



## CLOUD TRANSFORMATION

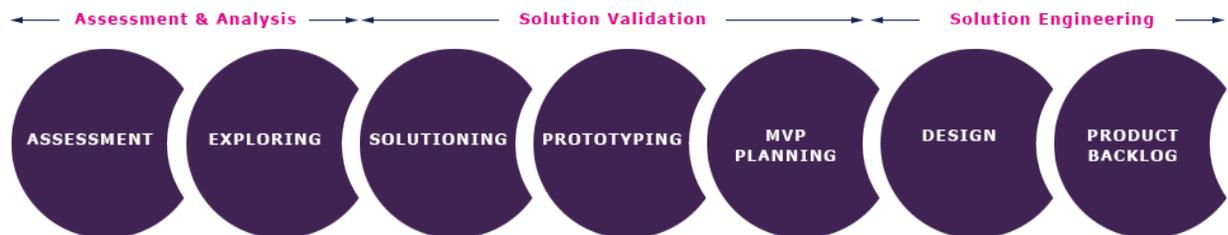
VividCloud clients usually know exactly what they want to achieve with a transformation of on-premises enterprise systems to the cloud. However, a key problem is often is how to rearchitect or refactor monolithic legacy applications running in a traditional data center to get the most benefit from a cloud transformation.

### CLOUD MIGRATION

VividCloud approaches complicated cloud migration projects with a disciplined process that begins with Discovery. The purpose of Discovery is threefold: understand the client's problem and their known requirements, develop and validate a path to migration, and build an MVP.

This process provides the team with a roadmap to a successful migration to the cloud.

#### STAGES



VividCloud Discovery Process

### CLOUD-NATIVE ARCHITECTURE

VividCloud develops a new cloud-native architecture that are optimized for the unique requirements and characteristics of the legacy software it replaces. The new architecture will utilize technologies and design approaches such as microservices, containers, serverless, and EC2 instances to achieve the optimum design around operational cost and performance.



## MONOLITHIC APPLICATIONS

A key driver of cloud transformation projects is the desire to address the limitations and high maintenance costs of aging monolithic applications.

All processes are tightly coupled in monolithic application and run as a single service. If one process in the application experiences a spike in demand, the entire architecture must be scaled. Another issue is that monolithic designs with many dependent and tightly coupled processes increase the impact of a single process failure. Another undesirable characteristic of monolithic applications is that adding or improving features becomes more complex as the code base grows.

## RE-ENGINEERED LEGACY SYSTEMS

VividCloud reengineers legacy systems and applications with a cloud native microservices architecture, where applications are built as independent components that run each application process as a service. These services communicate via well-defined interfaces using lightweight APIs. Services are built for business capabilities and each service performs a single function. Because they run independently, each service can be independently updated, deployed, and scaled to meet demand for specific functions of an application.

## VALIDATED DESIGN

Using the advantages of a Lean approach and Agile development methodology, VividCloud will build Proof of Concepts (PoC) that prototype key parts of the proposed infrastructure. The design can then be iterated until the performance, scalability, connectivity, and other foundation requirements for the migrated system are validated.

## MVP'S

VividCloud believes strongly in the value of MVPs and pilot projects that deliver a core set of the system's functionality as early as possible. Functioning software that performs key tasks well and delivers clear value is a powerful success milestone for most project development teams.

## A BUILD PIPELINE

Cloud transformations mean more than simply running enterprise systems on a cloud platform. Legacy applications usually were created with the traditional elements of a monolithic design, waterfall development methodology, and conventional QA and



release engineering. Managers, architects, and developers all seek relief from the limitations of these dated methods, and CI/CD is an important tool to gain all the benefits promised by a cloud transformation.

VividCloud brings Continuous Integration (CI), Continuous Delivery or Continuous Deployment (CD) is a set of operating principles and practices that enable application development teams to deliver code and infrastructure changes more frequently and reliably. The flow of code changes from developers to deliver/deployment is known as the CI/CD pipeline.

CI/CD is one of the best practices for DevOps teams to implement. It is an agile methodology best practice because it enables software development teams to focus on meeting business requirements, code quality, and security because deployment steps are automated.