Server Infrastructure Provisioned Through Amazon Web Services (AWS)

Outline: The following document provides clients of Technology Leadership, LLC (referred to throughout this document as “TLC”) a basic understanding of the process and steps involved in obtaining networked server infrastructure resources needed to pursue their business objectives.

The description includes which resources are needed, why, preferred account set-up methodology for securing Amazon Web Services (referred to as “AWS”) and Chef Configuration Management server, defining roles for staff, and options for how to support development and production environments.

Depending on the contracting corporation's budgets, staffing levels, in-house expertise, business objectives, security requirements, and many other factors, the details of the implementations will differ, so this particular document can only serve as a very high level general framework for engagement.

TLC also provides comprehensive Android development. This creates an unique opportunity for organizations to leverage their AWS infrastructure deployments outlined here with their Android product development efforts for compelling and tightly integrated product offerings and solutions not available elsewhere. This is important to note here, as design considerations that account for this integration now can save significant costs in money and time associated with redesign in the future.

Background: Amazon Web Services (AWS) is a collection of software and hardware 'objects' that can be organized and implemented in various ways. One of the primary services within the AWS collection is the Amazon Elastic Computing Cloud, otherwise known as EC2. EC2 is likewise, a collection of hardware and software 'objects' geared towards the provisioning of virtual servers known as server instances. A collection of server instances, including databases, load balancers and other optional components together form what is referred to here as a 'deployment'.

Because AWS contains many different product offerings, and because each of these offerings require both account management as well as technical management, the user interface via AWS is relatively complicated and is oftentimes not as consistent or user friendly as one might desire. Therefore, AWS provides an extensive set of APIs needed to perform many of the virtual hardware provisioning tasks in a more automated fashion. These APIs are the basis of the “Systems as a Service” provisioning model.

Without the use of APIs, all tasks must be done 'manually', one step at a time. As any system engineer will attest, there are hundreds of tasks to build a single fully operational server; potentially tens of thousands of tasks to build a multiple server environment. This is where Chef Configuration Management (Chef) server becomes extremely important.

Chef server offers an extremely consistent and powerful toolset managed through a user interface that vastly reduces the time to develop, test and deploy server instances in the Amazon EC2 cloud. The power behind Chef is its use of “cookbooks”, “roles” and “recipes” that combine what would be thousands of AWS and command-line server provisioning steps and commands into a few point-and-click tasks within the Chef web UI to set-up a specific set of tasks, and a single ‘chef-client’ command to execute them.

Chef integrates AWS (and many other cloud service providers) through its own API known as Knife. The knife API of chef interfaces with AWS APIs to be able to provision such as components as Elastic IP's, Elastic Block Storage (EBS), S3 Storage and others, in ways that are either very manually derived,

or not available at all, within the AWS user interface. These AWS service components are billed directly within an AWS account, but can be managed via Chef from any remote PC, or any customer owned AWS instance.

I. Obtaining an Amazon Web Services Account

TLC recommends the customer provide a valid credit card for the initial creation of a customer's AWS account and to receive direct billing from Amazon for AWS services. This is important to help meet legal, insurance, auditing, billing and tax requirements. Though it is possible to have the account under the TLC corporate account, this is highly discouraged. Because of the significant financial and legal liabilities posed to TLC, a significant fee above the cost of AWS services will be added to the customer's invoices, if AWS services are not billed directly to the customer.

The customer should provide to TLC an email account on their corporate mail server as the primary identifying email address associated with the AWS account. This account optionally can be a shared account, or have automatic cc and/or forwarding enabled. This helps ensure the customer always has access to the account in the event that AWS services and/or account passwords need to be modified without involving TLC staff, as the customer will presumably always have access to the email account and/or its forwards.

An option is to add additional email addresses to the primary AWS account with proper access to billing and password functions. However, for security and audit purposes, this should not be the sole method for customer account access. This strategy also assumes that the customer will be able to navigate and successfully change information within the AWS system without impairing their deployments.

Because of the many product offerings, the relative complexity in creating and distributing secure credentials, and the potential need to update information between the RS and AWS systems during an unscheduled event, it is customary for TLC to make all of the necessary purchases, create the accounts, and move the credential information between AWS and RS systems. TLC will add additional account manager and technical manager accounts to the the primary AWS account(s), as needed.

II. Building a Chef Server

As with AWS services, the Chef environment is fairly complicated. Chef is open source, and compiled binaries made available through Opscode are free. Once installed, Chef server can download any of a about 1000 free cookbooks. TLC will install the Chef server and create or modify cookbooks and recipes, as needed. TLC will always require at least the Amazon API key and secret key within the accounts they are managing. Access to customer's AWS services account via AWS console is also recommended, though not strictly required.

III. Providing Services

The exact details of what is to be provided in terms of number of server instances, number of geographic zones, whether EC2 will connect to a private cloud or network, and the multitude of other provisioning options and combinations do not have to be formalized prior to engagement.

However, having a general outline of what the service is, what the goals are, and the anticipated or available budget must be agreed upon before work begins. TLC will work with customers to

understand their business and develop estimates and targets for time and materials, as well as any other requirements that can be known or predicted.

To the extent that a successful infrastructure deployment requires business planning, product design, and project management, TLC can and does offer those services, as well. However, it should be understood that those services are in addition to, not part of, the actual infrastructure provisioning within the AWS and Chef environments.

Similarly, day to day operation of an existing environment often includes working with development teams, testing, deploying software and bug fixes, remedying failed, stalled or defunct tasks, responding to audits and inquiries, and providing daily operational support: well beyond a few days of up-front engineering.

IV. Roles and Responsibilities

As outlined in section III, there are quite a few job functions that a customer needs for a successful infrastructure implementation, perhaps a few more to keep the operations running day to day. Because of the close connection between technology and business, the technology personnel must make quite a few business decisions, and/or must contribute regularly to the design and implementation of the business plan.

Depending on their management's experience and skill level, the customer will understand some or all of the roles and responsibilities needed, but that does not mean those responsibilities have been assigned, nor all issues addressed. TLC will do as much as possible to leverage the customer's resources to increase efficiency and minimize TLC involvement, in effort to reduce fees and charges related to work rendered.

Managing the human and technical resources outside of AWS can be a shared responsibility between the customer and TLC to achieve those efficiencies. The corporation's management team should explain to their staff TLC's role and monitor their staff’s participation, accordingly.

Customers that make available competent management staff will reduce the time and effort TLC will spend doing management functions. Similarly, system administration and operational functions that can be effectively performed by the customer's staff will reduce TLC involvement and commensurate fees associated with those support functions.