ELASTIC BEANSTALK

**AWS Management Console might have been used to create the launch configuration** - By default, basic monitoring is enabled when you create a launch template or when you use the AWS Management Console to create a launch configuration. This could be the reason behind only the basic monitoring taking place.

**AWS CLI was used to create the launch configuration** - Detailed monitoring is enabled by default when you create a launch configuration using the AWS CLI or an SDK.

**SDK was used to create the launch configuration** - Detailed monitoring is enabled by default when you create a launch configuration using the AWS CLI or an SDK.

**The default configuration was Auto Scaling was not set** - This is an invalid statement and given only as a distractor.

* You must use saved configurations to migrate an Elastic Beanstalk environment between AWS accounts. You can save your environment's configuration as an object in Amazon Simple Storage Service (Amazon S3) that can be applied to other environments during environment creation, or applied to a running environment. Download the saved configuration to your local machine. Change your account-specific parameters in the downloaded configuration file, and then save the changes. For example, change the key pair name, subnet ID, or application name (such as application-b-name). Upload the saved configuration from your local machine to an S3 bucket in Team B's account. From this account, create a new Beanstalk application by choosing 'Saved Configurations' from the navigation panel.
* **Define a dev environment with a single instance and a 'load test' environment that has settings close to production environment**
* AWS Elastic Beanstalk makes it easy to create new environments for your application. You can create and manage separate environments for development, testing, and production use, and you can deploy any version of your application to any environment. Environments can be long-running or temporary. When you terminate an environment, you can save its configuration to recreate it later.
* It is common practice to have many environments for the same application. You can deploy multiple environments when you need to run multiple versions of an application. So for the given use-case, you can set up 'dev' and 'load test' environmen
* CLOUDFORMATION
* Q: What elements of my application can I control when using AWS Elastic Beanstalk?
* With AWS Elastic Beanstalk, you can:
* Select the operating system that matches your application requirements (e.g., Amazon Linux or Windows Server 2016)
* Choose from several Amazon EC2 instances including On-Demand, Reserved instances, and Spot instances
* Choose from several available database and storage options
* Enable login access to Amazon EC2 instances for immediate and direct troubleshooting
* Quickly improve application reliability by running in more than one Availability Zone
* Enhance application security by enabling HTTPS protocol on the load balancer
* Access built-in Amazon CloudWatch monitoring and getting notifications on application health and other important events
* Adjust application server settings (e.g., JVM settings) and pass environment variables
* Run other application components, such as a memory caching service, side-by-side in Amazon EC2
* Access log files without logging in to the application servers

Q: What are the Cloud resources powering my AWS Elastic Beanstalk application?

AWS Elastic Beanstalk uses proven AWS features and services, such as Amazon EC2, Amazon RDS, Elastic Load Balancing, Auto Scaling, Amazon S3, and Amazon SNS, to create an environment that runs your application. The current version of AWS Elastic Beanstalk uses the Amazon Linux AMI or the Windows Server 2012 R2 AMI.

Q: Can I use Amazon S3 to store application data, like images?

Yes. You can use Amazon S3 for application storage. The easiest way to do this is by including the AWS SDK as part of your application’s deployable file. For example, you can include the AWS SDK for Java as part of your application's WAR file.

Q: What database solutions can I use with AWS Elastic Beanstalk?

AWS Elastic Beanstalk does not restrict you to any specific data persistence technology. You can choose to use Amazon Relational Database Service (Amazon RDS) or Amazon DynamoDB, or use Microsoft SQL Server, Oracle, or other relational databases running on Amazon EC2.

Q: How do I set up a database for use with AWS Elastic Beanstalk?

Elastic Beanstalk can automatically provision an Amazon RDS DB instance. The information about connectivity to the DB instance is exposed to your application by environment variables. To learn more about how to configure RDS DB instances for your environment, see the Elastic Beanstalk Developer Guide.

Q: How do I make my application private?

By default, your application is available publicly at myapp.elasticbeanstalk.com for anyone to access. You can use Amazon VPC to provision a private, isolated section of your application in a virtual network that you define. This virtual network can be made private through specific security group rules, network ACLs, and custom route tables. You can also easily control what other incoming traffic, such as SSH, is delivered or not to your application servers by changing the EC2 security group settings.

Q: Can I restrict access to specific AWS Elastic Beanstalk resources?

Yes. You can allow or deny permissions to specific AWS Elastic Beanstalk resources, such as applications, application versions, and environments.

Q: Will my application be available during the maintenance windows?

Since managed platform updates use an immutable deployment mechanism to perform the updates, your application will be available during the maintenance window and consumers of your application will not be impacted by the update.