

دانشگاه صنعتی اصفهان

مرکز ابررایانش

گزارش کارآموزی



مرکز ابر رایانش ملی شیخ بهایی

National High Performance Computing Center



نویسنده: مجتبی ملائی

استاد کارآموزی: دکتر فاطمه دلدار
سرپرست کارآموزی: آقای محمد صالح محمدی

چکیده

من و همکارانم در این دوره کارآموزی ضمن آشنایی با مرکز ابرایانش و افراد متخصص شاغل در این مرکز با استفاده از ابزار های ceph و kubervirt .kubernetes بستری ایجاد کردیم تا بتوانیم ماشین های مجازی ویندوز را با هدف در دسترس بودن کامل^۱ و با قابلیت هایی مانند تخصیص منابع به صورت برخط و امکان خودکار سازی^۲ به صورت ایمن در اختیار مصرف کنندگان قرار دهیم.

فهرست مطالب

۱	معرفی	
۳	پیش‌زمینه من	۱.۱
۳	مرکز ابرایانش شیخ بهائی	۲.۱
۳	منظور از یک خوشه HPC چیست؟	۱.۲.۱
۴	معماری	۲.۲.۱
۴	نحوه کار	۳.۲.۱
۵	پروژه	۳.۱
۲	آماده سازی	
۶	آماده سازی کامپیوترها	۱.۲
۶	آماده سازی سرورها	۲.۲
۶	Proxmox	۱.۲.۲
۳	ارکستراسیون	
۸	Kubernetes معرفی	۱.۳
۸	نحوه کار	۲.۳
۹	کاربرد در پروژه	۳.۳
۹	جزئیات نصب k8s	۴.۳
۴	پیاده‌سازی شبکه	
۱۱	calico	۱.۴
۱۱	multus	۲.۴
۱۱	macvtap	۳.۴
۱۱	bridge	۴.۴
۱۱	kubemacpool	۵.۴
۵	بستر مجازی سازی در k8s	
۱۲	Kubevirt	۱.۵
۱۲	منابع ماشین‌ها	۲.۵
۱۲	انتقال زنده	۳.۵
۱۲	تخصیص منابع به صورت زنده	۴.۵
۶	Windows image	
۱۳	گرفتن اطلاعات	۱.۶
۱۳	تنظیمات اضافی	۲.۶
۷	kubevirt-manager	
۸	ناظارت	
۱۵	Prometheus	۱.۸

۱۵	Graphana	۲.۸
۱۶	آ	اطلاعات اضافی

فصل ۱

معرفی

۱.۱ پیش‌زمینه من

من کارآموزی‌ام را در بعد از ترم ششم آغاز کردم. در این دوران با برخی از گرایش‌ها، زمینه‌ها و مسیرهای مختلف در حوزه کامپیوتر آشنا شدم که از جمله آنها برنامه نویسی سمت سرور^۱، امنیت سایبری، شبکه و هوش مصنوعی بود. در این میان به فناوری‌های سمت سرور مانند Linux علاقه‌مند شدم و این موضوع باعث شد تا تصمیم بگیرم که کارآموزی‌ام را در حوزه‌های مربوط به سرور‌ها بگذرانم.

۲.۱ مرکز ابررایانش شیخ بهائی

شبکه ملی محاسبات ابری در سال ۱۳۸۸ با هدف ارائه خدمات کامل و جامع محاسباتی به صنایع و مراکز علمی و تحقیقاتی کشور و با حمایت معاونت علمی و فناوری ریاست جمهوری افتتاح شد. مرکز ملی محاسبات ابری^۲ (NHPCC) به عنوان یکی از چندین مرکز ابررایانه کشور، در آن زمان بخشی از این طرح ملی را در قالب تأسیس مرکز ملی ابررایانه شیخ بهائی اجرا کرد. تمهیدات در نظر گرفته شده در مرکز ملی محاسبات ابری شیخ بهائی به گونه‌ای است که امکانات سخت‌افزاری و نرم‌افزاری جامع و کامل و همچنین خدمات آموزشی و مشاوره‌ای فنی و تخصصی برای استفاده بهینه (در محل یا از راه دور) در دسترس باشد.

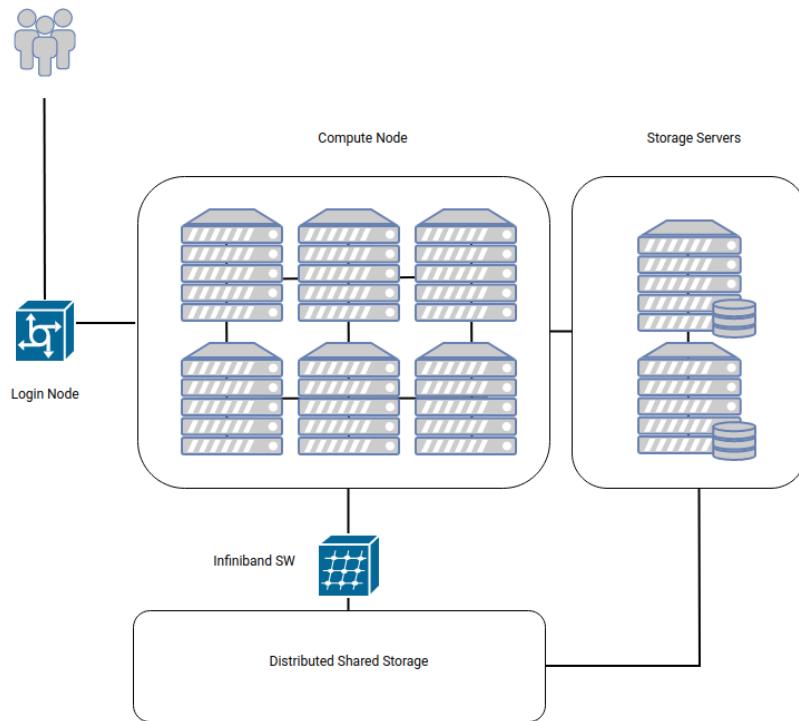
۱.۲.۱ منظور از یک خوش HPC چیست؟

محاسبات با کارایی بالا یا HPC^۳ هنر داشتن مجموعه‌ای از گره‌های پردازشی قدرتمند است که به عنوان یک خوشه (یا گروه‌هایی از آنها) به هم متصل شده‌اند تا مجموعه داده‌های عظیم یا مسائل ریاضی پیچیده چند بعدی را به صورت موازی و در سریع‌ترین زمان ممکن حل کنند. بنابراین HPC در مورد محاسبات است. همچنین ممکن است اصطلاح ابررایانه را بشنوید که مترادف خوشه HPC (یا گروه‌هایی از آنها) است. این نام بیشتر به دیدگاه سخت‌افزاری و قدرت آن اشاره دارد که معمولاً به عنوان یک واحد ممیز شناور به نام فلاپ (مانند ترافلاپس، پتافلاپس و ...) محاسبه می‌شود.

به عنوان مثال، می‌توانید دانش‌آموزی را در نظر بگیرید که یک کد بزرگ (نوشته شده در بیش از هزاران خط) دارد و اگر تمایل داشته باشد کد خود را در لپ‌تاپ شخصی خود اجرا کند، چند سال طول می‌کشد تا پردازش تمام شود و نتیجه نهایی را ببیند؟ بنابراین اگر کد خود را به ابررایانه منتقل کنند، در مدت زمان بسیار کوتاهی به نتایج دست می‌یابد.

¹ back-end
National High Performance Computing Center^۴
High Performance Computing^۵

۲.۲.۱ معماری



شکل ۱.۱: اجزای یک خوشه

ممکن است یک خوشه پایه از این اجزای اصلی تشکیل شده باشد:

- یک گره ورود (گره سر)
- انواع گره‌های محاسباتی
- یک فضای ذخیره‌سازی مشترک توزیع شده
- سرورهای GPU (برای بهره‌مندی از محاسبات گرافیکی عظیم)
- یک سوئیچ Infiniband برای اتصال گره‌ها
- یک زمانبند کار و یک سیستم صف‌بندی
- یک گره انتقال
- یک گره پسپردازش

۳.۲.۱ نحوه کار

کاربران به گره ورود (گره سر) وارد می‌شوند و فایل‌های ورودی خود را آپلود می‌کنند. سپس کارهای خود را به زمانبند ارسال می‌کنند و در صفحه انتظار قرار می‌گیرند. سپس زمانبند شروع به ارسال کارهای آنها به تمام گره‌های محاسباتی دیگری که کاربر مجاز به دسترسی به آنها است، می‌کند.

۳.۱ پروژه

در برخی از موارد مرکز نیاز دارد تا ماشین‌های مجازی ویندوزی را در اختیار مشتریان خود بگذارد. دلیل استفاده از ویندوز راحتی و سادگی بیشتر برای دانشجویان و مشتریانی است که رشته و تخصص آن‌ها در حوزه کامپیوتر نیست و نیاز دارند تا از قدرت محاسباتی مرکز استفاده کنند. برای این افراد کار با سیستم عامل Linux که به طور معمول در خوش‌ها استفاده می‌شود بسیار دشوار است.

نیاز دیگر این بود که اختصاص ماشین‌ها به مشتریان به صورت خودکار و با کمترین مداخله ممکن از طرف یک نیروی انسانی صورت بگیرد. به همین دلیل استفاده از ابزار هایی مانند Proxmox که در این مرکز استفاده می‌شود ممکن نبود. از طرفی در دسترس بودن و سلامتی این ماشین‌ها مهم ترین رکن آن‌ها است. در واقع این پردازش‌ها به دلیل زمان بر بودن آن‌ها از اهمیت خاصی برخوردار هستند. به عنوان مثال در صورتی که سرور میزبان دچار خطأ شود، ماشین مجازی ویندوز نیز از کار می‌افتد و تمام منابع و زمانی که برای پردازش استفاده شده بود به هدر می‌رود. از دید یک مصرف کننده این اتفاق کاملاً غیر قابل تحمل است زیرا که ممکن است مصرف کننده زمان لازم برای این اجرای دوباره این پردازش را نداشته باشد. هدف این پروژه برطرف سازی این مشکلات و نیاز‌ها بود. در کنار این‌ها قابلیت‌هایی مانند تخصیص منابع به صورت زنده، اضافه کردن و حذف دیسک‌ها به صورت زنده و برقرار سازی سطح مناسبی از اهداف مهم دیگر این پروژه بودند.

فصل ۲

آماده سازی

۱.۲ آماده سازی کامپیووترها

در اولین ورود به مرکز، نیاز بود تا کامپیووتر های مورد نیاز خودمان را شخصی سازی کنیم. در واقع میباشد سیستم عامل دیگری بر روی کامپیووترها نصب می کردیم تا اطلاعات قبلی آنها پاک شود و همچنین بتوانیم ابزار های مورد نیاز را روی آنها نصب کنیم. کامپیووترهای مرکز دارای BIOS هستند و همچنین از کارت های گرافیکی قدیمی Nvidia استفاده می کنند. به همین دلیل هنگامی که میخواستیم سیستم عامل های جدید را بر روی آنها نصب کنیم دچار خطأ می شدند. بنابراین نهایتاً مجبور به استفاده از ISO های قدیمی شدیم. پس از نصب سیستم عامل Manjaro بر روی سیستم ها، باید نرم افزار ها و ابزار های لازم مانند مروگر، ابزارهای مدیریت بسته، درایورها، کامپایلرهای ابزارهای نصب، ویراشگرهای کد و مدیریت را نصب می کردیم.

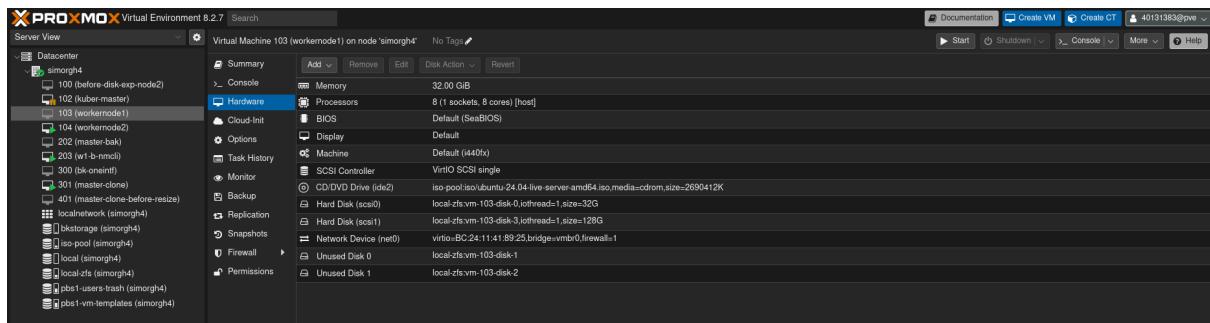
۲.۲ آماده سازی سرورها

پس از آماده سازی کامپیووترها، یک سرور قدرتمند به نام "Rخش" در اختیار ما قرار گرفت. این سرور شامل ۶۴ هسته پردازشی و ۱ ترابایت حافظه RAM می شد. از ما خواسته شد تا با استفاده از ابزاری به نام Proxmox که یک ابزار برای مدیریت مجازی سازی سطح یک است، چند ماشین مجازی لینوکس ایجاد کنیم. این ماشین های مجازی لینوکسی برای ما سرور های واقعی را شبیه سازی می کردند. در واقع در این پروژه در فاز توسعه، به جای استفاده از چند سرور فیزیکی، از یک سرور فیزیکی چند سرور مجازی ساختیم و از آنها برای توسعه پروژه استفاده کردیم. در این سرور های مجازی ما از Ubuntu Server 2022 به دلیل سازگاری، پایداری بهتر و سادگی استفاده کردیم.

۱.۲.۲ Proxmox

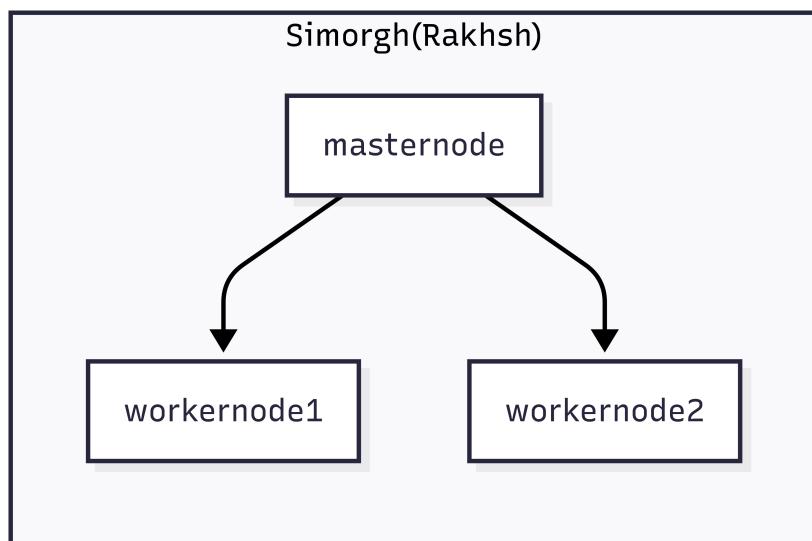
Proxmox VE یک پلتفرم مجازی سازی همه کاره و منبع باز^۱ است که دو فناوری مجازی سازی مبتنی بر کرنل^۲ و مجازی سازی در سطح سیستم عامل^۳ را به صورت یکپارچه در کنار هم ارائه می دهد. این ابزار قدرتمند با ارائه یک رابط مدیریتی تحت وب یکپارچه و کاربرپسند، امکان مدیریت مت默کر ماشین های مجازی و حامل ها،^۴ پیکربندی شبکه های پیچیده، تنظیم سیستم های ذخیره سازی متعدد (مانند ZFS، Ceph) و پیاده سازی قابلیت های خوشبندی و جایه جایی زنده^۵ را فراهم می کند تا راه حلی مقیاس پذیر، باثبات و حرفه ای برای دیتابانک های کوچک و بزرگ ایجاد کند.

open-source^۱
KVM^۲
LXC^۳
Containers^۴
Live Migration^۵



شکل ۱.۲: تصویری از صفحه مدیریت سرورها

همانطور که در تصویر بالا دیده می‌شود. از برخی ماشین‌ها back-up گرفته شده است و برخی از آنها متوقف شده‌اند. امکان back-up گیری و تغییر زنده منابع از مهم‌ترین قابلیت‌های این ابزار است. توپولوژی مورد استفاده ما در این پروژه به شکل زیر بود.



شکل ۲.۲: توپولوژی ماشین‌ها

در واقع ماشین‌های مجازی ویندوز بر روی masternode workernode ها ایجاد می‌شوند و ماشین های دیگر را کنترل می‌کند. در بخش ارکستراسیون به صورت مفصل نحوه این ارتباط توضیح داده خواهد شد.

فصل ۳

ارکستراسیون

ارکستراسیون در دنیای فناوری اطلاعات به فرآیند خودکارسازی استقرار، مدیریت، مقیاس‌گذاری و شبکه‌سازی برای حامل‌ها اطلاق می‌شود. در یک معماری مبتنی بر میکروسرویس‌ها، که یک برنامه به دهها یا صدها جزء مستقل تقسیم می‌شود، ارکستراسیون نقشی حیاتی در هماهنگ‌سازی این اجزا، توزیع بار پردازشی، و تضمین در دسترس بودن و مقاومت در برابر خطا^۱ برنامه ایفا می‌کند. این مفهوم، مدیریت چرخه عمر حامل‌ها را از سطح یک میزبان منفرد فراتر برد و آن را در مقیاس یک خوش از سرورها تعریف می‌کند.

۱.۳ Kubernetes معرفی

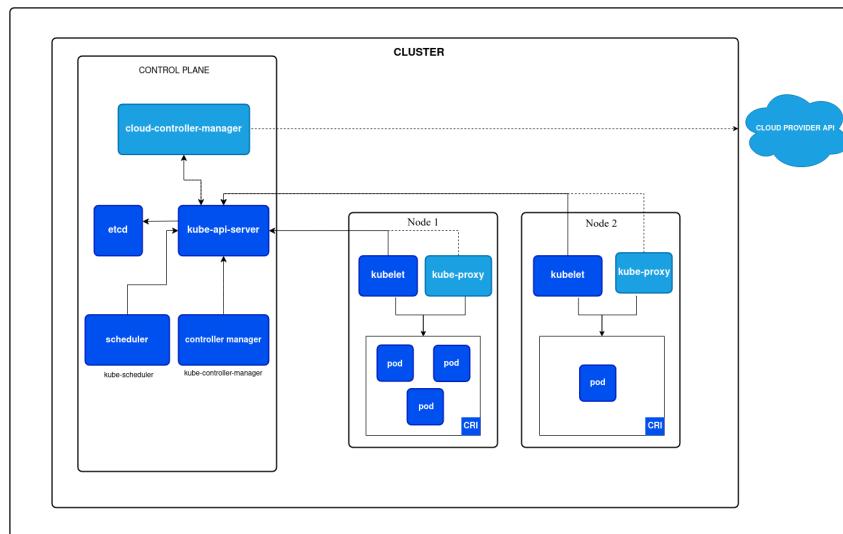
Kubernetes یک پلتفرم ارکستراسیون مبتنی بر حامل^۲ متن‌باز و گستردگی است که به عنوان یکی از اصلی‌ترین راه حل‌ها در این حوزه شناخته می‌شود. این پلتفرم با به کارگیری یک معماری Master-Worker^۳، امکان تعریف حالت مطلوب^۴ برنامه را از طریق فایل‌های اعلانی^۵ فراهم می‌سازد و سپس به صورت خودکار وظایفی مانند زمان‌بندی حاملها بر روی گره‌های کارگر، نظارت بر سلامت آن‌ها، و بازیابی خودکار را انجام می‌دهد. مزایای کلیدی Kubernetes شامل مقیاس‌پذیری الاستیک^۶ و خودکار (هم به صورت افقی و هم عمودی)، بهینگی در مصرف منابع، قابلیت حمل بالا^۷ بین محیط‌های ابری مختلف، و تسهیل فرآیندهای DevOps^۸ و CI/CD^۹ است. در نتیجه، این پلتفرم به عنوان استاندارد اصلی برای اجرا و مدیریت برنامه‌های کاربردی مدرن و مقیاس‌پذیر در محیط‌های هایبرید^{۱۰} و چندابری^{۱۱} تبدیل شده است.

۲.۳ نحوه کار

Kubernetes با دریافت تعریف حالت مطلوب برنامه از کاربر، به صورت خودکار فرآیندهای مدیریت و هماهنگی حامل‌ها را انجام می‌دهد. کاربران مشخص می‌کنند که برنامه‌شان باید چگونه اجرا شود و Kubernetes مسئولیت رساندن سیستم به آن حالت و حفظ آن را بر عهده می‌گیرد. سیستم به طور مداوم وضعیت فعلی را با حالت مطلوب مقایسه می‌کند. اگر بین این دو تفاوتی وجود داشته باشد، Kubernetes به صورت خودکار اقدامات لازم را انجام می‌دهد. به عنوان مثال، اگر یک حامل از کار بیفتاد، سیستم به طور

resilience ^۱
Container Orchestration Platform ^۲
Control Plane and Worker Nodes ^۳
Desired State ^۴
Declarative YAML/JSON ^۵
Elastic Scalability ^۶
High Portability ^۷
Development and Operations ^۸
Continuous Integration and Continuous Delivery/Deployment ^۹
Hybrid Cloud ^{۱۰}
Multi-Cloud ^{۱۱}

خودکار حامل جدیدی را جایگزین آن می‌کند تا برنامه همیشه در حالت مطلوب باقی بماند. Kubernetes حامل‌ها را بین گره‌های مختلف توزیع می‌کند و اطمینان حاصل می‌نماید که از منابع سخت‌افزاری به بهترین شکل ممکن استفاده شود. همچنین شبکه‌های لازم برای ارتباط بین حامل‌ها و تعادل بار بین آنها را به صورت خودکار مدیریت می‌کند تا برنامه به صورت یکپارچه و با کارایی بالا قابل دسترسی باشد.



شکل ۱.۳: معماری کوبرنتیز

ممکن است نام K8s را هم به جای kubernetes شنیده باشید که معنای آن حرف k در ابتداء و ۸ حرف میان آن و حرف s در انتهای است.

۳.۳ کاربرد در پروژه

k8s به دلیل مقاومت در برابر خطأ، بازگشت به حالت مطلوب پس از وقوع خطأ، مقیاس پذیری و انعطاف پذیری خوب و ساختار مازولار^{۱۲} آن یک گزینه عالی برای مدیریت ماشین‌های مجازی ویندوزی و دیگر چیزهای مورد نیاز آنها است. در اصل یکی از نیازهای اصلی پروژه که در دسترس بودن دائمی و مقاومت در برابر خطأ است که به کمک k8s رفع می‌شود. نکته حائز اهمیت این است که k8s به تنهایی تمام کارها را انجام نمی‌دهد. در واقع این ابزار فقط بستر را برای حامل‌ها فراهم می‌سازد. به دلیل امکان تعریف منابع دلخواه و در نتیجه آن، قابلیت توسعه پذیری و بالای این ابزار، افزونه‌های بسیار زیادی برای انجام کارهای مختلف وجود دارد. برخی از این افزونه‌ها مربوط به سیستم داخلی خود k8s است مانند افزونه‌های شبکه که در ادامه معرفی می‌شوند و یا افزونه‌هایی که یک سرویس کامل را اجرا می‌کنند مانند kubevirt که در ادامه پژوهه به آن نیاز داشتیم.

۴.۳ جزئیات نصب k8s

در هنگام نصب k8s نکات زیادی باید رعایت شوند. مثلاً بسته به نوع اینکه گره اصلی^{۱۴} است یا کارگر^{۱۵} دستورات متفاوتی وجود دارد. اما همه گره‌ها پیش‌نیازهای مورد نیاز یکسانی را دارند. مثلاً حداقل منابع RAM و CPU و Disk و بعد از آن یک شبکه که در آن گره‌ها بتوانند بهم متصل شوند، همچنین خاموش بودن Swap و بارگذاری یکسری کرنل مازولهای^{۱۶} و باینری‌های مورد نیاز از جمله این پیش‌نیازها هستند. بعد از آنکه این پیش‌نیازها فراهم شد، با استفاده از دستورات مربوط به پیوستن گره‌ها گره‌های کارگر را به گره اصلی متصل می‌کنیم.

Modular^{۱۲}

plugins^{۱۳}

master^{۱۴}

worker^{۱۵}

Kernel modules^{۱۶}

username	hostname	NIC(ens18)	NIC(ens19)	role	CPU	RAM	primary disk
master	masternode	192.168.170.190	-	control plane	4	10 GB	64 GB
worker1node	worker1node	192.168.170.191	used for br1	worker	8	32 GB	64 GB
worker2node	worker2node	192.168.170.192	used for br1	worker	8	32 GB	64 GB

جدول ۱.۳: مشخصات سرورهای مجازی در ۱۸۰

جدول بالا ویژگی‌ها و مشخصات هر گره را نشان می‌دهد. در فصل بعدی قسمت شبکه هر نود کاملاً توضیح داده خواهد شد.

فصل ۴

پیاده‌سازی شبکه

بحث شبکه یکی از چالشی ترین قسمت‌های این پروژه بود. زیرا ما سه شبکه متفاوت را مدیریت می‌کردیم. شبکه اول، شبکه داخلی و اصلی بین گره‌ها است. این شبکه بر روی رابط ens18 هر گره و با آدرس آیپی $192.168.170.19x/24$ به صورت آیپی ثابت مشخص شده است. این تنظیمات بر روی `netplan` هر گره نوشته شده است و توسط گره‌ها اجرا می‌شود. شبکه دوم شبکه داخلی `k8s` است که بر پایه شبکه اول پیاده‌سازی شده است. مدیریت این شبکه با `calico` است که یک CNI برای مدیریت شبکه داخلی `k8s` است. این شبکه برای ارتباط بین `pod`‌ها استفاده می‌شود. رنج آیپی آن نیز $10.0.0.0/8$ است. این شبکه به دلیل داخلی بودن قابل دسترسی از بیرون `k8s` نیست. شبکه سوم برای ماشین‌های ویندوز در نظر گرفته شده است. این شبکه به صورت عمومی باشد تا کاربران بتوانند به آن وصل شوند. چون دو شبکه دیگر کاملاً داخلی هستند. این شبکه بر روی رابط `ens19` است و رنج آیپی آن نیز $192.168.172.0/24$ است که توسط DHCP سرور آیپی‌ها به ماشین‌های ویندوز اختصاص داده می‌شوند.

calico ۱.۴

multus ۲.۴

macvtap ۳.۴

bridge ۴.۴

kubemacpool ۵.۴

ماشین‌های ما در این پروژه دارای سه شبکه مختلف هستند. سه شبکه متفاوت را مدیریت می‌کردیم. شبکه اول، شبکه داخلی و اصلی بین گره‌ها است. این شبکه بر روی رابط ens18 هر گره و با آدرس آیپی $192.168.170.19x/24$ به صورت آیپی ثابت مشخص شده است. این تنظیمات بر روی `netplan` هر گره نوشته شده است و توسط گره‌ها اجرا می‌شود. شبکه دوم شبکه داخلی `k8s` است که بر پایه شبکه اول پیاده‌سازی شده است. مدیریت این شبکه با `calico` است که یک CNI برای مدیریت شبکه داخلی `k8s` است. این شبکه برای ارتباط بین `pod`‌ها استفاده می‌شود. رنج آیپی آن نیز $10.0.0.0/8$ است. این شبکه به دلیل داخلی بودن قابل دسترسی از بیرون `k8s` نیست. شبکه سوم برای ماشین‌های ویندوز در نظر گرفته شده است. این شبکه به صورت عمومی باشد تا کاربران بتوانند به آن وصل شوند. چون دو شبکه دیگر کاملاً داخلی هستند. این شبکه بر روی رابط `ens19` است و رنج آیپی آن نیز $192.168.172.0/24$ است که توسط DHCP سرور آیپی‌ها به ماشین‌های ویندوز اختصاص داده می‌شوند.

interface^۱
IP Address^۲

^۳ یک منبع تعریف شده در `k8s` که یک یا تعدادی حامل را در خود نگه می‌دارد. کوچک ترین واحد محاسباتی در

فصل ۵

بسترهای مجازی در k8s

Kubevirt 1.0

۲.۵ منابع ماشین‌ها

۳.۵ انتقال زنده

۴.۵ تخصیص منابع به صورت زنده

فصل ٦

Windows image

١.٦ گرفتن اطلاعات

٢.٦ تنظیمات اضافی

این بخش در مورد اینستال کردن ویندوز است. اینستال کردن میتواند برای نصب نرم افزار، نصب سیستم عامل یا برای اینکه یک دستگاه را برای دیگران قابل استفاده کرد. اینستال کردن میتواند برای نصب نرم افزار، نصب سیستم عامل، نصب دستگاه یا برای اینکه یک دستگاه را برای دیگران قابل استفاده کرد. اینستال کردن میتواند برای نصب نرم افزار، نصب سیستم عامل، نصب دستگاه یا برای اینکه یک دستگاه را برای دیگران قابل استفاده کرد. اینستال کردن میتواند برای نصب نرم افزار، نصب سیستم عامل، نصب دستگاه یا برای اینکه یک دستگاه را برای دیگران قابل استفاده کرد. اینستال کردن میتواند برای نصب نرم افزار، نصب سیستم عامل، نصب دستگاه یا برای اینکه یک دستگاه را برای دیگران قابل استفاده کرد. اینستال کردن میتواند برای نصب نرم افزار، نصب سیستم عامل، نصب دستگاه یا برای اینکه یک دستگاه را برای دیگران قابل استفاده کرد. اینستال کردن میتواند برای نصب نرم افزار، نصب سیستم عامل، نصب دستگاه یا برای اینکه یک دستگاه را برای دیگران قابل استفاده کرد. اینستال کردن میتواند برای نصب نرم افزار، نصب سیستم عامل، نصب دستگاه یا برای اینکه یک دستگاه را برای دیگران قابل استفاده کرد.

فصل ٧

kubevirt-manager

فصل ۸

نظارت

Prometheus 1.8

Graphana ۲.۸

پیوست آ

اطلاعات اضافی

متنی است که در اینجا نمایش داده شده است. این متن ممکن است محتوا را در مورد این پروژه ارائه کند. متنی است که در اینجا نمایش داده شده است. این متن ممکن است محتوا را در مورد این پروژه ارائه کند. متنی است که در اینجا نمایش داده شده است. این متن ممکن است محتوا را در مورد این پروژه ارائه کند. متنی است که در اینجا نمایش داده شده است. این متن ممکن است محتوا را در مورد این پروژه ارائه کند. متنی است که در اینجا نمایش داده شده است. این متن ممکن است محتوا را در مورد این پروژه ارائه کند.