Cloud Computing
Tutorial Session 2

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

Project Assignment #2

- Goal
  - Write web service to determine whether number is prime (compute intensive)
  - Create automatically scaling web application
- Q: What infrastructural components will be required?

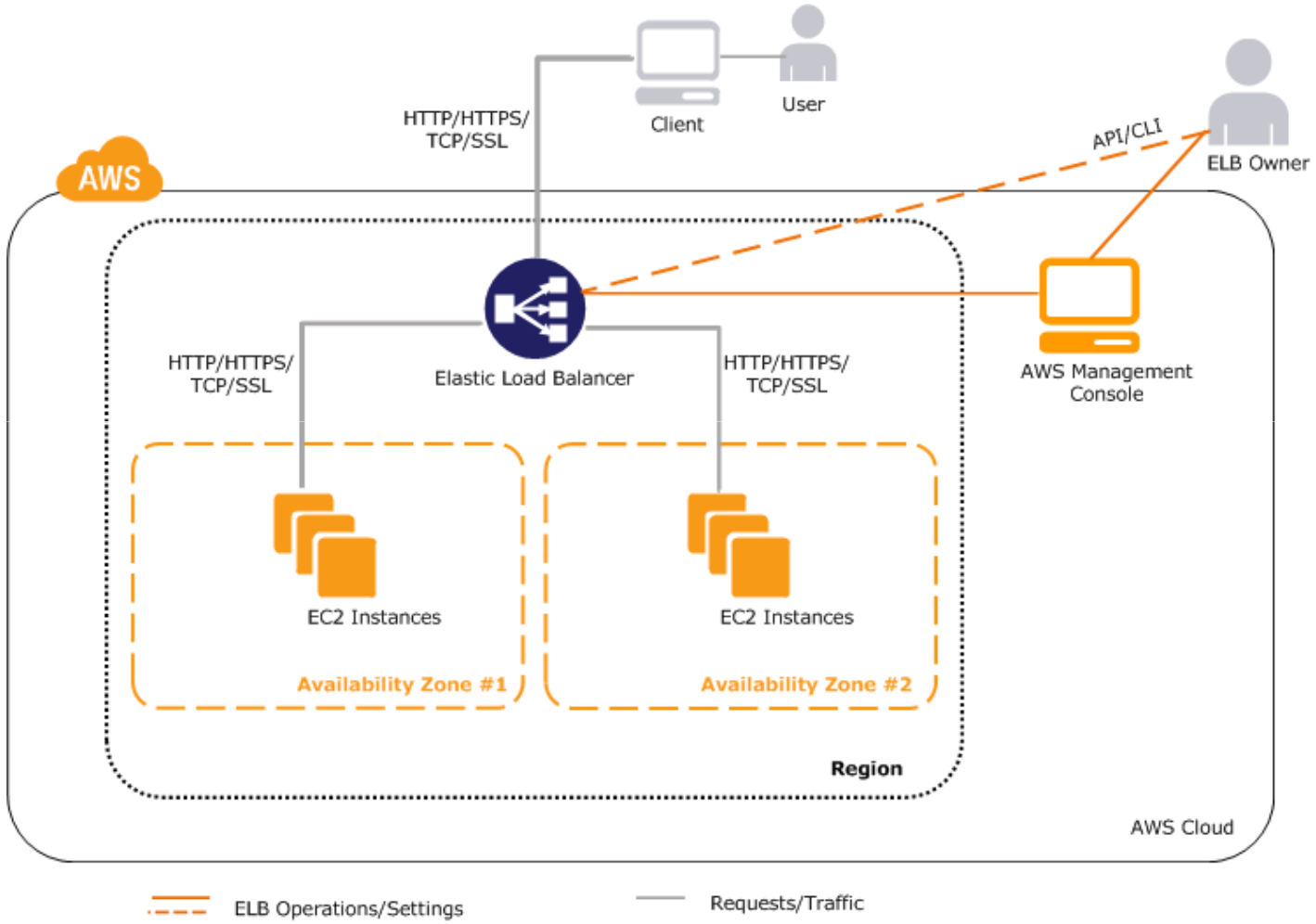
Focus of this tutorial

- Provide basic familiarity with additional AWS Services
  - Elastic Load Balancing
    - Network load balancers for groups of instances
  - Auto Scaling Groups
    - Elastically scaling groups of instances
  - Cloud Watch
    - Instance monitoring and alarms

Elastic Load Balancing (ELB)

- Distributes incoming network traffic to a group of EC2 instances (same region, different availability zones allowed)
  - Forwarded protocols are configurable: HTTP(S), SSL or plain TCP
  - HTTP(S) and SSL offer „sticky“ sessions and encryption offloading
- Transparent to clients
- Performs health checks on instances and only routes traffic to healthy instances
- Benefits
  - Better scalability (avoids some bottlenecks)
  - Increased fault tolerance (instance failure)
  - SSL encryption offloading

Elastic Load Balancing: Architecture



Source: [1]

Elastic Load Balancing: Technical Details

- Load balancer registers DNS name
- Clients resolve DNS name to an ELB IP address
  - DNS server may return different IP addresses pointing to different ELB entry points
- Client connects to IP address at application-specific port (e.g. TCP/80 for web servers)
- ELB handles incoming connection
  - Chooses healthy EC2 instance (with HTTP session stickyness)
  - Internally forwards traffic to port (e.g. TCP/80) of this instance
  - May do additional SSL encryption offloading

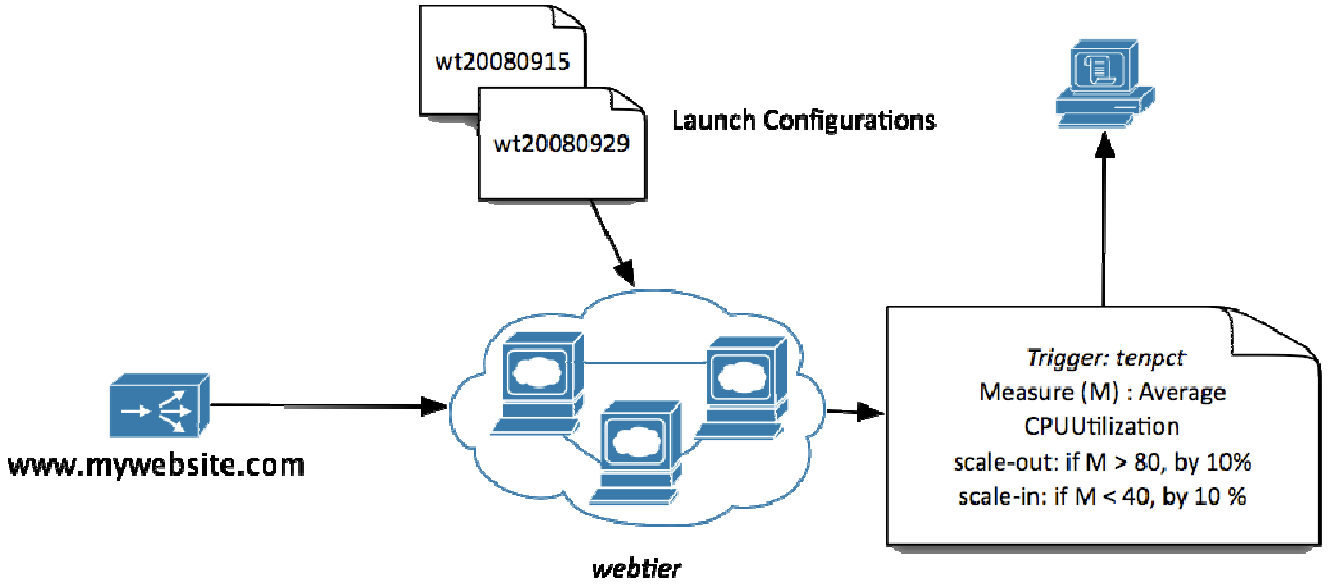
Elastic Load Balancing: Health Checks

- ELBs periodically check if EC2 instances are alive
  - Uses „Ping Protocol“ (!= ICMP Ping).
    - HTTP/HTTPS: make GET request to an URL
    - TCP: opens and closes a connection
  - Configuration parameters
    - Time between health checks
    - Check response timeout
    - Thresholds for (un)healthiness

Auto Scaling (AS) Groups

- Service that automatically launches/terminates EC2 instances
  - Launch configurations
    - Describe how to launch new instances (AMI, instance type, etc…)
  - Policies
    - Describe scaling actions such as or „add 10% more instances“ or „remove one instance“
  - Groups
    - Connects launch configs, policies and elastic load balancers
    - Has min, max and desired size
    - Does host level health checks so that min size is guaranteed

Auto Scaling Groups



Source: [2]

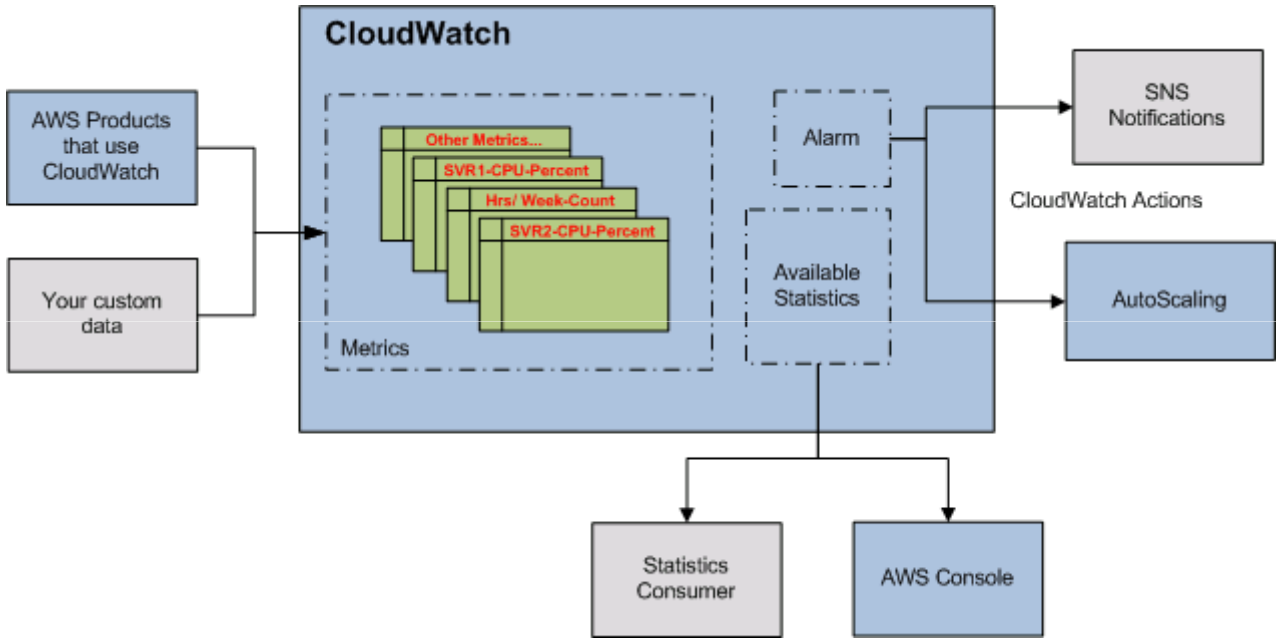
Auto Scaling Groups

- Types of scaling
  - Manual e.g. using command-line tools
  - Scheduled („add three instances at 18.05.2012 12:00 and shut them down at 18:00“)
  - Using Policies and Cloud Watch Alarms
    - Cloud Watch alarms send notifications to policies under specific conditions, such as „avg. CPU utilization >= 90%“
    - Notifications are sent to configurable ARNs (ARNs identify Amazon AWS resources, such as policies)

Cloud Watch (CW)

- Provides **monitoring services** for instances and applications
- Metrics
  - Describe performance aspects of instances/applications
  - Time series data produced either by AWS monitoring or your own services
  - Default: 5min resolution / detailed: 1min resolution
  - Defined by namespace, dimensions and (optional) unit of measure
- Alarms
  - Define actions to be taken given certain conditions

Cloud Watch (CW)



Source: [3]

Cloud Watch Metrics

- AWS provides 7 default metrics for every instance (measured on hosts, not inside instances):
  - Avg CPU utilization
  - 4 disk related metrics (data read/written, read/write operations)
  - 2 network related metrics (network in/out)
  - Default resolution 5min (1min resolution is available, default for auto scaling group instances)
- Custom metrics can be published by user applications

Cloud Watch

- Alarm
  - Watches a single metric over number of time periods
  - Triggers action when …
    - a statistic (min/max/avg/…)
    - over samples (e.g. one sample every minute for 5 minutes)
    - of a metric (e.g. CPU utilization)
    - is < = > than a threshold (e.g. > 90%)
  - Has a state (OK, ALARM, INSUFFICIENT_DATA)
  - Actions: SNS Notification or Auto Scaling Policy

References

- **[1]:** http://docs.amazonwebservices.com/ElasticLoadBalancing/latest/DeveloperGuide/SvcIntro_arch_workflow.html
- **[2]:** http://docs.amazonwebservices.com/AutoScaling/latest/DeveloperGuide/images/AutoscalingDesign2.png
- **[3]:** http://docs.amazonwebservices.com/AmazonCloudWatch/latest/DeveloperGuide/images/CW-Overview.png