Chapter 5
Cloud Management

Setting up appropriate management procedures is vital both for operators and for users of cloud services. Services must be described, provisioned, and billed. In order to achieve the required service scalability and reliability, automated processes are employed. When services are transferred from the local context into the public cloud, security issues and risk assessment play an important role. This chapter deals with the related cloud management aspects.

5.1 Service Level Agreements (SLAs)

A *service level agreement* is an agreement between a service provider and a service consumer related to the service level (quality of service). Such an agreement can be reached by signing a formal and legally binding contract, or informally in case of different departments of a company using the services. This is referred to as an *operation level agreement* (OLA). In terms of quality, the SLA implies a mutual agreement with respect to security, priorities, responsibilities, guarantees, and billing modalities. In addition, the SLA specifies metrics such as availability, throughput, response times, and others. By nature, SLAs always consider the output side, i.e. they are drafted from the service consumer's perspective. A provider can stand out by delivering a service in a superior quality or in a particularly innovative way. From the business perspective, it is possible to agree on different quality levels, e.g. Basic, Silver, Gold, Platinum.

Cloud computing SLAs are interesting when it comes to controlling resource allocation and dynamic resource usage. There are two phases which are essential for *service level management*:

- Agreement on the quality of service
- Service monitoring at runtime

C. Baun et al., *Cloud Computing*, DOI 10.1007/978-3-642-20917-8_5,

With the current cloud offerings, agreeing and monitoring specific SLAs is only possible in a very basic way, and these offerings are usually made on a *best effort* basis. In case of failures or service disruptions, the provider issues a corresponding credit note.

With respect to the cloud architecture, developers are called upon to insert a layer into the cloud stack on which both aspects, i.e. service agreement and service monitoring, will be dealt with. SLA@SOI [128] is a project supported by the European Commission in the EU Seventh Framework Programme. It examines the aspects of multi-level SLAs in a market with competing offerings.

5.2 Lifecycle and Automation

Each cloud service goes through a well-defined lifecycle: The service provider defines the scope and quality of the services and describes their properties in a service catalog. The consumers select the desired services from the catalog and instantiate them as required, while SLAs need to be taken into account and monitored. At the end of the utilization period, service orders are closed, service modules are dissociated and resources are reset. An accounting procedure adds up the usage of all resources. Thus, it is possible to track the current status in a fine-grained and time-resolved manner. The consumers are billed either periodically or each time the costs incurred exceed a certain threshold. Most billing procedures use credit cards, while some providers, such as Zimory, also accept direct debiting or invoicing [141].

Services are often provisioned as so-called ensembles. An ensemble is a group of similar resources which are managed in a fully automatic way. This approach facilitates the scalable provisioning of services with a nearly constant management overhead. In this context, automation includes the following steps:

1. Monitoring
2. Analysis
3. Scheduling
4. Execution

A suitable monitoring procedure constantly checks the quality of service and an analysis component examines and evaluates the monitoring component output. In case of malfunctions or deviations from the agreed performance parameters, an appropriate troubleshooting process is selected from a previously defined portfolio (scheduling). The executing unit finally implements the process by providing additional resources, e.g. for an application or a request. The associated components form a closed loop which is cycled through. Process automation is an essential feature of nearly all cloud architectures because it allows for dynamic scalability and fault tolerance.

5.3 Management Services and Tools

For the management of their services, cloud providers offer a broad range of tools (see Table 5.1). Some are command line-based, others can be accessed from Web portals. From the virtually innumerable number of solutions, we selected some exemplary tools for different areas of application which will be presented below.

5.3.1 Monitoring

Periodic acquisition of performance data is important both to the cloud provider and to the cloud consumer who wants to judge the quality of service. CloudClimate, for example, gathers performance data of different cloud service providers and publishes them on a website [63], including performance metrics such as CPU, memory usage and hard disk access. These values are displayed in charts for the past month. This not only enables the users to directly compare the performance of different providers, but in addition, malfunctions and periods with unusually high loads can be identified.

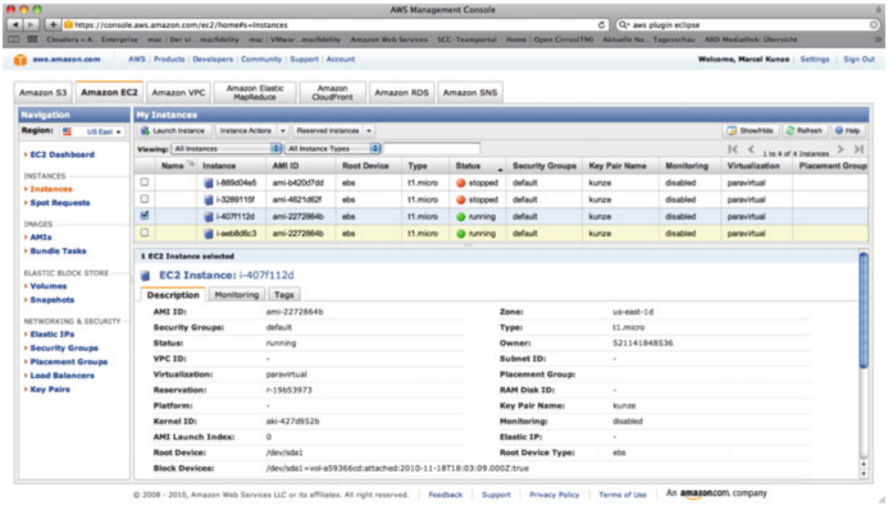
Amazon CloudWatch is a special Web service that monitors the Amazon Web Services [37] performance. CloudWatch allows to view resource usage and displays the current performance data as well as access patterns. This service collects data on CPU usage, disk access, and network traffic. To enable CloudWatch, the user allocates this service to an EC2 instance. The resulting monitoring data can be processed using either the Web service API or command line procedures.

5.3.2 Control

To support the management of its infrastructure services, Amazon not only offers a set of command line tools and libraries for various programming languages [51], but also a convenient Web-based management console [53] (see Fig. 5.1).

With this console, the following services can be controlled and monitored: Amazon S3, Amazon EC2, Amazon Virtual Private Cloud, Amazon Elastic MapReduce, Amazon CloudFront, Amazon Relational Database Service, and Amazon Simple Notification Service. This includes the following EC2 operations:

- **Instances:** Start, stop, restart, remove, access the console, view log files
- **Images:** Start, register, delete, define access rights
- **Memory:** Create, delete, allocate, release, take snapshots
- **Network:** Manage IP addresses
- **Structure:** Manage placement groups and load distribution
- **Security:** Manage security groups and keys



	Name	Instance	AMI ID	Root Device	Type	Status	Security Groups	Key Pair Name	Monitoring	Virtualization	Placement Group
☐		i-889d04e5	ami-b420d7dd	ebs	t1.micro	● stopped	default	kunze	disabled	paravirtual	
☐		i-3289115f	ami-4821d62f	ebs	t1.micro	● stopped	default	kunze	disabled	paravirtual	
☑		i-407f112d	ami-2272864b	ebs	t1.micro	● running	default	kunze	disabled	paravirtual	
☐		i-aeb8d6c3	ami-2272864b	ebs	t1.micro	● running	default	kunze	disabled	paravirtual	

1 EC2 Instance selected

EC2 Instance: i-407f112d

Description | Monitoring | Tags

AMI ID:	ami-2272864b	Zone:	us-east-1d
Security Groups:	default	Type:	t1.micro
Status:	running	Owner:	521141848536
VPC ID:	-	Subnet ID:	-
Virtualization:	paravirtual	Placement Group:	
Reservation:	r-19b53973	RAM Disk ID:	-
Platform:	-	Key Pair Name:	kunze
Kernel ID:	aki-427d952b	Monitoring:	disabled
AMI Launch Index:	0	Elastic IP:	-
Root Device:	/dev/sda1	Root Device Type:	ebs
Block Devices:	/dev/sda1=vol-a59366cd:attached:2010-11-18T18:03:09.000Z:true		

Fig. 5.1 Managing infrastructure with the AWS console

These management operations can be performed separately for the different availability zones. Besides the official EC2 command line tools from Amazon, there are a number of freely available tools that can be used to interact with AWS-compatible cloud services, such as s3cmd [121] and the Euca2ools from Eucalyptus.

An alternative graphical console is available as an open source plug-in for the Firefox browser: *ElasticFox* organizes infrastructure management in a similar way [72]. An advantage of ElasticFox is that it allows to manage not only EC2-based public clouds, but also Eucalyptus-based private clouds. A screenshot of the ElasticFox console is shown in Fig. 5.2.

There are multiple solutions which are targeted at using the Amazon S3 storage service. The Firefox S3Fox Organizer plug-in is of particular interest [122].

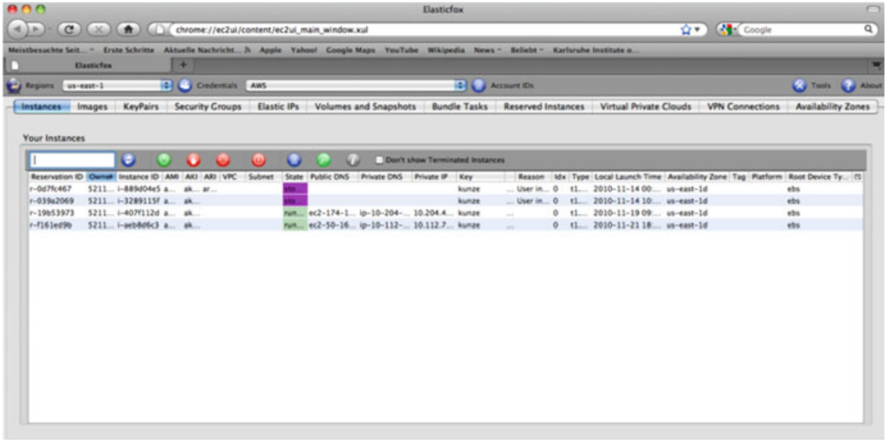
With this tool, a customer can

- Upload, download, delete, or hierarchically organize files,
- Control the visibility of data on the Internet (also temporarily),
- Manage access rights (especially set up access control lists), and
- Automatically synchronize local folders with S3.

The user interface is similar to a traditional FTP client (see Fig. 5.3). It contains two views showing the local and the cloud file structures side-by-side; files can be copied by simple drag-and-drop.

The existing tools for working with cloud services belong to one of the following three basic categories:

- Online tools (services)
- Browser plug-ins
- Command line tools



chrome://ec2ui/content/ec2ui_main_window.xul

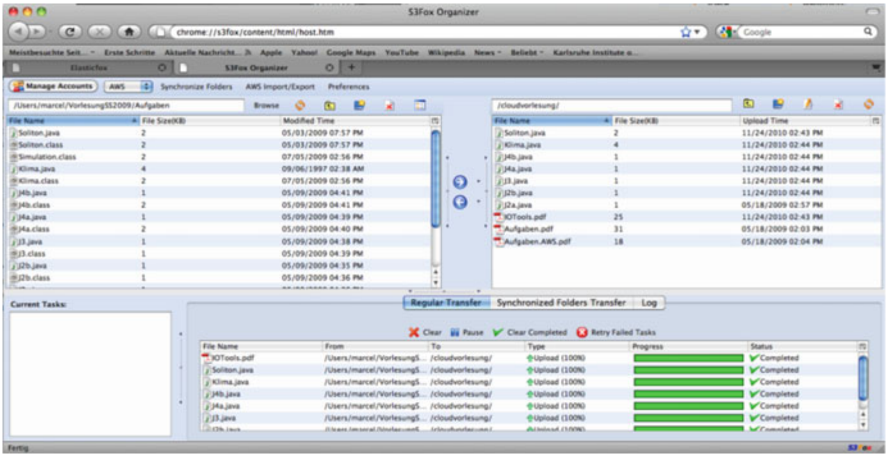
Regions: us-east-1 Credentials: AWS

Your Instances

□ Don't show Terminated Instances

Reservation ID	Owner	Instance ID	AMI	AKI	ARI	VPC	Subnet	State	Public DNS	Private DNS	Private IP	Key		Reason	Idx	Type	Local Launch Time	Availability Zone	Tag	Platform	Root Device Ty...
r-0d7fc467	5211...	i-889d04e5	a...	ak...	ar...			sto...				kunze	...	User in...	0	t1....	2010-11-14 00:...	us-east-1d			ebs
r-039a2069	5211...	i-3289115f	a...	ak...				sto...				kunze	...	User in...	0	t1....	2010-11-14 10:...	us-east-1d			ebs
r-19b53973	5211...	i-407f112d	a...	ak...				run...	ec2-174-1...	ip-10-204-...	10.204.4...	kunze	...		0	t1....	2010-11-19 09:...	us-east-1d			ebs
r-f161ed9b	5211...	i-aeb8d6c3	a...	ak...				run...	ec2-50-16...	ip-10-112-...	10.112.7...	kunze	...		0	t1....	2010-11-21 18:...	us-east-1d			ebs

Fig. 5.2 Infrastructure management with ElasticFox



chrome://s3fox/content/html/host.htm

AWS

/Users/marcel/VorlesungSS2009/Aufgaben

File Name	File Size(KB)	Modified Time
Soliton.java	2	05/03/2009 07:57 PM
Soliton.class	2	05/03/2009 07:57 PM
Simulation.class	2	07/05/2009 02:56 PM
Klima.java	4	09/06/1997 02:38 AM
Klima.class	2	07/05/2009 02:56 PM
J4b.java	1	05/09/2009 04:41 PM
J4b.class	2	05/09/2009 04:41 PM
J4a.java	1	05/09/2009 04:39 PM
J4a.class	2	05/09/2009 04:40 PM
J3.java	1	05/09/2009 04:38 PM
J3.class	1	05/09/2009 04:39 PM
J2b.java	1	05/09/2009 04:35 PM
J2b.class	1	05/09/2009 04:36 PM

/cloudvorlesung/

File Name	File Size(KB)	Upload Time
Soliton.java	2	11/24/2010 02:43 PM
Klima.java	4	11/24/2010 02:44 PM
J4b.java	1	11/24/2010 02:44 PM
J4a.java	1	11/24/2010 02:44 PM
J3.java	1	11/24/2010 02:44 PM
J2b.java	1	11/24/2010 02:44 PM
J2a.java	1	05/18/2009 02:57 PM
IOTools.pdf	25	11/24/2010 02:43 PM
Aufgaben.pdf	31	05/18/2009 02:03 PM
Aufgaben.AWS.pdf	18	05/18/2009 02:04 PM

File Name	From	To	Type	Progress	Status
IOTools.pdf	/Users/marcel/VorlesungS...	/cloudvorlesung/	Upload (100%)		✓ Completed
Soliton.java	/Users/marcel/VorlesungS...	/cloudvorlesung/	Upload (100%)		✓ Completed
Klima.java	/Users/marcel/VorlesungS...	/cloudvorlesung/	Upload (100%)		✓ Completed
J4b.java	/Users/marcel/VorlesungS...	/cloudvorlesung/	Upload (100%)		✓ Completed
J4a.java	/Users/marcel/VorlesungS...	/cloudvorlesung/	Upload (100%)		✓ Completed
J3.java	/Users/marcel/VorlesungS...	/cloudvorlesung/	Upload (100%)		✓ Completed

Fig. 5.3 S3Fox organizer for storage services

The online tools, which are typically implemented as services, include tools such as the AWS Management Console [53] and a corresponding offering from Ylastic [139]. A drawback of these solutions is that customers must trust the tool provider with respect to data privacy and data security because the access data will always be stored with the provider. In addition, the AWS Management Console will only work with services that are part of Amazon Web Services. This means that private cloud services are excluded. Currently, Ylastic only supports Amazon Web Services and infrastructure services based on Eucalyptus.

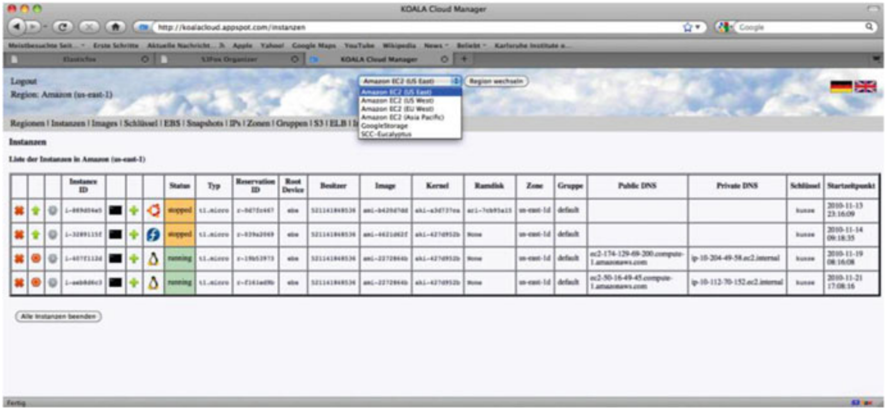
While browser plug-ins, such as ElasticFox and Hybridfox, boast many functions, they require a local installation and thus involve a certain maintenance effort. Each time a service provider switches to a new version, there is a certain risk of interface incompatibilities. What is more, the plug-ins cannot be used with alternative products, such as Chrome, Opera, Safari, Internet Explorer, etc.

Command line tools, such as the EC2 API Tools from Amazon and the Euca2ools from Eucalyptus, also require local installation. Since they have no graphical interface, they are less user-friendly. Just like the other Amazon tools, the EC2 API Tools only work with the AWS public cloud services. The Euca2ools are able to interact with public and private cloud services that are compatible with Amazon Web Services.

KOALA [101, 102] is a new solution for controlling cloud services, which overcomes the limitations of the existing tools. It is a software service capable of controlling public and private cloud infrastructures compatible with AWS interfaces (see Fig. 5.4). Cloud services from Amazon, Eucalyptus, Nimbus, and OpenNebula as well as the management of Google Storage are supported. The service itself can be operated in Google App Engine or alternatively in a compatible private cloud (AppScale and TyphoonAE). KOALA has been published under an open source license (Table 5.1).

Table 5.1 Cloud management tools

Name	Provider	Design	License	Costs	EC2	S3	EBS	ELB	Requirements
KOALA [101]	KIT	SaaS	Apache v2.0	Free	Yes	Yes	Yes	Yes	Browser
AWS console [53]	Amazon	SaaS	proprietary	Free	Yes	Yes	Yes	Yes	Browser
Ylastic [139]	Ylastic	SaaS	proprietary	$25/month	Yes	Yes	Yes	Yes	Browser
ElasticFox [72]	Amazon	Plug-in	Apache v2.0	Free	Yes	No	Yes	No	Firefox
S3Fox [122]	Suchi	Plug-in	proprietary	Free	No	Yes	No	No	Firefox
Euca2ools [74]	Eucalyptus	Shell	BSD	Free	Yes	No	Yes	No	Installation
API tools [50]	Amazon	Shell	Apache v2.0	Free	Yes	No	Yes	Yes	Installation
GSUtil [94]	Google	Shell	Apache v2.0	Free	No	Yes	No	No	Installation
s3cmd [121]	M. Ludvig	Shell	GPLv2	Free	No	Yes	No	No	Installation
AWS toolkit [54]	Amazon	Eclipse	Apache v2.0	Free	Yes	No	No	No	Eclipse



KOALA Cloud Manager
http://koalacloud.appspot.com/instanzen
Google
Meistbesuchte Seit... Erste Schritte Aktuelle Nachricht... Apple Yahoo! Google Maps YouTube Wikipedia News Beliebt Karlsruhe Institute a...
Elasticfox
S3Fox Organizer
KOALA Cloud Manager
Logout
Amazon EC2 (US East)
Region wechseln
Amazon EC2 (US East)
Amazon EC2 (US West)
Amazon EC2 (EU West)
Amazon EC2 (Asia Pacific)
GoogleStorage
SCC-Eucalyptus
Regionen | Instanzen | Images | Schlüssel | EBS | Snapshots | IPs | Zonen | Gruppen | S3 | ELB | I
Alle Instanzen beenden
Fertig

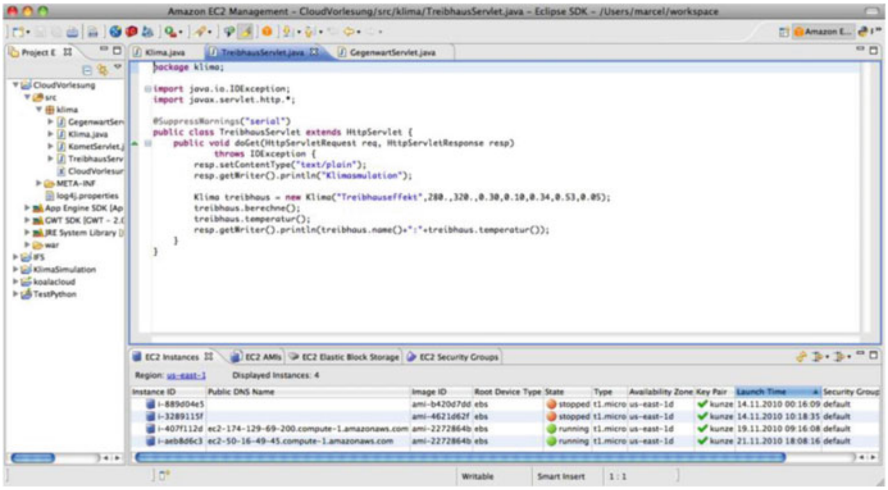
Region: Amazon (us-east-1)

Instanzen

Liste der Instanzen in Amazon (us-east-1)

			Instance ID				Status	Typ	Reservation ID	Root Device	Besitzer	Image	Kernel	Ramdisk	Zone	Gruppe	Public DNS	Private DNS	Schlüssel	Startzeitpunkt
✖	⬆	○	[illegible]	■	✚		stopped	t1.micro	[illegible]	ebs	[illegible]	[illegible]	[illegible]	[illegible]	us-east-1d	default			kunze	2010-11-13 23:16:09
✖	⬆	○	[illegible]	■	✚		stopped	t1.micro	[illegible]	ebs	[illegible]	[illegible]	[illegible]	None	us-east-1d	default			kunze	2010-11-14 09:18:35
✖	●	○	[illegible]	■	✚		running	t1.micro	[illegible]	ebs	[illegible]	[illegible]	[illegible]	None	us-east-1d	default	ec2-174-129-69-200.compute-1.amazonaws.com	ip-10-204-49-58.ec2.internal	kunze	2010-11-19 08:16:08
✖	●	○	[illegible]	■	✚		running	t1.micro	[illegible]	ebs	[illegible]	[illegible]	[illegible]	None	us-east-1d	default	ec2-50-16-49-45.compute-1.amazonaws.com	ip-10-112-70-152.ec2.internal	kunze	2010-11-21 17:08:16

Fig. 5.4 KOALA cloud manager



```
package klima;

import java.io.IOException;
import javax.servlet.http.*;

@SuppressWarnings("serial")
public class TreibhausServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp)
            throws IOException {
        resp.setContentType("text/plain");
        resp.getWriter().println("Klimasmulation");

        Klima treibhaus = new Klima("Treibhauseffekt",280.,320.,0.30,0.10,0.34,0.53,0.05);
        treibhaus.berechne();
        treibhaus.temperatur();
        resp.getWriter().println(treibhaus.name()+":"+treibhaus.temperatur());
    }
}
```

Region: us-east-1 Displayed Instances: 4

Instance ID	Public DNS Name	Image ID	Root Device Type	State	Type	Availability Zone	Key Pair	Launch Time	Security Group
i-889d04e5		ami-b420d7dd	ebs	● stopped	t1.micro	us-east-1d	✓ kunze	14.11.2010 00:16:09	default
i-3289115f		ami-4621d62f	ebs	● stopped	t1.micro	us-east-1d	✓ kunze	14.11.2010 10:18:35	default
i-407f112d	ec2-174-129-69-200.compute-1.amazonaws.com	ami-2272864b	ebs	● running	t1.micro	us-east-1d	✓ kunze	19.11.2010 09:16:08	default
i-aeb8d6c3	ec2-50-16-49-45.compute-1.amazonaws.com	ami-2272864b	ebs	● running	t1.micro	us-east-1d	✓ kunze	21.11.2010 18:08:16	default

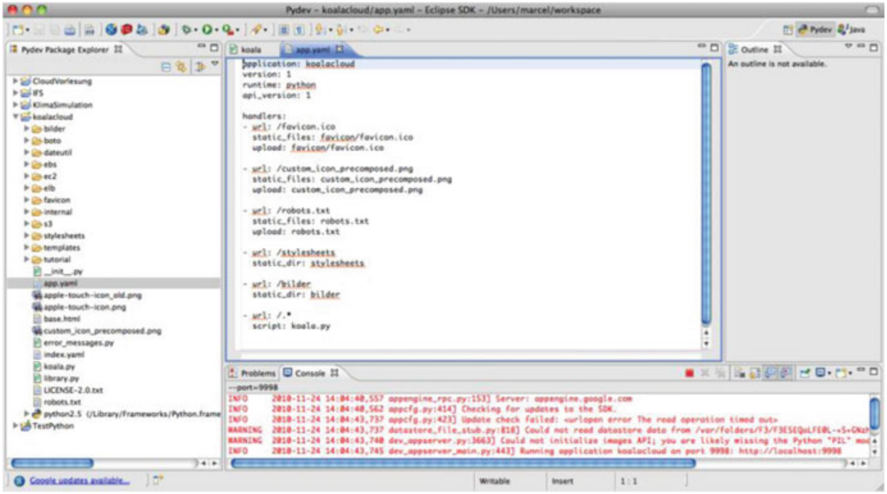
Fig. 5.5 Amazon web service plug-in for eclipse

5.3.3 Development

In a mainly service-oriented landscape, operation and development are often closely intertwined, so that besides the management tools, development tools play an important role. Let us take Eclipse [70] as an example of an integrated cloud development and management environment. Eclipse is a very popular application development platform for which a wealth of plug-ins exist, accommodating the most diverse programming environments. The gEclipse project, for example, features a comprehensive graphical interface for users, developers, and operators of grid and cloud infrastructures [81]. There are also specific Amazon Web Services extensions, such as the AWS Eclipse ToolKit [54] which enables developers to develop and test distributed and scalable Java applications based on Tomcat containers for Amazon EC2 (see Fig. 5.5). Tools for managing security groups, AMI libraries, EC2 instances, and EBS memory are included as well.

With its Google Web Toolkit (GWT), Google provides another way to support cloud application development. Google Web Toolkit is an SDK for the development of Java or Python programs to be run on the App Engine platform [84]. Besides the GWT, this development environment also includes a local Web server for testing the programs. Once developers are happy with the way an application works, they can package it into a so-called WAR file for upload to the scalable Google infrastructure. As an extension of the SDK, there is also a Google plug-in for Eclipse targeted at the interactive development of GWT applications in a graphical, integrated development environment [91].

With a simple click on the App Engine button, locally developed programs can be published to the Google infrastructure (see Fig. 5.6).



```
application: koalacloud
version: 1
runtime: python
api_version: 1

handlers:
- url: /favicon.ico
  static_files: favicon/favicon.ico
  upload: favicon/favicon.ico

- url: /custom_icon_precomposed.png
  static_files: custom_icon_precomposed.png
  upload: custom_icon_precomposed.png

- url: /robots.txt
  static_files: robots.txt
  upload: robots.txt

- url: /stylesheets
  static_dir: stylesheets

- url: /bilder
  static_dir: bilder

- url: /.*
  script: koala.py
```

```
--port=9998
INFO     2010-11-24 14:04:40,557 appengine_rpc.py:153] Server: appengine.google.com
INFO     2010-11-24 14:04:40,562 appcfg.py:414] Checking for updates to the SDK.
INFO     2010-11-24 14:04:43,737 appcfg.py:423] Update check failed: <urlopen error The read operation timed out>
WARNING  2010-11-24 14:04:43,737 datastore_file_stub.py:818] Could not read datastore data from /var/folders/F3/F3ESEQoLFE0L-+S+GNzH
WARNING  2010-11-24 14:04:43,740 dev_appserver.py:3663] Could not initialize images API; you are likely missing the Python "PIL" mod
INFO     2010-11-24 14:04:43,745 dev_appserver_main.py:443] Running application koalacloud on port 9998: http://localhost:9998
```

Fig. 5.6 Google app engine plug-in for eclipse

5.4 Security Management

Security is not only related to safely accessing resources, but also covers data privacy issues. A study by Berkeley University researchers shows that, for the so-called *cloud sourcing*, no specific challenges or problems exist that would require measures other than those currently implemented [1]. With respect to security, the same safety objectives apply as those common to the operation of services in a local data center [28]:

- Confidentiality
- Integrity
- Availability
- Authenticity
- Accountability
- Pseudonymity

Compliance with these safety objectives has to be considered as an integral part of the SLA negotiated between the service provider and the service consumer. The Zimory cloud marketplace, for example, allows to define SLAs with customized security requirements for cloud services [141].

Due to the openness of this approach, the various categories of cloud services have different characteristics:

- IaaS: Highest flexibility, the customer is responsible for security
- PaaS: Medium flexibility, both the customer and the provider are responsible for security
- SaaS: Lowest flexibility, the provider is responsible for security

Many providers simply identify customers by their credit card number.

This also ensures a smooth billing process. Keys or PKI-based processes usually allow controlled resource access and thus prevent unauthorized use. To guarantee the desired confidentiality, data encryption is used for transmission and storage. In this context, it is worth noting that storing encrypted data in the cloud may in some cases be safer than storing unencrypted data on a local PC or in a company data center. In addition, some providers use auditing processes to record all activities related to the use of their resources. This kind of monitoring enables the providers to comply even with the strictest legal provisions.

To guarantee information security, a certification according to ISO/IEC 27001 – or SAS 70 in the U.S. – is of great importance. Here, the processes required to establish and monitor security are stipulated in a binding manner. High-profile providers, such as Amazon or Microsoft, have undergone this certification for their resource centers and can thus offer a higher security standard than smaller company data centers.

There are scenarios where outsourcing of tasks into the cloud is indeed advantageous with respect to security issues: For a company intending to collaborate with external partners in a common project, the firewalls of the partner companies often become almost insurmountable obstacles to establishing common processes. If these processes, however, are transferred, e.g. to the Amazon cloud, it is possible for the cloud provider to autonomously adjust the firewall access rules in such a way that all project partners can collaborate smoothly and without compromising the security guidelines of the participating companies.

For more information on this topic, we refer to the comprehensive compendium on cloud security published by the *Cloud Security Alliance* [66].

5.5 Risk Management

With respect to risk assessment, cloud sourcing is hardly different from the classical "cloud-less" situation: When data has been transferred to the cloud, there is always a risk that access might no longer be available in case of a service disruption or bankruptcy of the provider. These issues, however, are basically the same as with any outsourcing process and can be handled by concluding suitable contracts or SLAs. To mitigate this problem, it is possible to call upon a second, independent cloud service provider as a backup solution to store copies of all business-critical data.

A further risk is the dependency on the proprietary technology and interfaces of a certain provider (vendor lock-in): This dependency problem is a minor one with IaaS and a major one with PaaS and SaaS: When developing a service, e.g. on top of Google App Engine, the developer is locked to the Google-specific infrastructure, and a platform change can be rather complex, time-consuming, and costly. In this context, it should be noted that similar dependencies on software manufactures, platforms, and infrastructures also exist when services are operated in a local data center. Conventional knowledge will therefore help to solve this problem:

- If possible, standardized processes should have priority over a proprietary solution.
- Software should be developed in a way to ensure the highest possible independence from a single platform.
- If dependencies are inevitable, proper encapsulation should be ensured for maximum flexibility.

For an evaluation of the opportunities and risks of cloud computing, see Chap. 8.

5.6 Legal Compliance

Neither cloud service offerings nor their use may infringe laws, social values, morals, or ethics. This is ensured by the concept of *compliance*. According to established case-law, the same laws are applicable to cloud computing as to renting (in case of paid services) or lending (with respect to unpaid services).

The geographical location of the cloud provider is decisive in the determination of the laws that will apply to the data stored. The data may have been replicated so that they exist in different places in the world at the same time and therefore might be subject to different data privacy laws: According to article 4 of the European Data Protection Directive, the pertaining national data protection legislation of the respective country is applicable [138]. For this reason, Amazon, for example, offers cloud services for different regions, such as the U.S., Europe, or Asia. Thus, it is possible, to specify the appropriate judicial area for the services by selecting the desired zone.

In this context, the legal aspects related to personal data processing deserve extra attention. Country-specific laws cover this topic in each country, and for the EU, directive 2000/31/EC is applicable. If data processing takes place outside of Europe, the customer should make sure that an appropriate data protection level exists (e.g. Safe Harbor Agreement in the U.S.). If international resources are involved in the commissioned processing of personal data in the cloud, the consent of the person concerned must always be obtained. From a legal point of view, this kind of data processing is still permitted, though, as soon as the personal character of the data is removed by measures such as encryption or anonymization.

For industries such as healthcare and finance explicit regulations regarding data protection exist such as the Health Insurance Portability and Accountability Act (HIPPA) or the Payment Card Industry Data Security Standard (PCI DSS) [130].