Project 2

Cloud Base Software Integration System

Contents

Introduction

Cloud Base Software Integration System (SaaS) is not a new term for building an enterprise system where different applications can be connected and managed from the developing stage to the build and deploy stages. The SaaS or widely known as Enterprise Systems are configured to work together to facilitate the efficient definition, development, test and deployment of a software product. In this paper, we will install and configure a Cloud Base Software Integration System step by step. The objective outcomes:

- Deploy and configure the tools to support an enterprise software development environment.
- Create and configure Continuous Integration pipelines for software builds.
- Create and configure Continuous Delivery pipelines for automated deployments.
- Setup software development workflows within the enterprise software development environment.

Model

In this paper, we will follow the environment model in Figure 1.

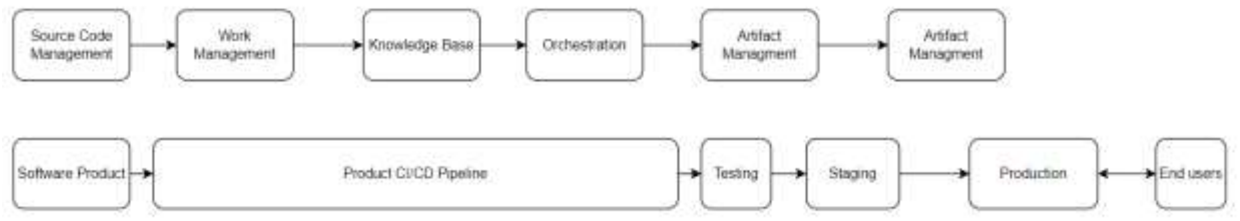
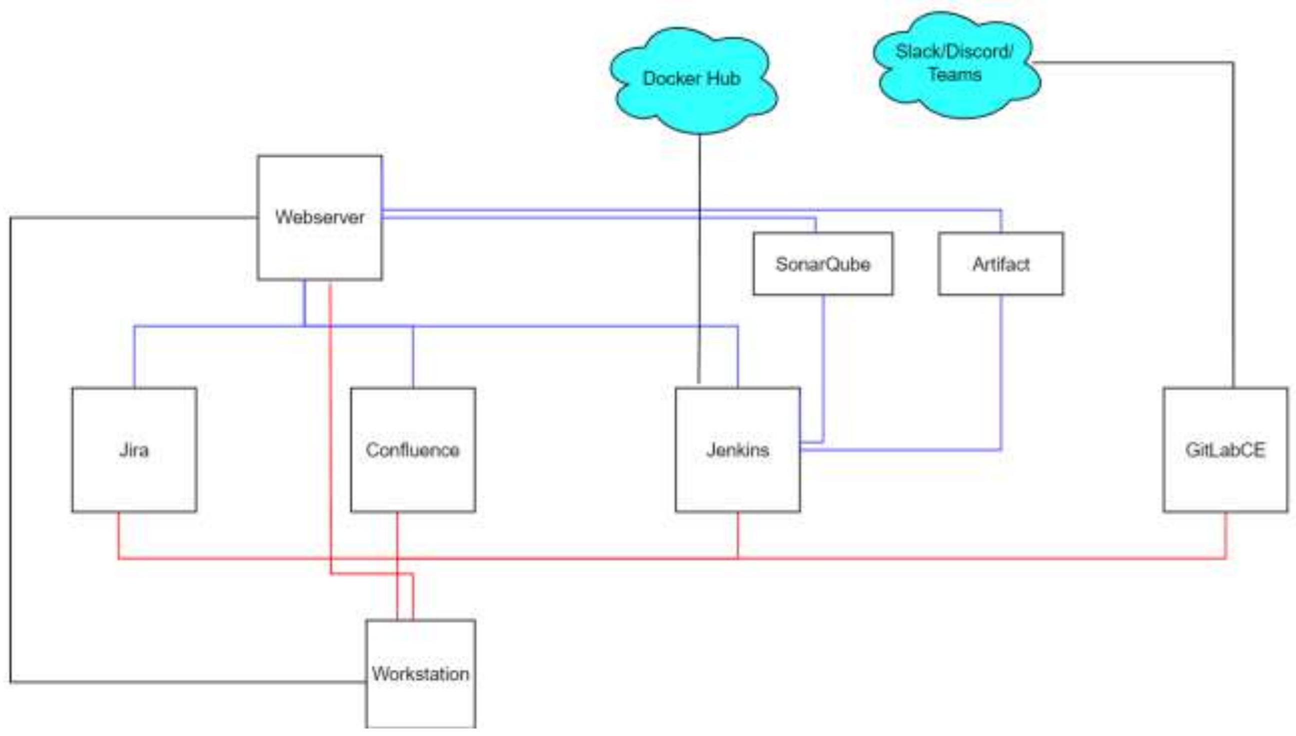


Figure 1: SaaS Development Environment

Once all services and servers are installed and deployed, our system will be connected as shown in Figure 2.



Legends:

+ **Red** line is for internal access such as SSH.

+ **Black** line is for direct access via browser.

+ **Blue** line is for internal routing from one service to another automatically.

Noted that all the servers are hosted in Azure in the same availability zone.

GitLabCE

Environment setup

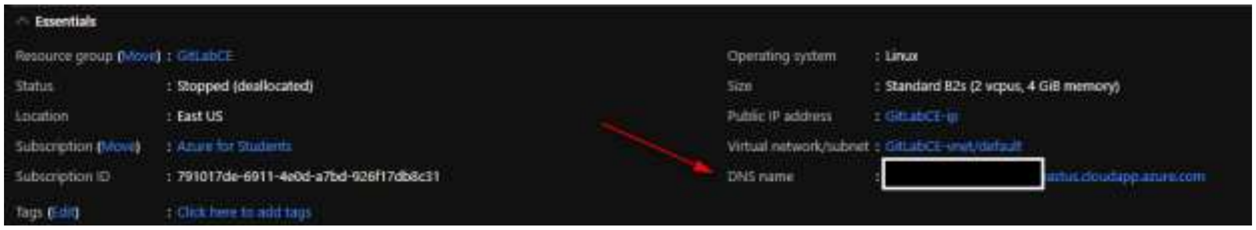
- Resource group name: GitLabCE
- Virtual machine name: GitLabCE
- Image: Ubuntu 18.04 LTS – Gen1

- Size: Standard_B2s
- Authentication Type: SSH public key
- Inbound Ports: Allow HTTP (80), HTTPS (443), SSH (22)

DNS setup

When the VM is created, view the newly created VM's resources. It should already be assigned a Public IP address. However, this address may change when the VM is stopped and restarted.

Select the Not Configured link next to the DNS name label under the VM's details and change it as desired.



Resource group (Move)	: GitLabCE	Operating system	: Linux
Status	: Stopped (deallocated)	Size	: Standard B2s (2 vcpus, 4 GiB memory)
Location	: East US	Public IP address	: GitLabCE-ip
Subscription (Move)	: Azure for Students	Virtual network/subnet	: GitLabCE-vnet/default
Subscription ID	: 791017de-6911-4e0d-a7bd-926f17db8c31	DNS name	: [redacted]astus.cloudapp.azure.com
Tags (Edit)	: Click here to add tags		

GitLab CE Installation

Using your ssh key, ssh into the VM.
Follow the instructions at the following URL to install GitLab CE with Ubuntu 18.04 LTS VM.
https://about.gitlab.com/install/#ubuntu

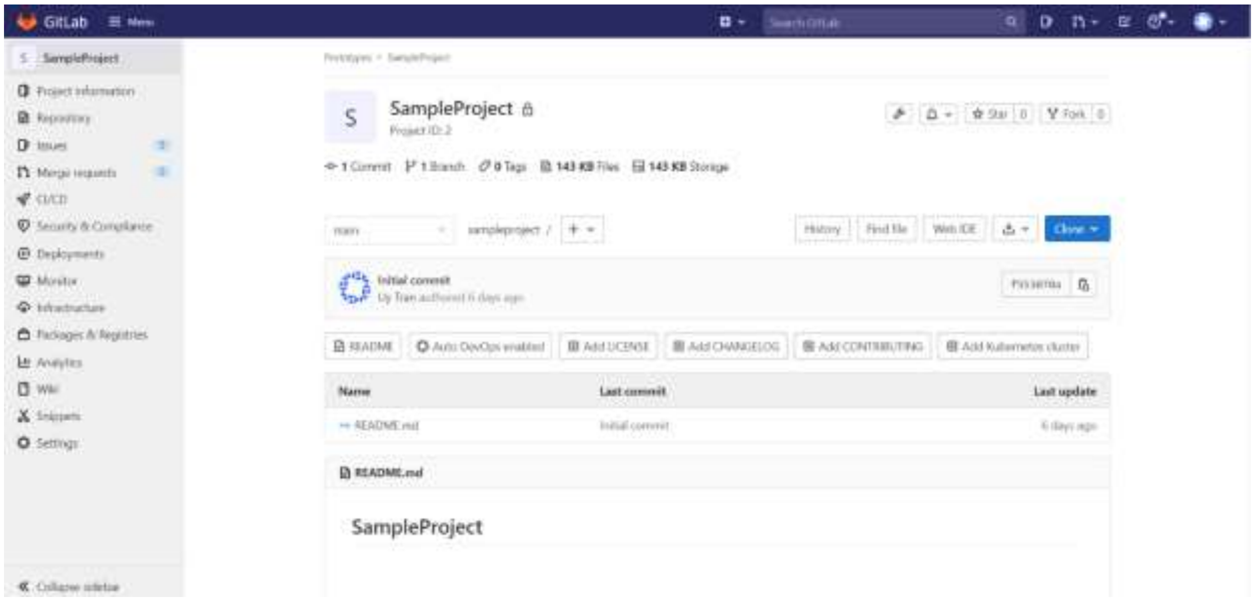
** Notes during the installation:

- Make sure you replace **"ee"** with **"ce"** to install the free Community Edition.
- Skip the Postfix installation, the e-mail notifications would not be used.
- Use the DNS name that you specified above for your GitLab URL (i.e., https://gitlab-utran2.westus2.cloudapp.azure.com). Make sure to include the leading **https://**
- Make sure ports 80 and 443 are opened for inbound access.
- Skip Steps 4 and 5 in the GitLab installation instructions.

Once the installation is done, go to the specified file (/etc/gitlab/initial_root_password) on the VM to find the root password. Make sure the root password is stored safely.

Then, go to the link https://<VM_DNS> on a browser and login to the GitLab web application with root as the username and root password.

In GitLab, you may create users or groups for others' access and projects as desired.



Knowledge Base

Environment setup

- Resource group name: Confluence
- Virtual machine name: Confluence
- Image: Ubuntu 18.04 LTS
- Authentication Type: SSH public key
- Inbound Ports: Allow SSH (22) only

Install Confluence

Follow the steps here to install Confluence with a Linux installer:

https://confluence.atlassian.com/doc/confluence-installation-guide-135681.html

- Download the Confluence installation to the workstation and then use *sftp* or *scp* to transfer the file to the VM. Or use *wget* to download directly to the VM.

- Setup a DNS Name for Confluence VM in the Azure Portal. (i.e., https://confluence-utran2.westus2.cloudapp.azure.com)
- Based on the prerequisites, the following ports need to be open on VM: 8090 and 8091.
- Setup Confluence for a Production Installation (not a Trial Installation) and My Own Database (not Built-in).
- For the license when setting up Confluence, we can obtain a free 30-day trial for Confluence (Server).
- **Make sure you store your admin password for Confluence in your password safe.**

Install MySQL

Login to your Virtual Machine using ssh.
Follow the instructions here to install MySQL:
https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-18-04

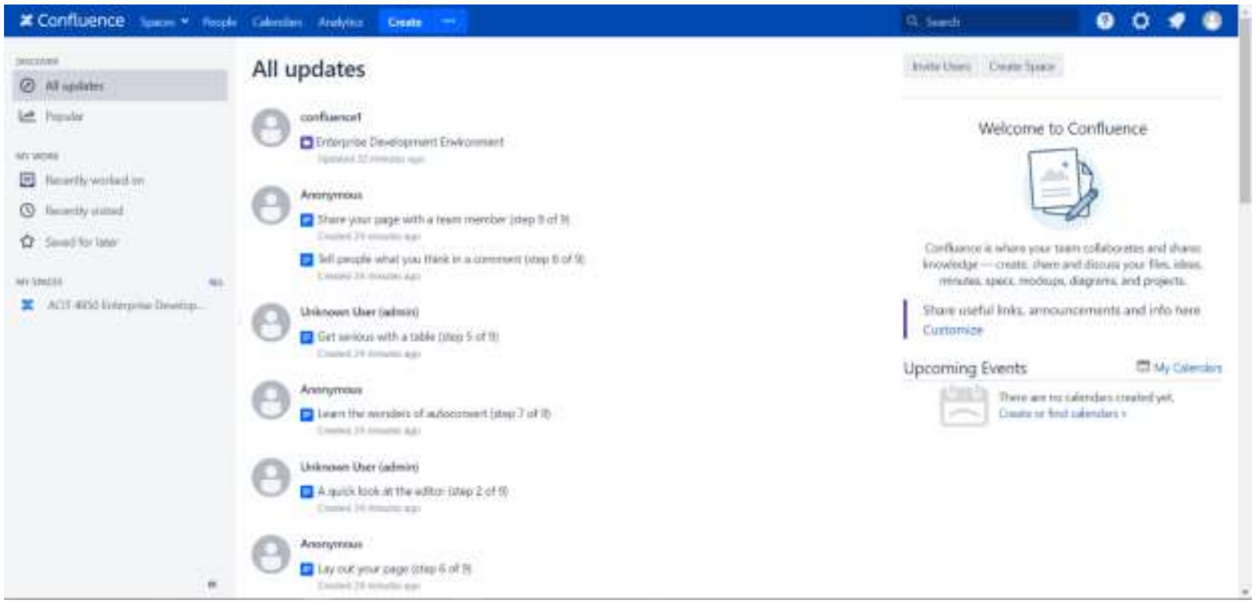
Configure your MySQL installation for Confluence based on the instructions here:
https://confluence.atlassian.com/doc/database-setup-for-mysql-128747.html

- Skip Item 1, it has been done.
- For Item 2, follow the Linux specific instructions
- Item 4 is the instructions above for the Confluence installation, we can skip it.
- Make sure to test the connection from Confluence to MySQL successfully.

Configure Knowledge Base

As Admin,

- create a test user,
- create a new test Space with the admin and test user as Space Admin,
- verify that we can create a page in the new Space.



All updates

confluence1
Enterprise Development Environment
Updated 22 minutes ago

Anonymous
Share your page with a team member (step 9 of 9)
Created 24 minutes ago
Tell people what you think in a comment (step 8 of 9)
Created 24 minutes ago

Unknown User (admin)
Get serious with a table (step 5 of 9)
Created 24 minutes ago

Anonymous
Learn the wonders of autoconvert (step 7 of 9)
Created 24 minutes ago

Unknown User (admin)
A quick look at the editor (step 2 of 9)
Created 24 minutes ago

Anonymous
Lay out your page (step 6 of 9)
Created 24 minutes ago

Welcome to Confluence

Confluence is where your team collaborates and shares knowledge — create, share and discuss your files, ideas, minutes, specs, mockups, diagrams, and projects.

Share useful links, announcements and info here
Customize

Upcoming Events
My Calendars
There are no calendars created yet.
Create or find calendars ›

Web Server

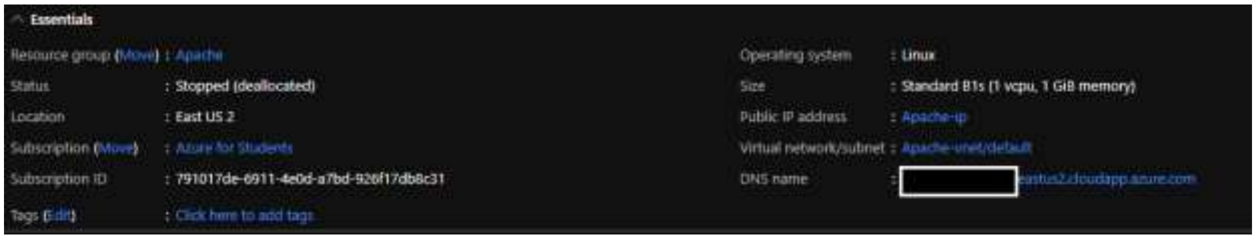
Environment setup

- Resource group name: Apache
- Virtual machine name: Apache
- Image: Ubuntu 18.04 LTS – Gen1
- Authentication Type: SSH public key
- Inbound Ports: Allow SSH (22)

DNS setup

When the VM is created, view the newly created VM’s resources. It should already be assigned a Public IP address. However, this address may change when the VM is stopped and restarted.

Select the Not Configured link next to the DNS name label under the VM’s details and change it as desired.



Essentials			
Resource group (Move)	: Apache	Operating system	: Linux
Status	: Stopped (deallocated)	Size	: Standard B1s (1 vcpu, 1 GiB memory)
Location	: East US 2	Public IP address	: Apache-ip
Subscription (Move)	: Azure for Students	Virtual network/subnet	: Apache-vnet/default
Subscription ID	: 791017de-6911-4e0d-a7bd-926f17db8c31	DNS name	: [redacted]eastus2.cloudapp.azure.com
Tags (Edit)	: Click here to add tags		

Install Apache2

Login to Virtual Machine using ssh.

Run the following commands to install Apache2:

- `sudo apt update`
- `sudo apt install apache2`

Verify your Apache2 installation:

- `apache2 -version`

Apache2 version 2.4.x should be installed.

Open up access to Apache on the Ubuntu firewall:

- `sudo ufw app list`

Open up the Apache profile, which allows access on port 80

- `sudo ufw allow 'Apache'`

Check the status of Apache2 with the following command:

- `sudo systemctl status apache2`

The Apache HTTP Server should be started.

On web browser, check that we can access the Apache server by going to the following URL:

- http://<**Apache**_Virtual_Server_DNS_Name >
- A page titled "Apache2 Ubuntu Default Page" is shown

Configure Reverse Proxy

Apache Reverse Proxy so that JIRA and Confluence are available on the following URLs:

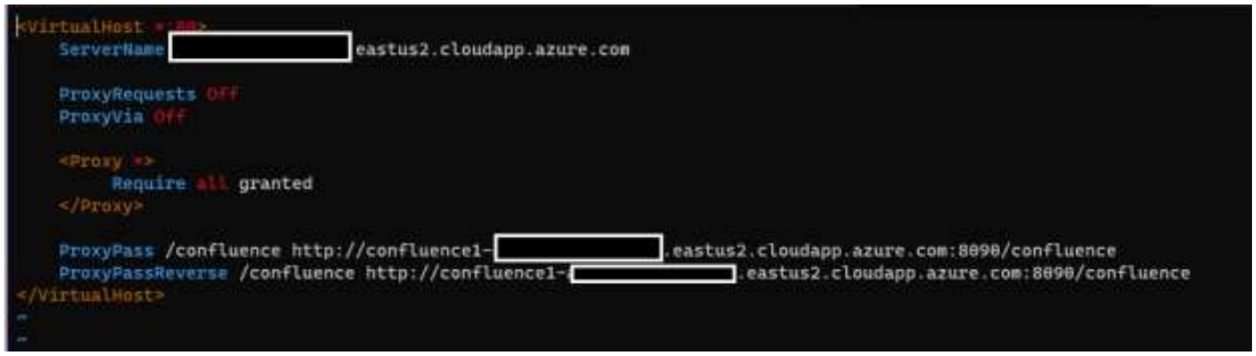
- http://<**Apache**_Virtual_Server_DNS_Name>/**jira**
- http://< **Apache**_Virtual_Server_DNS_Name>/**confluence**

Follow the steps here to setup Apache2 as a reverse proxy with Atassian products.

https://confluence.atlassian.com/kb/proxying-atlassian-server-applications-with-apache-http-server-mod_proxy_http-806032611.html

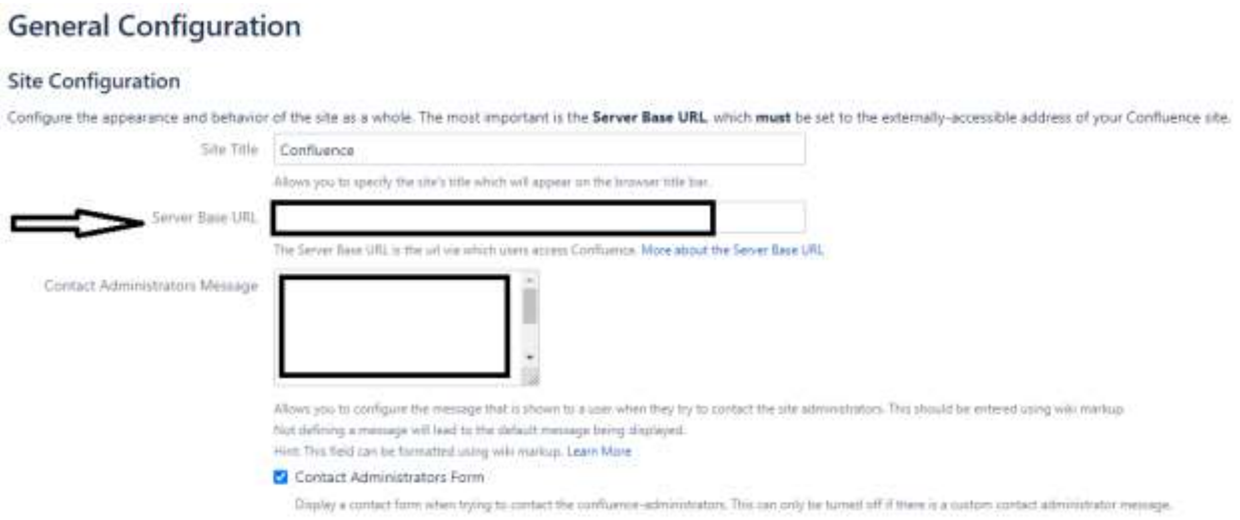
- Before editing any configuration file, **BACK UP** the file first

- Edit the existing 000-default.conf (and later the default-ssl.conf) on Apache



```
<VirtualHost *:80>
    ServerName [redacted]eastus2.cloudapp.azure.com

    ProxyRequests Off
    ProxyVia Off

    <Proxy *>
        Require all granted
    </Proxy>

    ProxyPass /confluence http://confluence1-[redacted].eastus2.cloudapp.azure.com:8090/confluence
    ProxyPassReverse /confluence http://confluence1-[redacted].eastus2.cloudapp.azure.com:8090/confluence
</VirtualHost>
~
~
```

- We also have to change your root URL for Confluence/JIRA in the admin settings of the applications to be that of the reverse proxy, otherwise users will get warnings that the URL being used doesn't match the one configured in the application.



General Configuration

Site Configuration

Configure the appearance and behavior of the site as a whole. The most important is the **Server Base URL**, which **must** be set to the externally-accessible address of your Confluence site.

Site Title: Confluence

Allows you to specify the site's title which will appear on the browser title bar.

Server Base URL: [redacted]

The Server Base URL is the url via which users access Confluence. More about the Server Base URL

Contact Administrators Message: [redacted]

Allows you to configure the message that is shown to a user when they try to contact the site administrators. This should be entered using wiki markup.
Not defining a message will lead to the default message being displayed.
Hint: This field can be formatted using wiki markup. Learn More

[✓] Contact Administrators Form

Display a contact form when trying to contact the confluence-administrators. This can only be turned off if there is a custom contact administrator message.

- We also need to add the Confluence */synchrony* endpoint in the VirtualHost of the Apache server

Network Security Groups

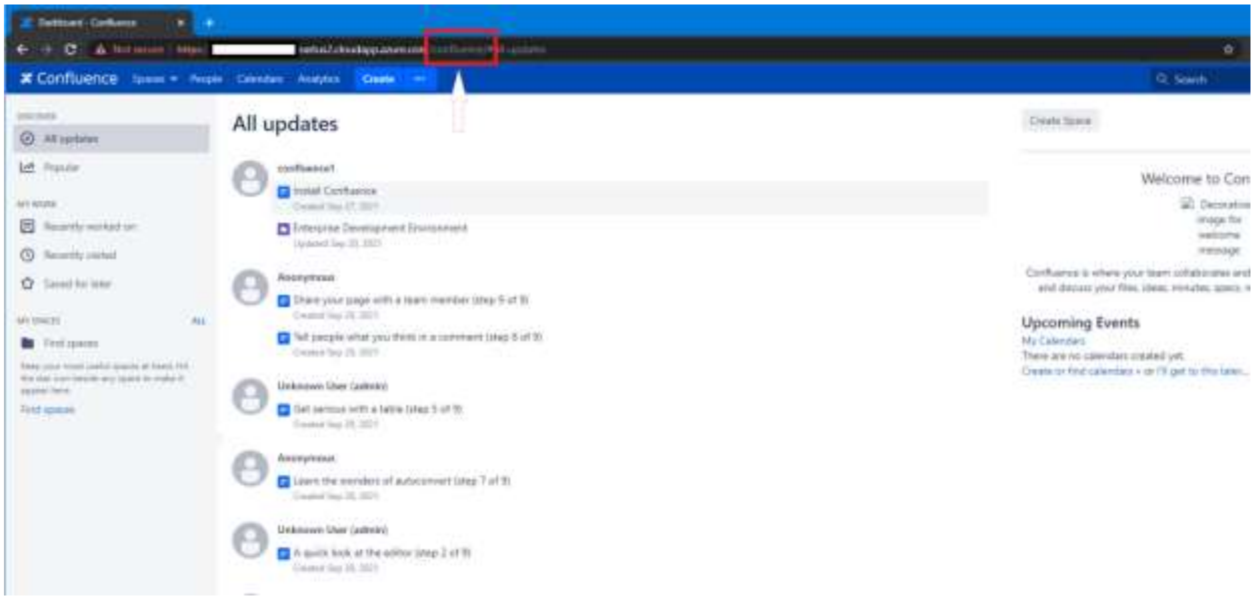
Update Network Security Groups for Virtual Machines running Confluence and JIRA.

Confluence Virtual Machine

- Change the inbound rules such that access to port 8090 and 8091 is only allowed from the Apache Virtual Machine. Use the public IP of the Apache VM and make sure that IP is reserved when shutdown that VM.

JIRA Virtual Machine

- Change the inbound rules such that access to port 8080 is only allowed from the Apache Virtual Machine. Use the public IP of the Apache VM and make sure that IP is reserved when shutdown that VM.



Orchestration Tool

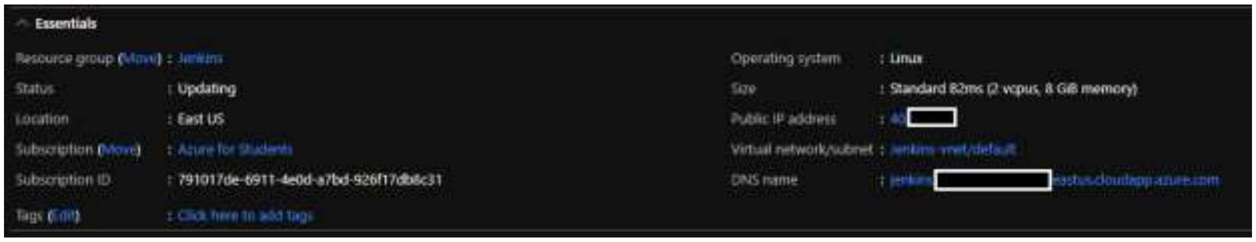
Environment setup

- Resource group name: Jenkins
- Virtual machine name: Jenkins
- Image: Ubuntu 18.04 LTS
- Authentication Type: SSH public
- Inbound Ports: Allow SSH (22), TCP (8080)

DNS setup

When the VM is created, view the newly created VM's resources. It should already be assigned a Public IP address. However, this address may change when the VM is stopped and restarted.

Select the Not Configured link next to the DNS name label under the VM's details and change it as desired.



Essentials

Resource group (Move)	: Jenkins	Operating system	: Linux
Status	: Updating	Size	: Standard B2ms (2 vcpus, 8 GiB memory)
Location	: East US	Public IP address	: 40[redacted]
Subscription (Move)	: Azure for Students	Virtual network/subnet	: jenkins-vnet/default
Subscription ID	: 791017de-6911-4e0d-a7bd-926f17db6c31	DNS name	: jenkins[redacted]eastus.cloudapp.azure.com
Tags (Edit)	: Click here to add tags		

Install Docker and Jenkins

Login to Jenkins VM using ssh.

Follow the instructions here for installing Docker and Jenkins on Ubuntu 18.04:

https://linuxhint.com/install_jenkins_docker_ubuntu/

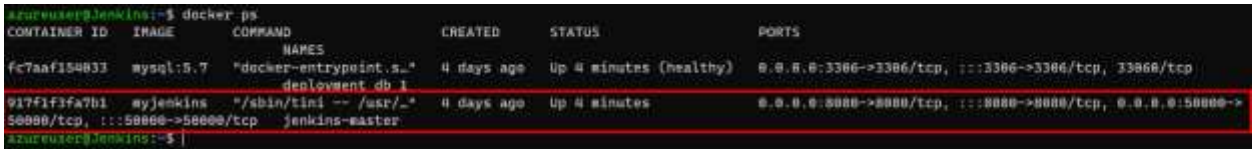
We will need to use *sudo* for most of the commands as they require root permissions. We can change this so we can run docker commands as a non-root user with these post-install steps:

https://docs.docker.com/engine/install/linux-postinstall

Command to start (including restart if the VM is restarted)

```
docker run -p 8080:8080 -p 50000:50000 --restart always --name=jenkins-master --mount
source=jenkins-log,target=/var/log/jenkins --mount source=jenkins-
data,target=/var/jenkins_home -d myjenkins
```

Verify the Jenkin container is running

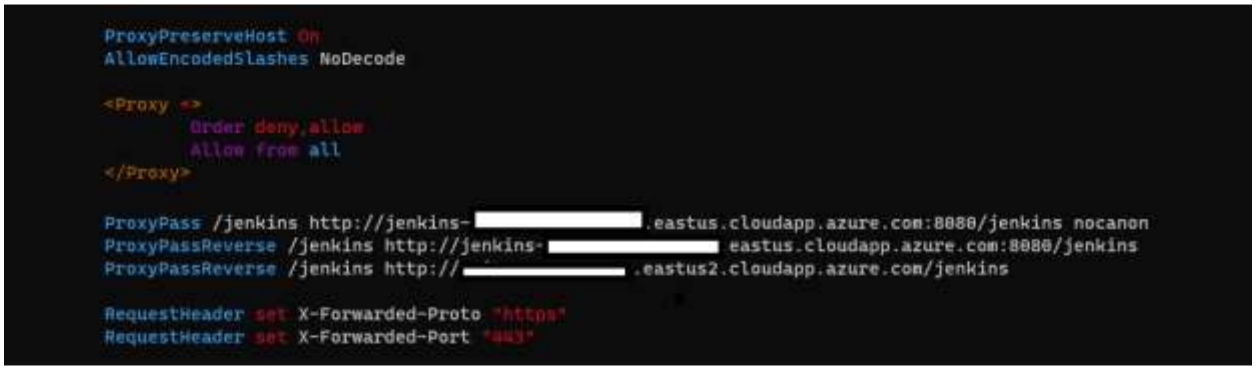


```
azureuser@Jenkins:~$ docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED      STATUS                  PORTS
                 NAMES
fc7aaf154033   mysql:5.7   "docker-entrypoint.s…"   4 days ago   Up 4 minutes (healthy)   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp
                 deployment_db_1
917f1f3fa7b1   myjenkins   "/sbin/tini -- /usr/…"   4 days ago   Up 4 minutes             0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:50000->
50000/tcp, :::50000->50000/tcp   jenkins-master
azureuser@Jenkins:~$ 
```

Reverse Proxy for Jenkins

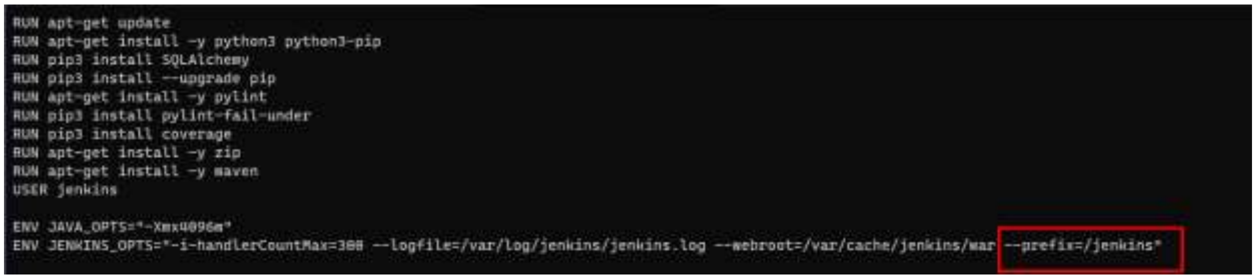
SSH into Apache VM (containing Apache2 configured as a reverse proxy).

Add the following to the default-ssl.conf file for Jenkins:



```
        ProxyPreserveHost On
        AllowEncodedSlashes NoDecode

        <Proxy *>
                Order deny,allow
                Allow from all
        </Proxy>

        ProxyPass /jenkins http://jenkins-[redacted].eastus.cloudapp.azure.com:8080/jenkins nocanon
        ProxyPassReverse /jenkins http://jenkins-[redacted] eastus.cloudapp.azure.com:8080/jenkins
        ProxyPassReverse /jenkins http://[redacted].eastus2.cloudapp.azure.com/jenkins

        RequestHeader set X-Forwarded-Proto "https"
        RequestHeader set X-Forwarded-Port "443"
```

Moreover, Jenkins installation must include the /jenkins context path. SSH into Jenkins VM and update the following in the Dockerfile:

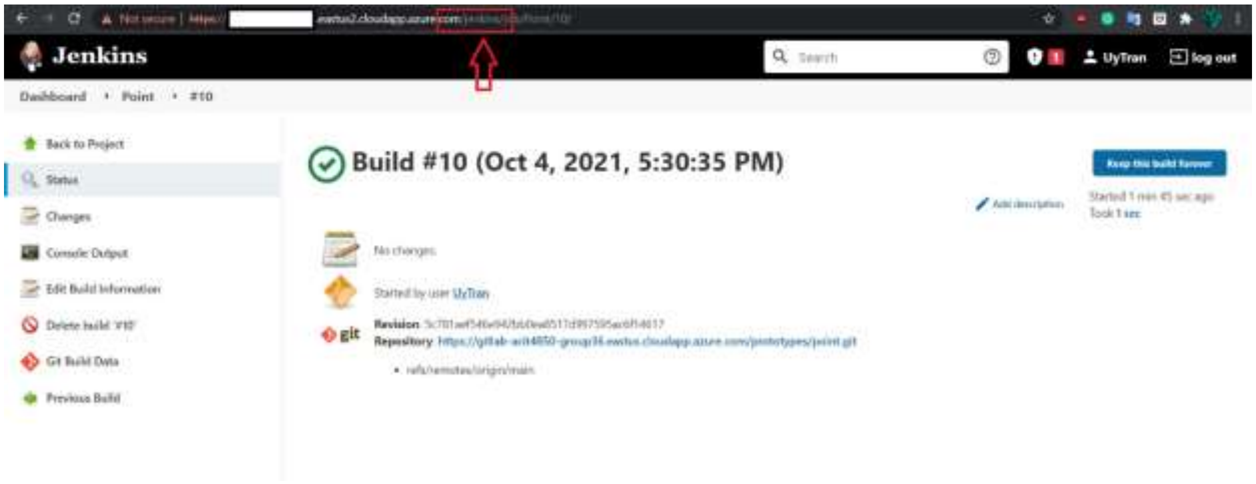


```
RUN apt-get update
RUN apt-get install -y python3 python3-pip
RUN pip3 install SQLAlchemy
RUN pip3 install --upgrade pip
RUN apt-get install -y pylint
RUN pip3 install pylint-fail-under
RUN pip3 install coverage
RUN apt-get install -y zip
RUN apt-get install -y maven
USER jenkins

ENV JAVA_OPTS="-Xmx4096m"
ENV JENKINS_OPTS="-i-handlerCountMax=300 --logfile=/var/log/jenkins/jenkins.log --webroot=/var/cache/jenkins/war --prefix=/jenkins"
```

Run the docker stop, build and start command above to re-build and re-start the Jenkins image.

Make sure only the 8080 port is open on Jenkins VM and it is tied to **the IP address** of the Apache VM

Go to browser, navigate to *https://<Apache DNS Name>/jenkins* and complete Jenkins installation



Build #10 (Oct 4, 2021, 5:30:35 PM)

Started 1 min 45 sec ago
Took 1 sec

No changes.

Started by user UyTran

Revision: [illegible]
Repository: https://gitlab-acit4850-group36.eastus.cloudapp.azure.com/prototypes/point.git

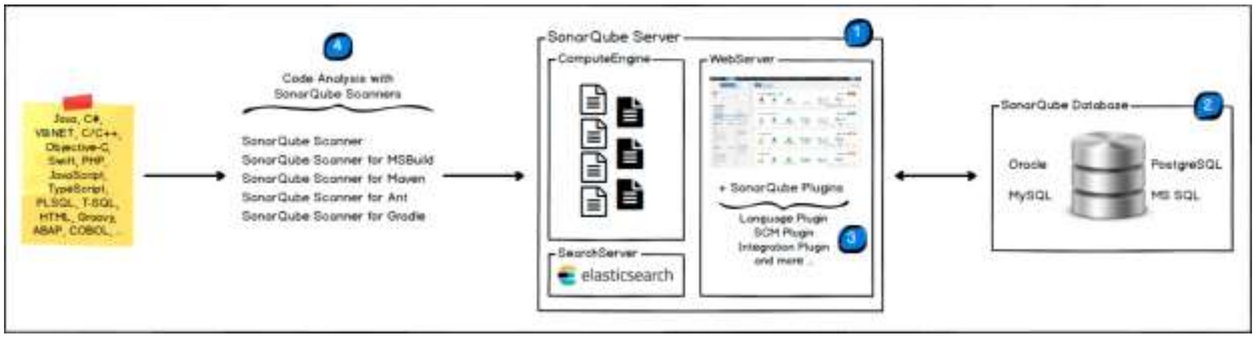
- refs/remotes/origin/main

SonarQube

Introduction

SonarQube is known as a “glorified” lint tool with the following features:

- Scanners to perform the static analysis run on CI tool
- Server to receive and analyze the results (i.e., the snapshots) and present them to the user
- Plugin for integration or language specific processing
- Database to store the results/analysis for comparison (i.e., the snapshots), custom rules and to record flagged “false positives”
- IDE Plugins to catch issues even before the CI pipeline


Java, C#, VB.NET, C/C++, Objective-C, Swift, PHP, JavaScript, TypeScript, PL/SQL, T-SQL, HTML, Groovy, ABAP, COBOL, ...
4
Code Analysis with SonarQube Scanners
SonarQube Scanner
SonarQube Scanner for MSBuild
SonarQube Scanner for Maven
SonarQube Scanner for Ant
SonarQube Scanner for Gradle
SonarQube Server
1
ComputeEngine
WebServer
+ SonarQube Plugins
Language Plugin
SCM Plugin
Integration Plugin
and more ...
3
SearchServer
elasticsearch
SonarQube Database
2
Oracle
MySQL
PostgreSQL
MS SQL

Reference: https://docs.sonarqube.org/7.1/ArchitectureandIntegration.html

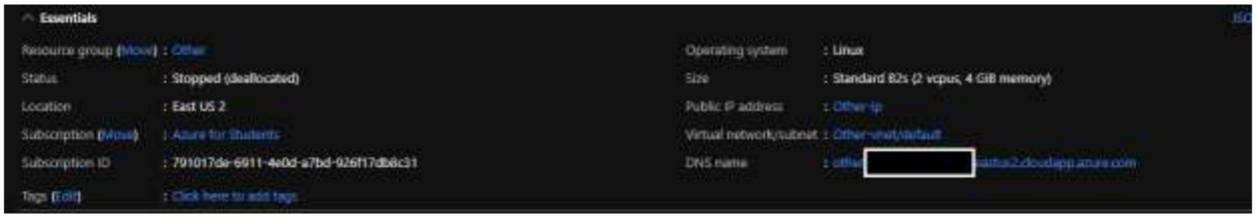
Environment setup

- Resource group name: Other
- Virtual machine name: Other
- Image: Ubuntu 18.04 LTS
- Authentication Type: SSH public key
- Inbound Ports: Allow SSH (22), TCP (9000)

DNS setup

When the VM is created, view the newly created VM’s resources. It should already be assigned a Public IP address. However, this address may change when the VM is stopped and restarted.

Select the Not Configured link next to the DNS name label under the VM’s details and change it as desired.



Essentials

Resource group (Move)	: Other	Operating system	: Linux
Status	: Stopped (deallocated)	Size	: Standard B2s (2 vcpus, 4 GiB memory)
Location	: East US 2	Public IP address	: Other-ip
Subscription (Move)	: Azure for Students	Virtual network/subnet	: Other-vnet/default
Subscription ID	: 791017de-6911-4e0d-a7bd-926f17db8c31	DNS name	: other[redacted]eastus2.cloudapp.azure.com
Tags (Edit)	: Click here to add tags		

SonarQube installation

The basic SonarQube installation is documented in the links below. We will use PostgreSQL as the database. Here is a tutorial on installing SonarQube with PostgreSQL on Ubuntu:

https://www.fosstechnix.com/install-sonarqube-on-ubuntu

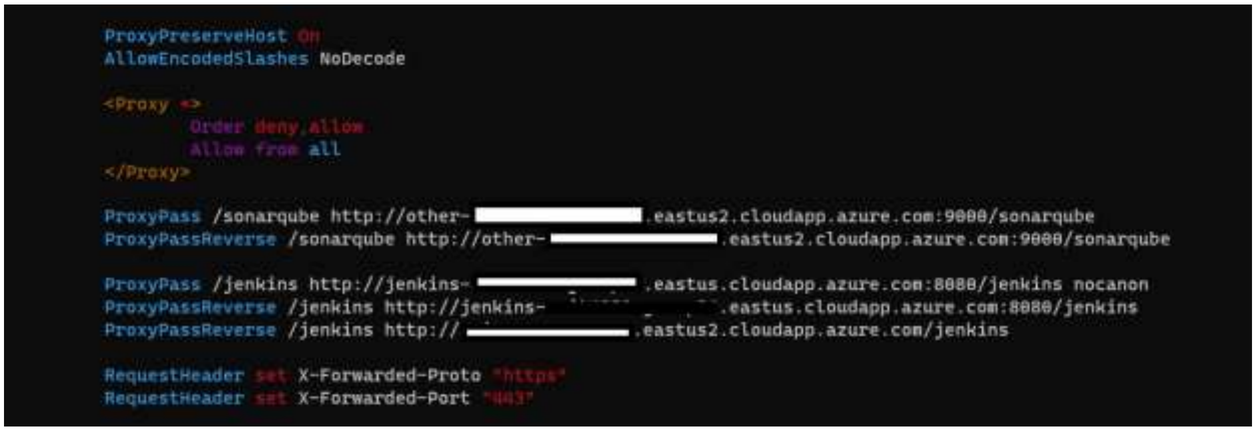
- Setup with systemd so that it will restart automatically when restart the VM.
- In sonar.properties, uncomment out the sonar.jdbc.url for postgresql. It should look like:
  `sonar.jdbc.url=jdbc:postgresql://localhost:5432/sonarqube`
- Also set the context path:
  `sonar.web.context=/sonarqube`
  `sonar.web.port=9000`
- We should be able to access running SonarQube at http://<Other DNS>:9000/sonarqube

Reverse Proxy

Documentation:

https://docs.sonarqube.org/latest/setup/operate-server/

(Refer to section Using an Apache Proxy).

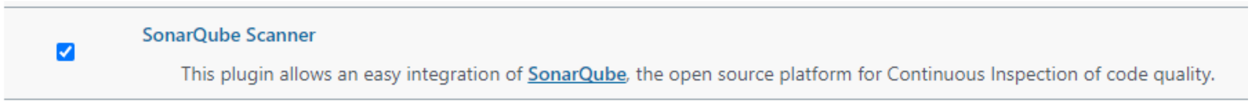


```
ProxyPreserveHost On
AllowEncodedSlashes NoDecode

<Proxy *>
        Order deny,allow
        Allow from all
</Proxy>

ProxyPass /sonarqube http://other-[redacted].eastus2.cloudapp.azure.com:9000/sonarqube
ProxyPassReverse /sonarqube http://other-[redacted].eastus2.cloudapp.azure.com:9000/sonarqube

ProxyPass /jenkins http://jenkins-[redacted].eastus.cloudapp.azure.com:8080/jenkins nocanon
ProxyPassReverse /jenkins http://jenkins-[redacted].eastus.cloudapp.azure.com:8080/jenkins
ProxyPassReverse /jenkins http://[redacted].eastus2.cloudapp.azure.com/jenkins

RequestHeader set X-Forwarded-Proto "https"
RequestHeader set X-Forwarded-Port "443"
```

SonarQube Jenkins Plugin

Add the Plugin:

- Login to Jenkins server
- Go to Manage Jenkins
- Go to Manage Plugins
- Select the Available tab
- Find the "SonarQube Scanner" plugin
- Select and install this plugin

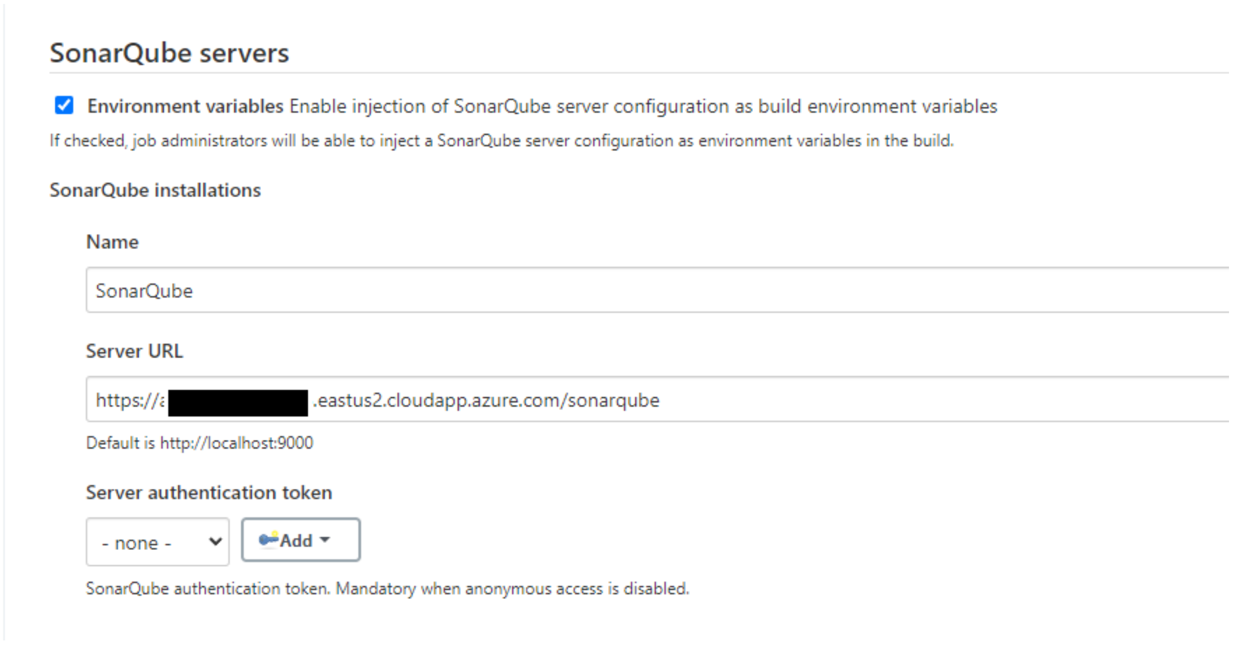


☑ SonarQube Scanner

This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.

Configure the Plugin:

- Go to Manage Jenkins
- Select Configure System
- Find the SonarQube servers section of the configuration
- Check the Environment variables checkbox
- Add A SonarQueb server
- Enter a SonarQube installation name. Use 'SonarQube'
- Enter the URL of SonarQube server
- Save the configuration



SonarQube servers

☑ **Environment variables** Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name

SonarQube

Server URL

https://[redacted].eastus2.cloudapp.azure.com/sonarqube

Default is http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

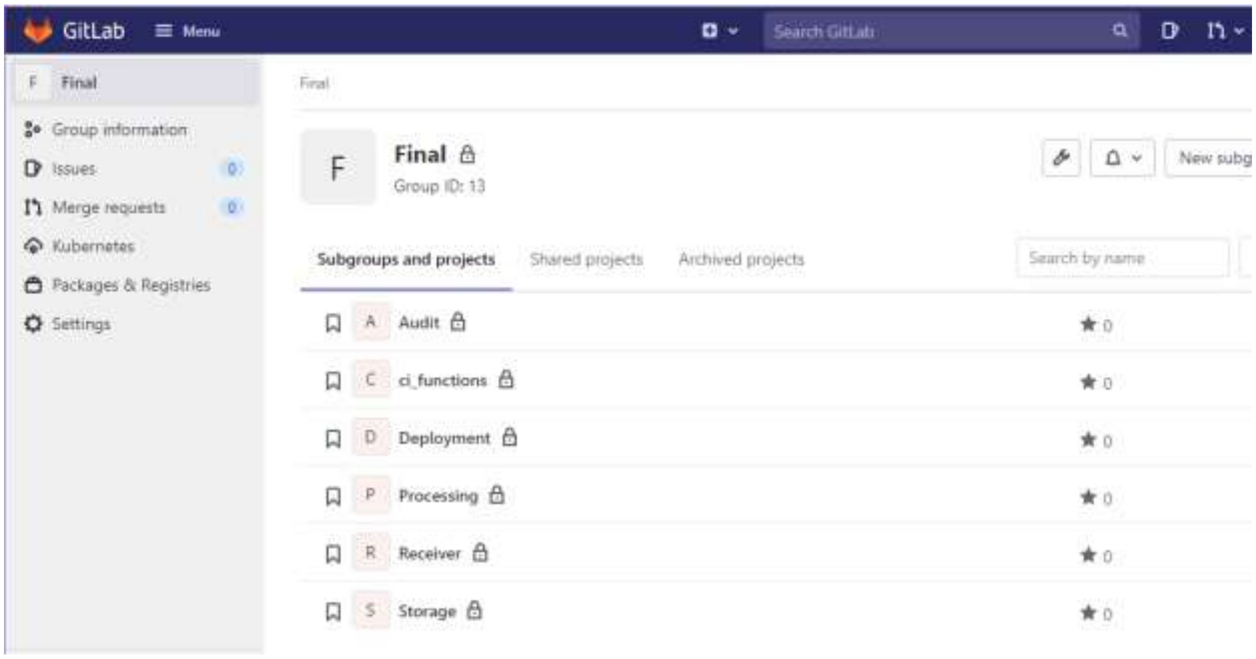
Note: For any Java project, add Maven to Jenkins Docker Image, SSH to Jenkins server

- Add the following to your Jenkins dockerfile: `RUN apt-get install -y maven`
- Stop and remove running Jenkins container
- Re-build the Jenkins image and Run newly built Jenkins image

Create, Build, and Deploy an application

GitLabCE

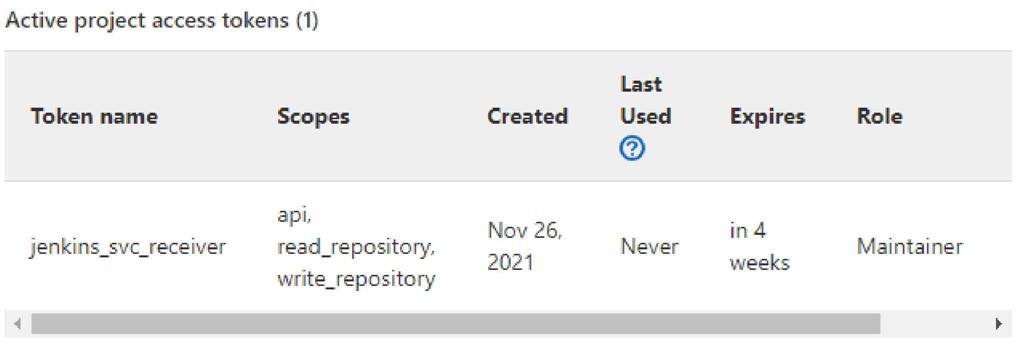
a) Create projects



b) Create access token

In order to let Jenkins access and run the pipeline job, we need an access token for each service.

- Going to each project, navigate to Setting on the left panel and click "Access Token"
- Define a name and token access for the project
- Give certain permission associated with that token



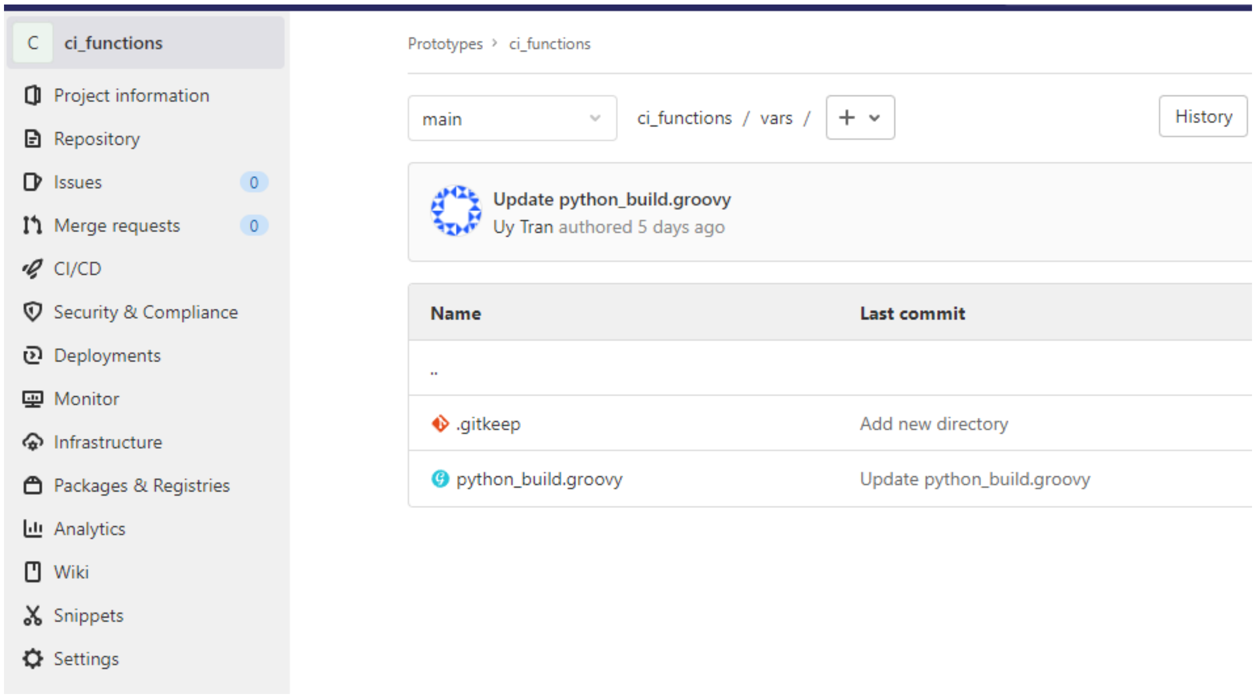
Active project access tokens (1)

Token name	Scopes	Created	Last Used	Expires	Role
jenkins_svc_receiver	api, read_repository, write_repository	Nov 26, 2021	Never	in 4 weeks	Maintainer

- Doing the same steps to create more 3 access tokens for other services.

c) <u>Define Jenkins file and share library</u>

Creating a Shared Library that defines an entire pipeline. Many repositories frequently have the same type and structure of code, so the same pipeline applies. This is especially true for microservices type projects where there are many small applications that need to be built, often in the same language.



Update python_build.groovy
Uy Tran authored 5 days ago

Name	Last commit
..	
.gitkeep	Add new directory
python_build.groovy	Update python_build.groovy

In this python_build.groovy to have pipeline structure for each stage. There are 4 stages:

- Stage “Build”: install the python libraries for each service
- Stage “Python Lint”: a static code analysis tool used to scan errors, bugs, stylistic, syntax errors in code
- Stage “Package”: build the service to image and upload to Docker hub
- Stage “Zip Artifact”: zip all python files and store in Jenkins for each build

- Stage "Build & Deliver WebApp": build and run the whole application. It is only when choose Build with Params in Jenkins, not automatically run when the pipeline is triggered.

Optional (not cover in this demo)

- Stage "Coverage": to run the unit tests and generate reports.

d) <u>Create webhook for automatically trigger a pipeline</u>

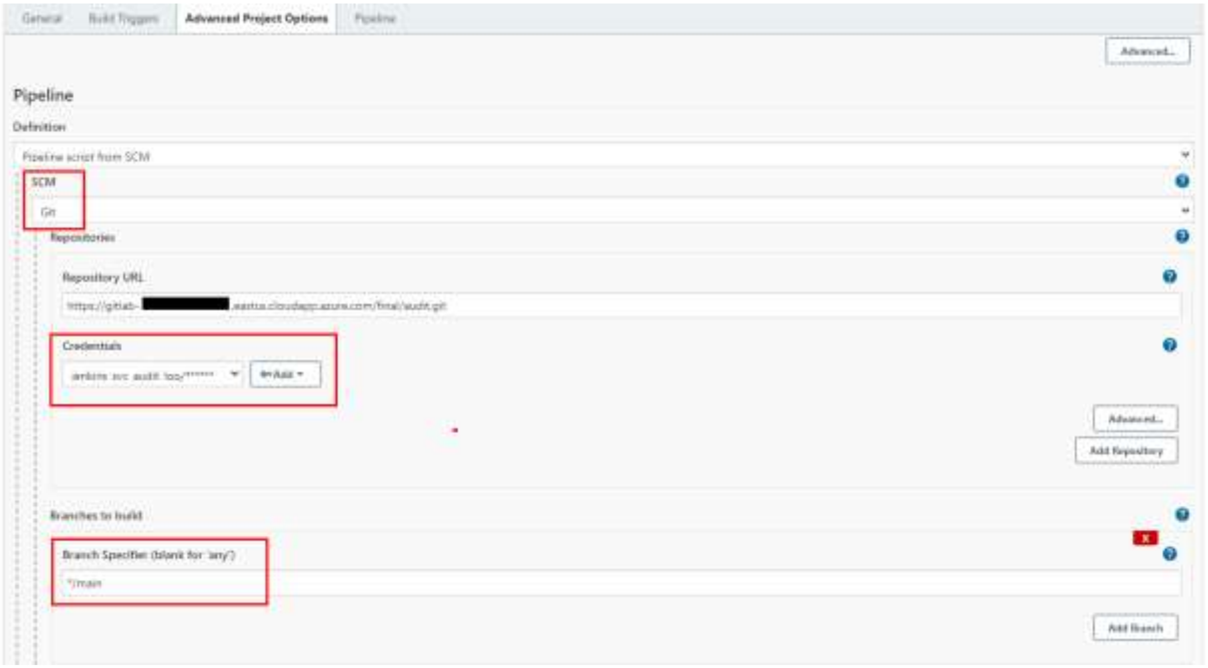
Jenkins

a) <u>Create a pipeline for a gitlab repo</u>

On Jenkins, create 4 pipelines associated with each service. The configuration below is applied the same for all services.

- Advanced Project Option

- Choose Git for the pipeline
- GitLab repo
- Credentials is the access token that we created above
- Brand that will be pull for the pipeline



Pipeline

Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL: https://gitlab-[redacted].eastus.cloudapp.azure.com/final/audit.git

Credentials: jenkins svc audit too/****** Add

Branches to build

Branch Specifier (blank for 'any'): */main

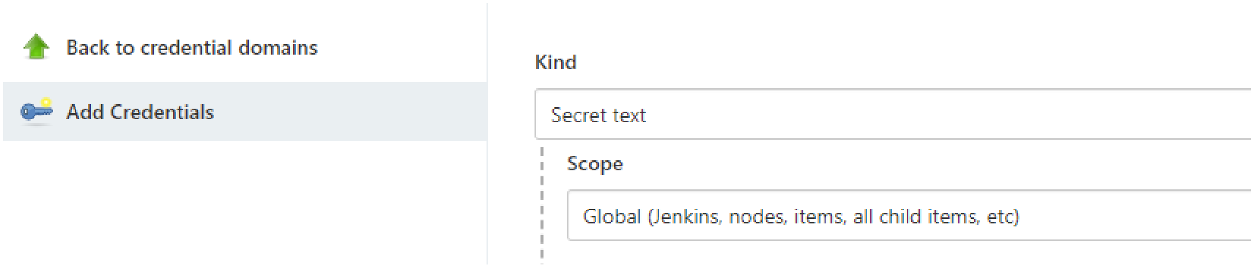
b) <u>Create Docker hub access token</u>

When pulling the image to Docker Hub, Jenkins needs an access token from Docker Hub.

Once the access token is created, go to Jenkins Dashboard and go to:

- Navigate to Manage Credentials

- Add a new credentials with the kind as "Secret Text"

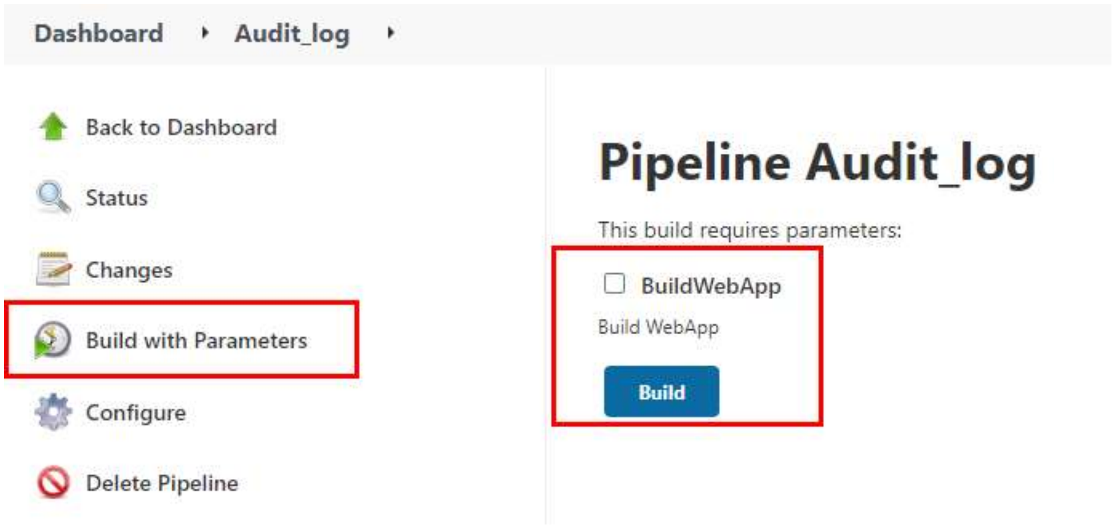


Kind: Secret text

Scope: Global (Jenkins, nodes, items, all child items, etc)

c) <u>Build with parmas</u>

When all the services pass all stages and ready to be deployed, on Jenkins, click to build the pipeline with params (at any services)



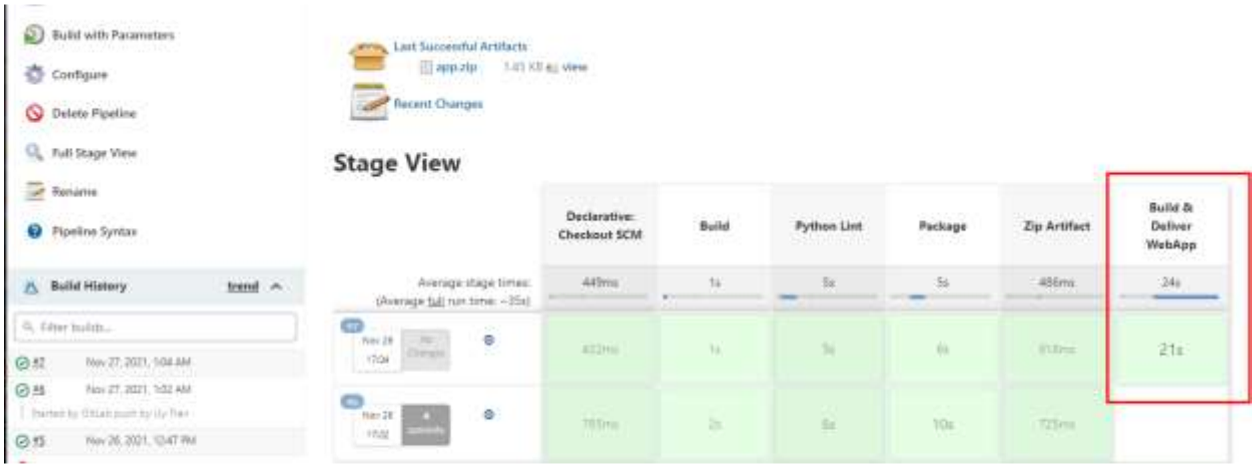
Pipeline Audit_log

This build requires parameters:

BuildWebApp

Build WebApp

Verify all stages passed

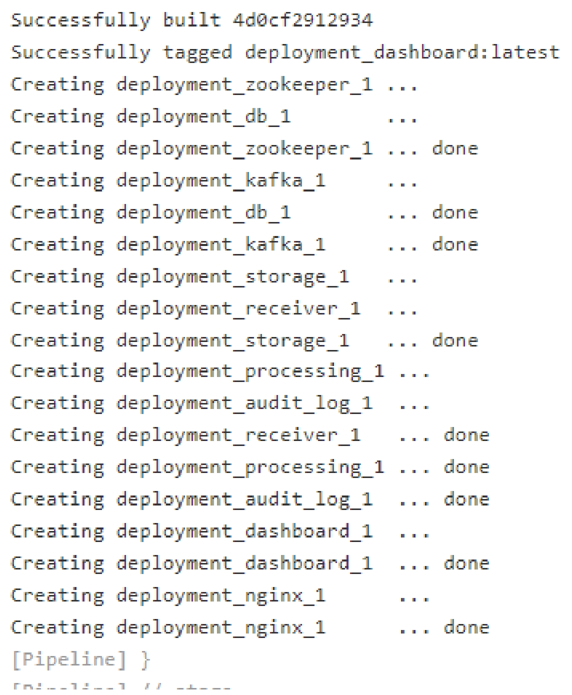


Stage View

Average stage times:
(Average full run time: ~35s)

Declarative: Checkout SCM	Build	Python Lint	Package	Zip Artifact	Build & Deliver WebApp
449ms	1s	5s	5s	486ms	24s
432ms	1s	5s	6s	613ms	21s
781ms	2s	6s	10s	725ms	

d) <u>Verify output logs</u>

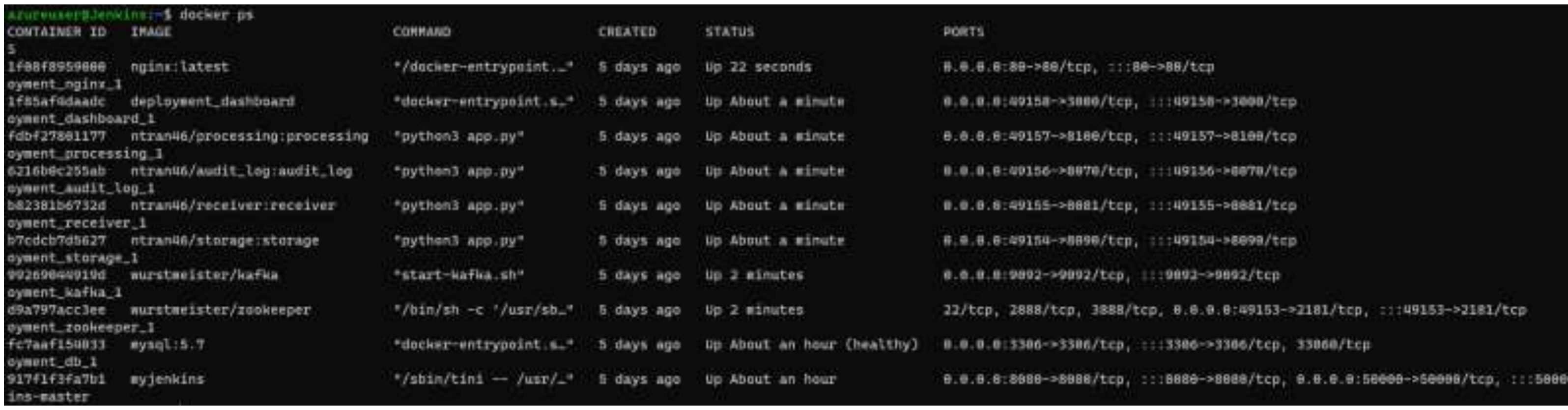


```
Successfully built 4d0cf2912934
Successfully tagged deployment_dashboard:latest
Creating deployment_zookeeper_1 ...
Creating deployment_db_1        ...
Creating deployment_zookeeper_1 ... done
Creating deployment_kafka_1     ...
Creating deployment_db_1        ... done
Creating deployment_kafka_1     ... done
Creating deployment_storage_1   ...
Creating deployment_receiver_1  ...
Creating deployment_storage_1   ... done
Creating deployment_processing_1 ...
Creating deployment_audit_log_1  ...
Creating deployment_receiver_1   ... done
Creating deployment_processing_1 ... done
Creating deployment_audit_log_1  ... done
Creating deployment_dashboard_1  ...
Creating deployment_dashboard_1  ... done
Creating deployment_nginx_1      ...
Creating deployment_nginx_1      ... done
[Pipeline] }
```

New deployment has been deployed successfully.

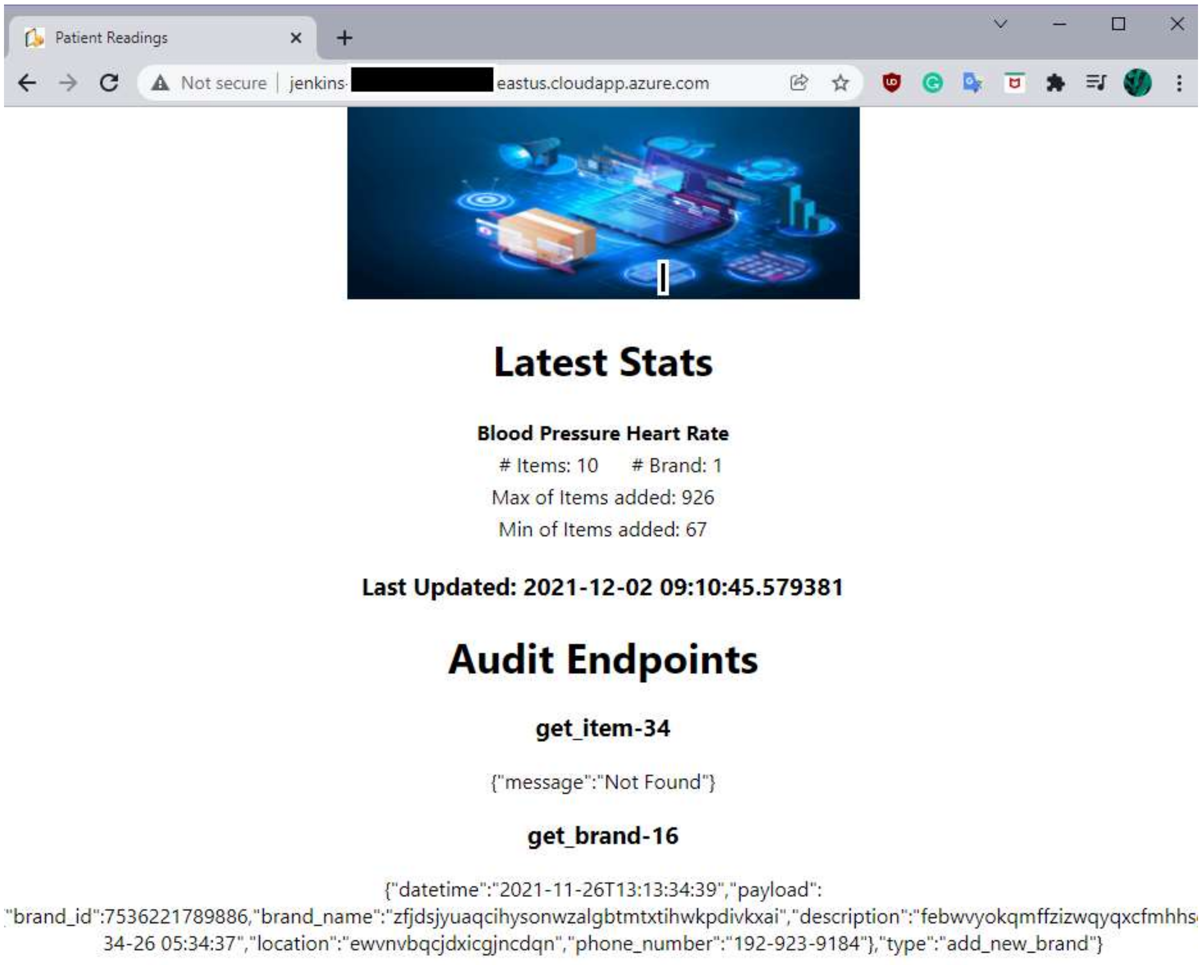
e) <u>Verify deployed application is running</u>

SSH to Jenkins severs, verify all services (containers) are up



```
azureuser@Jenkins:~$ docker ps
CONTAINER ID   IMAGE                           COMMAND                  CREATED       STATUS                    PORTS
s
1f08f8959800   nginx:latest                    "/docker-entrypoint.…"   5 days ago    Up 22 seconds             0.0.0.0:80->80/tcp, :::80->80/tcp
oyment_nginx_1
1f85af0daadc   deployment_dashboard            "docker-entrypoint.s…"   5 days ago    Up About a minute         0.0.0.0:49158->3000/tcp, :::49158->3000/tcp
oyment_dashboard_1
fdbf27801177   ntran46/processing:processing   "python3 app.py"         5 days ago    Up About a minute         0.0.0.0:49157->8100/tcp, :::49157->8100/tcp
oyment_processing_1
6216b0c255ab   ntran46/audit_log:audit_log     "python3 app.py"         5 days ago    Up About a minute         0.0.0.0:49156->8070/tcp, :::49156->8070/tcp
oyment_audit_log_1
b82381b6732d   ntran46/receiver:receiver       "python3 app.py"         5 days ago    Up About a minute         0.0.0.0:49155->8081/tcp, :::49155->8081/tcp
oyment_receiver_1
b7cdcb7d5627   ntran46/storage:storage         "python3 app.py"         5 days ago    Up About a minute         0.0.0.0:49154->8090/tcp, :::49154->8090/tcp
oyment_storage_1
99269044919d   wurstmeister/kafka              "start-kafka.sh"         5 days ago    Up 2 minutes              0.0.0.0:9092->9092/tcp, :::9092->9092/tcp
oyment_kafka_1
d9a797acc3ee   wurstmeister/zookeeper          "/bin/sh -c '/usr/sb…"   5 days ago    Up 2 minutes              22/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:49153->2181/tcp, :::49153->2181/tcp
oyment_zookeeper_1
fc7aaf154033   mysql:5.7                       "docker-entrypoint.s…"   5 days ago    Up About an hour (healthy)   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp
oyment_db_1
917f1f3fa7b1   myjenkins                       "/sbin/tini -- /usr/…"   5 days ago    Up About an hour          0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0:50000->50000/tcp, :::5000
ins-master
```

Access the webapp with Jenkin URL



Latest Stats

Blood Pressure Heart Rate

Items: 10 # Brand: 1

Max of Items added: 926

Min of Items added: 67

Last Updated: 2021-12-02 09:10:45.579381

Audit Endpoints

get_item-34

{"message":"Not Found"}

get_brand-16

{"datetime":"2021-11-26T13:13:34:39","payload":
"brand_id":7536221789886,"brand_name":"zfjdsjyuaqcihysonwzalgbtmtxtihwkpdivkxai","description":"febwvyokqmffzizwqyqxcfmhhs
34-26 05:34:37","location":"ewvnvbqcjdxicgjncdqn","phone_number":"192-923-9184"},"type":"add_new_brand"}