**جواب سوال ۱:**

میدونیم که هر IP address v4از ۴قسمت که به هر قسمت اوکتت میگیم تشکیل شده. به عبارتی این ادرس از ۴قسمت یک بایتی تشکیل شده (هر بایت=۸بیت).

منظور از اوکتت همین ۸بیته.

پس درمجموع هر IP address از ۳۲بیت تشکیل شده.

از طرفی هر IP address شامل دو قسمت میشه:

۱- net id: آدرس کلی شبکه

۲- host id: برای آدرس‌دهی به نودها و هاست های شبکه

آدرس ip توسط چیزی بنام subnet mask مشخص میشه که کدوم قسمت net id و کدوم قسمت host id هست.

اما چگونه؟ در subnet mask معادل تمامی بیت هایی که قراره بعنوان net id استفاده بشه، یک هست و برای host id صفر.

بطور مثال 255.255.255.0 یک subnet mask هست. واین یعنی ۲۴ بیت اول بعنوان net id و باقیمانده بیتها بعنوان host id درنظر گرفته میشن.

بطور مثال: IP address 212.128.19.20 روی sub mask مذکور یعنی:

212.128.19.0/24 آدرس اون شبکه اصلیه هست.

ما ۳نوع کلاس برای IP addressها داریم که هرکدومشون subnet mask مخصوص خودشونو دارن.

بطور خلاصه هرکلاس با subnet maskهاشون رو بخوام بگم:

کلاسc: بصورت 255.255.255.0

کلاسb: بصورت 255.255.0.0

کلاسa: بصورت 255.0.0.0

این ۳نوع کلاس رو‌بعنوان classfull میشناسن.

بطور مثال آدرس شبکه مربوط به ای پی 212.128.19.20 روی هرکدوم از کلاسها با توجه به subnet maskشون به این شکله:

کلاسc: بصورت 212.128.19.0/24

کلاسb: بصورت 212.128.0.0/16

کلاسa: بصورت 212.0.0.0/8

بعضی اوقات وقتی از یک کلاس برای آدرسدهی شبکه استفاده میکنیم، میبینیم تعداد بیت هایی که برای هاست هامون داریم استفاده میکنیم بیشتر یا کمتر از حد نیازمون هست. برای همین نیاز داریم تعداد بیت ها رو کم یا زیادشون کنیم. برای اینجور دستکاری کردن توی بیت هایی که طبق استاندارد ۳کلاس مذکور هستن، از متودی بنام CIDR(classless inter-domain routing) استفاده میکنیم که درنتیجه آن آدرسهای حاصله دیگه از subnet mask هیچکدوم از ۳کلاس اصلی پیروی نمیکنند و به همین دلیل بهشون میگن classless.

بطور مثال در مثال بالا میخوایم به آدرس مربوط به کلاسc دوبیت اضافه کنیم تا بتونیم به هاست های بیشتری آدرس‌دهی داشته باشیم. پس آدرس شبکه تبدیل میشه به:

212.128.19.0/22

میخوایم از آدرس مربوط به کلاسa، نه بیت کم کنیم تا چون تعداد بیتها خیلی بیش از حد نیازمونه. پس آدرس شبکه تبدیل میشه به:

212.0.0.0/17

**جواب سوال ۲:**

REST يك مدل معماري براي طراحي برنامه‌هاي كاربردي شبكه است كه ترکیبی از چند مدل معماری مبتنی بر شبکه مي‌باشد و محدودیت‌هايی جهت تعریف يك واسط اتصال یکنواخت براي آن درنظر گرفته شده است.

REST مخفف شده‌ی عبارت Representational State Transfer و به معنای "انتقال بازنمودی حالت" است.

ايده اصلي معماري REST اين است كه به جاي استفاده از مكانيزم هاي پيچيده اي مانند CORBA، RPC يا SOAP براي اتصال ماشين ها ، ازHTTP ساده براي برقراري ارتباط بين ماشين‌ها استفاده شود. مدل REST شش قيد براي معماری برنامه‌هاي شبكه تعريف مي‌كند: كلاينت سرور (client-server) باشد. بدون حالت (stateless) باشد. قابليت cache داشته باشد. (cacheable) سيستم لايه‌بندي شده (layered system) داشته باشد. واسط يكنواخت (uniform interface) داشته باشد. داراي قابليت كد در صورت نياز (code on demand) باشد. (كه اين محدوديت اختياري مي‌باشد) به سيستمي كه اين قيود را رعايت نمايد، RESTful مي‌گويند. از لحاظ رويكرد برنامه نويسي REST جايگزيني ساده براي سرويس‌هاي وب است. و همچنين استفاده از معماري REST در برنامه‌نويسي كارايي، سادگي، انعطاف‌پذيري، امكان مشاهده و نظارت، قابليت حمل و قابليت اطمينان را افزايش مي دهد.

SOAP پروتکل دسترسی آسان به اشیاء و مخفف عبارت Simple Object Access Protocol می باشد.

این پروتکل برای رد و بدل کردن اطلاعات بین برنامه ها استفاده می شود. اطلاعات در SOAP به صورت پیام (Message) و از طریق پروتکل های موجود در اینترنت مانند HTTP منتقل می‏شود. به زبان ساده‏ تر، SOAP یک پروتکل برای دستیابی به یک سرویس ارایه شده در وب (Web Service) می باشد. پروتکل SOAP یکی از عمومی ترین استاندارد هایی است که در وب سرویس ها استفاده می شود.

وقتی یک برنامه شروع به ارتباط با وب سرویس می کند، پیغام های SOAP وسیله ای برای ارتباط و انتقال دیتا بین آن دو هستند. یک پیغام SOAP به وب سرویس فرستاده می شود و یک تابع را در آن به اجرا در می آورد، وب سرویس نیز از محتوای پیغام SOAP استفاده کرده و عملیات خود را آغاز می کند و در انتها نیز نتایج را با یک پیغام SOAP دیگر به برنامه اصلی می فرستد.

SOAP یک پروتکل ارتباطی است.

برای ارسال پیام استفاده می‏شود.

برای محیط اینترنت و شبکه طراحی شده است.

وابسته به محیط پیاده سازی و اجرا نیست. ( Platform Independent)

مبتنی بر XML است.

از دیوارهای آتش (Firewall) گذر می‏کند ودیوارهای آتش مانع آنها نمی شوند (Block نمی‏شوند).

مزایای وب سرویس RESTful:

سرعت: خدمات وب RESTful سریع هستند زیرا هیچ ویژگی سخت افزاری مانند SOAP وجود ندارد. REST پهنای باند و منابع کمتری مصرف می کند.

زبان و پلت فرم مستقل: وب سرویس RESTful را می توان در هر زبان برنامه نویسی نوشته و در هر پلت فرم اجرا کرد.

می توان از SOAP استفاده کرد: وب سرویس های RESTful می توانند از وب سرویس های soap برای پیاده سازی استفاده کنند.

اجازه دسترسی به فرمت های مختلف دیتا: وب سرویس RESTful اجازه دسترسی به فرمت داده های مختلف مانند متن ساده، HTML، XML و JSON را می دهد .

مزایای وب سرویس SOAP:

WS Security: SOAP امنیت خود را به نام WS Security تعریف می کند.

زبان و پلت فرم مستقل: وب سرویس های SOAP را می توان در هر زبان برنامه نویسی نوشته و در هر پلت فرم اجرا کرد.

معایب وب سرویس SOAP:

کند بودن: SOAP از فرمت XML استفاده می کند که باید خوانده شود. XML استانداردهای بسیاری را تعریف می کند که باید در هنگام توسعه برنامه های SOAP دنبال شود. بخاطر همین کند است و پهنای باند و منابع بیشتری را مصرف می کند.

وابستگی به WSDL : SOAP از WSDL استفاده می کند و مکانیزم دیگری برای کشف سرویس ندارد.

**تفاوت‌ها و شباهت‌ها:**

مشخصات SOAP:

SOAP یک پروتکل است

SOAP برپایه دسترسی ساده به اشیاست

SOAP نمیتواند از rest استفاده کند چون یک پروتکل است

SOAP با استفاده از services interface ها برای نمایش منطق کسب و کار استفاده می کند.

JAX-WS یک api جاوا برای وب سرویس soap است

SOAP استانداردها را به شدت دنبال می کند.

SOAP نیاز به پهنای باند و منابع بیشتری نسبت به REST دارد.

SOAP امنیت خود را تعریف می کند.

SOAP فقط فرمت XML را پشتیبانی میکند.

SOAP کمتر از REST ترجیح داده می شود.

مشخصات REST:

REST یک سبک معماری است

REST بر پایه انتقال بازنمودی حالت است.

REST می تواند از وب سرویس SOAP استفاده نماید زیرا یک مفهوم است و می تواند از هر پروتکل مانند HTTP، SOAP استفاده کند.

REST از URI برای نمایش منطق کسب و کار استفاده می کند.

JAX-RS یک api جاوا برای وب سرویس REST است

REST مانند SOAP استانداردهای بیشماری را تعریف نمی کند.

REST نیاز به پهنای باند و منابع کمتر از SOAP دارد.

وب سرویس RESTful اقدامات امنیتی را از انتقال اصولی به ارث می برند.

REST فرمت های مختلف مانند متن ساده، HTML، XML، JSON و غیره را نیز پشتیبانی میکند.

REST بیشتر از SOAP ترجیح داده می شود.

**جواب سوال ۳:**

Status code های HTTP معمولا به 5 دسته تقسیم می شوند:

دسته ی 100: کدهای این قسمت معمولا شامل اطلاعاتی درباره ی شروع درخواست و ارسال آن هستند.

دسته ی 200: این کدها زمانی برگردانده می شوند که درخواست و پردازش آن موفقیت آمیز بوده باشد.

دسته ی 300: این دسته از کدها معمولا به redirect ها مربوط هستند، زمانی که یک resource با resource دیگری جایگزین شده است.

دسته ی 400: این کدها مشخص می کنند که درخواست ارسال شده دارای مشکل بوده است (خطاهای کلاینت).

دسته ی 500: این کدها می گویند که درخواست صحیح بوده است اما در هنگام پردازش آن در سمت سرور با خطا مواجه شده ایم (خطاهای سرور).

تعداد تمامی کدهای وضعیت حدود 40 کد است اما همه ی آن ها برای ما مهم نیستند.مهمترین آنها براساس نتیجه سرچ‌ها:

کدهای دسته ی 200

از این دسته معمولا فقط با کد 200: “Everything is OK.” مواجه می شویم. این کد فقط زمانی برگردانده می شود که resource های وب دقیقا به شکل صحیح کار کنند، زمانی که درخواست کاملا صحیح و پردازش آن نیز صحیح بوده است و هیچ خطایی در هیچ قسمتی از برنامه نداشته ایم.

کدهای دسته ی 300

کد 301: این کد معمولا با پیام The requested resource has been moved permanently همراه است که به ما می گوید resource مورد نظر ما (مثلا یک تصویر یا یک صفحه ی وب یا یک فایل و...) جابجا شده است. معمولا از این کد برای redirect کردن دائمی انواع url ها استفاده می شود.

کد 302: این کد با پیام The requested resource has moved, but was found. همراه است و می گوید که resource مورد نظر ما پیدا شده است البته در جایی غیر از آدرس مشخص شده! معمولا از این کدها در redirect کردن موقت url ها استفاده می شود.

کد 304: این کد معمولا با پیام The requested resource has not been modified since the last time you accessed it همراه است و کار آن اعلام عدم تغییر browser cache است. از این کد استفاده می شود تا به مرورگر اعلام کنیم resource های ما از بازدید قبلی تغییری نداشته اند و مرورگر می تواند از همان resource های دانلود شده ی قبلی استفاده کند تا سرعت بارگذاری صفحات بالاتر برود.

کدهای دسته ی 400

کد 401: این کد معمولا با پیام Unauthorized یا Authorization Required ارسال می شود. این پیام به ما می گوید که resource مورد نظر نیاز به احراز هویت دارد و مدارک ارسال شده معتبر نیست. اگر از HTTP authentication استفاده می کنید احتمالا با چنین خطاهایی روبرو شده اید. این مسئله معمولا زمانی اتفاق می افتد که دسترسی به سایت خود را از روی حالت public تغییر داده باشید (مثلا زمانی که می خواهید وب سایت فقط در دسترس افراد خاصی باشد).

کد 403: پیام Access to that resource is forbidden که همراه این کد وضعیت ارسال می شود به ما می گوید که کاربر برای دسترسی به یک resource مجوزی ندارد. مثلا بدون login کردن سعی در اتصال به صفحه ی خاصی داشته ایم.

کد 404: شایع ترین و شناخته شده ترین کد وضعیت در دنیا است و به کاربر می گوید The requested resource was not found یعنی سرور چنین resource ای ندارد و اصلا نمی داند که این resource تا به حال وجود داشته است یا خیر! مثلا آدرس https://www.roxo.ir/askavskbafbkuafdsuhfudshfdsku854asf در روکسو وجود ندارد بنابراین با خطای 404 مواجه خواهید شد.

کد 405: معمولا با پیام Method not allowed برگردانده می شود. این اتفاق زمانی می افتد که سرور میزبانی از متد خاصی پشتیبانی کند اما resource هدف چنین متدی را قبول نمی کند.

کد 406: معمولا با پیام Not acceptable response ارسال می شود و می گوید resource ما می تواند تنها محتوایی را تولید کند که بر اساس header های Accept، غیرقابل قبول هستند.

کد 408: این کد با پیام The server timed out waiting for the rest of the request from the browser به کاربر ارسال می شود و می گوید که سرور درخواست کامل را از طرف مرورگر دریافت نکرده است. ممکن است در زمان تراکم نت packet های داده به طور کامل ارسال نشوند و با چنین خطایی روبرو شویم.

کد 410: این کد با پیام The requested resource is gone and won’t be coming back ارسال می شود و می گوید که resource مورد نظر موجود نیست. البته تفاوت آن با کد 404 در این است که شرایط پیش آمده از قبل پیش بینی شده بوده و یک اتفاق ناخواسته نیست.

کد 429: پیام Too many requests مشخص می کند که این کد برای اعلام «تعداد درخواست ها بیش از حد مجاز» ارسال می شود. به زبان ساده تر، هنگامی که تعداد درخواست های ارسال شده از سمت کاربر بیش از حد مجاز باشد، این کد برگردانده خواهد شد.

کد 499: پیام همراه این کد Client closed request است. این کد توسط وب سرور NGINX برگردانده می شود و می گوید که کلاینت (کاربر) درخواست را زمانی بسته که NGINX در حال پردازش آن بوده است.

کدهای دسته ی 500

کد 500: پیام There was an error on the server and the request could not be completed همراه این کد ارسال می شود. این کد یک کد کلی است که می گوید سرور نتوانسته است درخواست را به طور کامل پردازش کند (به هر دلیلی). این مشکل ممکن است به دلیل پلاگین های وردپرس یا PHP بد یا قطع اتصال به پایگاه داده رخ دهد.

کد 501: این کد به همراه پیام Not Implemented ارسال می شود. در واقع این کد می گوید سرور دارای توانایی لازم برای پردازش و انجام درخواست نمی باشد بنابراین تقریبا همیشه به دلیل وجود خطا روی سرور است و باید توسط خود مدیر یا برنامه نویس سایت حل شود.

کد 502: که کد معروف Bad Gateway است و زمانی رخ می دهد که یک سرور پاسخ نامناسبی را از یک سرور دیگر دریافت کرده باشد. مثلا برخی اوقات پردازش یک درخواست بیش از حد طول می کشد و سرور نیز آن را قطع می کند، در چنین حالتی نیز کد 502 برگردانده می شود.

کد 503: پیام The server is unavailable to handle this request right now مشخص کننده ی این نوع کد وضعیت است و می گوید که سرور توانایی پردازش درخواستی را ندارد. به طور مثال زمانی که تعداد درخواست ها به حد نصاب برسد و سرور توانایی پردازش درخواست های بیشتری را نداشته باشد.

کد 504: به همراه پیام The server, acting as a gateway, timed out waiting for another server to respond ارسال می شود. این وضعیت زمانی پیش می آید که دو سرور مختلف درگیر پردازش یک درخواست باشند و سرور اول آنقدر منتظر پاسخ سرور دوم بماند که time out کند (زمان پردازش بیش از حد شده و اتصال قطع شود).

کد 521: پیام Web server is down مشخص می کند کارایی این کد چیست؛ این کد مخصوص سرویس Cloudflare است و می گوید که مرورگر با موفقیت به Cloudflare متصل شده است اما Cloudflare نمی تواند به سرور اصلی متصل شود.

**جواب سوال ۴:**

می‌توان گفت که سوکت به ترکیب یک آدرس ماشین (آی‌پی یا IP) و یک شماره درگاه (پورت یا Port) گفته می‌شود. در برقراری ارتباط بین کامپیوترها در یک شبکه، دو چیز بسیار مهم است:

آدرس ماشینی که می‌خواهیم اطلاعاتی از آن بگیریم یا به آن ارسال کنیم.

برنامه‌ای از آن ماشین که درخواست اطلاعات کرده؛ و یا اینکه می‌خواهیم اطلاعاتی از آن برنامه کسب کنیم.

این دو، یعنی آدرس ماشین و شماره برنامه، به وسیله‌ی سوکت در شبکه مشخص می‌شوند. سوکت یک ارتباط قابل اطمینان جهت انتقال داده‌ها بین دو ماشین مجزا می‌باشد. سوکت، برنامه‌نویسان را از پیچیدگی‌های فرآیند برقراری ارتباط بین دو ماشین مانند جزئیات کد کردن بسته‌ها، فرآیند ارسال داده‌ها در شبکه، ارسال مجدد بسته‌های خراب و … دور ساخته و برنامه‌نویسان، به راحتی قادر به توسعه‌ی برنامه‌ها‌ی تحت شبکه می‌باشند.

بطور کلی یک سوکت چهار عمل اصلی زیر را انجام می‌دهد:

۱-اتصال به ماشین راه دور

۲-ارسال داده ها

۳-دریافت داده ها

۴-بستن یا خاتمه اتصال

سوکت‌ها دارای انواع مختلفی هستند اما دو نوع از سوکت‌ها بیش‌ترین استفاده را دارند که بر اساس پروتکل‌های TCP و UDP دسته‌بندی شده‌اند. اگر بخواهیم به معرفی مختصر آن‌ها بپردازیم:

**سوکت‌های استریم(STREAM):**

اساس کار این سوکت‌ها بر پایه‌ی پروتکل TCP طراحی شده و با آن کار می‌کنند. در این سوکت‌ها باید قبل از جابه‌جایی اطلاعات، یک اتصال امن و قدرتمند ایجاد شده تا داده‌ها با نظم و دقت، ارسال و دریافت شوند. در ارتباطات با پروتکل‌های HTTP ،FTP و SMTP از این نوع سوکت‌ها استفاده می‌شود.

**سوکت‌های دیتاگرام(DATAGRAM):**

این نوع سوکت‌ها براساس پروتکل UDP کار می‌کند و هیچ اتصالی از قبل برای جابه‌جایی داده‌ها لازم ندارد. در این سوکت‌ها، انتقال صحیح و کامل داده‌ها اهمیت ندارد و هیچ وقت رسیدن داده به مقصد بررسی نمی‌شود. مهم‌ترین دلیل استفاده از این سوکت‌ها، سرعت بالای انتقال داده‌هاست و بیش‌تر در انتقال صوت و تصویر استفاده می‌شود.

**جواب سوال ۵:**

آدرس آی پی عمومی (Public IP address)، آدرسی است که به هر دستگاه متصل به اینترنت اختصاص می‌یابد و لذا می‌توان از اینترنت به آن دسترسی یافت. وقتی با کامپیوتر، گوشی یا هر وسیله دیگری به اینترنت متصل می‌شوید، آدرس آی پی منحصربه‌فردی به آن وسیله اختصاص می‌یابد که به آن آدرس آی پی عمومی (Public IP address) می‌گویند. در اینترنت، هیچ دو وسیله‌ای را نمی‌توان یافت که آدرس آی پی عمومی آن‌ها یکسان باشد.

آدرس‌های آی پی خصوصی (Private IP address) آدرس‌هایی هستند که برای استفاده در شبکه‌های خصوصی و غیرمرتبط با اینترنت رزرو شده‌اند. آدرس‌های آی پی خصوصی در اینترنت کاربرد ندارند بطور مثال برای ایجاد شبکه در یک شرکت یا سازمان استفاده میشود.

چون تعداد آدرس‌های آی پی IPv4 نسبت به جمعیت کاربران اینترنت بسیار محدود است و نمی‌توان به هر کاربر اینترنت یک آی پی عمومی منحصربه‌فرد اختصاص داد، با استفاده از ترجمه آدرس شبکه (NAT) به آدرس عمومی می‌توان این مشکل را حل کرد.

**آدرس‌ ip رزرو شده**

کارگروه مهندسی اینترنت (IETF) و نهاد متصدی انتساب اعداد در اینترنت (IANA) شماری از آدرس‌های آی پی را رزرو کرده‌اند. این آدرس‌ها برای کاربردهای خاصی رزرو شده‌اند. در ادامه برخی از آدرس‌های آی پی رزرو شده به‌اختصار توضیح داده می‌شوند.

**آدرس آی پی ۰.۰.۰.۰ چیست؟**

آدرس آی پی ۰.۰.۰.۰ یک آدرس غیرقابل مسیریابی است که معمولا نامعتبر یا ناشناخته بودن مقصد را نشان می‌دهد. البته این آدرس روی دستگاه‌های کلاینت و سرور کاربردها و معانی متفاوتی دارد.

آدرس آی پی ۰.۰.۰.۰ در کامپیوتر شخصی و کلاینت‌ نشان می‌دهد که دستگاه به شبکه TCP/IP متصل نیست و احتمالا آفلاین است. گاهی نیز اگر تخصیص آدرس آی پی دچار نقص شود، DHCP به‌طور خودکار این آدرس آی پی را به دستگاه اختصاص می‌دهد. وقتی آدرس آی پی کامپیوتر ۰.۰.۰.۰ باشد، نمی‌تواند با هیچ‌یک از تجهیزات متصل به شبکه مرتبط شود. اما کاربرد آدرس آی پی ۰.۰.۰.۰ در سرورهای شبکه متفاوت است. برخی تجهیزات به‌ویژه سرورهای شبکه بیش از یک کارت شبکه دارند و به هر کارت شبکه نیز آی پی متفاوتی اختصاص می‌یابد. اپلیکیشن‌های TCP/IP با استفاده از آدرس ۰.۰.۰.۰ ترافیک شبکه را روی همه آدرس‌های آی پی اختصاص یافته به کارت‌های شبکه آن سرور که به چند شبکه متصل است، مانیتور می‌کنند.

**آدرس آی پی ۱۲۷.۰.۰.۱ چیست؟**

آدرس آی پی ۱۲۷.۰.۰.۱ را هاست محلی (لوکال‌هاست) نیز می‌نامند. هر کامپیوتری برای اشاره به خود از آدرس آی پی ۱۲۷.۰.۰.۱ استفاده می‌کند. اما با این آدرس آی پی نمی‌توان به شبکه یا به اینترنت متصل شد. می‌توانید به کامپیوترتان یک آدرس آی پی خصوصی مثل ۱۹۲.۱۶۸.۱.۱۱۵ اختصاص دهید تا با روتر و دیگر گره‌های شبکه مرتبط شود. اما حتی در این‌صورت نیز هویت فردی کامپیوترتان با آدرس آی پی ۱۲۷.۰.۰.۱ تعریف می‌شود. به عبارت دیگر، آدرس آی پی ۱۲۷.۰.۰.۱ فقط توسط خود کامپیوتر و روی همان کامپیوتر کاربرد دارد. مثلا اگر روی کامپیوترتان یک وب‌سرور دارید و می‌خواهید صفحات وب را پیش از انتشار در اینترنت ابتدا به‌صورت محلی آزمایش کنید، می‌‌توانید صفحه‌های وب را روی آدرس آی پی ۱۲۷.۰.۰.۱ اجرا کنید تا فقط روی هاست محلی یعنی کامپیوترتان نمایش داده شود.

**آدرس آی پی ۸.۸.۸.۸ و آدرس‌ آی پی ۸.۸.۴.۴ چیست؟**

آدرس آی پی ۸.۸.۸.۸ و نیز ۸.۸.۴.۴ متعلق به سرورهای DNS عمومی گوگل است. گوگل این سرویس را برای افزایش سرعت وب‌گردی راه‌اندازی کرده است. این دو آدرس ضریب اطمینان‌پذیری زیادی دارند و می‌توان گفت همیشه آنلاین هستند. لذا کاربران برای اطمینان از این‌که آیا کامپیوترشان آنلاین است یا نه می‌توانند آن را با آدرس آی پی ۸.۸.۸.۸ یا ۸.۸.۴.۴ پینگ کنند. اگر پس از پینگ با پیغام خطا مواجه شدند، به احتمال زیاد ایراد از سمت کاربر است و او باید تنظیمات کامپیوتر یا شبکه را بازنگری کند.

**رنج ip چیست؟**

دامنه آدرس آی پی، محدوده آدرس‌های آی پی مجاز و معتبر را تعیین می‌کند. در فرمت IPv4 دامنه آدرس‌های آی پی از ۰۰۰.۰۰۰.۰۰۰.۰۰۰ شروع می‌شود و با ۲۵۵.۲۵۵.۲۵۵.۲۵۵ خاتمه می‌یابد. پس برای مثال، آدرس آی پی ۱۰.۰.۰.۰ معتبر اما آدرس آی پی پی ۳۰۰.۰.۰.۰ نامعتبر است، چون خارج از دامنه مجاز آدرس‌های آی پی است. علاوه بر این، بسته به کاربردهای مختلف می‌توان دامنه‌های دیگری نیز تعریف کرد. مثلا پیش‌تر، در توضیح آدرس‌های آی پی عمومی و خصوصی گفته شد که هر یک از آن‌ها دامنه یا محدوده مشخصی دارند. کلاس‌های آدرس آی پی از دیگر مواردی هستند که هر یک دامنه مختص خود را دارند. اما کلاس‌های آدرس آی پی چه هستند و چه کاربردی دارند؟

**کلاس‌بندی آدرس‌های آی پی**

آدرس‌های IPv4 در پنج رده یا کلاس تقسیم‌بندی و با حروف A تا E مشخص شده‌اند. در این‌جا تنها سه رده نخست آن‌ها موضوع بحث است. هر یک از این رده‌ها یا کلاس‌های آدرس آی پی حداکثر تعداد شبکه‌ها و حداکثر هاست‌های ممکن هر شبکه را که مبتنی بر آن کلاس ایجاد شده‌اند، مشخص می‌کند. این سه کلاس عبارتند از:

کلاس C: کوچک‌ترین کلاس آدرس‌های آی پی است که به‌ازای هر شبکه، تنها ۲۵۶ آدرس آی پی ارائه می‌دهد.

دامنه آدرس‌: از ۱۹۲.۰.۰.۰ تا ۲۲۳.۲۵۵.۲۵۵.۲۵۵

تعداد کل آدرس‌های آی پی: ۵۳۶.۸۷۰.۹۱۲

کلاس B: به‌ازای هر شبکه تا ۶۵.۵۳۶ آدرس آی پی ارائه می‌دهد.

دامنه آدرس: از ۱۲۸.۰.۰.۰ تا ۱۹۱.۲۵۵.۲۵۵.۲۵۵

تعداد کل آدرس‌های آی پی: ۱.۰۷۳.۷۴۱.۸۲۴

کلاس A: بزر‌گترین کلاس و به‌ازای هر شبکه ۱۶.۷۷۷.۲۱۶ آدرس آی پی ارائه می‌دهد.

دامنه آدرس: از ۰.۰.۰.۰ تا ۱۲۷.۲۵۵.۲۵۵.۲۵۵

تعداد کل آدرس‌های آی پی: ۲.۱۴۷.۴۸۳.۶۴۸

کلاس D برای کاربردهای مولتی‌کست و کلاس E برای کاربردهای آینده رزرو شده‌اند. هر کدام از این دو کلاس در کل ۲۶۸.۴۳۵.۴۵۶ آدرس آی پی دارند اما ظرفیت آدرس آی پی‌ آن‌ها به‌ازای هر شبکه، تعریف نشده است

**جواب سوال ۷:**

سرور ابری یک نوع server می باشد که در رایانش ابری ایجاد میشود و بر روی بستر اینترنت برای بسیاری از کاربران ارائه میشود. در حقیقت می توان گفت که سرورهای ابری همان سرورهای اختصاصی مجازی محسوب میشوند که بر روی زیر ساخت ابری قرار می گیرند و در محیط رایانش ابری ارائه میشوند. لازم به گفتن است که در حالت معمول سرورهای مجازی فقط بر روی یک server فیزیکی ارائه میشوند و با سخت افزار مربوط به آن راه اندازی می شود.

سرور ابری مانند سرورهای مجازی عمل می‌کنند. در این مدل سرور کاربر با داشتن پنل کاربری و بدون هیچ سخت افزار فیزیکی، می‌تواند سرور خود را از راه دور مدیریت کند. تنها تفاوت سرور ابری با سرورهای مجازی (VPS) این است که اطلاعات آن از راه دور کنترل و پردازش می‌شود.

**کاربرد سرور ابری**

امروزه حجم گسترده‌ای از خدمات فناوری‌های IT به سمت زیرساخت‌های ابری تمایل پیدا کرده و تمایل این سرویس‌ها به استفاده از خدمات ابری روز به روز بیشتر می‌شود. این تمایلات به شکلی بوده است که اغلب کارشناسان این عرصه اعتقاد دارند که آینده فناوری‌های IT در رایانش ابری تعریف می‌شود. در حقیقت رایانش ابری این امکان را فراهم کرده تا کاربران بدون نیاز به مدیریت و دخالت مستقیم، به منابع رایانشی مانند شبکه و سرورها دسترسی آسانتری داشته باشند. مفهوم ابر از صنعت تلفن بر گرفته شده است. حال اینکه رایانش ابری مفهوم ابر را به شکلی گسترش داده که سرورها را نیز علاوه بر زیرساخت‌های شبکه در بر گرفته است.

سرویس‌های زیرساخت ابری، یک زیرساخت مجازی را به شکل یک سرویس به کاربران خود ارائه می‌دهند. ازاین‌ جهت کاربران به‌جای خرید سخت‌افزار، نرم‌افزار، مرکز داده‌ها و تجهیزات شبکه، تمام این زیرساخت‌ها را در قالب یک سرویس خریداری می‌کنند.

در اصل سرویس‌دهندگان خدمات زیرساخت ابری، این امکان را به شرکت‌ها می‌دهند تا در زمان تامین و نگهداری زیرساخت‌های فناوری اطلاعات صرفه جویی کنند و بر روی مسائل مهم‌تری مانند کسب‌وکار خود متمرکز شوند. علاوه بر این، با استفاده از این خدمات ، نیازی به سرمایه‌گذاری اولیه جهت تأمین موارد فوق نبوده و هزینه‌های عملیاتی به میزان قابل‌توجهی کاهش خواهد یافت.

**جواب سوال ۸:**

معمولا سه مدل از سرویس های ابری برای مقایسه با هم وجود دارند:

\*نرم افزار به عنوان سرویس یا SaaS

\*پلتفرم به عنوان سرویس یا PaaS

\*زیرساخت به عنوان سرویس یا IaaS

SaaS: نرم افزار به عنوان سرویس

دقت کرده‌اید در این چند ساله اخیر چقدر راحت می‌توانیم از بعضی سرویس‌های سطح وب به جای نرم‌افزارها استفاده کنیم؟ مثلا قبلا مجبور بودیم برای نوشتن یک محتوا یا یک تکه متن، نرم افزار آفیس را روی ویندوز نصب کنیم، بعد به سراغ word رفته و در آن بنویسیم. اما امروز به راحتی می‌توانیم از google docs کمک بگیریم، آن هم بدون نیاز به نصب هیچ نرم‌افزاری! یا اصلا همین gmail که بدون نیاز به نصب و دانلود هیچ برنامه‌ای روی تلفن همراه یا رایانه به ما خدمات می‌دهد.

در واقع همه این‌ها را مدیون خدمات رایانش ابری و البته SaaS هستیم!

SaaS چیست؟

خود واژه SAAS از سرواژه کلمات Software As A Service و به معنای نرم‌افزار به عنوان سرویس برداشته شده. در واقع saas یکی از بهترین راهکارهایی است که در اختیار شرکت‌های نرم‌افزاری قرار گرفته تا بتوانند به کمک آن یک برنامه را در سطح وب راه اندازی کنند.

آشنایی با SaaS

در سرویس saas، شرکت شما می‌تواند نرم‌افزار مورد نظر را به جای ارائه در فایل‌های دانلودی یا فروش روی سی دی، فلاش مموری یا هر نوع فضای ذخیره سازی دیگر، روی یک سیستم ابری بارگذاری کند.

در ادامه دیگر کاربر نیاز به نصب نرم‌افزار روی سیستمش ندارد. چرا؟ چون پشت صحنه نرم‌افزار شما دیگر نه روی سیستم کاربر، بلکه روی یک سیستم ابری در یک شرکت ثالث اجرا شده و کاربر صرفا نیاز است به کمک اینترنت به برنامه دسترسی پیدا کند.

البته ممکن است نیاز شما نه اشتراک برنامه با کاربر، بلکه صرفا راه‌اندازی یک نرم افزار درون شرکتی باشد که روی سیستم اعضای مجموعه فعالیت کند. در این حالت هم SaaS یک راه‌حل عالی است.

بگذارید یک مثال بزنم تا هم خیال خودم و هم شما را راحت کنم:

الان شما برای استفاده از ابزارهایی مثل Google Sheet (که یک برنامه مشابه اکسل مایکروسافت است) چه نرم‌افزاری روی سیستم خود نصب می‌کنید؟ خب درواقع هیچ!

شما همان کارهایی با اکسل می‌کردید را روی گوگل شیت انجام می‌دهید؛ بدون آنکه نیاز باشد برنامه خاصی نصب کنید.

خب در واقع اینجا بارِ کارهای پشت صحنه نرم افزار از روی دوش سیستم شما برداشته و به کمک سرویس saas، به عهده یک رایانه ابری گذاشته شده. این یعنی به جای دانلود و نصب گوگل شیت روی رایانه شما، این برنامه الان روی سیستم‌های رایانش ابری یک شرکت دیگر سوار شده و به شما خدمات می‌دهند.

**مزایای استفاده از SaaS چیست؟**

۱) کاهش مشکلات کاربر

خب اولین مزیت استفاده از SaaS، کاهش درگیری‌های ذهنی کاربر نهایی با برنامه است. در واقع کاربری که به کمک یک مرورگر به نرم‌افزارش دسترسی دارد، نیاز به نگرانی بابت به‌روزرسانی، مشکلات نرم‌افزاری و… نخواهد داشت.

۲) کاهش مشکلات توسعه‌دهنده

مزیت دوم ولی در مورد خود شرکت سازنده نرم‌افزار است. شرکتی که به جای سرویس‌های سنتی و عادی، از سرویس SaaS برای اشتراک گذاری نرم‎افزارش استفاده می‌کند، بسیاری از مشکلات سخت افزاری را از روی دوش خود برداشته و آن را به شرکت ارائه دهنده سرویس SaaS منتقل می‌کند.

۳) کاهش هزینه

و اما مورد سوم، اینکه شرکت سازنده نرم افزار قرار نیست برای اشتراک‌گذاری نرم‌افزارش با کاربران نهایی (یا اعضای شرکت خودش) هزینه زیادی برای خرید سخت افزارهای شبکه، زیرساخت‌ها و… هزینه کند. بلکه به کمک SAAS، شرکت به اندازه میزان مصرفش، هزینه بابت خدمات پرداخت می‌کند و قرار نیست دیگر برای خرید یک سرور و تعداد زیادی تجهیزات پول بدهد.

با همه این مزیت‌ها، نکته جالب اینجاست که کارکرد برنامه مورد نظر که روی SaaS سوار شده، نسبت به برنامه‌ای که روی سیستم کاربر سوار است، با هیچ کاهش کیفیت و سرعتی روبرو نیست.

۴) کاهش مشکلات خرید محصول

چهارمین مزیتی که در مورد saas می‌خواهم در این مقاله به آن اشاره کنم، هم مربوط به مشتری و هم سازنده است. احتمالا می‌دانید ما اغلب باید برای استفاده از نرم‌افزارهای غیر رایگان، اقدام به خرید فایل آن یا لایسنسش کنیم. اما به کمک SaaS، دیگر نیازی به چنین کاری نبوده و شرکت سازنده می‌تواند آن را در اختیار کاربرانی قرار بدهد که قبلا اشتراک سیستم را به شکل آنلاین خریداری کرده و الان واردش می‌شوند.

به این ترتیب یک مرحله سخت که شامل ساخت لایسنس و… بود از دوش سازنده برداشته شده و همینطور به راحتی بیشتر کاربران کمک می‌کند.

علاوه بر همه مزیت‌های گفته شده، SaaS به‌طور خلاصه شامل این مزیت‌ها هم هست:

قابلیت دسترسی بالا صرفا به کمک یک مرورگر و خط اینترنت،

قابلیت خصوصی‌سازی بالای نرم افزار نسبت به نیاز هر مشتری،

راه‌اندازی سریع و بدون نیاز به دانش تخصصی و فنی،

کاهش هزینه‌ها برای سازنده و کاربر،

و… .

**PaaS چیست؟**

سرویس PAAS از سرواژه‌های عبارت Platform As A Service برداشته شده و به معنی پلتفرم به عنوان سرویس است. در واقع PaaS بعد از IaaS و قبل از SaaS، لایه دوم سرویس پلتفرم ابری است که در اختیار توسعه‌دهندگان قرار می‌گیرد.

PaaS یک چارچوب (framework) برای توسعه دهندگان فراهم می کند که از می توانند از آن برای ایجاد برنامه‌های سفارشی استفاده کنند.

PaaS توسعه‌دهنده‌ها را از خرید یا اجاره مستقیم سیستم‌های زیرساختی و سخت‌افزاری مثل زیرساخت شبکه، سرور، سیستم عامل مورد نیاز و… بی‌نیاز می‌کند.

در این مدل تمامی امور مربوط به سرور، ذخیره سازی و شبکه توسط یک شرکت یا ارائه‌دهنده به عنوان واسط، مدیریت می شود؛ در این صورت توسعه‌دهندگان می‌توانند تنها بر روی توسعه و مدیریت برنامه خود تمرکز کنند.

مثلا در حال حاضر، برنامه‌نویسان می‌توانند با اتصال به سرویس Microsoft Azure که یکی از سرویس‌های زیرمجموعه شرکت مایکروسافت است، بدون نیاز به انواع نرم‌افزار و دیگر زیرساخت‌های نگهداری اطلاعات، شروع به برنامه‌نویسی کنند.

**قابلیت‌های PaaS چیست؟**

به شکل خلاصه به ویژگی‌ها و قابلیت‌های این سرویس اشاره‌ای می‌کنم تا بیشتر با آن آشنا شوید:

خدمات میزبانی وب

خدمات نرم افزار سرور

ذخیره سازی اطلاعات و دیتاها

ارائه سیستم عامل‌های مورد نیاز

ابزارهای طراحی و توسعه نرم افزار

محیط برنامه‌نویسی نرم افزار طرف سرور

دسترسی به شبکه

خدمات پشتیبانی

**مزایای PaaS چیست؟**

حالا بعد از این‌همه تعریف، اصلا چرا برنامه نویسان و توسعه دهندگان نرم افزار، باید اقدام به استفاده از سرویس paas کنند؟ در ادامه و با توضیح مزایای این سرویس، خودتان به خوبی دلیل این موضوع را متوجه می‌شوید:

۱) قابلیت خصوصی سازی

یکی از قابلیت‌های جذاب paas که باعث جذب توسعه‌دهندگان نرم افزار به سمت آن شده، قابلیت خصوصی شدن آن است. این یعنی شما به عنوان یک برنامه‌نویس می‌توانید قابلیت‌هایی که سرویس paas در اختیارتان می‌گذارد را نسبت به نیازهایتان کم و زیاد کرده و از آنها نهایت استفاده را داشته باشید.

۲) به‌روزرسانی مداوم

سرویس دهندگان PaaS، همیشه پلتفرم و جزئیات آن را به‌روز نگه می‌دارند. این یعنی شما به عنوان یک توسعه‌دهنده نیاز به نگرانی بابت به‌روزرسانی مداوم سیستم‌ها، زیرساخت‌ها، نرم افزار و سخت افزار نداشته و می‌توانید بدون درگیر کردن ذهن‌تان به این موضوعات فنی، صرفا روی تولید یک نرم افزار و محصول نهایی عالی تمرکز کنید.

۳) هزینه‌های به صرفه

یک لحظه شرایطی را برای خودتان متصور شوید که مجبور باشید بدون وجود یک پلتفرم PaaS، اقدام به کدنویسی و تست دائم نرم افزار کنید. احتمالا در چنین حالتی نیاز دارید هزینه‌های زیادی را برای اجاره زیرساخت‌هایی مثل سخت افزارها و سرورها بدهید. همینطور نیاز به بررسی مداوم این سخت افزارها، بروزرسانی‌شان، رفع مشکلات احتمالی و… خواهید داشت. مطمئنا همه اینها، هم از نظر زمانی و هم از نظر مالی، هزینه‌های زیادی را به شما تحمیل می‌کنند.

در ادامه باید این را بدانید که در پلتفرم paas، قرار به صرف هزینه زمانی برای نگهداری از شبکه نیست. همینطور در این پلتفرم‌ قرار نیست شما مقدار زیادی هزینه پرداخت کرده و فضاهای ذخیره‌سازی اطلاعات و سخت افزارهای سنگین بخرید. شما صرفاً به اندازه آنچه از آن استفاده کرده‌اید می‌پردازید! بله! همه کاربران این سرویس‌ها، فقط به اندازه مصرف‌شان پول می‌دهند و نه چیزی بیشتر.

۴) کاهش نیاز به نیروی کار

اگر تا الان وارد دنیای توسعه نرم افزار به شکل جدی و در مقیاس‌های بزرگ شده باشید، می‌دانید که نگهداری از زیرساخت‌ها نیاز به یک یا چند نفر نیرو متخصص و دائم دارد. در مقابل و در زمان استفاده از paas، با توجه به اینکه فعالیت نگهداری و حل مشکلات زیرساخت‌ها به عهده یک شرکت سوم است، نسبت به استخدام متخصصان این حیطه بی‌نیاز می‌شوید.

۵) ارتباط آسان

تا همین چند وقت پیش هنوز خیلی از تیم‌ها اصرار داشتند باید تمام فعالیت‌های برنامه نویسی و توسعه نرم افزار از درون شرکت و به شکل حضوری انجام شود. اما بعد از بحران کرونا و بعد از اینکه شرکت‌ها مجبور به ادامه فعالیت به شکل دورکاری شدند، اهمیت فعالیت‌های از راه دور بیشتر از قبل پررنگ شد.

در این میان مجموعه‌هایی که از PaaS استفاده می‌کردند، با کمترین مشکلات روبرو شدند. چرا؟ چون در واقع برای ادامه فعالیت و توسعه محصول از طریق این پلتفرم‌ها، صرفا یک خط اینترنت و رایانه مورد نیاز بوده و دیگر نیاز به زیر ساخت‌های تخصصی حذف شده است. این موضوع، ارتباط از راه دور و ادامه همکاری را به شدت راحت کرده.

با سیستم‌های PaaS، شما می‌توانید از هر مکان و از هر سیستمی، به اطلاعات‌تان دسترسی داشته باشید.

۷) امنیت و ایمنی

اول از همه و با توجه به اینکه paas یک سرویس ابری است، امنیت بالاتری نسبت به سرویس‌های عادی و مجازی دارد. نکته بعدی اینکه این سرویس‌ها به دلیل داشتن فضای ذخیره سازی زیاد، دائما از داده‌های شما پشتیبان ذخیره کرده و امنیت آنها را تضمین می‌کنند.

با توجه به ابری بودن این سیستم، ایمنی بالاتری هم متوجه آن است. این یعنی حتی اگر اطلاعات شما در یکی از سرورها به دلایلی به مشکل بخورد، دیتاها به کمک دیگر سرورها نگهداری شده و در دسترس تیم‌تان یا خودتان قرار می‌گیرد. به این صورت دیگر نگرانی‌های مربوط به بحث امنیت و ایمنی به کل از بین می‌رود.

**IaaS چیست؟**

اولین و پایه‌ای‌ترین لایه در خدمات رایانش ابری، IaaS است که قبل از لایه‌های SaaS و PaaS قرار گرفته. این خدمات ابری امروز یکی از بهترین راهکارها برای حل مشکلات شرکت‌های بزرگِ نیازمند نگهداری اطلاعات هستند.

زیرساخت به عنوان سرویس یا همان Infrastructure as a Service، به نحوی مهم‌ترین خدمات در دنیای رایانش ابری است. در این سرویس، شرکت و مجموعه می‌توانند به جای خرید یا اجاره یک سرور برای نگهداری از دیتاها و اطلاعات کسب و کار، به اندازه نیاز، فضای ابری در اختیار گرفته و از آن استفاده کنند.

**کاربردهای IaaS چیست؟**

IaaS خدمات خود را به صورت کامل برای دسترسی و نظارت بر چیزهایی مانند محاسبات، شبکه سازی، ذخیره‌سازی و سایر سرویس‌ها فراهم می سازد و به کسب و کارها اجازه می‌دهد تا به‌جای خریداری منابع سخت‌افزاری ثابت، بر اساس نیاز و تقاضای خود از منابع استفاده کنند. در ادامه می‌خواهم به رایج‌ترین کاربردهای IaaS که شرکت‌های مختلف به خاطرشان اقدام به خرید این خدمات می‌کنند، اشاره کرده و آنها را برای‌تان توضیح بدهم.

۱) تحلیل داده

امروزه بخش اعظمی از فعالیت کسب و کارهای اینترنتی و حتی بعضی از کسب و کارهای فیزیکی، به تحلیل داده‌ها و اطلاعات به دست آمده از مشتری، محصول و خدمات اختصاص پیدا کرده. حالا در این میان زمانی که حجم داده بزرگ باشد، نه تنها برای ذخیره آنها بلکه برای تحلیل‌شان نیاز به ساختارهای قدرتمند حس می‌شود. IaaS با توجه به ساختار ابری‌ و زیرساخت انعطاف‌پذیر و قوی‌اش، می‌تواند در تحلیل داده‌ها کمک زیادی به شرکت‌های بزرگ بکند.

۲) بازیابی اطلاعات

با توجه به امنیت و یکپارچه بودن سیستم رایانش ابری، شرکت‌هایی که از آن استفاده می‌کنند نگرانی کمتری نسبت به ایمنی اطلاعات خودشان داشته و می‌توانند برای بازیابی اطلاعات، روی آن حساب کنند.

۳) آزمایش و توسعه

اغلب محصولات اینترنتی و نرم‌افزاری قدرتمند، مثل اپلیکیشن‌های همه‌گیر یا نرم افزارهای حسابداری بزرگ، برای ورود به بازار نیاز به گذراندن آزمایش‌های مختلفی دارند. علاوه بر این، این اپلیکیشن‌ها و نرم‌افزارها برای توسعه و افزایش کارایی هم نیازمند فضای ذخیره‌سازی و البته ساختار قدرتمند نرم‌افزاری و سخت‌افزاری قوی هستند. اینجاست که باز هم پای IaaS وسط آمده و در این پروسه، به مجموعه‌ها کمک زیادی می‌کند.

**مزایای IaaS چیست؟**

۱) امنیت و ایمنی

سرویس‌های ابری نسبت به سرویس‌های فیزیکی و ثابت در محل، امنیت و ایمنی بسیار بالاتری دارند. مثلا اگر برحسب اتفاق یا اصلا بلایای طبیعی، اطلاعات مجموعه شما در یکی از سرورهای رایانش ابری‌تان از بین برود، بقیه سرورهای موجود در زیرساخت ابری، اطلاعات را در اختیارتان قرار می‌دهند! بنابراین نه تنها آنها را از دست نمی‌دهید بلکه به راحتی می‌توانید به ادامه فعالیت‌های شرکت‌تان مشغول شوید.

۲) Up-Time بالا

احتمالا می‌دانید که هیچ شرکت هاستینگی در دنیا ادعای ۱۰۰ درصد بودن زمان up-time را ندارد. اما اصولا با توجه به اینکه در خدمات رایانش ابری، چندین سرور مختلف از اطلاعات شما نگهداری می‌کنند، اطلاعات موجود در سرورهای ابری IaaS شما، همیشه در اختیارتان قرار گرفته و این سرویس، آپ تایم بسیار بسیار بالایی دارد.

۳) انعطاف و مقیاس‌پذیری

در اغلب مواقع زمانی که قصد خرید سرورهای سنتی و عادی برای نگهداری از اطلاعات مجموعه‌تان را داشته باشید، خیلی دست‌تان برای انتخاب جزئیات باز نیست. یا حداقل اگر نیاز به گسترش کار داشته باشید، شاید کمی برای افزایش توان و زیرساخت سرورها به مشکل بخورید. اما در مورد IAAS دیگر چنین مشکلاتی وجود ندارد و می‌توانید به راحتی و با توجه به انعطاف‌پذیر بودن این سرویس، ترکیب‌های مورد نظرتان را از ارائه دهنده درخواست کنید. (مثلا افزایش تعداد cpu‌ها)

۴) کاهش هزینه

و در نهایت اینکه شرکت‌ها در زمان استفاده از IaaS و البته دیگر خدمات رایانش ابری، نیاز ندارند بیش از مصرف‌شان پول بپردازند! این یعنی شرکت در هر بازه زمانی صرفا به اندازه آن مقداری پول پرداخت می‌کند که طی آن، از سرویس‌ها استفاده کرده باشد.

**نحوه ارائه خدمات در IaaS چگونه است؟**

IaaS، زیرساخت های محاسباتی ابری شامل سرورها، شبکه، سیستم های عملیاتی و ذخیره سازها را از طریق تکنولوژی مجازی‌سازی (Virtualization) در اختیار سازمان‌ها قرار می دهد.

این سرورهای ابری معمولا از طریق یک داشبورد یا API در اختیار مشتری قرار می‌گیرند و بعد از آن مشتری‌های IaaS، اختیار کاملی بر روی تمامی زیرساخت ارائه شده دارند.

IaaS قابلیت‌ها و فناوری‌های مشابهی مانند یک دیتاسنتر سنتی ارائه می کند، با این تفاوت که نیازی به مدیریت و نگهداری فیزیکی از سرورها نیست. کاربران IaaS می‌توانند بر روی سرورها و هاردهای خود دسترسی مستقیم داشته باشند، اما این دسترسی از طریق یک دیتاسنتر مجازی که بر روی ابر قرار دارد، انجام می‌شود.

برخلاف SaaS و PaaS، در این مدل، کاربران مسئولیت مدیریت و بررسی مواردی همچون برنامه‌ها، سیستم عامل‌ها، زمان اجرا، میان‌افزارها و داده‌ها را بر عهده دارند. با این حال، ارائه دهندگان IaaS، سرورها، هارد دیسک‌ها، شبکه، مجازی‌سازی و ذخیره‌سازی را مدیریت می‌کنند. بعضی ارائه‌دهندگان نیز خدمات بیشتری خارج از لایه مجاز‌ساز، مانند پایگاه‌های داده و صف‌های پیام را نیز ارائه می‌کنند.