viettel
security

Viettel Endpoint Detection & Response (VCS-aJiant)

Phiên bản 3.3.0 EDR – Năm 2021

Ngày cập nhật: 31/12/2021

Tài liệu Hướng dẫn Triển khai Cài đặt (Installation Guide)

Mục lục

Thuật ngữ

Viết tắt	Diễn giải	Ghi chú
VCS	Viettel Cyber Security	Công ty An ninh mạng Viettel
VCS-aJiant	Tên giải pháp Endpoint Detection & Response do Công ty An ninh mạng viettel phát triển	
EDR	Endpoint Detection & Response	Tên 1 dòng sản phẩm giám sát phát hiện và phản ứng với các bất thường phía Endpoint
HA	High Availability	Tính sẵn sàng cao

1. GIỚI THIỆU

1.1. Thực trạng hiện nay

Ngày nay, các tổ chức, doanh nghiệp tiếp tục gặp rất nhiều khó khăn với việc phát hiện, xác định, điều tra và giảm thiểu các dạng phần mềm độc hại tiên tiến trong hệ thống. Các công nghệ phòng chống mã độc truyền thống như Anti Virus dựa trên chữ ký đang bị vượt qua một cách cố ý bởi những kẻ tấn công chuyên nghiệp có trình độ cao với các bộ công cụ tấn công, phần mềm độc hại được tùy chỉnh và hướng mục tiêu cụ thể. Nhiều tổ chức đã thừa nhận rằng các phương pháp phòng thủ chống phần mềm độc hại truyền thống của họ đã thất bại và một chiến lược mới phải được tạo ra để xác định những vi phạm này tại endpoint. Một số lượng đáng kể các vi phạm dữ liệu gần đây từ các dạng phần mềm độc hại nâng cao đã làm tăng sự quan tâm của khách hàng đối với các Giải pháp phát hiện và phản ứng cho lớp endpoint (EDR) mà VCS-aJiant là một trong số đó.

1.2. Sự phát triển của công nghệ

Công nghệ của Giải pháp VCS-aJiant giúp bù đắp các thiếu sót của các công nghệ dựa trên chữ ký mà các tổ chức đang sử dụng như Anti Virus hay IPS/IDS để cung cấp khả năng phát hiện bất thường dựa trên hành vi và cho cái nhìn sâu hơn về các thông tin cụ thể có liên quan trên endpoint để phát hiện và giảm thiểu các mối đe dọa nâng cao.

1.3. VCS-aJiant

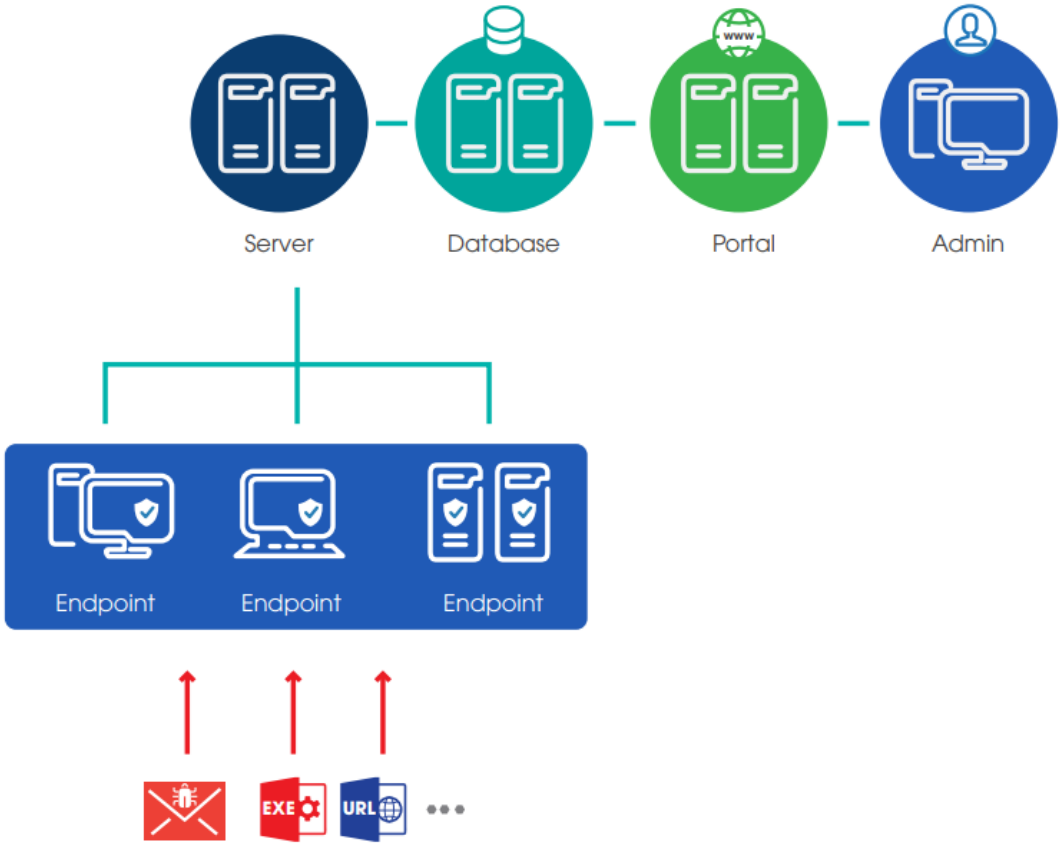
VCS-aJiant có khả năng cung cấp thông tin chi tiết về việc lây nhiễm phần mềm độc hại và các hành vi mở rộng phạm vi tấn công (lateral movement) của những kẻ tấn công khi chúng thực hiện việc dò quét hoặc sử dụng thông tin bị đánh cắp trong mạng nội bộ đối với các hệ thống và ứng dụng.

Ngoài ra, VCS-aJiant cũng bổ sung cho các công nghệ bảo mật hiện có như giải pháp quản lý sự kiện và thông tin bảo mật (SIEM), các công cụ giám định mạng (Network Forensics) và các thiết bị phòng chống mối đe dọa tiên tiến (Advanced Threat Detection), đồng nghĩa là bổ sung vào danh mục các giải pháp phản ứng sự cố an toàn thông tin của tổ chức.

2. TỔNG QUAN

2.1. Kiến trúc hạ tầng

Mô hình triển khai VCS-aJiant như sau:



Các thành phần của hệ thống bao gồm:

- **Endpoint**: Là thành phần được cài đặt trên từng máy tính, có nhiệm vụ giám sát các dấu hiệu bất thường trên máy tính, gửi log về server tập trung. Các thông tin giám sát bao gồm các hành vi liên quan đến File, Process, Memory, Registry, Network trên máy tính người dùng và server.
- **Cụm server xử lý tập trung và lưu trữ**: Là thành phần xử lý dữ liệu do Endpoint gửi về, đóng vai trò chính trong việc phân tích và xử lý dữ liệu theo thời gian thực.
- **Thành phần Web Portal**: Là thành phần mà người quản trị sẽ sử dụng để điều tra, giám sát và phân tích các thông tin của hệ thống, phản ứng khi có cảnh báo bất thường tại các agents

2.2. Công nghệ được sử dụng

VCS-aJiant sử dụng cộng nghệ Filter Driver (cho phép chạy và theo dõi ở mức Kernel-based) thu thập các thông tin bao gồm File, Process, Registry, Network trên máy tính người dùng và server. Các dấu hiệu về file bao gồm (modified, delete, changed attribute), về registry (delete key/value, set value, rename key/value, create key với access nghi ngờ. Các dấu hiệu nghi ngờ về Memory được định kì quét rà soát liên tục. Các hành vi được xác định là nghi ngờ được đẩy về hệ thống Back-end phân tích tập trung.

Luồng nghiệp vụ điều tra tấn công được thiết kế khép kín theo kịch bản incident response (IR Flow), hỗ trợ phát hiện và phân tích các dấu hiện bất thường ngay trên một giao diện duy nhất. Cung cấp các chức năng điều tra (Forensic) sâu trên Endpoint. Hỗ trợ lấy file nghi ngờ (Get Artifact), đẩy công cụ rà quét (Tool Deployment), cho phép thực hiện điều tra, cung cấp bằng chứng theo thời gian thực (Process Analysis, Live Response), cho phép thực hiện phản ứng khi phát hiện mối đe dọa.

Ngay khi xác minh được bất thường, Endpoint cung cấp các công cụ gỡ bỏ mã độc trên diện rộng (Response Scenario) bao gồm: cô lập mạng máy bị nhiễm (network containment), kill process, delete file/registry.

3. HƯỚNG DẪN CÀI ĐẶT BACKEND

3.1. Điều kiện đảm bảo cài đặt

- Cấu hình IP:
  - Mô hình AllInOne: cấu hình 1 IP tĩnh cho node server.
  - Mô hình MultiNode: cần 3 IP tĩnh (1 public virtual ip cho kết nối từ agents đến server, 2 ip cho loadbalancer)
- Mở kết nối đến server qua các port sau:

Mục đích	**Port**	**Protocol**
Agent kết nối server	4443, 5672, 8443, 8888	TCP
Truy cập Portal	80, 443	TCP

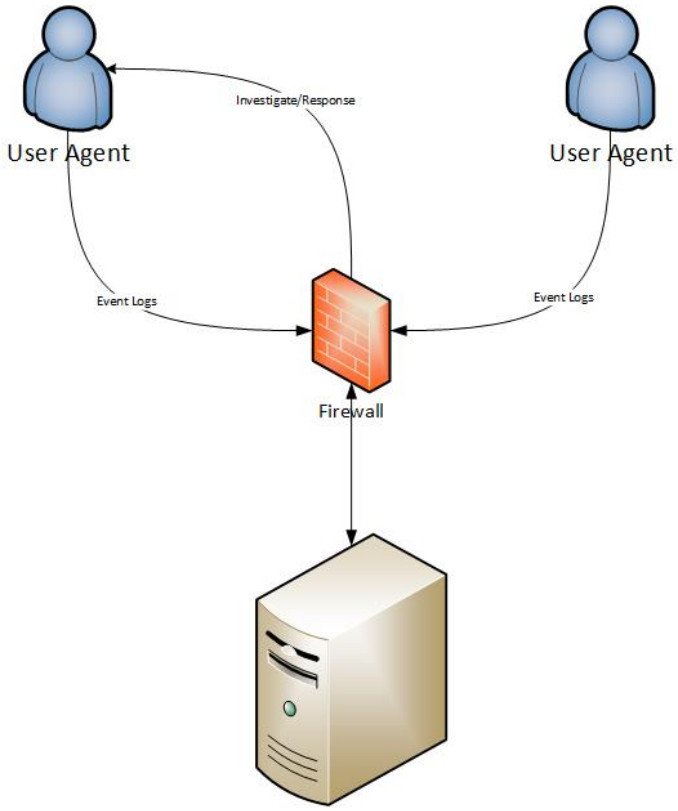
- Server cần bật tính năng ảo hoá Intel VT-x/AMD-V để chạy máy ảo build bộ cài agent Windows.

3.2. Topo của khách hàng

Tùy theo quy mô khách hàng triển khai, có thể lựa chọn triển khai hệ thống backend của VCS-aJiant theo 2 mô hình AllInOne hoặc MultiNode.

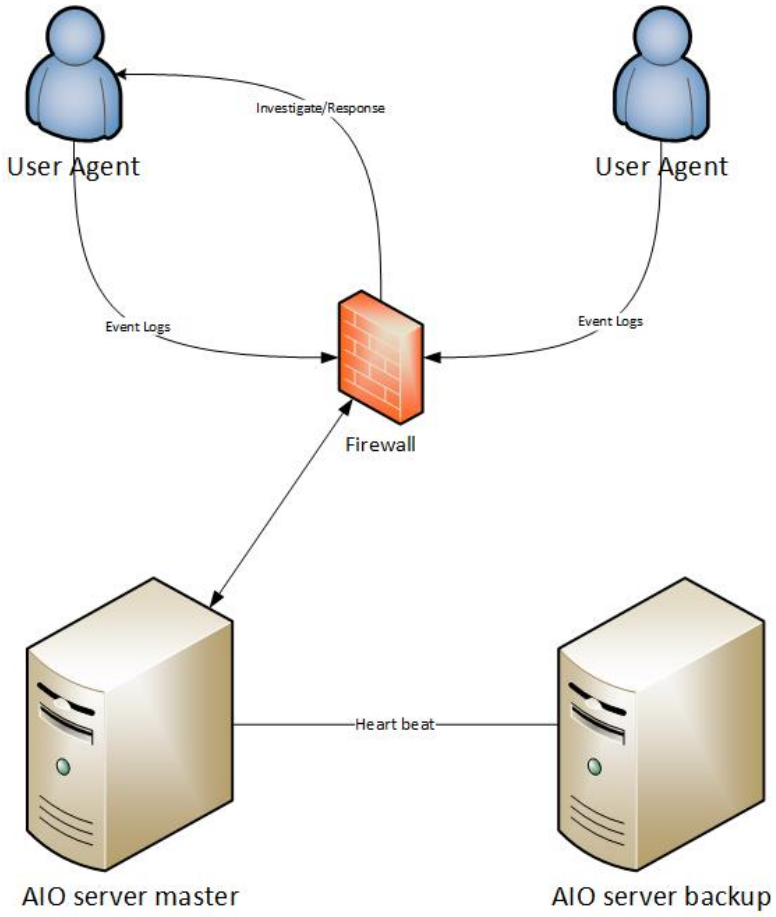
3.2.1. AllInOne Topo

Với mô hình triển khai AllInOne, tất cả các dịch vụ backend VCS-aJiant được triển khai trên 1 node server duy nhất. Mô hình này phù hợp khi triển khai với số lượng khách hàng nhỏ (<3000 agent).



Hình 1. Mô hình triển khai backend VCS-aJiant AllInOne

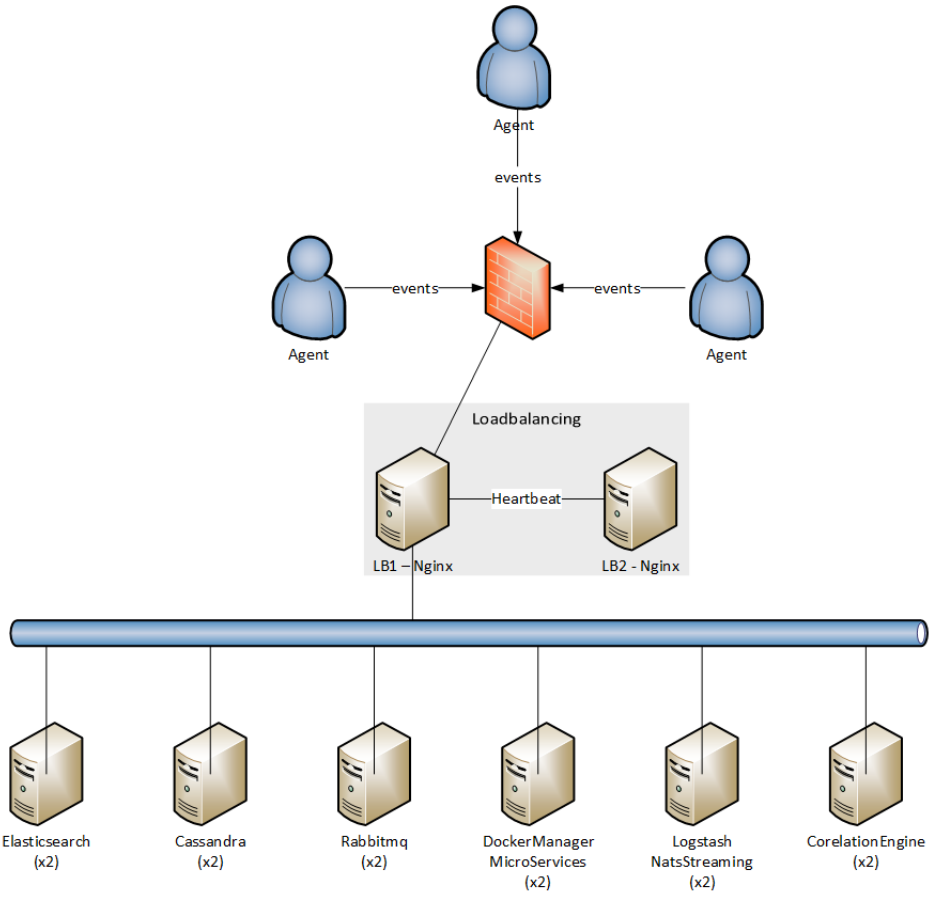
Với các khách hàng có số agent nhỏ (<3000) nhưng vẫn yêu cầu có HA, thì có thể triển khai mô hình AllInOne có HA. Mô hình này gồm 2 node AllInOne, trong đó có 1 node chạy chính và 1 node dự phòng. Database của 2 node được đồng bộ với nhau.



Hình 2. Mô hình triển khai VCS-aJiant AllInOne có HA

3.2.2. MultiNode Topo

Với mô hình backend MultiNode, các dịch vụ được chạy trên nhiều node máy ảo khác nhau giữa các server vật lý. Mô hình Multinode hỗ trợ cân bằng tải, HA cho các dịch vụ và HA cho dữ liệu.

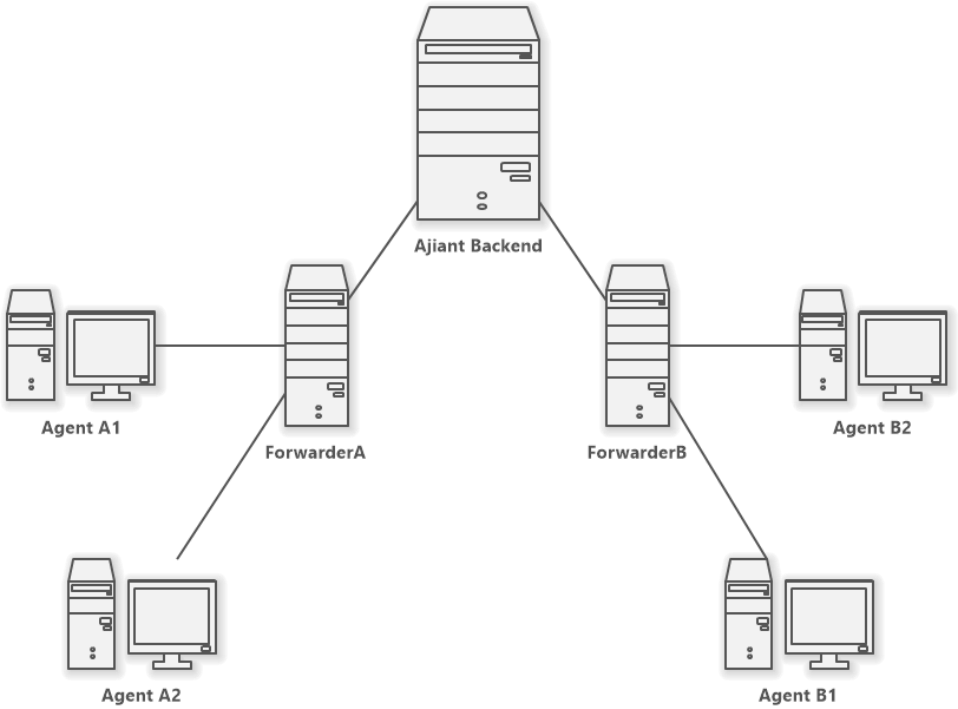


Hình 3. Mô hình triển khai backend MultiNode

3.2.3. MSSP Topo

Trong mô hình MSSP (Managed Security Service Provider) bao gồm 3 thành phần chính:

- Server tập trung: chứa các dịch vụ backend, được cài đặt theo cách thông thường theo mô hình AllInOne hoặc MultiNode
- Forwarder: forward các gói tin từ agent đến server tập trung và ngược lại. Forwarder đóng vai trò trung gian giữa agent và server đích.
- Agents: chỉ giao tiếp với forwarder



Hình 4. Mô hình triển khai MSSP qua forwarder

3.3. Định cỡ tài nguyên (Sizing)

Mô hình AllInOne không hỗ trợ được số lượng agents trên 3000. Với số lượng agents trên 3000, cần cài đặt mô hình backend MultiNode.

Tùy theo số lượng agents cần triển khai, lượng tài nguyên cần thiết như sau:

3.3.1. Cấu hình server tiêu chuẩn

Cấu hình mỗi server vật lý thông thường chia thành các loại sau:

- Loại 1:
  - CPU: 24 core vật lý
  - RAM: 128 GB
- Loại 2:
  - CPU: 12 core vật lý
  - RAM: 64 GB
- Loại 3:
  - CPU: 8 core vật lý
  - RAM: 32 GB
- Loại 4:
  - CPU: 4 core vật lý
  - RAM: 16 GB

3.3.2. Sizing tài nguyên CPU và RAM theo số agents

Chú ý: với mô hình có multinode cần 3 (hoặc 5) server vật lý (số lẻ) để tính năng HA chạy hiệu quả

Số agents	Mô hình cài đặt	Không có HA		Có HA	
		Số server vật lý	Loại server	Số server vật lý	Loại server
0 → 200	All In One	1	4	2	4
200 → 2k	All In One	1	3	2	4
2k → 5k	All In One	1	2	2	2
	Multi Node	N/A	N/A	3	2
5k → 10k	Multi Node	N/A	N/A	3	2
10k → 25k	Multi Node	N/A	N/A	3	1
25k → 50k	Multi Node	N/A	N/A	5	1

3.3.3. Sizing dung lượng ổ cứng theo số agents

Yêu cầu hiệu năng ổ cứng tối thiểu như sau:

Thông số	Giá trị yêu cầu	Cách kiểm tra
Tốc độ đọc	>= 200 MB/s	Chạy lệnh sau trên node cần kiểm tra (xem *buffered disk reads*): `$ sudo hdparm -Tt /dev/sda`
Tốc độ ghi	>= 100 MB/s	Chạy lệnh sau trên node cần kiểm tra: `$ sudo dd if=/dev/sda of=largefile bs=1M count=200`

Độ trễ ổ cứng	< 1 ms	Cài thêm gói ioping để kiểm tra độ trễ ổ cứng:
		`$ apt update && apt install -y ioping`
		Chạy lệnh sau trên node cần kiểm tra:
		`$ ioping -c 10 .`

IOPS	read: iops >= 500	Cài thêm gói fio để kiểm tra IOPS ổ cứng: `$ apt update && apt install -y fio`

	write: iops >= 200	Chạy lệnh sau trên node cần kiểm tra: $ fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1 --name=tempfile --filename=tempfile --bs=4k --iodepth=64 --size=4G --readwrite=randrw --rwmixread=75

Dung lượng ổ cứng theo số agents triển khai

Số lượng agents	**Tổng dung lượng HDD chia đều các server vật lý (cài VM và lưu log 1 năm)**	
	Không có HA dữ liệu	**Có HA dữ liệu (1 bản sao dữ liệu)**
0 → 50	160 GB	
50 → 200	250 GB	
200 → 500	500 GB	1 TB
500 → 2k	1 TB	2 TB
2k → 5k	2 TB	4 TB
5k → 7k	3 TB	6 TB
7k → 10k	4 TB	8 TB
10k → 15k	6 TB	12 TB
15k → 20k	8 TB	16 TB
20k → 25k	10 TB	20 TB
25k → 30k	12 TB	25 TB
30k → 40k	15 TB	30 TB

3.4. Bảng thông tin phiên bản hỗ trợ nâng cấp

Thông tin về phiên bản hỗ trợ nâng cấp được mô tả trong bảng sau đây:

		Version muốn nâng cấp		
		3.3.0 EDP	**3.3.0 EDR**	**3.3.0 EPP**
Version hiện tại	**3.1.0 EDR**	✓	✓	X
	3.3.0 EPP	✓	✓	
	3.3.0 EDR	✓		X
	3.3.0 EDP		X	X

3.5. Các bước cài đặt backend AllInOne

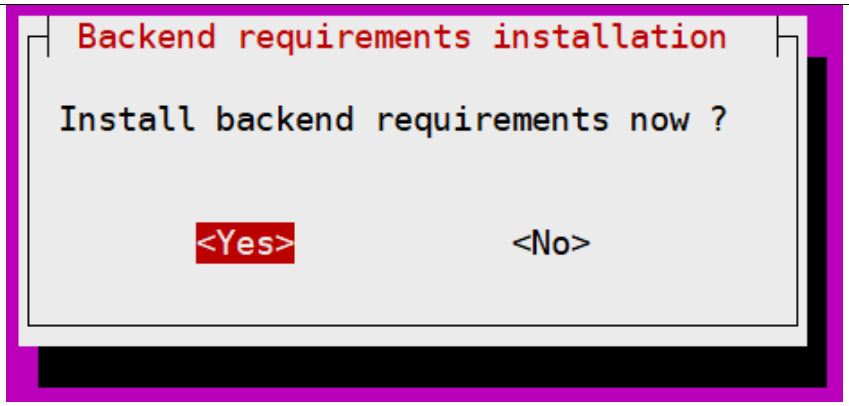
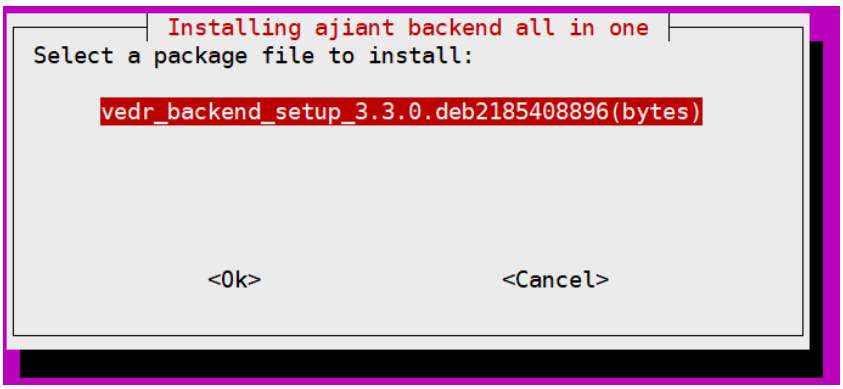
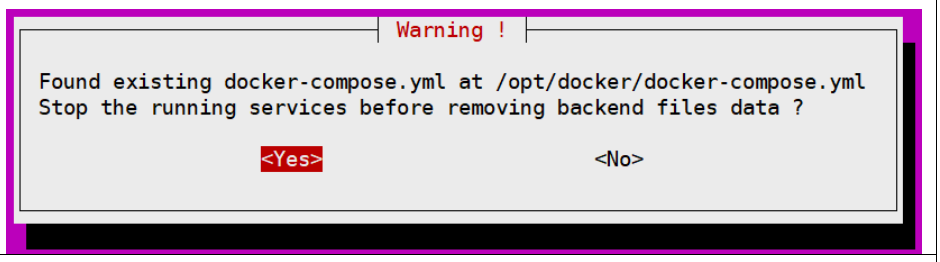
Chú ý: Trước khi cài đặt backend nên quy hoạch sử dụng tên miền cho hệ thống tập trung thay vì sử dụng IP để đơn giản cho quá trình đổi server có thể phát sinh sau giai đoạn triển khai.

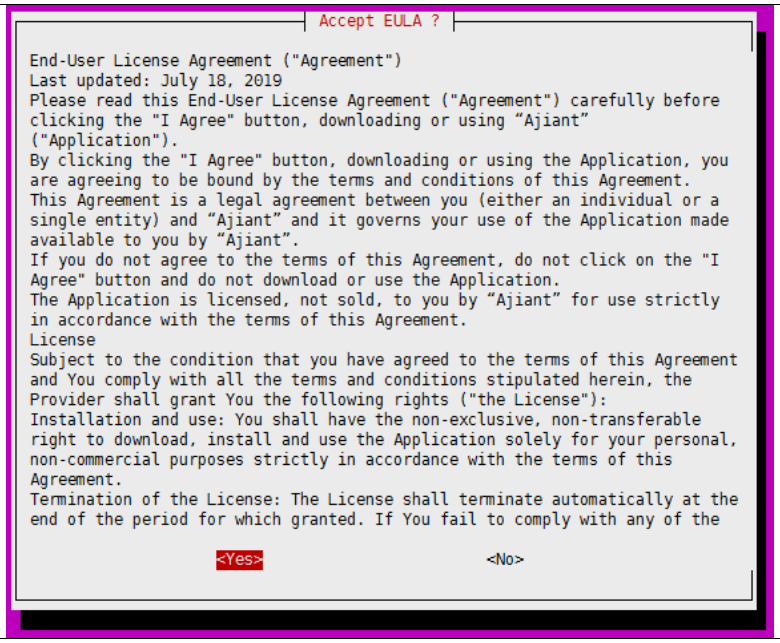
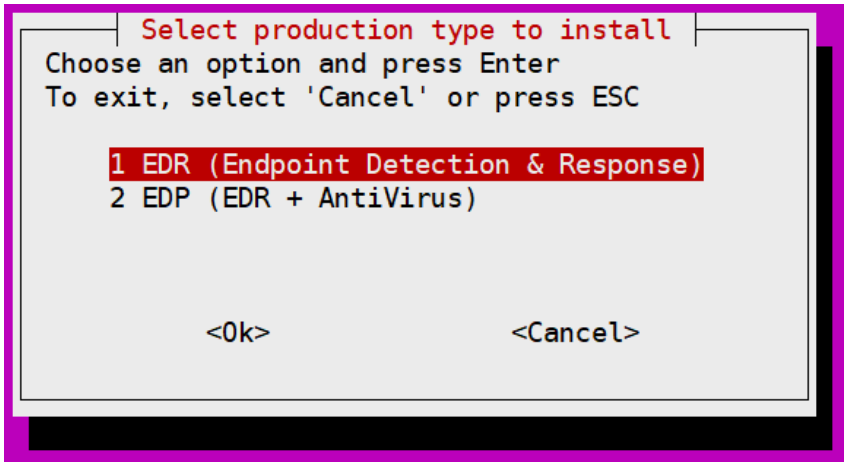
3.5.1. Cài đặt backend AllInOne version 3.3.0

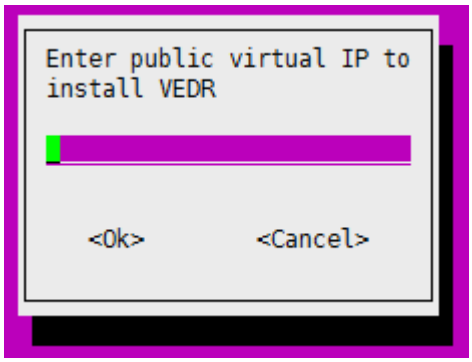
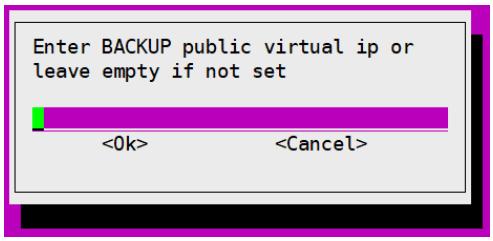
Bước	Tên	Thực hiện
1	Cài hệ điều hành server	Cài đặt 1 node server Ubuntu Server 20.04.3 x64
2	Đồng bộ thời gian cho server	Chỉnh đúng thời gian và múi giờ cho server, khuyến nghị cấu hình NTP để đồng bộ thời gian cho server. Chú ý: thời gian server không đúng có thể gây ra lỗi license, timestamp event log, timestamp alert…

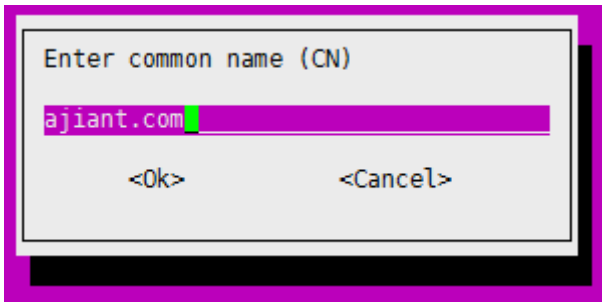
3	Chạy script cài đặt	Copy các file sau lên server và đặt cùng 1 thư mục:
		- install_vedr_backend.sh
		- vedr_requirement.tgz
		- Win7x86.tar.gz
		- vedr_backend_setup_3.3.0.deb
		`ubuntu@ubuntu:~/setup$ ll`
		`total 4921616`
		`drwxrwxr-x 2 ubuntu ubuntu 4096 Dec 28 04:07 ./`
		`drwxr-xr-x 7 ubuntu ubuntu 4096 Dec 28 04:06 ../`
		`-rw-rw-r-- 1 ubuntu ubuntu 7283 Dec 15 09:23 install_vedr_backend.sh`
		`-rw-rw-r-- 1 ubuntu ubuntu 2150973968 Dec 15 09:22 vedr_backend_setup_3.3.0.deb`
		`-rw-rw-r-- 1 ubuntu ubuntu 347358255 Dec 15 03:11 vedr_requirements.tgz`
		`-rw-rw-r-- 1 ubuntu ubuntu 2541368973 Dec 15 02:55 Win7x86.tar.gz`
		Vào thư mục chứa 4 file ở trên, chạy script:
		$ sudo bash install_vedr_backend.sh

4	Cài đặt các gói cần thiết	Cài các gói cần thiết cho server (chọn Yes) hoặc skip (chọn No). Việc cài đặt các gói cần thiết sẽ diễn ra tự động sau đó. **Chú ý:** cài gói offline chỉ thành công với phiên bản Ubuntu Server x64 20.04.3 trở lên. Các phiên bản Ubuntu thấp hơn đều có thể gây lỗi.

		 Sau khi cài đặt các gói cần thiết, nếu trong thư mục đã có sẵn file cài đặt vedr_backend_setup_3.3.0.deb thì sẽ được hiện ra trong danh sách file muốn cài đặt.  Lựa chọn đúng file muốn cài và chọn OK hoặc ấn Enter để tiếp tục. Nếu phát hiện có phiên bản VEDR được cài trước đó, cần xác nhận có dừng tất cả dịch vụ lại trước khi cài bản mới không (chọn Yes) 
5	Đồng ý với điều khoản sử dụng (EULA)	Khi được hỏi chấp nhận điều khoản sử dụng (**EULA**), chọn **Yes** để tiếp tục cài đặt, chọn **No** nếu không chấp nhận và thoát cài đặt.

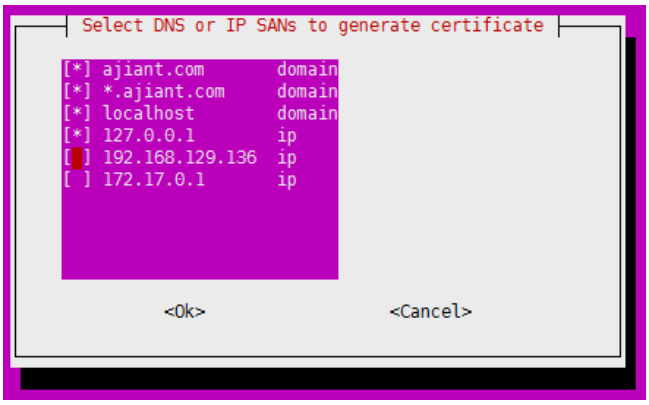
		
6	Chọn phiên bản backend để cài đặt	File vedr_backend_setup_3.3.0.deb hỗ trợ cài đặt 2 phiên bản EDR và EDP: • EDR: chỉ có tính năng của EDR • EDP: có thêm tính năng Antivirus 
7	Nhập địa chỉ của server	Sau khi bộ cài DEB được unpack ra **/opt/docker**, sẽ đến phần cấu hình cài đặt VEDR Backend.

		  Nhập địa chỉ theo dạng tên miền hoặc public ip đã được quy hoạch để các agent kết nối đến server, nếu không sử dụng địa chỉ backup thì bỏ trống. Chú ý: nên quy hoạch sử dụng tên miền thay cho IP để đơn giản cho quá trình đổi server có thể phát sinh sau giai đoạn triển khai.
8	Khởi tạo chứng thư (certificate) cho hệ thống	Nếu quá trình cài đặt phát hiện chứng thư cũ đã tồn tại khi nâng cấp backend - Chọn Yes nếu muốn dùng chứng thư cũ - Chọn No nếu muốn thiết lập lại chứng thư mới  Nếu cài mới thì đến luôn màn tạo mới chứng thư. Trong màn hình tiếp theo (trường hợp tạo mới chứng thư) Nhập CommonName (nhập tên miền hoặc public ip của server). Mặc định là **ajiant.com**. Có thể nhập tên miền mà agent kết nối lên server vào CommonName.

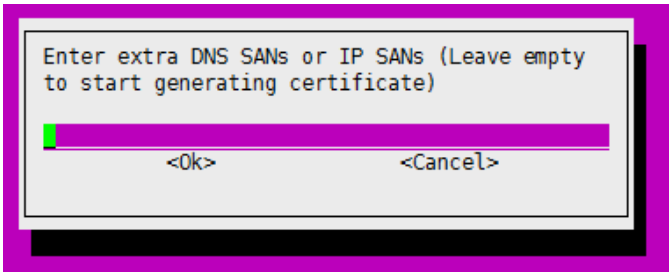


Nhập DNS SANs và IP SANs từ danh sách tùy chọn.

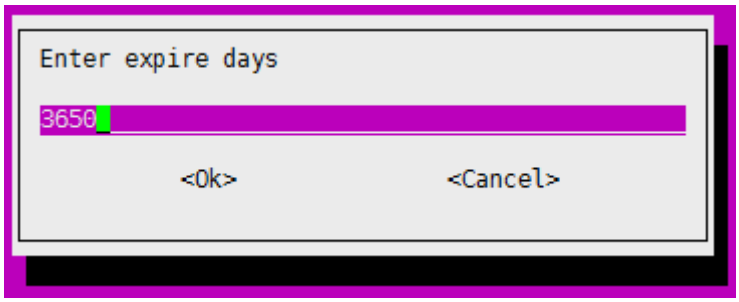
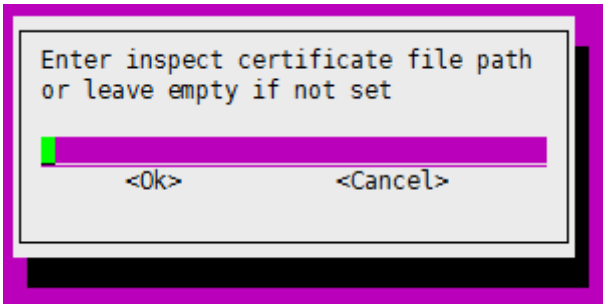
Chú ý: trong danh sách IP SANs phải có tên miền hoặc public ip để agent có thể kết nối đến server.

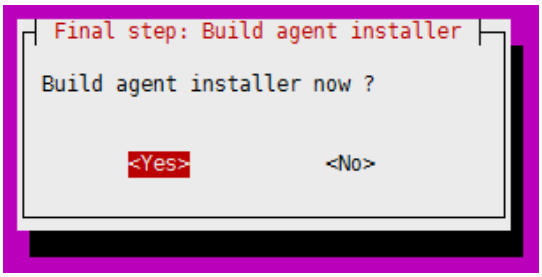


Nếu agent kết nối qua tên miền thì nhập tên miền vào ô DNS SANs.

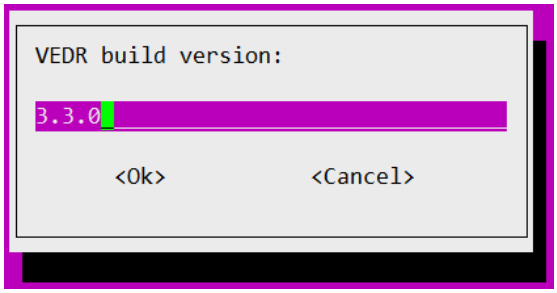


Nhập thời gian hết hạn chứng thư theo ngày (mặc định là 3650):

		 Kết quả sẽ sinh 2 file trong thư mục: /opt/docker/nginx/certs/ - cert.crt - cert.key Common name: edr.viettel.com.vn DNS SANs: edr.viettel.com.vn localhost IP SANs: 127.0.0.1 192.168.129.136 Successfully generated certificate Certificate: /opt/docker/nginx/certs/cert.crt Private Key: /opt/docker/nginx/certs/cert.key
9	Nhập đường dẫn cert inspect	 Nếu không dùng cert inspect để rỗng và enter bỏ qua bước này. Nếu có dùng cert inspect thì nhập đường dẫn cert inspect. Chọn OK. Script sẽ copy file cert_insp vào: /opt/docker/config/cert_insp.crt
10	Tạo bộ cài đặt agent	Khi được xác nhận tạo bộ cài agent không, chọn Yes để tiếp tục.

<table>
<tr>
<td></td>
<td></td>
<td>


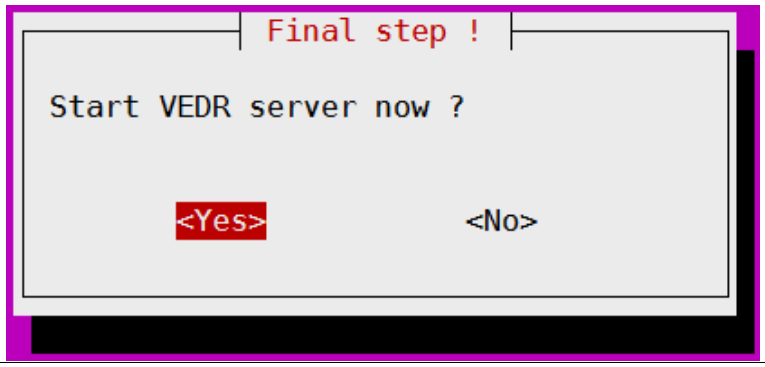
Nhập phiên bản bộ cài agent hoặc ấn Enter để lấy phiên bản mặc định tự sinh



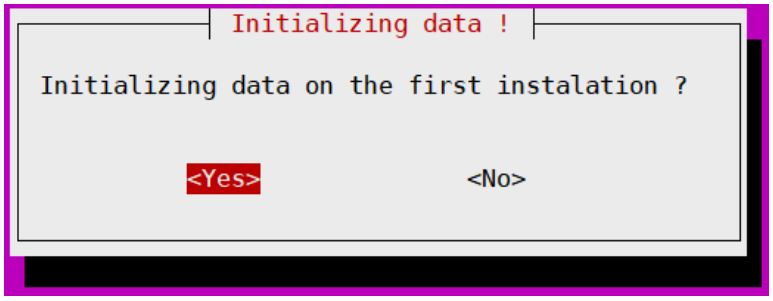
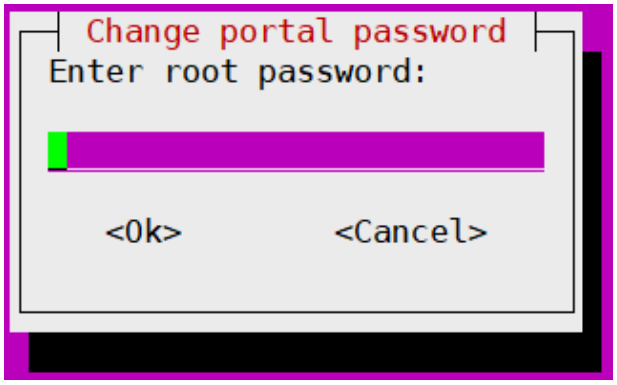
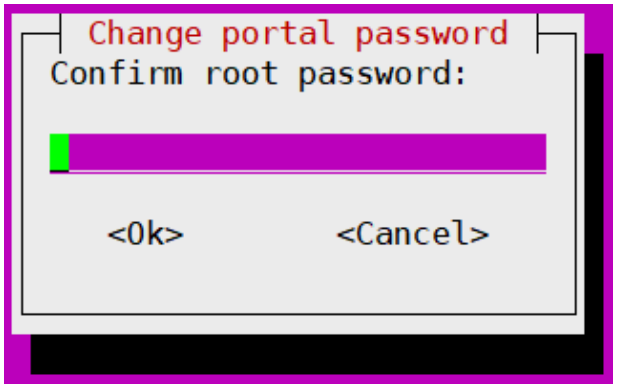
Chờ khoảng 10 phút để server build bộ cài agent (build agent windows có thể đến 5 phút 1 lần build do phải khởi động máy ảo vbox)

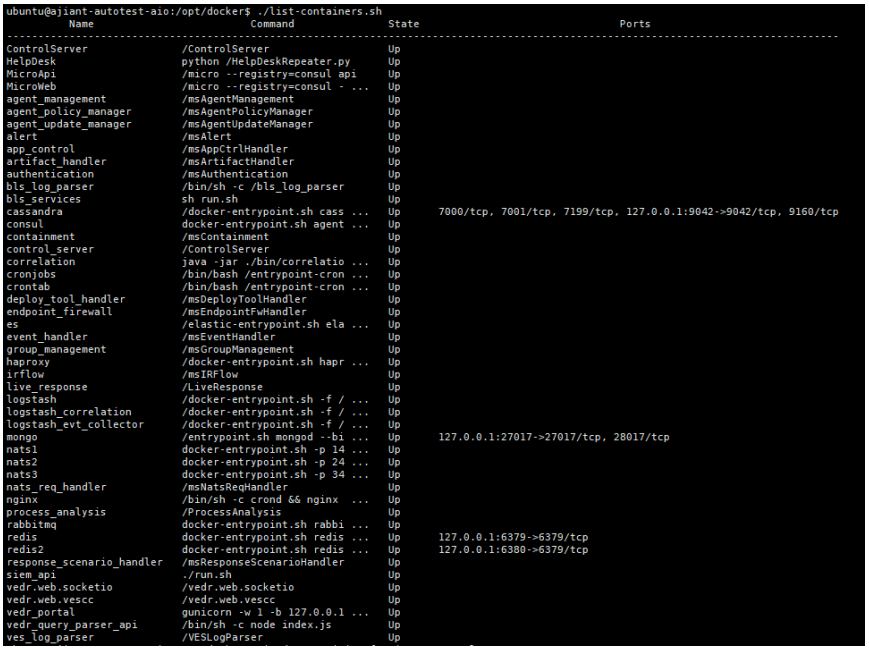
Nếu quá trình cài đặt thành công, server sẽ sinh ra 2 file MSI (cho windows), 1 file DEB, 2 files RPM và 1 file DMG trong thư mục: /opt/docker/repo/public/

ajiant_windows_3.3.0_x64_full.msi
ajiant_windows_3.3.0_x86_full.msi
ajiant_ubuntu_3.3.0_x64_full.deb
ajiant_centos7_3.3.0_x64.rpm
ajiant_centos6_3.3.0_x64.rpm
ajiant_macos_3.3.0_x64_full.dmg

</td>
</tr>
<tr>
<td>11</td>
<td>Khởi động server vedr backend</td>
<td>


</td>
</tr>
</table>

		Chọn ‘y’ hoặc ‘enter’ để khởi động ngay vedr backend hoặc chọn ‘n’ để khởi động sau. Nếu chọn “n”, để khởi động VEDR backend sau đó thì chạy lệnh sau: $ sudo bash /opt/docker/start_vedr.sh
12	Khởi tạo dữ liệu database	 Khi được hỏi có khởi tạo dữ liệu không thì chọn Yes để thực hiện khởi tạo (kể cả cài mới và nâng cấp). Bước tiếp theo là đặt password cho user root, có thể ấn Enter để bỏ qua nếu muốn user root giữ mật khẩu cũ.  Xác nhận password đã nhập  Nếu thành công thì password mới sẽ được thiết lập cho user root.

<table>
<tr>
<td>13</td>
<td>Kiểm tra lại các dịch vụ đang chạy, các port kết nối</td>
<td>
Khi khởi động thành công, chạy script sau để kiểm tra các dịch vụ:
$ cd /opt/docker
$ sudo docker-compose ps
Hoặc
$ cd /opt/docker
$ ./list-containers.sh
Nếu tất cả các service ở trạng thái “UP” là OK.


Kiểm tra các port dịch vụ:
$ telnet <PUBLIC_IP> 8888
$ telnet <PUBLIC_IP> 5672
$ telnet <PUBLIC_IP> 80
$ telnet <PUBLIC_IP> 443
$ telnet <PUBLIC_IP> 4443
$ telnet <PUBLIC_IP> 8443
</td>
</tr>
</table>

3.5.2. Cấu hình HA cho backend AllInOne

Với khách hàng cài đặt backend AllInOne và có yêu cầu HA thì trước hết cần cài đặt 2 node backend AllInOne. Sau đó thực hiện cấu hình HA theo các bước sau:

STT	Công việc	Thực hiện
1	Cài đặt các gói haproxy & keepalived	Copy file haproxy-keepalived.tgz vào 2 node Giải nén bộ file cài đặt: $ tar -zxf haproxy-keepalived.tgz Kết quả giải nén ra thư mục **haproxy-keepalived** Cài đặt các gói DEB: $ sudo dpkg -i haproxy-keepalived/$(lsb_release -s -c)/*.deb Có thể chạy gộp 2 lệnh trên thành 1 lệnh duy nhất: $ tar -zxf haproxy-keepalived.tgz && sudo dpkg -i haproxy-keepalived/$(lsb_release -s -c)/*.deb

2	Cấu hình keepalived	Quy hoạch 01 địa chỉ virtual ip (VIP) cùng dải với ip 2 node AllInOne
		Cấu hình keepalived trên 2 node, tạo file **/etc/keepalived/keepalived.conf** với nội dung trên 2 node như sau:
		Node 1:
		global_defs {
		router_id lb1
		}
		vrrp_script chk_haproxy {
		script "killall -0 haproxy"
		interval 2
		weight 2
		}
		vrrp_instance VI_1 {
		virtual_router_id 51
		advert_int 1
		priority 100
		state MASTER
		interface <net interface, vd: ens160/eth0>
		virtual_ipaddress {
		<VIP> dev <net interface, vd: ens160/eth0>
		}
		authentication {

```
                auth_type PASS
                auth_pass 123456
        }
        track_script {
                chk_haproxy
        }
}
```

Node 2:

```
global_defs {
        router_id lb2
}
vrrp_script chk_haproxy {
        script "killall -0 haproxy"
        interval 2
        weight 2
}
vrrp_instance VI_1 {
        virtual_router_id 51
        advert_int 1
        priority 99
        state MASTER
        interface <net interface, vd: ens160/eth0>
        virtual_ipaddress {
                <VIP> dev <net interface, vd: ens160/eth0>
        }
        authentication {
                auth_type PASS
                auth_pass 123456
        }
        track_script {
                chk_haproxy
```

		}
		}
		Khởi động dịch vụ keepalived:
		$ sudo service keepalived start
		Kiểm tra lại virtual ip đã được gắn vào node 1 chưa:
		$ ip a
		Kiểm tra VIP xuất hiện cùng ip chính không.
		Kiểm tra dịch vụ qua VIP:
		$ ping <VIP>
		$ telnet <VIP> 443

3	Cấu hình cluster cho cassandra	Mở file /opt/docker/docker-compose.yml
		Sửa cấu hình service Cassandra trên 2 node:
		+ Network_mode: host
		+ CASSANDRA_BROADCAST_ADDRESS=<ip node hiện tại>
		+ CASSANDRA_RACK=<2 node nhập giá trị khác nhau, vd: rack1,rack2>
		+ CASSANDRA_SEEDS=<danh sách ip của clusters, gồm cả ip 2 node>
		Khởi động lại Cassandra trên 2 node:
		$ cd /opt/docker/
		$ docker-compose up -d cassandra

```
    cassandra:
        image: cassandra
        restart: always
        container_name: cassandra
        network_mode: host
#       user: 1000:1000
        ports:
            - 9042:9042
            - 8778:8778
        environment:
            - CASSANDRA_CLUSTER_NAME=EDR
            - MAX_HEAP_SIZE=500M
            - HEAP_NEWSIZE=500M
            - CASSANDRA_BROADCAST_ADDRESS=10.0.0.131
            - CASSANDRA_ENDPOINT_SNITCH=GossipingPropertyFileSnitch
            - CASSANDRA_DC=dc1
            - CASSANDRA_RACK=rack1
            - CASSANDRA_SEEDS=10.0.0.131,10.0.0.132
            - CASSANDRA_AUTO_BOOTSTRAP=false
        volumes:
            - ./data:/var/lib/cassandra
            - ./log:/var/log/cassandra
            - ./java:/usr/share/java
```

		Kiểm tra lại trạng thái cluster Cassandra:
		$ /opt/docker/cacssandra/nodetool.sh status
		Nếu hiện 2 node ở trạng thái UP (**UN**) là OK
		`edr@EDR-Cassandra1:/opt/docker$ /opt/docker/cassandra/nodetool.sh status`
		`Datacenter: dc1`
		`==========`
		`Status=Up/Down`
		`|/ State=Normal/Leaving/Joining/Moving`
		`--  Address  Load  Tokens  Owns  Host ID  Rack`
		`UN  10.0.0.131  5.5 MiB  256  ?  86725091-29f7-4392-ade1-764014bc36b7  rack1`
		`UN  10.0.0.132  7.19 MiB  256  ?  2f9527ee-1cb9-4e06-87fd-12fb89bc97a9  rack2`

4	Cấu hình cluster cho elasticsearch	Mở file /opt/docker/elasticsearch/elasticsearch.yml
		Sửa các cấu hình như sau:
		+ network.host: 0.0.0.0
		+ discovery.zen.ping.unicast.hosts: ["<ip node 1>", "<ip node 2>"]
		`cluster.name: "ES-EDR"`
		`node.name: ES`
		`network.host: 0.0.0.0`
		`http.cors.enabled: true`
		`http.cors.allow-origin: "*"`
		`discovery.zen.ping.unicast.hosts: ["10.0.0.121", "10.0.0.122"]`
		`discovery.zen.minimum_master_nodes: 1`
		Restart lại elasticsearch trên 2 node:
		$ cd /opt/docker/
		$ ./restart-containers.sh elasticsearch
		Chờ để elasticsearch khởi động xong, kiểm tra lại trạng thái cluster:
		$ curl http://127.0.0.1:9200/_cluster/health?pretty
		Nếu thấy status=yellow hoặc green và number_of_data_nodes=2 là OK

5	Cấu hình đồng bộ repo từ node chính sang node dự phòng	Tại node 1, chạy 3 lệnh cấu hình để cho phép ssh từ node 1 sang node 2 mà không cần mật khẩu (chạy quyền user thường, không phải root):
		Sinh cặp key ssh:
		$ ssh-keygen
		Copy public key sang node 2:
		$ ssh-copy-id <user>@<ip-node-2>
		Thử ssh sang node 2. Nếu không cần nhập mật khẩu là OK:

		$ ssh <user>@<ip-node-2>
		Chạy thử lệnh sau để đồng bộ thư mục từ node 1 sang node 2:
		$ rsync -avr --delete /opt/docker/repo/ <user>@<ip-node-2>:/opt/docker/repo/
		Tại node 1, chạy lệnh "crontab -e" và thêm dòng cấu hình sau:
		0 */10 * * * rsync -avr --delete /opt/docker/repo/ <user>@<ip-node-2>:/opt/docker/repo/

6	Kiểm tra hoạt động của các dịch vụ	Bật trình duyệt vào portal qua VIP: https://<VIP>/login Thử kiểm tra agents đã cài đặt có online không.

3.6. Các bước cài đặt backend MultiNode

Chú ý: Trước khi cài đặt backend nên quy hoạch sử dụng tên miền cho hệ thống tập trung thay vì sử dụng IP để đơn giản cho quá trình đổi server có thể phát sinh sau giai đoạn triển khai.

3.6.1. Cấu hình các node máy ảo

Cấu hình cài đặt các máy ảo trên mỗi 1 server vật lý:

Để tiện cấu hình về sau, khi cài OS nên đặt chung tài khoản và mật khẩu/key ssh, chung mật khẩu sudo giữa các node.

STT	Loại node VM	Cấu hình	Ghi chú
1	Loadbalancer	CPU: 8 virtual core RAM: 4 GB HDD: 40 GB Network: 1 IP DCN + 1 IP dải nội bộ 10.0.0.0/24	Ldirectord Corosync Pacemaker
2	Docker manager	CPU: 8 virtual core RAM: 8 GB HDD: 80 GB Network: 1 IP dải nội bộ 10.0.0.0/24	Docker manager Microservices Docker registry
3	Queue	CPU: 16 virtual core RAM: 12 GB HDD: 120 GB Network: 1 IP dải nội bộ 10.0.0.0/24	Rabbitmq Logstash Nats streaming Redis

4	Database	CPU: 16 virtual core RAM: 32 GB HDD: 2 TB Yêu cầu HDD: - Tốc độ đọc >= 200 MB/s - Tốc độ ghi >= 100 MB/s Network: 1 IP dải nội bộ 10.0.0.0/24	Elasticsearch Mongodb
5	Cassandra	CPU: 16 virtual core RAM: 16 GB HDD: 120 GB Network: 1 IP dải nội bộ 10.0.0.0/24	Cassandra
6	Correlation	CPU: 12 virtual core RAM: 16 GB HDD: 40 GB Network: 1 IP dải nội bộ 10.0.0.0/24	Correlation engine
Tổng		Tổng: CPU: 76 virtual core RAM: 88 GB HDD: 2.4 TB	

Giữa các server vật lý cần có đường truyền tốc độ cao (khuyến nghị sử dụng đường truyền tối thiểu 1Gbs). Khi cài đặt xong các máy ảo cần test thử tốc độ mạng tại các node giữa các server vật lý khác nhau.

Chú ý kiểm tra tốc độ đọc ghi ổ cứng của node elasticsearch:

- Lệnh kiểm tra tốc độ đọc:

```
$ hdparm -Tt /dev/sda
```

- Lệnh kiểm tra tốc độ ghi:

```
$ dd if=/dev/sda of=largefile bs=1M count=100
```

- Lệnh kiểm tra độ trễ ổ cứng:

```
$ ioping -c 10 .
```

- Lệnh kiểm tra IOPS:

```
$ fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1 --name=tempfile --
filename=tempfile --bs=4k --iodepth=64 --size=4G --readwrite=randrw --rwmixread=75
```

3.6.2. Cấu hình network interfaces

Tại các node cần xác định 1 IP làm local virtual ip. Ví dụ: 10.0.0.100

Khi cấu hình network interfaces, cần đặt gateway là local virtual ip.

Ví dụ với node có IP 10.0.0.101:

```
# The primary network interface
network:
  version: 2
  ethernets:
    ens33:
      addresses: [10.0.0.101/24]
      gateway4: 10.0.0.100
```

Riêng node Loadbalancer cần có 2 network interfaces. Chỉ đặt gateway cho interface có ip public, không đặt gateway cho interface dải local.

Ví dụ LB1 có ip local 10.0.0.11 và ip public 10.30.161.11:

```
# The primary network interface
network:
  version: 2
  ethernets:
    ens33:
      addresses: [10.0.0.11/24]
    ens38:
      addresses: [10.30.161.11/24]
      gateway4: 10.30.161.1
```

3.6.3. Cài đặt backend MultiNode

Bước	Thực hiện
1	Cài đặt các node VM theo cấu hình ở trên. Đồng bộ thời gian cho các server trong cụm multimode: có thể dùng giải pháp NTP.

	Chú ý: thời gian server không đúng có thể gây ra lỗi license, timestamp event log, timestamp alert,…
	
	Đồng thời trên mỗi node VM (trừ các node LB) cài đặt các gói phần mềm cần thiết như sau.
	
	Copy 2 file sau đặt chung vào 1 thư mục:
	- Gói phần mềm cần thiết cho Ubuntu 20: **packages.tgz**
	- Script cài đặt: **install-common-requiments.sh**
	
	Chạy lệnh cài đặt các gói:
	
	$ sudo bash install-common-requirements.sh
	
	`ubuntu@ajiant-staging-dockermanager-3:~/setup$ ll`
	`total 103244`
	`drwxrwxr-x 2 ubuntu ubuntu 4096 Dec 30 07:01 ./`
	`drwxr-xr-x 6 ubuntu ubuntu 4096 Dec 30 07:01 ../`
	`-rw-rw-r-- 1 ubuntu ubuntu 1659 Dec 21 11:15 install-common-requirements.sh`
	`-rw-rw-r-- 1 ubuntu ubuntu 105709503 Dec 21 09:18 packages.tgz`
	`ubuntu@ajiant-staging-dockermanager-3:~/setup$`
	
	Chú ý quan sát quá trình cài đặt gói thành công không.

2	Trên node lb cài đặt các gói cần thiết của LoadBalancer theo các bước sau:
	
	• **Lựa chọn 1**: loadbalancer dùng bộ giải pháp pacemaker, corosync, ldirectord.
	
	Copy file lb-packages-ubuntu20.tgz vào 2 node Loadbalancer
	
	Giải nén bộ file cài đặt:
	
	$ tar -zxf lb-packages-ubuntu20.tgz
	
	Kết quả giải nén ra 2 thư mục **pacemaker-corosync** và **ldirectord-ipvsadm**:
	
	Cài đặt các gói DEB:
	
	$ sudo dpkg -i pacemaker-corosync/*.deb
	$ sudo dpkg -i ldirectord-ipvsadm/*.deb
	
	Có thể chạy gộp 3 lệnh trên thành 1 lệnh duy nhất:
	
	$ tar -zxf lb-packages-ubuntu18.tgz && sudo dpkg -i pacemaker-corosync/*.deb && sudo dpkg –i ldirectord-ipvsadm/*.deb
	
	• **Lựa chọn 2**: loadbalancer dùng bộ giải pháp haproxy, keepalived
	
	Copy file haproxy-keepalived.tgz vào 2 node loadbalancer
	
	Giải nén bộ file cài đặt:
	
	$ tar -zxf haproxy-keepalived.tgz
	
	Kết quả giải nén ra thư mục **haproxy-keepalived**

	Cài đặt các gói DEB:
	`$ sudo dpkg -i haproxy-keepalived/$(lsb_release -s -c)/*.deb`
	Có thể chạy gộp 2 lệnh trên thành 1 lệnh duy nhất:
	`$ tar -zxf haproxy-keepalived.tgz && sudo dpkg -i haproxy-keepalived/$(lsb_release -s -c)/*.deb`
	
	Phần cấu hình 2 node Loadbalancer sẽ ở phần tiếp theo (dùng ansible playbook)

3	Cài đặt node ajiant_registry
	Chọn node làm ajiant docker registry (thường chọn node manager1).
	Chạy script cài đặt các gói cần thiết theo yêu cầu nếu chưa cài ở bước 1:
	`$ sudo bash install-common-requirements.sh`
	Đưa các file cài đặt ansible sau vào node registry:
	- install-ansible.sh
	- ansible.tgz
	Cài đặt gói ansible:
	`$ sudo bash install-ansible.sh`
	
	Tạo thư mục setup, chuẩn bị file ansible_playbook_3.3.0.zip và copy vào trong thư mục setup của node registry:
	`$ mkdir setup && cd setup`
	Giải nén file zip:
	`$ unzip ansible_playbooks_3.3.0.zip`

```
ubuntu@ajiant-staging-dockermanager-1:~/setup$ ll
total 54800
drwxrwxr-x  8 ubuntu ubuntu     4096 Nov 11 18:02 ./
drwxr-xr-x  8 ubuntu ubuntu     4096 Nov 11 17:22 ../
drwxr-xr-x  3 root   root       4096 Jan  7  2020 ansible/
-rw-rw-r--  1 ubuntu ubuntu       65 Nov  9 13:48 ansible.cfg
-rw-rw-r--  1 ubuntu ubuntu  6554712 Nov  9 13:48 ansible.tgz
drwxrwxr-x  2 ubuntu ubuntu     4096 Nov 11 17:26 group_vars/
-rw-rw-r--  1 ubuntu ubuntu     3303 Nov 11 17:42 hosts
-rw-rw-r--  1 ubuntu ubuntu      127 Nov 11 16:37 init-all-data.yml
-rw-rw-r--  1 ubuntu ubuntu      330 Nov 11 17:42 init-database.yml
-rw-rw-r--  1 ubuntu ubuntu     1219 Nov 11 16:37 install_ajiant_registry.sh
-rw-rw-r--  1 ubuntu ubuntu     3608 Nov 11 18:02 install_ajiant_services.sh
-rw-rw-r--  1 ubuntu ubuntu     1119 Nov  9 13:48 install-ansible.sh
-rw-rw-r--  1 ubuntu ubuntu     2126 Nov  9 13:48 install-common-requirements.sh
-rw-------  1 ubuntu ubuntu     1680 Nov  9 16:57 key.pem
-rw-rw-r--  1 ubuntu ubuntu      854 Nov 10 11:01 licence.txt
-rw-rw-r--  1 ubuntu ubuntu      736 Nov 11 17:04 main.yml
drwxr-xr-x  3 root   root       4096 Jan 13  2020 packages/
-rw-rw-r--  1 ubuntu ubuntu 49475799 Nov  9 13:48 packages.tgz
drwxrwxr-x 13 ubuntu ubuntu     4096 Nov 11 16:37 playbook/
drwxrwxr-x  6 ubuntu ubuntu     4096 Nov  9 16:53 roles/
drwxrwxr-x  4 ubuntu ubuntu     4096 Nov 10 11:30 vbox/
ubuntu@ajiant-staging-dockermanager-1:~/setup$ sudo bash install_ajiant_registry.sh
```

Chuẩn bị file **ajiant_swarm_node_setup_3.3.0.deb** và copy vào trong thư mục **setup/ansible/playbook/**

Chuẩn bị file **ajiant_registry_setup_3.3.0.deb** và copy vào thư mục **setup/**

Chuẩn bị folder **vbox** và copy vào trong folder **setup/**

```
ubuntu@ajiant-staging-dockermanager-1:~/setup$ ll vbox/
total 192916
drwxrwxr-x 3 ubuntu ubuntu     4096 Dec 30 14:04 ./
drwxrwxr-x 6 ubuntu ubuntu     4096 Dec 30 14:02 ../
drwxr-xr-x 2 root   root       4096 Dec 21 08:48 focal/
-rw-rw-r-- 1 ubuntu ubuntu     1814 Dec 21 09:30 install_vbox.sh
-rw-r--r-- 1 root   root   11134336 Nov  9 10:10 Oracle_VM_VirtualBox_Extension_Pack-6.1.26.vbox-extpack
-rw-rw-r-- 1 ubuntu ubuntu 93194508 Dec 21 09:09 vbox-packages.tgz
-rw-rw-r-- 1 ubuntu ubuntu 93194508 Dec 30 14:04 Win7x86.tar.gz
ubuntu@ajiant-staging-dockermanager-1:~/setup$
```

Chạy lệnh sau để cài ajiant_registry:

```
$ sudo bash install_ajiant_registry.sh
```

Xem chi tiết quá trình thực hiện cài đặt tại mục 3.5.4

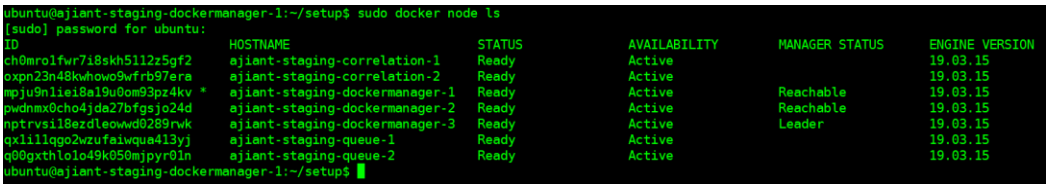
4

Cấu hình ansible theo các bước sau:
Bước 1: Cấu hình file **hosts**: đổi tên file template có sẵn **host.x.x.template** thành file **hosts**. Mở file **hosts** điền thông tin ip của các node theo loại dịch vụ chạy trên node.

- registry: điền ip nội bộ của node registry
- elasticsearch: điền ip nội bộ các node database
- cassandra: điền ip nội bộ các node cassandra
- mongodb: điền ip nội bộ các node database
- ms: điền ip nội bộ các node docker managers
- controlserver: điền ip nội bộ các node docker managers
- repo: điền ip nội bộ các node docker managers
- nats: ip các node chạy nats streaming server

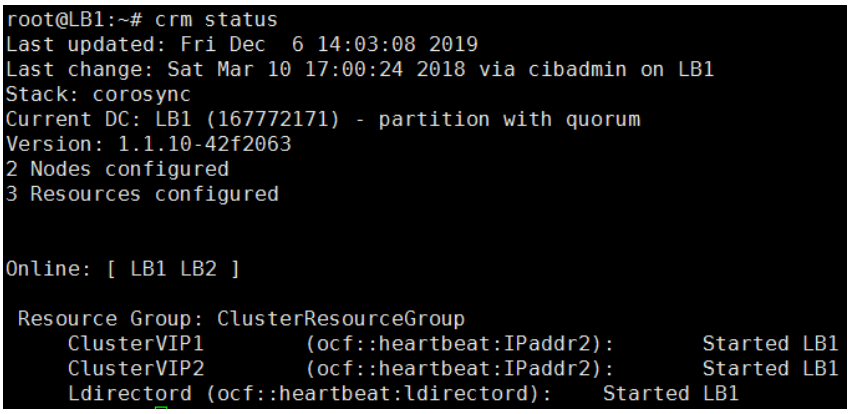
	- rabbitmq: điền ip nội bộ các node queue
	- redis: điền ip nội bộ các node queue
	- correlation và zoo: điền ip nội bộ các node correlation
	- lb1: ip nội bộ node loadbalancer 1
	- lb2: ip nội bộ node loadbalancer 2
	- loadbalancer: ip nội bộ các node loadbalancer
	- managers: điền ip nội bộ các node docker managers
	- workers: điền ip nội bộ các node ngoại trừ node docker managers
	- [all:vars] cấu hình biến tổng thể khi chạy ansible tasks: ansible_user, swarm_manager_master
	  - PUBLIC_VIP: địa chỉ tên miền hoặc public virtual IP của server
	  - LOCAL_VIP: local virtual IP của server **(chú ý LOCAL_VIP trùng với gateway cấu hình ở mục 3.2).** Để trống LOCAL_VIP nếu không có dải mạng nội bộ.
	  - COROSYNC_BIND_ADDRESS: điền dải ip local
	  - REGISTRY_HOST: điền ip nội bộ các node registry
	
	Bước 2: Cấu hình file **group_vars**: Mở file **group_vars/all.yml** và cấu hình các biến sau:
	- microservices_hosts: cấu hình danh sách hostname của các node chạy micro services
	- rabbitmq_hosts: cấu hình danh sách hostname của các node sẽ chạy rabbitmq server
	- correlation_hosts: cấu hình danh sách hostname của các node sẽ chạy correlation engine
	- redis_hosts: cấu hình danh sách hostname của các node sẽ chạy redis server
	- registry_hosts: cấu hình danh sách hostname của node registry
	- rack_node_labels: cấu hình tag số hiệu rack cho các node, các node chạy cùng server vật lý cần có tag chung số hiệu rack

5	Chạy ansible playbook để cấu hình thiết lập môi trường docker swarm, cho các node (trừ LB) cùng tham gia docker swarm
	$ ansible-playbook -i hosts init-swarm.yml -vkK
	SSH password:
	BECOME password[defaults to SSH password]:
	Nhập password ssh và password để lên quyền root
	Quan sát quá trình cài đặt xem có thành công không.
	
	Khi cài đặt thành công, kiểm tra danh sách node trong swarm:
	$ sudo docker node ls

	 Như vậy tất cả các node (trừ LB) đã cùng ở trong docker swarm. **Chú ý**: kiểm tra các HOSTNAME ở trên có giống hostname cấu hình trong **group_vars/all.yml** không !
6	Chỉ làm bước này khi bước trên thành công Chạy script cấu hình các dịch vụ backend ajiant. $ sudo bash install_ajiant_services.sh SSH password: BECOME password[defaults to SSH password]: Nhập password ssh và password để lên quyền root

7	Chạy lệnh sinh bộ cài agent (yêu cầu phải cài trước virtualbox và unpack gói máy ảo):
	$ cd /opt/ajiant && sudo bash gen_agent_installer.sh
	Khi kết thúc sinh bộ cài agent thì sẽ sinh ra 2 file MSI (cho windows), 1 file DEB, 2 files RPM và 1 file DMG trong thư mục: **/opt/ajiant/repo/public/**
	- ajiant_windows_3.3.0_x64_full.msi
	- ajiant_windows_3.3.0_x86_full.msi
	- ajiant_ubuntu_3.3.0_x64_full.deb
	- ajiant_centos7_3.3.0_x64.rpm
	- ajiant_centos6_3.3.0_x64.rpm
	- ajiant_macos_3.3.0_x64_full.dmg
	Chạy đồng bộ repo từ registry ra các node thuộc nhóm repo:
	$ sudo ansible-playbook -i hosts playbook/sync-repo.yml

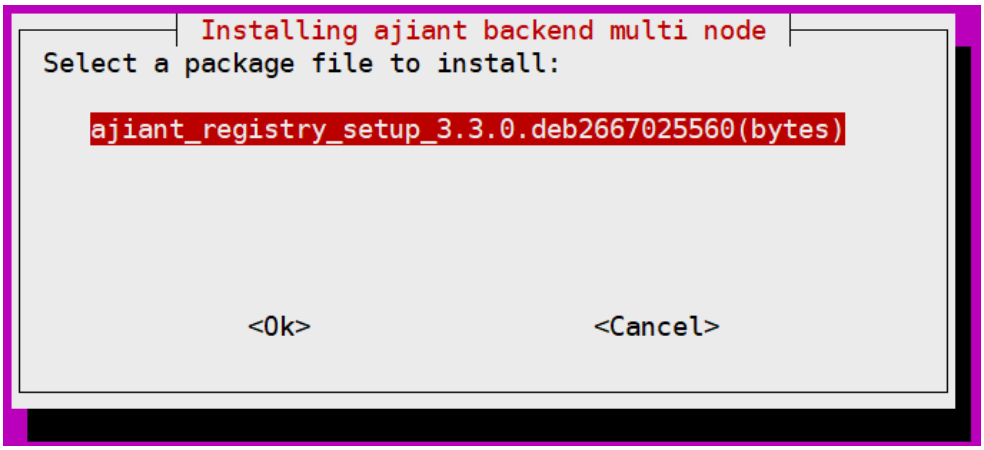
8	Vào 2 node Loadbalancer kiểm tra cài đặt LB: $ sudo crm status

	 Tại cả LB1 và LB2 đều hiện cùng kết quả Started là LB1 (hoặc cùng là LB2) Khi đó LB1 là LoadBalancer master, còn LB2 là LoadBalancer slave. Thử lệnh sau tại node master: $ sudo ipvsadm

10	Kiểm tra lại các port dịch vụ: 80, 443, 4443, 8443, 8888, 5672
	$ telnet <virtual_ip> 80
	$ telnet <virtual_ip> 443
	$ telnet <virtual_ip> 4443
	$ telnet <virtual_ip> 8443
	$ telnet <virtual_ip> 8888
	$ telnet <virtual_ip> 5672

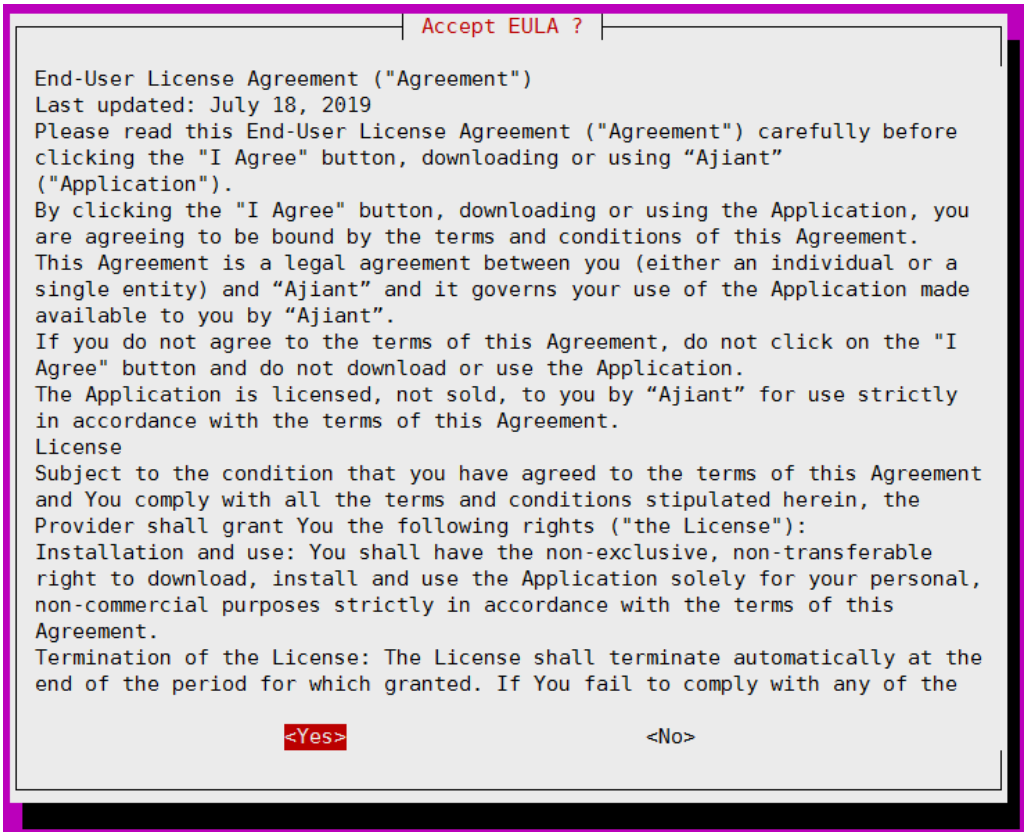
3.6.4. Cài đặt AJiantRegistry

STT	Thực hiện
1	Trên node registry, chạy file shell script: $ sudo bash install_ajiant_registry.sh
3	Nếu có các file cài đặt ajiant_registry_setup_3.3.0.deb thì sẽ được hiện ra trong danh sách file muốn cài đặt.

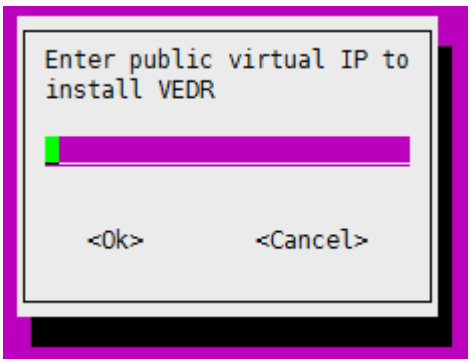
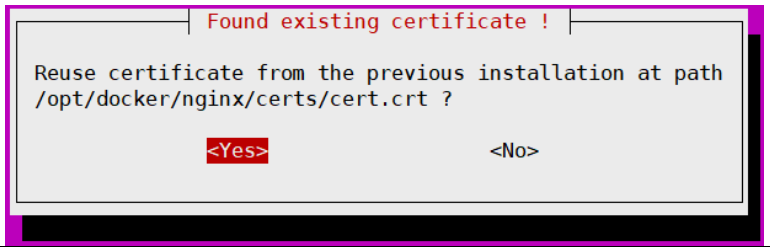


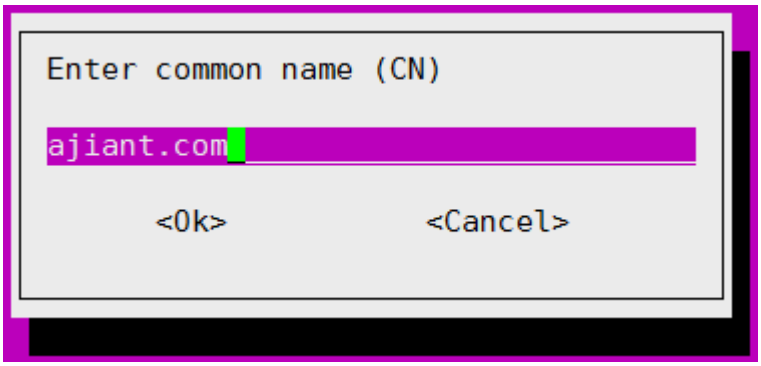
Lựa chọn đúng file muốn cài và chọn OK hoặc ấn Enter để tiếp tục

4 Accept EULA, chấp nhận điều khoản cài đặt



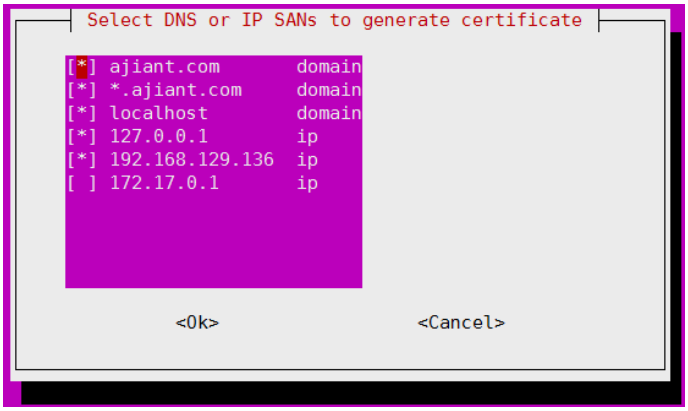
Chọn Yes để tiếp tục hoặc chọn No nếu không chấp nhận điều khoản cài đặt và thoát cài đặt.

5	Sau khi bộ cài DEB được unpack ra **/opt/ajiant**, sẽ đến phần cấu hình cài đặt VEDR Backend.  Nhập địa chỉ theo dạng tên miền hoặc public ip đã được quy hoạch để các agent kết nối đến server, nếu không sử dụng địa chỉ backup thì bỏ trống. Chú ý: nên quy hoạch sử dụng tên miền thay cho IP để đơn giản cho quá trình đổi server có thể phát sinh sau giai đoạn triển khai.
6	Nếu quá trình cài đặt phát hiện chứng thư cũ đã tồn tại khi nâng cấp backend - Chọn Yes nếu muốn dùng chứng thư cũ - Chọn No nếu muốn thiết lập lại chứng thư mới 
7	Trong màn hình tiếp theo (trường hợp tạo mới chứng thư) Nhập CommonName (nhập tên miền hoặc public ip của server). Mặc định là **ajiant.com**. Có thể nhập tên miền mà agent kết nối lên server vào CommonName.

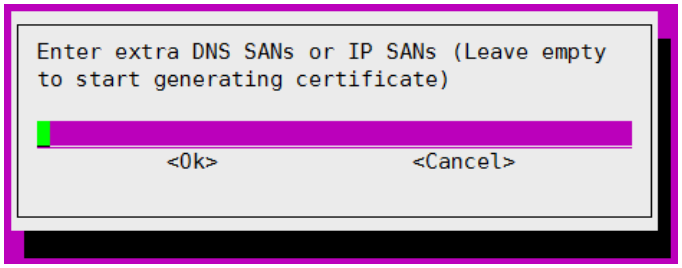


Nhập DNS SANs và IP SANs từ danh sách tùy chọn.

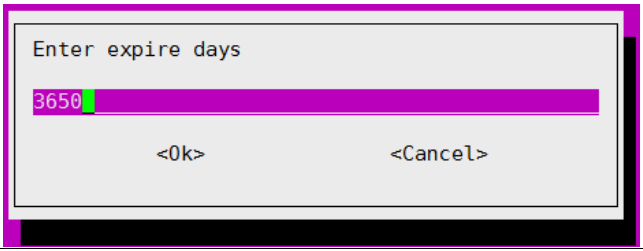
Chú ý: trong danh sách IP SANs phải có tên miền hoặc public ip để agent có thể kết nối đến server.

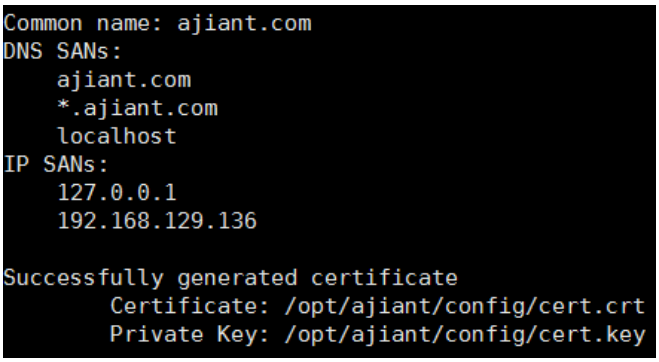


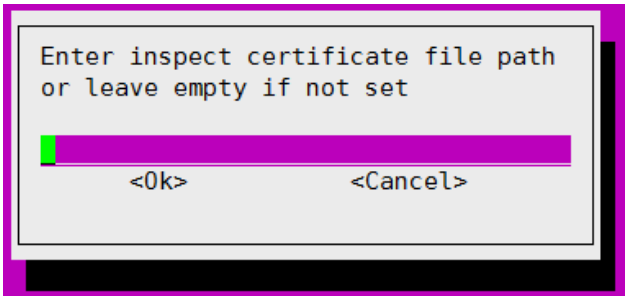
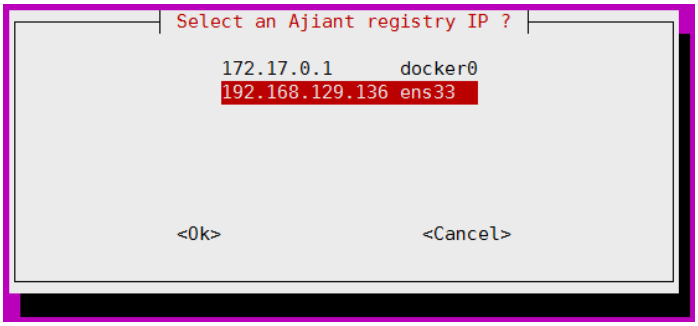
Nếu agent kết nối qua tên miền thì nhập tên miền vào ô DNS SANs.

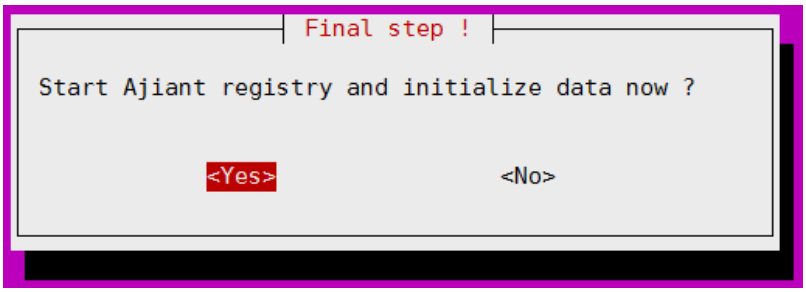
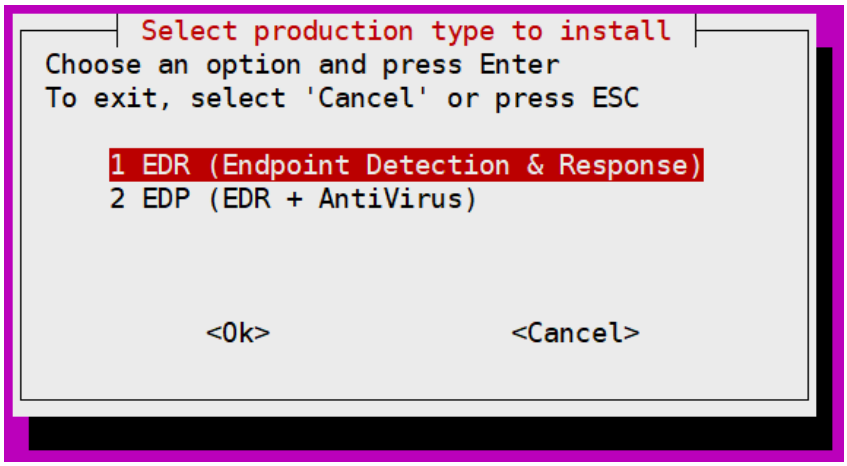


Nhập thời gian hết hạn chứng thư theo ngày (mặc định là 3650):



<table>
<tr><td></td><td>Kết quả sẽ sinh 2 file trong thư mục: /opt/ajiant/config/
- cert.crt
- cert.key
Khi tạo chứng thư thành công sẽ có thông báo như sau:
 

Nếu không dùng cert inspect để rỗng và enter bỏ qua bước này.
Nếu có dùng cert inspect thì nhập đường dẫn cert inspect. Chọn OK. Script sẽ copy file cert_insp vào: /opt/ajiant/config/cert_insp.crt
 
</td></tr>
<tr><td>8</td><td>Chọn ip của docker registry trong danh sách (cần chọn dải nội bộ để các node VM khác truy cập)
 
</td></tr>
<tr><td>9</td><td>Khởi tạo registry:</td></tr>
</table>

	 Chọn Yes để khởi tạo docker registry cho Ajiant backend.
10	Bộ cài multinode chỉ hỗ trợ cài đặt 2 phiên bản EDR và EDP, không hỗ trợ phiên bản EPP • EDR: chỉ có tính năng của EDR • EDP: có thêm tính năng Antivirus  Theo quá trình cài đặt diễn ra đến khi kết thúc thành công.

3.7. Hướng dẫn nâng cấp hệ thống

Lưu ý trước khi nâng cấp hệ thống cần backup những file sau:

/opt/docker/nginx/certs/cert.crt

/opt/docker/nginx/certs/cert.crt

3.7.1. Mô hình AllInOne

Chú ý:

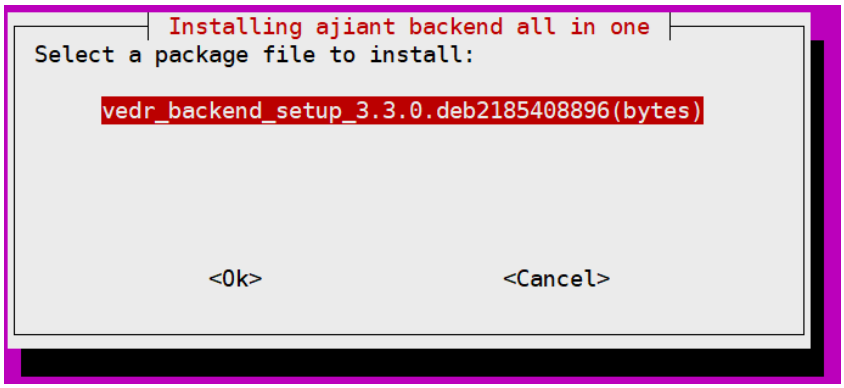
- Với mô hình AllInOne cần backup thêm file sau để phục hồi khi cần:

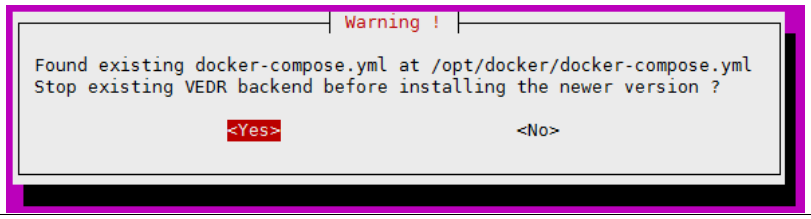
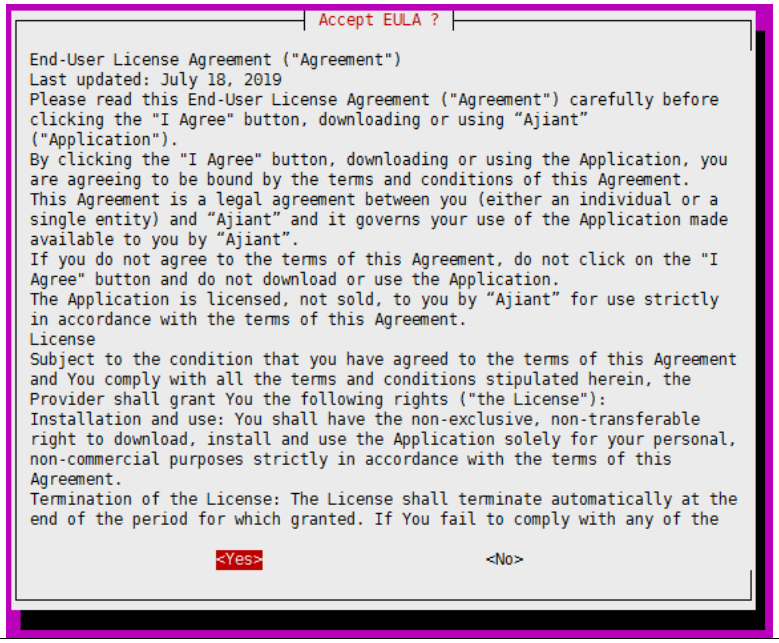
  /opt/docker/docker-compose.yml

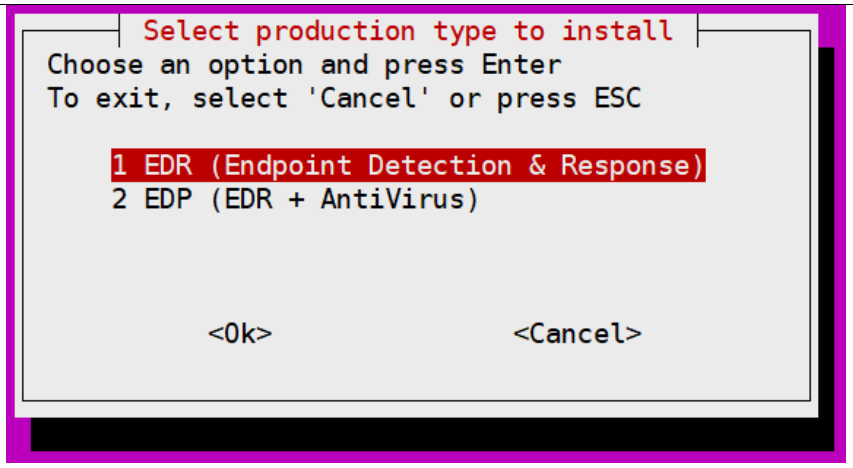
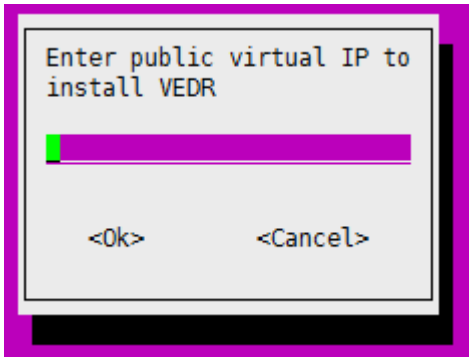
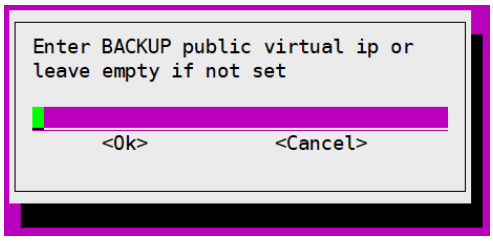
- Việc nâng cấp version backend cần thực hiện lần lượt theo thứ tự 2.0.0 -> 3.0.0 -> 3.1.0 -> 3.3.0

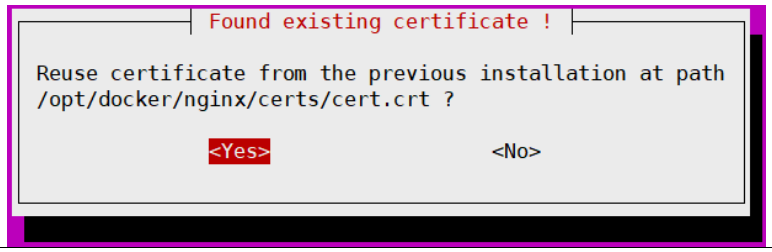
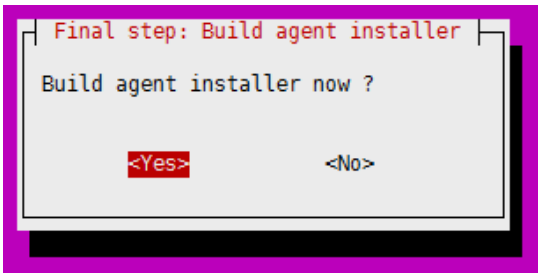
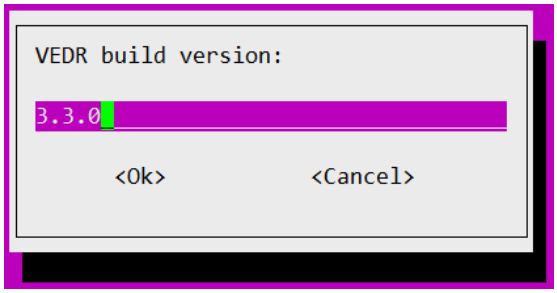
- File bộ cài backend vừa sử dụng vừa để cài mới vừa để nâng cấp từ version cũ.
- Khi thực hiện nâng cấp, các files agent của nhóm update release sẽ tự động được cập nhật. Do đó nếu không muốn update agent hàng loạt thì khuyến nghị tạo nhóm update khác (ví dụ: pre_release), sau đó chuyển tất cả agents sang nhóm pre_release, rồi mới chuyển dần các agent sang nhóm release.

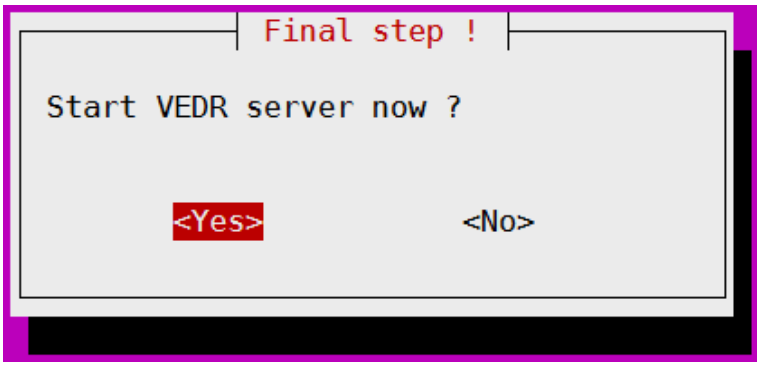
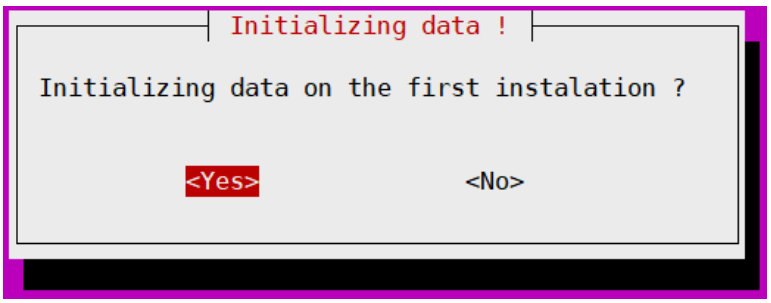
Cụ thể như sau:

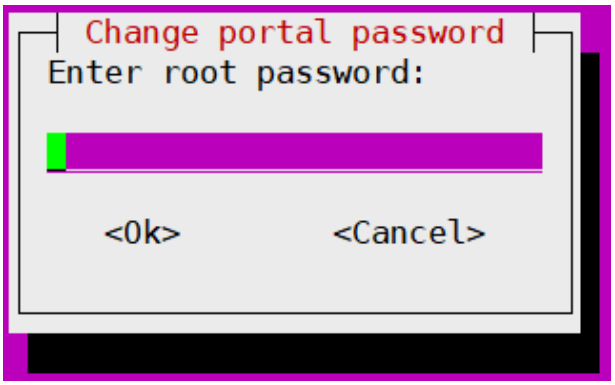
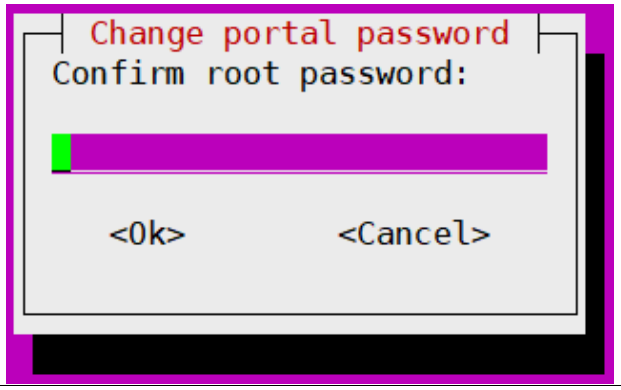
Bước	Tên	Thực hiện
1	Kiểm tra tài nguyên hệ thống cũ	Yêu cầu ổ cứng còn trống khoảng 10 GB trở lên. Kiểm tra ổ cứng trống bằng lệnh: $ df -h
2	Chạy script cài đặt	Copy các files sau lên server đặt cùng 1 thư mục setup: - install_vedr_backend.sh - vedr_backend_setup_3.3.0.deb Vào thư mục setup ở trên, chạy script: $ sudo bash install_vedr_backend.sh Nếu có các file cài đặt vedr_backend_setup_3.3.0.deb thì sẽ được hiện ra trong danh sách file muốn cài đặt.  Lựa chọn đúng file muốn cài và chọn OK hoặc ấn Enter để tiếp tục.

		Trong quá trình nâng cấp, trình cài đặt phát hiện có phiên bản backend được cài trước đó, cần xác nhận có dừng tất cả dịch vụ lại trước khi cài bản mới không (chọn YES) 
3	Đồng ý với điều khoản sử dụng (EULA)	Khi được hỏi chấp nhận điều khoản sử dụng (**EULA**), chọn **Yes** để tiếp tục cài đặt, chọn **No** nếu không chấp nhận và thoát cài đặt. 
4	Chọn phiên bản backend để cài đặt	File vedr_backend_setup_3.3.0.deb hỗ trợ cà đặt 2 phiên bản EDR và EDP: • EDR: chỉ có tính năng của EDR • EDP: có thêm tính năng Antivirus

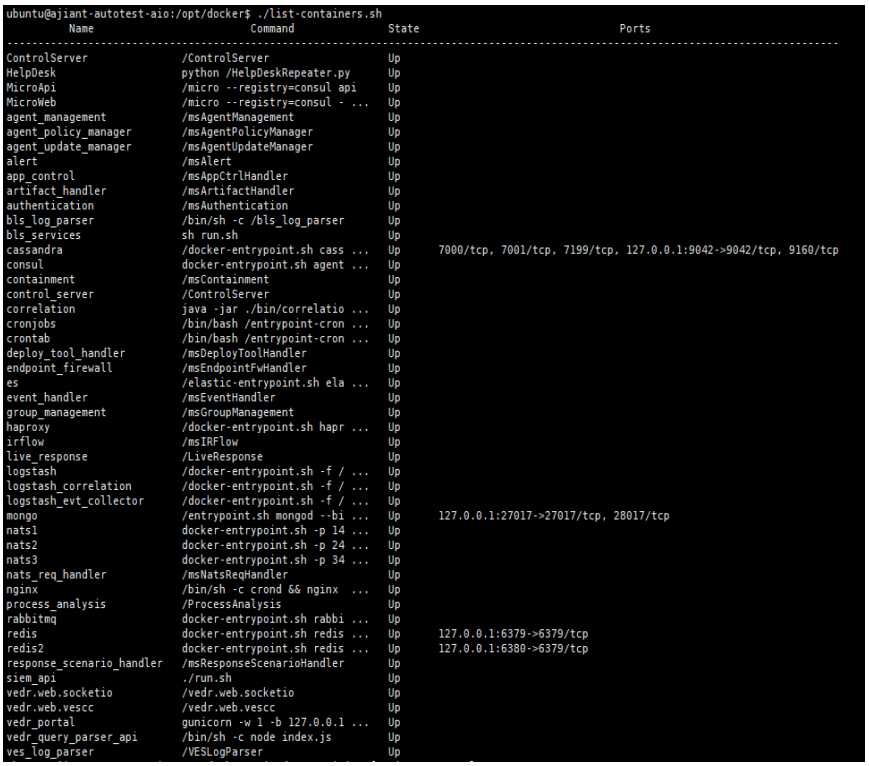
		
5	Nhập địa chỉ của server	Sau khi bộ cài DEB được unpack ra **/opt/docker**, sẽ đến phần cấu hình cài đặt VEDR Backend.   Nhập địa chỉ theo dạng tên miền hoặc public ip đã được quy hoạch để các agent kết nối đến server. Khi nâng cấp thì thông tin địa chỉ này được điền sẵn giá trị ở lần cài trước đó.
6	Khởi tạo chứng thư (certificate) cho hệ thống	Nếu quá trình cài đặt phát hiện chứng thư cũ đã tồn tại khi nâng cấp version backend - **Chọn Yes nếu muốn dùng chứng thư cũ (recommend)** - Chọn No nếu muốn thiết lập lại chứng thư mới

		
7	Tạo bộ cài đặt agent	Khi được xác nhận tạo bộ cài agent không, chọn Yes để tiếp tục.  Nhập phiên bản bộ cài agent hoặc ấn Enter để lấy phiên bản mặc định tự sinh  Chờ khoảng 10 phút để build bộ cài agent (***build agent windows có thể đến 5 phút 1 lần build do phải khởi động máy ảo virtual box***) Nếu quá trình cài đặt thành công, sẽ sinh ra 2 file MSI (cho windows), 1 file DEB, 2 files RPM và 1 file DMG trong thư mục: **/opt/docker/repo/public/** - ajiant_windows_3.3.0_x64_full.msi - ajiant_windows_3.3.0_x86_full.msi - ajiant_ubuntu_3.3.0_x64_full.deb - ajiant_centos7_3.3.0_x64.rpm - ajiant_centos6_3.3.0_x64.rpm - ajiant_macos_3.3.0_x64_full.dmg

8	Khởi động server vedr backend	 Chọn 'y' hoặc 'enter' để khởi động ngay vedr backend hoặc chọn 'n' để khởi động sau. Nếu chọn "n", để khởi động VEDR backend sau đó thì chạy lệnh sau: $ sudo bash /opt/docker/start_vedr.sh
9	Khởi tạo dữ liệu database	 Khi được hỏi có khởi tạo dữ liệu không thì chọn Yes để thực hiện khởi tạo (kể cả khi cài mới và nâng cấp). Trong trường hợp các dịch vụ database khởi động xong, trình cài đặt sẽ hỏi có thử kết nối lại database không. Nên chờ khoảng 15-30 phút đảm bảo các database chạy lênh bình thường. Database elasticsearch có thể khởi động khá lâu do có nhiều dữ liệu cũ. Thử chạy lệnh sau để kiểm tra cluster health: $ curl http://127.0.0.1:9200/_cluster/health?pretty Nếu ở trạng thái RED thì cần chờ thêm, nếu YELLOW hoặc GREEN thì có thể tiếp tục chạy. Bước tiếp theo là đặt password cho user root, có thể ấn Enter để bỏ qua nếu muốn user root giữ mật khẩu cũ.

		 Xác nhận password đã nhập 

10	Kiểm tra lại các dịch vụ đang chạy, các port kết nối	Khi khởi động thành công, chạy script sau để kiểm tra các dịch vụ:
		$ cd /opt/docker
		$ sudo docker-compose ps
		Hoặc
		$ ./list-containers.sh
		Nếu tất cả các service ở trạng thái “UP” là OK.



```
ubuntu@ajiant-autotest-aio:/opt/docker$ ./list-containers.sh
          Name                        Command               State                              Ports
-------------------------------------------------------------------------------------------------------------------------
ControlServer               /ControlServer                   Up
HelpDesk                    python /HelpDeskRepeater.py      Up
MicroApi                    /micro --registry=consul api     Up
MicroWeb                    /micro --registry=consul - ...   Up
agent_management            /msAgentManagement               Up
agent_policy_manager        /msAgentPolicyManager            Up
agent_update_manager        /msAgentUpdateManager            Up
alert                       /msAlert                         Up
app_control                 /msAppCtrlHandler                Up
artifact_handler            /msArtifactHandler               Up
authentication              /msAuthentication                Up
bls_log_parser              /bin/sh -c /bls_log_parser       Up
bls_services                sh run.sh                        Up
cassandra                   /docker-entrypoint.sh cass ...   Up      7000/tcp, 7001/tcp, 7199/tcp, 127.0.0.1:9042->9042/tcp, 9160/tcp
consul                      docker-entrypoint.sh agent ...   Up
containment                 /msContainment                   Up
control_server              /ControlServer                   Up
correlation                 java -jar ./bin/correlatio ...   Up
cronjobs                    /bin/bash /entrypoint-cron ...   Up
crontab                     /bin/bash /entrypoint-cron ...   Up
deploy_tool_handler         /msDeployToolHandler             Up
endpoint_firewall           /msEndpointFwHandler             Up
es                          /elastic-entrypoint.sh ela ...   Up
event_handler               /msEventHandler                  Up
group_management            /msGroupManagement               Up
haproxy                     /docker-entrypoint.sh hapr ...   Up
irflow                      /msIRFlow                        Up
live_response               /LiveResponse                    Up
logstash                    /docker-entrypoint.sh -f / ...   Up
logstash_correlation        /docker-entrypoint.sh -f / ...   Up
logstash_evt_collector      /docker-entrypoint.sh -f / ...   Up
mongo                       /entrypoint.sh mongod --bi ...   Up      127.0.0.1:27017->27017/tcp, 28017/tcp
nats1                       docker-entrypoint.sh -p 14 ...   Up
nats2                       docker-entrypoint.sh -p 24 ...   Up
nats3                       docker-entrypoint.sh -p 34 ...   Up
nats_req_handler            /msNatsReqHandler                Up
nginx                       /bin/sh -c crond && nginx  ...   Up
process_analysis            /ProcessAnalysis                 Up
rabbitmq                    docker-entrypoint.sh rabbi ...   Up
redis                       docker-entrypoint.sh redis ...   Up      127.0.0.1:6379->6379/tcp
redis2                      docker-entrypoint.sh redis ...   Up      127.0.0.1:6380->6379/tcp
response_scenario_handler   /msResponseScenarioHandler       Up
siem_api                    ./run.sh                         Up
vedr.web.socketio           /vedr.web.socketio               Up
vedr.web.vescc              /vedr.web.vescc                  Up
vedr_portal                 gunicorn -w 1 -b 127.0.0.1 ...   Up
vedr_query_parser_api       /bin/sh -c node index.js         Up
ves_log_parser              /VESLogParser                    Up
```

Kiểm tra các port dịch vụ:

```
$ telnet <PUBLIC_IP> 8888
$ telnet <PUBLIC_IP> 5672
$ telnet <PUBLIC_IP> 80
$ telnet <PUBLIC_IP> 443
$ telnet <PUBLIC_IP> 4443
$ telnet <PUBLIC_IP> 8443
```

Thử vào portal đăng nhập vào hệ thống và kiểm tra một số tính năng cơ bản theo tài liệu test UAT.

3.7.2. Mô hình MultiNode

Với mô hình MultiNode hiện tại chỉ hỗ trợ cài mới, chưa hỗ trợ nâng cấp hệ thống.

Khi nâng cấp cần đội giải pháp tham gia thực hiện.

3.8. Hướng dẫn kích hoạt license tập trung

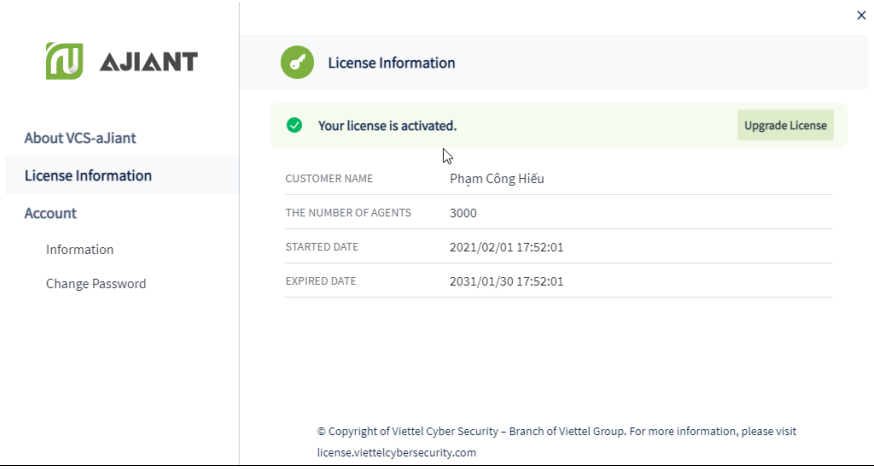
Tính năng license tập trung phục vụ bài toán kinh doanh, cho phép cung cấp sản phẩm theo hợp đồng đã ký kết với đơn vị, quy định về thời hạn sử dụng sản phẩm và số lượng agents tối đa cho phép cài đặt tại đơn vị.

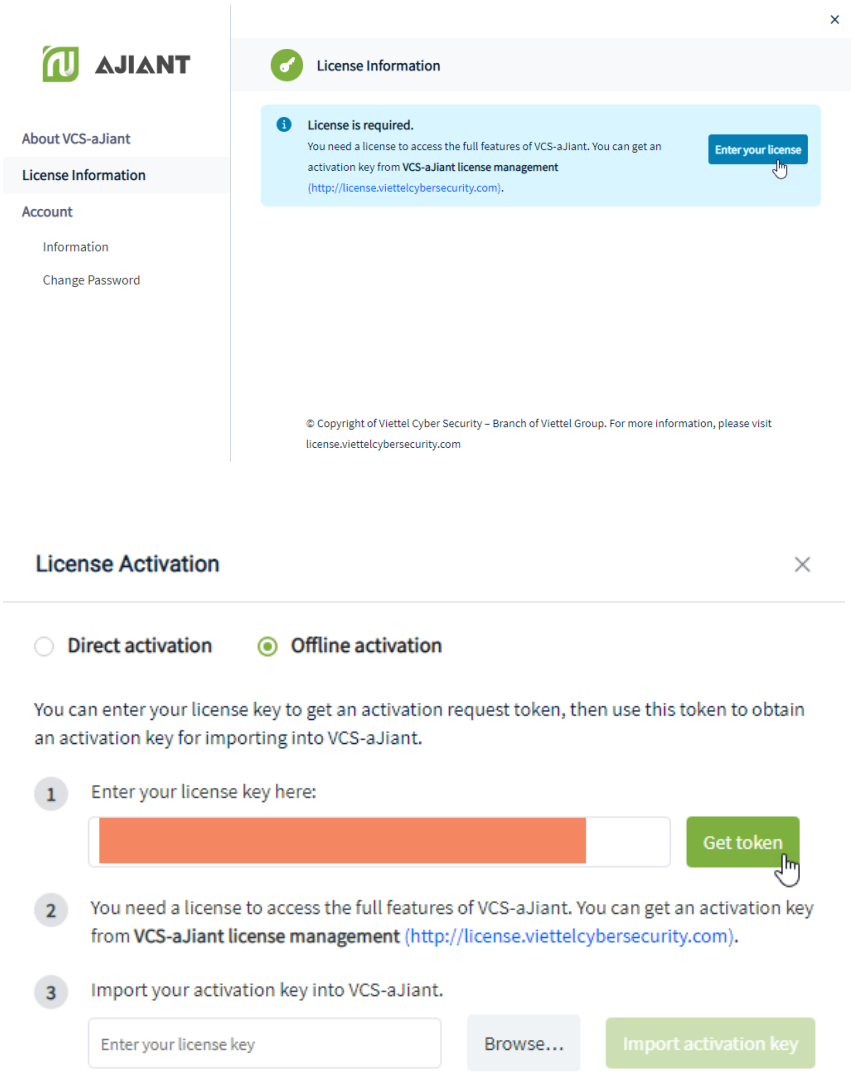
Về luồng quản lý license, sản phẩm gửi thông tin lên hệ thống quản lý license tập trung của VCS để kích hoạt, gia hạn, hủy hoặc định kỳ kiểm tra thời gian còn lại của license để quyết định cho phép agent hoạt động (trường hợp hết license, đơn vị vẫn có thể truy cập giao diện tuy nhiên không còn nhận được log từ agent nữa).

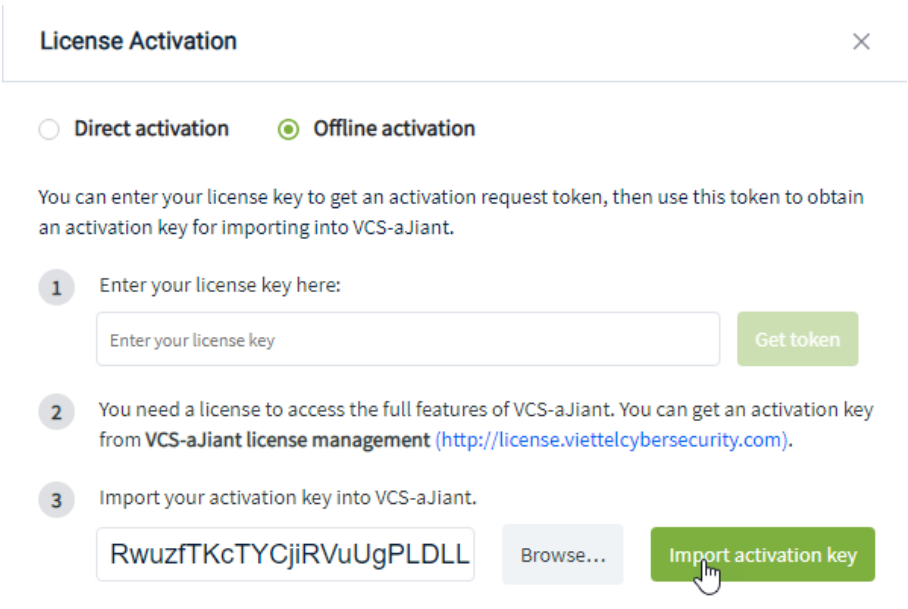
Lưu ý về cách thức kích hoạt license theo khách hàng:

- Kích hoạt online: áp dụng với khách hàng cho phép truy cập ngoại mạng từ hệ thống nội bộ.
- Kích hoạt offline: áp dụng với khách hàng **không** cho phép truy cập ngoại mạng từ hệ thống nội bộ (hệ thống bị cô lập).

Để tránh việc tính năng của phần mềm phụ thuộc vào đường Internet, khuyến nghị sử dụng cách kích hoạt license offline theo các bước sau:

Bước	Tên	Thực hiện
1	Tạo license key cho từng khách hàng	Gửi các thông tin qua email cho PM của dự án: • Họ tên, số điện thoại và địa chỉ email của đầu mối bên khách hàng. • Số lượng agent tối đa và thời hạn license của khách hàng theo Hợp đồng. PM dự án sẽ tạo license key cho từng khách hàng dựa vào các thông tin trên. Lưu ý: họ tên của đầu mối bên khách hàng sẽ hiển thị trên trang thông tin license 
2	Lấy token	Đăng nhập portal VCS-aJiant, click vào icon thông tin **User > License Information > Enter your license > Offline activation**

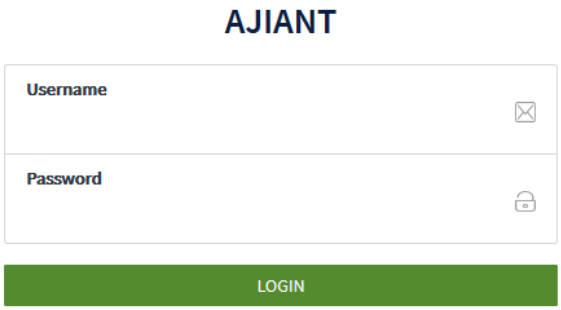
		 Nhập license key vào **Enter your license key here**, click nút Get token, hệ thống sẽ cho tải về file **offline_request.txt**
3	Kích hoạt license	Gửi file offline_request.txt cho PO để PO tạo tiếp activation key dưới dạng 1 file text. Sau khi PO gửi lại activation key, import file text chứa activation key vào mục **Import your activation key into VCS-aJiant**



3.9. Đăng nhập Portal

Sau khi cài đặt thành công, vào trình duyệt, truy cập portal tại địa chỉ sau:

https://<ajiant-ip>



Đăng nhập bằng tài khoản quản trị đã được cung cấp

3.10. Tải bộ cài agent từ repo

Tải về bộ cài agent về tại địa chỉ URL:

https://<AJIANT_IP>/repo/

Hoặc

https://<AJIANT_IP>:8443/repo/

Index of /repo/

../		
release/	28-Dec-2021 10:33	-
ajiant_centos6_3.3.0_x64_full.rpm	28-Dec-2021 11:04	52827752
ajiant_centos7_3.3.0_x64_full.rpm	28-Dec-2021 11:05	54602684
ajiant_macos_3.3.0_x64_full.dmg	28-Dec-2021 09:37	34210789
ajiant_ubuntu_3.3.0_x64_full.deb	28-Dec-2021 11:03	54155360
ajiant_windows_3.3.0_x64_full.msi	28-Dec-2021 11:03	67727360
ajiant_windows_3.3.0_x86_full.msi	28-Dec-2021 11:03	63311872

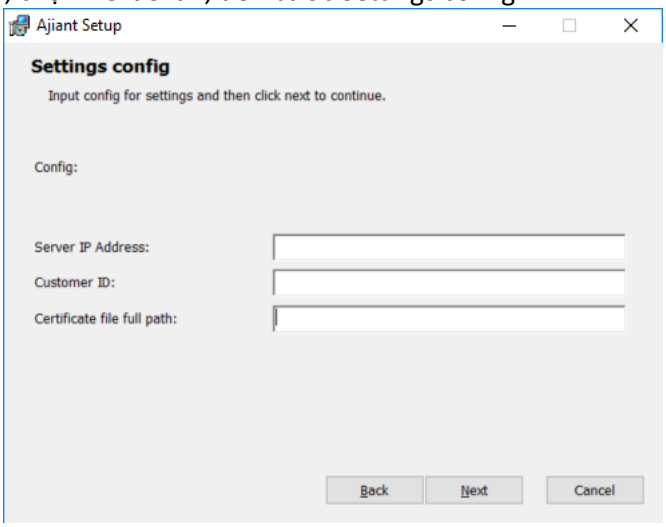
3.11. Hướng dẫn cài đặt các thành phần của mô hình MSSP

3.11.1. Cài đặt server forwarder

Liên hệ Phòng Hạ Tầng & Công Nghệ của VCS để lấy hướng dẫn triển khai server forwarder.

3.11.2. Cài đặt agent

Khi cài đặt agent MSSP, chọn Next 3 lần, đến bước Settings config:



Nhập các thông tin sau:

- Server IP Address: nhập ip forwarder
- Customer ID: nhập ID của khách hàng đã được cấp
- Certificate file full path: nhập đường dẫn ip forwarder được sinh ở bước trước

Các bước còn lại thực hiện như cũ.

3.12. Hướng dẫn thay certificate cho Portal

Chú ý: hướng dẫn thay certificate cho portal sau chỉ áp dụng với bộ cài AllInOne phiên bản 3.3.0

Các bước thực hiện thay certificate cho portal:

- **Bước 1:** Copy cert mới (bao gồm 1 file .key và 1 file .crt) vào thư mục **/opt/docker/tools/change_portal_cert/**
- **Bước 3:** vào thư mục change_portal_cert gõ lệnh:

```
$ cd /opt/docker/tools/change_portal_cert/
$ sudo bash change_portal_cert.sh
```

Output hiện "Nginx restarted" tức là đã thay cert thành công

4. HƯỚNG DẪN CÀI ĐẶT AGENT

4.1. Hướng dẫn cài đặt trên Windows

4.1.1. Yêu cầu đảm bảo cài đặt

Hệ điều hành:

Tương thích cài đặt với các máy Windows 7 SP1, Windows Server 2008 R2 và các phiên bản mới hơn.

Cấu hình phần cứng:

Yêu cầu tối thiểu:

- RAM 2GB
- CPU 1 core
- Dung lượng cài đặt 128MB

Kết nối mạng:

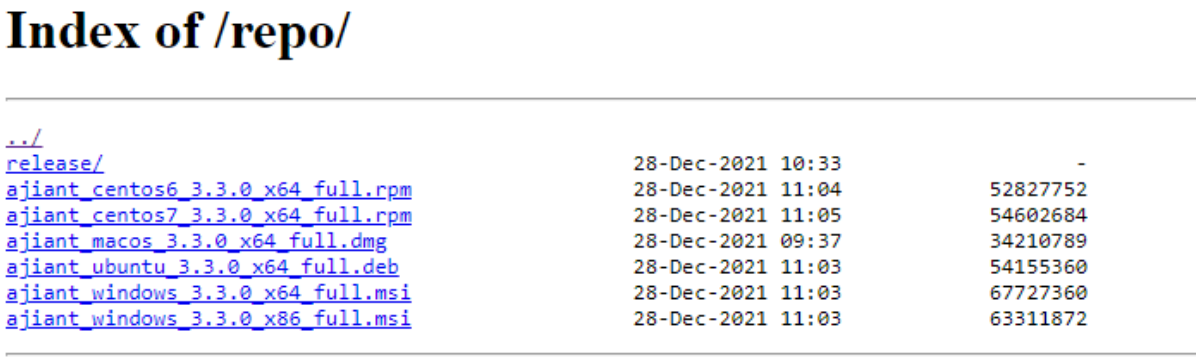
Đảm bảo thông kết nối tới hệ thống quản lý tập trung theo các port (4443, 5672, 8443, 8888)

4.1.2. Hướng dẫn cài đặt

a. Với bộ cài cho server on premise

Cài đặt thông qua hệ thống tập trung:

B1. Tải bộ cài agent (x86 và x64) thông qua địa chỉ https://<server-ip>/repo/ theo version mới nhất.



Index of /repo/

../		
release/	28-Dec-2021 10:33	-
ajiant_centos6_3.3.0_x64_full.rpm	28-Dec-2021 11:04	52827752
ajiant_centos7_3.3.0_x64_full.rpm	28-Dec-2021 11:05	54602684
ajiant_macos_3.3.0_x64_full.dmg	28-Dec-2021 09:37	34210789
ajiant_ubuntu_3.3.0_x64_full.deb	28-Dec-2021 11:03	54155360
ajiant_windows_3.3.0_x64_full.msi	28-Dec-2021 11:03	67727360
ajiant_windows_3.3.0_x86_full.msi	28-Dec-2021 11:03	63311872

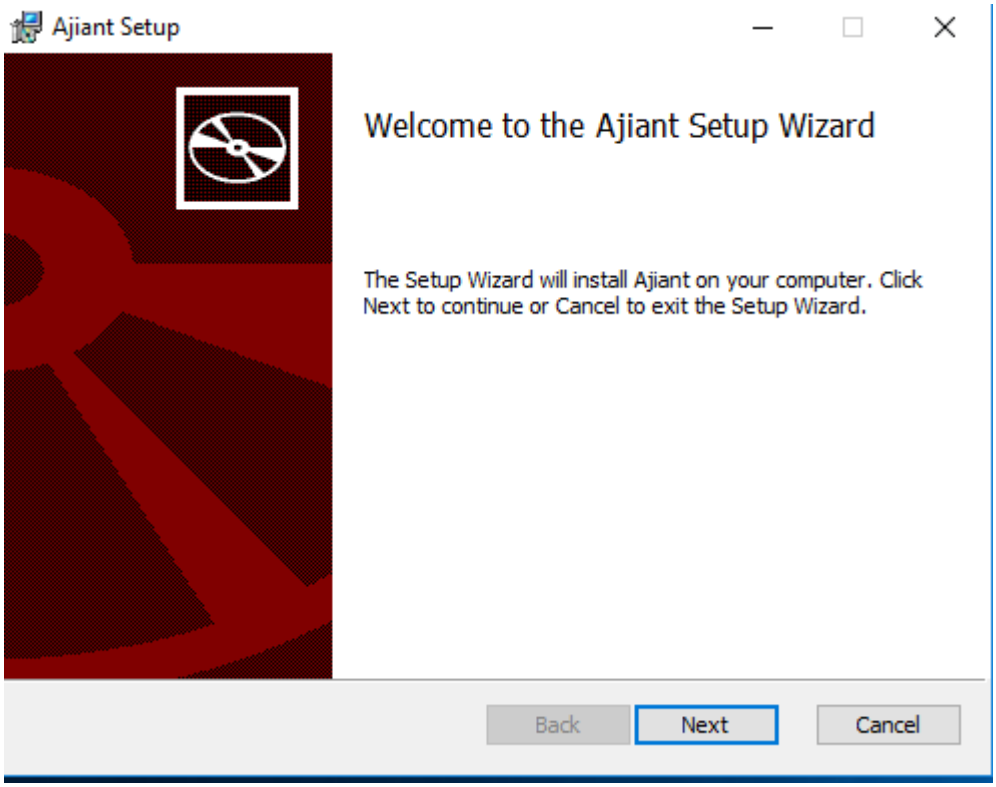
B2. Với các hệ thống quản trị tập trung AD thực hiện tạo package cài đặt từ xa cho bộ cài VCS-aJiant. Đẩy package xuống các máy cần cài đặt. Cấu hình tham số cài đặt cho bộ cài VCS-aJiant như sau:

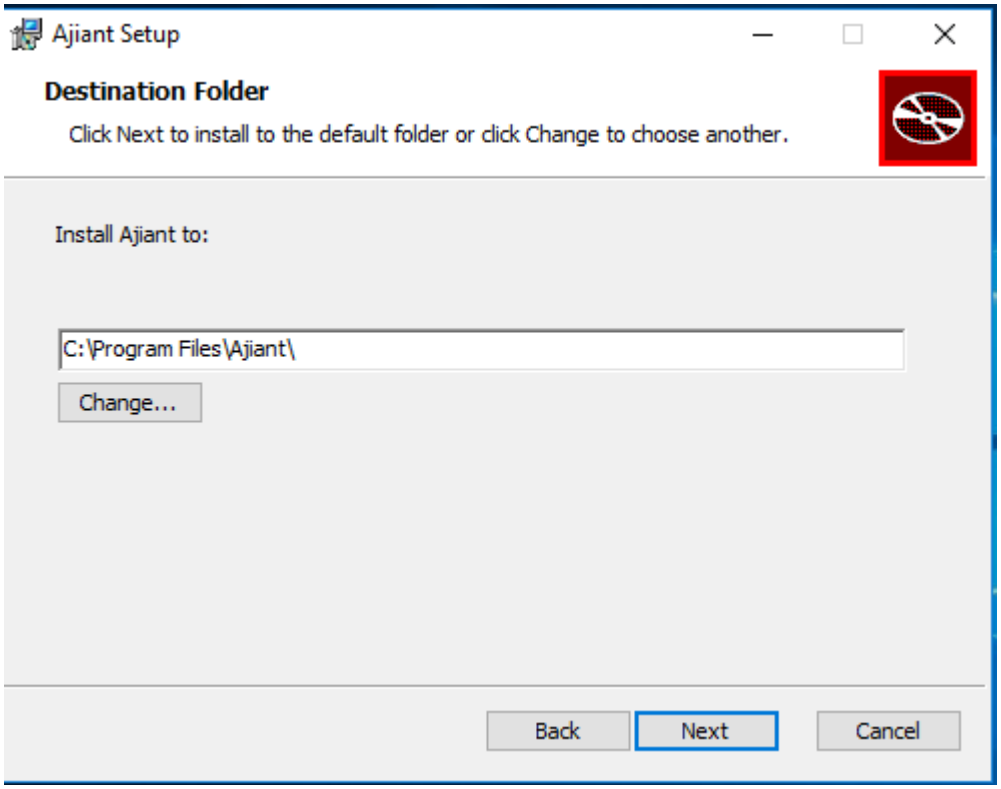
Msiexec.exe /i <setup_file>.msi /qn

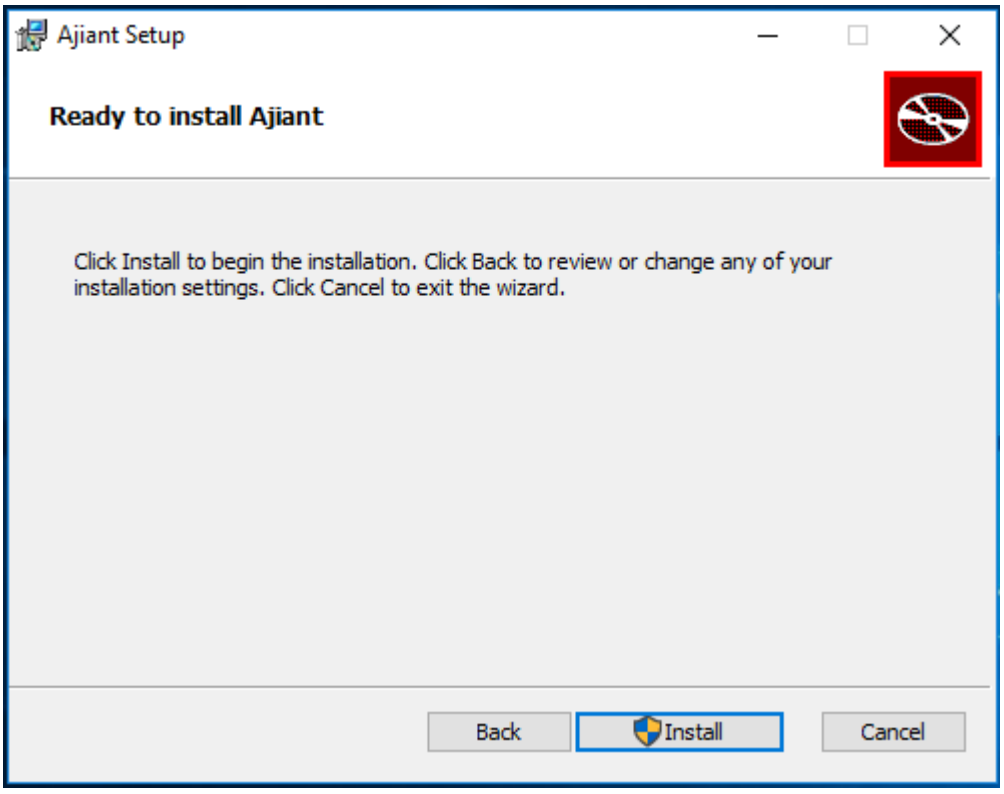
Cài đặt trực tiếp:

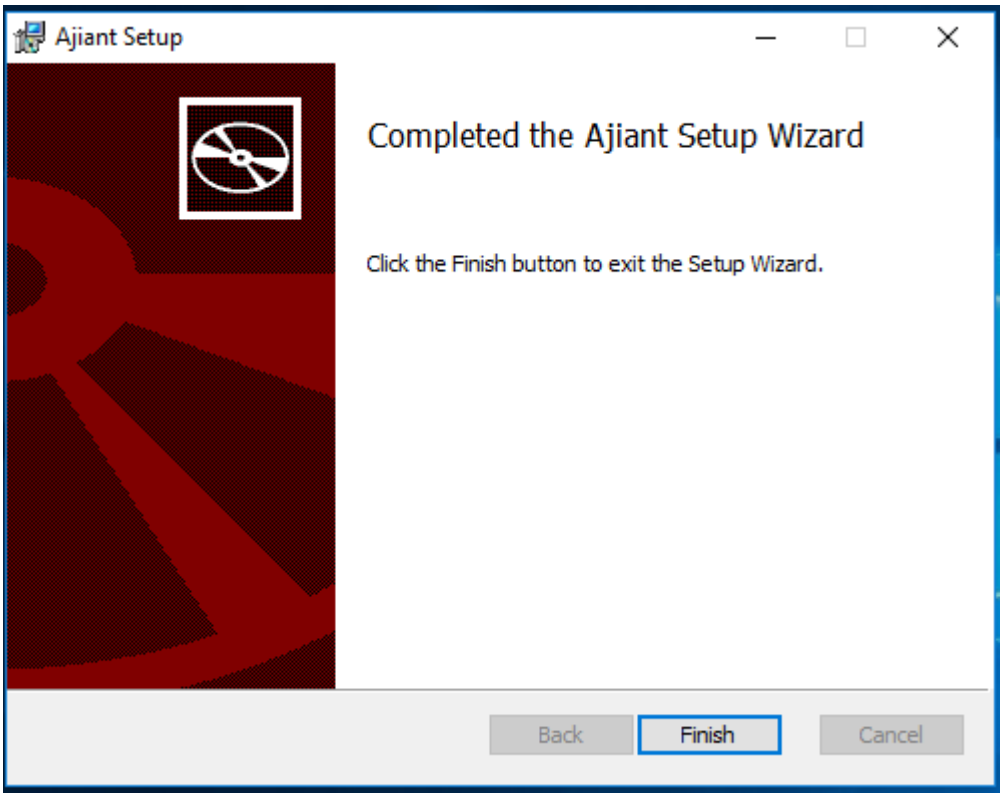
B1. Tải bộ cài (x86 và x64) thông qua địa chỉ https://<server-ip>/repo/ theo version mới nhất.

B2. Click đúp bộ cài, tiến hành cài đặt theo các bước (cần quyền administrator để thực hiện cài đặt)




Ajiant Setup
Destination Folder
Click Next to install to the default folder or click Change to choose another.
Install Ajiant to:
C:\Program Files\Ajiant\
Change...
Back
Next
Cancel


Ajiant Setup
Ready to install Ajiant
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel



b. Với bộ cài cho server MSSP

Cài đặt thông qua hệ thống tập trung:
B1. Tải bộ cài agent (x86 và x64) thông qua địa chỉ https://<server-ip>/repo/ theo version mới nhất.

B2. Với các hệ thống quản trị tập trung AD thực hiện tạo package cài đặt từ xa cho bộ cài VCS-aJiant. Đẩy package và file certificate xuống các máy cần cài đặt. Cấu hình tham số cài đặt cho bộ cài VCS-aJiant như sau:
Msiexec.exe /i <setup_file>.msi IP="<Server_IP>" ID="<Customer_ID>" CERT="<Cert_File_FullPath>" /qn

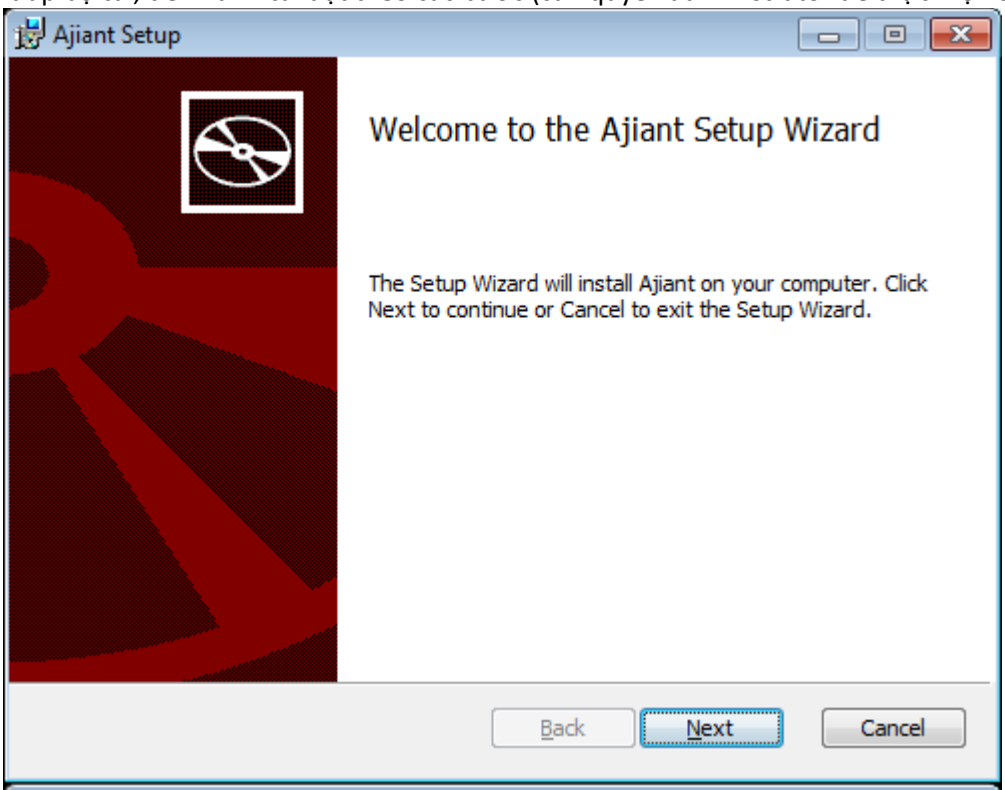
Trong đó:

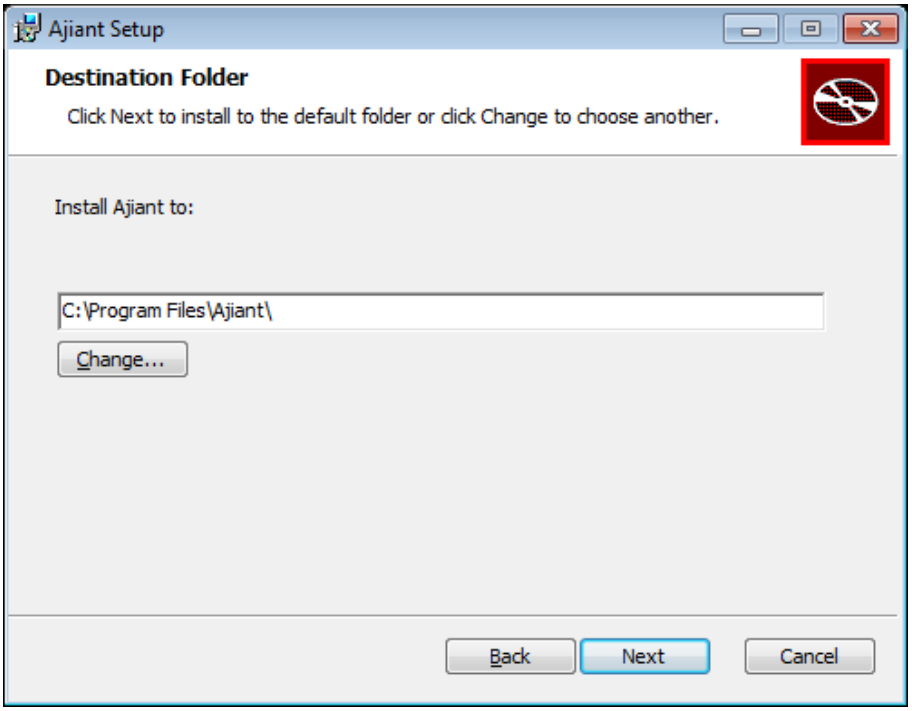
- Setup_file: là tên file msi được đẩy xuống máy
- Server_IP: là địa chỉ IP của server forwarder
- Customer_ID: là ID của khách hàng, được cung cấp sẵn
- Cert_File_FullPath: là đường dẫn fullpath của file certificate.

Ví dụ: *Msiexec.exe /i Ajiant_3.3.0_x64.msi IP="1.2.3.4" ID="VCS" CERT="c:\certificate\vcs.crt" /qn*

Cài đặt trực tiếp:

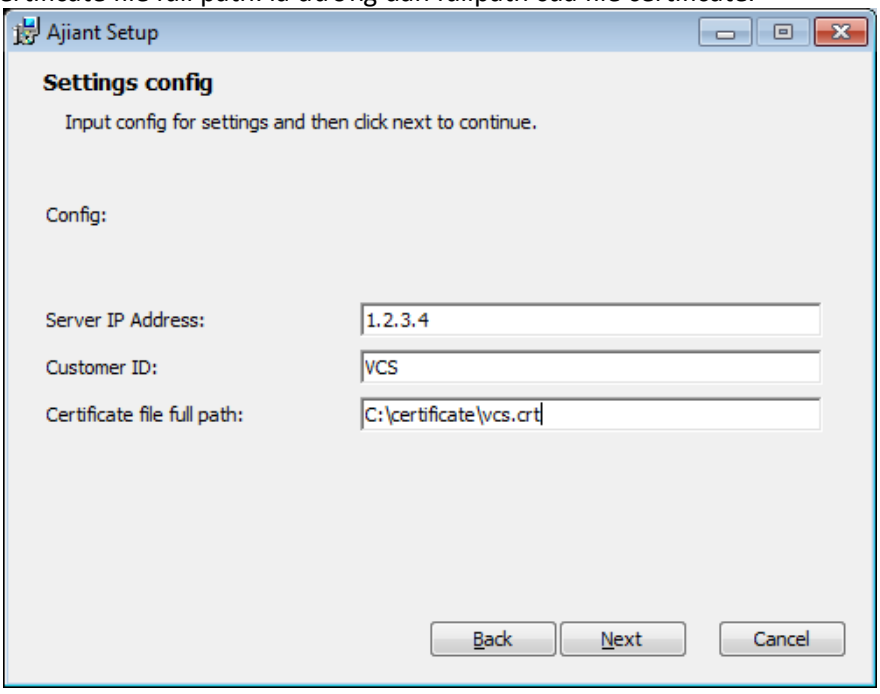
B1. Tải bộ cài (x86 và x64) thông qua địa chỉ https://<server-ip>/repo/ theo version mới nhất.
B2. Click đúp bộ cài, tiến hành cài đặt theo các bước (cần quyền administrator để thực hiện cài đặt)

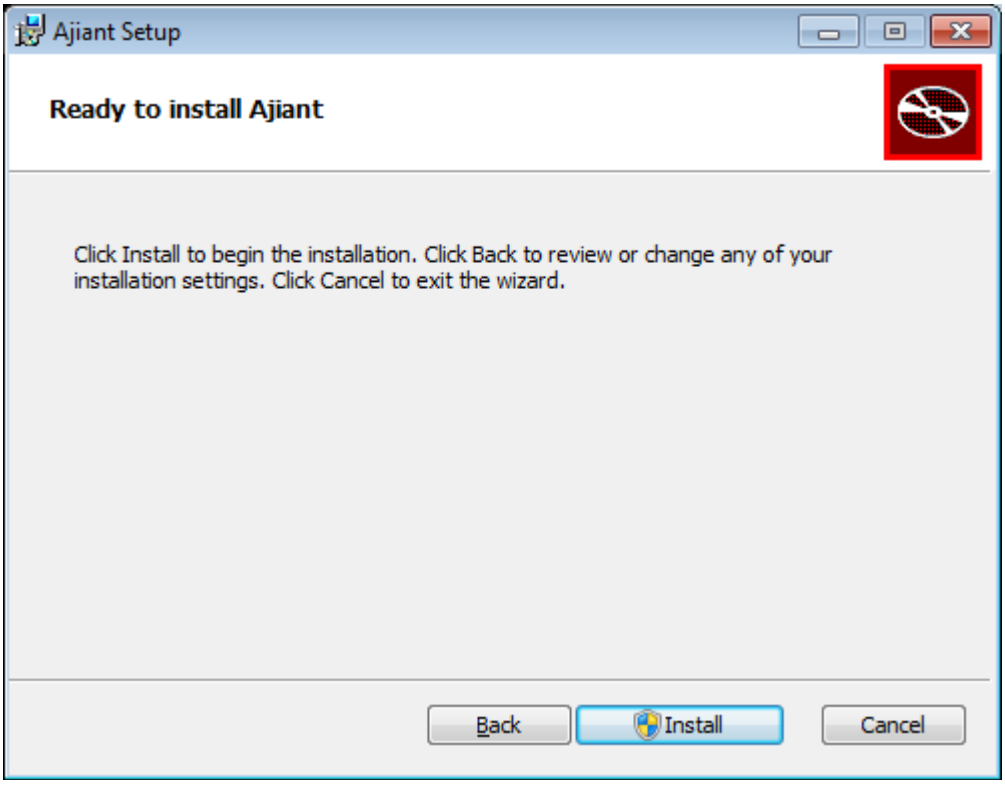


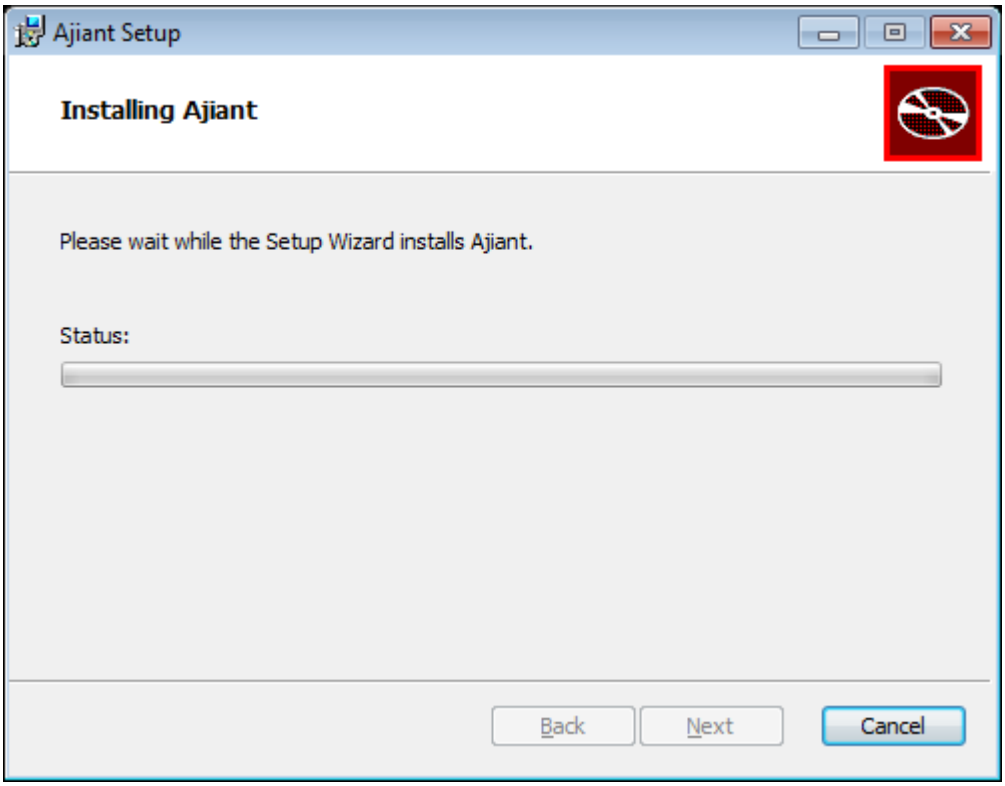


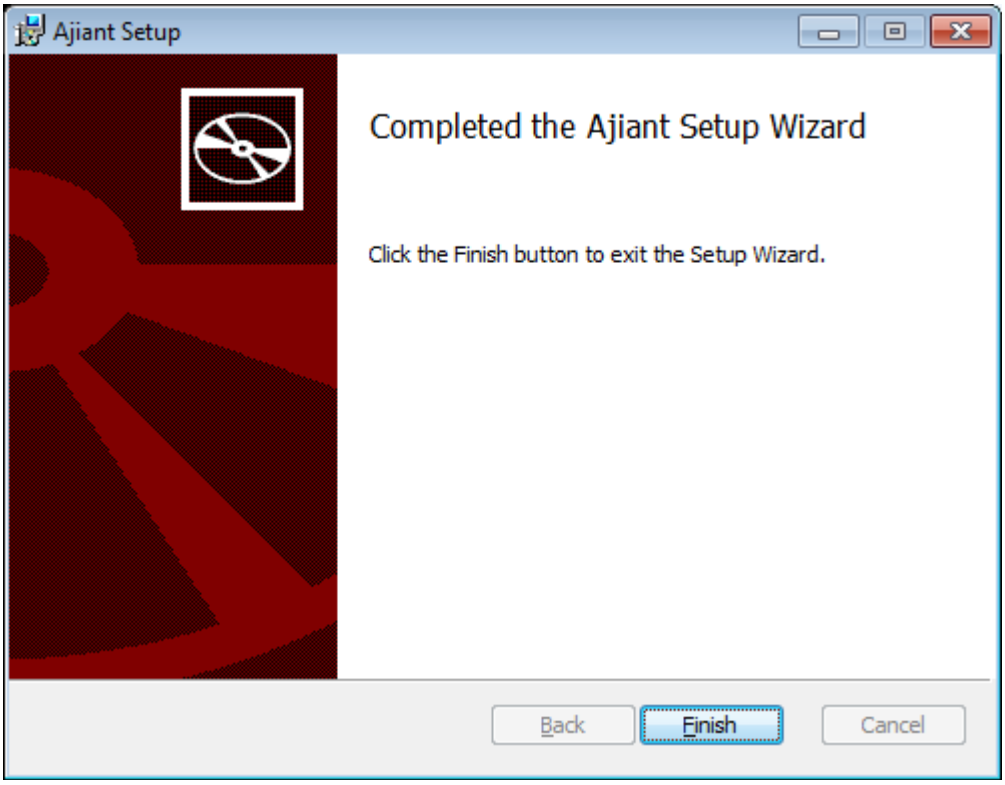
Nhập các thông tin cài đặt:

- Server IP Address: là địa chỉ IP của server forwarder
- Customer ID: là ID của khách hàng, được cung cấp sẵn
- Certificate file full path: là đường dẫn fullpath của file certificate.



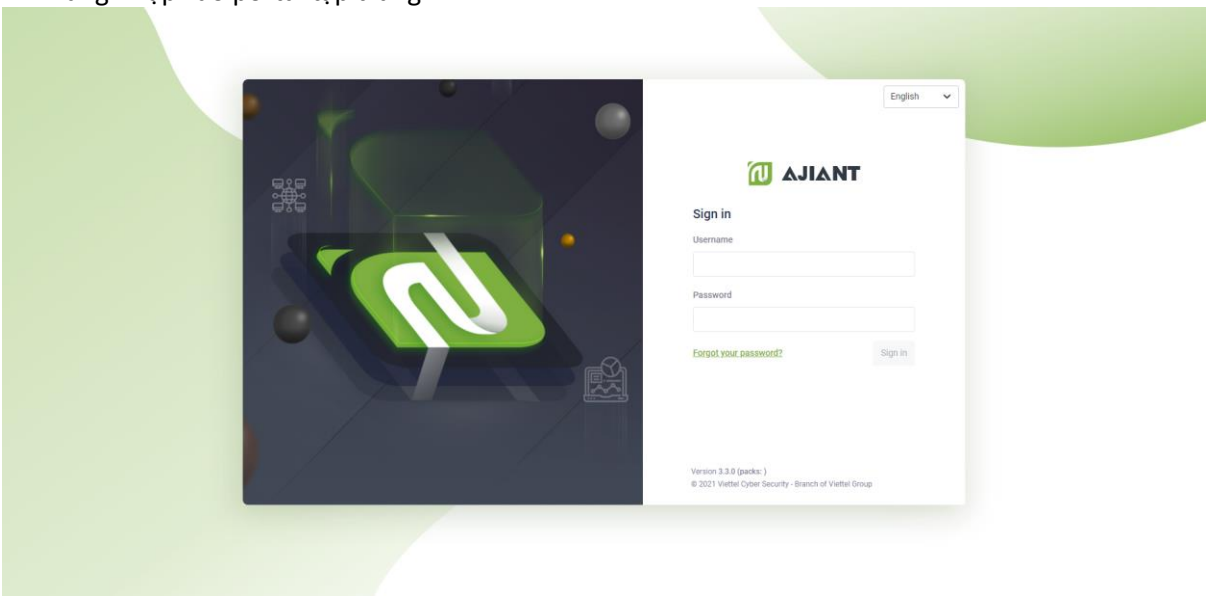

Ajiant Setup
Ready to install Ajiant
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel


Ajiant Setup
Installing Ajiant
Please wait while the Setup Wizard installs Ajiant.
Status:
Back
Next
Cancel

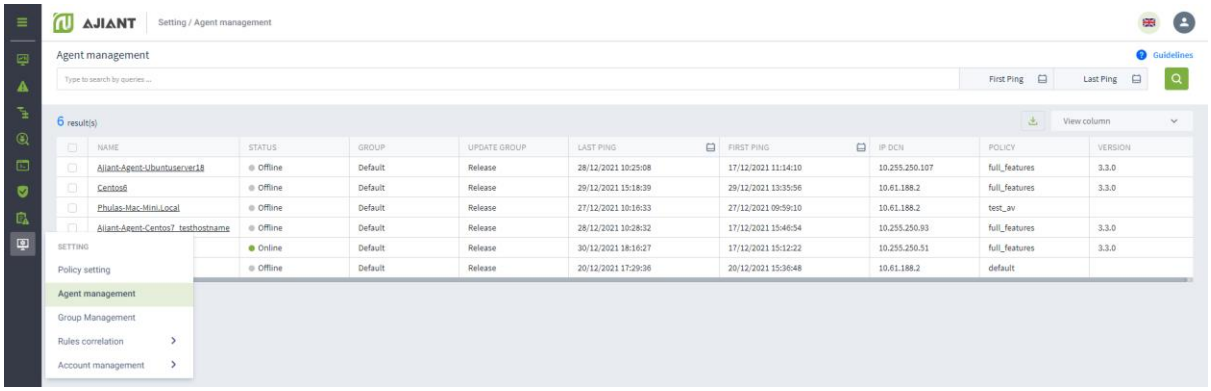


4.1.3. Kiểm tra cài đặt

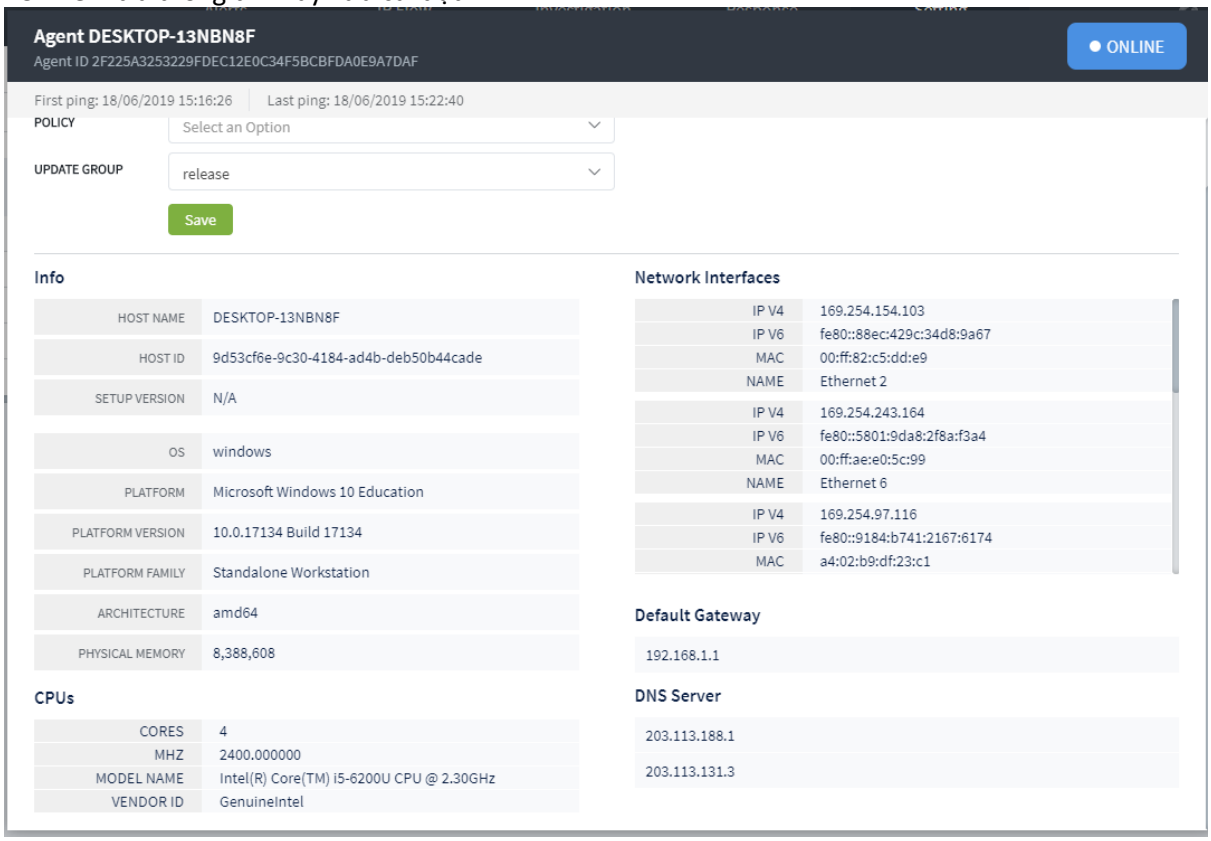
B1. Đăng nhập vào portal tập trung



B2. Vào mục quản lý agent (Agent Management)



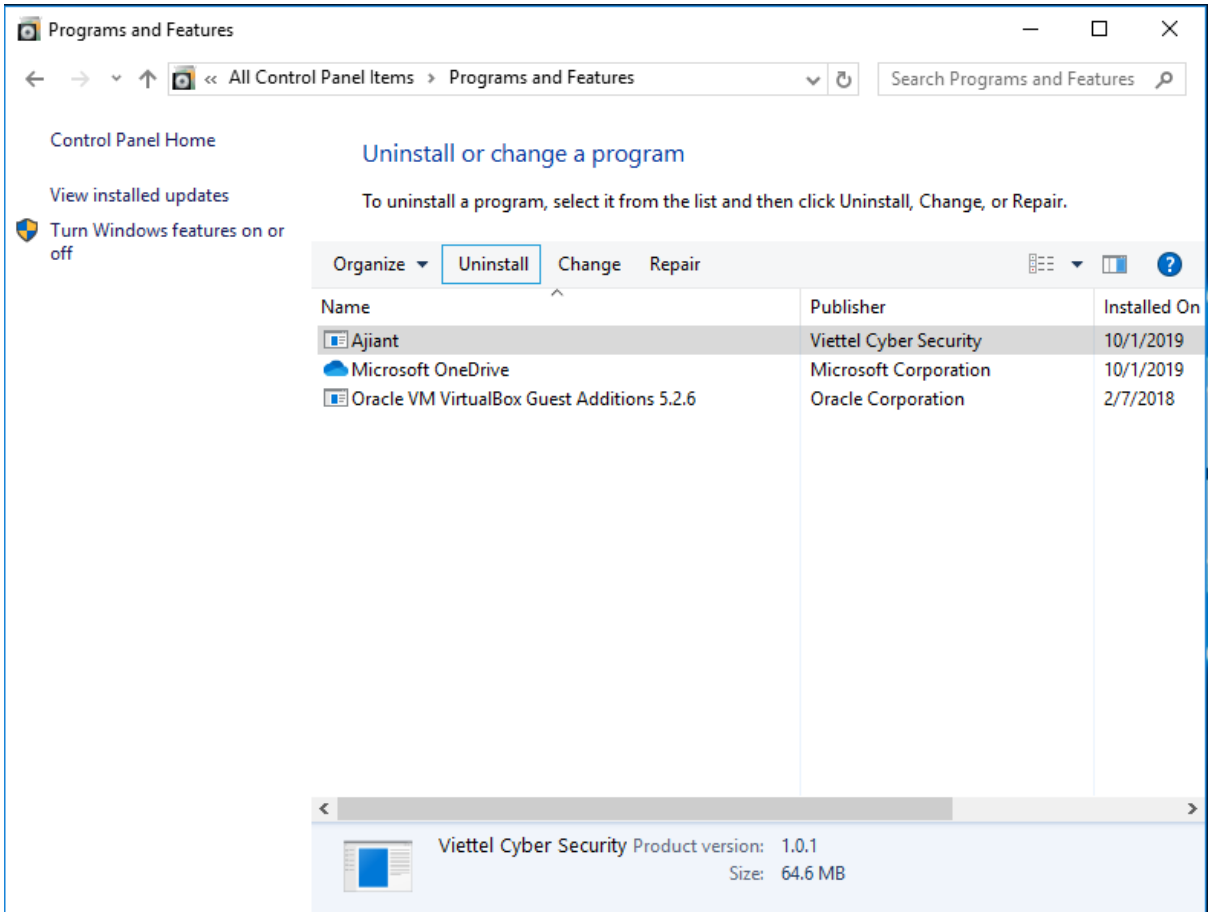
B3. Kiểm tra thông tin máy vừa cài đặt



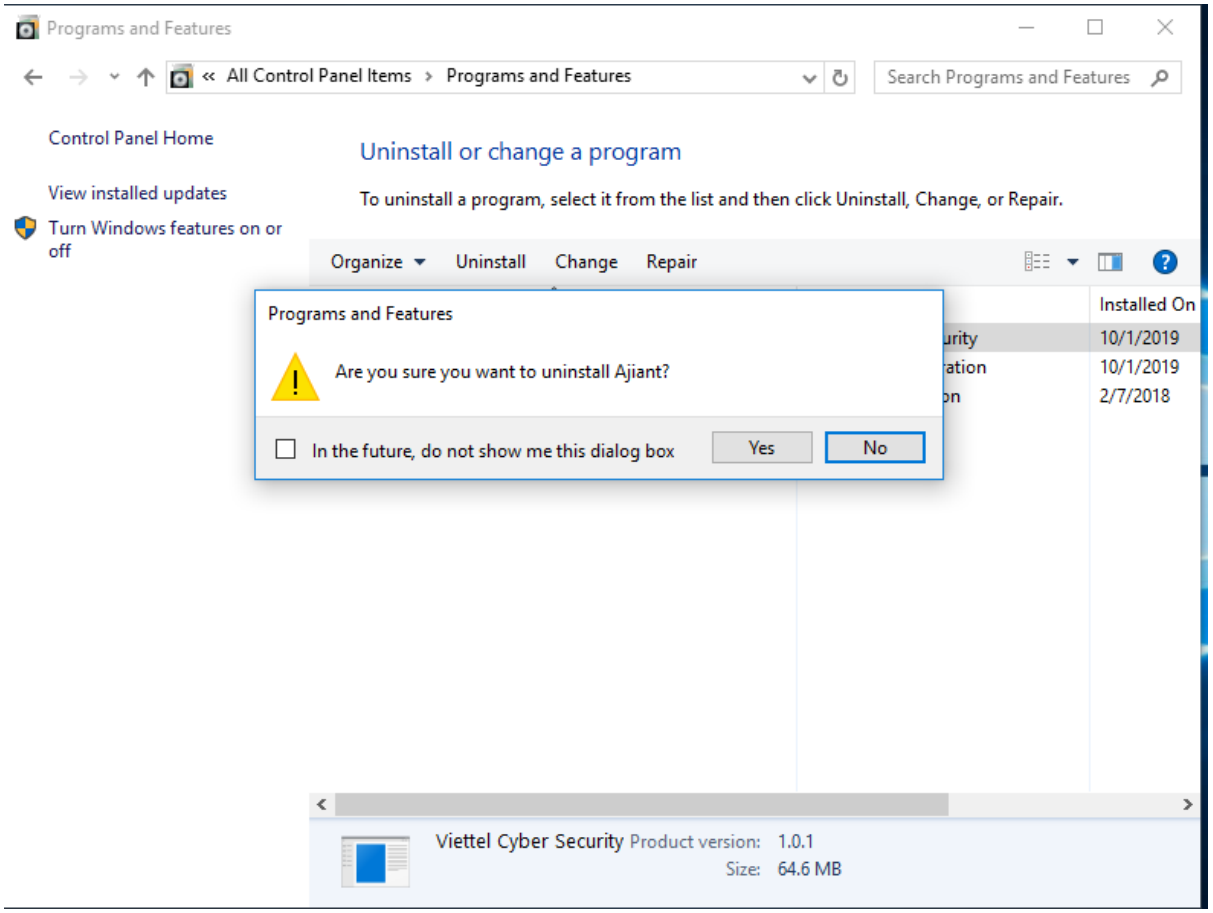
4.1.4. Hướng dẫn gỡ cài đặt

<u>Gỡ cài đặt thông qua hệ thống tập trung:</u>
Với các hệ thống quản trị tập trung AD thực hiện đẩy lệnh gỡ bỏ agent xuống các máy như sau:
wmic product where name="Ajiant" call uninstall

<u>Gỡ cài đặt trực tiếp</u>:
Trong trình quản lý **Programs and Features** của Windows thực hiện chọn gỡ bỏ agent



Thực hiện theo hướng dẫn để hoàn tất quá trình gỡ bỏ



4.2. Hướng dẫn cài đặt trên Ubuntu và CyOS

4.2.1. Yêu cầu đảm bảo cài đặt

Phiên bản OS: tương thích cài đặt với các máy Ubuntu 18.04 x64 Desktop, Ubuntu 18.04 x64 Server, CyOS x64

Phiên bản linux kernel: 4.15, 4.18, 5.0
Các phiên bản khác, VCS-aJiant sẽ thiếu hai tính năng gồm Event Log và Network Containment

Cấu hình phần cứng:

Yêu cầu tối thiểu:

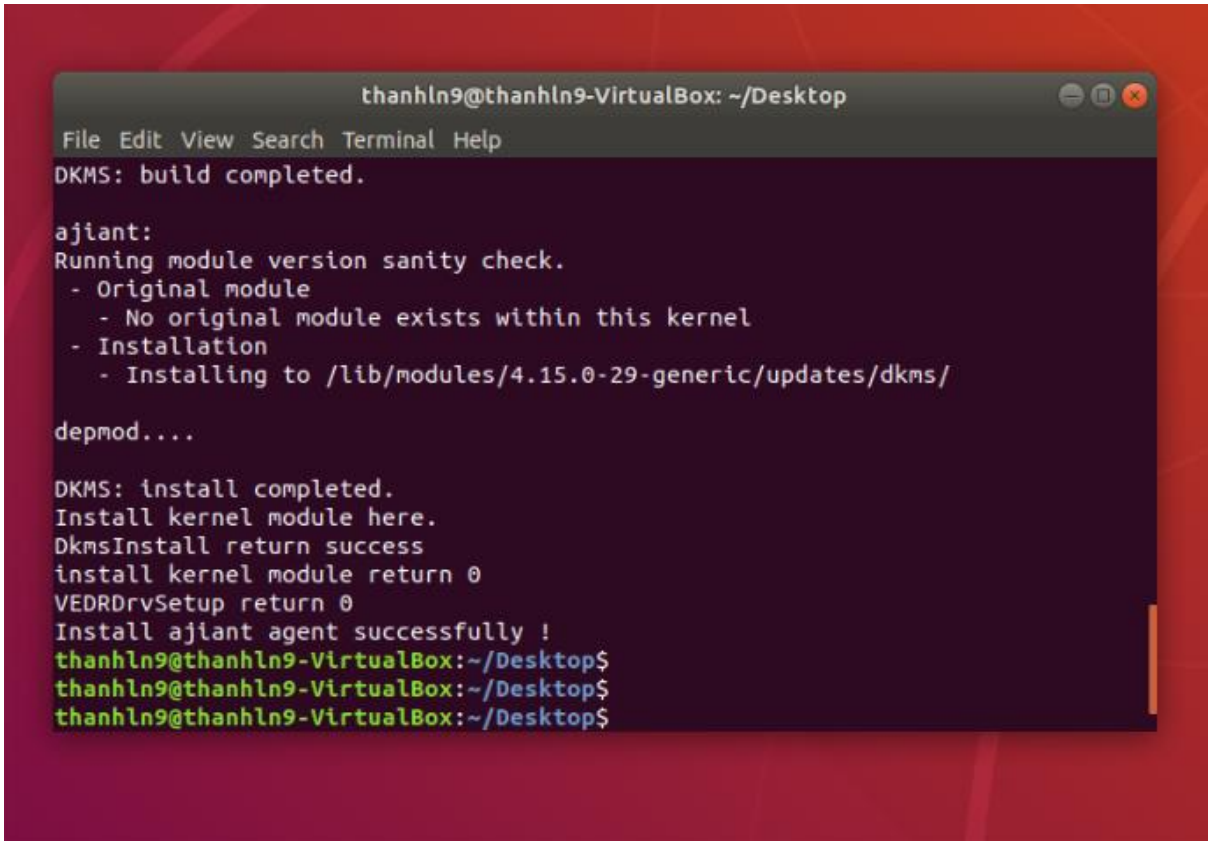
- RAM 2GB
- CPU 1 core
- Dung lượng cài đặt 128MB

Kết nối mạng:

Đảm bảo thông kết nối tới hệ thống quản lý tập trung theo các port (4443, 5672, 8443, 8888).

4.2.2. Hướng dẫn cài đặt

B1. Tải bộ cài agent ubuntu thông qua địa chỉ https://<server-ip>/repo/ theo version mới nhất.
B2. Chạy lệnh sau để cài đặt: *sudo dpkg -i <đường dẫn tới package>*
Ví dụ: *sudo dpkg -i ajiant_ubuntu_3.3.0_x64.deb*

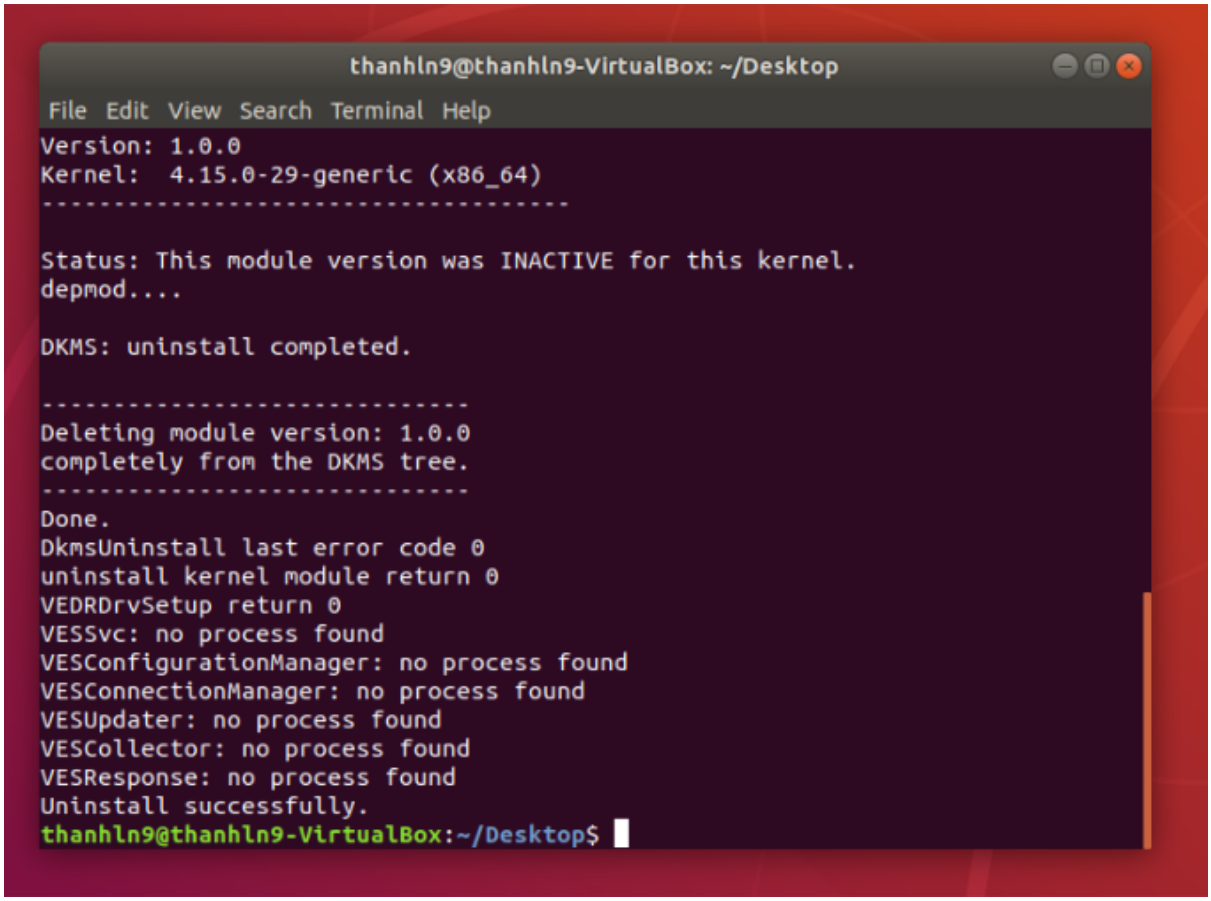


4.2.3. Kiểm tra cài đặt

Tương tự phần Windows

4.2.4. Hướng dẫn gỡ cài đặt

Chạy lệnh sau để gỡ cài đặt: *sudo dpkg -r ajiant*



4.3. Hướng dẫn cài đặt trên CentOS6 và CentOS7

4.3.1. Yêu cầu đảm bảo cài đặt

Phiên bản OS: tương thích cài đặt với các máy Centos6 và CentOS7

Phiên bản linux kernel:

- CentOS6: 2.6.32-754
- CentOS7: 3.10.0-1160, 3.10.0-1127, 3.10.0-1062, 3.10.0-957, 3.10.0-862, 3.10.0-693, 3.10.0-514

Các phiên bản khác, VCS-aJiant sẽ thiếu hai tính năng gồm Event Log và Network Containment

Cấu hình phần cứng:

Yêu cầu tối thiểu:

- RAM 2GB
- CPU 1 core
- Dung lượng cài đặt 128MB

Kết nối mạng:

Đảm bảo thông kết nối tới hệ thống quản lý tập trung theo các port (4443, 5672, 8443, 8888).

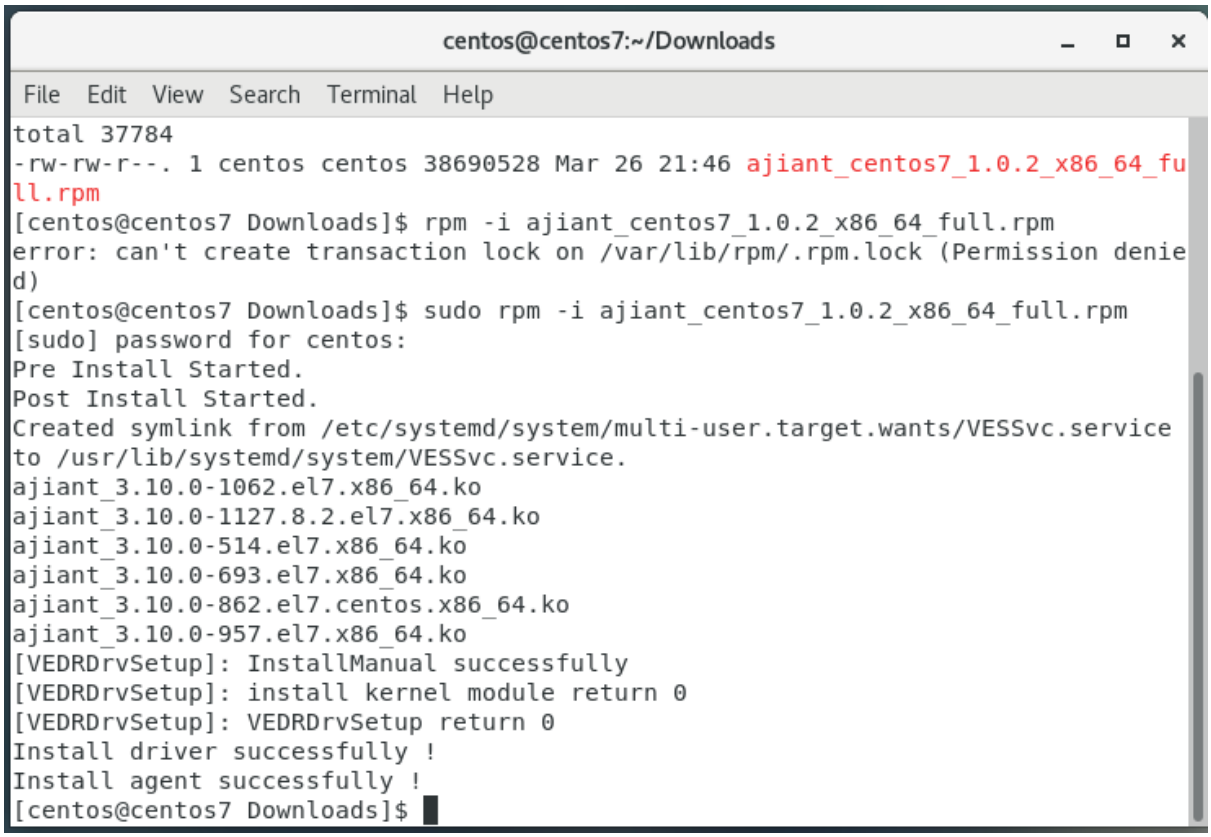
4.3.2. Hướng dẫn cài đặt

Cài đặt bằng dòng lệnh:

B1. Tải bộ cài linux thông qua địa chỉ https://<server-ip>/repo/ theo version mới nhất.

B2. Chạy lệnh sau để cài đặt: *sudo rpm -i <đường dẫn tới package>*

Ví dụ: *sudo rpm -i ajiant_centos7_3.3.0_x64_full.rpm*

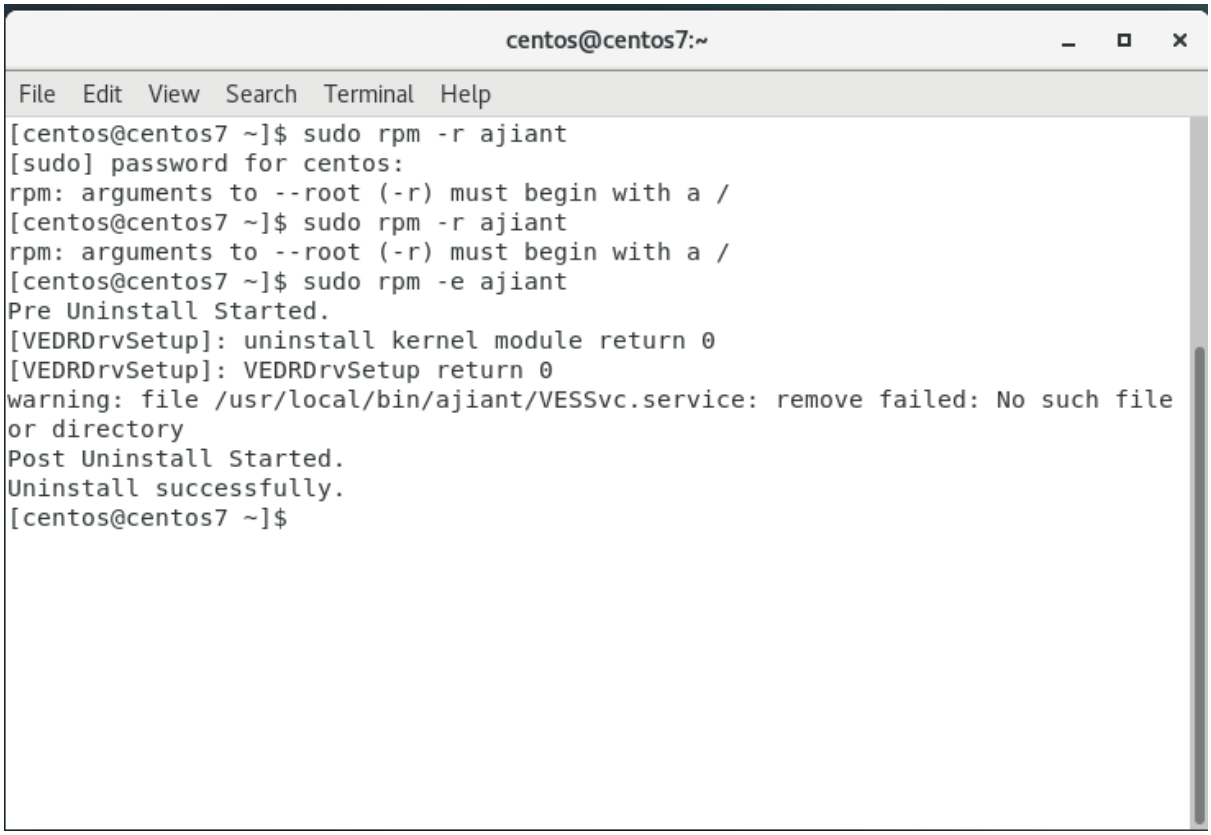


4.3.3. Kiểm tra cài đặt

Tương tự phần Windows

4.3.4. Hướng dẫn gỡ cài đặt

Chạy lệnh sau để gỡ cài đặt: *sudo rpm -e ajiant*



4.4. Hướng dẫn cài đặt trên MacOS

4.4.1. Yêu cầu đảm bảo cài đặt

Phiên bản OS: tương thích cài đặt với các máy MacOS Catalina, Bigsur, Monterey

Cấu hình phần cứng: dùng cho Mac mini hoặc macbook đời 2016 trở về sau

Kết nối mạng:

Đảm bảo thông kết nối tới hệ thống quản lý tập trung theo các port (4443, 5672, 8443, 8888).

4.4.2. Hướng dẫn cài đặt

Hướng dẫn disable SIP để có thể chạy được các ứng dụng ngoài App Store.

B1. Restart lại máy đến khi thấy màn đen, bấm giữ phím **Command + R** để vào chế độ Recovery. Với Mac M1 thì sau khi restart, giữ phím nguồn để vào chế độ Recovery.

B2. Sau khi máy đã vào chế độ Recovery, chọn mục **Utilities -> Terminal**

B3. Gõ lệnh **csrutil disable**

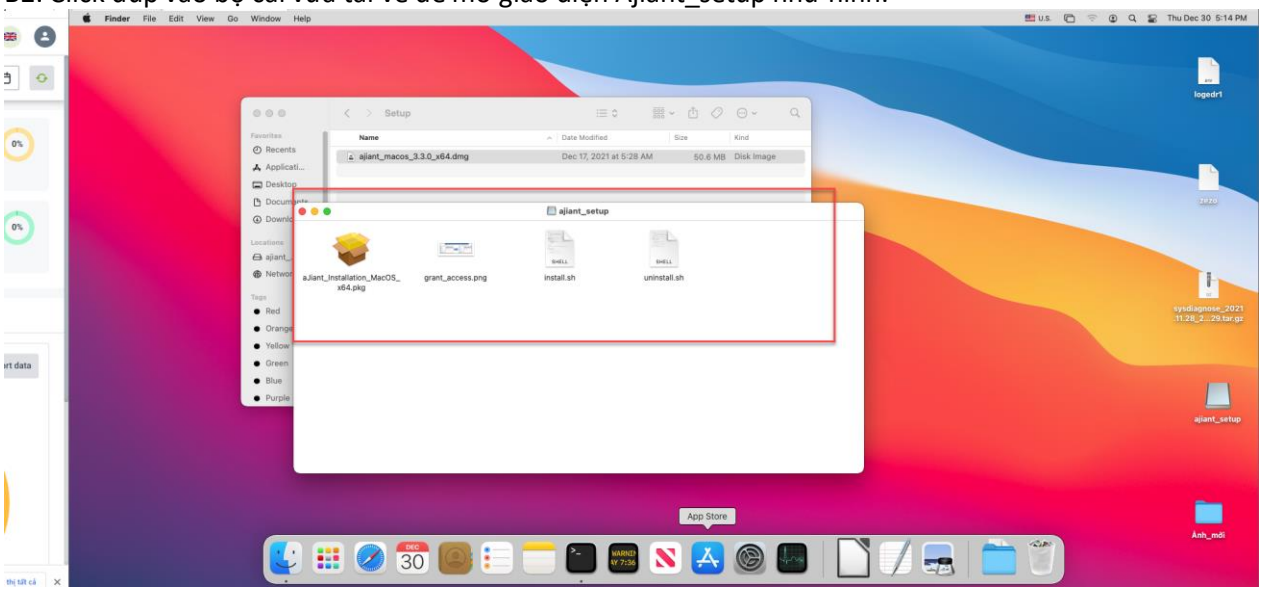
B4. Enter và máy hiển thị **System Integrity Protection is off** là đã thành công.

B5. Gõ reboot để máy khởi động lại

Cài đặt bằng dòng lệnh:

B1. Tải bộ cài MacOS thông qua địa chỉ https://<server_ip>/repo/ theo version mới nhất.

B2. Click đúp vào bộ cài vừa tải về để mở giao diện Ajiant_setup như hình:



B3: Mở ứng dụng Termial, gõ lệnh theo cấu trúc:

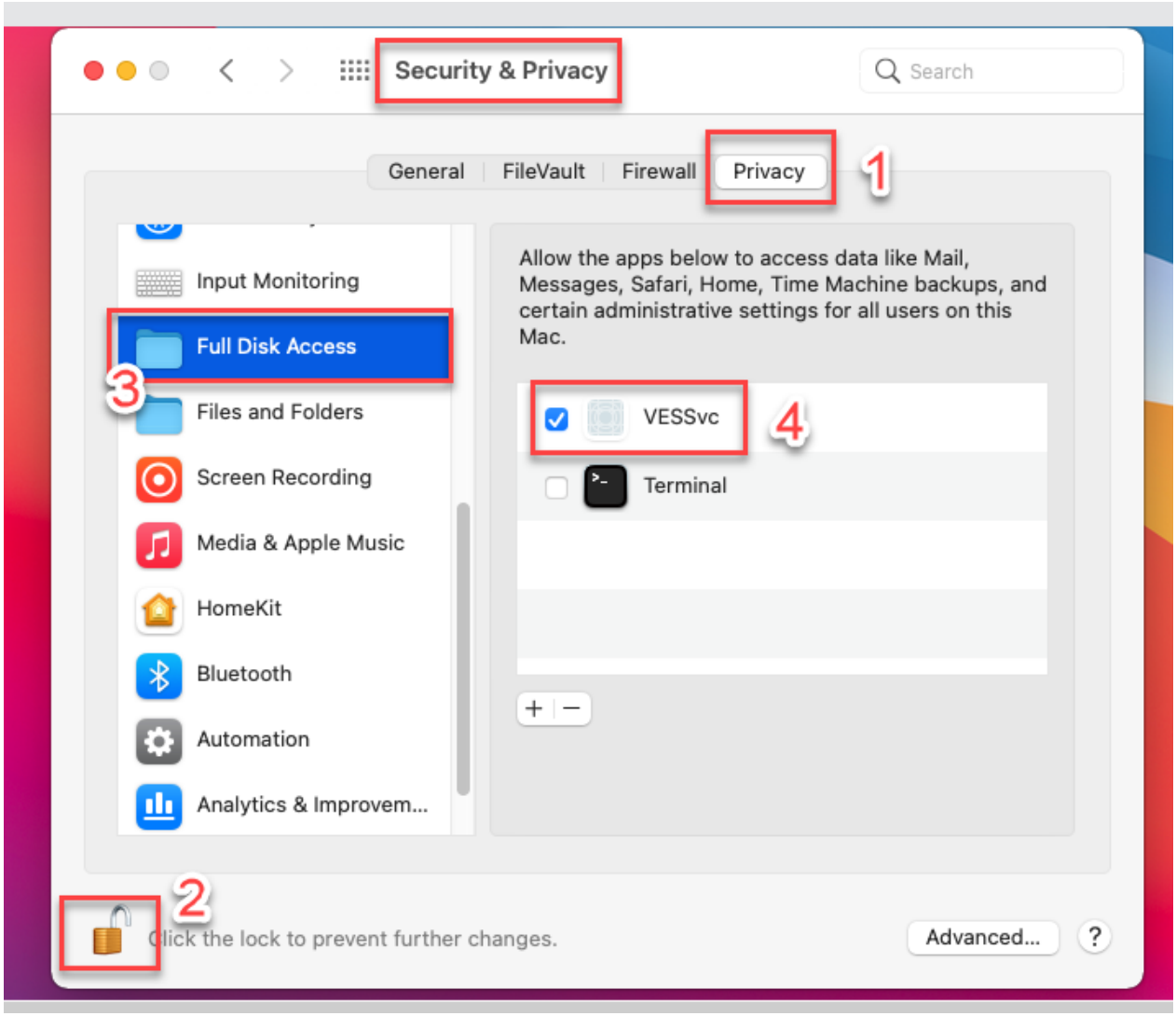
sudo bash ./install.sh [server_address] [update_port]

Lưu ý: trong trường hợp thay đổi update_port thì cần phải nhập thêm địa chỉ update_port đang thiết lập. Nếu không thay đổi update port mặc định là 8443 thì không cần nhập địa chỉ update port.

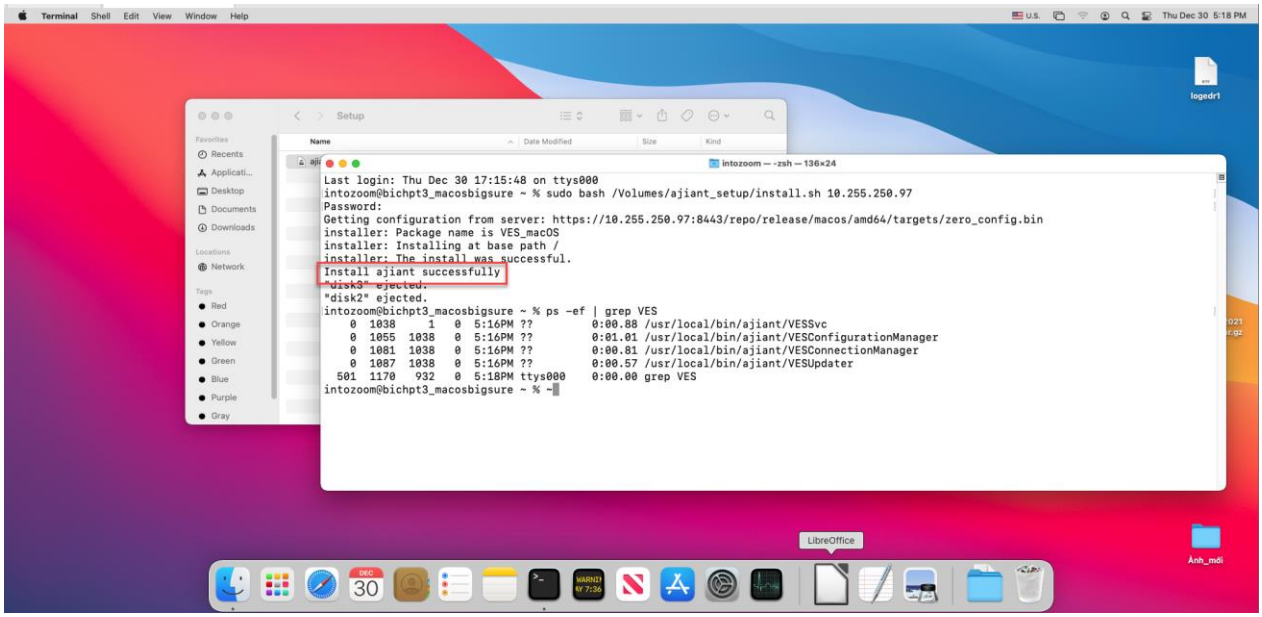
Ví dụ: *sudo bash /Volumes/ajiant_setup/install.sh 10.30.161.3*

B4. Cấp quyền cho phép phần mềm Ajiant được phép truy cập các thư mục:

Mở ứng dụng System Preferences/ Security & Privacy chọn lần lượt thứ tự các bước từ 1 đến 4 như hình:



Cài đặt hoàn tất sẽ hiển thị thông báo thành công ngay trên cửa sổ Terminal.

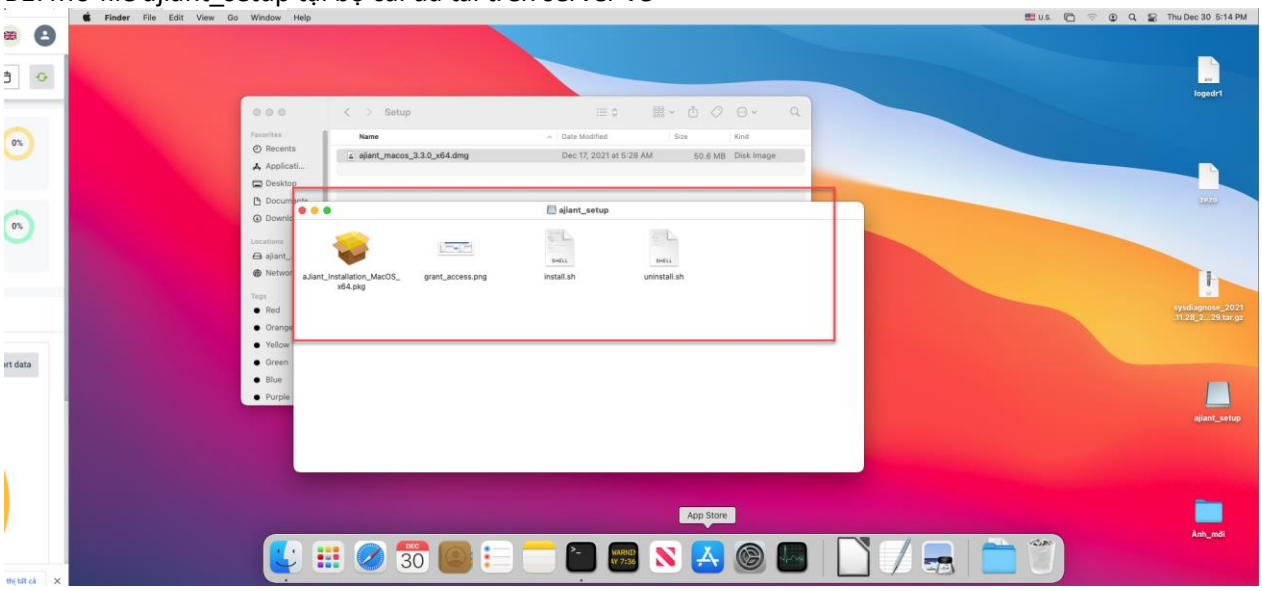


4.4.3. Kiểm tra cài đặt

Tương tự phần Windows

4.4.4. Hướng dẫn gỡ cài đặt

B1: Mở file ajiant_setup tại bộ cài đã tải trên server về



B2: Mở terminal chạy file uninstall.sh để gỡ cài đặt.

Lệnh: *sudo bash /Volumes/ajiant_setup/uninstall.sh*

5. KHẮC PHỤC SỰ CỐ

5.1. Các lỗi thường gặp khi cài đặt và nâng cấp backend

5.1.1. Lỗi build bộ cài agents

Quá trình sinh bộ cài MSI có thể xem log như sau:

```
$ tailf /opt/docker/agentbuild/Win7x86/Logs/VBox.log
```

Lỗi build bộ cài agent có thể do các nguyên nhân sau:

- Cài virtual box lỗi: cần cài đặt lại gói virtual box và gói mở rộng (extension pack)
- Giải nén file máy ảo bị lỗi: kiểm tra lại file máy ảo copy vào vào server (**Win7x86.tar.gz có kích thước 2.4GB**) và file sau khi giải nén (**Win7x86.vdi có kích thước khoảng 5GB**)
- Server chưa được bật tính năng ảo hoá Intel VT-x/AMD-V.

5.1.2. Không đăng nhập được portal

Tham khảo tài liệu Admin Guide mục 5.2.4

5.2. Khôi phục hệ thống

5.2.1. Khôi phục hệ thống sau khi nâng cấp gặp lỗi

Mô hình AllinOne	**Mô hình MultiNode**
Thay lại file *docker-compose.yml* vào ***/opt/docker/docker-compose.yml***	Thay lại file *ajiant-stack.yml* vào **/opt/ajiant/ajiant-stack.yml**
Chạy lệnh:	Chạy lệnh:
`$ docker-compose up –d`	`$ ./DeployStack.sh`
Kiểm tra lại:	
`$ ./list-containers.sh`	
Xem các service có ở trạng thái "**Up**" không	
Chờ một thời gian và kiểm tra lại các thông tin sau: - Portal có đăng nhập được không ? - Có agent online không ? - Có thấy log event gửi lên không ? - Kiểm tra rabbitmq management xem các log có được xử lý hết không ?	

5.2.2. Khôi phục toàn bộ (rollback)

Nếu hệ thống backend gặp lỗi không thể phục hồi lại (lỗi phần cứng server, lỗi ổ cứng) thì cần phải cài đặt lại hệ thống backend với các file cert và licence đã lưu lại ở bước cài đặt:

/opt/docker/nginx/certs/cert.crt

/opt/docker/nginx/certs/cert.crt

Các bước thực hiện tương tự mục 3.4 và 3.5 với file cert đã có sẵn

5.3. Thông tin đầu mối hỗ trợ

- Trung tâm giám sát và phản ứng trên không gian mạng, Công ty An ninh mạng Viettel

  Email: **soc247@viettel.com.vn**

- Bộ phận chăm sóc khách hàng, Công ty An ninh mạng Viettel

  Email: **cskh_anm@viettel.com.vn**