### **Knowledge Transfer (KT) Document : SRE**

#### Document Information

* Document Title: Overview of Deployment process by Cloud Bees, Tableau, Informatica, HIP, Tableau, Tiger, WFA, Beyond Trust, RPA.
* Version:1.1
* Date:29th Apr 2024 – 3rd may 2024
* Prepared By: SRE-Coforge
* Reviewed By: Satya Akella (Coforge)
* Table of Contents

[**Knowledge Transfer (KT) Document : SRE** 1](#_Toc165902921)

[1.1 Overview of the deployment process: 3](#_Toc165902922)

[1.2 Deployment Process: 3](#_Toc165902923)

[1.3 Deployment issues and enhancements: 4](#_Toc165902924)

[1.4 Conclusion: 4](#_Toc165902925)

[2.1 Introduction to Tableau: 5](#_Toc165902926)

[2.2 Architecture Overview: 5](#_Toc165902927)

[2.3 Application Overview : 6](#_Toc165902928)

[3. Introduction to Informatica: 7](#_Toc165902929)

[4. HIP (Hybrid Informatica Platform): 9](#_Toc165902930)

[4.1.1 What is HIP? 9](#_Toc165902931)

[4.1.2 Uses of HIP 9](#_Toc165902932)

[4.1.3 How is HIP Used? 9](#_Toc165902933)

[4.2 Application Integration Console: 9](#_Toc165902934)

[4.2.1 B2B Gateway: 10](#_Toc165902935)

[4.2.2 Data Integration: 10](#_Toc165902936)

[4.2.3 Administrator Console: 10](#_Toc165902937)

[4.2.4 Monitor: 10](#_Toc165902938)

[4.3 Agent Management: 10](#_Toc165902939)

[4.4 Connection Management: 10](#_Toc165902940)

[4.5 Server Environment: 10](#_Toc165902941)

[4.5.1 Server Setup and Configuration: 10](#_Toc165902942)

[4.5.2 Folder Structure and Management: 11](#_Toc165902943)

[4.6 Authentication and Runtime Environment: 11](#_Toc165902944)

[4.7 Connections and Data Integration: 11](#_Toc165902945)

[5. Introduction to Tiger: 11](#_Toc165902946)

[5.1 Components and Infrastructure of Tiger: 11](#_Toc165902947)

[5.2 User Interaction and Permissions: 12](#_Toc165902948)

[5.3 Application Details: 12](#_Toc165902949)

[5.4 RabbitMQ Configuration: 12](#_Toc165902950)

[5.5 Redis Cache Configuration: 12](#_Toc165902951)

[5.6 Apache Web Server Configuration: 12](#_Toc165902952)

[5.7 User Interface and Functionality: 12](#_Toc165902953)

[5.8 Importance and Impact of Tiger: 13](#_Toc165902954)

[5.9 Conclusion: 13](#_Toc165902955)

[6. Introduction to WFA (Workforce Analytics): 13](#_Toc165902956)

[6.1 Infrastructure Components: 13](#_Toc165902957)

[6.2 Environments and Gateways Servers: 14](#_Toc165902958)

[6.3 Application Configuration and Property Files: 14](#_Toc165902959)

[6.4 Monitoring and Observability: 14](#_Toc165902960)

[6.4.1 Specific Application Monitoring: 14](#_Toc165902961)

[6.4.2 Apache Web Servers: 15](#_Toc165902962)

[6.4.3 Granular Monitoring with AppDynamics: 15](#_Toc165902963)

[6.5 Frequent Issues and Support: 15](#_Toc165902964)

[6.6 Troubleshooting Approaches: 15](#_Toc165902965)

[6.8 Conclusion: 15](#_Toc165902966)

[7.Introduction to Beyond Trust: 16](#_Toc165902967)

[7.1 Authentication and Access Control: 16](#_Toc165902968)

[7.3 Usage of Beyond Trust: 16](#_Toc165902969)

[7.4 Account Management: 16](#_Toc165902970)

[7.5 Incident Handling: 16](#_Toc165902971)

[7.6 Security Measures: 16](#_Toc165902972)

[8. Introduction to RPA: 17](#_Toc165902973)

[8.1 RPA Environment: 17](#_Toc165902974)

[8.2 Architecture Overview: 17](#_Toc165902975)

[8.3 App Admin Responsibilities: 17](#_Toc165902976)

[8.4 Incident Handling and Audit Logs: 17](#_Toc165902977)

[8.5 Accessing Control Room and Server Maintenance: 17](#_Toc165902978)

[8.6 Troubleshooting and Common Issues: 17](#_Toc165902979)

[8.7 Conclusion: 18](#_Toc165902980)

### 1.1 Overview of the deployment process:

##### 1.1.1 Deployment Teams:

* + Two teams mentioned: App Deployment Automation and App Deployment.
  + App Deployment Automation responsible for creating, scheduling, and configuring pipelines.
  + App Deployment team utilizes these pipelines for deployment tasks.

##### 1.1.2 Deployment Process Overview:

* + Discussed the creation of deployment checklists for various applications.
  + Mentioned the backup and rollback strategy, aiming for automated processes.
  + Importance of communication with stakeholders during both successful and failed deployments.
  + Automated notifications via ServiceNow and email for deployment status updates.

##### 1.1.3 Deployment Tool Usage:

* + Primarily using CloudBees for deployment, isolating deployment from Jenkins.
  + Troubleshooting facilitated by CloudBees support for enterprise versions.

##### 1.1.4 Process Execution:

* + Initiation of deployment based on CR approval in version control.
  + Accessing details from ServiceNow for deployment plans and configurations.
  + Standard steps for pipeline execution include status email notifications.

##### 1.1.5 Pipeline Structure and Configuration:

* + The pipeline consists of stages for different environments like QE and Prod.
  + Each stage has its own configuration, specifying environment-specific properties like server resources and credentials.
  + Components within the pipeline, such as bots for RPA, are defined with their own processes.
  + Processes are scripted and configured to execute specific procedures for deployment.

### 1.2 Deployment Process:

* + The deployment process involves selecting the appropriate pipeline and stage based on the change request (CR).
  + Input parameters such as bot name(war file name) and version are specified during deployment initiation.
  + Procedures within the pipeline execute deployment scripts, pulling artifacts from repositories and deploying them to target environments.
  + Credentials and other sensitive information required for deployment are stored as secrets and accessed during execution.

##### 1.2.1 Deployment Monitoring and Validation:

* + Post-deployment, health checks are conducted to ensure the application is up and running.
  + Monitoring and observability tools continuously monitor the deployed components and alert in case of any issues.
  + Validation includes thorough checks to ensure the environment is functioning correctly and the deployed changes are reflected as expected.
  + App deployment team ensures the success of the deployment process, while SREs oversee the health and performance of the overall system.

##### 1.2.2 Troubleshooting and Debugging:

* 1. In case of deployment failures, troubleshooting involves identifying issues such as missing artifacts or incorrect configurations.
  2. Logs and error messages provide insights into the cause of failures, aiding in resolution.
  3. Collaboration between app deployment and SRE teams may be required for addressing issues like server space constraints or connectivity issues.

### 1.3 Deployment issues and enhancements:

##### 1.3.1 Post-Deployment Validation:

* + Mentioned the use of a static page called "version.json" for basic validation post-deployment.
  + Static page contains information like API version, build details, and branch details.
  + Validation involves checking if the static page returns an HTTP 200 OK status.

##### 1.3.2 Validation Gaps:

* + Highlighted gaps in validation, including issues beyond basic HTTP status checks.
  + Challenges include identifying context.xml issues, JDBC connectivity problems, and dependencies on file system locations.
  + Current deployment tools focus on basic validation, not comprehensive application health checks.

##### 1.3.3 Rollback Strategy:

* + Identified manual processes for rollback in case of deployment issues.
  + As of now there is no automated rollback mechanism in the deployment tool.
  + Manual tasks involve identifying the previous version, taking snapshots, and redeploying the stable version.

##### 1.3.4 Space Management:

* + Discussed challenges with server space management during deployment.
  + Manual intervention required to clean up occupied space, often involving SRE team assistance.
  + Proposed automation of space clean-up tasks, especially for lower environments.

### 1.4 Conclusion:

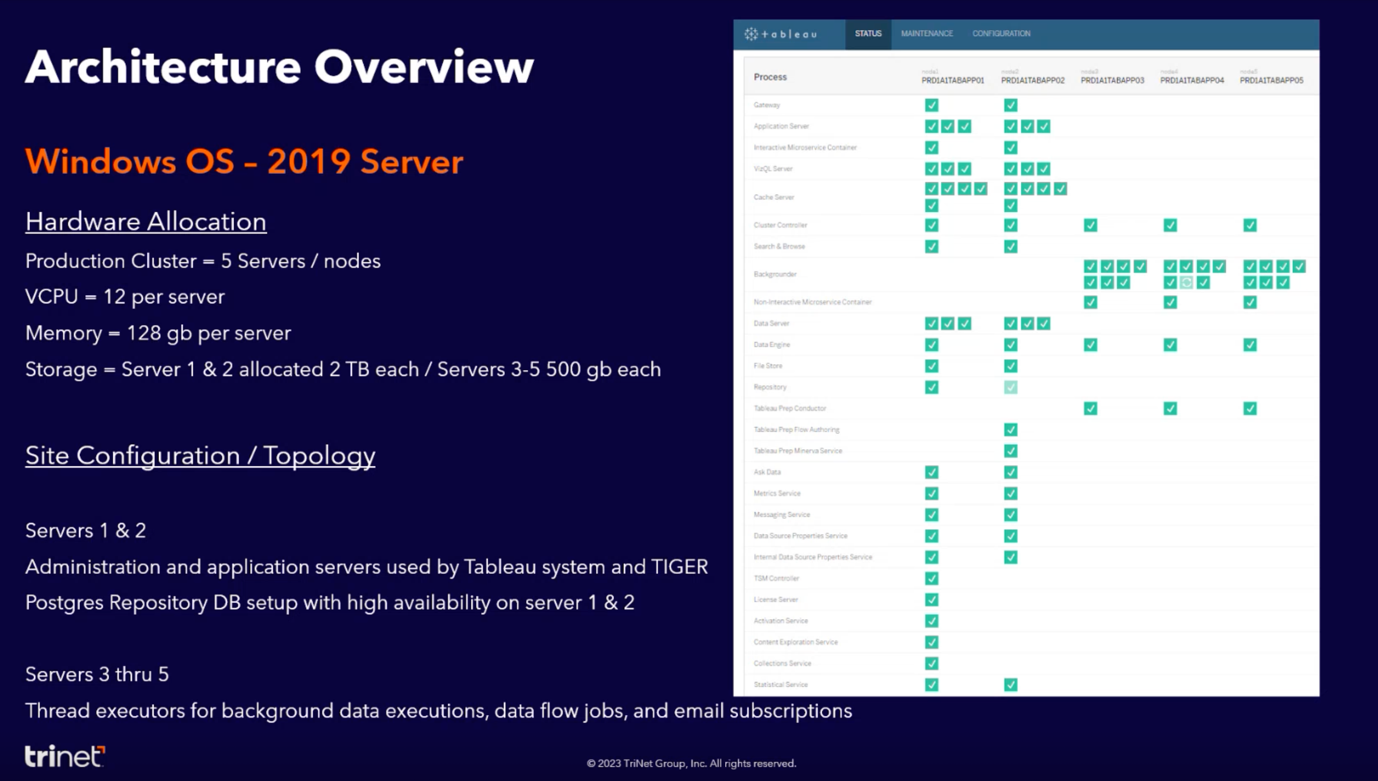
* + Recognized the importance of addressing gaps in post-deployment validation, rollback strategies, and space management.
  + Emphasized the need for automation and continuous improvement in deployment processes to enhance efficiency and reliability.

### **2.1 Introduction to Tableau**:

* Tableau is a reporting application adopted by TriNet about seven years ago, aiming to replace Cognos. It serves as the internal reporting platform for various departments within TriNet.
* **Usage Across Departments**: Tableau is utilized by multiple departments within TriNet, including human resources, technology, sales, customer experience, finance, treasury, project management, and executive leadership.
* **User Groups and Roles**: There are two main user groups for Tableau: developers and support staff who create and manage reports, and end users who access and view the reports. Permissions and access are managed through roles and Tiger, an internally developed application.
* **Data Sources**: Tableau connects to various data sources, including the data warehouse, EA products fee, Tiger Rabbit MQ data, ServiceNow, Salesforce, and Excel documents.
* **Frontend Dashboard**: Users interact with Tableau through a frontend dashboard where they can create, publish, and access reports. The dashboard provides a project tree structure for organizing reports and data connections.
* **Data Connections and Exploration**: Users can create and manage data connections to various databases, and administrators can explore and monitor data sources, workbooks, and data flows.
* **Operational Support**: Daily operational support for Tableau includes tasks such as updating database connection details, monitoring data refresh processes, and managing background tasks for extracts.

### 2.2 Architecture Overview:

TriNet's Tableau environment consists of production and staging environments. The production environment includes five Windows servers, with the first two servers serving as administration and repository servers, and the remaining servers acting as background extract runners.



* + 1. Infrastructure Setup:

Tableau is hosted on five servers, each with 12 CPUs, 128 GB of RAM, and 2 TB of storage. The setup includes two drives, one for the Windows operating system and another for Tableau applications to ensure optimal performance.

* + 1. Backup and Restore Process:

Tableau backups are stored on a network file share after being generated on the E drive. These backups are crucial for data recovery in case of system failure or site migration. The restore process typically takes about two hours.

##### 2.2.3 Data Sources and Drivers:

Tableau connects to various data sources such as Amazon Athena, Snowflake, and Oracle databases.

* + 1. Administration and Monitoring:

The Tableau server environment is managed through an administration interface, which allows for configuring services, monitoring server status, and generating support snapshots for troubleshooting.

### 2.3 Application Overview :

##### **2.3.1 Dashboard and UI Console**:

Dashboard and UI console that provide detailed information about extract refreshes, subscriptions, and data flows. We can drill down into specific reports, connections, or schedules to analyse performance and troubleshoot issues.

##### **2.3.2 Scheduling and Automation**:

We use scheduling features to automate tasks like data extracts, subscriptions, and data flows. We can create new schedules, manage existing ones, and adjust settings like frequency and parallelism.

##### **2.3.3 User Management and Access Control**:

User management and access control are configured to Active Directory. You rarely need to manage groups and users directly, as it's mostly handled through AD integration.

##### **2.3.4 System Health and Maintenance**:

We can regularly check the site status to ensure system health and performance. This includes monitoring server status, network traffic, and service allocation. We also perform tasks like license checks, log file analysis, and system upgrades.

##### 2.3.5 Infrastructure Management:

Infrastructure migrations and site switchovers are done for ensuring smooth transitions between data centres. Tools like Jenkins are used for automation and efficiency, especially for tasks like backups and system restarts. There are even specific Jenkins jobs created for site switchovers.

Overall, Tableau plays a critical role in Trinet's data management and reporting strategy, supporting various business functions with actionable insights and analytics capabilities.

### **3. Introduction to Informatica**:

Informatica Power Center is an ETL (Extract, Transform, Load) tool used for data integration and management. It allows users to extract data from various sources, transform it according to business requirements, and load it into target systems. Power Center is crucial for organizations that deal with large volumes of data and need to ensure its accuracy, consistency, and security.

##### **3.1 Informatica Power Client:**

Informatica PowerClient is a component of the Informatica platform that provides users with a web-based interface for managing and monitoring Informatica services and tasks.

##### 3.1.1 Environment Setup:

* + Three environments: Development (DV01), Quality Assurance (QEN1, QEN2), STG1(OCI) and Production (Ashburn and Phoenix).
  + Development runs on AWS while QE and Production run on OCI (Oracle Cloud Infrastructure).
  + Power Center sites are located in Phoenix and Ashburn.

##### **3.1.2 Components**:

There are fourcomponents of Informatica Power Center, such as

the Power Center client .

* Power Center Designer ​
* Power Center Repository Manager     ​
* Power Center Workflow Manager​
* Power Center Workflow Monitor

Each component plays a crucial role in designing, managing, and monitoring ETL workflows.

* **Power Center Designer:** Allows granular level management of sources, targets, mappings, sessions. Designer is used by data management Developer, QE, and support.
* **Power Center Repository Manager:** Functions as a library for storing and managing objects and content. Permissions are defined for folders within the repository.
* **Power Center Workflow Manager:** Workflow Manager enables designing and managing workflows and sessions with graphical representations.
* **Power Center Workflow Monitor:** Provides a user-friendly interface to monitor job statuses, retrieve logs, etc.

##### 3.1.3 Support and Operational Aspects:

* Support includes troubleshooting workflows, session management, and application upgrades.
* Permissions and access are managed through Active Directory groups, ensuring security and control.
* Routine maintenance tasks include password changes, application upgrades, and disaster recovery support.
* Access audits are facilitated for internal and external auditing purposes.

##### 3.1.4 Data Sources and Connections:

* Power Center interacts with various data sources including relational databases like Oracle, MongoDB, SQL, and applications like Salesforce, ServiceNow, etc.
* Reporting tools like Tableau and external reporting tools like WFA utilize data managed by Power Center.

##### 3.1.5 SOX Compliance:

* Access permission audited quarterly .
* Password change policy applies for Production: 180 days rotation for DB , Service and Domain accounts.

##### 3.1.6 Conclusion:

##### Informatica Power Center is a critical component in TriNet's data management infrastructure, facilitating ETL operations across different environments. It ensures data integrity, security, and efficiency in handling large volumes of data across diverse sources and destinations.

### 

### 4. HIP (Hybrid Integration Platform):

* + HIP is Hosted by Informatica, offering a cloud-based ETL solution.
  + Provides more extensive functionalities beyond ETL, such as encrypted file transfer and B2B integration.
  + Managed by Informatica, with connectivity maintained by the organization internally.

### 4.1.1 What is HIP?

* + HIP stands for Hybrid Integration Platform.
  + It serves as an integration platform between on-premises and cloud systems.
  + Data is securely transferred between on-premises and cloud using a secure agent, hence the term "hybrid."
  + Types of In Integration:
* Data Integration
* Application Integration

### 4.1.2 Uses of HIP

* + Cloud applications like Salesforce, Snow, Workday and data sources like EA are users of HIP.
  + On-premises applications such as PeopleSoft and Oracle DB. Oracle E-Suite also utilize HIP.

### 4.1.3 How is HIP Used?

* + HIP facilitates two main kinds of integrations: data integration and application integration.
  + Data integration involves scheduled jobs for transferring files or information between systems.
  + Application integration creates APIs for exchanging data between applications or services.
  + HIP components include Application Integration Console, B2B Gateway, Data Integration Administrator, and Monitor.

### 4.2 Application Integration Console:

* + Provides an overview of APIs and running jobs.
  + Shows job status, schedules, and detailed logs for each job.

### 4.2.1 B2B Gateway:

* + Acts as a security layer for encrypting data during transmission.
  + Ensures secure data transfer between HIP and end clients.

### 4.2.2 Data Integration:

* + Manages scheduled jobs and server environments.
  + Monitors processes and allows starting or stopping agent services.

### 4.2.3 Administrator Console:

It provides a complete overview of the HIP agents we have and what is the agent for. Whether it is a EVW agent or a secure agent that is running.

### 4.2.4 Monitor:

It tells us what are the processes running on the particular server on a particular secure agent.

### 4.3 Agent Management:

* + Agents serve as connectivity devices between HIP and infrastructure.
  + Agents communicate with Informatica Cloud for automatic upgrades.
  + Agents can be started or stopped independently of running processes.
  + Processes need to be manually restarted if agents are stopped while processes are running.

### 4.4 Connection Management:

* + BeyondTrust is used for securing passwords and managing password rotations.
  + Connection details include client IDs, secrets, URLs, and authentication methods.
  + Connections can be associated with specific processes, and changes to connections require republishing associated processes.

### 4.5 Server Environment:

* + Servers are categorized based on their usage for specific tasks like TriNet or EDW.
  + Each server hosts a set of agent services responsible for different processes.
  + The Data Integration Administrator allows monitoring and managing the status of servers and their associated processes.

### 4.5.1 Server Setup and Configuration:

* + Access to servers using a Sudo account named **hip\_user**.
  + Software installed at the root of the apps folder, within which lies the agent core for Informatica.
  + The **infra agent** service is used to **start and stop the agents**, similar to Power Center's startup process.
  + Shutting down the agent is done with "**infra agent shutdown**" and starting it with "**infra agent startup**".
  + Agent core log files provide details on start and stop issues.

### 4.5.2 Folder Structure and Management:

* + Folders under apps/hip are managed with ZFS, allowing for data processing with associated permissions.
  + Windows mounts are identified by "AD.Corp.local" and managed using prod mount.

### 4.6 Authentication and Runtime Environment:

* + Authentication is token-based, managed through Informatica Cloud, with runtime environment monitoring available in the admin console.
  + Agent hosts and communication with cloud infrastructure are managed through tokens.

### 4.7 Connections and Data Integration:

* + Administration console manages connections, with live changes reflecting immediately.
  + Testing connections ensures functionality before saving changes.
  + Dependencies of connections are crucial, especially when updating passwords to ensure published changes are absorbed by associated processes.

##### 4.8 Conclusion:

This comprehensive summary covers the main points discussed in HIP with Informatica.

## 5. Introduction to Tiger:

* TIGER- TriNet Information Gateway Enterprise Reporting
* Tiger is an internal reporting mechanism used by TriNet for various departments including human resources, finance, sales, treasury, and project management.
* It drives customer experience and facilitates targeted sales through data utilization.

5.1 Components and Infrastructure of Tiger:

* Tiger serves as the front end UI for Tableau, allowing users to interact with Tableau reports.
* Infrastructure components include an application server, Redis cache server, RabbitMQ for messaging, Mongo database, and Apache Web server attached to a load balancer.
* Users request access to reports through Tiger, triggering a workflow in ServiceNow for approval.

### 5.2 User Interaction and Permissions:

* Users access reports and datasets through Tiger and can request access based on their roles.
* Managers approve access requests within Tiger, and users can also publish roles and access report catalogues.
* Tiger has multiple environments including QEN 1, QEN 2, STG 1, and PROD, with sites in Phoenix and Ashburn.

### 5.3 Application Details:

* Tiger Infrastructure has various Applications & Services that points to targets like 1) Tableau 2) EA Dataware house.
* Applications are managed on the same server with separate log files and property files for each.
* Dependencies include RabbitMQ for messaging, MongoDB, and Redis for caching.

5.4 RabbitMQ Configuration:

* RabbitMQ is used for messaging between components and requires correct configuration for applications to function.
* RabbitMQ authentication is done with an account called analytics.

### 5.5 Redis Cache Configuration:

* Redis cache is used for performance and caching purposes.
* Tiger uses Redis cache standalone for internal purposes and Redis Enterprise for external clients(WFA).
* Configuration includes specifying ports and passwords in property files.

### 5.6 Apache Web Server Configuration:

* Apache Web server is used for serving Tiger and WFA.
* Configuration involves ensuring connectivity with Tableau and Tomcat on the application server.

### 5.7 User Interface and Functionality:

* Tiger provides a user-friendly interface with SSO login.
* Users can access reports, request permissions, and manage ownership through the interface.
* Access requests undergo approval workflows and are audited periodically for compliance.

### 5.8 Importance and Impact of Tiger:

* Tiger plays a crucial role in managing data access and integrity within TriNet.
* It evolved from a manual access request process to a structured workflow, ensuring proper data governance.

### 5.9 Conclusion:

* Tiger is a powerful tool developed by TriNet's data management team to streamline reporting and data access.
* It enables users to interact with Tableau reports and EA dataware house efficiently while maintaining data security and compliance.

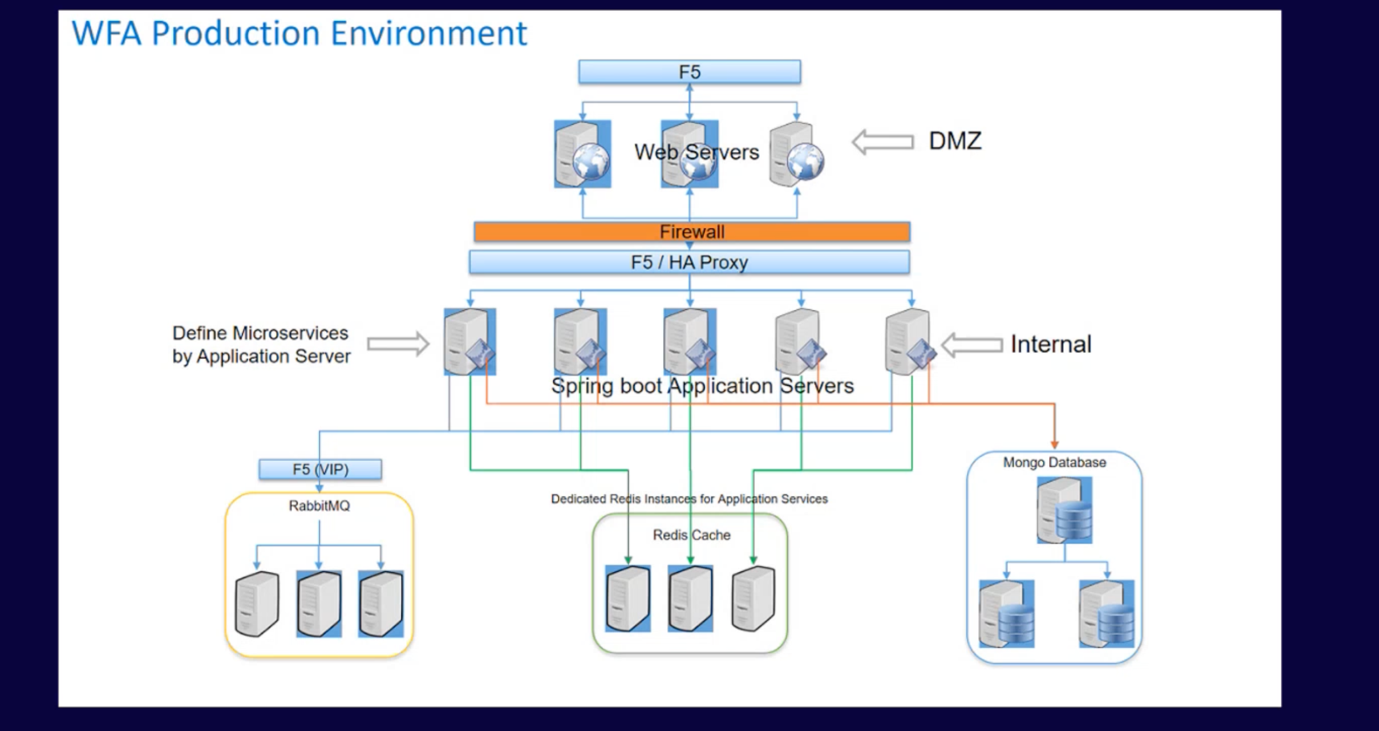
This summary captures the key components, functionalities, and significance of Tiger within the context of TriNet's internal reporting system.

### 6. Introduction to WFA (Workforce Analytics):

* + WFA is a front-end UI for external client reporting.
  + Accessible through a separate menu option on the Passport portal login.
  + Access is based on permissions and entitlements tied to client accounts.

### 6.1 Infrastructure Components:

* + Similar to Tiger, WFA utilizes application servers, Redis cache servers, RabbitMQ servers, and Apache web servers.
  + Multiple web servers are set up for failover and performance handling.
  + WFA employs Redis Enterprise for caching, with a cluster setup and administration console.
  + Mongo servers are also used, managed by the database management team.



### 6.2 Environments and Gateways Servers:

* + Various environments exist for WFA, including Dev,QE,Stage,production, and demo environments.
  + A WFA Gateway server serves as an entry point for internal colleagues, not visible to external clients.

### 6.3 Application Configuration and Property Files:

* + WFA and Tiger share configuration files and jar files, repurposing them with different application properties.
  + Each application has its own property file containing configurations for services like RabbitMQ, Redis, and Mongo.
  + Log files are also segregated per application for troubleshooting purposes.

### 6.4 Monitoring and Observability:

* + TriNet employs advanced monitoring tools like AppDynamics for observing application performance.
  + AppDynamics provides insights into application dependencies, traffic, and performance metrics.
  + A dedicated experience monitoring team utilizes these tools during infrastructure bridges to troubleshoot issues effectively.

### 6.4.1 Specific Application Monitoring:

* + AppDynamics allows detailed monitoring of applications like Redis and RabbitMQ.
  + It provides visibility into traffic, errors, and cluster status, aiding in quick issue identification and resolution.
  + Automatic queue creation in WFA RabbitMQ differs from Green Stack's manual deployment process.

### 6.4.2 Apache Web Servers:

* + Apache web servers in WFA are configured similarly to Tiger, with multiple instances for failover.
  + Monitoring tools like AppDynamics provide insights into Apache server status and performance.

### 6.4.3 Granular Monitoring with AppDynamics:

* + AppDynamics offers granular monitoring capabilities, allowing users to drill down into specific application components.
  + Detailed insights into errors, warnings, and performance metrics facilitate efficient troubleshooting and resolution.
  + Enables detailed analysis of application dependencies, errors, and performance metrics.
  + Offers insights into specific application components and issues.

### 6.5 Frequent Issues and Support:

* + Post-employment support and data setup issues are common.
  + Application services may experience downtime due to component failures or deployment impacts.

### Troubleshooting Approaches:

* + Utilizes App Dynamics for monitoring and troubleshooting.
  + Alerts from ServiceNow incidents and auto-resolution of some issues observed.
  + Multiple ways to troubleshoot, including log file analysis and monitoring tool dashboards.

### 6.8 Conclusion:

* + TriNet's investment in robust monitoring tools like AppDynamics enhances operational efficiency and ensures timely issue resolution across its application infrastructure.

Overall, the detailed summary provides a comprehensive overview of WFA infrastructure, application configuration, monitoring processes, and troubleshooting methodologies.

### 7.Introduction to Beyond Trust:

* + Beyond Trust serves as a password vault and storage locker for various entities.
  + It facilitates secure storage and retrieval of database passwords, domain passwords, and managed accounts.

### 7.1 Authentication and Access Control:

* + Requires multifactor authentication for access.
  + Access is based on permissions and privileges, with different users having visibility into different accounts based on their roles.

### 7.3 Usage of Beyond Trust:

* + Users can search for specific accounts based on filters such as domain or environment.
  + Accounts are retrieved based on the search criteria, and users can perform actions like copying passwords or displaying sensitive information.

### 7.4 Account Management:

* + Accounts are organized into smart groups for easier management.
  + Users can update passwords and view password history for managed accounts.
  + Accidental changes to sensitive information are mitigated by the system, ensuring security and integrity.

### 7.5 Incident Handling:

* + Accidental actions, such as inadvertently accessing sensitive information or changing passwords in production, are addressed promptly.
  + Proper procedures, such as stopping screen sharing and rectifying errors, are followed to mitigate any potential impact.

### 7.6 Security Measures:

* + Changes made within Beyond Trust are isolated to the system and do not affect the actual passwords in the identity directory.
  + Use of AD accounts and multifactor authentication enhances security and access control.

Overall, Beyond Trust serves as a crucial tool for securely managing and accessing sensitive information, with robust authentication measures and controls in place to safeguard against unauthorized access or accidental changes.

### 8. Introduction to RPA:

* + RPA (Robotic Process Automation) is a software technology used for automating business tasks, reducing human error, and enhancing security.
  + Automation 360 is the RPA software utilized in TriNet, hosted on Windows Server 2019 standard editions.

### 8.1 RPA Environment:

* + Three environments: development, testing (QA), and production, with two sites in Phoenix and Ashburn.
  + Production environment comprises 17 servers acting as bot runners, with Control Room serving as the management portal.

### Architecture Overview:

* + Bot creators, including app admins and developers, develop and push code to central repositories.
  + Control Room facilitates management of bot runs, scheduled buttons, and user access.
  + Bot runners are production servers where the automation processes are executed.

### 8.3 App Admin Responsibilities:

* + Infrastructure updates, access provisioning, and handling access-related issues.
  + Adding input files to bot folders, managing bot runs, and resolving incidents related to bot failures.

### 8.4 Incident Handling and Audit Logs:

* + Incidents related to bot runs are automatically created, and developers can also create incidents for assistance.
  + Audit logs provide detailed information about bot runs, facilitating bug identification and resolution.

### Accessing Control Room and Server Maintenance:

* + App admins access Control Room to manage users, roles, and perform daily maintenance tasks.
  + Password changes are done every six months, with documentation available for setup and configuration.

### 8.6 Troubleshooting and Common Issues:

* + Bot failures are addressed by restarting the bot service and checking status and logs for issues.
  + Common issues are documented, and regular maintenance ensures smooth bot operations.

### 8.7 Conclusion:

Overall, RPA plays a crucial role in automating business processes at TriNet, with app admins responsible for ensuring the smooth operation of bot runs, managing access, and handling incidents effectively. Detailed documentation and regular maintenance practices contribute to the reliability and efficiency of the RPA system.