Environments Overview

Authors:

- Adam Edwards

Overview

This document contains an overview of the Devops environments used in managing a product's lifecycle, with a product being a platform, service(s) or application(s).

Environments

Development Environments

Development environments are those actively used to develop systems. They frequently change and are broken by developers as they work with them.

Sandbox

A sandbox environment is the environment most used by a developer with it often being their local machine. In cloud engineering this may be a separate environment in and of itself for cloud engineers to rapidly develop and break things without impacting others.

Resources here are usually instances used by a single individual.

Development (dev)

A dev environment is like a sandbox environment but resources here are shared and worked upon by many developers. Some products may not have development environments at all.

It's usually the first environment where a component of a large system is tested with other components of the system.

Test environments

Test environments are used to perform QA (quality assurance, aka testing) before promoting systems to production.

SIT

SIT stands for system integration testing. With other systems is often performed here but it does not have access to production data of other systems.

Staging (or pre prod)

A staging environment is used to stage production releases. Often the staging environment will have access to the production data including that of other systems to allow for proper load testing and validation of outputs. It is the most production like environment outside of production.

Production environment

A production environment is where services run live for use by users or other services. This is the most controlled environment with regards to change management as changes here have the greatest risk of impact to service delivery.

Long lived vs short lived environments

Resources within an environment almost always have an associated cost to run, so those outside of production should do their best to be ephemeral / short lived. Often, it's worth keeping resources deployed within an environment for extended periods of time, but each case is different and requires consideration.

One good case for resources in a non-prod environment not being ephemeral is a data warehouse. A data warehouse often requires a large volume of data to test with and it is costly to load, so it's common to leave these resources running in the long term.