Cloud Computing
Übung 1

Björn Lohrmann
Complex and Distributed IT-Systems
Bjoern.lohrmann@tu-berlin.de

Formalities

- Project Assignment Groups
  - Groups must consist of 3 students (max. 1 group with 2 to 4 people)
  - You must define group membership via ISIS (ISIS-Abstimmung : „Übungsgruppeneinteilung “)
- After this tutorial
  - Opportunity to find group partners will be provided
- Project Assignment 1
  - Will be uploaded on ISIS (~tonight)
  - Submission via ISIS until 10.5.2012 23:59
  - Approx. 12-13 days of time to complete assignment

Formalities

- Amazon AWS grant codes
  - One grant code ($100) per group
  - Will be sent to all group members via ISIS message as soon as groups are complete
  - If your group is full but you still have not received a grant code please email me at bjoern.lohrmann@tu-berlin.de
- Help for project assignment
  - Use the ISIS Discussion forum
  - I will check forums once every 1-2 weekdays for new messages
  - Answering each others questions might be faster
  - Non-public issues via email to me

Amazon AWS

- Amazon Web Services (AWS)
  - Offers many cloud-based services…
    - Elastic Compute Cloud (EC2)
    - Simple Storage Service (S3)
    - Elastic Block Store (EBS)
    - Database Services (SimpleDB, Relational Database Service, …)
    - Messaging (Simple Queue Service, Simple Notification Service, …)
    - Monitoring (CloudWatch)
    - The list goes on…

Amazon AWS

- Pay-as-you-go model: Pay by the hour, Mbyte, …
- Administration via…
  - Command-Line with API/AMI Tools
    - Based on Web-Service API (SOAP)
    - Required for project assignment
  - Browser: AWS Management Console
- AWS accounts
  - Only one account in group can redeem grant code
  - This account can share the grant money via IAM (Identity and Access Management)

AWS: Regions

- Regions
  - Encapsulate different regions of the world with Amazon infrastructure
  - Currently 7 regions (**eu-west-1**, us-east-1, us-west-1, us-west-2, …)
  - User has to choose a region where his services should run
  - Contract: 99,95% availability within region
  - Traffic between regions via Internet (internet charges apply)

AWS: Availability Zones

- Availability Zones
  - Independent data center locations within a region
  - Availability zones are independent of each other
    - Outage in one zone does not affect others
  - High-bandwidth, low-latency network links between zones
  - No charges for inter-zone traffic within the same region

Setting up the API/AMI tools

- Download the API and AMI tools on Amazon AWS site
- Requirements
  - Java Runtime Environment
  - X.509 certificate and private key (Security Credentials)
  - Environment variables:
    - EC2_HOME=<api-tool-directory>
    - EC2_AMITOOL_HOME=<ami-tool-directory>
    - EC2_PRIVATE_KEY=<private-key-path>
    - EC2_CERT=<X.509-cert-path>
    - EC2_URL=<regional-aws-endoint>

Regional AWS Endpoint

- Configure env-variables except EC2_URL
  - Execute `ec2-describe-regions`
  - Output:

```
REGION  eu-west-1       ec2.eu-west-1.amazonaws.com
REGION  sa-east-1       ec2.sa-east-1.amazonaws.com
REGION  us-east-1       ec2.us-east-1.amazonaws.com
REGION  ap-northeast-1  ec2.ap-northeast-1.amazonaws.com
REGION  us-west-2       ec2.us-west-2.amazonaws.com
REGION  us-west-1       ec2.us-west-1.amazonaws.com
REGION  ap-southeast-1  ec2.ap-southeast-1.amazonaws.com
```

  - Set EC2_URL =https://ec2.eu-west-1.amazonaws.com
  - Also consider: `ec2-describe-availability-zones`

EC2

- EC2 is a Infrastructure-as-a-Service cloud
  - Runs Xen 3.x virtualization
  - EC2-Terminology:
    - Instances
      - Virtual machines
      - ID Pattern: i-XXXXXXXX
    - AMI (Amazon machine images)
      - Disk images to boot instances from
      - ID Pattern: ami-XXXXXX
    - AKI (Amazon kernel images)
      - Kernel images to boot instances with
      - ID Pattern: aki-XXXXXXX
    - ARI (Amazon Ramdisk Images)
      - Ramdisks to boot kernels with
      - ID Pattern ari-XXXXXXX

EC2

- More terminology
  - Instance Type
    - Determines hardware properties of instance
    - Currently 13 types
    - Example:
      - m1.small (1 vCPU with 1 EC2 Compute Unit, 1.7GB RAM, 160GB local instance storage)
  - Key pairs
    - SSH keys for remote access to instances
    - Is injected automatically into instance image at instance creation time
    - See `ec2-describe-keypairs` and `ec2-create-keypairs`
  - Security Groups
    - Define firewall rules
    - You should always allow tcp:22 ingoing (for ssh)

Payment Model

- Pay-as-you go
  - EC2-instances
    - Hourly fee depending on instance type and OS type
    - Example: Linux on m1.small costs $0,09 per hour
  - S3
    - Monthly fee depending on S3 space consumed
    - Fixed fee per Webservice request ($0,01 per 1k/10K requests)
    - Example: $0,125 per month for the first TiB
  - EBS
    - Model similar to S3 (I/O requests instead of web-requests)
  - Traffic
    - Intra-region traffic is generally free
    - For inter-region traffic, special rules apply (service dependent)

EC2 AMIs

- AMIs contain the disk images
- Two types of AMIs:
  - EBS-backed
    - Root device in network block device (EBS)
    - Persistent on instance failure/termination
    - Size limit 1TiB
  - Instance-store-backed
    - Root device in local instance storage
    - Not persistent not instance failure/termination
    - Size limit 10GiB

EC2 AMIs

- Usual instance-store backed AMI creation process
  - Create instance with pre-defined AMI
  - Customize AMI (install software, kernels, etc…)
  - Bundle root file system as image (from within instance)
    - `ec2-bundle-vol`
  - Upload bundle to S3 bucket
    - `ec2-upload-bundle`
  - Register the AMI
    - `ec2-register`
- Consult the EC2 documentation for further details

EC2 AKIs

- AKIs contain a kernel image (ARIs a matching RAM-disk)
- Uploading your own AKIs is not possible
- **Problem**: Kernel and remaining OS must be compatible
- **Solution**: Amazon offers „PV-Grub“ AKIs
  - PV-Grub is a paravirtual mini-OS
  - PV-Grub mounts the root device and boots user-provided kernel from /boot
  - User has to provide Grub configuration in /boot/grub/menu.lst
  - Refer to EC2 documentation („Enabling your own kernels“)

- **Note**: All Linux kernels running in Amazon EC2 must have paravirtual guest support (except for some expensive high-performance instance types)

EC2 Instances

- Most important commands
  - List instances: `ec2-describe-instances`
  - Create instance: `ec2-run-instance`
  - Stop instance: `ec2-terminate-instance`

Project Assignment

- Goals:
  - Benchmark paravirtualization on EC2 platform
  - Get familiar with API & AMI tools
  - Think about effects of paravirtualization and resource pooling in clouds
- Benchmarks scripts are provided
- Csv files to collect data
- R script to plot data

Formalities (Again)

Students without project group can find project partners right now!