Integrated Webserver Build and Configuration Automation using Ansible
Final Presentation

IWBA

By

Abey Mathew

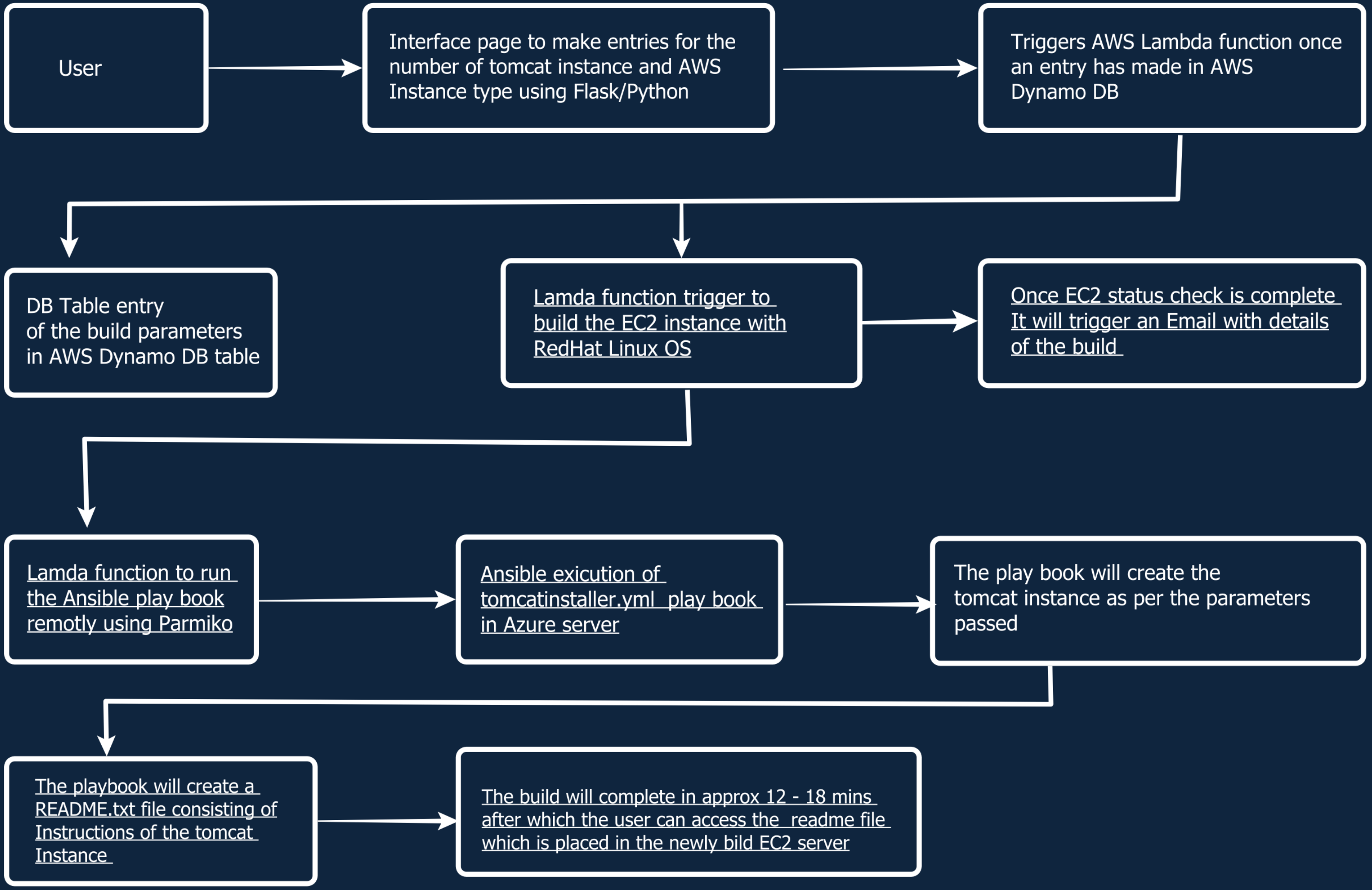
github.com/abeyva/iwba

Objective

IWBA

The project involves automating the deployment of a Tomcat server based on user-submitted parameters. Users can raise a work ticket or use a web interface to specify the build parameters for a new server, the Tomcat instance count/name. The parameters are passed to an AWS Lambda function which will create an EC2 instance (Red Hat Linux) instance based to instance type parameter. Then a second Lambda function is triggered using SNS. The tomcat instance parameters are then passed using Pramiko module to Azure server, which runs an Ansible playbook to install Tomcat, configure the server, update server.xml, and output the configuration details to a README file. During the build process the user will get an Email having the full details of the build.

Architecture diagram



User

Interface page to make entries for the number of tomcat instance and AWS Instance type using Flask/Python

Triggers AWS Lambda function once an entry has made in AWS Dynamo DB

DB Table entry of the build parameters in AWS Dynamo DB table

Lamda function trigger to build the EC2 instance with RedHat Linux OS

Once EC2 status check is complete It will trigger an Email with details of the build

Lamda function to run the Ansible play book remotly using Parmiko

Ansible exicution of tomcatinstaller.yml play book in Azure server

The play book will create the tomcat instance as per the parameters passed

The playbook will create a README.txt file consisting of Instructions of the tomcat Instance

The build will complete in approx 12 - 18 mins after which the user can access the readme file which is placed in the newly bild EC2 server

AWS Lambda (Python) Steps

- User Submission: Users submit build parameters (Tomcat instance name, Instance type etc.) via a webpage build on flask.

- **First Lambda function**: The first lambda function passes the parameters to DynamoDB table this function is also responsible for triggering the EC2 build.

- Provision EC2: The Lambda function reads parameters and provisions a Red Hat Linux EC2 instance as per the instance type.

- **Second Lambda function**: Using SNS the parameters are passed to the second Lambda function which is responsible for remotely connecting Ansible master server and passing the parameters for setting up the tomcat instances.

- Python Parmiko: Pramiko is responsible for connecting to the remote Ansible master server which is hosted in Microsoft Azure cloud the second Lambda function uses this module to trigger the Ansible build.

Ansible Playbook (YAML) Steps

- Login to EC2: The playbook logs into the EC2 instance.
- Create Directories: Creates /local/apps and other necessary directories.
- Create User’s and Groups: Creates the username and group name as specified by the input parameters
- Install JDK & Setup Tomcat: Installs OpenJDK, configures users/groups, and downloads the Tomcat package.
- Update Configuration: Modifies server.xml as per user parameters.
- Generate Output: Writes Tomcat instance details and ports into README.txt.

Technology used

AWS Cloud Platform 70% (Central India Zone)
Azure Cloud 30% (Central India Zone)

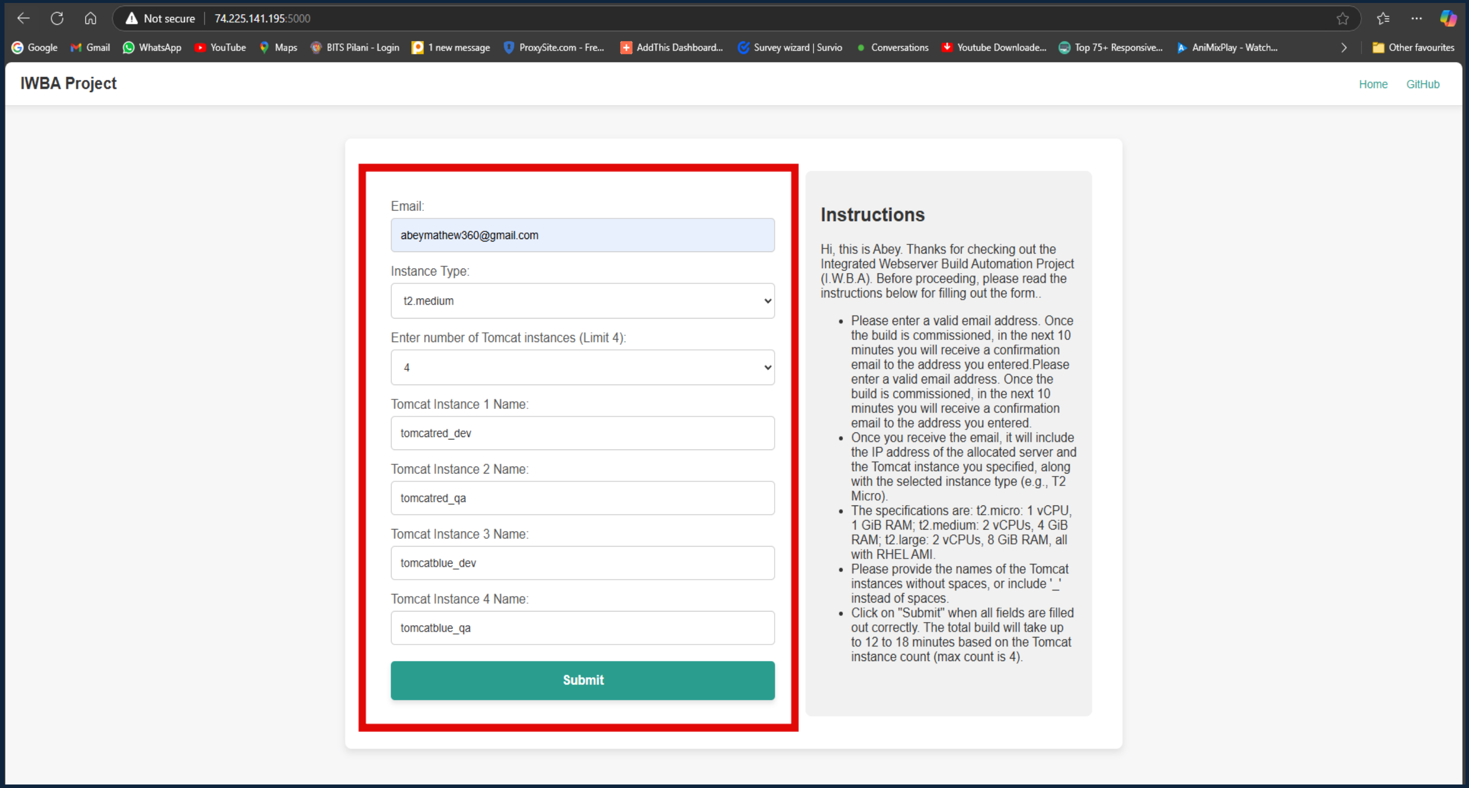
- **EC2 (Elastic Compute Cloud):** A scalable compute service that allows users to run virtual
- servers in the cloud.
- **DynamoDB**: A NoSQL database service that stores build parameters and configurations for
- server instances.
- **Lambda**: A serverless computing service that automatically runs code in response to events, such as user requests or changes in DynamoDB.
- **Tomcat**:An open-source Java servlet container that provides a pure Java HTTP web server
- environment for running Java-based applications.
- **OpenJDK**:An open-source implementation of the Java Platform, Standard Edition, used to run Java
- **Python**: Lamda functions are written in python
- **Pramiko**: For running the shell command remotely similar to Jenkins operates

Server specification: -

- **Ansible core server**: At least 4 GB of RAM running CENT OS 8.5 hosted on Microsoft Azure
  Test servers: created on virtual box with Cent OS 9 stream as OS.
- **Target server EC2 Instance**: This is hosted on AWS Cloud it runs on Red hat Linux OS.

Build Parameters Input Screen

Screenshot of the interface where users specify parameters such as the User email, Tomcat instance name, AWS server instance type, etc once the relevant fields ae filled the user can proceed for submission.



IWBA Project

Email: abeymathew360@gmail.com

Instance Type: t2.medium

Enter number of Tomcat instances (Limit 4): 4

Tomcat Instance 1 Name: tomcatred_dev

Tomcat Instance 2 Name: tomcatred_qa

Tomcat Instance 3 Name: tomcatblue_dev

Tomcat Instance 4 Name: tomcatblue_qa

Instructions

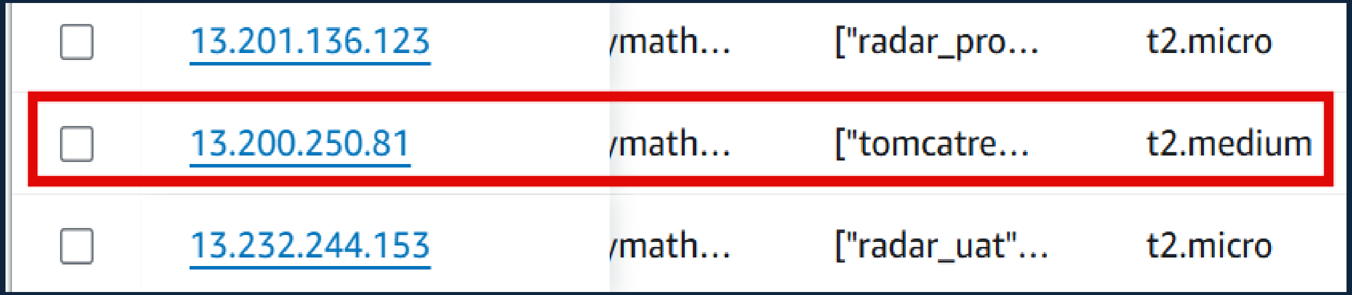
Hi, this is Abey. Thanks for checking out the Integrated Webserver Build Automation Project (I.W.B.A). Before proceeding, please read the instructions below for filling out the form..

- Please enter a valid email address. Once the build is commissioned, in the next 10 minutes you will receive a confirmation email to the address you entered.Please enter a valid email address. Once the build is commissioned, in the next 10 minutes you will receive a confirmation email to the address you entered.
- Once you receive the email, it will include the IP address of the allocated server and the Tomcat instance you specified, along with the selected instance type (e.g., T2 Micro).
- The specifications are: t2.micro: 1 vCPU, 1 GiB RAM; t2.medium: 2 vCPUs, 4 GiB RAM; t2.large: 2 vCPUs, 8 GiB RAM, all with RHEL AMI.
- Please provide the names of the Tomcat instances without spaces, or include '_' instead of spaces.
- Click on "Submit" when all fields are filled out correctly. The total build will take up to 12 to 18 minutes based on the Tomcat instance count (max count is 4).

github.com/abeyva/iwba

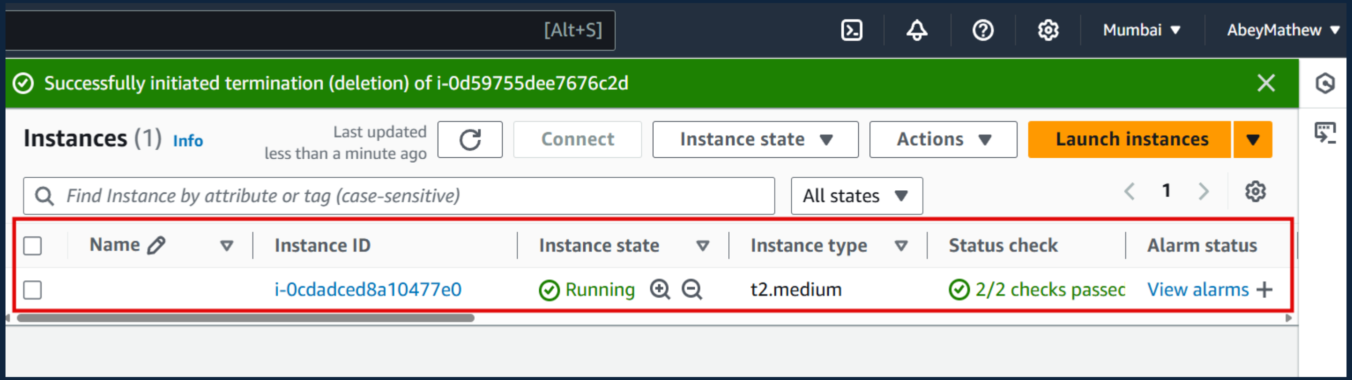
EC2 Instance Creation on AWS and DB entry

Screenshot of the DynamoDB table showing the server entry created after the request is raised.



□	13.201.136.123	/math...	["radar_pro...	t2.micro
□	13.200.250.81	/math...	["tomcatre...	t2.medium
□	13.232.244.153	/math...	["radar_uat"...	t2.micro

Screenshot from the AWS Management Console showing the created EC2 instance based on user parameters.



Successfully initiated termination (deletion) of i-0d59755dee7676c2d

Instances (1) Info

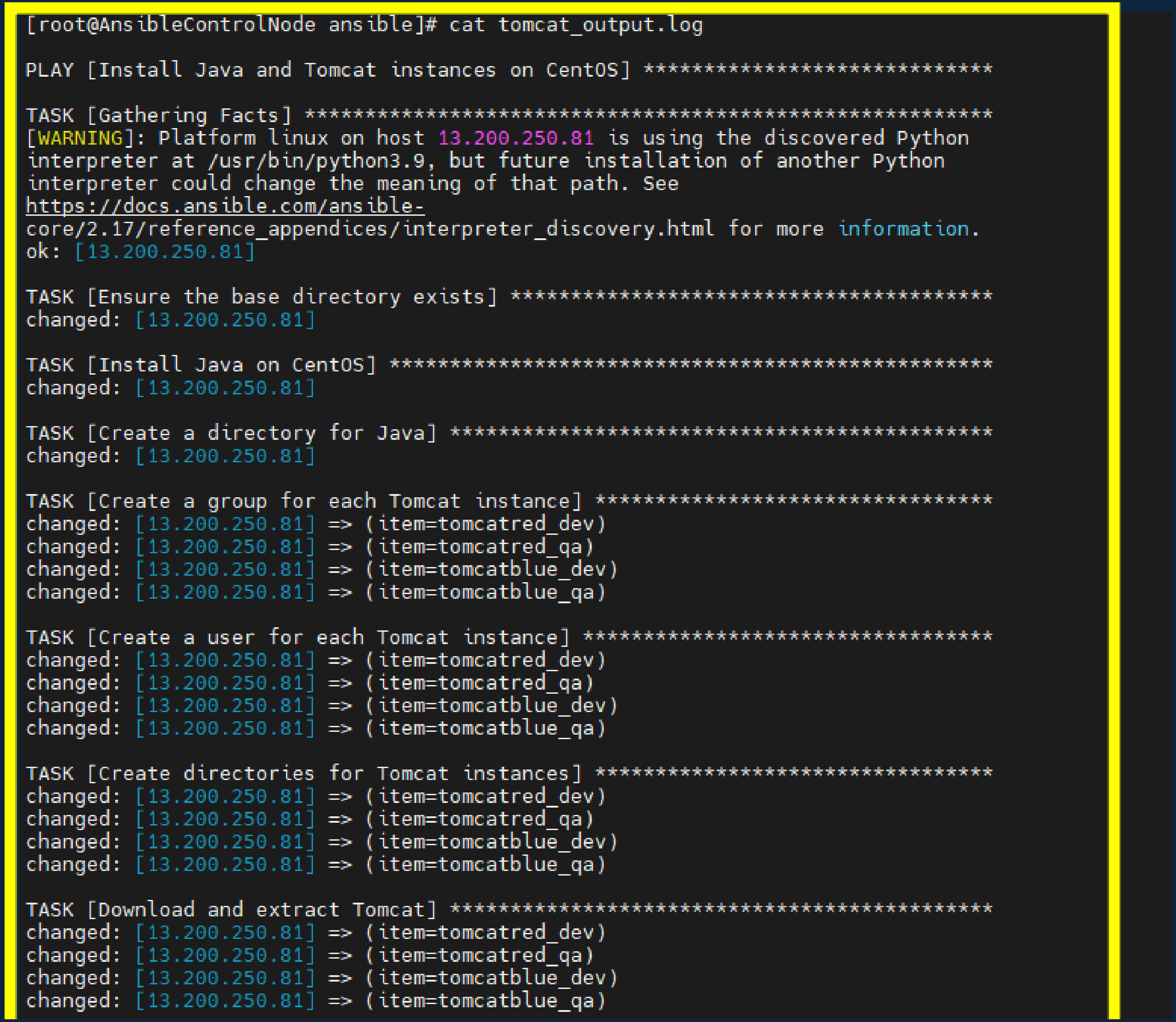
Last updated less than a minute ago

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□		i-0cdadced8a10477e0	✓ Running	t2.medium	✓ 2/2 checks passec	View alarms +

github.com/abeyva/iwba

Ansible Playbook Execution

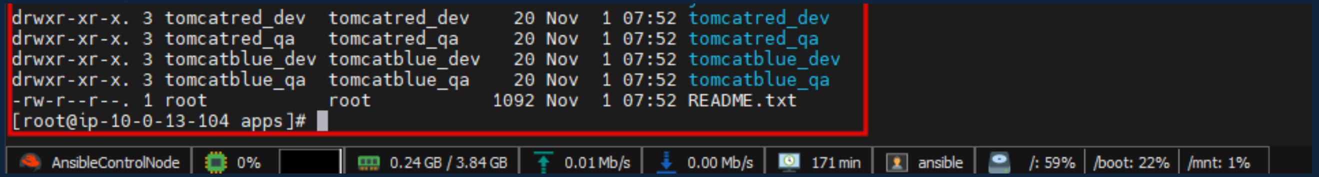
Screenshot of the log showing the execution of the Ansible playbook on Azure server.



```
[root@AnsibleControlNode ansible]# cat tomcat_output.log

PLAY [Install Java and Tomcat instances on CentOS] *****************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host 13.200.250.81 is using the discovered Python
interpreter at /usr/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [13.200.250.81]

TASK [Ensure the base directory exists] ****************************************
changed: [13.200.250.81]

TASK [Install Java on CentOS] **************************************************
changed: [13.200.250.81]

TASK [Create a directory for Java] *********************************************
changed: [13.200.250.81]

TASK [Create a group for each Tomcat instance] *********************************
changed: [13.200.250.81] => (item=tomcatred_dev)
changed: [13.200.250.81] => (item=tomcatred_qa)
changed: [13.200.250.81] => (item=tomcatblue_dev)
changed: [13.200.250.81] => (item=tomcatblue_qa)

TASK [Create a user for each Tomcat instance] **********************************
changed: [13.200.250.81] => (item=tomcatred_dev)
changed: [13.200.250.81] => (item=tomcatred_qa)
changed: [13.200.250.81] => (item=tomcatblue_dev)
changed: [13.200.250.81] => (item=tomcatblue_qa)

TASK [Create directories for Tomcat instances] *********************************
changed: [13.200.250.81] => (item=tomcatred_dev)
changed: [13.200.250.81] => (item=tomcatred_qa)
changed: [13.200.250.81] => (item=tomcatblue_dev)
changed: [13.200.250.81] => (item=tomcatblue_qa)

TASK [Download and extract Tomcat] *********************************************
changed: [13.200.250.81] => (item=tomcatred_dev)
changed: [13.200.250.81] => (item=tomcatred_qa)
changed: [13.200.250.81] => (item=tomcatblue_dev)
changed: [13.200.250.81] => (item=tomcatblue_qa)
```

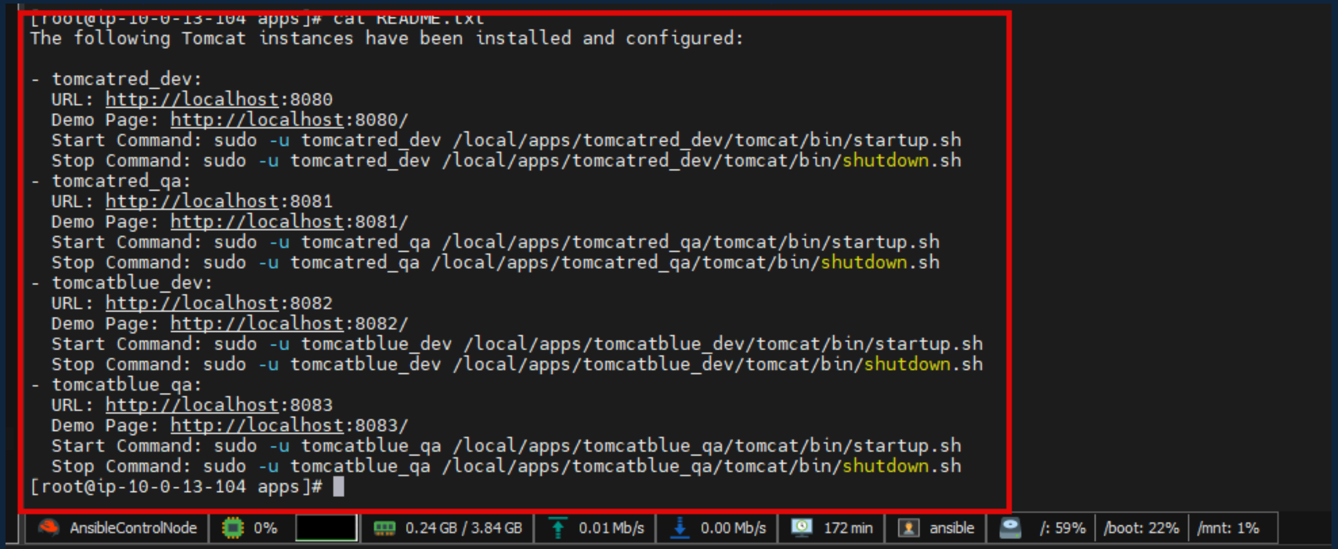
README.txt Output

Screenshot showing ls command on the installation path which is /local/apps



```
drwxr-xr-x. 3 tomcatred_dev  tomcatred_dev    20 Nov  1 07:52 tomcatred_dev
drwxr-xr-x. 3 tomcatred_qa   tomcatred_qa     20 Nov  1 07:52 tomcatred_qa
drwxr-xr-x. 3 tomcatblue_dev tomcatblue_dev   20 Nov  1 07:52 tomcatblue_dev
drwxr-xr-x. 3 tomcatblue_qa  tomcatblue_qa    20 Nov  1 07:52 tomcatblue_qa
-rw-r--r--. 1 root           root           1092 Nov  1 07:52 README.txt
[root@ip-10-0-13-104 apps]#
```

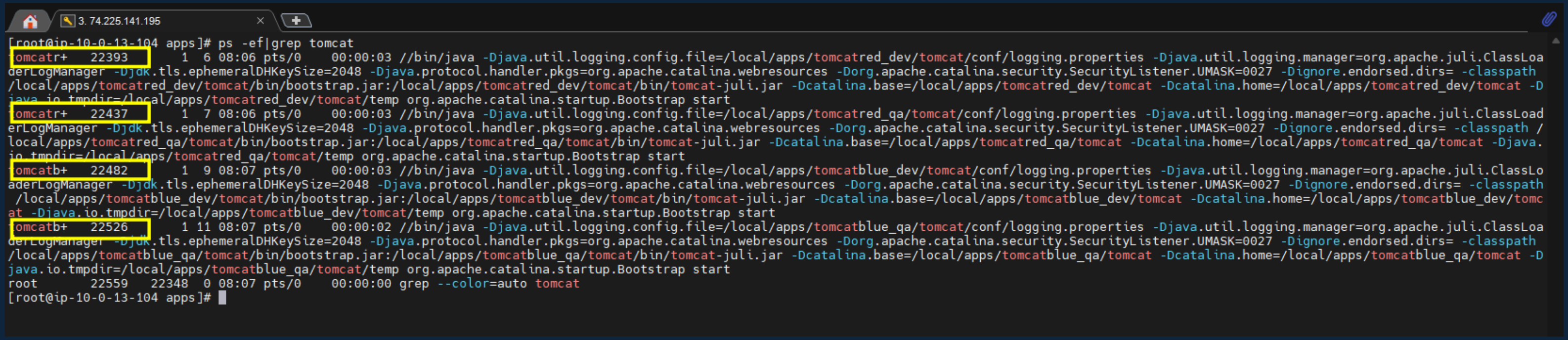
Screenshot showing the generated README.txt file containing Tomcat instance details, configured ports, and startup scripts



```
[root@ip-10-0-13-104 apps]# cat README.txt
The following Tomcat instances have been installed and configured:

- tomcatred_dev:
  URL: http://localhost:8080
  Demo Page: http://localhost:8080/
  Start Command: sudo -u tomcatred_dev /local/apps/tomcatred_dev/tomcat/bin/startup.sh
  Stop Command: sudo -u tomcatred_dev /local/apps/tomcatred_dev/tomcat/bin/shutdown.sh
- tomcatred_qa:
  URL: http://localhost:8081
  Demo Page: http://localhost:8081/
  Start Command: sudo -u tomcatred_qa /local/apps/tomcatred_qa/tomcat/bin/startup.sh
  Stop Command: sudo -u tomcatred_qa /local/apps/tomcatred_qa/tomcat/bin/shutdown.sh
- tomcatblue_dev:
  URL: http://localhost:8082
  Demo Page: http://localhost:8082/
  Start Command: sudo -u tomcatblue_dev /local/apps/tomcatblue_dev/tomcat/bin/startup.sh
  Stop Command: sudo -u tomcatblue_dev /local/apps/tomcatblue_dev/tomcat/bin/shutdown.sh
- tomcatblue_qa:
  URL: http://localhost:8083
  Demo Page: http://localhost:8083/
  Start Command: sudo -u tomcatblue_qa /local/apps/tomcatblue_qa/tomcat/bin/startup.sh
  Stop Command: sudo -u tomcatblue_qa /local/apps/tomcatblue_qa/tomcat/bin/shutdown.sh
[root@ip-10-0-13-104 apps]#
```

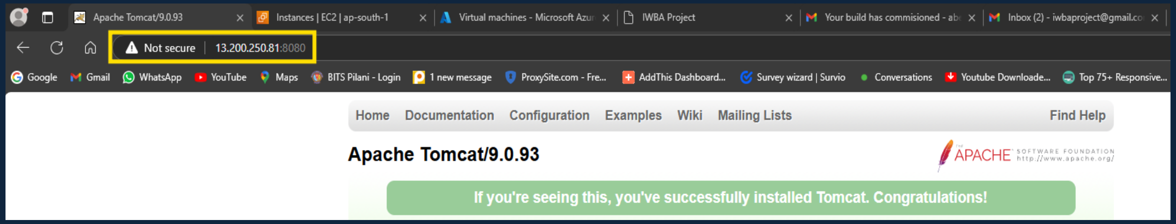
Tomcat instance running

Screenshot showing 4 tomcat instance running after executing the start-up script.



```
[root@ip-10-0-13-104 apps]# ps -ef|grep tomcat
tomcatr+   22393      1  6 08:06 pts/0    00:00:03 //bin/java -Djava.util.logging.config.file=/local/apps/tomcatred_dev/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoa
derLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath
/local/apps/tomcatred_dev/tomcat/bin/bootstrap.jar:/local/apps/tomcatred_dev/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/local/apps/tomcatred_dev/tomcat -Dcatalina.home=/local/apps/tomcatred_dev/tomcat -D
java.io.tmpdir=/local/apps/tomcatred_dev/tomcat/temp org.apache.catalina.startup.Bootstrap start
tomcatr+   22437      1  7 08:06 pts/0    00:00:03 //bin/java -Djava.util.logging.config.file=/local/apps/tomcatred_qa/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoad
erLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath /
local/apps/tomcatred_qa/tomcat/bin/bootstrap.jar:/local/apps/tomcatred_qa/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/local/apps/tomcatred_qa/tomcat -Dcatalina.home=/local/apps/tomcatred_qa/tomcat -Djava.
io.tmpdir=/local/apps/tomcatred_qa/tomcat/temp org.apache.catalina.startup.Bootstrap start
tomcatb+   22482      1  9 08:07 pts/0    00:00:03 //bin/java -Djava.util.logging.config.file=/local/apps/tomcatblue_dev/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLo
aderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath
 /local/apps/tomcatblue_dev/tomcat/bin/bootstrap.jar:/local/apps/tomcatblue_dev/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/local/apps/tomcatblue_dev/tomcat -Dcatalina.home=/local/apps/tomcatblue_dev/tomc
at -Djava.io.tmpdir=/local/apps/tomcatblue_dev/tomcat/temp org.apache.catalina.startup.Bootstrap start
tomcatb+   22526      1 11 08:07 pts/0    00:00:02 //bin/java -Djava.util.logging.config.file=/local/apps/tomcatblue_qa/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoa
derLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath
/local/apps/tomcatblue_qa/tomcat/bin/bootstrap.jar:/local/apps/tomcatblue_qa/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/local/apps/tomcatblue_qa/tomcat -Dcatalina.home=/local/apps/tomcatblue_qa/tomcat -D
java.io.tmpdir=/local/apps/tomcatblue_qa/tomcat/temp org.apache.catalina.startup.Bootstrap start
root       22559  22348  0 08:07 pts/0    00:00:00 grep --color=auto tomcat
[root@ip-10-0-13-104 apps]# 
```

Screenshot showing tomcat homage of instance's running on separate ports.

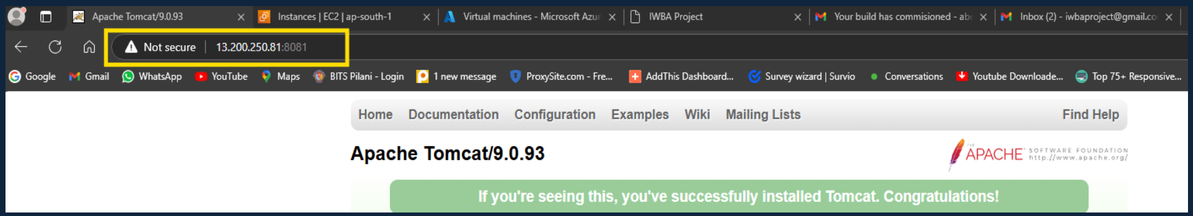


Not secure | 13.200.250.81:8080

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/9.0.93

If you're seeing this, you've successfully installed Tomcat. Congratulations!



Not secure | 13.200.250.81:8081

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/9.0.93

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Conclusion

- Automated Server Setup: Tomcat server is deployed automatically based on user-specified parameters.
- EC2 Instance Provisioning: Red Hat Linux EC2 instances are created dynamically via AWS Lambda.
- Customized Tomcat Configuration: Each Tomcat instance is set up with unique configurations (instance name, ports) specified by the user.
- Ansible Playbook Execution: The entire Tomcat installation process, including directory setup, JDK installation, and server.xml configuration, is automated.
- User-Friendly Access: The server details, configured ports, and startup scripts are stored in a README.txt file.
- Efficient Workflow: The process minimizes manual intervention, reducing errors and improving deployment speed.

Future scope

- Support for Additional Application Servers: Expand the deployment options to include other popular application servers, such as WebLogic, JBoss, or Websphere. This would make the solution more versatile for organizations with diverse technology stacks.
- User-Friendly API for External Integration:Create a RESTful API for integrating the deployment system with other applications, allowing third-party tools or internal systems to interact with the system programmatically, enhancing interoperability.
- Self-Healing Mechanisms for Fault Tolerance:Add self-healing capabilities, such as automated restart or replacement of failed EC2 instances, to maintain service availability and minimize disruptions, especially in high-availability environments.
- Automated Configuration Drift Detection:Introduce configuration drift detection to monitor and alert on unintended changes in server configurations. This would ensure that all instances remain consistent with the intended setup, reducing configuration errors.

github.com/abeyva/iwba

Thank you