# Deploying Immutable Infrastructures in AWS.

## Goal and Summary

The goal of this presentation is to explain the challenges of managing a scaling web app and how DevOps principles like immutable infrastructure can help address the challenges. We will explain concepts like ELB, Autoscaling, immutable infrastructure. We will also explain how the different components can automatically be scaled in AWS . There will also be a demo of how bootstrapping and autoscaling can be used effectively. If time allows we can show a demo of how Packer and Vagrant can be used to effectively test automation that will be used to deploy immutable infrastructure.

## Outline

Present a common n-tier web app infrastructure. Just draw it on white board. Speak about the challenges of scaling that application for load. Walk through the different layers and what can be done to scale.

Explain how AWS scales ELB and how Autoscaling addresses the scaling of the servers.

Introduce the problem of updating components like application updates, os updates, security updates and how that plays an infrastructure that scales on demand. The time to bootstrap needs to be minimized. Introducing new version of OS components can cause issues with applications if not tested properly.

Explain the challenge of how as the environment morphs; how do you ensure things are not going to break. How can you ensure the test in dev and qa will be the same as in prod.

Explain the concept of immutable infrastructure. Deploy everything once via an AMI and test that trough out all environments then deploy in prod.

Benefits of this approach

Things are well tested in all environments before they are deployed

Minimizes the amount of time for bootstrapping

Immutable boxes are more secure since no interaction is needed to update code

And more.

Demo time

Show how one instance can be launched and bootstrapped and the resulting instance can be converted into an AMI

Update the autoscaling group and deploy the new AMI.

Show the new version of the web page

Explain how tools like Vagrant and Packer can be used to simplify this entire process. Demo Ansible code with Packer and Vagrant.

Questions