پیشنهادفنی
زیرساخت ابری

ابر فردوسی

شـرکت ابر فردوسی

آدرس: مشهد، میدان آزادی، دانشگاه فردوسی مشهد، مرکز تحقیقات پیشرفته
شماره تلفن: ۰۵۱-۳۸۸۰۴۲۴۷

فهرست شکل‌ها

فهرست جداول

۱ معرفی زیر ساخت ابری

این روزها بهره‌وری انرژی یکی از دغدغه‌های اصلی بسیاری از شرکت‌ها و سازمان‌ها است و هر چیزی که بتواند به سازمان‌ها در صرفه جویی هزینه‌ها و کارآمدتر شدن انرژی کمک شایانی کند بسیار مهم می‌باشد. زیرساخت ابری آمده است تا هزینه‌های عملیاتی را کاهش و چابکی را افزایش دهد. یکی از برتری‌های زیرساخت ابری در این زمینه، تخصیص منابع است. زیرساخت ابری می تواند به سازماندهی و تخصیص منابع به گونه‌ای کمک کند که فشار بیشتری بر شبکه وارد نشود، اما اطمینان حاصل شود که منابع در دسترس کسانی است که به آن‌ها نیاز دارند. البته مزیت آن کاهش هزینه‌ها ، انعطاف پذیری بیشتر و بهره‌وری بیشتر است. همچنین تخصیص منابع کارآمد بر اساس نیاز کاربر باعث از بین بردن لحظاتی که ظرفیت‌های خالی استفاده نشده وجود دارد می‌شود.

در ادامه به معرفی معماری زیرساخت ابری و بررسی اجزای آن پرداخته شده است.

۱-۱ معماری زیرساخت ابری

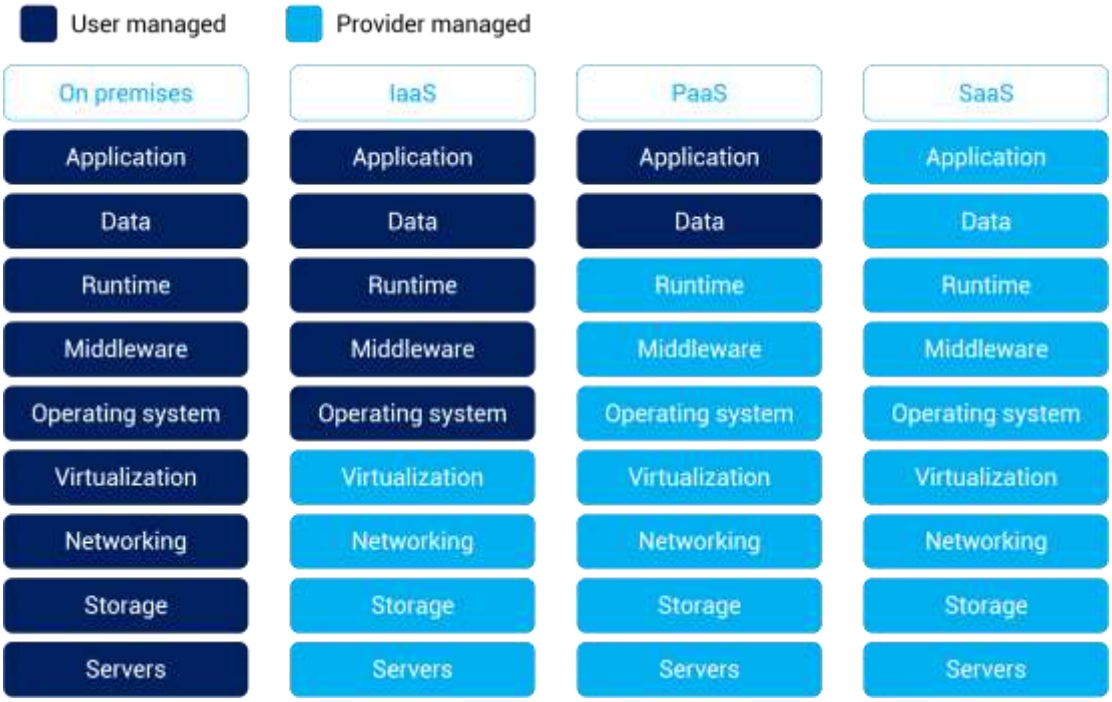
در معماری رایانش ابری معمولا خدمات در سه لایه در اختیار قرار داده می‌شود. این لایه‌ها به شرح زیر می‌باشد.

۱. **زیر ساخت به عنوان خدمت[1](IaaS):** در این نوع از سرویس‌ها پایه‌ترین زیرساخت‌ها مانند ماشین‌های محاسباتی با پیکربندی‌های متفاوت، فضای ذخیره سازی و دیگر منابع سخت افزاری در اختیار کاربران به صورت اجاره‌ای قرار می‌گیرد.

۲. **بستر به عنوان خدمت[2](PaaS):** در این دسته از خدمات نرم افزارها و سرویس‌هایی در اختیار کاربران قرار می‌گیرد تا با استفاده از این سرویس‌ها کاربران بتوانند برنامه‌ها و نرم افزارهای کاربردی ایجاد نمایند. نمونه‌هایی از این موارد عبارتند از سرویس توزیع بار، پایگاه داده‌ها، وب سرورها و سایر مواردی که توسط تولید کنندگان نرم‌افزاری مورد استفاده قرار می‌گیرد.

۳. **نرم افزار به عنوان خدمت[3] (SaaS):** در این دسته نرم افزارهای کاربردی به عنوان خدمات به کاربران ارائه می‌شود.

[1] Infrastructure as a service
[2] Platform as a service
[3] Software as a service



شکل ۱-۱ لایه‌های معماری رایانش ابری

شـــکل ۱-۱ لایه‌های مختلف معماری رایانش ابری را نمایش می‌دهد در لایه زیر ســـاخت به عنوان خدمت(IaaS)، خدمات می‌تواند به صورت خدمات محاسباتی (ماشین مجازی)، خدمات ذخیره‌سازی (فایل سیستم و ذخیره اشیاء) و خدمات شبکه تقسیم شوند. برای پیاده سازی سامانه‌ای که بتواند این خدمات را در اختیار کاربران قرار دهد، پروژه‌های مختلفی تا کنون شـــکل گرفته اســـت، در ادامه به معرفی برخی از مهم‌ترین پروژه‌ها در خصوص پیاده سازی زیرساخت به عنوان خدمت پرداخته شده است.

۱-۱-۱ پروژه OpenStack

پروژه OpenStack یک ســـکوی نرم‌افزاری متن‌باز اســـت که در دنیای رایانش ابری کاربرد دارد. پروژه OpenStack در سال ۲۰۱۰ به‌وسیله سازمان تحقیقات فضایی NASA و شرکت خدمات مرکز داده Rackspace، به عنوان یک خدمت IaaS کلید خورد. هم‌اکنون بیش از ۵۰۰ شرکت به این پروژه پیوسته‌اند و از آن حمایت می‌کنند که این امر ســـبب می‌شـــود که این پروژه احتمالاً محبوب ترین ابزار نرم‌افزاری برای ایجاد محیط‌های IaaS باشد. این پروژه توسط OpenStack Foundation مدیریت می‌شود که یک بنیاد غیرانتفاعی است که در سپتامبر ۲۰۱۲ تأسیس شده‌است.

سکوی OpenStack یک مجازی ساز[1] نیست بلکه روی مجازی‌ساز پیاده‌سازی می‌شود تا بتوان منابع را به بهینه‌ترین شکل ممکن کنترل و استفاده نماید. یکی از مهم‌ترین ویژگی‌های OpenStack این است که منحصر به یک نرم‌افزار یا سخت‌افزار خاص نیست، بنابراین قابل پیاده‌سازی بر روی کلیه مجازی سازها از جمله Hyper-V، Open-vz، Xen، KVM و ESXi می‌باشد.

[1] Hypervisor

۱-۱-۲ پروژه Apache CloudStack

پروژه CloudStack یک نرم‌افزار متن‌باز رایانش ابری برای ساخت، مدیریت و پیاده‌سازی سرویس زیر ساخت ابری است. این پروژه ابتدا به عنوان یک پروژه start-up معروف به VMOps در سال ۲۰۰۸ آغاز شد. این شرکت در نهایت نام خود را به Cloud.com تغییر داد و بسیاری ازکد منابع را در می ۲۰۱۰ تحت مجوز (GPLv3) به CloudStack منتشر کرد. شرکت Cloud.com در جولای ۲۰۱۱ توسط Citrix خریداری شد و بقیه کدهای CloudStack (دوباره تحت GPLv3) در آگوست ۲۰۱۱ منتشر شد. در آوریل ۲۰۱۲، شرکت Citrix دوباره مجوز CloudStack را تحت مجوز نرم افزار Apache 2.0 صادر کرد و CloudStack را به بنیاد Apache ارسال کرد.

این پروژه از محدوده وسیعی از مجازی‌سازها از شرکت های مختلفی همچون مایکروسافت، سیتریکس، KVM و VMware پشتیبانی می کند. بر خلاف OpenStack که بر روی جنبه ابری هسته تمرکز داشت، CloudStack به دنبال فراهم آوردن همه چیز تحت یک منبع واحد است. بیش از ۱۵۰ سازمان شناخته شده از Apache CloudStack (یا توزیع تجاری CloudStack) استفاده می کنند.

۱-۱-۳ پروژه OpenNebula

پروژه OpenNebula يك سكوي نرم افزاري متن باز براي رايانش ابري است و وظيفه كلي آن مديريت تهيه ماشين‌هاي مجازي براي فراهم نمودن زير ساخت به عنوان سرويس در رايانش ابری است و جايگزيني مهم براي افرادي است كه نمي‌خواهند از ابرهاي تجاري استفاده نمايند. اين سكو به ميزان زيادي از قابليت سفارشي سازي براي راهبران و كاربران پشتيباني مي‌نمايد . اين پروژه ابتدا در سال ۲۰۰۵ به صورت یک پروژه تحقیقاتی توسط Ignacio M. Llorente و Ruben S. Montero شروع شد و سپس اولین انتشار این نرم‌افزار در سال ۲۰۰۸ به منظور مدیریت ماشین‌های مجازی بر روی زیرساخت‌های توزیع شده، اتفاق افتاد.

این سکو تحت دو نسخه Community و Enterprise توسعه داده شده که نسخه CE، متن‌باز و تحت مجوز نسخه ۲ آپاچی و نسخه‌ی EE به صورت تجاری می‌باشد.

سکوی OpenNebula دارای چند بخش پایه می‌باشد که در ادامه معرفی خواهند شد.

- **Host:** ماشین‌های فیزیکی که لایه مجازی ساز راپشتیبانی می‌کنند.
- **Template:** تعریف ماشین مجازی که به آن base هم می‌گویند.
- **Virtual Machine**: به نمونه‌های ایجاد شده از template اشاره دارد.
- **Cluster** : مجموعه‌ای از ماشین‌های فیزیکی که میان آن‌ها منابع ذخیره‌سازی و شبکه به اشتراک گذاشته شده است.
- **Image:** به دیسک‌های یک ماشین مجازی گفته می‌شود.
- **Virtual Network**: گروهی از IPها که به طور اتوماتیک به ماشین‌های مجازی تخصیص داده می‌شود. این مورد شبکه‌های مجازی را ایجاد می‌کند تا vmها بتوانند با دنیای بیرون و بین خودشان ارتباط برقرار کنند. OpenNebula با یک virtual router appliance نیازهای شبکه خود را پاسخ می‌دهد.

۴-۱-۱ VMware Cloud Suite

در واقع VMware vCloud Suite مجموعه‌ای است از آنچه که برای برپا سازی یک زیرساخت ابری به آن نیاز است، و مجموعه‌ی نرم افزارهای شرکت VMware در زمینه رایانش ابری می باشد. این زیرساخت ابری بر پایه قلب مجازی سازی این شرکت (vSphere, ESXi) بنا نهاده شده است. در این معماری، سرویس‌های مبتنی بر محاسبات ابری، شبکه، امنیت و دسترسی ارائه می‌شود. خودکار سازی امور به صورت هوشمند و تنظیمات بر پایه سیاست‌های از پیش تعین شده یک مرکز داده پویا می‌باشد.

اجزای اصلی تشکیل دهنده VMware vCloud Suite به شرح زیر می‌باشد:

- **VMware vSphere :** قلب مدیریت و برپا سازی مجازی سازی و رایانش ابری شرکت VMware است.
- **VMware vCenter Site Recovery Manager :** سامانه‌ای مناسب برای مواقع بحران که در کوتاه‌ترین زمان بازگردانی و سلامت مجدد سیستم را ضمانت می‌کند و مجهز به سیستم بازیابی هوشمند و خودکار است.
- **VMware vCloud Networking and Security :** زیرساخت شبکه مجازی را می‌توان با این نرم افزار برپا سازی نمود و با استفاده از VMware vShield امنیت این ساختار را تضمین نمود.
- **VMware vCloud Automation Center :** خودکار سازی به صورت هوشمند و بر اساس سیاست‌های از پیش تعین شده مهم‌ترین ویژگی آن است.
- **VMware vCenter Operations Management Suite :** یکپارچه سازی، بالابردن کارایی و سرعت، استفاده بهتر از منابع سیستم و مدیریت پویا و پایا از قابلیت‌های این بسته نرم‌افزاری می‌باشد.
- **VMware vCloud Director:** توسعه و برپا سازی رایانش ابری به صورت درون سازمانی و برون سازمانی یا عمومی با ایجاد دسترسی‌های گوناگون انتظاری است که این نرم‌افزار برآورده می‌نماید.

اجزای تکمیل کننده VMware vCloud Suite به شامل موارد زیر می‌باشند:

- **VMware Virtual SAN :** راهکار نرم افزاری ذخیره‌سازی شرکت VMware است که به سادگی منابع محلی را بین چندین مرکزداده به اشتراک می‌گذارد. در مجموع راهکاری ارزان و کارا برای شرکت‌های کوچک و سازمان‌های بزرگ است.
- **VMware NSX :** امنیت جامع شبکه‌های مجازی و مراکز داده است که کاملا نرم افزاری بوده و جدا از سخت افزار فعالیت می‌نماید. در واقع SDN شرکت VMware است که با نام NSX معرفی شده است.
- **VMware vCenter Log Insight :** تجزیه و تحلیل رویدادها رخ داده در کل مرکزداده و زیرساخت بصورت کاملا مجتمع و کارا.
- **VMware IT Business Management :** محاسبه هزینه نرم افزارها و سرویس‌های رایانش ابری را بر عهده دارد.

۲ معماری پیشنهادی

در بخش ۱-۱ به معرفی پروژه‌های مختلف در خصوص پیاده‌سازی زیرساخت ابری پرداخته شد. از بین پروژه‌های معرفی شده پروژه OpenStack به دلایل زیر به عنوان راهکار پیشنهادی برای پیاده‌سازی زیرساخت ابری انتخاب گردیده است.

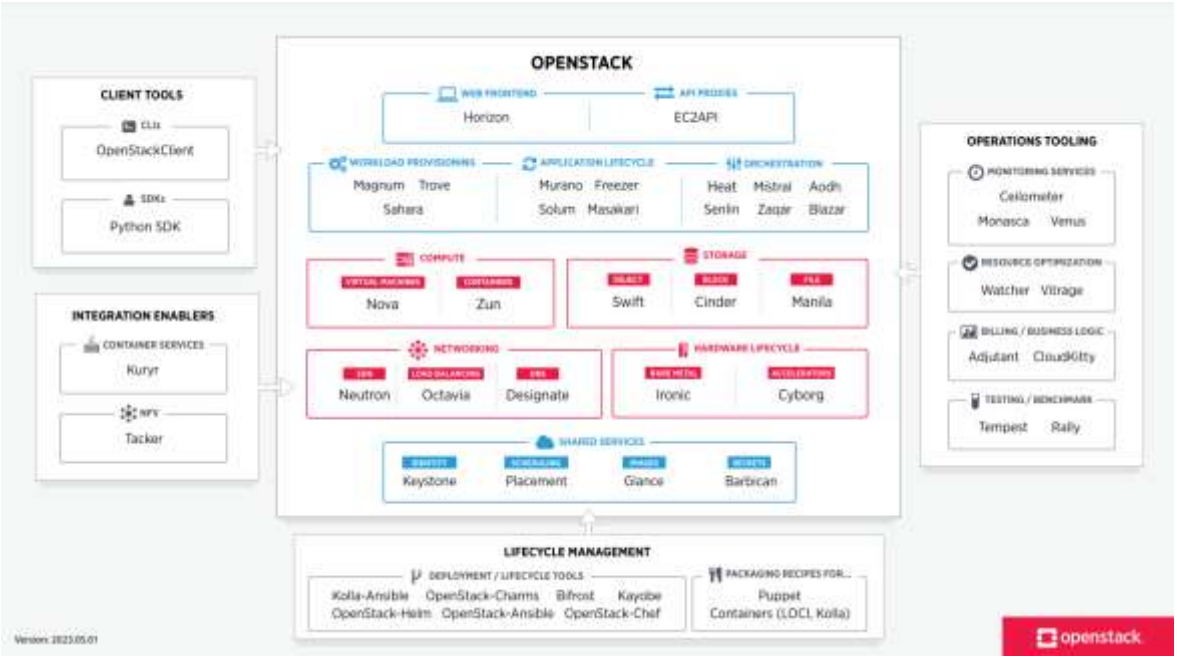
- تجربه‌های موفق در استفاده این ابزار برای سرویس IaaS
- متن باز بودن پروژه و امکان سفارشی سازی تمامی مولفه‌های آن براساس نیاز و کاربرد خاص
- عدم وابستگی به یک Vendor خاص
- قابلیت ارایه سرویس به صورت Bare-Metal و هم به صورت Virtual Machine و همچنین سرویس های Container-based
- استفاده گسترده در شرکت‌های تجاری بزرگ و همچنین مراکز تحقیقاتی مهم مانند CERN و NASA
- پایدار بودن پروژه و تعداد بالای مشارکت کننده‌های آن
- یکپارچه سازی بسیار بالای این پروژه با سایر ابزارهای متن باز
- وجود نیروی متخصص در دسترس برای توسعه و پیاده سازی این ابزار

در ادامه این فصل ابتدا برخی از مهم‌ترین پروژه‌های تشکیل‌دهنده OpenStack معرفی و سپس به بیان طراحی زیرساخت ابری مبتنی بر بستر OpenStack پرداخته می‌شود.

۲-۱ معرفی پروژه‌های اصلی OpenStack

سکوی OpenStack مجموعه‌ای از پروژه‌های گوناگون می‌باشد که هریک از آن‌ها مسولیت خاصی را برعهده دارند. به عنوان مثال پروژه Nova وظیفه خدمات محاسباتی ، پردازشی و مدیریت ماشین مجازی را برعهده دارد و پروژه Neutron وظیفه خدمات شبکه و دیوارآتش را برعهده دارد.

تعداد این پروژه‌ها براساس نسخه انتشار یافته OpenStack متفاوت می‌باشد اولین نسخه OpenStack یعنی Austin که درسال ۲۰۱۰ انتشار یافت فقط شامل دو پروژه Nova وSwift بود درحالی‌که جدیدترین نسخه پایدار انتشار یافته از OpenStack (در زمان نگارش این سند) شامل ۴۱ عدد پروژه مختلف می‌باشد. کاربرد هریک از این پروژه‌ها مختلف می‌باشد و در زمان استقرار زیرساخت لزومی به نصب همه این پروژه‌ها وجود ندارد و براساس نیاز می‌توان انتخاب نمود که کدامیک از این پروژه‌ها نصب گردد. لازم به ذکر است که بین برخی از این پروژه‌ها وابستگی‌هایی وجود دارد و باید درهنگام نصب این نوع از نیازمندی‌ها رعایت گردد.



شکل ۱-۲ پروژه‌های تشکیل دهنده سکوی OpenStack

معماری OpenStack به صورت پیمانه‌ای می‌باشد یعنی هر فرد و یا موسسه‌ای می‌تواند بنا به نیاز خود کدها و مولفه‌های آن را کم، زیاد و دستکاری کند. متن باز بودن، پیمانه‌ای بودن و وجود پروژه‌های مختلف در آن منجر به محبوبیت و کاربردهای گوناگونش در عرصه‌های مختلف شده است. شکل ۱-۲ پروژه‌های تشکیل دهنده OpenStack را براساس نوع کاربردشان نمایش می‌دهد. در ادامه برخی از مهمترین پروژه‌های این سکو توضیح داده شده است.

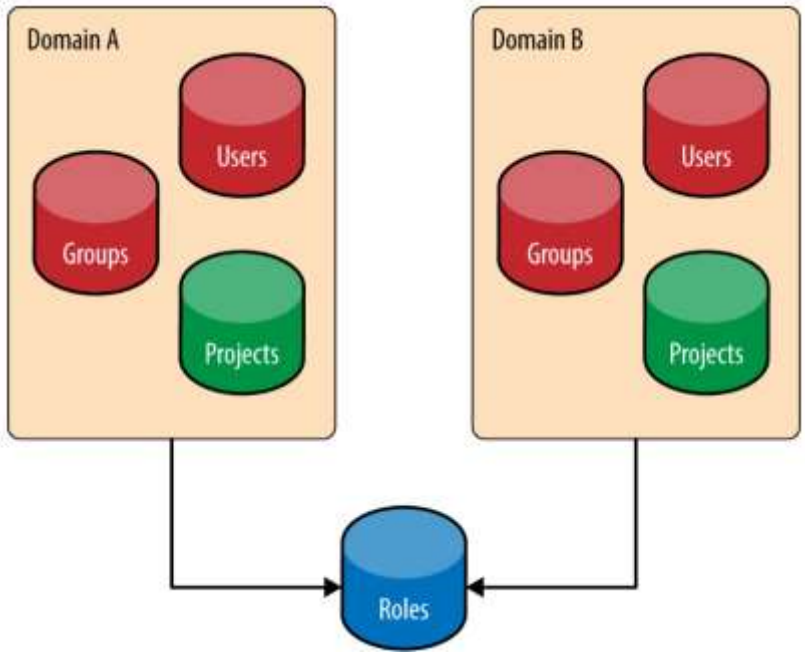
۱-۱-۲ پروژه Keystone

به طور کلی هر ابزاری که در حوزه ابری پیاده‌سازی می‌شود در نهایت باید دارای سطوح دسترسی و مجوزهای عبور و سیستم احراز هویت باشد، پروژه Keystone به عنوان OpenStack Identity Service شناخته می‌شود و یک سیستم یکپارچه برای مدیریت کاربران OpenStack ایجاد می‌کند. کارکرد Keystone تقریبا شبیه کاری که اکتیودایرکتوری (LDAP) در سرویس‌های مایکروسافتی انجام می‌دهد می‌باشد با این تفاوت که روش‌های مختلف احراز هویتی در Keystone از جمله Username و Password ، روش استفاده از Token برای لاگین کردن و حتی مکانیزم لاگین آمازونی AWS در Keystone قابل استفاده هستند. تمام کاربرانی که در فضای OpenStack تعریف می‌شوند توسط پروژه Keystone مدیریت می‌شوند.

پروژه Keystone حاوی چندین مفهوم هست که چگونگی ارتباط آن با OpenStack را مشخص می‌کند. در ادامه به بیان برخی از مهمترین مفاهیم استفاده شده در keystone پرداخته شده است:

- **پروژه (Project):** پروژه شامل مفهومی است که در آن منابع مانند سرورها، ایمیج‌ها و ... برای سایرسرویس‌ها، گروه‌بندی یا ایزوله‌سازی می‌شود. در واقع مهمترین هدف سرویس Keystone، ثبت پروژه‌ها و تعیین کسانی است که باید به پروژه‌ها دسترسی داشته باشند. لذا برای دسترسی کاربران یا گروه‌ها به یک پروژه از مفهومی به نام تخصیص Role استفاده می‌شود.

- **نقش (Role):** Roleها در Keystone، برای مجوزدهی مورد استفاده قرار می‌گیرند. یک Actor (کاربر یا گروه) می‌تواند چندین Role برای اهداف مختلفی داشته باشد. به عنوان مثال Role ادمین، می‌تواند برای پروژه Development به کاربر Alireza تخصیص داده شود.
- **دامنه(Domain):** در ابتدا هیچ مکانیزمی به منظور محدود کردن پروژه‌ها برای کاربران سازمان‌های مختلف نبود؛ که این امر سبب اختلال در نامگذاری پروژه‌ها و ایجاد پروژه‌هایی با نام مشابه می‌شد. همچنین همین مسئله برای نام‌های کاربران اتفاق می‌افتاد؛ لذا نیاز شد تا پروژه‌ها و کاربران را درون یک NameSpace تعریف کنند تا قادر به ایزوله کردن آنها باشد؛ اینجا بود که مفهوم Domain به وجود آمد. در واقع Domain، امکان دسته‌بندی منابع درون ابر را فراهم می‌آورد؛ به عنوان مثال یک فضای ابری می‌تواند دو Domain داشته باشد که منابع هر Domain کاملا از هم، مجزا هستند.
- **کاربران و گروه‌هایی از کاربران (Actorها):** کاربران و گروه‌ها به عنوان استفاده‌کنندگان نهایی ازفضای ابری محسوب می‌شوند و به آنها Actor گفته می‌شود که می‌توان آنها را با تخصیص مجوز و دادن Role، به Domain یا پروژه اضافه کرد.



شکل ۲-۲ مفاهیم استفاده شده در Keystone

برای درک بهتر، در شکل ۲-۲ رابطه بین Domainها، پروژه‌ها، کاربران و گروه‌ها نشان داده شده است. همان‌طور که مشخص می‌باشد کاربران، گروه‌ها و پروژه‌ها در داخل Domianها محدود می‌شوند، لذا می‌توانند در بین Domainها مشترک باشند.

۲-۱-۲ پروژه Nova

پروژه Nova از OpenStack را می‌توان به عنوان هسته اصلی پردازشی این سکو به حساب آورد. این پروژه وظیفه پشتیبانی از انواع مختلف مجازی‌سازها را برعهده دارد. به عبارت دیگر این Nova می‌باشد که ماشین‌های مجازی KVM لینوکسی ، ماشین‌های مجازی ESXi محصول VMWare ، ماشین‌های مجازی Hyper-V مایکروسافتی ، ساختار ماشین مجازی Container Based ای مثل Docker و

حتی LXC ها را با هم یکپارچه می‌نماید. در حقیقت API های Nova مشخصات ماشین مجازی خواسته‌شده (ازقبیل میزان RAM، CPU، Disk، Network و ..) را دریافت کرده و سپس از طریق تعامل با Hypervisor پس از ساخت آن ماشین، شرایط را برای ارائه به کاربران فراهم می‌نماید. بالاترین میزان سازگاری با Nova را KVM Hypervisor دارا می‌باشد و توصیه خود OpenStack هم استفاده از راهکار مجازی سازی KVM Linux می‌باشد.

به هر Hypervisor ای که در اختیار Nova باشد، اصطلاحا یک **Compute Node** گفته می‌شود. همچنین به هر Virtual Machine ای که توسط Nova کنترل می‌شود اصطلاحا یک **Instance** گفته می‌شود. زمانی‌که Hypervisor ها تحت عنوان Compute Node در اختیار سرویس Nova قرار گرفته باشند، این Nova است که تصمیم می‌گیرد که هر کدام از Instance ها بر روی کدام میزبان قرار بگیرند. به عبارت دیگر این Nova است که ارتباطات بین Hypervisor و Virtual Machine را مدیریت می کند تا (در یک مقیاس انبوه) با دنبال نمودن میزان مصرف منابع بر روی هر میزبان ، درخواست‌هایی نظیر ساخت، تغییر سایز و یا حذف یکInstance صورت پذیرد. عبارت **Flavor** در اصطلاح Nova منظور فایل تنظیمات برای ساخت Instance می‌باشد. هنگام ساخت یک Instance باید تعیین شود که بر اساس کدام Flavor ساخته می‌شود. به عبارت دیگر برای ساخت یک Instance باید Flavor آن را تعیین نمود، چراکه Flavor اندازه آن Instance را تعیین می‌کند. درون یک Flavor می‌توان میزان منابع یک Instance و میزان بهره‌گیری از آن‌ها را تعیین نمود، به عنوان مثال به کمک Flavor می توان تعیین نمود که هر Instance چه تعداد دیسک، و روی هر دیسک چه مقدار فضا داشته و همچنین هر Instance روی آن دیسک چقدر I/O بتواند استفاده کند. چنین مواردی برای تعیین سایر منابع یک Instance از قبیل RAM، CPU، Network و همچنین تعیین سایر شرایط یک Instance از قبیل میزان swap، Bandwidth، Secure Boot و ... نیز وجود دارد.

۲-۱-۳ پروژه Neutron

پیکربندی شبکه مجازی یا همان Virtual Networking در محیط OpenStack بر عهده پروژه Neutron می‌باشد. این سرویس قابلیت per-project networking را در اختیار قرار می‌دهد به این معنی که شبکه‌های مجازی ایزوله می توانند داخل پروژه‌های مختلف ساخته و ایجاد شوند و این امکان باعث ایزوله شدن ماشین‌های مجازی می‌شود.

هر ماشین مجازی در OpenStack برای ارتباط با دنیای بیرون باید دارای آدرس IP باشد. دو نوع از آدرس‌های IP برای ماشین های مجازی وجود دارند: fixed IPs و floating IPs. آدرس های fixed IPs در زمان بوت ماشین مجازی تخصیص داده می‌شوند درحالیکه آدرس های floating IPs می توانند پس از ایجاد ماشین های مجازی به آنها تخصیص داده شوند و بین چندین ماشین مجازی جابه‌جا شوند. آدرس‌های fixed IPs ضروری هستند ولی ماشین های مجازی بدون آدرس‌های floating IPs هم می توانند اجرا شوند. یکی از سناریوهای متداول استفاده از آدرس‌های floating IPs مهیا کردن آدرس هایIP عمومی برای یک ابر خصوصی با در اختیار داشتن تعداد محدودی از آدرس های IP است.

۲-۱-۴ پروژه Cinder

پروژه Cinder قابلیت Block Storage را برای استفاده در Nova فراهم می‌کند. این سرویس وظیفه ذخیره‌سازی بلاکی ، ایجاد، اتصال و لغو اتصال بلاک ذخیره‌سازی به ماشین مجازی را برعهده دارد. در

سکوی OpenStack طیف وسیعی از سرویس دهندگان ذخیره سازی شناخته شده در سطح Enterprise پشتیبانی می‌شود.

جدول ۱ برخی از درایورهای پشتیبانی شده در Cinder

Ceph RADOS Block Device (RBD)	LVM
NFS driver	DataCore SANsymphony volume driver
Dell VNX driver	Dell Unity driver
Fujitsu ETERNUS DX driver	Hitachi block storage driver
HPE MSA Fibre Channel and iSCSI drivers	HPE 3PAR, HPE Primera and HPE Alletra 9k
Huawei volume driver	IBM FlashSystem 840/900 driver
Lenovo Fibre Channel and iSCSI drivers	NetApp unified driver
NEC Storage V series driver	NEC Storage M series driver
TOYOU NetStor Cinder driver	Veritas ACCESS iSCSI driver
VMware VMDK driver	Virtuozzo Storage driver
Windows iSCSI volume driver	Windows SMB volume driver
YADRO Cinder Driver	Zadara Storage VPSA volume driver

جدول ۱ لیست برخی از درایورهای پشتیبانی شده توسط پروژه Cinder را نمایش می‌دهد. یکی از معروف ترین و محبوب‌ترین دریوارهای مورد استفاده در Cinder سکوی متن باز Ceph می‌باشد که یکپارچه سازی بسیار بالایی با سکوی OpenStack دارد.

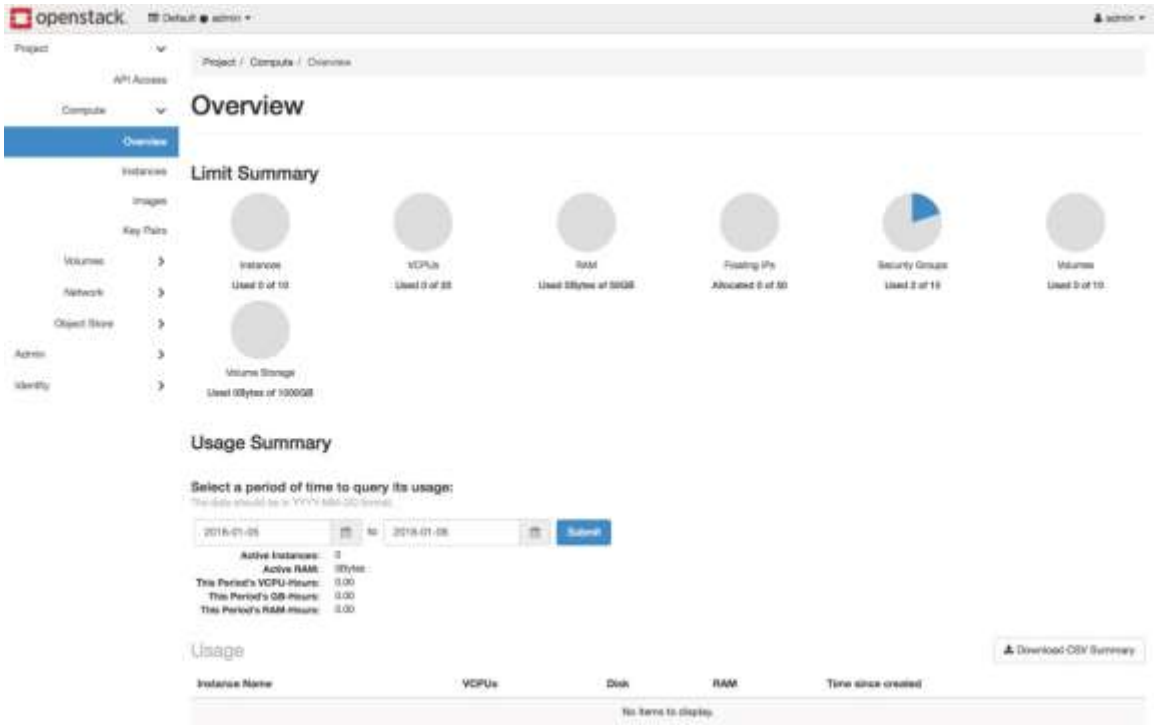
سکوی Ceph به عنوان یکی از معروف‌ترین ابزارها در حوزه Software-Defined Storage (SDS) شناخته می‌گردد و قابلیت ارایه هر سه مدل سرویس Object Storage، Block Storage و File Storage را دارا می‌باشد.

۲-۱-۵ پروژه Glance

کاربرد پروژه Glance در OpenStack مدیریت و انجام کارهای مرتبط با Imageهای دیسک و Imageهای سیستم عامل (سرورها) در فضای ابری می‌باشد. این مولفه وظیفه شناسایی ، ثبت کردن و ارائه سرویس‌های مربوط به Imageها را بر عهده دارد. عملیات مربوط به ذخیره‌سازی اطلاعات کاربران ارتباطی به این مولفه ندارد. ایجاد کردن قالب‌های آماده از سیستم عامل‌ها و گرفتن Snapshotها و در نهایت ایجاد کاتالوگی آماده از سرورها برای انتخاب کردن و فعال سازی در فضای ابری با استفاده از Glance انجام می‌گردد.

۲-۱-۶ پروژه Horizon

پروژه Horizon رابط کاربری تحت وبی است که از طریق آن اکثر پروژه‌های OpenStack قابل مشاهده و پیکربندی می‌باشند. این داشبورد یکی از چندین راه برقراری ارتباط با OpenStack می‌باشد. غیر از این رابط کاربری راهبر سامانه نیز می‌تواند از طریق رابط تحت خط فرمان یا CLI نیز اقدام به مدیریت و پیکربندی OpenStack نماید. شکل ۲-۳ نمایی از صفحه داشبور Horizon را نمایش می‌دهد.

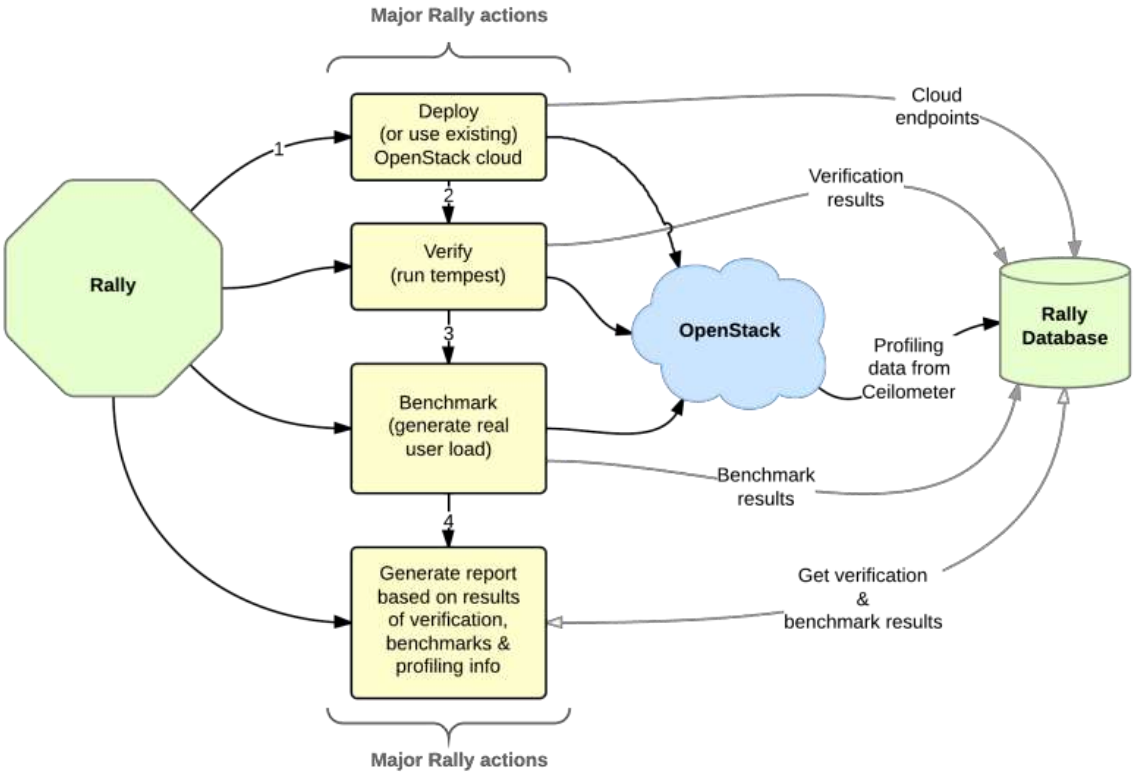


شکل ۳-۲ رابط کاربری تحت وب Horizon

۲-۱-۷ پروژه Rally

Rally یک پروژه Benchmark-as-a-Service برای سکوی OpenStack می‌باشد. این پروژه در نظر گرفته شده است تا ابزاری برای محک زدن OpenStack ارایه دهد که قادر به انجام موارد آزمایشی خاص، پیچیده و تکرار پذیر در سناریوهای استقرار واقعی باشد.

یکی دیگر از کاربردهای این پروژه فراهم آوردن یک سیستم OpenStack CI/CD می‌باشد که این امکان را مهیا می‌سازد تا SLA، عملکرد و پایداری OpenStack را به طور مداوم بهبود دهد. تیم OpenStack QA بیشتر روی CI/CD کار می‌کند که تضمین می‌کند، بروزرسانی‌ها وصله‌های نرم‌افزاری جدید عملکرد و کارایی OpenStack را خراب نمی‌کنند. شکل ۲-۴ چرخه کاری Rally را نمایش می‌دهد که شامل استقرار OpenStack (یا استفاده از یک OpenStack از قبل استقرار یافته)، اجرای آزمون‌ها، بنچمارک گرفتن و گزارش دهی می‌باشد.

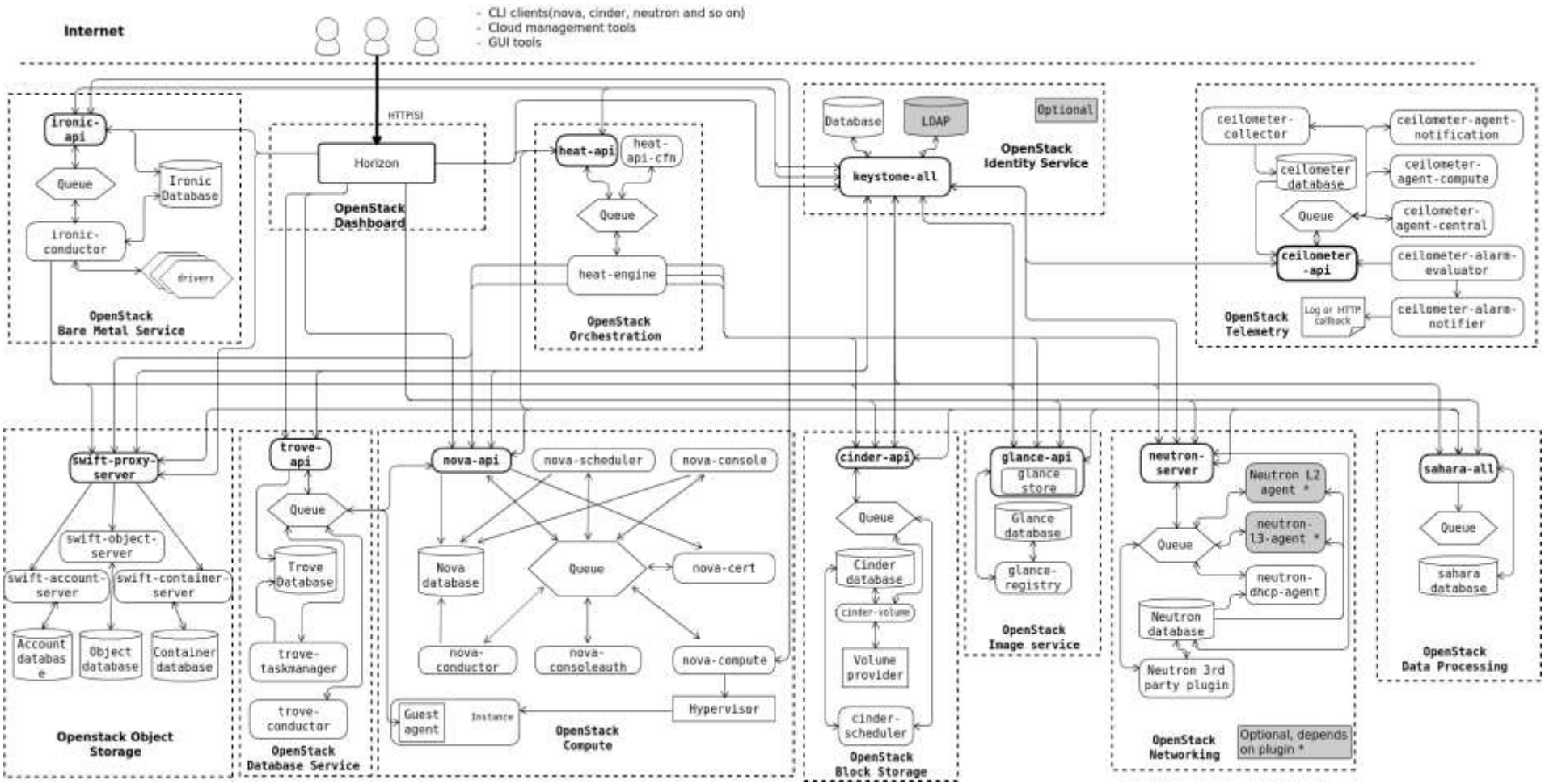


شکل ٤-٢چرخه کاری Rally

۲-۲ ارتباط بین پروژه‌های OpenStack

شـــکل ۲-۵ نحوه ارتباط منطقی بین پروژه‌های مختلف OpenStack را نمایش می‌دهد. همانطور که مشخص است هر پروژه OpenStack خودش از تعدادی مولفه تشکیل می‌شود که از طریق RestAPI و یا Message Queue با یکدیگر ارتباط دارند.

روال ساخت یک ماشین مجازی به صورت خلاصه بدین صورت است که ابتدا درخواست کاربر توسط Keystone احراز هویت می‌گردد و سطح دسترسی آن نیز بررسی می‌شود در صورت موفقیت آمیز بودن این مرحله یک Token برای مدت زمان مشخصی به کاربر اختصاص می‌یابد و سپس درخواست به سمت Nova ارسال می‌گردد تا ماشین مجازی (بخش محاسباتی) براساس مشخصات درخواست شده ایجاد گردد. پس از آن Cinder براســاس درایوری که برای آن مشـخص شـده اسـت اقدام به ایجاد بلاک ذخیره سازی می‌نماید در انتها نیز Neutron برای ماشین مجازی آدرس IP رزرو می‌نماید.



شکل ۲-۵ نحوه ارتباط داخلی پروژه‌های تشکیل دهنده سکوی OpenStack

۳ نحوه پیاده‌سازی

در این فصل درباره نحوه پیاده سازی سکوی OpenStack ، ابزارها و معماری‌های مختلف آن پرداخته شده است.

۳-۱ روش‌های پیاده‌سازی

برای پیاده‌سازی زیرساخت OpenStack روش‌ها و راه‌های گوناگونی وجود دارد. جدول ۲ انواع روش‌های استقرار و نصب خودکار OpenStack را نمایش می‌دهد. علاوه بر موارد ذکر شده امکان نصب OpenStack به صورت دستی و مرحله به مرحله نیز وجود دارد که برای محیطهای عملیاتی توصیه نمی‌گردد و صرفا جنبه یادگیری و آزمون دارد.

جدول ۲ روش های استقرار OpenStack

نام روش	توضیحات
Openstack-Helm	استفاده از Helm
Kolla-Ansible	استفاده از Docker Image و Ansible
Kayobe	استفاده از Docker Image و Ansible
Openstack-Ansible	استفاده از LXC و Ansible
Openstack-Charm	استفاده از Charm و JuJu
BIFROST	استفاده از مولفه Ironic
Openstack-chef	استفاده از Chef

از بین روش‌های نصب و استقرار OpenStack، روش نصب Kolla-Ansible وOpenstack-Ansible روش‌هایی هستند که بیشتر مورد استفاده قرار می‌گیرند. جدول ۳ مقایسه بین روش‌های نصب OpenStack را بر مبنای آخرین نسخه آن‌ها (در زمان نگارش سند) نمایش می‌دهد. همانطور که مشخص است روش Kolla-ansible بیشترین تعداد اجزای نصب OpenStack را شامل می‌گردد از این رو انتخاب مناسبی می‌باشد.

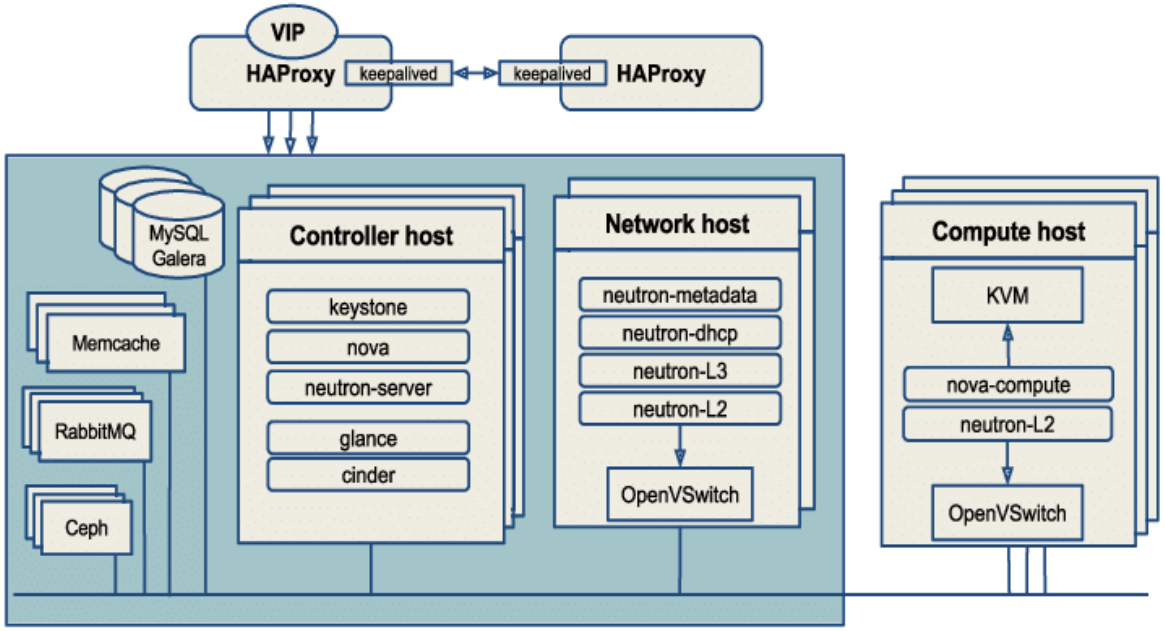
جدول ۳ مقایسه ابزارهای استقرار OpenStack

نام روش	تعداد اجزای OpenStack که توسط ابزار استقرار پشتیبانی می شوند	ویژگی های خاص ابزارهای استقرار	قبل از اینکه ابزار استقرار کنترل شود، چه اتفاقی باید بیفتد.	ابزار استقرار از چه فناوری های پایه ای استفاده می کند.	هنگام استفاده از این ابزار استقرار، مسیر اصلی ارتقاء پشتیبانی شده چیست؟
Openstack-Helm	۲۴	all-in-one encrypted-local-comms, nova-cells, offline-installation	kubernetes-cluster	helm, kubernetes, oci-containers, sles-opensuse, ubuntu	online , per-version
Kolla-Ansible	۳۹	all-in-one encrypted-local-comms, nova-cells, offline-installation	os-installed	ansible, deb-packages, debian, git, oci-containers, redhat-centos, rpm-packages, source-tarballs, ubuntu	online , per-version
Kayobe	۳۹	all-in-one encrypted-local-comms, nova-cells, offline-installation	bare-metal os-installed	ansible, git, oci-containers, redhat-centos, rpm-packages, source-tarballs	online , per-version
Openstack-Ansible	۳۰	all-in-one, supports-heterogeneous-versions	os-installed	ansible, deb-packages, debian, git, redhat-centos, rpm-packages, sles-opensuse, source-tarballs, ubuntu	online , per-version
Openstack-Charm	۲۱	nova-cells	bare-metal	deb-packages, juju, ubuntu	online , per-version
BIFROST	-	-	-	-	-
Openstack-chef	۱۱	all-in-one	env-bootstrap	chef, deb-packages, redhat-centos, rpm-packages, ubuntu	-

۳-۲ چیدمان و نحوه استقرار

نحوه چیدمان و استقرار سکو OpenStack می‌تواند هم به صورت HA و هم به صورت non-HA و یا ترکیبی از این دو حالت صورت پذیرد. در ساده‌ترین حالت ممکن این امکان وجود دارد که تمامی سرویس‌های کنترلی در OpenStack به صورت All-In-One و در یک سرور مستقر شوند این شیوه کمترین میزان منابع و کمترین پیچیدگی ممکن را خواهد داشت اما هیچ گونه افزونگی را به‌همراه نخواهد داشت و با از بین رفتن سرور تمامی زیرساخت مختل می‌شود.

رویکرد دیگری که می‌توان در نظر داشت این است که برخی از سرویس‌های مهم صرفا به صورت HA و برخی دیگر به صورت non-HA استقرار یابند. این روش باعث می‌شود که منابع بیشتری نسبت به حالت All-In-One و منابع کمتری نسبت به زمانی‌که تمامی سرویس‌ها به صورت HA مستقر شوند مصرف گردد. این راهکار برای زمانی‌که محدودیت در میزان منابع وجود دارد می‌تواند مورد استفاده قرار بگیرد و اینکه کدام سرویس‌ها به صورت HA مستقر شوند نیز به صورت انتخابی خواهد بود. مثلا می‌توان مشخص کرد که پایگاه داده به صورت HA مستقر شود اما سایر سرویس‌ها به صورت HA نباشند.

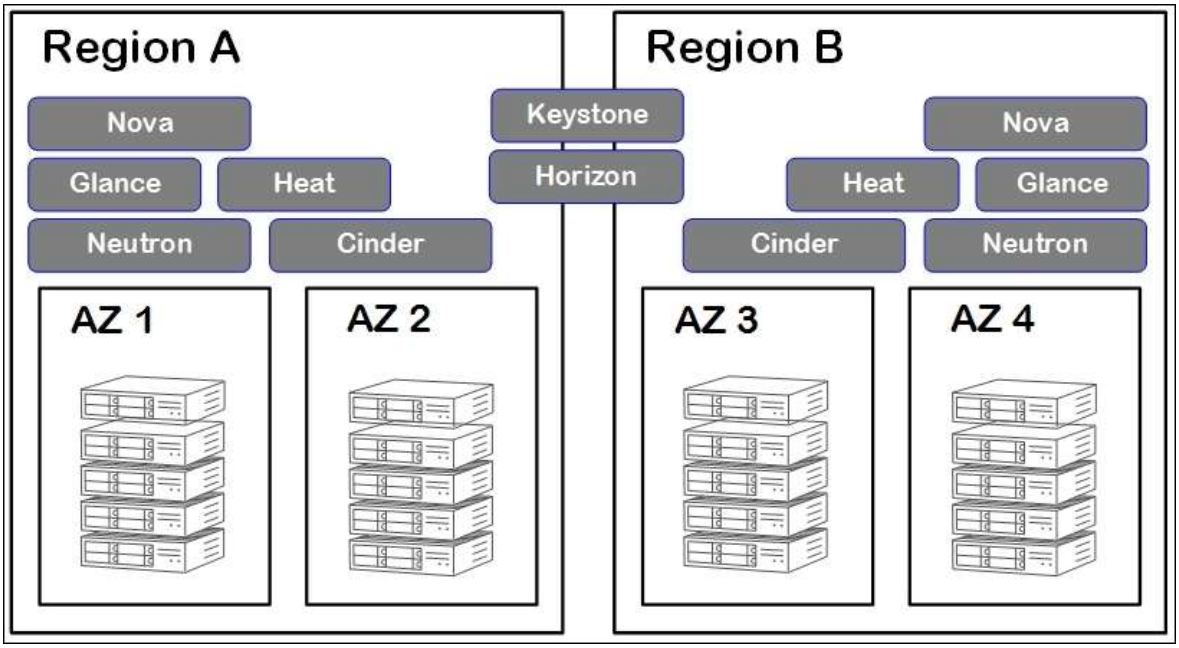


شکل ۳-۱ نحوه چیدمان سرویس‌هایOpenStack به صورت HA

کامل‌ترین حالت ممکن این است که تمامی سرویس‌ها به صورت HA مستقر شوند و علاوه بر اینکه هر سرویس در چندین سرور مستقر شود خود سرورها نیز در Rack های مختلف قرار گرفته باشند که از بین رفتن یک Rack باعث قطع شدن و اختلال در خدمات نشود.

شکل ۳-۱ نحوه چیدمان سرویس‌های OpenStack را در حالت HA نمایش می‌دهد. همانطور که مشخص است برای استقرار سرویس‌های کنترلی از سه سرور مختلف استفاده شده است و برای سرویس‌های بخش شبکه نیز از دو سرور استفاده گردیده است. استقرار پایگاه داده نیز به صورت Galera صورت گرفته است. ارتباط سرویس‌ها با یکدیگر از طریق HAProxy صورت می‌پذیرد که خود این سرویس نیز توسط سرویس Keepalived به صورت HA مورد استفاده قرار می‌گیرد.

علاوه بر موضـــوع HA اســـتقرار OpenStack می‌تواند به صـــورت Single-Region و یا Multi-Region صـورت پذیرد. زمانیکه اولین خوشـه OpenStack نصـب و مسـتقر می‌گردد، آن خوشـه به عنوان Region اصـلی در نظر گرفته می‌شـود و تمامی سـرویس‌های مورد نیاز و ضـروری OpenStack نیز بالا می‌آیند.



شکل ۲-۳ استقرار Multi-Region سکوی OpenStack

شـکل ۲-۳ نحوه چیدمان سـرویس‌های OpenStack را به صـورت Multi-Region نمایش می‌دهد. در صـــورتیکه خوشـــه دومی به عنوان Region دوم مســـتقر شـــود، در این حالت دیگر Keystone و همچنین Horizon به صـورت جداگانه مستقر نمی‌گردند و از همان سـرویس‌های خوشـه اول استفاده خواهد شـد اما سـایر سـرویس‌ها در خوشـه دوم به صـورت مسـتقل از خوشـه اول نصـب و مسـتقر خواهند شـد و تنها Keystone و Horizon به صـــورت مشـــترک مســـتقر می‌گردند. علت این امر بدین خاطر اســـت که احراز هویت کاربران و مدیریت آن‌ها به صـورت متمرکز و توسط یک سـرویس واحد صـورت خواهد پذیرفت و به ازای هر Region یک سرویس احراز هویت مستقر نمی‌گردد.

۴ نگه‌داری و نظارت

در این فصل به نحوه و چگونگی نگه‌داری و نظارت بر سامانه پرداخته می‌شود.

۴-۱ مدیریت لاگ ها

مدیریت لاگ به‌معنای مجموعه فعالیت‌ها و فرایندهایی اطلاق می‌گردد که برای تولید، جمع‌آوری، پردازش، ذخیره، آرشیو و دور انداختن لاگ‌ها انجام می‌شود. ابزارهای مدیریت لاگ برای مدیریت تمامی این لاگ‌ها که توسط محیط سیستم، شبکه، نرم‌افزار و کاربر تولید شده استفاده می‌شود.

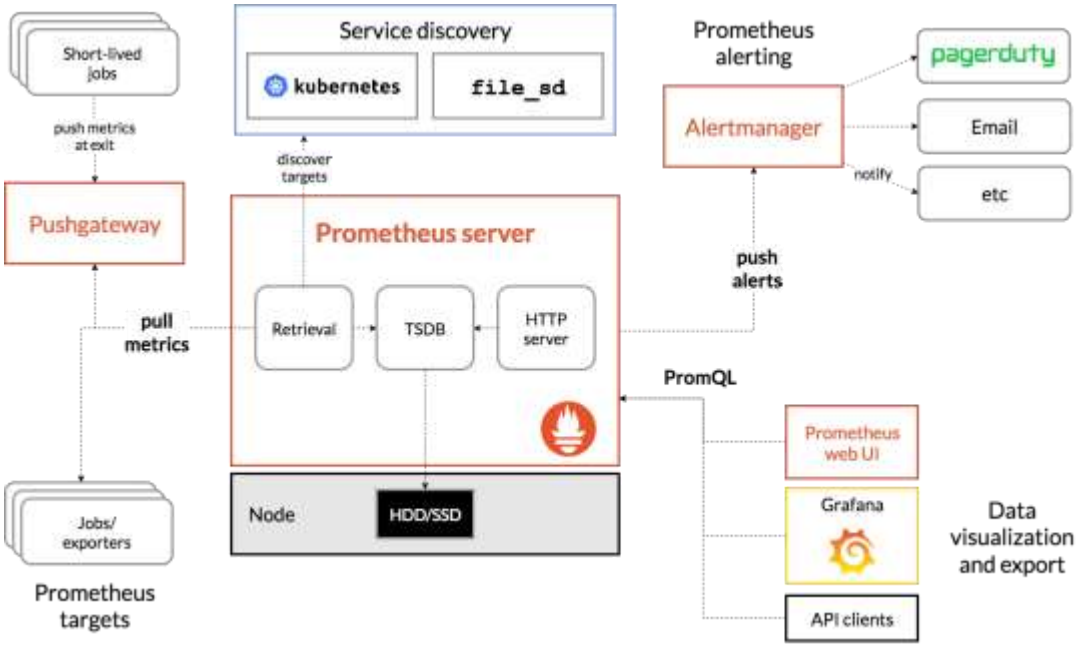
یکی از پراستفاده‌ترین و معروف‌ترین ابزارها در زمینه مدیریت، ابزار ELK می‌باشد که مخفف سه واژه Elasticsearch و Logstash و Kibana است. این استک توسط شرکت Elastic توسعه و انتشار می‌گردد از حدود سال ۲۰۲۱ شرکت Elastic مدل مجوزدهی خود را تغییر داد و بخش‌هایی از این ابزارها را به صورت تجاری درآورد این موضوع باعث گردید که شرکت Amazon یک Fork از آن بگیرد و نسخه متن باز خودش به نام Opensearch را معرفی نماید. امروزه در بسیاری از پروژه‌ها از ابزار Opensearch به عنوان جایگزین Elasticsearch استفاده می‌گردد.

۴-۲ نظارت و هشداردهی

ابزارهای نظارت و هشداردهی، میزان منابع مصرفی سرورها، عملیات و فعالیت‌های کاربران، برنامه‌های کاربردی و سرویس‌های شبکه را مشاهده، پایش و ردیابی می‌کنند. این نرم‌افزارها همچنین امکاناتی برای نمودارسازی و بصری‌سازی داده‌ها نیز به همراه دارند.

ابزار Prometheus یکی از انواع ابزارهای نظارت سرور است که به صورت متن باز بوده و داده‌ها را بر اساس خط زمانی (Time series) آن‌ها جمع آوری می‌کند. این ابزار در ابتدا توسط شرکت Soundcloud توسعه داده شد اما اکنون توسط بنیاد CNCF پشتیبانی می‌شود.

شکل ۴-۱ معماری ابزار Prometheus را نمایش می‌دهد. این ابزار در واقع یک پایگاه داده خط زمانی یا TSDB می‌باشد که متریک‌ها و شاخص‌های گوناگون از قبیل میزان مصرف CPU، Ram و... توسط برخی برنامه‌های جانبی که به آن‌ها exporter گفته می‌شود توسط Prometheus خوانده می‌شود. برای بصری سازی و ترسیم نمودار براساس داده‌ها و اطلاعات موجود در پایگاه داده Prometheus از ابزار Grafana استفاده می‌گردد.



شکل ۱-٤ معماری ابزار Prometheus

۴-۳ امنیت

امنیت اطلاعات یکی از جنبه‌های حیاتی در ارایه خدمات زیرساختی مانند OpenStack است. امنیت اطلاعات در این زمینه از اهمیت بسزایی برخوردار است زیرا حفاظت از اطلاعات حساس و امنیت شبکه‌ها و سرورها از عواقب ناخوشایندی مانند نفوذ و دسترسی غیرمجاز جلوگیری می‌کند. این موضوع نشان از اهمیت اجرای استراتژی‌ها و فرآیندهای مدیریت امنیت اطلاعات در سازمان‌ها دارد.

به منظور بالا بردن سطح امنیت در ارایه خدمات زیرساختی میتوان برخی نکات را رعایت نمود:

- تفیکیک ترافیک کاربران با ترافیک مدیریت سرورها (مانند جداسازی VLAN)
- ایجاد قوانین دیواره آتش در تمامی سرورهای و محدود کردن دسترسی فقط برای آدرسIPهای مجاز
- الزام به استفاده از فرآیند احراز هویت مبتنی بر کلید عمومی و خصوصی در هنگام SSH زدن به سرورها
- اعمال پچ‌های امنیتی، بروزرسانی‌های دوره‌ای و ایمن کردن سرور نسبت به رخنه‌های امنیتی
- بررسی امنیتی ایمیج های مورد استفاده در زیرساخت

۴-۴ پشتیبان‌گیری و بازیابی

علی رغم اینکه طراحی با در نظر داشتن معماری HA می‌باشد اما همچنان تهیه نسخه پشتیبان از سرورها و پایگاه داده‌ها امری ضروری بحساب می‌آید. از اینرو توسط برخی اسکریپت‌ها به صورت دوره‌ای (روزانه، هفتگی، ماهانه) نسخه‌های پشتیبانی از منابع مهم گرفته می‌شود و بر روی سرورهای دیگری ذخیره می‌گردند. همچنین به صورت تصادفی برخی از این نسخه‌های پشتیبان بازیابی میگردد تا از صحت فرآیندهای صورت گرفته اطمینان حاصل شود.

۵ موارد استفاده از OpenStack

با رشد و بلوغ بیشتر سکوی OpenStack موارد استفاده از این سکو روز به روز گسترش می‌یابد و امروزه شاهد آن هستیم که بسیاری از سازمان‌ها خدمات زیرساختی خود را بر روی ابر داخلی خودشان که مبتنی بر OpenStack می‌باشد ایجاد می‌کنند. از آنجایی که این سکو کاملا متن باز می‌باشد می‌توان آن را برای هر نوع کاربردی سفارشی سازی نمود. شکل ۱-۵ نمایی از داشبود سفارشی‌سازی شده Horizon را نمایش می‌دهد که مربوط به مرکز CERN می‌باشد.

شکل ۱-۵ داشبورد Horizon سفارشی سازی شده مرکز تحقیقاتی CERN

برخی از مهم ترین موارد کابرد OpenStack شامل موارد زیر می‌شود:

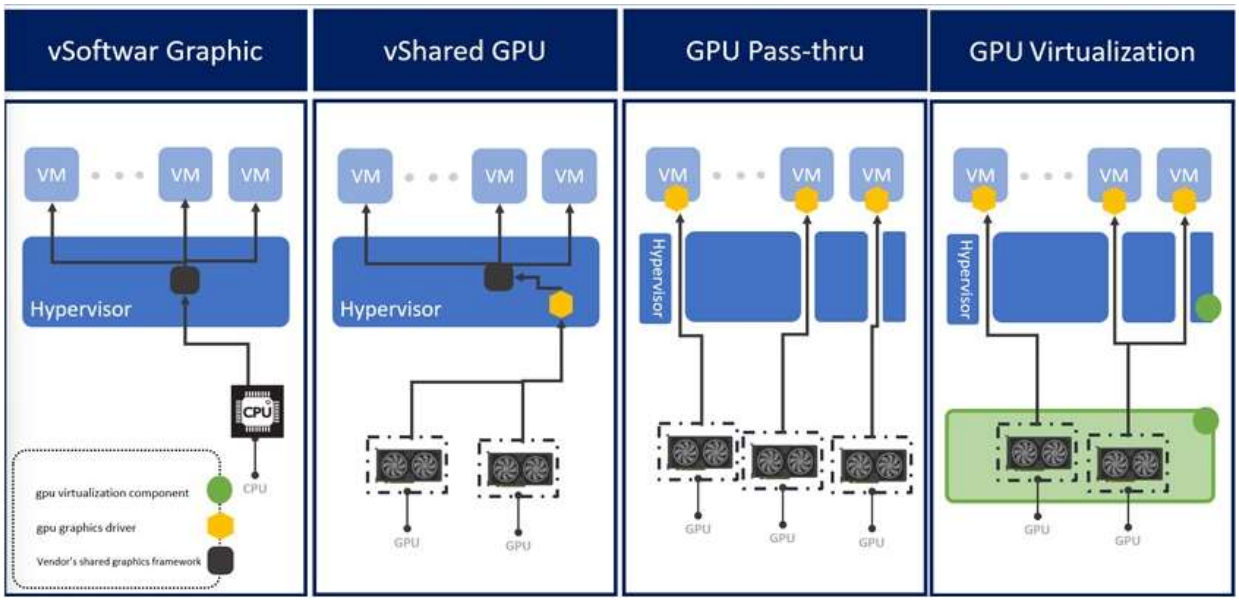
- DevOps Function
- Big Data
- AI and ML
- Storage
- NFV
- IT as a Service

۵-۱ خدمات vGPU

کارت گرافیک مجازی به معنی عینی کلمه (virtual graphics processing unit) به دسته خاصی از کارت‌های گرافیک اشاره دارند که با ایده اشتراک بهینه منابع پردازش گرافیکی برای چند ماشین مجازی تولید شده‌اند. اما در بیان کلی‌تر کارت‌های گرافیک مجازی درواقع همان کارت گرافیک هستند که در ماشین‌های مجازی مورد استفاده قرار می‌گیرند.

به صورت کلی می‌توان به پنج مدل متفاوت از پردازش گرافیکی در یک ماشین استفاده کرد.

۱) Bare metal (مستقیم)
۲) vSoftware Graphic (گرافیک نرم‌افزاری)
۳) vShared GPU (گرافیک اشتراکی)
۴) GPU Pass-thru (تخصیص مستقیم)
۵) GPU Virtualization (کارت گرافیک مجازی)



شکل ۲-۵ انواع کارت گرافیک مجازی

شکل ۲-۵ انواع استفاده از کارت گرافیک مجازی را نمایش می‌دهد. باتوجه به محدودیت‌های تحریم و هزینه خرید لایسنس روشی که در حال حاضر پیشنهاد می‌گردد استفاده از شیوه "**تخصیص مستقیم**" می‌باشد که باعث می‌گردد ماشین مجازی دسترسی کامل به تمام منابع کارت گرافیک داشته باشد.

۵-۲ خدمات مربوطه به توسعه و انتشار نرم‌افزار

توسعه عملیات یا دواپس (DevOps) مجموعه‌ای از روش‌ها و فرایندها و ابزارهایی است که با تمرکز بر ارتباطات و همکاری و یکپارچگی بین تیم‌های توسعه نرم‌افزار و عملیات فناوری اطلاعات، ارزش‌های تولیدشده را به‌طور سریع و مداوم به مشتریان نهایی می‌رساند. هدف، ایجاد فرهنگ و محیطی بوده‌است که در آن بیلدها و تست‌ها و انتشار نرم‌افزار می‌تواند انجام گردد.

در ادامه برخی از ابزارهای مهم و کاربردی که در فرآیند مربوط به توسعه و انتشار نرم‌افزار مورد استفاده قرار می‌گیرند معرفی گردیده است.

۵-۲-۱ خدمات CI/CD

عبارت CI/CD مخفف Continuous Integration Continuous Delivery یا Deployment می‌باشد. همان طور که مشخص است این عبارت از دو بخش تشکیل می‌گردد؛ بخش اول CI مربوط به تست و بیلد برنامه‌ها و بخش دوم CD مربوط به استقرار می‌باشد. یک عبارت مهم که در هردو بخش وجود دارد تاکید بر استمرار زمانی می‌باشد. در این شیوه به کمک برخی از ابزارها می‌توان به طور پیوسته کدها را بیلد، تست و دیپلوی نمود.

ابزاری که عمدتا به عنوان سیستم کنترل نسخه (Version Control System) استفاده می‌گردد، ابزار گیت می‌باشد. در اصل این پیوستگی با استفاده از پوش کردن در یک سرور گیت رخ می‌دهد. با انجام تنظیماتی می‌توان گفت هر زمان به برنچ خاصی پوش شد کد آن بیلد، تست و دیپلوی گردد. در کنار گیت یک سری ابزارهای دیگر نیز باید حضور داشته باشند تا یک pipeline را تشکیل بدهند تا این پیوستگی کامل گردد. ابزار گیتلب (Gitlab) تمامی این فرایند را می‌تواند انجام دهد و می‌توان از آن به عنوان گیت سروری که مجموعه‌ی ابزارهای CI/CD را دارد استفاده کرد.

۵-۲-۲ خدمات مخزن OCI artifacts

در سال ۲۰۱۵ توسط شرکت Docker و برخی از رهبران صنعت کانتینر برخی مشخصات و ساختارهایی پیرامون قالب‌های کانتینر و زمان اجرای آنها ایجاد گردید که به آن Open Containers Initiative یا OCI گفته می‌شود. داکر ایمیج و Heml chart نمونه‌هایی از OCI Artifacts می‌باشند.

امروزه با توجه به کانتینری شدن اکثریت سرویس‌ها نیاز به یک فضای ابری برای ذخیره سازی این نوع از Artifactها احساس می‌شود. ابزار Harbor یک سرویس ذخیره سازی و مدیریت ایمیج داکر (Docker Images) و سایر OCI Artifactها می‌باشد. این سرویس به توسعه دهندگان این اجازه را می‌دهد تا ایمیج داکر خود را ایجاد ، ذخیره و به اشتراک بگذارند. این سرویس، تحت وب کار می‌کند و پروتکل مورد استفاده آن برای مدیریت ایمیج‌ها، پروتکل HTTP و HTTPS است.

۶ سخت‌افزار مورد نیاز

با توجه به معماری ارائه شده برای زیرساخت ابری می‌توان مولفه‌های تشکیل دهنده زیرساخت را به صورت زیر دسته بندی نمود:

۱. **مولفه‌های کنترلی بخش مدیریتی(Controller Node):** این مولفه‌ها شامل سرویس‌هایی از OpenStack می‌شوند که بخش مرکزی سامانه را مدیریت می‌نمایند(مانند سرویس‌هایAPI هریک از مولفه‌ها) و همچنین برخی از سرویس‌هایی که باعث ایجاد ارتباط بین مولفه‌های مختلف می‌گردند(مانند RabbitMQ وMemcached .)
۲. **مولفه‌های کنترلی بخش شبکه(Network Node):** این مولفه‌ها شامل سرویس‌هایی از OpenStack می‌شوند که بخش شبکه (مانند DHCP، Router و ...) سامانه را مدیریت می‌نمایند.
۳. **مولفه‌های کنترلی بخش محاسباتی(Compute Node):** این مولفه‌ها شامل سرویس‌هایی از OpenStack می‌شوند که مدیریت یک گره محاسباتی را برعهده دارند و اصلی‌ترین سرویس این بخش سرویس Nova-Compute می‌باشد.
۴. **مولفه‌های کنترلی بخش ذخیره‌سازی(Storage Node):** تعداد و نوع مولفه‌های این بخش وابستگی به نوع درایور انتخابی برای مولفه Cinder دارد و باتوجه به هر درایور متفاوت می‌باشد.
۵. **مولفه‌ها بخش نظارت و هشدار دهی(Monitoring Node):** مولفه های این بخش شامل برخی از سرویس‌هایی می‌شوند که علاوه بر سرویس های نظارتی خود OpenStack امکانات بیشتری جهت نظارت برزیرساخت را ارایه می‌دهند از جمله این سرویس های میتوان به Prometheus ، Zabbix، Grafana ، Opensearch و ... اشاره نمود.
۶. **پایگاه داده:** پایگاه داده‌ای که تمامی اطلاعات زیرساخت سکوی OpenStack در آن ذخیره می‌گردد، در پیاده‌سازی OpenStack از پایگاه داده MariaDB استفاده می‌گردد.

میزان منابع سخت افزاری برای بخش محاسباتی(Compute Node) و بخش ذخیره‌سازی(Storage Node) وابستگی شدیدی به میزان بارکاری و نوع استفاده دارد و متناسب با نیاز کاربر مشخص می‌گردد. همچنین باتوجه به اینکه این بخش‌ها قابلیت مقیاس پذیری افقی(Horizontal Scaling) را دارا می‌باشند لذا امکان افزودن منابع پس از استقرار و براساس میزان نیاز وجود دارد و نگرانی از این نظر وجود نخواهد داشت. از اینرو میزان منابع سخت افزاری این دو بخش در ادامه ذکر نگردیده است.

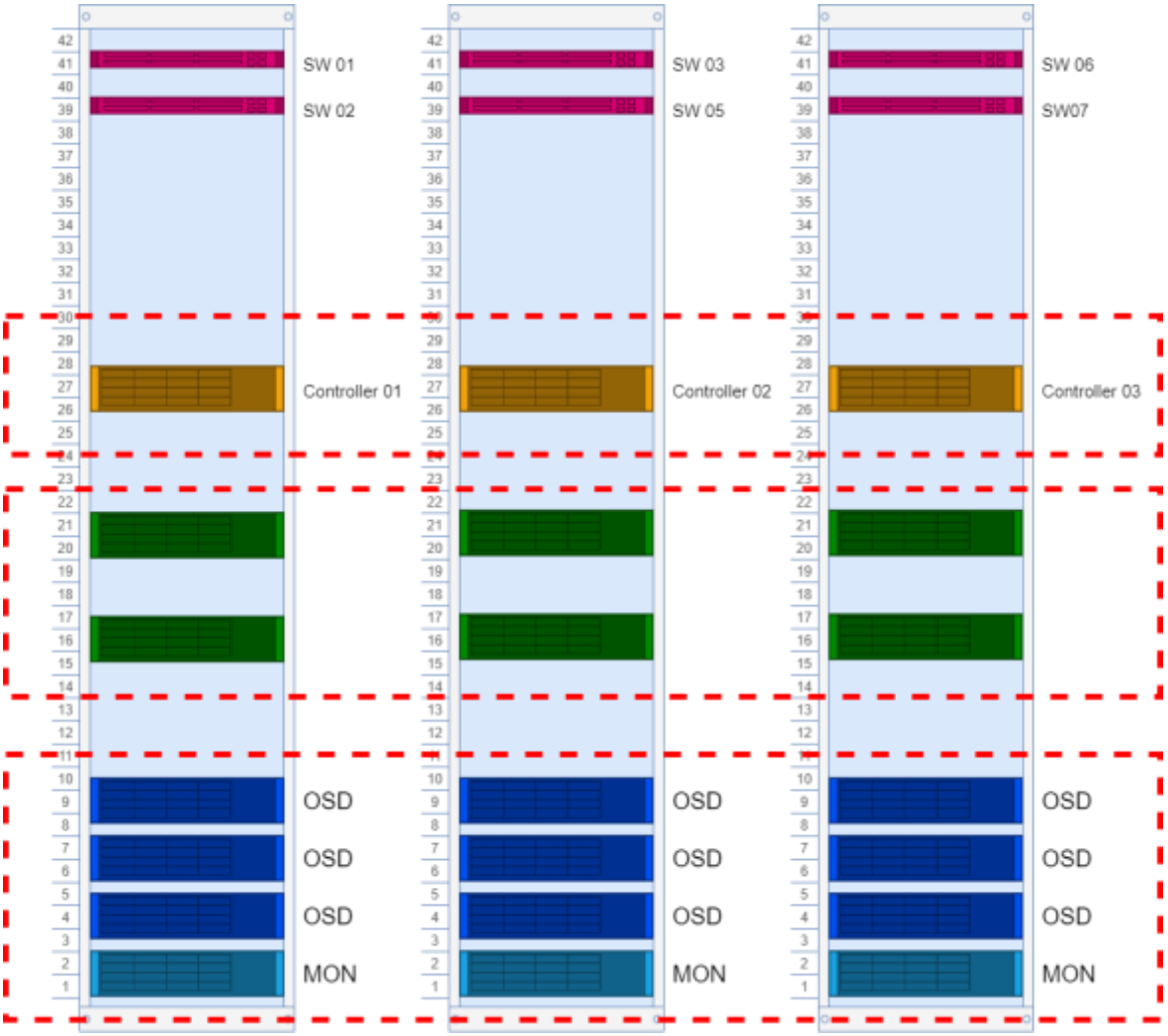
جدول ٤ میزان منابع سخت افزاری

نوع خدمت و مولفه	میزان منابع مصرفی به ازای یک گره				تعداد گره ها با در نظر گرفتن HA	امکان استقرار به صورت ماشین مجازی وجود دارد ؟
	CPU (core)	Ram (GB)	Storage (GB)	Network Adaptor		
Controller	۱۲	۱۶	۱۰۰	۲	۳	بله

Network	۶	۶	۵۰	۲	۲	بله
Compute	براساس نیاز و منابع موجود					خیر
Storage	وابسته به نوع درایور انتخابی					خیر
Monitoring	۶	۶	۱۵۰	۱	۲	بله
DataBase	۸	۸	۱۰۰	۲	۳	بله

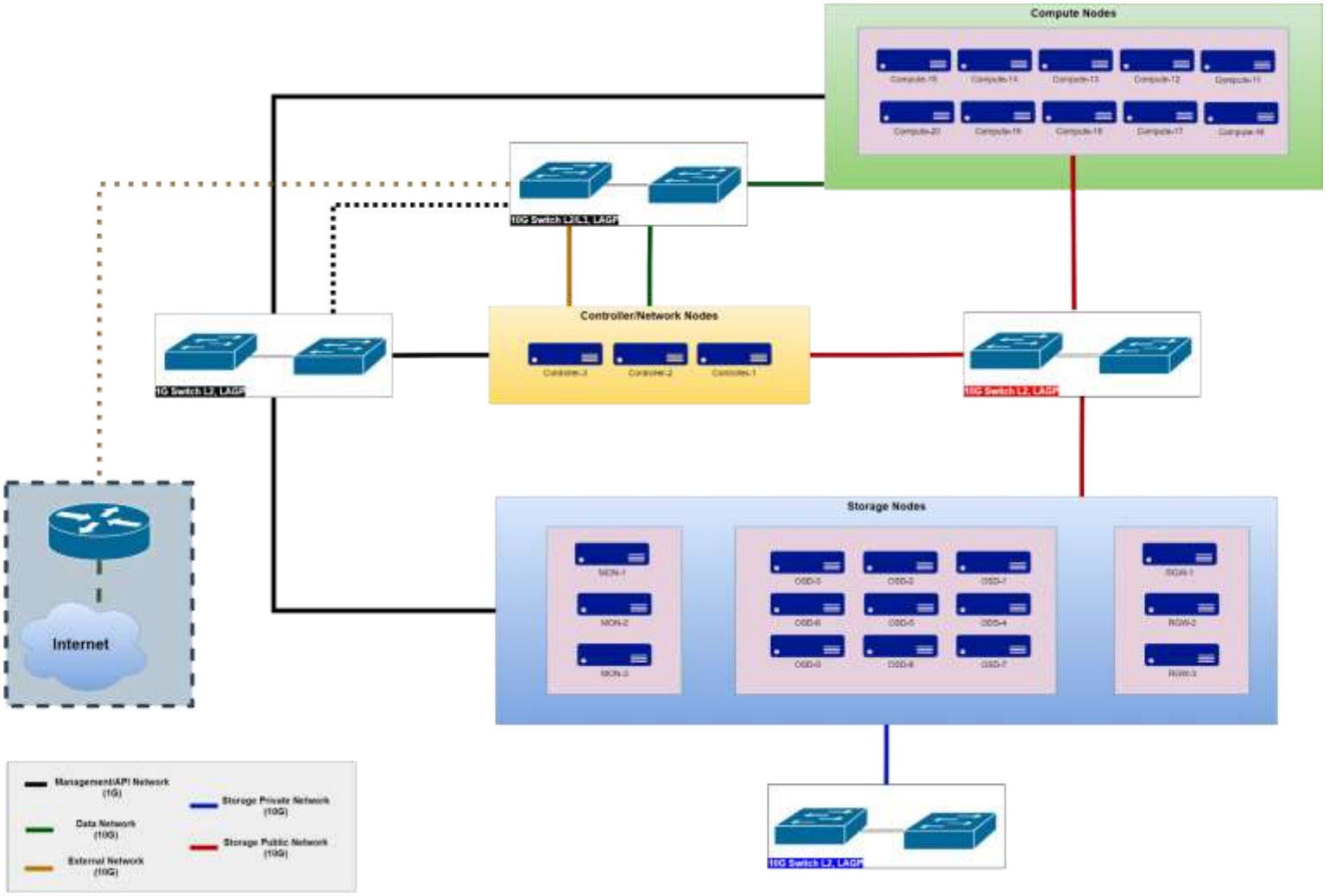
نکته قابل ذکر آن است که برای آنکه در تمامی بخش‌های زیرساخت طراحی به صورت HA در نظرگرفته شود تمامی منابع در تمامی بخش ها باید حداقل به صورت زوج در نظر گرفته شود به عنوان نمونه برای ارتباط سرورهای سخت افزاری به یکدیگر از دو عدد سوییچ شبکه به صورت تجمیع شده استفاده شود تا از بین رفتن یک سوییچ باعث ایجاد اختلال در سرویس‌دهی نگردد.

شکل ۱-۶ نحوه چیدمان سخت افزارها شامل سرورها و سوییچ‌های شبکه را در رک‌ها نمایش می‌دهد. این چیدمان با در نظر گرفتن استفاده از ابزار Ceph به عنوان درایور بخش ذخیره‌سازی می‌باشد. در پایین‌ترین بخش که سرورهای آن با رنگ آبی مشخص شده است سرورهای بخش ذخیره سازی قرار گرفته‌اند. در بخش دیگر سرورهای محاسباتی با رنگ سبز مشخص گردیده‌اند تعداد و مشخصات آنها وابسته به نوع کاربردشان می‌تواند متفاوت در نظر گرفته شود. در بخش دیگر سرورهایی که با رنگ زرد مشخص شده‌اند سرورهای بخش کنترلی می‌باشند و در بالاترین بخش نیز سوییچ‌های شبکه برای بحث ارتباط بین بخش های مختلف در نظر گرفته شده است.



شکل ۱-۶ نحوه چیدمان سرورها در رک با در نظر گرفتن HA

شکل ۲-۶ طرح منطقی و چیدمان سخت افزاری زیرساخت ابری را از منظر نوع کارکرد مولفه‌های مختلف نمایش می‌دهد. در این طرح نیز از ابزار Ceph به عنوان درایور بخش ذخیره سازی استفاده گردیده است، برای بخش ذخیره‌سازی عمدتا از دو نوع شبکه استفاده می‌گردد. یک شبکه اختصاصی (Private) برای عملیات همگام‌سازی و توزیع بار بین سرورهای ذخیره‌سازی که به صورت داخلی بایکدیگر ارتباط دارند و یک شبکه عمومی(Public) که جهت ارایه سرویس ذخیره سازی به بخش محاسباتی مورد استفاده قرار می‌گیرد این جداسازی عمدتا به دلایل پرفرمنسی می‌باشد تا بار ترافیکی بخش ذخیره‌سازی باعث کندی بخش های محاسباتی نگردد.



شکل ۲-۶ طرح منطقی چیدمان سخت افزاری زیرساخت ابری