# Tổng quan

## ****IaaS Compute Cloud là gì?****

**IaaS (Infrastructure as a Service)** là mô hình điện toán đám mây trong đó nhà cung cấp cung cấp tài nguyên hạ tầng **ảo hóa** như:

* Máy ảo (VM)
* CPU, RAM, Disk
* Mạng, IP, tường lửa
* Load balancer, snapshot, v.v...

Người dùng không cần lo phần cứng vật lý, chỉ cần khởi tạo và quản lý các tài nguyên này qua web hoặc API.

### ****Private IaaS Compute Cloud****

* **Định nghĩa:**

Private IaaS là môi trường đám mây **triển khai trong nội bộ doanh nghiệp**, dành riêng cho tổ chức đó. Doanh nghiệp **toàn quyền kiểm soát hạ tầng** và chỉ cấp quyền truy cập cho các nhóm/nhân viên nội bộ.

* Đặc điểm:
* Triển khai trên server vật lý trong doanh nghiệp
* Tường lửa và mạng riêng, không chia sẻ với bên ngoài
* Bảo mật cao, kiểm soát truy cập chặt chẽ
* Tùy biến theo yêu cầu doanh nghiệp
* **Ví dụ:**

Công ty XYZ có 5 máy chủ KVM tại văn phòng → Cài CloudStack → Cấp quyền cho các nhóm như Dev, QA tự tạo VM để test phần mềm.

* **Dev** có thể tạo VM Ubuntu để chạy pipeline
* **QA** có thể tạo máy Windows để test tính năng
* Tài nguyên, mạng được cô lập, không ai ngoài công ty truy cập được

➡️ **Đây là Private IaaS.**

## ****Public IaaS Compute Cloud****

### 📌 Định nghĩa:

Public IaaS là mô hình trong đó **nhà cung cấp dịch vụ cloud (VD: AWS, Azure, DigitalOcean)** sở hữu hạ tầng vật lý và **bán hoặc cho thuê tài nguyên hạ tầng qua internet** đến khách hàng.

### ✔️ Đặc điểm:

* Triển khai quy mô lớn, dùng chung hạ tầng vật lý
* Tính đàn hồi cực cao: tạo hàng trăm VM trong vài phút
* Truy cập qua internet, sử dụng theo mô hình trả phí theo giờ hoặc theo tài nguyên

### 💡 Ví dụ:

Bạn tạo 1 tài khoản AWS → chọn tạo EC2 (VM) → chọn OS, RAM, IP → bấm "Launch" → có VM chạy ngay.

* Dùng IP công cộng để SSH vào
* Chạy web app, web server, hoặc cluster
* Trả tiền theo giờ hoặc tháng

➡️ **Đây là Public IaaS.**

## 🆚 ****4. So sánh nhanh Public vs Private IaaS****

| **Tiêu chí** | **Private Cloud** | **Public Cloud** |
| --- | --- | --- |
| Hạ tầng | Doanh nghiệp tự sở hữu | Nhà cung cấp sở hữu |
| Truy cập | Nội bộ (VPN, mạng LAN) | Qua Internet |
| Bảo mật | Cao, kiểm soát toàn diện | Phụ thuộc nhà cung cấp |
| Chi phí | Đầu tư ban đầu lớn | Trả theo tài nguyên dùng |
| Quy mô mở rộng | Giới hạn bởi phần cứng | Gần như vô hạn |
| Ví dụ | CloudStack trên server nội bộ | AWS EC2, Google Cloud, Azure VM |

## 🔁 ****5. Trường hợp kết hợp: Hybrid Cloud****

Khi doanh nghiệp:

* Dùng **Private Cloud** để lưu dữ liệu nhạy cảm
* Dùng **Public Cloud** để xử lý cao điểm, thử nghiệm

→ Gọi là **Hybrid Cloud** – kết hợp ưu điểm cả hai mô hình.

📌 **Tóm lại:**

* **Private IaaS**: Dành cho nội bộ, kiểm soát cao → CloudStack, OpenStack, OLVM
* **Public IaaS**: Mở cho mọi người, mở rộng cực nhanh → AWS, GCP, Azure
* Tùy theo mục đích sử dụng mà chọn mô hình phù hợp.

## Giới thiệu cloudstack

* Cloudstack là một nền tảng mã nguồn mở IaaS (Infrastructure as a Service) quản lý tài nguyên storage, network, compute; dùng để triển khai, quản lý và mở rộng hạ tầng private hoặc public IaaS compute cloud (điện toán đám mây)
* CloudStack hoạt động với nhiều hypervisors. Một single cloud có thể bao gồm nhiều hypervisor. Đến thời điểm version hiện tại, CloudStack support:
* BareMetal (via IPMI)
* Hyper-V
* KVM
* LXC
* vSphere (via vCenter)
* Xenserver
* Xen Project

## So sánh Cloudstack với vCenter và OLVM

| **Tiêu chí** | **CloudStack** | **vCenter (VMware)** | **OLVM (Oracle Linux Virtualization Manager)** |
| --- | --- | --- | --- |
| **Mô hình triển khai** | Nền tảng quản lý hạ tầng cloud (IaaS), hỗ trợ multi-tenant và multi-hypervisor. | Quản lý trung tâm cho môi trường vSphere (VMware ESXi). | Quản lý ảo hóa trên nền KVM, sử dụng oVirt làm backend. |
| **Hỗ trợ đa hypervisor** | ✅ (KVM, VMware, Xen, Hyper-V, v.v.) | ❌ (Chỉ hỗ trợ VMware ESXi) | ❌ (Chỉ hỗ trợ KVM) |
| **Khả năng mở rộng (Scalability)** | Rất cao, được thiết kế cho môi trường cloud công cộng và riêng lớn. | Cao, nhưng tập trung cho môi trường doanh nghiệp VMware. | Trung bình, thích hợp cho môi trường vừa và nhỏ. |
| **Tính năng cloud** | ✅ Có sẵn các tính năng như self-service, elastic scaling, auto-provisioning, API mạnh mẽ. | ❌ Không phải nền tảng cloud, thiếu self-service portal tự nhiên. | ❌ Không có tính năng cloud gốc, chủ yếu là quản lý VM. |
| **Chi phí** | ✅ Mã nguồn mở, không yêu cầu giấy phép thương mại. | ❌ Chi phí cao, yêu cầu license VMware (vSphere, vCenter, vSAN…). | ✅ Mã nguồn mở, miễn phí, nhưng hỗ trợ thương mại có phí. |
| **Tích hợp API / Automation** | ✅ REST API mạnh, hỗ trợ automation và orchestration tốt. | ✅ Có API nhưng chặt chẽ với hệ sinh thái VMware. | ✅ Có REST API, tương thích với Ansible và Terraform. |
| **Tính sẵn sàng cao (HA)** | ✅ Tích hợp HA cho VM, router, storage, v.v. | ✅ Có HA nhưng phụ thuộc vào vSphere HA. | ✅ Có HA cho VM nếu cấu hình đúng. |
| **Quản lý multi-tenant / người dùng** | ✅ Hỗ trợ đa người dùng, phân quyền, domain, account rất mạnh. | ❌ Không hỗ trợ native multi-tenant. | ❌ Hạn chế trong quản lý đa tenant. |
| **Giao diện người dùng (UI)** | Web UI thân thiện, đầy đủ chức năng. | Rất tốt, trực quan, chuyên nghiệp. | Đơn giản, dễ dùng nhưng không hiện đại bằng VMware. |

### Tổng kết ưu điểm nổi bật của CloudStack:

* **Đa dạng hypervisor** → phù hợp với môi trường hỗn hợp.
* **Mô hình cloud thực thụ** → hỗ trợ self-service, elastic compute.
* **Chi phí thấp** → mã nguồn mở, không cần license đắt đỏ.
* **Quản lý người dùng mạnh** → hỗ trợ domain/account phân quyền rõ ràng.
* **Tích hợp automation tốt** → REST API, hỗ trợ Ansible, Terraform.

### Khi nào nên chọn CloudStack?

* Khi bạn cần triển khai **private cloud** hoặc **hybrid cloud** có khả năng mở rộng.
* Khi bạn cần quản lý đa tenant hoặc phân quyền chi tiết.
* Khi bạn muốn tiết kiệm chi phí bằng giải pháp **mã nguồn mở**.
* Khi bạn có môi trường với **nhiều loại hypervisor khác nhau**.

### 1. ****Self-Service (Tự phục vụ)****

**Định nghĩa:**  
Self-service trong môi trường cloud nghĩa là **người dùng (ví dụ: developer, team IT nội bộ)** có thể tự **tạo, quản lý và xóa tài nguyên ảo hóa** như máy ảo (VM), mạng, ổ đĩa… thông qua một **giao diện web hoặc API**, **không cần liên hệ với quản trị viên hệ thống**.

**Ví dụ:**

* Một lập trình viên đăng nhập vào cổng web CloudStack.
* Họ chọn cấu hình (CPU, RAM, OS…) và nhấn “Tạo VM”.
* Sau vài phút, VM của họ đã sẵn sàng để sử dụng.

✅ **Lợi ích:**

* Tiết kiệm thời gian chờ đợi từ đội IT.
* Tự chủ trong phát triển, thử nghiệm, triển khai.
* Phù hợp với DevOps, Agile, CI/CD.

### 2. ****Elastic Compute (Tính đàn hồi trong tính toán)****

**Định nghĩa:**  
Elastic compute nghĩa là **khả năng mở rộng tài nguyên tính toán linh hoạt**, tùy theo nhu cầu thực tế. Bạn có thể dễ dàng **tăng/giảm số lượng máy ảo, thay đổi cấu hình CPU/RAM** theo thời điểm hoặc theo tải công việc.

**Ví dụ:**

* Ban ngày, hệ thống e-commerce cần 10 VM để đáp ứng người dùng.
* Ban đêm, nhu cầu giảm, hệ thống tự động hạ xuống 3 VM.
* Khi có sự kiện flash sale, hệ thống tự động scale lên 20 VM.

✅ **Lợi ích:**

* Tối ưu chi phí (chỉ dùng tài nguyên khi cần).
* Hiệu suất cao, sẵn sàng cho đột biến tải.
* Tích hợp tốt với autoscaling, monitoring.

### Tóm tắt dễ hiểu:

| **Tính năng** | **Ý nghĩa đơn giản** | **Lợi ích** |
| --- | --- | --- |
| **Self-service** | Người dùng tự tạo & quản lý tài nguyên | Tăng tốc độ, giảm phụ thuộc IT |
| **Elastic compute** | Tài nguyên tăng/giảm linh hoạt theo nhu cầu | Tối ưu chi phí & hiệu suất |

Để triển khai **CloudStack để quản lý 3 KVM host** hiện tại bạn đang dùng (đang chạy đơn lẻ, không có OLVM), bạn có thể thực hiện theo các **bước chính sau**:

## 🔧 ****TÓM TẮT KIẾN TRÚC TRIỂN KHAI****

CloudStack cần các thành phần chính:

1. **Management Server** (quản trị trung tâm).
2. **Database (MySQL/MariaDB)** – thường cài cùng máy với management.
3. **KVM Hosts** – 3 máy chủ KVM của bạn.
4. **Secondary Storage** – lưu template, ISO, snapshot.
5. **Primary Storage** – lưu VM đang chạy.
6. **System VM (router, template, console proxy)** – CloudStack tự tạo.

## 🧩 ****CÁC BƯỚC TRIỂN KHAI CLOUDSTACK VỚI 3 KVM HOST****

### ****Bước 1: Chuẩn bị****

* Hệ điều hành đề xuất: **CentOS 7**, **Rocky Linux 8**, hoặc **AlmaLinux 8/9**.
* Kết nối mạng đầy đủ giữa management và 3 KVM host.
* Mỗi KVM host cần có **libvirt + qemu-kvm**, có SSH key để CloudStack quản lý.
* Cấu hình IP tĩnh cho tất cả các node.

### ****Bước 2: Cài đặt CloudStack Management Server****

Trên 1 máy chủ riêng biệt (hoặc VM):

1. Cài đặt MariaDB (hoặc MySQL), Java, NTP.
2. Cài CloudStack theo hướng dẫn chính thức:

bash

CopyEdit

yum install -y cloudstack-management

1. Cài database:

bash

CopyEdit

cloudstack-setup-databases cloud:password@localhost --deploy-as=root

1. Khởi động các dịch vụ CloudStack:

bash

CopyEdit

cloudstack-setup-management

### ****Bước 3: Chuẩn bị các KVM Host****

Trên **mỗi host KVM**, thực hiện:

1. Cài các gói cần thiết:

bash

CopyEdit

yum install -y qemu-kvm libvirt libvirt-client bridge-utils cloudstack-agent

1. Kích hoạt dịch vụ libvirtd và bật khởi động cùng hệ thống:

bash

CopyEdit

systemctl enable --now libvirtd

1. Thiết lập bridge mạng (ví dụ cloudbr0), dùng cho CloudStack quản lý.
2. Mở tường lửa hoặc tắt firewalld (nếu cần kiểm tra kết nối ban đầu).
3. Kết nối KVM host về CloudStack management (qua SSH).

### ****Bước 4: Cấu hình trong CloudStack GUI****

Truy cập GUI qua trình duyệt:

arduino

CopyEdit

http://<CloudStack\_Management\_IP>:8080/client

1. **Đăng nhập mặc định**:
   * User: admin, Password: password
2. **Tạo cấu trúc hạ tầng CloudStack:**
   * Zone → Pod → Cluster → Host → Storage

(3 host KVM của bạn sẽ nằm trong cùng **Cluster**).

1. **Thêm primary storage**:
   * Dùng NFS, iSCSI hoặc Ceph (phổ biến nhất là NFS).
2. **Thêm secondary storage**:
   * Cũng dùng NFS, chứa template & ISO.

### ****Bước 5: Import Template & Tạo VM****

1. Import template (ví dụ: CentOS, Ubuntu) từ URL vào secondary storage.
2. Tạo VM từ template, chọn KVM host cụ thể hoặc để CloudStack tự chọn.

## 🎯 Lưu ý quan trọng:

* CloudStack cần ít nhất **1 IP public** để System VM hoạt động.
* System VM (Router VM, Console VM, v.v.) được tự động deploy.
* Nên cấu hình NTP đồng bộ thời gian giữa management và các host.
* Nên dùng DNS nội bộ hoặc /etc/hosts để các node nhận diện nhau chính xác.

### Tình huống của bạn:

* **3 host KVM đang chạy production** với nhiều VM thật.
* **OS đang dùng là Oracle Linux** (OL7/8/9).
* Chưa có hệ quản trị tập trung như OLVM hay CloudStack.

👉 Việc tích hợp CloudStack để quản lý các host này **không thể thực hiện trực tiếp mà không ảnh hưởng** nếu làm không đúng cách.

## ✅ Giải pháp khả thi: “Tích hợp dần mà không làm gián đoạn production”

Dưới đây là **kịch bản triển khai CloudStack an toàn**, không ảnh hưởng VM đang chạy:

### 🔸 ****1. Tạo vùng tách biệt cho CloudStack****

CloudStack không thể "quản lý" những VM KVM đã tạo bên ngoài hệ thống của nó.  
Bạn cần **tạo một cụm/zone riêng biệt**, và nếu cần, **di chuyển VM dần sang hạ tầng CloudStack**.

**Gợi ý:**

* Triển khai CloudStack trên một server riêng (VM hoặc bare-metal).
* Cấu hình 1 trong 3 host KVM (sau khi di chuyển bớt VM) để làm node CloudStack **thử nghiệm**.
* Sau đó, di chuyển từng VM (nếu cần) bằng cách export/import qcow2 và tạo template trong CloudStack.

### 🔸 ****2. Không nên cài cloudstack-agent trực tiếp trên host đang chạy VM production****

Vì:

* CloudStack agent sẽ quản lý libvirt và có thể thay đổi các bridge/mạng mặc định.
* Có thể xung đột với VM đang chạy nếu CloudStack cố gắng can thiệp.
* CloudStack yêu cầu host "sạch" để quản lý VM hoàn toàn.

### 🔸 ****3. Các bước triển khai an toàn gợi ý****

#### Giai đoạn 1: **Tách 1 host để thử nghiệm**

1. **Di chuyển tạm các VM đang chạy trên 1 trong 3 host sang 2 host còn lại** (nếu đủ tài nguyên).
2. Cài CloudStack agent trên host đã rảnh.
3. Tích hợp host đó vào hệ thống CloudStack.

#### Giai đoạn 2: **Thử nghiệm khởi tạo VM từ CloudStack**

* Tạo VM mẫu.
* Kiểm tra kết nối, NAT, mạng nội bộ.
* Test hiệu suất và hoạt động ổn định.

#### Giai đoạn 3: **Lập kế hoạch di chuyển**

* Nếu ổn định, export từng VM .qcow2 từ 2 host còn lại → import vào CloudStack → tạo template.
* Khi các VM đã chạy ổn định trên CloudStack, có thể tái sử dụng các host còn lại để đưa vào hệ thống.

## ❌ Không thể:

CloudStack không hỗ trợ:

* Nhận diện hoặc quản lý các VM đã tạo **ngoài hệ thống CloudStack**.
* Import tự động từ libvirt/KVM cũ (phải thủ công).

## ✅ Nếu bạn muốn vừa quản lý vừa giữ nguyên VM hiện tại?

Bạn có 2 lựa chọn:

| **Giải pháp** | **Ghi chú** |
| --- | --- |
| 🔸 **Dùng Ansible/libvirt để quản lý** | Giữ nguyên setup, quản lý VM hiện tại qua virsh, libvirt, Ansible, Cockpit. |
| 🔸 **Triển khai CloudStack song song** | Từ từ di chuyển sang CloudStack theo cách có kiểm soát. |

### 📌 Kết luận

* Không thể “gắn” CloudStack vào host KVM đang chạy production mà không ảnh hưởng nếu không di chuyển VM.
* Bạn nên triển khai CloudStack song song, thử nghiệm trên một host, và dần migrate.
* Sau khi migrate xong, các host mới được CloudStack quản lý hoàn toàn.

Nếu bạn muốn:

* **Kịch bản di chuyển chi tiết từng bước**
* **Script hỗ trợ export qcow2 + import template**
* **Sơ đồ mạng gợi ý trong CloudStack cho môi trường sản xuất**

👉 Mình có thể chuẩn bị cụ thể giúp bạn. Bạn đang dùng Oracle Linux bản mấy vậy (7, 8, hay 9)? Và bạn dùng bridge mạng nào trong KVM (bridge thủ công, virbr0, hay OVS)?

2/2

##### You said:

tôi dùng Oracle linux 8, trong KVM tôi dùng Bridge mode Open

###### ChatGPT said:

Updated saved memory

Cảm ơn bạn! Với thông tin này, mình sẽ gợi ý chi tiết phù hợp với môi trường Oracle Linux 8 + KVM + bridge thủ công.

## ✅ ****Hướng triển khai CloudStack an toàn với KVM đang dùng bridge mode thủ công****

### 🔧 ****Tóm tắt hiện trạng****

* Oracle Linux 8 (OL8)
* Đang chạy production VM trên KVM
* Mạng KVM dùng **bridge thủ công** (ví dụ: br0, openbr0, không dùng virbr0)

## 📘 ****Gợi ý kịch bản triển khai dần – không gián đoạn VM hiện tại****

### 🔸 ****Giai đoạn 1: Chuẩn bị CloudStack****

1. **Triển khai 1 máy riêng (hoặc VM)** cài CloudStack Management + Database (CentOS 7, Rocky/Alma 8 đều được).
2. Cấu hình IP tĩnh, đặt hostname đầy đủ (/etc/hosts nên ánh xạ tên các node).
3. Cài CloudStack management server.

### 🔸 ****Giai đoạn 2: Chọn 1 host KVM để tích hợp****

* Di chuyển toàn bộ VM trên **1 host tạm thời không tải**.
* Trên host đó:
  + Đảm bảo bridge mạng vẫn hoạt động (openbr0 chẳng hạn).
  + Cài thêm CloudStack Agent:

bash

CopyEdit

yum install -y cloudstack-agent

* Cấu hình lại cloudstack-agent để dùng đúng bridge:
  + Mở file /etc/cloudstack/agent/agent.properties, sửa dòng:

ini

CopyEdit

network.bridge.type=native

libvirt.network.bridge=openbr0

* + Khởi động agent:

bash

CopyEdit

systemctl enable --now cloudstack-agent

### 🔸 ****Giai đoạn 3: Thêm host vào CloudStack GUI****

* Truy cập GUI: http://<management\_ip>:8080/client
* Tạo Zone, Pod, Cluster → thêm host (KVM vừa cài agent).
* CloudStack sẽ ssh sang host và kiểm tra kết nối.

### 🔸 ****Giai đoạn 4: Thiết lập Storage****

* **Primary Storage**: tạo trên NFS/iSCSI (bạn có thể dùng NFS mount từ 1 NAS hoặc 1 máy khác).
* **Secondary Storage**: nơi chứa template/ISO. Cũng có thể dùng NFS.

### 🔸 ****Giai đoạn 5: Import template & tạo VM****

* Import template từ URL (CentOS, Ubuntu, Oracle Linux…)
* Tạo VM mới → test console, SSH, kết nối mạng

### 🔸 ****Giai đoạn 6: Di chuyển dần VM production vào CloudStack****

* Trên 2 host còn lại, export từng VM ra file .qcow2:

bash

CopyEdit

virsh shutdown <vm>

virsh dumpxml <vm> > vm.xml

qemu-img convert -O qcow2 /var/lib/libvirt/images/vm.img /mnt/export/vm.qcow2

* Import vào CloudStack làm template.
* Tạo lại VM từ template → gắn disk → boot → test → go-live.

## 📌 Tài nguyên phụ (mình có thể hỗ trợ nếu cần):

* File cấu hình mẫu cho agent.properties
* Sơ đồ mạng gợi ý cho bridge openbr0
* Script tạo NFS làm Primary/Secondary Storage nhanh
* Script tự động import qcow2 thành template CloudStack

# ****Thuyết minh: Giới thiệu về Apache CloudStack****

## 1. ****Giới thiệu chung****

### Apache CloudStack là gì?

Apache CloudStack là một nền tảng mã nguồn mở, cho phép **triển khai và quản lý hạ tầng đám mây IaaS (Infrastructure as a Service)**. Với CloudStack, các tổ chức có thể biến trung tâm dữ liệu vật lý thành **dịch vụ đám mây riêng hoặc cộng đồng**, cho phép người dùng **tự tạo, cấu hình và quản lý máy ảo** thông qua giao diện web hoặc API.

## 2. ****Công nghệ lõi & Kiến trúc****

CloudStack hỗ trợ nhiều hypervisor và dịch vụ tích hợp:

| **Thành phần** | **Vai trò** |
| --- | --- |
| **KVM**, **VMware**, **XenServer** | Hypervisor được hỗ trợ |
| **NFS, iSCSI, Ceph, GlusterFS** | Storage linh hoạt |
| **CloudStack Management Server** | Trung tâm điều khiển |
| **System VMs** | Cung cấp routing, console, DHCP, VPN |
| **API/GUI** | Giao diện quản trị & tích hợp tự động hóa |

### 🖼️ Sơ đồ kiến trúc tổng quát:

* Zones → Pods → Clusters → Hosts → Storage

## 3. ****Các công nghệ mới & nổi bật****

| **Công nghệ** | **Mô tả** |
| --- | --- |
| **Template & ISO management** | Tự tạo template cho VM nhanh chóng |
| **SDN Support (CloudStack + Tungsten, Nuage)** | Hỗ trợ mạng định nghĩa bằng phần mềm |
| **VM Autoscaling** | Tự động mở rộng VM theo tải |
| **API RESTful** | Cho phép tích hợp với CI/CD, Ansible, Terraform |
| **UI mới (Primate)** | Giao diện web hiện đại, dễ sử dụng |
| **Storage tiering** | Quản lý nhiều lớp lưu trữ khác nhau |

## 4. ****Ưu điểm của CloudStack****

| **Ưu điểm** | **Diễn giải** |
| --- | --- |
| ✅ **Mã nguồn mở & miễn phí** | Không tốn chi phí license như VMware vCenter |
| ✅ **Triển khai nhanh** | Có thể triển khai cloud private trong vài giờ |
| ✅ **Tự phục vụ (Self-Service)** | Người dùng tạo/huỷ VM qua web/API |
| ✅ **Tính đàn hồi (Elastic Compute)** | Scale up/down dễ dàng |
| ✅ **Tự động hóa tốt** | Hỗ trợ script, CLI, API |
| ✅ **Đa hypervisor** | Không bị khóa với một công nghệ (vendor-lock-in) |
| ✅ **Cộng đồng lớn, tài liệu rõ ràng** | Được hỗ trợ bởi Apache Software Foundation |

## 5. ****Ví dụ ứng dụng thực tế****

### 🎯 ****Doanh nghiệp nội bộ****

* **Tình huống**: Một công ty có 5 server KVM đang chạy VM rời rạc.
* **Giải pháp**: Dùng CloudStack để gom lại, tạo portal quản lý tập trung.
* **Lợi ích**: Quản trị đơn giản hơn, cấp phát VM cho dev/test nhanh chóng.

### ☁️ ****Dịch vụ hosting****

* **Tình huống**: Nhà cung cấp dịch vụ muốn tạo “cloud VPS”.
* **Giải pháp**: Dùng CloudStack để cho khách hàng tự tạo VM, tự quản lý.
* **Lợi ích**: Tự động hoá 100%, không cần quản trị viên can thiệp từng khách.

### 🧪 ****Môi trường thử nghiệm CI/CD****

* **Tình huống**: Team DevOps cần VM mới mỗi khi chạy pipeline.
* **Giải pháp**: Tích hợp CloudStack API với GitLab CI hoặc Jenkins.
* **Lợi ích**: Mỗi lần chạy pipeline có VM riêng biệt, sạch sẽ, tự hủy khi xong.

## 6. ****So sánh với nền tảng khác****

| **Tính năng** | **CloudStack** | **vCenter** | **OLVM** |
| --- | --- | --- | --- |
| Mã nguồn mở | ✅ | ❌ (thu phí) | ✅ |
| Đa hypervisor | ✅ | ❌ (chỉ VMware) | ❌ (chỉ KVM) |
| Tự phục vụ (GUI/API) | ✅ | ❌ (phức tạp hơn) | Trung bình |
| Tích hợp DevOps (API) | ✅ | Hạn chế | Hạn chế |
| Cộng đồng mã nguồn mở | Rộng | Thương mại | Giới hạn |
| Quản lý storage, network đa dạng | ✅ | ✅ | ❌ |

## 7. ****Kết luận****

Apache CloudStack là một giải pháp **mã nguồn mở mạnh mẽ, linh hoạt**, phù hợp với:

* Doanh nghiệp muốn tự quản lý hạ tầng cloud riêng.
* Các nhà cung cấp dịch vụ VPS/Cloud Hosting.
* Các team DevOps/Dev cần hạ tầng nhanh, tự động.

Nếu bạn đang dùng KVM và muốn có **quản trị tập trung, tự phục vụ, dễ tích hợp API**, CloudStack là **ứng viên hàng đầu**, thay thế hợp lý cho OLVM hoặc vCenter nếu muốn giảm chi phí license và thoát khỏi vendor-lock-in.

Tính năng **Tự phục vụ (Self-Service)** trong Apache CloudStack được thiết kế để người dùng cuối có thể **tự tạo, quản lý và xoá máy ảo, ổ đĩa, mạng, snapshot**... mà **không cần quản trị viên can thiệp trực tiếp**. Để đảm bảo an toàn và phân quyền chặt chẽ, CloudStack sử dụng **cơ chế phân quyền theo tài khoản (Account), domain và role (vai trò)**.

## 🔐 ****Cơ chế phân quyền người dùng trong CloudStack****

### 1. ****Tổ chức theo Domain (Miền)****

CloudStack cho phép tạo **cây phân cấp domain** tương tự như tổ chức/doanh nghiệp:

pgsql

CopyEdit

Root Domain

├── Domain A (Ví dụ: Sales)

│ └── Account: user1, user2

├── Domain B (Ví dụ: IT)

│ └── Account: admin, dev1

Mỗi domain có thể được quản lý riêng và tách biệt về tài nguyên.

### 2. ****Tài khoản (Account)****

Mỗi người dùng trong CloudStack là một **Account**, và tài nguyên mà họ tạo (VM, disk, network, snapshot...) sẽ thuộc về account đó.

### 3. ****Các loại Vai trò (Roles) mặc định****

| **Vai trò** | **Quyền hạn** | **Dành cho** |
| --- | --- | --- |
| **Root Admin** | Toàn quyền quản trị toàn hệ thống | Quản trị viên hệ thống |
| **Domain Admin** | Quản lý domain con, tài nguyên người dùng | Quản trị viên bộ phận |
| **User** | Tự quản lý tài nguyên của chính mình | Người dùng cuối |
| **Custom Role** | Tùy chỉnh chi tiết hành vi | (Tuỳ chỉnh theo nhu cầu) |

## 📊 ****Ví dụ phân quyền thực tế****

### 🎯 Ví dụ 1: Dev tự tạo VM

| **Thành phần** | **Quyền** |
| --- | --- |
| User: dev1 trong domain IT | Có quyền tạo 5 VM, mỗi VM tối đa 4 vCPU, 8GB RAM |
| Domain Admin it-admin | Có thể giới hạn quota cho dev1, gán template dùng được |
| Root Admin | Toàn quyền xem log, audit, quota toàn hệ thống |

### 🎯 Ví dụ 2: Phòng Kế toán chỉ xem console, không xóa VM

Tạo một **Custom Role**:

* Có quyền:
  + listVirtualMachines, viewConsole
* Không có quyền:
  + destroyVirtualMachine, rebootVirtualMachine

## ⚙️ ****Tùy chỉnh Role nâng cao****

Từ CloudStack 4.9+, hỗ trợ **Role-Based Access Control (RBAC)** rất chi tiết:

Bạn có thể tạo role mới và quy định chính xác các API nào role đó được dùng (hơn 400 API control point).

Ví dụ tạo Role “ReadOnlyUser”:

json

CopyEdit

{

"name": "ReadOnlyUser",

"description": "Chỉ xem, không thay đổi tài nguyên",

"rule": [

{"rule": "allow", "permission": "read", "api": "\*"}

]

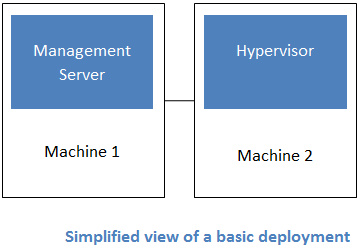
}

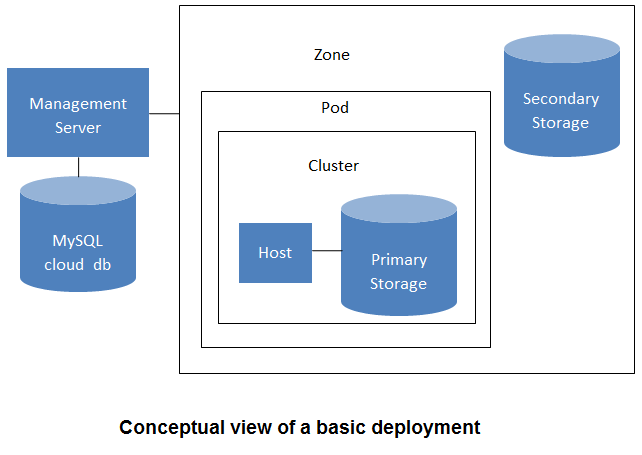
## ✅ ****Lợi ích của Self-Service + Phân quyền****

* Giảm tải cho Admin → không cần tạo VM thủ công
* Chủ động cấp phát theo nhu cầu người dùng
* Quản lý quota, giới hạn tài nguyên từng nhóm
* An toàn, tách biệt quyền rõ ràng giữa phòng ban

## Mô hình kiến trúc cloudstack

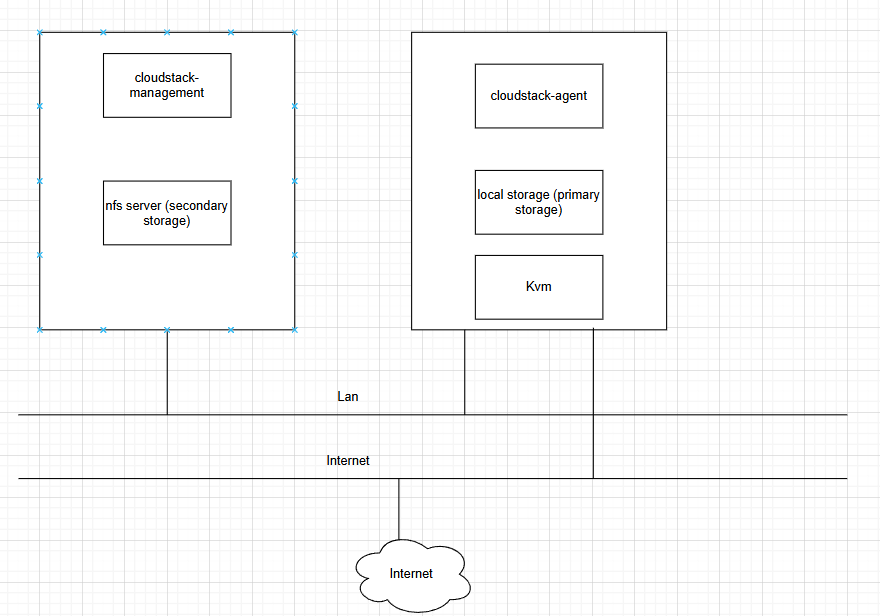
Mô hình thành phần chính bên trong cloudstack





|  |  |
| --- | --- |
| Thành phần | Giải thích |
| Management server | Là server đảm nhiệm chức năng quản lý, phân phối công việc cho các thành phần hệ thống |
| Hypervisor | Là các máy chủ đảm nhiệm việc thực thi, tạo và run vm khi được yêu cầu từ management  Mỗi máy đều được cài cloudstack agent để nhận nhiệm vụ từ management |
| Region | Phân biệt các vùng hệ thống, có thể theo địa chỉ địa lý ( chỉ sử dụng với hệ thống lớn , đa quốc gia) |
| Zone | Vùng tách biệt như một datacenter, bao gồm network, pod, host riêng biệt |
| Pod | Được ví như một tủ rack trong hệ thống datacenter, chứa nhiều host và cluster.  Mỗi pod chứa 1 subnet riêng để phân biệt dùng cho đường traffic management và storage |
| Cluster | Một nhóm các host chung loại hypervisor, bao gồm thông tin primary storage |
| Host | Là máy chủ chứa tài nguyên như ram, cpu,… được cài đặt hypervisor (kvm, vmware,…) |
| Primary storage | Kho lưu trữ vm, chứa toàn bộ data của vm instance được tạo |
| Secondary storage | Lưu trữ file template, iso, backup của vm |

Mô hình cài đặt được sử dụng trong tài liệu:



Trong đó gồm:

* 2 đường mạng
* 2 máy chủ (1 sv management cài nfs server, 1 cài agent sử dụng kvm)

## Hypervisor - KVM

### Giới thiệu KVM

KVM (Kernel-based Virtual Machine) là công nghệ ảo hoá mã nguồn mở cho linux, cho phép một hệ thống Linux hoạt động như một hypervisor (trình ảo hóa), từ đó có thể tạo và chạy nhiều máy ảo (Virtual Machines - VM) trên một máy vật lý duy nhất.

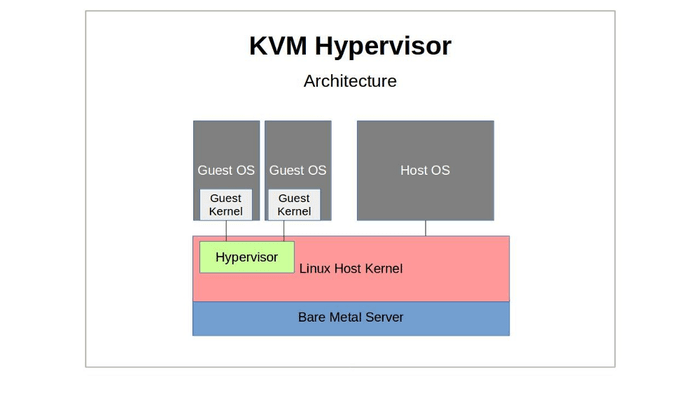
KVM được phát triển và được tích hợp trực tiếp vào kernel như một module kernel

KVM chỉ có thể được sử dụng với các chương trình ảo hoá phần cứng như Intel-VT or AMD-V.

### Tính năng KVM

|  |  |
| --- | --- |
| Tính năng nổi bật | Giải thích về tính năng |
| Hiệu năng cao | Khả năng ảo hoá của kvm cho hiệu suất các thiết bị ảo hoá gần tương đương thiết bị vật lý |
| Khả năng mở rộng | Là một module kernel, KVM có khả năng xử lý lượng thông tin lớn từ các VM. KVM có sẵn thiết lập cho phép dựng cluster mở rộng các node, sẵn sàng cho thiết lập hạ tầng cloud |
| Bảo mật | KVM là một phần kernel linux, là một phần quan trọng trong hệ thống linux hiện nay, được cập nhật, patching liên tục |
| Khả năng phát triển | KVM được tạo ra lần đầu vào năm 2006 và vẫn tiếp tục phát triển cho đến nay. Trên khắp thế giới có hơn 1000 develop làm việc và tiếp tục phát triển đóng góp cho KVM code |
| Chi phí | Vì KVM là giải pháp mã nguồn mở, giúp doanh nghiệp, người sử dụng tiết kiệm chi phí để sử dụng |

### Mô hình hoạt động của KVM



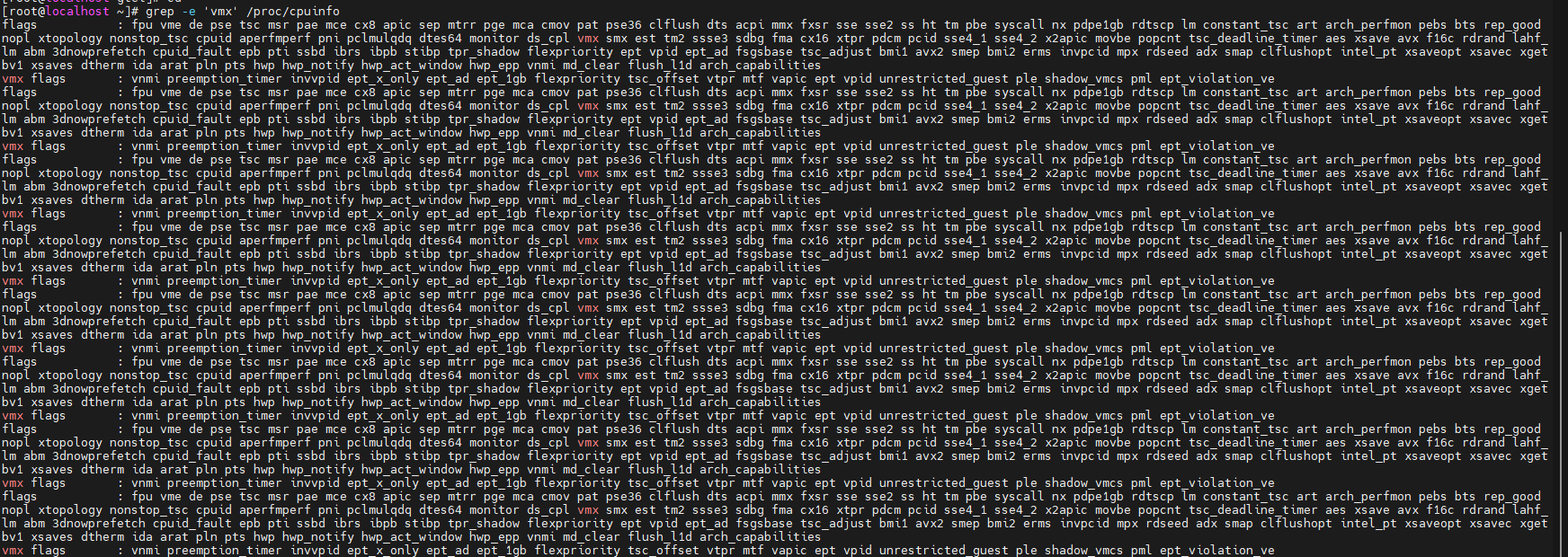
KVM hỗ trợ giao tiếp xử lý với hệ thống thiết bị hardware trực tiếp, trong khi qemu hoặc libvirtd quản lý hardware ảo

### Cài đặt KVM

Lưu ý: tất cả gói cài đều được lấy từ internet nên đảm bảo tất cả host đều đi được tới internet

Kiểm tra host có hỗ trợ ảo hoá cpu không:

|  |
| --- |
| grep -e 'vmx' /proc/cpuinfo |



Kiểm tra kvm module đã được load vào hệ thống chưa:

|  |
| --- |
| Lsmod | grep kvm |

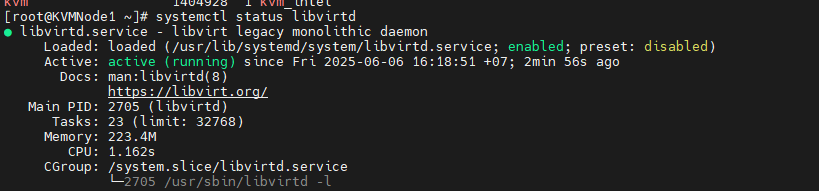


Cài đặt libvirt và các gói bổ trợ ảo hoá kvm:

|  |
| --- |
| dnf install qemu-kvm libvirt virt-manager virt-install epel-release -y |

Sau khi cài xong, bật và check service libvirtd:

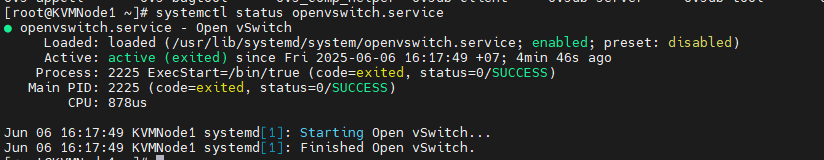
|  |
| --- |
| Systemctl start libvirtd  Systemctl enable libvirtd  Systemctl status libvirtd |



Do không sử dụng interface ban đầu của hệ thống mà dùng openvswitch nên cài đặt thêm:

Cài đặt openvswitch:

|  |
| --- |
| Dnf install centos-release-nfv-openvswitch  Dnf install openvswitch3.5  Systemctl start openvswitch  Systemctl enable openvswitch  Systemctl status openvswitch |



Để sử dụng openvswitch cần tạo các bridge thuộc openvswitch quản lý và assign cho bridge đó interface ban đầu được sử dụng trên hệ thống.

Bridge mới sẽ thay thế nhiệm vụ của bridge bị thay thế.

Config interface lan và internet:

|  |
| --- |
| Ovs-vsctl add-br <interface moi>  Ovs-vsctl add-port <interface moi> <interface cu> |

Trong đó:

- interface moi: tên interface mới được tạo để thay thế interface cũ

- interface cu: tên interface ban đầu sử dụng bị name interface thay thế



Do interface mới sẽ thay thế interface cũ nên cần gắn ip lại cho interface hoặc dùng dhcp để nhận lại ip

|  |
| --- |
| Ip addr add <ip>/<subnet> dev <interface> |

Với dhcp

|  |
| --- |
| Dhclient <interface> |

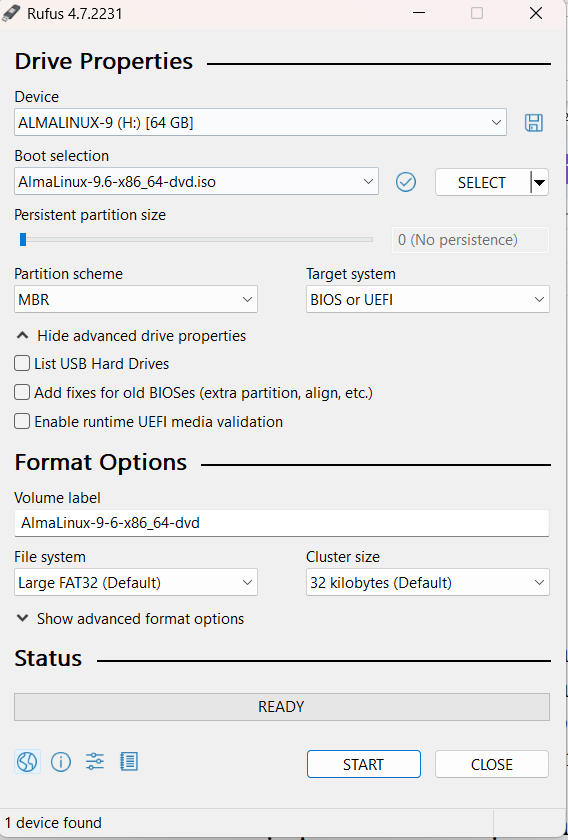
Nếu interface bị tắt thì bật lên

|  |
| --- |
| Ip link set <interface> up |

1. **Chuẩn bị usb boot và thực hiện cài đặt OS**
   1. Tạo usb boot

ISO được sử dụng để boot cài đặt là almalinux 9.6

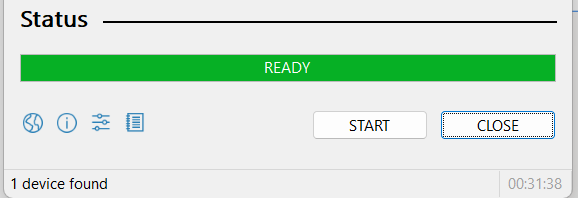
Sử dụng rufus để tạo usb boot



Giải thích 1 số thông số:

|  |  |
| --- | --- |
| Thông số | Giải thích |
| Device | Thiết bị usb boot |
| Boot selection | File iso, template được dùng để tạo boot |
| Partition scheme | Thông tin phân vùng dữ liệu |
| Target system | Loại boot hỗ trợ |
| Volume label | Tên ổ volume disk sau khi được tạo |
| File system | Cấu trúc file system được sử dụng |

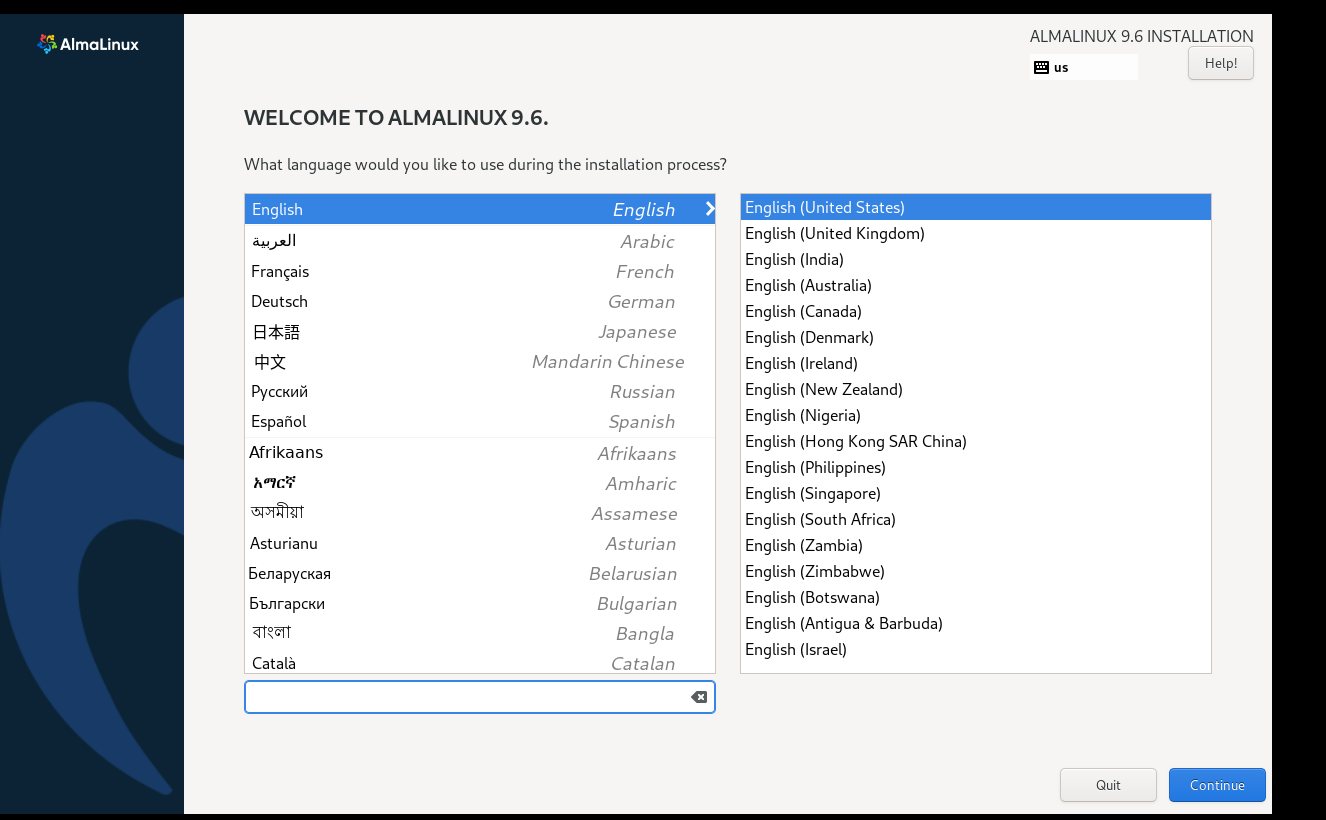
Khi tạo xong có báo ready:



* 1. Cài đặt OS

Sau khi tạo usb boot xong và thực hiện boot:

Chọn ngôn ngữ và bàn phím:

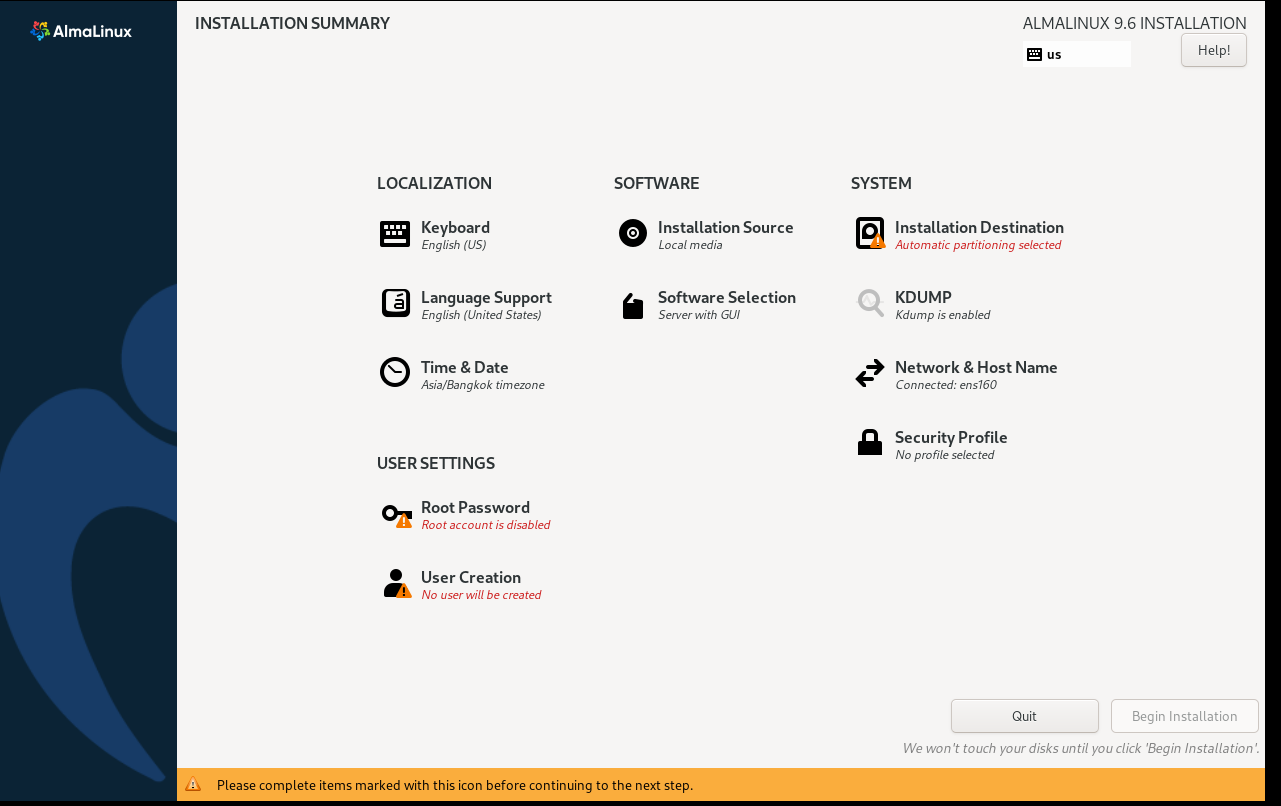


Các thành phần config bắt buộc:

- User settings

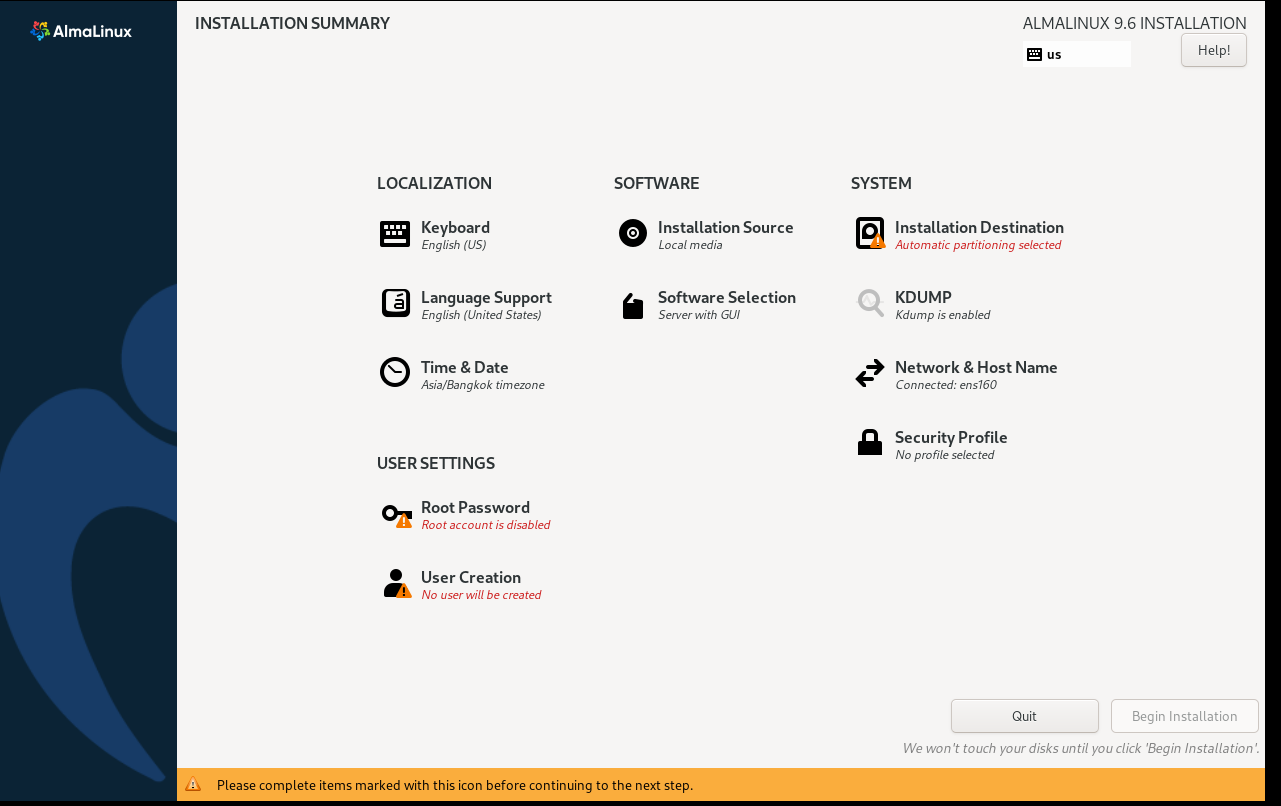
- Sofware selection

- Installation Destination



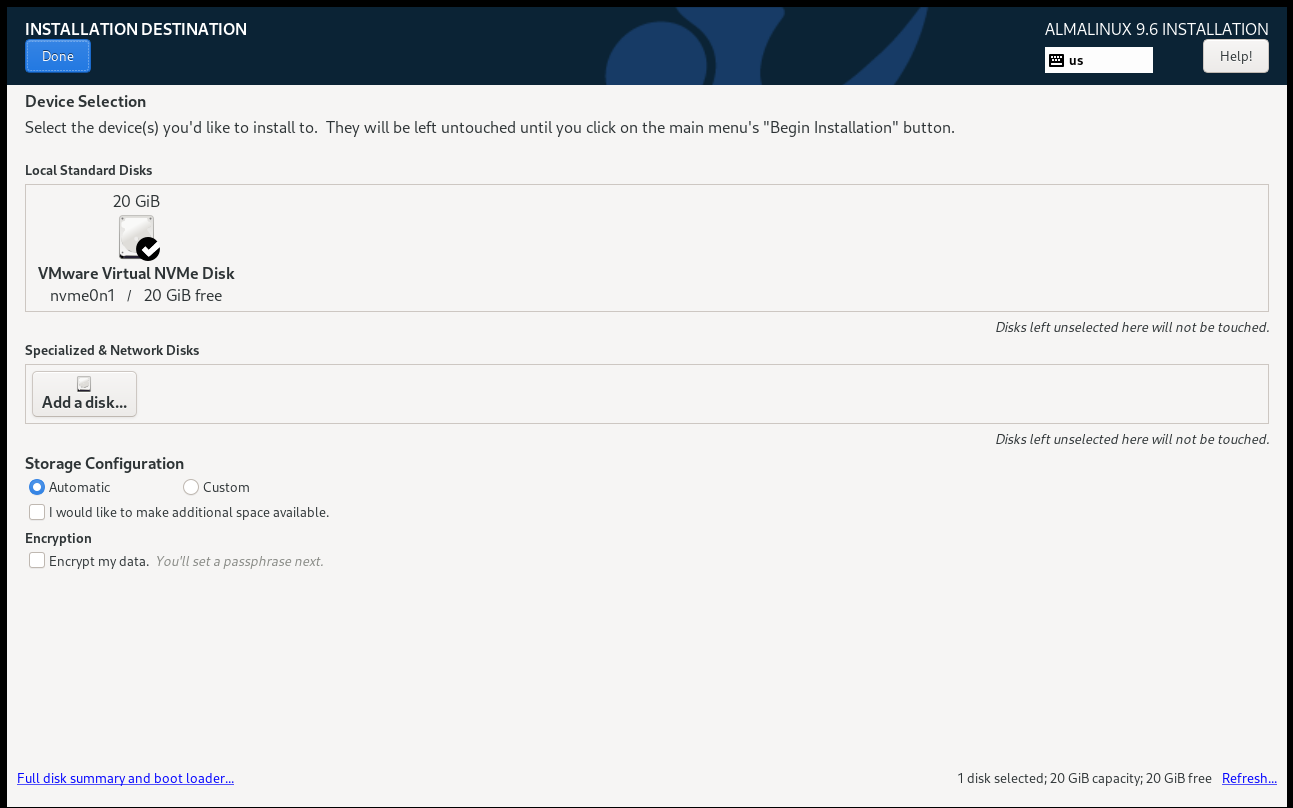
Config sofware selection:

- chọn server vì không sử dụng GUI



Installation Destination:

- Chọn đúng disk để cài đặt, có thể custom lại nếu cần thiết

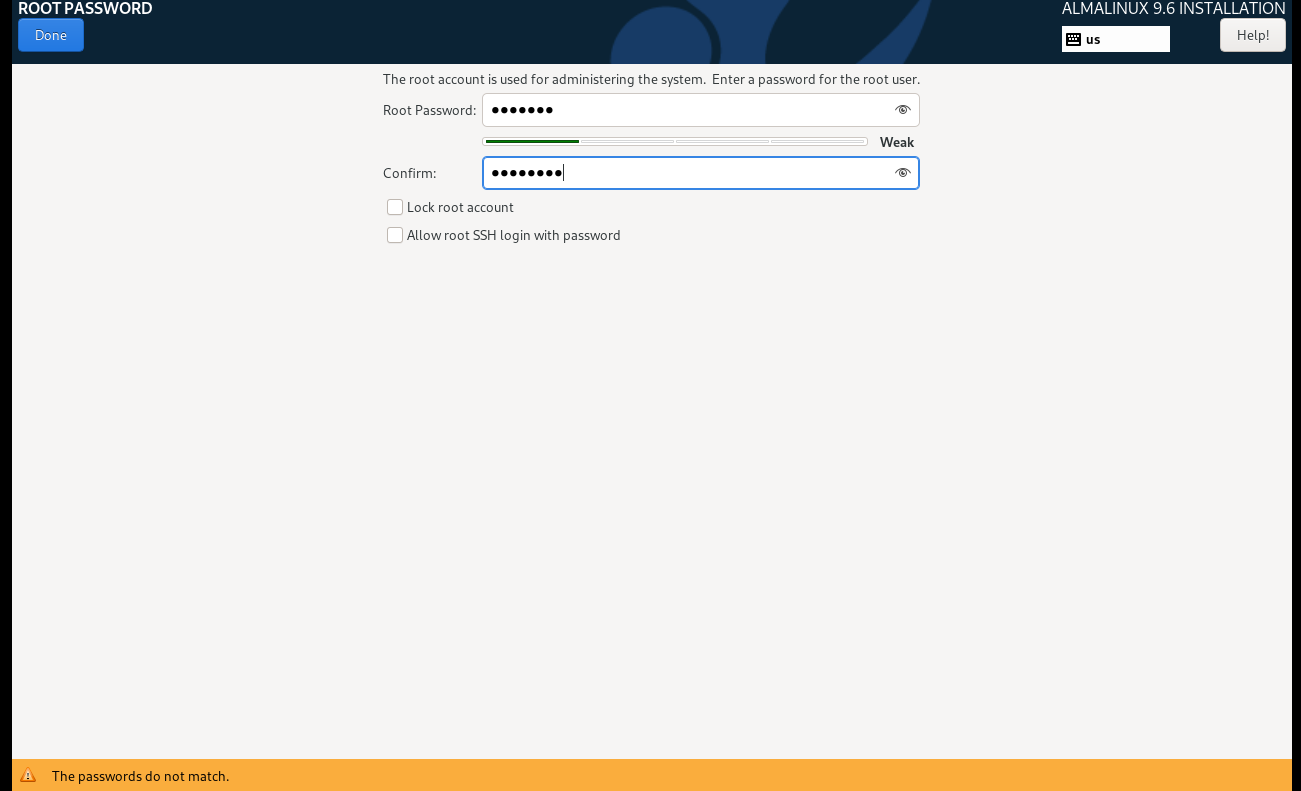


User setting:

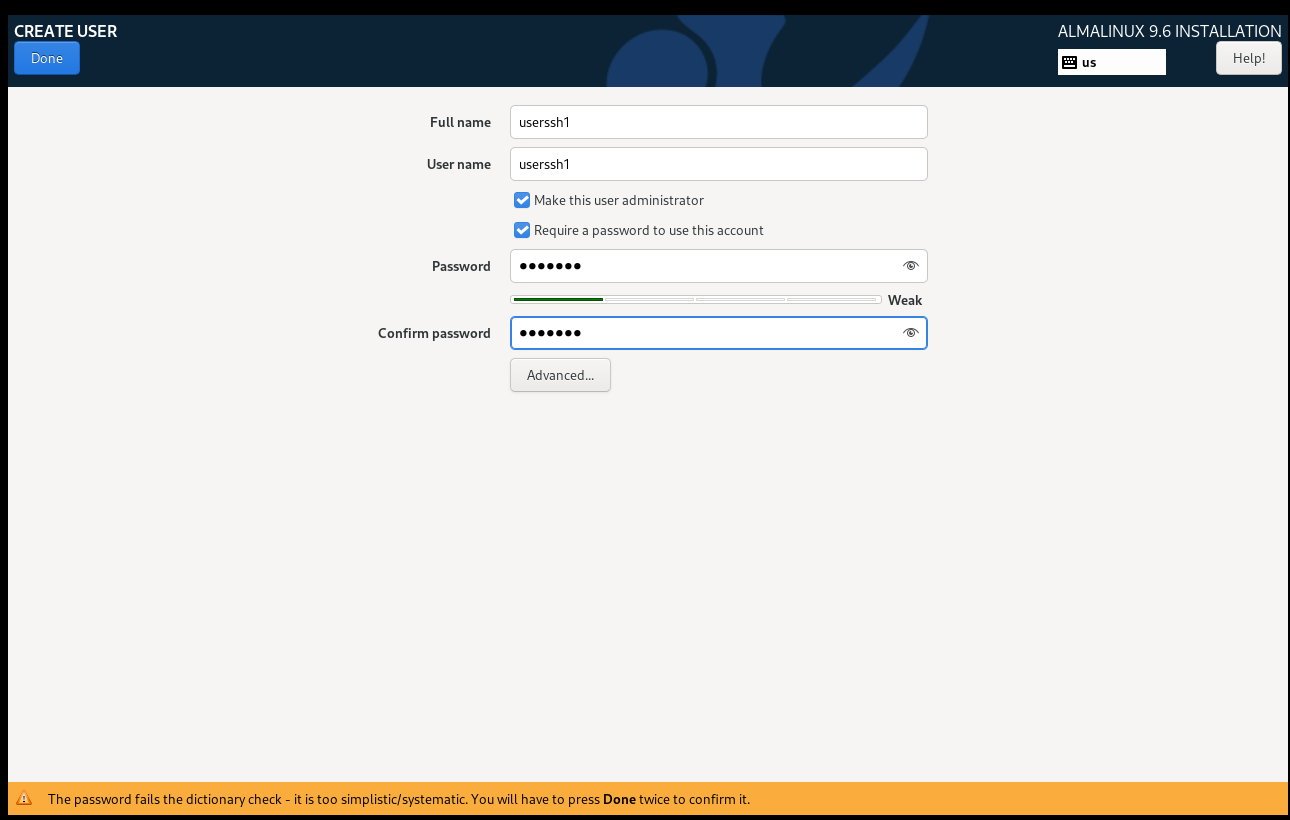
- config password root account

- có thể tích không cho ssh bằng root nếu không muốn dùng password ssh bằng user root

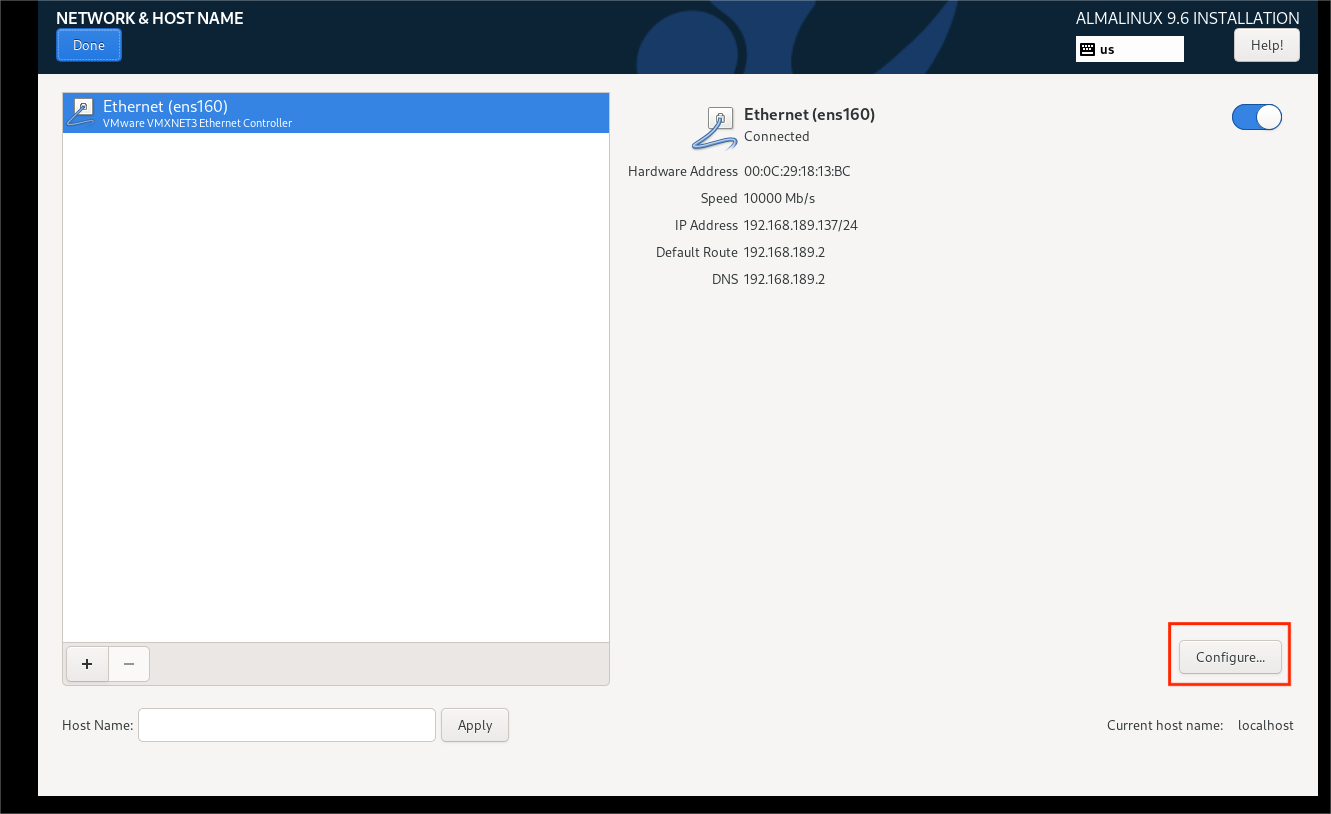
+ trong trường hợp tích không ssh root password thì cần config thêm 1 user khác để ssh



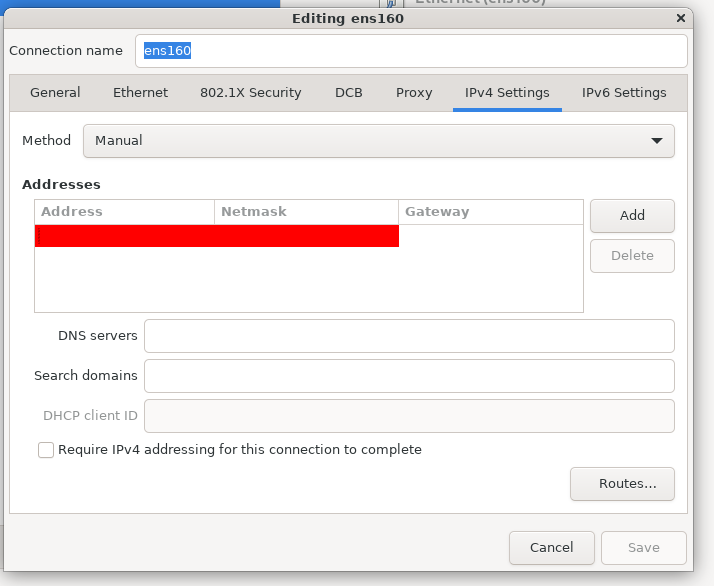
User để ssh ( tích make this user administrator để cho phép dùng lệnh sudo)



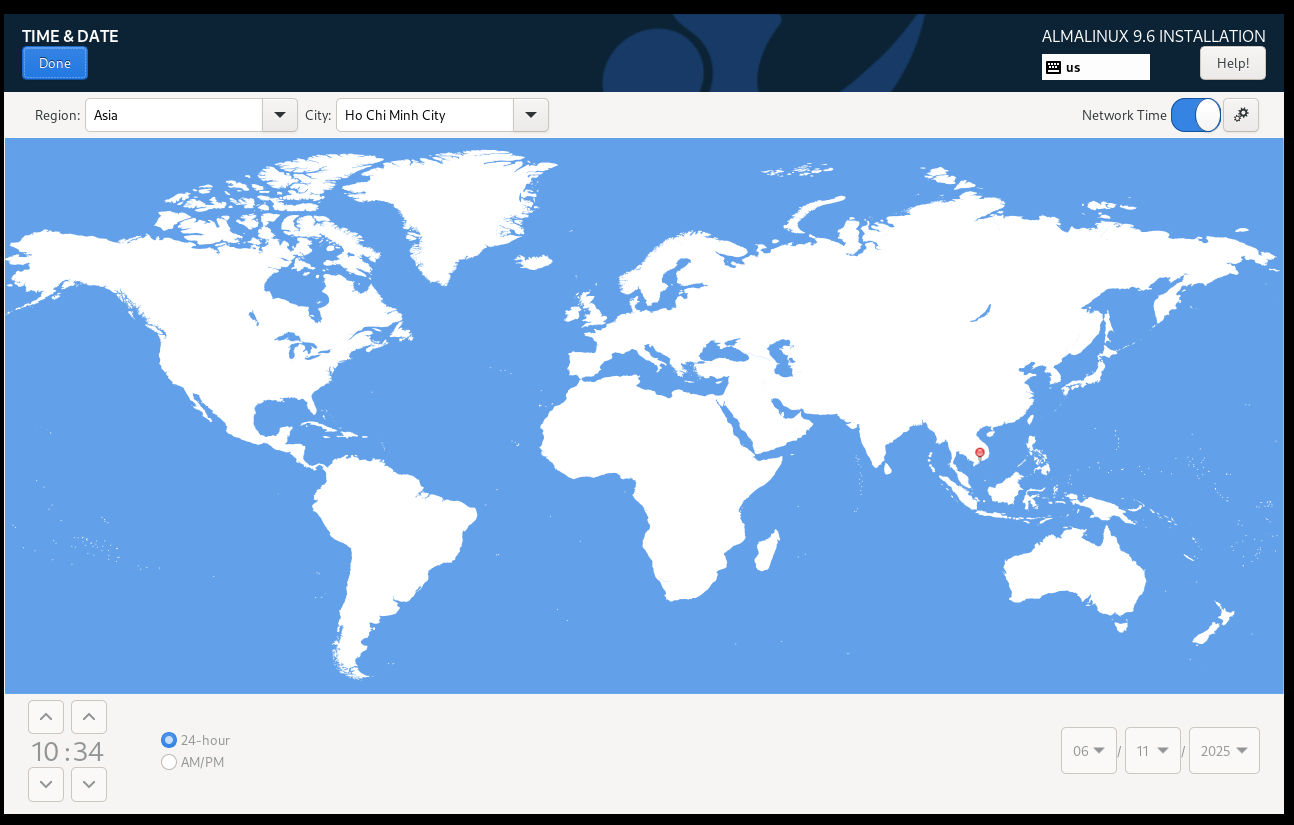
Với network, nếu có dhcp thì không cần thiết phải config, có thể thay đổi ip nếu muốn:



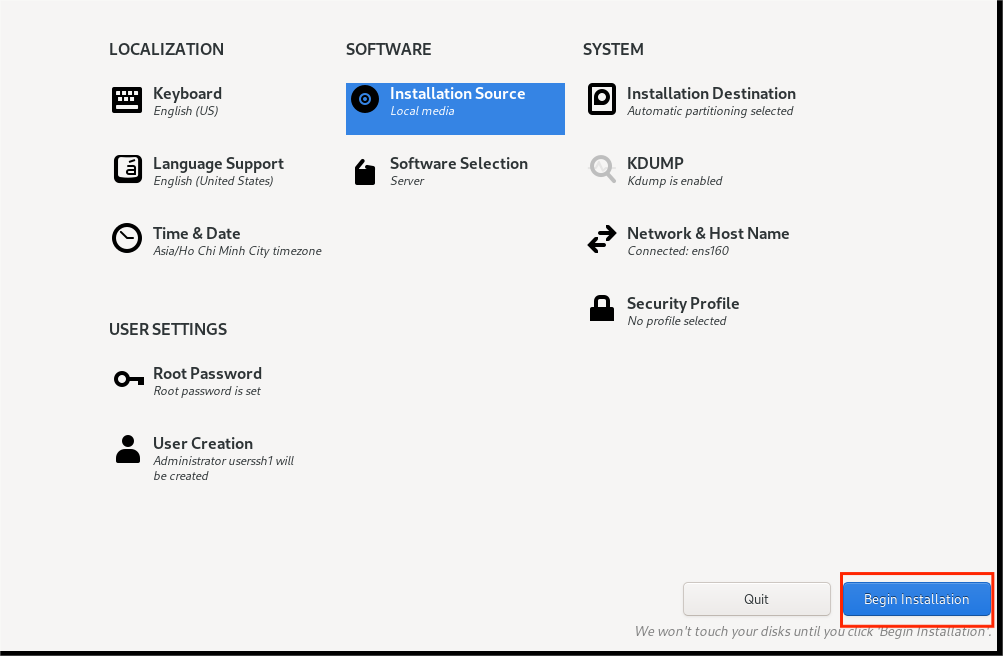
Chuyển method ipv4 thành manual và điền thông tin ip.



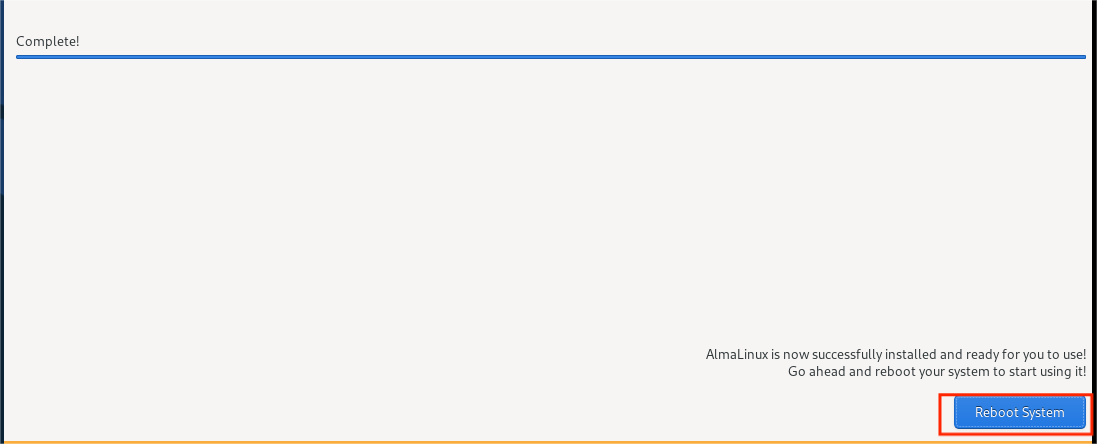
Với time thì chuyển sang giờ vn và lưu ý thời gian phải chính xác



Chọn begin để bắt đầu cài



Khi hoàn thành chọn reboot :



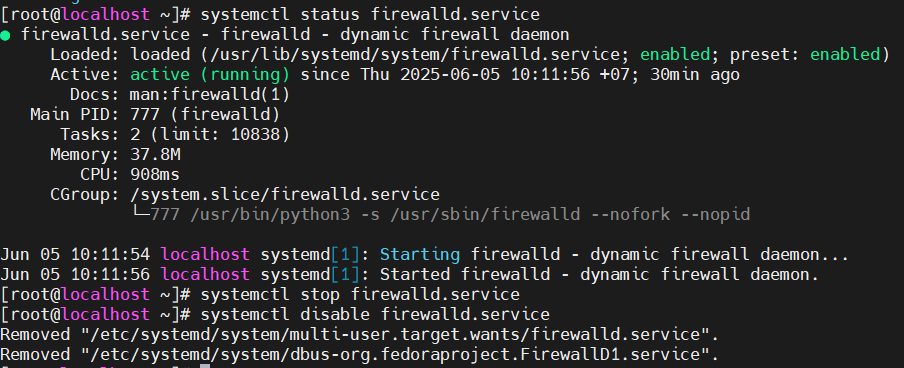
1. **Cài đặt cloudstack**

Lưu ý: tất cả gói cài đều được lấy từ internet nên đảm bảo tất cả host đều đi được tới internet

* 1. Chuẩn bị hạ tầng, thiết bị
     1. Chuẩn bị trên host management

Vì firewalld có thể chặn traffic giữa các thành phần cloudstack nên tắt firewalld:

|  |
| --- |
| Systemctl status firewalld  Systemctl stop firewalld  Systemctl disable firewalld |

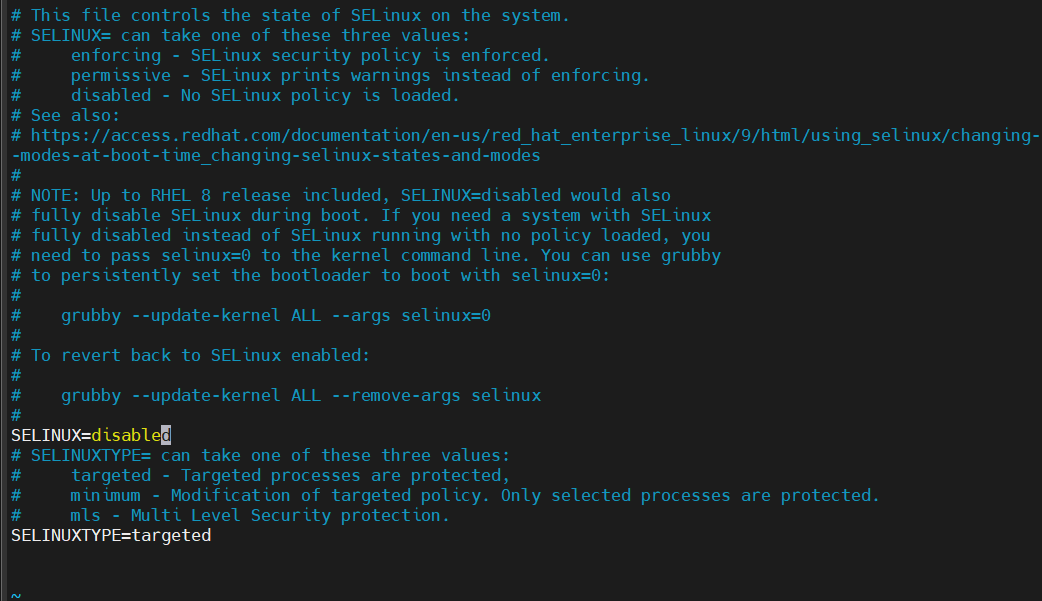


Tương tự selinux cũng có thể chặn không cho bật và export port nên tắt selinux.

Config disable selinux:

Gõ lệnh setenforce

|  |
| --- |
| setenforce 0 |

Sửa trong file theo đường dẫn /etc/selinux/config

Để chia sẻ storage nên cài đặt nfs server:

|  |
| --- |
| dnf -y install nfs-utils |

Config cài đặt thư mục share tại thư mục /etc/exports :



Trong đó: - /data : thư mục share (nếu không có thư mục thì cần tạo thư mục bằng mkdir )

- \* : cho phép tất cả client truy cập

- rw :Cho phép client đọc và ghi vào thư mục /data.

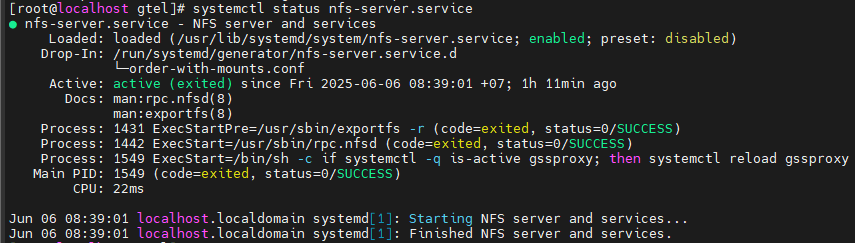
- async :Cho phép NFS ghi dữ liệu không đồng bộ

- no\_root\_squash :Cho phép user trên client vẫn có quyền root khi truy cập NFS

- no\_subtree\_check :Bỏ qua kiểm tra đường dẫn thư mục con.

Bật và kiểm tra service nfs server:

|  |
| --- |
| systemctl enable rpcbind  Systemctl start rpcbind  Systemctl start nfs-server  Systemctl enable nfs-server  Systemctl status nfs-server |



Add package repo cho package cloudstack:

Tạo thêm file tại vị trí file /etc/yum.repos.d/cloudstack.repo và sửa lại nội dung của file

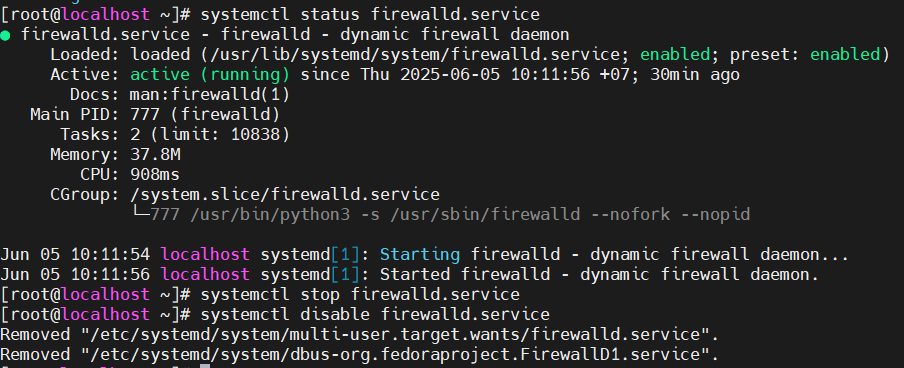
|  |
| --- |
| [cloudstack]  name=cloudstack  baseurl=http://download.cloudstack.org/centos/$releasever/4.20/  enabled=1  gpgcheck=0 |

* + 1. Chuẩn bị trên host agent-kvm

Vì firewalld có thể chặn traffic giữa các thành phần cloudstack nên tắt firewalld:

Config tắt firewalld:

|  |
| --- |
| Systemctl status firewalld  Systemctl stop firewalld  Systemctl disable firewalld |



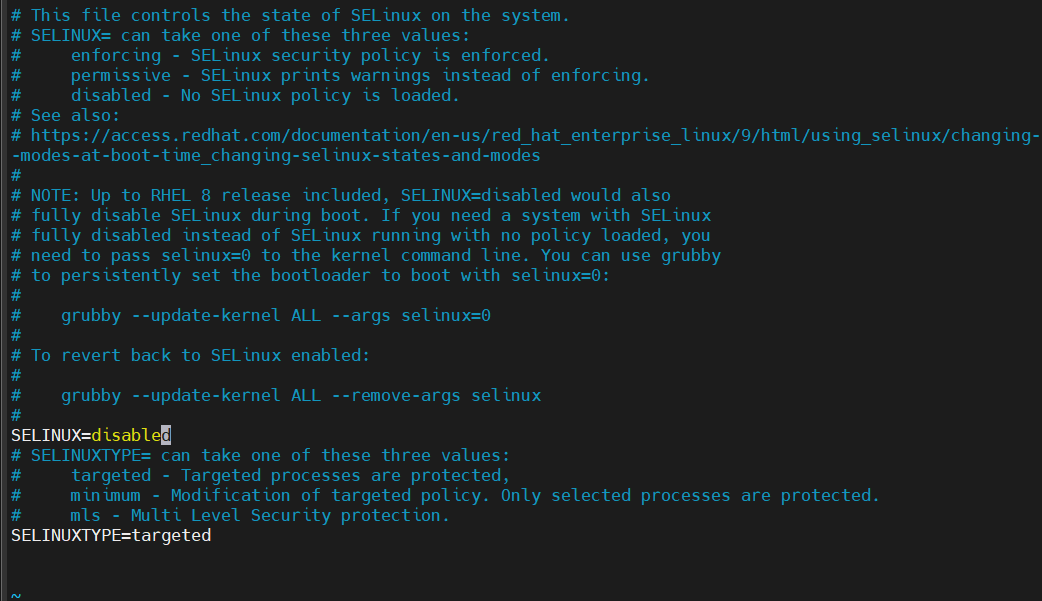
Tương tự selinux cũng có thể chặn không cho bật và export port nên tắt selinux.

Config disable selinux:

Gõ lệnh setenforce

|  |
| --- |
| setenforce 0 |

Sửa trong file theo đường dẫn /etc/selinux/config



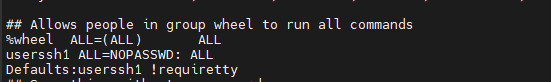
Add package repo cho package cloudstack:

Tạo thêm file tại vị trí file /etc/yum.repos.d/cloudstack.repo và sửa lại nội dung của file

|  |
| --- |
| [cloudstack]  name=cloudstack  baseurl=http://download.cloudstack.org/centos/$releasever/4.20/  enabled=1  gpgcheck=0 |

Gán quyền sudo không cần nhập pass cho userssh1.

Sửa file /etc/sudoers theo cấu hình:



Trong đó userssh1 là user được dùng để ssh vào server khi cấu hình add host của cloudstack

* 1. Cài đặt thành phần của cloudstack
     1. Cài đặt trên host management

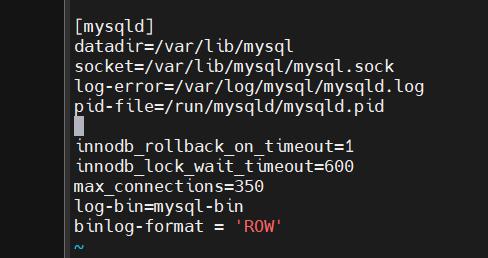
Cloudstack sử dụng mysql server làm db nên cần cài đặt db mysql.

Cài đặt mysql-server:

|  |
| --- |
| dnf -y install mysql-server |

Sau khi cài đặt xong, config nội dung trong file /etc/my.cnf.d/mysql-server.cnf theo nội dung dưới vào mục section [mysqld]:

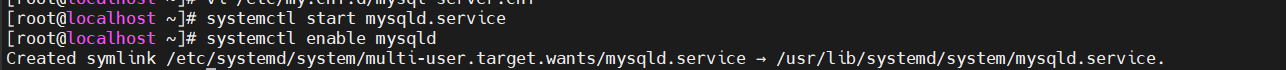
|  |
| --- |
| innodb\_rollback\_on\_timeout=1  innodb\_lock\_wait\_timeout=600  max\_connections=350  log-bin=mysql-bin  binlog-format = 'ROW' |



Giải thích ý nghĩa nội dung config mysqld:

|  |  |
| --- | --- |
| Nội dung | Giải thích |
| innodb\_rollback\_on\_timeout=1 | Thực hiện rollback trạng thái khi bị timeout trong lúc xử lý thao tác dữ liệu |
| innodb\_lock\_wait\_timeout=600 | Thời gian timeout của một thao tác dữ liệu |
| max\_connections=350 | Số lượng connect max tới db cùng lúc |
| log-bin=mysql-bin | Bật khả năng lưu log khi thay đổi dữ liệu trong mysql, các file được đánh thứ tự thực hiện |
| binlog-format = 'ROW' | Định dạng file log ghi lại là theo dòng được thay đổi |

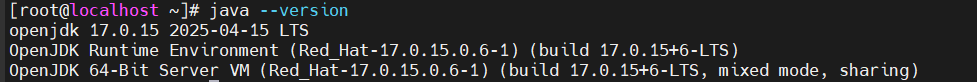
Lưu file và thực hiện bật service mysql



Cài gói cloudstack-management:

|  |
| --- |
| dnf -y install cloudstack-management |

Sau khi cài xong check xem java 17 đã được cài chưa? Phải đảm bảo java 17 đã được cài tự động theo gói cloudstack



Cài đặt cơ sở dữ liệu database mysql để cloudstack sử dụng:

|  |
| --- |
| cloudstack-setup-databases cloud:<dbpassword>@localhost --deploy-as=root:<password> |

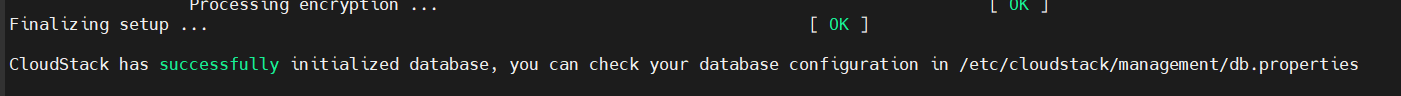
Trong đó:

- dbpassword: password của user cloud trong db

- localhost: địa chỉ của db

- <password>: password của user root

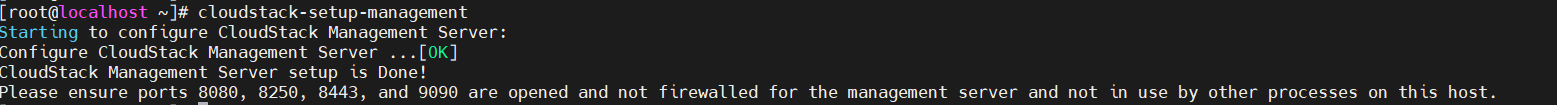
Khi hoàn thành trên màn hình sẽ hiển thị thông báo hoàn thành cài đặt:



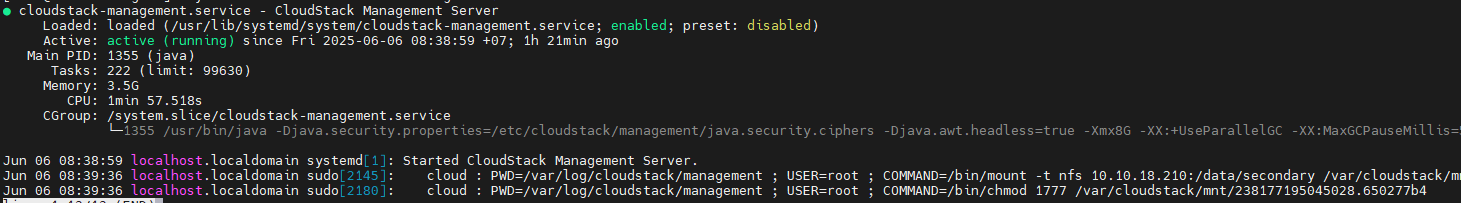
Có thể vào xem thông tin của file config được lưu

|  |
| --- |
| cloudstack-setup-management |

Hoàn thành phần config cloudstack-management



Kiểm tra tình trạng service:



Vào địa chỉ http://<IP-management>:8080 để vào giao diện web của cloudstack

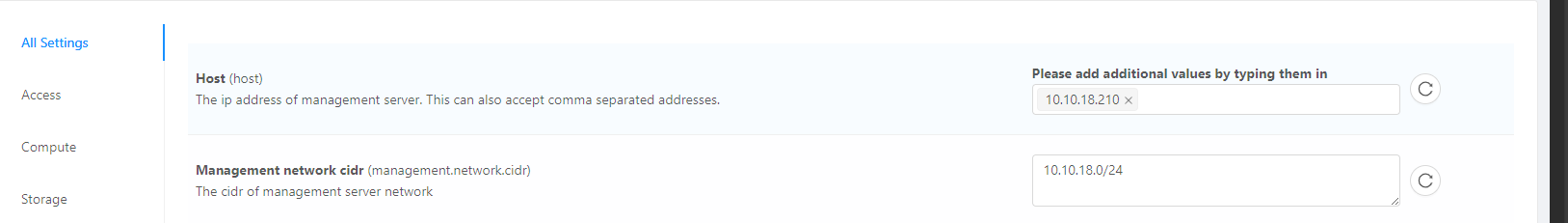
Account admin default

Username: admin

Password: password

Domain: /

Vào configuration --> global settings --> all settings , sửa lại thông tin host và management network cidr như hình, với ip và dải là thông tin của host management



* + 1. Cài đặt trên host agent - kvm

Cấp nhật package và cài đặt cloudstack-agent

|  |
| --- |
| Dnf update && dnf install cloudstack-agent |

1. **Cấu hình cloudstack**
   1. Cấu hình các thành phần chính cloudstack (zone, pod, cluster, host, primary storage, secondary storage, traffic …)

Giải thích khái niệm:

|  |  |
| --- | --- |
| Tên | Giải thích |
| Zone | Vùng tách biệt như một datacenter, bao gồm network, pod, host riêng biệt |
| Pod | Được ví như một tủ rack trong hệ thống datacenter, chứa nhiều host và cluster.  Mỗi pod chứa 1 subnet riêng để phân biệt dùng cho đường traffic management và storage |
| Cluster | Một nhóm các host chung loại hypervisor, bao gồm thông tin primary storage |
| Host | Là máy chủ chứa tài nguyên như ram, cpu,… được cài đặt hypervisor (kvm, vmware,…) |
| Primary storage | Kho lưu trữ vm, chứa toàn bộ data của vm instance được tạo |
| Secondary storage | Lưu trữ file template, iso, backup của vm  Được chia sẻ sử dụng cho toàn bộ các pod trong cùng zone |
| Guest traffic | Là luồng dữ liệu mạng được sử dụng để kết nối giữa các instance sau khi được tạo ra |
| Management traffic | Là dải mạng được sử dụng để quản lý các thành phần hệ thống như system vm, host,… |
| Public traffic | Là dải mạng được sử dụng để export vm ra ngoài internet, dải user thông thường sử dụng để truy cập kết nối tới vm |

Lưu ý: Luồng traffic được giới thiệu là các luồng traffic thường xuyên được sử dụng và bắt buộc có khi cấu hình

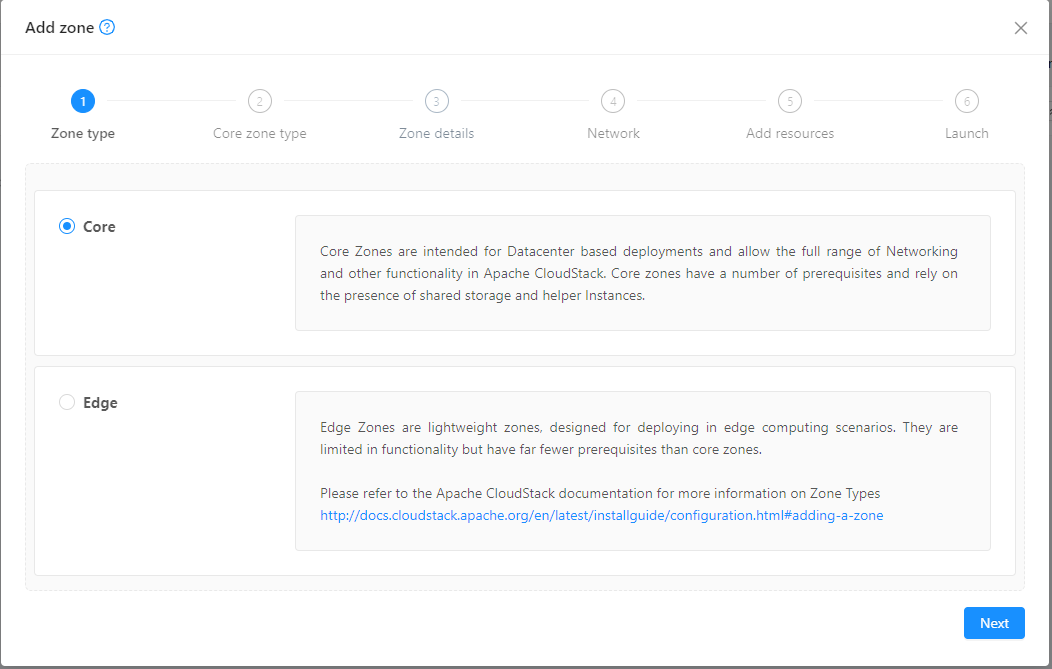
Việc cấu hình có thể thực hiện toàn bộ khi add zone hoặc cấu hình riêng lẻ từng thành phần.

Config tại mục Infrastructure

Thực hiện cấu hình config add zone:

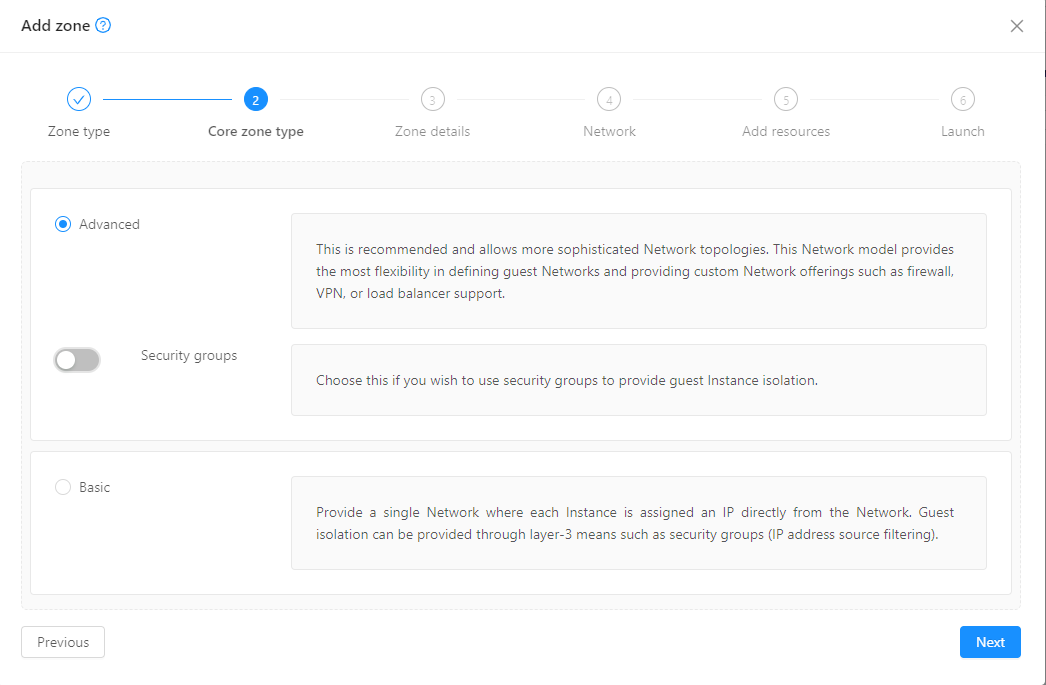
Chọn core vì có thể triển khai đầy đủ cloudstack.

Edge option sẽ config zone không được đầy đủ các thành phần như core setup, system vm sẽ không được cài đặt, các đường traffic network như management, public, guest không được config và chỉ duy nhất 1 hypervisor host được add vào zone



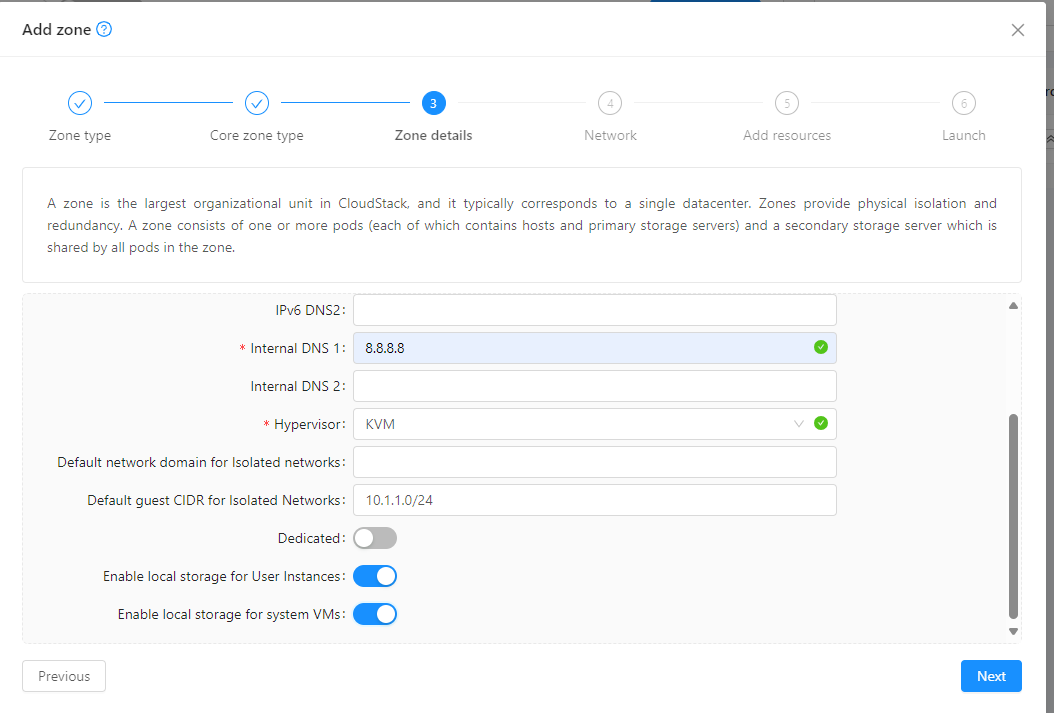
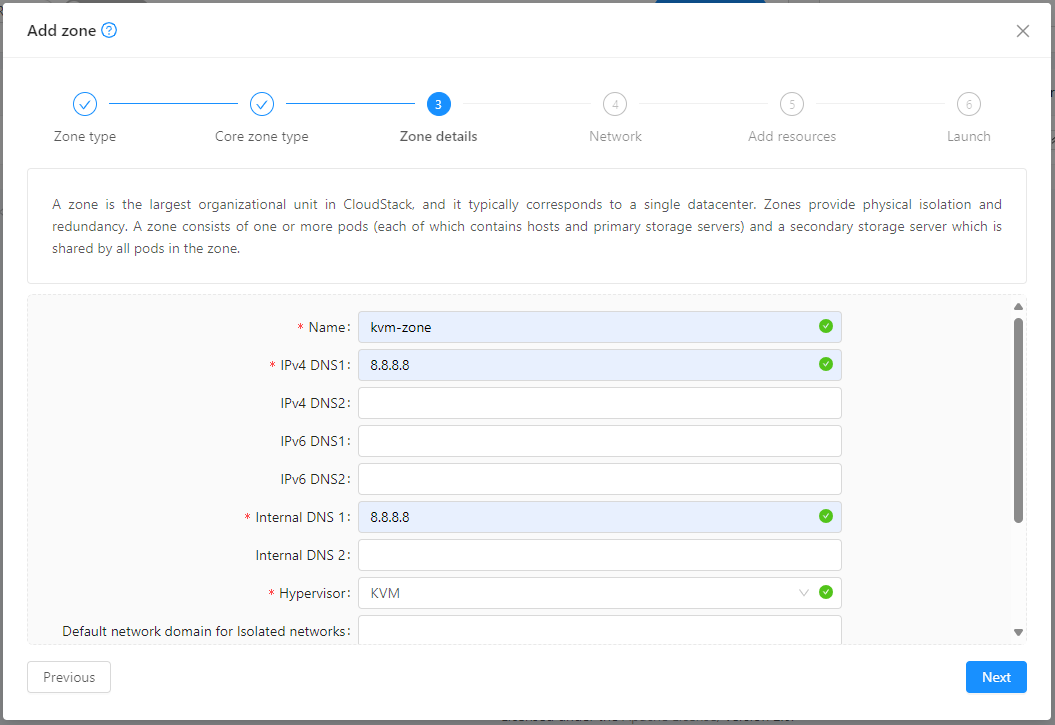
Config zone type advanced cho phép cấu hình chi tiết về lớp network, config vlan, …

Với zone type basic không có sự phân chia rõ ràng về network, các đường traffic được sử dụng chung một dải mạng --> khó cho việc mở rộng sau này



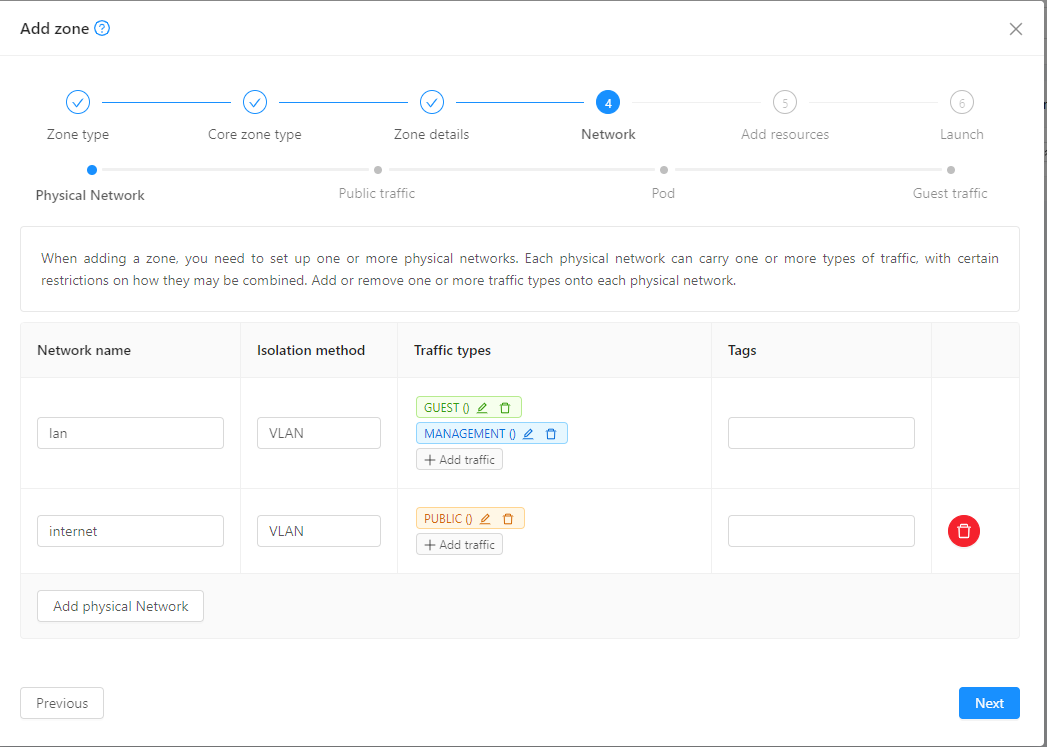
Config cấu hình thông tin dns, loại hypervisor được sử dụng trong các host thuộc zone, dải ip mặc định được gán private cho vm instance tạo ra sau này,một số option config zone:

* Local storage for User Instance: cho phép sử dụng local storage của host để làm nơi lưu trữ primary storage cho vm instance được tạo
* Local storage for System vm: cho phép các system vm được sử dụng local storage để lưu trữ
  + System vm là các máy ảo đặc biệt do hệ thống tự động tạo ra để thực hiện các chức năng quản lý và hỗ trợ hạ tầng ảo hóa.
* Delicated: option tuỳ chọn zone chỉ cho phép một nhóm user chỉ định được sử dụng

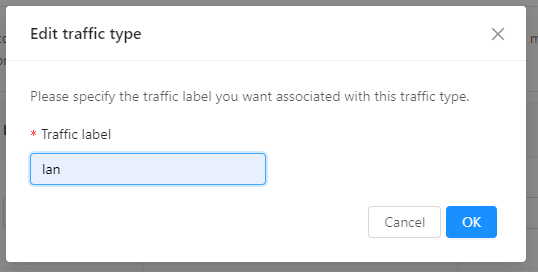


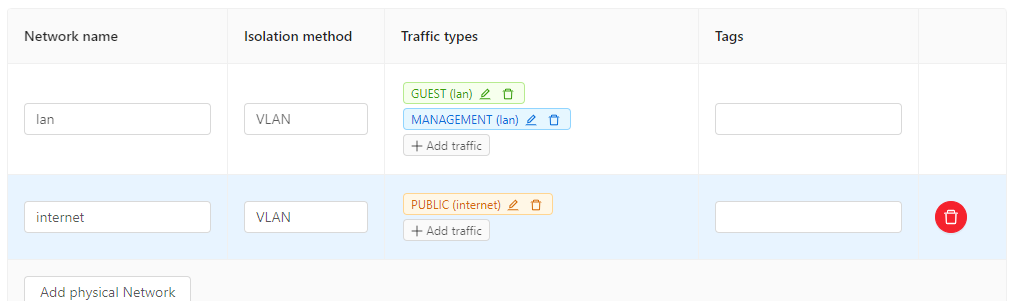
Cấu hình network traffic, mỗi traffic có thể gắn với các đường interface vật lý khác nhau, bắt buộc có 3 dải traffic gồm guest, management và public để cụm zone có thể hoạt động bình thường:

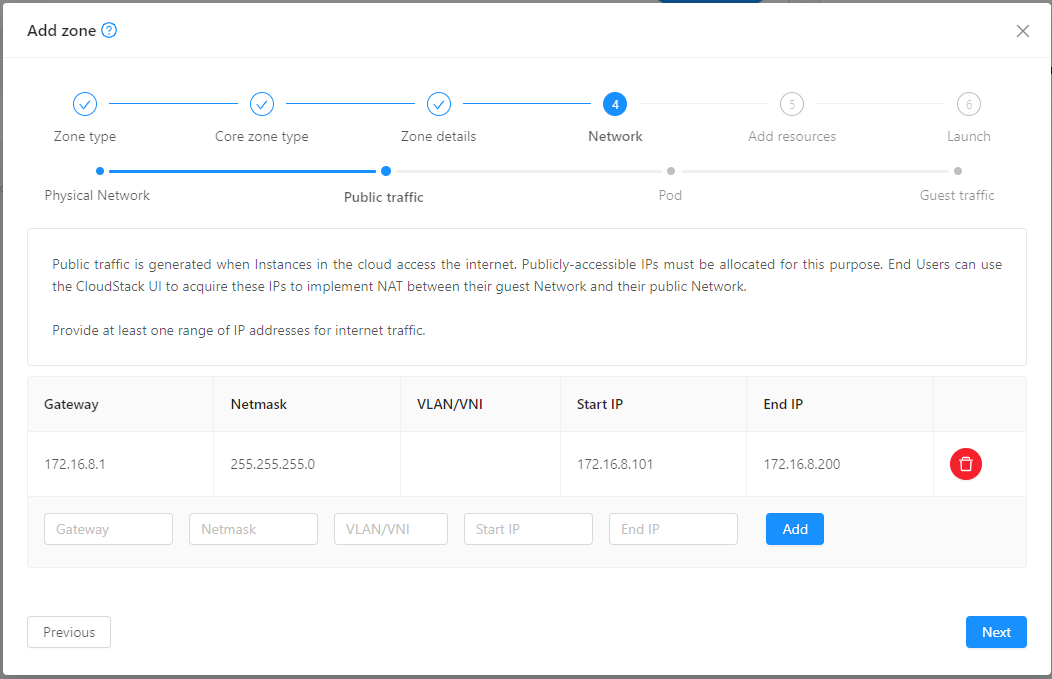
* Guest traffic: Đường traffic nội bộ sử dụng giữa các vm instance
* Management: Đường traffic kết nối quản lý các thiết bị, vm hệ thống quản lý
* Public: Đường được sử dụng trong quá trình export vm instance ra internet, cho phép user thông thường truy cập tới vm instance



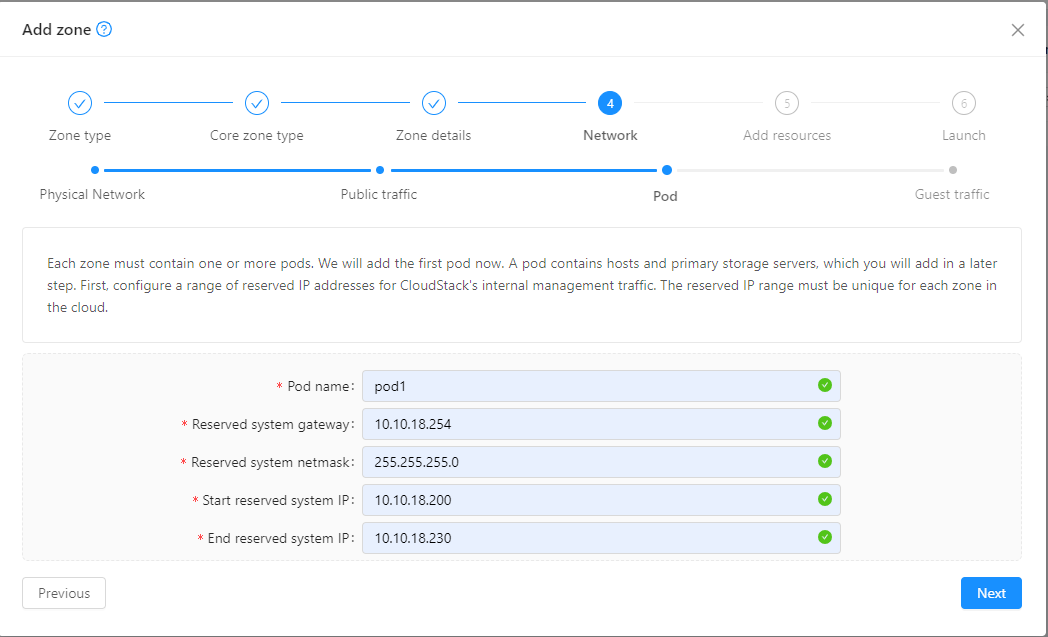
Mặc định cloudstack sẽ gán interface được dùng cho các traffic này là cloud0, chọn hình bút viết bên cạnh thông tin traffic để sửa label (tên interface) và gán interface cho traffic đó





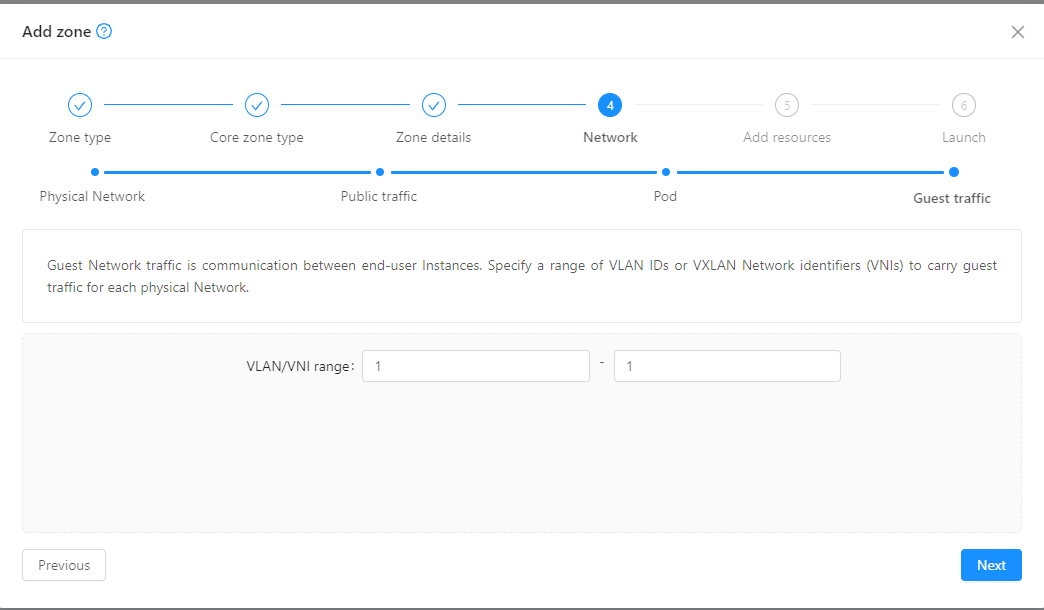
Dải public sẽ được dùng để người dùng bthg truy cập qua giao diện vào các instance sau khi được tạo ra 

Dải revered của pod là dải được gán cho system vm:

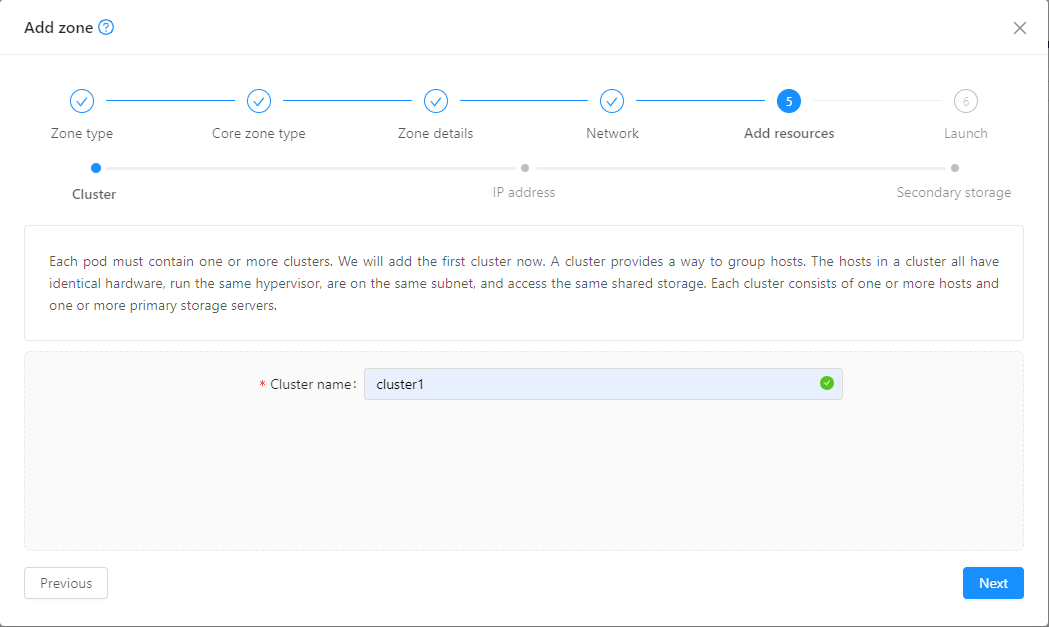


Vlan range được sử dụng cho vlan của guest traffic các vm, có thể thay đổi sau này.

Có thể config range 0-0 để sử dụng untag vlan cho phép truy cập trực tiếp không thông qua vlan giữa các máy



Config cluster, host, storage,…

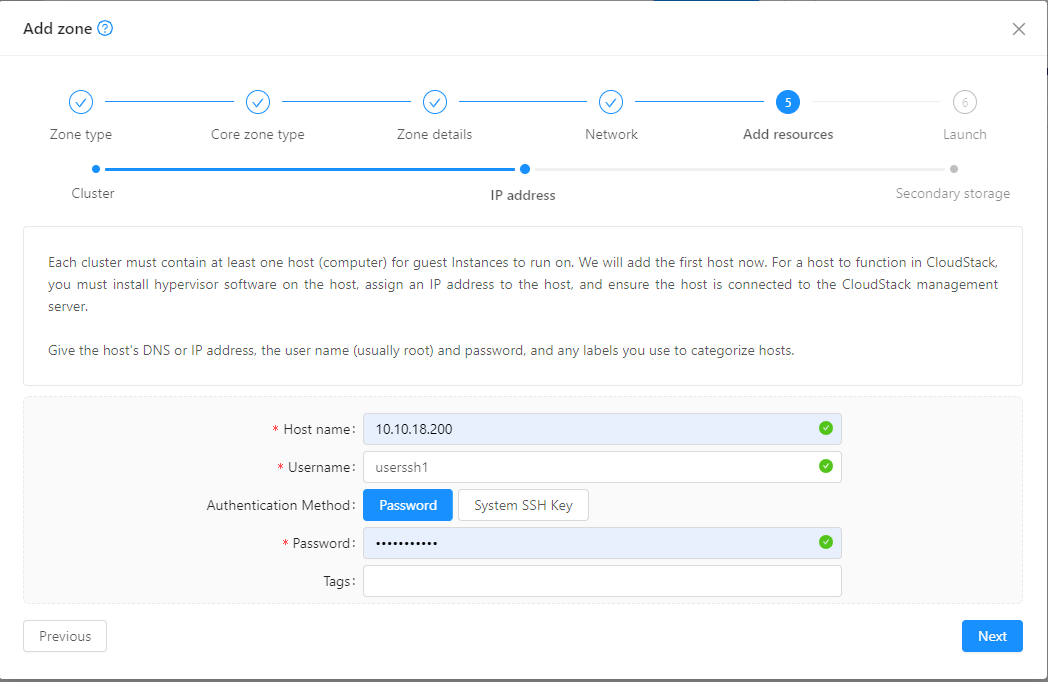


Config thông tin host đầu để add vào zone, cluster, trong đó:

* Hostname: url , ip server hypervisor (kvm)
* Username: username được sử dụng để ssh vào host
* Authentication method: phương thức ssh
* Password: thông tin password ssh dùng username được khai báo

Một phương thức ssh khác là dùng ssh key để ssh, để thực hiện cần copy ssh public key của management sang cho host server:

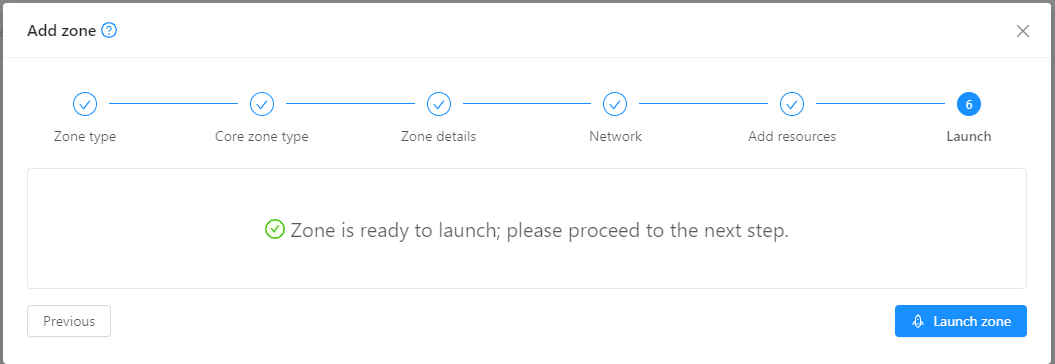
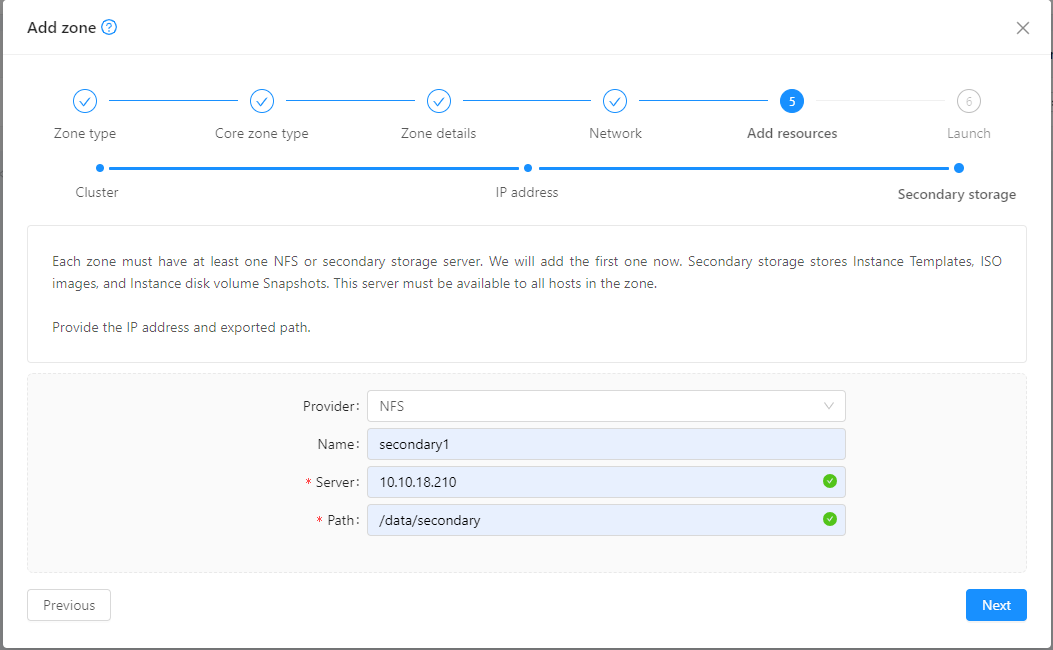
* SSH public key tại đường dẫn /var/cloudstack/management/.ssh/id\_rsa.pub
* Copy nội dụng key sang file /root/.ssh/authorized\_keys ( nếu ssh bằng user root)



Thực hiện add secondary storage, secondary storage thực hiện nhiệm vụ lưu trữ snapshot, backup, image template, iso,…

Điền thông tin server nfs đã chuẩn bị sẵn:

* Provider: phương thức kết nối
* Name: tên secondary
* Server: địa chỉ kết nối, ip
* Path: đường dẫn chia sẻ nfs



Sau khi tạo xong truy cập server agent bổ sung thêm vào cuối file /etc/cloudstack/agent/agent.properties :

|  |
| --- |
| network.bridge.type=openvswitch |

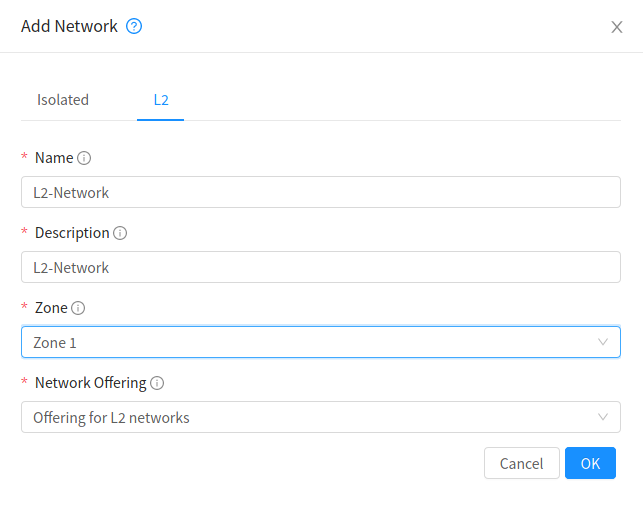
Guest network gồm 3 loại:

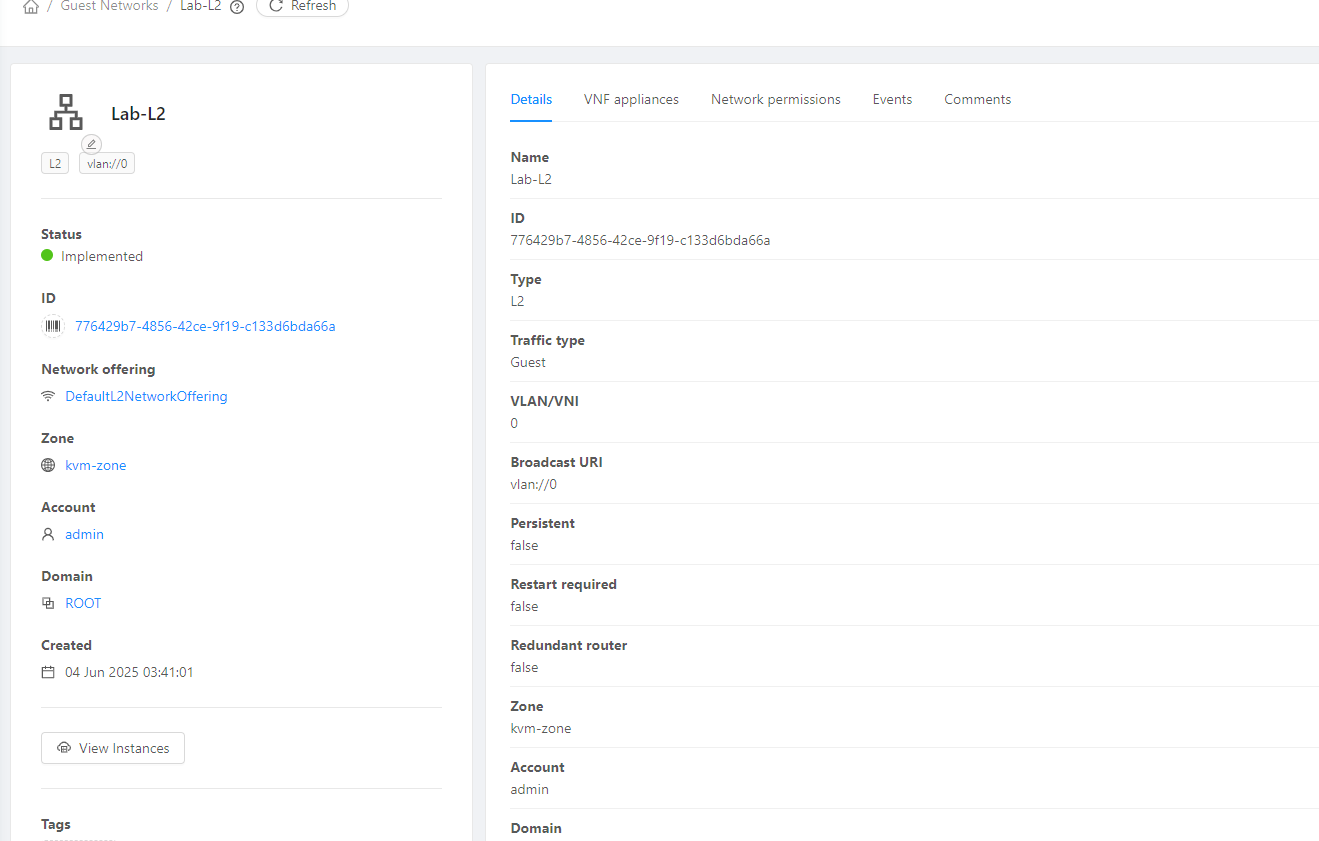
* Isolated Networks: Loại mạng được truy cập bởi user, mỗi user sẽ có mạng riêng biệt, các user khác không thể truy cập, kết nối tới. Dải ip sẽ được cấp qua một router ảo, các dịch vụ fw, Nat cũng được thực hiện qua đó
* Shared Network: Loại mạng cho phép các instance không thuộc cùng user dùng chung.
* L2 network: Network không được trang bị sẵn các dịch vụ khác như fw, router, không có virtual router, instance sử dụng phải tự cấu hình ip,…

Tạo guest network L2 nếu không có đường guest network chính xác tại mục network --> guest network

Cấu hình thông tin theo vlan 0 để đường network sử dụng vlan untag

VD cấu hình network guest, vì sử dụng network offering nên một số thông tin đã được config theo offer đó

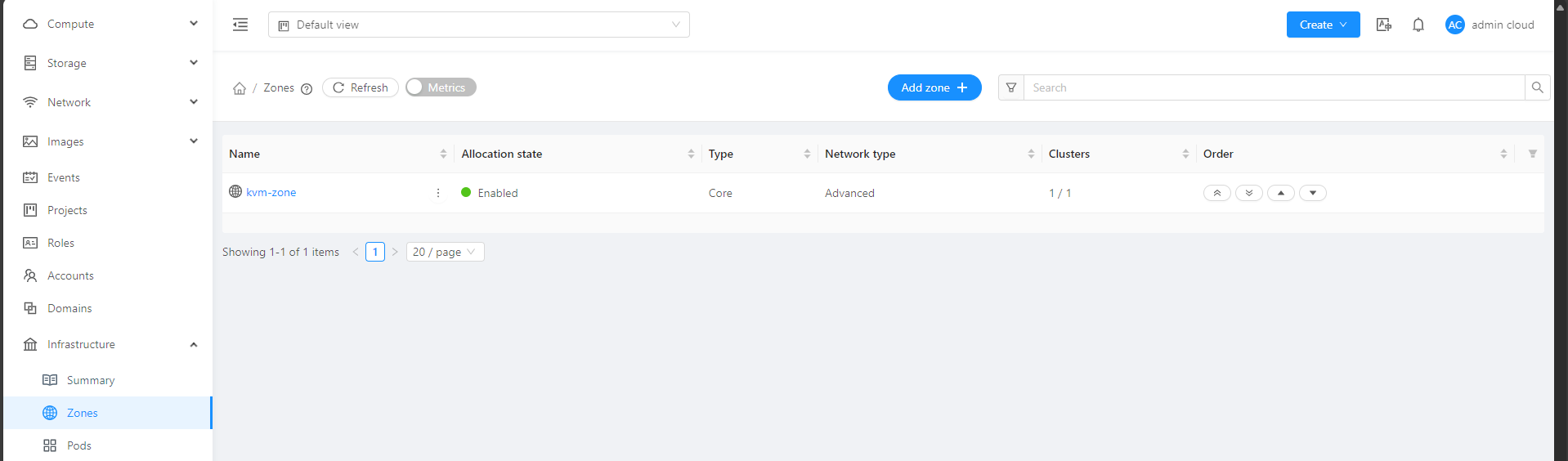




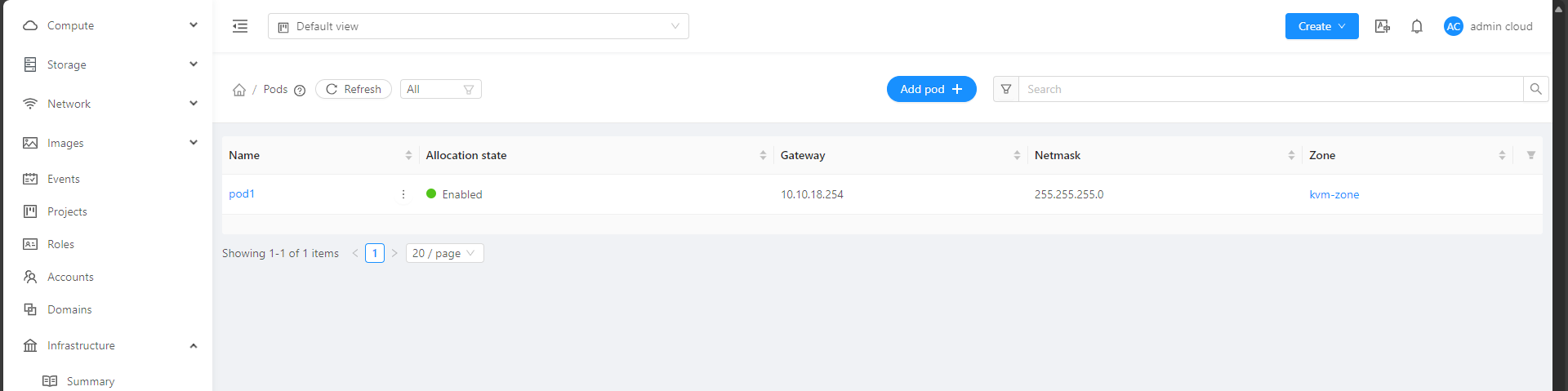
* 1. Rà soát và kiểm tra toàn bộ config chính

Rà soát và đảm bảo các thông tin config đã chính xác

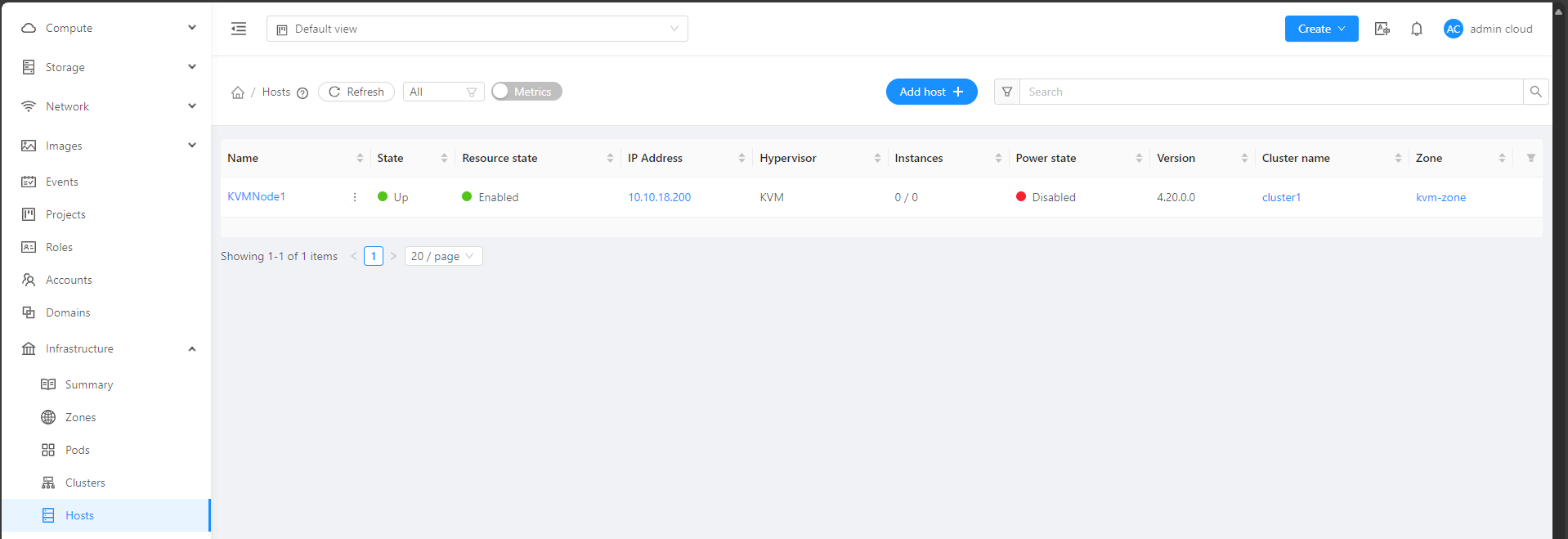
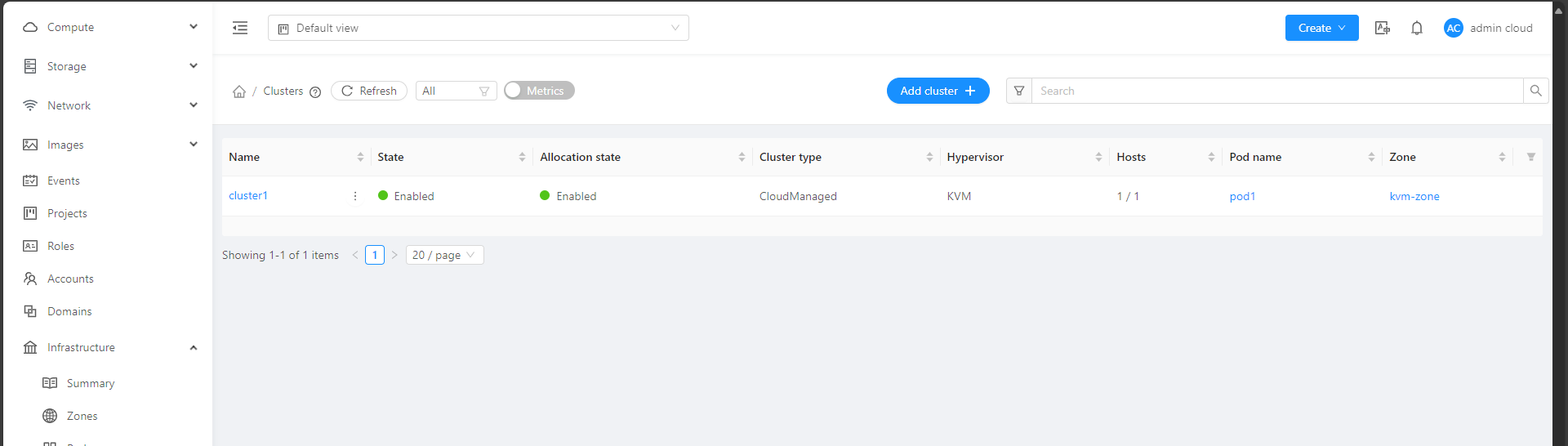
Kiểm tra zone đã enable



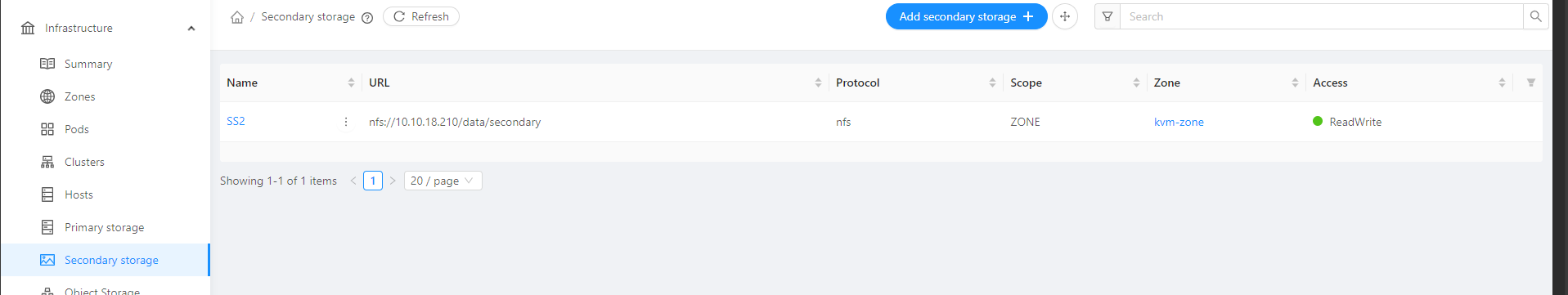
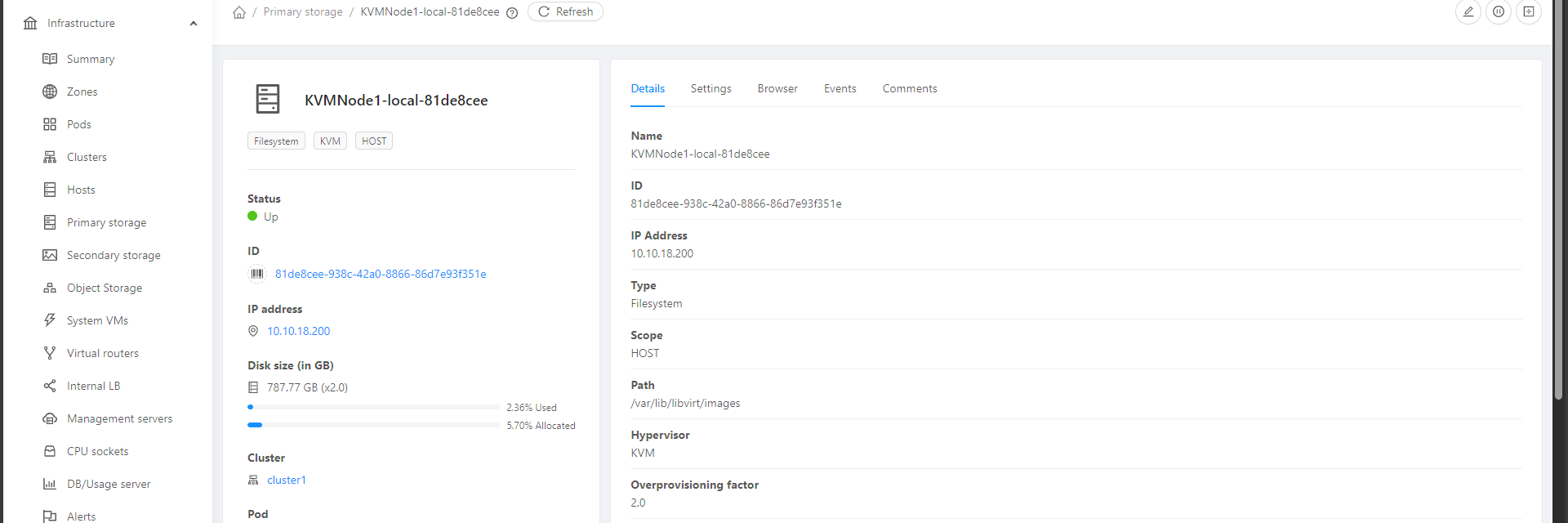
Kiểm tra pod đã enable và các thông tin về zone, network



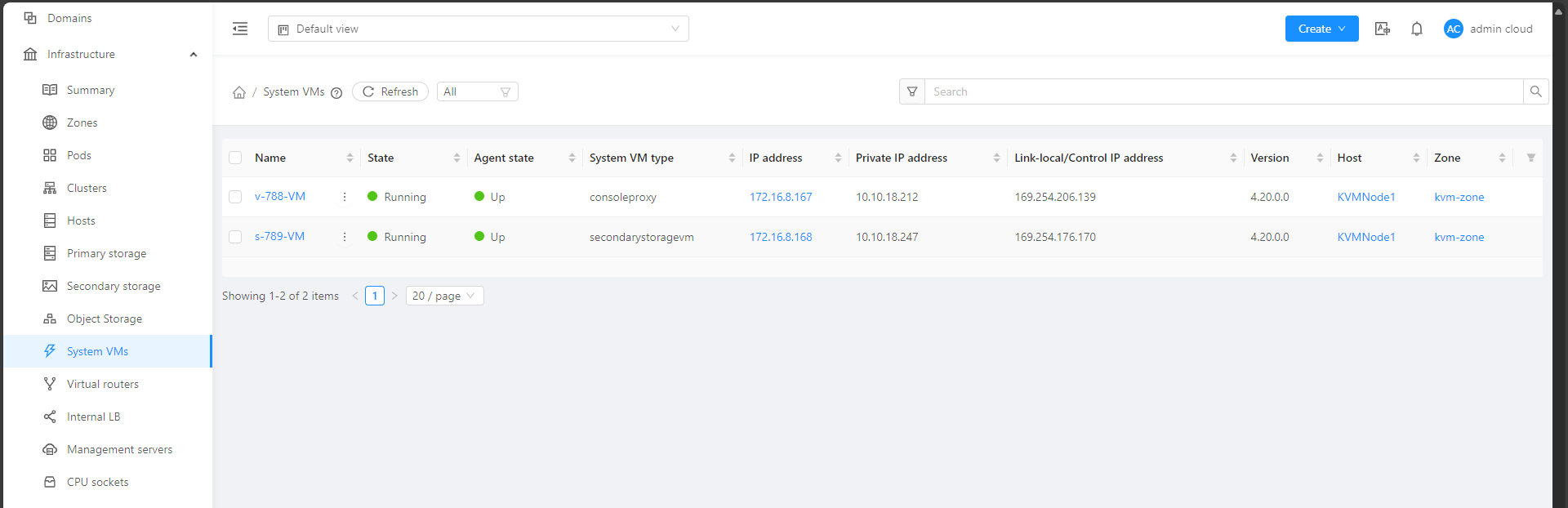
Kiểm tra cluster, số lượng host đã nhận đủ, thông tin loại hypervisor của cluster



Kiểm tra đã có primary storage và secondary storage:



Đảm bảo đã có system vm đang running



Trong trường cần các offer khác ngoài mặc định thì cần cấu hình thêm offer tại mục service offer

Với option disk local thì offer sử dụng cũng phải dùng storage type: local

1. **Vận hành**
   1. Upload template, iso

Sau khi hoàn thành cấu hình cơ bản cho zone, trước khi tạo instance vm thì cần upload template hoặc file iso vm.

Tại mục images --> templates --> Register Template from URL

Có thể lấy template tại : <http://dl.openvm.eu/cloudstack/>

Sử dụng upload bằng url, cloudstack cần có kết nối internet để download template-iso về.

Trong trường hợp không có internet có thể sử dụng module http.server của python nhanh chóng tạo một web server đơn giản để cloudstack kết nối tới và lấy template - iso

Dùng python3 và lệnh để tạo web server share file qua http:

Gõ lệnh tại vị trí thư mục của template/iso:

|  |
| --- |
| Python3 -m http.server 8080 |

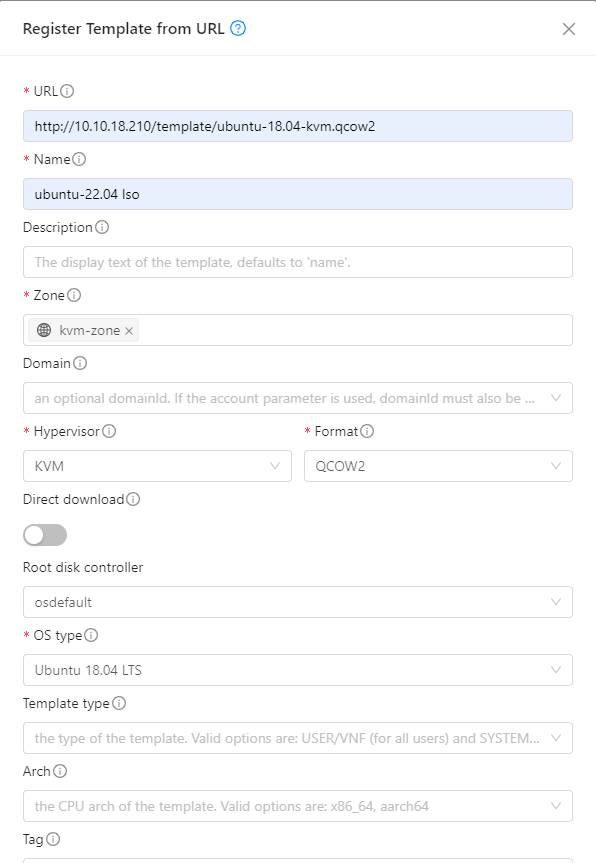
Trong đó 8080 là port được sử dụng để mở web server

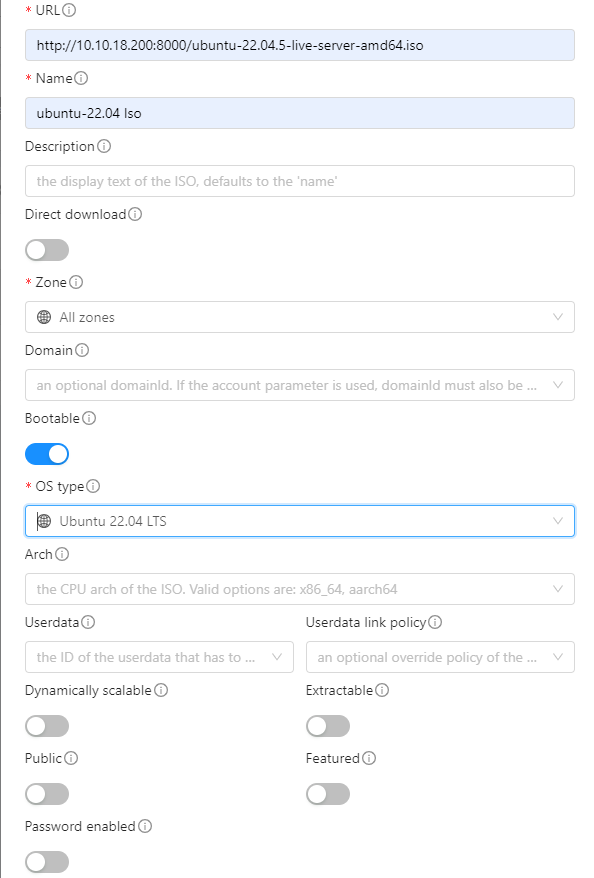


Sau khi server mở port có thể truy cập thông qua web, tìm đúng đường dẫn tới template/iso muốn upload và

Các option có dấu \* là các thông bắt buộc có, bao gồm thông tin đường dẫn url file, thông tin zone được sử dụng, tên template, tên iso, loại file, thông tin hypervisor,…

Root disk controller: sử dụng virtio ( phù hợp với kvm)

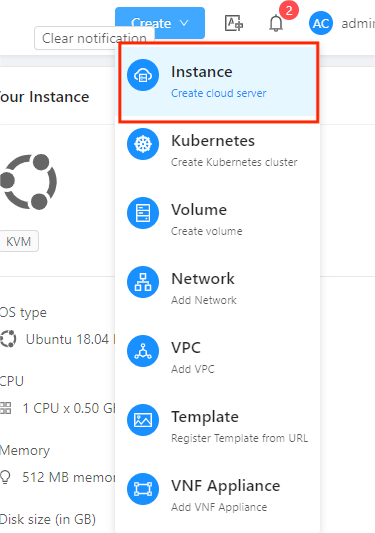




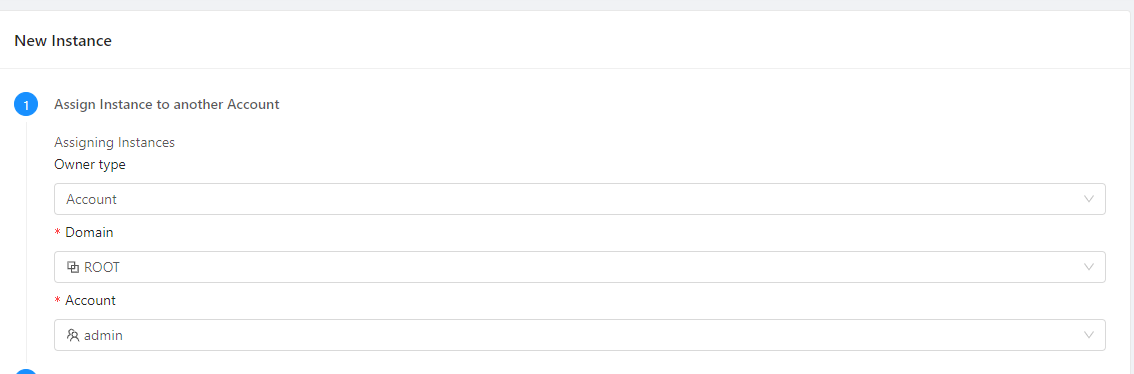
* 1. Tạo vm instance

Trước khi tạo vm instance cần kiểm tra lại:

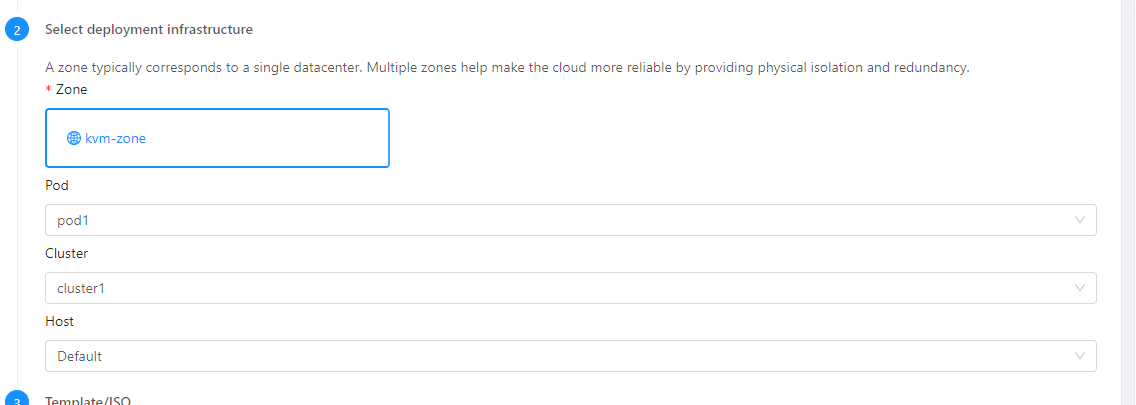
* Zone đã active?
* Storage primary có khả thi ( dùng local hay dạng share storage nfs ,… )
* Template, iso đã có chưa?
* Thông tin đường guest network đã chuẩn xác

Tạo instance bằng nút create 

Mục 1 cho phép config instance tạo gắn cho user khác, domain chỉ định.

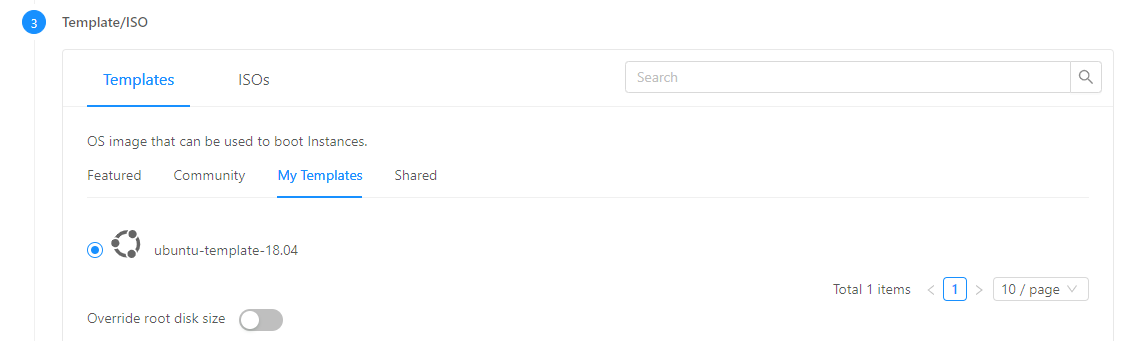


Mục 2 là thông tin khai báo zone, pod, cluster, host instance sẽ thuộc về



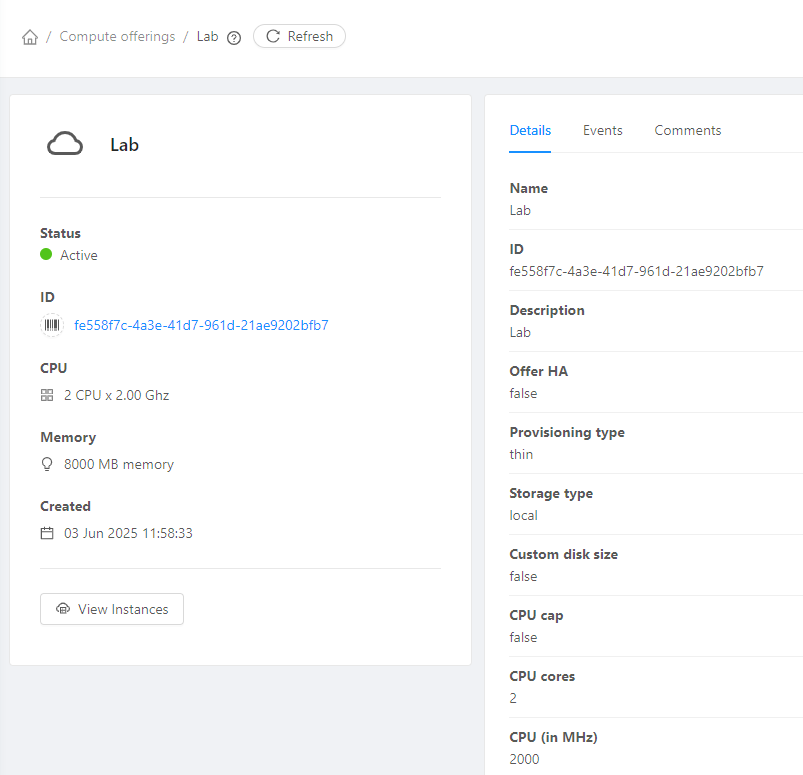
Mục 3 là thông tin về template, iso được sử dụng

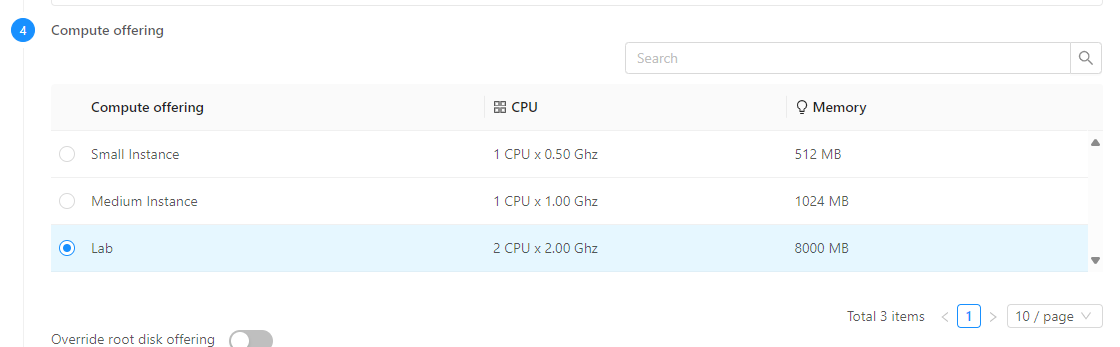
Để tạo instance vm thì cần template hoặc file iso boot để cài vm. Template này được upload trước đó



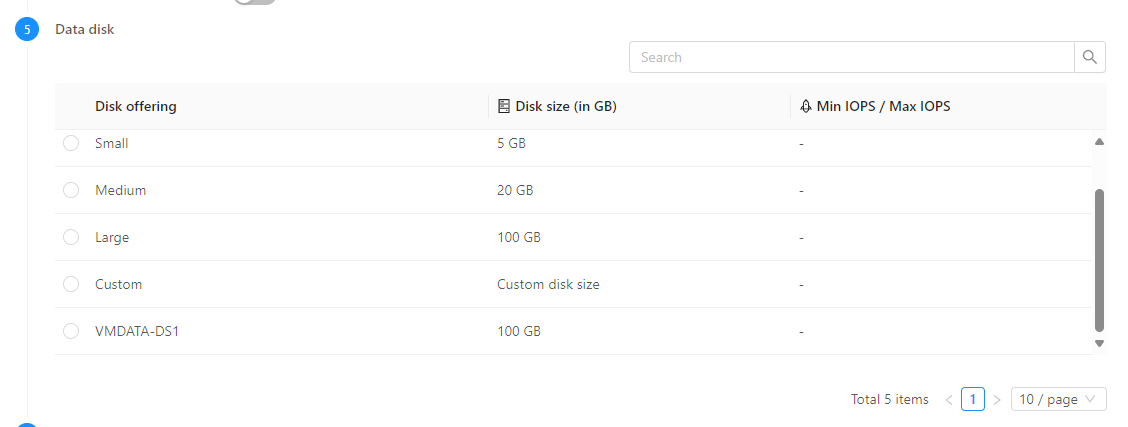
Mục 4 là thông tin về cấu hình cpu, ram, disk root

Lưu ý: về disk nếu muốn sử dụng disk local host thì cần cấu hình compute offering sử dụng storage type là local

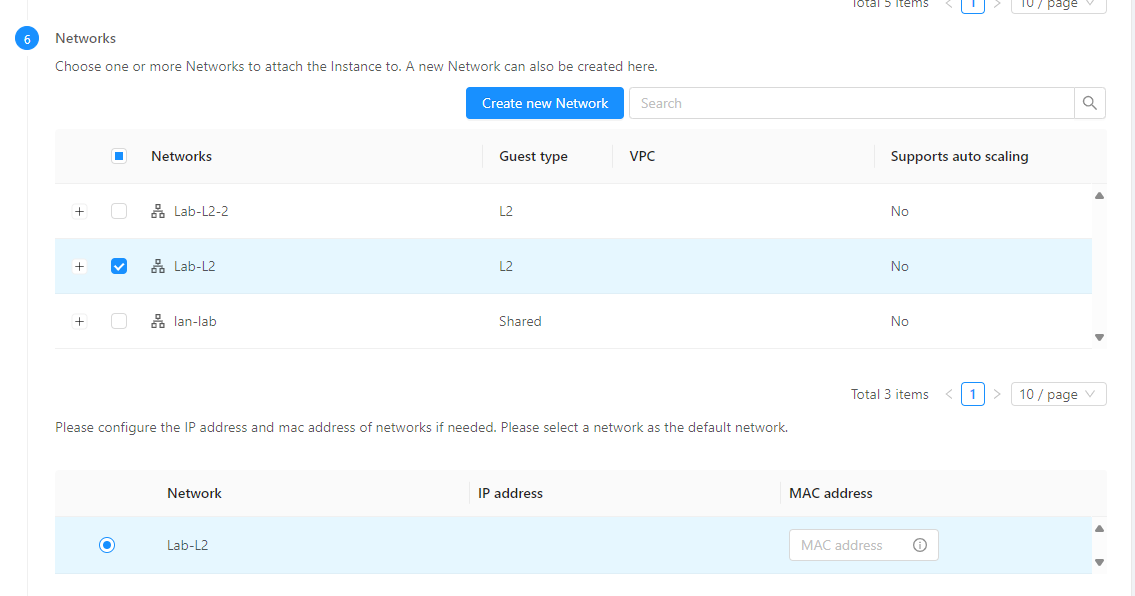


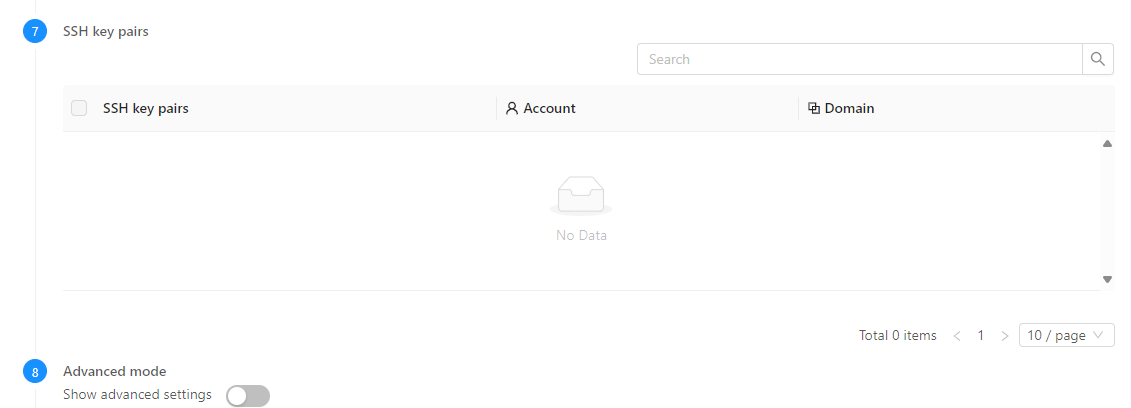


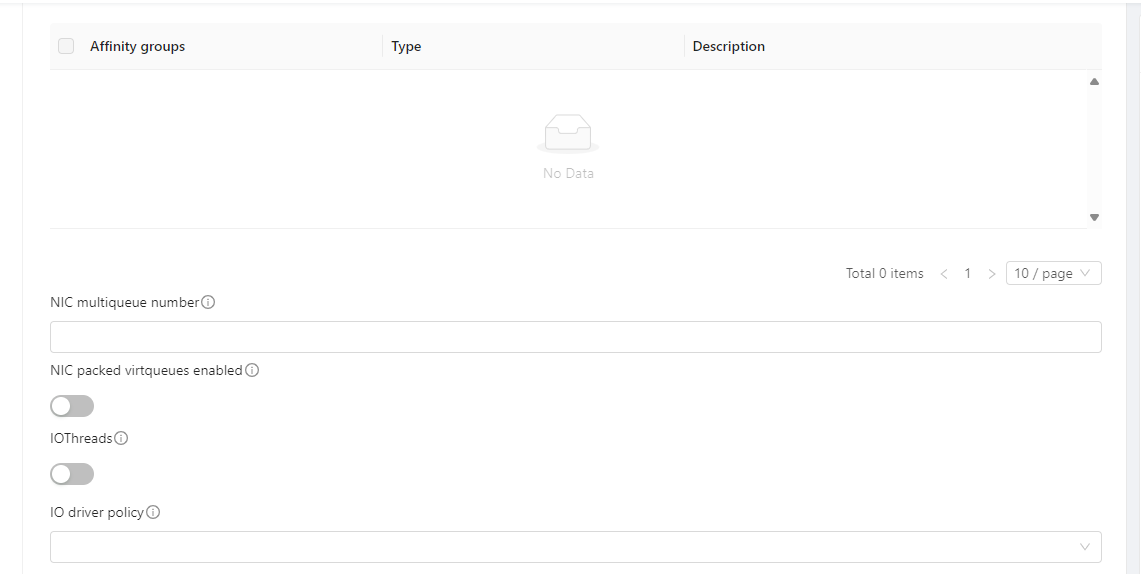
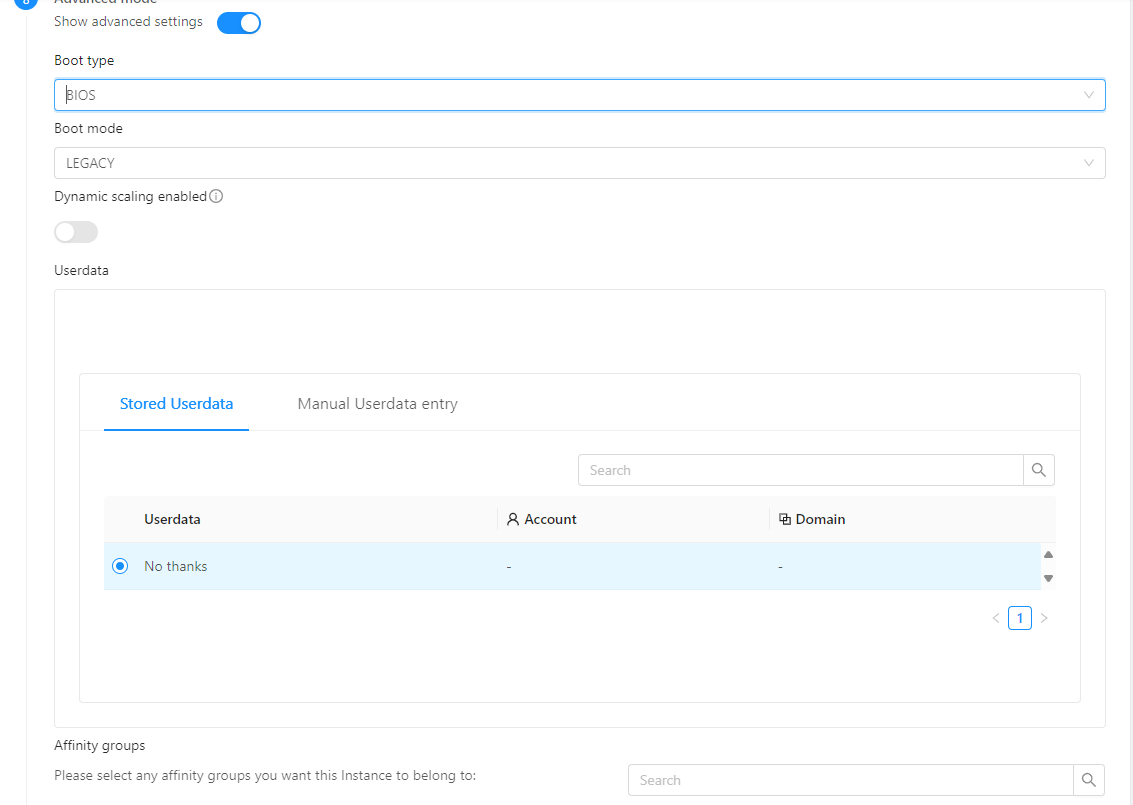
Mục 5 là data disk, tuỳ theo nhu cầu để chọn , mặc định sẽ không sử dụng ( lưu ý tương tự về storage type như mục 4)



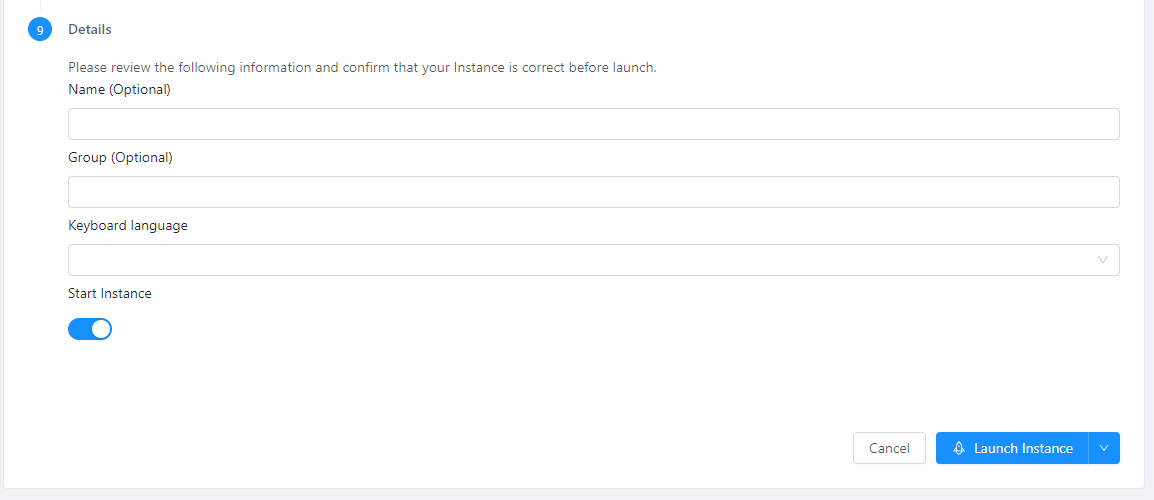
Mục 6 là thông tin guest network được sử dụng, chọn đúng guest network L2 đã config

Mục 7 là thông tin ssh public key được sử dụng để add vào vm

Mục 8 là option tuỳ chọn cấu hình nâng cao bao gồm các thông tin loại boot, config userdata, groupdata, …

Mục 9 là một số thông tin option bổ sung cho instance bao gồm tên instance, group , layout keyboard.

Option start instance cho phép thực hiện start tự động instance vm sau khi tạo



Sau khi tạo xong sử dụng ip private để ssh vào

1. **Phụ lục**
   1. Troubleshoot

|  |  |
| --- | --- |
| Vấn đề | Hướng dẫn |
| Log file của management server | Toàn bộ được lưu tại /var/log/cloudstack/managemnet/ |
| File log của agent server | Toàn bộ được lưu tại /var/log/cloudstack/agent/ |
| Log dịch vụ trên cloudstack | Có thể xem tại mục infrastructure --> alert trên giao diện web |
| Các vm instance được tạo ra không ssh được vào | Đảm bảo sử dụng đúng đường network để ssh và đã config vlan đúng ( nếu không dùng vlan thì phải để là vlan 0) |
| Không có lệnh dhclient | Có thể sử dụng nmtui để config network |
| Không connect, ping được tới 8.8.8.8 , google | Hãy đảm bảo config ip đã chính xác, có 1 đường tới internet  Lỗi có thể nằm ở config route, do có nhiều đường route cho traffic đi nên khi ping thì server không biết đi đường nào đúng  --> bỏ 1 gateway của đường mạng lan đi ( nếu vẫn muốn giữ lại gateway thì cần config chính xác route) |
| Nếu không thấy nút xoá delete | Cho resource vào trạng thái maintain  Load lại trang sẽ xuất hiện nút xoá |
| Mất kết nối host | Các ip phải chính xác và management server phải kết nối dc tới port chỉ định |
| Trong trường hợp systemvm lỗi không start, check log agent không thấy báo lỗi | Xoá system vm bị lỗi để cloudstack tự tạo lại |
| Lỗi systemvm chưa running, agent state ko up | Kiểm tra interface cloud0 đã được up chưa? Nếu chưa thì bật lên   |  | | --- | | Ip link set cloud0 up | |

* 1. Tài liệu tham khảo
* Doc tài liệu chính thức của cloudstack: <https://docs.cloudstack.apache.org/>
* Doc hướng dẫn cài đặt kvm: <https://www.linuxbuzz.com/install-kvm-on-rockylinux-almalinux/>