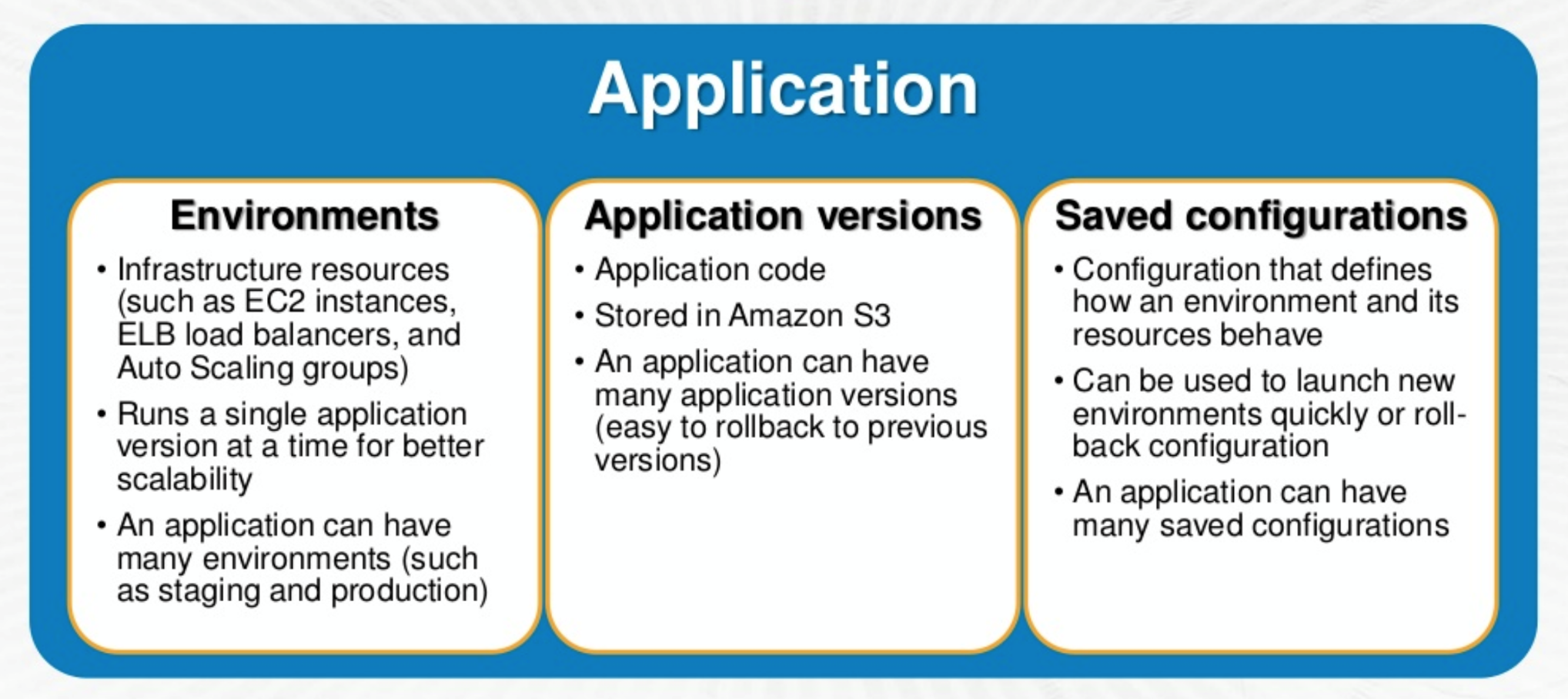
AWS Elastic Beanstalk

# AWS Elastic Beanstalk

* AWS Elastic Beanstalk helps to quickly deploy and manage applications in the AWS Cloud without having to worry about the infrastructure that runs those applications.
* Elastic Beanstalk reduces management complexity without restricting choice or control.
* Elastic Beanstalk enables automated infrastructure management and code deployment, by simply uploading, for applications and includes
  + Application platform management
  + Capacity provisioning
  + Load Balancing
  + Auto scaling
  + Code deployment
  + Health Monitoring
* Once an application is uploaded, Elastic Beanstalk automatically launches an environment, creates and configures the AWS resources needed to run the code. After your environment is launched, it can be managed and used to deploy new application versions
* AWS resources launched by Elastic Beanstalk are fully accessible i.e. EC2 instances can be sshed into
* Elastic Beanstalk provides developers and systems administrators an easy, fast way to deploy and manage the applications without having to worry about AWS infrastructure.
* CloudFormation, using templates, is a better option if the internal AWS resources to be used are known and fine grained control is needed

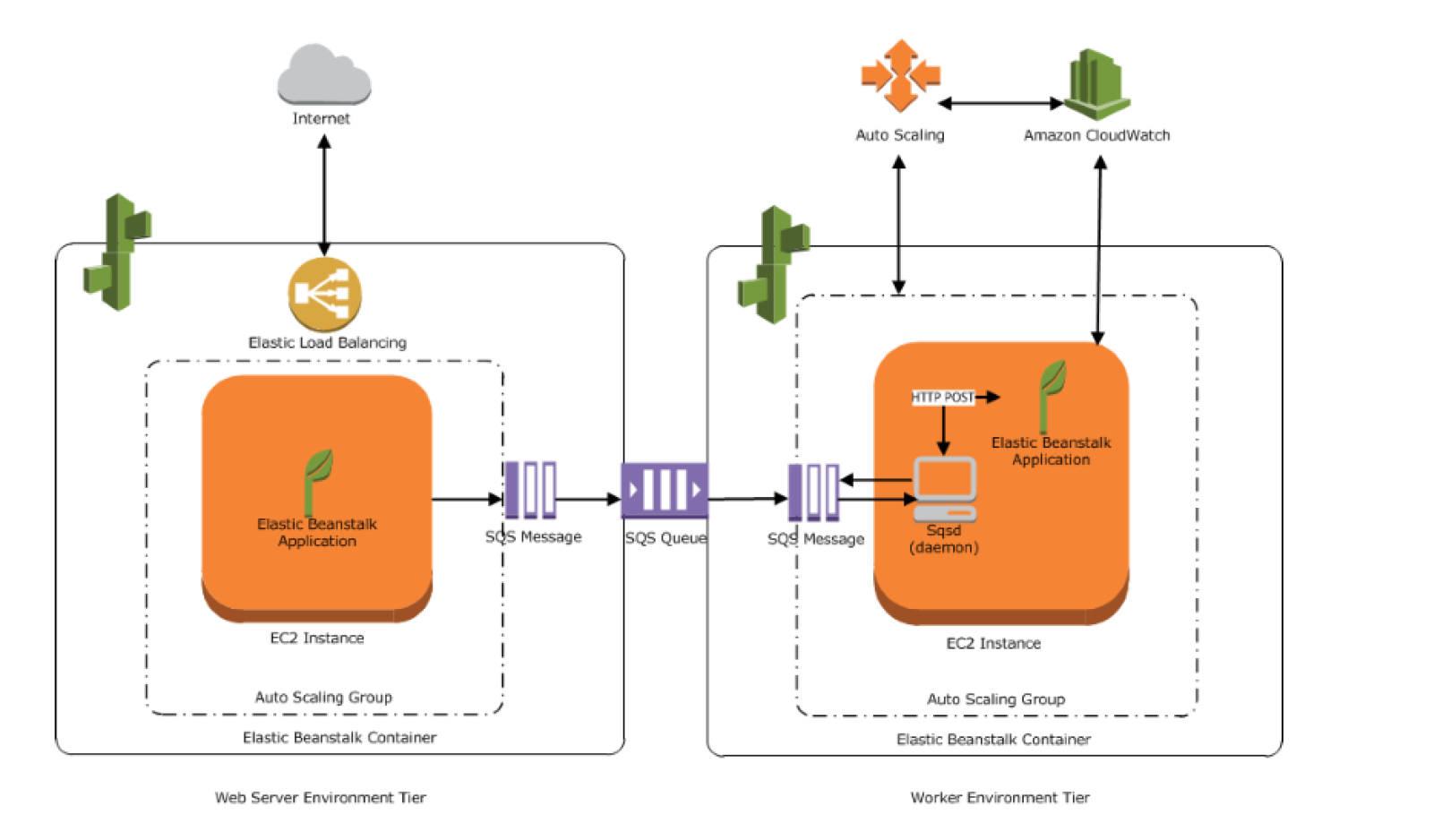
## Elastic Beanstalk Components



* **Application**
  + An Elastic Beanstalk application is a **logical collection** of Elastic Beanstalk components, including environments, versions, and environment configurations.
* **Application Version**
  + An application version refers to a specific, labeled iteration of deployable code for a web application
  + Applications can have many versions and each application version is unique and points to an S3 object
  + Multiple versions can be deployed for an Application for testing differences and helps rollback to any version if case of issues
* **Environment**
  + An environment is a version that is deployed onto AWS resources
  + An environment runs a single application version at a time, but same application version can be deployed across multiple environments
  + When an environment is created, Elastic Beanstalk provisions the resources needed to run the application version you specified.
* **Environment Configuration**
  + An environment configuration identifies a collection of parameters and settings that define how an environment and its associated resources behave
  + When an environment’s configuration settings is updated, Elastic Beanstalk automatically applies the changes to existing resources or deletes and deploys new resources, depending upon the change
* **Configuration Template**
  + A configuration template is a starting point for creating unique environment configurations

## 

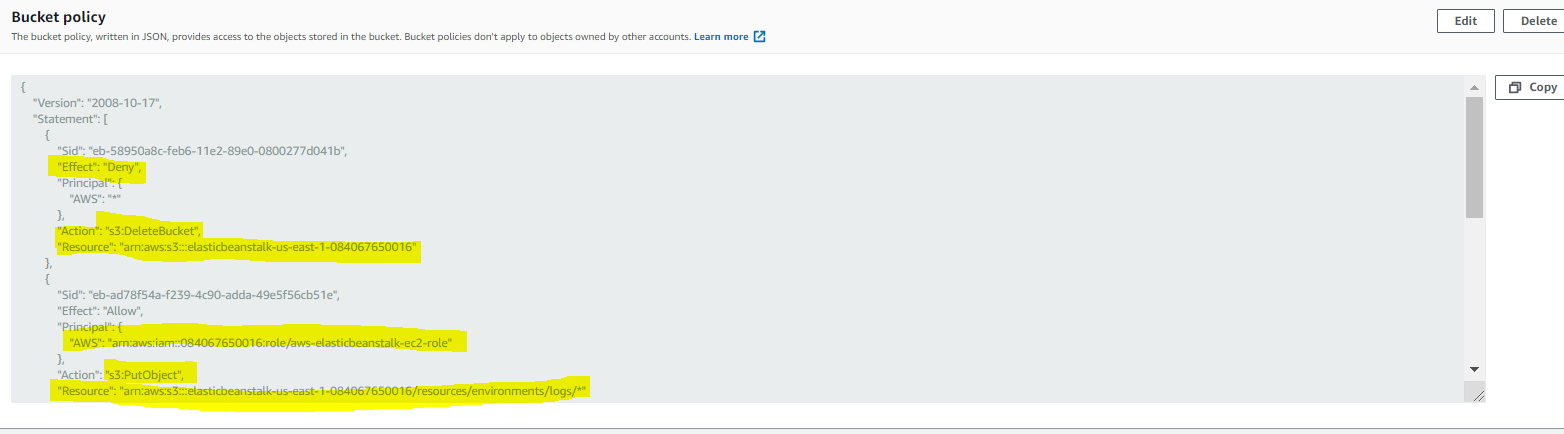
## Elastic Beanstalk Architecture



* Elastic Beanstalk environment requires an **environment tier, platform, and  
  environment type**
* Environment tier determines whether Elastic Beanstalk provisions resources to support a web application that handles HTTP(S) requests or a web application that handles background-processing tasks
* **Web Environment Tier**
  + An environment tier whose web application processes web requests is known as a web server tier.
  + AWS resources created for a web environment tier include a Elastic Load Balancer, an Auto Scaling group, one or more EC2 instances
  + Every Environment has a CNAME url pointing to the ELB, aliased in Route 53 to ELB url
  + Each EC2 server instance that runs the application uses a container type, which defines the infrastructure topology and software stack
  + A software component called the **host manager (HM) runs on each EC2 server** instance and is responsible for
    - Deploying the application
    - Aggregating events and metrics for retrieval via the console, the API, or the command line
    - Generating instance-level events
    - Monitoring the application log files for critical errors
    - Monitoring the application server
    - Patching instance components
    - Rotating your application’s log files and publishing them to S3
* **Worker Environment Tier**
  + An environment tier whose web application runs background jobs is known as a worker tier
  + AWS resources created for a worker environment tier include an Auto Scaling group, one or more Amazon EC2 instances, and an IAM role.
  + **For the worker environment tier, Elastic Beanstalk also creates and provisions an SQS queue, if one doesn’t exist**
  + When a worker environment tier is launched, Elastic Beanstalk installs the necessary support files for the programming language of choice and a daemon on each EC2 instance in the Auto Scaling group reading from the same SQS queue
  + Daemon is responsible for pulling requests from an SQS queue and then sending the data to the web application running in the worker environment tier that will process those messages
  + **Elastic Beanstalk worker environments support SQS dead letter queues which can be used to store messages that could not be successfully processed. Dead letter queue provides the ability to sideline, isolate and analyze the unsuccessfully processed messages**
* **One environment cannot support two different environment tiers** because each requires its own set of resources; a worker environment tier and a web server environment tier each require an Auto Scaling group, but Elastic Beanstalk supports only one Auto Scaling group per environment.

## Elastic Beanstalk with other AWS Services

* Elastic Beanstalk supports VPC and launches AWS resources, such as instances, into the VPC
* Elastic Beanstalk supports IAM and helps you securely control access to your AWS resources.
* CloudFront can be used to distribute the content in S3, after an Elastic Beanstalk is created and deployed
* CloudTrail
  + Elastic Beanstalk is integrated with CloudTrail, a service that captures all of the Elastic BeanstalkAPI calls and delivers the log files to an S3 bucket that you specify.
  + CloudTrail captures API calls from the Elastic Beanstalk console or from your code to the Elastic Beanstalk APIs and help to determine the request made to Elastic Beanstalk, the source IP address from which the request was made, who made the request, when it was made etc.
* RDS
  + Elastic Beanstalk provides support for running RDS instances in the Elastic Beanstalk environment **which is ideal for development and testing but not for production.**
  + **For a production environment, it is not recommended because it ties the lifecycle of the database instance to the lifecycle of application’s environment. So it the Elastic beanstalk environment is deleted, the RDS instance is deleted as well**
  + It is recommended to launch a database instance outside of the environment and configure the application to connect to it outside of the functionality provided by Elastic Beanstalk.
  + Using a database instance external to your environment requires additional security group and connection string configuration, but it also lets the application connect to the database from multiple environments, use database types not supported with integrated databases, perform blue/green deployments, and tear down your environment without affecting the database instance.
* S3
  + Elastic Beanstalk creates an S3 bucket named elasticbeanstalk-region-account-id for each region in which environments is created.
  + Elastic Beanstalk uses the bucket to store application versions, logs, and other supporting files.
  + It applies a bucket policy to buckets it creates to allow environments to write to the bucket and prevent accidental deletion



## Elastic Beanstalk Deployment Strategies

Refer to post [AWS Elastic Beanstalk Deployment Strategies](https://jayendrapatil.com/aws-elastic-beanstalk-deployment-strategies/)

## AWS Certification Exam Practice Questions

1. An organization is planning to use AWS for their production roll out. The organization wants to implement automation for deployment such that it will automatically create a LAMP stack, download the latest PHP installable from S3 and setup the ELB. Which of the below mentioned AWS services meets the requirement for making an orderly deployment of the software?
   1. **AWS Elastic Beanstalk**
   2. AWS CloudFront
   3. AWS CloudFormation
   4. AWS DevOps
2. What does Amazon Elastic Beanstalk provide?
   1. A scalable storage appliance on top of Amazon Web Services.
   2. **An application container on top of Amazon Web Services**
   3. A service by this name doesn’t exist.
   4. A scalable cluster of EC2 instances
3. You want to have multiple versions of your application running at the same time, with all versions launched via AWS Elastic Beanstalk. Is this possible?
   1. However if you have 2 AWS accounts this can be done
   2. AWS Elastic Beanstalk is not designed to support multiple running environments
   3. **AWS Elastic Beanstalk is designed to support a number of multiple running environments**
   4. However AWS Elastic Beanstalk is designed to support only 2 multiple running environments
4. A .NET application that you manage is running in Elastic Beanstalk. Your developers tell you they will need access to application log files to debug issues that arise. The infrastructure will scale up and down. How can you ensure the developers will be able to access only the log files?
   1. Access the log files directly from Elastic Beanstalk
   2. **Enable log file rotation to S3 within the Elastic Beanstalk configuration**
   3. Ask your developers to enable log file rotation in the applications web.config file
   4. Connect to each Instance launched by Elastic Beanstalk and create a Windows Scheduled task to rotate the log files to S3
5. Your team has a tomcat-based Java application you need to deploy into development, test and production environments. After some research, you opt to use Elastic Beanstalk due to its tight integration with your developer tools and RDS due to its ease of management. Your QA team lead points out that you need to roll a sanitized set of production data into your environment on a nightly basis. Similarly, other software teams in your org want access to that same restored data via their EC2 instances in your VPC .The optimal setup for persistence and security that meets the above requirements would be the following. **[PROFESSIONAL]**
   1. Create your RDS instance as part of your Elastic Beanstalk definition and alter its security group to allow access to it from hosts in your application subnets. (Not optimal for persistence as the RDS is associated with the Elastic Beanstalk lifecycle and would not live independently)
   2. Create your RDS instance separately and add its IP address to your application’s DB connection strings in your code. Alter its security group to allow access to it from hosts within your VPC’s IP address block. (RDS is connected using DNS endpoint only)
   3. **Create your RDS instance separately and pass its DNS name to your app’s DB connection string as an environment variable. Create a security group for client machines and add it as a valid source for DB traffic to the security group of the RDS instance itself.**(Security group allows instances to access the RDS with new instances launched without any changes)
   4. Create your RDS instance separately and pass its DNS name to your DB connection string as an environment variable. Alter its security group to allow access to it from hosts in your application subnets. (Not optimal for security adding individual hosts)
6. Your must architect the migration of a web application to AWS. The application consists of Linux web servers running a custom web server. You are required to save the logs generated from the application to a durable location. What options could you select to migrate the application to AWS? (Choose 2) **[PROFESSIONAL]**
   1. Create an AWS Elastic Beanstalk application using the custom web server platform. Specify the web server executable and the application project and source files. Enable log file rotation to Amazon Simple Storage Service (S3). (EB does not work with Custom server executable)
   2. Create Dockerfile for the application. Create an AWS OpsWorks stack consisting of a custom layer. Create custom recipes to install Docker and to deploy your Docker container using the Dockerfile. Create custom recipes to install and configure the application to publish the logs to Amazon CloudWatch Logs (although this is one of the option, the last sentence mentions configure the application to push the logs to S3, which would need changes to application as it needs to use SDK or CLI)
   3. Create Dockerfile for the application. Create an AWS OpsWorks stack consisting of a Docker layer that uses the Dockerfile. Create custom recipes to install and configure Amazon Kinesis to publish the logs into Amazon CloudWatch. (Kinesis not needed)
   4. **Create a Dockerfile for the application. Create an AWS Elastic Beanstalk application using the Docker platform and the Dockerfile. Enable logging the Docker configuration to automatically publish the application logs. Enable log file rotation to Amazon S3.**(Use Docker configuration with awslogs and EB with Docker)
   5. **Use VM import/Export to import a virtual machine image of the server into AWS as an AMI. Create an Amazon Elastic Compute Cloud (EC2) instance from AMI, and install and configure the Amazon CloudWatch Logs agent. Create a new AMI from the instance. Create an AWS Elastic Beanstalk application using the AMI platform and the new AMI.**(Use VM Import/Export to create AMI and CloudWatch logs agent to log)
7. Which of the following groups is AWS Elastic Beanstalk best suited for?
   1. **Those who want to deploy and manage their applications within minutes in the AWS cloud.**
   2. Those who want to privately store and manage Git repositories in the AWS cloud.
   3. Those who want to automate the deployment of applications to instances and to update the applications as required.
   4. Those who want to model, visualize, and automate the steps required to release software.
8. When thinking of AWS Elastic Beanstalk’s model, which is true?
   1. Applications have many deployments, deployments have many environments.
   2. Environments have many applications, applications have many deployments.
   3. **Applications have many environments, environments have many deployments.**(Applications group logical services. Environments belong to Applications, and typically represent different deployment levels (dev, stage, prod, forth). Deployments belong to environments, and are pushes of bundles of code for the environments to run.)
   4. Deployments have many environments, environments have many applications.
9. If you’re trying to configure an AWS Elastic Beanstalk worker tier for easy debugging if there are problems finishing queue jobs, what should you configure?
   1. Configure Rolling Deployments.
   2. Configure Enhanced Health Reporting
   3. Configure Blue-Green Deployments.
   4. **Configure a Dead Letter Queue**(Elastic Beanstalk worker environments support SQS dead letter queues, where worker can send messages that for some reason could not be successfully processed. Dead letter queue provides the ability to sideline, isolate and analyze the unsuccessfully processed messages. Refer [link](http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features-managing-env-tiers.html#worker-deadletter))
10. When thinking of AWS Elastic Beanstalk, which statement is true?
    1. Worker tiers pull jobs from SNS.
    2. Worker tiers pull jobs from HTTP.
    3. Worker tiers pull jobs from JSON.
    4. **Worker tiers pull jobs from SQS.**(Elastic Beanstalk installs a daemon on each EC2 instance in the Auto Scaling group to process SQS messages in the worker environment. Refer [link](http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features-managing-env-tiers.html))
11. You are building a Ruby on Rails application for internal, non-production use, which uses MySQL as a database. You want developers without very much AWS experience to be able to deploy new code with a single command line push. You also want to set this up as simply as possible. Which tool is ideal for this setup?
    1. AWS CloudFormation
    2. AWS OpsWorks
    3. AWS ELB + EC2 with CLI Push
    4. **AWS Elastic Beanstalk**
12. What AWS products and features can be deployed by Elastic Beanstalk? Choose 3 answers.
    1. **Auto scaling groups**
    2. Route 53 hosted zones
    3. **Elastic Load Balancers**
    4. **RDS Instances**
    5. Elastic IP addresses
    6. SQS Queues
13. AWS Elastic Beanstalk stores your application files and optionally server log files in \_\_\_\_.
    1. Amazon Storage Gateway
    2. Amazon Glacier
    3. Amazon EC2
    4. **Amazon S3**
14. When you use the AWS Elastic Beanstalk console to deploy a new application \_\_\_\_.
    1. Need to upload each file separately
    2. Need to create each file and path
    3. **Need to upload a source bundle**
    4. Need to create each file