«بـــــــه نام خـــالق آرامـــش»

نام کتاب: رایانش ابری (بخش سوم)

نام نویسنده: محمد کاظم اکبری، مرتضی سرگلزایی جوان

تعداد صفحات: ۱۶۶ صفحه

تاریخ انتشار: سال ۱۳۸۹

نسخه از dot net framework خودش می‌تواند باعث مشکلاتی شود و پایداری کل سیستم را کاهش دهد. مجازی‌سازی Dot net framework می‌تواند این مشکلات را به شکل‌های زیر حل کند:

- **عدم نیاز به نصب:** با مجازی‌سازی dot net framework، دیگر نیازی به نصب آن روی یک ماشین به روش قبل نیست. به این معنا که تغییر در محیط سیستم عامل میزبان رخ نمی‌دهد و در زمان نیز صرفه‌جویی می‌شود.
- **برطرف شدن تداخل ها:** بدلیل اینکه dot net framework در ماشین نصب نمی‌شود و برنامه در محیط مجزای خودش اجرا می‌شود، تداخل ها بصورت مجازی رفع می‌شوند. به عبارت دیگر چندین نسخه از dot net framework می‌توانند روی یک ماشین بدون مشکل اجرا شوند.
- **استقرار سریع و آسان:** مجازی‌سازی برنامه کاربردی معمولا شامل بسته‌بندی کل برنامه‌ها در داخل یک فایل قابل اجرا است. این کار نیاز به نصب را از بین می‌برد و می‌توان از روی چهارچوب‌های مختلف چندین کپی بر روی سیستم‌های مختلف تهیه کرد یا اینکه به شیوه ای ساده‌تر همه بسته ها را در یک سرور مرکزی قرار داد و کاربران می‌توانند از این چهارچوب‌ها در thin client های خود استفاده کنند.

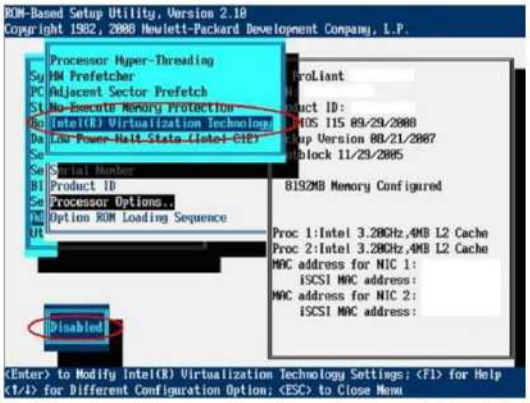
با استفاده از تکنولوژی مجازی‌سازی، dot net framework می‌تواند بسیار موثرتر بکار گرفته شود. تداخل ها حذف شود و برنامه‌های قدیمی و جدید روی یک ماشین یا یک سیستم عامل بدون مشکل کار کنند.

۶-۶- مجازی‌سازی سخت‌افزاری

۶-۶-۱- تکنولوژی مجازی سازی اینتل

تکنولوژی مجازی‌سازی اینتل (Intel VT) راه‌حل سخت‌افزاری برای کمک به مجازی‌سازی است. در گذشته مجازی‌سازی بصورت سنتی کاملا توسط نرم‌افزار انجام می‌شد که فشار زیادی به منابع سخت‌افزاری وارد می‌کرد. امروزه، اکثر پردازنده‌ها دارای تکنولوژی مجازی‌سازی هستند. این ویژگی فشار کمتری به سیستم فیزیکی وارد می‌کند، زیرا سخت‌افزار به ارائه بهتر مجازی‌سازی نرم‌افزاری کمک می‌کند.

تکنولوژی مجازی‌سازی اینتل منابع سخت‌افزاری یک ماشین نظیر پردازنده، chipset و BIOS را به شکل بهینه بکار می‌گیرد تا بخشی از بار کاری مجازی‌سازی نرم‌افزاری به سخت‌افزار منتقل شود. این نوع از تکنولوژی می‌تواند نرخ کارآیی نزدیک به حالت واقعی را دربر داشته باشد، زیرا مجازی‌سازی بطور مستقیم توسط خود سخت‌افزار انجام می‌شود و تنها به نرم‌افزار وابسته نیست.



شکل ۶-۱۱- فعال کردن تکنولوژی مجازی‌سازی سخت‌افزار اینتل در BIOS [۸]

مزایای Intel VT شبیه به مزایای دیگر روش های مجازی‌سازی است. برای مثال، به کمک آن می‌توان چندین ماشین مجازی را روی یک ماشین واحد مستقر کرد. امکان مجزا سازی ماشین‌ها را از همدیگر فراهم می‌کند و .... تنها تفاوت آن این است که Intel VT نسبت به حالت عادی استفاده از مجازی‌سازی نرم‌افزاری بر روی آن، بسیار کارآمدتر است. دیگر نکته مهم در مورد Intel VT این است که بطور یکپارچه با تکنولوژی‌های مجازی‌سازی نرم‌افزاری موجود کار می‌کند و اعمال هیچگونه تغییری در ماشین‌های مجازی لازم نیست و Intel VT بطور اتوماتیک کار می‌کند. این تکنولوژی به شما امکان اجرا سیستم‌های عامل میهمان 64 بیتی را روی بر روی ماشین‌های فیزیکی ۳۲ بیتی می‌دهد و بخاطر تفکیک بهتر ماشین‌های

مجازی، امنیتی بیشتری را فراهم می‌کند. تنها کاری که در خصوص استفاده از این تکنولوژی لازم است انجام دهید، این است که مطمئن شوید ویژگی Intel VT در قسمت BIOS سیستم شما فعال شده باشد.
تکنولوژی مجازی‌سازی اینتل بطور خاص اشاره به راه‌حل مجازی‌سازی سخت‌افزاری[1] آن اشاره دارد که تحت عنوان Intel VT شناخته می‌شود. می‌توان گفت که مجازی‌سازی Hardware-assisted از دیگر اشکال مجازی‌سازی بسیار موثرتر است زیرا پردازنده فیزیکی حجم زیادی از بار کاری هسته سیستم عامل میزبان و نرم‌افزار مجازی‌سازی را بر عهده دارد.
تکنولوژی Intel VT میزبان، شبیه‌سازی نرم‌افزاری را نیز کاهش می‌دهد، زیرا بسیاری از کارها بصورت عادی و بدون نیاز به شبیه‌سازی قابل انجام است. مزایایی که مجازی‌سازی Intel ارائه کرده است به این ترتیب می‌باشد:

- کاهش تعداد ماشین‌های فیزیکی سازمان
- مدیریت ساده‌تر محیط و زیرساخت کسب و کار
- بهبود در دسترس پذیری کسب و کار و قابلیت اطمینان ضمن کاهش downtime
- بهبود بهره‌وری منابع و کاهش هزینه‌های عملیاتی نظیر توان، سرمایش و هزینه‌های مدیریتی
- و ...

با توجه به اینکه Intel VT بسیاری از حجم کارهای نرم‌افزاری را بر عهده دارد، توانایی اجرای انواع مختلف سیستم عامل های مجازی شده با اندازه های بسیار بزرگتر و تنوع بیشتر را خواهد داشت. حتی می‌تواند سیستم عامل هایی را پشتیبانی کند که پیش از این بدون اعمال تغییرات در محیط مجازی بطور موثر کار نمی‌کردند(مشابه روش Para-virtualization). Intel VT بگونه‌ای طراحی شده است که به آسانی قابل استفاده باشد و بطور خودکار با نرم‌افزارهای مجازی‌سازی کار کند و کاربران مجبور به انجام پیکربندی خاصی برای استفاده از آن نباشند. تنها کاری که لازم است انجام دهند این است که Intel VT را در BIOS سیستم خود فعال کنند و بقیه کارها خودش انجام می‌شود. بدلیل سهولت استفاده، Intel VT نه تنها در پردازنده سرورها، بلکه در پردازنده رایانه‌های رومیزی هم قرار گرفته است. به این معنا که حتی کاربران خانگی نیز می‌توانند از مزایای تکنولوژی مجازی‌سازی اینتل استفاده کنند. در حقیقت Intel VT یکی از دلایلی است که بسیاری از کاربران خانگی به سمت اجرای ماشین‌های مجازی در رایانه‌های خود سوق پیدا کرده‌اند و می‌توانند در اغلب اوقات کارآیی در حد نزدیک به حالت واقعی را دریافت کنند.

[1] hardware-assisted virtualization

۶-۷- تکنولوژی‌های مجازی‌سازی فضای ذخیره‌سازی

مجازی‌سازی ذخیره‌سازی در ساده‌ترین شکل خود شامل گروه بندی چندین دستگاه ذخیره‌سازی فیزیکی با همدیگر است (مهم نیست که از نظر فیزیکی این دستگاه ها کجا قرار دارند)، بنابراین بعنوان یک واحد ذخیره‌سازی بزرگ به نظر خواهند رسید. تکنولوژی‌های مجازی‌سازی زیادی در این خصوص وجود دارد. NAS [۱]، SAN [۲] و iSCSI [۳] نمونه هایی از آن هستند.

۶-۷-۱- NAS (Network-attached Storage)

یک NAS یک سرور است که به سرویس اشتراک فایل اختصاص داده شده است. این سرویس در ارتباط با دیگر سرویس‌ها نظیر e-mail یا اشتراک چاپگر نیست و تنها مخصوص اشتراک‌گذاری فایل است. تکنولوژی NAS مزایای زیادی برای کاربران دارد. برای مثال ظرفیت ذخیره‌سازی بیشتر را به شبکه می‌افزاید بدون اینکه نیاز باشد سرورهای موجود خاموش شوند، بنابراین زمان دسترسی داده‌ها و uptime افزایش می‌یابد. در زمان استفاده از تکنولوژی NAS، فضای ذخیره‌سازی نیازی نیست که بخشی فیزیکی از یک سرور باشد. یک سرور هنوز همه درخواست های داده ای که دریافت می‌کند پردازش خواهد کرد، اما نهایتا دستگاه NAS است که داده را به کاربر تحویل می‌دهد. دستگاه NAS می‌تواند هرجایی در یک شبکه LAN قرار داشته باشد، یعنی می‌توان آن را به دلخواه متمرکز یا توزیع شده در شبکه مستقر کرد. یک NAS می‌تواند خودش از تعداد زیادی دستگاه های NAS تشکیل شده باشد. بعبارت دیگر می‌توان یک استخر [۴] حجیم از فضای ذخیره‌سازی ایجاد کرد.

۶-۷-۲- SAN (Storage Area Network)

یک SAN زیر مجموعه‌ای از یک شبکه است که تنها شامل دستگاه های ذخیره‌سازی است. این دستگاه ها می‌توانند هم متشکل از سرورهایی با تعداد دیسک زیاد باشند و هم rack هایی که در آنها تعداد زیادی دیسک قرار داشته باشد. SAN ها بگونه‌ای طراحی شده‌اند که توسط همه سرور در شبکه محلی سازمان و نیز WAN سازمان قابل دسترسی باشند. وقتی فضای بیشتر به SAN افزوده می‌شود، فوراً توسط دیگر سرورها در کل سازمان قابل دسترسی خواهد بود. سرورها مانند یک gateway بین SAN ها و کاربران نهایی عمل می‌کنند. همچنین در هدایت ترافیک نیز کمک می‌کنند. SAN ها برای سازمان مزایای زیادی دارند. یک مزیت آن امکان استفاده بهینه تر از منابع سرورها است. زیرا یک سرور نیاز نیست که خودش درگیر ذخیره‌سازی داده شود و تنها ترافیک را به سمت دیگری هدایت می‌کند و بنابراین توان پردازشی زیادی آزاد می‌شود. در نهایت اینکه با استفاده از تکنولوژی SAN، کاربران نهایی بجای دسترسی به بخش

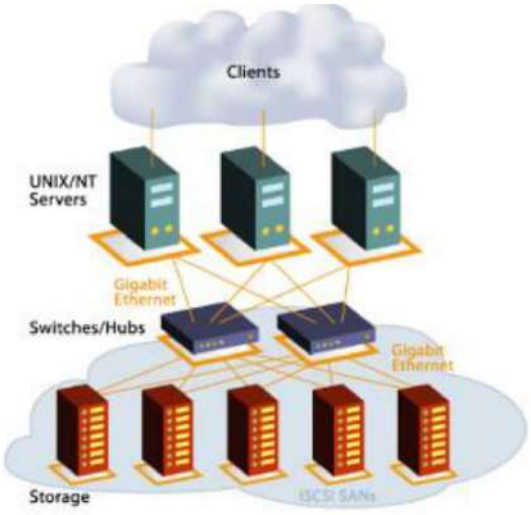
---

[1] Network attached Storage
[2] Storage Area Network
[3] Internet SCSI
[4] pool

کوچکی از منابع ذخیره‌سازی که بصورت عادی در یک سرور موجود است، به کل منابع ذخیره‌سازی سازمان خود دسترسی خواهند داشت. در نتیجه کاربران توانایی ذخیره‌سازی بیشتری خواهند داشت که بدون SAN این کار غیر ممکن بود.

iSCSI -۳-۷-۶

واژه iSCSI مخفف Internet Small Computer System Interface است. SCSI یک واسط است که برای اتصال دستگاه های جانبی (نظیر هارد دیسک) به رایانه مورد استفاده قرار می‌گیرد. این واسط مانند یک کنترلر عمل می‌کند و نیاز به نرم‌افزاری برای پردازش درخواست های داده ندارد و خودش به تنهایی عمل می‌کند.



شکل ۶-۱۲- نمونه ای از تکنولوژی iSCSI [۹]

مزیت استفاده از SCSI بجای IDE [1] برای اتصال هارد دیسک به رایانه این است که SCSI نرخ تبادل داده سریع‌تری دارد. خصوصاً اینکه در یک محیط مبتنی بر سرور که چندین کاربر همزمان درخواست دسترسی به درایوها را دارند، SCSI بازدهی بهتری نسبت به IDE دارد. همچنین SCSI به شما امکان اتصال همزمان چندین دستگاه جانبی را بصورت زنجیره وار به یک واسط SCSI می‌دهد. یعنی می‌توان تعداد زیادی دیسک را به یک ماشین متصل کرد که بیشتر از تعدادی است که توسط IDE امکان‌پذیر است. ISCSI یک توسعه بر روی تکنولوژی SCSI است که در آن از تکنولوژی IP استفاده می‌کند. ISCSI می‌تواند از شبکه

[1] Integrated Drive Electronics

Gigabit Ethernet استفاده کند و قادر است که به تجهیزات شبکه نظیر سوئیچ‌ها و روترهایی که واسط iSCSI داشته باشند متصل شود. در این حالت دستگاه های ذخیره‌سازی iSCSI می‌توانند به سرعت در شبکه مورد دسترسی قرار بگیرند.

همانطور که مشاهده شد، تکنولوژی‌های مجازی‌سازی فضای ذخیره‌سازی مزایای زیادی برای سازمان‌ها فراهم می‌کنند. وقتی که چندین فضای ذخیره‌سازی فیزیکی بصورت یک فضای واحد به نظر می‌رسد یا اینکه چندین فضای توزیع شده مانند فضای محلی قابل دسترسی هستند، راه‌حل‌های ذخیره‌سازی می‌توانند با انعطاف بیشتر و یا قدرت مدیریت بیشتری نسبت به قبل بکار گرفته شوند.

۶-۸- نرم‌افزارهای مجازی‌سازی

نرم‌افزار مجازی‌سازی به هر نوع نرم‌افزاری گفته می‌شود که در ارتباط با شبیه‌سازی سخت‌افزار و تقسیم کردن منابع سخت‌افزاری و نرم‌افزاری است. بعضی از اکثر نرم‌افزار های مجازی‌سازی سیستم‌های فیزیکی[1] شامل VMware ESX server و Windows Server Hyper-V می‌باشد. برنامه‌های مجازی‌سازی سیستم‌های فیزیکی مستقیما بر روی سخت‌افزار فیزیکی همانند یک سیستم عامل معمولی نصب می‌شوند. سپس سیستم‌های عامل میهمان می‌توانند روی آن نصب شوند.

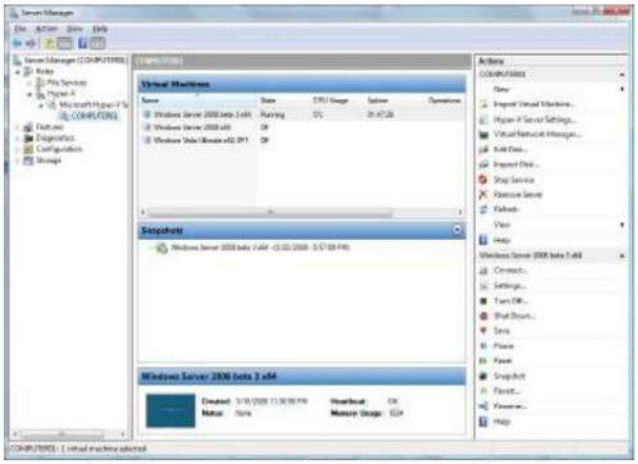
نقطه مقابل برنامه مجازی‌سازی سخت‌افزار فیزیکی، مجازی‌سازی مبتنی بر برنامه کاربردی است. این برنامه مستقیما روی سیستم عامل میزبان نصب می‌شود و شبیه هر برنامه کاربردی دیگری مورد استفاده قرار می‌گیرد. سپس در داخل این برنامه می‌توان تعدادی سیستم عامل میهمان با سیستم عامل های مختلف نصب کرد.

روش دوم مجازی‌سازی که مبتنی بر برنامه کاربردی است به اندازه مجازی‌سازی سخت‌افزار فیزیکی کارآیی ندارد، زیرا یک برنامه کاربردی نمی‌تواند مستقیما با سخت‌افزار صحبت کند و با سیستم عامل میزبان خود ارتباط برقرار می‌کند. بعضی از برنامه‌های کاربردی مجازی‌سازی که امروز موجود هستند شامل VMware workstation و Microsoft Virtual PC می‌باشند. دیگر نرم‌افزارهای مجازی‌سازی شامل نرم‌افزارهای مجازی‌سازی برنامه‌های کاربردی هستند. در این نرم‌افزارها سعی می‌شود که یک محیط مجازی مجزا ایجاد شود تا در داخل آن برنامه کاربردی اجرا شود. این محیط‌ها کاملا از هم ایزوله هستند بطوریکه تداخل نرم‌افزاری بوجود نخواهد آمد و مهم تر اینکه تغییرات اعمال شده در این محیط‌ها بر روی سیستم شما دائمی نخواهند بود. برنامه‌های مشهوری که در این دسته قرار می‌گیرند شامل Symantec's Altiris SVS و Microsoft's Softgrid می‌باشد. در ادامه لیستی از نرم‌افزارهای مجازی‌سازی که امروزه موجود هستند آورده شده است:

- **EasyVZ**: یک برنامه کد متن باز در لینوکس با واسط گرافیکی برای مدیریت سرورهای مجازی خصوصی شان
- **Fabric Server:** نرم‌افزاری برای مجازی‌سازی برنامه‌های کاربردی. به کمک آن می‌توان یک برنامه کاربردی را از سیستم عامل حذف کرد و در داخل یک محیط مجزا قرار داد.
- **FluidVM**: یک برنامه مدیریت مجازی‌سازی که از چندین روش مختلف مجازی‌سازی پشتیبانی می‌کند.
- **GridServer**: یک برنامه مجازی‌سازی که توسط DataSynapse ارائه شده است. به کمک آن می‌توان برنامه‌ها را مجازی کرد و از سیستم سخت‌افزاری و منابع سیستم عامل مجزا کرد.

[1] Bare metal

- **HP Integrity Virtual Machines**: یک نرم‌افزار مجازی‌سازی مبتنی بر برنامه کاربردی که توسط HP ارائه شده است. به کمک آن می‌تواند چندین ماشین مجازی ایجاد کرد که بر روی هر سرور مبتنی بر پردازنده Itanium که سیستم عامل آن HP-UX اجرا شود. پردازنده Itanium متعلق به اینتل و ۶۴ بیتی است و تفاوت عمده ای با معماری استاندارد x86 دارد.
- **Hercules Emulator**: یک شبیه ساز کدمتن باز که می‌تواند روی ویندوز، لینوکس و MAX OS X اجرا شود و به شما امکان اجرای برنامه‌های طراحی شده در mainframe های IBM را در سیستم‌های PC می‌دهد.
- **Hyper-V (Windows Server Virtualization)**: این نرم‌افزار اصلی مجازی‌سازی مایکروسافت است که در دو نگارش موجود است. هم در Windows Server 2008 موجود است و هم می‌توان آن را بصورت مستقل بر روی سخت‌افزار فیزیکی نصب کرد.
- **HyperVM**: یک برنامه مدیریت مجازی‌سازی محصول مایکروسافت است. این برنامه تعداد از تکنولوژی‌های مجازی‌سازی را با پلت‌فرم‌های مختلف و ماشین‌های مختلف پشتیبانی می‌کند.



شکل ۶-۱۳-نمونه از صفحه برنامه HyperVM مایکروسافت [۱۰]

- **Leostorm P>V direct**: یک ابزار برای تبدیل ماشین فیزیکی به مجازی. این برنامه می‌تواند ماشین‌های فیزیکی مایکروسافت را به ماشین‌های مجازی VMware یا سرورهای مجازی Microsoft تبدیل کند.
- **Leostorm Virtual Desktop Connection Broker**: یک برنامه مجازی‌سازی که می‌تواند برای مانیتور کردن و کنترل تعدادی از ماشین‌های مجازی که تحت یک برنامه یکسان اجرا

شده‌اند مورد استفاده قرار بگیرد. مثلا اگر از Microsoft Virtual PC استفاده می‌کنید، می‌توانید با استفاده از این برنامه همه ماشین‌های مجازی آن را مانیتور و کنترل کنید.

- **LivePC:** به کمک این برنامه می‌توان PC های مجازی ایجاد کرد و به اشتراک گذاشت.
- **Mac-on-Linux:** این یک برنامه مبتنی بر لینوکس است که به کمک آن می‌توان ماشین‌های مبتنی بر Mac را روی سیستم عامل لینوکس اجرا کرد.
- **Mac-on-Mac (MoM):** یک برنامه برای اجرای انواع مختلف سیستم‌های عامل Mac و لینوکس در داخل ویندوز یا Mac.
- **Microsoft Virtual PC:** یک نرم‌افزاری مجازی‌سازی مبتنی بر برنامه کاربردی که تنها می‌توان سیستم‌های عامل ویندوز را در آن اجرا کرد. البته امکان اجرای برخی لینوکس ها هم وجود دارد، اما بطور رسمی پشتیبانی نمی‌شوند و تضمینی برای سازگاری آنها نیست.
- **Microsoft Virtual Server:** شبیه Microsoft Virtual PC برای استفاده از سخت‌افزارهای سرور که در آن سیستم‌های عامل بیشتری (از جمله لینوکس Red Hat و Suse) پشتیبانی می‌شود.
- **MojoPac:** یک برنامه مجازی‌سازی که به شما امکان اجرای PC های مجازی را روی هارد دیسک، فلش دیسک، iPod و حتی تلفن‌های همراه می‌دهد. توسط آن می‌توان هر کاری که یک PC عادی اجرا می‌کند را انجام داد (مثلا انجام بازی های با پردازش بالا). نکته جالب در مورد آن قابلیت حمل برنامه است.
- **MokaFive (Moka5) Live PC Engine:** یک برنامه مدیریت مجازی‌سازی که با تکنولوژی LivePC کار می‌کند. موتور LivePC به کاربر امکان ایجاد، اجرا و به اشتراک‌گذاری LivePC های مجازی را می‌دهد.
- **Oracle VM:** نرم‌افزار مجازی‌سازی شرکت اوراکل است که از هر دو سیستم عامل ویندوز و لینوکس پشتیبانی می‌کند و کنسول مدیریت تحت وب مخصوص به خودش را دارد.
- **Parallels Desktops for MAC:** یک برنامه full virtualization مبتنی بر Mac که توسط موسسه Parallels ایجاد شده است. این برنامه از تعدادی از سیستم عامل های ۳۲ بیتی x86 نظیر توزیع‌های مختلف لینوکس، ویندوز، Solaris و ... پشتیبانی می‌کند. البته در حال حاضر ۶۴ بیتی ها را پشتیبانی نمی‌کند.
- **Parallels Server for MAC:** مشابه Parallels Desktop برای برنامه‌های Mac است که در سرورها اجرا می‌شود و تعداد زیادی از سیستم عامل ها (۳۲ بیتی و ۶۴ بیتی) را پشتیبانی می‌کند.

- **Parallels Workstation**: همانند Parallels Desktops است، با این استثنا که برای اجرا در محیط‌های ویندوز یا لینوکس طراحی شده است. از طریق آن می‌توان چندین ماشین مجازی x86 را هم زمان اجرا کرد.
- **PearPC**: برنامه مجازی‌سازی مبتنی بر PowerPC است که هم در لینوکس و هم در ویندوز قابل استفاده است و می‌توان سیستم عامل‌های Mac OS X، سیستم عامل های مبتنی بر یونیکس و توزیع‌های مختلف لینوکس را در آن اجرا کرد.
- **Q**: این یک برنامه کدمتن باز مبتنی بر برنامه شبیه ساز پردازنده QEMU است که بطور رایگان قابل دسترس است و در محیط‌های مکینتاش اجرا می‌شود. با استفاده از آن می‌توان ویندوز یا هر کدام از سیستم عامل های x86 را در Mac اجرا کرد. Q نسبت به QEMU کاربرپسند تر است زیرا یک واسط گرافیکی برای مدیریت و پیکربندی کنسول آن وجود دارد ولی QEMU فقط واسط خط فرمان دارد.
- **QEMU**: شبیه ساز اصلی پردازنده است که ابزارهایی برای اجرای سیستم عامل های مختلف را فراهم می‌کند. بنابراین هم یک شبیه ساز است و هم یک روش مجازی‌سازی. دیگر نکته جالب در خصوص QEMU قابلیت حمل آن است. محیط QEMU می‌تواند روی هر PC یا حتی ماشین‌هایی که کاربر بر روی آنها محدودیت دارد اجرا شود.
- **Quick Transit**: برنامه ای است که توسط شرکت Transitive توسعه داده شده است و امکان اجرا نرم‌افزارهای خاص یک پردازنده یا پلت‌فرم را در دیگر پلت‌فرم‌های نرم‌افزاری فراهم می‌آورد. این برنامه مبتنی بر لینوکس است اما برای دیگر محیط‌ها نظیر Apple نیز توسعه داده شده است که در آن به این تکنولوژی Rosetta گفته می‌شود.
- **SIMH**: این یک برنامه مجازی‌سازی است که روی بسیاری از سیستم عامل ها از جمله ویندوز، لینوکس، Mac OS X، OpenBSD و ... اجرا می‌شود.
- **SVISTA (Serenity Virtual System)**: یک برنامه مجازی‌سازی ساخته شده توسط Serenity Systems International که روی انواع مختلف ماشین‌های فیزیکی x86 اجرا می‌شود و می‌تواند ماشین‌های مختلف مبتنی بر x86 را ایجاد کند. ماشین‌های مجازی SVISTA می‌توانند شامل سیستم عامل های ویندوز، لینوکس و BSD باشند.
- **Sun VDI software**: یک برنامه مجازی‌سازی رومیزی که توسط Sun Microsystems ایجاد شده است. این برنامه سرویس‌های desktop virtualization را با جایگزینی چندین میزکار فیزیکی با میزکارهای مجازی که در سرورهای راه دور ذخیره شده‌اند فراهم می‌کند. کاربران می‌توانند به میزکارهای مجازی خود از طریق ماشین‌های thin client دسترسی داشته باشند.
- **Sun xVM Ops Center**: یک برنامه اصلی مدیریت Sun Microsystems است که به مدیران هم در مدیریت دارایی های فیزیکی و هم دارایی ها مجازی کمک می‌کند. این برنامه

مزایای زیاد دارد. مثلا اینکه مدیران می‌توانند کل دارایی های فیزیکی و مجازی سازمان خود را از یک کنسول واحد مدیریت کنند. به روز رسانی آنها می‌تواند بطور متمرکز مدیریت شود. همچنین آنها می‌توانند گزارش جزئیات مربوط به یک یا چند ماشین خاص را مشاهده کنند.

- **Sun xVM Server**: این برنامه، نرم‌افزار مجازی‌سازی سخت‌افزارهای فیزیکی Sun است که می‌تواند روی سیستم‌های x86 و نیز ۶۴ بیتی اجرا شود. Sun xVM فرمت خاصی برای دیسک های خود ندارد و در عوض از فرمت های VMware ESX Server و Microsoft Hyper-V استفاده می‌کند بطوریکه این تکنولوژی‌ها می‌توانند با هم دیگر کار کنند.
- **Sun xVM VirtualBox**: این نرم‌افزار مجازی‌سازی مبتنی بر برنامه کاربردی است که روی پلت‌فرم‌های سخت‌افزاری x86 اجرا می‌شود و می‌توان انواع مختلف سیستم عامل ها را در آن اجرا کرد.

شکل ۶-۱۴-نمونه ای از اجرای سیستم عامل ویندوز در مکینتاش با استفاده از Sun xVM VirtualBox [۱۱]

- **VDSmanager**: این برنامه ای است که به مدیران امکان مدیریت سرورهای مجازی خصوص را می‌دهد. مرکز کنترل آن تحت وب است و یک واسط گرافیکی نسبتا ساده را برای مدیران فراهم می‌کند که از طریق آن می‌توان سرورهای مجازی را ایجاد و مدیریت کرد. در هر VPS مجموعه سخت‌افزارهای شبیه‌سازی شده خود و آدرس IP مخصوص به خود را دارد.

- **VMmark:** این یک برنامه ارزیابی (benchmark) ماشین مجازی رایگان است که توسط VMware توسعه داده شده است. از این برنامه می‌توان برای اندازه گیری کارآیی یک سرور مجازی وقتی که زیر بار زیادی قرار می‌گیرد استفاده کرد. این ابزار مناسبی برای مدیران است که به آنها امکان ارزیابی کارآیی ماشین‌های مجازیشان را می‌دهد.
- **VMware ESX Server:** برنامه مجازی‌سازی سخت‌افزار فیزیکی است که توسط VMware توسعه داده شده است. این یکی از راه‌حل‌های مجازی‌سازی است که بطور گسترده استفاده می‌شود و سیستم عامل های مختلف را پشتیبانی می‌کند.
- **VMware Fusion:** ابزاری است که توسط VMWare برای ماشین‌های مکینتاش توسعه داده شده است. این ابزار مبتنی بر برنامه کاربردی است که بر روی سیستم عامل Mac نصب می‌شود و می‌توان انواع مختلف ماشین‌های x86 را روی آن اجرا کرد (نظیر ویندوز، لینوکس و Solaris).
- **VMware Player:** نرم‌افزار رایگانی است که توسط VMware ارائه شده است و از آن می‌توان برای اجرای ماشین‌های مجازی که توسط دیگر محصولات VMware ایجاد شده است استفاده کرد. ولی توسط خود این ابزار نمی‌توان ماشین مجازی ایجاد کرد.

شکل ۶-۱۵-نمونه ای از صفحه برنامه VMware Player

- **VMware Server:** یک راه‌حل مجازی‌سازی که برای ایجاد، اجرا و تغییر ماشین‌های مجازی قابل استفاده است. این سرویس بصورت کلاینت-سرور است که کاربران می‌توانند از راه دور با استفاده از thin client به ماشین‌ها دسترسی داشته باشند.

- **VMware ThinApp:** پیش از این بنام Thinstall شناخته می‌شود که یک برنامه مجازی‌سازی است که برنامه‌ها را بصورتی که قابل حمل باشند مجازی می‌کرد. اکثر برنامه‌های کاربردی مبتنی بر لینوکس می‌توانند توسط این برنامه بصورت قابل حمل ایجاد شوند. البته محدودیت‌هایی نیز وجود دارد. VMware ThinApp نمی‌تواند مجازی‌سازی را برای برنامه‌هایی که نیاز به درایور خاصی برای اجرا دارند انجام دهد. همچنین نمی‌تواند مجازی‌سازی را برای نرم‌افزارهایی که قفل سخت‌افزاری برای اجرا دارند یا نیاز به محیط محاسباتی خاصی دارند انجام دهد. بعبارت دیگر شما نمی‌توانید برنامه‌های مجازی مبتنی بر لینوکس یا Mac را بر روی سیستم عامل ویندوز اجرا کنید و برعکس.
- **VMware Workstation:** ابزار مجازی‌سازی مبتنی بر برنامه کاربردی است که به کاربران امکان ایجاد انواع مختلف ماشین‌های مجازی مبتنی بر x86 را می‌دهد. VMware از انواع مختلف سیستم‌های عامل میهمان پشتیبانی می‌کند. یکی از بزرگترین ویژگی‌های آن امکان تهیه تصویر لحظه ای[1] از سیستم عامل در حال اجرا است. به این معنا که ماشین‌های مجازی کاربران می‌تواند به آسانی بازیابی شود یا یک شخص می‌تواند در صورت نیاز به یک وضعیت قبل برگردد.
- **Virtual DOS Machine (VDM):** VDM به شما امکان نصب سیستم عامل های قدیمی 16 بیتی را در سخت‌افزارهای امروزی می‌دهد. تکنولوژی VDM در حال حاضر در همه ویندوز های ۳۲ بیتی در دسترس است. اما از آخرین نگارش سیستم‌های عامل Windows NT 64-bit حذف شد و بنابراین در آنها نمی‌توان برنامه‌های ۱۶ بیتی DOS یا ویندوز قدیمی را اجرا کرد.
- **Virtual Iron:** Virtual Iron یک برنامه مجازی‌سازی است که می‌تواند مستقیما بر روی سخت‌افزارهایی که از آن پشتیبانی کنند، بدون نصب سیستم عاملی دیگر نصب شود. Virtual Iron یکی از اولین شرکت‌هایی بود که نرم‌افزارهای مجازی‌سازی را توسعه داده و از تکنولوژی hardware-assisted virtualization نیز بهره برداری می‌کند. در بستر آن می‌توان سیستم‌های عامل ۳۲ بیتی و ۶۴ بیتی را بدون تغییر یا کارآیی نزدیک به حالت عادی اجرا کرد. Virtual Iron دارای یک برنامه مدیریت مجازی‌سازی نیز است که به مدیران امکان کنترل، تغییر یا مانیتور کردن محیط مجازیشان را می‌دهد.
- **Virtual Machine Manager:** این یک برنامه مدیریتی رایج برای مدیریت ماشین‌های مجازی است. این برنامه بصورت رایگان با تعداد از توزیع‌های لینوکس نظیر Red Hat Enterprise Linux 5 به بعد، Fedora 6 به بعد و Ubuntu 8.04 به بعد موجود است. این برنامه امکان ایجاد، تغییر، کنترل و مشاهده جزئیات آماری عملکرد ماشین‌های مجازی و آمار وضعیت بهره‌وری منابع توسط ماشین‌های مجازی را فراهم می‌کند.

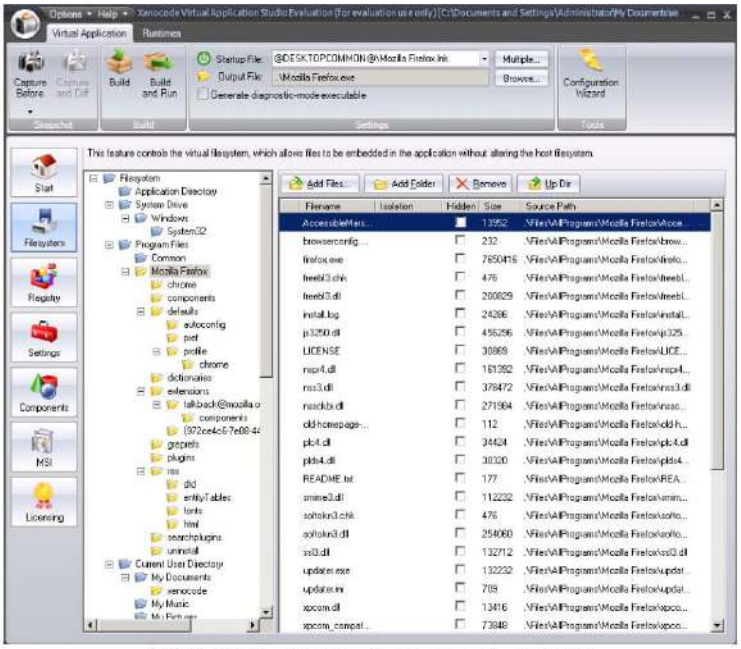
[1] snapshots

- **VirtualBox**: این یک ابزار مجازی‌سازی مبتنی بر برنامه کاربردی است که توسط شرکت آلمانی Innotek توسعه داده شده است. این ابزار می‌تواند سخت‌افزار مبتنی بر x86 استاندارد را شبیه‌سازی کند و انواع مختلف سیستم عامل های میهمان را اجرا کند. از جمله سیستم عامل هایی که می‌توانند روی آن اجرا شوند می‌توان به Windows XP، Vista، Linux، Mac OS X و ... نام برد. VirtualBox از سیستم عامل های میهمان ۶۴ بیتی بر روی سخت‌افزارهای ۶۴ بیتی هم پشتیبانی می‌کند. همچنین می‌تواند از تکنولوژی‌های hardware-assisted اینتل و AMD هم استفاده کند. یکی از معایب استفاده از VirtualBox این است که دارای فرمت اختصاصی برای دیسک های مجازی خود است. به این معنا که ماشین‌های مجازی آن با ماشین‌های مجازی ساخته شده توسط دیگر برنامه‌های مجازی‌سازی ناسازگار است. البته VirtualBox می‌تواند دیسک های VMDK [1] که فرمت متعلق به VMware است را بخواند و بر روی آنها بنویسد. یعنی VirtualBox از ماشین‌های مجازی ساخته شده توسط VMware پشتیبانی می‌کند ولی VMware نمی‌تواند از ماشین‌های مجازی VirtualBox پشتیبانی کند.
- **Virtuozzo**: برنامه Virtuozzo راه‌حل مجازی‌سازی در سطح سیستم عامل است که توسط شرکت Parallels ایجاد شده است. این برنامه از سال ۲۰۰۱ از سیستم عامل لینوکس پشتیبانی می‌کرد و از ۲۰۰۵ ویندوز را نیز پشتیبانی می‌کرد. توسط Virtuozzo می‌توان تعدادی محیط مجازی (که به آنها VE یا container گفته می‌شود) ایجاد کرد. این محیط‌ها بطور کامل از هم ایزوله هستند و بعنوان ماشین‌های مجزا یا سرورهای مجزا روی یک ماشین فیزیکی واحد کار می‌کنند. وقتی که Virtuozzo در ویندوز XP اجرا می‌شود، همه container ها می‌توانند فقط محیط ویندوز XP را اجرا کنند. اما در لینوکس انعطاف پذیری بیشتر است. زیرا در حالتی که در یک میزبان مبتنی بر لینوکس اجرا می‌شود، می‌توان توزیع‌های مختلف لینوکس را در container ها اجرا کرد.
- **Vx32**: این ابزار مجازی‌سازی در سطح برنامه کاربردی است. به کمک آن می‌توان محیط‌های مستقل از سیستم عامل و مجزا اجرا کرد که در آنها برنامه‌های مبتنی بر x86 بتوانند اجرا شوند. در این محیط‌ها می‌توان هرگونه افزودنی (plug-ins) که در زبان های مختلف نوشته شده باشند اجرا کرد. Vx32 تا حدی مشابه Java virtual machine است، اما با استفاده از Vx32 کاربران می‌توانند کدهای نوشته شده در هر زبانی را (چه امن باشد و چه نباشد) اجرا کنند. مشکل اصلی Vx32 در حال حاضر این است که با سخت‌افزارهای غیر از x86 سازگار نیست
- **Win4Lin**: برنامه Win4Lin یک راه‌حل مجازی‌سازی است که در سیستم عامل های مبتنی بر لینوکس کار می‌کند. این برنامه به شما امکان اجرای انواع مختلف سیستم عامل های ویندوز را در یک پنجره جداگانه در محیط لینوکس می‌دهد. سیستم عامل های اصلی ویندوز که توسط

[1] Virtual Machine Disk Format

Win4Lin پشتیبانی می‌شوند به این شرح هستند: Win95، Win98، WinME، Win2000 و WinXP. از این برنامه در حال حاضر دو نسخه اصلی وجود دارد: Win4Lin Pro Desktop و Win4Lin Virtual Desktop Server. نسخه Win4Lin Pro Desktop مانند یک نسخه بهبود یافته از برنامه QEMU است. نسخه سرور کمی متفاوت است. در این نسخه کاربران می‌توانند به محیط‌های سیستم عامل ویندوز از طریق thin client متصل شوند.

- **XenMan**: این یک برنامه مدیریت مجازی‌سازی مبتنی بر لینوکس است. این برنامه برای کاربران یک واسط گرافیکی فراهم میآورد که از طریق آن قادر خواهند بود که اجرا یک ماشین مجازی را کنترل کنند و یا تصویر لحظه ای از آن تهیه کنند و ...
- **Xenocode**: این یک نرم‌افزار مجازی‌سازی برنامه کاربردی است به کاربران اجازه اجرا برنامه‌ها را بصورت مجازی و بدون نیاز به نصب اجرا کنند. Xenocode برنامه‌ها را در یک فایل قابل اجرا بسته‌بندی می‌کند. این فایل می‌تواند مانند هر برنامه قابل اجرای دیگری بصورت عادی در محیط ویندوز اجرا شود. هیچ تداخلی بین برنامه‌های Xenocode و برنامه‌های ویندوزی که از DLL مشترک یا ورودی‌های رجیستری مشترک استفاده می‌کنند بوجود نخواهد آمد، زیرا Xenocode برنامه‌های مجازی را در یک محیط مجزا از بقیه سیستم اجرا می‌کند و آنها یک کپی از منابع مربوط به خود را در اختیار دارند. Xenocode مزایای زیادی دارد. مهمترین آنها به این شرح می‌باشد:
  - نیازی به نصب برنامه‌ها نسبت و بنابراین خیلی سریع استقرار می‌یابند.
  - بعضی برنامه‌های قدیمی که در محیط‌های جدیدتر (نظیر ویستا) کار نمی‌کنند توسط این برنامه می‌توانند اجرا شوند.
  - در زمان هایی که کاربران مجوز نصب برنامه‌ها را ندارند یا نمی‌توانند برنامه‌های خاصی را اجرا کنند، می‌توانند از این ابزار برای اجرای برنامه‌های خود استفاده کنند (زیرا هیچ تغییری در محیط سیستم عامل میزبان داده نمی‌شود).
  - تداخل نرم‌افزارها در محیط مجازی بوجود نمی‌آید.
  - امنیت افزایش می‌یابد. زیرا هیچ تغییری بصورت دائمی در سیستم عامل میزبان انجام نمی‌شود. همچنین برنامه‌های اجرا شده در محیط مجزای خود هستند و بنابراین برنامه‌های مخرب می‌توانند تست شوند و اثر مخرب آنها ایزوله می‌شود.



شکل ۶-۱۶- تصویر نمونه ای از محیط برنامه Xenocode [۱۲]

Xenocode یک برنامه مجازی‌سازی بسیار خوب است. اگر چه از ۲۰۰۲ معرفی شد اما هنوز بطور گسترده مورد استفاده کاربران بسیار زیادی قرار می‌گیرد. مراکزی نظیر NASA، Siemens و Phillips و ... از این نرم‌افزار استفاده می‌کنند.

پرسش‌های مروری فصل ۶
۱- مجازی سازی را تعریف کنید. اهداف مجازی سازی چیست؟
۲- ماشین مجازی شامل چه چیزهایی می‌شود؟
۳- در کدام یک از انواع مجازی سازی از فوق ناظر استفاده می‌شود؟
۴- مجازی سازی سرور چیست؟ انواع مختلف آن را نام ببرید و مقایسه کنید.
۵- سطح مجازی سازی را تعریف کنید
۶- چگالی مجازی سازی چیست؟
۷- آیا ماشین‌های مجازی از همدیگر مجزا هستند؟
۸- برای حل مشکلات پیاده‌سازی مجازی سازی نرم افزاری کامل، از چه روشی استفاده می‌شود؟
۹- تفاوت مجازی سازی نرم افزار با مجازی سازی برنامه کاربردی چیست؟

تحقیق و پژوهشی فصل ۶
۱- اگر یک سرور بخاطر مشکل سخت افزار از کار بیفتد، چه اتفاقی برای ماشین‌های مجازی درون آن می‌افتد؟
۲- service desk مجازی چیست؟ در مورد این مفهوم تحقیق کنید.
۳- انواع مختلف RAID را بررسی کنید و با هم مقایسه کنید
۴- جزئیات عملکرد Mirroring و Stripping چگونه است؟
۵- مجازی سازی میزکار چه تفاوتی با میزکارهای ابری در لایه سرویس دارد؟
۶- مجازی سازی سرور چه تاثیری در نحوه استقرار برنامه‌های کاربردی دارد؟
۷- مدل محاسباتی Plura Processing از چه روش‌های مجازی‌سازی استفاده می‌کند.
۸- مهاجرت زنده (Live) به چه معناست؟
۹- در خصوص جدیدترین دستاوردها در خصوص مجازی سازی تحقیق کنید.
۱۰- فرآیند تعدیل‌بار بکمک مجازی‌سازی چگونه انجام می‌شود. ابزارهایی را برای این کار پیدا کنید؟
۱۱- چگونه می‌توان با استفاده از مجازی‌سازی یک سیستم محاسبات داوطلبانه را بصورت سرویس مورد استفاده قرار داد؟

مراجع

[1] Ivanka Menken, Gerard Blokdijk, "Cloud Computing Virtualization Specialist Complete Certification Kit," 2008

[2] Clark Scheffy, "Virtualization For Dummies, AMD Special Edition," Wiley Publishing, Inc. www.wiley.com, 2007

[3] John Allspaw, "TheArt of Capacity Planning," O'Reilly, 2008

[4] Kevin Sloan, "Security in a virtualized world," Network Security, Volume 2009, Issue 8, August 2009, Pages 15-18

[5] SAN: *http://www.sunstarco.com/Network Attached Storage/StoreAge/StoreAge SVM.htm, accessed April 2010*

[6] Parallels Virtuozzo: *http://download.swsoft.com/virtuozzo/virtuozzo4.0/docs/ en/lin/ VzLinuxTmplMgmt/ 26225.htm, accessed April 2010*

[7] SystemGuard*: http://technet.microsoft.com/en-us/library/bb608285.aspx, accessed April 2010*

[8] How to enable the Intel VT virtualisation feature: *http://itbod.wordpress.com/2009/10/07/ how-to-enable-the-intel-vt-virtualisation-feature-on-a-supported-hp-proliant-server-running-esx-3-5-so-that-64-bit-guest-vm%E2%80%99s-can-run/ , accessed April 2010*

[9] iSCSI: *http://www.tomshardware.com/reviews/flexible-data-storage-networks,971-2.html , accessed April 2010*

[10] Microsoft HyperVM: *http://www.win2008workstation.com/win2008/installing-hyper-v-in-server-2008-x64, accessed April 2010*

[11] Sun xVM VirtualBox: *http://blogs.sun.com/perry/resource/VB_Win2008.jpg, accessed April 2010*

[12] Xenocode:*http://4sysops.com/archives/review-xenocode-virtual-application-studio, accessed April 2010*

[13] Data Center Efficiency: *http://cleantechnologyaliance.ning.com/group/datacenterefficiency*, accessed at May 2010

فصل ۷-استانداردهای رایانش ابری

- مقدمه ای بر استانداردهای ابری
- ارتباطات
- امنیت
- زیرساخت
- سرویس

۷-۱- مقدمه ای بر استانداردهای ابری

استانداردها برای محیط‌های رایانش ابری اهمیت زیادی دارند و امکان اتصال به ابر، توسعه و ارائه محتوا را فراهم می‌کنند. در این بخش استانداردهای رایجی که در رایانش ابری مورد استفاده قرار می‌گیرند را بررسی می‌کنیم. در این بین، استانداردهایی که برای توسعه برنامه‌های کاربردی در ابر مورد استفاده قرار می‌گیرند نیز بررسی خواهند شد.

۷-۱-۱- برنامه‌های کاربردی

یک برنامه کاربردی ابری، شامل معماری نرم‌افزاری است که ابر از آن استفاده می‌کند تا نیاز به نصب و اجرا در رایانه کاربر برطرف شود. برنامه‌های کاربردی زیادی هستند که می‌توانند در ابر اجرا شوند، اما به استانداردهایی بین کاربر و ابر نیاز هست که کاربر بتواند به ابر برای استفاده از برنامه کاربردی متصل شود. در ادامه به پروتکل هایی که نیاز است تا ارتباطات بین دو طرف را مدیریت کند، نگاهی می‌اندازیم.

۷-۲- ارتباطات

رایانه‌ها نیاز به روشی دارند که بتوانند با همدیگر صحبت کنند. ارتباطات را می‌توانید همانند صحبت کردن از طریق تلفن با یک شخص دیگری که فارسی صحبت نمی‌کند و شما متوجه زبان آن نمی‌شوید در نظر بگیرید. بنابراین راهی برای فهمیدن همدیگر ندارید. شاید بتوانید بعضی کلمات را حدس بزنید، اما در کل مکالمه پیش نمی‌رود. در رایانه‌ها هم موضوع از همین قرار است. البته اگر زبان آنها مشترک نباشد، حتی یک کلمه را هم نمی‌توانند تشخیص دهند. بنابراین بدون زبان مشترک، ارتباط برقرار نمی‌شود.

۷-۲-۱- HTTP

برای دریافت یک صفحه وب از سرویس دهنده ابری، معمولا از پروتکل HTTP [1] استفاده می‌شود که یک مکانیسم برای انتقال داده بین ابر و سازمان شما است. HTTP یک پروتکل بدون وضعیت است (stateless). مزیت این روش این است که میزبان‌ها نیاز ندارند تا اطلاعاتی را از درخواست کاربران نگهداری کنند. البته توسعه‌دهندگان وب از روش های دیگری برای نگهداری اطلاعات وضعیت کاربران استفاده می‌کنند. مثلا اگر بخواهند محتوای وب را برای کاربر سفارشی کنند یا اینکه مسیر پیشرفت کاربر را از یک صفحه به صفحه دیگر حفظ کنند، از کوکی ها[2] استفاده می‌کنند.

[1] Hypertext Transfer Protocol
[2] Cookies

HTTP زبانی است که ابر و رایانه شما از آن برای برقراری ارتباط استفاده می‌کنند. این زبان بسیار ساده است و تنها کافیست به مرورگر خود بگویید که چه صفحه ای را می‌خواهید مشاهده کنید. مرورگر شما با ارسال پیغام زیر، کار را شروع می‌کند:

```
GET/HTTP/1.0
Host: www.velte.com
```

و سرور مقصد پیغام زیر را در پاسخ ارسال می‌کند:

```
HTTP/1.0 200 OK
Content-Type: text/html
<head>
<title>Thank you for visiting Velte Publishing. </title>
{بقیه اطلاعات صفحه مقصد}
</body>
```

در خط اول درخواست مرورگر، پیغام GET/HTTP/1.0 می گوید که مرورگر می‌خواهد صفحه خانگی یک سایت را ببیند و در حال استفاده از نسخه ۱،۰ پروتکل HTTP است. در خط دوم پیغام Host: www.velte.com می‌گوید که چه سایتی مدنظر است.

نکته: لازم است که حتما آدرس سایت در این قسمت مشخص شود. زیرا بسیاری از سایت‌ها از یک آدرس IP اشتراکی و یکسان استفاده می‌کنند و هر چند سایت با یک آدرس IP یکسان در یک سرور قرار دارند.

خط اول پاسخ سرور، که پیغام HTTP/1.0 200 OK را نوشته است، به مرورگر می‌گوید که آن هم با پروتکل HTTP 1.0 صحبت می‌کند و درخواست موفقیت آمیز بوده است. اگر صفحه درخواستی پیدا نشود، پاسخ HTTP/1.0 404 Not Found داده خواهد شد. خط دوم پاسخ سرور، پیغام Content-Type: text/html است که به مرورگر می‌گوید که یک صفحه وب را دریافت خواهد کرد. به این ترتیب مرورگر خواهد فهمید که داده دریافت شده را چگونه تفسیر کند و چگونه با آن رفتار کند. مثلا اگر در خط دوم پیغام Content-Type:image/jpg نوشته شده بود، یعنی مرورگر باید خود را برای دریافت یک تصویر آماده می‌کرد.

HTTP 1.1: در مثال فوق از HTTP 1.0 استفاده شده بود، اما در حال حاضر مرورگرها از نگارش ۱،۱ استفاده می‌کنند. درخواست و پاسخ شامل اطلاعات بیشتری است، اما تفاوت ها خیلی زیاد و متمایز نیست. تفاوت اصلی بین این دو این است که در اصل مرورگرهای وب درخواست های هر صفحه، هر تصویر و هر بخش خاص از صفحه را از هم جداکرده‌اند. با استفاده از HTTP 1.1، یک مرورگر و سرور می‌توانند برای باز نگه‌داشتن اتصال مذاکره کنند و همه اجزای صفحه را بدون نیاز به ایجاد session های جدید، ارسال کنند.

درخواست های HTTP: پروتکل HTTP هشت روش را برای تعیین عملیات مورد نظر خود در سرور بکار می‌گیرند. در جدول انواع مختلف این درخواست ها لیست شده است.

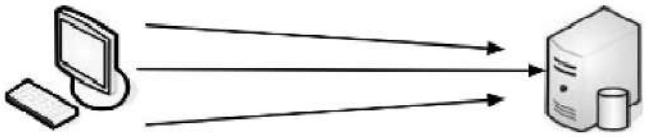
جدول ۷-۱- انواع درخواست های HTTP

Request	Description
HEAD	Asks for the response identical to the one that would correspond to a GET request, but without the response body. This is good for retrieving metainformation in the response headers, but without transporting the entire content.
GET	Requests information from a server.
POST	Submits data to be processed to the server. The data is included in the body of the request. The result of the request might be the creation of the resource or updating the existing resource.
PUT	Uploads a representation of the resource.
DELETE	Deletes the specified resource.
TRACE	Echoes the request back to the browser so that the client can see which servers are adding or changing in the request.
OPTIONS	Returns HTTP methods that the server supports for the given URL. This can be used to check the functionality of a web server.
CONNECT	Converts the request connection to a transparent TCP/IP tunnel. It's usually used to facilitate SSL-encrypted communication through an unencrypted HTTP proxy.

بطور کلی HTTP رایج ترین راه برای اتصال مرورگر شما به ابر است. یکی دیگر از پروتکل های قابل استفاده XMPP است.

XMPP -۲-۲-۷

گفته می‌شود که پروتکل XMPP [1] یکی دیگر از رویدادهای بزرگ در خصوص رایانش ابری خواهد بود. مشکل این است که تبادل اطلاعات در سرویس‌های ابری فعلی –نظیر SOAP و دیگر پروتکل های مبتنی بر HTTP- همه یکطرفه است. پروتکل XMPP این امکان ارتباط دوطرفه را فراهم کرده است و نیاز به عملیات polling را برطرف کرده است.



شکل ۷-۱- HTTP نیاز به چندین رویداد polling دارد تا وضعیت آن از مرورگر وب به روز رسانی شود

XMPP که به عنوان Jabber هم شناخته می‌شود، پروتکلی است که Google، Apple، AOL، IBM، و LiveJournal همه بر روی آن به توافق رسیده اند، اما به دلیل اینکه ایده جدیدی است، هنوز بطور گسترده از آن استفاده نمی‌شود.

[1] Extensible Messaging and Presence Protocol

شکل ۷-۲- XMPP یک ارتباط بین کلاینت و سرور فراهم می‌کند

مشکل عملیات Polling: وقتی شما می‌خواهید که سرویس‌های بین دو سرور را با هم همسان کنید، رایج ترین روش این است که کلاینت بطور منظم میزبان را چک کند. به این کار polling گفته می‌شود. این کاری است که ما معمولا وقتی ایمیل خود را چک می‌کنیم به نوعی روی سرور ایمیل polling انجام می‌دهیم تا ببینیم که آیا پیغام جدیدی دریافت کرده ایم یا نه. روش عملکرد API ها در اکثر سرویس‌های وب نیز به همین ترتیب است. سایت High Scalability در ۲۰۰۸ گزارش کرده است که Twitter بطور میانگین در ثانیه حدود ۲۰۰ تا ۳۰۰ اتصال برقرار کرده است که نقطه اوج آن ۸۰۰ درخواست بوده است و حتی در زمان سخنرانی Macworld، به علت تعداد زیاد poll ها، سرویس از دسترس خارج شده بود.

بزرگترین مشکل XMPP این است که ایده جدیدی است و معمولا تصور می‌شود که هر چیز جدیدی باید بر اساس استانداردهای موجود باشد و در حالی که HTTP به خوبی کار می‌کند، برای رایانش ابری چندان ایده‌آل نیست.

XMPP در ابتدا برای پیغام رسانی فوری توسعه داده شده بود. برخی ویژگی‌های آن به این ترتیب می‌باشد:

- امکان ارتباطات دو طرف، حذف نیاز به polling
- مبتنی بر XML و قابلیت توسعه آسان، که مناسب سرویس‌های ابری است.
- کارا و قابل توسعه به میلیون ها کاربر همزمان بر روی یک سرور واحد

۷-۳- امنیت

امن کردن session های ابر از آن جهت مهم است که بسیاری از شرکت‌ها علاقه‌مند شده‌اند تا وارد ابر شوند. امن کردن session های ابر می‌تواند از طریق رمزنگاری و احراز هویت انجام شود. روش های رمزنگاری رایج در همه مرورگرها بصورت استاندارد انجام می‌شود. احراز هویت موضوع دیگری است که گزینه‌های مختلفی در پیش روی شماست. در این قسمت درباره SSL [1] که بطور گسترده برای رمزنگاری مورد استفاده قرار می‌گیرد و نیز OpenID که یکی از روش‌های احراز هویت است صحبت خواهیم کرد.

[1] Secure Sockets Layer

۷-۳-۱- SSL

SSL تکنولوژی امنیتی استاندارد برای برقراری یک اتصال رمز شده بین یک سرور وب و مرورگر است. به کمک آن می‌توان اطمینان حاصل کرد که داده‌های ارسال شده بین مرورگر و سرور وب، امن باقی می‌ماند. برای ایجاد یک اتصال SSL، بر روی سرور وب، نیاز به گواهینامه SSL می‌باشد. وقتی سرویس دهنده ابری بخواهد از SSL استفاده کند، تعدادی سئوال در خصوص هویت سرویس دهنده و هویت سایت از سمت مرکز صادر کننده گواهینامه[1] از سرویس دهنده ابری پرسیده خواهد شد. سپس سرویس دهنده ابری دو کلید رمزنگاری تولید می‌کند (یک کلید عمومی و یک کلید خصوصی). کلید عمومی محرمانه نیست و در داخل فایل درخواست امضای گواهینامه (CSR[2]) قرار داده می‌شود (این فایل شامل اطلاعات سرویس دهنده است) ولی کلید خصوصی را نزد خودش محرمانه نگه می‌دارد. سپس باید CSR را ثبت کند. در حین فرآیند درخواست گواهینامه SSL، مرکز اعتبارسنجی و صدور گواهینامه، اطلاعات را ارزیابی می‌کند و یک گواهینامه SSL صادر می‌کند که شامل اطلاعات سرویس دهنده است و مجوز استفاده از SSL را دریافت می‌کند. هر بار که مرورگر شما به سایت سرویس دهنده متصل شود، مرورگر وب صحت گواهینامه سرور را بررسی می‌کند و سپس در صورت صحت آن یک ارتباط امن بین رایانه شما و سرویس دهنده ابری برقرار می‌کند.

شکل ۷-۳- برقراری یک ارتباط SSL بین کاربر و سرویس دهنده ابری

این یک فرآیند نسبتاً ساده است و در پس زمینه انجام می‌شود. تنها تفاوتی که شاید مشاهده کنید این است که بخاطر بررسی گواهینامه، صفحه کمی دیرتر باز می‌شود.

معمولا گواهینامه SSL شامل نام دامنه سرویس دهنده ابری، نام شرکت، آدرس، شهر، ایالت و کشور آن است. همچنین شامل تاریخ انقضای گواهینامه و جزئیات مربوط به صادر کننده گواهینامه است. وقتی مرورگر شما سعی می‌کند که بطور امن به ابر متصل شود، گواهینامه SSL سایت را دریافت می‌کند و بررسی می‌کند که منقضی نشده باشد و گواهینامه برای همان سایت صادر شده باشد. همچنین بررسی می‌کند که صادر کننده گواهینامه نیز معتبر و قابل اطمینان باشد. اگر مشکلی در این بررسی ها پیش بیاید، مرورگر به کاربر اطلاع خواهد داد که ارتباط SSL امن نیست.

---

[1] Certification Authority
[2] Certificate Signing Request

۱- مرورگر گواهینامه سایت را بررسی می‌کند تا اطمینان حاصل کند سایتی که به آن متصل می‌شود، واقعی است و کس دیگری آن را شنود یا جعل نمی‌کند.
۲- مرورگر و سایت وب بر روی نوع روش رمزنگاری مورد استفاده به توافق می‌رسند.
۳- مرورگر و سرور هر کدام یک کد منحصر به فرد برای هم ارسال می‌کنند تا در زمان ارسال اطلاعات رمزنگاری مورد استفاده قرار بگیرد.
۴- مرورگر و سرور با استفاده از روش رمزنگاری مورد استفاده شروع به برقراری ارتباط می‌کنند.
۵- مرورگر آیکن مربوط به رمزنگاری را در صفحه نمایش می‌دهد که نشان دهنده این است که صفحات امن هستند.

OpenID -۲-۳-۷

OpenID یک راه‌حل کدمتن باز برای مشکل نیاز به نام کاربری و کلمه عبور واحد برای دسترسی به سایت‌های مختلف است، بنابراین پیمایش در وب را تسهیل می‌کند. این راه‌حل به شما امکان می‌دهد تا بتوانید سرویس دهنده OpenID خود را که بهتر با نیازهای شما منطبق است و به آن اعتماد دارید، انتخاب کنید. همچنین OpenID رایگان است و برای شرکت‌ها بسیار مناسب است، زیرا هزینه کمتری برای مدیریت حساب کاربری و کلمه عبور دربر دارد.
OpenID هنوز در مرحله پذیرش قرار دارد و در حال محبوب شدن است. شرکت‌هایی نظیر AOL، Microsoft، Sun و Novell شروع به پذیرش و فراهم کردن OpenID ها کرده‌اند. OpenID یک محصول از جامعه کدمتن باز برای حل مشکلی است که با تکنولوژی‌های موجود قابل حل نبود. OpenID یک روش ساده و سبک برای احراز هویت کاربران، با استفاده از همان تکنولوژی است که برای احراز هویت در سایت‌ها بکار گرفته می‌شود. هر کس می‌تواند یک کاربر OpenID باشد یا بطور رایگان سرویس دهنده آن باشد.

PCI DSS -۳-۳-۷

طبق استاندارد امنیت داده در صنعت کارت های پرداخت (PCI DSS)[1]، بیان شده است که یک سازمان می‌تواند تنها یک وظیفه اصلی محاسباتی را در هر سرور پیاده‌سازی کند[2]. اما آیا این معنای یک سرور فیزیکی است؟ پاسخی منفی است. شما می‌توانید چندین سیستم مجازی شده داشته باشید. فقط کافیست مطمئن شوید که از هم دیگر کاملا مجزا و ایزوله هستند.

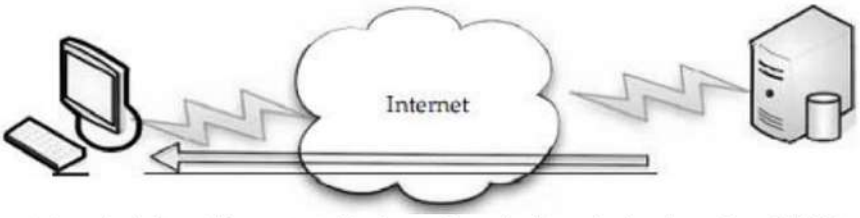
[1] Payment Card Industry Data Security Standards (PCI DSS) requirement 2.2.1
[2] implement only one primary function per server

۷-۴- زیرساخت

مجازی‌سازی راهی برای ایجاد زیرساخت در محیط رایانش ابری است. قبلا در مورد مجازی‌سازی صحبت کرده ایم، اما در این بخش خواهیم دید که مجازی‌سازی چگونه استاندارد می‌شود و این کار به چه صورت محقق خواهد شد.

۷-۴-۱- مجازی‌سازی

هر زمان که چیز جدیدی در دنیای محاسبات اتفاق می‌افتد، رقبای مختلف می‌خواهند که پیاده‌سازی های خود را به عنوان یک استاندارد ارائه کنند. مجازی‌سازی تا حدی متفاوت است و بازیگران اصلی این صنعت با هم کار کرده‌اند تا با هم یک استاندارد را توسعه دهند.



شکل۷-۴- در یک محیط مجازی شده، برنامه‌ها روی یک سرور اجرا شده و در سمت کلاینت نمایش داده می‌شوند. سرورها می‌توانند محلی باشند یا در سمت دیگر ابر قرار داشته باشند

VMWare، AMD، BEA Systems، BMC Software، Broadcome، Cisco، Dell، Emulex، HP، IBM، Intel، Mellanox، Novel، QLogic، Red Hat و انجمن بین‌المللی کامپیوتر، همه با هم کار کرده‌اند تا در استاندارد مجازی‌سازی را پیش ببرند.

VMware گفته است که دسترسی شرکای خود را به کدهای منبع VMware ESX Server و واسط‌های آن از طریق برنامه جدیدی بنام VMware Community Source فراهم می آورد. این برنامه بدین منظور طراحی شده است که شرکا بتوانند در یک مدل توسعه مشارکتی این محصول موثر باشند. مجازی‌سازی با توجه به مزایای بی چون و چرایی که دارد، در حال کسب یک پذیرش گسترده است. هدف این اقدامات این است که کاربران نهایی از طرق مختلف نفع ببرند:

- **توسعه راه‌حل‌های مجازی‌سازی:** انتظار می‌رود که در دسترس بودن استاندارد های باز برای واسط‌های مجازی‌سازی و ماهیت مشارکتی VMware Community Source سبب ایجاد راه حال های مجازی‌سازی جدید شود.

- **توسعه قابلیت همکاری**[1] **و پشتیبانی:** انتظار می‌رود که واسط‌های استاندارد برای فوق‌ناظرها امکان قابلیت همکاری را برای مشتریان با محیط‌های مجازی ناهمگن فراهم کند.
- **پشتیبانی تکنولوژی‌های جدید از مجازی‌سازی:** فروشنده‌هایی که در زمینه تکنولوژی‌های دیگر فعالیت می‌کنند (مثلا سخت‌افزارها یا برنامه‌های کاربردی)، می‌توانند تکنولوژی‌های فعلی را بگونه‌ای بهینه کنند تا تکنولوژی‌های جدید برای اجرا در محیط‌های مجازی را معرفی کنند.

۷-۴-۲- استانداردهای باز فوق‌ناظر

فوق‌ناظرها، اجزای اصلی زیرساخت مجازی هستند و امکان تقسیم‌بندی سیستم‌های رایانه‌ای را فراهم کرده‌اند. یک استاندارد باز فوق‌ناظر می‌تواند زمینه نوآوری را برای مشتریان و فروشندگان مختلف فراهم آورد. شرکت VMware با ارائه Virtual Machine Hypervisor Interfaces (VMHI) که مبتنی بر محصولات مجازی خودش است در فراهم آمدن چهارچوبی برای توسعه این استاندارد باز همکاری کرده است. پذیرش این استاندارد باز انتظار می‌رود که سبب تسهیل قابلیت همکاری و پشتیبانی در محیط‌های مجازی ناهمگن شود.

همکاری و مشارکت بر روی استانداردهای باز فوق‌ناظر، برای ایجاد قابلیت همکاری و بهینه‌سازی عملکرد، متمرکز بر موارد زیر است:

- چهارچوب‌های بین پلت‌فرم برای مدیریت ماشین‌های مجازی stand-alone
- API های مجازی‌سازی مشارکتی بین فوق‌ناظرها و سیستم‌های عامل مهمان
- قالب های ماشین مجازی و امکان مهاجرت و بازیابی در پلت‌فرم‌های مختلف

۷-۴-۳- OVF

در نتیجه تلاش‌های VMware و دیگر شرکای صنعت مجازی‌سازی، یک استاندارد بنام OVF [2] توسعه داده شد. OVF تعیین می‌کند که چگونه ابزارهای مجازی[3] بتوانند در یک فرمت مستقل از فروشنده قرار بگیرند تا در هر فوق‌ناظری اجرا شوند. این یک فرمت مستقل از سکو، قابل توسعه و باز برای بسته‌بندی و توزیع ابزارهای مجازی است که از یک یا چند ماشین مجازی تشکیل شده‌اند.

OVF به مشتریان و توسعه‌دهندگان امکان انتخاب هر فوق‌ناظری را بر اساس قیمت، ترجیحات یا کارآیی می‌دهد و از قفل شدن در یک فروشنده جلوگیری می‌کند. این استاندارد بسته‌بندی و توزیع برای ابزارهای مجازی در افزایش سرعت پذیرش ابزارهای مجازی نقش مهمی ایفا می‌کند. این استاندارد دارای ویژگی‌های زیر می‌باشد:

---
[1] interoperability
[2] Open Virtualization Format
[3] appliances

- بهینه شده برای توزیع
- قابلیت حمل و توزیع ابزارهای مجازی
- پشتیبانی از فرآیند استاندارد بررسی یکپارچگی و صحت محتوا
- فراهم کردن یک الگوی پایه برای مدیریت مجوزهای نرم‌افزاری
- یک تجربه ساده و اتوماتیک از دید کاربر
- یک رویکرد قوی و کاربر پسند برای ساده سازی فرآیند نصب
- بازبینی کل بسته و اطمینان از اینکه همه ماشین‌های مجازی نصب شده باشند
- بررسی سازگاری با سخت‌افزار محلی مجازی
- بسته‌بندی قابل حمل ماشین مجازی
- امکان بهره گیری از بهبودهای خاص هر پلت‌فرم
- پشتیبانی از محدوده گسترده ای از هارد دیسک های مجازی مورد استفاده برای ماشین‌های مجازی و قابل توسعه برای در برگرفتن فرمت های آتی
- مستقل از فروشنده و پلت‌فرم
- عدم نیاز به استفاده از یک پلت‌فرم میزبان، پلت‌فرم مجازی‌سازی یا سیستم عامل مهمان خاص
- قابل توسعه
- طراحی شده برای همگام شدن با پیشرفت تکنولوژی در صنعت
- امکان محلی سازی
- ...

۷-۵- سرویس

یک سرویس وب، طبق تعریف W3C، یک سیستم نرم‌افزاری است که برای پشتیبانی از تعامل های متقابل ماشین به ماشین در شبکه طراحی شده است. در رایانش ابری نیز اجزای مختلف می‌توانند از طریق سرویس‌های وب در دسترس قرار بگیرند. سرویس‌های وب معمولا API های وبی هستند که از طریق شبکه هایی نظیر اینترنت قابل دسترسی هستند و روی یک سیستم راه دور که میزبان سرویس‌های درخواست شده است، اجرا می‌شوند. در این بخش، ما در خصوص چند نمونه از سرویس‌های وب رایج نظیر REST و SOAP و JSON صحبت می‌کنیم.

داده: داده می‌تواند با استفاده از مکانیسم ها و ساختارهای مختلفی مورد استفاده قرار بگیرد. دو نمونه از رایج ترین آنها JSON و XML می‌باشد. هر دوی این‌ها بر اساس استاندارد های پیش رو در صنعت – HTML و جاوااسکریپت- هستند که برای کمک به ارائه و استفاده از داده مورد استفاده قرار می‌گیرند.

۷-۵-۱- JSON

JSON [1] یک قالب سبک رایانه‌ای برای تبادل داده‌ها است. از این فرمت برای ارسال داده‌های ساخت یافته در شبکه استفاده می‌شود. این فرمت معمولا می‌تواند بعنوان یک روش دیگر مشابه با XML مورد استفاده قرار بگیرد. اصول JSON بر اساس زیرمجموعه‌ای از جاوااسکریپت است که معمولاً همراه با آن زبان استفاده می‌شود. با این حال JSON یک فرمت مستقل از زبان است و کدهای مورد نیاز برای تحلیل و تولید آن در چندین زبان برنامه‌نویسی مختلف موجود است. همین موضوع باعث شده است که وقتی در تبادل داده با جاوااسکریپت درگیر هستیم (مثلا در برنامه‌های AJAX)، بتوان آن را بعنوان یک جایگزین خوب بجای XML استفاده کرد.

۷-۵-۲- مقایسه XML با JSON

زمانی بهتر است از JSON استفاده کنیم که برای ارسال یا دریافت داده از جاوااسکریپت استفاده می‌کنیم. دلیل این کار این است که وقتی از XML در جاوااسکریپت استفاده می‌شود، مجبور خواهیم بود که اسکریپت‌هایی برای استفاده از کتابخانه های مورد نیاز برای استفاده از اشیاء DOM به منظور استخراج داده‌های مورد نیاز بنویسیم. اما در JSON اشیاء همچنان به همان صورت یک شیء باقی می‌مانند. بنابراین کار اضافه بیشتری نیاز ندارند. این کار میزان سربار، استفاده از CPU و میزان کدی که برای برنامه می‌نویسید را کاهش می‌دهد. برای مثال، کد JSON نمونه زیر شیء توصیف کننده یک شخص را نشان می‌دهد:

```
{
    "firstName": "Johnny",
    "lastName": "Johnson",
```

[1] JavaScript Object Notation

```
    "address": {
        "streetAddress": "123 Main Street",
        "city": "Minneapolis",
        "state": "MN",
        "postalCode": 55102
    },
    "phoneNumbers": [
        "612 555-9871",
        "952 555-1598"
    ]
}
```

همانطور که ملاحظه شد، این شیء شامل نام شخص، آدرس و آرایه‌ای از شماره‌های تلفن شخص است.

XML -۳-۵-۷

زبان XML [1] یک استاندارد و روشی خود تعریف برای کد کردن متن و داده است بطوری که محتوا با کمترین تعامل انسانی قابل دسترسی و قابل مبادله در بین طیف مختلف سخت‌افزارها، سیستم‌های عامل و برنامه‌های کاربردی باشد. XML روشی استاندارد برای ارائه متن و داده در فرمتی که بتواند مستقل از پلت‌فرم استفاده شود، فراهم آورده است. همچنین می‌تواند با طیف گسترده‌ای از ابزارهای توسعه و برنامه‌نویسی و ابزارهای دیگر مورد استفاده قرار بگیرد.

XML بسیار شبیه HTML است (هر دوی آنها بر اساس زبان SGML هستند که از سال ۱۹۸۶ یک استاندارد شده است). البته دو تفاوت عمده بین این دو وجود دارد:

- در HTML از نشانه ها برای تعریف محتوا و ظاهر متن استفاده می‌شود در صورتیکه در XML محتوا و ساختار آن توصیف می‌شود، ظاهر نمایش متن بطور مجزا بر اساس کاربرد به هر شکل دلخواه قابل تعیین است.
- نشانه ها در XML می‌توانند توسط خود توسعه‌دهندگان، متناسب با هر کاربرد دلخواه تعریف شوند در حالی که نشانه‌ها در HTML توسط موسسه W3C تعیین می‌شوند.

از نظر عملکردی، XML استفاده از پایگاه‌داده را برای سازمان شما ساده‌تر می‌کند. سیستم‌های پایگاه‌داده رابطه‌ای نمی‌توانند همه نیاز سیستم‌های الکترونیکی را بر طرف کنند. همچنین نمی‌توانند ساختارهای داده‌ای خاص و تودرتو را به همان صورت که هستند مدیریت کنند. این‌ها بخشی از نیازمندی‌های رایج در محیط‌های ابری می‌باشد. اگرچه امکان کار با XML در اغلب پایگاه‌های داده سنتی فراهم شده است، اما فرآیند تبدیل مورد استفاده در آنها معمولا سربار زیادی دارد و با پیچیدگی زیادی همراه است.

پایگاه‌های داده مبتنی بر XML این فرآیند را ساده کرده‌اند، زیرا آنها XML را با همان ساختار اصلی خود و به شکل سلسله مراتبی ذخیره می‌کنند. به این ترتیب پرس‌وجوها بسیار سریعتر اجرا می‌شوند زیرا نیاز

[1] Extensible Markup Language

نیست که داده‌های درختی XML به جداول رابطه‌ای در پایگاه‌داده نگاشت شوند. دیگر مزایای XML به این شرح می‌باشد:

- **ساختار داده ای خود تعریف:** XML نیازی به جداول توصیفی و الگوهای رابطه‌ای، توصیف نوع داده‌های خارجی و ... را ندارد. همچنین در حالیکه HTML فقط کمک می‌کند که از شکل صحیح نمایش داده مطمئن شویم، XML تضمین می‌کند که داده‌ها قابل استفاده نیز باشند.
- **یکپارچه‌سازی با پایگاه‌داده:** سندهای XML می‌توانند شامل هر نوع داده ای باشند (از متن و عدد گرفته تا اشیاء چندرسانه‌ای و اشیاء فعال نظیر جاوا).
- **عدم نیاز به برنامه‌نویسی مجدد در صورت نیاز به تغییر:** اسناد و سایت‌های وب می‌توانند با XSL [1] تغییر کنند، بدون اینکه نیاز باشد در فرمت و ساختار داده‌ها تغییر داده شود.
- **دید یکپارچه از داده:** XML بطور ویژه برای محیط‌های ابری مناسب است، زیرا داده‌هایی که در چندین سرور پخش می‌شوند می‌توانند بگونه‌ای بنظر برسند که معادل یک سرور باشد.
- **باز و قابل توسعه:** ساختار XML به شما امکان افزودن عناصر مورد نیازتان را می‌دهد. شما می‌توانید به آسانی سیستم خود را با تغییرات کسب و کار منطبق کنید.
- **پشتیبانی گسترده:** XML در W3C به عنوان یک استاندارد در صنعت مطرح است و توسط بسیاری از تولیدکنندگان پیشرو در عرصه نرم‌افزار پشتیبانی می‌شود و ابزارهای بسیاری برای کار با آن توسعه داده شده است.
- **شامل اطلاعات زمینه قابل خواندن برای ماشین:** نشانه ها، مشخصه ها و عناصر ساختاری اطلاعات زمینه‌ای را برای بررسی و استفاده در ابزارهای مورد نیاز فراهم می‌کنند.
- **محتوا در مقایسه با ارائه:** نشانه های XML، بجای شکل ارائه، معنای هر شیء را توصیف می‌کنند. به عبارت دیگر، XML ساختار سند را توصیف می‌کند و برنامه کاربردی آن را به همان شکل که توصیف شده است ارائه می‌کند.

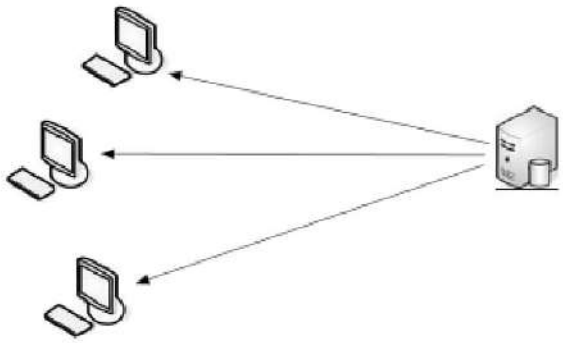
۷-۵-۴- سرویس‌های وب

سرویس‌های وب تعیین می‌کنند که داده چگونه از ابر به کلاینت ارسال شود. دو پروتکلی که در این رابطه مورد بررسی قرار می‌گیرند REST و SOAP هستند که بهترین گزینه‌ها برای نیازهای ابری هستند.

[1] Extensible Stylesheet Language

REST -۵-۵-۷

پروتکل REST[1] یک راه برای دریافت محتوای اطلاعاتی از یک سایت وب است. برای نمونه REST می‌تواند توسط سرویس دهنده ابری شما استفاده شود تا اطلاعات به روز شده عضویت را فراهم کند. هر چند وقت سرویس دهنده ابری می‌تواند صفحه وبی آماده کند که شامل محتوا و بخش‌های XML باشد. اعضا و مشترکین تنها کافیست آدرس URL آن صفحه که فایل XML در آن قرار دارد را داشته باشند تا با استفاده از یک مرورگر وب آن را بخوانند.



شکل ۷-۵- کلاینت ها درخواست اطلاعات را به سرور وب ارسال می‌کنند، وب سایت محتوای خود را به روز رسانی می‌کند و با استفاده از REST اطلاعات را برای آنها ارسال می‌کند

عملکرد REST بسیار شبیه SOAP[2] است اما استفاده از آن ساده‌تر است. استفاده از SOAP نیازمند نوشتن یا استفاده از یک برنامه سرور و کلاینت است ولی با این حال قابلیت‌های بیشتری دارد. برای مثال، اگر شما بخواهید محتوای خبرخوان را از ابر خود فراهم کنید تا سایت‌ها عضو آن شوند، آن مشترکین ممکن است لازم باشد تا از SOAP استفاده کنند، زیرا امکان تعاملات بیشتری را بین سرور و کلاینت فراهم می‌کند. REST از رویکرد انتشار مشابه RSS [3] استفاده می‌کند. RSS از چهارچوبی برای توصیف منابع[4] استفاده می‌کند که یک روش استاندارد برای توصیف سایت وب است.

منابع: یک بخش مهم در REST وجود منابع است. هر منبع اطلاعات خاصی را مشخص می‌کند و با استفاده از یک URL مشخص از طریق HTTP قابل دسترسی است. برای استفاده از این منابع، اجزای شبکه با استفاده از واسط استاندارد (نظیرHTTP) با هم ارتباط برقرار می‌کنند و شکل بازنمایی منابع را مبادله می‌کنند. مثلا یک منبع اگر "مثلث" باشد، ممکن است بصورت یک چندضلعی با سه ضلع مساوی

[1] Representational state transfer
[2] Simple Object Access Protocol
[3] RDF Site Summary
[4] Resource Description Framework (RDF)

توصیف شود، یا اینکه ممکن است ترکیبی از سه نقطه که با کاما در یک لیست از هم جدا شده‌اند مشخص شود.

مزایا: REST شامل مزایای زیر است:

- زمان پاسخ بهتر و بار سرور کمتر را بخاطر پشتیبانی از caching در ارائه بازنمایی‌ها فراهم می‌کند.
- مقیاس‌پذیری سرور با کاهش نیاز به نگه داری وضعیت sessionها بهبود یافته است.
- یک مرورگر می‌تواند به هر برنامه کاربردی و هر منبعی دسترسی داشته باشد. بنابراین نرم‌افزار سمت کلاینت کمتری مورد نیاز است.
- با توجه به اینکه از hyperlink ها برای بازنمایی استفاده می‌کند، یک مکانیسم کشف منبع مجزا نیاز ندارد.
- سازگاری بهتر در دراز مدت و مشخصه های قابل تکامل نسبت به RPC بخاطر:
  - امکان تکامل تدریجی اسنادی نظیر HTML
  - امکان افزودن پشتیبانی از انواع داده جدید بدون از بین رفتن پشتیبانی از انواع داده قدیمی

یک مزیت در زمان استفاده از برنامه‌های REST در ابر این است که REST به کاربران این امکان را می‌دهد تا پرس‌وجوهای خاص را ذخیره (bookmark) کنند و آن پرس‌وجوها را از طریق ایمیل یا برنامه‌های پیغام رسان برای دیگران ارسال کنند. این شکل بازنمایی از مسیر یا نقطه ورود به یک برنامه کاربردی بسیار قابل حمل است.

۷-۵-۶- SOAP

پروتکل SOAP [1] روشی است برای اینکه یک برنامه بتواند با دیگر برنامه‌ها در همان سیستم عامل یا دیگر سیستم عامل ها با استفاده از HTTP و XML (که ابزارهای تبادل اطلاعات هستند) ارتباط برقرار کند. فراخوانی تابع از راه دور (RPC) بین اشیائی نظیر DCOM یا COBRA مورد استفاده قرار می‌گیرد، اما HTTP بگونه‌ای طراحی نشده است که بتواند از آن استفاده کند. مشکل اصلی RPC سازگاری آن است، همچنین دیوارآتش و پروکسی سرورها جلوی این نوع ترافیک را می‌گیرند. بدلیل اینکه پروتکل‌های وب نظیر HTTP و XML در تمام سیستم عامل های بزرگ از قبل نصب شده و موجود هستند، یک راه‌حل آسان برای حل مشکل برقراری ارتباط بین برنامه‌هایی که در سیستم عامل‌ها و شبکه‌های مختلف قرار دارند، بحساب می‌آیند.

[1] Simple Object Access Protocol

SOAP تعیین می‌کند که چگونه سرآیند HTTP و فایل XML را بگونه‌ای آماده کند تا یک برنامه روی یک رایانه بتواند یک برنامه روی رایانه دیگر را فراخوانی کند و اطلاعات را به آن ارسال کند. همچنین این پروتکل شرح می‌دهد که چگونه برنامه فراخوانی شده، نتیجه را برگرداند.

یکی از مزایای SOAP این است که فراخوانی ها با احتمال بیشتری می‌توانند از دیوار آتش عبور کنند. بدلیل اینکه تقاضاهای HTTP معمولا مجوز عبور از دیوارآتش را دارند، برنامه‌هایی که از SOAP استفاده می‌کنند می‌تواند با برنامه‌ها در هرجایی ارتباط برقرار کنند.

مثال: به مثال زیر که نمونه ای از SOAP است نگاه کنید. شما می‌توانید ببینید که آن بر اساس HTTP است. در حقیقت اولین خط این درخواست بسیار شبیه به استاندارد درخواست HTTP تعیین می‌شود:

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-
encoding">
  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

و به همین ترتیب پاسخ SOAP نیز مشابه پاسخ HTTP است که یک نمونه در ادامه آمده است:

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-
encoding">
  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>
</soap:Envelope>
```

با توجه به اهمیت استاندارد ها، در این بخش ما در خصوص استانداردهایی صحبت کردیم که در زمان کار با ابر می‌توان آنها را بکار گرفت. همچنین نحوه عملکرد آنها نیز مورد بررسی قرار گرفت.

پرسش‌های مروری فصل ۷
۱- HTTP چیست؟
۲- XMPP چه تفاوتی با HTTP دارد؟
۳- Polling به چه معناست؟
۴- OpenID چیست؟ چه کاربردهایی دارد؟
۵- فرآیند صدور گواهینامه SSL به چه صورت است؟
۶- CSR چیست و شامل چه اطلاعاتی است؟
۷- XML چه ویژگی‌هایی دارد؟ تفاوت آن با HTML چیست؟

تحقیق و پژوهشی فصل ۷
۱- نحوه عملکرد OpenID را تشریح کنید. چند نمونه از سایت‌هایی را که از آن استفاده می‌کنند را نام ببرید.
۲- چه پروژه های دیگری مشابه OpenID وجود دارند که امکان دسترسی را به یک سرویس خاص فراهم می‌کنند؟
۳- PCI DSS شامل چه مواردی است؟
۴- نحوه عملکرد و معماری گروتکل SOAP و REST را بررسی و با هم مقایسه کنید.
۵- در خصوص Interoperability (قابلیت همکاری) و وضعیت توسعه آن تحقیق کنید.
۶- استانداردهای مورد نیاز در لایه‌های زیرساخت، پلت‌فرم و سرویس را با هم مقایسه کنید.
۷- در خصوص opencloud و استانداردهای مورد استفاده در آن تحقیق کنید.

مراجع

[1] Anthony T. Velte, Toby J. Velte, Ph.D. Robert Elsenpeter, "Cloud Computing: A Practical Approach," McGraw Hill, 2010

[2] Polling: *http://en.wikipedia.org/wiki/Polling_%28computer_science%29, accessed May 2010*

[3] Don't Use Polling for Real-time Feeds: *http://highscalability.com/blog/2010/1/11/strategy-dont-use-polling-for-real-time-feeds.html, accessed May 2010*

[4] Web Standards and Toturials: *http://www.w3schools.com, accessed May 2010*

[5] Cloud Computing Interoperability Forum, *cloudforum.org*

[6] The Open Cloud Consortium, *opencloudconsortium.org*

[7] Cloud Interoperability Magazine, *cloudinterop.ulitzer.com*

[8] Thrift - A software framework for scalable cross-language services development, *http://incubator.apache.org/thrift*

[9] OpenID Foundation website, *http://openid.net*

[10] Why should I use OpenID?, *http://openidexplained.com*

فصل ۸-مقیاس‌پذیری زیرساخت ابری

- مقدمه ای بر مقیاس‌پذیری زیرساخت ابری
- برنامه‌ریزی ظرفیت
- تقاضای مورد انتظار
- تعیین تقاضای مورد انتظار
- تحلیل موارد غیرقابل انتظار
- تاثیر بار
- نقاط مقیاس‌پذیری
- مقیاس ابر
- مقیاس‌پذیری پویا
- مقیاس‌پذیری پیش فعال
- مقیاس‌پذیری واکنشی
- سیستم‌ها و ابزارهای مانیتورینگ
- مدیریت مقیاس‌پذیری پیش فعالانه
- مدیریت مقیاس‌پذیری واکنشی
- مقیاس‌پذیری عمودی

۸-۱- مقدمه ای بر مقیاس‌پذیری زیرساخت ابری

یکی از پرکاربردترین ویژگی‌های زیرساخت‌های ابری، توانایی مقیاس‌پذیری خودکار آنها هم بصورت افقی و هم بصورت عمودی می‌باشد. این مقیاس‌پذیری در حالی است که بر روی برنامه در حال اجرا بدون تاثیر بوده یا تاثیر بسیار کمی دارد. این ویژگی بطور اساسی، رویکرد مدیران IT را در خصوص زیرساخت‌های فناوری اطلاعات خود و نیز مدیران مالی را نسبت به سرمایه گذاری بر روی IT تغییر داده است. اما این ویژگی یک شمشیر دو لبه است.

مزیت واضح مقیاس‌پذیری ابر این است که شما تنها برای منابعی که از آنها استفاده می‌کنید، پرداخت انجام می دهید و بنابراین نیازمند برنامه‌ریزی دقیق در خصوص ظرفیت و منابع مورد نیاز می‌باشد. رویکرد غیر ابری این است که زیرساخت را با توجه به پیک مصرف خریداری کنیم که باعث هدر رفتن منابع می‌شود. اما اشکال ابر این است که معمار سیستم، برنامه‌ریزی منابع را ممکن است نادیده بگیرد. علاوه بر این، اتکای بیش از حد به مقیاس‌پذیری ابر می‌تواند منجر به این شود که سازمان به تقاضا پاسخ دهد صرف نظر از اینکه این تقاضا شاید فایده تجاری برای سازمان نداشته باشد.

در این قسمت در خصوص موضوعاتی بحث می‌کنیم که در مقیاس‌پذیری زیرساخت ابری مطرح است و بنابراین شما می‌توانید بصورت هوشمند، بدون اینکه در خطر بیفتید تمام توانایی‌های ابر را بر روی زیرساخت IT خود بکار بگیرید. موفقیت شما با برنامه‌ریزی منطقی ظرفیت شروع خواهد شد.

۸-۱-۱- برنامه‌ریزی ظرفیت

برنامه‌ریزی ظرفیت به معنای ایجاد یک استراتژی است که تضمین کند زیرساخت شما می‌توانید از تقاضای منابع که بر آن اعمال می‌شود پشتیبانی کند. بحث در خصوص پیچیدگی برنامه‌ریزی خود به اندازه یک کتاب کامل است. این موضوع بطور کامل در کتابی از John Allspaw بنام The Art of Capacity Planning[1] که توسط O'Reilly منتشر شده است، بررسی گردیده است. در اینجا ما مفاهیم اصلی که برای مقیاس‌پذیری مورد نیاز است بررسی می‌کنیم:

- از الگوی مصرفی خود که چگونه در طول روز ، یا در طول یک هفته، در روزهای تعطیل و نیز در فصول مختلف تغییر می‌کند مطلع باشید.
- بدانید که چگونه برنامه‌های شما به بار کاری خود پاسخ می‌دهند بطوری که بتوانید تعیین کنید چه زمانی و چه نوع از ظرفیت اضافی را نیاز خواهید داشت.
- از ارزش سیستم‌های خود برای کسب و کار آگاه باشید، به این ترتیب خواهید دانست که چه موقع افزودن ظرفیت، ارزش ایجاد می‌کند و چه موقع فاقد ارزش است.

[1] http://oreilly.com/catalog/9780596518578/index.html

بعضی ها به توانایی محیط‌های ابری در مقیاس‌پذیری اتوماتیک بر اساس تقاضا نگاه می‌کنند و فکر می‌کنند که دیگر نیاز نیست در برنامه‌ریزی ظرفیت درگیر شوند. بعضی دیگر به برنامه‌ریزی منابع نگاه می‌کنند و به فکر ده ها یا هزاران دلار جهت هزینه‌های مشاوره آن می افتند. هر دو این تفکر ها، باورهای خطرناکی هستند که باید کنار گذاشته شوند.

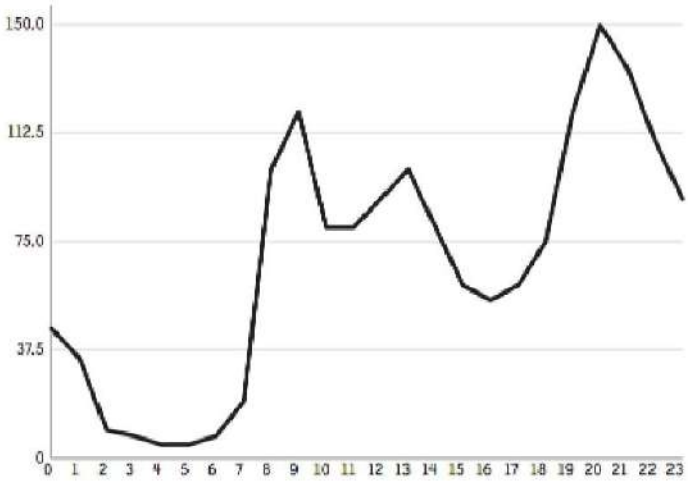
برنامه‌ریزی ظرفیت همانقدر در ابر اهمیت دارد که در یک زیرساخت فیزیکی اهمیت داشته است و نیاز نیست که در برخی پروژه های برنامه‌ریزی ظرفیت عجیب درگیر شوید تا بتوانید طرح مناسب خود را ایجاد کنید. در نهایت، هدف شما این است که به سادگی بدانید وقتی هزینه بیشتری را با افزایش مقیاس زیرساخت خود متحمل می شوید، آن هزینه اضافی از اهداف شما پشتیبانی می‌کند یا نه.

۸-۱-۲- تقاضای مورد انتظار

مطمئناً شما لازم است بدانید که چه تقاضاهایی انتظار دارید که به برنامه شما ارسال شود. حتما لازم نیست که این پیش بینی خیلی دقیق باشد که مثلا تعداد صفحاتی که روزانه از سایت شما نمایش داده می‌شود را بدانید. حداقل کافیست که مقدار مورد انتظار خود را تا حدی بیان کنید که قادر باشید موارد زیر را انجام دهید:

- یک زیرساخت برای پشتیبانی از بار مورد انتظار طراحی کنید.
- تشخیص دهید که چه موقع بار واقعی به بار مورد انتظار همگرا می‌شود.
- تاثیر تغییر نیازمندی های برنامه کاربردی را بر روی زیرساخت خود بدانید.

واضح‌ترین ارزش پیش‌بینی تقاضا –در صورتیکه بدانید چگونه سیستم به بار ورودی پاسخ می‌دهد- این است که خواهید دانست چه تعداد سرور و از چه نوعی نیاز دارید و آن سرورها چه منابعی نیاز دارند. اگر شما ایده ای در خصوص تعداد افرادی که از برنامه کاربردی وب شما استفاده می‌کنند نداشته باشید، مثل این خواند بود که ایده ای در خصوص اینکه زیرساختی که فراهم کرده‌اید، درست کار خواهد کرد یا نه نخواهید داشت. ممکن است در عرض یک ساعت پس از استقرار از کار بیفتد یا اینکه مقدار زیادی پول را بخاطر زیرساخت غیر ضروری هدر بدهید. احتمالا نمی‌توان انتظار داشت که شما بتوانید آینده را به خوبی پیش‌بینی‌کنید. هدف برنامه‌ریزی تقاضا نیز این نیست که نقاط اوج تقاضای غیر قابل انتظار را حذف کند، چرا که همیشه چنین وضعیت های ناخواسته ای وجود دارد. هدف برنامه‌ریزی ظرفیت این است که به شما کمک کند تا در خصوص موارد مورد انتظار برنامه‌ریزی کنید، موارد غیرقابل انتظار را شناسایی کنید و عکس‌العمل مناسب را برای برطرف کردن آن انجام دهید.



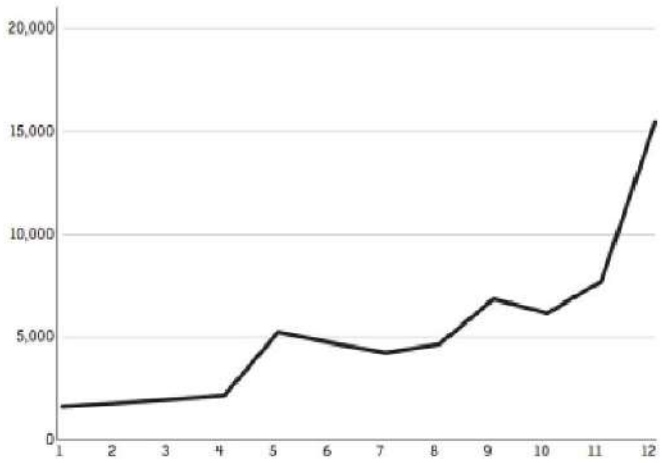
شکل ۸-۱- نمایی از وضعیت بار روزانه یک سایت تجارت الکترونیک

برای مثال سناریویی را در نظر داشته باشید که در آن زیرساختی دارید که از ۱۰ میلیون تراکنش در ثانیه پشتیبانی می‌کند و شما رشدی با میانگین ۱ تا ۵ میلیون تراکنش در ثانیه دارید. اگر شما به درستی این بار را پیش بینی کرده باشید، خواهید دانست که آیا این موج ناگهانی قابل انتظار است (که در این صورت نگرانی وجود ندارد) یا اینکه غیرقابل انتظار بوده است. بدون تخمین صحیح، شما نخواهید دانست که با تغییرات چه باید بکنید.

۸-۱-۳- تعیین تقاضای مورد انتظار

شکل ۸-۱ و شکل ۸-۲ نمودارهایی هستند که ترافیک مورد انتظار را برای یک سایت تجارت الکترونیک در یک دوره یک روزه و نیز نقاط اوج تقاضا برای ۱۲ ماه بعد نشان می‌دهند. نمودار روزانه، نقاط اوج تقاضا را در صبح، عصر و اوایل شب نشان می‌دهد. بعنوان یک شرکت در حال رشد، و با در نظر داشتن اینکه دو محصول در ماه های می و سپتامبر ارائه می‌شوند، ما انتظار داریم که حجم تقاضا به تدریج در طول سال افزایش پیدا کند. نهایتا اینکه ما یک افزایش خرید فصلی در پایان سال داریم.

ما چگونه این اعداد را بدست آورده ایم؟ همانند خیلی چیزهای دیگر، این موضوع به نوع کاربرد بستگی دارد. در مورد نمودار روزانه، الگوهای تاریخی مبنای تشکیل آن هستند. اگر شما یک کسب و کار استوار دارید، الگوهای تاریخی انتظارات شما هستند. در هر صورت یک کسب و کار در حال رشد، تغییر رفتار را در تغییر بازار خواهد دید.



شکل ۸-۲- بار مورد انتظار بر روی سایت تجارت الکترونیک در ۱۲ ماه بعد

یک موقعیت چالش برانگیز دیگر حالتی است که در خصوص آن داده تاریخی در اختیار ندارید. اگر شما بازار خود را بشناسید، می‌تواند بهترین راه برای پیش بینی وضعیت در این موارد باشد. در یک سطح سالیانه، بیشتر نمودارهای شما بر اساس اطلاعاتی است که از دیگر بخش‌های کسب و کار بدست آمده است. اگر شما کالاهایی را به مشتریان می‌فروشید، می‌توانید تغییرات فصلی را خصوصا در پایان سال پیش‌بینی کنید. برای اینکه بتوانید اطلاعات را بهتر تحلیل کنید، لازم است که با بخش‌های مختلف فروش، بازاریابی، تولید و دیگر بخش‌هایی که ترافیک را به سمت سایت شما هدایت می‌کنند ارتباط نزدیکی داشته باشید. مثلا همانطور که در شکل ۸-۲ نشان داده شده است، لازم است که در خصوص محصولاتی که قرار است در آینده ارائه شوند نیز اطلاع داشته باشید.

۸-۱-۴- تحلیل موارد غیرقابل انتظار

در برخی موارد شرکت محصولی را ارائه می هد که منجر به ترافیکی بیش از حد انتظار می‌شود یا اینکه ممکن است حالتی پیش بیاید که در انجام تخمین ها اشتباه کرده باشید و .... به هر حال هر زمان که با ترافیک غیرقابل انتظاری مواجه شدید، مهم است که علت آن را بتوانید تشخیص دهید. در ادامه خواهیم دید که ترافیک غیر منتظره می‌تواند هم خوب باشد و هم بد.

۸-۱-۵- تاثیر بار

توانایی مقیاس‌پذیری یک برنامه وب یا سایت مستقیما به این بستگی دارد که بدانیم محدودیت‌های منابع در کجا قرار دارد و افزودن منابع مختلف چه تاثیری بر روی برنامه کاربردی دارد. مثلا اینکه همیشه اضافه

کردن یک سرور دیگر نمی‌تواند مشکل کارآیی را حل کند. افزودن یک سرور کاربردی به یک زیرساخت مشتمل بر سرور پایگاه‌داده با I/O زیاد، تنها مشکل را بدتر می‌کند. بنابراین باید با شناخت الگوهای استفاده، تست های مختلف را برای بررسی و شناسایی مسائل موجود در زیرساخت انجام داد.

با فرض اینکه یک برنامه وب و پایگاه‌داده کارآمد داشته باشیم، برنامه وبی که روی ابر مستقر می‌شود همه محدودیت‌های ظرفیت زیر را بصورت بالقوه دارد:

- پهنای باند تعدیل‌کننده‌بار
- CPU و RAM در تعدیل‌کننده‌بار.
- توانایی تعدیل‌کننده‌بار در پخش صحیح بار بین سرورهای کاربردی.
- پهنای باند بین تعدیل‌کننده‌بار و سرورهای کاربردی.
- CPU و RAM سرور برنامه کاربردی.
- عملیات I/O برای خواندن دیسک در سرور کاربردی.
- عملیات I/O برای نوشتن بر روی دیسک در سرور کاربردی.
- پهنای باند بین سرور برنامه کاربردی و تجهیزات ذخیره‌سازی شبکه (نظیر دیسک های ذخیره‌سازی block storage در آمازون).
- پهنای باند بین سرور کاربردی و سرور پایگاه‌داده.
- عملیات I/O دیسک برای خواندن پایگاه‌داده.
- عملیات I/O دیسک برای نوشتن بر روی پایگاه‌داده.
- میزان فضای دیسک برای پشتیبانی از ذخیره‌سازی.

موارد زیادی وجود دارد که یک برنامه ممکن است با کمبود ظرفیت مواجه شود. البته باید در نظر داشت که این‌ها فقط محدودیت هستند و مشکلات اصلی بیشتر به خود طراحی بر می گردد. معماری برنامه و پایگاه‌داده قبلا مورد بحث قرار گرفت. بطور خلاصه برای مقیاس‌پذیری افقی یک برنامه بهتر است از قواعد زیر پیروی کنید:

- استفاده از سریع ترین دستگاه ذخیره‌سازی موجود برای پایگاه‌داده.
- اجتناب از نگهداری داده‌های تراکنش در لایه سرور کاربردی.
- امکان ایجاد چند کپی از سرور کاربردی برای اجرا با یک پایگاه‌داده یکسان بدون هر گونه ارتباط بین سرورهای برنامه کاربردی.
- ایندکس‌گذاری صحیح پایگاه‌داده.
- در صورت امکان استفاده از تنظیمات master/slave با عملیات خواندن هدایت شده به slave ها

- افزونگی های خود را بگونه‌ای طراحی کنید که ترافیک بین نواحی دسترسی به حداقل برسد.

۸-۱-۶- نقاط مقیاس‌پذیری

بسته به برنامه شما، رایج ترین نقاط فشار اولیه یکی از سه مورد زیر می‌باشد:

- CPU در سرور برنامه کاربردی.
- RAM در سرور برنامه کاربردی.
- I/O در سرور برنامه کاربردی.

هر برنامه ای نقطه فشار دارد. اگر نداشته باشد، می‌تواند در یک سیستم ساده Intel 386 با بار نامحدود کار کند. مثلا برای برنامه Valtira (یک برنامه مبتنی بر جاوا که برای شرکتی با همین نام طراحی شده بود)، اولین گلوگاه همیشه CPU بود. هنگامیکه بصورت افقی توسعه داده می‌شود این مشکل حل می‌شد، اما بسته به محتوایی که در استقرار Valtira وجود داشت، ما با گلوگاه های جدیدی هم در پهنای باند شبکه و هم I/O دیسک پایگاه‌داده مواجه شدیم. در ابر نیز این گلوگاه I/O است. بنابراین نقطه مقیاس‌پذیری بعد ما، شکستن عملیات خواندن بین slave های پایگاه‌داده و استفاده از master تنها برای عملیات نوشتن بود. نقطه انسداد دیگر، عملیات I/O دیسک در زمان نوشتن در پایگاه‌داده master است. در آن نقطه، ما نیاز داریم که پایگاه‌داده را تقسیم کنیم یا دنبال یک راه‌حل پایگاه‌داده گرانتر باشیم. به عبارت دیگر، ما احاطه کاملی بر روی عملکرد برنامه خود، محدودیت‌های آن و چگونگی عملکرد در مقابل محدودیت‌ها داریم. بدون آن دانش، ممکن است در تشخیص راه‌حل گمراه شویم و راه‌حل را در افزودن سرور برنامه کاربردی ببینیم.
البته شما می‌توانید تحلیل‌های پرهزینه‌ای را برای تعیین تعداد دقیق کاربرانی که برای هر عمل مقیاس‌پذیری نیاز دارید، انجام دهید. ولی واقعا این کار ضرورتی ندارد. فقط برنامه خود را با توجه به شرایط محیطی مورد انتظار اجرا کنید، تست های بار در حال اجرا انجام دهید و به تاثیر بار و افزایش مقیاس دقت کنید. این کار زمان کمتر و هزینه بسیار کمتری دارد و اطلاعاتی که بدست می آورید به اندازه کافی خوب خواهد بود.

۸-۱-۷- مقیاس ابر

ابر به شما این قدرت را می‌دهد که بتوانید منابع محاسباتی خود را تغییر دهید تا با نیازمندی های حجم بار متناسب شود. شما می‌توانید ظرفیت را هم بصورت دستی (با اجرا یک دستور در خط فرمان یا از طریق واسط وب) و یا بصورت برنامه‌نویسی (از طریق تغییرات از پیش تعریف شده در ظرفیت یا از طریق نرم‌افزاری که بطور خودکار ظرفیت را تنظیم کند تا متناسب با تقاضای واقعی شود) تغییر دهید.
امکان تنظیم دستی ظرفیت یک مزیت عمده نسبت به محاسبات سنتی است. اما قدرت واقعی مقیاس‌پذیری در ابر بخاطر مقیاس‌پذیری پویای آن است.

۸-۱-۸- مقیاس‌پذیری پویا

این اصطلاح –که اغلب که آن را مقیاس‌پذیری ابری نیز می گوییم- به نرم‌افزار امکان تنظیم منابع زیرساخت را بدون هیچ گونه تعاملی فراهم می‌کند. مقیاس‌پذیری پویا می‌تواند بصورت پیش فعال[1] یا واکنشی[2] باشد.

۸-۱-۹- مقیاس‌پذیری پیش فعال

این مورد شامل یک زمان بندی برای تغییر زیرساخت بر اساس نمودارهای تقاضاهای پیش بینی شده است. اگر شما برنامه را متناسب با شکل ۸-۱ در نظر بگیرید، می‌توانیم ابزارهای مدیریتی ابر را بگونه‌ای پیکربندی کنیم که با حداقل زیرساخت موجود بتواند نیازمندی های موجود در طول ساعات اولیه صبح را جواب بدهد و سپس به ظرفیت اضافه کنیم، دوباره ظرفیت را تا ظهر مقداری کاهش دهیم و .... این استراتژی منتظر افزایش تقاضا نمی‌شود، در عوض بر اساس یک برنامه مشخص کار می‌کند.

۸-۱-۱۰- مقیاس‌پذیری واکنشی

در این استراتژی، زیرساخت شما متناسب با تغییرات تقاضا با افزودن یا کاهش ظرفیت واکنش نشان می‌دهد.

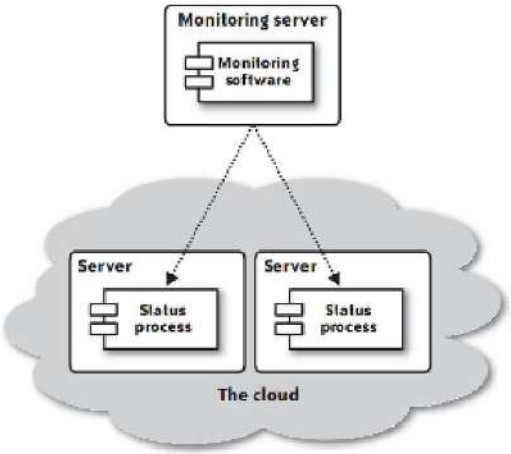
۸-۱-۱۱- سیستم‌ها و ابزارهای مانیتورینگ

ابزارهای مدیریت و مانیتورینگ در زیرساخت ابر، برای مدیریت آن بسیار حیاتی هستند. نمونه های بسیاری از این ابزارها نظیر enStratus، RightScale و Morph وجود دارد که می‌توان بر اساس هزینه، نوع برنامه ای که باید مدیریت شود و اجزای زیرساخت از آن ها استفاده کرد.
هر ابزاری که انتخاب کنید باید حداقل دارای قابلیت‌های زیر باشد:

- زمان بندی تغییرات در ظرفیت برای استقرار برنامه‌های کاربردی.
- مانیتورکردن وضعیت برای تقاضاهای خیلی زیاد یا کمتر از حد عادی.
- تنظیم ظرفیت بطور خودکار بر اساس موج‌های درخواست غیرقابل‌انتظار یا کاهش ناگهانی تقاضا.

مانیتورینگ تنها بررسی ظرفیت نیست، بلکه باید خطاها و خرابی ها را نیز پیدا کرده و آنها را بازیابی کند. به این ترتیب یک سیستم اطلاع رسانی[3] بخش مهمی از مانیتورینگ خرابی ها است. شما باید گزارش کاملی از هر تغییری در ظرفیت داشته باشید و هر چیز غیرعادی مثلا با ایمیل به شما اطلاع داده شود.

---
[1] proactive
[2] reactive
[3] Notification System



شکل ۸-۳-معماری کلی استفاده از یک سیستم مانیتورینگ برای بررسی سلامت ابر

معماری یک سیستم مانیتورینگ در شکل ۸-۳ نشان داده شده است. در مقایسه با بازیابی از سوانح[1]، لازم نیست که برای کارهایی نظیر برنامه‌ریزی ظرفیت، سرور مانیتورینگ خارج از ابر باشد. مانیتورینگ پهنای باند نیز یکی از موارد مهم و ضروری می‌باشد. با مانیتورینگ می‌تواند وضعیت هر یک از سرورها را بررسی کرد تا تصویری از وضعیت فعلی منابع در اختیار داشته باشید. هر نمونه ابری، پردازش‌هایی دارد که می‌تواند علائم حیاتی آن را به سرور مانیتورینگ گزارش کند.

برای مقاصد امنیتی، بهتر است که سرور مانیتورینگ خودش وضعیت نمونه ها را بپرسد، تا اینکه نمونه آن را گزارش کنند. به این ترتیب می‌توانید سرور مانیتورینگ را پشت دیوار آتش قرار دهید تا امکان ورود ترافیک به سمت سرور مانیتورینگ وجود نداشته باشد. همچنین مهم است که در نظر داشته باشید، سرور مانیتورینگ باید متناسب با افزایش تعداد گره ها، امکان مقیاس‌پذیری داشته باشد.

فرآیندی که حیات نمونه ها را چک می‌کند باید برای هر نمونه بر اساس عملکرد آن متفاوت باشد. یک تعدیل‌کننده‌بار بطور مداوم باید از نظر وضعیت CPU و RAM بررسی شود. یک سرور پایگاه‌داده باید از نظر وضعیت عملکرد I/O نیز بررسی شود. مشکل ترین مانیتورینگ مربوط به سرور برنامه کاربردی است. در این مورد نه تنها باید قادر باشید بررسی کنید که چقدر از منابع استفاده شده است، بلکه چگونگی فعالیت آن را نیز بدانید.

[1] Disaster Recovery

سپس سرور مانیتورینگ از ابزارهای تحلیلی برای بررسی همه داده‌ها استفاده می‌کند تا بداند که چه زمان باید مقیاس را کاهش یا افزایش داد، فعالیت های غیرقابل انتظار را تشخیص دهد و عملکرد مناسب را در پاسخ به فعالیت های غیرمنتظره انجام دهد.

۸-۱-۱۲- مدیریت مقیاس‌پذیری پیش فعالانه

یک سیستم مقیاس‌پذیر پیش فعال شما را قادر می سازد تا تغییر ظرفیت را بگونه‌ای زمان بندی کنید تا با تغییرات مورد انتظار شما بر اساس نیاز برنامه کاربردی تطبیق داشته باشد. وقتی از مقیاس‌پذیری پیش فعال استفاده می‌کنید، باید میزان استاندارد انحراف معیار مورد انتظار را بدانید. برای اینکار نیاز نیست به کتاب های آمار مراجعه کنید بلکه منظور این است که شما باید تقریبا بدانید که ترافیک عادی شما چقدر از انتظارات شما منحرف می‌شود. اگر شما حدود ۱۰۰۰ نمایش صفحه در ساعت را برای ظهر پیش بینی کرده باشید ولی این رقم به ۱۱۰۰ برسد، آیا واقعا غیرمنتظره است؟ شاید نه. ظرفیت شما در هر زمان باید قادر باشد تا جوابگوی حداکثر ظرفیت مورد انتظارتان باشد.

۸-۱-۱۳- مدیریت مقیاس‌پذیری واکنشی

مقیاس‌پذیری واکنشی به شما این امکان را می‌دهد تا به سرعت در مقابل تقاضای غیرقابل انتظار، واکنش نشان دهید. اما اگر شما نتوانید برنامه‌ریزی ظرفیت انجام دهید و در عوض تنها به مقیاس‌پذیری واکنشی برای مدیریت یک برنامه وب تکیه کنید، با مشکل مواجه خواهید شد. مبنای کار این مقیاس‌پذیری، بهره‌وری[1] است. به عبارتی، وقتی منابع حافظه یا پردازش شما به سطح خاصی از مصرف می رسد، منابع بیشتری را به محیط محاسباتی اضافه می‌کنید. منطق مانیتورینگ چنین وضعیتی بسیار ساده است، اما در حقیقت، این کار تنها در برخی از موارد مورد نیاز است. مثال‌هایی از مشکلات این روش در ادامه ذکر شده است.

- افزایش پردازش سرور برنامه کاربردی که به یک سرور پایگاه‌داده I/O bound متصل است. در نتیجه بار روی سرور افزایش می یابد و موقعیت بدتر می‌شود.
- یک حمله که همه منابع خود را مصرف می‌کند. در نتیجه تلاش‌های فزاینده ای از اجرای نمونه های جدید بوجود خواهد آمد و هزینه‌های شما سر به فلک می کشد.
- یک موج ناخواسته در فعالیت وب که به زیرساخت شما تحمیل می‌شود اما بر روی کاربر تاثیر بسیار ناچیزی دارد. در عین حال شما می‌دانید که بار فعالیت کاهش خواهد یافت، اما سیستم مانیتورینگ شما نمونه های بیشتری را با توجه به بار تحمیل شده اجرا می‌کند.

[1] Utilization

یک سیستم مانیتورینگ خوب، ابزارهایی را فراهم می‌کند که این مشکلات بالقوه را با مقیاس‌پذیری واکنشی کاهش دهد. به ندرت می‌توان سیستمی را دید که بتواند به شکل ایده عالی قادر به انجام سناریوی آخر باشد. این مورد نیاز مند شناخت محدوده مشکل و الگوهای فعالیتی است که نشان بدهد اجرای نمونه جدید ضرورتی ندارد.

به هر حال، قوانین سیستم مانیتورینگ شما برای مقیاس‌پذیری واکنشی تعریف شده است و باید همیشه یک ناظر روی آن داشته باشید. این ناظر می‌تواند تعداد نمونه هایی که سیستم مانیتورینگ می‌تواند بطور خودکار اجرا کند و نیز میزان هزینه‌ای که از طرف شما بتواند خرج کند را محدود کند. همچنین در حالتی که حمله ای بر روی زیرساخت شما صورت بگیرد، سیستم شما را باید به شکل مناسبی متوقف کند تا از هزینه کردن بیش از حد و اجرای نمونه های زیاد جلوگیری شود.

۸-۱-۱۴- مقیاس‌پذیری عمودی

تا اینجا بر روی مقیاس‌پذیری افقی صحبت کردیم که به معنای اضافه کردن سرورهای جدید بود. مقیاس‌پذیری عمودی، به معنای جابجا کردن سرور موجود با یک نمونه دیگر یا تغییر مشخصات سخت‌افزاری آن است. همه محیط‌های مجازی و خصوصا محیط‌های ابری در مقیاس‌پذیری افقی خیلی خوب عمل می‌کنند. اما وقتی میخواهیم مقیاس‌پذیری عمودی انجام دهیم، ابر دارای تعدادی نقاط قوت مهم و تعداد نقاط ضعف مهم است.

قدرت ابر با مقیاس‌پذیری عمودی، سهولت استفاده از پیکربندی‌های مختلف کوچکتر یا بزرگتر می‌باشد. برای مثال آمازون در حال حاضر ۵ انتخاب مختلف با اندازه های مختلف برای شما فراهم کرده است. اما اگر یک بخش از برنامه شما نیاز به RAM بیشتر از نمونه های پشتیبانی شده توسط آمازون داشت، شما بدشانس خواهید بود. در مقابل GoGrid می‌تواند درجه بیشتری از سفارشی سازی را انجام دهد که از جمله آن امکان طراحی پیکربندی‌های I/O بیشتر است ولی هیچ کدام از این گزینه‌ها از نظر سفارشی سازی پیکربندی به پای سیستم سفارشی سازی Dell نمی رسد.

وقتی ما با استفاده از مقیاس‌پذیری عمودی سعی می‌کنیم که از نودهای بزرگتر استفاده کنیم، درست است که کارآیی بیشتری بدست می آوریم، اما در حالتی که از نودهای کوچکتر ولی با تعداد بیشتر استفاده می‌کنیم، می‌توانیم همان کارآیی را با دسترس پذیری بیشتری بدست آوریم. البته این مقایسه از نظر هزینه تمام شده نیز قابل توجه است که یک نمونه از آن که مربوط به آمازون است در جدول ۸-۱ نشان داده شده است.

جدول ۸-۱ -گزینه‌های مختلف انتخاب سرور در آمازون

Configuration	Capacity	Cost
Eight Amazon medium	8,000 page views/minute	$0.80/hour
Two Amazon large	10,000 page views/minute	$0.80/hour
One Amazon extra-large	10,000 page views/minute	$0.80/hour

طبق این جدول، با فرض اینکه شما ۸ سرور متوسط اضافه کرده باشید و هزینه مقیاس‌پذیری افقی آنها را بصورت خطی در نظر بگیریم، چنانچه بخواهید ظرفیت را افزایش دهید، می‌توانید بجای استفاده از سرور نهم، از دو سرور بزرگتر استفاده کنید. به این ترتیب اغلب پیش می‌آید که بخاطر مسائل مالی، برنامه را بصورت عمودی بدهیم. نکته این است که مقیاس‌پذیری عمودی یک نمونه خاص از همان مقیاس‌پذیری افقی است که شامل مراحل زیر می‌باشد:

۱- افزودن یک نمونه از سیستم بزرگتر در ابر، همانند مقیاس‌پذیری افقی با این تفاوت که شما نمونه های موجود را تکثیر نمی‌کنید.

۲- انتظار برای اجرای برنامه در سیستم جدید.

۳- حذف نمونه قدیمی و کوچکتر از سیستم

وقتی شما مقیاس‌پذیری افقی و عمودی را کنار هم داشته باشید، زیرساختی خواهید داشت که می‌توانید بهترین استفاده را از منابع محاسباتی آن ببرید.

پرسش‌های مروری فصل ۸
۱- برنامه‌ریزی ظرفیت به چه معناست؟
۲- تقاضای مورد انتظار چه ویژگی‌هایی دارد؟
۳- چگونه می‌توان تقاضای مورد انتظار را برآورده کرد؟
۴- محدودیت‌های مختلف یک برنامه وب را نام ببرید. چه راخ حلی برای هر کدام وجود دارد؟
۵- نقاط مقیاس پذیر برنامه‌های کاربردی کدامند؟
۶- مقیاس‌پذیری پیش فعال و واکنشی را تعریف کنید.

تحقیق و پژوهشی فصل ۸
۱- هر کدام از روش‌های مقیاس‌پذیری در کدام یک از سیستم‌های محاسباتی و به چه شکل قابل استفاده است؟ (سوپر کامپیوترها، کلاسترها، محاسبات توری، داوطلبانه و ابری)
۲- یک نمونه برنامه ابری را انتخاب نموده و پس از شناسایی محدودیت‌های بالقوه آن، ابعاد مختلف مقیاس پذیری آن را تحلیل کنید.
۳- مقیاس پذیری در لایه های مختلف زیرساخت، پلت فرم و سرویس را با هم مقایسه کنید.
۴- چند نمونه از ابزارهای مانیتورینگ را مورد بررسی قرار دهید و با هم مقایسه کنید.
۵- معماری یک برنامه مقیاس پذیر را مورد بررسی قرار دهید و نحوه پیاده سازی آن را بررسی کنید. چه ملاحظات و استراتژی‌هایی می‌توان در آن در نظر گرفت؟

مراجع

[1] George Reese, "Cloud Application Architectures," O'Reilly, 2009

[2] John Allspaw, "TheArt of Capacity Planning," O'Reilly, 2008

فصل ۹- امنیت در رایانش ابری

- مقدمه ای بر امنیت در توده‌های ابر
- فواید امنیت برای رایانش ابری
- تشخیص ریسک
- ریسک‌های خط و مشی و سازماندهی
- ریسک‌های فنی
- ریسک‌های قانونی
- ریسک‌های عمومی (نه فقط مربوط به ابر)
- آسیب‌پذیری‌ها
- دارایی ها
- چارچوب تضمین اطلاعات
- نیازمندی های تضمین اطلاعات
- توصیه های قانونی

۹-۱- مقدمه ای بر امنیت در توده‌های ابر

سازمان‌ها به منظور بکارگیری خدمات ابری نیاز به تضمین امنیت کافی دارند تا دارایی‌های اطلاعاتی سازمانشان به مخاطره نیفتد. تا پیش از این ما امنیت را با استفاده از مکانیسم‌ها و سیاست‌های امنیتی مختلف برای سازمان خود فراهم می‌آوردیم و دارایی‌های سازمان خود را در پشت دیواره‌های آتش، DMZ، VPN و ... پنهان می‌کردیم. حال با ظهور رایانش ابری، نه تنها داده‌های شخصی و خصوصاً سازمانی ما در توده ابری قرار دارد که حتی محل آن را هم نمی‌دانیم، بلکه برنامه‌های کاربردی و ابزارهای مختلفی که از آنها استفاده می‌کنیم نیز در وضعیتی قرار دارند که نمی‌دانیم واقعاً چه کدی را بر روی داده‌های ما اجرا می‌کنند. مثلا اینکه بسیاری از برنامه‌های کاربردی که در سکوی SalesForce قابل اجرا هستند توسط شرکت‌ها ی دیگر تهیه می‌شود. بیش از ۸۰۰ برنامه نوشته شده توسط شرکت‌ها ی ثالث توسط حدود ۴۰۰۰۰ مشتری استفاده می‌شود. این برنامه‌ها بر روی سکوی SalesForce اجرا می‌شوند ولی نمی‌توان از محتوای آنها اطمینان داشت. به این ترتیب هیچ کنترلی بر روی آنچه که اتفاق می‌افتد نداریم و عملا کل دارایی اطلاعاتی خود را از کنترل خود خارج کرده ایم. از سوی دیگر ماهیت خدمات ابری به‌گونه‌ای است که از طریق مرورگر ارائه می‌شوند. بنابراین تهدیدات رایجی که تا کنون در خصوص مرورگرها وجود داشته است، رایانش ابری را نیز براحتی مورد هدف قرار داده‌است. در جدول ۹-۱، مقایسه ای بین وضعیت آنچه که سازمان‌ها در ابر بدست می‌آورند با حالت سنتی انجام شده است.

جدول ۹-۱- مقایسه خدمات ابری و خدمات معمولی

خدمات روی زمین	خدمات روی ابر
سیستم در پشت دیوارآتش، NAT و دیگر gatewayها قرار دارد.	سیستم کاملاً قابل مشاهده است و از هرجا قابل دسترسی است.
نفوذگر باید مهارت زیادی برای آگاهی از وضعیت داخلی بکار گیرد.	مانند ترسیم یک تصویر بزرگ در مقابل نفوذگر می‌باشد.
پایگاه‌داده مشتمل بر اطلاعات یک سازمان است. با نفوذ به یک سایت، تنها سیستم‌های یک سازمان به مخاطره می‌افتد.	شامل داده‌های هزاران سازمان و مشتری مختلف است. با نفوذ به یک سایت، سیستم‌های چندین شرکت به خطر می‌افتد.
کاربردها مبتنی بر وب و یا مبتنی بر سیستم‌عامل ارائه می‌شود.	همه کاربردهای آن مبتنی بر وب است.
امکان اعمال رویه‌های نظارتی دقیق وجود دارد.	امکان نظارت دقیق بر اتفاقات درون ابر وجود ندارد.
سطح امنیت اکثر سازمان‌ها برای کاربردهای تحت وب کم است.	امکان فراهم آوردن سطح امنیت بیشتری وجود دارد.

معمولا ارائه دهندگان سرویس، ادعا می‌کنند که با توجه به SLA[1] تضمین کافی برای رفع مشکلات موجود فراهم می‌شود، اما SLA نیز به تنهایی کافی نیست. اینکه چه تضمینی برای رعایت خود SLA وجود دارد نیز مسئله دیگری است. در جدول ۹-۲ نمونه‌ای از تهدیدات شناخته شده در خصوص بکارگیری توده‌های ابر آورده شده است و در جدول۹-۳ موضوعات امنیتی مطرح در رایانش ابری که اخیرا توسط موسسهٔ Gartner گزارش شده است[۱]، تشریح گردیده است. نمونه‌ای از حملات صورت گرفته در ابر که تا کنون به وقوع پیوسته است به این شرح می‌باشد

- چندین مورد حمله XSS[2] در خدمات گوگل.
- حمله XSS در بخش نظرسنجی BlogSpot و امکان مشاهدهٔ ایمیل‌های دریافتی در Gmail.
- هک شدن سایت Monster در سال گذشته و به سرقت رفتن اطلاعات میلیون‌ها کاربر.
- درج فیلتر با استفاده از حمله CSRF در حساب کاربری شخص برای ارسال خودکار ایمیل‌های او به یک ایمیل دیگر.
- ترکیب XSS و CSRF[3] و دستکاری در URL برای حمله به Picasa و سرقت تصاویر کاربران.
- بروز مشکل در سیستم ایندکس ZOHO و دسترسی یک کاربر غیرمجاز به اسناد دیگر کاربران.

جدول ۹-۲- برخی از تهدیدات شناخته شده در خصوص استفاده از رایانش ابری

تهدیدات	شرح
قطع شدن سرویس	قطع شدن اتصال با اینترنت یا ارتباط با سرویس‌دهنده و سایر اختلالات مشابه
حملات DoS	مختل کردن یا از کار انداختن خدمات سرویس دهندگان
XSS	قرار دادن کدها و اسکریپتهای مخرب در داخل صفحات وب
CSRF	ترکیبی از حمله XSS و استفاده از URLهای تغییر شکل داده‌شده
سرویس‌های نامعتبر	ارائه سرویس توسط سرویس‌دهندگان نامعتبر
DNS poisoning exploit	ارجاع کاربر به یک سایت تقلبی با ظاهری مشابه سایت اصلی

جدول۹-۳- هفت موضوع امنیتی مطرح در خصوص استفاده از رایانش ابری

موضوعات امنیتی	شرح
دسترسی کاربران مجاز Privileged User Access	با توجه با اینکه اطلاعات حساس مربوط به سازمان در خارج از مرزهای آن، ذخیره شده و یا پردازش می‌شوند، بنابراین اینکه در سمت سرویس‌دهندگان، پرسنلی که مسئول مدیریت

[1] Service Level Agreement
[2] Cross-Side-Scripting
[3] Cross-site request forgery

موضوعات امنیتی	شرح
	سرویس‌ها و یا داده‌ها هستند، چه کسانی هستند و چه کنترلی بر روی آنها وجود دارد بسیار مهم است. دسترسی افرادی نظیر مدیر سیستم که در طرف سرویس‌دهنده نیاز است با سیستم سروکار داشته باشند، چون معمولاً تعهد طولانی مدتی با سازمان ما ندارند، یک موضوع امنیتی مهم است.
رعایت قوانین Regulatory Compliance	حتی در زمانی که از یک سرویس دهنده استفاده می‌شود، نهایتاً این مشتریان هستند که مسئول امنیت و حفظ جامعیت داده‌های خود هستند. بنابراین پیشنهاد می‌شود از سرویس‌دهندگانی که دارای مدارک و گواهینامه‌های معتبر در خصوص امنیت و حسابرسی هستند استفاده شود.
مکان داده‌ها Data Location	در زمان استفاده از یک توده ابری، احتمالاً نمی‌توانید اطلاع دقیقی از محل میزبانی داده‌ها بدست آورید. حتی ممکن است ندانید که داده‌ها در کدام کشور ذخیره شده‌اند. از سرویس‌دهندگان بپرسید که آیا داده‌ها را در مکان‌های خاصی و یا شرایط قضایی و قانونی خاصی ذخیره‌سازی و پردازش می‌کنند و یا اینکه تعهدی در خصوص رعایت نیازمندی‌های مربوط به حفظ حریم خصوصی مشتریان متقبل می‌شوند؟
تفکیک داده Data Segregation	با توجه به اینکه داده‌ها در توده ابر در یک محیط اشتراکی ذخیره می‌شود، چگونه می‌توان از عدم دسترسی مشتریان دیگر به داده‌های خود مطمئن شد؟ استفاده از رمزنگاری می‌تواند تا حدی مؤثر باشد. در این خصوص سرویس‌دهندگان باید بتوانند مدارکی در خصوص الگوهای رمزنگاری طراحی شده و ارزیابی شده توسط متخصصان باتجربه ارائه کنند. همچنین باید در نظر داشت که برخی اشتباهات در رمزنگاری ممکن است باعث شود که داده کاملاً غیرقابل استفاده شود.
بازیابی Recovery	حتی اگر ندانید که داده شما کجا ذخیره می‌شود، یک سرویس‌دهنده باید بتواند پاسخگوی این مسئله باشد که در صورت وقوع یک سانحه، چه اتفاقی بر سر داده‌های شما خواهد آمد. هر راه‌حلی که مبتنی بر توزیع داده[1] در چندین سایت مختلف نباشد، آسیب‌پذیر خواهد بود. از سرویس‌دهنده بپرسید که آیا در صورت وقوع سانحه توانایی این را دارد که یک بازیابی کامل انجام دهد، و اینکه این کار چه مدت طول خواهد کشید.
پشتیبانی و بررسی Investigative Support	در زمان بروز مشکل چگونه می‌توان به داده‌ها دسترسی مستقیم داشت؟ اگرچه در توده‌ی ابر، بررسی بی‌جا و یا فعالیت‌های غیرمجاز ممکن است غیرممکن باشد. اما باید در نظر داشت که از طرفی انجام بررسی‌های ضروری نیز بسیار مشکل خواهد بود. چرا که داده‌های چندین مشتری معمولاً با هم در یکجا قرار دارند و یا در چندین مجموعه میزبان پخش شده‌اند و مدام در حال تغییر هستند. حتی اگر سرویس‌دهنده چنین تعهدی را هم قبول کند، باید مطمئن شوید که تجربه موفقی از پشتیبانی چنین فعالیتی را داشته است یا نه.
دسترسی طولانی مدت Long-Term Viability	شکل ایده‌آل این است که سرویس‌دهندگان هیچ وقت از کار نمی‌افتند. ولی باید بتوان مطمئن شد که اگر چنین اتفاقی رخ دهد، داده‌های ما هنوز هم باقی خواهند ماند. از سرویس‌دهندگان بپرسید که چگونه می‌توان داده را پس‌گرفت و اینکه آیا فرمت داده‌ها به‌گونه‌ای خواهد بود که بتوانید آنها را در یک برنامه جایگزین قرار دهید؟

بیشتر مشکلات به این دلیل رخ می‌دهد که ما نمی‌توانیم داخل ابر را ببینیم. بنابراین نیاز اصلی کاربران برای کاهش ریسک‌هایی که با آن مواجه هستند، مشاهده درون توده ابر است؛ اما با توجه به اینکه این کار عملاً

[1] Replication

بخاطر ماهیت توده ابر غیرممکن است، برخی از راه‌حل‌های ارائه شده برای حل این موضوع به این شرح می‌باشد:

- استفاده از SLA دقیق و وجود ضمانت قابل قبول برای رعایت آن
- وجود تضمین مناسب برای تداوم فعالیت تجاری
- وجود تضمین مناسب برای بازیابی از سوانح[1]
- بیان جزئیات مربوط به سیاست‌های امنیتی و پیاده‌سازی‌های انجام شده
- ارائه اطلاعات کامل زیرساختی در زمان انجام مذاکرات و نیز در هر لحظه از طول دوره ارائه سرویس
- نشان دادن رویه‌ها و سیاست‌ها توسعه نرم‌افزار، سیاست‌های تست امنیتی و سیاست‌های اعلام آسیب‌پذیری
- انجام عملیات تست توسط سرویس‌دهندگان ثالث
- انجام تست‌های نفوذ[2] به‌صورت منظم و دوره‌ای و در دسترس بودن نتایج تست در صورت تقاضا
- و...

البته اینکه آیا فروشنده‌ها این سطح از شفافیت را در خدمات خود ارائه دهند، خود یک موضوع دیگر است.

[1] Disaster Recovery
[2] Intrusion

۹-۲- فواید امنیت برای رایانش ابری

در این بخش که برگرفته از یک تحقیق جامع در خصوص ریسک‌های رایانش ابری می‌باشد، کارهای انجام شده در زمینه ریسک‌های امنیتی ابر و راه های کاهش آنها بررسی شده است[۳]. اینکه درباره فواید گسترده رایانش ابری از نظر اقتصادی، فنی و معماری بحث کنیم بسیار حائز اهمیت می‌باشد. اما در تجربه مستقیمی که توسط متخصصین این حوزه در دنیای واقعی به دست آمده است برای تحلیل مسائل امنیتی رایانش ابری می بایست فواید امنیت در این زمینه نیز مرور شوند. رایانش ابری پتانسیل بالایی برای ارتقای امنیت دارد. در ادامه توضیحاتی درباره راههای کلیدی که رایانش ابری می‌تواند در آنها سهیم باشد آورده شده است.

۹-۲-۱- امنیت و مقیاس‌پذیری

به طور عمومی تمامی انواع معیارهای امنیتی اگر در مقیاس بزرگ پیاده‌سازی شوند بسیار ارزان تر خواهند بود. بنابراین هرچه سرمایه گذاری بهتری در امنیت انجام شود حفاظت بهتری در مقابل خطرات حاصل خواهد شد. این سرمایه گذاری ها شامل انواع معیارهای امنیتی مانند فیلترینگ، مدیریت زمان[۱]، مقاوم‌سازی ماشین مجازی، منابع انسانی و مدیریت و آزمایش آنها، افزونگی سخت‌افزار و نرم‌افزار، احراز هویت بسیار قوی و مدیریت هویت ها می‌باشند. این سرمایه گذاری ها از جهت دیگر موجب ارتقای همکاری ما بین شرکا خواهد شد. دیگر فواید مقیاس عبارتند از:

- **چندین محل**: اغلب فراهم کنندگان ابر منابع اقتصادی برای نسخه برداری از محتواها در چندین محل[۲] را دارا می‌باشند. این امر موجب افزایش افزونگی و همچنین مستقل شدن از خرابی ها شده و یک سطح بازیابی از سوانح[۳] را فراهم می‌کنند.
- **شبکه های لبه**[۴]: ذخیره‌سازی، پردازش و تحویل در نزدیکی لبه های شبکه به این معناست که سرویس‌هایی با کیفیت بیشتر و قابلیت اعتماد بالاتر ارائه می‌شوند. از سوی دیگر این موضوع باعث می‌شود که مشکلات محلی شبکه بر سراسر ابر تاثیر نداشته باشد.
- **پاسخ های بجا به اتفاقات**[۵]: سیستم‌های با مقیاس بزرگتر که به خوبی اجرا می‌شوند می‌توانند قابلیت‌های بهتری را برای پاسخ به اتفاقات ارائه دهند. برای مثال در زمینه شناسایی سریع نرم‌افزارهای بدخواه .

---

[1] Patch management
[2] Replicate
[3] Disaster Recovery
[4] Edge networks
[5] Improve timeliness of response to incidents

- **مدیریت تهدید**[1]: تامین کنندگان ابر می‌توانند از عهده استخدام متخصصین برای مقابله با تهدید های امنیتی بر آیند در حالیکه شرکت‌های کوچک توانایی این کار را ندارند.

۹-۲-۲- امنیت بعنوان یک عامل تمایز در بازار

امنیت با اولویت‌ترین نگرانی مشتریان ابر می‌باشد. مشتریان تصمیمات خرید خود را براساس شهرت در زمینه محرمانگی، صحت و قابلیت ارتجا[2]، و سرویس‌های امنیتی که توسط تامین کننده ارائه می‌شود اتخاذ می‌کنند. حساسیت های مشتریان در زمینه های فوق الذکر در ابر بسیار بیشتر از محیط‌های سنتی می‌باشد. این امر موجب شده است تا رقابت بسیار شدیدی میان تامین کنندگان ابر در زمینه موضوعات امنیتی برقرار باشد.

۹-۲-۳- استاندارد کردن واسطها برای خدمات امنیتی مدیریت شده

تامین کنندگان بزرگ ابر می‌توانند واسطهایی استاندارد برای خدمات امنیتی مدیریت شده[3] به مشتریان خود پیشنهاد دهند. این قابلیت موجب ایجاد بازاری می‌شود که در آن مشتریان می‌توانند ما بین تامین کنندگان مختلف سوییچ کنند و ارزان ترین هزینه‌ها را برای خود انتخاب کنند.

۹-۲-۴- مقیاس کردن سریع و هوشمند منابع

لیستی از منابع ابر که می‌توانند به سرعت تغییر مقیاس داده شوند عبارتند از منبع ذخیره‌سازی، زمان CPU، حافظه، درخواست های سرویس وب و نمونه های ماشین مجازی و سطح دانه بندی کنترل منابع می‌باشند.

یک تامین کننده ابر[4] پتانسیل تخصیص دوباره منابع به صورت پویا به منظور اهدافی چون فیلترینگ، شکل‌دهی به ترافیک[5] ، رمزنگاری و ... را دارد. این قابلیت برای افزایش پشتیبانی دفاعی در برابر حملات احتمالی می‌باشند (مانند حمله DDos). وقتی این قابلیت تخصیص دوباره منابع با متدهای بهینه‌سازی منابع ادغام می‌شود باعث می‌شود که تامین کنندگان ابر قادر به محدود کردن تاثیر حملات مخرب بر منابع باشند. حملاتی که دسترس پذیری منابع ابر را تهدید می‌کنند. رسیدن به این موضوع نیازمند این است که تامین کنندگان تلفیقی از ساختارهای دفاعی، مدیریت منابع و روش های بهینه‌سازی را پیاده‌سازی کنند.

---

[1] Threat management
[2] Resilience
[3] Managed Security Services (MSS)
[4] Cloud Provider
[5] Traffic shaping

۹-۲-۵- حسابرسی و جمع آوری شواهد

در IaaS پشتیبانی از تکثیر[1] ماشین‌های مجازی بر حسب تقاضا وجود دارد. در موارد مشکوک به نفوذ یک مشتری می‌تواند تصویری از یک ماشین مجازی در حال اجرا ایجاد کرده که برای تحلیل‌های بعدی قانونی استفاده شود. با توجه به وجود ذخیره سازها، چندین نسخه از این ماشین‌های مجازی می‌توانند نگهداری شده و سپس به صورت موازی مورد تحلیل قرار بگیرند که این امر موجب کاهش زمان بررسی برای موارد امنیتی می‌شود. این بررسی ها برای کشف اتفاقات امنیتی و ردیابی حمله کنندگان و از بین بردن ضعف های امنیتی انجام می‌شود.

۹-۲-۵-۱-بروز رسانی های دقیق‌تر، موثرتر و بهینه تر بهمراه پیش فرض ها

تصاویر ماشین‌های مجازی و ماژولهای نرم‌افزاری که توسط مشتریان استفاده می‌شوند را می‌توان با استفاده از بروزرسانی‌هایی با آخرین وصله های امنیتی مقاوم‌سازی نمود. همچنین سرویس‌های ابر در IaaS این امکان را بوجود می‌آورند که نمایش لحظه ای از زیرساخت مجازی در حال اجرا گرفته و نگهداری شود تا برای مقایسه با baseline ها مورد استفاده قرار گیرد (مثلا مطمئن شدن از تغییر نکردن قواعد نرم‌افزاری در دیواره‌های آتش) [۴]. در محیط‌های ابری بروزرسانی‌ها با سرعت بسیار بیشتری نسبت به محیط‌های سنتی انجام می‌شود. در محیط‌های سنتی بروزرسانی‌ها با استفاده از وصله‌ها انجام می شد. در مدل‌های PaaS و SaaS برنامه‌های کاربردی از نظر امنیتی بیشتر مقاوم‌سازی می‌شوند تا قابلیت‌های بهتری از نظر قابلیت حمل و مقاومت در مقابل خرابی نسبت به برنامه‌های کاربردی مورد استفاده داشته باشند. همچنین در مدل‌های PaaS و SaaS بروزرسانی‌ها برای کمینه کردن آسیب‌پذیری‌ها به صورت متمرکز انجام می‌شود.

۹-۲-۵-۲- حسابرسی و توافقنامه‌های سطح سرویس

حسابرسی و توافقنامه‌های سطح سرویس موجب مدیریت ریسک بهتر می‌شوند. نیاز برای محدود کردن سناریو های مختلف ریسک در توافقنامه های سطح سرویس و تاثیرات محتمل نفوذهای امنیتی بر شهرت ابر، تامین کنندگان ابر را تشویق به سخت گیری در حسابرسی های داخلی و طریقه تشخیص ریسک می‌کند. اغلب حسابرسی ها برای افشا کردن ریسک‌هایی است که ممکن است خود را نمایان نکنند اما تاثیرات خود را داشته باشند.

۹-۲-۶- فواید تمرکز منابع

در بخش ریسک‌ها در همین بخش گفته خواهد شد که تمرکز منابع خود یک ریسک می‌باشد. با این حال داشتن تمرکز در منابع فوایدی چون دسترسی فیزیکی ارزان تر، کنترل دسترسی فیزیکی و بکاربردن خط و

[1] Cloning

مشی های امنیتی ارزان تر و آسان تر، کنترل بر مدیریت داده، مدیریت وصله ، مدیریت وقایع و پروسه های نگهداری، را دارد.

۹-۲-۷- پروسه تشخیص ریسک

سطح ریسک بر اساس احتمال اتفاق یک سناریو تخمین زده می‌شود و به تاثیر منفی‌ای نگاشت می‌شود. احتمال اتفاق یک سناریو با توجه به احتمال ایجاد خطر به واسطه یک آسیب‌پذیری تعیین می‌شود.
احتمال اتفاق هر سناریو و تاثیر آن بر کسب و کار بر اساس نظر تعدادی از متخصصین در این زمینه تعیین شده است. در مواردی که امکان قضاوت درست و تخمین با احتمال مناسب وجود نداشته است با مقدار N/A مشخص شده است. در بسیاری از موارد تخمین احتمال وقوع وابسته به مدل ابر و معماری مدنظر می‌باشد.
در جدول ۹-۴ سطح ریسک بعنوان یک تابع برای تاثیر بر کسب و کار و احتمال وقوع سناریو نشان داده شده است. ریسک نتیجه شده در مقیاسی از ۰ تا ۸ بیان می‌شود و می‌تواند با معیارهای قبول ریسک سنجیده شود. این مقیاس از ریسک را می‌توان برای سهولت درک به رتبه بندی سه گانه زیر نگاشت کرد:

- ریسک کم: ۰ - ۲
- ریسک متوسط: ۳ – ۵
- ریسک زیاد: ۶ – ۸

جدول ۹-۴ – بر اساس سطوح تخمین ریسک در ISO/IEC 27005:2008 [۵]

	احتمال وقوع سناریو	بسیار پایین (خیلی نامحتمل)	پایین (نامحتمل)	متوسط (ممکن)	بالا (محتمل)	بسیار بالا (مکرر)
تاثیر بر کسب و کار	بسیار پایین	۰	۱	۲	۳	۴
	پایین	۱	۲	۳	۴	۵
	متوسط	۲	۳	۴	۵	۶
	بالا	۳	۴	۵	۶	۷
	بسیار بالا	۴	۵	۶	۷	۸

۹-۳- تشخیص ریسک

۹-۳-۱- سناریو های مورد کاربرد

برای اهداف تشخیص ریسک در رایانش ابری در این بخش سه سناریو تحلیل شده است:

- یک دید SME از رایانش ابری
- تاثیر رایانش ابری بر قابلیت ارتجاع سرویس
- رایانش ابری و eGovernment (برای کاربرد eHealth)

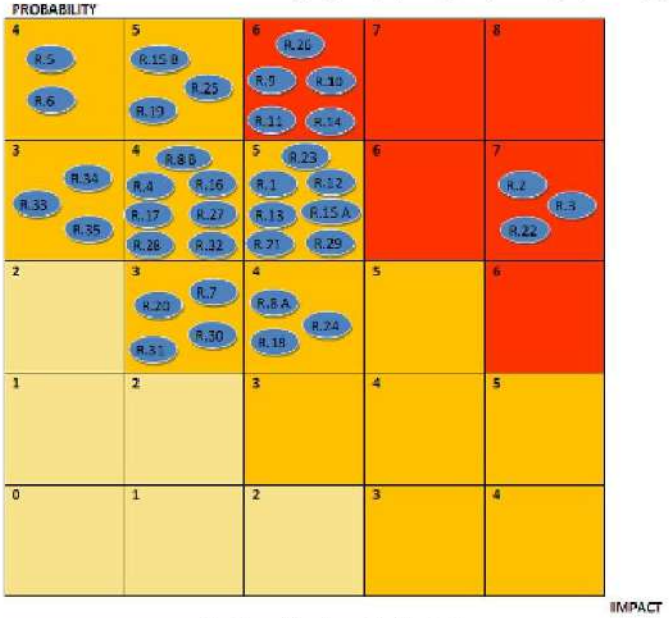
سناریویی که در اینجا بررسی می‌شود برای یک مشتری ابر یا یک تامین کننده خاص و واقعی نمی‌باشد بلکه با این هدف می‌باشد که سناریوای ارائه شود که برای سازمان‌های مختلف وجه اشتراکی داشته باشد.

۹-۳-۲- ریسک‌ها

این نکات در ارتباط با تعریف ریسک می بایست مدنظر قرار بگیرند:

- ریسک همیشه باید در ارتباط با فرصت های کسب و کار و میل به ریسک کردن مطرح شود. برخی اوقات ریسک در عوض فرصت ها بوجود می‌آیند.
- سرویس‌های ابر فقط برای دسترسی آسان به ذخیره‌سازی از وسایل مختلف نمی‌باشد، بلکه شامل فوایدی چون ارتباط آسان میان چندین نقطه همکاری می‌باشد. بنابراین در تحلیل ریسک‌ها علاوه بر ریسک‌های نگهداری داده در مکان‌های مختلف ریسک‌های ارسال یک داده از نقطه ای به نقطه دیگر نیز می بایست مدنظر قرار بگیرند. بنابراین ریسک‌های استفاده از رایانش ابری باید با ریسک‌های استفاده از راه‌حل‌های سنتی مقایسه شوند.
- سطوح مختلف ریسک‌ها با توجه به معماری های مختلف ابر که وجود دارند بسیار متفاوت می‌باشد.
- ممکن است که مشتریان ابر ریسک‌هایی را متوجه تامین کننده ابر کنند که در چنین مواردی ریسک‌ها را باید با توجه به هزینه و سود حاصل مورد بررسی قرار داد. بنابراین تمامی ریسک‌ها قابل مواجهه نمی‌باشند: اگر یک ریسک منجر به شکست کسب و کار شود و صدمات جدی به شهرت کسب و کار وارد کند، برای هیچ کسب و کاری امکان تاوان دادن در مقابل آن وجود نخواهد داشت.
- تحلیل‌های ریسک در این گزارش برای فناوری ابر انجام شده است و مختص هیچ محاسبه ابری خاصی که توسط کمپانی‌ای ارائه شده باشد نمی‌باشد.
- سطح ریسک‌ها از دید مشتریان ابر بیان شده است. در جاهایی که نقطه نظر تامین کننده ابر بیان شده باشد مشخصا ذکر می‌شود.

شکل زیر توزیع احتمال ریسک‌ها و تاثیرات آنها را نشان می‌دهد:



شکل ۹-۱- توزیع احتمال ریسک‌ها

ریسک‌های تعیین شده در مرحله تشخیص ریسک در ۳ دسته قرار می‌گیرند:

- خط و مشی و سازماندهی
- فنی
- قانونی

هر ریسک در جدولی مشتمل بر موارد زیر نشان داده می‌شود:

- سطح احتمال
- سطح تاثیر
- ارجاع به آسیب‌پذیری‌ها
- ارجاع به دارایی های تحت تاثیر واقع شده
- سطح ریسک

برای قابلیت فهم بهتر، خانه‌هایی برای مقایسه بین ریسک‌های رایانش ابری و ریسک‌های استاندارد فناوری اطلاعات در جداول اضافه شده است.

۹-۴- ریسک‌های خط و مشی و سازماندهی

۹-۴-۱-۱- ر ۱. Lock-in

احتمال	بالا	مقایسه ای: بالاتر
تاثیر	متوسط	مقایسه ای: مساوی
آسیب‌پذیری‌ها	آ۱۳. کمبود فناوری‌ها و راه‌حل‌های استاندارد آ۴۶. انتخاب ضعیف تامین کننده آ۴۷. فقدان افزونگی تامین کننده آ۳۱. فقدان تکامل و شفافیت شرایط استفاده	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	بالا	

در حال حاضر تعداد کمی ابزار، داده استاندارد و یا واسطهای سرویس برای تضمین داده و قابلیت حمل خدمات وجود دارند [۶]. برای یک مشتری بسیار سخت می‌باشد که از یک تامین کننده به تامین کننده دیگر جابجا شود یا اینکه بخواهد داده و خدمات را از/به محیط فناوری اطلاعات جابجا کند. از طرف دیگر تامین کنندگان ابر نیز ممکن است انگیزه‌ای برای جلوگیری (مستقیم یا غیر مستقیم) از قابلیت حمل خدمات و داده‌های مشتریانشان داشته باشند.

این استقلال در فراهم کردن خدمت که در هر تامین کننده ابر وجود دارد، با توجه به تعهدات تامین کننده ابر، ممکن است موجب شکست های سنگین در کسب و کار و حتی ورشکستگی تامین کننده ابر شود (ر۵ را ببینید).

اکتساب تامین کننده ابر (ر۶) در صورت افزایش احتمال تغییر در خط و مشی تامین کننده و یا توافقنامه ها مانند شرایط استفاده[1] نیز ممکن است تاثیر مشابهی را داشته باشد.

این نکته بسیار مهم است که بدانید که ماهیت تعطیل شدن با توجه به نوع ابر متفاوت است:

[1] Terms of Use (ToU)

۹-۴-۱-۲- SaaS lock-in

- داده مشتریان در یک شمای پایگاه‌داده طراحی شده توسط تامین کننده SaaS نگهداری می‌شود. اغلب تامین کننده های SaaS از فراخوانی های API برای خواندن رکوردهای داده استفاده می‌کنند. حال اگر تامین کننده ای این قابلیت را به مشتری ارائه ندهد، مشتری نیاز به توسعه برنامه ای خواهد داشت که داده‌هایش را استخراج کرده و در پایگاه‌داده تامین کننده دیگری وارد کند. این نکته قابل توجه است که تعدادی توافقنامه برای ساختار رکوردهای کسب و کار وجود دارد (مثلا ممکن است که رکورد مشتری در تامین کننده SaaS ای دارای فیلدهایی باشد که در تامین کننده دیگر وجود ندارد)، از طرف دیگر ممکن است که فرمت های مختلفی برای وارد کردن و خارج کردن داده وجود داشته باشد مثلا XML که در تامین کننده های مختلف مورد استفاده قرار می‌گیرند. تامین کننده جدید می‌تواند در قبال دریافت مبلغی در انجام این کار کمک لازم را به مشتری ارائه دهد. در صورتیکه روتین های لازم برای این کار توسط تامین کننده فراهم نشود مشتری مجبور به نوشتن این روتین ها می‌باشد. از آنجایی که مشتریان به این جنبه از خدمات قبل از تصمیم گیری برای انتقال به یک تامین کننده توجه می‌کنند، یک تامین کننده ابر با ایجاد راه انقال ساده داده به همراه هزینه مناسب می‌تواند در دراز مدت سود مناسبی را کسب کند.
- lock in برنامه کاربردی از بارزترین انواع lock in می‌باشد (مختص ابر هم نیست). هر تامین کننده SaaS برنامه کاربردی مختص به خود برای مشتریان را ارائه می‌دهد. مشتریان SaaS که دارای حجم زیاد کار بر روی یک ابر هستند ممکن است هزینه زیادی را در انتقال از یک تامین کننده به دیگری پرداخت کنند (مانند هزینه یادگیری مجدد). در صورتی که مشتریان برنامه‌های کاربردی خاص خود را داشته باشند که مستقیما از API ها استفاده کرده باشد هم در صورتی که بخواهند تغییر تامین کننده بدهند دوباره باید API های خود را هم تغییر دهند.

۹-۴-۱-۳- PaaS lock-in

این نوع از lock in هم در لایه API و هم در لایه مولفه اتفاق می افتد. برای نمونه تامین کننده PaaS ممکن است که یک مرکز داده پشتیبان را پیشنهاد دهد. در این حالت مشتری هم باید از کدهای خاص API ارائه شده توسط تامین کننده استفاده کند و هم روتین های دسترسی به مرکز داده پشتیبان را کدنویسی کنند. این کدها لزوما ما بین تامین کننده های PaaS قابل انتقال نمی‌باشد. اگر هم API های آنها سازگار باشند ممکن است که مدل‌های دسترسی داده در آنها متفاوت باشد.

- در لایه API، PaaS lock-in در مواقعی اتفاق می افتد که تامین کننده های مختلف API های متفاوت عرضه کنند.

- در لایه زمان اجرا[1]، PaaS lock-in در زمان اجراهای استاندارد به صورتی سفارشی می‌شوند که در محیط ابر به صورت امن عمل کنند. مثلا در زمان اجراهای جاوا ممکن است که فراخوانی های خطرناک حذف یا دستکاری شوند.
- مانند SaaS، PaaS هم از lock in داده رنج می برد. ولی در این حالت مسئولیت کامل ساختن روتین ها بر عهده مشتریان است.

IaaS lock-in-۴-۱-۴-۹

این بخش از lock-in ها وابسته به خدمات زیرساختی مصرف شده می‌باشد. برای مثال، یک مشتری که از ذخیره سازهای ابر استفاده می‌کند نمی‌تواند توسط فرمت های ماشین مجازی غیر سازگار تحت تاثیر قرار گیرند.

- تامین کننده های محاسبات IaaS به طور نمونه ماشین‌های مجازی مبتنی بر فوق ناظر را ارائه می‌دهند. نرم‌افزار و متاداده‌های ماشین مجازی برای ارائه قابلیت حمل با هم همراه می‌شوند – مثلا در ابر تامین کننده. امکان انتقال ما بین تامین کننده ها بعد از پیاده‌سازی استانداردهای باز مانند OVF [۷] امکان‌پذیر می‌باشد.
- ارائه های تامین کننده های ذخیره‌سازی در IaaS ممکن است که از ذخیره داده‌های مبتنی بر کلید ساده تا ذخیره های مبتنی بر فایل متفاوت باشند. مجموعه ویژگی‌ها می‌توانند به طور قابل توجهی متفاوت باشند، بنابراین سمانتیک های ذخیره‌سازی هم متفاوت هستند.
- در سرویس‌های ذخیره‌سازی IaaS نیز lock in داده محرز می‌باشد. هرچه مشتریان ابر داده بیشتری را به ذخیره‌سازی ابر ارسال کنند lock in داده افزایش پیدا می‌کند، مگر اینکه تامین کننده ابر قابلیت جابجایی داده را فراهم کند.

۹-۴-۲- ر۲. فقدان نظارت

احتمال	خیلی بالا	مقایسه ای: بالاتر
تاثیر	خیلی بالا (بسته به سازمان: برای IaaS خیلی بالا، برای SaaS کم)	مقایسه ای: مساوی
آسیب‌پذیری‌ها	آ۳۴. نقش ها و مسئولیت های گنگ آ۳۵. اجرای ضعیف تعاریف نقش ها آ۲۱. همگام سازی مسئولیت ها یا تعهدات پیمانی خارج از ابر آ۲۳. عبارات SLA با توافق‌های متداخل برای ذینفعان مختلف آ۲۵. حسابرسی و تصدیق مشتریان امکان‌پذیر نباشد آ۲۲. برنامه‌های کاربردی میان ابری موجب بوجود آمدن وابستگی های پنهان می‌شوند	

[1] Runtime

	آ۱۳. کمبود فناوری‌ها و راه‌حل‌های استاندارد آ۲۹. ذخیره‌سازی داده در چندین حوزه و عدم شفافیت در این باره آ۱۴. عدم وجود توافق اجرایی منبع آ۱۶. عدم وجود کنترل بر روی پروسه تشخیص آسیب‌پذیری آ۲۶. طرح های تصدیق با زیرساخت ابر جور نباشد آ۳۰. فقدان اطلاعات درباره حوزه ها آ۳۱. فقدان تکامل و شفافیت شرایط استفاده آ۴۴. مالکیت نامشخص دارایی ها
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۳. وفاداری و تجربه کارکنان د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس
ریسک	بالا

در استفاده از زیرساخت‌های ابر، مشتری کنترل چند موضوع که بر امنیت تاثیرگذار هستند را به تامین کننده ابر واگذار می‌کند. برای مثال شرایط استفاده (ToUs) ممکن است از بررسی کردن پورت ها، تشخیص آسیب‌پذیری‌ها و تست های نفوذ ممانعت بعمل آورند. از سوی دیگر ممکن است تداخل‌هایی مابین رویه‌های مقاوم‌سازی انجام شده توسط مشتری و محیط ابر بوجود بیاید (ر۲۰ را ببینید). از دگر سو توافقنامه های سطح سرویس ممکن است تعهدی را در قبال سرویس‌های تامین کننده ابر ارائه نکنند، بنابراین یک شکاف امنیتی در این بین به وجود می‌آید.

علاوه بر این تامین کننده ابر ممکن است که خدمات خود را به تامین کننده شخص سومی برونسپاری کند که تضمین های قبلی در این حالت شاید وجود نداشته باشند. یا اینکه کنترل تامین کننده ابر تغییر کند بنابراین شرایط سرویس‌ها نیز تغییر می‌کند.

فقدان نظارت و کنترل ابر ممکن است که بر استراتژی سازمان تاثیرگذار باشد و بر ظرفیت تامین اهداف آن تاثیر بگذارد. اتلاف کنترل و نظارت ممکن است منجر به عدم برآوردن نیازمندی های امنیتی، کاهش محرمانگی، صحت و دسترس پذیری داده، و زوال کارایی و کیفیت خدمات شود.

۹-۴-۳- ر۳. چالش های مطلوبیت[1]

احتمال	خیلی بالا – وابسته به PCI	مقایسه ای: بالاتر
تاثیر	بالا	مقایسه ای: مساوی
آسیب‌پذیری‌ها	آ۲۵. حسابرسی و تصدیق مشتریان امکان‌پذیر نباشد آ۱۳. کمبود فناوری‌ها و راه‌حل‌های استاندارد آ۲۹. ذخیره‌سازی داده در چندین حوزه و عدم شفافیت در این باره آ۲۶. طرح های تصدیق با زیرساخت ابر جور نباشد آ۳۰. فقدان اطلاعات درباره حوزه ها آ۳۱. فقدان تکامل و شفافیت شرایط استفاده	
دارایی های تحت تاثیر	د۲۰. تصدیق	
ریسک	بالا	

سازمان‌هایی که به گرید مهاجرت[2] می‌کنند سرمایه‌گذاری‌های قابل توجهی را در بدست آوردن گواهی‌ها یا در جهت بدست آوردن مزیت رقابتی یا در جهت دستیابی به استانداردهای صنعتی یا نیازمندی های قانونی انجام می‌دهند. این سرمایه‌گذاری‌ها در مهاجرت به ابر ممکن است که با خطر مواجه شوند:

- اگر تامین کننده ابر قادر به فراهم کردن شواهدی برای مطلوبیت خود با توجه به نیازمندی های موجود نباشد.
- اگر تامین کننده ابر اجازه حسابرسی به مشتری ابر[3] ندهد.

در موارد مشخص، استفاده از زیرساخت‌های عمومی ابر دلالت بر این دارد که انواع مشخصی از مطلوبیت قابل دستیابی نمی‌باشد و بنابراین سرویس‌های میزبانی ابر که مورد نیاز سرویس‌های دیگر می‌باشند قابل استفاده نمی‌باشند. برای مثال EC2 می گوید که امکان رسیدن به مطلوبیت در PCI در سکوی آنها نمی‌باشد. بنابراین سرویس‌های میزبان برای تراکنش‌های کارت های اعتباری قابل استفاده نمی‌باشند.

۹-۴-۴- ر۴.فقدان شهرت کسب و کار بعلت فعالیت های co-tenant

احتمال	پایین	
تاثیر	بالا	
آسیب‌پذیری‌ها	آ۶. فقدان جداسازی منبع آ۷. فقدان جداسازی های شهرتی آ۵. آسیب‌پذیری‌های فوق ناظر	

[1] Compliance
[2] Migrating
[3] Cloud Customer

دارایی های تحت تاثیر	د۱. شهرت کمپانی د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس
ریسک	متوسط

اشتراک منابع به این معنا می‌باشد که فعالیت های بدخواه[1] که توسط یک مستاجر[2] انجام می‌شوند ممکن است که بر شهرت دیگر مستاجرها تاثیر بگذارند. برای مثال spamming، بررسی پورت‌ها یا ارسال محتواهای بدخواهانه از زیر ساخت‌های ابر ممکن است منجر به:

- دسته‌ای از آدرس‌های IP که مسدود شده‌اند، شامل حمله کننده‌ها و دیگر مستاجران بی گناه
- توقیف منابع بعلت فعالیت‌های همسایه ها[3]

تاثیر این موارد می‌تواند موجب بدتر شدن سرویس تحویل شده و اتلاف داده‌ها شود، و از سویی شهرت سازمان را در معرض خطر قرار دهد.

۹-۴-۵- ر۵. انقضا یا خرابی سرویس ابر[4]

احتمال	N/A	
تاثیر	خیلی بالا	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۴۶. انتخاب ضعیف تامین کننده آ۴۷. فقدان افزونگی تامین کننده آ۳۱. فقدان تکامل و شفافیت شرایط استفاده	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۳. وفاداری و تجربه کارکنان د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	متوسط	

[1] malicious
[2] tenant
[3] neighbor subpoenaed
[4] Cloud service termination or failure

مانند هر بازار نوظهور در عرصه فناوری اطلاعات، فشارهای رقابتی، یک استراتژی کسب و کار نامناسب، عدم پشتیبانی مالی لازم و ... می‌توانند موجب خارج شدن برخی تامین کننده ها یا مجبور ساختن آنها به بازساختاردهی سرویس‌های پیشنهادی می‌شود. به عبارت دیگر می‌توان گفت که این امکان وجود دارد که در یک دوره کوتاه یا متوسط برخی خدمات رایانش ابری از بین بروند.

تاثیر این خطر بر روی مشتریان ابر از آنجایی که موجب اتلاف و تباهی کارایی سرویس‌ها و کیفیت آنها می‌شود کاملا مشهود می‌باشد.

از سوی دیگر شکست در خدماتی که به ابر برونسپاری شده است باعث می‌شود که توانایی مشتریان ابر برای انجام وظایف خود افزایش پیدا کند و آنها برای تامین نیازهای خود مسئولیت ها را به کارکنان خود بدهند.

۹-۴-۶- ر۶.تملک تامین کننده ابر[1]

احتمال	N/A	
تاثیر	متوسط	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۳۱. فقدان تکامل و شفافیت شرایط استفاده	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۳. وفاداری و تجربه کارکنان د۴. دارایی فکری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	متوسط	

تملک تامین کننده ابر می‌تواند موجب افزایش احتمال تغییر استراتژیک سازمان و باعث در خطر قرار گرفتن توافق‌های غیر الزام آور (مثلا سرمایه گذاری برای امنیت، کنترل های امنیتی خارج از قرارداد) شود. در نتیجه ممکن است برآوردن نیازمندی های امنیتی غیر ممکن شود. تاثیر نهایی این ریسک می‌تواند از بین رفتن دارایی های حیاتی سازمان باشد. دارایی هایی مانند: شهرت سازمان، اعتماد مشتری، وفاداری و تجربه کارمندان.

[1] Cloud computing acquisition

۹-۴-۷- ر۷. شکست زنجیره تامین

احتمال	پایین	مقایسه ای: بالاتر
تاثیر	متوسط	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۳۱. فقدان تکامل و شفافیت شرایط استفاده آ۲۲. برنامه‌های کاربردی میان ابری موجب بوجود آمدن وابستگی های پنهان می‌شوند آ۴۶. انتخاب ضعیف تامین کننده آ۴۷. فقدان افزونگی تامین کننده	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	پایین	

یک تامین کننده رایانش ابری می‌تواند برخی فعالیت های تخصصی در زنجیره تولید خود را به شرکای شخص سوم برونسپاری کند. در چنین حالتی سطح امنیت تامین کننده ابر وابسته به سطح امنیت هر پیوند ما بین تامین کننده و شخص سوم و میزان وابستگی تامین کننده ابر به شخص سوم می‌باشد. هرگونه تداخل در این زنجیر و یا عدم همکاری مابین مسئولیت های طرفین می‌تواند منجر به موارد زیر شود: در دسترس نبودن سرویس‌ها، فقدان محرمانگی داده، صحت و دسترس پذیری، از بین رفتن شهرت و منافع اقتصادی به خاطر شکست در تامین نیازهای مشتری، تخطی از SLA و شکست سرویس‌ها و غیره.

به طور کلی، عدم وجود شفافیت در قرارداد می‌تواند مسئله ای برای کل سیستم باشد. اگر تامین کننده به صراحت اعلام نکند که کدام یک از سرویس‌های اصلی فناوری اطلاعات برونسپاری شده‌اند مشتری قادر به ارزیابی دقیق خطرات احتمالی نمی‌باشد. فقدان شفافیت می‌تواند منجر به کاهش سطح اعتماد به تامین کننده شود.

۹-۵- ریسک‌های فنی

۹-۵-۱- ر۸. فرسودگی منابع

احتمال	الف. اضافه کردن ظرفیت برای مشتری مقدور نباشد: متوسط	مقایسه ای: N/A
	ب. تامین ظرفیت توافق شده مقدور نباشد: پایین	مقایسه ای: بالاتر
تاثیر	الف. اضافه کردن ظرفیت برای مشتری مقدور نباشد: پایین / متوسط	مقایسه ای: N/A
	ب. تامین ظرفیت توافق شده مقدور نباشد: بالا	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۱۵. مدل‌های نادقیق استفاده از منبع آ۲۷. سرمایه گذاری ناکافی بر روی زیرساخت‌ها آ۲۸. عدم وجود خط و مشی بستن منابع آ۴۷. فقدان افزونگی تامین کننده	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۱۰. تحویل سرویس د۱۱. کنترل دسترسی / احراز هویت / مجازشناسی	
ریسک	متوسط	

سرویس‌های ابر سرویس‌های بلادرنگ هستند. بنابراین یک سطح از ریسک محاسبه شده، در تخصیص تمام منابع سرویس ابر وجود دارد، این امر به خاطر آن است که منابع مبتنی بر نتایج آماری، تخصیص داده شده‌اند. مدلسازی غیر دقیق استفاده از منابع یا تدارک غیر دقیق بر منبع و عدم سرمایه گذاری بر زیرساخت‌ها می‌تواند از دید تامین کننده ابرمنجر به موارد زیر شود:

- دسترس ناپذیری سرویس: شکست در سناریو های برنامه کاربردی که از یک منبع خاص به شدت استفاده می‌کنند. مثلا استفاده شدید از حافظه برای شبیه‌سازی در کاربردهایی چون پیش بینی قیمت های سهام.
- توافق برای کنترل دسترسی: در برخی موارد این امکان وجود دارد که در رخداد فرسودگی منبع سیستم را وادار به 'fail open' کند.
- ضررهای اقتصادی و شهرت: بعلت شکست در تامین تقاضاهای مشتریان.
- اتفاقات مخرب به واسطه تخمین نامناسب نیازهای منابع

از دید مشتری ابر، انتخاب بد تامین کننده و فقدان افزونگی لازم در آنها منجر می‌شود به:

- دسترس ناپذیری سرویس: شکست در تحویل سرویس‌ها.
- توافق برای کنترل دسترسی: قرار دادن محرمانگی و صحت داده‌ها در خطر.
- ضررهای اقتصادی و شهرت: بعلت شکست در تامین نیازهای مشتریان.

نتایج این ریسک با نتایج ر۱۵ یکی است.

۹-۵-۲- ر۹. شکست در جداسازی

احتمال	پایین (ابر خصوصی) متوسط (ابر عمومی)	مقایسه ای: بالاتر
تاثیر	خیلی بالا	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۵. آسیب‌پذیری‌های فوق ناظر آ۶. فقدان جداسازی منبع آ۷. فقدان جداسازی‌های شهرتی آ۱۷. احتمال رخداد وارسی‌های داخل شبکه‌ای آ۱۸. احتمال اجرای بررسی‌های co-residence	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	بالا	

مالکیت چندگانه و اشتراک منابع از مشخصه های تعریف شده برای رایانش ابری می‌باشد. ظرفیت های محاسباتی، ذخیره‌سازی و شبکه میان چندین کاربر به اشتراک گذاشته می‌شود. این دسته از ریسک‌ها شامل شکست در مکانیزم های جداسازی ما بین منابع ذخیره‌سازی، حافظه، مسیریابی و حتی شهرت بین مالکان مختلف زیرساخت‌های مشترک می‌شود. (برای نمونه می‌توان به SQL injection attack اشاره کرد که در آن داده‌های چندین مشتری در یک جدول قرار داشته و به واسطه این حمله افشا می‌شود)

احتمال رخداد چنین سناریویی را در مدل ابر در نظر بگیرید، این احتمال در ابرهای خصوصی کمتر و در ابرهای عمومی بیشتر می‌باشد.

۹-۵-۳- ر۱۰. کارمند خودی بدخواه در تامین کننده ابر

احتمال	متوسط (پایین تر از سنتی)	مقایسه ای: پایین‌تر
تاثیر	خیلی بالا (بالاتر از سنتی)	مقایسه ای: بالاتر (مجتمع) مقایسه ای: یکسان (مشتری منفرد)
آسیب‌پذیری‌ها	آ۳۴. نقش ها و مسئولیت های گنگ آ۳۵. اجرای ضعیف تعاریف نقش ها آ۳۶. بکارنرفتن اصل دانستن با توجه به نیاز آ۱. آسیب‌پذیری‌های AAA آ۳۹. آسیب‌پذیری‌های سیستم یا سیستم عامل آ۳۷. ناکافی بودن رویه های فیزیکی امنیتی آ۱۰. غیر ممکن بودن پردازش داده‌ها در حالت رمز شده آ۴۸. آسیب‌پذیری‌های برنامه کاربردی یا مدیریت ضعیف وصله	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۳. وفاداری و تجربه کارکنان د۴. دارایی فکری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	بالا	

اقدامات بدخواهانه یک کارمند داخلی می‌تواند بر محرمانگی، صحت و دسترس پذیری انواع داده، IP و تمام انواع سرویس تاثیر گذار باشد. و به طور مستقیم بر شهرت سازمان، اعتماد مشتری و تجربه مشتریان اثر بگذارد. این ریسک می‌تواند با قدرت بیشتری در رایانش ابری دیده شود زیرا در معماری ابر افرادی با نقش های پر ریسک بیشتر وجود دارند. از جمله این نقش ها می‌توان به مدیران سیستم تامین کننده ابر، حسابرسان و مدیران سرویس‌های امنیتی که وظیفه گزارشات تشخیص نفوذ و مسئولیت اتفاقات را برعهده دارند نام برد. از آنجایی که استفاده از ابرها در حال گسترش است، کارمندان تامین کننده های ابر تبدیل به اهدافی برای گروه‌های خلافکار شده‌اند[۸و۹].

۹-۵-۴- ر۱۱.به خطر افتادن واسط مدیریت (دستکاری، دسترس پذیری زیرساخت)

احتمال	متوسط	مقایسه ای: بالاتر
تاثیر	خیلی بالا	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۱. آسیب‌پذیری‌های AAA آ۴. دسترسی از راه دور به واسط مدیریت آ۳۸. پیکره بندی نامناسب آ۳۹. آسیب‌پذیری‌های سیستم یا سیستم عامل آ۴۸. آسیب‌پذیری‌های برنامه کاربردی یا مدیریت ضعیف وصله	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس د۱۴. واسط مدیریت سرویس ابر	
ریسک	متوسط	

واسط‌های مدیریت مشتری در ابرهای عمومی از طریق اینترنت قابل دسترسی هستند و دسترسی به مجموعه بزرگی از منابع به واسطه آنها امکانپذیر می‌باشد. با وجود این موضوعات ریسک‌های قابل توجهی آنها را تهدید می‌کنند بخصوص اینکه آسیب‌پذیری‌های دسترسی از راه دور و مرورگرهای اینترنت به شدت این ریسک‌ها می افزاید. این ریسک‌ها وقتی در نظر می گیریم که واسط‌های مشتریان قادر به کنترل ماشین‌های مجازی هستند و واسط‌های تامین کننده کلیه عملیات ابر را کنترل می‌کند بیشتر خود را نشان می‌دهند. این ریسک با سرمایه گذاری در امنیت توسط تامین کننده ها قابل کاهش است.

۹-۵-۵- ر۱۲. استراق سمع داده در انتقال

احتمال	متوسط	مقایسه ای: بالاتر
تاثیر	بالا	مقایسه ای: یکسان
آسیب‌پذیری‌ها	آ۱. آسیب‌پذیری‌های AAA آ۸. آسیب‌پذیری‌های رمزنگاری ارتباط آ۹. فقدان یا ضعف در رمزنگاری آرشیو ها و داده در انتقال آ۱۷. احتمال رخداد وارسی های داخل شبکه ای آ۱۸. احتمال اجرای بررسی‌های co-residence	

	آ۳۱. فقدان تکامل و شفافیت شرایط استفاده
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۴. دارایی فکری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۲۳. پشتیبان داده یا بایگانی داده
ریسک	متوسط

در رایانش ابری که دارای یک معماری توزیع شده می‌باشد، انتقال داده بسیار بیشتر از زیرساخت‌های سنتی می‌باشد. برای نمونه، در همگام سازی چندین تصویر ماشین توزیع شده با هم داده می بایست منتقل شود و این انتقال در زیرساخت ابر انجام می‌شود.

حملات Sniffing، Spoofing، man in the middle، side channel و reply از جمله تهدیدهای محتمل می‌باشند. به هر صورت در برخی از موارد تامین کننده ابر اقدامات لازم برای تامین امنیت در مقابل این تهدیدها را انجام نمی‌دهد.

۹-۵-۶- ر۱۳. هدر رفتن داده در بارگذاری و دریافت

احتمال	متوسط (N/A)	
تاثیر	بالا	
آسیب‌پذیری‌ها	آ۱. آسیب‌پذیری‌های AAA آ۸. آسیب‌پذیری‌های رمزنگاری ارتباط آ۱۷. احتمال رخداد وارسی های داخل شبکه ای آ۱۸. احتمال اجرای چک های co-residence آ۱۰. غیر ممکن بودن پردازش داده‌ها در حالت رمز شده آ۴۸. آسیب‌پذیری‌های برنامه کاربردی یا مدیریت ضعیف وصله	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۳. وفاداری و تجربه کارکنان د۴. دارایی فکری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR	

	د۱۲. اعتبارنامه د۱۳. دایرکتوری کاربر (داده) د۱۴. واسط مدیریت سرویس ابر
ریسک	متوسط

این ریسک شبیه به ریسک قبلی می‌باشد با این تفاوت که فقط بر انتقال داده بین تامین کننده و مشتری تمرکز دارد.

۹-۵-۷- ر۱۴. حذف ناامن یا غیر موثر داده[1]

احتمال	متوسط	مقایسه ای: بالاتر
تاثیر	خیلی بالا	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۲۰. قابل قبول شدن رسانه های حساس	
دارایی های تحت تاثیر	د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۱۲. اعتبارنامه	
ریسک	متوسط	

هرگاه که یک تامین کننده تغییر می‌کند، منابع تغییر مقیاس داده می‌شوند، سخت‌افزارهای فیزیکی تخصیص دوباره داده می‌شوند و داده‌ها ممکن است مدت زمان بیشتری نسبت به آنچه قرار بوده باشند در ابر در دسترس باشند. ممکن است که برنامه حذفی که در خط و مشی امنیتی آورده شده را نتوان به طور کامل اجرا کرد و حذف کامل داده‌ها تنها در صورت از بین بردن فیزیکی دیسک های سخت‌افزاری امکان‌پذیر باشد. وقتی یک درخواست برای حذف یک منبع ابر ایجاد شود ممکن است که به نتیجه کامل منجر نشود. وقتی که حذف کامل مورد نیاز باشد نیاز به انجام عملیاتی است که توسط API های استاندارد پشتیبانی نمی‌شود.

اگر از روش‌های موثر رمزنگاری استفاده شود این ریسک تا حد بسیار زیادی کاهش پیدا می‌کند.

۹-۵-۸- ر۱۵. Distributed denial of service (DDoS)

احتمال	مشتری: متوسط	مقایسه ای: پایین‌تر
	تامین کننده: پایین	مقایسه ای: N/A
تاثیر	مشتری: بالا	مقایسه ای: بالاتر

[1] Insecure or ineffective deletion of data

	تامین کننده: خیلی بالا	مقایسه ای: پایین‌تر
آسیب‌پذیری‌ها	آ۳۸. پیکره بندی نامناسب آ۳۹. آسیب‌پذیری‌های سیستم یا سیستم عامل آ۵۳. فیلترینگ ناکافی منابع	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس د۱۴. واسط مدیریت سرویس ابر د۱۶. شبکه (ارتباطات و غیره)	
ریسک	متوسط	

۹-۵-۹- ر۱۶. Economic denial of service (EDoS)

احتمال	پایین	
تاثیر	بالا	
آسیب‌پذیری‌ها	آ۱. آسیب‌پذیری‌های AAA آ۲. آسیب‌پذیری‌های تدارکاتی کاربران آ۳. آسیب‌پذیری‌های قطع تدارک کاربران آ۳. آسیب‌پذیری‌های قطع تدارک کاربران آ۲۸. عدم وجود خط و مشی بستن منابع	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	متوسط	

سناریو های مختلفی وجود دارد که در آن منابع مشتریان ابر توسط روش‌های بدخواهانه شخص سوم ها مورد استفاده قرار گرفته و تاثیرات اقتصادی دارند:

- دزدی هویت: فرد حمله کننده از حساب کاربری دیگری استفاده کرده و منابع مشتری را برای منافع شخصی مورد استفاده قرار می‌دهد و حساب آن فرد را از بین می برد.
- اگر مشتری ابر محدودیت استفاده برای حساب کاربری خود ایجاد نکند موجب می‌شود که آن حساب را در مقابل این تهدید ضعیف تر کند.

- حمله کننده از یک کانال عمومی برای ضربه اقتصادی وارد کردن به مشتریان استفاده می‌کند- مثلا جایی که پرداخت به صورت پرداخت به‌ازای هر درخواست http صورت می‌گیرد انجام حمله DDos موجب ایجاد خسارت می‌شود.

حملات EDoS موجب وارد شدن ضربات اقتصادی به مشتریان می‌شود، در بدترین حالت ممکن است موجب برشکستگی مشتری یا تاثیرات اقتصادی بر آنها شود.

۹-۵-۱۰- ر۱۷. فقدان کلیدهای رمزنگاری

احتمال	پایین	مقایسه ای: N/A
تاثیر	بالا	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۱۱. روش های ضعیف مدیریت کلید آ۱۲. تولید کلید: آنتروپی پایین برای تولید اعداد تصادفی	
دارایی های تحت تاثیر	د۴. دارایی فکری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۱۲. اعتبارنامه	
ریسک	متوسط	

این ریسک شامل افشای کلیدهای رمز (SSL، رمزنگاری فایل، کلیدهای خصوصی مشتری و ...) یا کلمه های عبور مشتریان برای اشخاص بدخواه می‌باشد. از افشای این کلیدها برای استفاده غیر مجاز در احراز هویت و امضای دیجیتال استفاده خواهد شد.

۹-۵-۱۱- ر۱۸. جستجو ها و پویش های بدخواهانه[1]

احتمال	متوسط	مقایسه ای: پایین‌تر
تاثیر	متوسط	مقایسه ای: پایین‌تر
آسیب‌پذیری‌ها	آ۱۷. احتمال رخداد وارسی های داخل شبکه ای آ۱۸. احتمال اجرای بررسی‌های co-residence	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۹. تحویل سرویس – سرویس‌های بلادرنگ	

[1] Undertaking malicious probes or scans

	د۱۰. تحویل سرویس
ریسک	متوسط

جستجوها و پویش های بدخواهانه، نوعی از خطر هستند که به صورت غیر مستقیم دارایی های سازمان را تهدید می‌کنند. از این جستجوها می‌شود برای جمع آوری اطلاعات به جهت تلاش‌های نفوذ و هک استفاده کرد. از جمله عواقب آن می‌توان از بین رفتن محرمانگی، صحت و دسترس پذیری داده و سرویس را نام برد.

۹-۵-۱۲- ر۱۹. به خطر افتادن موتورهای سرویس

احتمال	پایین	
تاثیر	خیلی بالا	
آسیب‌پذیری‌ها	آ۵. آسیب‌پذیری‌های فوق‌ناظر آ۶. فقدان جداسازی منبع	
دارایی های تحت تاثیر	د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	متوسط	

هر یک از معماری های ابر برروی یک سکوی خاص ساخته شده‌اند، موتور سرویس در بالای منابع سخت‌افزاری فیزیکی قرار می‌گیرد و وظیفه مدیریت منابع مشتریان را در سطوح مختلف بر عهده دارد. برای نمونه در ابرهای IaaS مولفه های نرم‌افزار موتور سرویس می‌تواند فوق‌ناظر باشد. در برخی از موارد موتور سرویس توسط سکوهای متن باز ساخته و پشتیبانی می‌شوند. سفارشی‌سازی بیشتر این موتورها را تامین کنندگان ابر برعهده می‌گیرند.

مانند تمامی نرم‌افزارهای دیگر کدهای موتور سرویس هم دارای آسیب‌پذیری‌هایی هستند و مستعد برای مورد حمله قرار گرفتن می‌باشند. یک حمله کننده می‌تواند موتور سرویس را به واسطه هک کردن ماشین مجازی (ابرهای IaaS) هک کردن محیط زمان اجرا (ابرهای PaaS) هک کردن application pool و یا به واسطه API ها مورد خطر قرار دهند.

هک کردن موتور سرویس موجب ایجاد دسترسی به داده‌های موجود در آنها و ایجاد قابلیت نظارت ودستکاری بر اطلاعات داخلی برای فرد حمله کننده می‌شود.

۹-۵-۱۳-ر۲۰. تداخل بین رویه‌های مقاوم‌سازی توسط مشتریان و محیط ابر

احتمال	پایین	
تاثیر	متوسط	
آسیب‌پذیری‌ها	آ۳۱. فقدان تکامل و شفافیت شرایط استفاده آ۲۳. عبارات SLA با قولهای متداخل برای ذی نفعان مختلف آ۳۴. نقش ها و مسئولیت های گنگ	
دارایی های تحت تاثیر	د۴. دارایی فکری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی	
ریسک	پایین	

تامین کنندگان ابر باید به صورت کاملا روشن تفکیک مسئولیت ها را در ابر مشخص کنند و باید برای هر مشتری حداقل کارها که باید برای امن کردن محیط ابر انجام دهد مشخص شود. شکست مشتریان در امن کردن موفق محیط موجب آسیب‌پذیری ابر خواهد شد. از سوی دیگر تامین کننده ها باید راهنمایی برای مشتریان تهیه کنند تا آنها بتوانند به طور موفقیت آمیز کارهای امن سازی را انجام دهند.

مشتریان ابر باید از مسئولیت های خود به طور کامل آگاه باشند. در برخی موارد مشتریان به طور نادرست تصور می‌کنند که تمام مسئولیت ها بر عهده تامین کننده می‌باشد، و او تمام کارهای امنیتی را انجام می‌دهد. این تصور مشتریان و عدم اطلاع رسانی کامل توسط تامین کننده موجب بوجود آمدن ریسک می‌شود. بنابراین ضروری است که مشتری از مسئولیت های خود آگاه باشد و به آنها عمل کند.

۹-۶- ریسک‌های قانونی

۹-۶-۱- ر۲۱. Subpoena and E-Discovery

احتمال	بالا	
تاثیر	متوسط	
آسیب‌پذیری‌ها	آ۶. فقدان جداسازی منبع آ۲۹. ذخیره‌سازی داده در چندین حوزه و عدم شفافیت در این باره آ۳۰. فقدان اطلاعات درباره حوزه ها	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	بالا	

رخداد توقیف سخت‌افزار فیزیکی، منتج از حکم قانونی آژانس‌های مسئول و تمرکز ذخیره‌سازی ابر در یک نقطه می‌تواند به این معنا باشد که مشتریان در معرض خطر افشای داده‌هایشان می‌باشند [۱۰و۱۱و۱۲].

۹-۶-۲- ر۲۲. ریسک‌های ناشی از تغییرات حوزه های قضایی[1]

احتمال	خیلی بالا	
تاثیر	بالا	
آسیب‌پذیری‌ها	آ۳۰. فقدان اطلاعات درباره حوزه ها آ۲۹. ذخیره‌سازی داده در چندین حوزه و عدم شفافیت در این باره	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	بالا	

[1] changes of jurisdiction

داده مشتریان ممکن است در چند حوزه قضایی نگهداری شوند، برخی از این حوزه ها با ریسک زیادی همراه هستند (مثلا مرکز داده‌هایی که در کشورهای پر ریسک قرار دارند). در این حوزه ها ممکن است فقدان قوانین بازدارنده و محدود کننده زمینه های لازم برای افشا یا سرقت داده‌ها را فراهم کند. لازم به ذکر است که در اینجا نسبت به تمامی حوزه ها وجود ریسک مطرح نشده و فقط حوزه هایی که به موجب فشارهای قانونی خاص مورد توجه هستند منظور باشد.

۹-۶-۳- ر۲۳. ریسک‌های پشتیبانی داده

احتمال	بالا	
تاثیر	بالا	
آسیب‌پذیری‌ها	آ۳۰. فقدان اطلاعات درباره حوزه ها آ۲۹. ذخیره‌سازی داده در چندین حوزه و عدم شفافیت در این باره	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	بالا	

رایانش ابری چند ریسک پشتیبانی داده هم برای مشتریان ابر و هم تامین کنندگان دارند.

- برای یک مشتری ابر (در نقش کنترل کننده داده) ممکن است چک کردن پردازش های داده که توسط تامین کننده انجام می‌شود مشکل باشد، و بنابراین مطمئن شدن از انجام از طریق قانونی نیز مشکل می‌باشد. باید کاملا روشن باشد که مشتری ابر مسئول شخصی پردازش های داده‌های شخصی می‌باشد، حتی وقتیکه این پردازش ها توسط تامین کننده ابر انجام می‌شود. شکست در اجرای قوانین پشتیبانی از داده موجب مجازات های مختلفی می‌شود که در کشورهای مختلف با توجه به قوانین متفاوت است.
- ممکن است نفوذ های امنیت داده ای وجود داشته باشند که کنترل کننده از آنها آگاه نشود.
- مشتری ابر ممکن است کنترل خود به داده پردازش شده توسط تامین کننده ابر را از دست بدهد. در حالت افزایش انتقال های داده این مورد افزایش می یابد.
- تامین کننده ابر ممکن است داده‌هایی را دریافت کند که جمع آوری آنها توسط مشتری منع قانونی داشته باشد.

۹-۶-۴- ر۲۴. ریسک‌های صدور مجوز[1]

احتمال	متوسط	مقایسه ای: بالاتر
تاثیر	متوسط	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۳۱. فقدان تکامل و شفافیت شرایط استفاده	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۲۰. تصدیق	
ریسک	متوسط	

شرایط صدور مجوز، مانند توافقنامه‌های اولیه و چک‌کردن‌های برخط صدور مجوز ممکن است برای محیط ابر کاربرد نداشته باشند. برای مثال، در حالت پرداخت موردی برای هر زمان استفاده از ماشین در محیط ابر ممکن است صورتحساب مشتری به صورت نمایی افزایش پیدا کند در حالیکه او به طور مدام از یک مورد استفاده می‌کند.

[1] Licensing risks

۹-۷- ریسک‌های عمومی (نه فقط مربوط به ابر)

در زمینه تحلیل‌های ریسک ما با یک سری از ریسک‌ها مواجه هستیم که فقط مربوط به محیط ابر نمی‌باشد، و در همه حالات باید آنها را مدنظر قرار بدهیم.

۹-۷-۱- ر۲۵. خرابی های شبکه

احتمال	پایین	مقایسه ای: یکسان
تاثیر	خیلی بالا	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۳۸. پیکره بندی نامناسب آ۳۹. آسیب‌پذیری‌های سیستم یا سیستم عامل آ۶. فقدان جداسازی منبع آ۴۱. فقدان یا ضعف در پیوستگی کسب و کار و طرح های بازیابی بد	
دارایی های تحت تاثیر	د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	متوسط	

یکی از شایعترین ریسک‌ها که در صورت وقوع، چندین هزار مشتری به طور همزمان درگیر می‌شوند.

۹-۷-۲- ر۲۶. مدیریت شبکه[1]

احتمال	متوسط	مقایسه ای: یکسان
تاثیر	خیلی بالا	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۳۸. پیکره بندی نامناسب آ۳۹. آسیب‌پذیری‌های سیستم یا سیستم عامل آ۶. فقدان جداسازی منبع آ۴۱. فقدان یا ضعف در پیوستگی کسب و کار و طرح های بازیابی بد	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۳. وفاداری و تجربه کارکنان د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس د۱۶. شبکه (ارتباطات و غیره)	
ریسک	بالا	

[1] IE, network congestion / mis-connection / non-optimal use

۹-۷-۳- ر۲۷. دستکاری ترافیک شبکه

احتمال	پایین	
تاثیر	بالا	
آسیب‌پذیری‌ها	آ۲. آسیب‌پذیری‌های تدارکاتی کاربران آ۳. آسیب‌پذیری‌های قطع تدارک کاربران آ۸. آسیب‌پذیری‌های رمزنگاری ارتباط آ۱۶. عدم وجود کنترل بر روی پروسه تشخیص آسیب‌پذیری	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس	
ریسک	متوسط	

۹-۷-۴- ر۲۸. Privilege escalation

احتمال	پایین	مقایسه ای: پایین‌تر
تاثیر	بالا	مقایسه ای: بالاتر (برای تامین‌کننده ابر)
آسیب‌پذیری‌ها	آ۱. آسیب‌پذیری‌های AAA آ۲. آسیب‌پذیری‌های تدارکاتی کاربران آ۳. آسیب‌پذیری‌های قطع تدارک کاربران آ۵. آسیب‌پذیری‌های فوق‌ناظر آ۳۴. نقش ها و مسئولیت های گنگ آ۳۵. اجرای ضعیف تعاریف نقش ها آ۳۶. بکارنرفتن اصل دانستن با توجه به نیاز آ۳۸. پیکره بندی نامناسب	
دارایی های تحت تاثیر	د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۱۱. کنترل دسترسی / احراز هویت / مجازشناسی د۱۳. دایرکتوری کاربر (داده)	
ریسک	متوسط	

۹-۷-۵- ر۲۹. Social engineering attacks (IE, Impersonation)

احتمال	متوسط	مقایسه ای: یکسان
تاثیر	بالا	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۳۲. فقدان آگاهی امنیتی آ۲. آسیب‌پذیری‌های تدارکاتی کاربران آ۶. فقدان جداسازی منبع آ۸. آسیب‌پذیری‌های رمزنگاری ارتباط آ۳۷. ناکافی بودن رویه های فیزیکی امنیتی	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۴. دارایی فکری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۱۱. کنترل دسترسی / احراز هویت / مجازشناسی د۱۲. اعتبارنامه	
ریسک	متوسط	

۹-۷-۶- ر۳۰. اتلاف یا به خطر افتادن گزارش‌های عملیاتی

احتمال	پایین	مقایسه ای: پایین‌تر
تاثیر	متوسط	مقایسه ای: یکسان (برای مشتری)
آسیب‌پذیری‌ها	آ۵۲. فقدان یا ضعف در رویه های گزارش گیری و نگهداری آ۱. آسیب‌پذیری‌های AAA آ۲. آسیب‌پذیری‌های تدارکاتی کاربران آ۳. آسیب‌پذیری‌های قطع تدارک کاربران آ۱۹. فقدان آمادگی قانونی آ۳۹. آسیب‌پذیری‌های سیستم یا سیستم عامل	
دارایی های تحت تاثیر	د۲۱. گزارش‌های عملیاتی (مشتری و تامین کننده ابر)	
ریسک	پایین	

۹-۷-۷- ر۳۱. اتلاف یا به خطر افتادن گزارش‌های امنیتی (دستکاری تحقیقات قانونی)

احتمال	پایین	مقایسه ای: پایین‌تر
تاثیر	متوسط	مقایسه ای: پایین‌تر (برای مشتری)
آسیب‌پذیری‌ها	آ۵۲. فقدان یا ضعف در رویه های گزارش گیری و نگهداری آ۱. آسیب‌پذیری‌های AAA آ۲. آسیب‌پذیری‌های تدارکاتی کاربران آ۳. آسیب‌پذیری‌های قطع تدارک کاربران آ۱۹. فقدان آمادگی قانونی آ۳۹. آسیب‌پذیری‌های سیستم یا سیستم عامل	
دارایی های تحت تاثیر	د۲۲. گزارش‌های امنیتی	
ریسک	پایین	

۹-۷-۸- ر۳۲. گم شدن یا دزدیده شدن پشتیبان های داده

احتمال	پایین	مقایسه ای: پایین‌تر
تاثیر	بالا	مقایسه ای: یکسان (برای مشتری)
آسیب‌پذیری‌ها	آ۳۷. ناکافی بودن رویه های فیزیکی امنیتی آ۱. آسیب‌پذیری‌های AAA آ۲. آسیب‌پذیری‌های تدارکاتی کاربران آ۳. آسیب‌پذیری‌های قطع تدارک کاربران	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس د۲۳. پشتیبان داده یا بایگانی داده	
ریسک	متوسط	

۹-۷-۹- ر۳۳. دسترسی غیر مجاز به جزئیات ماشین‌ها و ابزار

احتمال	خیلی پایین	مقایسه ای: پایین‌تر
تاثیر	بالا (برای داشتن تاثیر خیلی زیاد باید هدف حمله مشخص باشد(مثلا یک ماشین خاص را هدف بگیرد))	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۳۷. ناکافی بودن رویه های فیزیکی امنیتی	
دارایی های تحت تاثیر	د۱. شهرت کمپانی د۲. اعتماد مشتری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۲۳. پشتیبان داده یا بایگانی داده	
ریسک	پایین	

از آنجایی که تامین‌کنندگان ابر بر مرکز داده‌های بزرگ تمرکز کرده‌اند و همچنین کنترل‌های محیط فیزیکی نیز قوی‌تر است، بنابراین نتایج ناشی از نفوذ بسیار پرخطرتر می‌شود.

۹-۷-۱۰- ر۳۴. دزدی تجهیزات رایانه‌ای

احتمال	خیلی پایین	مقایسه ای: پایین‌تر
تاثیر	بالا	مقایسه ای: خیلی بالا
آسیب‌پذیری‌ها	آ۳۷. ناکافی بودن رویه های فیزیکی امنیتی	
دارایی های تحت تاثیر	د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۱۷.سخت‌افزار فیزیکی	
ریسک	پایین	

۹-۷-۱۱- ر۳۵. بلایای طبیعی

احتمال	خیلی پایین	مقایسه ای: پایین‌تر
تاثیر	بالا	مقایسه ای: بالاتر
آسیب‌پذیری‌ها	آ۴۱. فقدان یا ضعف در پیوستگی کسب و کار و طرح‌های بازیابی بد	
دارایی های تحت تاثیر	د۱. شهرت کمپانی	

	د۲. اعتماد مشتری د۵. داده‌های شخصی حساس د۶. داده شخصی د۷. داده حیاتی شخصی د۸. داده HR د۹. تحویل سرویس – سرویس‌های بلادرنگ د۱۰. تحویل سرویس
ریسک	پایین

به طور کلی ریسک ناشی از بلایای طبیعی در محیط ابر بسیار کمتر از محیطهای سنتی می‌باشد، این موضوع به خاطر ارائه چندین سایت و مسیر شبکه توسط تامین کننده ها می‌باشد

۹-۸- آسیب‌پذیری‌ها

لیست آسیب‌پذیری‌هایی که در ادامه آورده شده یک لیست جامع نیست، ولی به هرصورت جزئیات لازم برای تامین اهداف تحلیل این گزارش را شامل می‌شود. این آسیب‌پذیری‌ها هم از نوع آسیب‌پذیری‌های مختص به ابر و هم از نوع آسیب‌پذیری‌های عمومی هستند.

آ۱. آسیب‌پذیری‌های AAA

یک سیستم که در احراز هویت[1] ، مجازشناسی[2] و حسابرسی[3] ضعیف باشد می‌تواند با مشکلاتی چون دسترسی غیرمجاز به منابع، استفاده نامناسب از منابع و اتفاقات امنیتی مواجه شود. بنابراین:

- دسترسی ناامن به ذخیره‌سازی ابر وجود خواهد داشت
- نقش های نامناسب وجود دارند
- داده‌های اعتباری بر روی منابع ذخیره زودگذر قرار دارند

از سوی دیگر ابر با حملات مبتنی بر کلمه عبور دست به گریبان خواهد بود، بنابراین نیاز به احراز هویت هایی که فراتر از استفاده از کلمه عبور می‌باشد.

آ۲. آسیب‌پذیری‌های تدارکاتی کاربران

- مشتری قادر به کنترل پروسه تدارک نیست.
- هویت مشتری در مرکز ثبت[4] تایید نشده است.
- تاخیر میان مولفه های ابر وجود دارد.
- چندین نسخه کپی از داده‌های هویت مشتریان ساخته شده است.
- داده‌های اعتباری مشتریان در مقابل استراق سمع و تکرار آسیب پذیرند.

آ۳. آسیب‌پذیری‌های قطع تدارک[5] کاربران

هنگامیکه قطع تدارک یک مشتری صورت می‌گیرد به خاطر تاخیرهای موجود در ابر تا مدتی داده‌های اعتباری او همچنان معتبر می‌باشند.

---

[1] Authentication
[2] Authorization
[3] Accounting
[4] Registration
[5] De-provisioning

آ۴. دسترسی از راه دور به واسط مدیریت

از نظر تئوری این نوع از دسترسی در ماشین‌های کاری آسیب‌پذیری‌هایی را بوجود می‌آورند، مانند احراز هویت ضعیف درخواست ها و پاسخ ها.

آ۵. آسیب‌پذیری‌های فوق‌ناظر

حملات لایه فوق‌ناظر بسیار جذاب می‌باشند: این لایه در حقیقت منابع فیزیکی و اجرای ماشین‌های مجازی را کنترل می‌کند، بنابراین هرگونه آسیب‌پذیری در این لایه بسیار حیاتی خواهد بود. از کار انداختن این لایه به معنای از کار انداختن همه ماشین‌های مجازی می‌باشد. کارهای انجام شده درباره آسیب‌پذیری‌های این لایه در [۱۳و۱۴و۱۵] آمده است. برای دیدن مثالهای بیشتر درباره این آسیب‌پذیری به [۱۶] مراجعه شود.

آ۶. فقدان جداسازی منبع

منبع استفاده شده توسط یک مشتری می‌تواند منبع مورد استفاده مشتری دیگر را تحت تاثیر قرار دهد.
زیرساخت‌های رایانش ابری IaaS بر مبنای طراحی های معماری هستند که در آنها منابع فیزیکی توسط چند ماشین مجازی به اشتراک گذاشته می‌شوند و بنابراین چندین مشتری به صورت اشتراکی از آنها استفاده می‌کنند.
آسیب‌پذیری‌های لایه فوق‌ناظر ممکن است موجب دسترسی غیر مجاز به این منابع مشترک شود. برای مثال ماشین‌های مجازی مشتری۱ و مشتری۲ دارای درایوهای هارد مجازی بر روی LUN [1] مشترک باشند. مشتری۲ شاید قادر به نگاشت کردن درایو هارد مشتری۱ بر روی ماشین مجازی خود بوده و بتواند داده‌های درون آن را ببیند.
در ابرهای IaaS از فوق‌ناظرهایی استفاده می‌شود که دارای API های کاملی برای تامین مدیریت مشتریان می‌باشند. آسیب‌پذیری‌ها در این زمینه می‌تواند منجر به دسترسی غیر مجاز به داده‌ها شود. حتی ممکن است حمله کنندگان قادر باشند که دارایی های درون ابر را تغییر دهند. برای مثال‌های بیشتر در این زمینه آسیب‌پذیری‌ها به [۱۶] مراجعه شود.

آ۷. فقدان جداسازی های شهرتی

فعالیت های یک مشتری بر شهرت مشتری دیگر تاثیر می گذارد.

آ۸. آسیب‌پذیری‌های رمزنگاری ارتباط

این آسیب‌پذیری‌ها در زمره نگرانی هایی درباره خوانده شدن داده‌ها در هنگام جابجایی می‌باشند.

[1] Logical Unit Number

آ۹. فقدان یا ضعف در رمزنگاری آرشیو ها و داده در انتقال

شکست در رمزنگاری داده در حال انتقال، داده‌های نگهداری شده در آرشیوها و پایگاه‌های داده، تصویرهای ماشین‌های مجازی، داده‌های قانونی و گزارش‌های حساس موجب می‌شود که داده‌های دیگر در ابر در معرض خطر قرار بگیرند. طبعا هزینه‌های پیاده‌سازی مدیریت کلید (آ۱۱) و هزینه‌های پردازش باید در نظر گرفته شده و با هزینه‌های این ریسک مقایسه شوند.

آ۱۰. غیر ممکن بودن پردازش داده‌ها در حالت رمز شده

رمزنگاری داده به طور کلی کار سختی نیست، ولی با توجه به یافته های اخیر رمزنگاری همریختانه [۱۷]، سیستم‌های تجاری از رمزنگاری در حین پردازش داده استفاده نخواهند کرد. در تخمینی که توسط Bruce Schneier انجام شده این نتیجه بدست آمده که جستجوی وب اگر بر روی داده‌های رمز شده انجام شود، زمان محاسبه حدودا یک میلیارد برابر می‌شود [۱۸]. این موضوع به این معناست که مادامیکه مشتریان ابر به کارهایی غیر از ذخیره داده احتیاج دارند باید به تامین کننده ابر اعتماد کنند.

آ۱۱. روش های ضعیف مدیریت کلید

زیرساخت‌های رایانش ابری نیازمند مدیریت و ذخیره‌سازی انواع مختلف کلید می‌باشد؛ برای نمونه می‌توان کلید نشست را برای پشتیبانی از انتقال داده (کلیدهای SSL)، کلیدهای رمزکردن فایل، جفت کلیدهای مشخص کننده تامین کننده ابر، جفت کلیدهای مشخص کننده مشتریان و توکن‌های مجازشناسی را نام برد [۱۹]. از آنجا که ماشین‌های مجازی دارای زیر ساخت سخت‌افزاری ثابتی نیستند و ابر مبتنی بر محتوی نیز از نظر جغرافیایی توزیع شده می‌باشد بکاربردن کنترل‌های استاندارد مانند HSM[1] مشکل‌تر می‌باشد. اطلاعات بیشتر در [۲۰و۲۱و۲۲].

آ۱۲. تولید کلید: آنتروپی پایین برای تولید اعداد تصادفی

ترکیب تصاویر استاندارد سیستم ، فناوری‌های مجازی‌سازی و کمبود وسایل ورودی به این معناست که سیستم‌ها دارای آنتروپی پایین‌تری نسبت به RNG[2] های فیزیکی هستند؛ مراجعه شود به امنیت رایانش ابری [۲۳]. این بدین معناست که یک حمله کننده به ماشین مجازی ممکن است به خاطر تشابه آنتروپی های مورد استفاده برای اعداد تصادفی قادر به حدس زدن کلیدهای تولید شده توسط ماشین مجازی باشد. این آسیب‌پذیری به آسانی قابل حل می‌باشد ولی اگر در گام طراحی مد نظر قرار نگیرد پیامدهای مهمی خواهد داشت.

[1] Hardware Security Module
[2] Random Number Generator

آ۱۳. کمبود فناوری‌ها و راه‌حل‌های استاندارد

کمبود استاندارد به این معناست که در برخی مواقع داده وابسته به تامین کننده می‌باشد. اینکه قطع عملیات را تامین کننده باید انجام دهد ریسک بسیار بزرگی است.

آ۱۴. عدم وجود توافق اجرایی منبع[1]

نبود توافق اجرای منبع به این معناست که اگر تامین کننده PaaS یا SaaS برشکست شود، مشتریانش هیچ پشتیبانی نخواهند داشت.

آ۱۵. مدل‌های نادقیق استفاده از منبع

سرویس‌های گرید به طور خاص در مقابل فرسودگی منبع آسیب پذیر می‌باشد زیرا آنها به طور آماری مورد استفاده قرار می‌گیرند. همچنین بسیاری از تامین کننده ها اجازه رزرو منابع را به صورت پیشرفته به مشتریان می‌دهند. این الگوریتم در این مواقع می‌توانند با شکست مواجه شوند:

- مدلسازی نادقیق استفاده از منبع، که می‌تواند منجر به رزرو اضافی یا استفاده بیش از حد شود (۳۴)، (۳۵) و (۳۶).
- خرابی الگوریتم های تخصیص منابع بعلت اتفاقات غیرعادی
- خرابی الگوریتم های تخصیص منابع که از دسته بندی کارها استفاده می‌کنند به خاطر عدم دسته بندی مناسب.

آ۱۶. عدم وجود کنترل بر روی پروسه تشخیص آسیب‌پذیری

محدودیت‌هایی که به واسطه شرایط استفاده در زمینه های اسکن کردن پورت ها و تست های آسیب‌پذیری بوجود می‌آید باعث می‌شود که مسئولیت برخی از اقدامات امن سازی بر عهده مشتری باشد که این امر خود یک آسیب‌پذیری جدی است.

آ۱۷. احتمال رخداد وارسی های داخل شبکه ای

مشتریان ابر می‌توانند بررسی های پورت و دیگر تست های استاندارد را بر روی دیگر مشتریان روی شبکه داخلی اجرا کنند.

آ۱۸. احتمال اجرای چک های co-residence

حملات side attack موجب می‌شوند که حمله کنندگان قادر باشند مشتری ای که منابع را به اشتراک می گذارد شناسایی کنند.

---

[1] No source escrow agreement

آ۱۹. فقدان آمادگی قانونی

علی رغم اینکه ابر پتانسیل ارتقای آمادگی قانونی را دارا می‌باشد، بسیاری از تامین کننده ها سرویس‌ها و شرایط استفاده لازم برای این امر را پدید نمی‌آورند. برای مثال، تامین کننده های SaaS امکان دسترسی به گزارش‌های IP مشتریان را فراهم نمی‌کنند. تامین کننده های IaaS هم ممکن است سرویس‌های قانونی مانند تصویرهای دیسک را فراهم نکنند.

آ۲۰. قابل قبول شدن رسانه های حساس[1]

مالکیت مشترک منابع ذخیره‌سازی فیزیکی به این معنی خواهد بود که داده‌های حساس ممکن است فاش شوند. این فاش شدن به این علت است که ممکن است خط و مشی های نابود کردن داده که در انتهای چرخه حیات مورد استفاده قرار می‌گیرند قابل پیاده‌سازی نباشند، برای نمونه به خاطر اینکه دیسک توسط مالک دیگری مورد استفاده است، یک فایل را نتوان به طور فیزیکی از بین برد.

آ۲۱. همگام سازی مسئولیت ها یا تعهدات پیمانی خارج از ابر[2]

مشتریان ابر معمولا از مسئولیت هایی که در شرایط استفاده به آنها داده شده است آگاه نیستند. تمایلاتی وجود دارد تا مسئولیت های فعالیت هایی که بی کس مانده مانند بایگانی رمزنگاری در تامین کننده ابر را واگذار کنند.

آ۲۲. برنامه‌های کاربردی میان ابری موجب بوجود آمدن وابستگی های پنهان می‌شوند[3]

در زنجیره تامین سرویس‌های ابر وابستگی های پنهان وجود دارند، و هنگامیکه شخص ثالث در عملیات ابر وارد شود معماری تامین کننده ابر از عملیات پشتیبانی نمی‌کند

آ۲۳. عبارات SLA با قول‌های متداخل برای ذی نفعان مختلف

این عبارات در توافقنامه های سطح سرویس حتی ممکن است با دیگر عبارات موجود یا با عبارات دیگر توافقنامه در تناقض باشند

آ۲۴. عبارات SLA شامل ریسک‌های زیادی برای کسب و کار باشند

توافقنامه های سطح سرویس ممکن است ریسک کسب و کار زیادی برای تامین کننده داشته باشند. از دیدگاه مشتری، SLA ها ممکن است عباراتی داشته باشند که تعابیر زیان آور از آنها حاصل شود- برای مثال ممکن است عبارتی وجود داشته باشد که از آن بتوان این نتیجه را گرفت که تامین کننده ابر (CP) حق نگهداری و دسترسی به هر محتوی ذخیره شده بر روی زیر ساخت ابر را دارد.

---

[1] Sensitive media sanitization
[2] Synchronizing responsibilities or contractual obligations external to cloud
[3] Cross-cloud applications creating hidden dependency

آ۲۵. حسابرسی و تصدیق مشتریان امکان‌پذیر نباشد

تامین کننده ابر نتواند هیچ تضمینی به مشتری در زمینه حسابرسی بدهد.

برای نمونه برخی از تامین کننده‌های ابر از فوق‌ناظرهای متن باز استفاده می‌کنند [۲۴] که در آنها تصدیق‌های معمول وجود ندارد [۲۵] که وجود آنها یک نیازمندی پایه‌ای برای سازمان‌ها می‌باشد. توجه به این نکته ضروری است که بدانیم که هیچ تقارن مستقیمی ما بین تصدیق‌های صورت گرفته و سطح آسیب‌پذیری وجود ندارد.

آ۲۶. طرح های تصدیق با زیرساخت ابر جور نباشد

کنترل خاصی برای ابر وجود ندارد که بدین معنی باشد که آسیب‌پذیری‌های امنیتی احتمالا از بین رفته اند.

آ۲۷. سرمایه گذاری ناکافی بر روی زیرساخت‌ها

سرمایه‌گذاری بر روی زیرساخت‌ها نیازمند زمان می‌باشد. اگر مدل‌های پیش بینی آن اشتباه از آب در آیند موجب مختل شدن سرویس برای مدت طولانی می‌شود.

آ۲۸. عدم وجود خط و مشی بستن منابع [1]

اگر راه‌حل قابل انعطاف و تطبیق پذیری برای مشتری و تامین کننده ابر در محدودیت گذاری بر روی منابع وجود نداشته باشد، استفاده غیر قابل پیش بینی از منابع مشکل ساز خواهد شد.

آ۲۹. ذخیره‌سازی داده در چندین حوزه و عدم شفافیت در این باره

mirror کردن داده‌ها و افزونگی در ذخیره‌سازی بدون اطلاع مشتری از محل ذخیره شدن داده‌ها موجب بوجود آمدن سطح بالایی از آسیب‌پذیری می‌شود. کمپانی‌ها ممکن است از توافقات عدول کنند، مخصوصا وقتی اطلاعات روشنی درباره حوزه‌های ذخیره‌سازی وجود نداشته باشد.

آ۳۰. فقدان اطلاعات درباره حوزه ها

داده‌ها ممکن است در حوزه هایی نگهداری و/یا پردازش شوند که بسیار آسیب پذیر هستند. اگر اطلاعات لازم در اینباره برای مشتریان ابر وجود نداشته باشد، مشتری قادر به اجتناب از آنها نخواهد بود.

آ۳۱. فقدان تکامل و شفافیت شرایط استفاده

آسیب‌پذیری‌هایی که مختص ابر نیستند

[1] Resource capping

در بخش تحلیل ریسک‌ها به این نکته اشاره شد که برخی ریسک‌ها مختص محیط ابر نیستند. در این بخش هم آسیب‌پذیری‌هایی که فقط مربوط به محیط ابر نیستند ذکر می‌شود، این آسیب‌پذیری‌ها حتما باید در پیاده‌سازی سیستم‌های مبتنی بر ابر در نظر گرفته شوند.

آ۳۲. فقدان آگاهی امنیتی

مشتریان وقتیکه به ابر مهاجرت می‌کنند نسبت به خطرات موجود آگاهی لازم را ندارند، مخصوصا آن ریسک‌هایی که مختص به محیط ابر می‌باشد. این عدم آگاهی ممکن است در تامین کننده هم از جهت ناآگاه بودن از روش‌های کاهش خطا وجود داشته باشد.

آ۳۳. فقدان پروسه های بررسی

از آنجایی که نقش های حساسی در محیط ابر برای تامین کننده وجود دارند، در این مقیاس اگر بررسی های لازم بر روی آن نقش ها انجام نشود خسارت هایی را به بار می آورد.

آ۳۴. نقش ها و مسئولیت های گنگ

این آسیب‌پذیری‌ها مربوط به عدم مشخص شدن کافی نقش ها و مسئولیت ها در سازمان تامین کننده ابر می‌باشد.

آ۳۵. اجرای ضعیف تعاریف نقش ها

در تامین کننده های ابر عدم تفکیک مناسب نقش ها موجب بوجود آمدن نقش هایی می‌شود که از امتیازات بالایی برخوردار هستن و این امر موجب آسیب‌پذیری سیستم می‌شود. مثلا هیچ فردی نباید دسترسی کامل به تمام بخش‌های ابر را داشته باشد.

آ۳۶. بکارنرفتن اصل دانستن با توجه به نیاز

این آسیب‌پذیری به نقش ها و مسئولیت ها مربوط می‌شود. افراد نباید به چیزهایی که نیاز ندارند دسترسی داشته باشند.

آ۳۷. ناکافی بودن رویه های فیزیکی امنیتی

مشتمل بر:

- فقدان کنترل های فیزیکی محیط (استفاده از کارتهای هوشمند برای احراز هویت).
- فقدان استفاده از پوشش های الکترومغناطیسی برای دارایی هایی که در مقابل استراق سمع آسیب پذیر هستند.

آ۳۸. پیکره بندی نامناسب

این دسته از آسیب‌پذیری‌ها مشتمل است بر: استفاده ناکافی از رویه های مقاوم‌سازی، خطاهای انسانی و مدیران دوره ندیده.

آ۳۹. آسیب‌پذیری‌های سیستم یا سیستم عامل

آ۴۰. نرم‌افزار غیرقابل اطمینان[1]

آ۴۱. فقدان یا ضعف در پیوستگی کسب و کار و طرح های بازیابی بد

آ۴۲. فقدان یا ناکامل بودن موجودی دارایی ها

آ۴۳. فقدان یا ناکامل بودن دسته بندی دارایی ها

آ۴۴. مالکیت نامشخص دارایی ها

آ۴۵. شناسایی ضعیف نیازمندی های پروژه

این مورد شامل فقدان در نظر گرفتن نیازمندی های امنیتی، نقش سیستم‌ها و کاربردهای کاربران و نیازمندی های ناکامل کسب و کار و .. می‌باشد.

آ۴۶. انتخاب ضعیف تامین کننده

آ۴۷. فقدان افزونگی تامین کننده

آ۴۸. آسیب‌پذیری‌های برنامه کاربردی یا مدیریت ضعیف وصله

این دسته از آسیب‌پذیری‌ها شامل: باگ های کدهای برنامه کاربردی، رویه های وصله کردن متداخل میان مشتری و تامین کننده، بکاربردن وصله‌های تست نشده، آسیب‌پذیری‌ها در مرورگرها و ... .

آ۴۹. آسیب‌پذیری‌های مصرف منابع [2]

آ۵۰. نفوذ به NDA[3] بوسیله تامین کننده

---

[1] Untrusted software
[2] Resource consumption vulnerabilities
[3] Non-disclosure agreement

آ۵۱. احتمال از دست دادن داده

آ۵۲. فقدان یا ضعف در رویه های گزارش گیری و نگهداری

آ۵۳. فیلترینگ ناکافی منابع

۹-۹- دارایی ها

دارایی	توضیح یا ارجاع به عناصر توضیح داده شده در بالا	مالک [بازیگر یا سازمان درگیر]	مقدار ملاحظه شده [خیلی پایین – پایین – متوسط – بالا – خیلی بالا]
د۱. شهرت کمپانی		CC[1]	خیلی بالا
د۲. اعتماد مشتری	حسن نیت را هم شامل می‌شود، می‌توان با شکایات آنرا اندازه گرفت	CC	خیلی بالا
د۳. وفاداری و تجربه کارکنان		CC	بالا
د۴. دارایی فکری[2]		CC	بالا
د۵. داده‌های شخصی حساس	(مانند آنچه در European data protection directive)	CC / CP[3]	خیلی بالا
د۶. داده شخصی	(مانند آنچه در European data protection directive)	CC / CP	متوسط (ارزش عملیاتی) / بالا (در صورت گم شدن)
د۷. داده حیاتی شخصی	(تمام داده‌هایی که در دسته داده‌های شخصی در European data protection directive قرار دارد و در سازمان بعنوان حیاتی مارک شده)	CC / CP	بالا (ارزش عملیاتی) / بالا(در صورت گم شدن)
د۸. داده HR	داده مربوط به دیدگاه عملیاتی	CC	بالا
د۹. تحویل سرویس – سرویس‌های بلادرنگ	تمام سرویس‌هایی که زمان در آن بسیار اهمیت دارد و سطح دسترس پذیری آنها نزدیک به ۱۰۰٪ می‌باشد	CC / CP	خیلی بالا
د۱۰. تحویل سرویس		CC / CP	متوسط
د۱۱. کنترل دسترسی / احراز هویت / مجازشناسی (admin vs others)		CC / CP	بالا
د۱۲. اعتبارنامه	برای مراجعین و افرادی که به سیستم دسترسی دارند	CC	خیلی بالا

[1] Cloud customer
[2] Intellectual property
[3] Cloud provider

د۱۳. دایرکتوری کاربر (داده)	اگر از کار بیفتد دیگر کسی قادر به وارد شدن به سیستم نیست	CC	بالا
د۱۴. واسط مدیریت سرویس ابر	واسط مدیریت می‌باشد (چه مبتنی بر وب چه مبتنی بر شل از راه دور[1]) که تمامی سرویس‌هایی که توسط ابر ایجاد می‌شود با آن مدیریت می‌شود.	CP / CC	خیلی بالا
د۱۵. API های واسط مدیریت		CP / CC / EuropeanHealth	متوسط
د۱۶. شبکه (ارتباطات و غیره)	شامل ارتباطات درون و بیرون از ابر می‌شود	CP / CC	بالا
د۱۷.سخت‌افزار فیزیکی		CP / CC	پایین (وابسته به اینکه چقدر از آن را از دست بدهید) / متوسط (در صورتیکه دزدیده شود و از آن پشتیبانی نشود)
د۱۸. ساختمان های فیزیکی		CC / CP	بالا
د۱۹. برنامه کاربردی CP (کد اصلی)		CP / CC	بالا
د۲۰. تصدیق	ISO, PCI, DSS, etc	CC / CP	بالا
د۲۱. گزارش‌های عملیاتی (مشتری و تامین کننده ابر)	این گزارش‌ها برای تقویت و بهینه کردن پروسه های کسب و کار استفاده شده و اهداف حسابرسی را تسهیل می‌کنند	CC / CP	متوسط
د۲۲. گزارش‌های امنیتی	بعنوان شواهد نفوذ و موارد قانونی مورد استفاده هستند	CC / CP	متوسط
د۲۳. پشتیبان داده یا بایگانی داده		CP / CC	متوسط

[1] Remote shell

۹-۱۰- چارچوب تضمین اطلاعات

یکی از مهمترین توصیه های این گزارش مجموعه‌ای از معیارهای تضمین است که برای موارد زیر طراحی شده است:

- برای دسترسی به ریسک اتخاذ شده از سرویس‌های ابر
- برای مقایسه بین پیشنهادات تامین کننده های ابر متفاوت
- امنیتی موثر برای تامین کننده های شخص ثالث کاری است که بدون داشتن تجربه و تخصص لازم بسیار مشکل است.
- برای کاهش بار تضمین روی تامین کننده های ابر. یک از مهم ترین ریسک‌هایی که برای زیر ساخت ابر بیان می‌شود مربوط به نیازمندیهای تضمین NIS[1] می‌باشد. تامین کننده های ابر فهمیده‌اند که تعداد زیادی از مشتریان درخواست حسابرسی از زیرساخت‌ها و خط‌مشی‌هایشان را دارند. این موضوع بار امنیتی بزرگی را بوجود می آورد و موجب افزایش خطر حملات به زیر ساخت ها می‌شود. تامین کننده ها نیاز به تبیین یک چارچوب برای مقابله با این موضوع دارند.

این بخش از توصیه‌ها پرسش ایی را فراهم می‌کند که سازمان می‌تواند از تامین کننده ابر داشته باشد تا از حمایت اطلاعات خود مطمئن شوند. هدف این پرسش ها ایجاد یک baseline حداقل می‌باشد. بنابراین هر سازمان ممکن است نیازمندی هایی اضافه تر از این baseline داشته باشد. به هر صورت این پاسخ ها باید در چاچوب معنایی مناسبی قرار بگیرند تا بتوان از آن پاسخ های پایدارتر و دارای قابلیت مقایسه بدست آورد. برای داشتن چنین پاسخ هایی طبعا نیاز به معیارهای قابل سنجش می‌باشد. این معیارهای فوق الذکر باید برای همه تامین کننده ها یکسان باشند تا قیاس بین آنها را در سازمان‌ها ممکن باشد.

۹-۱۰-۱- تقسیم مسئولیت ها

با توجه به وقایع امنیتی که ممکن است رخ بدهد نیاز به تعریف روشن و فهم مسئولیت ها و نقش های مربوط به امنیت در مشتری و تامین کننده وجود دارد. خطوط این تقسیم‌بندی به طور گسترده ای بین SaaS و Iaas تفاوت دارد. در جدول زیر معمول ترین و منطقی ترین این تقسیم‌بندی نشان داده شده است. در هر مورد باید مشخص شود که کدامیک از سرویس، مشتری یا تامین کننده مسئولیت را بر عهده دارند. اگر شرایط استفاده استانداردی موجود باشد در آن مشخص شده که مشتری باید از تمامی مسئولیت هایش آگاه باشد.

[1] Network and information security

Software as a service-۱-۱-۱۰-۹

مشتری	تامین کننده
- قبول قوانین حمایت داده برای داده‌های جمع آوری شده و پردازش شده توسط مشتری - نگهداری سیستم مدیریت هویت - مدیریت سیستم مدیریت هویت - مدیریت سکوی احراز هویت	- زیر ساخت پشتیبانی فیزیکی (facilities, rack space, power, cooling, cabling, etc) - امنیت و دسترس پذیری زیرساخت فیزیکی (سرور، ذخیره‌سازی، پهنای باند و غیره) - مدیریت وصله سیستم عامل و رویه های مقاوم‌سازی (چک کردن هرگونه تداخل بین مقاوم‌سازی انجام شده توسط مشتری با خط مشی امنیتی) - پیکره بندی سکوی امنیتی (firewall rules, IDS/IPS tuning, etc) - مانیتورینگ سیستم‌ها - نگهداری سکوی امنیتی (firewall, Host IDS/IPS, antivirus, packet filtering) - جمع آوری گزارش و مانیتوریگ امنیتی

Platform as a Service -۲-۱-۱۰-۹

مشتری	تامین کننده
- نگهداری سیستم مدیریت هویت - مدیریت سیستم مدیریت هویت - مدیریت سکوی احراز هویت	- زیر ساخت پشتیبانی فیزیکی (facilities, rack space, power, cooling, cabling, etc) - امنیت و دسترس پذیری زیرساخت فیزیکی (سرور، ذخیره‌سازی، پهنای باند و غیره) - مدیریت وصله سیستم عامل و رویه های مقاوم‌سازی (چک کردن هرگونه تداخل بین مقاوم‌سازی انجام شده توسط مشتری با خط مشی امنیتی) - پیکره بندی سکوی امنیتی (firewall rules, IDS/IPS tuning, etc) - مانیتورینگ سیستم - نگهداری سکوی امنیتی (firewall, host IDS/IPS, antivirus, packet filtering) - جمع آوری گزارش و مانیتوریگ امنیتی

۹-۱۰-۱-۳- Infrastructure as a service

مشتری	تامین کننده
- نگهداری سیستم مدیریت هویت - مدیریت سیستم مدیریت هویت - مدیریت سکوی احراز هویت - مدیریت وصله‌های سیستم عامل مهمان و روند های مقاوم‌سازی - پیکره بندی سکوی امنیتی مهمان (firewall rules, IDS/IPS tuning, etc) - مانیتورینگ سیستم‌های مهمان - نگهداری سکوی امنیتی (firewall, host IDS/IPS, antivirus, packet filterig) - جمع آوری گزارش و مانیتوریگ امنیتی	- زیر ساخت پشتیبانی فیزیکی (facilities, rack space, power, cooling, cabling, etc) - امنیت و دسترس پذیری زیرساخت فیزیکی (سرور، ذخیره‌سازی، پهنای باند و غیره) - سیستم‌های میزبان (hypervisor, virtual firewall, etc)

وقتی مشتریان مسئول امنیت زیرساخت‌هایشان هستند (مانند IaaS) این موارد را باید در نظر بگیرند:

۹-۱۰-۱-۴- امنیت برنامه کاربردی در IaaS

تامین کننده های برنامه کاربردی IaaS با برنامه کاربردی مانند یک عنصر جعبه سیاه برخورد می‌کنند و بنابراین و کاملا نگرانی های مدیریتی و عملیاتی برنامه‌های کاربردی مشتریان را رها کرده‌اند. تمام لایه‌های برنامه بر روی سرور مشتری اجرا می‌شود و بوسیله خود مشتریان مدیریت می‌شود. به همین دلیل مشتریان باید مسئولیت کامل امنیت برنامه‌هایشان را بر عهده داشته باشند. در اینجا یک خلاصه چک لیست و توضیحات برای بهترین اقداماتی که مشتریان می‌توانند انجام دهند آورده شده:

- برنامه‌های ابر باید برای مدل‌های کاری مبتنی بر اینترنت بوجود آمده باشند.
- باید طراحی آنها به گونه ای باشد که در مقابل آسیب‌پذیری‌های معمول وب حفاظت لازم را داشته باشند[۲۶].
- مشتریان مسئول بروزرسانی و ارتقای برنامه‌های خود هستند بنابراین باید از داشتن یک استراتژی وصله اطمینان حاصل شود.
- مشتریان نباید وسوسه شوند که از پیاده‌سازی های شخصی شده احراز هویت، مجازشناسی و حسابرسی (AAA) استفاده کنند، زیرا ممکن است این پیاده‌سازی ها ضعف هایی داشته باشند.

به طور خلاصه: برنامه‌های کاربردی بنگاه باید به همراه کنترل های زیادی اجرا شوند تا امنیت میزبان و شبکه، دسترسی کاربر، و امنیت کنترل‌های برنامه کاربردی را تامین کنند [۲۷]. برای اطلاعات بیشتر می‌توانید به یادداشت های امنیتی محصولات تولیدکننده های بزرگ مانند مایکروسافت، اوراکل، سان و غیره مراجعه نمایید.

۹-۱۱- نیازمندی های تضمین اطلاعات

۹-۱۱-۱- امنیت پرسنل

اکثر سوالات مربوط به پرسنل همانهایی هستند که شما از پرسنل فناوری اطلاعات و افراد مربوط به فناوری اطلاعات می پرسید.

- رویه ها و خط مشی شما در استخدام مدیران فناوری اطلاعات و افرادی که به سیستم دسترسی دارند چیست؟ این‌ها باید شامل:
  - بررسی‌های قبل از استخدام (هویت، ملیت و وضعیت، تاریخچه کاری و مراجع آنها، سابقه مجرمانه و بررسی میدانی(برای کارکنان اولویت بالا))
- آیا خط مشی های متفاوت با توجه به محل نگهداری داده و برنامه‌های کاربردی مورد اجرا وجود دارد؟
  - مثلا خط مشی استخدام یک منطقه ممکن است با منطقه دیگر تفاوت داشته باشد
  - ممکن است داده حساسی در بخش خاصی با پرسنل خاص ذخیره شده باشد.
- برنامه آموزش امنیت شما برای کارمندان چیست؟
- آیا پروسه ارزیابی مداوم وجود دارد؟
  - هرچند وقت انجام می‌شود؟
  - مرورهای دوباره
  - بازبینی های دسترسی ها و سطوح اولویت
  - بازبینی خط مشی ها

۹-۱۱-۲- تضمین زنجیره تامین

این سوالات وقتی مطرح است که تامین کننده ابر برخی از عملیات امنیتی را به شخص ثالث محول می‌کند (مثلا استفاده از تامین کننده خارجی برای مدیریت هویت در سیستم‌های عامل). همچنین این سوالات شخص ثالث هایی که به زیرساخت تامین کننده ابر دسترسی دارند نیز شامل می‌شود. این فرض وجود دارد که این پرسشنامه به صورت تناوبی برای شخص ثالث هم بکار می‌رود.

- عملیات کلیدی امنیتی که در زنجیره تامین شما برونسپاری شده‌اند و با شما در ارتباطند را تعریف کنید.
- جزئیات روند های تضمین دسترسی شخص ثالث ها به زیر ساخت شما را شرح دهید.
  - آیا شما آنها را حسابرسی و بازرسی می‌کنید؟

- آیا SLA هایی با سطح پایین تر نسبت به آنچه به مشتریانتان پیشنهاد داده اید توسط برونسپاران گارانتی شده؟
- چه معیاری برای مشخص شدن تضمین سطح سرویس شخص ثالث وجود دارد؟
- آیا تامین کننده ابر با خط مشی امنیتی بکار رفته توسط شخص ثالث موافق است؟

۹-۱۱-۳- امنیت عملیات

انتظار می‌رود که هر توافق تجاری با تامین کننده های خارجی سطوح سرویس برای تمام سرویس‌های شبکه را شامل شود. علاوه بر این توافقات، مشتری نهایی باید مطمئن باشد که تامین کننده از کنترل های مناسب برای تضمین افشا نشدن غیرمجاز داده‌ها استفاده می‌کند.

- جزئیات تغییرات خط مشی و رویه های کنترلی را بگویید. این پاسخ باید شامل پروسه مورد استفاده در بازشناسی ریسک‌ها بعنوان نتایج تغییرات و شفاف کردن خروجی‌هاست.
- خط مشی دسترسی از راه دور را تعریف کنید.
- آیا تامین کننده از رویه های اجرایی مستند برای سیستم‌های اطلاعاتی استفاده می‌کند؟
- آیا محیط تعریف شده ای برای کاهش ریسک وجود دارد، برای مثال، توسعه، تست و محیطهای اجرایی از هم جدا هستند؟
- کنترل های میزبان و شبکه بکار رفته برای پشتیبانی از سیستم‌های میزبان و شبکه و اطلاعات مشتری را تعریف کنید. این‌ها باید شامل جزئیات گواهی ها هم باشد.
- کنترل های مورد استفاده برای پشتیبانی در برابر کدهای بدخواه را مشخص کنید.
- آیا پیکره بندی های امن برای مجازشناسی کدها استفاده شده است؟
- خط مشی و روندهای پشتیبان گیری را با جزئیات بیان کنید. این‌ها باید شامل جزئیات روند های مدیریت رسانه ها شود.

گزارش‌های حسابرسی برای اتفاقاتی که نیاز به بازرسی دارند مورد استفاده قرار می‌گیرد؛ برای عیب یابی هم می‌توان از آنها استفاده کرد. برای چنین اهدافی مشتری نیاز به دسترس بودن چنین اطلاعاتی را دارد:

- آیا تامین کننده می‌تواند جزئیات اطلاعات ضبط شده بر روی گزارش‌های حسابرسی را تعیین کند؟
  - تا چه مدت این اطلاعات باقی می مانند؟
  - آیا اطلاعات این گزارش‌ها را می‌توان تقسیم‌بندی کرد تا برای مشتریان نهایی هم قابل استفاده باشد؟
  - چه کنترل هایی برای حمایت گزارش‌ها از دسترسی غیر مجاز مورد نیاز است؟
  - چه متدی برای چک کردن صحت گزارش‌های حسابرسی مورد استفاده است؟
- چگونه گزارش‌های حسابرسی بازبینی می‌شود؟ چه رخدادهایی باعث حرکت‌های اتخاذ شده است؟

- این گزارش‌ها در چه زمانی و با چه دقتی تهیه می‌شود؟

۹-۱۱-۴- تضمین نرم‌افزار

- کنترل هایی که برای حمایت از صحت سیستم عامل و برنامه‌های کاربردی استفاده می‌شود را تعریف کنید. شامل همه استانداردهای مورد استفاده مانند، OWASP[۲۸]، SANS Checklist [۲۹]، SAFECode[۳۰].
- چگونه متوجه می شوید که نسخه های تازه وارد بازار شده نیازهای شما را بدون ریسک تامین می‌کند؟ آیا این‌ها قبل از استفاده بازبینی شده؟
- چه راهکارهایی برای امن نگه داشتن برنامه‌های کاربردی وجود دارد؟

۹-۱۱-۵- مدیریت وصله

- جزئیات رویه های مدیریت وصله مورد استفاده را فراهم کنید.
- آیا می‌توانید تضمین کنید که پروسه مدیریت وصله تمامی لایه های فناوری ابر را پوشش می‌دهد(مثلا شبکه(مولفه های زیرساخت، روتر و سوییچ)، سرور سیستم عامل، نرم‌افزار مجازی‌سازی، زیر سیستم‌های امنیتی و برنامه کاربردی(firewalls, antivirus gateway))؟

۹-۱۱-۶- کنترلهای معماری شبکه

- کنترلهای مورد استفاده برای مقابله با حملات DDoS را تعریف کنید.
  - عمقا مقابله کنید (deep packet analysis, traffic throttling)
  - آیا شما دفاع مناسب برای حملات داخلی مانند آنچا برای حملات خارجی وجود دارد دارید؟
- چه سطوحی از جداسازی استفاده می‌شود؟
  - برای ماشین‌های مجازی، ماشین‌های فیزیکی، شبکه، ذخیره‌سازی، مدیریت شبکه و غیره.
- آیا معماری ابر عملیات را هنگامیکه تامین کننده سرویس از کمپانی جدا می‌شود ادامه می‌دهد(آیا وابستگی به سیستم LDAP مشتری وجود دارد)؟
- آیا زیرساخت شبکه مجازی استفاده شده توسط تامین کننده ابر (معماری 802.1q [۳۱]) توسط استانداردها امن شده است(آیا با پیکره بندی های امنیتی خاص در مقابل حملاتی مانند MAC spoofing, ARP poisoning attacks امن شده)؟

۹-۱۱-۷- معماری میزبان

- آیا تامین کننده تضمین می‌کند که تصاویر ماشین‌ها مقاوم‌سازی شده‌اند؟
- آیا تصاویر مقاوم شده در برابر دسترسی غیر مجاز محافظت می‌شوند؟
- آیا تامین کننده می‌تواند تصویر مجازی‌سازی شده که احراز هویت نشده است را تایید کند؟
- آیا هاست می‌تواند با حداقل پورت های مورد نیاز کار کند؟

۹-۱۱-۸- PaaS – امنیت برنامه کاربردی

اگر بخواهیم بطور عمومی صحبت کنیم، تامین کننده سرویس PaaS مسئول امنیت لایه‌های نرم‌افزاری پلت‌فرم می‌باشد، و توصیه های این متن برای تامین کننده PaaS می‌تواند اصولی را نشان دهد که برای مدیریت و طراحی سکوی PaaS کمک کند. اینکه اطلاعات جزئیات امن سازی سکوها را از تامین کننده PaaS بدست آورد بسیار مشکل است – سوالاتی که در ادامه این متن می‌آید می‌تواند در این امر کمک کند.

- درخواست اطلاعات درباره اینکه چگونه برنامه‌های کاربردی چند مالکی از هم جدا می‌شوند – توضیح سطح بالا از معیارهای جداسازی مورد نیاز است.
- چه تضمین هایی تامین کننده PaaS می‌تواند برای محدود شدن دسترسی به داده شما ارائه دهد؟
- معماری سکوی شما باید از نوع معماری کلاسیک sandbox باشد – آیا تامین کننده ابر پلت‌فرم را برای آسیب‌پذیری‌ها و باگ ها مانیتور می‌کند؟
- تامین کننده PaaS باید قادر به پیشنهاد دادن برخی ویژگی‌های امنیتی باشد – آیا این‌ها شامل احراز هویت، مجازشناسی و غیره می‌باشد؟

۹-۱۱-۹- SaaS – امنیت برنامه کاربردی

مدل SaaS این ایده را دارد که تامین کننده مدیریت تمامی برنامه‌های کاربردی که به کاربر نهایی تحویل می‌شود را بر عهده دارد. بنابراین تامین کننده های SaaS مسئول تامین امنیت برنامه‌های کاربردی هستند. مشتریان مسئول پروسه های امنیتی عملیاتی هستند. با این حال سوالات زیر می‌تواند کمک باشد:

- چه کنترلهای مدیریتی فراهم شده و آیا این کنترلها را می‌توان برای اختصاص دادن امتیاز خواندن و نوشتن به کاربران استفاده کرد؟
- آیا کنترل دسترسی SaaS ریزدانه است و می‌تواند برای خط مشی سازمان‌ها سفارشی شود؟

۹-۱۱-۱۰- اشغال منابع

- در رویداد اضافه بار منبع (پردازش، حافظه، ذخیره‌سازی، شبکه):

- چه اطلاعات اضافه ای درباره امتیازات مربوط داده شده؟
- آیا تغییری در زمان انجام کار و نیازمندی ها ایجاد شده؟

- تا چه میزان می‌توانید گسترش مقیاس دهید؟ آیا تامین کننده می‌تواند این تضمین را بدهد که حداکثر منابع در حداقل زمان ارائه می‌شوند؟
- گسترش مقیاس تا چه حد سریع انجام می‌شود؟ آیا تامین کننده دسترس پذیری منابع در حداقل زمان را تضمین می‌کند؟
- چه پروسه هایی برای سروکار داشتن با فرآیندهای در مقیاس بزرگ وجود دارند؟

۹-۱۱-۱۱- مدیریت هویت و دسترسی

کنترل های زیر برای سیستم‌های مدیریت دسترسی و هویت توسط تامین کننده ابر مورد استفاده قرار می‌گیرند:

۹-۱۱-۱۱-۱- مجازشناسی

- آیا حساب کاربری‌ای وجود دارد که دارای امتیاز دسترسی به تمام سیستم ابر باشد؟ اگر بله، برای چه عملیاتی (خواندن/نوشتن/حذف)؟
- حساب های کاربری با سطح امتیاز بالا چگونه احراز هویت و مدیریت می‌شوند؟
- تصمیمات کلیدی چگونه مجازشناسی می‌شوند؟
- آیا نقش های با امتیاز بالا به یک شخص داده می‌شود؟
- آیا از کنترل دسترسی مبتنی بر نقش[1] استفاده می‌کنید؟

۹-۱۱-۱۱-۲- تدارک هویت

- چه چک هایی بر روی هویت حساب ها و ثبت نام ها انجام می‌شود؟ آیا از استاندارد خاصی استفاده می‌شود؟
- آیا سطوح مختلف چک کردن هویت بر اساس منابع مورد نیاز انجام می‌شود؟
- چه پروسه هایی برای حذف هویت وجود دارد؟

۹-۱۱-۱۱-۳- مدیریت داده شخصی

- چه کنترل های حمایتی و ذخیره‌سازی داده برای دایرکتوری کاربر (مثلا AD[2] و LDAP[3]) و دسترسی به آن استفاده می‌شود؟

---

[1] Role-based access control (RBAC)
[2] Active Directory
[3] Lightweight Directory Access Protocol

- آیا دایرکتوری کاربر قابل مرور می‌باشد؟
- آیا من نیاز به دسترسی پایه ای به داده مشتری توسط تامین کننده ابر دارم؟

۹-۱۱-۱۱-۴- مدیریت کلید

برای کلیدهای تحت کنترل تامین کننده ابر:

- آیا کنترل های امنیتی برای خواندن و نوشتن آن کلیدها وجود دارد؟ برای مثال، خط مشی های قوی کلمه عبور، ذخیره کلید در سیستم جدا، ماژول‌های امنیتی سخت‌افزاری برای کلیدهای اصلی گواهی ها، احراز هویت مبتنی بر کارت هوشمند و غیره.
- آیا کنترل های امنیتی برای بکاربردن کلیدها برای امضا و رمزنگاری داده وجود دارد؟
- آیا رویه هایی برای رخدادهای مقایسه کلید وجود دارد؟ مثلا لیست کلیدهای لغو شده.
- آیا لغو کلیدها می‌تواند با چند سایت به طور همزمان کار کند؟
- آیا تصاویر سیستم مشتری پشتیبانی و رمز شده است؟

۹-۱۱-۱۱-۵- رمزنگاری

- رمزنگاری در چندین مکان می‌تواند انجام شود، جایی که:
  - داده در حال انتقال است
  - داده در حال استراحت است
  - داده در پردازشگر یا حافظه است
- نام کاربری و کلمه عبور؟
- آیا خط مشی خوش تعریفی برای اینکه آیا داده باید رمز شود یا خیر وجود دارد؟
- چه کسی کلیدهای دسترسی را در اختیار دارد؟
- کلیدها چگونه حفاظت می‌شوند؟

۹-۱۱-۱۱-۶- احراز هویت

- چه فرم هایی از احراز هویت برای عملیاتی که نیاز به تضمین بالا دارند استفاده می‌شود؟
- آیا احراز هویت دو فاکتوره برای مدیریت مولفه های حیاتی استفاده می‌شود؟

۹-۱۱-۱۱-۷- به خطر افتادن یا دزدی اعتبارها

- آیا مکانیزم های کشف انحراف وجود دارد؟ مانند چند ورودی[1] و ورود در زمان غیر معمول و ... .
- چه اقداماتی در صورت دزدی یا خرابی انجام می‌شود؟

[1] Multiple login

۹-۱۱-۸-چارچوب های مدیریت هویت

- آیا سیستم، اجازه استفاده از زیر ساخت IDM[1] که قابل تعامل با سیستم‌های با تضمین بالا و پایین می‌باشد را می‌دهد؟
- آیا تامین کننده ابر قابل تعامل با تامین کننده های شخص ثالث هویت است؟
- آیا قابلیت استفاده از single sign on وجود دارد؟

۹-۱۱-۹- کنترل دسترسی

- آیا سیستم اعتبار مشتریان اجازه تفکیک نقش ها و مسئولیت ها برای دامنه های مختلف را می‌دهد؟
- دسترسی به تصاویر سیستم مشتری چگونه مدیریت می‌شود؟

۹-۱۱-۱۰-احراز هویت

- تامین کننده ابر چگونه خود را برای مشتری تعیین هویت می‌کند؟
  - وقتی که مشتری فرمان های API را می فرستد؟
  - وقتی که گزارش مشتری وارد واسط مدیریت می‌شود؟
- آیا شما از مکانیزم فدرال برای احراز هویت پشتیبانی می‌کنید؟

۹-۱۱-۱۲-مدیریت دارایی

این نکته که تامین کننده لیستی از دارایی های سخت‌افزاری و نرم‌افزاری زیر کنترلش داشته باشد مهم می‌باشد. این لیست برای چک کردن اینکه تمام سیستم از کنترل مشخصی و درستی بهره می برد استفاده می‌شود، و از آن نمی‌توان استفاده غیر مجاز کرد.

- آیا تامین کننده از انبارداری اتوماتیک برای دارایی ها استفاده می‌کند؟
- آیا لیستی از دارایی ها که توسط مشتری ای در یک بازه زمانی استفاده شده وجود دارد؟

سوالات زیر در جایی که مشتری نهایی ار داده‌ها را گسترش می‌دهد استفاده می‌شود.

- آیا داده‌ها در شرایط حساس استفاده می‌شوند؟
  - اگر بله، آیا تامین کننده در تفکیک درست داده‌ها کمک می‌کند؟

۹-۱۱-۱۳- قابلیت حمل داده و سرویس‌ها

برای فهم ریسک‌های ناشی از vendor lock in این سوالات باید پاسخ داده شود.

- آیا رویه ها و API های مستندی برای خارج کردن داده از ابر وجود دارد؟

[1] Identity Management

- آیا فروشنده فرمت های متعارفی را برای خارج کردن داده استفاده می‌کند؟
- در مورد SaaS آیا واسط‌های API استاندارد استفاده شده؟
- آیا پروسه های تست برای استخراج داده از ابر وجود دارد؟
- آیا کاربر می‌تواند به تنهایی خروج داده را اجرا کرده و عملیات را تایید کند؟

۹-۱۱-۱۴-مدیریت دوام کسب و کار[1]

فراهم آوردن دوام برای یک سازمان دارای اهمیت می‌باشد.

- آیا تامین کننده متد مستندی که جزئیات تاثیر شکست را بیان می‌کند نگهداری می نماید؟
  - چه RPO[2] و RTO[3] هایی برای سرویس‌ها وجود دراد؟ با جزئیات.
  - آیا فعالیت های امنیت اطلاعات به پروسه استرداد آدرس شده‌اند؟
  - راه های ارتباط با مشتری نهایی ها چیست؟
  - آیا نقش ها و مسئولیت های تیم مشخص است؟
- آیا تامین کننده اولویت ها برای بازیابی را دسته بندی کرده است؟
- در هنگام در دسترس نبودن سایت اصلی حداقل فاصله از سایت ثانوی چقدر است؟

۹-۱۱-۱۵-مدیریت واقعه و پاسخ

مدیریت واقعه و پاسخ بخشی از مدیریت دوام کسب و کار می‌باشد. هدف این پروسه مشخص کردن تاثیر رخداد های غیرمنتظره و بالقوه نابود کننده و رساندن این تاثیر به یک سطح قابل پذیرش است.

برای ارزیابی قابلیت یک سازمان در حداقل کردن امکان اتفاق افتادن رویدادها و کم کردن تاثیر آنها سوالات زیر باید پرسیده شود:

- آیا تامین کننده پروسه رسمی برای کشف، شناسایی، تحلیل و پاسخ به رویدادها دارد؟
- قابلیت‌های کشف چگونه ساختار داده شده‌اند؟
  - مشتری ابر چگونه چیزهای غیرعادی را و رخداد های امنیتی را به تامین کننده گزارش می‌دهد؟
  - چه تسهیل کننده هایی تامین کننده در اختیار قرار می‌دهد؟
  - آیا سرویس مانیتورینگ بلادرنگ امنیتی[4] وجود دارد؟ آیا این سرویس برونسپاری شده است؟
  - گزارش‌های امنیتی تا چه مدت زمانی نگهداری می‌شوند؟

---

[1] Business Continuity Management
[2] Recovery Point Objective
[3] Recovery Time Objective
[4] Real Time Security Monitoring (RTSM)

- سطوح سخت گیری چگونه تعریف شده‌اند؟
- پروسه های تعدیل چگونه تعریف شده‌اند؟
- وقایع چگونه مستند شده و شواهد چگونه نگهداری می‌شود؟
- علاوه بر احراز هویت، مجازشناسی و حسابرسی چه کنترل های دیگری انجام می‌شود؟
- آیا تامین کننده تصویری قانونی از ماشین مجازی را به مشتری پیشنهاد می‌دهد؟
- تامین کننده هر چند وقت بازیابی از سوانح را انجام می‌دهد؟
- آیا تامین کننده داده‌های مربوط به سطح رضایت از SLA ها را جمع آوری می‌کند؟
- آیا تامین کننده تست های آسیب‌پذیری را انجام می‌دهد؟ هر چند وقت؟

۹-۱۱-۱۶-امنیت فیزیکی

موضوعات بالقوه ای برای زیر ساخت های فناوری اطلاعات که زیر نظر شخص ثالث هستند بوجود می‌آید – مانند برونسپاری های معمول، این موضوعات باعث می‌شود که تاثیر یک نفوذ فیزیکی مثل تاثیر امنیت پرسنل در سازمان تواند چندین مشتری را تحت تاثیر قرار دهد.

- چه تضمینی برای امنیت فیزیکی محلی می‌توانید به مشتری ارائه دهید؟ مثلا تضمین هایی که در ISO 27001/1 وجود دارد.
  - چه کسی غیر از پرسنل فناوری اطلاعات دسترسی بدون نظارت به زیر ساخت های فناوری اطلاعات دارد؟ (مثلا مدیران، حراست و مشاوران).
  - قوانین دسترسی هر چند وقت بازبینی می‌شود؟
  - شناسایی ریسک چگونه انجام می‌شود؟ هر چند وقت انجام می‌شود؟
  - آیا شما پرسنلی که به داده‌های حفاظت شده دسترسی دارند را مانیتور می‌کنید؟
  - برای نصب و راه اندازی تجهیزات چه رویه ها و خط مشی ای دارید؟
  - آیا اقلام تحویلی برای ریسک‌ها بازرسی می‌شود؟
  - آیا انباره فیزیکی بروز شده برای استفاده بعنوان مرکز داده وجود دارد؟
  - آیا کابل های شبکه از مسیر هایی که گذر عمومی دارد می گذرد؟
  - آیا تجهیزاتی خارج از سایت برای کار استفاده می‌شود؟
  - آیا پرسنل شما از تجهیزات همراه (رایانه همراه، تلفن هوشمند و ...) که به مرکز داده دسترسی دارد استفاده می‌کنند؟ چگونه مباحث امنیتی برای آنها تامین شده؟
  - برای از بین بردن سیستم‌ها و رسانه‌های قدیمی چه رویه‌هایی وجود دارد؟
  - برای انتقال تجهیزات از یک سایت به سایت دیگر چه پروسه‌های مجازشناسی انجام می‌شود؟
  - بازرسی های تجهیزات هر چند وقت انجام می‌شود؟
  - چک های حقوقی برای تامین نیازمندی های قانونی هر چن وقت انجام می‌شود؟

۹-۱۱-۱۷- کنترل های محیطی

- چه رویه ها و خط مشی ای وجود دارد تا تضمین کند که موضوعات محیطی بر روی سرویس رسانی اختلال ایجاد نمی‌کنند؟
- برای مقابله با خسارات آتش، سیل و زلزله از چه متدی استفاده می‌کنید؟
  - در مواقع حادثه چه موارد امنیتی اضافه ای دارید؟
  - آیا سایت ثانوی وجود دارد؟
- آیا دما و رطوبت محیط را مانیتور می‌کنید؟
- آیا ساختمان خود را در برابر رعد و برق مقاوم کرده‌اید؟
- آیا ژنراتور برق مورد استفاده در هنگام قطع برق دارید؟
  - چقدر کار می‌کند؟
  - هر چند وقت بازبینی می‌شود؟
- آیا تمام موارد (برق و آب و ...) قابل استفاده در محیط شما هستند؟
- آیا سیستم تهویه مناسب را دارا هستید؟
- آیا از توصیه های نگهداری ابزار محیط خود استفاده می‌کنید و آنها را رعایت می‌کنید؟
- آیا وقتی تجهیزات برای تعمیر ارسال می‌شود از قبل داده‌های آن پاک شده است؟
  - چگونه این کار را انجام می دهید؟

۹-۱۱-۱۸- نیازمندی های قانونی

مشتریان ابر باید الزامات قانونی را که در داخل و خارج از چارچوب ابر وجود دارد را به طور کامل مراعات کنند. این الزامات در سطوح ملی و فرا ملی هم مطرح می‌باشد.
سوالات قانونی کلیدی که مشتری باید از تامین کننده ابر بپرسد عبارتند از:

- تامین کننده ابر در کدام کشور قرار دارد؟
- آیا زیر ساخت ابر در همان کشور یا کشور های دیگر است؟
- آیا تامین کننده ممکن است از زیر ساخت های دیگر کمپانی ها هم استفاده کند؟
- داده‌ها از نظر فیزیکی در کجا قرار دارند؟
- آیا حوزه ها بر اساس شرایط قرارداد بخش بندی شده است؟
- آیا ممکن است سرویسی از قرار داد حذف شود؟

پرسش‌های مروری فصل ۹
۱- چند نمونه از ریسک های رایانش ابری را توضیح دهید؟
۲- چند نمونه از آسیب‌پذیری‌های رایج را نام ببرید.
۳- Fail open را تعریف کنید؟
۴- AAA چیست؟
۵- Single sign on به چه معناست؟
۶- بررسی‌های co-residence چگونه بررسی هایی هستند؟
۷- مجازی شناسی، احراز هویت و حسابرسی چه تفاوتی با هم دارند؟
۸- Lock-in چیست؟ کدام دارایی‌ها را تهدید می‌کند؟ چه راه حل‌هایی برای آن وجود دارد؟

تحقیق و پژوهشی فصل ۹
۱- امنیت یک سرور فیزیکی و یک سرور مجازی را با هم مقایسه کنید.
۲- طرح تداوم کسب و کار چیست؟
۳- امنیت به عنوان سرویس (Security as a Service) چیست؟
۴- جدیدترین مخاطراتی که در زمینه رایانش ابری گزارش شده است را جستجو نمایید.
۵- در خصوص روش‌های حمله Social engineering تحقیق کنید.
۶- در خصوص آسیب‌پذیری‌های فوق ناظر تحقیق کنید.
۷- حملات XSS و CSRF به چه شکل انجام می‌شوند؟
۸- Privilege escalation به چه معناست؟
۹- اعتماد و شهرت به چه معناست؟
۱۰- در خصوص زیرساخت IDM تحقیق کنید.
۱۱- در خصوص آخرین روش‌هایی که برای افزایش حریم خصوصی ارائه شده است تحقیق کنید.
۱۲- فرآیند SecSDLC (چرخه عمر تولید نرم افزار بصورت ایمن) و مراحل انجام آن را بررسی کنید.
۱۳- چند نمونه ابزار مجازی (Virtual Appliance)که برای کارهای امنیتی ارائه شده اند را بررسی نموده و باهمدیگر و همچنین با نمونه‌های‌غیرمجازی آن‌ها مقایسه نمایید.

مراجع

[1] Gartner, "Seven cloud-computing security risks," Network World , July 2008

[2] John, W. R. and F. R. James. "Cloud Computing Implementation, Management, and Security", CRC Press, 2010

[3] Enica: European Network and Information Security Agency, "Cloud Computing- Benefits, risks and recommendations for information security", 2009

[4] SUN - Project Kenai: *http://kenai.com/projects/suncloudapis/pages/HelloCloud #Examining_the_Virtual_Data_Center, Accessed at May 2010.*

[5] ISO/IEC. ISO/IEC 27001:2008 Information technology - Security Techniques – Information security risk management; Annex E: Information security risks assessment approaches, 2008

[6] Data Liberation Front, Google, *http://www.dataliberation.org, Accessed at May 2010*

[7] Wikipedia: *http://en.wikipedia.org/wiki/Open_Virtualization_Format*

[8] BBC: *http://news.bbc.co.uk/2/hi/uk_news/scotland/glasgow_and_west/6089736.stm*

[9] Retailresearch.org: *http://www.retailresearch.org/reports/fightinternalfraud.php*

[10] Enterprise Storage Forum, *www.enterprisestorageforum.com/continuity/news/article.php/3800226*

[11] Electronic Discovery Navigator, *http://www.ediscoverynavigator.com/statutesrules/*

[12] Find Law http://technology.findlaw.com [Online] http://technology.findlaw.com/articles/01059/011253.html

[13] Samuel T King, Peter M Chen, Yi-Min Wang, Chad Verbowsk, Helen J Wang, Jacob R Lorch SubVirt: Implementing malware with virtual machines. 2006

[14] Secunia: *http://secunia.com/advisories/37081/*

[15] *http://secunia.com/advisories/36389/*

[16] Ormandy, Tavis: *http://taviso.decsystem.org/virtsec.pdf*

[17] Gentry, Craig: *http://delivery.acm.org/10.1145/1540000/1536440/p169 gentry.pdf?key1=1536440&key2=6166986521&coll=GUIDE&dl=GUIDE&CFID=60359435&CFTOKEN=10086693*

[18] Schneier, Bruce: *http://www.schneier.com/blog/archives/2009/07/homomorphic_enc.html*

[19] www.spywarewarrior.com:*http://www.spywarewarrior.com/uiuc/ss/revoke/pgp revoke.htm*

[20] RSA Laboratories, PKCS#11:*http://www.rsa.com/rsalabs/node.asp?id=2133*

[21] Jun Zhou, Mingxing He: *http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4716141*

[22] Clulow, Tyler Moore and Jolyon:*http://people.seas.harvard.edu/~tmoore/ifipsec-pres.pdf*

[23] Andrew Bechere, Alex Stamos, Nathan Wilcox: *http://www.slideshare.net/astamos/cloud-computing-security*

[24] Open Source Xen Community: *http://xen.org/*

[25] Common Criteria Recognition Agreement (CCRA)*: http://www.commoncriteriaportal.org/ [Online]*

[26] OWASP: *http://www.owasp.org/index.php/OWASP_Top_Ten_Project*

[27] OWASP: *http://www.owasp.org/index.php/Category:OWASP_Guide_Project*

[28] OWASP: *http://www.owasp.org/index.php/Main_Page*

[29] SANS Institute: *http://www.sans.org/reading_room/whitepapers/securecode/a_security_checklist_for_web_applcation_design_1389?show=1389.php&cat=securecode*

[30] Software Assurance Forum for Excellence in Code (SAFECode), *http://www.safecode.org*

[31] IEEE Standards Association: *http://standards.ieee.org/getieee802/download/802.1Q2005.pdf*

[32] Steve Mansfield-Devine, "Danger in the clouds," Network Security, Volume 2008, Issue 12, December 2008, Pages 9-11

فصل ۱۰- ضمائم

- مرجع سرویس‌های وب آمازون
- GOGRID
- VMWARE ESX SERVER
- EUCALYPTUS

۱۰-۱- مرجع سرویس‌های وب آمازون

سرویس‌های وب آمازون از طریق پروتکل های SOAP و REST قابل دسترسی هستند. همچنین امکان دسترسی به آنها از طریق دستورات سطح بالاتر خط فرمان نیز امکان‌پذیر است. این دستورات در این قسمت تشریح می‌شوند. انتزاع های سطح بالاتر متفاوت دیگری نیز وجود دارد که SOAP و API ها را در زبان های برنامه سازی دیگر پیاده‌سازی کرده‌اند. اگر شما ابزارهای برای مدیریت EC2 و S3 ایجاد می‌کنید، باید بهترین کتابخانه ای را که مناسب با زبان برنامه‌نویسی و قابلیت‌های مورد نیازتان هست انتخاب کنید.

۱۰-۱-۱- مرجع دستورات خط فرمان EC2

ابزارهای خط فرمان EC2 بر اساس API های سرویس‌های وب توسعه داده شده‌اند. در حقیقت یک نگاشت یک به یک بین یک خط فرمان و ورودی‌های آن با فراخوانی یک API با نام و پارامتر های مشابه وجود دارد. هر دستور قالب زیر را دارد:

```
command [GENERAL OPTIONS] [COMMAND OPTIONS]
```

برای مثال، دستور اجرای یک نمونه EC2 به این صورت است:

```
ec2-run-instances -v ami-123456 -g dmz
```

در این نمونه، سوئیچ v- یک گزینه عمومی برای مشاهده جزئیات خروجی است و ami-123456 -g dmz گزینه‌های خاص مربوط به این دستور هستند. گزینه‌های عمومی به این صورت هستند:

```
-C certificate
```

گواهینامه ای که برای احراز هویت درخواست سرویس وب به آمازون مورد استفاده قرار می‌گیرد. این مقدار، می‌تواند بجای متغییر محیطی EC2_CERT مورد استفاده قرار بگیرد.

```
--connection-timeout
```

تعیین زمان timeout اتصال SOAP بر حسب ثانیه

```
--debug
```

نمایش اطلاعات خطایابی

```
--headers
```

نمایش ستون عنوان

```
--help
```

نمایش اطلاعات کمکی در خصوص یک دستور

-K privatekey

کلید خصوصی برای احراز هویت درخواست سرویس وب به آمازون. این مقدار می‌تواند بجای متغییر محیطی EC2_PRIVATE_KEY مورد استفاده قرار گیرد.

--region region

انتخاب ناحیه ای که باید به دستور اعمال شود.

--request-timeout

تعیین زمان timeout یک درخواست SOAP بر حسب ثانیه

--show-empty-fields

نمایش ستون های خالی که در پاسخ یک دستور بصورت nil هستند.

-U url

تعیین آدرس URL سرویس‌های وب آمازون برای فراخوانی API. این مقدار می‌تواند بجای متغییر محیطی EC2_URL مورد استفاده قرار گیرد.

-v

نمایش جزئیات خروجی یک دستور. در این حالت درخواست ها و پاسخ های SOAP نمایش داده می‌شود.

۱۰-۱-۱-۱- دستور ec2-add-group

ec2-add-group groupname -d description

افزودن یک گروه امنیتی جدید به محیط EC2. نام گروهی که تعیین می‌کنید، در شناسایی گروه به دیگر دستورات مورد استفاده قرار می‌گیرد. description توضیحاتی است که شما برای یادآوری هدف ایجاد گروه می‌توانید به آن اضافه کنید. گروه جدید امکان دسترسی خارجی به نمونه های موجود در خود را نمی‌دهد. شما باید از ec2-authorize برای تعیین مجوزهای دسترسی به آن، قبل از اینکه هرگونه ترافیکی را به سمت آن گروه هدایت کنید، استفاده نمایید.
مثال:

```
$ ec2-add-group mydmz -d DMZ
GROUP mydmz DMZ
```

۱۰-۱-۱-۲-دستور ec2-add-keypair

ec2-add-keypair keyname

ایجاد یک جفت کلید RSA 2048-bit. نامی که شما تعیین می‌کنید، برای ارجاع به کلید با استفاده از دیگر دستورات مورد استفاده قرار می‌گیرد. این دستور، کلید عمومی را در آمازون قرار می‌دهد و کلید خصوصی را بر روی stdout نمایش می‌دهد. شما باید بطور امن کلید خصوصی را برای استفاده در دسترسی به نمونه هایی که اجرا می‌کنید با استفاده از این جفت کلید ذخیره کنید.
مثال:

```
$ ec2-add-keypair georgekey

----- KEYPAIR georgekey  2e:82:bb:91:ca:51:22:e1:1a:84:c8:19:db:7c:8b:ad:f9:5e:27:3e
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAkot9/09tIy5FJSh0X5vLGCu3so5Q4qG7cU/MBm45c4EVFtMDpU1VpAQi1vn9
r7hr5kLr+ido1d1eBmCkRkHuyhfviJmH1FTOWm6JBhfOsOgDU0pInyQOP0nRFLx4eyJfYsiK/mUm
hiYC9Q6VnePjMUiHSahOL95C8ndAFBlUAuMDDrXMhLypOGRuWkJo+xtlVdisKjlOT0l33q3VSeT6
NBmZwymWOguGWgKWMpzpDLhV9jhDhZgaZmGUKP0wPQqdV6psA9PuStN1LJkhWVYuQTqH9UUol
vJn
ZXx5yE2CSpPW+8zMb4/xUuweBQ6grw8O3IxhKWbFCpGGhkpk5BB+MQIDAQABAoIBAQCIs6U6mA
4X
5l7MFdvRIFSpXIbFAutDLInbjvOlAAeJzt0saHWbKvP7x3v0jElxNRk6OC1HMqIh9plyW46Cl5i4
XvGsvIOvt9izFS+vRmAiOJx5gu8RvSGpOiPXMyU0wFC4ppi6TQNN2oGhthQtsFrMK3tAY8dj8fMD
mehll2b+NPZRWPp9frm3QtwLIOMeWm1ntknCVSjBqj21XRg3UPbE8r8ISlSGryqJBA0KjnOj+cMN
2SBx8iC+BHxD9xSUvXs4hVjUpQofzd+8BAZbsXswj+/ybuq1GlNwzpUKKEfH1rN3TZztywN5Z9Hb
EbkOtgRYi/2htSpbuDq5b/cTaxIRAoGBAOLRgfZhEwnGQvveMOhRLLko1D8kGVHP6KCwlYiNow07
G8FkP6U3wcUrsCTtvOFB/79FeWVT+o7v25D34cYFtGbfnp3Zh9bdTBi18PbIHQHvD4tIAIF+4PcO
XMRsJCrzhChOLY1G/laMi5EKFcx6RU8Pjup92YbEbi/fkybcrmS9AoGBAKVmGI5PV00A10LQkTov
CnLuyfAPL9s8w6eOy+9WMRd8+27tI6H8yGuEdsF9X9bOJQsnTM3+A+RC8ylVXWPgiflbcpbPsZ8a
HVuTg37D/iFpl42RzrMtzhgLCahvNotirNyAYnklBsOlmtsQdJSJ0GPpv4SOloSoPT+jbP4ONUiF
AoGAWU48aZHXOSYDAcB+aTps7YqR5zqDbZ767SoZ9mYuKOt5BjA+jwLhHI0TEbc5g0fFNr5YCfmC
0fzG6tFu59UfLtIlVelsfsErUR9x/PjV0wkZibGT4Wjfkubox738j5zKEESX0uR9B/7WhQj/hD8w
QuzRTKq4l0OITvksq0SAtdECgYAqpr1GVWdp0AGylR4eJutG4BTq9r+chXrexpAIU+2s5OnhnP1H
VGxKbYpCMxZ3ygj7a1L++7X9MtaJnh3LF6f8yXwvL7faE13ms4+BLQFnlFckhqkKw5EV2iLPcH5c
S0HQSrsaClZINXhNbVziwPcgDLL6d9qQsuG4e2gry3YqEQKBgFHqE4UJCOd5WiAG0N0cYDTF/wh6
iujW5tY90F63xAn2B236DGE+8o2wGwU77u59LO7jyx4WyR8TpcorL79zZuzm0Vjn9nslAu7tkS6O
wmdEM0O2LrGnKGydSdRF50NH8Tgb060txh+hWyWtvkP0tSOZyGu1z7S7JS0ZPX42Arm8
END RSA PRIVATE KEY-----
```

۱۰-۱-۱-۳- دستور ec2-allocate-address

```
ec2-allocate-address
```

اختصاص یک آدرس IP معتبر جدید و نمایش آدرس اختصاص داده شده بر روی صفحه. این آدرس جدید تنها برای شما در دسترس خواهد بود تا آن را با نمونه هایی که بخواهید اختصاص دهید. آمازون بابت این آدرس‌های IP از شما هزینه دریافت می‌کند.

مثال:

```
$ ec2-allocate-address
ADDRESS 67.202.55.255
```

۱۰-۱-۱-۴- دستور ec2-associate-address

```
ec2-associate-address -i instanceid ipaddress
```

اختصاص یک آدرس بدست آمده از طریق ec2-allocate-address به نمونه EC2 در حال اجرا. چنانچه این آدرس به نمونه دیگری اختصاص داده شده بود، توسط این دستور از آن جدا می‌شود. بنابراین مهم است که بطور تصادفی از این دستور استفاده نکنید.

مثال:

```
$ ec2-associate-address -i i-12b3ff6a 67.202.55.255
ADDRESS 67.202.55.255    i-12b3ff6a
```

۱۰-۱-۱-۵- دستور ec2-attach-volume

```
ec2-attach-volume volumeid -i instanceid -d device
```

اختصاص یک حافظه ذخیره‌سازی ثابت به یک نمونه EC2 در حال اجرا بطوری که بصورت یک درایو برای آن قابل دسترس باشد. نام درایو مستقل از پلت‌فرم است. مثلا در سیستم عامل لینوکس بصورت /dev/sdh و در ویندوز بصورت xvdh است. این حافظه ها که از نوع block storage هستند را می‌توانید mount نموده و یا توسط دستورات مدیریت دیسک در سیستم عامل format کنید. خروجی این دستور وضعیت اتصال حافظه را به همراه اطلاعات آن نمایش می‌دهد.

مثال:

```
$ ec2-attach-volume vol-81aeb37f -i i-12b3ff6a -d /dev/sdf
ATTACHMENT vol-81aeb37f i-12b3ff6a /dev/sdf attaching 2008-12-17T22:36:00+0000
```

۱۰-۱-۱-۶-دستور ec2-authorize

```
ec2-authorize groupname -P protocol (-p portrange
| -t icmptypecode) [-u sourceuser ...] [-o sourcegroup ...]
[-s sourceaddress]
```

اجازه دسترسی ترافیک شبکه به یک نمونه EC2 اجرا شده در گروه خاص توسط این دستور امکان‌پذیر است. شما می‌توانید ترافیک ورودی را بر اساس معیارهای مختلفی اعتبارسنجی کنید:

- بر اساس ترافیک ورودی از یک زیر شبکه خاص

- بر اساس گروه عضویت نمونه EC2 مبدا
- بر اساس پروتکل (TCP، UDP، ICMP)
- بر اساس پورت مقصد

بطور پیش فرض، یک گروه امکان ورود هیچ ترافیکی را به یک نمونه EC2 خاص نمی‌دهد، مگر نمونه هایی که در همان گروه قرار داشته باشند. برای ورود ترافیک، شما باید بطور خاص اجازه ورود آن را بدهید.
مثال:

```
# Grant port 80 access to all traffic regardless of source
$ ec2-authorize mydmz -P tcp -p 80 -s 0.0.0.0/0
GROUP        mydmz
PERMISSION   mydmz   ALLOWS   tcp   80   80   FROM   CIDR   0.0.0.0/0

# Grant access to the app server group from the DMZ group
$ ec2-authorize myapp -u 999999999999 -o mydmz
GROUP        myapp
PERMISSION   myapp   ALLOWS   all   FROM   USER   999999999999 GRPNAME mydmz


# Grant access to a range of ports from a specific IP address
$ ec2-authorize mydmz -P udp -p 3000-4000 -s 67.202.55.255/32
GROUP        mydmz
PERMISSION   mydmz   ALLOWS   udp   3000   4000   FROM   CIDR 67.202.55.255/32
```

۱۰-۱-۱-۷- دستور ec2-bundle-instance

```
ec2-bundle-instance instanceid -b s3bucket -p prefix -o accesskey (-c policy | -w secretkey)
```

این دستور فقط مربوط به نمونه های ویندوزی است که نمونه ویندوزی را دسته بندی کرده و آن را برای ذخیره‌سازی در S3 آماده می‌کند و سپس باید توسط ec2-register ثبت شود.
مثال:

```
$ ec2-bundle-instance i-12b3ff6a -b mybucket -p myami -o 999999999999 -w
lY1zp/1iKzSAg9B04lQ0T3gMxje7IfnXtN5asrM/dy==
BUNDLE  bun-abd5209d8  i-12b3ff6a  mybucket myami  pending  2008-12-
18T13:08:18+0000  2008-12-18T13:08:18+0000
```

۱۰-۱-۱-۸- دستور ec2-cancel-bundle-task

ec2-cancel-bundle-task bundleid

این دستور فقط مربوط به نمونه های ویندوزی است که فرآیند bundle را که در حال انجام است متوقف می‌کند.

مثال:

```
$ ec2-cancel-bundle-task bun-abd5209d8

BUNDLE bun-abd5209d8 i-12b3ff6a mybucket myami canceling 2008-12-
18T13:13:29+0000 2008-23-18T13:13:29+0000
```

۱۰-۱-۱-۹-دستور ec2-confirm-product-instance

ec2-confirm-product-instance productcode -i instanceid

به صاحب یک AMI امکان بررسی اینکه آیا یک نمونه خاص دارای کد محصول می‌باشد را می‌دهد.

مثال:

```
$ ec2-confirm-product-instance zt1 -i i-12b3ff6a

ztl i-12b3ff6a false
```

۱۰-۱-۱-۱۰- دستور ec2-create-snapshot

ec2-create-snapshot volumeid

ایجاد یک تصویر تفاضلی از درایو خاص و ذخیره‌سازی آن در S3. بهتر است زمانی این تصویر تهیه شود که سیستم فایل در حالت ثابت "frozen" قرار داشته باشد تا از پایداری آن اطمینان حاص شود. پس از اینکه خروجی دستور با موفقیت نمایش داده شد، می‌توان مجدداً نوشتن بر روی دیسک را آغاز کرد.

مثال:

```
$ ec2-create-snapshot vol-12345678

SNAPSHOT snap-a5d8ef77 vol-12345678 pending 2008-12-20T20:47:23+0000
```

۱۰-۱-۱-۱۱- دستور ec2-create-volume

ec2-create-volume (-s size | --snapshot snapshotid) -z zone

ایجاد یک درایو جدید با اندازه مشخص یا بر اساس تصاویر تهیه شده در همان ناحیه دسترسی. پارامتر اندازه بر حسب گیگابایت می‌باشد.

مثال:

```
# Create a new volume of 10 GB
$ ec2-create-volume -s 10 -z eu-west-1a
VOLUME  vol-12345678  10  eu-west-1a  creating  2008-12-20T20:47:23+0000

# Create a volume based on a stored snapshot
$ ec2-create-volume --snapshot snap-a5d8ef77 -z eu-west-1a
VOLUME  vol-12345678  10  eu-west-1a  creating  2008-12-20T20:47:23+0000
```

۱۰-۱-۱-۱۲- دستور ec2-delete-group

```
ec2-delete-group group
```

حذف یک گروه امنیتی از حساب کاربری شما. تا زمانی که نمونه ای در یک گروه امنیتی قرار داشته باشد، نمی‌توانید آن را حذف کنید.

مثال:

```
$ ec2-delete-group mydmz
GROUP mydmz
```

۱۰-۱-۱-۱۳- دستور ec2-delete-keypair

```
ec2-delete-keypair keypair
```

حذف کلید عمومی مربوط به یک زوج کلید از حساب کاربری شما.

مثال:

```
$ ec2-delete-keypair georgekey
KEYPAIR georgekey
```

۱۰-۱-۱-۱۴- دستور ec2-delete-snapshot

```
ec2-delete-snapshot snapshotid
```

حذف یک تصویر از حساب کاربری شما.

مثال:

```
$ ec2-delete-snapshot snap-a5d8ef77
SNAPSHOT snap-a5d8ef77
```

۱۰-۱-۱-۱۵- دستور ec2-delete-volume

ec2-delete-volume volumeid

حذف یک درایو از حساب کاربری شما.
مثال:

```
$ ec2-delete-volume vol-12345678
VOLUME vol-12345678
```

۱۰-۱-۱-۱۶- دستور ec2-deregister

ec2-deregister imageid

ابطال[1] یک تصویر ماشین مجازی بطوری که دیگر نمی‌توانید نمونه ای را از آن ایجاد کنید. شما باید AMI مربوطه را بطور جداگانه از S3 حذف کنید تا حافظه اشغال شده توسط آن نیز آزاد شود.
مثال:

```
$ ec2-deregister ami-f822a39b
IMAGE ami-f822a39b
```

۱۰-۱-۱-۱۷- دستور ec2-describe-addresses

ec2-describe-addresses [ipaddres1 [...ipaddressN]]

نمایش اطلاعات مربوط به یک آدرس IP معتبر. اگر آدرس خاصی تعیین نکرده باشید، همه آدرس‌هایی را که به منطقه EC2 خود اختصاص داده اید، نمایش می‌دهد:
مثال:

```
# SHOW ALL ALLOCATED
$ ec2-describe-addresses
ADDRESS 67.202.55.255 i-12b3ff6a
ADDRESS 67.203.55.255
# SHOW A SPECIFIC ADDRESS
$ ec2-describe-addresses 67.202.55.255
ADDRESS 67.202.55.255 i-12b3ff6a
```

[1] Deregisters

۱۰-۱-۱-۱۸- دستور ec2-describe-availability-zones

ec2-describe-availability-zones [zone1 [...zoneN]]

نمایش اطلاعات مربوط به یک ناحیه دسترسی EC2 خاص. اگر ناحیه ای را تعیین نکنید، همه ناحیه های موجود در منطقه EC2 شما لیست می‌شود.

مثال:

```
# SHOW ALL ALLOCATED

$ ec2-describe-availability-zones

AVAILABILITYZONE    us-east-1a  available
AVAILABILITYZONE    us-east-1b  available
AVAILABILITYZONE    us-east-1c  available

# SHOW A SPECIFIC ZONE

$ ec2-describe-availabilty-zones us-east-1a

AVAILABILITYZONE    us-east-1a  available

# SHOW ALL EU ZONES

$ ec2-describe-availability-zones --region eu-west-1

AVAILABILITYZONE    eu-west-1a  available
AVAILABILITYZONE    eu-west-1b  available
```

۱۰-۱-۱-۱۹- دستور ec2-describe-bundle-tasks

ec2-describe-bundle-tasks [bundle1 [...bundleN]]

این دستور فقط برای نمونه های ویندوز قابل استفاده است که اطلاعات مربوط به یک فعالیت bundle خاص را نمایش می‌دهد. اگر فعالیت مشخصی تعیین نشده باشد، لیست همه فعالیت های bundle نمایش داده می‌شود.

مثال:

```
# SHOW ALL TASKS

$ ec2-describe-bundle-tasks

BUNDLE  bun-abd5209d8  i-12b3ff6a  mybucket myami  pending  2008-12-
18T13:08:18+0000  2008-12-18T13:08:18+0000
BUNDLE  bun-abd5209d9  i-12b3ff7a  mybucket myami  pending  2008-12-
18T13:08:18+0000  2008-12-18T13:08:18+0000

# SHOW SPECIFIC TASK
```

```
$ ec2-describe-bundle-tasks bun-abd5209d8

BUNDLE  bun-abd5209d8  i-12b3ff6a  mybucket myami  pending  2008-12-
18T13:08:18+0000  2008-12-18T13:08:18+0000
```

۱۰-۱-۱-۲۰- دستور ec2-describe-group

```
ec2-describe-group [group1 [...groupN]]
```

نمایش اطلاعات مربوط به یک گروه امنیتی خاص. اگر گروهی مشخص نشده باشد، همه گروه‌های امنیتی حساب کاربری شما نمایش داده خواهد شد.

مثال:

```
# SHOW ALL GROUPS

$ ec2-describe-group

GROUP mydmz DMZ
PERMISSION   mydmz   ALLOWS   tcp   80   80   FROM   CIDR   0.0.0.0/0
GROUP myapp App
PERMISSION   myapp   ALLOWS   all   FROM   USER   999999999999 GRPNAME mydmz

# SHOW A SPECIFIC GROUP

$ ec2-describe-group mydmz

PERMISSION   mydmz   ALLOWS   tcp   80   80   FROM   CIDR   0.0.0.0/0
```

۱۰-۱-۱-۲۱- دستور ec2-describe-image-attribute

```
ec2-describe-image-attribute imageid (-l | -p)
```

نمایش مشخصه های یک AMI خاص. شما می‌توانید تعیین کنید که مجوزهای اجرا یا کدهای محصول را نیز مشاهده کنید.

مثال:

```
# SHOW LAUNCH PERMISSIONS

$ ec2-describe-image-attribute ami-f822a39b -l

launchPermission  ami-f822a39b  userId 999999999999

# SHOW PRODUCT CODE

$ ec2-describe-image-attribute ami-f822a39b -p

productCodes ami-f822a39b productCode zz95xy
```

۱۰-۱-۱-۲۲- دستور ec2-describe-images

ec2-describe-images [imageid1 [...imageidN]] [-a] [-o ownerid] [-x ownerid]

مشاهده اطلاعات مربوط به یک تصویر خاص یا هر تصویری که با پارامترهای مشخص شده مطابقت داشته باشد. اگر شما پارامتری را تعیین نکنید، لیست همه تصاویری که متعلق به شما باشد را مشاهده خواهید کرد.

گزینه‌های مرتبط با آن به این صورت می‌باشد:

-a

مشاهده لیست AMI هایی که کاربر مجوز اجرای آنها را دارد.

-o ownerid

مشاهده لیست AMI هایی که متعلق به یک کاربر خاص است. همچنین می‌توان از شناسه های خاص استفاده کرد: amazon (برای تصاویر عمومی)، self (تصاویر متعلق به شما)، explicit (اشاره به تصاویری که شما مجوز اجرای آنها را دارید).

-x ownerid

مشاهده لیست AMI هایی را که یک کاربر یا کاربران خاصی اجازه اجرای آن را دارند. علاوه بر شماره شناسه استاندارد، می‌توانید از عبارت self برای مشاهده تمام AMI هایی که اجازه اجرا آنها را دارید استفاده کنید. یک روش مناسب برای پیداکردن تصویری که بتوانید از آن برای شروع کار استفاده کنید، استفاده از این دستور بصورت زیر می‌باشد:

ec2-describe-images -o amazon

مثال:

```
# SHOW ALL OWNER IMAGES
$ ec2-describe-images
IMAGE ami-f822a39b myami/myami.manifest.xml 999999999999 available private
zz95xy i386 machine aki-a71cf9ce ari-a51cf9cc

# SHOW IMAGES FOR A SPECIFIC USER
$ ec2-describe-images -o 063491364108
IMAGE ami-48de3b21 level22-ec2-images-64/ubuntu-7.10-gutsy-base-
64-20071203a.manifest.xml 063491364108 available public x86_64 machine
IMAGE ami-dd22c7b4 level22-ec2-images-64/ubuntu-7.10-gutsy-base-
64-20071227a.manifest.xml 063491364108 available public x86_64 machine
```

۱۰-۱-۱-۲۳-دستور ec2-describe-instances

ec2-describe-instances [instanceid1 [...instanceidN]]

نمایش اطلاعات مرتبط با نمونه ای خاص. اگر نمونه ای تعیین نشده باشد لیست همه نمونه های مربوط به یک حساب کاربری نمایش داده می‌شود.

مثال:

```
# SHOW ALL INSTANCES

$ ec2-describe-instances

RESERVATION r-3d01de54 999999999999 default
INSTANCE i-b1a21bd8 ami-1fd73376 pending 0 m1.small 2008-10-
22T16:10:38+0000 us-east-1a aki-a72cf9ce ari-a52cf9cc
RESERVATION r-3d01cc99 999999999999 default
INSTANCE i-ccdd1b22 ami-1fd73376 pending 0 m1.small 2008-10-
22T16:10:38+0000 us-east-1a aki-a72cf9ce ari-a52cf9cc

# SHOW A SPECIFIC INSTANCE

$ ec2-describe-instances i-b1a21bd8

RESERVATION r-3d01de54 999999999999 default
INSTANCE i-b1a21bd8 ami-1fd73376 pending 0 m1.small 2008-10-
22T16:10:38+0000 us-east-1a aki-a72cf9ce ari-a52cf9cc
```

۱۰-۱-۱-۲۴-دستور ec2-describe-keypairs

ec2-describe-keypairs [keypairid1 [...keypairidN]]

نمایش اطلاعات مرتبط با یک جفت کلید خاص. اگر کلید خاصی تعیین نشود، همه کلیدهای شما را نمایش می‌دهد.

مثال:

```
$ ec2-describe-keypairs

KEYPAIR georgekey 98:21:ff:2a:6b:35:71:6e:1f:36:d9:f2:2f:d7:aa:e4:14:bb:1d:1a
```

۱۰-۱-۱-۲۵-دستور ec2-describe-regions

ec2-describe-regions [region1 [...regionN]]

نمایش اطلاعات مربوط به یک منطقه خاص. اگر منطقه ای تعیین نشده باشد، همه مناطق را نشان می‌دهد:

مثال:

```
$ ec2-describe-regions

REGION  eu-west-1  eu-west-1.ec2.amazonaws.com
REGION  us-east-1  us-east-1.ec2.amazonaws.com
```

۱۰-۱-۱-۲۶- دستور ec2-describe-snapshots

```
ec2-describe-snapshots [snapshotid1 [...snapshotidN]]
```

نمایش اطلاعات مربوط به یک تصویر خاص. اگر تصویری تعیین نشده باشد، همه تصاویر حساب کاربری نمایش داده می‌شود.

مثال:

```
$ ec2-describe-snapshots

SNAPSHOT snap-a5d8ef77  vol-12345678  pending  2008-12-20T20:47:23+0000 50%
```

۱۰-۱-۱-۲۷- دستور ec2-describe-volumes

```
ec2-describe-volumes [volumeid1 [...volumeidN]]
```

نمایش اطلاعات مربوط به یک درایو خاص. اگر درایوی تعیین نشده باشد، همه درایو های حساب کاربری نمایش داده می‌شود.

مثال:

```
$ ec2-describe-volumes

VOLUME vol-81aeb37f  5  snapa5d8ef77  us-east-1a  in-use 2008-12-17T22:36:00+0000
ATTACHMENT vol-81aeb37f i-12b3ff6a /dev/sdf attached 2008-12-17T22:36:00+0000
```

۱۰-۱-۱-۲۸- دستور ec2-detach-volume

```
ec2-detach-volume volumeid [-i instanceid] [-d device] –force
```

جداکردن یک درایو خاص از یک نمونه ای که در حال حاضر به آن متصل است. شما باید مطمئن شوید که قبل از جدا کردن درایو، آن درایو از سیستم فایل unmount شده باشد، در غیر اینصورت احتمال خراب شدن داده‌ها وجود دارد. اگر درایو جدا نشد، می‌توانید با سوئیچ force-- این کار را تکرار کنید.

مثال:

```
$ ec2-detach-volume

ATTACHMENT vol-81aeb37f i-12b3ff6a /dev/sdf detaching 2008-12-17T22:36:00+0000
```

۱۰-۱-۱-۲۹- دستور ec2-disassociate-address

ec2-disassociate-address idaddress

جدا کردن یک آدرس IP معتبر از یک نمونه ای که در حال حاضر آدرس به آن اختصاص داده شده است.
مثال:

$ ec2-disassociate-address 67.202.55.255

ADDRESS 67.202.55.255

۱۰-۱-۱-۳۰- دستور ec2-get-console-output

ec2-get-console-output instanceid [-r]

نمایش خروجی مراحل اجرای یک نمونه. با استفاده از سوئیچ r– (raw) خروجی بدون قالب بندی خاصی نمایش داده می‌شود.
مثال:

$ ec2-get-console-output i-b1a21bd8

```
i-b1a21bd8
2008-12-23T20:03:07+0000
Linux version 2.6.21.7-2.fc8xen (mockbuild@xenbuilder1.fedora.redhat.com) (gcc
version 4.1.2 20070925 (Red Hat 4.1.2-33)) #1 SMP Fri Feb 15 12:39:36 EST 2008
BIOS-provided physical RAM map:
sanitize start
sanitize bail 0
copy_e820_map() start: 0000000000000000 size: 000000006ac00000 end:
000000006ac00000 type: 1
 Xen: 0000000000000000 - 000000006ac00000 (usable)
980MB HIGHMEM available.
727MB LOWMEM available.
NX (Execute Disable) protection: active
Zone PFN ranges:
  DMA             0 ->   186366
  Normal     186366 ->   186366
  HighMem    186366 ->   437248
early_node_map[1] active PFN ranges
    0:        0 ->   437248
ACPI in unprivileged domain disabled
Detected 2600.043 MHz processor.
Built 1 zonelists.  Total pages: 433833
Kernel command line:  root=/dev/sda1 ro 4
Enabling fast FPU save and restore... done.
Enabling unmasked SIMD FPU exception support... done.
```

```
Initializing CPU#0
CPU 0 irqstacks, hard=c136c000 soft=c134c000
PID hash table entries: 4096 (order: 12, 16384 bytes)
Xen reported: 2600.000 MHz processor.
Console: colour dummy device 80x25
Dentry cache hash table entries: 131072 (order: 7, 524288 bytes)
Inode-cache hash table entries: 65536 (order: 6, 262144 bytes)
Software IO TLB disabled
vmalloc area: ee000000-f4ffe000, maxmem 2d7fe000
Memory: 1711020k/1748992k available (2071k kernel code, 28636k reserved, 1080k
data, 188k init, 1003528k highmem)
```

۱۰-۱-۱-۳۱- دستور ec2-get-password

```
ec2-get-password instanceid -k keypair
```

این دستور فقط برای نمونه های ویندوزی قابل استفاده است که کلمه عبور مدیر سیستم را بر اساس زوج کلید برای اجرای نمونه فراهم می‌کند.

مثال:

```
$ ec2-get-password i-b1a21bd8 -k georgekey

sZn7h4Dp8
```

۱۰-۱-۱-۳۲-دستور ec2-modify-image-attribute

```
ec2-modify-image-attribute imageid -l -a value
ec2-modify-image-attribute imageid -l -r value
ec2-modify-image-attribute imageid -p productcode [-p productcode]
```

تغییر مشخصه یک تصویر. سوئیچ l– مشخصه های مجوز اجرا را مشخص می‌کند و سوئیچ p– کد محصول را تعیین می‌کند.

مثال:

```
# Add access

$ ec2-modify-image-attribute ami-f822a39b -l -a 123456789

launchPermission ami-f822a39b ADD userId 123456789


# Remove access

$ ec2-modify-image-attribute ami-f822a39b -l -r 123456789

launchPermission ami-f822a39b REMOVE userId 123456789
```

```
# Add product code
$ ec2-modify-image-attribute ami-f822a39b -p crm114
productCodes ami-f822a39b   productCode crm114
```

۱۰-۱-۱-۳۳-دستور ec2-reboot-instances

```
ec2-reboot-instances instanceid1 [...instanceidN]
```

راه اندازی مجدد یک نمونه خاص از خط فرمان. خروجی خاصی برای این دستور وجود ندارد، مگر اینکه با خطایی مواجه شود.

مثال:

```
ec2-release-address
```

۱۰-۱-۱-۳۴-دستور ec2-release-address ipaddress

رهایی آدرسی که در حال حاضر به شما اختصاص داده شده است. وقتی این دستور اجرا شد، شما نمی‌توانید آدرس آزاد شده را برگردانید.

مثال:

```
$ ec2-release-address 67.202.55.255
ADDRESS 67.202.55.255
```

۱۰-۱-۱-۳۵-دستور ec2-register

```
ec2-register s3manifest
```

ثبت تصویر ماشینی که فایل خلاصه آن (manifest) در مکان مشخص شده قرار دارد.

مثال:

```
$ ec2-register myami/myami.manifest.xml
IMAGE ami-f822a39b
```

۱۰-۱-۱-۳۶-دستور ec2-reset-image-attribute

```
ec2-reset-image-attribute imageid –l
```

تنظیم مجدد مشخصه مربوط به مجوز اجرای یک تصویر برای یک تصویر ماشین مجازی مشخص

مثال:

```
$ ec2-reset-image-attribute ami-f822a39b -l

launchPermission ami-f822a39b RESET
```

۱۰-۱-۱-۳۷-دستور ec2-revoke

```
ec2-revoke groupname [-P protocol] (-p portrange | -t icmptypecode)
[-u sourceuser ...] [-o sourcegroup ...] [-s sourceaddress]
```

حذف مجوز های داده شده از یک گروه امنیتی خاص. گزینه‌های آن متناسب با گزینه‌هایی است که در زمان تخصیص مجوز با استفاده از ec2-authoriza استفاده کرده بودید.

مثال:

```
$ ec2-revoke -P tcp -p 80 -s 0.0.0.0/0

GROUP  mydmz
PERMISSION  mydmz  ALLOWS  tcp  80  80  FROM  CIDR  0.0.0.0/0
```

۱۰-۱-۱-۳۸-دستور ec2-run-instances

```
ec2-run-instances imageid [-n count] [-g groupname1 [... -g groupnameN]]
[-k keypair] -d customdata | -f customfile] [ -t type] [ -z zone]
[ --kernel kernelid] [ --ramdisk ramdiskid] [ -B devicemapping]
```

تلاش برای اجرای یک یا چند نمونه EC2 بر اساس یک AMI و گزینه‌های تعیین شده که به صورت زیر می‌باشد:

-B devicemapping

تعیین نحوه نگاشت دستگاه ها با نمونه ای که اجرا می‌شود. شما می‌توانید یکی از انواع نام های مجازی را انتخاب کنید:

- ami : فایل سیستم ریشه که توسط نمونه دیده می‌شود
- root : فایل سیستم ریشه که توسط کرنل دیده می‌شود.
- swap : درایو swap که توسط نمونه دیده می‌شود.
- ephemeralN :N امین دستگاه ذخیره‌سازی جانبی

-d customdata

داده‌هایی که باید برای نمونه در حال اجرا در دسترس باشد. اگر بخواهید داده‌های زیادی را تعیین کنید، می‌توانید آنها را در داخل یک فایل قرار دهید و از سوئیچ f– استفاده کنید.

-f customfile

نام یک فایل با داده‌های زمان اجرا که باید برای نمونه ای که اجرا می‌شود در دسترس قرار گیرد.

-g groupname

نام گروه امنیتی که قوانین آن باید بر نمونه (ها)ی در حال اجرا اعمال شود. شما می‌توانید چندین گروه امنیتی را برای آنها تعیین کنید. اگر چندین گروه امنیتی تعیین کنید، دسترسی به نمونه بر اساس اجتماع مجوزهای آنها انجام می‌شود.

-k keypair

کلید عمومی برای EC2 برای اینکه در زمان راه اندازی در نمونه اجرا شده قرار بگیرد.

--kernel kernelid

شناسه کرنل که نمونه با آن اجرا می‌شود.

-n count

حداقل تعداد نمونه هایی که باید با این دستور اجرا شود. اگر EC2 نتواند حداقل تعداد را اجرا کند، هیچ کدام اجرا نخواهد شد.

--ramdisk ramdiskid

شناسه RAM که نمونه با آن اجرا شود.

-t type

نوع نمونه مشخصه های مربوط به CPU، RAM و ... را تعیین می‌کند: m1.small، m1.large، m1.xlarge، c1.medium، c1.xlarge.

-z zone

ناحیه دسترسی که در آن نمونه ها باید اجرا شوند. اگر ناحیه ای تعیین نشده باشد، نمونه ها در ناحیه ای که توسط خود EC2 در زمان اجرا بهتر تشخیص داده می‌شود، اجرا خواهد شد.

مثال:

```
# Launch exactly 1 instance anywhere

$ ec2-run-instances ami-f822a39b

RESERVATION r-a882e29b7 999999999999 default
INSTANCE i-b1a21bd8 ami-f822a39b pending 0 m1.small 2008-12-23T21:37:13+0000 us-east-1c

# Launch at least 2 instances in us-east-1b

$ ec2-run-instances ami-f822a39b -n 2 -z us-east-1b

RESERVATION r-ac82e29b8 999999999999 default
INSTANCE i-b1a21be9 ami-f822a39b pending 0 m1.small 2008-12-23T21:37:13+0000 us-east-1b
```

INSTANCE i-b1a21bf0 ami-f822a39b pending 0 m1.small 2008-12-23T21:37:13+0000 us-east-1b

Launch exactly 1 instance with the specified keypair in the myapp group

$ ec2-run-instances ami-f822a39b -g myapp -k georgekey

RESERVATION r-a882e29b7 999999999999 default
INSTANCE i-b1a21bd8 ami-f822a39b pending georgekey 0 m1.small 2008-12-23T21:37:13+0000 us-east-1c

۱۰-۱-۱-۳۹-دستور ec2-terminate-instances

ec2-terminate-instances

خاتمه دادن اجرای نمونه (ها)ی مورد نظر.

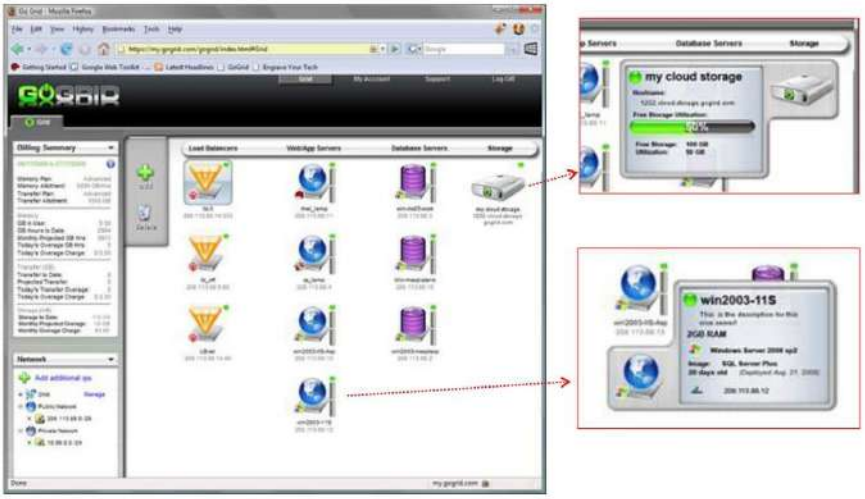
مثال:

$ ec2-terminate-instances i-b1a21bd8

INSTANCE i-b1a21bd8 running shutting-down

GoGrid -۲-۱۰

بطور کلی برای ایجاد یک زیرساخت ابری بیش از یک راه وجود دارد. سرویس‌های وب آمازون (AWS) که رویکرد سرویس زیرساخت[1] را ارائه داده است، به مشتریان، تعدادی سرویس سفارشی اما با مقیاس‌پذیری بسیار بالا پیشنهاد داده است تا زیرساخت مورد نیاز خود را در محیطی کاملا مجازی ایجاد کنند. رویکرد متفاوت دیگر GoGrid است که محیطی بسیار شبیه مرکز داده، اما در داخل ابر ارائه می‌کند. برای بسیاری از مشتریان، رویکرد GoGrid ساده‌تر و راحت تر است، زیرا از تکنولوژی‌های سنتی نظیر VLAN ها، شبکه ها، تجهیزات سخت‌افزاری (تعدیل‌کننده‌بار[2] و دیوار آتش)، و تجهیزات ذخیره‌سازی فایل (SAN یا NAS) استفاده می‌کند. آنچه که در این رویکرد ارائه می‌شود تحت عنوان مرکز ابری[3] یا مرکز داده در ابر معرفی می‌شود.



شکل ۱۰-۱-نمونه‌ای از زیرساخت ابری ارائه شده توسط GoGrid (سرورها می‌توانند مجازی یا فیزیکی باشند)

۱۰-۲-۱- انواع ابرهای زیرساخت

ابرهای زیرساخت می‌توانند به دو شکل ساخته شوند: "سرویس زیرساخت" و "مرکز ابری". هر دوی این‌ها همه قابلیتهایی را که یک نفر از IaaS انتظار دارد، فراهم می‌کنند:

- مقیاس‌پذیری بر حسب تقاضا[4]

---

[1] Service Infrastructure
[2] Load Balancer
[3] CloudCenter
[4] Scale on Demand

- پرداخت بر اساس استفاده[1]
- تبدیل هزینه‌های سرمایه گذاری[2] به هزینه‌های عملیاتی[3]
- واسط برنامه‌نویسی و واسط گرافیکی کاربر
- اجزای اساسی زیرساخت (ذخیره‌سازی، سرورها، شبکه، توان[4]، سرمایش[5]) و ...

اگرچه هر دو بصورت اولیه، ارزش یکسانی دارند، اما تفاوت عمده ای در رویکرد آنها وجود دارد:

۱۰-۲-۲- زیرساخت سرویس (Service Infrastructure)

این رویکرد شبیه AWS است. زیرساخت سرویس بطور کلی سرویس‌های وب سفارشی در داخل ابر هستند. این ها می‌توانند بطور جداگانه یا با همدیگر برای ارائه یک برنامه وب یا انجام کارهای مختلف استفاده شوند. برای مثال، آمازون سرورها، تجهیزات ذخیره‌سازی، پایگاه‌داده، امکانات پیغام رسانی و صف، فرآیند پرداخت و ... را فراهم کرده است. هر یک از این سرویس‌های وب یک راه‌حل سفارشی و منحصر به فرد است. در S3 از پروتکل S3 و مکانیسم های ذخیره‌سازی استفاده می‌شود. سرویس صف AWS SQS از پروتکل سفارشی و غیراستاندارد و نیز فرمت پیغام رسانی خود استفاده می‌کند. مشابه آن سرویس SimpleDB در خصوص پایگاه‌داده عمل می‌کند. این سرویس‌ها بصورت سفارشی طراحی شده‌اند تا آمازون بتواند بستر ابری خود را به بیش از ۵۰۰۰۰ سرور و هزاران محصول مختلف گسترش دهد.

۱۰-۲-۳- مراکز ابری (CloudCenters)

اکثر رقبای AWS از این رویکرد استفاده می‌کنند. متدلوژی مورد استفاده این است که سرویس‌های استاندارد مرکز داده با استفاده از تکنولوژی‌ها و پروتکل های استاندارد، بصورت ابری فراهم شود. برای سیستم ذخیره‌سازی، از پروتکل های آشنا و رایج SMB/CIFS و NFS استفاده شود. پایگاه ها داده با استفاده از SQL و RDBMS استاندارد فراهم شود. دیواره‌های آتش و تعدیل کننده های بار بجای استفاده از نرم‌افزارهای سفارشی توزیع شده، مبتنی بر تجهیزات سخت‌افزاری باشند. در نهایت اینکه انتخاب باید بین زیرساخت سفارشی با پروتکل های خاص خود و زیرساخت استاندارد و مشابه به سرویس‌های مراکز داده با همان استاندارد های صنعتی انجام شود.

---

[1] Pay as you go
[2] CapEx
[3] OpEx
[4] Power
[5] Cooling

۱۰-۲-۴- مراکز ابری با جزئیات بیشتر

GoGrdi اولین و بزرگترین مرکز ابری در ایالات متحده است که این رویکرد را بکارگرفته است. از دیگر شرکت‌هایی که با این روش کار می‌کنند می‌توان FlexiScale، ElasticHosts و AppNexus را نام برد. از مزایای اصلی آن امکان بکارگیری مستقیم مهارت‌ها و زیرساخت فعلی و تبدیل آن به محیط قابل انعطاف ابری نام برد. رویکرد GoGrid در نهایت "Cloud-bridging" یا اتصال و یکپارچه‌سازی مراکز داده داخلی با ابرهای خارجی را به شیوه بسیار ساده‌تری فراهم کرده است.

مراکز داده سنتی از اجزای زیر تشکیل شده‌اند:

- امنیت محیطی با استفاده از دیوارهای آتش سخت‌افزاری و سیستم‌های تشخیص نفوذ
- تعدیل بار با استفاده از سخت‌افزارهای تعدیل کننده
- تقسیم‌بندی شبکه با ایجاد اجزای مختلف شبکه و VLAN ها
- ترکیبی از سیستم‌های عامل بر روی سخت‌افزارهای فیزیکی و مجازی
- به اشتراک‌گذاری فایل با استفاده از NAS
- ذخیره‌سازی با استفاده از SAN
- پشتیبانی از سرویس‌هایی نظیر :DNS، DHCP، server imaging، مدیریت موجودی، مدیریت دارایی و مانیتورینگ
- تامین توان، پهنای باند، سیستم‌های خنک کننده و پشتیبان گیری برای همه این سرویس‌ها و دستگاه ها
- پشتیبانی ۷/۲۴

مراکز ابری بسیار شبیه مراکز داده هستند که اکثر این سرویس‌ها را با اندکی تفاوت ارائه می‌دهند. علاوه بر این، مراکز ابری، برخلاف مراکز داده معمولی، کارآیی را از طریق واسطهای گرافیکی و API ها افزایش می‌دهند.

۱۰-۲-۵- مقایسه GoGrid با مراکز داده معمولی

مشکل اصلی مراکز داده معمولی، نیاز به ایجاد آنها با حداکثر ظرفیت است. شما به عنوان متخصص IT، میبایست بطور صحیح پیش بینی کنید و این ظرفیت را بصورت داخلی ایجاد و مدیریت کنید. مراکز ابری به شما امکان استفاده از مجدد از تخصص داخلی مراکز داده تان را با فراهم کنندگان ابری دیگر می‌دهد. استفاده مجدد از دانش و تخصص به این معناست که زمان کمتری برای یادگیری ایده های جدید لازم است که صرف شود و در عوض وقت بیشتری روی اهداف کسب و کار خود می گذارید، ضمن اینکه از همه مزایای

رایانش ابری بهره مند می شوید (نظیر افزایش ظرفیت بر حسب تقاضا، تعدیل خودکار بارکاری، پرداخت بر اساس استفاده).

۱۰-۲-۶- مقیاس‌پذیری افقی و عمودی

استقرار برنامه روی GoGrid مشابه یک مرکز داده معمولی است، اما ابزارهای جدید بسیاری وجود دارد که می‌تواند این فرآیند را ساده‌تر و سریعتر انجام دهند. شبیه هر ابر دیگر، شما گزینه‌هایی برای مقیاس‌پذیری افقی[1] دارید همچنین امکان مقیاس‌پذیری عمودی[2] را نه تنها با نمونه های ابری مجازی شده بلکه با سخت‌افزارهای فیزیکی اختصاصی نیز خواهید داشت. این کار مشابه یک مرکز داده معمولی است، که در آن ترکیبی از سرورهای فیزیکی و مجازی وجود دارد. شما می‌توانید دسترسی مستقیم به سخت‌افزارهای فیزیکی با تعداد زیادی RAM، ذخیره‌سازی مستقیم با سرعت بالا[3] و تعداد زیادی هسته‌های پردازشی اختصاصی که همه در یک شبکه خصوصی مجازی یکسان قرار دارند، داشته باشید. با بکارگیری صحیح ابزارها، شما می‌توانید به طور افقی بار کاری stateless خود را روی سرورهای مجازی گسترش دهید و برای کارهای stateful از سرورهای فیزیکی برای گسترش افقی استفاده کنید که در صورت نیاز برای گسترش عمودی نیز مناسب تر هستند. علت این موضوع در ادامه آمده است:

۱۰-۲-۷- مقیاس‌پذیری افقی "Scaling out"

مقیاس‌پذیری افقی بهترین روش برای سرورها و کاربردهایی است که stateless هستند (نظیر سرورهای وب و سرورهای کاربردی و فرآیند های دسته ای[4]). برای این نوع از بارهای کاری، افزودن یک سرور اضافی معمولا پیکربندی خاصی یا قابل توجهی نیاز ندارد. به آسانی می‌توان سرورهای بیشتر را اضافه کرد و ظرفیت را افزایش داد.

۱۰-۲-۸- مقیاس‌پذیری عمودی "Scaling Up"

مقیاس‌پذیری عمودی بیشتر برای برنامه‌هایی مورد نیاز است که stateful هستند (نظیر پایگاه‌های داده و سرورهای فایل). در این موارد، افزودن سرورهای بیشتر به سادگی باعث افزایش ظرفیت نمی‌شود. معمولا لازم است که پیکربندی‌ها تغییر کند یا بطور عمده از اول انجام شود، یا معماری تغییر کند یا حداقل بطور خودکار وضعیت داده روی سرورهای جدید تعدیل شود. به این ترتیب حجم زیادی از داده باید همزمان شده و یا تعدیل بار شوند. این علتی است که معمولا ترجیح داده می‌شود تا بجای اینکه تعداد زیادی سرور بکار

---

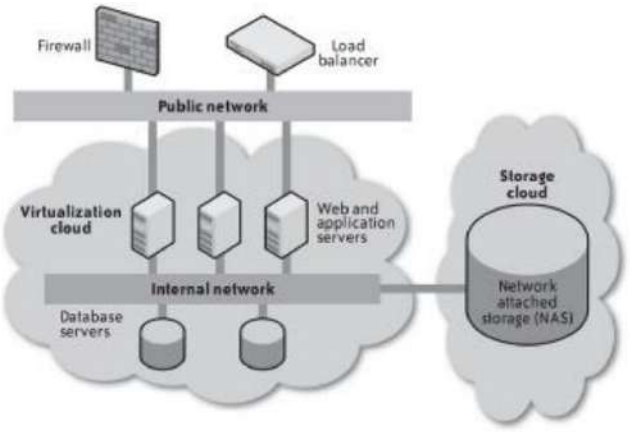
[1] Scale Out
[2] Scale Up
[3] DAS
[4] Batch Processing

گرفته شوند، از سرورهای بزرگتر استفاده شود. همچنین دلیل اینکه این سرورها تمایلی به پویایی (حتی در محیط ابری) ندارند نیز طبیعت آنها است.

GoGrid هر دو بعد مقیاس‌پذیری را پشتیبانی می‌کند. همیشه مقیاس‌پذیری افقی کافی نیست، و مقیاس‌پذیری عمودی نیز در برخی موارد یک تاکتیک مهم است. آمازون با علم بر این موضوع، سرورهای مختلف با اندازه های مختلف در AWS فراهم کرده است. البته مجازی‌سازی نیز یک استراتژی برای چندمنظوره کردن یک سرور است، و به این معناست که می‌تواند چندین مشتری یا برنامه را در بر بگیرد. اما اگر بخشی از برنامه شما بتواند از همه ظرفیت یک سرور فیزیکی استفاده کند، دلیلی ندارد که آن را روی یک سرور فیزیکی اجرا کنیم. در این حالت مجازی‌سازی سربار غیر ضروری روی سیستم اضافه می‌کند و اگر نیاز به توان محاسباتی یا حافظه بیشتری نسبت به آنچه که در ابر ارائه شده است داشته باشید، مجبور خواهید بود که معماری خود را تغییر دهید تا بصورت افقی آن را گسترش دهید. از سویی دیگر، اکثر برنامه‌های کاربردی از برخی جهات از نظر مقیاس‌پذیری عمودی در نهایت به محدودیت بر می خورند و سپس مجبور خواهید بود که آنها را بصورت افقی گسترش دهید. در هر صورت در هر روش از مقیاس‌پذیری، شما باید تقاضا را اندازه گیری کنید و ظرفیت را با آن تطبیق دهید.

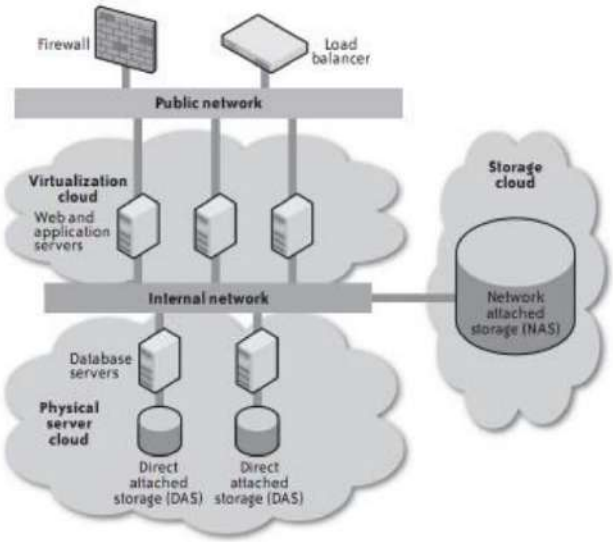
۱۰-۲-۹- معماری های استقرار GoGrid

یک نمونه عمومی از استقرار GoGrid که در شکل ۱۰-۲ نمایش داده شده است، مشابه یک مرکز داده می‌باشد. در مراکز داده فیزیکی معمولی، سرورهای کاربردی در سمت اینترنت قرار دارند تا با کاربران تعامل کنند، در حالی که سرورهای پشتیبانی در سمت DMZ محافظت می‌شوند و همه سیستم‌ها بصورت امن یک NAS را به اشتراک می گذارند. مشابه مراکز داده سنتی، دو بخش شبکه (VLAN) وجود دارد. یکی برای بخش عمومی و اصطلاحاً جلوی شبکه که در سمت frontend قرار دارد و دیگری برای بخش backend که از آدرس‌های IP خصوصی استفاده می‌کند. درست همانند مرکز داده، یک NAS تحت عنوان GoGrid Cloud Storage وجود دارد تا برای کارهای ذخیره‌سازی، آرشیو، تهیه کپی پشتیبان و ... مورد استفاده قرار گیرد.



شکل ۱۰-۲-نقش مجازی‌سازی و ابر در GoGrid

مقیاس‌پذیری عمودی GoGrid خیلی متفاوت با مقیاس‌پذیری افقی نیست، بجز اینکه همه پایگاه‌های داده که کارآیی بالایی نیاز دارند، روی سخت‌افزار فیزیکی اختصاصی اجرا می‌شوند (شکل ۱۰-۳).



شکل ۱۰-۳-نقش میزبانی فیزیکی در GoGrid

۱۰-۲-۱۰- تمرکز بر روی برنامه‌های کاربردی وب

معماری مرکز ابری برای برنامه‌های وب نسبت به کارهای دسته ای مناسب تر است، زیرا برای کارهای دسته ای بسیاری موارد زیرساختی در مراکز داده، نظیر دیواره‌های آتش، تعدیل کننده های بار و VLAN ها را معمولا نیاز ندارند. بسیاری از کارهای دسته ای در محیط‌های توری به خوبی عمل می‌کنند. GoGrid برای کارهای دسته ای نیز قابل استفاده است و البته چنین پردازش هایی در برخی موارد از اجزای مهم بسیاری از برنامه‌های کاربردی وب نیز ممکن است باشند.

۱۰-۲-۱۱- مقایسه رویکردها

وقتی که مرکز ابری (GoGrid) را با سرویس زیرساخت (AWS) مقایسه می‌کنیم، مهم است که هر دو تجربه مراکز داده سنتی و نوع برنامه ای که نصب می‌کنید را بخاطر آورید. شاید نگاهی به مراکز داده معمولی، مراکز ابری و سرویس زیرساخت به شما در این مقایسه کمک کند. در

جدول ۱۰-۱ برخی از قابلیت‌های هر یک از سه نوع زیر ساخت با هم مقایسه شده است.

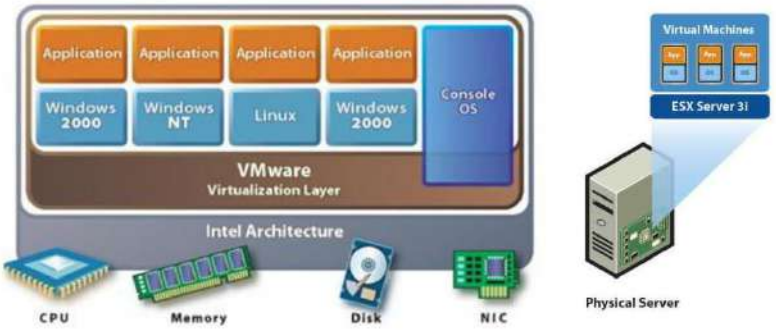
جدول ۱۰-۱- مقایسه سه نوع زیرساخت مختلف

Functionality	Traditional data center	GoGrid (cloud center)	Amazon (service infrastructure)
Firewall	Perimeter hardware firewall	Perimeter hardware firewall (Q1 2009 release)	Custom distributed software firewall
Load balancer	Hardware load balancer	Hardware load balancer	Roll-your-own software load balancer (possible 2009 release of custom load balancer service)
Network isolation	VLAN	VLAN	Faux "VLAN" separation using distributed software firewall
Private networks	Yes (VLAN)	Yes (VLAN)	No
Network protocols	No limitations	No limitations	Restricted; no multicast, no broadcast, GRE and related may not work
OS choices	Unlimited	Some limits	Some limits
DNS	Yes; managed in-house	Yes; managed by GoGrid	No
Persistent local storage	Yes	Yes	No
Persistent network storage	Yes	Yes	Yes
Mixed virtual and physical servers	Yes	Yes	No

تفاوت های بین مراکز ابری و سرویس زیرساخت کاملا مشهود است. با AWS (مدل سرویس زیرساخت)، مقدار کمی از تخصص فعلی شما در شبکه، ذخیره‌سازی و ... قابل استفاده است. شما باید تخصص های جدیدی برای مدیریت S3 و حتی توسعه مهارت های مدیریت سرورها برای مدیریت EC2 بکار بگیرید. در مقابل رویکرد GoGrid (مدل مراکز ابری) بسیار شبیه استفاده از کنسول VMWare VirtualCenter یا دیگر سیستم‌های مدیریت مجازی‌سازی است. علاوه بر سرورها، شما می‌توانید شبکه، DNS، ذخیره‌سازی، دیوارهای آتش و ... را با همان واسط گرافیکی کنترل کنید. در هر حال با هر دو رویکرد، شما قادر خواهید بود که از مزایای مقیاس‌پذیری بر حسب تقاضا و پرداخت بر اساس میزان استفاده که بطور استاندارد در رایانش ابری وجود دارد استفاده کنید.

VMware ESX Server -۳-۱۰

ESX محصولی از VMware است که اجازه مجازی‌سازی را برای نصب چندین سیستم عامل همزمان می‌دهد. این محصول مبتنی بر لینوکس است که می‌توان آن را بر روی سیستم‌های سرور نصب کرد. در حقیقت ESX یک فوق‌ناظر است که می‌توان ماشین‌های مجازی را بر روی آن اجرا کرد. البته برای استفاده از قابلیت‌های آن باید با برنامه VSphere client به آن وصل شد.



شکل ۱۰-۴- معماری کلی برنامه VMware ESX

در شکل مشاهده می‌کنید که با استفاده از برنامه vShpereClient به یکی از سیستم‌هایی که ESX بر روی آن نصب شده است متصل شده ایم و می‌توانیم ماشین‌های مجازی روی آن را مدیریت کنیم.

شکل ۱۰-۵- صفحه برنامه vShpereClient

برنامه vShpereClient تنها به شما اجازه می‌دهد که هر لحظه به یک ماشین فیزیکی متصل شوید و ماشین‌های مجازی روی آن را مدیریت کنید. به این ترتیب می‌توان منابع سیستم را با توجه به ماشین‌های مجازی مدیریت و کنترل کرد و بر روی نحوه استفاده از منابع توسط هر یک از ماشین‌های مجازی بطور دقیق نظارت داشت.

نکته ای که در اینجا باید به آن اشاره کرد این است که ما با نصب ESX بر روی سرورها و استفاده از vShpereClient نمی‌توانیم بگوییم که به رایانش ابری دست پیدا کرده ایم، زیرا رایانش ابری تنها مجازی‌سازی یک سرور نیست، و مجازی‌سازی تنها یکی از تکنولوژی‌هایی است که به استقرار محاسبات‌ابری کمک می‌کند.

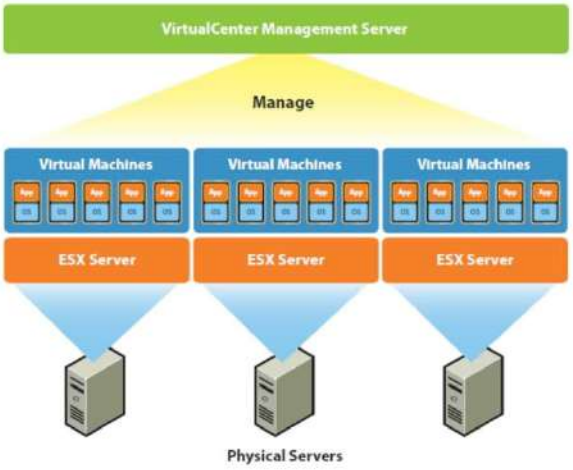
شکل ۱۰-۶- صفحه برنامه vShpereClient در حال نظارت بر وضعیت استفاده از منابع توسط ماشین‌های مجازی

به چنین ابزارهایی که تنها به ما کمک می‌کنند تا یک سرور فیزیکی را بصورت چندین سرور مجازی مورد استفاده قرار دهیم، می‌توان صرفا ابزارهای مدیریت مجازی‌سازی نامید. البته کار به همین جا ختم نمی‌شود و VMware محصول دیگری را نیز ارائه داده است که ما را یک قدم به رایانش ابری نزدیک تر می‌کند. این محصول تحت عنوان VCenter در ادامه معرفی شده است.

VCENTER -۱-۳-۱۰

همانطور که گفته شد با VSphere Client شما فقط می‌توانید در آن واحد فقط به یک سرور ESX متصل شوید. برای اینکه بخواهید در آن واحد چندین سرور ESX را با هم مدیریت کنید، باید یک سرور Vcenter راه اندازی کنید و ESX ها را به آن وصل کنید. حال اگر شما به VCenter متصل شوید می‌توانید چند

ESX را با هم ببینید. شما می‌توانید سرور VCenter را در یکی از ESX ها به صورت ماشین مجازی راه اندازی کنید یا اینکه آن را در یک ماشین فیزیکی مجزا نصب کنید.



شکل ۱۰-۷- شمای کلی عملکرد سرور VCenter

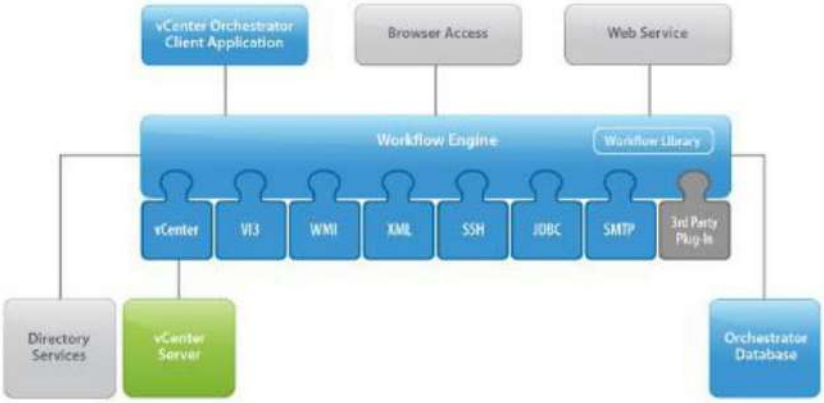
همانطور که در تصاویر مشاهده می‌کنید، سرورVCenter به شما امکان مدیریت متمرکز و یکپارچه مجموعه سرورهای ESX را می‌دهد. با این کار می‌توان از یک دید سطح بالا، ماشین‌های مجازی و کلیه منابع سخت‌افزاری موجود را یکجا مدیریت کرد و کنترل بسیار خوبی را بر روی منابع سازمان فراهم می‌آورد.

شکل ۱۰-۸- تصویری از برنامه vSphere Client که به سرور VCenter متصل شده است

نکته مهم اینجا این است که با اینکه ما یک دید متمرکز نسبت به کلیه منابع بدست آورده ایم، آیا هنوز می‌توانیم بگوییم که این ساختار رایانش ابری در لایه IaaS است؟ پاسخ منفی نیست. زیرا اگر چه ما به آنچه در مورد ویژگی‌های رایانش ابری در لایه IaaS اشاره کردیم به طور کامل دست پیدا نکرده ایم، ولی بسیار به آن نزدیک شده ایم. به عبارت دیگر ما یک زیرساخت بسیار قابل انعطاف و مجازی در اختیار داریم که می‌توانیم آنرا ابر خصوصی در داخل یک سازمان بنامیم که از طریق آن سرویس‌های سازمان بطور قابل انعطاف قابل ارائه است. تنها اشکال عمده ای که بر این زیرساخت وارد است، این است که کارها در آن بطور خودکار انجام نمی‌شود. این نیازمندی توسط محصول دیگری بنام Orchestrator بر طرف شده است که در ادامه توضیح داده خواهد شد.

Orchestrator -۲-۳-۱۰

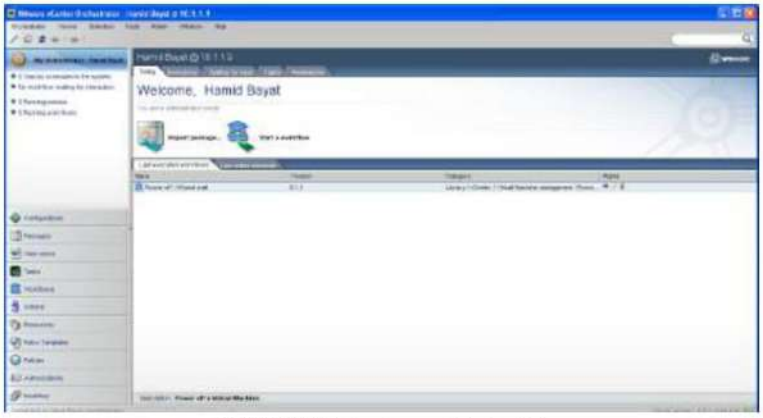
این برنامه بیشتر برای خودکار کردن جریان فعالیت ها و مدیریت بهتر مجازی‌سازی منابع بکار می‌رود. بنابراین هرچه که ما تاکنون بصورت دستی انجام می‌دادیم (ایجاد ماشین مجازی، اجرا، حذف، کنترل وضعیت و ...) به کمک این محصول می‌توان بصورت اتوماتیک انجام داد. برای این کار باید فرآیند یا جریان کاری که باید انجام شود را بطور دقیق مشخص کنید و با استفاده از ابزارهای ارائه شده در Orchestrator پیاده‌سازی نمایید. مدیریت اجرای این جریان کار توسط Workflow Engine که هسته مرکزی Orchestrator را تشکیل می‌دهد انجام می‌شود.



شکل ۱۰-۹- معماری کلی برنامه Orchestrator

در شکل ۱۰-۱۰ تصویری از صفحه برنامه Orchestrator را که از طریق آن می‌توان مدیریت جریان کارها را انجام داد مشاهده می‌نمایید. ایجاد جریان کار در این برنامه از طریق اجزای از پیش آماده شده و نیز

APIها که بیش از ۸۰۰ فعالیت مختلف را تحت پوشش قرار می‌دهند امکان‌پذیر است. همچنین می‌توان برای حالت های پیشرفته تر از اسکریپت نویسی نیز در آن استفاده کرد.



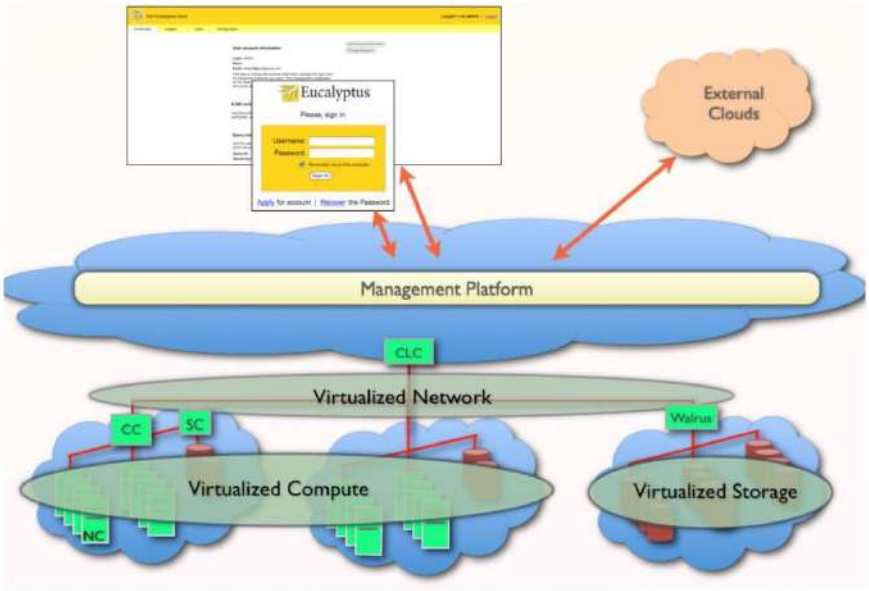
شکل ۱۰-۱۰-تصویری از برنامه Orchestrator

Eucalyptus -۴-۱۰

امروزه روش های مختلفی برای توزیع داده و توان پردازشی وجود دارد. کاربران عموما می‌توانند منابع را بر اساس مشخصه های مختلف آنها، از قبیل مشخصه های سخت‌افزاری، میزان حافظه و ظرفیت، نوع اتصال شبکه و یا در برخی موارد موقعیت فیزیکی آنها پیدا کنند. محاسبات توری از جمله این روش ها بحساب می‌آیند. منابع در محاسبات توری غیرهمگن است. بنابراین فرآیند برنامه‌نویسی و استفاده از این منابع به منظور استفاده بهینه از آنها بسیار پیچیده و مشکل است. علی رغم اینکه برخی کاربران تخصص لازم برای بهره برداری از این منابع غیرهمگن را دارند، خیلی از کاربران ترجیح می‌دهند که محیط این منابع سخت‌افزاری و نرم‌افزاری بطور یکپارچه در اختیارشان قرار گیرد. در یک محیط یکپارچه فعالیت های مربوط به توسعه برنامه‌هایی حتی در مقیاس خیلی بزرگ بسیار ساده‌تر است. رایانش ابری مبتنی بر همین رویکرد است تا بتواند با ارائه یک سطح انتزاعی تر از معرفی منابع، یک محیط یکپارچه برای توسعه چنین برنامه‌هایی فراهم آورد. اما معماری توزیع شده مناسب برای رایانش ابری چگونه است؟ چه مشخصه هایی از منابع موجود باید در ماشین‌های مجازی در نظر گرفته شود تا بتوان از منابع بطور بهینه استفاده کرد. شبکه ماشین‌های مجازی چگونه باید باشد تا قابل انعطاف، کارا و ایمن باشد؟ و بسیاری سئوالات از این قبیل از جمله سئوالاتی است که در خصوص رایانش ابری مطرح است و سعی شده است در سیستم اکالیپتوس به آنها پاسخ داده شود. به کمک این سیستم کاربران قادر خواهند بود تا ماشینهای مجازی قرار داده شده بر روی منابع فیزیکی مختلف را اجرا نموده و کنترل کنند. اکالیپتوس خدماتی از نوع IaaS را ارائه می‌دهد. معماری اکالیپتوس بگونه‌ای است که قابل حمل، ماژولار و با کاربری آسان و متناسب با زیرساخت‌های موجود در محیط‌های آکادمیک می‌باشد. اکالیپتوس از معدود بسته‌های نرم‌افزاری کدمتن بازی است که امکان استفاده از آن بر روی چندین کلاستر با همدیگر نیز وجود دارد. اکالیپتوس دارای یک معماری باز و ساده برای پیاده‌سازی قابلیت‌های مختلف توده‌های ابری در سطح زیرساخت است و در آن از تکنولوژی وب سرویس‌ها استفاده شده است.

محیط کاری Eucalyptus یک محیط کاری نرم‌افزاری کد باز برای پیاده‌سازی رایانش ابری مفروض است و از زیرساخت فناوری اطلاعات سازمان -بدون نیاز به تغییر، استفاده از سخت‌افزار خاص منظوره یا بازپیکره‌بندی- استفاده می‌کند. ابر Eucalyptus منابع مراکز داده مانند ماشین‌ها، شبکه‌ها و سیستم‌های ذخیره‌سازی را به ابری تبدیل می‌کند که توسط فناوری اطلاعات محلی قابل کنترل و خصوصی‌سازی است. این ابر تنها معماری ابری است که مانند ابرهای عمومی از واسط‌های برنامه‌نویسی برنامهٔ کاربردی (API) پشتیبانی می‌کند و در حال حاضر با زیرساخت ابر سرویس‌های وب Amazon کاملاً سازگار است.

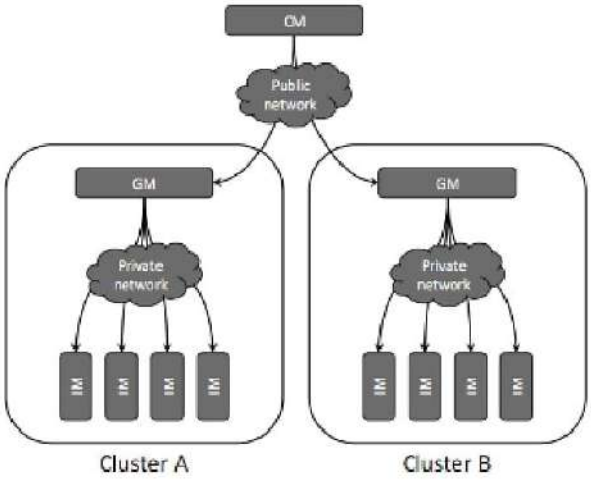
اکالیپتوس به کاربران و مدیران امکان ایجاد زیرساختی را فراهم آورده است که از طریق آن بتوانند ماشین‌های مجازی قابل کنترل توسط کاربران را ایجاد کرده و بر منابع موجود کنترل داشته باشند. سیستم کاملا ماژولارو دارای API های خوش تعریفی می‌باشد به گونه ای که محققان بتوانند نتایج تحقیقات خود و اجزای طراحی شده توسط خود را توسط آن مورد آزمایش و ارزیابی قرار دهند.



شکل ۱۰-۱۱- مدل مفهومی ابر اکالیپتوس. CLC کنترلر ابر است که به عنوان واسط اصلی ارتباط با منابع مجازی شده بحساب می‌آید. CC ها کنترل کننده های مربوط به هر مجموعه از منابع هستند. NC ها ماشین‌های نهایی هستند که ماشین‌های مجازی روی آنها اجرا خواهد شد. SC کنترلر ذخیره‌سازی است (مشابه EBS در آمازون) که توسط Walrus کنترل می‌شوند (Walrus مشابه S3 در آمازون است). یک پلت‌فرم مدیریتی برای استفاده از ابر فراهم شده است که سطوح دسترسی مختلف را به ابر برای مدیریت، توسعه و ... فراهم می‌کند.

اکالیپتوس بر روی منابعی مختلفی از یک لپ تاپ گرفته تا کلاسترهای لینوکسی با ۴۸ تا ۶۴ نود با موفقیت پیاده‌سازی شده است و به این ترتیب افراد مختلفی می‌توانند با منابعی که در اختیار دارند این سیستم را بکارگیرند و توده ابری خود را ایجاد کنند. امکان نصب این سیستم بدون نیاز به نصب نرم‌افزارهای اضافی فراهم است و بنابراین می‌توان گفت که اکالیپتوس می‌تواند نقش مهمی را در ایجاد یک جامعه تحقیقاتی در خصوص رایانش ابری فراهم آورد.

در این سیستم، کاربران با استفاده از یک مرورگر وب می‌توانند وارد توده ابری شوند(Sign Up)، مجوزهای رمزنگاری مورد نیاز برای واسط برنامه‌نویسی را دریافت کنند و پرس‌وجوهای خود را اجرا کنند (مثلا در خصوص تصاویر ماشین مجازی در دسترس). همچنین مدیران می‌توانند علاوه بر این کارها، حساب های کاربران را نیز مدیریت کنند و حتی آنها را غیرفعال و یا حذف کنند.



شکل ۱۰-۱۲- ساختار سلسله مراتبی اکالیپتوس برای بکارگیری منابع در یک توده ابر

اکالیپتوس شامل اجزای زیر می‌باشد:

- مدیریت نمونه ها[1]: برای مدیریت هر یک از نمونه های ماشین مجازی یک IM اختصاص داده شده است.
- مدیریت گروه[2]: برای مدیریت و زمان بندی هر مجموعه از ماشین‌های مجازی (شبکه ای از نمونه های مجازی) از GM استفاده می‌شود.
- مدیریت ابر[3]: نقطه ورود به هر توده ابری برای کاربران و مدیران سیستم می‌باشد. امکان درخواست وضعیت منابع، زمان بندی سطح بالا و ... از طریق CM امکان‌پذیر است.

سیستم اکالیپتوس، در تهیه زیرساخت مورد نیاز برای بکارگیری منابع موجود در قالب یک توده ابری مورد استفاده قرار می‌گیرد. بنابراین همانطور که گفته شد محققان می‌توانند ماژول هایی که خود طراحی کرده‌اند در آن بکارگیرند و مورد ارزیابی قرار دهند. اما نباید فراموش کرد که رایانش ابری برای تسهیل کسب و کار فراهم آمده اند. بنابراین راه‌حل‌های مبتنی بر ابر می بایست متناسب با شرایط کسب و کار باشند. در یک

[1] Instance Manager
[2] Group Manager
[3] Cloud Manager

سازمان منابع مختلفی وجود دارد. به نظر میرسد اکالیپتوس صرفا منابع ذخیره‌سازی و پردازشی را به منظور ایجاد نمونه ماشین‌های مجازی در نظر گرفته است ولی برای بکارگیری منابع داده ای و پردازشی موجود و یکپارچه‌سازی آنها با شرایط ابری هنوز راه‌حلی ارائه نداده است که این موضوع بسیار مهمی است که نیازمند تحقیق و بررسی جداگانه جهت ارائه یک راه‌حل مناسب و مقرون بصرفه متناسب با شرایط موجود سازمان‌ها می‌باشد.

مراجع

[1] Daniel Nurmi and et al, "The Ecalyptus Open-source Cloud-computing System," Proceedings of Cloud Computing and Its Applications, Chicago, Illinois, 2008

[2] Daniel Nurmi and et al, "Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems," UCSB Computer Science Technical Report Number 2008

[3] Eucalyptus Public Cloud: eucalyptus.cs.ucsb.edu/wiki/EucalyptusPublicCloud

فرهنگ اصطلاحات مورد استفاده

اصطلاح	شرح
Accounting	حسابرسی
Amazon Machine Image (AMI)	ماشین مجازی آمازون
Appliances	ابزارهای مجازی
Application Programming Interface	واسط برنامه‌نویسی برنامه کاربردی
Application virtualization	مجازی‌سازی برنامه کاربردی
Authentication	احراز هویت
Authorization	اعتبارسنجی
Bare metal	سخت‌افزار فیزیکی
Batch jobs	کارهای دسته‌ای، وظایف دسته‌ای
Batch Processing	محاسبات دسته‌ای، پردازش دسته‌ای
Bookmark	نشانه
Business Continuity	تداوم کسب و کار
Business Continuity Management	مدیریت تداوم کسب و کار
CapEx	هزینه‌های سرمایه‌گذاری
Certification Authority (CA)	مرکز صدور گواهی‌نامه دیجیتال
Cloning	تکثیر
Cloud Computing	رایانش ابری
Cloud Customer	مشتری ابری
Cloud Provider	سرویس‌دهنده ابری
Cloud Service	سرویس ابری
Cloud Storage	ذخیره‌سازی ابری، ابر ذخیره‌سازی
Cloud Center	مرکز ابری
Cluster Computing	محاسبات کلاستری
Collaborative annotation	درج توضیحات بصورت گروهی
Co-location	مکان اشتراکی
Compliance	موافقت

اصطلاح	شرح
Content Distribution Network (CDN)	شبکه توزیع محتوا
Cookie	کوکی، کلوچه
Cooling	سرمایش
Cooperative Virtualization	مجازی‌سازی ترکیبی
Cross-language	میان‌زبانی
Data Center	مرکز داده
Deployment	استقرار، تحویل، ارائه
Desktop	میزکار، دسک تاپ
Disaster Recovery	بازیابی از سوانح
Edge Networks	شبکه‌های لبه
Elastic	قابل ارتجاع، قابل انعطاف
Grid Computing	محاسبات توری
Groupware	گروه افزار
Guest	میهمان
Hardware Security Module (HSM)	ماژول امنیت سخت افزاری
Hardware virtualization	مجازی سازی سخت افزاری
Hardware-assisted virtualization	مجازی سازی سخت افزاری
Horizontal Scaling	مقیاس پذیری افقی
Host	میزبان
Hypertext Transfer Protocol	پروتکل انتقال ابرمتن
Hypervisor	فوق ناظر
Identity Management	مدیریت هویت
Image	تصویر
Incremental	افزایشی
Infrastructure as a service	زیرساخت بعنوان سرویس
Instant Messaging (IM)	پیغام رسانی فوری
Intellectual property	مالکیت معنوی
Interoperability	قابلیت همکاری
Intrusion	نفوذ

اصطلاح	شرح
IT as a Service	منابع IT به عنوان سرویس
Load Balancer	تعدیل کننده بار
Loosely coupled	کمترین جفت شدگی
Malicious	مضر، بداندیش، بدخواه
Mean time between failures (MTBF)	زمان متوسط بین خرابی
Migrating	مهاجرت
Monitoring	نظارت، مانیتورینگ
Multiple login	ورود چندگانه
Notification	هشدار
Notification System	سیستم هشدار
Offline	آفلاین
On Demand Deployment	استقرار بر حسب تقاضا
Online	آنلاین
OpEx	هزینه‌های عملیاتی
Packaging	بسته‌بندی
Patch	وصله
Patch management	مدیریت وصله
Pay as you go	پرداخت بر حسب تقاضا
Platform	سکو، پلت فرم
Platform as a service	سکو به عنوان سرویس
Pool	استخر
Pool of resources	استخری از منابع
Power	توان
Pricing	قیمت‌گذاری
Private Cloud	ابر خصوصی
Proactive	پیش فعال
Public Cloud	ابر عمومی
Reactive	واکنشی
Real Time	بلادرنگ

اصطلاح	شرح
Recovery	بازیابی
Redundancy	افزونگی
Replication	تکرار
Resilience	حالت ارتجاعی
Resource Description Framework (RDF)	چهارچوب توصیف منابع
Role-based access control (RBAC)	کنترل مبتنی بر نقش
Rollback	عقبگرد
Runtime	زمان اجرا
Scale on Demand	مقیاس‌پذیری بر حسب تقاضا
Scale Out	مقیاس پذیری افقی
Scale Up	مقیاس پذیری عمودی
Service Infrastructure	زیرساخت سرویس
Service Level Agreement (SLA)	سطح توافق سرویس
Snapshots	تصویر لحظه‌ای
Software as a Service	نرم‌افزار به عنوان سرویس
Storage Virtualization	مجازی سازی ذخیره سازی
Surge Computing	محاسبات ناگهانی
Tenant	مستاجر
Terms of Use (ToU)	شرایط استفاده
Threat management	مدیریت تهدید
Traffic shaping	شکل‌دهی به ترافیک
Transactional Computing	محسابات تراکنشی
Untrusted software	نرم افزار غیرقابل اطمینان (ناپایدار)
Usage-based model	مدل مبتنی بر استفاده
Utility	سودمند
Utilization	بهره برداری، مصرف
Virtual Appliances	ابزارهای مجازی
Virtual Machine Image	تصویر ماشین مجازی
Virtual Machine (VM)	ماشین مجازی

شرح	اصطلاح
مجازی‌سازی	Virtualization
سرویس وب	Web Service
وب تاپ، میزکار تحت وب	Webtop

شاخص ها

منابع تکمیلی جهت مطالعه بیشتر در جامعه آزاد رایانش ابری ایران:

وبسایت رسمی جامعه آزاد رایانش ابری ایران	http://www.occc.ir
دانشنامه آزاد رایانش ابری ایران	http://wiki.occc.ir
سایت پرسش و پاسخ تخصصی در حوزه رایانش ابری	http://ask.occc.ir
وبلاگ انگلیسی جامعه آزاد رایانش ابری ایران	http://blog.occc.ir/
اسناد و اسلایدهای مرتبط	http://docs.occc.ir
سیستم تجمیع اخبار	http://news.occc.ir
سیاره ابری	http://planet.occc.ir
شبکه رایانش	http://tv.occc.ir
بوردهای تخصصی و آزاد جامعه رایانش ابری	http://link.occc.ir/board
گروه پستی جامعه آزاد رایانش ابری ایران	http://link.occc.ir/mailinglist