Elastic Beanstalk

AWS Elastic Beanstalk

- PAAS platform
- Quick Deployment and application Management
- Do not need to know IAAS
- Reduce Management Complexity
- Just Upload your app
- Beanstalk manages
  - Capacity Provisioning
  - Load Balancing
  - Scaling
  - App Health Monitoring

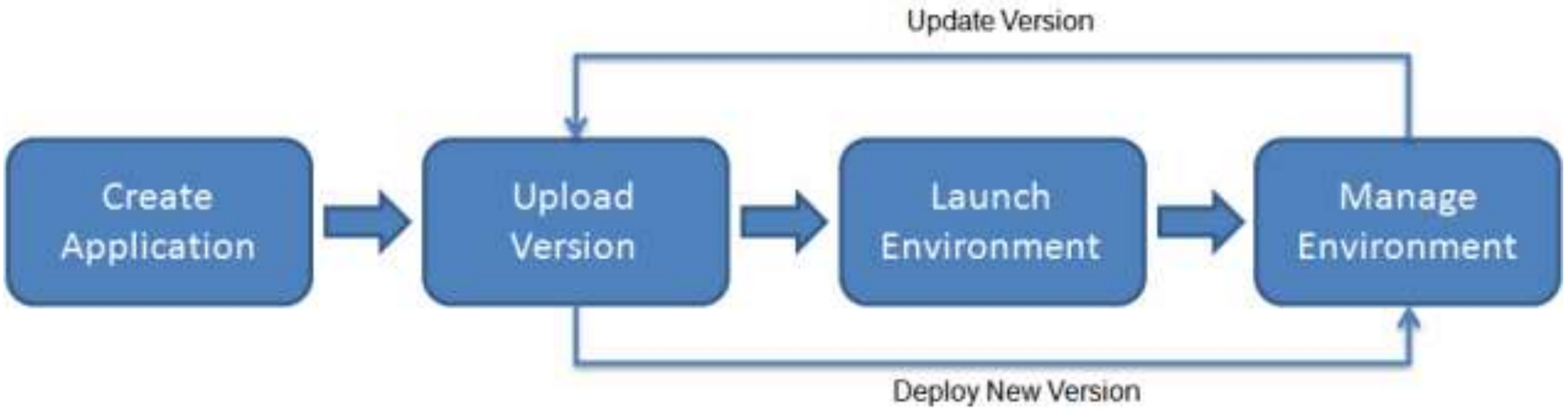
AWS Elastic Beanstalk

- Supports apps written in
  - Go
  - Java
  - .NET
  - Node.js
  - PHP
  - Python
  - Ruby
  - Docker platforms

AWS Elastic Beanstalk

- You can use Beanstalk using
  - Console
  - AWS CLI
  - AWS EB CLI – Another CLI geared for Elastic Beanstalk
    - Requires Python 3.8+
    - On Windows also requires IIS Server to be installed
- Basic Flow
  - Create an Application locally
  - Upload an Application Version (WAR/JAR)
  - Provide Configure the application
  - AWS Beanstalk will automatically launch an environment and create resources

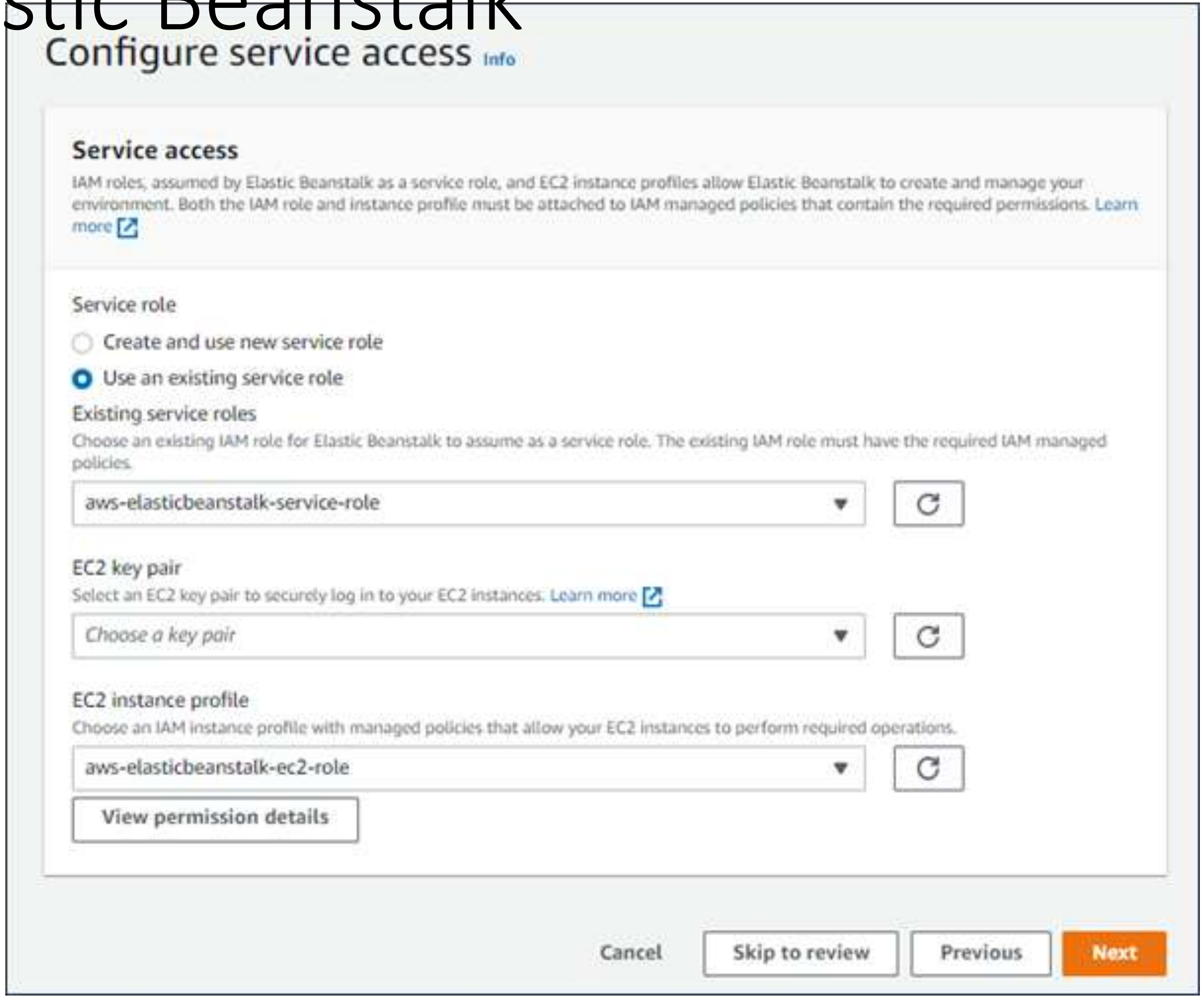
AWS Elastic Beanstalk



AWS Elastic Beanstalk

- There is no additional charges for AWS Elastic Beanstalk
- However, it will use resources which may attract charges

AWS Elastic Beanstalk



Configure service access Info

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. Learn more

Service role

○ Create and use new service role

● Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role ▼

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. Learn more

Choose a key pair ▼

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

aws-elasticbeanstalk-ec2-role ▼

AWS Elastic Beanstalk

- An AWS Elastic Beanstalk application is a logical collection of components
- These are
  - Environments
  - Versions
  - Environment Configurations

AWS Elastic Beanstalk

- **An AWS Elastic Beanstalk application** is conceptually similar to a folder
- AWS Elastic Beanstalk allows you to run applications as environments.
- **An Application version** is a **specific, labeled iteration** of deployable code for a web application
- An Application Version points to an Amazon Simple Storage Service (Amazon S3) object that contains the deployable code.
- An Application can have multiple versions.

AWS Elastic Beanstalk

- **An AWS Elastic Beanstalk Environment** is a collection of AWS resources
- Each Environment runs only one Application Version at a time.
- We can run same application version or different application versions on multiple environments at same time.
- To do so, we first choose an Environment Tier.
- An environment Tier selects the type of application and controls what resources are required. Examples are:
  - Web Server Environment Tier
  - Backend services run in worker environment tiers

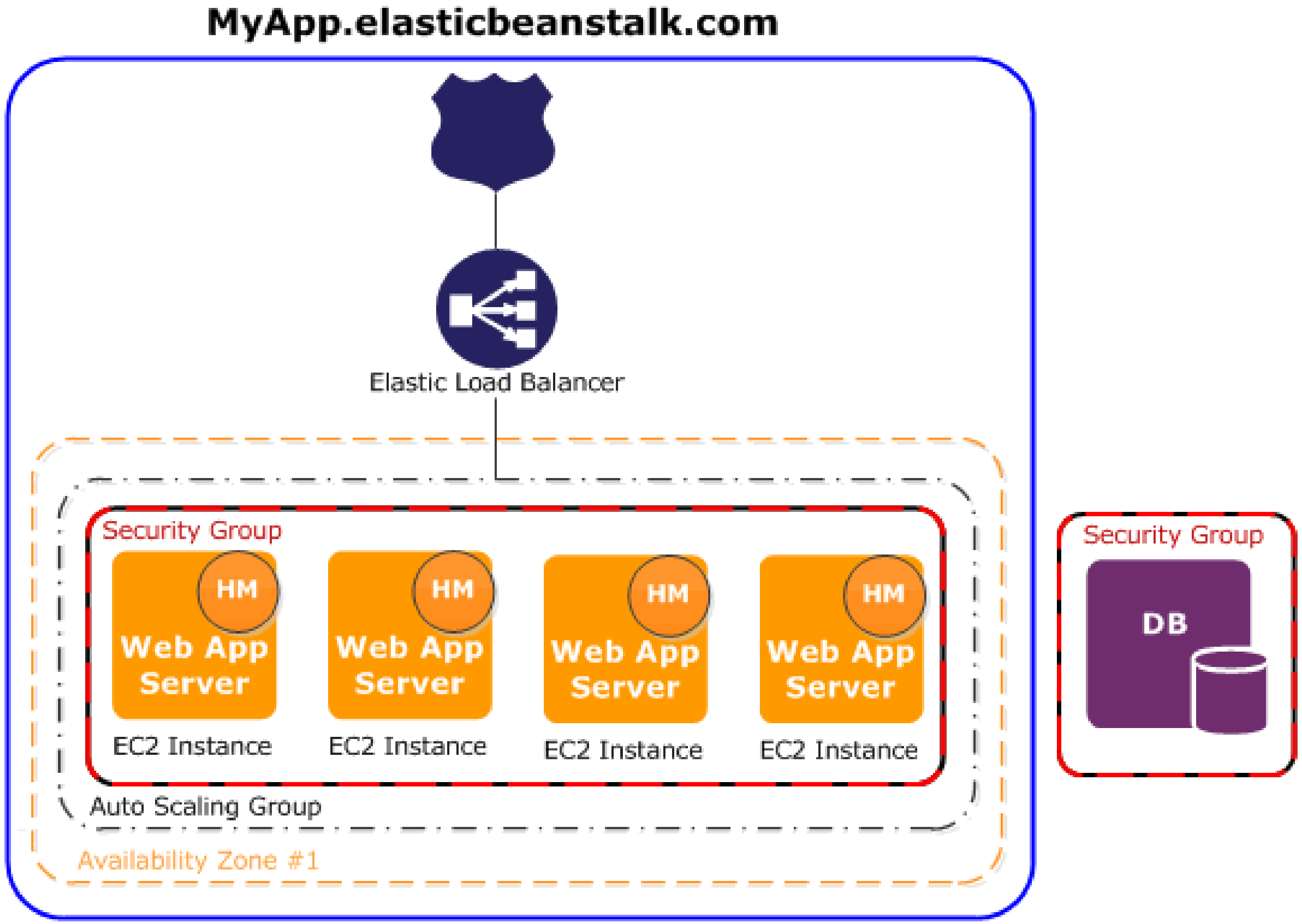
AWS Elastic Beanstalk

- An Environment Configuration is a collection of parameters and settings that define how an environment and its associated resources behave.
- When you update an environment’s configuration settings, Elastic Beanstalk automatically applies the changes to existing resources or deletes and deploys new resources (depending on the type of change).
- You can use templates for re-use - these are “saved Configurations”

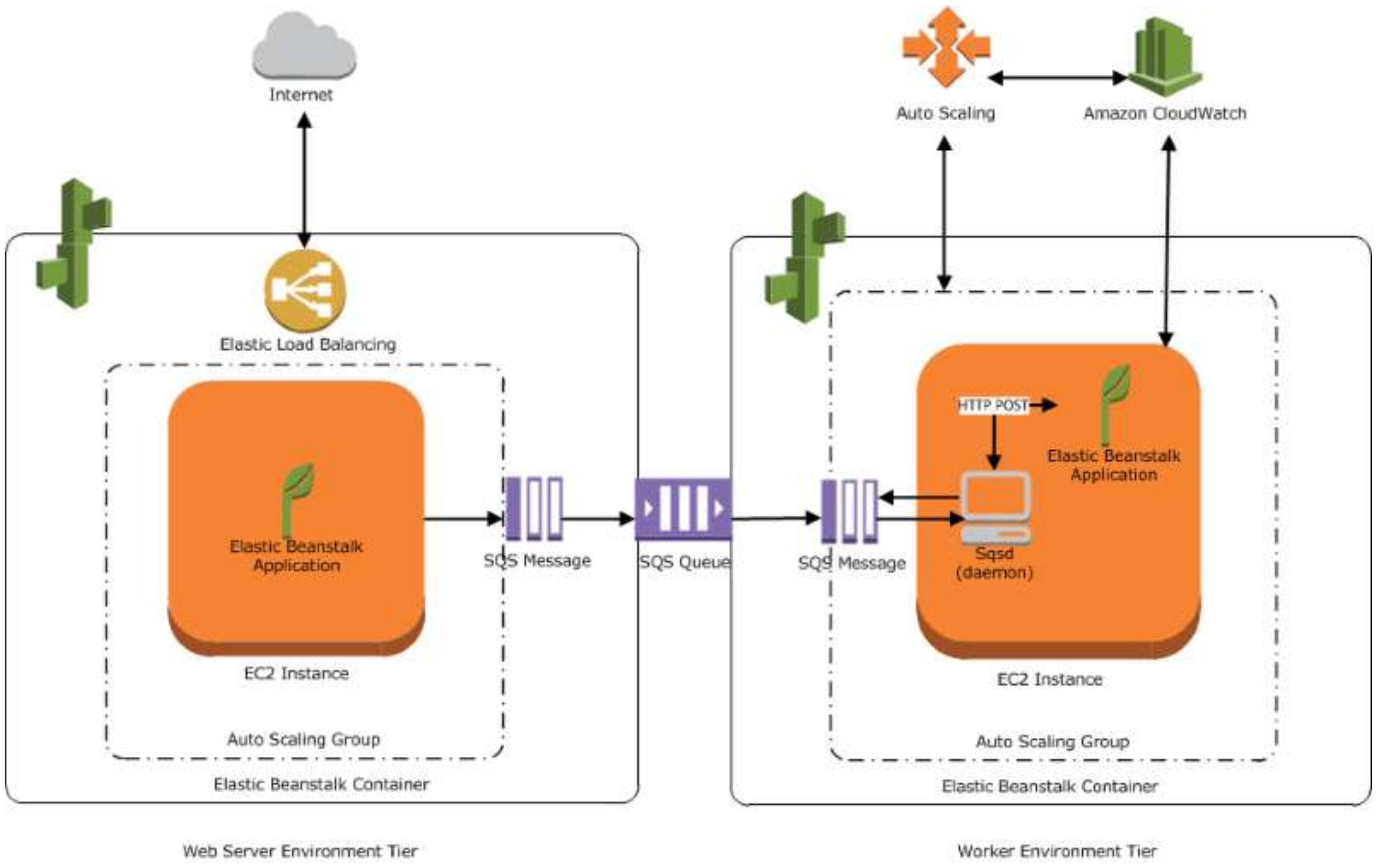
AWS Elastic Beanstalk

- A Platform is a combination of an operating system, programming language runtime, web server, application server, and Elastic Beanstalk components.
- Design and target your web application to a platform.
- Elastic Beanstalk provides a variety of platforms on which you can build your applications.

AWS Elastic Beanstalk



AWS Elastic Beanstalk



AWS Elastic Beanstalk

- Beanstalk Applications do not have persistent storage as default. You can configure either
  - EBS
  - EFS
  - S3
  - DynamoDB
  - RDS
- Security of data in/out is customer security.
- Network Traffic has to be over HTTPS not HTTP (get certificate from AWS Certificate Manager – ACM)

AWS Elastic Beanstalk

- Required Roles
  - Service Role - assume this role to run the application. Assigned via managed policies
  - Instance Profile – Applied to EC2 Instances. Allows them to write/read logs to S3 for example.

AWS Elastic Beanstalk

- A service role is the IAM role that Elastic Beanstalk assumes when calling other services on your behalf.
- 2 Managed roles are attached
  - AWSElasticBeanstalkEnhancedHealth
  - AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy
- An instance profile is in an IAM role that's applied to Amazon EC2 instances that are launched in your Elastic Beanstalk environment.

AWS Elastic Beanstalk

- An instance profile is in an IAM role that's applied to Amazon EC2 instances that are launched in your Elastic Beanstalk environment.
- Used when
  - Retrieve application versions from Amazon Simple Storage Service (Amazon S3)
  - Write logs to Amazon S3
  - In worker environments, read from an Amazon Simple Queue Service (Amazon SQS) queue
  - In worker environments, publish instance health metrics to Amazon CloudWatch
-

AWS Elastic Beanstalk

- Uses these policies
  - AWSElasticBeanstalkWebTier
  - AWSElasticBeanstalkWorkerTier
  - AWSElasticBeanstalkMulticontainerDocker
-