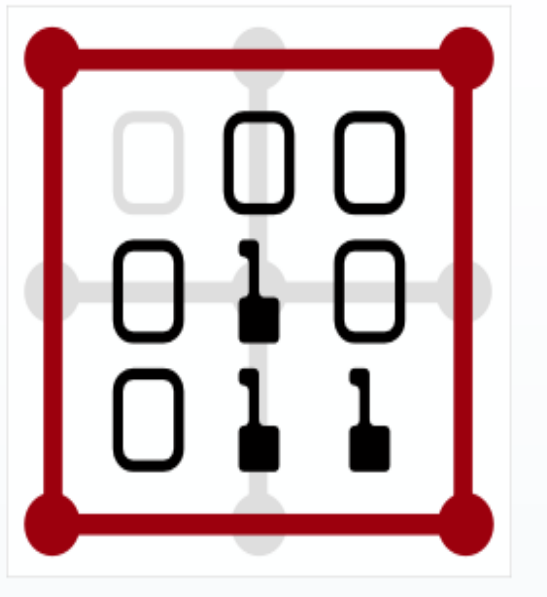




Bilgisayar Mühendisliği Bölümü

Bitirme Projesi



Yazılım Geliştirme ve Sistem Yönetiminde Sanal Makinelere Dayalı İş Akışı

Aydın Tunç DOYAK

Proje Danışmanı : Yrd. Doç. Dr. Recai Oktaş

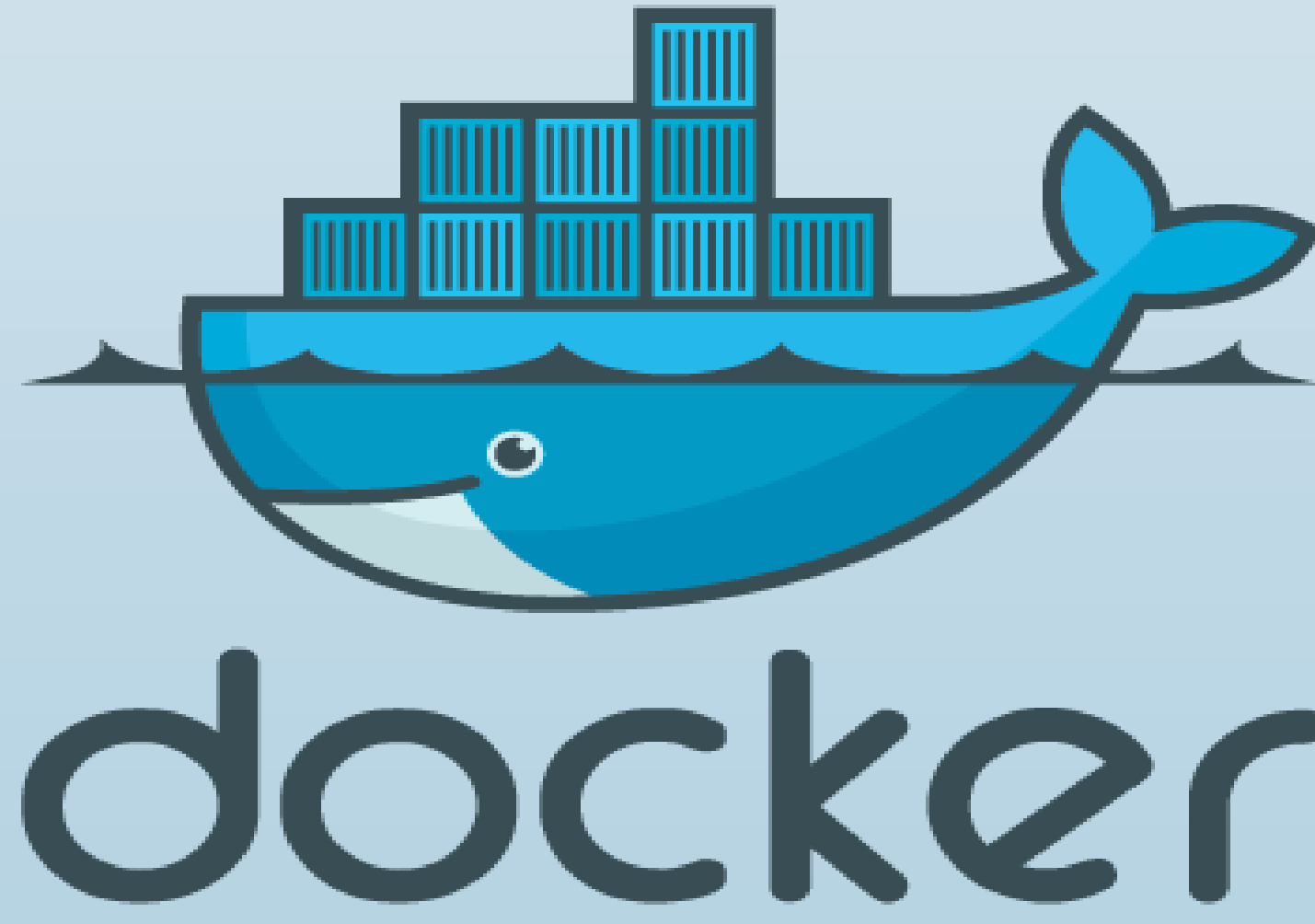
E-posta : aydintd@bil.omu.edu.tr

Özet : Günümüzde “üretim” seviyesindeki pek çok servis sanal makineler üzerinde konuşlanmakta ve bunun doğal bir uzantısı olarak “geliştirme” süreçleri de sanal makineler üzerinde gerçekleşmektedir. Projenin amacı geliştiricilerin rol aldığı “geliştirme” süreciyle, sistem yöneticilerinin rol aldığı “üretim” sürecinin belirli bir proje bağlamında daha uyumlu ve verimli bir şekilde gerçekleşmesini sağlayacak iş akışlarını modellemek ve bu iş akışlarını destekleyecek alt yapı ve araçları oluşturmaktır.

Anahtar Kelimeler : Sanal makine, PaaS (Platform as a Service), Packer, Vagrant, Docker, Dokku, VirtualBox, KVM

Giriş

Bu projede; üniversitemizde ve bölümümüzde geliştirme ve sistem yönetimi süreçlerinin birbirleriyle daha uyumlu gitmesi, geliştiricilerin ve sistem yöneticilerinin projelerde daha verimli sonuçlar elde edilebilmesi için sanal makineler ve günümüz sanal makine teknolojilerine dayalı altyapının modellenmesi ve gerekli araçların oluşturulması amacıyla bir sanal makine üretim fabrikasının oluşturulması, güncel ve ihtiyaca yönelik sanal makinelerin geliştiricilere ve sistem yöneticilerine test veya ürün makineler olarak sunulması ayrıca geliştirme sonrası ortaya çıkan ürünlerin sunucu bilgisayarlarda konuşlandırılması için bir **PaaS** (Platform as a Service) altyapısının kurulması ve işler hale getirilmesi amaçlanmaktadır.



Yöntem

Sanal makinelerin yönetimi için, test ve ürün makinelerin kurulması, konfigüre edilmesi ve günümüzde kullanılan sanal makine yönetim sistemlerinin (Proxmox, Xen) desteklediği sanal makine formatlarında oluşturulmaları için Packer kullanılıyor. Projenin ikinci ayağı olan Bilgisayar Mühendisliği Bölümünde kullanılmak üzere bir PaaS sistemi (Heroku vb.) kurulması, konfigüre edilmesi için ise Docker tabanlı Dokku sistemi kullanılıyor.

Packer

Sanal makine üretim fabrikasının altyapısını oluşturmak için **Packer** sanal makine üretim teknolojisi kullanılıyor. Packer, üretilmesi istenen sanal makine için inşa aracına (**hypervisor**) bir **JSON** dosyası aracılığıyla besleniyor. Bu JSON dosyasında hangi inşa aracının kullanılacağı, makineye hangi işletim sisteminin kurulacağı, sanal makinenin disk formatı, sanal makine için ayrılacak disk boyutu ve bellek miktarı, sanal makineler özellik kazandıracak **provision betiklerinin** yeri ve boot esnasında kullanıcıdan istenen bilgilerin bulunduğu ve işletim sisteminin türevine göre değişen bir preseed dosyasının yeri bulunuyor. Bu dosyalara girilen bilgiler doğru olduğu takdirde, tek bir build komutuyla, fabrika; sunucu makinenizin gücüne göre değişen zaman aralıklarında rahatlıkla sanal makineler üretebiliyor ve otomatik bir sanal makine üretim altyapınız oluyor. Üretilmiş boş sanal makineler **shell** gibi programlama dilleriyle ya da chef, puppet gibi **orquestrasyon** araçlarıyla rahatlıkla provision edilebiliyor. (Şekil 1.a)

PaaS

Platform as a Service alt yapısında **Dokku** servisi kullanılıyor. Dokku, arka planda **Docker** Linux Container Engine ile çalışan ve **Heroku Buildpack**lerini kullanan bir platform servisi. Yani bir Heroku klonu. Geliştiriciler, Dokku servisine **SSH** açık anahtarını göndererek, Heroku'nun desteklediği Rails, Python, PHP, Java gibi bir çok programlama dilinde geliştirilmiş web uygulamalarını **git** aracılığıyla push edebiliyorlar. Dokku kendisine bir uygulama push edildiğinde Heroku Buildpacklerini kullanarak hangi dilde yazıldığını algılıyor ve sunucu bilgisayarın kaynaklarını kullanan izole bir Docker konteynırına alıyor, sonra da otomatik olarak **nginx** web sunucusu ayarlarını yapıyor. Eğer uygulama bir **veritabanına** ihtiyaç duyuyorsa geliştiriciler Dokku'nun MySQL, PostgreSQL, MariaDB gibi veritabanlarını veritabanı eklentilerini Dokku servisine push edilen uygulamalara bağlayarak konuşlandırılabilir.

Sonuç

Fabrikada şu an Ubuntu ve Debian işletim sistemlerinin server ve desktop sürümlerinin sanal makineleri otomatik olarak üretilirebiliyor. Yani hiç bir sistem yöneticisi bu iki işletim sistemi için elle sanal makine kurmak zorunda kalmıyor. Bu projenin sistem yöneticilerine %80 zaman kazandıracakları öngörülmektedir. Dokku projesindeyse, bölümde kullanılan servislerin tek bir sisteme konuşlandırılıp, merkezi bir yerden yönetimini yapmak; kaynak yönetimi ve performans olarak sistem yöneticilerinin işini kolaylaştırıp, sistemi çok daha güvenli kılacağı öngörülmektedir.

Kaynaklar

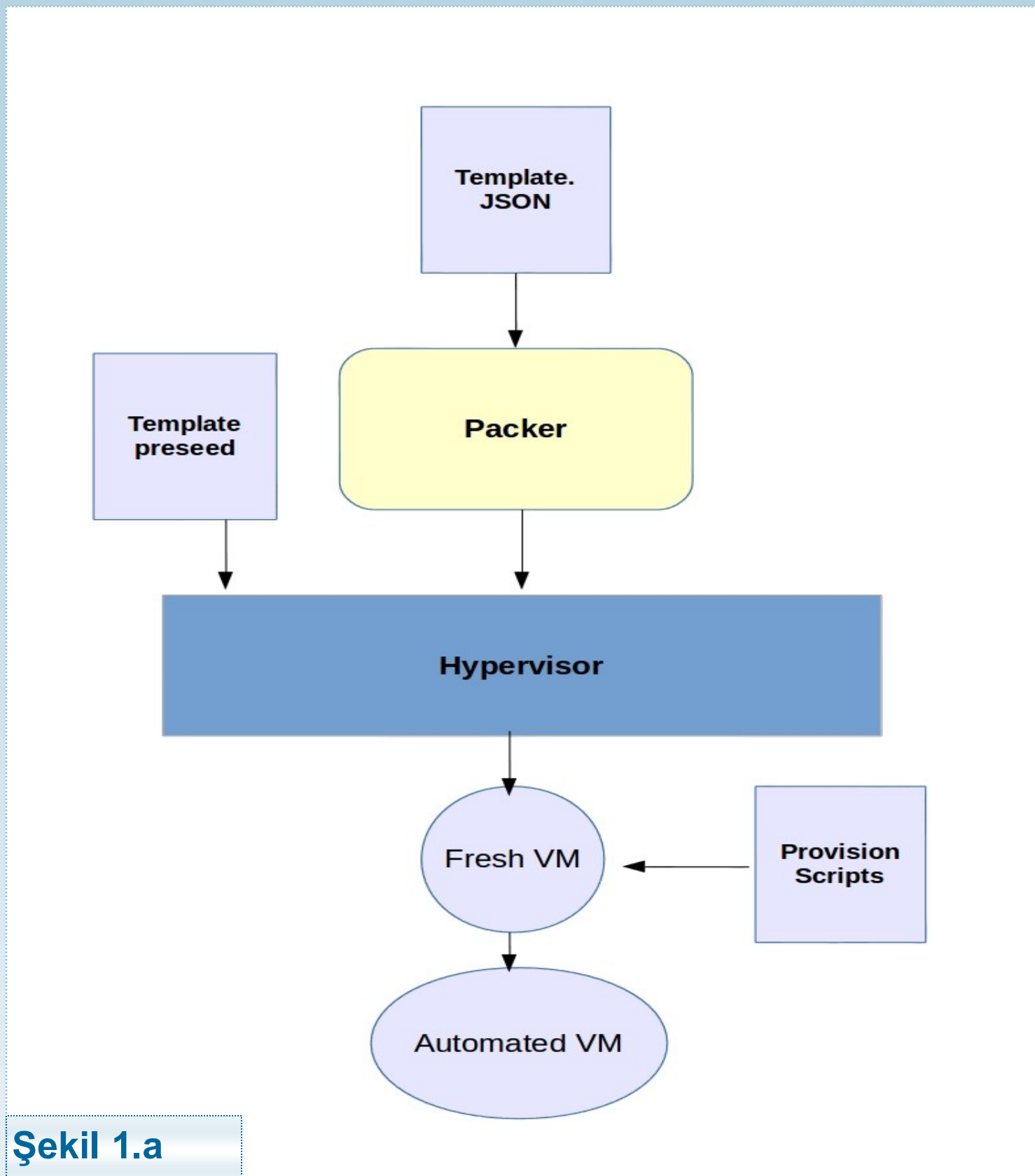
Packer : <http://packer.io/documents>

Vagrant : <http://vagrantup.com>

Fabrika : <http://github.com/aydintd/fabrika>

Docker : <http://docker.io>

Dokku : <http://github.com/progrium/dokku>



Şekil 1.a

Teknolojiler

Vagrant : Test ortamlarını kolaylıkla oluşturmak için kullanılan bir özgür yazılım

Packer : Sanallaştırma otomasyonu sağlayan çok faydalı bir paket program

Docker : Linux konteynır motoru

Github : Sürüm takip sistemi

Trello : İş takip sistemi

