components (*e.g.*, 6 hours of manual effort); manual deployment from multiple assets; manual validation across all components; break glass of privileged accounts; and resource intensive processes (*e.g.*, 10-15 deployments per week taking approximately three days).

[0006] Moreover, the manual work required in current processes is prone to human error which results in instability and unavailability of the platform for users.

[0007] These and other drawbacks exist.

SUMMARY OF THE INVENTION

[0007a] According to a first aspect of the present invention, there is provided a system that implements an orchestration engine in a software platform environment, the system comprising:

an electronic input configured to receive an input configuration file via a web layer;

a database configured to communicate with a data access layer and further configured to store states, data and logs; and

an orchestration engine comprising a computer processor, the orchestration engine coupled to the electronic input and the database and further programmed to perform the steps of:

receiving deployment input data from the input configuration file;

initiating a deployment data enrichment process based on the deployment input data;

performing a pre-validation service that submits a pre-validation service invocation request and receives one or more validation results;

applying an operation execution that deploys an application and restarts a subset of software components;

performing a post-validation service of an entire infrastructure that submits a post-validation service invocation request and receives one or more validation results;

generating and providing, via an interactive interface, status relating to the deployment data enrichment process, wherein the interactive interface comprises a deployment status portion, a feed details portion, a deployment summary portion and a deployment history portion;

generating and transmitting one or more alerts to one or more impacted teams for intervention; and

displaying history data of upgrade operations listed in a descending order of timestamp, wherein the history data includes a history page detailing data of a name of an upgrade activity, user identifier of person or entity who submitted the operation, status of the operation, time stamp, actions, and sub-tasks statistics data under each operation.

[0007b] According to a second aspect of the present invention, there is provided a method that implements an orchestration engine in a software platform environment, the method comprising the steps of:

receiving, via an electronic input, deployment input data from an input configuration file, wherein the electronic input is configured to receive an input configuration file via a web layer;

initiating, via an orchestration engine, a deployment data enrichment process based on the deployment input data, wherein the orchestration engine comprises a computer processor and is coupled to the electronic input and a database, wherein the database is configured to communicate with a data access layer and further configured to store states, data and logs;

performing a pre-validation service that submits a pre-validation service invocation request and receives one or more validation results;

applying an operation execution that deploys an application and restarts a subset of software components;

performing a post-validation service of an entire infrastructure that submits a post-validation service invocation request and receives one or more validation results;

generating and providing, via an interactive interface, status relating to the deployment data enrichment process, wherein the interactive interface comprises a deployment status portion, a feed details portion, a deployment summary portion and a deployment history portion;

generating and transmitting one or more alerts to one or more impacted teams for intervention; and

displaying history data of upgrade operations listed in a descending order of timestamp, wherein the history data includes a history page detailing data of a name of an upgrade activity, user identifier of person or entity who submitted the operation, status of the operation, time stamp, actions, and sub-tasks statistics data under each operation.

[0008] There is disclosed a system that implements an orchestration engine in a software platform environment, such as Splunk. The system comprises: an electronic input configured to receive an input configuration file via a web layer; a database configured to communicate with a data access layer and further configured to store states, data and logs; and an orchestration engine comprising a computer processor, the orchestration engine coupled to the electronic input and the database and further programmed to perform the steps of: receiving deployment input data from the input configuration file; initiating a deployment data enrichment process based on the deployment input data; performing a pre-validation service that submits a pre-validation service invocation request and receives one or more validation results; applying an operation execution that deploys an application and restarts a subset of software components; performing a post-validation service of an entire infrastructure that submits a post-validation service invocation request and receives one or more validation results;

and generating and providing, via an interactive interface, status relating to the deployment data enrichment process, wherein the interactive interface comprises a deployment status portion, a feed details portion, a deployment summary portion and a deployment history portion.

[0009] There is disclosed a method that implements an orchestration engine in a software platform environment, such as Splunk. The method comprises the steps of: receiving, via an electronic input, deployment input data from an input configuration file, wherein the electronic input is configured to receive an input configuration file via a web layer; initiating, via an orchestration engine, a deployment data enrichment process based on the deployment input data, wherein the orchestration engine comprises a computer processor and is coupled to the electronic input and a database, wherein the database is configured to communicate with a data access layer and further configured to store states, data and logs; performing a pre-validation service that submits a pre-validation service invocation request and receives one or more validation results; applying an operation execution that deploys an application and restarts a subset of software components; performing a post-validation service of an entire infrastructure that submits a post-validation service invocation request and receives one or more validation results; and generating and providing, via an interactive interface, status relating to the deployment data enrichment process, wherein the interactive interface comprises a deployment status portion, a feed details portion, a deployment summary portion and a deployment history portion.

[0010] An embodiment may include a specially programmed computer system comprising one or more computer processors, interactive interfaces, electronic storage devices, and networks. The computer implemented system and method described herein provide unique advantages to entities, security analysts, users and other participants. An embodiment is directed to eliminating inefficient and resource intensive processes to deliver application code into a distributed environment with confidence. The innovative solution further addresses inefficiencies relating to human error, frequent releases and

downtime. An embodiment seeks to facilitate and improve required logging and traceability. In addition, users may engage in an automated or one click deployment solution that reduces significant execution time for application code releases. This further eliminates manual intervention, automated backups, document reviews thereby significantly reducing or even eliminating human error and providing an end-to-end validation. An embodiment further promotes ease of use by providing a single platform to track execution progress and historical changes, monitor change related logs, track errors, trace and correlate deployment issues.

[0011] These and other advantages will be described more fully in the following detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] In order to facilitate a fuller understanding of the present invention, reference is now made to the attached drawings. The drawings should not be construed as limiting the present invention, but are intended only to illustrate different aspects and embodiments of the present invention.

[0013] Figure 1 is an exemplary system diagram, according to an embodiment of the present invention.

[0014] Figure 2 is a detailed illustration of Orchestrator, according to an embodiment of the present invention.

[0015] Figure 3 is an exemplary process flow, according to an embodiment of the present invention.

[0016] Figure 4 is an exemplary user interface, according to an embodiment of the present invention.

WHAT IS CLAIMED IS:

1. A system that implements an orchestration engine in a software platform environment, the system comprising:

an electronic input configured to receive an input configuration file via a web layer;

a database configured to communicate with a data access layer and further configured to store states, data and logs; and

an orchestration engine comprising a computer processor, the orchestration engine coupled to the electronic input and the database and further programmed to perform the steps of:

receiving deployment input data from the input configuration file;

initiating a deployment data enrichment process based on the deployment input data;

performing a pre-validation service that submits a pre-validation service invocation request and receives one or more validation results;

applying an operation execution that deploys an application and restarts a subset of software components;

performing a post-validation service of an entire infrastructure that submits a post-validation service invocation request and receives one or more validation results;

generating and providing, via an interactive interface, status relating to the deployment data enrichment process, wherein the interactive interface comprises a deployment status portion, a feed details portion, a deployment summary portion and a deployment history portion;

generating and transmitting one or more alerts to one or more impacted teams for intervention; and

displaying history data of upgrade operations listed in a descending order of timestamp, wherein the history data includes a history page detailing data of a name of an upgrade activity,

user identifier of person or entity who submitted the operation, status of the operation, time stamp, actions, and sub-tasks statistics data under each operation.

2. The system of claim 1, wherein the data access layer is configured to communicate with a service layer that executes one or more business services.

3. The system of claim 2, wherein the data access layer is configured to communicate with the web layer that executes the interactive interface that receives one or more inputs from a user.

4. The system of claim 1, wherein the deployment status portion comprises deployment stage progress.

5. The system of claim 4, wherein the deployment stage progress comprises process input and deployment agent restarts.

6. The system of claim 1, wherein the deployment summary portion comprises summary data relating to one or more operations.

7. The system of claim 1, wherein the deployment history portion comprises data relating to reference identifier, date, change identifier and status data.

8. The system of claim 1, wherein the orchestration engine is further programmed to perform the step of: collecting event log data.

9. The system of claim 1, wherein the actions include pause, resume, cancel or other actions depending on the current status of operation.

10. The system of claim 1, wherein the software platform environment is Splunk.

11. A method that implements an orchestration engine in a software platform environment, the method comprising the steps of:

receiving, via an electronic input, deployment input data from an input configuration file, wherein the electronic input is configured to receive an input configuration file via a web layer;

initiating, via an orchestration engine, a deployment data enrichment process based on the deployment input data, wherein the orchestration engine comprises a computer processor and is coupled to the electronic input and a database, wherein the database is configured to communicate with a data access layer and further configured to store states, data and logs;

performing a pre-validation service that submits a pre-validation service invocation request and receives one or more validation results;

applying an operation execution that deploys an application and restarts a subset of software components;

performing a post-validation service of an entire infrastructure that submits a post-validation service invocation request and receives one or more validation results;

generating and providing, via an interactive interface, status relating to the deployment data enrichment process, wherein the interactive interface comprises a deployment status portion, a feed details portion, a deployment summary portion and a deployment history portion;

generating and transmitting one or more alerts to one or more impacted teams for

intervention; and

displaying history data of upgrade operations listed in a descending order of timestamp, wherein the history data includes a history page detailing data of a name of an upgrade activity, user identifier of person or entity who submitted the operation, status of the operation, time stamp, actions, and sub-tasks statistics data under each operation.

12. The method of claim 11, wherein the data access layer is configured to communicate with a service layer that executes one or more business services.

13. The method of claim 12, wherein the data access layer is configured to communicate with a web layer that executes the interactive interface that receives one or more inputs from a user.

14. The method of claim 11, wherein the deployment status portion comprises deployment stage progress.

15. The method of claim 14, wherein the deployment stage progress comprises process input and deployment agent restarts.

16. The method of claim 11, wherein the deployment summary portion comprises summary data relating to one or more operations.

17. The method of claim 11, wherein the deployment history portion comprises data relating to reference identifier, date, change identifier and status data.

18. The method of claim 11, further comprising the step of: collecting event log data.

19. The method of claim 11, wherein the actions include pause, resume, cancel or other actions depending on the current status of operation.

20. The method of claim 11, wherein the software platform environment is Splunk.