Worksheet: Migration Plan

Application: _______________________________________

Migration Architect: _______________________________________

Style of Migration

[ ] All at once [ ] One component at a time
[ ] Inside Out [ ] Outside In

Notes:

Cloud Ready Analysis

What refactoring needs to happen before a migration is initiated?

<u>Cloud Ready Analysis</u>

- Will it run in the cloud (virtual servers)?
- Maintain state within the app itself?
  - aka, filesystem/disk
- Custom network settings?
- Unique load balancing?
- Twelve Factor App
  https://12factor.net
- Application analytics tools

Data Migration

What data needs to be migrated? What strategy will you use? What about offline data transfer such as AWS Snowball?

<u>Data Migration</u>

- Synchronized Primaries
- Primary/Replica Switch
- Read/Only Datasets

- Large Dataset / Offline Transfer

Post Cloud Refactorings (Required)

What refactoring must happen after application has moved to the cloud, but are important enough that you will implement them as part of the migration project? This includes required refactorings for including cloud-based services into the project, performance tunings, and usage of dynamic resource elements.

Required Cloud Changes

- Required performance tuning
- Cost containment
- Database changes
  - Start using S3, DynamoDB…
- Resource management
- CI/CD changes
- Operational procedures

Post Migration Possibilities (Optional)

What refactoring would be useful after the migration is complete, but not strictly required? This includes useful but not required refactorings to include cloud-based services into the project, performance tunings, and usage of dynamic resources allocations.

Optional Cloud Changes

- Optional performance tuning
- Cost optimizations
- Dynamic services
  - Such as Lambda…
- Cloud datastore usage/patterns
  - Increased use of S3, DynamoDB…

Production Go-Live Plans

What are your plans for switching production traffic from on-prem stack to cloud stack?

[ ] All at once [ ] Slow Ramp [ ] Canary Test

Load Shift Strategy

Notes

Production Go-Live

- All at once
  (0%->100% switch)
- Slow ramp
  (0%->100% over period of time)
- Canary Test
  (0%-n%, to test on small sample)
- Data migration impact?
  (does your data migration strategy support your go-live plan?)
- Strategy for load shifting
  (load balancer, DNS, …)

Reference

Migration Steps

Before Migration	During Migration	After Migration
•Instrument your entire system •Establish baselines •Create acceptance criteria from the baselines •Perform all planned system-level pre-migration refactorings •Reconfirm baselines post refactoring	•Do service-level pre-migration refactorings •Migrate data •Migrate the service •Resolve roadblocks/problems •Post-migration refactorings •Validate performance & acceptance criteria •Resolve any performance issues	•Planned post-migration system-level refactorings •Validate performance & acceptance criteria •Resolve any performance issues

Keys to a Successful Migration

1. Limit complexity of data migration
2. Reduce duration of in-progress migrations
3. Leave yourself backout options
4. Be conscious of interim performance issues
5. Do refactoring before you migrate

Migration Steps

Use this space to plan out the steps of your migration, using the information above as a reference.