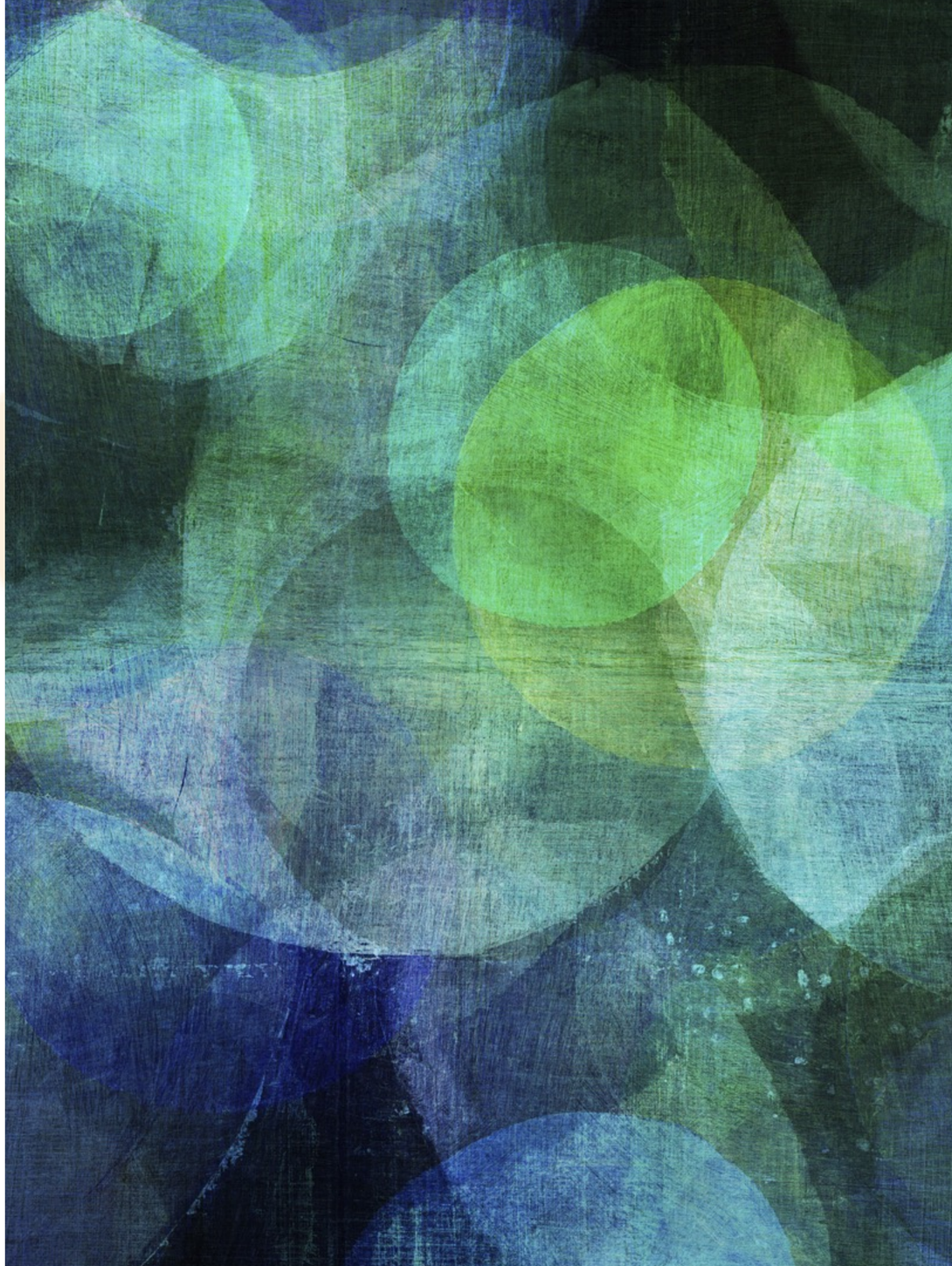
AWS DATA PROCESSING INFRASTRUCTURE 3A

Nan Dun
nan.dun@acm.org



COPYRIGHT POLICY 版权声明

All content included on the Site or third-party platforms as part of the class, such as text, graphics, logos, button icons, images, audio clips, video clips, live streams, digital downloads, data compilations, and software, is the property of BitTiger or its content suppliers and protected by copyright laws.

Any attempt to redistribute or resell BitTiger content will result in the appropriate legal action being taken.

We thank you in advance for respecting our copyrighted content.

For more info see https://www.bittiger.io/termsofuse and https://www.bittiger.io/termsofservice

所有太阁官方网站以及在第三方平台课程中所产生的课程内容，如文本，图形，徽标，按钮图标，图像，音频剪辑，视频剪辑，直播流，数字下载，数据编辑和软件均属于太阁所有并受版权法保护。

对于任何尝试散播或转售BitTiger的所属资料的行为，太阁将采取适当的法律行动。

我们非常感谢您尊重我们的版权内容。

有关详情，请参阅
https://www.bittiger.io/termsofuse
https://www.bittiger.io/termsofservice

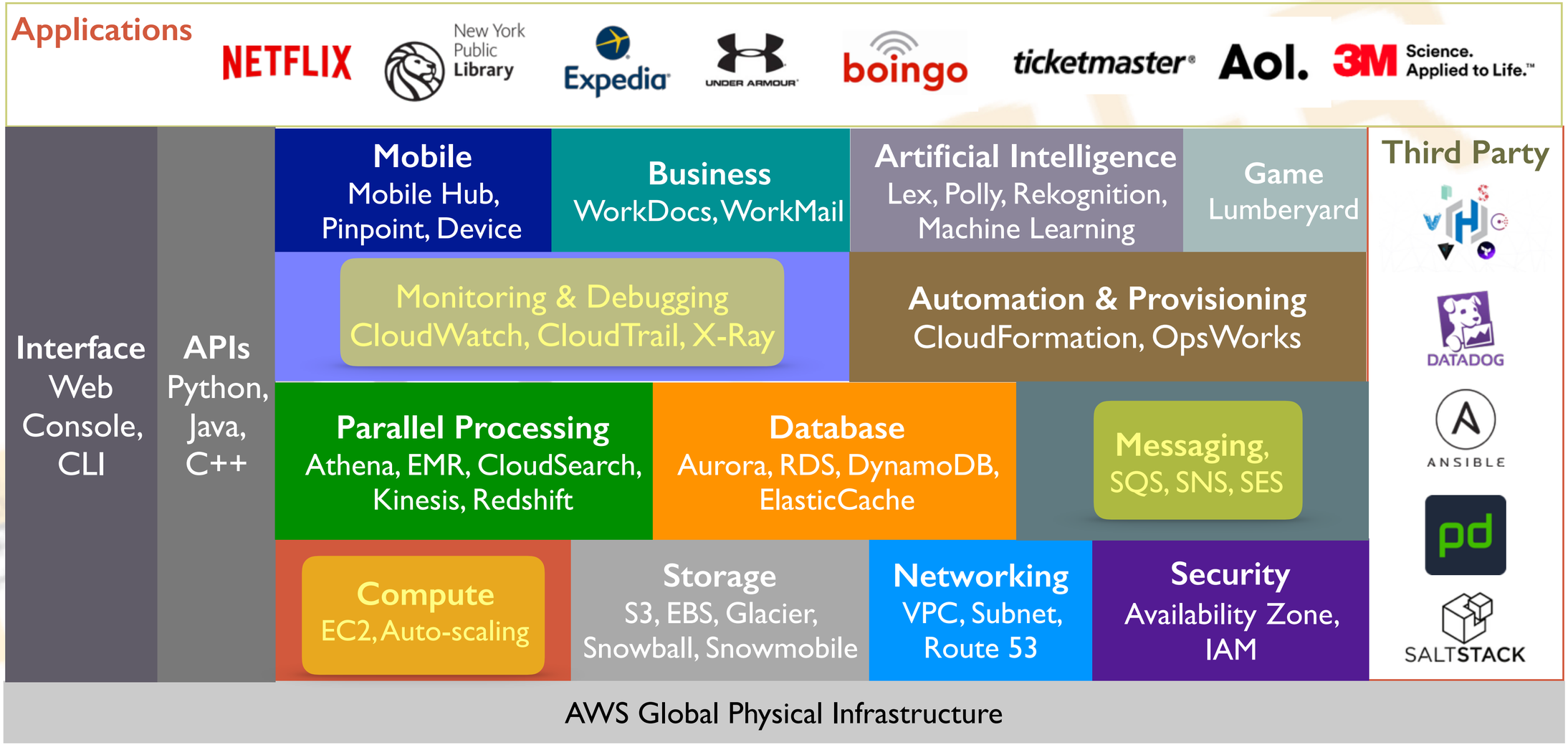
DISCLAIMER

I. "All data, information, and opinions expressed in this presentation is for informational purposes only. I do not guarantee the accuracy or reliability of the information provided herein. This is a personal presentation. The opinions expressed here represent my own and not those of my employer."

II. "The copyright of photos, icons, charts, trademarks presented here belong to their authors."

III. "I could be wrong."

TODAY'S TOPIC



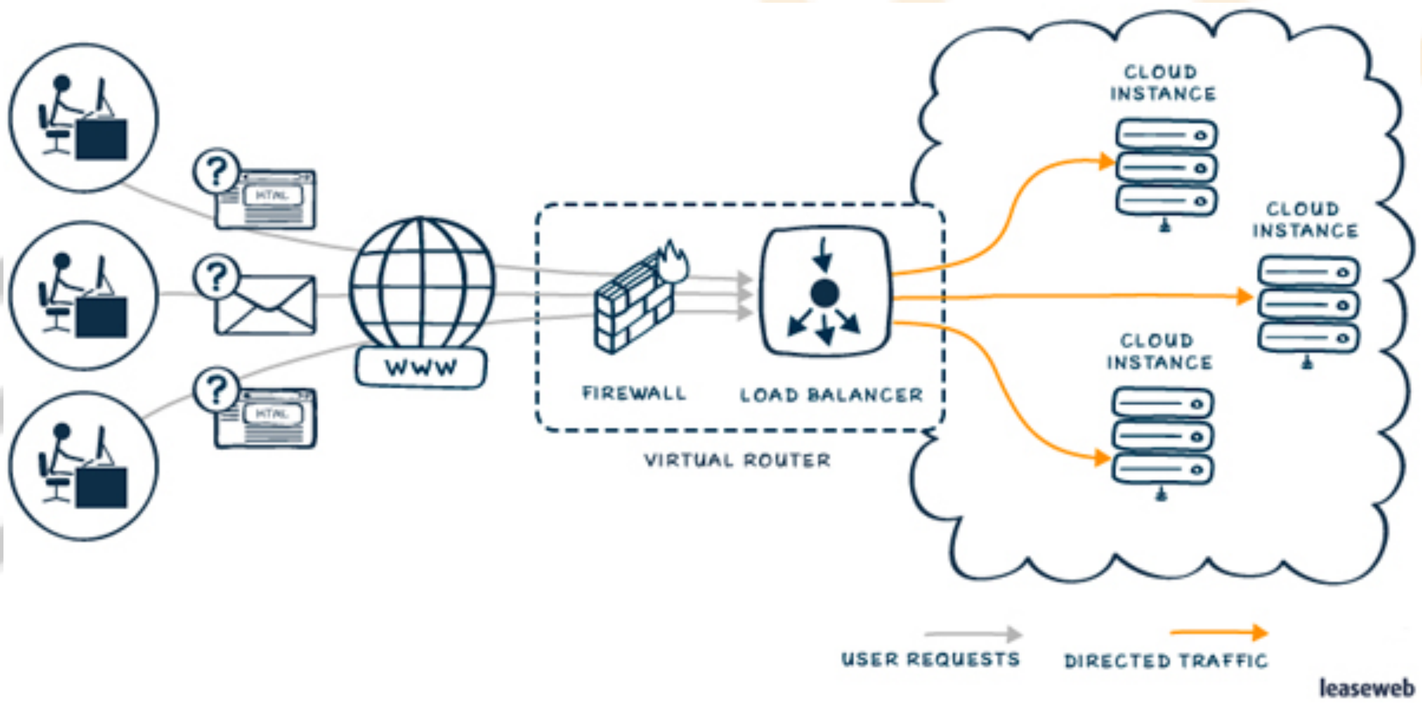
OUTLINE

- Elastic Load Balancer
- Auto-Scaling Group
- Spot Instances and Spotfleet
- CloudWatch
- CloudTrail

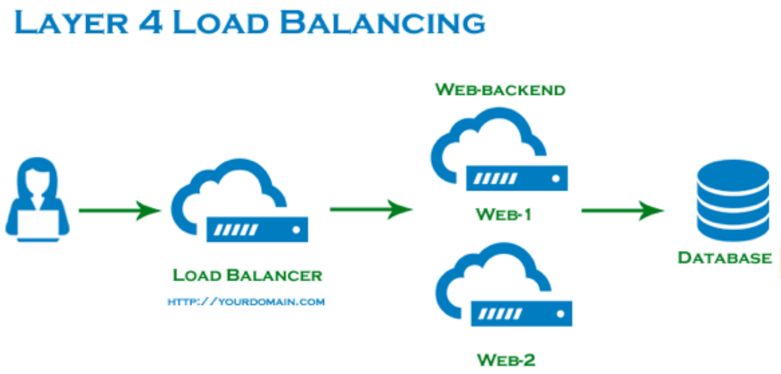
ELB

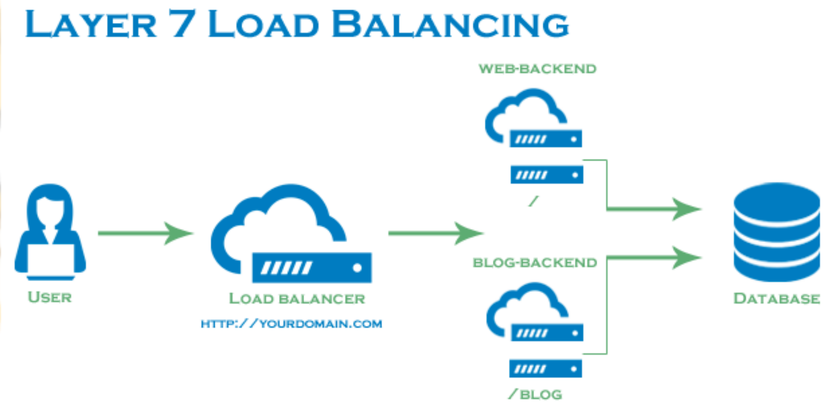
WHAT IS LOAD BALANCER?

- Automatically distribute incoming application traffic across multiple application, micro-services, and containers



LAYER 4 VS. LAYER 7 LOAD BALANCING





OSI (Open Source Interconnection) 7 Layer Model

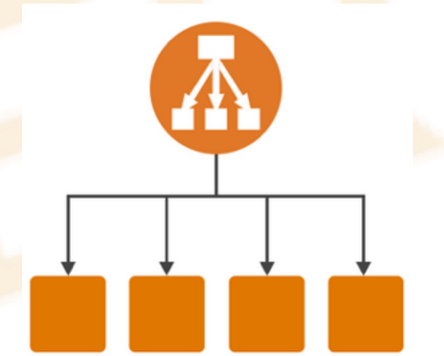
Layer	Application/Example		Central Device/ Protocols			DOD4 Model
Application (7) Serves as the window for users and application processes to access the network services.	**End User layer** Program that opens what was sent or creates what is to be sent. Resource sharing • Remote file access • Remote printer access • Directory services • Network management		**User Applications** SMTP		GATEWAY Can be used on all layers	Process
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	**Syntax layer** encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • **Character Set Translation**		JPEG/ASCII EBDIC/TIFF/GIF PICT			
Session (5) Allows session establishment between processes running on different stations.	**Synch & send to ports** (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.		**Logical Ports** RPC/SQL/NFS NetBIOS names			
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	**TCP** Host to Host, Flow Control. Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	PACKET FILTERING	TCP/SPX/UDP			Host to Host
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	**Packets** ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		**Routers** IP/IPX/ICMP			Internet
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	**Frames** ("envelopes", contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control		**Switch Bridge WAP** PPP/SLIP	Land Based Layers		Network
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	**Physical structure** Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts		**Hub**			

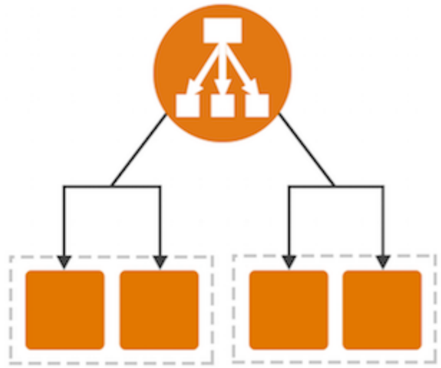
LAYER 4 VS. LAYER 7 (CONT.)

	Layer 4 (Network)	Layer 7 (Application)
Protocol	TCP and SSL	HTTP and HTTPS
Connection	THROUGH LB to Server	END at LB and Pooled
Header	NO MODIFICATION	MAY MODIFIED
Routing	ADDRESS (IP, TCP/UDP ports)	CONTENT (TEXT, DATA, VIDEO, etc.)
Computing Cost	MODERATE	HIGH

CLASSIC VS. APPLICATION LOAD BALANCER

	Classic	Application
Protocol	TCP, SSL, HTTP, HTTPS	HTTP, HTTPS
Platform	EC2-Classic, EC2-VPC	EC2-VPC
Heath Check	✓	Improved
CloudWatch	✓	Improved
Path-based Routing		✓
Container		✓
WebSocket & HTTP/2		✓

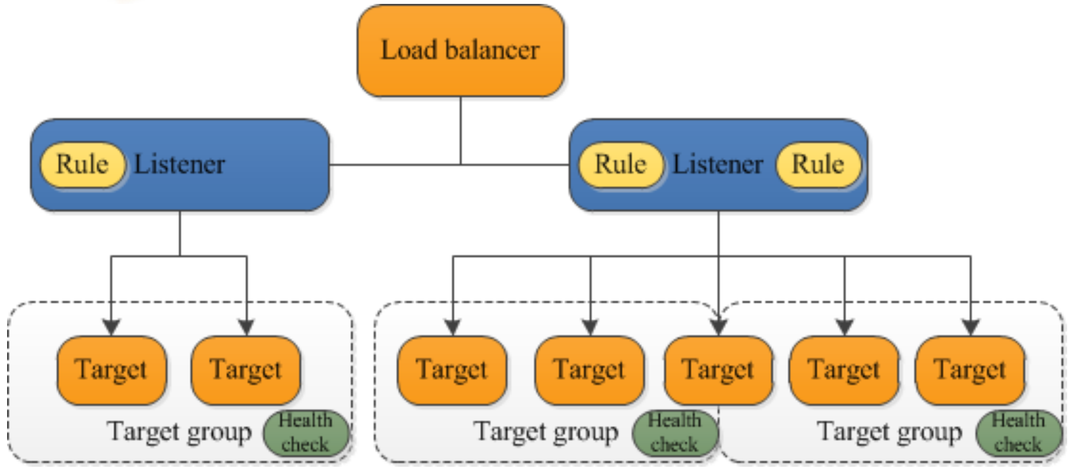




- $0.0225 per Application Load Balancer-hour (or partial hour)
- $0.008 per LCU-hour (or partial hour)

APPLICATION LOAD BALANCER

- Allows for multiple applications to be hosted behind one single load balancer
  - Instances can be registered to multiple ports, so requests can be routed to multiple containers one single instance



LISTENER (WHO)

- Define the protocol and port of incoming connections
- At least one listener to accept incoming traffic, up to 10
- Routing tables are defined in listeners

TARGET GROUPS (WHICH)

- Logical group of targets
- Dynamically registered to load balancer
- Can register with auto-scaling group

RULES (WHEN)

- Up to 10 rules
- Path pattern only now
  - Based on request path

DEMO

- Two web servers registered to one load balancer
  - webserver0.bittiger.info/webserver0/
  - webserver0.bittiger.info/webserver1/
  - web.bittiger.info ==> web-elb.amazonaws.com
    if webserver0: ==> webserver0.bittiger.info/webserver0/
    if webserver1: ==> webserver1.bittiger.info/webserver1/

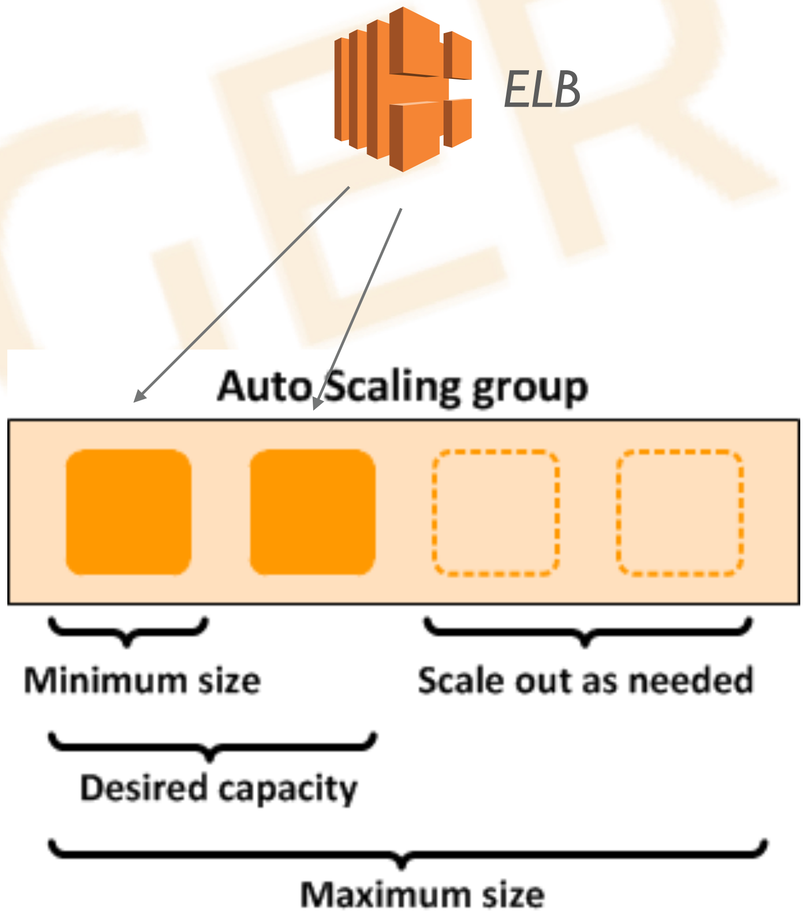
AUTO SCALING

AUTO SCALING

- Concepts
  - Auto Scaling Groups
  - Launch Configuration
  - Scaling Plan
- Benefits
  - Automatically adapt do demand
  - Availability and reliability
  - Auto-rolling customization

AUTO SCALING GROUP

- A group of EC2 instance that can be automatically launched or terminated based on demand, metrics, etc.
  - Minimum capacity: guaranteed # of instances running
  - Desired capacity: generally required # of instances running
  - Maximum capacity: upper limit of # of running instances
- Auto-balanced across AZs
- Price as on-demand instance



LAUNCH CONFIGURATION

- Specification of instances to be launched (what)
  - EC2 instance type, size
  - AMI
  - Security groups, SSH key, IAM instance profile
  - User data
    - A bootstrap to setup instance right after it launched?
      - Can install software, copy files, etc.
    - Why not Ansible?
      - No IP at boot time
      - Ansible don’t know when instances launched

USER DATA

- A bootstrap script will be executed during boot time
  - A shell script
- Now we have
  - AMI: static
  - User-data: static, load dynamically
  - Ansible: dynamic

SCALING PLANS

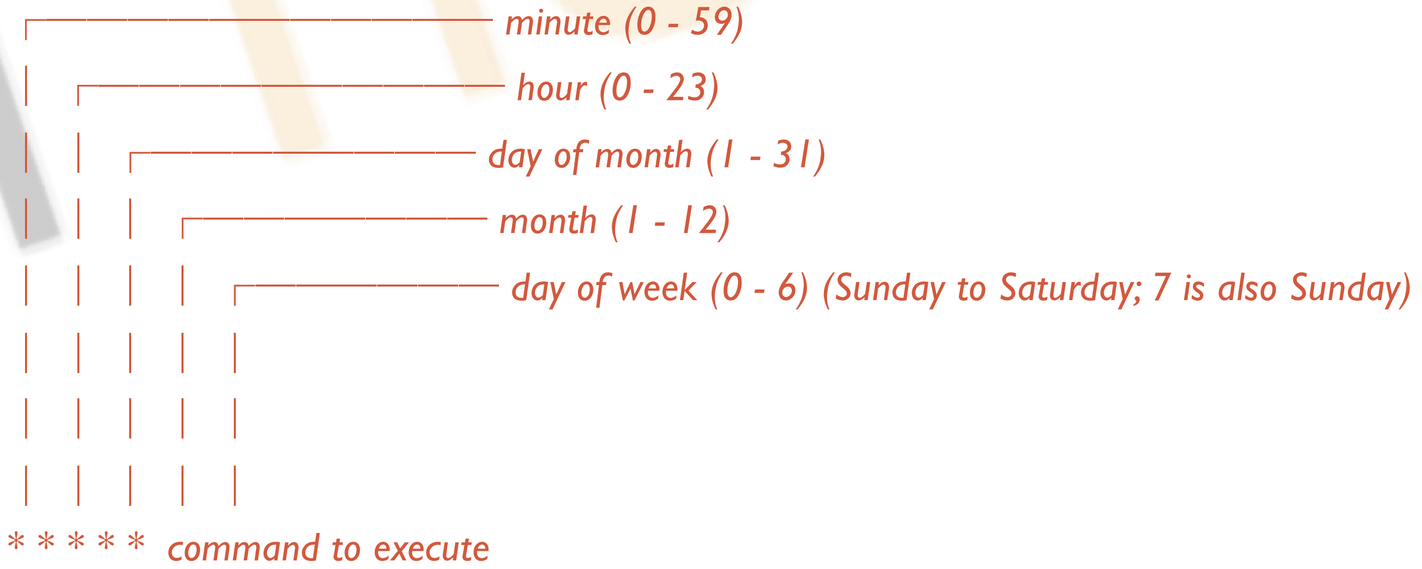
- Used to determine when to scale in/out
  - Out (launch instance): desired capacity > current capacity
  - In (terminate instance): desired capacity < current capacity
- Plans types
  - Default: ensure current capacity of *healthy* instances
  - Manual: modify desired capacity by console, CLI, APIs.
  - Scheduled: on a pre-defined time
  - Dynamic scaling: based on runtime metrics and instance health, e.g., CloudWatch
- Policies
  - Scaling and terminate policies

SCALING POLICIES

- Fix capacity
  - Set desired capacity to a number
- Increment / Decrement by an amount
  - Desired capacity +/- delta
- Increment / Decrement by a ration
  - Desired capacity +/- %

SCHEDULED SCALING

- Recurring event
  - cron syntax
- Individual event
  - Up to 135 event per group



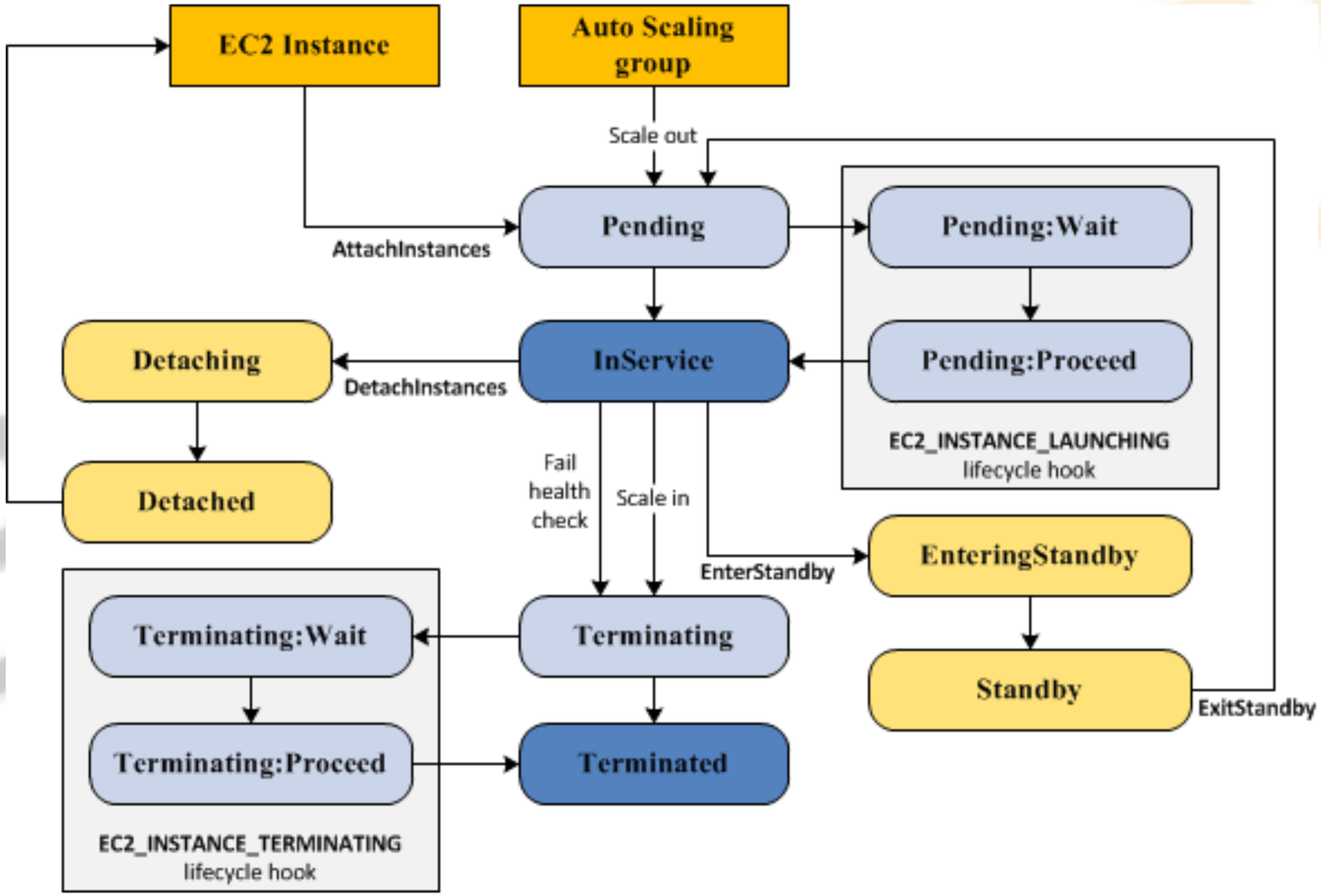
DYNAMIC SCALING POLICIES

- Triggered by CloudWatch metrics
  - CPU, bandwidth, S3 traffic, cost, …
- Metrics to policies mapping
  - High CPU utilization —> launch more
  - High traffic —> launch more instance with higher bandwidth
  - High cost —> terminate some instances
- Step scaling
  - Add 2 instances when 30 < CPUUtilization < 50
  - Add 4 instances when 50 <= CPUUtilization <= 70
  - Add 8 instances when 70 < CPUUtilization

TERMINATE POLICIES

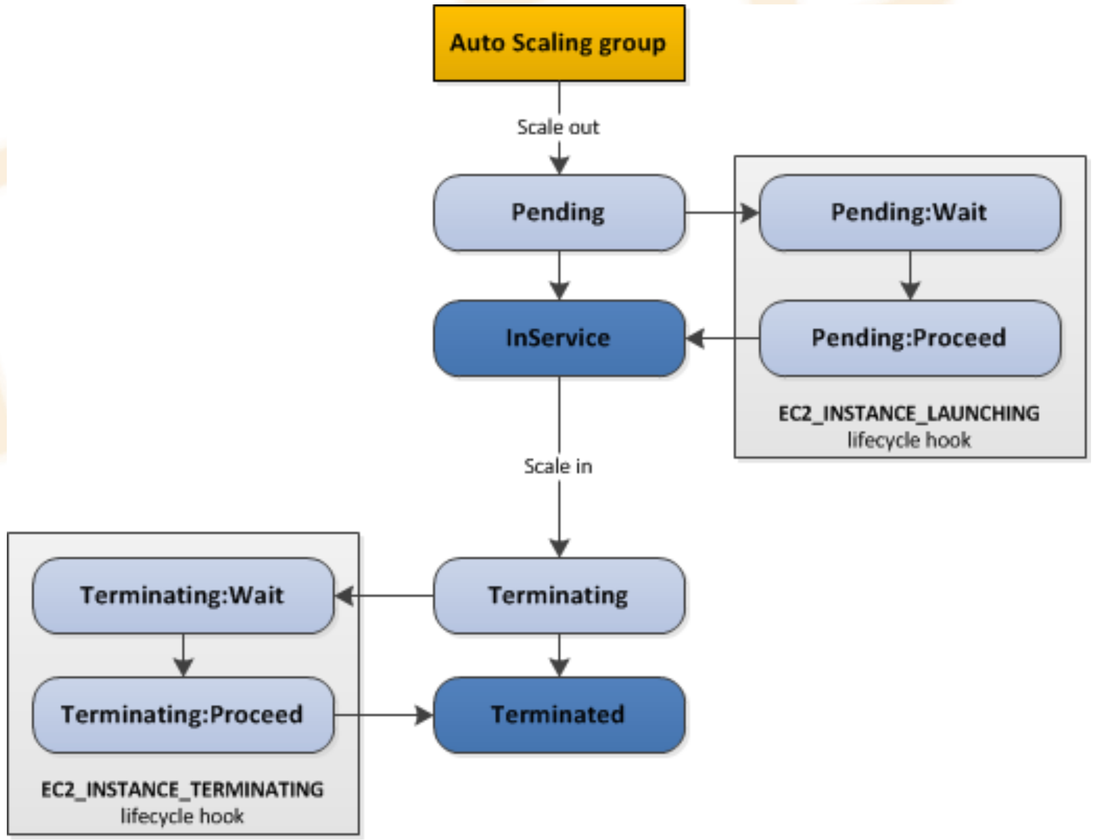
- Determine which instances are terminated first
  - Longest running
  - Oldest launch configuration
  - Closest to full filling hour
- Cross AZs migration may take place

INSTANCE LIFECYCLE



LIFECYCLE HOOKS (CALLBACKS)

- Do something before
  - Assigne Elastic IP
  - Register new instances with DNS
  - Register with SQS
- Do something after
  - Download data
  - Keep logs
  - Notify users via SNS



“

Whenever you find yourself on the side of majority, it is time to pause and reflect.

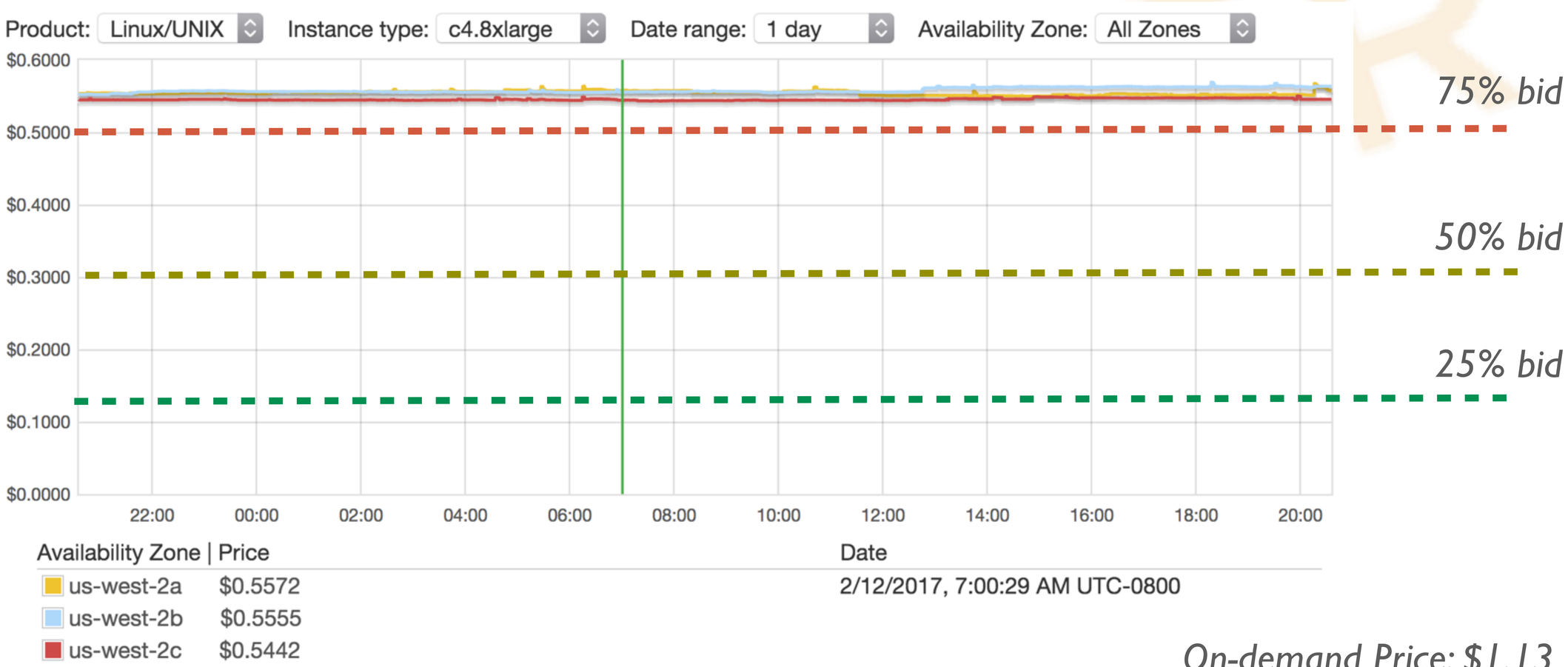
-Mark Twain

SPOT INSTANCES

SPOT INSTANCE

- “Idle” EC2 resources offered with a discount price based on market supply-demand
- Life cycle
  - Bid price > current spot price, your instances are run.C
  - Current spot price > bid price, instances are reclaimed by AWS and given to others
- Two-minute warning before reclaim

SPOT INSTANCES



BIDDING STRATEGY

- Bid what you are willing to pay, (up to 10x on-demand price)
  - Usually pay as on-demand price should be good enough
- Pay what market price is
- Pay per hour, do not pay if interrupted
  - Use full hour

BEST PRACTICE

- Diverse into various instance types, AZ, regions
- Reduce boot time
  - Bake everything in AMI
- Small jobs instead of large jobs
- Frequent stage in and stage out
- Hot swap

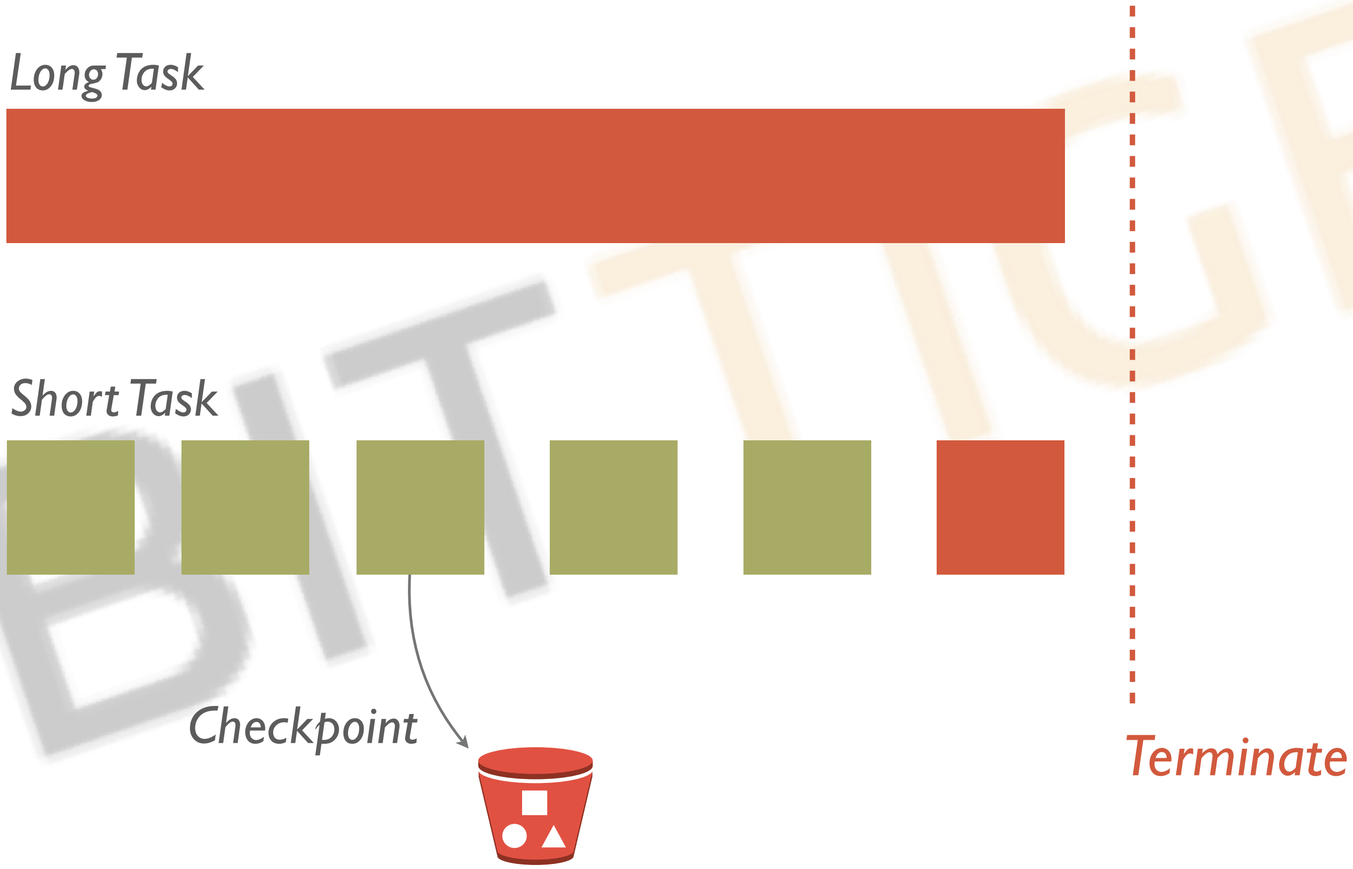
DIVERSIFY

- Distribute instances across AZs, even regions
- Distribute instances among different instance types, using spot fleet
- Challenges
  - Deployment across AZ, regions
  - Configurations according to different instance types
  - Cross-region performance: bandwidth, latency, etc.
  - Cross-region security, VPC, etc.

REDUCE BOOTSTRAP/SETUP TIME < 2MINS

- Do not use remote Ansible in this case
- Instances should boot and setup itself using user data
- Or bake everything into AMIs
- Challenges
  - Using git to distribute package might become a bottleneck, and cannot distribute large data
  - You may have to setup instances according to there roles
  - Or you will have many AMIs….

SMALL TASKS



HOT SWAP

- Monitoring termination status
  - Check status: http://169.254.169.254/latest/meta-data/spot/termination-time
  - If # of spot instances drop to some value, launch auto-scaling group
  - When # of spot instances get back to normal, scale down auto-scaling group
- Challenges
  - Coordinate resources is not as easy as you think
  - Migrate and resume tasks interrupted (even harder if you jobs has state, or even global state!)

SPOTFLEET

- A auto-scaling group of spot instances
  - Including different instance types
  - Including different AZs
- Demo

CLOUDWATCH

MONITORING IS IMPORTANT

- Why
  - Service Quality
  - Performance and cost
  - Trends
  - Troubleshooting and improvement
- Coverage
  - All AWS services

SAMPLE METRICS

- EC2
  - CPUUtilization
  - DiskReadBytes
  - DiskReadOps
  - DiskWriteBytes
  - DiskWriteOps
  - NetoworkIn
  - NetworkOut
- EBS
  - VolumeReadBytes
  - VolumeWriteBytes
  - VolumeReadOps
  - VolumeWriteOps
  - VolumeTotalReadTime
  - VolumeTotalWriteTime

METIRCS

- Concepts
  - Namespaces
  - Dimensions
  - Statistics
  - Percentile
- Source
  - Trusted Advisor
  - Graph
  - CLI/APIs

LIST METRICS

- aws cloudwatch list-metrics

  [--namespace <value>]

  [--metric-name <value>]

  [--dimensions <value>]

  [--cli-input-json <value>]

  [--starting-token <value>]

  [--max-items <value>]

  [--generate-cli-skeleton <value>]

LOGS

- Monitor and Alert
- Centralized access via S3
- Install awslogs agent in EC2 instances
  - Demo

CLOUDTRAIL

WHAT IS CLOUDTRAIL?

- A service that records AWS API calls for your account and delivers log files to you
  - A monitoring system only for API calls
- CloudTrail answers
  - What was the API call?
  - Who made the API call?
  - When was the API call made?
  - Which resources was API call acted on?
  - Where was the API call made, from where to where?

CLOUDTRAIL EXAMPLE

- EC2
  - Find out which user is denied to access in Lesson 2


```
{
    "Records": [{
        "eventVersion": "1.0",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "accountId": "123456789012",
            "userName": "Alice"
        },
        "eventTime": "2014-03-06T21:22:54Z",
        "eventSource": "ec2.amazonaws.com",
        "eventName": "StartInstances",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "205.251.233.176",
        "userAgent": "ec2-api-tools 1.6.12.2",
        "requestParameters": {
            "instancesSet": {
                "items": [{
                    "instanceId": "i-ebeaf9e2"
                }]
            }
        },
        ... additional entries ...
        ]
}
```

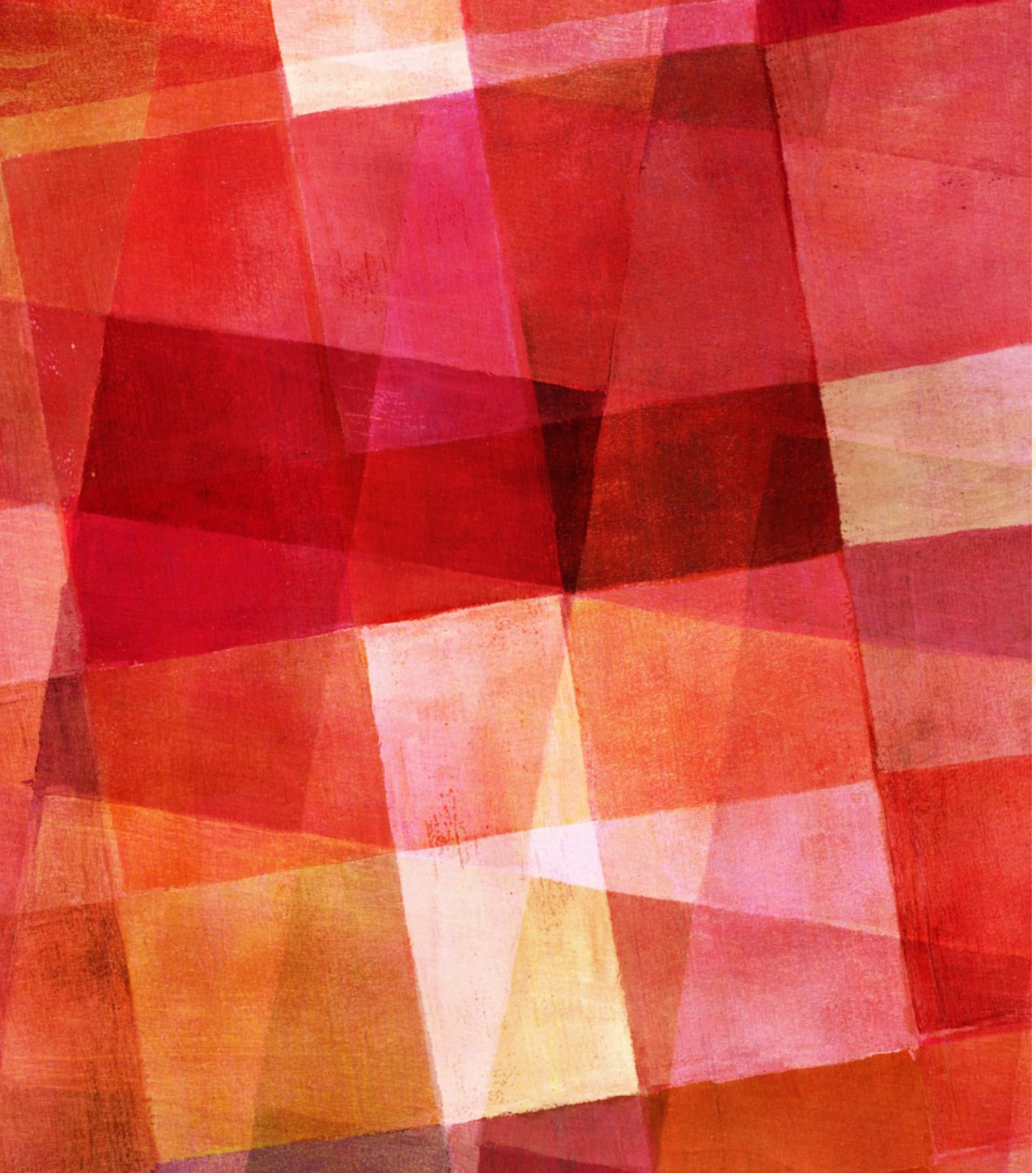
BUCKET POLICY FOR CLOUDTRAIL

```
{
    "Sid": "AWSCloudTrailAclCheck20150319",
    "Effect": "Allow",
    "Principal": {
        "Service": "cloudtrail.amazonaws.com"
    },
    "Action": "s3:GetBucketAcl",
    "Resource": "arn:aws:s3:::bucket"
},
```

```
{
    "Sid": "AWSCloudTrailWrite20150319",
    "Effect": "Allow",
    "Principal": {
        "Service": "cloudtrail.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::dun/AWSLogs/012345678901/*",
    "Condition": {
        "StringEquals": {
            "s3:x-amz-acl": "bucket-owner-full-control"
        }
    }
}
```

HOMEWORK

- Using console to manually launch ELB for two web servers
- Using console to create launch configuration and auto-scaling group with at least two instances, and terminated one of them, see what happens
- Using console to launch spot fleet across AZs, at least 4, kill one or two and see what happens
- Explore CloudWatch
- Create CloudTrail Alarms
  - http://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html#cloudwatch-alarms-for-cloudtrail-s3-bucket-activity



QUESTIONS

- bittiger-aws@googlegroups.com

BITTIGER