**Cloud services**

1. pengantar

Wikipedia komputasi awan seperti: pengiriman komputasi sebagai layanan daripada produk, tempat sumber daya bersama, perangkat lunak, dan informasi disediakan untuk komputer dan perangkat lain sebagai layanan meteran melalui jaringan (biasanya Internet). Gartner Group juga mencantumkan 'layanan berbasis' sebagai salah satu atribut kunci komputasi awan. Memang, kita bisa mengatakan bahwa salah satu ciri khas itu membuat cloud computing berbeda dari pengembangan perangkat lunak tradisional dan deploy-ment adalah fokus pada pemberian layanan. Lembaga Nasional Standar dan Teknologi (NIST) juga layanan situs sebagai komponen penting dari komputasi awan dan de fi nes tiga model layanan. Dalam bab ini kita akan mengeksplorasi berbagai cara di mana Anda dapat membangun dan menggunakan layanan di cloud. Kami akan memeriksa apa yang dikenal sebagai model SPI di mana surat-surat itu singkatan dari 'layanan', 'platform' dan 'infrastruktur' dan lihat caranya model ini berlaku untuk sistem yang saat ini tersedia di cloud. Sebagai yang muncul dan area yang berkembang pesat, sejumlah istilah utama yang digunakan dalam komputasi awan tidak belum memiliki sebutan yang disepakati. Namun, NIST menyediakan yang paling banyak digunakan definisi, khususnya yang berkaitan dengan model SPI, sehingga kami memasukkannya ke dalam bagian yang relevan di bawah ini. Kami juga menyertakan sejumlah contoh teknologi untuk masing-masing jenis layanan. Namun, daftar itu jauh dari lengkap, dan tentu saja, baru layanan dan penyedia layanan terus bermunculan dan mentransformasikannya dengan cepat mengubah 'cloudscape'. Di akhir bab ini, Anda akan mengembangkan virtual mesin yang Anda buat sebelumnya dan kemudian menggunakannya untuk membuat dan menyebarkan aplikasi Anda sendiri kation di cloud. Kami harus mencatat bahwa meskipun kami menyentuh pada penyimpanan data masalah dalam bab ini, topik ini dibahas secara rinci di Chap. Ada sejumlah teknologi yang telah diidentifikasi sebagai prekursor awan komputasi seperti grid, utilitas komputasi, komputasi otonom dan virtualisasi. Mungkin yang paling relevan dengan layanan cloud adalah pengembangan layanan web dan arsitektur berorientasi layanan (SOA). Kami memulai bab ini dengan memeriksa teknologi ini yang masih sangat relevan dengan evolusi layanan cloud yang sedang berlangsung.

1. Layanan Web

Layanan web mengacu pada perangkat lunak yang menyediakan cara standar integrase fasilitas yang ditawarkan oleh aplikasi web dan mendukung komunikasi antarorganisasi sations tanpa memerlukan pengetahuan rinci tentang bagaimana layanan diimplementasikan atau bahkan bahasa atau platform mana mereka diimplementasikan. Layanan harus bekerja tugas yang mudah digambarkan, terfokus, dan terisolasi sehingga tidak bergantung pada keadaan khusus layanan lainnya. Tidak seperti arsitektur client-server tradisional, web layanan menyediakan antarmuka terprogram dan lebih ramah-mesin daripada graphi antarmuka yang ramah-pengguna disajikan dalam browser atau aplikasi desktop. XML adalah format komunikasi khas untuk layanan web yang menyediakan keuntungan signi fi dari independensi platform. Uraian Layanan Web Bahasa (WSDL) dapat digunakan untuk menggambarkan layanan yang tersedia, ke struktur permintaan dan tanggapan dan untuk memberikan informasi mengenai jenis protocol diperlukan untuk memanggil layanan. Implementasi untuk bekerja dengan WSDL berdasarkan C #, Java dan bahasa lain tersedia. Banyak layanan web benar-benar gratis, sedangkan yang lain mungkin mengenakan biaya untuk penggunaannya. Jika Anda membuat layanan web, itu jelas berguna jika Anda dapat mempublikasikan fakta ini seperti itu pengguna potensial dapat menemukan, mengevaluasi, dan menggunakan layanan web yang Anda tawarkan. Direktori layanan web yang paling banyak digunakan tersedia melalui UDDI (Universal Keterangan, Penemuan dan Integrasi) spesifikasi yang menentukan cara untuk mempublikasikan dan temukan informasi tentang layanan web. SOAP (Protokol Akses Objek Sederhana) adalah format XML standar yang biasa digunakan untuk menandai informasi dan transportasi data. Koleksi standar (dikembangkan oleh World Wide Web Consortium (W3C)) yang digunakan untuk layanan web secara kolektif dikenal sebagai WS \* stack. Mungkin lebih mudah untuk memahami layanan web dengan melihat beberapa contoh. Jika Anda pergi ke halaman beranda WebserviceX.NET (<http://www.webservicex.net>), ratusan layanan web terdaftar. Anda dapat mencari berdasarkan kata kunci atau browser oleh popukategori larity atau subjek. Anda harus dapat melihat contoh yang biasa digunakan seperti itu sebagai generator barcode, konverter mata uang dan pemasok data cuaca. Jika Anda tertarik, Anda dapat menggunakan situs ini untuk mendapatkan deskripsi layanan WSDL bersama penataan SOAP yang diperlukan dan informasi berguna lainnya mengenai layanan.

1. Arsitektur Berorientasi Layanan

Tujuan arsitektur berorientasi layanan adalah menyatukan berbagai potongan fungsionalitas untuk membentuk aplikasi baru yang sebagian besar dibangun dari perangkat lunak yang ada jasa pergudangan. SOA menyediakan satu set metodologi untuk merancang perangkat lunak sebagai satu set layanan interoperable longgar digabungkan cocok untuk digunakan dalam komputasi terdistribusi lingkungan Hidup. Dalam cara yang agak mirip dengan pendekatan rekayasa perangkat lunak lainnya seperti orientasi objek, arsitektur berorientasi layanan memungkinkan kita untuk membuat komponen atau blok bangunan yang masuk akal bagi pengguna dan perancang manusia dan bias digunakan kembali untuk tujuan yang berbeda. Kerja internal komponen yang kompleks adalah tersembunyi, dan antarmuka yang sederhana dengan perilaku yang terdefinisi dengan baik disajikan. Pelayanan adalah modul mandiri yang menyediakan fungsionalitas bisnis standar dan independen dari negara atau konteks layanan lainnya. Layanan dijelaskan dalam bahasa definisi standar dan memiliki antarmuka yang dipublikasikan. The W3C mengacu pada SOA hanya sebagai: Seperangkat komponen yang dapat dipanggil, dan deskripsi antarmukanya dapat dipublikasikan dan ditemukan. SOA memungkinkan komunikasi antara penyedia dan konsumen dan memberikan sebuah mekanisme untuk mengelola layanan yang tersedia. Komunikasi harus dalam bentuk layanan atau fungsi perangkat lunak yang digabungkan secara longgar yang dapat digunakan kembali untuk berbeda tujuan. Singkatnya kita dapat mengatakan bahwa aplikasi enterprise yang mengikuti SOA Paradigma adalah kumpulan layanan yang bersama-sama melakukan logika bisnis yang rumit.

1. Interoperabilitas

Interoperabilitas adalah kunci persyaratan SOA. Komponen SOA harus independen bahasa pemrograman dan platform dan harus tidak terkait satu sama lain. SOA bertujuan untuk memungkinkan pengguna menggabungkan atau menggabungkan layanan kolaborasi dari angka sumber untuk membangun aplikasi baru yang berpotensi rumit. SOA memungkinkan organisasi untuk memaparkan dan mengakses layanan aplikasi dan untuk menggunakan informasi yang terikat layanan tersebut menggunakan antarmuka yang terdefinisi dengan baik.

1. Composability

Aplikasi sering dibangun dari kumpulan komponen: fitur yang disebut composability. Sistem composable menggunakan komponen untuk merakit layanan itu dapat disesuaikan untuk tujuan tertentu menggunakan bagian standar. Salah satu yang menarik Keuntungan dari pendekatan SOA adalah cara layanan dapat dikomposisikan kapan membuat aplikasi. Menggabungkan dan interleaving layanan adalah jantung layanan - komputasi berorientasi. SOA menyiratkan bahwa layanan dapat berpartisipasi sama dengan sederhana atau komposisi yang rumit. Composability telah dibawa ke level baru dengan cloud komputasi; misalnya, sumber daya komputasi dasar seperti penyimpanan dan kekuatan pemrosesan disediakan sebagai layanan yang dapat dikomposisikan untuk membuat lingkungan aplikasi yang dibutuhkan.

1. Transfer Negara Perwakilan (REST)

Perwakilan transfer negara (REST) ​​adalah gaya arsitektur perangkat lunak awalnya diperkenalkan oleh Roy Fielding yang merupakan salah satu penulis utama HTTP spesifikasi 1.0 dan 1.1. Seperti halnya HTTP, arsitektur REST memvisikan klien dan server dalam percakapan atau siklus permintaan / tanggapan. Permintaan dipusatkan pada rep- kebencian dari sumber daya dan menggunakan metode HTTP yang ada. Hal ini berbeda dengan sistem lain, seperti SOAP di mana pengguna dapat mendefinisikan metode dan sumber daya mereka sendiri identi fi ers. Dalam REST URI yang diberikan digunakan untuk mengakses status representasional dari sumber daya dan juga untuk memodifikasi sumber daya. Misalnya, URL di web dapat digunakan untuk memberikan informasi kepada pengguna tentang sumber daya atau untuk memodifikasi sumber daya:

• GET digunakan untuk mentransfer status representasi saat ini dari sumber daya dari server ke klien.

• PUT digunakan untuk mentransfer status representatif dari sumber daya dariklien ke server.

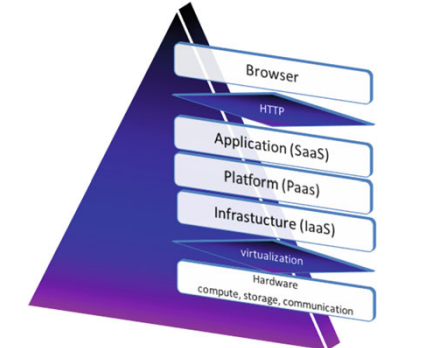
• POST digunakan untuk mentransfer status representasi baru dari sumber daya dari klien ke server.

• DELETE digunakan untuk mentransfer informasi yang diperlukan untuk mengubah sumber daya menjadi menghapus status representasional.

REST menggunakan HTTP untuk semua operasi CRUD (Create / Read / Update / Delete). REST adalah alternatif ringan yang semakin populer untuk mekanisme seperti RPC (remote proce- panggilan dure) dan SOAP. Seperti SOAP dan RPC, layanan ‘RESTful’ adalah platform dan lan- gauge independen tetapi dianggap lebih sederhana dan lebih mudah digunakan. Sementara layanan REST mungkin menggunakan XML dalam tanggapan mereka (sebagai salah satu cara untuk mengatur data terstruktur), REST permintaan jarang menggunakan XML; memang kita bisa menganggap halaman web itu sendiri sebagai layanan. SEBUAH contoh yang baik diberikan oleh situs yang menawarkan layanan ISBN yang mengembalikan detail buku saat menerima nomor ISBN. Isbndb.com menawarkan layanan semacam itu yang membutuhkan Anda cukup menggunakan HTTP GET dengan parameter termasuk kunci akses untuk memungkinkan penggunaan layanan (untuk mendapatkan kunci akses dan menggunakan isbndb.com, Anda harus menyelesaikannya secara gratis pendaftaran di http://isbndb.com/ dan ikuti instruksi untuk mendapatkan akun yang diverifikasi. Di akhir bab ini, kita akan membangun aplikasi web kecil menggunakan Google App Engine (lihat di bawah) yang akan menggunakan layanan RESTful yang disediakan oleh isbndb.com untuk mengambil dan memproses informasi buku dalam XML berdasarkan nomor ISBN dimasukkan oleh pengguna. Tidak seperti layanan SOAP yang memerlukan respons yang dikemas dalam XML, REST tidak meminta tanggapan untuk berada dalam format tertentu. Ilustrasi sederhana disediakan oleh Twitter yang akan mengembalikan hasil dalam berbagai format (RSS, ATOM dan JSON) menggunakan URL yang ditunjukkan di bawah ini untuk mengirim permintaan REST ke layanan pencarian Twitter. <http://search.twitter.com/search.atom?q=turing&count=5> 'Q' mengikuti tanda tanya adalah istilah kueri ('turing'), dan hitungan mewakili jumlah item maksimum yang akan dikembalikan. REST didasarkan pada protokol HTTP dan begitu juga arsitektur 'stateless'. Ini mudah untuk membuat aplikasi berbasis AJAX RESTful, dan mengembalikan respons dalam format JSON membuat aplikasi sederhana untuk memprogram sementara minimis-ing bandwidth yang dibutuhkan untuk respon. ISI di mana-mana di layanan cloud, misalnya, Amazon.com menawarkan API RESTful dengan kata kerja HTTP untuk S3 mereka solusi penyimpanan (lihat di bawah). Sumber daya adalah elemen kunci dari pendekatan RESTful. Sumber daya diidentifikasi oleh URL, dan status dan fungsionalitas aplikasi diwakili menggunakan sumber daya. SOA dapat dibangun menggunakan layanan REST - kadang-kadang dengan pendekatan disebut sebagai ROA (REST-oriented architecture). Keuntungan utama ROA adalah kemudahan implementasi, kelincahan desain dan pendekatan yang ringan. The lat-versi est dari WSDL sekarang mengandung kata kerja HTTP dan dianggap dapat diterima metode mendokumentasikan layanan REST. Ada juga alternatif yang dikenal sebagai WADL (Bahasa Deskripsi Aplikasi Web).

1. Tumpukan Cloud

Ini berguna untuk melihat penawaran cloud sebagai tumpukan, di mana setiap lapisan memiliki fitur yang berbeda dan kemampuan. Ini kadang-kadang disebut sebagai 'cloud ontology', dan beragam model kompleksitas yang berbeda telah diajukan. Kami menyajikan yang relatif sederhana model di sini seperti yang ditunjukkan pada Gambar. 5.1



Di bagian atas tumpukan adalah aplikasi dengan antarmuka pengguna yang dikirimkan ke pengguna melalui browser web. Di bagian bawah tumpukan terletak perangkat keras fisikmesin benar-benar menjalankan aplikasi, biasanya terletak di pusat data besar.Virtualisation teknologi memungkinkanuntuk abstraksi dan presentasi dari hard sumber daya ware secara diskrit dan terukur ke lapisan yang lebih tinggi. Sebenarnya, lapisan tertentu dapat didefinisikan sesuai dengan tingkat abstraksi, sehingga bisa kita katakana bahwa lapisan diklasifikasikan lebih tinggi di tumpukan awan jika layanan yang ditawarkan oleh lapisan itu dapat disusun dari layanan lapisan yang mendasarinya. Lapisannya juga bias diidentifikasi sesuai dengan jenis pengguna yang ditargetkan oleh lapisan. Model layanan membangun satu sama lain dan mendefinisikan bidang tanggung jawab di antara vendor, berakhir pengguna dan pengembang. Dalam bab ini, kami terutama tertarik pada tiga lapisan layanan di tengah tumpukan, dan bagian berikut menjelaskan masing-masing dengan contoh.

1. Software sebagai Layanan (SaaS)

Kemampuan yang diberikan kepada konsumen adalah menggunakan aplikasi penyedia yang berjalan pada infrastruktur cloud. Aplikasi dapat diakses dari berbagai perangkat klien melalui antarmuka klien yang tipis, seperti browser web (misalnya email berbasis web), atau antarmuka program. Konsumen tidak mengelola atau mengendalikan infrastruktur awan yang mendasarinya termasuk jaringan, server, sistem operasi, penyimpanan, atau bahkan capa aplikasi individu bilities, dengan kemungkinan pengecualian dari konfigurasi konfigurasi aplikasi user-speci fi c terbatas. NIST Perangkat lunak sebagai layanan (SaaS) adalah aplikasi yang di-host yang tersedia melalui Internet melalui browser web. SaaS, kadang-kadang disebut sebagai 'perangkat lunak berdasarkan permintaan', adalah yang paling banyak lengkap dari layanan cloud. Komputasi perangkat keras, perangkat lunak dan solusinya ditawarkan oleh vendor. Lapisan aplikasi cloud adalah satu-satunya layer yang terlihat oleh pengguna akhir yang merupakan pengguna target untuk lapisan ini. Pengguna akhir tidak mengelola atau mengontrol infrastruktur yang mendasarinya. Satu-satunya tanggung jawab mereka adalah untuk memasuki dan mengelola mereka data berdasarkan interaksi dengan perangkat lunak. Pengguna berinteraksi langsung dengan yang dihosting perangkat lunak melalui browser. Kita harus mencatat bahwa SaaS sudah ada sebelum konsep cloud computing muncul; Namun demikian, itu sekarang merupakan bagian integral dari model cloud. Membuat dan mengirimkan perangkat lunak melalui lapisan SaaS merupakan alternatif yang menarik aplikasi desktop yang lebih tradisional yang harus dipasang pada pengguna mesin. Dengan SaaS, aplikasi ditempatkan di cloud sehingga pekerjaan pengujian, memelihara dan memutakhirkan perangkat lunak sangat disederhanakan karena semuanya dapat terjadi satu tempat daripada diluncurkan ke desktop yang berpotensi ribuan pengguna. Konfigurasi dan pengujian berkurang dalam kompleksitas karena sentralisasi dan pembatasan yang telah ditetapkan dalam lingkungan penyebaran. Pengembang juga bisa menggunakan strategi sederhana ketika menerapkan upgrade dan fi xes. Selanjutnya, komposisi, seperti dibahas di atas, menjadi pilihan langsung segera setelah layanan cloud dikembangkan. Last but not least, penyedia juga manfaat dari perlindungan yang lebih besar kepada kekayaan intelektual mereka sebagai aplikasi tidak dikerahkan secara lokal dan bajakan versi perangkat lunak akan jauh lebih sulit diperoleh dan didistribusikan. Sejumlah karakteristik khas SaaS tercantum di bawah ini:

• Perangkat lunak tersedia secara global melalui Internet, gratis atau dibayar oleh sub- naskah berdasarkan penggunaan pelanggan.

• Kerja kolaboratif mudah diberikan dan umumnya didorong.

• Upgrade otomatis ditangani oleh vendor tanpa masukan pelanggan yang diperlukan.

• Semua pengguna memiliki versi perangkat lunak yang sama.

• Perangkat lunak ini secara otomatis akan meningkatkan permintaan.

• Biaya distribusi dan pemeliharaan berkurang secara signifikan.

Ada berbagai macam aplikasi SaaS yang sudah tersedia, dan jumlah mereka :

* + Salesforce.com

Salesforce.com (http://www.salesforce.com) menyediakan yang paling banyak digunakan layanan manajemen hubungan pelanggan (CRM) yang digunakan untuk mengelola keuangan, masalah logistik dan staf yang berkaitan dengan operasi sistem bisnis. CRM terdiri satu set proses bisnis bersama dengan perangkat lunak yang membantu mengelola pelanggan- kegiatan, informasi dan analisis terkait. Kegiatan-kegiatan ini termasuk:

• Penjualan

• Pemasaran dan prospek

• Analisis isi

• Akun

• Peramalan

• Kemitraan dan kolaborasi

• Pelayanan pelanggan

Salesforce.com menawarkan daftar lengkap fitur yang dapat diintegrasikan dengan aplikasi dari Google. Penawaran SaaS Salesforce.com sangat ketat - dapat dihilangkan dan disesuaikan untuk industri tertentu.

* + 5.8.2 Dropbox

Dropbox (https://www.dropbox.com) menggunakan penyimpanan cloud untuk memungkinkan pengguna menyimpan dan bagikan file dan folder di Internet. Dropbox dimulai dengan tier gratis hingga 2 GB penyimpanan dan mendukung beberapa klien pengguna di seluruh operasi desktop dan seluler sistem, termasuk versi untuk Windows, Linux, Mac OS X, Android dan iPhone. Apa saja fi les atau folder tempat pengguna atau pembaruan di folder dropbox dari mesin lokal mereka disinkronkan dengan versi web. Ketika seorang pengguna masuk ke komputer lain dengan Dropbox terpasang, pembaruan apa pun secara otomatis diterapkan. Dropbox juga menyediakan riwayat revisi sehingga pengguna dapat kembali ke versi lama dari file mereka. Dropbox saat ini menggunakan penyimpanan S3 milik Amazon (lihat di bawah) meskipun mungkin beralih ke penyimpanan yang berbeda penyedia di masa depan.

* 5.8.3 Layanan Google

Google (http://www.google.com/services/) menawarkan berbagai aplikasi cloud tions termasuk email, web hosting dan, tentu saja, pencarian. Sejumlah Google layanan menargetkan pengguna bisnis, seperti Google Analytics yang mendukung targetnya bisnis periklanan. Google Docs agak mirip dengan Microsoft Of ce dan berisi kata pro program cessor, spreadsheet, menggambar dan presentasi dan gratis untuk digunakan hingga a ukuran data tertentu. Dengan Google Docs, kolaborator dapat berbagi dokumen dan bekerja secara real time dari lokasi yang berbeda. REST API memungkinkan pengguna untuk mengakses secara langsung Google Data Protocol (yang mendasari produk Google) dan untuk mencari Google Konten dokumen. Setelah dipublikasikan ke web, pengguna diberikan kepada public URL dan kode HTML diperlukan untuk menanamkan dokumen di halaman web. Skrip Google Apps memberi pengguna tingkat kontrol yang tinggi terhadap sejumlah Google produk. Spreadsheet Google di antara produk lain dapat diakses dan dikonversikan di-troll menggunakan program JavaScript yang berjalan langsung di server Google. Sebenarnya ini berarti bahwa spreadsheet Google menawarkan lebih dari 'aplikasi spesifik pengguna' pengaturan konfigurasi fi 'dan berbicara dengan sangat keras berada di luar NIST definisi SaaS. Ini bukan situasi yang tidak biasa untuk produk yang sebaliknya secara alami jatuh ke dalam kategori SaaS.

* 5.8.4 Prezi

Prezi (http://prezi.com/) adalah teknologi kolaboratif, yang bebas digunakan kapan saja membuat dan menampilkan presentasi non-linear yang tersedia secara publik (‘prezi’) di awan hingga ukuran 100 MB. Pengguna hanya dapat mengganti Microsoft PowerPoint dengan Prezi untuk presentasi interaktif tetapi masih banyak lagi yang ditawarkan. Layanan ini juga memungkinkan pengguna untuk berkolaborasi secara real time untuk mengedit atau brainstorming, dan Prezi memungkinkan pengguna untuk mengunggah file video dan file gambar. Biaya dibayarkan untuk penyimpanan yang lebih besar dan untuk fitur tambahan seperti kemampuan untuk membuat Prezis of fl ine dan membuat Prezis menjadi pribadi.

1. Platform sebagai Layanan (PaaS)

Kemampuan yang diberikan kepada konsumen adalah untuk menyebarkan ke infrastruktur cloud aplikasi yang dibuat atau dibuat dengan yang dibuat menggunakan bahasa pemrograman, pustaka, layanan, dan alat yang didukung oleh penyedia. Konsumen tidak mengelola atau mengendalikan infrastruktur cloud yang mendasarinya termasuk jaringan, server, sistem operasi, atau penyimpanan, tetapi memiliki kontrol atas aplikasi yang digunakan dan mungkin konfigurasi konfigurasi untuk lingkungan aplikasi-hosting. NIST Bagi banyak bisnis, spesifikasi spesifik daripada persyaratan umum dapat berarti bahwa Aplikasi SaaS generik tidak akan memenuhi standar. PaaS menciptakan lingkungan yang dikelola di cloud di mana aplikasi yang kompleks dan khusus dapat dibuat, diuji, dan dikerahkan. Penyedia menyediakan platform perangkat keras bersama dengan perangkat lunak ronments yang dirancang khusus untuk mendukung pengembangan aplikasi cloud. The tar- dapatkan pengguna adalah pengembang yang membuat, menguji, menyebarkan, dan menyetel aplikasi di awan platform (untuk pengguna akhir hasilnya masih berupa aplikasi berbasis browser). Otomatis skalabilitas, pemantauan dan load balancing disediakan sehingga peningkatan permintaan untuk sumber daya seperti aplikasi web tidak akan menghasilkan degradasi dalam perfor- mance (tentu saja, ini mungkin berarti peningkatan biaya yang terjadi). Vendor PaaS juga akan memastikan ketersediaan aplikasi, jadi, misalnya, jika ada masalah pada perangkat keras yang mendasarinya, aplikasi akan secara otomatis rede dikerjakan untuk lingkungan kerja, tanpa efek yang merugikan pada pengalaman pengguna akhir. PaaS harus mempercepat pengembangan dan penerapan dan menghasilkan waktu yang lebih singkat ke pasar bila dibandingkan dengan pengembangan perangkat lunak tradisional menggunakan pusat data organisasi sendiri. Pengembang tidak mengontrol dan tidak bertanggung jawab atas awan yang mendasarinya infrastruktur tetapi memiliki kontrol atas aplikasi yang dikerahkan. Vendor adalah bertanggung jawab untuk semua aspek operasional dan pemeliharaan layanan dan juga akan memberikan informasi penagihan dan meteran. Vendor umumnya menawarkan berbagai alat dan utilitas untuk mendukung desain, integrasi, pengujian, pemantauan dan penyebaran- dari aplikasi. Biasanya set alat akan menyertakan layanan untuk mendukung:

• Kolaborasi dan kerja tim

• Manajemen data — terkadang disebut sebagai data sebagai layanan (DaaS)

• Otentikasi atau identifikasi - terkadang disebut sebagai otentikasi sebagai layanan (AaaS)

• Pemantauan dan manajemen kinerja

• Pengujian

• Layanan antrean

• Email dan pesan

• Komponen antarmuka pengguna

Seringkali alat pendukung dikon ﬁ gurasi sebagai layanan RESTful yang dapat dengan mudah tersusun saat membangun aplikasi. Adalah penting bahwa pengembang dapat menulis aplikasi tanpa perlu mengetahui rincian teknologi yang mendasari awan yang akan disebarkan. Bahasa pemrograman standar adalah sering tersedia, meskipun biasanya dengan beberapa pembatasan berdasarkan keamanan dan masalah skalabilitas. Dukungan tambahan untuk tugas-tugas pemrograman dan kon ﬁ gurasi adalah sering tersedia dari komunitas pengembang khusus untuk penawaran PaaS.

* Portabilitas

Portabilitas menjadi perhatian banyak organisasi yang mempertimbangkan mengembangkan aplikasi PaaS karena persyaratan yang berbeda dari berbagai vendor, jadi, misalnya, jika Anda membangun aplikasi menggunakan Google App Engine, mungkin ada pekerjaan yang signifikan terlibat jika Anda kemudian memutuskan untuk pindah ke Microsoft Azure. Menggunakan lan yang telah ditentukan Rujukan seperti Java pergi beberapa cara untuk memperbaiki kekhawatiran tentang portabilitas aplikasi tetapi tidak perlu menghilangkannya karena kode sering perlu disesuaikan sesuai dengan persyaratan khusus vendor dan kemungkinan besar program tersebut hanya akan bekerja dalam lingkungan spesifik. Cara di mana data bertahan sangat rentan terhadap strategi yang cukup berbeda antara vendor.

* 5.9.2 API Awan Sederhana

Cloud Sederhana bertujuan untuk menyediakan satu set antarmuka umum untuk penyimpanan file layanan, layanan penyimpanan dokumen, layanan antrian sederhana dan layanan infrastruktur keburukan, atau dengan kata-kata mereka sendiri, 'Simple Cloud API membawa teknologi cloud ke PHP dan filosofi PHP ke cloud ’(http://simplecloud.org/). Zend punya mengundang komunitas open source dan vendor perangkat lunak dari semua ukuran untuk berpartisipasi. IBM, Microsoft, Rackspace, Nirvanix, dan GoGrid telah bergabung dengan proyek ini sebagai kontributor.

* 5.9.3 Java

Banyak bahasa pemrograman tersedia pada berbagai penawaran PaaS, tetapi Java Saat ini tampaknya menjadi yang paling populer. Meskipun Java terkadang disebut untuk agak mengejek oleh pengembang sebagai 'COBOL abad ke dua puluh satu', Java adalah bahasa berorientasi objek yang kuat yang tersedia secara gratis dan menawarkan sejumlah fitur menarik seperti manajemen memori otomatis. Mungkin yang paling penting ada banyak pengembang dengan keahlian Java: saat ini dilaporkan sepuluh juta pengguna Java. Bahasa memiliki daya tarik tersendiri mereka yang membangun aplikasi web client-server. Aplikasi Java biasanya dikompilasi untuk membentuk sumber Java ke Java bytecode. Setelah dalam format ini kode dapat berjalan pada Java Virtual Machine (JVM) terlepas dari arsitektur komputer. Salah satu tujuan utama bahasa adalah jumlah diramalkan dalam ‘menulis sekali, jalankan di mana saja’ yang terkenal, yang berarti kode itu berjalan di satu platform (misalnya Windows) tidak perlu diedit untuk dijalankan pada yang berbeda satu (mis. Linux). Alasan lain untuk bandingnya adalah banyaknya dukungan alat porting, pustaka dan kerangka kerja juga tersedia secara bebas dan umumnya terbuka sumber. Ini termasuk sumber terbuka yang banyak digunakan dan Eclipse yang tersedia secara bebas Integrated Development Environment (IDE) yang mendukung pengembangan yang cepat dan debugging. Perpustakaan JUnit (yang dapat dengan mudah diintegrasikan ke dalam Eclipse) mendukung pengembangan berbasis tes dan pendekatan tangkas untuk produksi perangkat lunak. Web aplikasi ARchive (WAR) adalah format standar yang digunakan untuk mengemas dan mendistribusikan sumber daya yang digunakan untuk aplikasi Java Web yang bertujuan untuk memfasilitasi pengembangan, uji ing dan penyebaran. Sejumlah vendor seperti Amazon, Google dan VMForce telah dikembangkan ‘Plug-in’ untuk Eclipse sehingga penawaran Java PaaS mereka dapat dikembangkan, diuji, debugging dan disebarkan di lingkungan yang sudah akrab bagi banyak pengembang. Itu Format Warfile juga sering digunakan oleh sejumlah vendor PaaS, lagi-lagi mem-porting gaya pengembangan yang sudah dikenal dan mengurangi masalah portabilitas.

* 5.9.4 Google App Engine

Beberapa tahun yang lalu sebelum cloud benar-benar menjadi berita utama, Google secara terkenal memberikannya stiker yang mengatakan ‘Komputer saya yang lain adalah pusat data’. Google App Engine (GAE) adalah contoh PaaS yang baik dan memungkinkan pengguna untuk menulis dan menyebarkan aplikasi mereka infrastruktur yang sama yang menggerakkan produk Google. Kami akan memberikan sedikit lebih banyak waktu di sini untuk membahas GAE karena ini adalah produk yang akan kita gunakan pada akhir bab ini ketika Anda akan mengembangkan aplikasi cloud GAE Anda sendiri.

1. Bahasa pemrograman Python adalah yang pertama muncul ketika GAE berada awalnya dibuat tersedia untuk pengembang. Python System Development Kit (SDK) masih tersedia dan banyak digunakan.

2. Java telah ditambahkan bersama dengan integrasi dengan Eclipse. Menggunakan plug-in Eclipse pengembang dapat menjalankan dan menguji aplikasi Java mereka secara lokal dan menyebarkannya cloud dengan satu klik. Google menggunakan Java Virtual Machine dengan Jetty mesin servlet dan struktur FI PERANG standar. Bahasa pemrograman apa saja yang dapat berjalan pada interpreter berbasis JVM, seperti JRuby, Groovy, JavaScript (Rhino) dan Scala, juga dapat dijalankan di GAE meskipun mungkin ada sedikit lebih banyakbekerja dalam bentuk konfigurasi awal. GAE juga mendukung banyak stan Java- dards dan kerangka kerja seperti servlets, Spring Framework dan Apache Struts.

3. Baru-baru ini lingkungan 'eksperimental' telah ditambahkan untuk ‘Go’ yang baru (http://golang.org/) bahasa pemrograman konkuren yang dikembangkan oleh Google. Java, Python, dan Go runtime environments disediakan dengan API (aplikasi antarmuka program) untuk berinteraksi dengan lingkungan runtime Google. Aplikasi Google Engine menyediakan pengembang dengan lingkungan simulasi untuk membangun dan menguji aplikasi- secara lokal dengan sistem operasi apa pun yang mendukung versi yang sesuai Lingkungan bahasa Python, Java, atau Go. Namun, ada sejumlah penting pembatasan tant. Pengembang tidak dapat menulis ke sistem file, aplikasi Java dapat- tidak membuat utas baru, dan hanya sebagian dari kelas dan paket Java standar tersedia (Anda dapat melihat daftar kelas yang didukung di [http://code.google.com/appengine / docs / java / jrewhitelist.html](http://code.google.com/appengine%20/%20docs%20/%20java%20/%20jrewhitelist.html)). Pembatasan seperti ini telah menyebabkan kekhawatiran portabilitas, tetapi baru-baru ini nomor proyek seperti AppScale telah dapat menjalankan aplikasi Python, Java dan Go GAE kasi pada EC2 dan vendor cloud lainnya. Platform GAE juga memungkinkan pengembang untuk menulis kode dan mengintegrasikan aplikasi yang dirancang dengan layanan Google lainnya. GAE juga memiliki sejumlah layanan pendukung yang menguntungkan termasuk:

• Layanan web terintegrasi

• Otentikasi menggunakan Akun Google

• Penyimpanan tanpa skema yang skematis dan non-relasional menggunakan persistensi Java standar model

• Penyimpanan cepat di memori menggunakan cache nilai kunci (memcache)

• Antrian dan penjadwalan tugas GAE memungkinkan aplikasi untuk dilayani dari nama domain pengembang sendiri meskipun secara default ini akan menjadi nama yang ditetapkan pengembang untuk aplikasi mereka di appspot.com, misalnya, <http://myApp.appspot.com>. URL dapat dibagikan secara publik atau selektif. Kode yang diperlukan untuk aplikasition dapat disimpan pada kode Google yang dapat dikelola oleh berbagai sumber terbuka sistem kontrol versi seperti SVN (<http://subversion.apache.org/>).

Keuntungan besar, terutama bagi pengembang baru, adalah Anda dapat mulai menggunakannya GAE gratis tanpa kartu kredit diperlukan sebelum melanjutkan ke biaya lulus berdasarkan penggunaan (ini adalah salah satu alasan kami memilih tutorial menggunakan GAE). Pada saat menulis, setiap pengembang dapat menghosting hingga sepuluh aplikasi gratis. Sebuah aplikasi di akun gratis dapat menggunakan hingga 1 GB penyimpanan dan hingga lima juta tampilan halaman bulan. Untuk mempertahankan data menggunakan GAE, aplikasi harus menggunakan mesin aplikasi yang disediakan jasa . Data store adalah sistem penyimpanan nilai-kunci yang mirip dengan Amazon SimpleDB dan Layanan Tabel Windows Azure (lihat di bawah). GAE menyediakan ketersediaan tinggi oleh replikasi data dan sinkronisasi. Layanan memcache digunakan untuk meningkatkan kinerja dengan menggunakan jangka pendek, dalam cache lokal memori daripada pergi ke penyimpanan data. Meskipun data tidak dapat dibaca dari sistem file, dimungkinkan untuk membaca fi les yang merupakan paket sebagai bagian dari PERANG.

* 5.9.5 Google Web Toolkit

Google Web Toolkit (GWT) adalah seperangkat pustaka sumber terbuka di bawah Apache 2.0 lisensi, yang memungkinkan pengembang web untuk menulis aplikasi web AJAX sepenuhnya di Java. GWT akan mengkompilasi Java untuk mengoptimalkan kode JavaScript yang akan dijalankan di browser pengguna sementara memungkinkan untuk strategi pengujian Java yang matang, misalnya, menggunakan JUnit. GWT bertujuan untuk meningkatkan produktivitas pengembang dan pemeliharaan aplikasi web masalah keuangan dan termasuk dukungan untuk komunikasi asynchronous (AJAX), ini- manajemen tory, bookmark, internasionalisasi dan portabilitas lintas browser. Aplikasi GWT yang menggunakan HTML5 dapat memiliki tampilan terpisah untuk tablet dan seluler telepon. Ada juga plug-in GWT ke IDE gerhana dan solusi langsung cedure untuk menyebarkan aplikasi GWT ke GAE.

* 5.9.6 Microsoft Azure

Platform Layanan Azure menyediakan beragam layanan berbasis Windows untuk mengembangkan dan menyebarkan aplikasi berbasis Windows di cloud. Sementara Azure terutama dirancang sebagai tawaran PaaS di mana pengguna dapat membangun web host dan skala aplikasi di pusat data global Microsoft, itu juga termasuk Infrastruktur sebagai Fitur Layanan (IaaS). Platform ini terdiri dari komputasi, penyimpanan dan kain (net sumber daya kerja). Azure menyediakan penskalaan, penyeimbangan beban, dan sejumlah dukungan layanan seperti identitas dan otentikasi, enkripsi pesan, pemantauan dan pengelolaan.Tiga merek utama telah dikembangkan:

1. Sistem operasi cloud Windows Azure yang dapat diskalakan

2. SQL Azure yang merupakan versi scalable berbasis cloud dari Microsoft SQL Server

3. AppFabric yang merupakan kumpulan layanan yang mendukung layanan cloud Aplikasi Azure biasanya ditulis baik sebagai ‘peran web’, ‘peran pekerja’ atau kedua. Peran web digunakan untuk aplikasi web biasa yang di-host di server run-ning Microsoft IIS web server, sementara peran pekerja digunakan untuk aplikasi itu memerlukan operasi intensif komputasi. Peran mesin virtual saat ini dalam versi beta tetapi memungkinkan pengembang untuk menggunakan gambar mesin Windows Server yang dibuat khusus ke Windows Azure berpotensi menghemat waktu signi fi cant pada tugas-tugas yang terkait dengan instalasi. Azure menggunakan kombinasi teknologi layanan web standar REST, HTTP dan XML dan terintegrasi dengan Microsoft Visual Studio. Dengan menggunakan Microsoft Azure SDK, pengembang dapat membuat layanan yang memanfaatkan .NET Framework. Itu tujuannya adalah untuk memungkinkan pengembang untuk dapat memilih bahasa atau kerangka kerja dan untuk mengintegrasikan aplikasi cloud publik dengan aplikasi yang ada. Aplikasi ini harus diunggah melalui portal Microsoft Azure untuk dieksekusi di atas Windows Azure. Layanan tambahan tersedia, seperti eksekusi alur kerja dan manajemen, orkestrasi layanan web dan akses ke penyimpanan data SQL. Microsoft juga menawarkan perangkat pengembangan sistem untuk bahasa lain seperti Java, PHP dan Ruby.

* 5.9.7 Force.com

Force.com dari SalesForce.com adalah pendekatan untuk PaaS untuk mengembangkan CRM sistem dan aplikasi untuk perusahaan sosial dengan dukungan built-in untuk sosial dan fungsi seluler. Desain platform dan lingkungan runtime-nya berdasarkan teknologi Java. Platform menggunakan pemrograman berpemilik bahasa dan lingkungan yang disebut Kode Apex, yang memiliki reputasi untuk kesederhanaan pembelajaran dan pengembangan dan eksekusi yang cepat. Pengguna Force.com dapat membangunnya aplikasi sendiri yang berdiri sendiri atau terintegrasi dengan Salesforce.com.

* 5.9.8 VMForce

Salesfoce.com dan VMware telah bermitra untuk memperkenalkan VMForce, salah satu yang pertama platform cloud perusahaan untuk pengembang Java. VMForce menggunakan Eclipse dan memungkinkan untuk pengembangan dengan teknologi Java Web standar seperti JSP dan servlets. Setelah aplikasi yang dikembangkan dapat digunakan ke cloud dengan satu klik.

* 5.9.9 Heroku

Heroku adalah platform cloud untuk penyebaran aplikasi web Ruby on Rails secara instan. Ruby on Rails menyediakan kerangka kerja lengkap untuk pengembangan aplikasi web termasuk template dan sistem render dan metode untuk bertahan dan mengambil data. Ruby on Rails sangat populer di kalangan pengembang, memiliki rekam jejak yang baik untuk web yang cepat produksi aplikasi dan duduk sangat baik dengan metode perangkat lunak tangkas produksi. Server Heroku secara tidak terlihat dikelola dan aplikasi secara otomatis berskala dan seimbang.

* 5.9.10 Cloud Foundry

Cloud Foundry mengklaim sebagai penawaran PaaS open source pertama yang skalabel, menggunakan kerangka kerja standar industri seperti Spring (Java) dan Scala dan pilihan aplikasi layanan infrastruktur kation. Proyek ini diprakarsai oleh VMware tetapi telah diperoleh dukungan industri yang luas. Cloud Foundry bertujuan untuk meminimalkan pembatasan pada pengembangan opment dan penyebaran di cloud sehingga pengembang tidak perlu khawatir middleware atau infrastruktur vendor.

1. Infrastruktur sebagai Layanan (IaaS)

Kemampuan yang diberikan kepada konsumen adalah penyediaan pemrosesan, penyimpanan, jaringan, dan sumber daya komputasi mendasar lainnya di mana konsumen dapat menyebarkan dan menjalankan arbi- perangkat lunak yang tersembunyi, yang dapat mencakup sistem operasi dan aplikasi. Konsumen melakukannya tidak mengelola atau mengendalikan infrastruktur cloud yang mendasarinya tetapi memiliki kendali atas operasi sistem, penyimpanan, dan aplikasi yang dikerahkan; dan mungkin kontrol terbatas dari jaringan-pilihing komponen (misalnya, host fi rewalls). NIST ([http://csrc.nist.gov/publications/nist-pubs / 800-145 / SP800-145.pdf](http://csrc.nist.gov/publications/nist-pubs%20/%20800-145%20/%20SP800-145.pdf)) IaaS menyediakan pengembang dengan sumber daya infrastruktur sesuai permintaan seperti komputasi, penyimpanan, dan komunikasi sebagai layanan virtualisasi di awan. Penyedia mengelola seluruh infrastruktur dan mengoperasikan pusat data yang cukup besar untuk menyediakan sumber daya yang tampaknya tidak terbatas. Klien bertanggung jawab atas semua yang lain aspek penyebaran yang dapat mencakup sistem operasi itu sendiri, bersama dengan bahasa pemrograman, server web, dan aplikasi. IaaS biasanya menggunakan a model pay-as-you-go dengan vendor yang biasanya mengisi daya per jam. Setelah terhubung, pengembang bekerja dengan sumber daya seolah-olah mereka memilikinya. IaaS sebagian besar telah difasilitasi oleh kemajuan dalam virtualisasi sistem operasi yang memungkinkan tingkat indirection atau abstraksi berkaitan dengan penggunaan perangkat keras langsung (lihat Chap. 4). Mesin virtual (VM) adalah bentuk paling umum untuk menyediakan sumber daya tional, dan pengguna biasanya mendapatkan akses super-pengguna ke mesin virtual mereka. Virtualisasi bentuk sumber daya fundamental seperti daya komputasi, penyimpanan atau bandwidth jaringan disediakan dan dapat disusun untuk membangun yang baru lingkungan atau aplikasi perangkat lunak cloud. Virtualisasi memungkinkan IaaS pro- vider untuk mengontrol dan mengelola pemanfaatan sumber daya fisik yang efisien dan memungkinkan pengguna fleksibilitas yang belum pernah ada sebelumnya dalam konfigurasi sementara melindungi fisik infrastruktur pusat data. Model IaaS memungkinkan aplikasi yang ada langsung dipindahkan dari server organisasi ke perangkat keras pemasok cloud, berpotensi dengan sedikit atau tidak ada perubahan pada perangkat lunak.

* 5.10.1 Peralatan Virtual

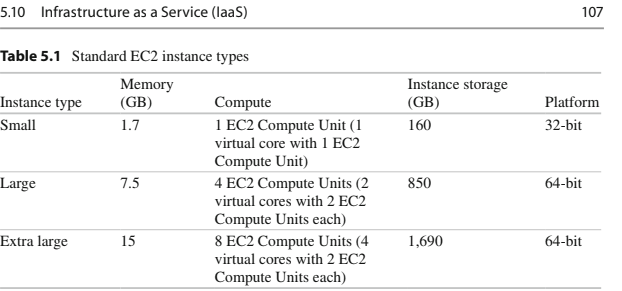
Alat virtual adalah gambar mesin virtual yang dirancang untuk dijalankan secara khusus platform virtualisasi, biasanya dipasok melalui IaaS. Alat virtual mungkin perangkat lunak seperti sistem operasi, kompiler, server web, sistem basis data dan aplikasi apa pun yang diperlukan sudah terpasang, dikonfigurasi dan dioptimalkan untuk menyelesaikanmasalah khusus yang dirancang. Dalam prakteknya banyak peralatan virtual sering digunakan untuk menjalankan satu aplikasi. Kita dapat menganggap alat sebagai aplikasi kation dikombinasikan dengan semua yang dibutuhkan untuk benar-benar dijalankan. Seringkali operasi sistem adalah versi dipreteli dari rekan serba guna nya, suatu proses kali disebut sebagai 'sistem operasi yang cukup' atau 'Juice'. Peralatan virtual biasanya dikontrol menggunakan antarmuka web dan dapat digunakan sebagai titik awal untuk membangun server yang lebih kompleks. Sejumlah pasar online telah disiapkan untuk memungkinkan pertukaran peranti siap pakai yang berisi berbagai macam sistem operasi dan peranti lunak kombinasi. VMWare memiliki pilihan peralatan terbesar (tersedia di http: //www.vmware.com/appliances/directory/) yang dapat segera digunakan untuk VMWare hypervisor. Di Chap. 4 tutorial, kami menggunakan VMWare untuk membuat Ubuntu mesin virtual dan kemudian menginstal Java dan Eclipse ke mesin kami. Sebuah alternative pendekatan akan mengunduh alat yang sesuai dengan perangkat lunak yang kami butuhkan sudah terpasang. Amazon Machine Image (AMI) digunakan untuk menyimpan salinan virtual tertentu mesin yang dapat berjalan di Amazon Compute Cloud (EC2) Elastis dan IaaS lainnya platform yang mendukung format. Amazon Web Services menawarkan ratusan siap membuat AMI, yang disebut Amazon sebagai 'solusi pra-bentukan' yang mengandung kesamaan kombinasi sistem operasi dan perangkat lunak aplikasi.

* 5.10.2 Layanan Web Amazon

Amazon Web Services (AWS) adalah salah satu bisnis berbasis cloud yang paling sukses berdasarkan standar SOA dan mencakup seluruh keluarga layanan terkait. Itu kebutuhan komputasi suatu organisasi tentu saja sangat bervariasi, dan beberapa aplikasi cenderung menghitung intensif, sementara yang lain akan menekankan penyimpanan EC2 terkenal karena kemampuannya memberikan solusi terukur untuk beragam persyaratan.

* 5.10.3 Amazon Compute Cloud (EC2) Elastis

EC2 menyediakan kekuatan komputasi virtual sebagai layanan dan dianggap sebagai pusat penerapan penawaran AWS dan sering disebut sebagai 'contoh model' dari IaaS. Pengguna dapat menyewa daya komputasi di infrastruktur Amazon dan membayar dengan tarif per jam menggunakan EC2. EC2 dapat dikustomisasi dengan cara yang mirip dengan server fisik dan dapat menjalankan berbagai sistem operasi termasuk Linux, Solaris dan Windows server Pengguna mendapatkan hak istimewa administrator dan dapat melakukan berbagai kegiatan di server seperti memulai dan berhenti, konfigurasi dan instalasi dari setiap sistem basis data, bahasa pemrograman atau paket perangkat lunak. Sumber daya komputasi yang tersedia pada instance EC2 terdiri dari kombinasi daya komputasi dan sumber daya lainnya seperti memori. EC2 ditawarkan sebagai meteran layanan, dan EC2 Compute Unit (CU) digunakan sebagai ukuran standar komputasi kekuasaan. Pengguna EC2 dapat memilih dari sejumlah jenis instance yang berbeda untuk paling cocok kebutuhan komputasi tertentu. Amazon Auto Scaling adalah seperangkat alat baris perintah yang memungkinkan peningkatan kapasitas EC2 ke atas atau ke bawah secara otomatis dan sesuai dengan kondisi tions speci fi ed oleh pengguna akhir. Contoh standar saat ini ditawarkan dengan sumber daya berikut sebagai ilus- trated dalam Tabel 5.1.



Ada banyak jenis contoh lain yang dirancang untuk memenuhi persyaratan tidak standar. EC2 juga menawarkan pengguna kemampuan untuk memilih dari sejumlah daerah dan di dalam setiap wilayah sejumlah zona ketersediaan tempat pusat data virtual tempat Instance EC2 akan berjalan.

* 5.10.4 Layanan Penyimpanan Amazon

Berbagai bentuk penyimpanan data juga disediakan oleh Amazon ‘as-a-service’. Amazon Simple Storage Service (S3) adalah sistem penyimpanan berbasis objek murni yang dapat diakses melalui antarmuka web berbasis REST. S3 tidak menyediakan kemampuan untuk berlari program dan bahkan tidak menyediakan sistem atau pengindeksan, tetapi memiliki tages dari keandalan, ketersediaan dan skalabilitas. Pengguna dapat mengalokasikan blok penyimpanan- usia hingga 5 TB dalam ukuran yang memiliki identifier unik yang ditetapkan pengguna dan kemudian dapat membaca atau menulis ke blok. S3 sering dikutip sebagai contoh yang baik dari cloud yang fokus sempurna layanan yang melakukan satu hal dengan cara yang sangat sempit. Layanan kemudian bias disusun dengan orang lain untuk menyediakan sumber daya yang dibutuhkan untuk pengembangan. Pengguna S3 dibebankan sesuai dengan volume data dan bandwidth jaringan digunakan saat menyimpan dan mengambil data. S3 adalah pilihan populer di antara aplikasi web pengembang tion dan dengan mudah dapat digunakan untuk menggantikan infrastruktur web hosting yang ada mendatang. Sistem Apache Hadoop fi (lihat Bab 4) juga dapat di-host pada S3 untuk menjalankan pekerjaan MapReduce. Pada saat penulisan Amazon S3 dilaporkan untuk menyimpan lebih dari 762 miliar objek. Dropbox dan Ubuntu One adalah contoh online layanan backup dan sinkronisasi yang menggunakan S3 untuk penyimpanan dan transfer. Amazon SimpleDB adalah penyimpanan data NoSQL nilai-kunci (lihat Bab 6) dan Relasional Amazon Layanan Database menyediakan contoh database MySQL di cloud. Seperti yang disebutkan di atas, Amazon S3 bekerja melalui HTTP. Amazon juga menawarkan blok sistem penyimpanan di mana sumber daya tampak menjadi cakram fisik pada server EC2. Ini memberikan peningkatan kinerja lebih dari S3 dan lebih cocok untuk kasus ini di mana beberapa operasi disk diperlukan. Instance penyimpanan dilampirkan ke par-contoh EC2 ticular, hanya dapat diakses oleh EC2 tertentu dan akan hilang segera setelah turunan EC2 berhenti. The Elastic Block Storage (EBS) menyediakan blok penyimpanan untuk EC2 tetapi tidak bergantung pada instance tertentu dan dapat dibagikan antara instance EC2. EC2 menyediakan Penyeimbang Beban Elastis yang sesuai dengan namanya menyeimbangkan beban di beberapa server. Load balancer juga menskalakan jumlah server naik atau turun tergantung pada persyaratan beban tertentu.

* 5.10.5 Amazon Elastic Beanstalk

AWS Elastic Beanstalk saat ini dalam versi beta tetapi berjanji untuk memberikan 'yang lebih mudah cara bagi Anda untuk menyebarkan dan mengelola aplikasi dengan cepat di cloud AWS ’. Itu pengembang hanya perlu mengunggah aplikasi mereka, dan Beanstalk Elastis akan menangani rincian penerapan, penyeimbangan beban, dan penskalaan. Dalam beberapa hal ini terjadi produk lebih seperti penawaran PaaS meskipun dalam hal ini pengembang dapat mengontrol infrastruktur yang mendasarinya jika diperlukan (Beanstalk akan menggunakan sumber daya seperti EC2 dan S3 saat menyebarkan aplikasi). Amazon dengan jelas menyatakan bahwa tujuannya adalah untuk menawarkan pengembang yang terbaik dari dunia PaaS dan IaaS. Rilis Elastic Beanstalk saat ini dibuat untuk pengembang Java, dan Aplikasi beanstalk elastis harus dikemas menggunakan Java WAR standar struktur. Sebuah plug-in AWS tersedia untuk Eclipse yang juga memungkinkan Anda untuk menguji dan menyebarkan aplikasi Anda ke satu atau beberapa contoh EC2 yang menjalankan Apache Tomcat server web.

* 5.10.6 FlexiScale

FlexiScale (http: // www. Fl exiscale.com/) menawarkan hosting dan beroperasi sesuai permintaan dengan cara yang agak mirip dengan Amazon EC2. FlexiScale mendukung Windows dan server Linux dan menggunakan hypervisor Xen.

* 5.10.7 GoGrid

GoGrid (http://www.gogrid.com/) memungkinkan pengguna untuk memanfaatkan berbagai jendela pra-pembukaan dan gambar Linux dalam berbagai ukuran contoh tetap dan juga menawarkan tumpukan di atasnya aplikasi seperti aplikasi web bervolume tinggi, e-commerce, dan basis data.

* 5.10.8 Eucalyptus (‘Arsitektur Komputasi Utilitas Elastis

untuk Menghubungkan Program Anda ke Sistem Berguna ’) Eucalyptus (http://www.eucalyptus.com/) adalah perangkat lunak berbasis Linux open source ware platform untuk menciptakan sistem IaaS komputasi awan berbasis computer kelompok. Proyek ini memiliki antarmuka yang dapat dihubungkan ke komputasi Amazon dan sistem penyimpanan cloud (EC2 dan S3), dan pengguna dapat berinteraksi dengan Eucalyptus awan menggunakan alat yang sama yang digunakan di Amazon. Eucalyptus mendukung banyak hypervisor teknologi dalam cloud yang sama dan menyediakan berbagai alat administrasi cloud untuk manajemen sistem dan akuntansi. Eucalyptus saat ini didukung oleh sebagian besar distribusi Linux. Ubuntu enterprise cloud adalah inisiatif baru untuk dibuat lebih mudah untuk menyediakan, menyebarkan, mengkon ﬁ gurasi dan menggunakan infrastruktur cloud berdasarkan Eucalyptus

* 5.10.9 Rackspace

Rackspace (http://www.rackspace.com/) menawarkan contoh IaaS solusi fi xed ukuran di awan. Server cloud Rackspace menawarkan berbagai gambar yang dibuat berbasis Linux, dan pengguna dapat meminta ukuran gambar yang berbeda.

1. Ringkasan Bab

Dalam bab ini, kita telah melihat bagaimana menggunakan layanan web dan layanan yang berorientasi arsitektur telah berkontribusi pada munculnya komputasi awan. Kami telah meninjau Model SPI dan memeriksa bagaimana hal itu dapat berguna untuk melihat komputasi awan sebagai konsisten- ing dari berbagai lapisan yang paling mudah diidentifikasi dengan mempertimbangkan yang khusus tanggung jawab vendor, pengembang, dan pengguna akhir. Kami harus mencatat bahwa meskipun model SPI sangat berguna dalam mengkategorikan dan memahami cloud-based sistem, lapisan agak kabur karena, misalnya, beberapa penawaran SaaS dapat diprogram dan beberapa produk PaaS menawarkan opsi opsi langsung kepada pengembang trol dari perangkat keras yang mendasarinya.

5.11.1 Akhir dari Bab Latihan

Dalam tutorial ini kita akan bekerja dengan penawaran cloud Google untuk membuat yang sederhana aplikasi web. Pilihan ini sebagian karena ketersediaan tingkat gratis Google sebagai kami menghargai bahwa banyak siswa dari komputasi awan dapat ditunda oleh keharusan membeli sumber daya cloud atau memberikan rincian kartu kredit. Buat akun Google App Engine (gratis) dengan mengeklik tautan Mendaftar di bawah Memulai di http://code.google.com/appengine/ dan ikuti petunjuknya. Tidak ada kartu kredit yang diperlukan untuk tier gratis, tetapi Anda harus dapat menerima Pesan teks SMS. Saat ini ada tiga bahasa pemrograman yang dapat Anda gunakan pada GAE, tapi kami akan menggunakan Java. Satu keuntungan dari ini adalah kita dapat menggunakan plug-GAE ke IDE gerhana yang tersedia gratis yang kami pasang ke VM kami sebelumnya.