ENGINE YARD CLOUD
BEST PRACTICES FOR DISASTER RECOVERY

This technical brief details the fault tolerance built into Engine Yard Cloud as well as the mechanisms of recovery should certain infrastructure suffer failure.

This document assumes the readers' knowledge on the following topics:

- Cluster configuration in Engine Yard Cloud
- Amazon Web Services terms such as S3, EBS, EC2

Fault tolerance vs. disaster recovery

- Fault tolerance is architecting your application to avoid interruptions in service. This means architecting mechanisms like redundancy that do not allow an entire application to go down despite certain components failing.
- Disaster recovery is a set of pre-planned mechanisms that a team employs to bring their site back online when it goes down. (Fault threshold was surpassed)

Support options for Fault Tolerance and Disaster Recovery

- Premium Support customers: Our support team steps in proactively
- Standard Support customers: Our support team assists after the customer has opened a ticket on the issue

The Engine Yard Cloud Platform as a Service (PaaS) utilizes Amazon Web Services (AWS) as its underlying infrastructure. When you create an instance with Engine Yard, you are booting up an AWS instance. Engine Yard Cloud boots this instance for you and automatically configures it with the appropriate Engine Yard platform components for your environment. Other Engine Yard technologies automate and manage key functions including cluster management, load balancing, high availability, database replication, and monitoring and alerting.

Engine Yard Best Practices

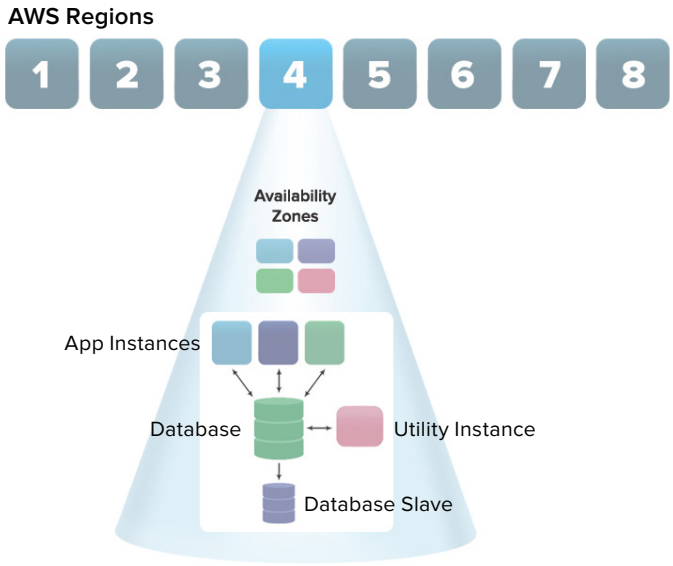
Backups

Engine Yard Cloud automatically performs snapshots of application data and backups of the database for our customers. This data is stored on an Amazon S3 bucket that we provision on our customer's behalf. You are given control over the frequency at which these backups should be performed.

We attach EBS volumes to all instances. Any application and database data is stored on the EBS volume as opposed to the volume on the instance. We do this because **EBS has its own layer of redundancy**. What's more, this further decouples the failure of an instance from the associated volume of data.

Regions and Availability Zones

Engine Yard allows customers to deploy their applications to any AWS region they choose; we support all eight AWS regions. Within each of these regions, Amazon has multiple discrete Availability Zones (AZs) (ranging from 2 to >=4). These AZs are separate data centers and are designed in such a way to contain disasters - each AZ runs independently and should be resilient to the failure of others.



There are multiple availability zones within a given region. Instances in your app are deployed across these separate availability zones.

Fault Tolerance and Disaster Recovery

This section elaborates how Engine Yard architects different instance types - application, database, and utility instances - to withstand failure and the tools provided to the customer to recover from failure.

Application Instances

If a customer has two or more application instances in a cluster, they are auto-healing. If an application instance is determined to be unresponsive, it is deregistered from the load balancer. At that time, it is decommissioned while a new application instance is brought up to take its place.

If the app instance that failed was the app master, an app slave will be promoted to replace it. This process usually only results in one to two minutes of downtime after the app master has been deemed unresponsive. Naturally, another app instance will be brought up in the background as a slave.

All these steps are performed without customer intervention. The customer will be notified that the takeover took place.

Note that application instances are all brought up in different AZs when possible. If application instances go down due to a failure of a given AZ, new application instances will be brought up in healthy AZs.

Database Instances

Due to the sensitive nature of avoiding data corruption, database instances require intervention to failover.

The situations below fall into two scenarios:
(1) Instance fails, but EBS volume is preserved
(2) EBS volume fails

Scenario 1: EBS volume is preserved
Our support team will be able to recover all up-to-date data in your database.

Scenario 2: EBS volume fails

A) No database slave present
Time to recover: Fast
Cost: Lowest
Risk to recent data: High

If the EBS volume fails, the instance can be rebuilt from the most recent snapshot or rebuilt with a clean volume and reloaded using a database dump. Any data not snapshotted or dumped will be lost.

Support can perform this procedure for our customers, but our customers can do so themselves if speed is of the essence. To do this, they must terminate all instances in the environment (by hitting “Stop”) and then reboot the cluster (“Boot”) to have all instances restored from snapshots. Depending on the size of the cluster, this usually takes 7-15 minutes. If a customer is unfamiliar with this procedure, we recommend they wait until consulting with support.

B) Database slave present
Time to recover: Fast
Cost: Low
Risk to recent data: Low

If the EBS volume fails, the database slave will have very recent data. Replication is asynchronous, but the slave usually does not lag behind the master by more than a few milliseconds. Typically, a replica will still have more recent data than the most recent snapshot of the master EBS volume. (Engine Yard has monitoring in place for replication, and we will be alerted if it fails to run or is significantly delayed.)

A customer’s best option is to have Support perform a manual failover, where the slave is promoted to master.

Utility Instances

If a utility instance fails, it will be up to either the customer or Support to replace it. Naturally, the customer can ask Engine Yard to replace it for them.

Failing Over to Another Region

All disaster recovery plans thus far were intraregional. This means they rely on at least one healthy Availability Zone to be running. While failure of an entire AWS region is incredibly rare, some business critical applications require the extra level of disaster recovery.

This behavior is not supported by default on Engine Yard. However, our Professional Services team can create a solution for you. In this scenario, the customer has two duplicate clusters running, just in separate regions. One cluster actively serves traffic, the redundant cluster does not. Instead, the redundant cluster receives database updates and deploys so that it is kept up-to-date. Should a failover be needed, support performs the failover. The procedure can take anywhere from 20 minutes to an hour as data is carefully migrated between regions.

Engine Yard, 500 Third Street, Suite 510, San Francisco, CA 94107
www.engineyard.com • sales@engineyard.com • 1-866-518-9273 • 1-415-624-8380