Unit 3
Platform as a Service

Hosting a Website in EC2

TAMAL DEY
DEPT OF CA

Method 1

Configure default apache server

Steps to follow to connect and configure:

1. Create a Sample web page
2. Create a ubuntu instance with ssh and http protocols (choose anywhere). If not available after launching the instance go and edit in the security groups of that instance.
3. Connect your instance through putty or ssh

```
chmod 400 My_Keypair.pem
```

```
ssh -i <your key a name>.pem ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com
```

```
scp -i My_Keypair.pem samplefile.txt ec2-user@ec2-184-72-204-112.compute-1.amazonaws.com:~
```

Copy files to a specific directory(/var/www/html/):

scp -i "WebSec.pem" hello.txt ubuntu@10.0.2.82:~/.

Optional

1. Create a Elastic-IP and associate with your EC2 instance

Configure your Web Server

1. **Update the server**: sudo apt-get update
2. **Install Lamp in server :** sudo apt-get install lamp-server^

   Above command will install apache web server, php and Mysql
3. **Visit default page**: Now copy and paste the public DNS of your instance in the address bar of the browser and click enter (index.html)
4. **Uploading Web Pages to WWW Folder:** After the web server has been installed you have to upload your web pages or project to the /var/www/html/ of apache web server.
5. You can locate the folder using the following command

   cd /var/www/html
6. **Create your working directory: (change mode to 777)**

   chmod 777 –R /var/www/html/

7. Create password for mysql

sudo mysql_secure_installation

A series of prompts where you can make some changes to your MySQL installation's security options. Press Y and then ENTER to accept the defaults for all the subsequent questions .Press No for access **remote login**.
Enter and then confirm a secure password of your choice. (**8 characters**)

mysql –u root -p

Alternative approach (if unable to log-in)

1. sudo su
2. sudo mysql
3. mysql >SELECT user,authentication_string,plugin,host FROM mysql.user;

   root user does in fact authenticate using the **auth_socket** plugin. To configure the root account to authenticate with a password, run the following **ALTER USER**
4. mysql >ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';
5. FLUSH PRIVILEGES; **Crtl +D** to exit from mysql

8. sudo apt-get install phpmyadmin

   Choose **apache** and use **same password** used in mysql

9. sudo service apache2 restart

   Latest changes will take effect by server

10. In browser type instance_public_ip/phpmyadmin and login with password

Your server has been configured

Optional

In case of any file /directory **creation** / **access permission error** use:

sudo chmod 777 **filename** or **directory name**

Additional settings to enable error in browser

11. Enable **php error** in browser

sudo **vi** /etc/php/7.2/apache2/php.ini
Go to line number 479 and make display_errors=On
Go to line number 490 and make display_startup_errors=On

Escape key :wq (Save &Quit) or :!q (only Quit)

sudo service apache2 restart

Task to perform: Create / copy a sample files

Create

1 PHP file
1 HTML , 1 CSS, 1 JS file (Connect each other)
use vi editor

Copy webhoting zip and extract

Copy the webhosing directory in the following path

/var/www/html/webhosting (Hint: winscp / scp-ubuntu)

scp -i "WebSec.pem" hello.txt ec2-user@10.0.2.82:~/.

Execute **query.sql** file in phpMyAdmin under database name **demo**

Perform changes in config.php

In browser type PublicIP/webhosting/login.php

You are good to go.

Method 2

Install Xampp

Login as superuser

Sudo su –

Update Ubuntu OS Packages: **sudo apt-get update**

Stop Default Apache: **sudo /etc/init.d/apache stop**

Install Xampp

Download 64bit xampp

wget https://www.apachefriends.org/xampp-files/7.0.23/xampp-linux-x64-7.0.23-0-installer.run

Make Execute Installation: sudo chmod +x xampp-linux-x64-7.0.23-0-installer.run

Run Installation

sudo ./xampp-linux-x64-7.0.23-0-installer.run

Follow **XAMPP instructions**

Type y for the prompted questions.

sudo /opt/lampp/lampp start

Open your browser and access *http://IP-ADDRESS/*

Click to phpmyadmin, you will find this Access forbidden screen.

Edit XAMPP configurations

sudo vim /opt/lampp/etc/extra/httpd-xampp.conf

Type I to go to insert mode.
Go to <Directory> or <LocationMatch>
Insert the following into the file after <Directory > line before the line ErrorDocument 403
/error/XAMPP_FORBIDDEN.html.var

AllowOverride AuthConfig Limit
Order deny, allow
Allow for all
Require all granted

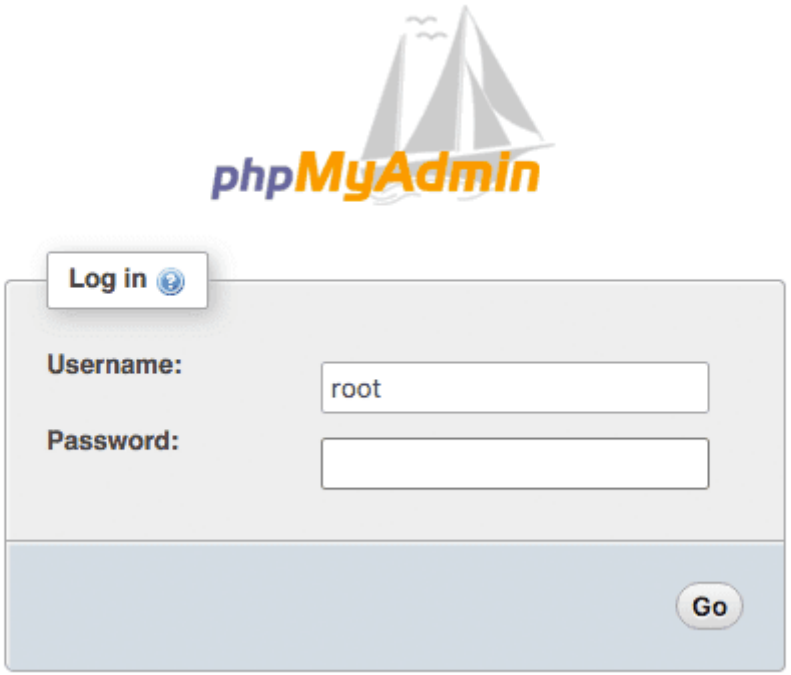
or

```
<LocationMatch
"^/(?i:(?:xampp|security|licenses|phpmyadmin|webalizer|serv
er-status|server-info))">
Order deny,allow
Allow from all
Allow from ::1 127.0.0.0/8 \
fc00::/7 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 \
fe80::/10 169.254.0.0/16
ErrorDocument 403 /error/XAMPP_FORBIDDEN.html.var
</LocationMatch>
```

Restart XAMPP: sudo /opt/lampp/lampp restart
Security Settings: sudo /opt/lampp/xampp security

XAMPP: Do you want to set a password? [yes] **no**
XAMPP: MySQL is accessable via network.
XAMPP: Normaly that's not recommended. Do you want me to turn it off? [yes] **yes**
XAMPP: Turned off.
XAMPP: The MySQL/phpMyAdmin user pma has no password set!!!
XAMPP: Do you want to set a password? [yes] **yes**
XAMPP: MySQL has no root passwort set!!!
XAMPP: Do you want to set a password? [yes] **yes**
XAMPP: The FTP password for user 'daemon' is still set to 'xampp'.
XAMPP: Do you want to change the password? [yes] **no**

Access PhyMyAdmin at *http://IP-Address/phpmyadmin/*



#sudo chmod -R 777 /opt/lampp/htdocs
#ls –ld /opt/lampp/htdocs
#exit [from su]
Whoami
sudo chown –R [Username]:[Groupname] /opt/lampp/htdocs
Create your own directories in htdocs
Move the web pages from your home machine to the ec2 through filezilla or winscp

References

LAMP

https://comtechies.com/how-to-host-a-dynamic-php-website-on-amazon-ec2-and-static-website-on-s3.html

XAMPP

https://www.9lessons.info/2015/12/amazon-ec2-setup-with-ubuntu-and-xampp.html

Application Link (Choose object-oriented)

https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php

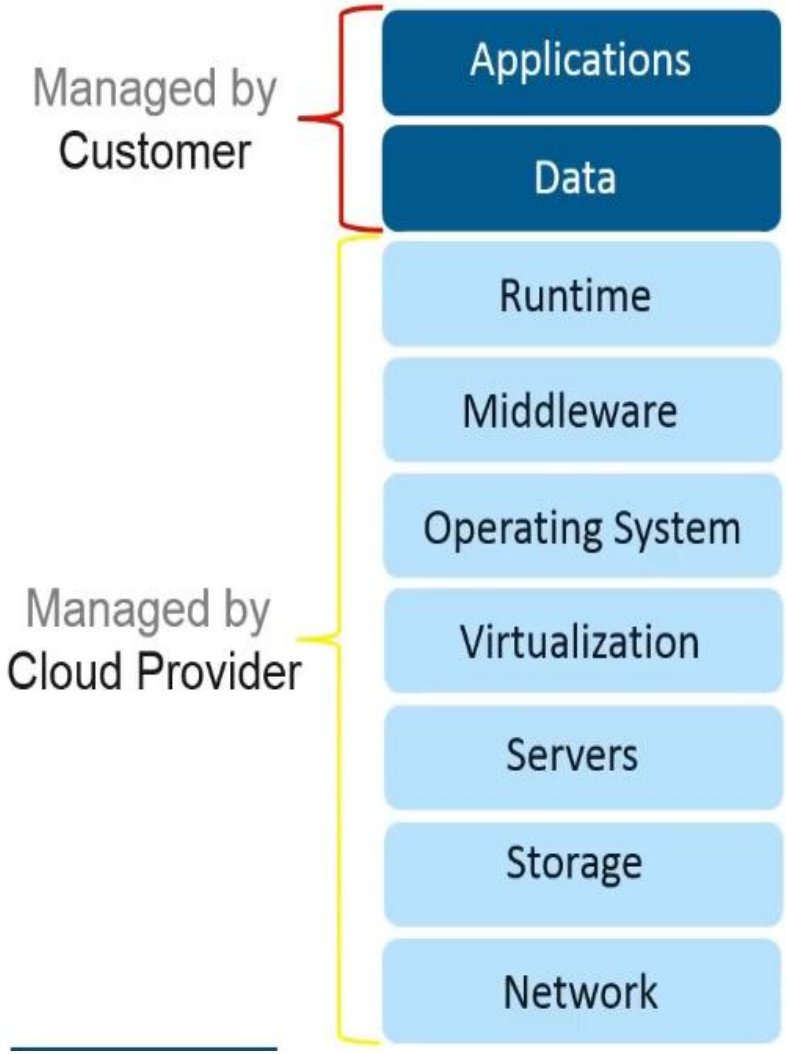
What is PaaS?

- Platform as a service (PaaS) is a cloud computing model in which a third-party provider delivers hardware and software tools , usually those needed for application development to users over the internet.
- A PaaS provider hosts the hardware and software on its own infrastructure.
- It is an abstracted and integrated cloud based computing environment that supports the development, running and management of applications
- PaaS is a category of cloud computing that provides a platform and environment to allow developers to build and deploy applications and services on the internet.
- An abstraction layer between your cloud application and your IaaS provider.

What is PaaS?

- PaaS services are hosted in the cloud and accessed by users simply via their web browser.
- Fundamentally provides elastic scaling of your application.
- You need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application.
- Platform as a Service (PaaS) is a way to rent hardware, operating systems, storage and network capacity over the Internet.
- The service delivery model allows the customer to rent virtualized servers and associated services for running existing applications or developing and testing new ones.

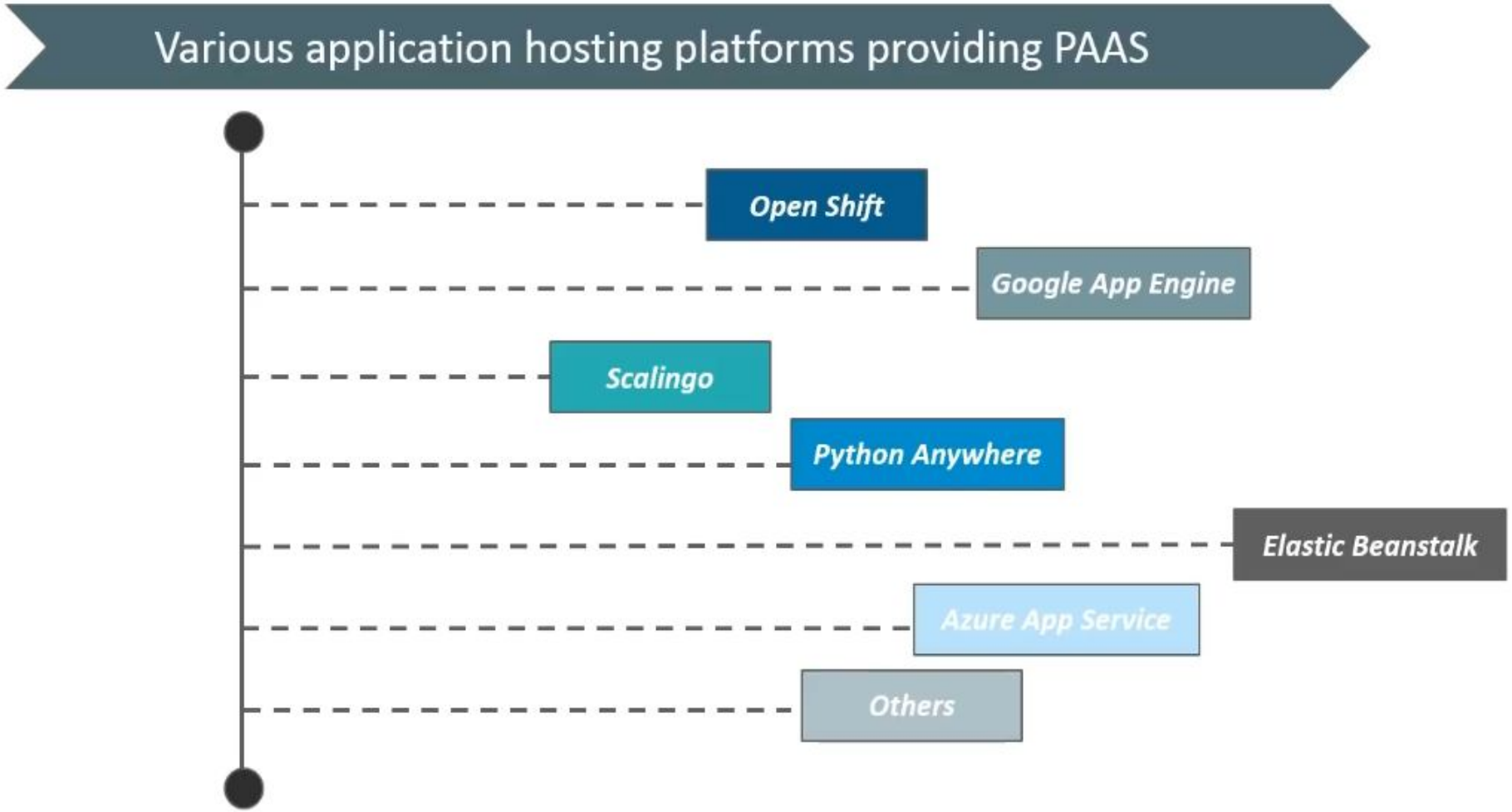
PAAS(Platform-As-A-Service)



Features for PaaS offering

- Operating system
- Server-side scripting environment
- Database management system
- Server Software
- Support
- Storage
- Network access
- Tools for design and development
- Hosting

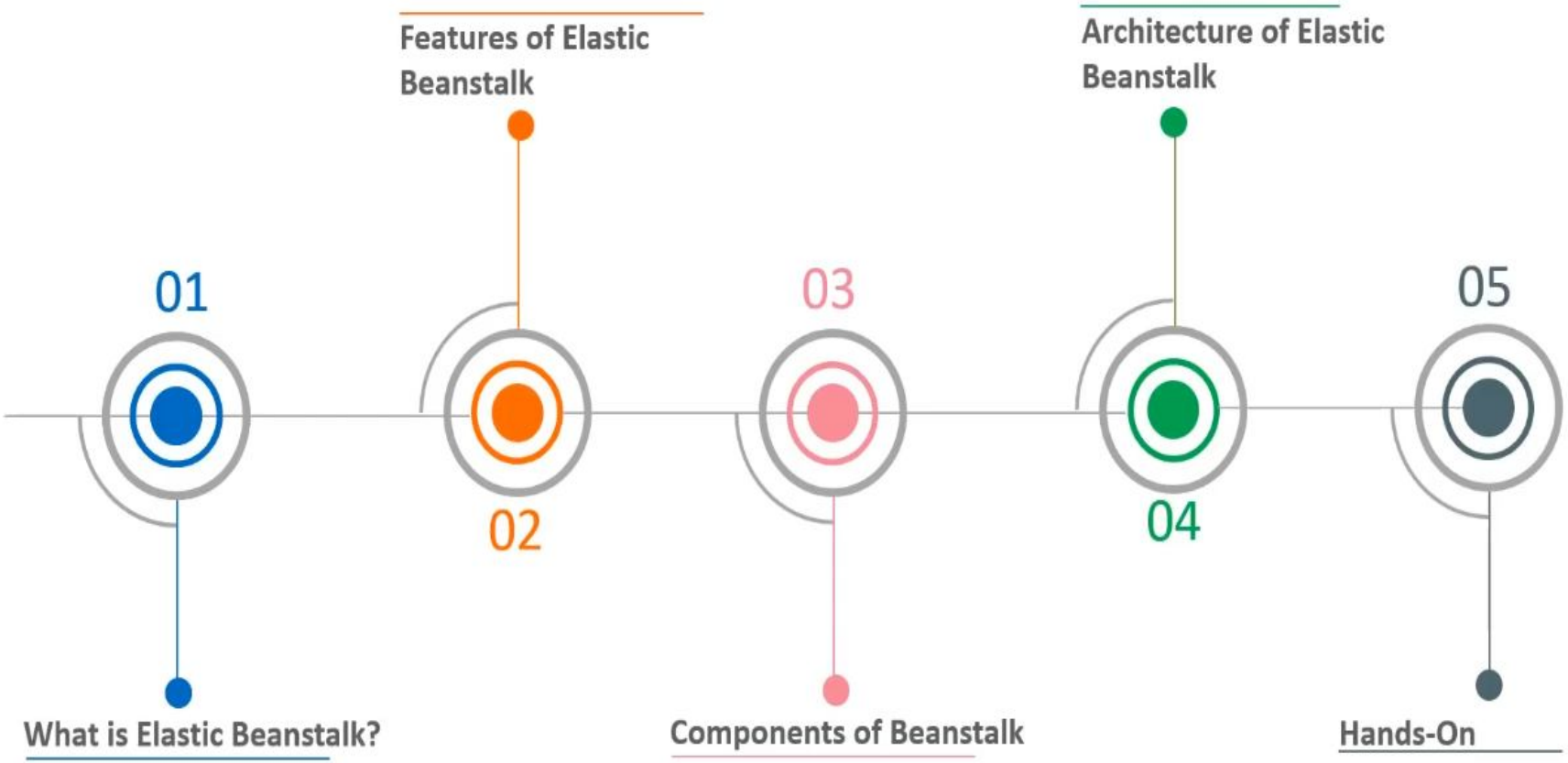
Web Hosting Platforms



Elastic Beanstalk

Tamal Dey
MCA,PESU

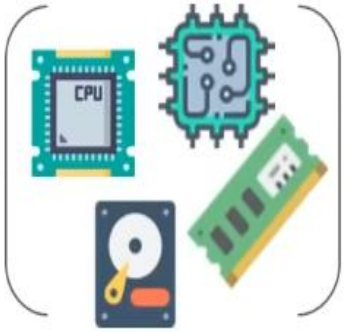
Outline



AWS Elastic Beanstalk is an PAAS service used for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js etc on familiar servers such as Apache, Nginx, Tomcat, and IIS.

Scenario: A Computer

Option A



You can go to a computer ware house and buy different components according to your requirement & assemble them.

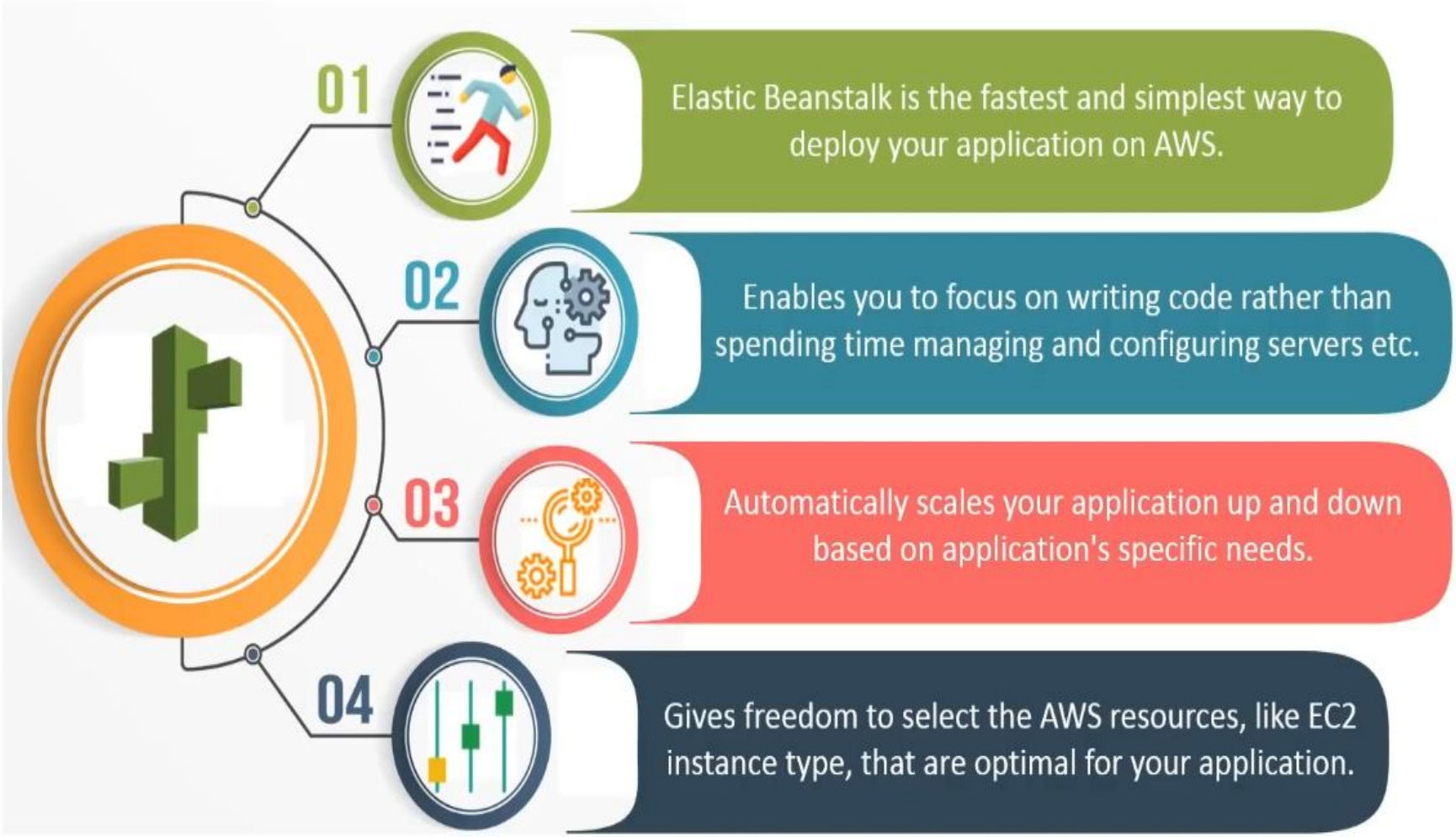
Deploying an application without using Elastic Beanstalk

Option B

You can visit a electronic retail store and buy a computer that fits your requirements.

Deploying an application using Elastic Beanstalk

Features of AWS Elastic Beanstalk



Beanstalk

AWS provides PaaS solution called Elastic Beanstalk. It is mature PaaS than other PaaS service provider.

Developers no need to worry about infrastructure to launch an web app. Developers just upload the application as binary file such as ***projectname.war***

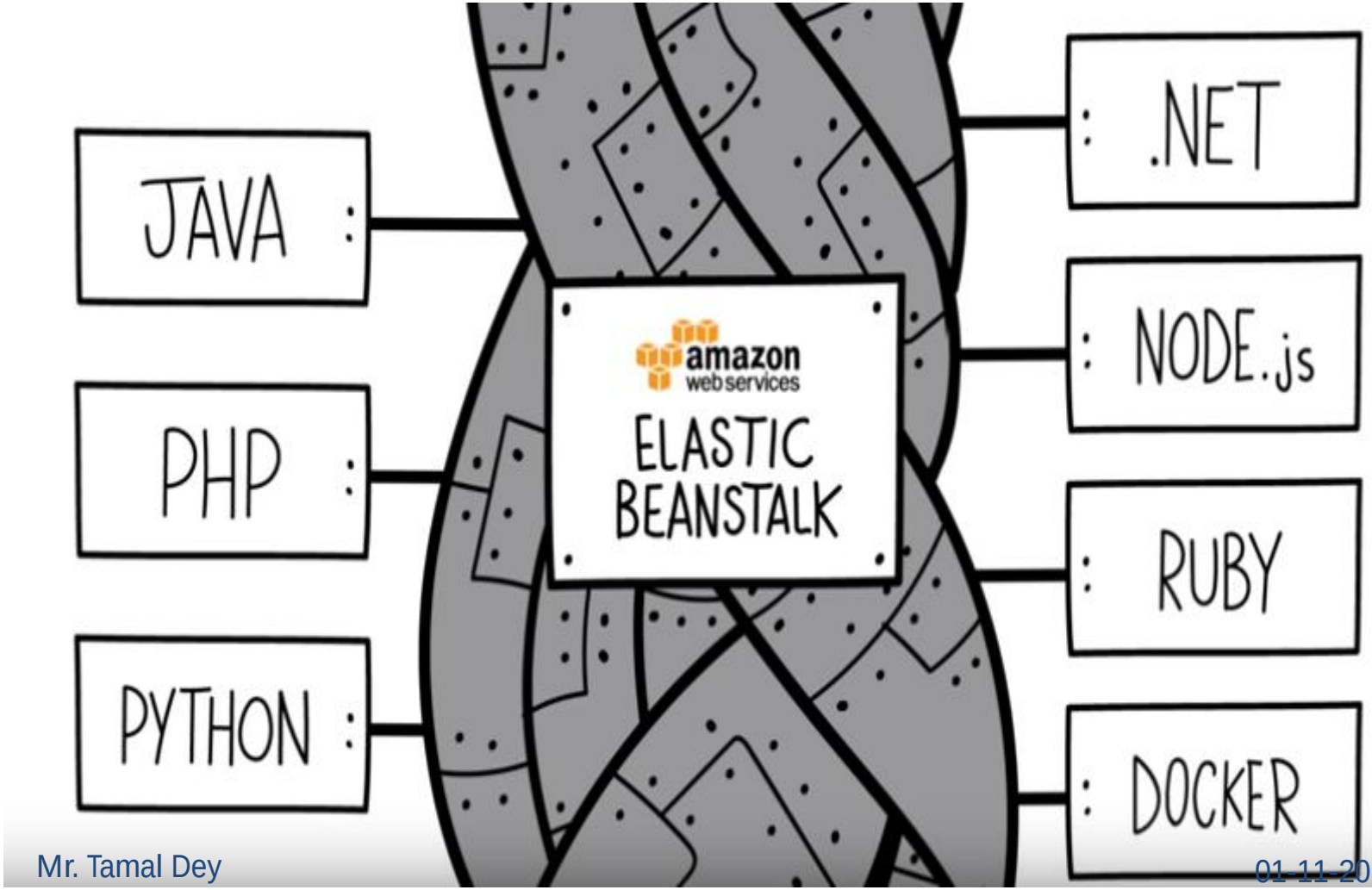
Developers can use application versioning to switch between previous version of application by using ***swap*** URL environment option.

It is possible to use an existing instance instead of provisioning a new environment. You can create AMI of your existing instance and use your AMI to create an instance.

Beanstalk

Whenever we want to host an application, we have configure system, manage servers, databases, scaling, load balancing etc.
Most of the time the programmers have to concentrate on these things
Beanstalk helps in reducing all these activities and helps to host/deploy your application very fast
Beanstalk is a service that deploys, manages and scales the web application for the users

Manage Applications



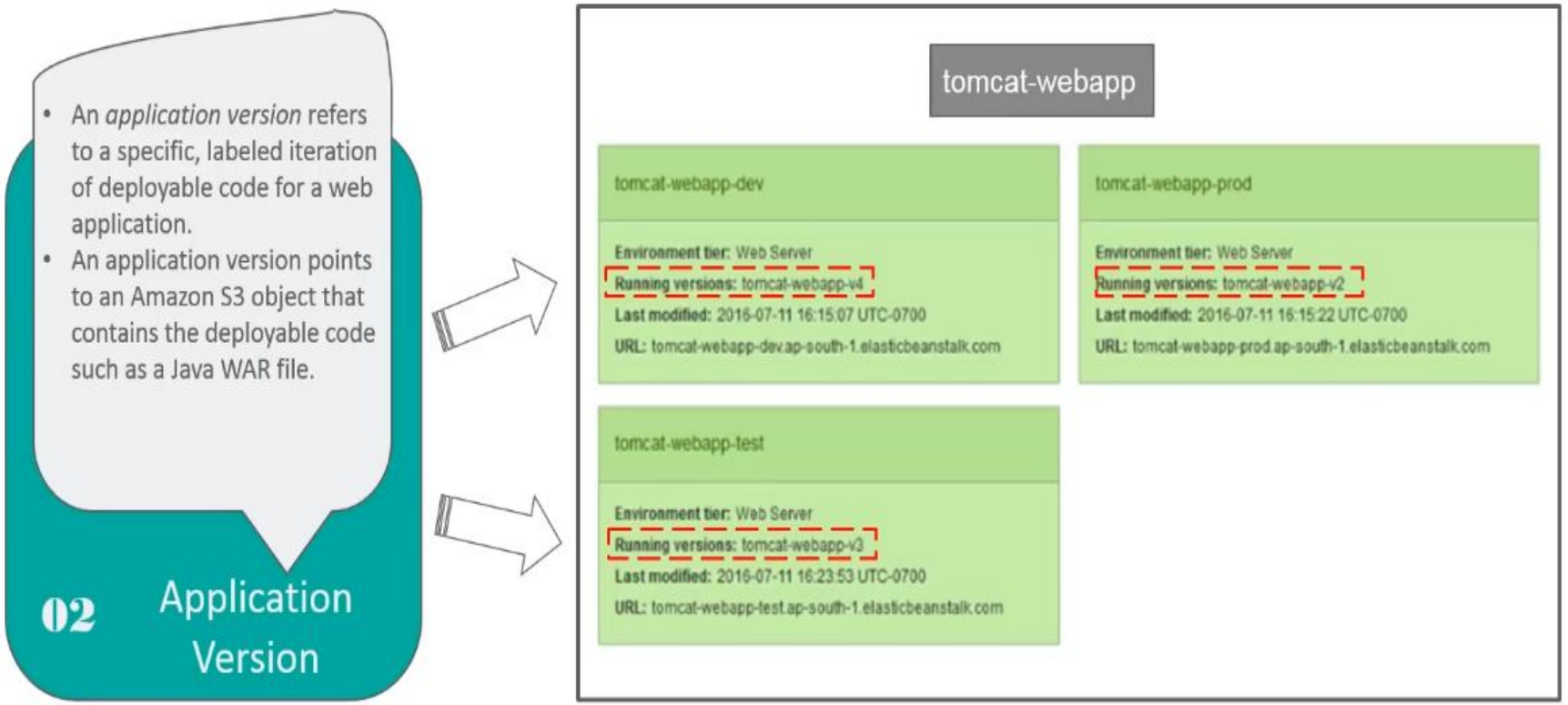
Manage Servers (Apache http, Apache tomcat, NginX, IIS)

Components Of Elastic Beanstalk

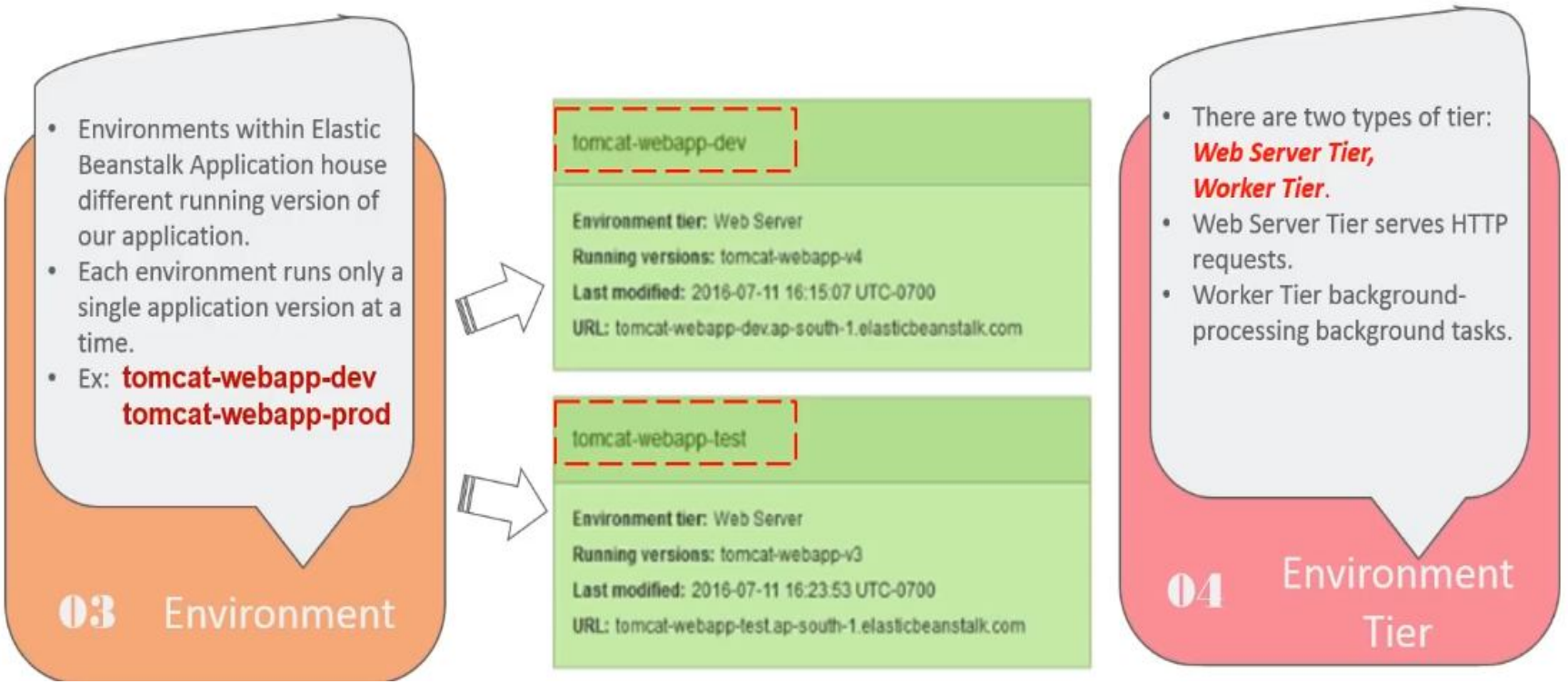
- An application is a collection of components including ***environments, versions*** and ***environment configuration.***
- An application in Elastic Beanstalk is conceptually similar to a folder.
- Ex : **tomcat-webapp**

01 Application

Components Of Elastic Beanstalk



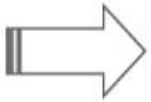
Components Of Elastic Beanstalk



Components Of Elastic Beanstalk

- Elastic Beanstalk reports the health of a web server environment depending on how the application running in it responds to the health check.
- Uses one of four colours to describe status.

05 Environment Health



Environment is being updated

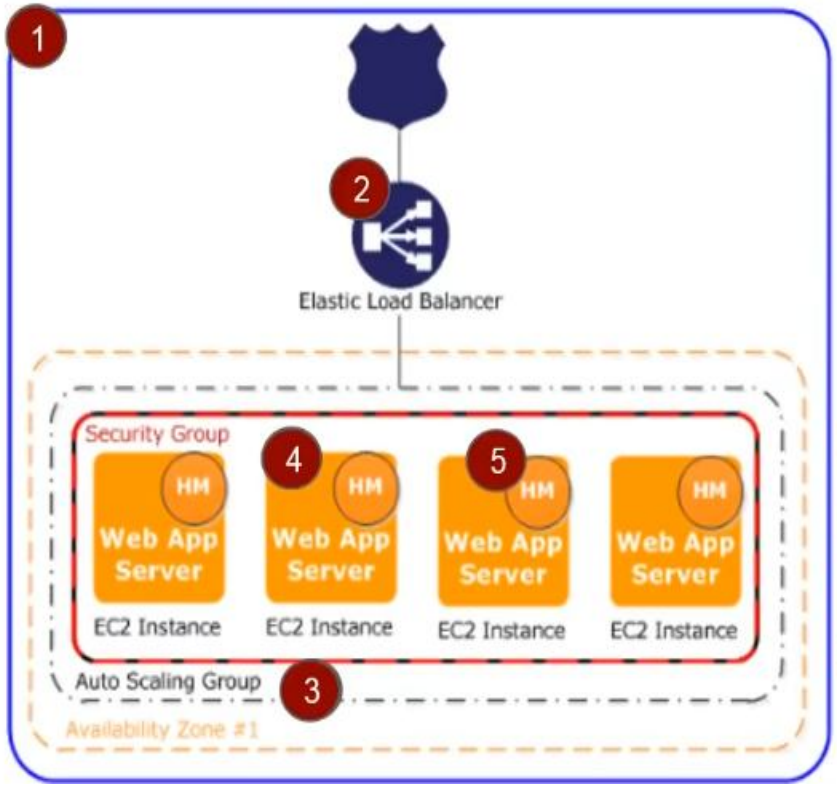
Passed recent health check

Failed one or more checks

Failed three or more checks

Web Server Environment

Web Server Environment Tier handles HTTP requests.



01 Beanstalk Environment

02 Elastic Load Balancer

03 Auto Scaling Group

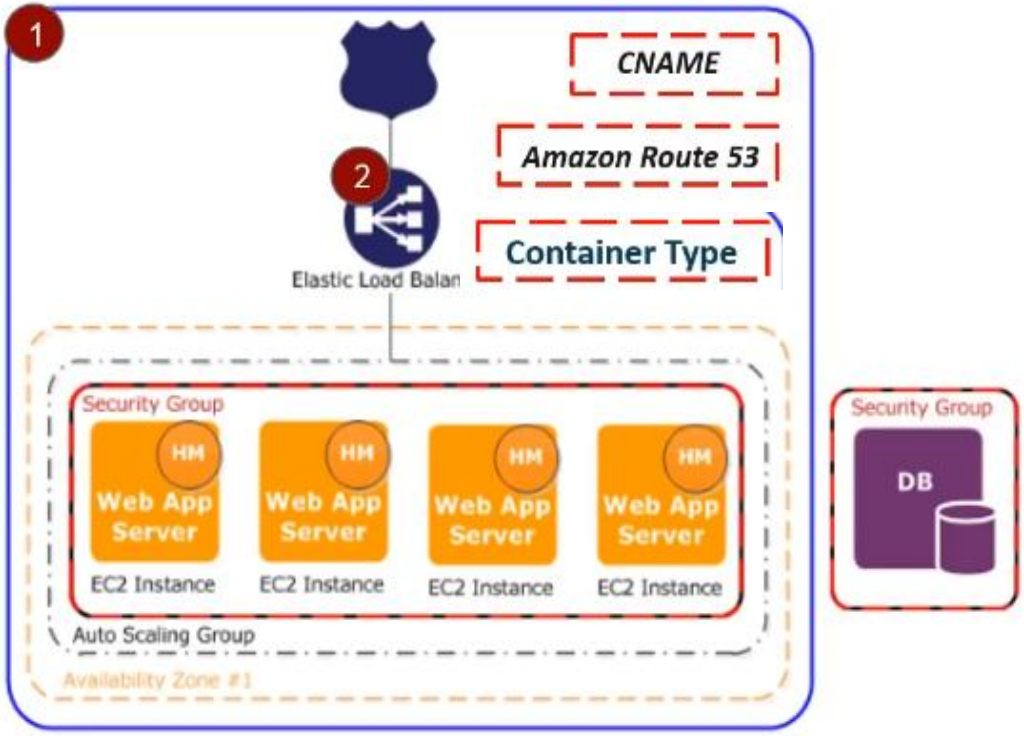
04 EC2 Instances

05 Host Manager

06 Security Groups

Web Server Environment

Web Server Environment Tier handles HTTP requests.



01 Beanstalk Environment

02 Elastic Load Balancer

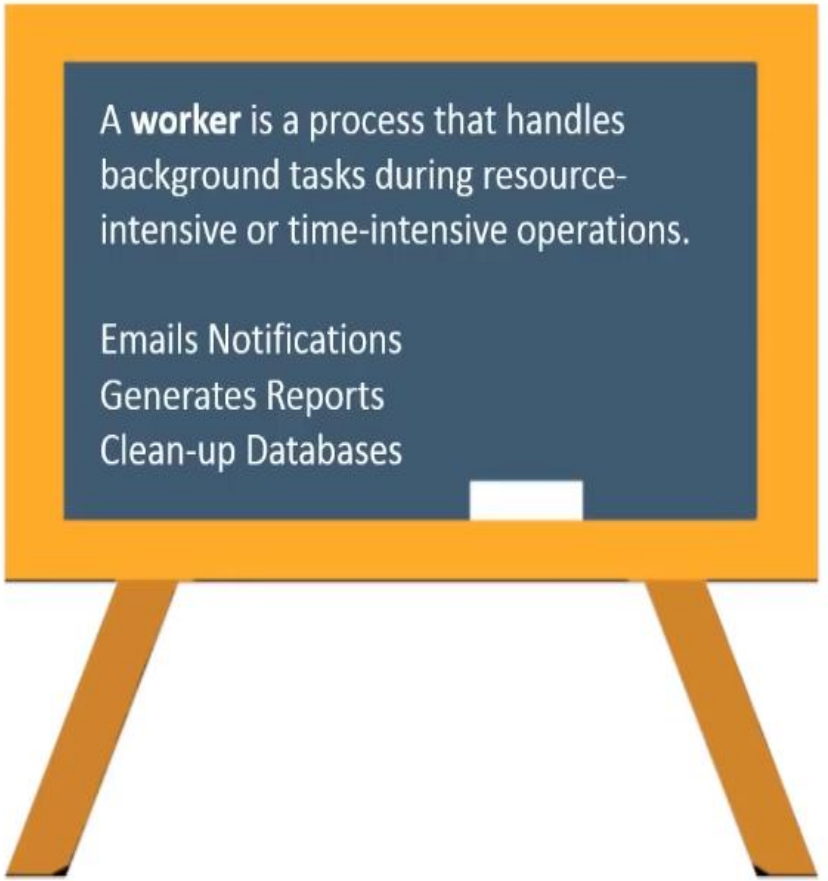
03 Auto Scaling Group

04 EC2 Instances

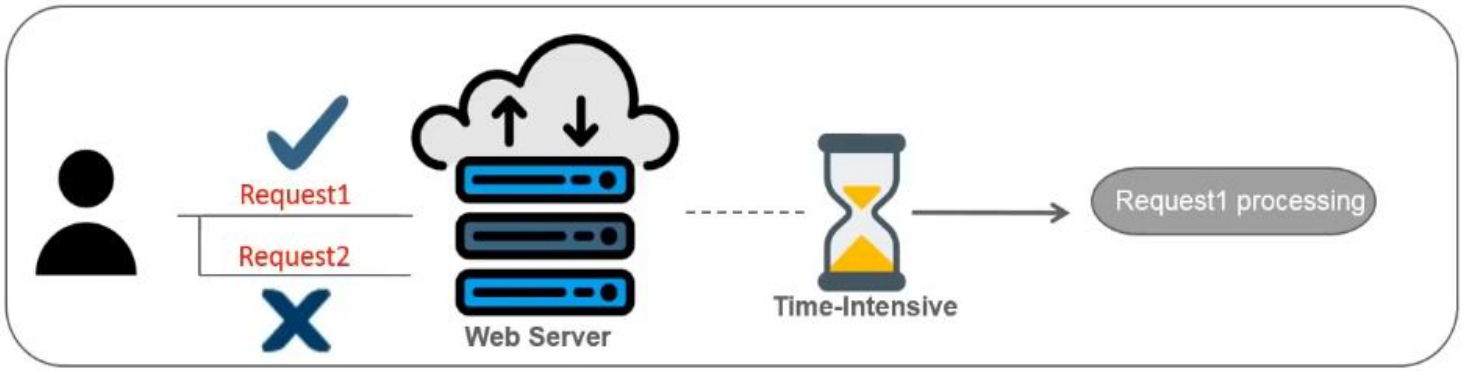
05 Host Manager

06 Security Groups

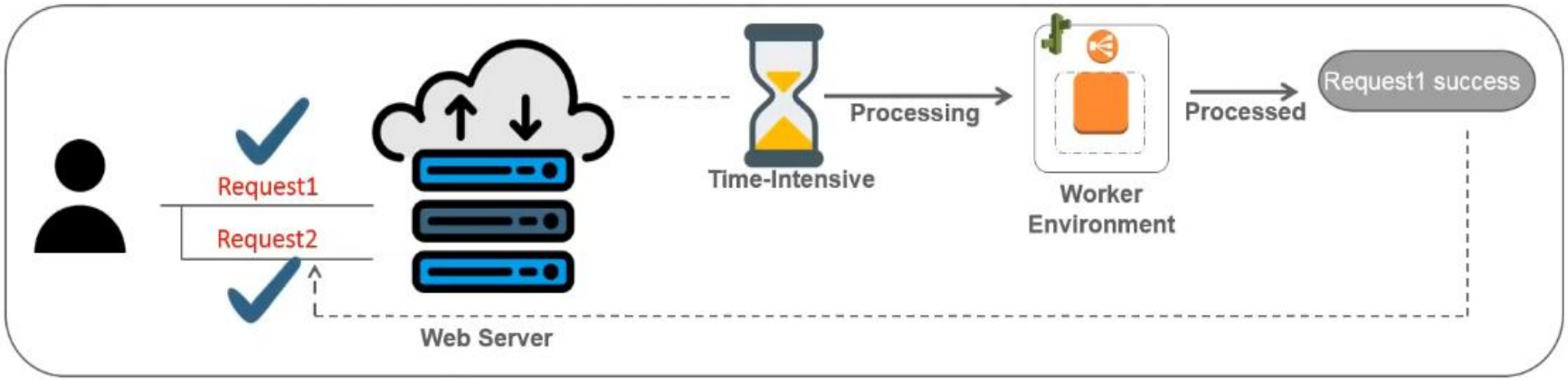
Worker Environment



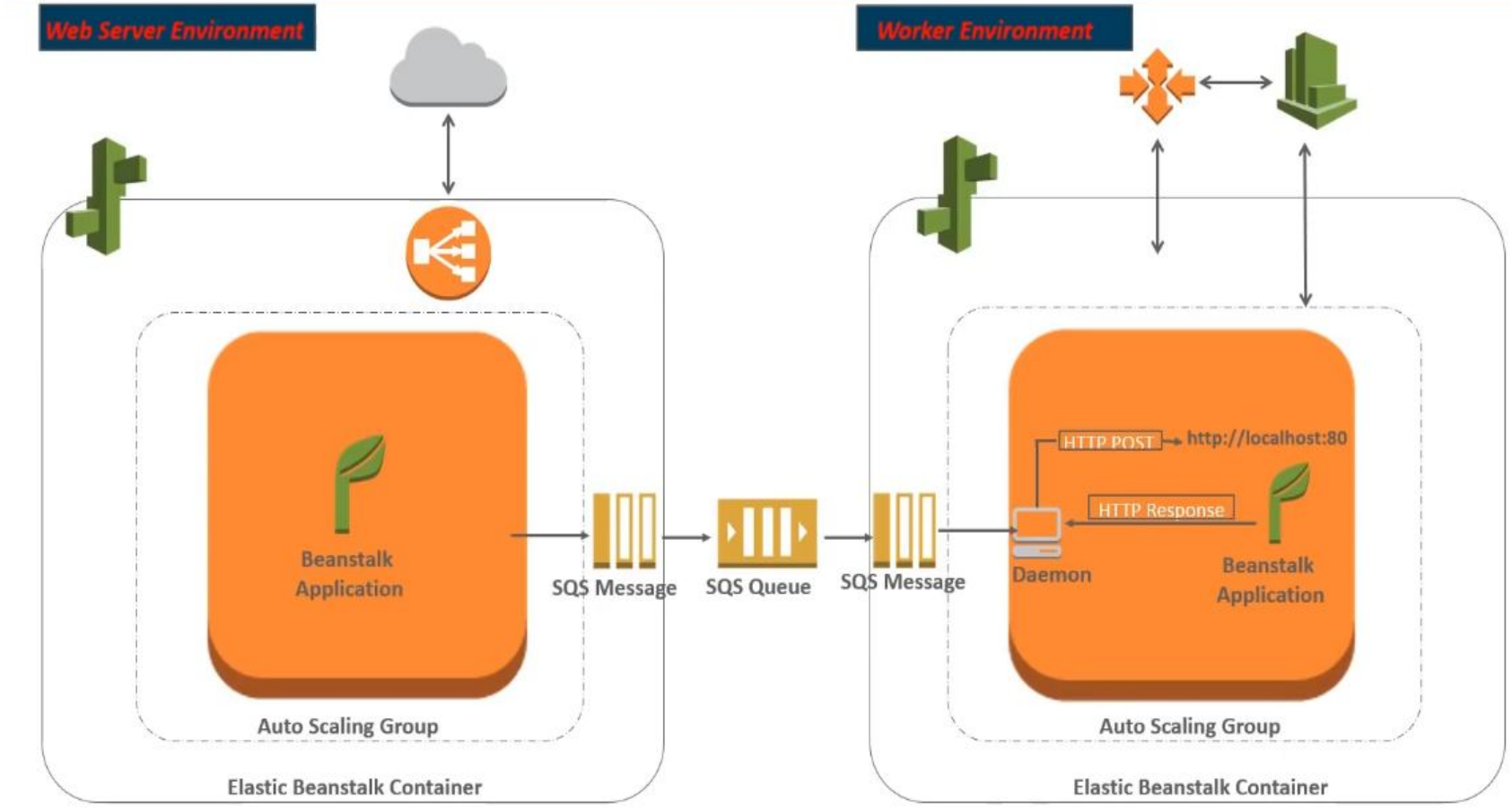
Why Do We Need Worker?



Performance and Requests Accepted

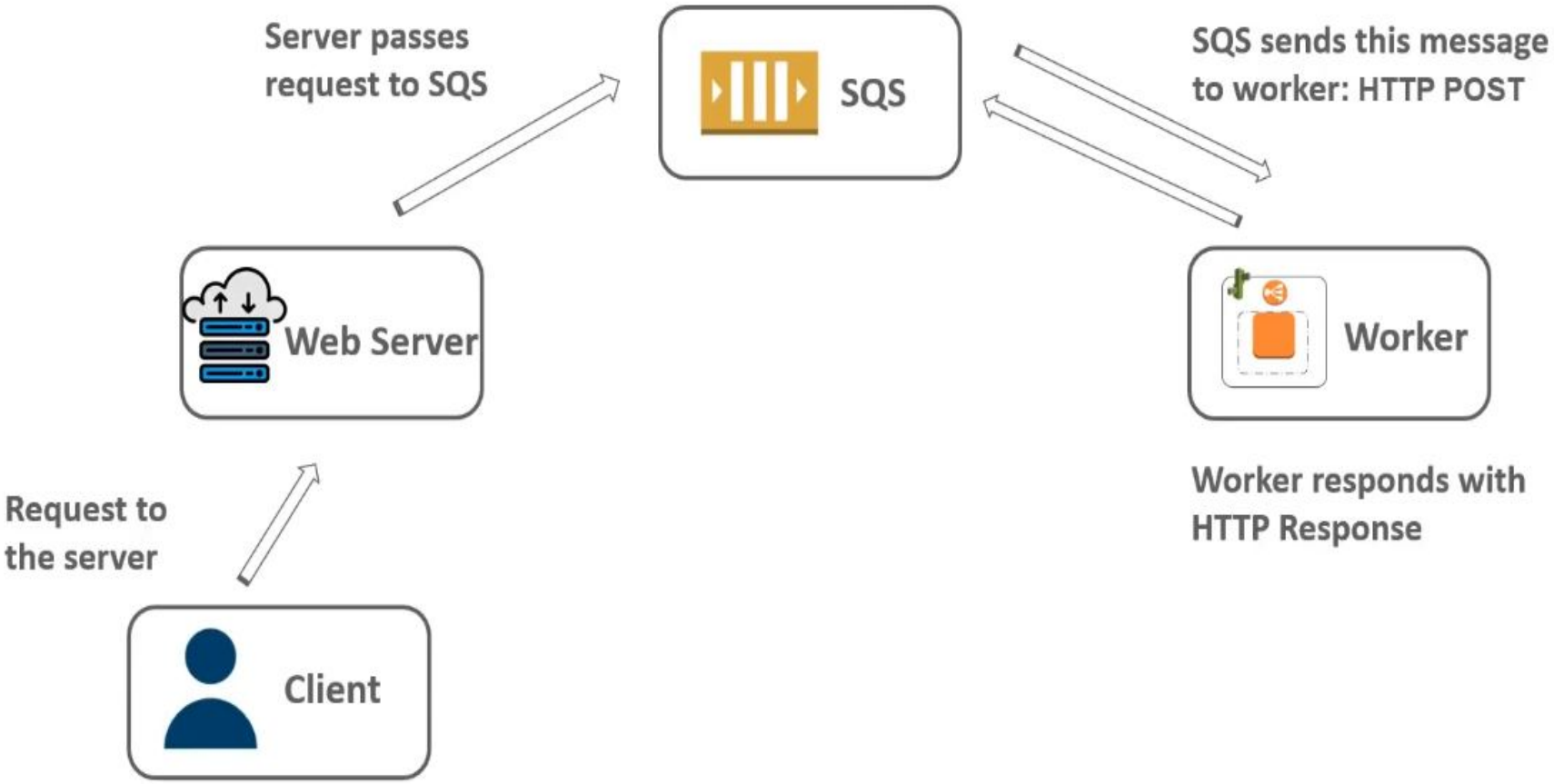


How Do These Two Environments Communicate?



that enables FIFO queues that are designed to guarantee that **messages** are processed exactly once that they are sent.

How Do These Two Environments Communicate?



Communication to other AWS services

Demo-How To Deploy An Application Using Beanstalk

Steps to Deploy Application

Go to AWS management console

Under Compute choose Elastic Beanstalk

Click to

Give the application name

Configure the environment

Choose env. tier : web server

Platform: PHP

Application Version

Sample application/ Your application code

Environment information

Give the domain name

Additional resources

Create an RDS instance

Configuration details

T2.micro

Select your ec2 key pair

Click save and launch

Wait for some time to get activated

Click on the application url on the page.

Add your application foldername/index.php in the url

- **For Static website**

  Service -> Elastic beanstalk -> Get Started -> Give application name -> Choose platform as your file type -> select upload your code -> In local file choose file -> upload -> create application -> Click on URL

- **For dynamic website**

  Service -> Elastic beanstalk -> Get Started -> Give application name -> Choose platform as your file type -> select upload your code -> In local file choose file -> upload -> Configure more option -> in Instances select Modify -> Under EC2 security groups select default -> Save -> In security -> Select key-pair -> save -> In Databases -> Give user Name and password -> Save -> Create application -> Click on URL

- Connect to database using workbench.

Summary of Steps

- Log into the AWS console and go to the compute section and select Elastic beanstalk
- Click on the Get started button
- Next you need to choose the type of Application code that needs to be deployed.
- Choose the sample application and click on "Create Application" or Choose for upload your code and upload your zipped folder.
- As the application gets created, you will be able to see the different events as the infrastructure gets created
- You should be able to see the sample URL located at the right hand of the screen. Click on the URL to view the application.

- If you make any changes to the file, save the file and then right click in Windows explorer and choose on Send to->Compressed(zipped) folder
- Now go to Elastic beanstalk and click on Upload and Deploy
- The changes will start getting deployed
- And now to the URL and see the new changes to your application.

Complete Project Hosting

Open Elastic Beanstalk
Create new Application
Sample name and choose a DNS name
Choose PHP application
Upload Web Hosting Project
Go to Configuration and choose PHP
Upload the project (webhosting.zip)
Once created go to configuration add a ec2 key in Security.
    Create the key before hand
Go to database and create a user name and password
    Wait for 10 Minutes
    Go to enpoint link created under database and view copy the endpoint link for future access.

RDS Configuration

Click the **security group** link in RDS and Edit the inbound rule

Add SSH and Check TCP(Default)

Modify the access for both protocols -> Connect Anywhere

Connect to EC2 instance via the keypair (Putty/SSH)

cd /var/app/current

sudo chmod 777 –R webhosting

cd webhosting

vi config.php

DB Server- 'RDS Endpoint Address'

DB Username- '____'

DB Password- '____'

DB name- 'demo'

Save the file Escape key + :wq

DB Creation

Connect to database from your EC2 Instance

Mysql –h ‘RDS enpoint address’ –u root –p
Create and use database demo
Create user table (refer query.sql file in webhosting directory)

Go to browser- BeanStack URL/webhosting/login.php

You are good to go

In case of any problem refer the RDS setup video in last slide

Video References

AWS Elastic Beanstalk

https://www.youtube.com/watch?v=96DJ2Og90hU

AWS Elastic Beanstalk - Part 1

https://www.youtube.com/watch?v=IUZnlCDQilY

Connect AWS Elastic Beanstalk Using SSH- Part 2

https://www.youtube.com/watch?v=jqxwCppc97s

Configure RDS

https://www.youtube.com/watch?v=KbdWjETGoew&t=44s
https://www.youtube.com/watch?v=NheohB3adV8
https://www.youtube.com/watch?v=I-xaBOISR2s&t=2s

Standard Applications vs. Containerized applications

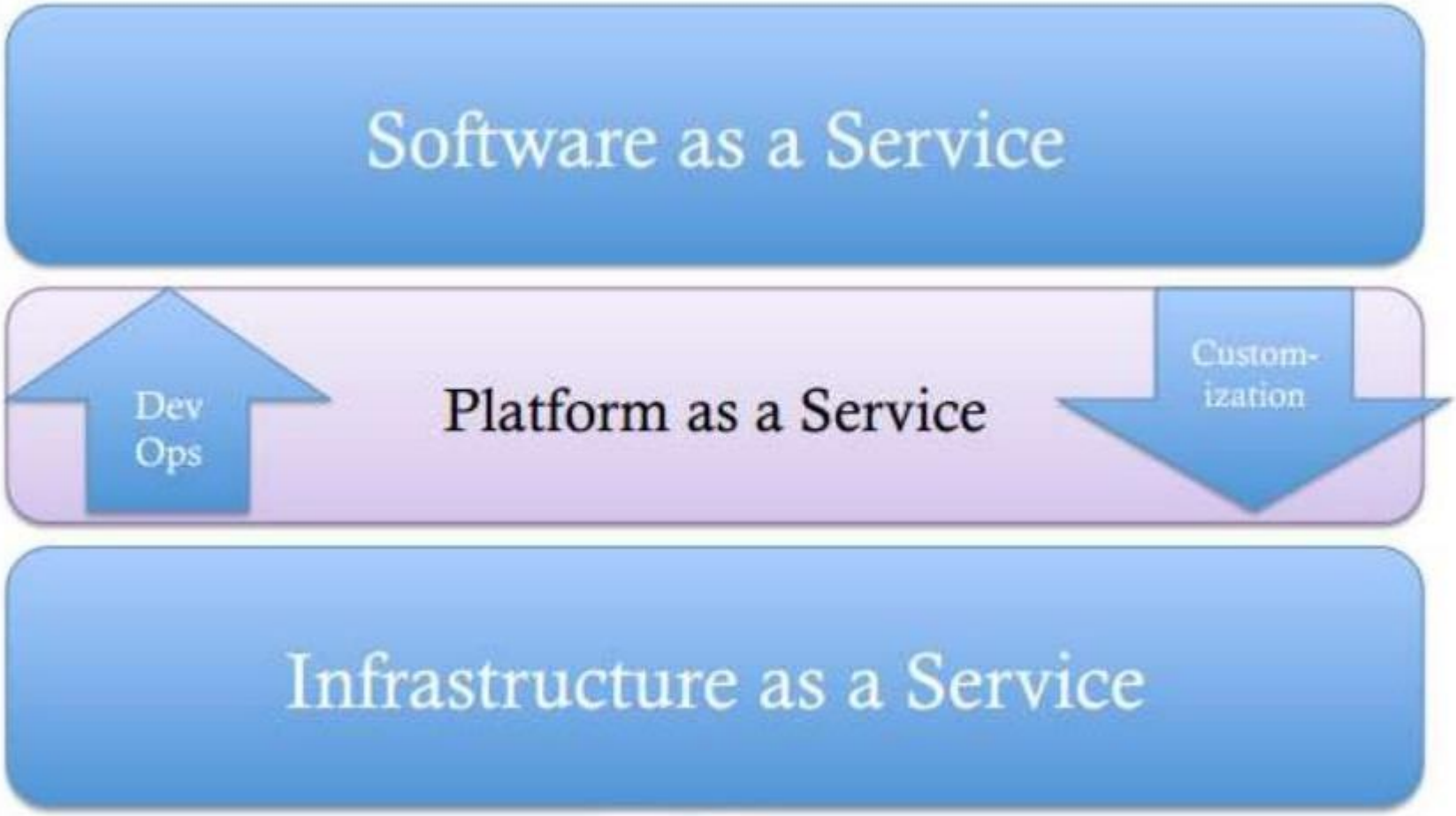
EBS & ECS-Docker

Tamal Dey
MCA,PESU

How PaaS works

- Platform as a Service allows users to create software applications using **tools** supplied by the provider.
- PaaS services can consist of preconfigured **features** that customers can subscribe
- The infrastructure and applications are managed for customers and support is available.
- Services are constantly updated, with existing features upgraded and additional features added.

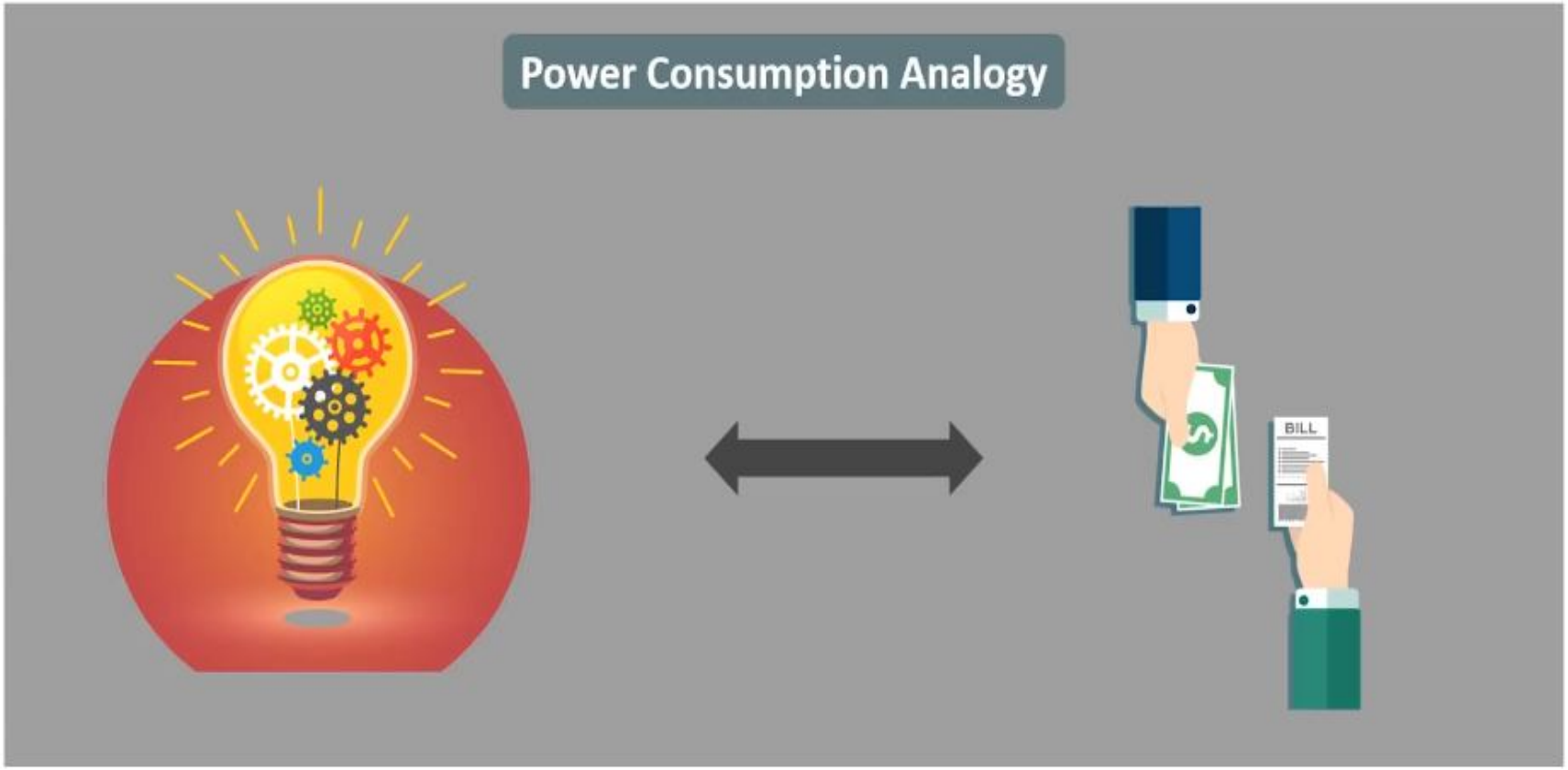
Evolving from different standards



Agenda

What is AWS?
What is Docker?
What is AWS ECS?
Demo: Running Docker in ECS

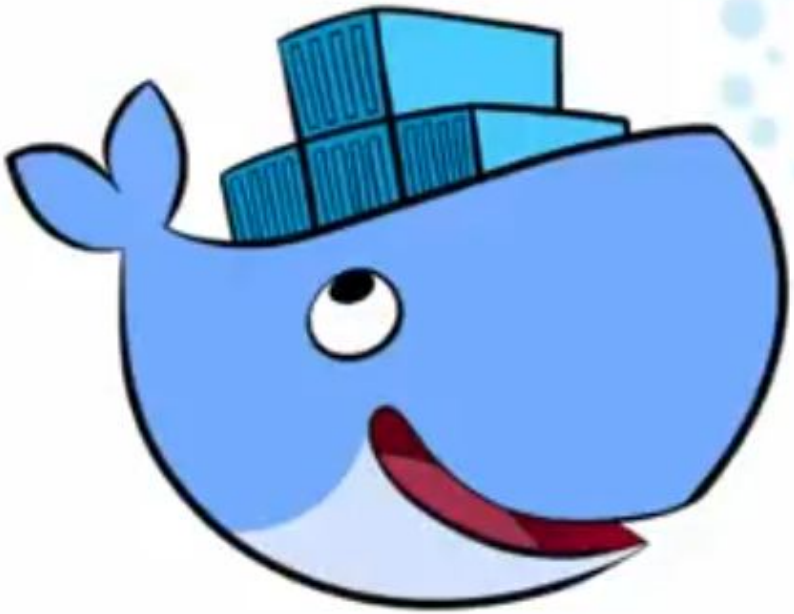
What is AWS?



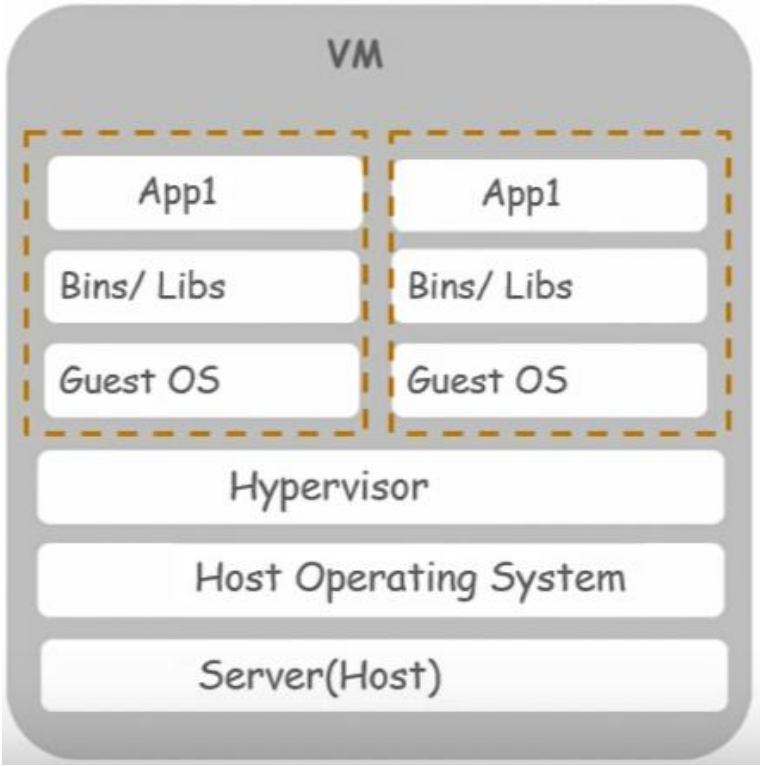
Example: Electricity

In High Level

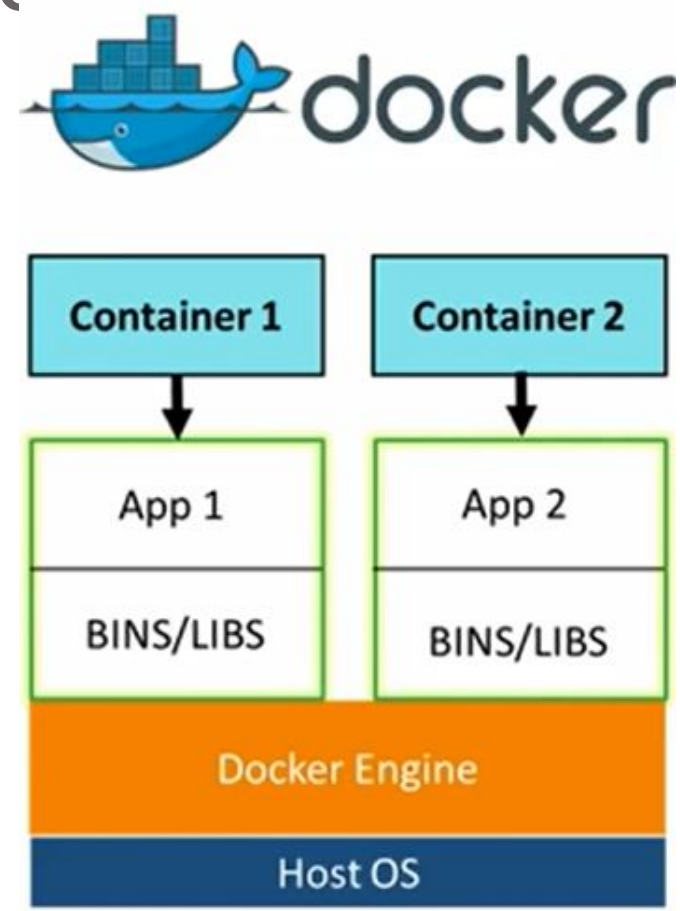
Operating System
Software to build upon
Dependencies to Run Software
Environment Variable
Client Pull the container



Virtualization vs Containers



Hyper-V, Virtual Box



No Guest OS
No Hypervisor
Docker Engine (holds tiny os)

Benefits of Docker

- Docker is a tool designed to make it easier to create, deploy, and run applications by using containers.
- Docker containers are lightweight alternatives to Virtual Machines, and it uses the host OS.
- You don't have to pre-allocate any RAM in containers.

Run a Docker Container from the Docker Registry (EBS)

https://hub.docker.com
https://www.youtube.com/watch?v=lBu7Ov3Rt-M&feature=youtu.be
https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/single-container-docker-configuration.html

How ***docker*** achieved containers?

It was made possible with the help of *LXCs (Linux containers)*

LXCs are *user space* interface for the Linux kernel containment which make it possible to run multiple isolated Linux containers, on one control host (the LXC host).

Linux Containers serve as a lightweight alternative to VMs as they don't require the hypervisors like.

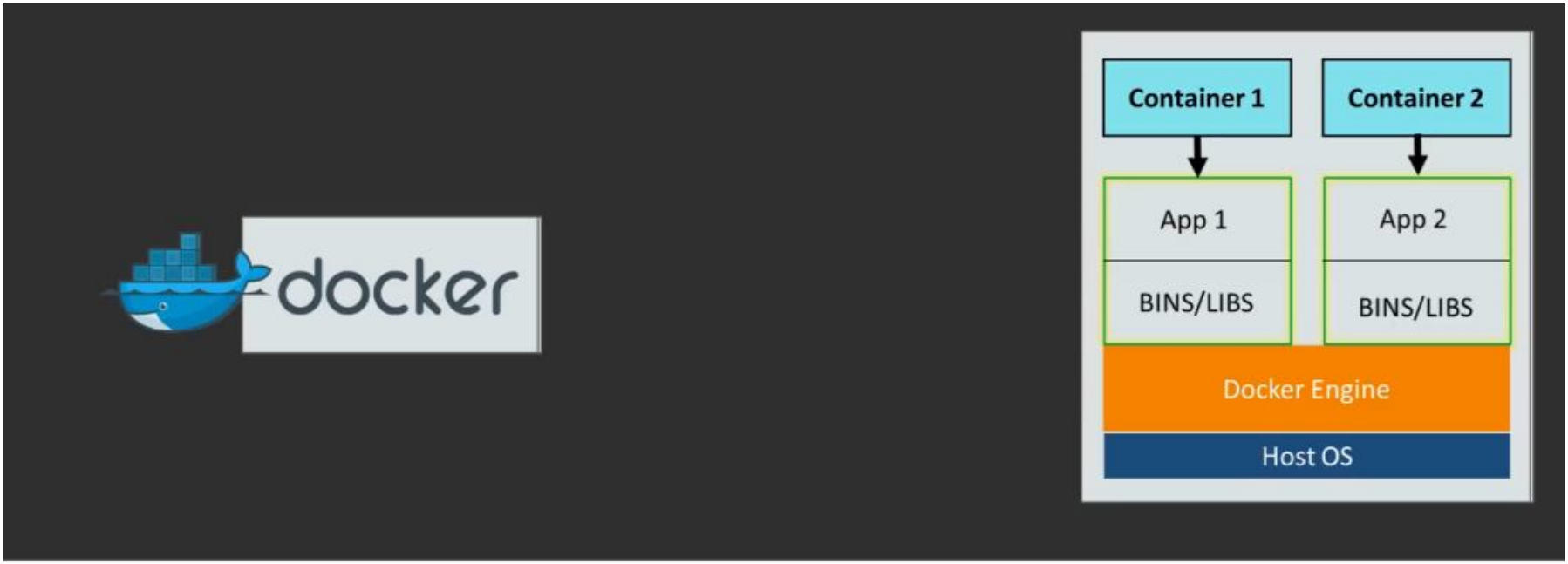
✓Virtualbox,

✓KVM,

✓Xen etc.

Runs in your machine but not in mine

What is Docker?



What Is Docker?

- Docker is a tool designed to make it easier to create, deploy, and run applications by using containers. Docker containers are lightweight alternatives to Virtual Machines, and it uses the host OS.
- You don't have to pre-allocate any RAM in containers.

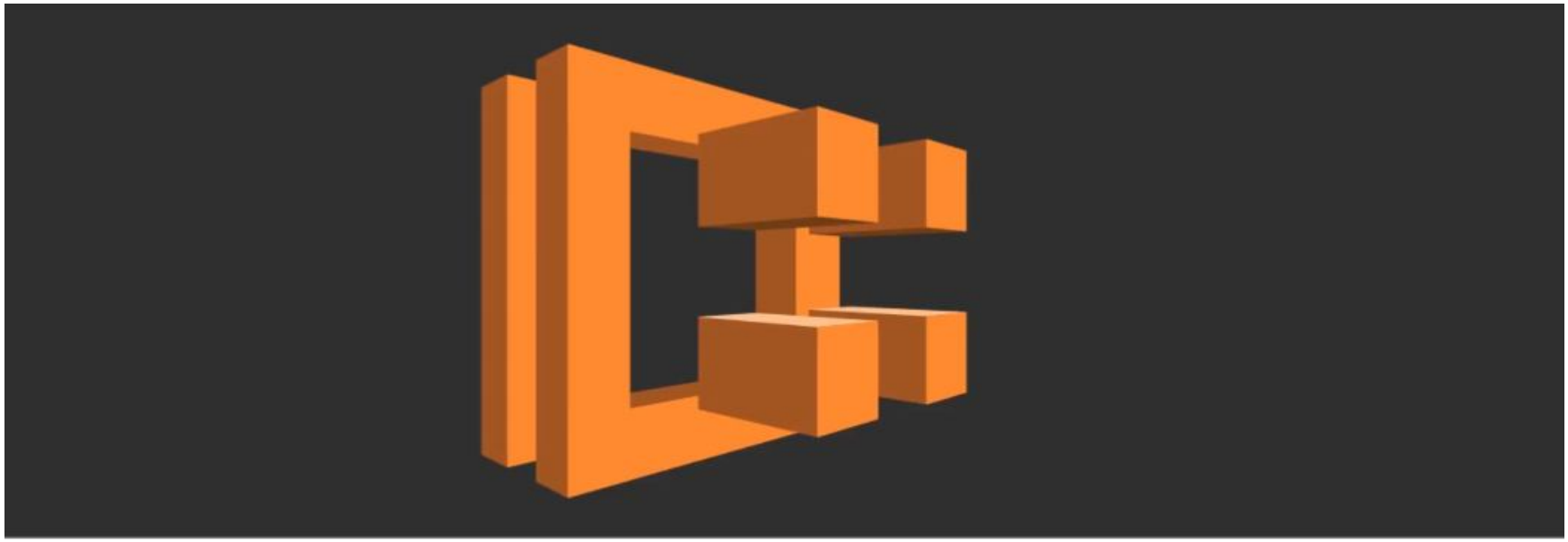
AWS Features

- Virtualization
- Middleware (Hypervisor)

Docker-Container Agent

- Containerization Application
- ECS-Elastic Container Service

What is AWS ECS?

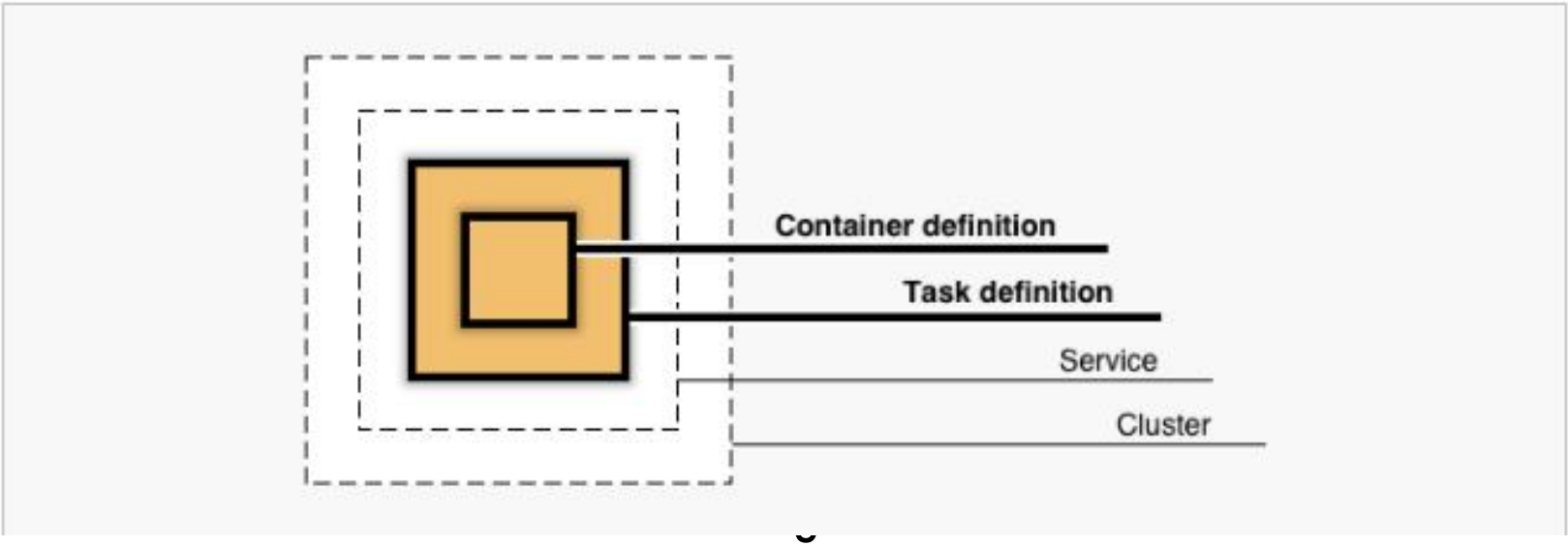


What Is AWS ECS?

Amazon Elastic ECS is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster, which lets you host your cluster on a serverless infrastructure.

Serverless Infrastructure

Diagram of ECS objects



EC

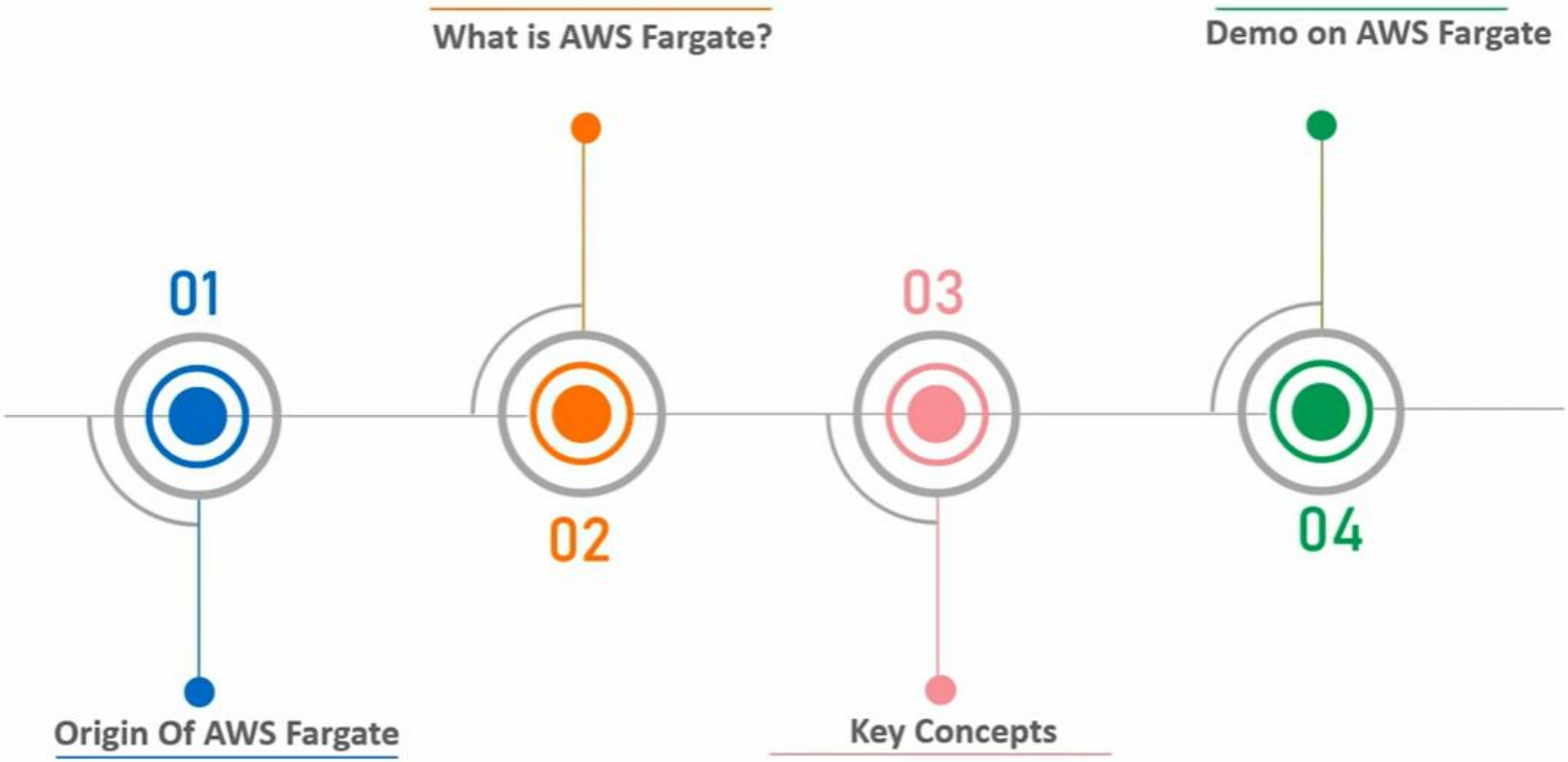
Container Definition: Image Details

Task Definition: Blueprint for the application

Service: Desired number of task and clusters

Cluster: Group your simultaneously running tasks

Agenda

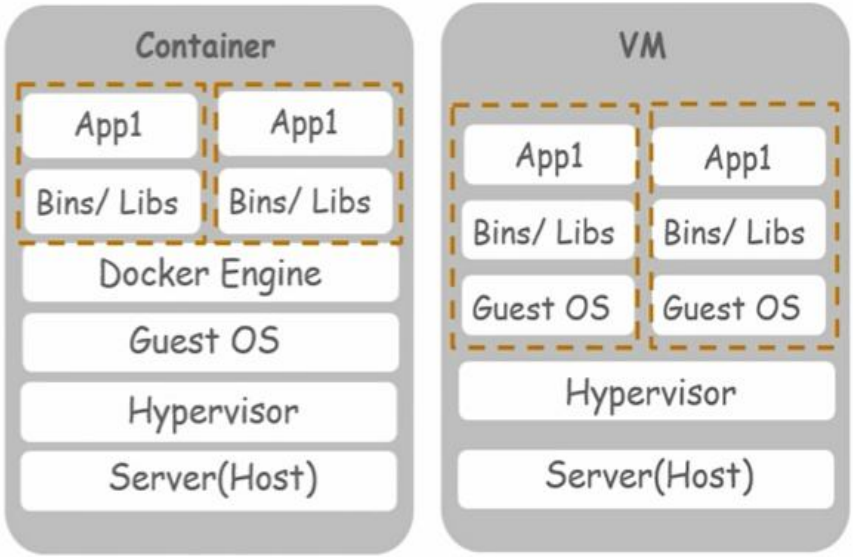

What is AWS Fargate?
Demo on AWS Fargate
01
03
02
04
Origin Of AWS Fargate
Key Concepts

Motivation

1 Run applications on Amazon EC2

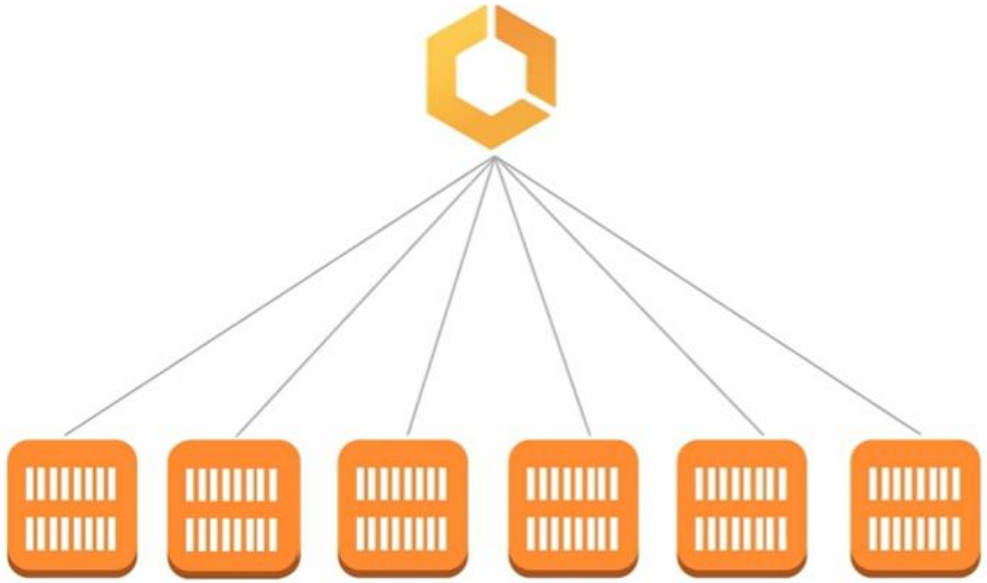
Motivation

2 Deploy applications on docker containers



Motivation

3 Deploy applications on docker containers + AWS ECS

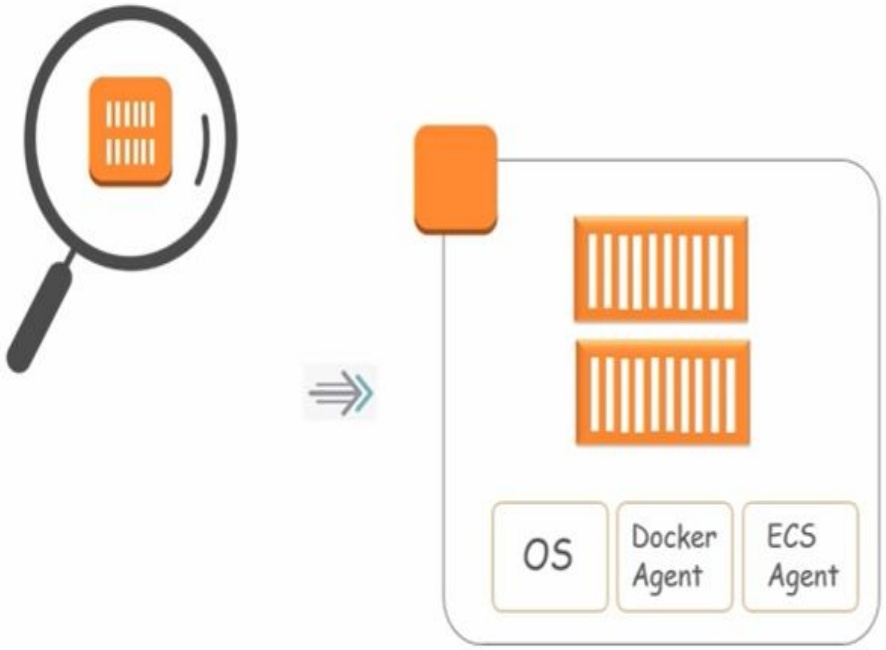


AWS Container Service handles:

- State of EC2 instances
- Applications running
- Resources available
- Resources consumed

Motivation

3 Deploy applications on docker containers + AWS ECS



Customers had to deal with:

- Managing fleet of EC2 instances
- Patching & upgrading software
- Scaling EC2 instance fleet

Motivation

AWS Fargate

Elastic Container Service (**Amazon ECS**)

Elastic Kubernetes Service. (**Amazon EKS**)

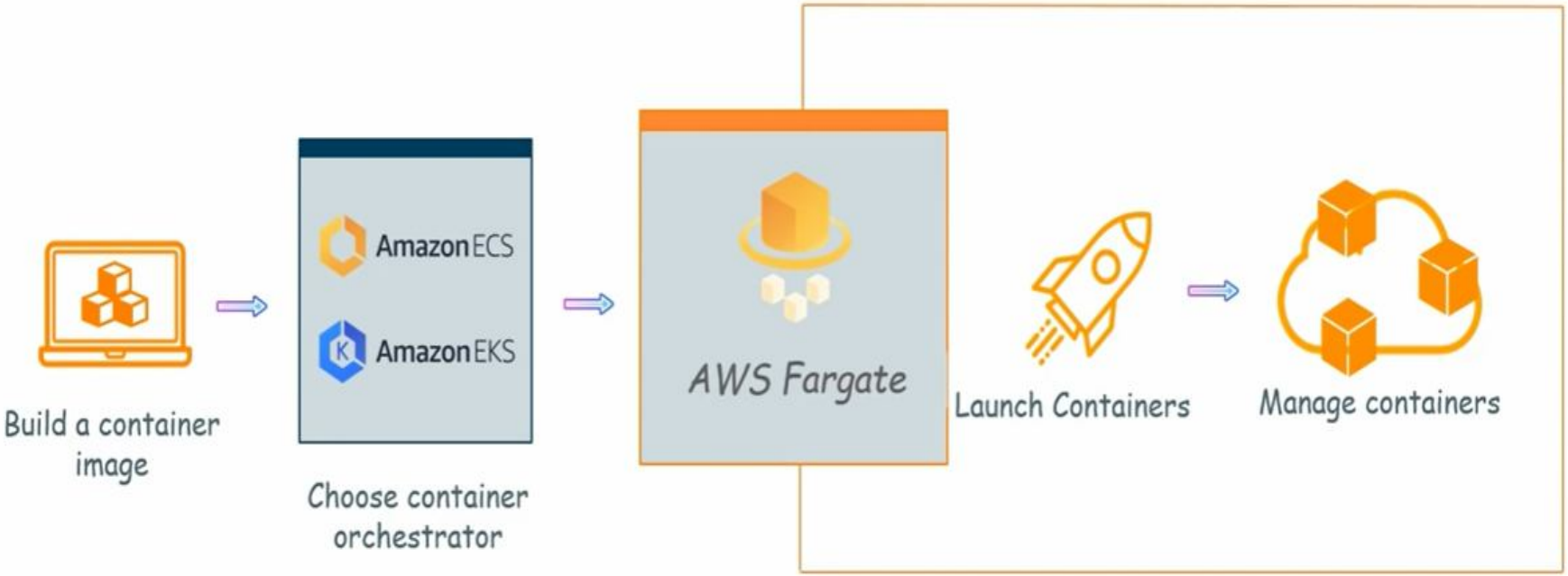
Kubernetes is an open-source container-orchestration system for automating application deployment, scaling, and management. It was originally designed by Google, and is now maintained by the Cloud Native Computing Foundation

AWS Fargate

AWS Fargate is a compute engine for Amazon Elastic Container Service(ECS) that allows you to run containers without having to provision, configure & scale clusters of VMs.

How does AWS Fargate work?

Simple steps to launch containers



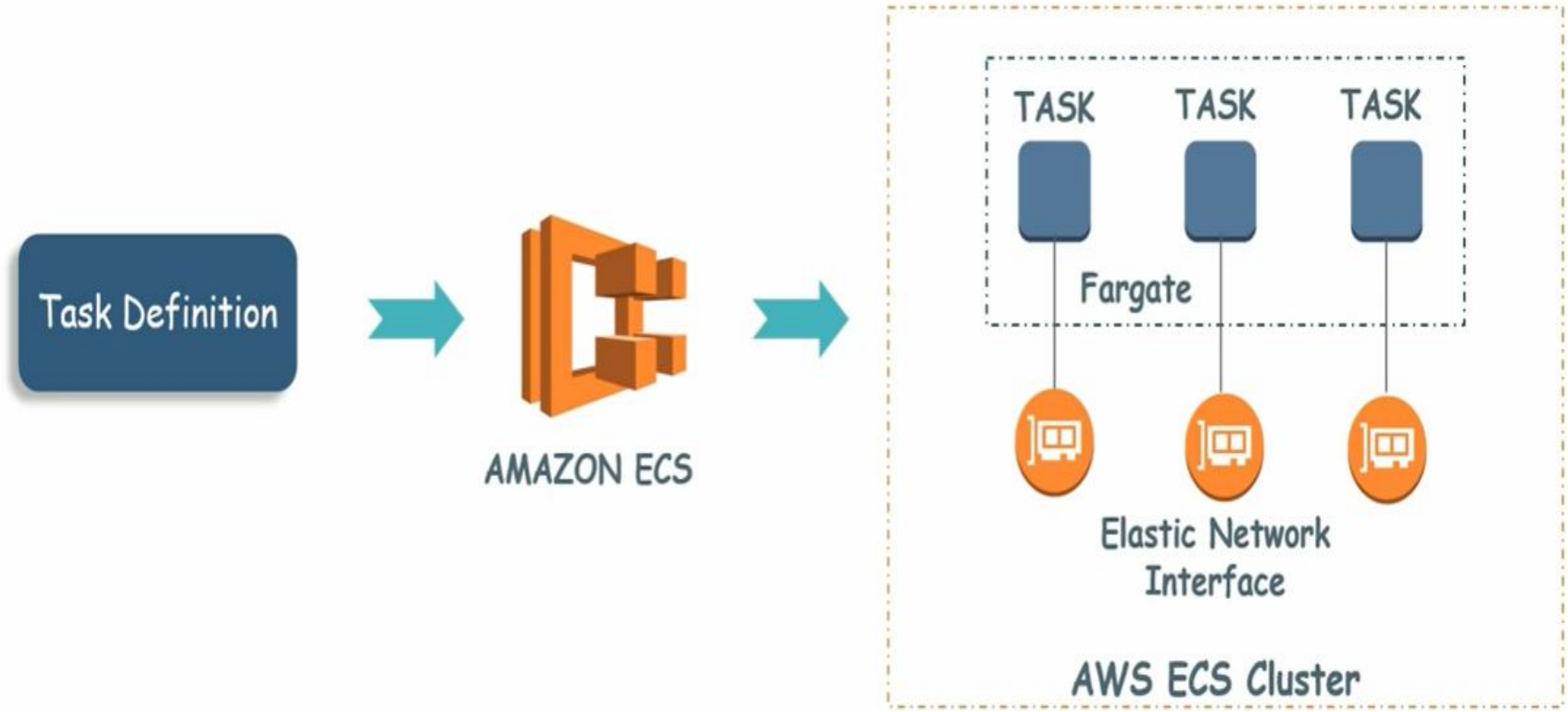
Key Concepts

Docker file

Task Definition

```
{
      "family": "webserver",
      "containerDefinitions": [
            {
                  "name": "web",
                  "image": "nginx",
                  "memory": "100",
                  "cpu": "99" },
       ],
       "requiresCompatibilities": [
             "FARGATE“
       ],
       "networkMode": "awsvpc",
       "memory": "512",
       "cpu": "256",
}
```

Key Concepts



Demo

Search for ECS in Dashboard

Create a **Sample-app**

- See Task Definition
- Number of desired task- 1
- Security Group- Automatic create new
- Load Balancer Type- Application Load balance
- Cluster name: SampleMyDocker

10 + Services will be created.

- View Service and Check all the tabs
- Tasks, Events, Auto scaling, Deployments , Metrices etc.

Application Look

Details – Click the link

Click on ECS Name and Load Balancer Link

Click on DNS to Run the Application in new browser window

or

Go to Details-> Load Balancer-> Target Group Link-

Find & Copy **Complete DNS** in Browser

Delete the service (Cluster) in action

Nginx Server Run

Search for ECS in Dashboard

Create a **nginx** definition

- Check task definition details (Name, N/w, Compatibility, Memory etc.)
- Edit definition, , Task 2, cluster, service.
- Create ECS service and View it (10 Minutes creation time)

Go to Details-> Target Group Link->Load Balancer->

- Find & Copy Complete DNS in Browser

Home Assignment

Create and Upload own Docker Image

Create docker image

https://www.howtoforge.com/tutorial/how-to-create-docker-images-with-dockerfile/

AWS ECS Fargate Tutorial | Running Containers Using AWS Fargate Service

https://www.youtube.com/watch?v=w-nEmKwfrx8

Running Docker In Production Using AWS ECS

https://www.youtube.com/watch?v=zp7gUCgyS34

Running Docker In Production Using AWS ECS

https://www.youtube.com/watch?v=-vgsxR_8DcE

Running Docker In Production Using AWS ECS

Benefits

- CONTAINERS WITHOUT INFRASTRUCTURE MANAGEMENT
- CONTAINERIZE EVERYTHING
- SECURE
- PERFORMANCE AT SCALE
- DESIGNED FOR USE WITH OTHER AWS SERVICES

How it works

- **Build Container Images**
  Create build and push your container images to a registry
- **Define Your Application**
  Select container images and resources needed for your application
- **Launch containers**
  ECS launches containers for your application
- **Manage Containers**
  ECS scales yout application and manages your containers for availability

Example

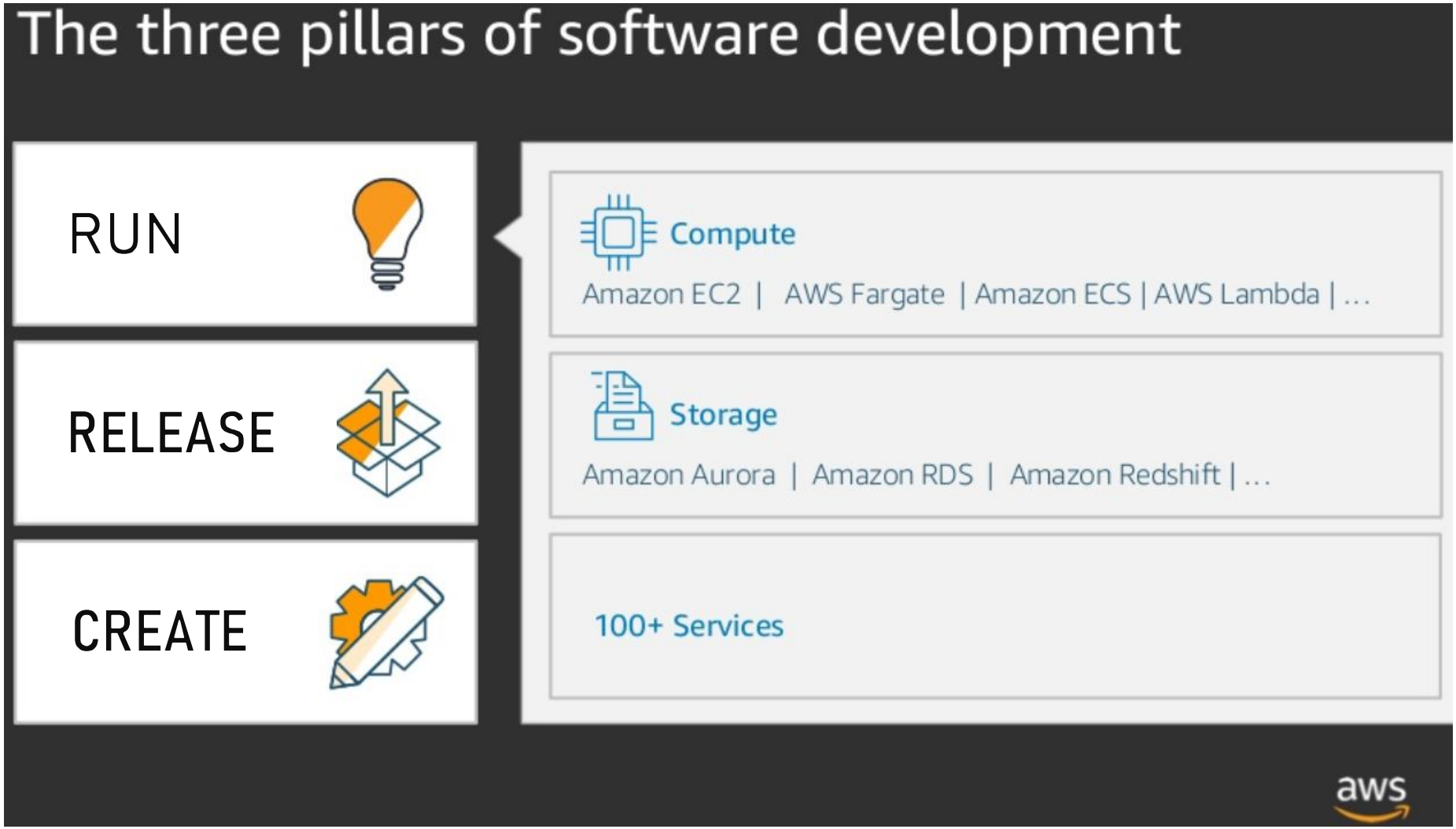
- Create repository and Upload Docker image
- Create Cluster
- Create task Definition
- Create Service and run the task
- Test your Application

PaaS- Development Environment

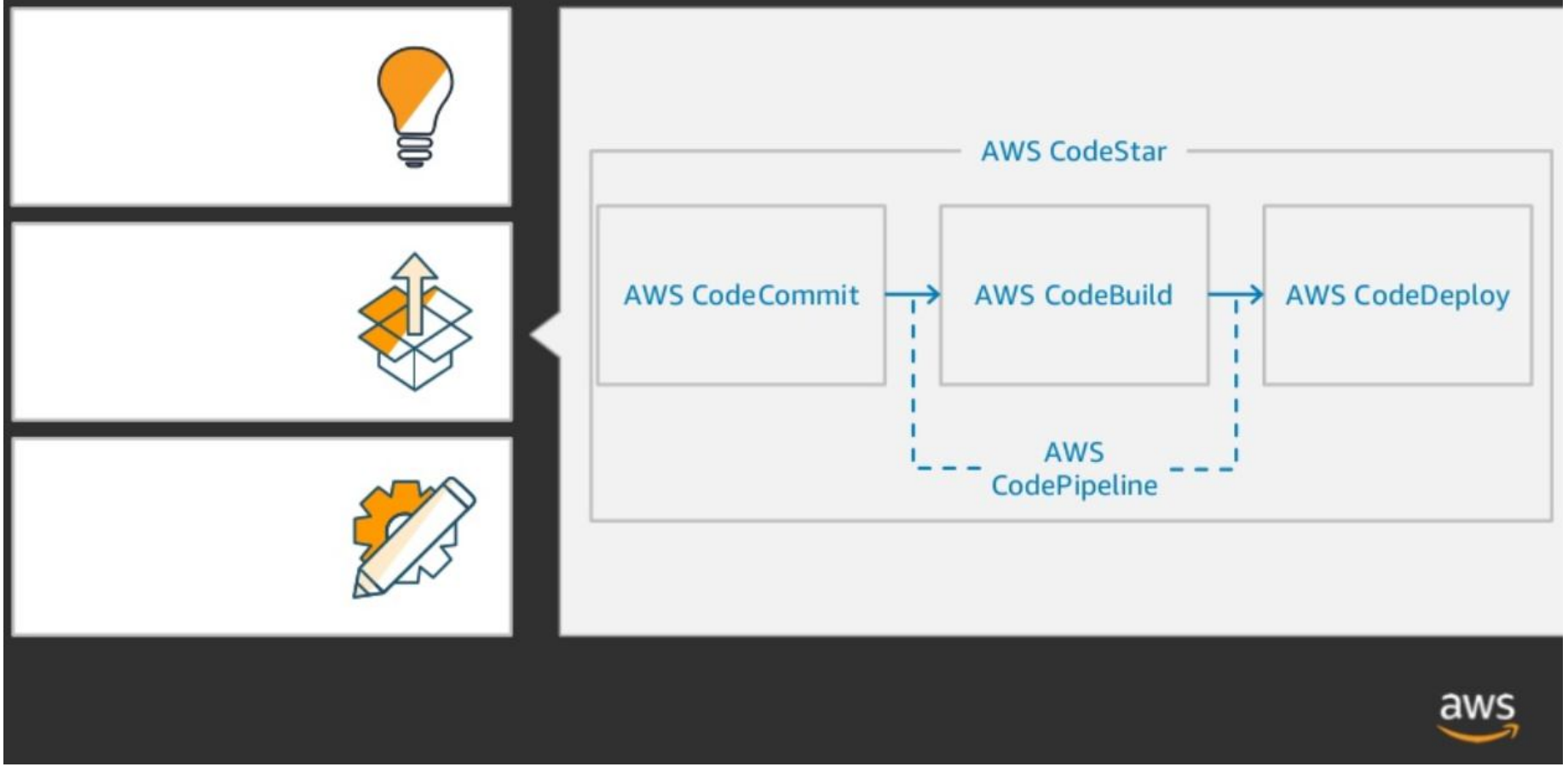
Cloud9

By: Tamal Dey, MCA, PESU

The three pillars of software development



The three pillars of software development



The three pillars of software development

Developer Tools

AWS Cloud9

A cloud IDE for writing, running, and debugging code

aws

1. Developers are building for the cloud, but use local machines

Pain points

- Rely on local machine's hardware/configuration
- Hard to multi-task on various projects
- Difficult to work from multiple locations

Why do we build IDE?

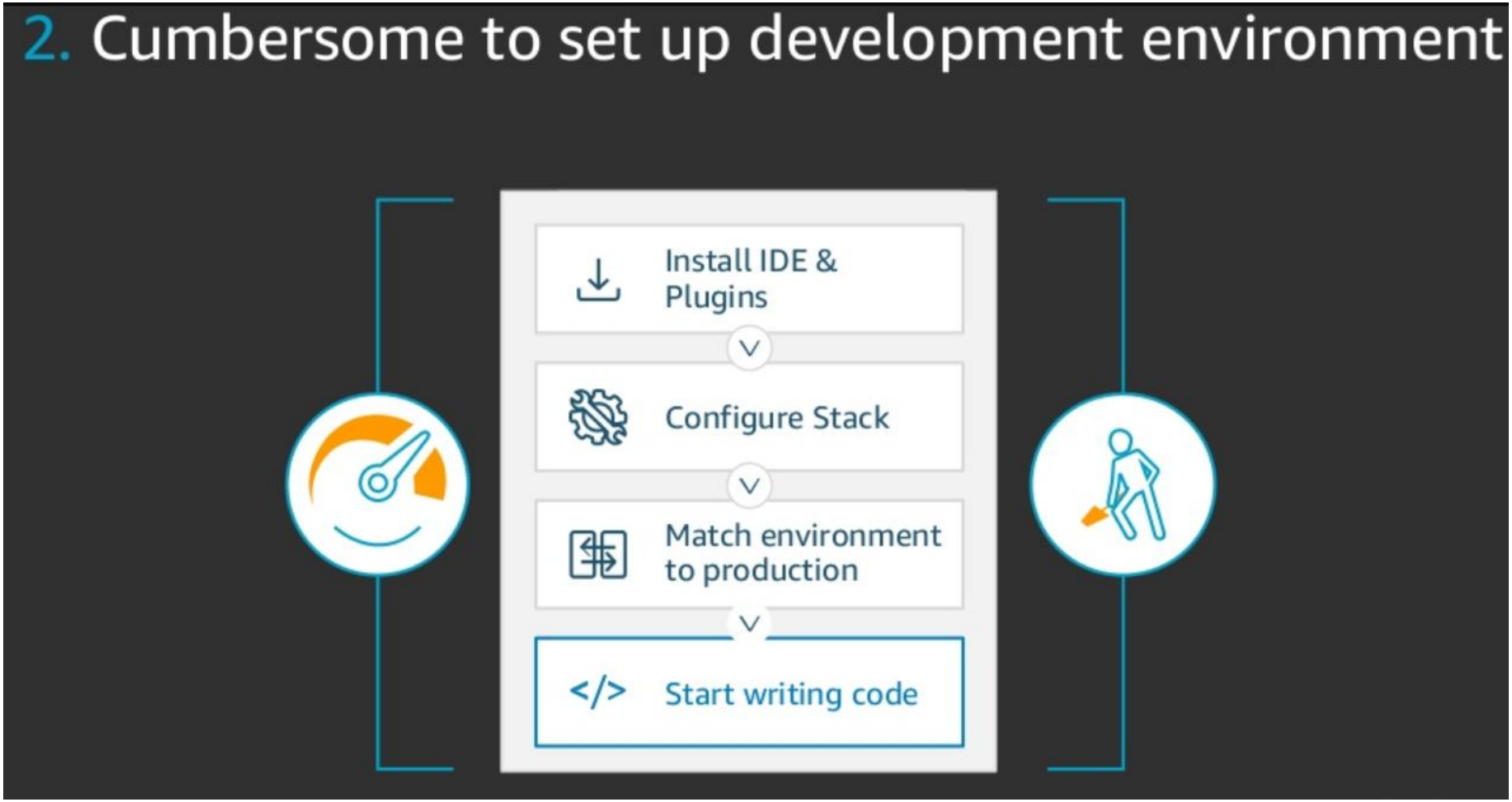
Building applications for cloud but use local machines for development

- Rely on local machines for hardware and software configuration
- Hard to do multitask on various projects
- Difficult to work from multiple locations

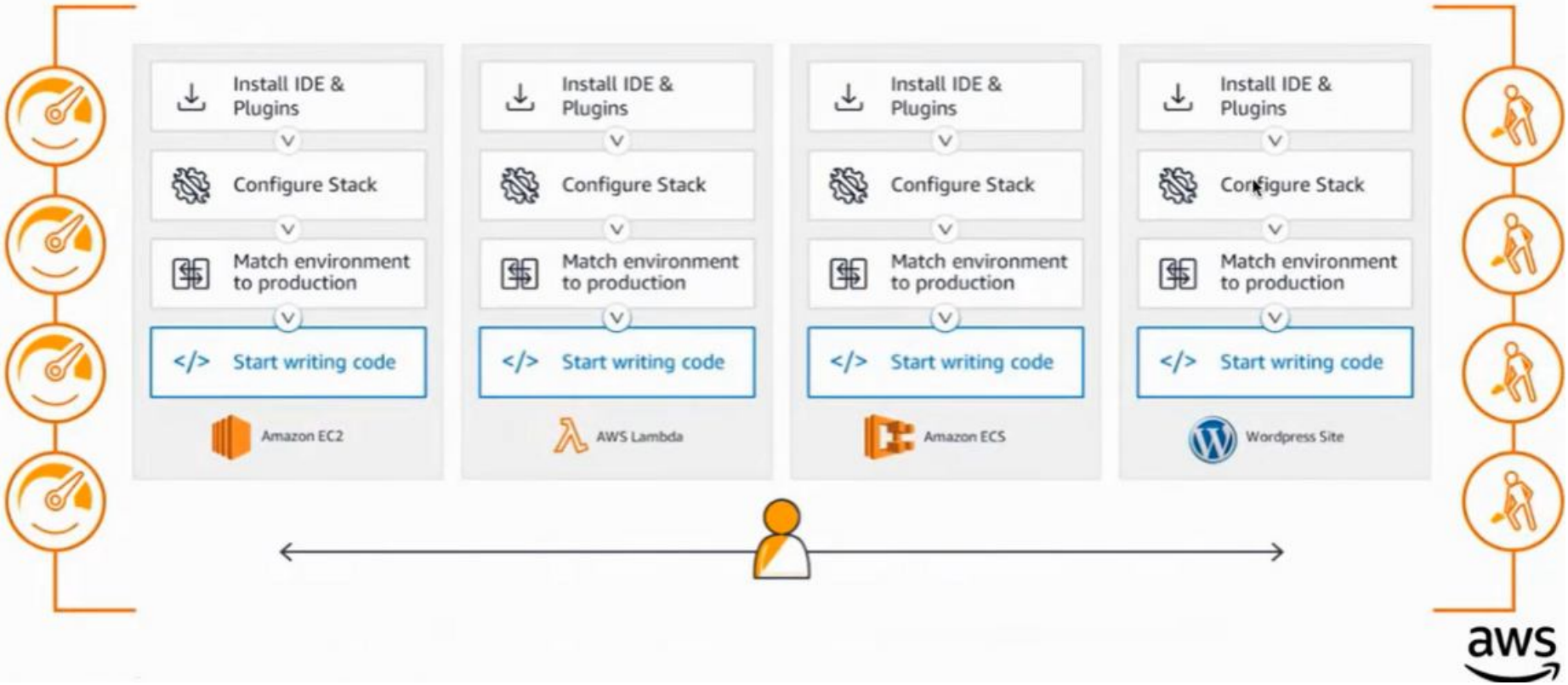
Relief from:

Cumbersome (large or heavy)to setup new environment for development

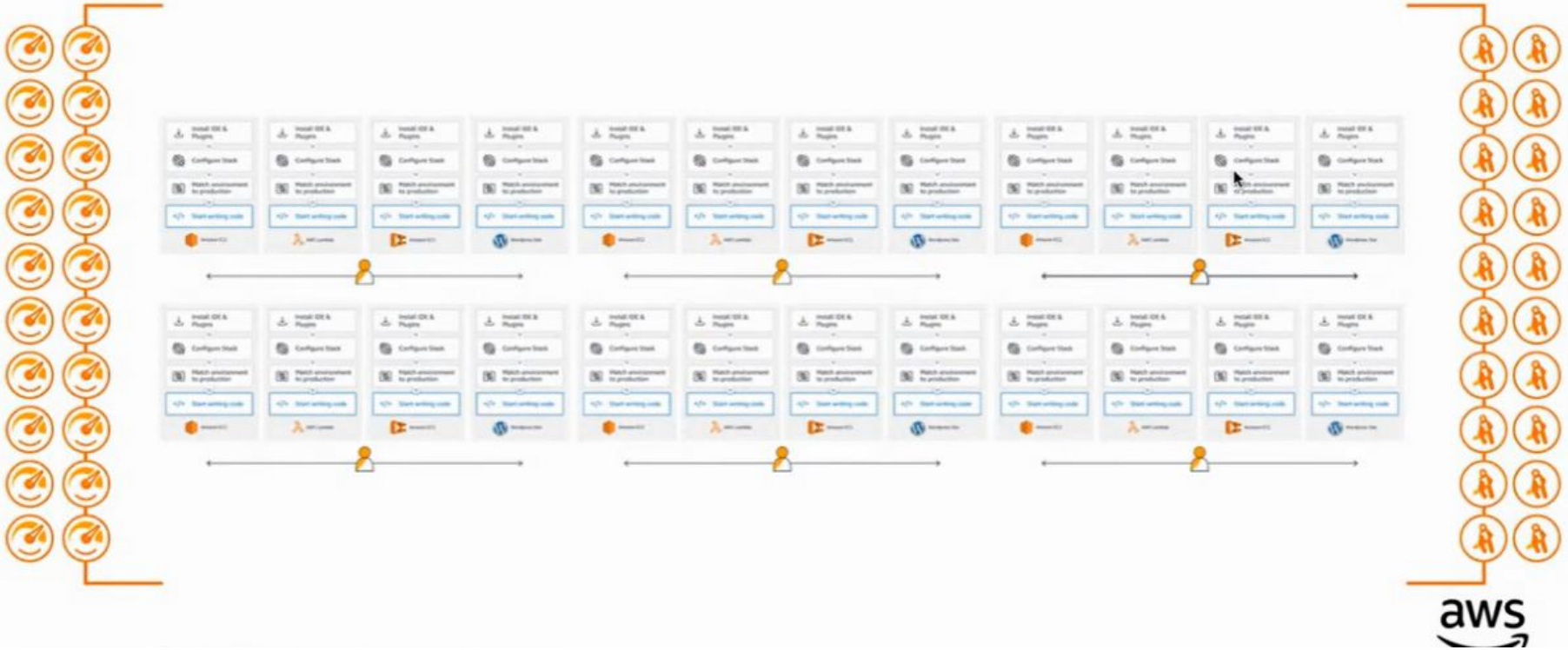
- Install IDE and plugin
- Configure your stack such as docker
- Match the environment in production
- ***Start writing code only***


2. Cumbersome to set up development environment
Install IDE & Plugins
Configure Stack
Match environment to production
</>
Start writing code

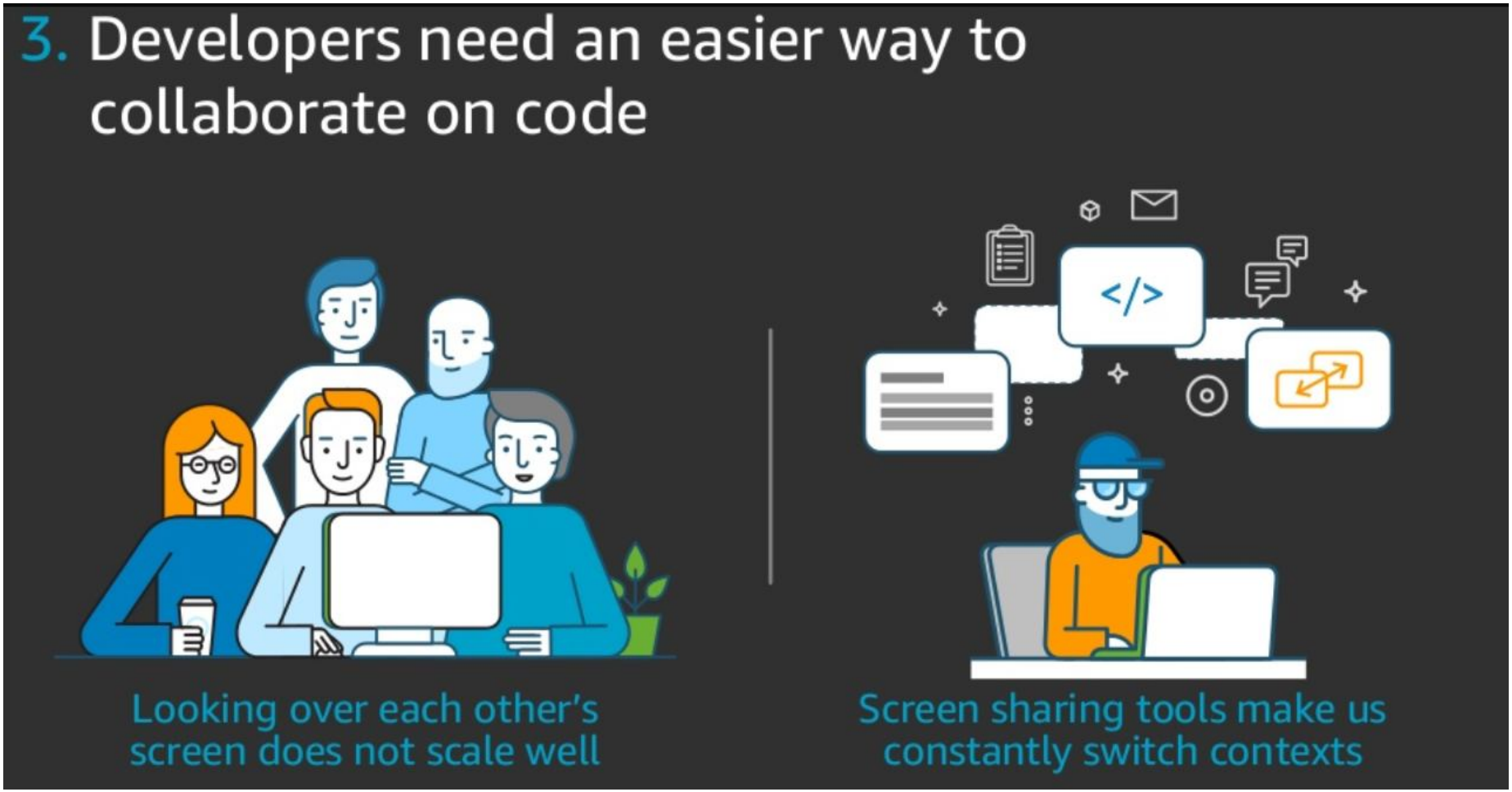
2. Cumbersome to setup development environment (contd.)

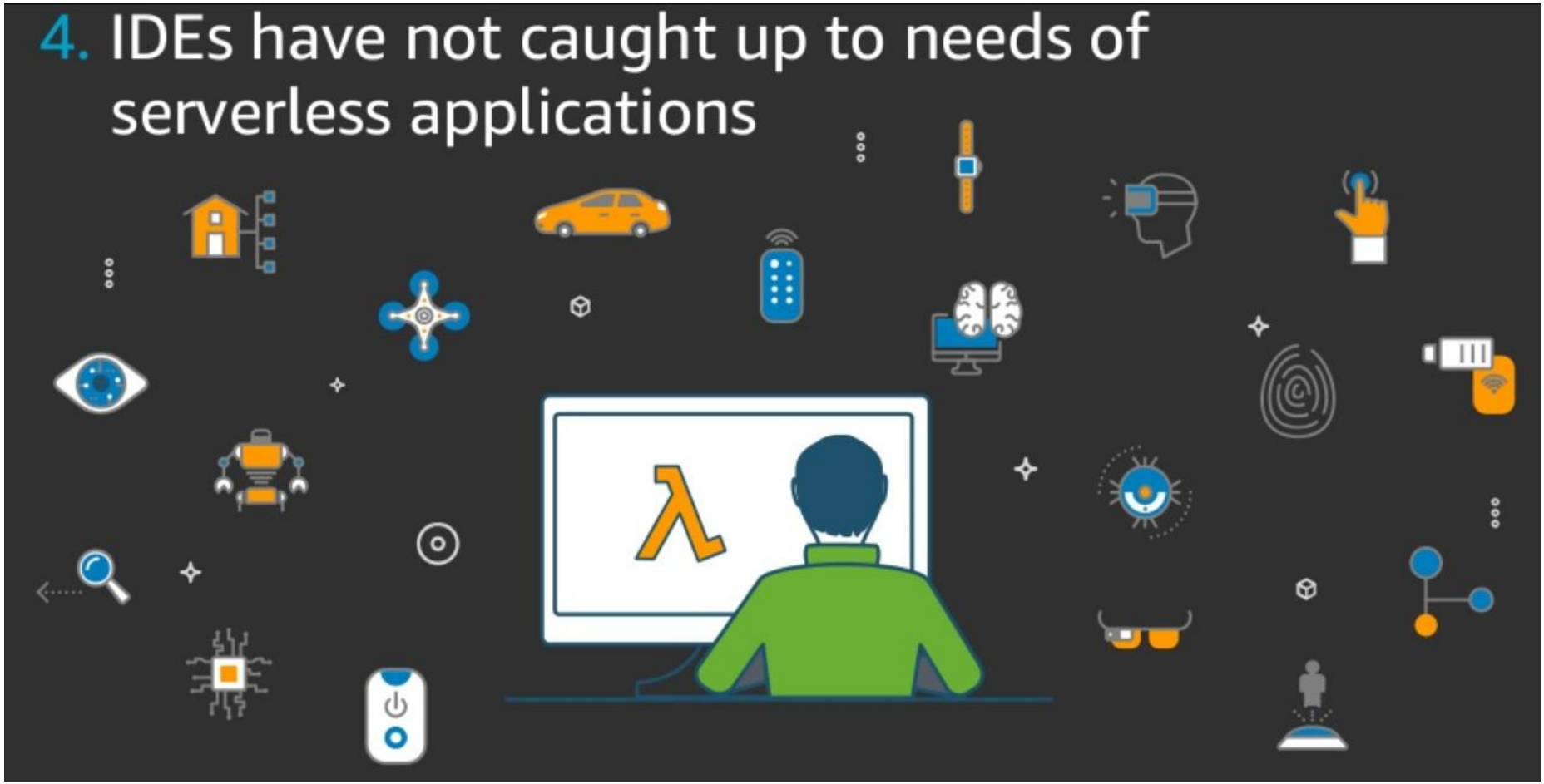


2. Cumbersome to setup development environment (contd.)

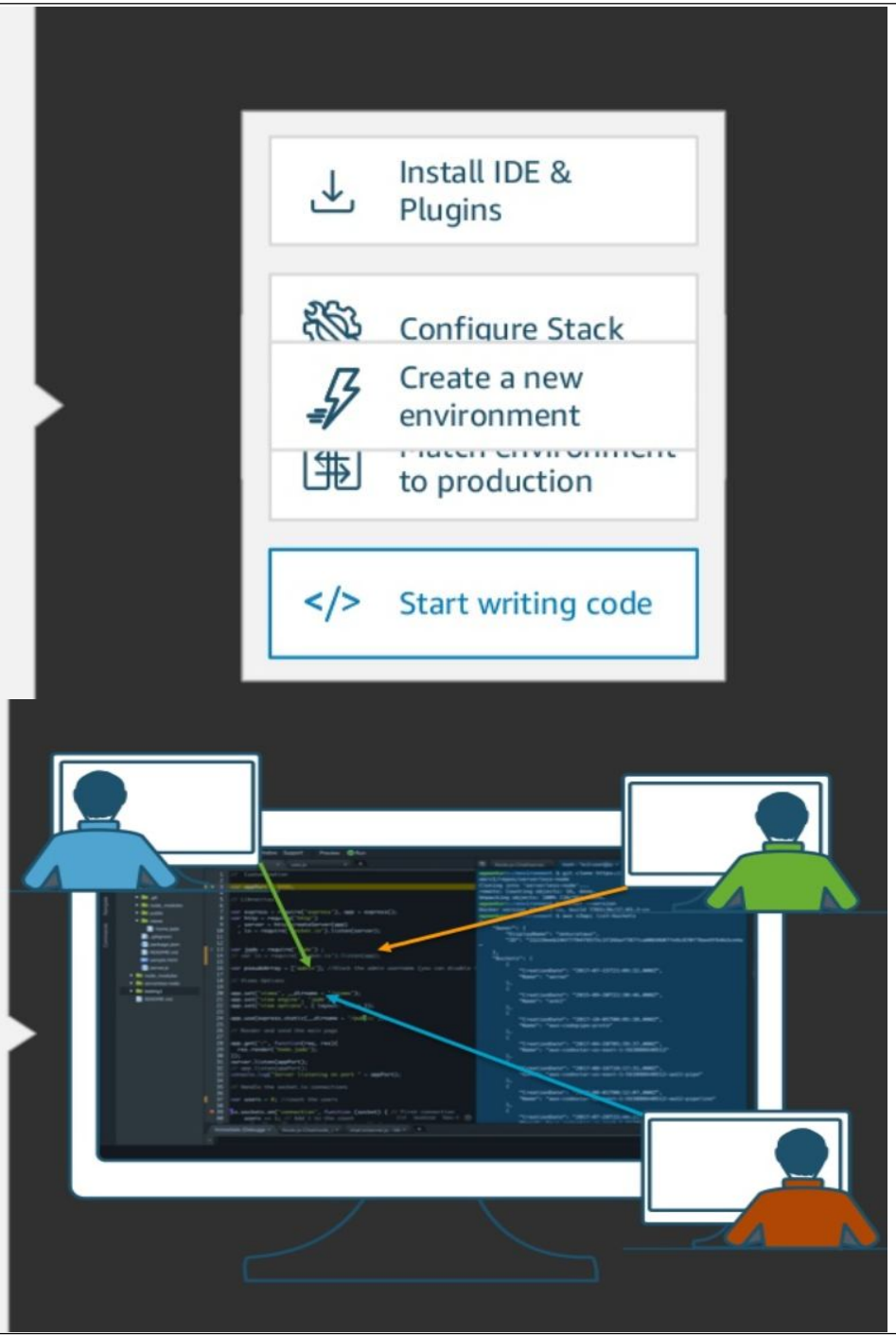


Why need an IDE?




4. IDEs have not caught up to needs of serverless applications

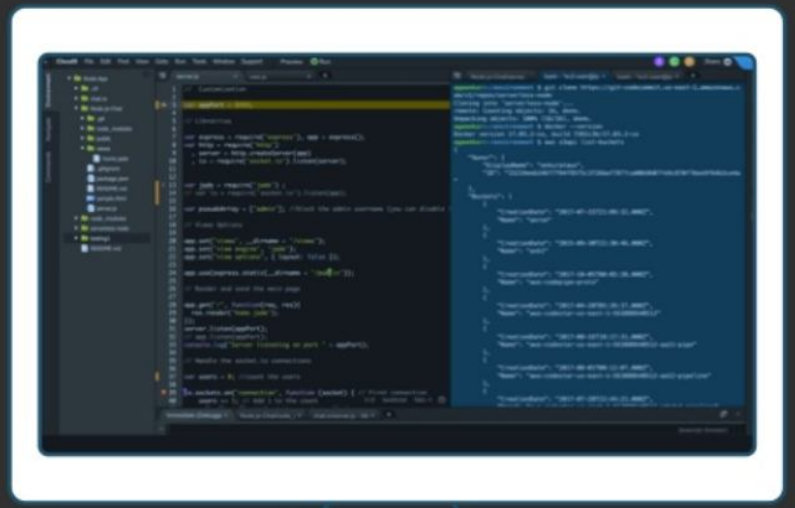
Start new projects quickly



Code together in real time

Build serverless applications with ease

Direct terminal access to AWS services



AWS Cloud9 availability


AWS Cloud9 availability
US East (Ohio)
(us-east-2)
US West (Oregon)
(us-west-2)
EU (Ireland)
(eu-west-1)
US East (N. Virginia)
(us-east-1)
Asia Pacific (Singapore)
(ap-southeast-1)
Generally Available today

Pricing

Managed EC2	Own Linux Server (SSH)
No Charge for IDE	No Charge for IDE
Standard compute and storage charges to run & store code (Amazon EC2, Amazon EBS)	

Cloud9 Features

IDE in the browser
Start new projects easily
Create a new environment
Code together in real time
Build serverless applications with ease
Direct terminal access to aws services

Steps

Go to cloud9 in devops
Create a new environment
Give name
Select ec2 instance type, where the application to be deployed
Wait for moment, here we go with cloud9 IDE
Search for CloudFormation in service link(Top left corner)
Go to terminal and type

Ubuntu commands: top
Git clone: visit index.html, demo.js, hello.rb, info.php
console.log(bar.abc);

Click on AWS Resources [Right Side] (static-demo Project Link)

Lamda function

```
{    "payload":"testing"}
callback(event);
```

To Share the IDE with Team Mates [Right Corner]

- Create IAM role
- Create username and assign to a Group. Name the group
- Add Cloud9 Administrative Permission
- Add IAM User to your IDE
- Provide the access link from IAM URI

Reading resources

https://www.youtube.com/watch?v=FvcILeg2vEQ

Code resource:

git clone https://github.com/Atlas7/demo-project
git clone https://github.com/nonken/static-demo.git
https://docs.c9.io/docs/

PaaS- Production Environment

CloudFront

By: Tamal Dey,
MCA, PESU

Agenda

01 What is AWS

02 Need For AWS CloudFront

03 What Is AWS CloudFront?

04 How Is Content Delivered?

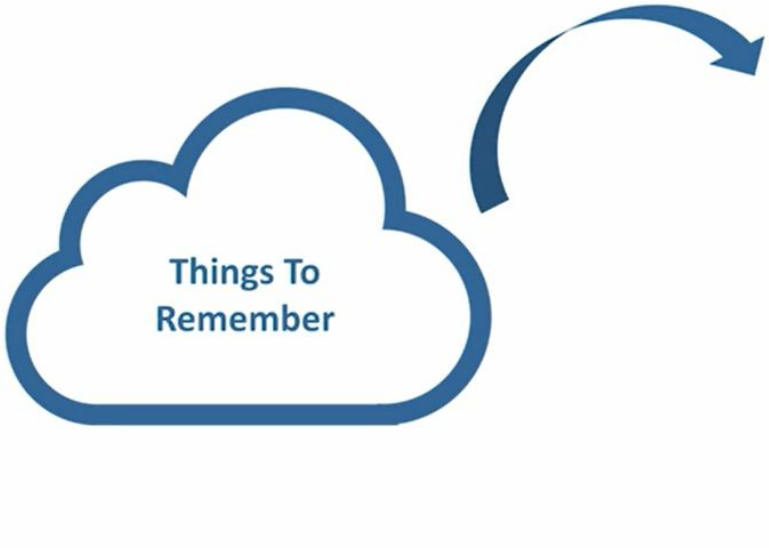
05 Applications

06 Demo: AWS CloudFront

What Is AWS?

Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.

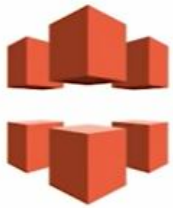
What Is AWS?



Need For AWS CloudFront

Scenario 1- Cloud Data Transmission

Scenario 2- CloudFront Transmission



What Is AWS CloudFront?

Amazon CloudFront is a web service that ***speeds up distribution*** of your static and dynamic web content, such as .html, .css, .js, and image files, to your users.

Key Notations:

How AWS CloudFront Delivers Content?

Applications/Benefits

- Accelerate Static Website Content Delivery
- Serve On-Demand or Live Streaming Video
- Encrypt Specific Fields Throughout System Processing
- Customize at the Edge
- Serve Private Content by using Lambda@Edge Customizations

Steps

Create a S3 bucket with restricting public access with standard bucket policy
Upload one (each) image, video and html file with read and write permission for owner

Go to CloudFront from Network & Content Delivery

Create Distribution

Step 1: Select delivery method

Choose web [For static content display]

Choose RTMP [For streaming live media]

Real-Time Messaging Protocol (**RTMP**)

Step 2: Create distribution

Settings Page1

Create Distribution

Origin Settings

Origin Domain Name cloudfronttoday.s3.amazonaws.com

Origin Path

Origin ID S3-cloudfronttoday

Restrict Bucket Access
- (•) Yes
- ( ) No

Origin Access Identity
- (•) Create a New Identity
- ( ) Use an Existing Identity

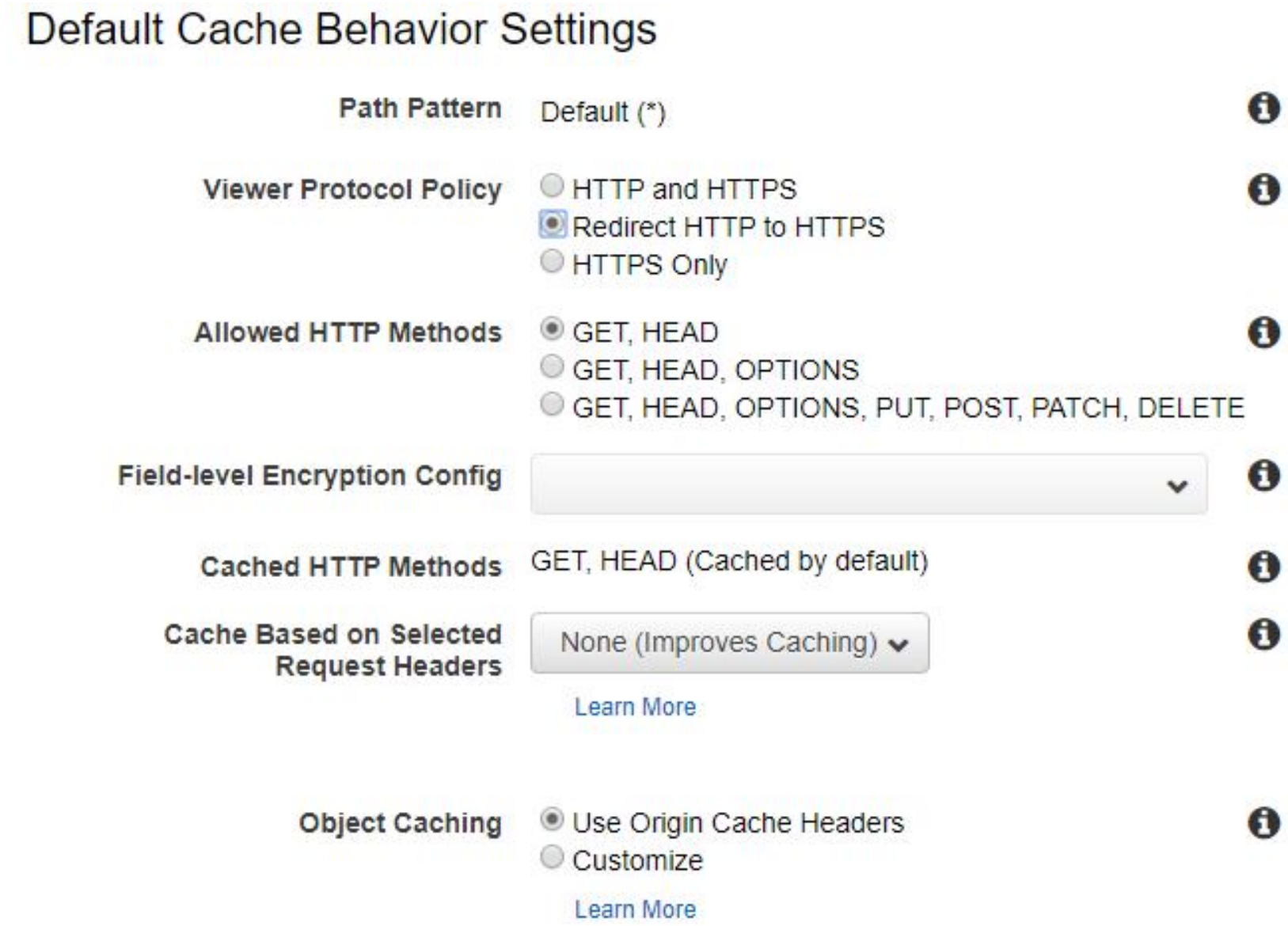
After (-)type Origin Domain Name cloudfronttoday.s3.amazonaws.com

Comment access-identity-

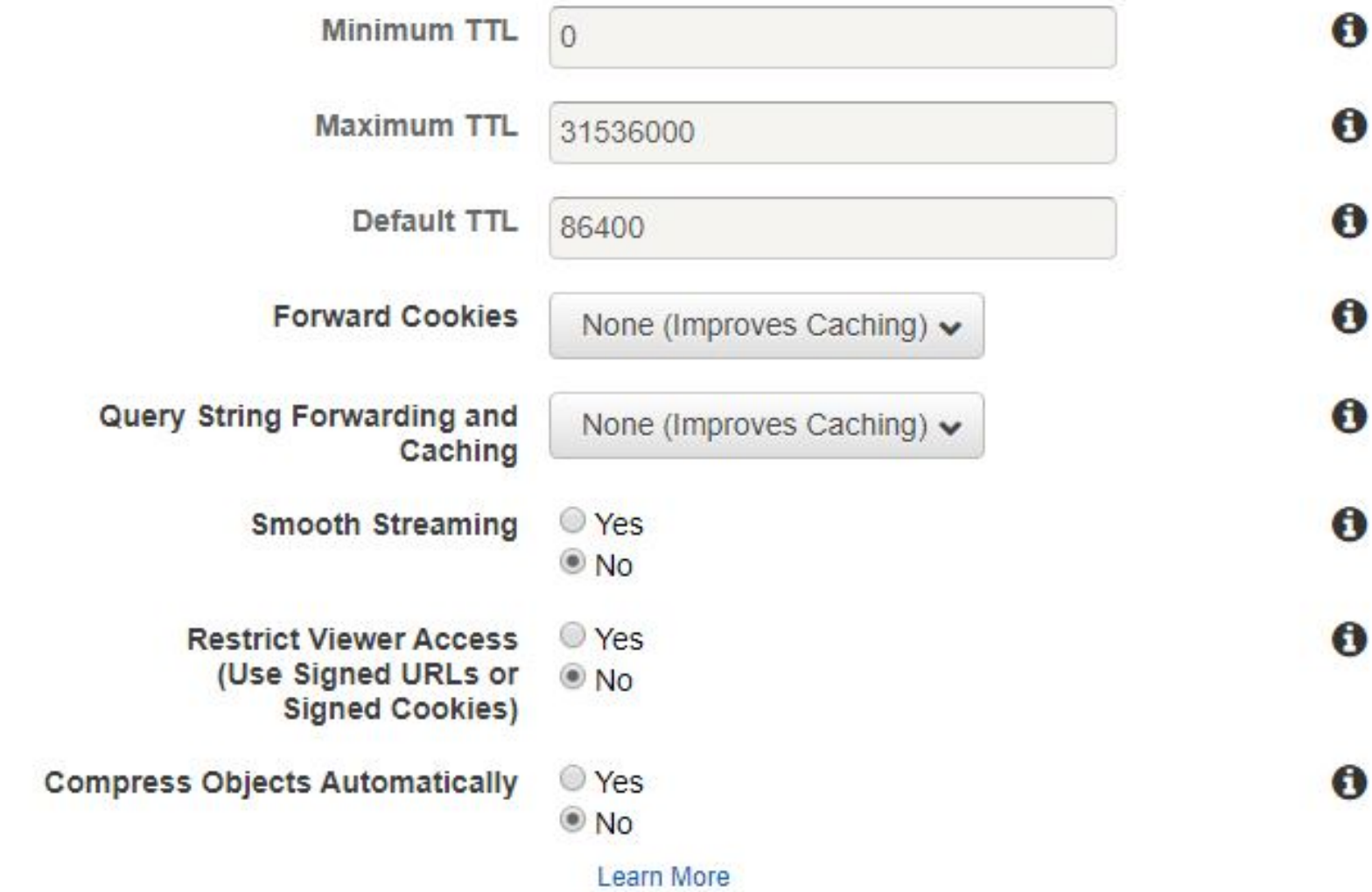
Grant Read Permissions on Bucket
- (•) Yes, Update Bucket Policy
- ( ) No, I Will Update Permissions

Origin Custom Headers | **Header Name** | **Value**

Settings Page2



Settings Page3



Settings Page4

Lambda Function Associations

CloudFront Event	Lambda Function ARN	Include Body
Select Event Type		

Learn More

Distribution Settings

Price Class Use All Edge Locations (Best Performance)

AWS WAF Web ACL None

Alternate Domain Names (CNAMEs)

SSL Certificate

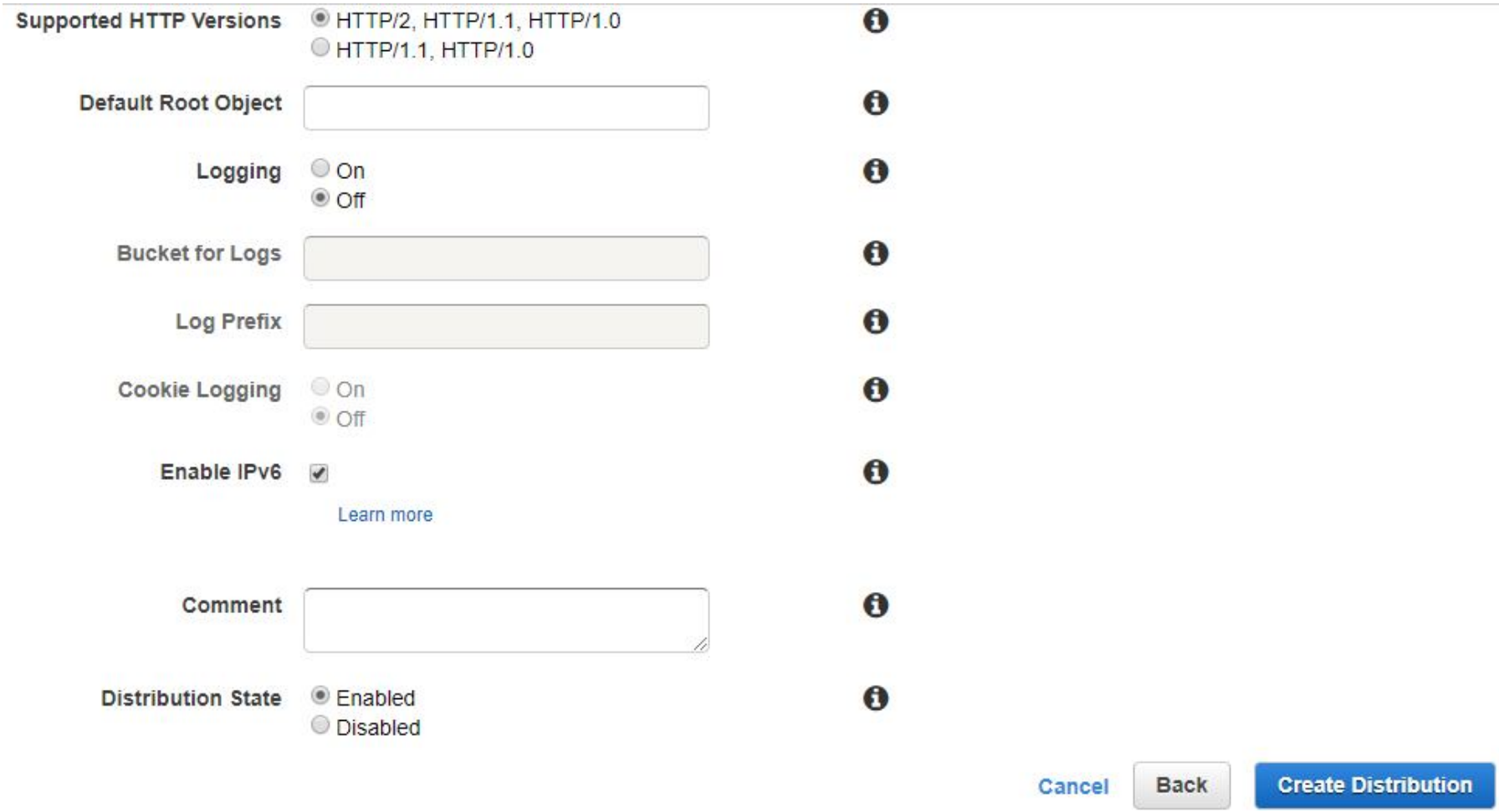
Default CloudFront Certificate (*.cloudfront.net)

Choose this option if you want your users to use HTTPS or HTTP to access your content with the CloudFront domain name (suc
https://d111111abcdef8.cloudfront.net/logo.jpg).
Important: If you choose this option, CloudFront requires that browsers or devices support TLSv1 or later to access your content

Custom SSL Certificate (example.com):

Choose this option if you want your users to access your content by using an alternate domain name, such as https://www.exam
You can use a certificate stored in AWS Certificate Manager (ACM) in the US East
(N. Virginia) Region, or you can use a certificate stored in IAM.

Settings Page5



Output Link

Click on Distribution [Top Left Corner]

Waiting Time [5 to 10 Minutes]

Click on DNS/filename

References

https://www.youtube.com/watch?v=sQNONcj0cvc
https://www.youtube.com/watch?v=AT-nHW3_SVI
https://www.youtube.com/watch?v=KIltfPRpTi4
https://www.youtube.com/watch?v=Ideptk0sJcM