

IT Production Environment

Jyoti Madabhushi

Tata Consultancy Services Ltd., Synergy Park, Non SEZ, Gacchibowli, Hyderabad, India

Abstract: *The production environment is where the application is actually used by the organization. It is the least volatile and most tightly controlled. One needs to upfront ensure understanding of system/solution performance characteristics. There should be an effective mechanism to establish SLAs, OLAs between business and IT solution providers. The environment should reduce the risk of deploying system/solution that cannot meet the user needs in terms of availability and response time. It should be a benchmark for capacity planning and other IT support organizations to monitor the systems effectively.*

Keywords: Servers, procedure, application, environment

Production environment refers to a server environment consists of hardware, software, data, operational plans and personnel that are necessary to keep the application working. Factors to be considered are availability, recoverability and performance. Application implies the application should be usable by intended users during the advertised hours. It can be disrupted due to a bug, database storage or application server failure. One way to promote availability is to decrease the single points of failure in an environment like using a static IP and a monitoring failover service ensures that users only access healthy load balancers. The ability to recover an application environment in the event of system failure or data loss is critical. Improving maintainability reduces the time required to perform a recovery process in the event of a failure. Application should perform as expected under average or peak load. Backup system, recovery plans, load balancing etc. are some tangible components that are utilized to transform existing setup into production environment.

A backup system grants the ability to create periodic backups of the data and restore data from backups. Backups allow for roll backs in the data to a previous state in the event of accidental deletion or undesired modification. Considerations for backup are backup selection, backup schedule, data retention period, disks space for backups, off site backups and backup restoration tests. As part of backup selection, we should backup any data that cannot be reliably reproduced from an alternate source. Backup schedule includes when and how frequently full or incremental backups can be done especially for active databases. One needs to use compression and incremental backups to decrease the disk space required by the backups. To protect backups against local disasters, a copy of the backups is to be maintained in a geographically different location. Periodically backup restoration process should be tested to ensure that the backup works correctly.

Recovery plans are a set of documented procedures to recover from potential failures or administration errors within the production environment. These plans enable one to utilise backup effectively and provide a blue print for rebuilding the environment or rolling back to a desired state when the need arises. Procedure documentation, automation tools, critical components, single points of failure and revisions are critical considerations. Procedure documentation is building a step by step document that can be followed to rebuild a failed server, adding steps for

restoring the various application data and configuration from backups. Script and configuration management software provide automation which can improve deployment and recovery processes. Bot the application and database servers are critical components because if either fails, the application will become unavailable. Critical components that do not have an automatic failover mechanism are single points of failure. These should be eliminated as far as possible. Revision includes the updation of documentation as recovery and deployment process improves.

Load balancing can be added to a server environment to improve performance and availability by distributing the work load across multiple servers. If one of the servers that is load balanced fails, the other server will handle the incoming traffic until the failed server becomes healthy again. Load balancing is not always required for a production environment but it is an effective way to reduce the number of single points of failure in a system if implemented correctly. It can improve performance by adding more capacity through performance scaling. Considerations include load balanceable components, application data replication, performance bottlenecks and single points of failure. Not all components can be load balanced easily. Special consideration can be made for certain types of software as databases. If a load balanced application server stores application data locally such as uploaded files, this data must be made available to other application servers via replication or shared file systems. This is necessary to ensure that the application data will be available no matter which application server is chosen to serve a user request. Bottlenecks imply if a load balancer does not have enough resources or is not configured properly, it can decrease the performance of the application. While a load balancing is used to eliminate single points of failure, a poorly planned load balancing can add more single points of failure.

Biggest benefits of monitoring systems is that they can be configured to trigger an action such as running a script or sending a notification when the server or service goes down or a certain resource such as CPU, memory or storage becomes over- utilized. These notifications enable one to react to any issues as soon as they occur which can help minimise or prevent the downtime of the application. Services to monitor, resources to monitor, data retention, problem detection rules and notification rules are the

considerations. One should monitor the state of all of the services that need to be in a healthy running state for the application to function properly. Resources to monitor include CPU, memory, storage, network utilization and the server as a whole. The period of time one retains monitoring data before discarding it along with the choice of items to monitor will affect the amount of disk space that the monitoring system will require. Problem detection includes the thresholds and rules that determine whether a service or resource is in ok state. Notification rules include the thresholds and rules that determine if a notification needs to be sent.

Centralised logging allows one to identify issues that span multiple servers by correlating their logs and metrics during a specific timeframe. It grants flexibility in terms of log retention because local logs can be offloaded from application server to a centralised log server that has its own independent storage. A log shipping agent is installed on each server and configured to send important app and database logs to the centralised logging server. The administrators of the system can view, filter and search all the important logs from a single console. Considerations include logs to gather, data retention, log filters and server clocks. The period of time that one retains logs before discarding them. This along with the choice of logs to gather will decide the amount of disk space that centralised logging system will require. Filtering logs will improve ability to query, analyse and graph the data in meaningful ways. The clocks of servers are synchronised and set to the same time zone so that log timeline is accurate across entire environment.

All the above components when put together leads to a full-fledged production environment.

End Notes

The change management process forces the support staff to be proactive and plan changes to the system rather than just execute them and wait for the consequences. Deployment of a new product must be carefully planned and downtime event if required must be coordinated. If a product is to be deployed across multiple hosts, partial deployment is done and tested for side-effects before doing wider deployment.

References

[1] Channel World, International Journal of Information Technology and Management

Author Profile

Jyoti Madabhushi, B.E, M.S. has got more than 25 years of work experience in IT industry with more than one and a half decade experience in project management. She has worked in various capacities from being a hands on technical person to project manager, program manager, portfolio manager to strategic business unit head mapping her portfolio to the changing trends in IT.