Migration Use Cases & Processes

Before diving into this topic, it's worth considering why a large-scale migration may be required so that we can understand some of the constraints that the discussion and guides below are attempting to work with. By large scale website migrations, we are specifically referring to relocation of a large number of websites, as compared to moving a single large website. This document is a summary of some use cases and best practices to help clients understand the different situations and guidelines we use in moving their data, websites and applications to our servers.

Some possible reasons for moving:

- Existing web hosting is located on servers owned by the company and they either wish to relocate these servers to a different location (where the existing IP address space cannot be used) or they wish to have a third party take over provision of web hosting services.
- The existing web hosting service is either in-house or provided by a third party that is no longer suitable. This could be due to:
    - Level of service is not acceptable
    - The requirements of the client have outgrown the capabilities or scale at which the existing provider operates at
    - Commercial reasons such as pricing are dictating the need to move
    - The existing service is currently located offshore and there is a desire to make it local
    - The client wishes to have a greater level of control over the hosting, i.e. currently outsourced and wish to bring in-house

The original hosting arrangements will define the options which exist for migration:

- Websites are hosted under a reseller plan or some form of shared hosting, in which the client does not have control of the server and root access cannot be provided. Websites may or may not all be hosted in the same place or with the same company.
- Websites are hosted on a server (either in an office or a data centre) that is dedicated to the client, and which they have full access to and control over.

Major challenges associated with large scale website migrations:

It's worth taking a look at what we consider to be some of the major challenges in large-scale website migrations to help you identify some of the issues that you will need to consider:

- ***Changes which depend on third parties.*** Having a large number of configuration changes or steps involved in a service migration in itself is often not a major problem. Many of the more repetitive tasks can be scripted and checklists can be used to maintain quality on those that can't. Changes upon which you are entirely dependent on third parties can be one of the more difficult aspects. These include:
  - References to IP addresses of the existing hosting environment from third parties. This may be in the form of hard coded addresses or firewall rules.
  - Externally hosted DNS. Many companies may control the DNS independently of the hosting service.
  - Challenges associated with changes which depend on third parties include:
    - Identification of the people who are able to make the requested change
    - Establishment of your authority to be able to request the change with the person who is able to make the change
    - Scheduling of changes to be made at the desired time and follow through to ensure that they are actually made
- ***Testing of new environment.*** In most cases website migrations will involve significant changes to the versions and installations of the programs or applications that support the websites. Even slight changes can cause significant problems. Thorough testing is absolutely critical to avoid outages and service disruption. Testing will need to be carried out by someone with intricate knowledge of the website or application that is being moved and how it is expected to operate. This will usually require involvement of either the client/end user or the developer of the website.
- ***Ensuring continuity and consistency of data on either side of the migration.*** Continuity of the data must be protected both in terms of dynamic (created by visitors to the website) data and the code base itself. If migrations are spread over a period of days, a lock out of some sort may be required to ensure now coding changes are not made during the migration.

- ***Ensuring compatibility of code with new versions of applications on new hosting environment.*** As most operating systems and applications only have a limited support life (in respect to availability of security updates), when a migration occurs, it will commonly (and sensibly) involve the new hosting environment running the latest stable version of the operating system and application to minimize the requirement of future upgrades. Changes to application versions may result in incompatibilities in the website code which must be taken into account and tested.
- ***Avoid mistakes.*** Humans by nature make mistakes, checklists and processes help to a large degree but they still don't provide a failsafe solution. For small data sets, manual migration is most often appropriate, as the data set or number of websites to be migrated increases, so do the chances of errors. Automating many of the tasks can reduce human involvement and in many cases reduce the chances of human error. With large numbers of sites, the overhead of automating/scripting becomes more acceptable and warrants consideration.

Possible approaches to large-scale website migration:

- **One at a time.** Moving a large number of websites one at a time is a very flexible approach, but is also very time consuming. Moving a large number of sites one at a time is simply a repetition of a single site move procedure. The major difference between the two (aside from the time), is the need to maintain very accurate records for the status of the migration tasks for each of the websites being moved. This becomes important as each site will take time and, in practice, the people performing the migration tasks will be constantly working on multiple sites concurrently as they wait for responses from clients, completion of testing, DNS propagation etc. The risk of steps being skipped, tasks forgotten, response missed etc. is increased.

- **All at once.** Moving everything in one hit can be fraught with risk, not to mention nerve racking. There are some very solid reasons for this approach though, as described below. If you are able to get past the constraints associated with this move approach, the biggest advantage is that you use significantly less time to complete the move than a "one at a time" approach. Generally, if all sites are able to be moved at the same time, the interaction with the end client or owner of the website will typically be much lower.

Reasons the “All at Once” approach may be required:

It may be possible and highly desirable to maintain the same IP addresses on the old and new infrastructure. If the migration is occurring across infrastructure within the same network provider (for example, migrating from co-location to dedicated servers or vice-versa with the same hosting company), it is often possible to keep the same IP address. A major motivation for the "all at once" approach will be if there are many references to the existing IP addresses which are not directly controlled (i.e. external DNS, hard coded IP address in third party applications, IP specific source/destination firewall rules). In such cases, the new infrastructure can be prepared on alternative IP addresses and at cut-over time, the original IP address is simply moved across, eliminating the need for DNS changes and so forth.

Common dependency constraints the nature of the way in which the websites have been developed and in some cases may make it difficult to have a varying numbers of sites operating from varying locations across the migration period. This may be due to a shared code base or common database backend, etc.

Pre-requisites for an “all at once” approach to migration:

- You must be able to maintain the same IP address on either side of the migration, or
- You must have full control over all services such as DNS that reference the IP addresses on the servers being migrated, or
- Be able to and find it acceptable to use a proxy to redirect all requests to all services from the old to the new infrastructure in the event that not all references to the IP address can be updated within a very short time period.
- You must have full control and full access to both the original and destination hosting environments.
- If moving email services, it is high desirable, if not mandatory, to be able to maintain the same authentication details for all users on either side of the migration.

Core components of a large scale website migration plan:

Since every migration is very different there's no sense in trying to provide a generic “one size fits all” migration plan. We can, however, put together a series of best practices from which we draw upon to produce a custom migration plan.

Existing infrastructure analysis:

For controlled infrastructure, the existing configuration should be analyzed to ensure there is a clear understanding of the starting point. This includes:

- Identification of applications that are in use
- Identification of versions of all applications that are in use
- Validation of the clients' description of infrastructure
- Security audit - ensure no evidence of server compromise. If the existing server has been compromised, we must ensure a greater degree of independence with the new system, ensure passwords are reset, etc.
- Identify any interaction with third party services which may have IP (source or destination) specific requirements/restrictions in place

Confirm viability of proposed migration:

Before any significant amount of work is carried out on the migration, it is prudent to ensure that it will be possible to complete the process within the client's technical and financial limits. This should include consideration of:

- Ensuring compatibility of website code with changes to application versions which may be requested or required as part of the migration. For example, newer operating system versions may mandate the use of new PHP/MySQL versions. The client may wish to adopt a newer PHP version in order to gain access to new functionality as part of the migration.
- Core application changes - For example, migration may in part be dictated by the limitations of a mail server application having been reached and the desire to adopt a different platform. The practicalities of such a migration must be thought through
- Ensure that a sufficient budget has been allocated to complete both the known migration work, but also an allowance to deal with any unforeseen tasks which may arise during the migration process
- A realistic timeframe must be set and ensure that this is acceptable to the client

Migration tracking register:

Migration of a number of websites will inherently involve a multitude of tasks and independent websites. To complete a successful migration, we will have a mechanism in place to track the status of all migration tasks. For any opportunity, we have a designated point of contact who will touch base with the designated client contact based on client timelines and requirements.

Rollback plan:

Any change to a live system has its associated risks. We maintain a rollback plan for any changes which have the capacity to effect any live service should be in place for the migration project. Most importantly:

- Ensure rollback capacity exists at every step of the migration
- Ensure full backups of original infrastructure before starting
- Consider making existing environment as stable as possible before starting to work with it
- Identify risks associated with any existing configuration and make them known to client

Identify sites/applications being migrated:

Create a list of the domain names, servers, software, data and other tools which have services that are to be migrated. This list should form part of the migration tracking register identified above. For controlled infrastructure, extract lists of the following:

- Domain names - from web server configuration
- DNS
- Mail domains
- System user accounts
- Mail accounts
- FTP accounts
- SSH accounts
- Database user accounts and database names
- Remote access services
- Local services

For non-controlled infrastructure:

- Request list of active domains from the hosting vendor or exact from self-service interfaces

Validation of list of actively hosted sites:

Check the accuracy of the list created above using an automated script. The script should check for each domain name that is to be migrated and that the relevant addresses (i.e. www or mx) are pointing to the IP address or range of IP addresses which are used as part of the original web hosting infrastructure. This automated check identifies sites that may have been moved away without notification to the provider. There is no value in migrating services which are not active.

Identification of DNS services:

Similar to above, a script should be used to identify and validate the name server settings for all domains to be migrated. This information should be incorporated into the tracking register.

“Sanity check” of server configuration:

From the information collected and vetted above, a “sanity check” is completed to ensure that both the correct services and required services are being migrated. This may involve a secondary migration team member and/or the client manually auditing lists of user accounts, domain names, servers, etc. to ensure accuracy.

DNS migration:

If DNS is to be migrated as part of the process, we like to perform the move at least 24 hours prior to migration of any other service. If DNS is currently not controlled by the client but it is to be, this is of particular importance.

DNS migration should occur as follows:

- See more detailed guides on DNS migration:
  - Using Whois to investigate domain and DNS records
  - How DNS works
- For controlled DNS:
  - Obtain copy of zone records
  - Configure server with all zone records to be hosted
  - Point records to existing services (i.e. the IP addresses of original location, not the new location)

- Set low TTLs
- Template as much as possible to reduce errors
- Re-delegate domain names
- Validate completion of delegation changes using a script to automate
- Make sure old DNS provider removes DNS records
- For externally hosted DNS (where delegation of the domain name is not changing):
    - Identify person responsible for making change, make contact with said person
    - Ensure authority has been granted to be able request changes
    - Request TTLs to be dropped on effected domains
    - Agree on procedure and timing for changes to zone records to occur to cut-over date/time
    - Or, consider taking control of the DNS if possible
    - Validate completion of zone record changes manually or via script
- Create sub-domain records pointing at the new server to facilitate testing during migration

Server security:

- For controlled environments:
    - Audit existing firewall configuration for security, currency and appropriate design
    - Identify non-standard security requirements, client specific configurations
    - Configure firewall rules in new environment
    - For service specific access requirements (i.e. pam based restrictions):
        - Review existing configuration
        - Review requirements with client
        - Configure new server
- For non-controlled environments:
    - Request details of any custom firewall configurations from existing hosting provider
- Consider and discuss with the clients the effects of modified security profile (typically more restrictive) on their clients:
    - It's important to be aware of possible consequences of changes, associated communication/awareness required with clients
    - Review client’s specific security obligations (i.e. associated with PCI compliance)

Web server configuration:

- Analyze configuration of existing hosting environment to determine configuration requirements
    - Consider:
        - Structure of existing configuration - one file, many files
        - Hard coded paths in client code
        - Use of web server modules, for example: suexec, mod_php, mod_perl, mod_proxy_ajp, mod_ssl etc.
        - Use of application servers such as: ruby on rails, tomcat, jboss, .NET applications.
        - Identification shared libraries and other code/program dependencies
        - SSL certificates
    - If no access to existing environment request configuration files:
        - SSL certificates
    - Confirm above analysis with client/developer
- Configure webserver:
    - Create fresh or import existing configuration files
    - Potential to script modifications to configuration files
    - Configure all application dependencies
    - Configure temporary sub-domain records to facilitate testing

Database server:

- Configure new databases based on previously created list
- Import client databases
- Review server wide database configuration parameters
- Review server wide performance tuning settings, may require adjustment associated with hardware changes
- Check and configure for applications which connect from remote sources

Mail server configuration:

In most migrations, consideration will need to be given for the way in which email is dealt with. As previously discussed, mail server migration will be dealt with in a separate article, suffice to say, it will need some attention in the migration plan.

User account configuration:

- Create user accounts and set passwords
- Consider attempting to maintain existing password rather than resetting by migrating passwords files

Initial file system data replication:

- Copy files from home directory for each website
- Check file permissions on copied files
- Perform dump of databases on original server and re-import on new server

Website statistics:

If website statistics which are generated from server log files are in use, consideration should be given to the effects of the migration on these:

- Moving from a non-controlled environment
  - Generally or not easily possible to provide contiguous reporting
  - Consider archiving statistics from the original environment and starting fresh on the new
- Moving from a controlled environment
  - May be possible to import old statistics and provide contiguous reports. This will require additional attention at the time of migration

Testing to verify:

With file system data copied over and snapshots of databases imported, initial testing should be carried out on the migrated sites:

- Test websites using sub-domain testing URL's created
- If there are URLs hard coded into the site, the person doing the testing (i.e. client, developer) may be required to set host records on the testing users workstations to permit testing

Auro

Auro