Section 14 ) AWS Elastic Beanstalk

Machine generated alternative text:
Typical architecture: Web App 3-tier 
Route 53 
53 
ELB 
PUBLIC SUBNET 
) Stephane Maarek 
group 
Avail 
zone 
ability zone 2 
M5 
Availability zone 3 
PRIVATE SUBNET 
ElastiCache 
Store / retrieve 
session data 
+ Cached data 
Amazon RDS 
Read / write data 
DATA SUBNET 

**This we can get quickly with beanstalk**

Machine generated alternative text:
Developer problems on AWS 
• Managing infrastructure 
• Deploying Code 
• Configuring all the databases, 10 
• Scaling concerns 
cers, etc 
• Most web apps have the same architecture (ALB + ASG) 
• All the developers want is for their code to run! 
• Possibly, consistently across different applications and environments 

Machine generated alternative text:
AWS Elastic Beanstalk Overview 
• Elastic Beanstalk is a developer centric view of deploying 
an application on AWS 
• It uses all the component's we've seen before: 
EC2, ASG, ELB, RDS, etc.. 
• But it's all in one view that's easy to make sense OP. 
• We still have full control over the configuration 
• Beanstalk is free but you pay for the underlying instances 

Machine generated alternative text:
Elastic Beanstalk 
• Managed service 
• Instance configuration / OS is handled by Beanstalk 
• Deployment strategy is configurable but performed by Elastic Beanstalk 
• Just the application code is th -oqsi •lity of the developer 
• Three architecture models: 
• Single Instance deployment: good for dev 
• LB + ASG: great for production or pre-production web applications 
• ASG only: great for non-web apps in production (workers, etc..) 

Machine generated alternative text:
Elastic Beanstalk 
• Elastic Beanstalk has three components 
• Application 
• Application version: each deployment gets assigned a version 
• Environment name (dev, test, prod.. naming 
• You deploy application versions to 
versions to the next environment 
• Rollback feature to previous applic •o AIO 
• Full control over lifecycle of environ 
Create Application 
Upload Version 
(+ alias) 
Create 
Environments 
and can promote application 
Release to 
environments 

Machine generated alternative text:
Elastic Beanstalk 
• Support for many platforms: 
. Go 
. Java SE 
• Java with Tomcat 
• .NET on Windows Server with IIS 
Node.js 
• 
• Python 
• Ruby 
Packer Builder 
• 
• Single Container Docker 
• Multicontainer Docker 
• Preconfigured Docker 
• If not supported, you can write 
your custom platform (advanced) 

**Beanstalk demo**

Machine generated alternative text:
Compute 
AWS Elastic Beanstalk 
End-to-end web 
application management. 

**(very important everything will happen behind the scene)**

Machine generated alternative text:
Elastic Beanstalk 
Environments 
Applications 
x 
my- •rst-we app- eansta 
Up to 100 Unicode characters, not including forward slash 
Application tags 
Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the 
Feedback 
English (US) v 
resource and is case-sensitive. Learn more 
Key 
Add tag 
50 remaining 
Platform 
Platform 
Node.js 
Platform branch 
Node.js 12 running on 64bit Amazon Linux 2 
Platform version 
5.2.1 (Recommended) 
Remove tag 
@ 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights r 

**Note:**

**We can check in event log of elastic beanstalk created**

Machine generated alternative text:
Elastic Beanstalk 
Environments 
Applications 
my-first-webapp-beanstalk 
Application versions 
Saved configurations 
' MyFirstWebappBeanstalk-env 
Configuration 
Logs 
Health 
Monitoring 
Alarms 
Managed updates 
Events 
Tags 
x 
) Environments ) MyFirstWebappBeanstalk-env 
Elastic Beanstalk 
Elastic Beanstalk is launching your environment. 
Events 
Events 
Severity 
Time 
INFO 
Type 
Details 
Created CloudWatch alarm named: awseb-e-bapfctm5k9-stack-AWSEBCloudwatchAlarmLow-1 A735Z44GTR85 
Created CloudWatch alarm named: awseb-e-bapfctm5k9-stack-AWSEBCloudwatchAlarmHigh-1 TMI MlJ80YF1 K 
Created Auto Scaling group policy named: arn:aws:autoscaling:eu-west-3:398678778168:scalingPolicy:c6dac02b-c1 
4a2f-9114-79967 e312271 :autoScalingGroupName/awseb-e-bapfctm5k9-stack-AWSEBAutoScal ingGroup- 
1931 HB61 FRCQZ:p01icyName/awseb-e-bapfctm5k9-stack-AWSEBAutoScalingScaleDown 153J2CW 
Created Auto Scaling group policy named: arn:aws:autoscaling:eu-west-3:398678778168:scalingPolicy:9c52de79-5a 
4005-9e7b-1 a13cb2b8095:autoScalingGroupName/awseb-e-bapfctm5k9-stack-AWSEBAutoScalingGroup- 
1931 HB61 FRCQZ:p01icyName/awseb-e-bapfctm5k9-stack-AWSEBAutoScalingScaleUpPolicy-1 OE5CWC9CHZPJ 
Waiting for EC2 instances to launch. This may take a few minutes. 
Created Auto Scaling group named: awseb-e-bapfctm5k9-stack-AWSEBAutoScalingGroup-1931 HB61 FRCQZ 
Environment health has transitioned to Pending. Initialization in progress (running for 36 seconds). There are no inst 
2020-10-07 
INFO 
UTC+0530 
2020-10-07 
INFO 
UTC+0530 
2020-10-07 
INFO 
UTC+0530 
2020-10-07 
INFO 
UTC+0530 
2020-10-07 
INFO 
UTC+0530 
2020-10-07 
INFO 
UTC+0530 
2020-10-07 00:41 
INFO 

Machine generated alternative text:
S3 buckets 
Q elasticbeanstalk-eu-west-3-3986787781$ 
Empty 
Delete 
Access O 
Objects can be public 
All access types 
1 
Buckets 
Region 
EU (Paris) 
Discover the console 
1 
Regions 
Date created 
Oct 7, 2020 
AM 
GMT+0530 
+ Create bucket 
Bucket name 
Edit public access settings 
El 
elasticbeanstalk-eu-west-3-398678778168 

Machine generated alternative text:
Elastic Beanstalk 
Environments 
Applications 
v my-first-webapp-beanstalk 
Application versions 
Saved configurations 
v Recent environments 
MyFirstWebappBeanstalk-env 
x 
Elastic Beanstalk 
Applications 
my-first-webapp-beanstalk 
URL 
MyFirstWebappBeanstalk- 
env.eba-2zv3t3mp.eu- 
west- 
3.elasticbeanstalk.com 
Running 
versions 
Sample 
Application 
Platform v 
Node.js 12 
running on 
64bit 
Amazon 
Linux 2 
Actions A 
Create environment 
Manage tags 
Delete application 
Restore terminated environment 
Swap environment URLs 
Application 'my-first-webapp-beanstalk' environments 
Q Filter results matching the display values 
Last 
modified v 
2020-10-07 
UTC+0530 
Environment name 
MyFirstWebappBeanstalk- 
env 
Health v 
0k 
Date 
created v 
2020-10- 
07 
UTC+0530 
state 
Supported 
name 
WebServer 

**Beanstalk second environment**

**Note**

**we have one environment but want to create one more environment**

Machine generated alternative text:
Elastic Beanstalk 
Environments 
Applications 
v my-first-webapp-beanstalk 
Application versions 
Saved configurations 
v Recent environments 
MyFirstWebappBeanstalk-env 
x 
Elastic Beanstalk 
Applications 
my-first-webapp-beanstalk 
URL 
MyFirstWebappBeanstalk- 
env.eba-2zv3t3mp.eu- 
west- 
3.elasticbeanstalk.com 
Running 
versions 
Sample 
Application 
Platform v 
Node.js 12 
running on 
64bit 
Amazon 
Linux 2 
Actions V 
Create a new environment 
Application 'my-first-webapp-beanstalk' environments 
Q Filter results matching the display values 
Platform 
state 
Supported 
Tier 
name 
WebServer 
Last 
modified v 
2020-10-07 
UTC+0530 
Environment name 
MyFirstWebappBeanstalk- 
env 
Date 
Health v 
created v 
2020-10- 
07 
UTC+0530 

Machine generated alternative text:
Elastic Beanstalk 
Environments 
Applications 
my-first-webapp-beanstalk 
Application versions 
Saved configurations 
Recent environments 
MyFirstWebappBeanstalk-env 
x 
Platform 
Node.js 
Platform branch 
Node.js 12 running on 64bit Amazon Linux 2 
Platform version 
5.2.1 (Recommended) 
Application code 
O Sample application 
Get started right away with sample code. 
Existing version 
Application versions that you have uploaded for my-first-webapp-beanstalk. 
- Choose a version 
O 
Upload your code 
Upload a source bundle from your computer or copy one from Amazon S3. 
Canc 
Configure more options 
Create environment 
Environment 
creation wizard 
Provide details that Elastic 
Beanstalk needs to create an 
environment. 
An environment is a collection c 
AWS resources running a versio 
of your application code. When 
you create an environment, 
Elastic Beanstalk provisions the 
resources needed to run the 
application version you 
specified. 
Key details you provide here: 
• Environment name 
• Domain —A subdomain for 
accessing your web 
application. 
• Platform —A managed 
platform or your own custor 
platform. 
• Application code— New 
uploaded code, an exiting 
application version, or an 
example application. 
• Configuration options — 

**Elastic beanstalk deployment modess**

Machine generated alternative text:
Elastic Beanstalk Deployment Modes 
Single Instance 
Great for dev 
Elastic IP 
51 
High Availability with Load Balancer 
Great for prod 

Machine generated alternative text:
Beanstalk Deployment Options for Updates 
• All at once (deploy all in one go) — fastest, but instances aren't available to 
serve traffic for a bit (downtime) 
• Rolling: update a few instances atim - bucket), and then move onto the 
next bucket once the first bu 
• Rolling with additional batche ike'+ålip , but spins up new instances to 
move the batch (so that the old 
•Cation is still available) 
• Immutable: spins up new instances in a new ASG, deploys version to these 
instances, and then swaps all the instances when everything is healthy 

Machine generated alternative text:
Elastic Beanstalk Deployment 
All at once 
• Fastest deployment 
• Application has downtime 
• Great for quick iterations in 
development environment 
• No additional cost 
v2 
v2 

Machine generated alternative text:
Elastic Beanstalk Deployment 
v2 
v2 
v2 
Rolling 
• Application is 
running below 
capacity 
• Can set the 
bucket size 
• Application is 
running both 
versions 
simultaneously 
• No additional 
le have written a note here. 
VI 
v2 
v2 

Machine generated alternative text:
Elastic Beanstalk Deployment 
Rolling with additional batches 
v2 
v2 
v2 
v2 
v2 
• Application is running 
at capacity 
• Can set the bucket 
Size 
• Application is running 
both versions 
simultaneously 
• Small additional cost 
• Additional batch is 
removed at the end 
of the deployment 
• Longer deployment 
• Good for prod 
VI 
VI 

Machine generated alternative text:
Elastic Beanstalk Deployment 
Immutable 
Current ASG CurrentASG Current ASG Current ASG 
• Zero downtime 
• New Code is deployed to new 
instances on a temporary ASG 
• High cost, double capacity 
• Longest deployment 
• Quick rollback in case of failures 
(just terminate new ASG) 
• Great for prod 
WS Certified Developer C Stephane Maarek 
Temp ASG 
v2 
Temp ASG 

Machine generated alternative text:
Elastic Beanstalk Deployment 
Blue / Green 
• Not a "direct feature" of Elastic Beanstalk 
• Zero downtime and release facility 
• Create a new "stage" environment and 
deploy v2 there 
• The new environment (green) can be 
validated independently and roll back if 
Issues 
• Route 53 can be setup using weighted 
policies to redirect a little bit of traffc to 
the stage environment 
• Using Beanstalk, "swap URLs" when done 
with the environment test 
AWS Certified Developer C Stephane Maarek 

Machine generated alternative text:
Elastic Beanstalk Deployment Summary 
from AWS Doc 
• https://docs.aws.amazon.com/elasticbeanstalldlatest/dg/using- 
features.deploy-existing-version.html 
Deployed TO 
Existing instances 
New and existing 
New instances 
Zero 
NO DNS 
Change 
Rollback ProcEs 
Manual 
Manual Redeploy 
swap URL 
All at 
Rolling 
Rolling with 
batch 
Impct Of Failed 
batch out of service; ary wccessful batctæs to 
running new version 
Minfrnal if first batch fans. similar to 
eee 
see 

Machine generated alternative text:
Elastic Beanstalk CLI 
• We can install an additional CLI called the "EB cli" which makes working with 
Beanstalk from the CLI easier 
• Basic commands are: 
• eb create 
• eb status 
• eb health 
• eb events 
. eb logs 
• eb open 
• eb deploy 
• eb config 
• eb terminate 
• It's helpful for your automated deployment pipelines! 

Machine generated alternative text:
Elastic Beanstalk Deployment Process 
• Describe dependencies 
(requirements.txt for Python, package.json for Node.js) 
• Package code as zip, and describe dependencies 
• Python: requirements.txt 
• Node.js: package.json 
• Console: upload zip file (creates new app version), and then deploy 
• CLI: create new app version using CLI (uploads zip), and then deploy 
• Elastic Beanstalk will deploy the zip on each EC2 instance, resolve 
dependencies and start the application 

Machine generated alternative text:
Beanstalk Lifecycle Policy 
• Elastic Beanstalk can store at most 1000 application versions 
• If you don't remove old versions, you won't be able to deploy anymore 
se a lifecycle policy 
• To phase out old application ve 
• Based on time (old versions a re 
• Based on space (when you h 
AH Y rsions) 
• Versions that are currently use on' e deleted 
• Option not to delete the source bundle in S3 to prevent data loss 

**Beanstalk lifecycle policy**

**Notes :**

**Whatever files we will upload in beanstalk that will save in s3**

**Step 1) click on setting**

Machine generated alternative text:
Elastic Beanstalk 
Environments 
Applications 
v my-first-webapp-beanstalk 
Application versions 
Saved configurations 
Recent environments 
MyFirstWebappBeanstalkprod 
MyFirstWebappBeanstalk-env 
x 
Elastic Beanstalk 
Applications 
Application versions 
Version label 
blue version 
Sample Application 
my-first-webapp-beanstalk 
Application versions 
Actions 
Source 
2020280XEm-nodejs-v2-blue.zip 
Sample Application 
Settings 
Upload 
C Refresh 
Description 
Date created 
2020-10-07T01 :38: 1 1 +05:30 
Deployed to 
MyFirstWebappBeanstalkprod 
MyFirstWebappBeanstalk-env 

**Step 2) enable lifecycle policy**

Machine generated alternative text:
Application version lifecycle settings 
to retain for future —oy•rTV1ts. This polo will déte 
cwreltly in 
of being Learn rnore 
Lifecy& 
LifEy& rule 
IBO 
Delete btnde from SS 
mitten a note here. 
x 

**Beanstalk extension**

Machine generated alternative text:
Elastic Beanstalk Extensions 
• A zip file containing our code must be deployed to Elastic Beanstalk 
• All the parameters set in the UI can be configured with code using files 
• Requirements: 
• in the .ebextensions/ directory in the root of source code 
• YAML / JSON format 
• .config extensions (example: logging.confg) 
• Able to modify some default settings using: option_settings 
• Ability to add resources such as RDS, ElastiCache, DynamoDB, etc... 
• Resources managed by .ebextensions get deleted if the environment goes away 

**Note : if we want setup with external db then we need to setup in this way**

Machine generated alternative text:
Ultimate AWS Certified Developer Associate 2020 - NEW! 
O environment-variables.config x 
OPEN EDITORS 
aws-cicd 
v beanstalk 
v nodejs-v3-ebextensions 
v .ebextensions 
O environment-variables.config 
app.js 
! cron.yaml 
o index_html 
package.json 
O environment-variables. config 
Icon 
nodejs-v2-blue.zip 
nodejs-v3-ebextensions.zip 
cli 
OUTLINE 
No timeline information was 
provided 
z•bextensions O environment-variablesconfig option_settings ) D3_USER 
4 
6 
8 
9 
12 
13 
14 
15 
16 
18 
You must place this file in .ebextensions 
g And must have a . config file 
So the file is at 
Note: Even though the narkup Language is Y*L, you must use . config extension 
opt ion_sett ings : 
g 
see: https://docs.aws.a.azon. 
ions -qenera I —e last ic bean s ta Ikapp licat ionenv iron—nt 
aWS : e t icbeansta I k : app 1 ication : env i ronent : 
DB_IRLi "jdbc; 
DB_USER: use 
This format 
option_setting • 
— næespace: 
opt ion_ 
value: Opt value 
See: https://docs.aws. 
html 
n.com/elast t inqs. 
Ln 10, Col 22 
Spaces: 2 
UTE-8 
YAML 

**Once we upload it then environment properties will automatically set up**

Machine generated alternative text:
Elastic 
Beanstalk 
v my-first-webapp- 
beanstalk 
Application versions 
Saved configurations 
v MyFirstWebappBeanstal 
GO to environment 
Monitoring 
Resource 
days 
Kep logs after terminating ewircgvne.t 
Environment properties 
The folbwmg ttw 

**Beanstalk and cloud formation**

Machine generated alternative text:
Elastic Beanstalk Under the Hood 
• Under the hood, Elastic Beanstalk relies on CloudFormation 
• CloudFormation is used to provision other AWS services (we'll see later) 
Elastic Beanstalk 
CloudFormation 
• Use case: you can define CloudFormation resources in your .ebextensions 
to provision ElastiCache, an S3 bucket, anything you want! 

**Cloudformation will tell everything**

**Note : cloudformation use in elastic beanstalk behind the scene**

Machine generated alternative text:
CloudFormation 
El Stacks (2) 
Stacks 
awseb-e-bapfctm5k9-stack 
awseb-e-bapfctm5k9-stack 
Parameters 
Delete 
Change sets 
Update 
Stack actions v 
Status reason 
Create stack 
c 
Stack info 
c 
Events 
Resources 
Physical ID 
Outputs 
Template 
Q Filter by stack name 
View nested 
Active V 
awseb-e-3mtupuhh34-stack 
2020-10-07 UTC*0530 
@ UPDATE COMPLETE 
awseb-e-bapfctm5k9-stack 
2020-10-07 00:41 UTC+0530 
@ UPDATE COMPLETE 
O 
o 
Status 
UPDATE COMPLETE 
CREATE COMPLETE 
CREATE COMPLETE 
Resources (16) 
Q 
Search resources 
Logical ID 
AWSEBAutoSca1ingGroup 
AWSEBAutoScalingLaunchCo 
nfiguration 
AWSEBAutoSca1ingScaleDow 
nPolicy 
Type 
AWS::AutoScaling• 
.:Aut 
oScalingGroup 
AWS::AutoScaling• 
.:Lau 
nchConfiguration 
AWS::AutoScaling::Scal 
ingPolicy 
awseb-e-bapfctm5k9-stack- 
AWSEBAutoSca1ingGroup- 
1931HB61FRCQZ e 
awseb-e-bapfctm5k9-stack- 
AWSEBAutoScalingLaunchConf 
iguration-109GB35BCV4DV e 
arn:aws:autoscaling:eu-west- 
c6dac02b-c133-4a2f-9114- 
79967e312271:autoScalingGro 
upName/awseb-e- 
bapfctm5k9-stack- 
AWSEBAutoSca1ingGroup- 
1931HB61FRCQZ:policyName/ 

**Note :**

**Using clouformation we can deploy s3,dynamo db ,elasticache**

**Beanstalk cloning**

Machine generated alternative text:
Elastic Beanstalk Cloning 
• Clone an environment with the exact same configuration 
• Useful for deploying a 'test" version ofyour application 
• All resources and configurati 
• Load Balancer type and config 
• RDS database type (but the d 
• Environment variables 
-'Ares ed: 
cloning 
served) 
not 
• After cloning an environment, you can change settings 

**Beanstalk migration**

Machine generated alternative text:
Elastic Beanstalk Migration: Load Balancer 
• After creating an Elastic Beanstalk 
environment, you cannot change 
the Elastic Load Balancer type 
(only the configuration) 
. To migrate: 
Beanstalk old 
Route 53 
or CNAME swap 
anstalk new 
ALB 
2. 
3. 
create a new environment with 
same confi uration except LB 
(can't clone 
deploy your application onto the 
new environment 
perform a CNAME swap or Route 
53 u date 

\

Machine generated alternative text:
RDS with Elastic Beanstalk 
• RDS can be provisioned with Beanstalk, which is great for dev / test 
• This is not great for prod as the database lifecycle is tied to the 
Beanstalk environment 
• The best for prod is to sepa Eely RDS database and provide 
our EB application with the c nn 
Beanstalk with R 
Amazon 
written a note here. 

Machine generated alternative text:
Elastic Beanstalk Migration: Decouple RDS 
2. 
3. 
4. 
5. 
Create a snapshot of RDS DB (as a 
safeguard) 
Go to the RDS console and protect 
the RDS database from deletion 
Create a new Elastic Beanstalk 
environment, without RDS, point your 
application to existing RDS 
perform a CNAME swap (blue/green) 
or Route 53 update, confirm working 
Terminate the old environment (RDS 
wont be deleted) 
Amazon 
Route 53 F--t 
or CNAME swap QD 
Beanstalk new 
Ive written a note here. 

**Elastic beanstalk with docker**

Machine generated alternative text:
Elastic Beanstalk — Single Docker 
• Run your application as a single docker container 
• Either provide: 
• Dockerflle: Elastic Beanstalk will bHild-aArun the Docker container 
• Dockerrun.aws.json (VI): Describe Iready built* Docker image is 
. Image 
• ports 
. Volurr-es 
. Etc... 
• Beanstalk in Single Docker Container does not use ECS 

Machine generated alternative text:
Elastic Beanstalk — Multi Docker Container 
• Multi Docker helps run multiple containers per EC2 instance in EB 
• This will create for you: 
• ECS Cluster 
• EC2 instances, configured to use the ECS Cluster 
• Load Balancer (in high availability mode) 
• Task definitions and execution 
• Requires a config Dockerrun.aws.json (v2) at the root of source code 
• Dockerrun.aws.json is used to generate the ECS task definition 
• Your Docker images must be pre-built and stored in ECR for example 

Machine generated alternative text:
Elastic Beanstalk + Multi Docker ECS 
Elastic Beanstalk Environment 
ECS Cluster + ASG 
stance 
php 
ntainer 
nglnx 
Container 
other 
Container 
EC2 
Instance 
php 
Container 
nglnx 
Container 
other 
Container 
Load Balancer 
beanstalk-url:80 
rt 
beanstalk-url:1234 
Port 1234 
written a note here. 

**Hands on**

**Dockerrun.aws.json contain the ecs task defintion for us**

Machine generated alternative text:
x Dockerrun.aws.json 
DOCKER-MULTEO... 
) ebextensions 
v php-app 
index.php 
scheduled.php 
o static.html 
v proxy / conf.d 
O default.conf 
! cron_ a 
errun.aws_json 
O 
3 
2 
5 
6 
9 
12 
13 
15 
16 
17 
2ø 
22 
23 
24 
27 
"AWSEBDockerrunVersion 
"volumes": I 
" name" : "php—app" , 
"host": 
" sourcePath": "var/app/current/php—app" 
"name" : "nginx 
"host": 
sourcePa 
" container-Def in i 
"name": "php—app" , 
" : "php: fpm", 
"essential": true, 
: 128, 
"nountP0ints": 
" sourceVolune": "php—app", 
"containerPath": 
" readOn y" : t rue 
t proxy/conf. d" 

**Beanstalk advanced concept**

Machine generated alternative text:
Elastic Beanstalk and HTTPS 
• Beanstalk with HTTPS 
• Idea: Load the SSL certificate onto the Load Balancer 
• Can be done from the Console (EB console, load balancer configuration) 
• Can be done from the code: .ebextensions/securelistener-alb.config 
• SSL Certificate can be provisioned using ACM (AWS Certificate Manager) or CLI 
• Must configure a security group rule to allow incoming port 443 (HTTPS port) 
• Beanstalk redirect HITP to HITPS 
• Configure your instances to redirect HTTP to HTTPS: 
s:// ithub.com/awsdocs/elastic-beanstalk-sam les/tree/master/confi uration-files/aws- 
n 
• OR configure the Application Load Balancer (ALB only) with a rule 
• Make sure health checks are not redirected (so they keep giving 200 OK) 

Machine generated alternative text:
Web Server vs Worker Environment 
• If your application performs tasks that are long to complete, offload these tasks to a dedicated 
worker environment 
• Decoupling your application into two tiers is common 
• Example: processing a video, generating a zip file, etc 
• You can define periodic tasks in a file cron. 
Worker Tier SQS + EC2 
Web Tier ELB + EC2 
requeS 
ALB 
Auto Scaling Group 
PUT 
Auto Scaling Group 
SQS Queue 

Machine generated alternative text:
Elastic Beanstalk — Custom Platform (Advanced) 
• Custom Platforms are very advanced, they allow to define from scratch: 
• The Operating System (OS) 
• Additional Software 
• Scripts that Beanstalk n-lns on these platforms 
• Use case: app language is incompatible with Beanstalk & doesn't use Docker 
• To create your own platform: 
• Define an AMI using Platformyaml file 
• Build that platform using the Packer software (open source tool to create AMIs) 
• Custom Platform vs Custom Image (AMI): 
• Custom Image is to tweak an existing Beanstalk Platform (Python, Node.js, Java... ) 
• Custom Platform is to create an entirely new Beanstalk Platform 