

Deploying Application to Cloud

1. Cloud computing concepts
2. Introduction to Cloud based deployments

Contents

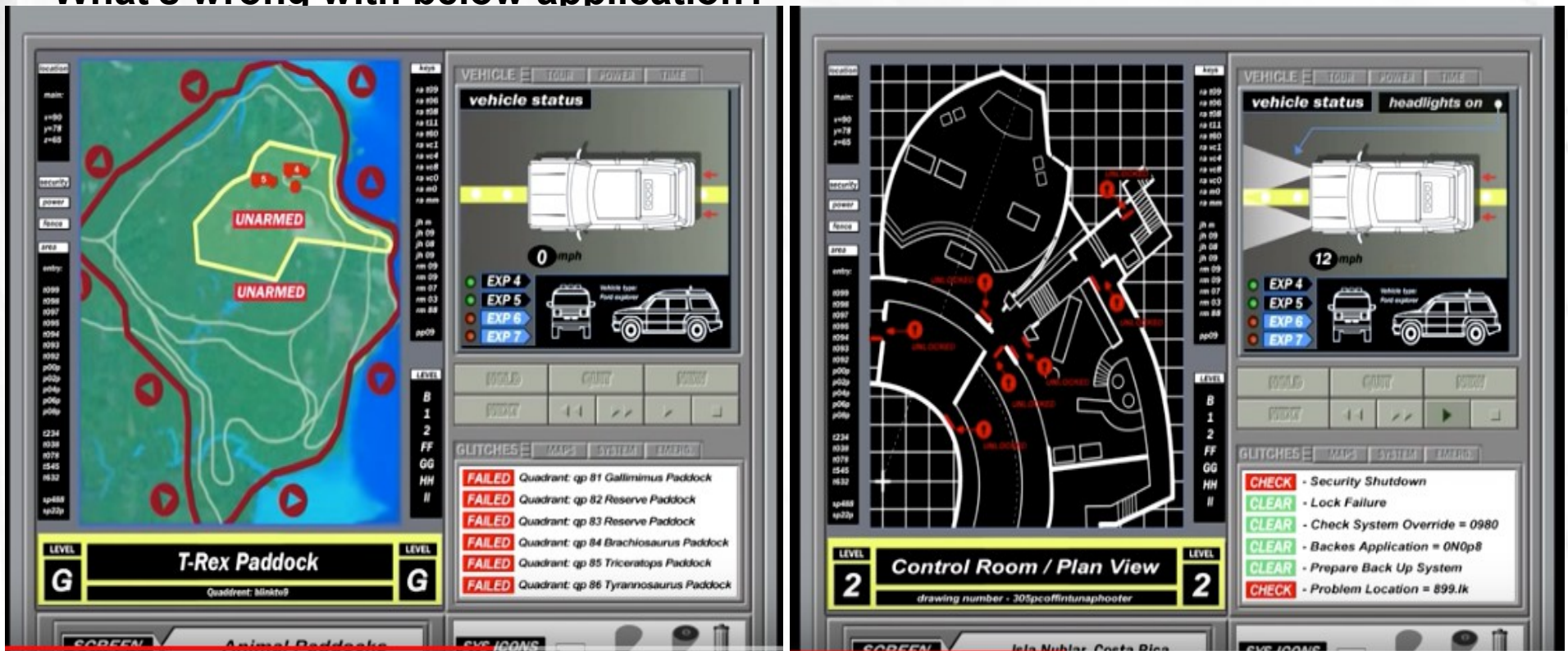
- Overview – Application development and Deployment
- Application Simplifications
- Cloud concepts
 - Containers
 - Virtual Cloud
- Containerized deployment on Google
 - Environment setup
 - Package your app into a Docker image
 - Run the container locally on your machine (optional)
 - Upload the image to a registry
 - Create a container cluster
 - Deploy your app to the cluster
 - Expose your app to the Internet
 - Scale up your deployment
 - Deploy a new version of your app

Building and deploying a computer application

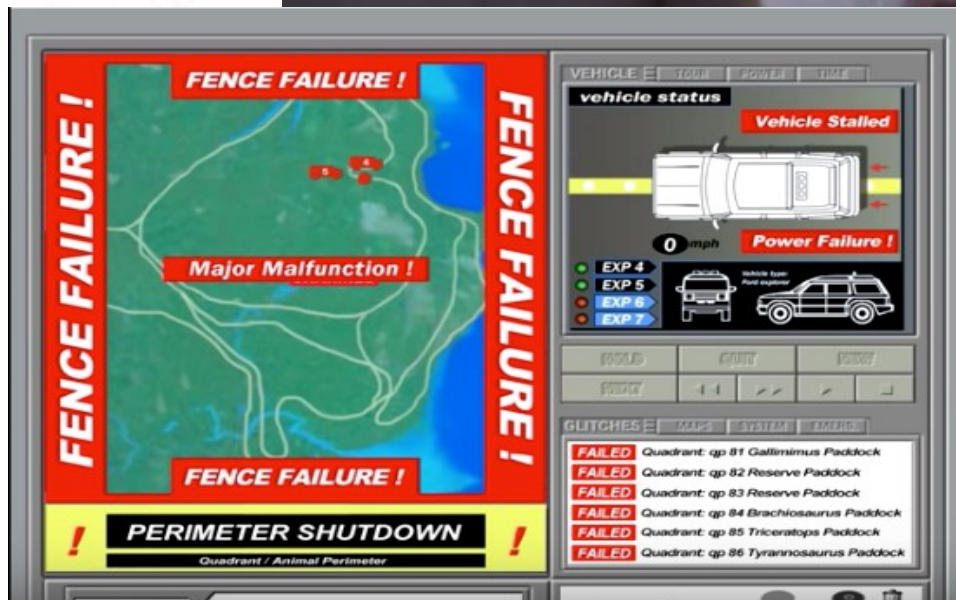
Typical stages involved in building a computer application are -

- Design and Development
- Testing – unit, functional and integration
- Deployment

What's wrong with below application?



Jurassic Park Controller App



What is wrong with above type of Systems?

- Single point of failure
- System not based on a modular design
- Singular deployment (no fail safe mode) on an dedicated environment
- Lengthier development time (recall 'Dennis Nedry' – the chief software engineer, complaining about how he is fixing one issues at a time inside the code! .. That gives us an idea about his development and testing strategy)
- Maintenance and Support is difficult / near impossible

Solutions ..

- **Consider**

- Modular design
- Distributed deployments that supports automated scaling, recovery, etc. For example use microservices architecture to design robust applications and containers for scalable deployments & testing.

- Using **Microservices**

- Allows us to divide and conquer complex applications
- Simplified scaling (refer tiger-styles example)
- Loose coupling
- Makes it easier to adopt to new technologies
- Autonomous and focused build (including testing), better productivity

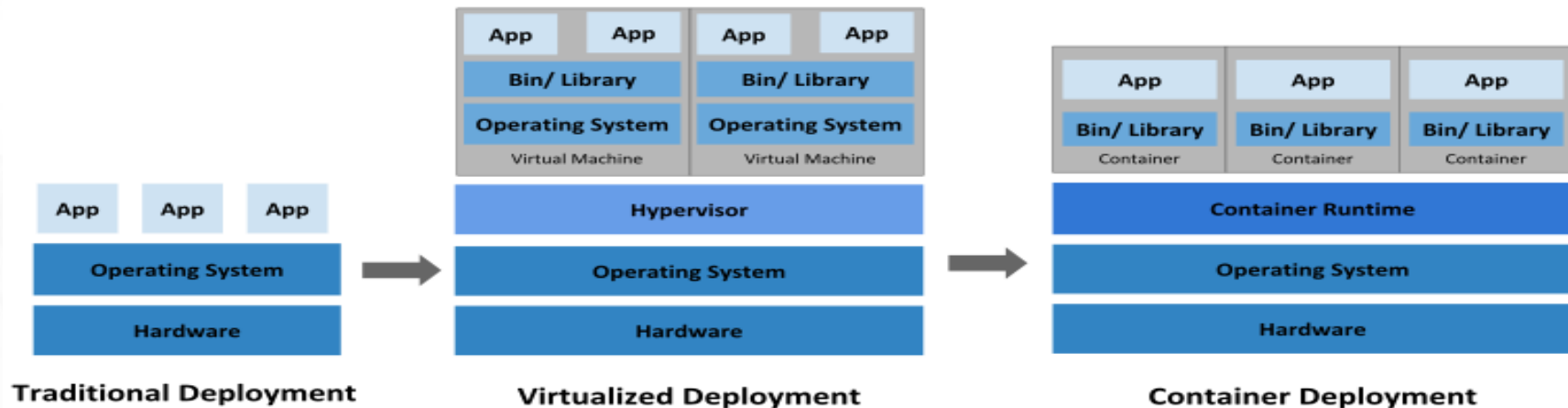
- What are **Microservices**?

Its a pattern for building distributed systems

- A suite of services, each running in its own process, each exposing an API
- Independently developed
- Independently deployable
- Each service is focused on doing one thing well

Example: <https://github.com/rnn0047/deployments-training>

• Simplified Application Deployment



Virtualized deployment allows us to run multiple virtual machine on single physical CPU thus allowing better utilization of resources with low h/w costs.

Containers are similar to virtual machine, but they have relax isolation properties to share the OS among applications. They offer more portability across cloud and OS distributions

Containers - It's a pattern for building distributed systems and offers separation of application from runtime environment

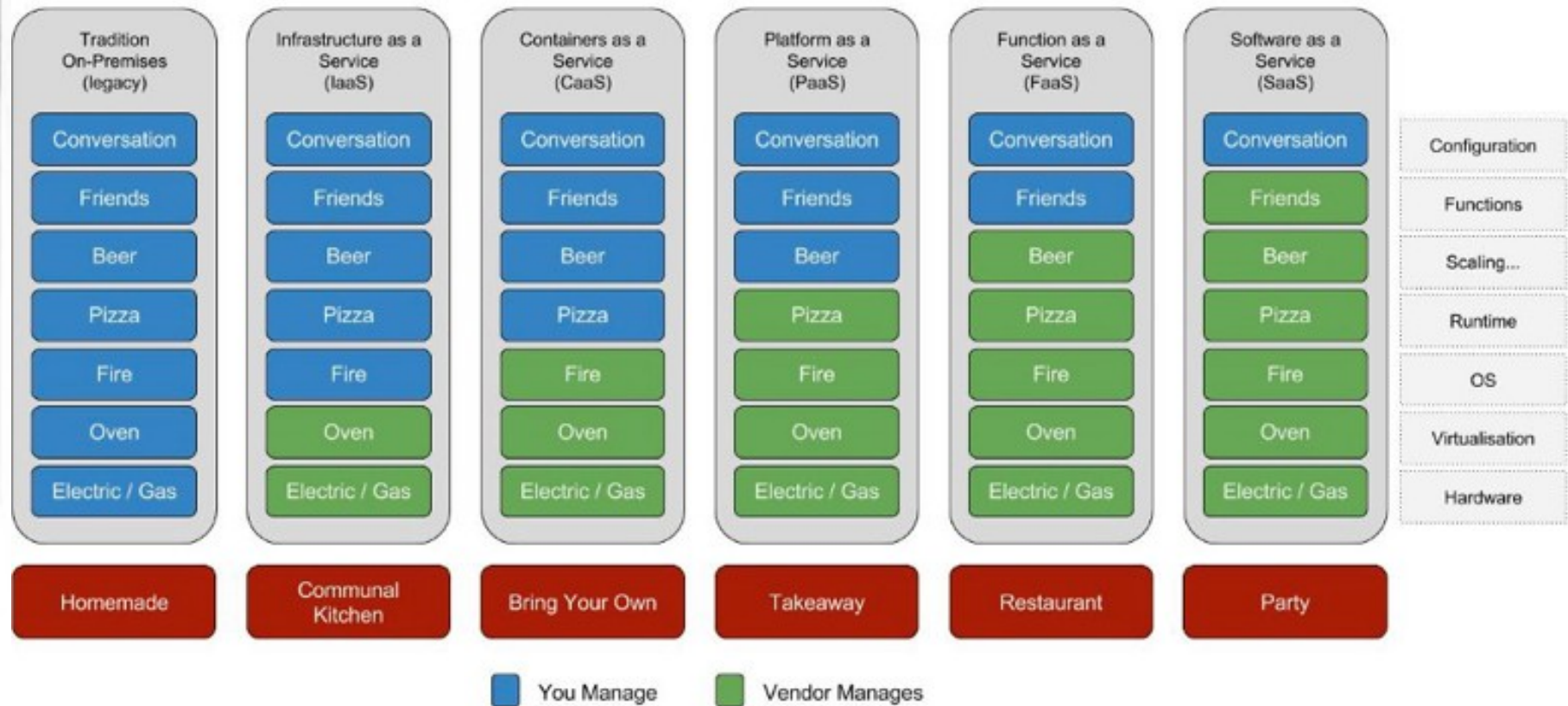
- A suite of services, each running in its own process, each exposing an API
- Independently developed
- Independently deployable
- Each service is focused on doing one thing well

For further details: <https://cloud.google.com/containers/>

- **Some key concepts ..**
- **Dockers** – tools for developing, deploying and running applications in containers. <https://docs.docker.com/get-started/>
- **Kubernetes** – orchestration tool for containers. Takes care of resiliency, scaling, service discover, storage orchestration, self healing, etc.
<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>
- **Container optimized OS** - these are VM's optimized for running docker containers on vendor platform like AWS, GOOGLE Cloud, Microsoft Azure, etc. They come with preinstalled tools like gcloud, kubernetes, kubectl, etc
- **Types of Cloud services for deployments** – On-Prem, IaaS, CaaS, PaaS, FaaS & SaaS (refer diagram on next slide)

- **Various flavor of Cloud services**

Using 'Pizza as a service' analogy



Deploying Containers to Google Cloud

Setup Environment

- Signup for free tier account at <https://cloud.google.com/free/> (enter name and project id that will be used to identify your project OR accept defaults)
- Make sure you have access to GCP console <https://console.cloud.google.com>
- Enable required APIs inside GCP (for example we need (a) GCE – Google compute engine. On Amazon it will be called EC2 instance. And (b) Deployment Manager API)
- [optional but recommended] Install and Authenticate Cloud SDK (<https://cloud.google.com/sdk/docs/quickstarts> or <https://cloud.google.com/sdk/docs/quickstart-windows>) which comes with command line tool like 'gcloud'. Authentication will be done at end of installation. Refer below snip and answer accordingly (it will also ask to name your project for deployment). This tool also downloads Docker and Kubectl
- [not required if step #3 is done] Set project id & compute zone using following commands
 - gcloud config set project [project id]
 - gcloud config set compute/zone us-east1-a
- Get Kubernetes command line tool using 'gcloud components install kubectl'. Run this command inside the Google Cloud SDK Shell (from start menu right click to run as administrator)

Setup Environment (continued from prior slide)

- Get Docker Community edition (<https://docs.docker.com/install/>) to build container image of the application
- Lastly get the git command line tools (<https://git-scm.com/downloads>)