T.C.
BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ
İKTİSADİ İDARİ BİLİMLER FAKÜLTESİ
YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ

SCALELİTE İLE BİGBLUEBUTTON ORGANİZASYONU VE ANSİBLE İLE TEMEL YAPILANDIRMA ÖLÇEKLENDİRME

Beyza YILMAZ

Gülçin AYDIN

DANIŞMAN

Dr. Öğr. Üyesi Hüseyin PARMAKSIZ

DÖNEM SONU PROJE

BİLECİK 2025

ÖZET

SCALELİTE İLE BİGBLUEBUTTON ORGANİZASYONU VE ANSİBLE İLE TEMEL YAPILANDIRMA ÖLÇEKLENDİRME

Bu çalışmada, açık kaynaklı video konferans sistemi BigBlueButton'un birden fazla sunucuda ölçeklendirilebilir şekilde yönetilmesi amacıyla kullanılan Scalelite yazılımı ve altyapının otomatikleştirilmesi için Ansible aracı incelenmiştir. Günümüzde çevrim içi eğitim ve toplantı sistemlerinin yaygınlaşmasıyla birlikte, bu tür çözümlerin sürdürülebilir, güvenilir ve esnek biçimde kurulup yönetilebilmesi büyük önem taşımaktadır.

Ödev kapsamında, VirtualBox ortamında Ubuntu sunucular kurulmuş; SSH bağlantıları üzerinden sistemlere erişim sağlanarak BigBlueButton'un temel kurulumu gerçekleştirilmiştir. Ardından, Scalelite ile bu sunucuların merkezi bir noktadan yönetimi sağlanmışve Ansible kullanılarak yapılandırma süreçleri otomatize edilmiştir. Çalışma boyunca temel düzeyden başlayarak adım adım uygulamalarla ilerlenmiş, karşılaşılan teknik sorunlar analiz edilip çözümleriyle birlikte belgelenmiştir.

Bu sürecin sonunda hem dağıtık sistemlerin nasıl organize edilebileceği hem de Ansible ile konfigürasyon yönetiminin nasıl daha verimli hâle getirilebileceği pratik örneklerle öğrenilmiştir. Bu ödev, sadece teknik bilgi kazandırmakla kalmamış, aynı zamanda sistem düşüncesi, problem çözme ve otomasyon kültürü konusunda da önemli katkılar sağlamıştır.

iÇİNDEKİLER

1 SCALELİTE İLE BİGBLUEBUTTON ORGANİZASYONU VE ANSİBLE İLE TEMEL YAPILANDIRMA ÖLÇEKLENDİRME

1.1 Projenin Amacı ve Hedeflenen Mimari

1.1.1 Amaç:

- Yerel ağda çalışan, halka açık alan adı ve sunucu gerektirmeyen, ölçeklenebilir bir BigBlueButton video konferans altyapısı kurmak.
- Yük dengeleme için Scalelite kullanmak.
- Kullanıcı dostu bir arayüz için Greenlight entegre etmek.
- Tüm sunucuların temel yapılandırma ve yönetimini Ansible ile otomatize etmek.

1.1.2 Hedeflenen Mimari:

- VM1 (bbb-1): BigBlueButton Sunucusu (10.89.4.7)
- VM2 (bbb-2): BigBlueButton Sunucusu (10.89.2.173)
- VM3 (scalelite): Scalelite Yük Dengeleyici, Greenlight Arayüzü, Ansible Kontrol Noktası, PostgreSQL Veritabanı Sunucusu, Redis Veritabanı Sunucusu (hepsi Docker container'ları içinde) (10.89.11.157)
- Host Makine (Windows): Sisteme erişim ve test için.

1.2 Kullanılan Temel Teknolojiler ve Kavramlar

1.2.1 BigBlueButton (BBB):

- Açık kaynaklı, web tabanlı video konferans sistemi.
- Toplantıların fiilen çalıştığı ana motor.

1.2.2 Scalelite:

- Birden fazla BBB sunucusu arasında yük dengelemesi yapan, toplantı isteklerini dağıtan yönlendirici.
- Ölçeklenebilirlik için kritik.
- Aktif toplantıların meta verilerini ve sunucu durumlarını saklamak için PostgreSQL'e,

geçici veriler ve iletişim için Redis'e ihtiyaç duyar.

1.2.3 Greenlight:

- BigBlueButton için kullanıcı dostu web arayüzü.
- Oda oluşturma, kullanıcı yönetimi gibi işlevler sunar.
- Kullanıcı ve oda bilgilerini saklamak için PostgreSQL'e, oturum yönetimi ve önbellekleme için Redis'e bağlanır.

1.2.4 Ansible:

- BT otomasyon aracı.
- Sunucu yapılandırması, yazılım dağıtımı ve karmaşık iş akışlarını otomatize etmek için kullanılır.

1.2.5 Docker Docker Compose:

- Container (kapsayıcı) teknolojisi.
- Scalelite, Greenlight, PostgreSQL ve Redis'i kendi izole ortamlarında kolayca çalıştırmamızı sağladı.
- docker-compose.yml dosyası ile bu çoklu container uygulamasının tamamı tanımlandı ve yönetildi.

1.2.6 Nginx:

- Yüksek performanslı web sunucusu ve ters proxy.
- Scalelite sunucumuzda (VM3), Docker içinde çalışarak https://scalelite.local adresine gelen istekleri karşıladı ve isteğin yoluna göre ya Greenlight arayüzüne ya da Scalelite API'sine yönlendirdi.
- SSL sonlandırmasını (kendi kendine imzalanmış sertifikalarla) Nginx yaptı.

1.2.7 PostgreSQL Veritabanı:

Ne İşe Yarar: Güçlü, açık kaynaklı, ilişkisel bir veritabanı yönetim sistemi. Verilerin kalıcı, düzenli ve güvenli saklanmasını sağlar.

Projedeki Amacı: *Scalelite İçin:* BBB sunucularının durumları, aktif toplantıların meta verileri, kayıt bilgileri gibi kritik verileri tutar. *Greenlight İçin:* Kullanıcı hesapları, oluşturulan odalar, oda ayarları, kullanıcı rolleri gibi uygulama verilerini saklar.

Nasıl Kullanıldı: docker-compose.yml içinde tek bir postgres servisi tanımlandı. Hem Scalelite API (scalelite adında bir veritabanı) hem de Greenlight (greenlight*productionadndabirveritaban)bute*

1.2.8 Redis Veri Deposu:

Ne İşe Yarar: Hızlı, anahtar-değer tabanlı, genellikle bellekte çalışan (in-memory) bir veri yapısı sunucusu. Önbellekleme, oturum yönetimi, mesaj kuyrukları için kullanılır. **Projedeki Amacı:** *Scalelite İçin:* BBB sunucularından gelen anlık durum bilgileri, kısa süreli görevler veya servisler arası iletişim için. *Greenlight İçin:* Kullanıcı oturum bilgilerini saklamak, sık erişilen verileri önbelleğe almak, arka plan görevlerini yönetmek için. **Nasıl Kullanıldı:** docker-compose.yml içinde tek bir redis servisi tanımlandı. Hem Scalelite API hem de Greenlight, kendi .env dosyalarındaki REDIS*U RLdeikeniaracylabutekRediscontainer'nabaland.*

1.2.9 Yerel Alan Adları (.local) ve hosts Dosyası:

- Halka açık DNS olmadan, yerel ağımızda makinelere isimle (bbb-1.local, scalelite.local) erişmemizi sağlayan yöntem.
- Her makinenin kendi hosts dosyasına (Linux'ta /etc/hosts, Windows'ta C:32) bu isimlerin hangi IP adreslerine karşılık geldiği yazıldı.

1.2.10 Kendi Kendine İmzalanmış SSL Sertifikaları (Self-Signed SSL):

- https:// üzerinden güvenli bağlantı için gerekli.
- openssl ile scalelite.local için üretildi ve Scalelite sunucusundaki Nginx tarafından kullanıldı. BBB sunucuları da kendi .local adresleri için kendi sertifikalarını üretti.
- Tarayıcılar bu sertifikalara başta güvenmez, uyarı verirler. Servislerin birbirine güvenmesi için ek ayarlar gerekti.

1.3 Kurulum Adımları ve Yaşanan Zorluklar

1.3.1 Sanal Makinelerin Hazırlanması

- 3 adet Ubuntu 22.04 sanal makine kuruldu.
- Ağ yapılandırması "Bridge Adapter" modunda yapıldı.
- Statik IP adresleri atandı: VM1 (bbb-1): 10.89.4.7, VM2 (bbb-2): 10.89.2.173, VM3 (scalelite): 10.89.11.157.

hosts Dosyası Ayarları:

- Her 3 sanal makinenin /etc/hosts dosyasına diğer tüm makinelerin IP ve .local isimleri eklendi.
  - Windows ana makinenin hosts dosyasına sadece 10.89.11.157 scalelite.local eklendi.

1.3.2 BigBlueButton Sunucularının Kurulumu (VM1 VM2)

Temel sistem güncellemeleri yapıldı (sudo apt update sudo apt upgrade -y)..

BigBlueButton kurulum script'i (bbb-install.sh) v3.0.x sürümü için indirildi ve çalıştırılabilir yapıldı.

2.7 versiyonunun kurulum dokümantasyonunu bulabildik fakat en güncel versiyonun 3.0 olması sebebiyle ilgili github reposunu bularak doğru kurulum script'i ile aşağıdaki şekilde kurulumu yapıldı.

Denemeler sonucu -v jammy-300 parametresinin çalışığı tespit edildi. Komut: sudo ./bbb-install.sh -v jammy-300 -s bbb-X.local -g.

Kurulum sonrası sudo bbb-conf –secret ile her BBB sunucusunun URL ve Secret bilgileri alınıp kaydedildi.

1.3.3 Scalelite Sunucusunun Kurulumu (VM3 - Scalelite Ansible Veritabanları)

- Gerekli paketler kuruldu: ansible, git, docker.io, docker-compose.

- Kullanıcı Docker grubuna eklendi ve SSH oturumu yeniden başlatıldı.
- .env Dosyası (/scalelite-run/dotenv dosyasından cp dotenv .env ile oluşturuldu):
  - DOMAIN NAME ayarlandı.
  - SECRET KEY BASE ve LOADBALANCER SECRET ayarlandı.
  - BIGBLUEBUTTON SERVERS ayarlandı.
  - REDIS URL ayarlandı.
  - DATABASE URL ayarlandı.

- Kendi Kendine İmzalanmış SSL Sertifikaları (scalelite.local için):
  - /scalelite-run/secrets klasörü oluşturuldu.
  - sertifika ve anahtar bu klasöre üretildi.

- Bu dosya zaten postgres ve redis servislerini içeriyordu.
- **Scalelite Veritabanı Hazırlama:** db:setup komutu çalıştırıldı.

1.3.4 Nginx Proxy (scalelite-proxy) Yapılandırması (Manuel Olarak)

- 1. /scalelite-run içinde mkdir nginx ile klasör oluşturuldu.
- 2. nano nginx/scalelite.conf ile özel Nginx yapılandırma dosyası oluşturuldu.
  İçeriği:
- Port 80'den 443'e (HTTPS) yönlendirme.
- Port 443 için SSL ayarları (oluşturduğumuz secrets/scalelite.crt ve secrets/scalelite.key dosyalarını işaret edecek şekilde).
- Artırılmış zaman aşımı (timeout) ayarları.

1.4 docker-compose.yml dosyası (/scalelite-run içindeki) düzenlendi:

- scalelite-proxy: servisi için image: nginx:1.25-alpine (standart Nginx imajı) kullanıldı.
- Bu servisin environment ve command satırları silindi (artık Nginx'i imajın kendi scripti'leri değil, bizim verdiğimiz config yönetecek).
- volumes bölümüne ./nginx/scalelite.conf:/etc/nginx/conf.d/default.conf ve ./secrets:/etc/nginx/ssl:ro satırları eklenerek bizim özel yapılandırmamız ve sertifikalarımız container'a bağlandı.

1.4.1 Greenlight Arayüzünün Kurulumu ve Entegrasyonu (VM3)

- docker-compose.yml dosyasına greenlight: servisi eklendi:
  - image: bigbluebutton/greenlight:v3
  - container name: scalelite-greenlight
  - env file: ./greenlight.env
  - depends on: (Diğer servislerin önce başlaması için)

- greenlight.env dosyası (/scalelite-run içinde) oluşturuldu ve yapılandırıldı:
  - SECRET KEY BASE: openssl rand -hex 64 ile üretildi (önce docker run ... rake secret denendi ama container içinde curl ve openssl olmaması/takılması gibi sorunlar yaşandı).
  - BIGBLUEBUTTON ENDPOINT: https://scalelite.local/bigbluebutton/api olarak değişirildi.
  - BIGBLUEBUTTON SECRET: Scalelite'in ana .env dosyasındaki SECRET KEY BASE değeri (a4818f...) kullanıldı.
  - ALLOW SELFSIGNED CERTIFICATE=true: Eklendi (kendi kendine imzalanmış sertifikaya güvenmesi için).
  - DATABASE URL= (Aynı PostgreSQL servisinde Greenlight için ayrı bir veritabanı).
  - REDIS URL=redis://redis:6379 (Aynı Redis servisi).

Zorluk: Greenlight "Toplantıyı Başlat" denildiğinde "bir sorun çıktı" hatası veya dönen düğme.

- Hata Ayıklama: Tarayıcı Konsolu (NotFoundError: Failed to execute 'insertBefore' on 'Node' gibi JavaScript hataları), Greenlight logları, Nginx logları ve Scalelite API logları incelendi. Sorunun Greenlight'ın Scalelite API'sine isteği gönderememesi veya zaman aşımına uğraması olduğu anlaşıldı fakat SSL problemlerinin giderilememesi sebebi ile çözülemedi.

1.4.2 Ansible Kontrol Noktasının Kurulumu (VM3)

- Ansible zaten VM3'e kurulmuştu.

- Şifresiz SSH Erişimi Ayarlandı:
  - scalelite (VM3) üzerinde ssh-keygen ile anahtar üretildi.
  - ssh-copy-id ubuntu@bbb-1.local ve ssh-copy-id ubuntu@bbb-2.local ile anahtarlar BBB sunucularına ubuntu kullanıcısı için kopyalandı.

- inventory.ini Dosyası Oluşturuldu (/ansible/inventory.ini):
  - bbb servers grubu (bbb1, bbb2).
  - scalelite server grubu (sl host için ansible connection=local).
  - all:vars altında ansible user=ubuntu.

- **Bağlantı Testi:** ansible all -i /ansible/inventory.ini -m ping ile tüm sunuculara erişim doğrulandı.

1.5 Sistemin Test Edilmesi (Hedeflenen Durum)

1.6 Scalelite Yük Dengeleme:

Amaç: Scalelite'in yükü bbb-1 ve bbb-2 arasında dağıtıp dağıtmadığını görmek. **Yöntem:** Greenlight (https://scalelite.local) üzerinden aynı anda iki veya daha fazla farklı toplantı başlatmak. **Doğrulama:**

- Scalelite API'si üzerinden getMeetings çağrısı yaparak (API-Mate veya curl ile, Scalelite'in LOADBALANCER SECRET'ı kullanılarak). Gelen XML yanıtında her toplantı için <serverID> (örn: bbb-1, bbb-2) kontrol edilir.
- Alternatif olarak her BBB sunucusuna SSH ile bağlanıp sudo bbb-conf –status ile aktif toplantı sayılarına bakılır.

(**Not:** Bu test, toplantı başlatma sorunu tam olarak çözüldüğünde tam anlamıyla yapılabilir.)

1.7 Ansible ile Yönetim:

Amaç: Ansible'ın tüm sunucuları merkezi olarak yönetebildiğini görmek. *Yapılanlar:* inventory.ini dosyası bbbservers ve scalelite server gruplarıyla ve ansible user=ubuntu ile yapılandırıldı. Ad-hoc Komutlar: ansible all -a "df -h", ansible bbb servers -b -a

"bbb-conf –status", ansible scalelite server -a "docker-compose -f /home/ubuntu/scalelite-run/docker-compose.yml ps" gibi komutlarla anlık kontroller yapıldı.

Playbook Örnekleri: check bbb status.yml: BBB sunucularının durumunu alıp göstermek için. create and show report.yml: Her sunucuya o sunucu hakkında bilgi içeren bir dosya oluşturup, içeriğini Ansible çıktısında göstermek için. **Sonuç:** Ansible ile tüm sunuculara komut gönderilebildiği, bilgi toplanabildiği ve otomasyon yapılabildiği doğrulandı.

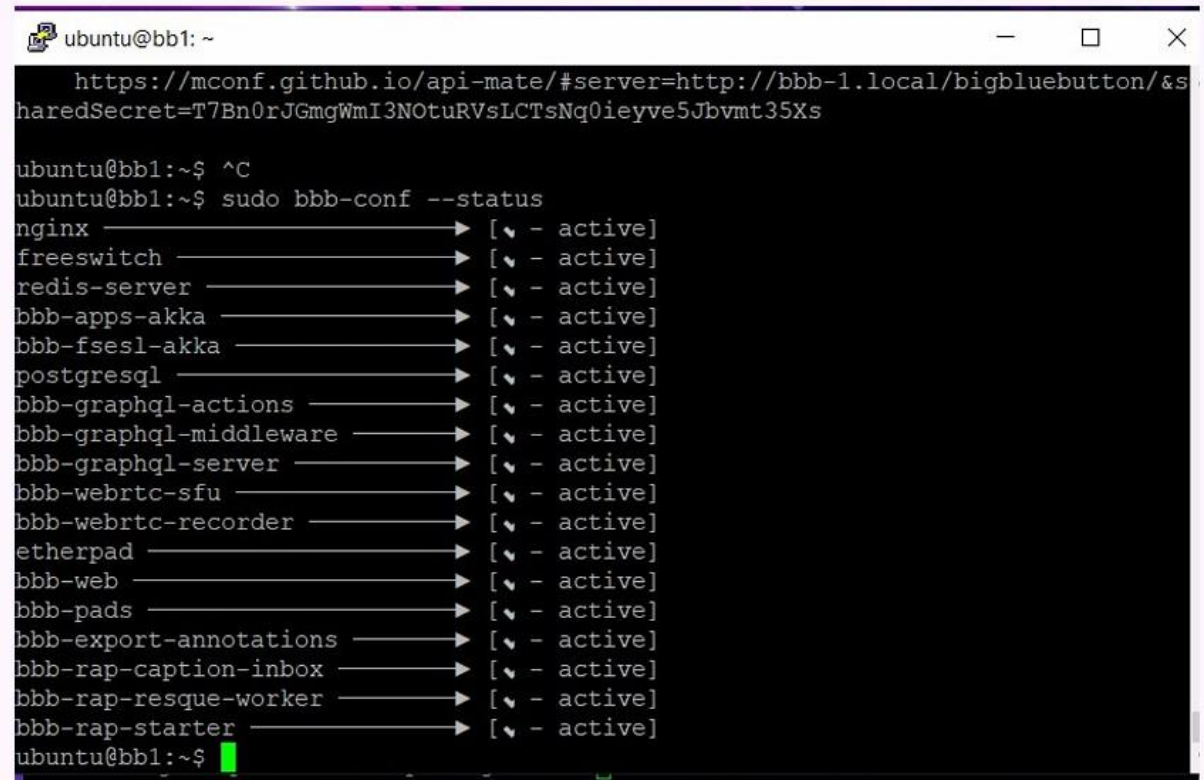
1.8 Projenin Mevcut Durumu ve Sonraki Adımlar (Toplantı Başlatma Sorunu Özelinde)

1.8.1 Mevcut Durum:

BBB sunucuları çalışıyor. Scalelite API'si curl ile test edildiğinde SUCCESS dönüyor. Greenlight arayüzü açılıyor, oda ve kullanıcı oluşturulabiliyor. Ancak Greenlight üzerinden "Toplantıyı Başlat" denildiğinde sorun yaşanıyor. En son Greenlight logları, kendi iç API'sine yaptığı POST .../start.json isteğinde status=400 ve 30 saniyelik bir duration gösteriyordu; Nginx ve Scalelite API loglarına ise bu sırada bir istek düşmüyordu.

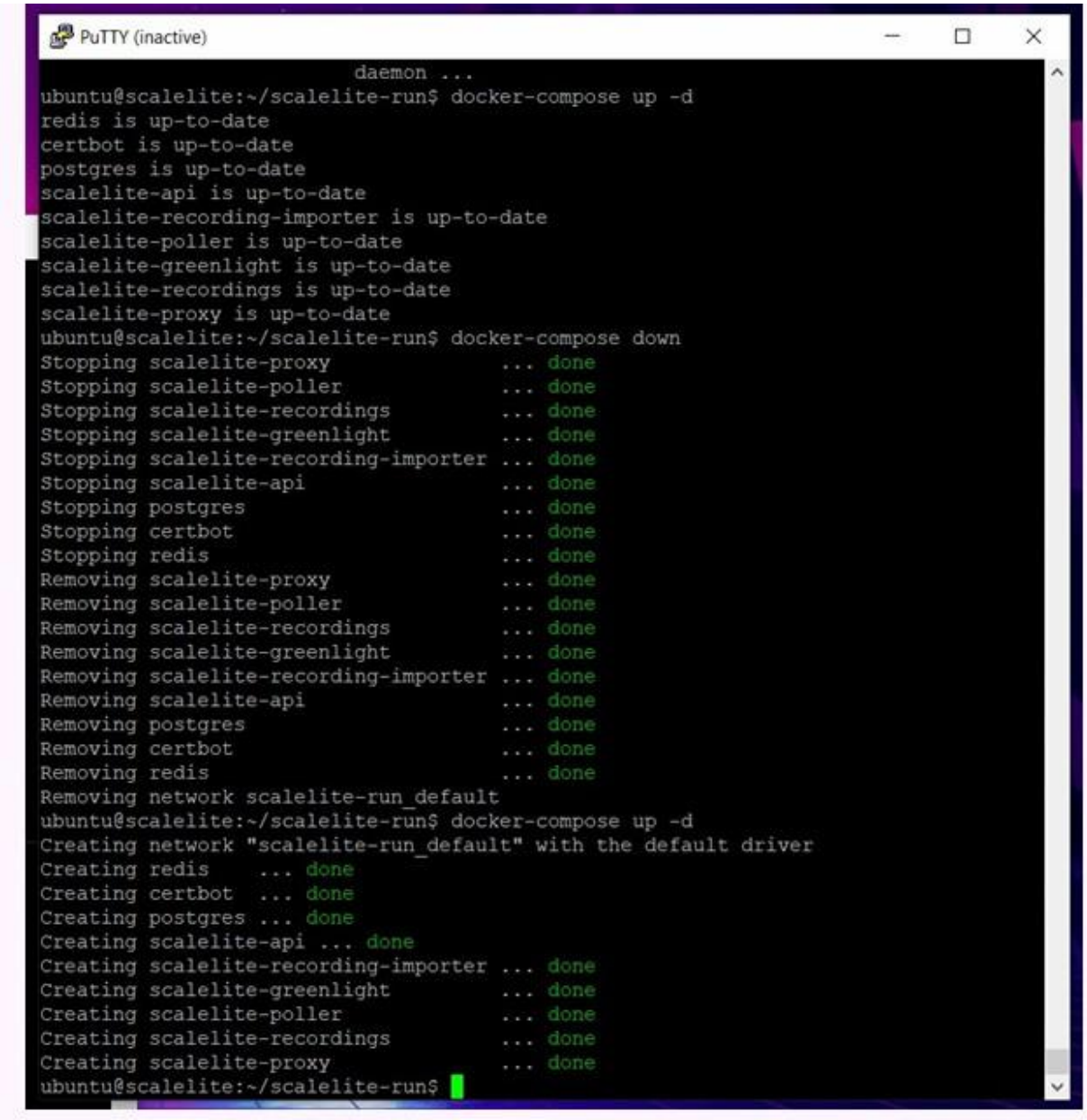
1.8.2 En Son Denenen Çözüm:

Greenlight'ın .env dosyasındaki BIGBLUEBUTTON ENDPOINT'in http://scalelite.local/bigbluebutton olarak ayarlanması ve Nginx'in bu HTTP isteğini alıp HTTPS'e yönlendirerek Scalelite API'sine iletmesi.

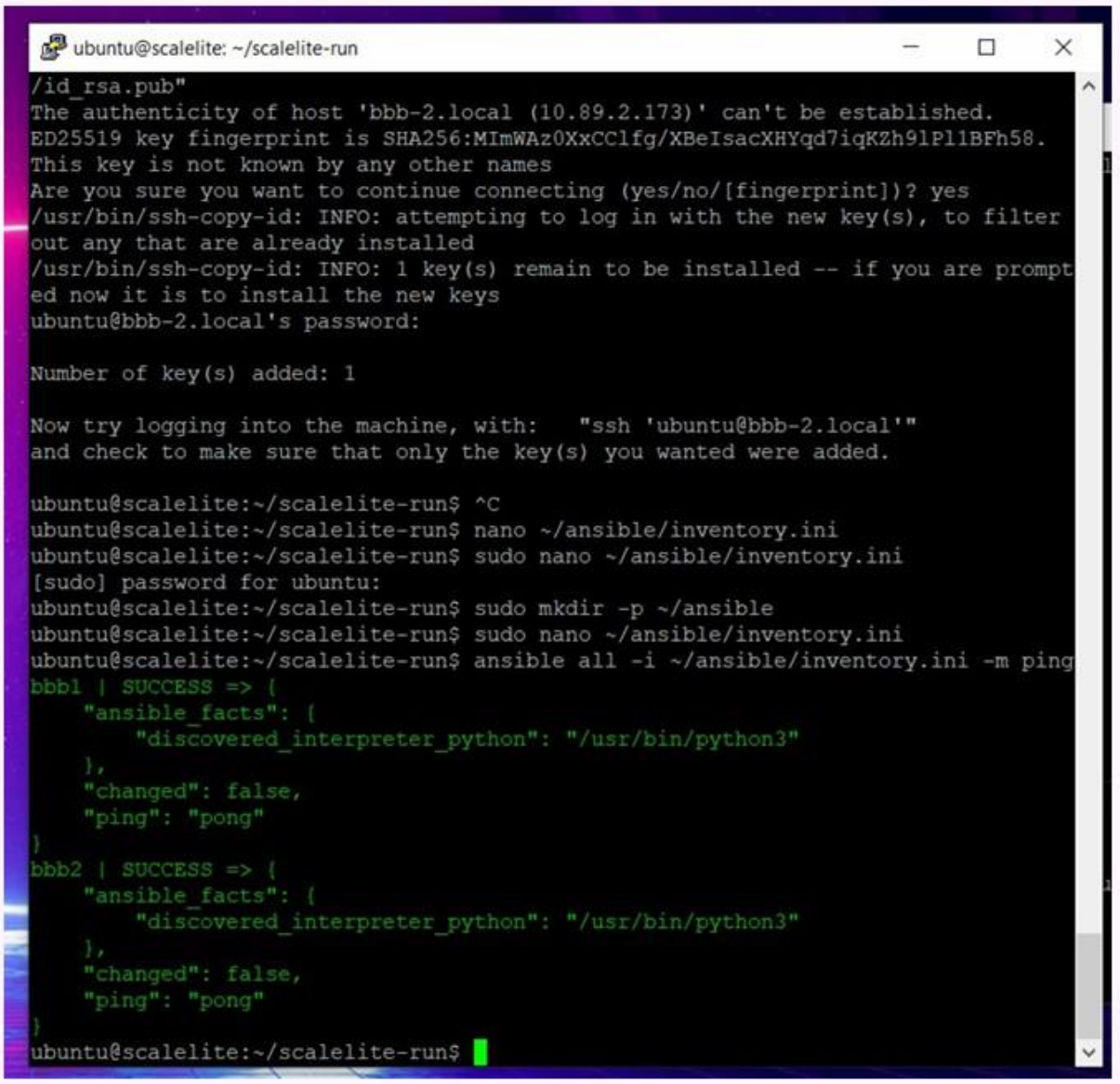


Şekil 1.1: BigBlueButton Sunucularının Kurulumu

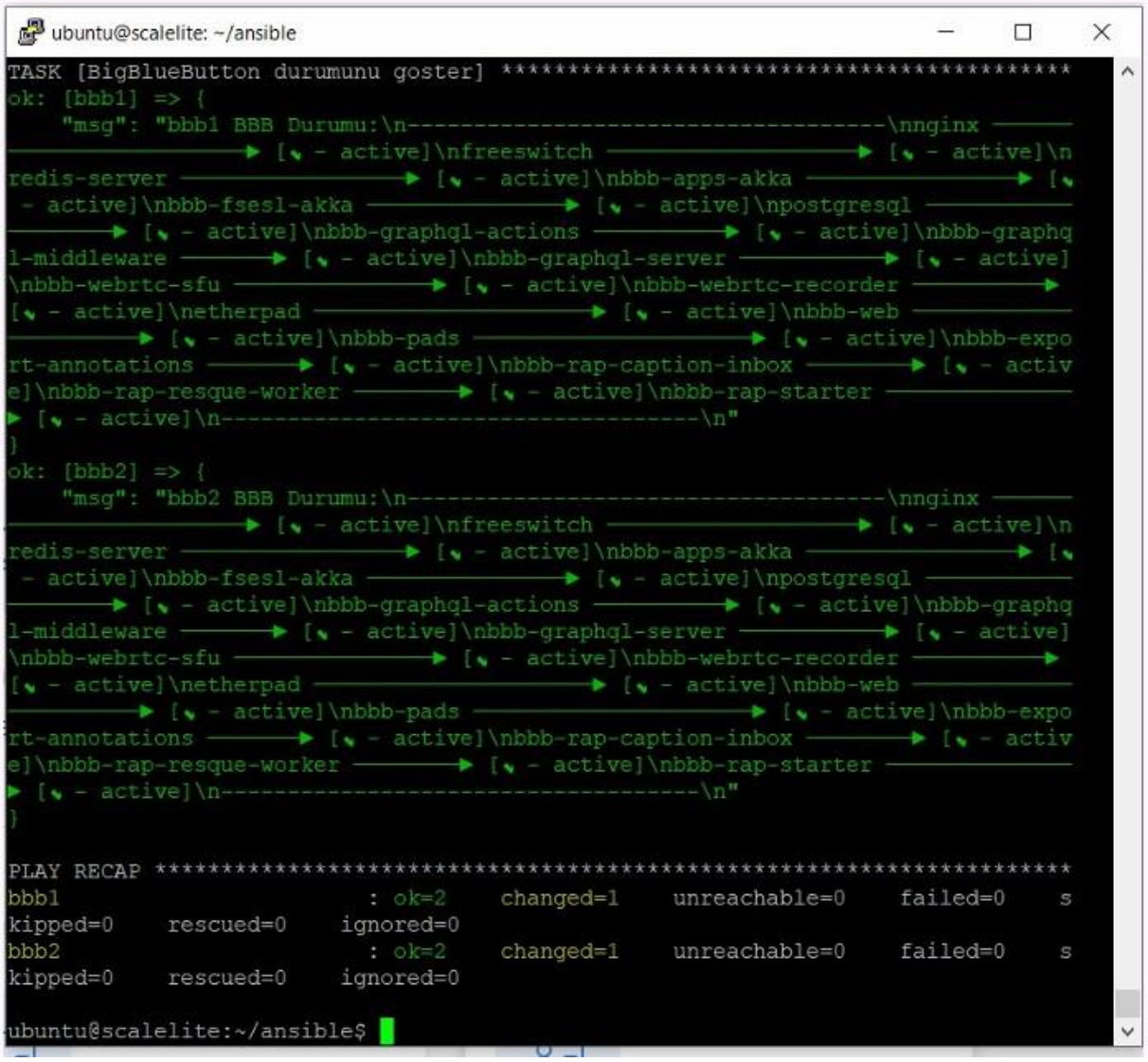
Şekil 1.2: Scalelite Veritabanı Hazırlama



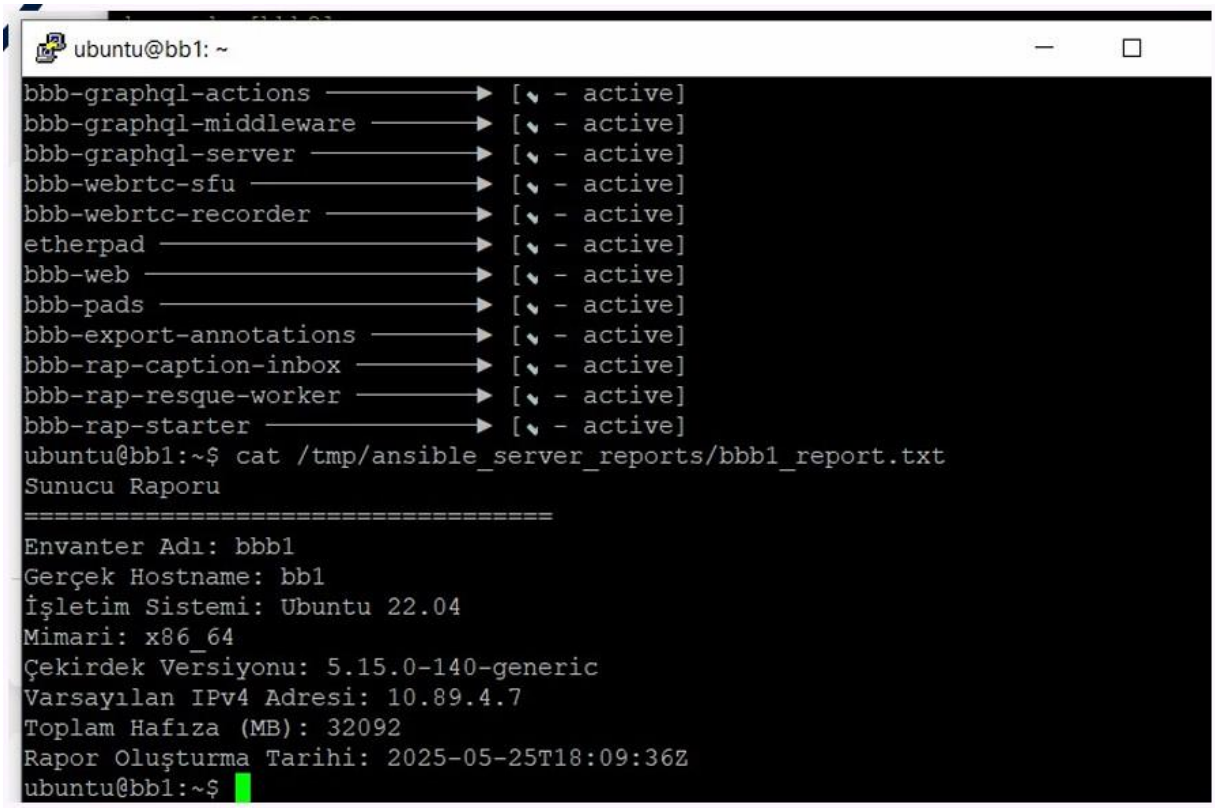
Şekil 1.3: Greenlight Arayüzünün Kurulumu ve Entegrasyonu



Şekil 1.4: Ansible Kontrol Noktasının Kurulumu



Şekil 1.5: Playbook Örnekleri



Şekil 1.6: Playbook Örnekleri

2 SONUÇLAR VE ÖNERİLER

Başarı Yolunda: Hedeflenen temel altyapı büyük ölçüde kuruldu. Çoğu servis çalışır durumda. "Toplantı başlatma" sorunu çözüldüğünde sistem tam fonksiyonel hale gelecek.

Öğrenilenler:

- Çok katmanlı (Greenlight -> Nginx -> Scalelite API -> BBB) ve kendi kendine imzalanmış SSL sertifikaları kullanan yerel sistemler kurmak, beklenenden daha fazla detay ve hata ayıklama gerektirebilir.
- Loglar (Docker, Nginx, Uygulama logları) ve tarayıcı geliştirici araçları, sorunların kök nedenini bulmada vazgeçilmezdir.
- Sistematik hata ayıklama, yani bir seferde tek bir şeyi değiştirip test etme ve "temiz bir sayfa"dan başlama (gerektiğinde eski konfigürasyonları silme), karmaşık sorunların çözümünde önemlidir.