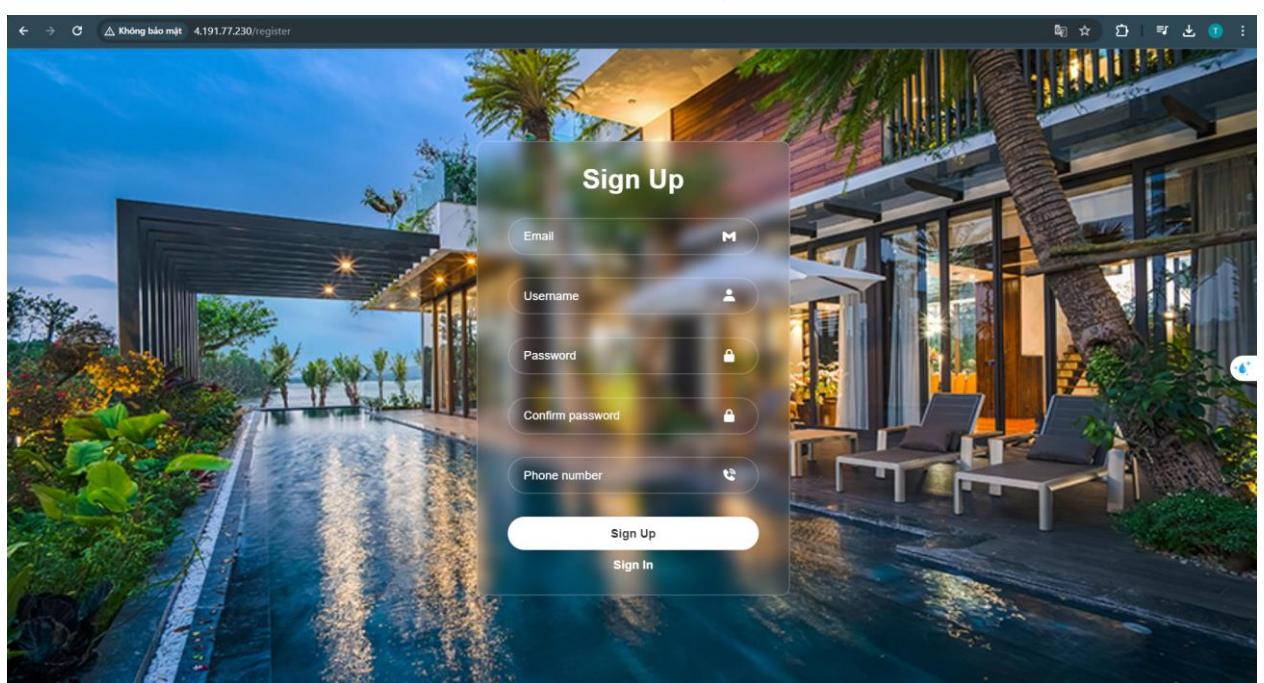
This screenshot shows a more detailed view of MySQL Workbench. The top navigation bar includes "File", "Edit", "View", "Query", "Database", "Server", "Tools", "Scripting", and "Help". The "Query" tab is selected. The "Navigator" pane on the left shows the schema "vexere_db" with its tables: Amenities, Bookings, Countries, HotelAmenities, Hotels, Reviews, RoomServices, Sequelizemeta, UriTranslations, and Users. The "Information" pane below it shows the "Table: Users" with its columns: id, name, email, password, numberPhone, birthDate, gender, type, cccid, address, createdAt, and updatedAt. The main area displays the results of a query: "SELECT * FROM vexere_db.Users;". The results grid shows 35 rows of data, each representing a user with their name, email, password, phone number, birth date, gender, type, CCCD, address, and creation and update dates. The "Result Grid" toolbar at the top of the results area includes icons for "Edit", "Copy", "Print", "Export/Import", and "Wrap Cell Content". The status bar at the bottom right indicates "Duration / Fetch 0.047 sec / 0.000 sec".

Hình 4.68: Kết nối database từ máy khác

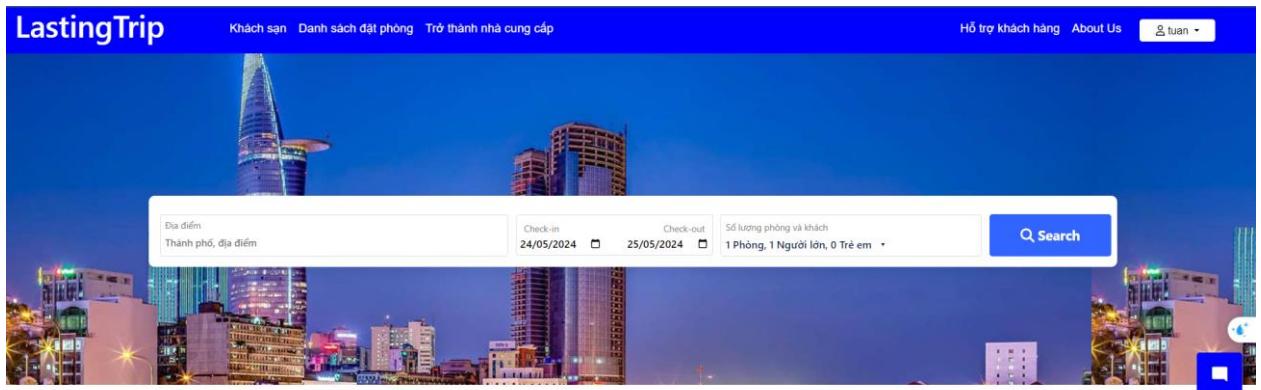
Website đã được deploy thành công lên ip 4.191.77.230



Hình 4.69: Giao diện trang chủ



Hình 4.70: Giao diện đăng nhập



Lựa chọn hàng đầu về khách sạn 5 sao sang trọng



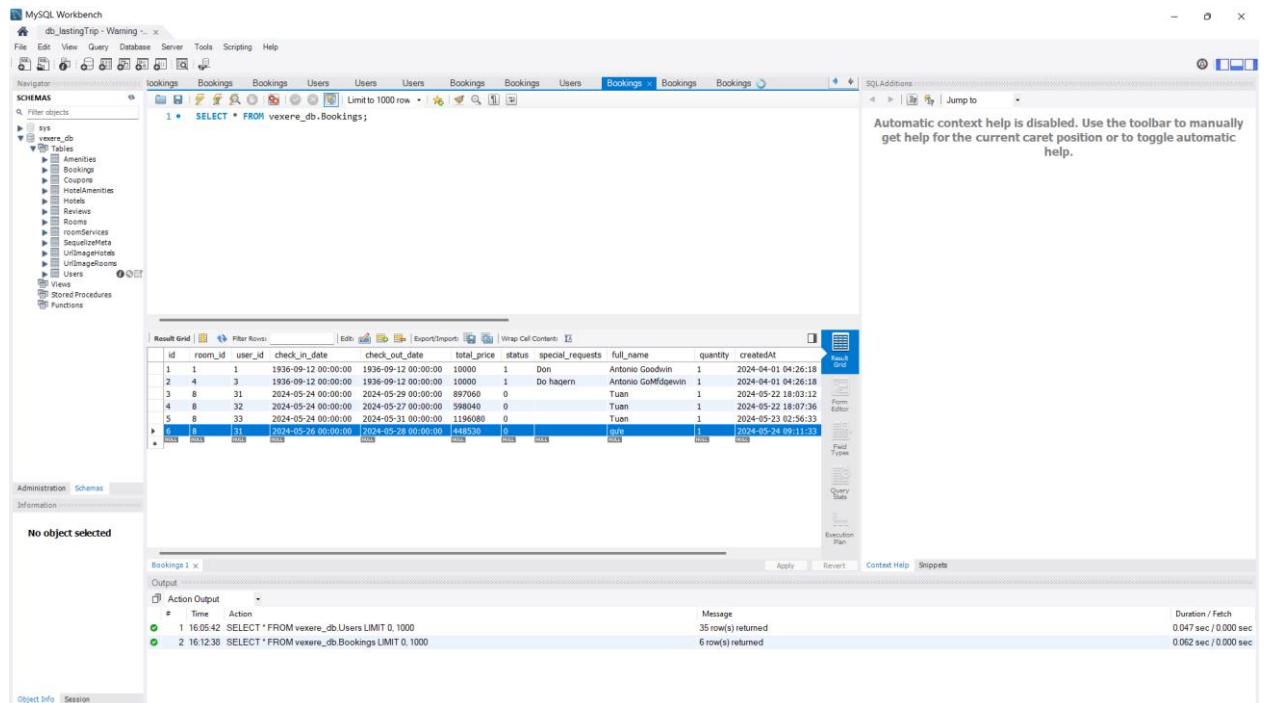
Hình 4.71: Giao diện sau khi đăng nhập



Thanh toán

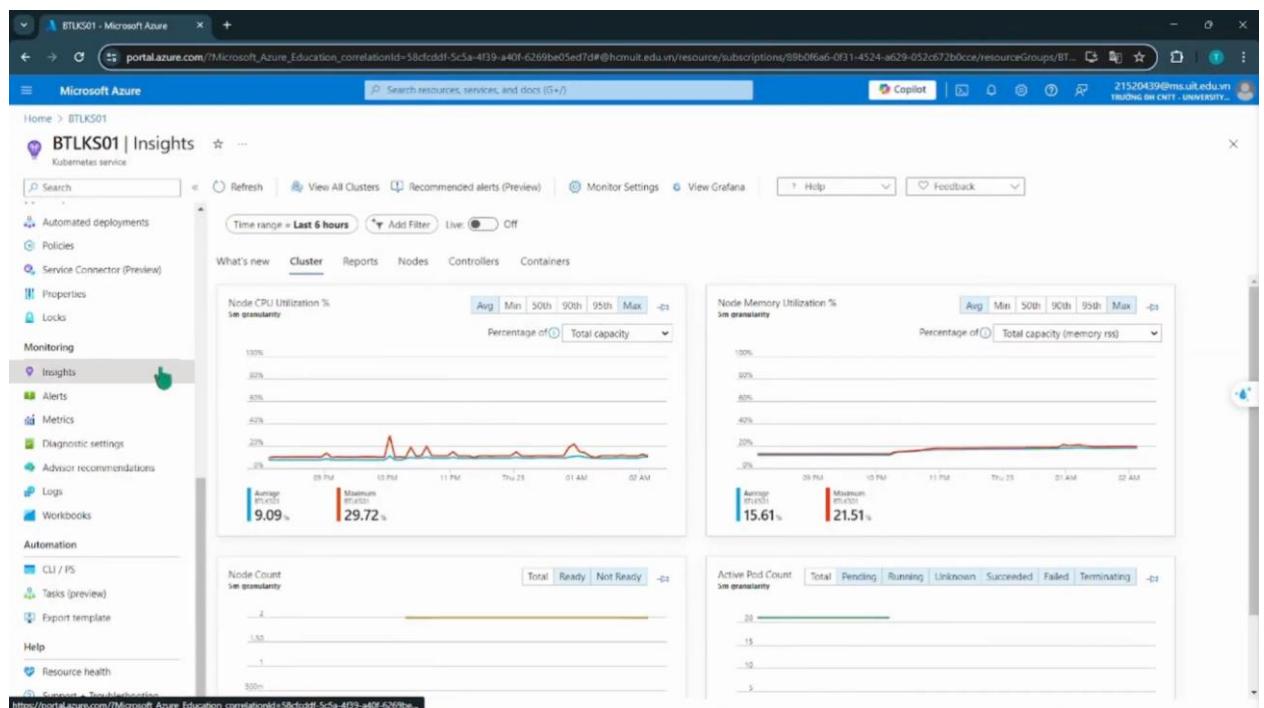
Họ tên *	que	Đơn hàng của bạn
Năm sinh*	123	Holiday Inn & Suites Saigon Airport 149,510 VND
Căn cước công dân *	123	Số lượng phòng
Địa chỉ	123	Tổng cộng 149,510 VND
Số điện thoại*	0342730606	<input type="radio"/> Thanh toán qua ngân hàng
Email Address *	Ka Ka Supper02@gmail.com	<input checked="" type="radio"/> Thanh toán trực tiếp
Yêu cầu đặc biệt		Place Order

Hình 4.72: Tiến hành đặt phòng



Hình 4.73: Đơn hàng đã được lưu vào database

Bước 10: Tại tab Insights của tab Monitoring trong AKS (Azure Kubernetes Service) có thể xem lại quá trình thực thi thời gian CPU.



Hình 4.74: Xem lại quá trình thực thi CPU ở Insights (Monitoring) (1)

prometheus-collector | Overview

Container

View in Log Analytics

Overview Live Logs Live Events

Container Name	Container ID
prometheus-collector	95bf57afe2586fbab85524c7a18cef91...
Namespace	Container Status
kube-system	running
Container Status Reason	Image
-	azuremonitor/containerinsights/ciprod/...
Image Tag	Container Creation Time Stamp
6.8.10-main-05-06-2024-079dabb	22:16:21 22/5/2024
Start Time	Finish Time
-	-
CPU Limit	CPU Request
200 m	50 m
Memory Limit	Memory Request
1 GB	150 MB
Last reported	-
1 min ago	-
Environment Variables	AZMON_COLLECT_ENV=FALSE

Hình 4.75: Xem lại quá trình thực thi của CPU ở Insights (Monitoring) (2)

Có thể vào View grafana để xem chi tiết các file.

grafana-2024322955-ea - Microsoft Edge

Dashboard - Grafana

grafana-2024322955-ea-a2bacscqabd5c6hdhk.grafana.azure.com/dashboards

Home > Dashboards

Dashboards

Create and manage dashboards to visualize your data

New

Search or jump to... Ctrl+K

Search for dashboards and folders

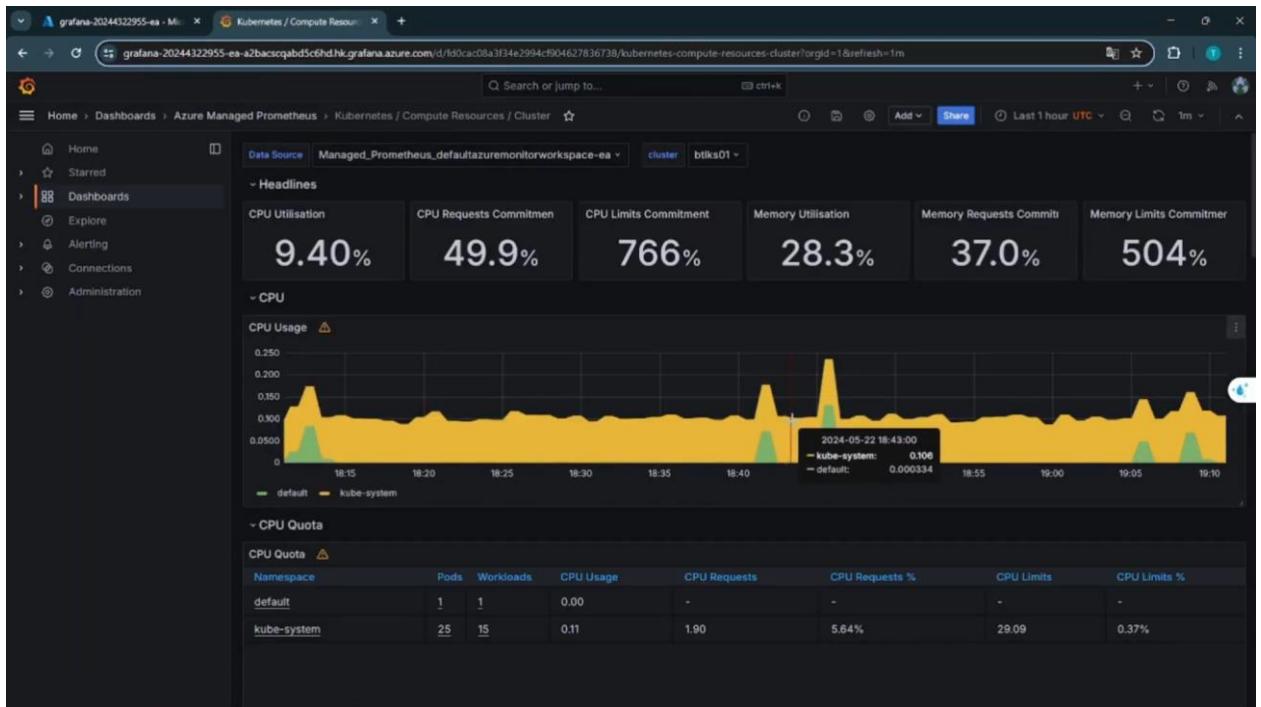
Filter by tag

Starred

Name Tags

- Azure Managed Prometheus
- Azure Monitor
- Microsoft Defender for Cloud

Hình 4.76: View Grafana



Hình 4.77: Xem chi tiết hoạt động của CPU tại view Grafana

Chương 5. TỔNG KẾT

5.1. Kết quả đạt được

- Hiểu được vấn đề và cách giải quyết bài toán.
- Tích hợp được nhiều công cụ khác nhau.
- Website đã được deploy và có thể sử dụng rộng rãi.
- Các công cụ hoạt động mượt mà, đúng logic, đúng lộ trình.

5.2. Ưu điểm

Tính linh hoạt: việc sử dụng cloud computing mang lại sự linh hoạt cao, cho phép điều chỉnh dự án web theo yêu cầu cụ thể của từng doanh nghiệp. Điều này giúp các nhà quản lý dự án dễ dàng thay đổi kế hoạch và tiến độ để phù hợp với các biến động.

Khả năng mở rộng: Cloud computing cho phép mở rộng hoặc thu hẹp quy mô dự án web một cách dễ dàng, đáp ứng nhanh chóng các nhu cầu thay đổi của người dùng mà không cần đầu tư thêm vào cơ sở hạ tầng.

Tối ưu chi phí: sử dụng dịch vụ đám mây giúp giảm thiểu chi phí vận hành và bảo trì. Các doanh nghiệp chỉ phải chi trả cho những tài nguyên thực sự sử dụng, giúp tiết kiệm chi phí đáng kể.

Bảo mật cao: Cloud computing cung cấp các giải pháp bảo mật mạnh mẽ, bảo vệ dữ liệu và ứng dụng web khỏi các mối đe dọa an ninh mạng. Điều này đảm bảo rằng dự án luôn an toàn và bảo mật.

Tính sẵn sàng cao: các dịch vụ cloud đảm bảo tính sẵn sàng cao, giúp dự án web hoạt động liên tục ngay cả khi xảy ra sự cố. Điều này giúp giảm thiểu rủi ro gián đoạn kinh doanh và đảm bảo trải nghiệm người dùng không bị ảnh hưởng.

5.3. Nhược điểm

Yêu cầu cập nhật kiến thức: Cloud computing phát triển nhanh chóng, yêu cầu các nhà quản lý dự án phải liên tục cập nhật kiến thức và kỹ năng mới để bắt kịp xu hướng công nghệ.

Chi phí ban đầu: mặc dù chi phí vận hành giảm, việc đầu tư ban đầu vào cloud computing có thể cao nếu cần thêm các tài nguyên như nhân lực có kỹ năng chuyên môn và cơ sở hạ tầng hỗ trợ.

Rủi ro tiềm ẩn: Cloud computing có thể mang lại các rủi ro mới như bảo mật dữ liệu và tính sẵn sàng của dịch vụ. Các nhà quản lý dự án cần có chiến lược quản lý rủi ro hiệu quả để đối phó với các thách thức này.

5.4. Hướng phát triển

Tăng cường tự động hóa: đẩy mạnh việc tự động hóa quy trình triển khai và quản lý ứng dụng để tăng hiệu quả và giảm thiểu lỗi do con người.

Cải thiện bảo mật và tính sẵn sàng: liên tục cải tiến các biện pháp bảo mật và tăng cường tính sẵn sàng của hệ thống để đảm bảo dự án web luôn hoạt động ổn định và an toàn.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] R. Awati, "Azure Kubernetes Service (AKS)," 26 5 2024. [Online]. Available: <https://www.techtarget.com/searchcloudcomputing/definition/Azure-Kubernetes-Service-AKS>.
- [2] GESO, "Software As A Service (SaaS) là gì? Ứng dụng trong doanh nghiệp," 06 04 2023. [Trực tuyến]. Available: <https://geso.us/software-as-a-service-saas-la-gi>. [Đã truy cập 27 05 2024].
- [3] V. N. Thoai, "PaaS là gì? Mọi thứ cần biết về mô hình Platform as a service," 18 03 2023. [Trực tuyến]. Available: <https://mona.media/paas-la-gi/>. [Đã truy cập 28 05 2024].
- [4] I. Lee, "what-is-saas," [Online]. Available: <https://www.wallarm.com/what/what-is-saas>. [Accessed 29 04 2024].
- [5] "Azure Kubernetes Service: Features, Architecture & Getting Started," [Online]. Available: <https://spot.io/resources/azure-kubernetes-service/azure-kubernetes-service-features-architecture-getting-started/>. [Accessed 20 05 2024].
- [6] "CI/CD using Jenkins on Azure Container Service (AKS)," Microsoft Learn, [Online]. Available: <https://learn.microsoft.com/en-us/samples/azure/azure-quickstart-templates/jenkins-cicd-container/>. [Accessed 27 05 2024].
- [7] TomArcherMsft, duffney, v-thepe, "Azure and Jenkins," 02 12 2021. [Online]. Available: <https://learn.microsoft.com/en-us/azure/developer/jenkins/overview>. [Accessed 29 05 2024].
- [8] Mike Jacobs, Delora Bradish, Julia Kulla-Mader, "Azure Pipelines architecture for Azure Web Apps," 27 07 2023. [Online]. Available: <https://learn.microsoft.com/en-us/azure/devops/pipelines/architectures/devops->

[pipelines-azure-web-apps-architecture?view=azure-devops](#). [Accessed 27 05 2024].

- [9] D. School, "What is an Azure Monitor?," [Online]. Available: <https://www.devopsschool.com/blog/what-is-an-azure-monitor/>. [Accessed 21 04 2024].
- [10] Microsoft, "Managed Grafana overview," [Online]. Available: <https://azure.microsoft.com/en-us/products/managed-grafana#overview>. [Accessed 22 05 2024].