Week 2: ServiceNow Administration Fundamentals

Submitted by: Ananya Herle: Dayananda Sagar College of Engineering
Superset ID: 5098517
Registered mail id: ananyaherle2003@gmail.com

What is ServiceNow:

- Servicenow: cloud software platform providing Application program as a service to its clients; Spans various solutions for different business problems wide across the enterprise (ITS mgmt, customer relationship mgmt, ITO mgmt, HR, Security, Intelligent apps..)
- Earlier ticketing applications to receive employee queries, today used to automate business processes with its low code no code approach

ServiceNow Architecture/ Security/ Interaction/ Components :

Servicenow architecture:

a) Application platform as a service, hence provides users with infrastructure, platform and applications
b) Multi instance and not multi-tenant, allowing companies to have uniques, dedicated databases that are not intermingled with other organizations.All necessary components to be provided to the client by servicenow is bundled into a stack called “INSTANCE”. An organization can have several instances, each independent of one another, to cater to different clients
c) Single data model, uses tables to store data in a relational format across the entire instance, also includes certain core, reusable functionalities
d) High availability ensured by cloning the instances across various cloud data centers; these redundant copies help overcome issues with availability of data
e) Frequent backups: 4 weekly full backups with 6 days of different daily backups

Difference between multi tenant and multi instance architecture;
This prevents the customers of an organization be affected from maintenance activities issued by other companies sharing the cloud infrastructure

- Security in servicenow can be role based alongside integration with 3rd party authentication services like
    a) LDAP(user authenticated via lightweight active directory account)
    b) Local database
    c) oAuth 2.o(uses client id and secret to authenticate users)
    d) SSO(single sign on uses details provided by identity provider)
    e) MFA
    f) Digest token(encrypted format of username and password)
- Interaction ways:
    a) Now platform desktop UI
    b) Mobile interface(Agent, now Mobile, servicenow onboarding)
    c) Service portal(domain_name/ sp; access servicenow features like knowledge base, catalog, help or view your requests using user friendly interface customized by application developers; uses widgets to provide functionality)---> options available depends on access given to users
- Roles in servicenow
    A) Requesters
    B) Fulfillers
    C) ITIL→ read and write access to ITSM activities and records
    D) Admins
    E) Specialized Admins
- Servicenow components> USER > GROUPS> ROLES
- Users, groups and roles are “**OUT OF THE BOX**” records; provided by servicenow instance it self as a part of baseline implementation
- Roles determine what user can see and access on the instance as most servicenow records/tables and modules/applications are secured using role based access
- User interface version latest> UI16
- UI elements: Banner frame, app navigator, content frame

- History tab of the application navigator can be customized to store user activity history for any “x” days
- For branding instance> goto system properties> basic configurations UI16

List and Filters:

- Context menu:
    a) Context menu
    b) Column context menu
    c) Field context menu
- View: create a different version of a file by applying certain filters or personalizing a list for specific users
- Group by can be done using any table field irrespective of whether its seen on the list
- Saving a filter done using
    a) Creating favorite
    b) Creating view
    c) Save filter> access by context menu> filter> your saved filter name
- Group by done by:
    a) List context menu
    b) Column context menu
- List records will be group as per the distinct values of the attribute concerned
- List layout allows addition/ removal of existing fields/ add new fields/ add new views
- List control> record with fields that decides whether list can be manipulated by the user and in what manner
- Searching
    a) List header> select field and value for searching
    b) Column search> enter field value in the input box listed below the column name, enabled by clicking search icon

Forms:

- Form elements
    a) Content frame(main section that displays form fields)
    b) form / record header
    c) Form context menu
    d) UI action buttons
    e) Sections
    f) Related links
    g) Related lists
- Save/ submit difference → Save: remains on form and displays additional options, does not add record to database
- Submit: adds record to database and return to list view
- Fields can be mandatory(assigned a red asterisk) or readonly (gray in color)
- Form configuration> form layout/ form designer
- Form layout> add or remove fields that exist in the table
- Form designer> manipulate appearance of the form or add field/ create sections
- Record header> toggle template bar> once template bar visible> add a template or use existing template

Knowledge Check Question answers:

- Which app visible to ESS users> **Self Service**
- Primary way for data access> **Native UI**
- Role should be added to user directly T/F: **F add to group**
- Baseline system roles: **itil and admin**
- Button to copy and incident and create new record: **Copy Incident**

Task Management:

- Task is record of the task table that is used to report an event of problem, change request or incident in servicenow
- Managed by **itil** users
- Can be issued by a requester and further is assigned to a user/ group for resolving or fulfillment
- Task workflow> example
    a) Employees computer stops working
    b) Raises an incident
    c) Incident record assigned to the IT group
    d) The record is then taken up for resolving by an appropriate person from the IT team
    e) Task record goes to status like new> in progress> resolved> closed
    f) Task records can be resolved even using automatic workflows

- Functions associated with tasks
    a) Approval
    b) Assignment(manual/automatic)
    c) SLA (ensuring that the task is fulfilled within promised time)
- Task is an **out of the box** servicenow table that is a base and parent table to incident, requested item, change, problem
- SLA configured using the Service level Agreement application
- Approval for task records> depends on state, if approval state is requested> the approver is notified via email
- Use workflows for automated approvals, by defining necessary conditions
- Task assignment:
    a) Manual(manually populate the assignment group and assigned to fields; ensure to assign to user who has access to the extended task table; manual assignment can be done by admin or itil roles)
    b) Assignment rules(trigger> table> assigned to which user/group> if necessary use script)
    c) Assignment lookup rules(if the **incident** record fields has the lookup rule defined values then it is assigned to a specified user/group)
    d) Custom script
    e) Predictive intelligence(intelligent agents used to predict field value> enable via plugin/ licensed application)
- Service Desk> used by users to view tasks assigned to them or to their group
    a) My work
    b) My groups work
    c) My approvals
    d) Knowledge
    e) SLA
    f) Callers
    g) Incidents
- Effective task management
    a) Work notes: used to pass important information/updates to other task record viewers→once posted added to activity
    b) Additional comments: used to update/ communicate with the task issuer or requester
    c) Activity tracker: used to track changes made to record
    d) Email: used to communicate and notify users about their task status→once sent, added to activity

Notifications:

- Inbound: when servicenow sends notification to users> example: email sent to a group upon task record assignment
- Outbound: Respond or email sent to servicenow from user to manipulate property of record
- Handled by **System Notification Application**
- All system emails recorded in system logs
- Modules under Notifications>
    a) **Digest Intervals**: define intervals for digest email functionality; used to reduce the number of email sent to a user for specific notifications in a given time period and replacing them by one email notification
    b) **Notifications module**: list of all notifications in current instance
    c) **Email Scripts**: custom javascripts for email notifications
    d) **Notification categories**: out of the box email categories; defines primary purpose of notification
    e) **Email Template**: list of predefined email templates; when template is used the email body and subject are populated by the template; however the email body filled by template can be overridden by new body content
    f) **Notification filters**: list of filters that can be applied on the notifications; applied from the user preference option> notifications section; Example> enable notifying user about those incident records that have critical priority; **out of the box filters: critical/ unsubscribe**
    g) Email restriction access: restrict access to users from specific type of emails
- **Servicenow also has out of the box notifications for the task records**
- Notification table> sysevent_email_action
- Creating new notification> entry of a new record to the notifications table
- Components of the notification form
    a) Name of notification
    b) Table name
    c) Notification category
    d) When to send(send when 1) record updated/inserted 2) triggered by flow designer 3) event)
    e) Whom to send(user/group / subscribable)
    f) What to send(content type/ subject/ body/ template/ include attachments/ from mail id/ push message(only for mobile))
- Dynamic content>> ${field name belonging to table} ex: ${number}
- Alpha numeric code on end of email: watermark> used by instance to recognise the where the response by user must be updated

- Email scripts> add custom script in email body in the following format **${mail_script.<task record type_parameter>}**
- Inbound mail: SN can respond or cause actions due to receiving of inbound email from external users; defined by **system policy**>**email> inbound actions**
- Create a new record for an inbound email action
    a) Name
    b) Target table
    c) Action type(record action/ reply)
    d) When to run(type of email/ condition/ order of execution/ roles)
    e) Action(script)

Knowledge Management:

- Knowledge bases are organizational repositories to store important documentation and knowledge articles,allowing users to resolve certain issues by themselves before raising tickets, troubleshooting or self help tips
- Hierarchy of knowledge base> **categories**> **articles**
- Knowledge mgmt application: out of the box servicenow app
- Homepage > categories, most viewed, most useful, featured content
- Categories featured to user depends on their access
- Ui buttons> create an article(**available to people with create access**)/ post a question
- Article contents:
    a) Article number: KB____
    b) Title
    c) Article body
    d) Rating
    e) Mark as helpful
    f) Flag
    g) Feedback/ comment
- Create article module creates new record to the article table> opens up a form for the same; specify name, base, category, short description and article body; use search for duplicates button to check if article exists
- Import article module, allows importing of word files and post the same as KB articles
- Publishing knowledge base articles controlled by workflows that may pause the publishing process and wait for approval> hence article remains in unpublished state until approved
- Feedback provided by users to articles managed by the **feedback** module

- Workflows for KB>
    A) Approval publish
    B) Approval retire
    C) Instant publish
    D) Instant retire
    E) Publish subflow
    F) Retire subflow
- Article lifecycle> when created **DRAFT**> upon clicking publish **SENT FOR APPROVAL**> if approved **PUBLISHED**> else **UNPUBLISHED**
- **By default the workflow forward the request to the knowledge manager for approval to publish or retire knowledge articles**
- User criteria used to manage access to articles(can read, can't read, can contribute, can't contribute)> add user criteria in knowledge base form(related lists); **by default only can read and can contribute shown; configure related lists to showcase can't read and can't contribute**
- **Knowledge_admin and system admin can add and configure knowledge bases**
- User criteria diagnostic tool tells the user criteria a user on SN platform has for a specific article
- Knowledge bases can also be accessed using service portals
- Importing articles:
    a) Knowledge application>articles> import articles or knowledge base homepage, click on UI button create article
    b) Publish the article

Knowledge check answers:

- Tables that extend task: **problem, change request, incident**
- Name two assignment fields of task: **assignment group, assigned to**
- How incident can be assigned automatically on basis of category and CI: **assignment rule**
- Which functionality of SN can create or update records based on emails sent to instance: **inbound action**
- Dynamic content in emails using: **Email scripts**
- What extension of file must be used for import in KB: **doc/ docx**
- Visibility of articles defined using: **user criteria**
- Which field provides informations about task from which article is created or related to: **Source Task**

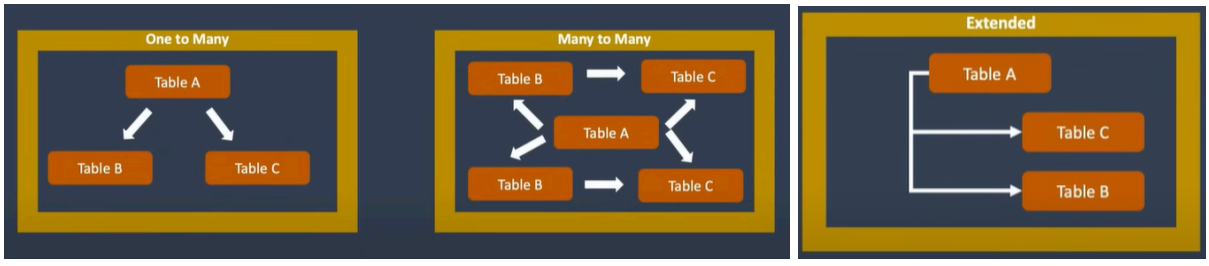
Service Catalog:

- Marketplace similar application allowing servicenow users to order for necessary software, hardware, devices or configuration item resources or report tasks using record producers
- All items are categorized into categories; users are posed questions or variable sets to find the device/ service that best suits their requests
- Available to users using **self service** module; categories and items visible to users determined by **user criteria**
- Manage service catalog features and items using **servicecatalog** application
- To define new categories: **service catalog> maintain categories> new**
- To define/maintain catalog items: **service catalog> maintain items**
- Categories can also be extended
- Fulfillment of requests for catalog items satisfied by workflows; requested item lifecycle> **REQ(sc_request)> RITM(sc_req_item)> SC_TASK(catalog task)**
- Once catalog item is defined preview it using **try it** button
- Roles:
    a) **System admin**(all access)
    b) **Catalog admin**(admin og service catalog application> no scripting access)
    c) **Catalog manager**(manages tem in a particular catalog)
    d) **Catalog editor**(update or edit a catalog and catalog items)
- Catalog editors can assign other editor but cannot change catalog manager
- Catalog managers can add other catalog managers
- Service catalog components
    a) order(catalog items that the user can order)
    b) Order guide(set of relevant items that may be required by the user> order guides suggest and assist user to buy all those relevant items)
    c) Record producers: simplified version of task record forms that are user friendly: which include question that the user can easily understand
- Order forms components:
    a) Variables: questions/ field to be filled by user for requesting a catalog item
    b) Variable sets: collection of variables; reusable(requested for, on behalf of)
- Order fulfillment adheres to workflows or flows(scriptable/ drag and drop)or execution plans(choose either one)
- User criteria for catalog→ available for and not available for
- An order guide must show individual catalog items the user may have to purchase depending upon the initial details provided by them.This order guide is dynamically altered using rules
- Forms are for **itil** users familiar with servicenow instance and technical aspects

- For general users wanting to submit tickets, simple record producer UI can be used
- Variable types: select box, multiple choice, multiline, single line, email, reference
- Approvers must approve the request laid by a user to move catalog items between sc_request> sc_req_item> sc_task(has to be fulfilled by someone)
- Request stages> **manager approval> department approval> configuration of item> delivered(customisable; not mandatorily the same)**
- How to create a catalog item>
    a) Create a catalog
    b) Under catalog related list> create a category
    c) To create catalog item> go to the catalog related list> catalog item> new
    d) Fill in basic details and save> further add variables/ variable sets
    e) Preview using try it
- Person who is assigned to fulfill request is provide the view of the variables answered by the initiator of request
- Standard process that do not require any approvals from other higher order users can follow execution plans for req processing

Tables in SN:

- System definition> tables/ tables and columns module for creation of tables
- Dictionary: module that provides a detailed information about table and columns
- Each record has a 32 character unique key called **sys_id**
- Field components: field label(for user); field name(for system), field value
- To configure fields> right click on field> configure dictionary option to open dictionary entry of field
- Table relationships:
    a) One to many (incident is one to many relationship with sys_user, cmdb_ci, sys_user_group)--> reference fields, glide list, document id field
    b) Many to many (bidirectional relationship→ group and role table)
    c) Database views
    d) parent / child (task→incident)

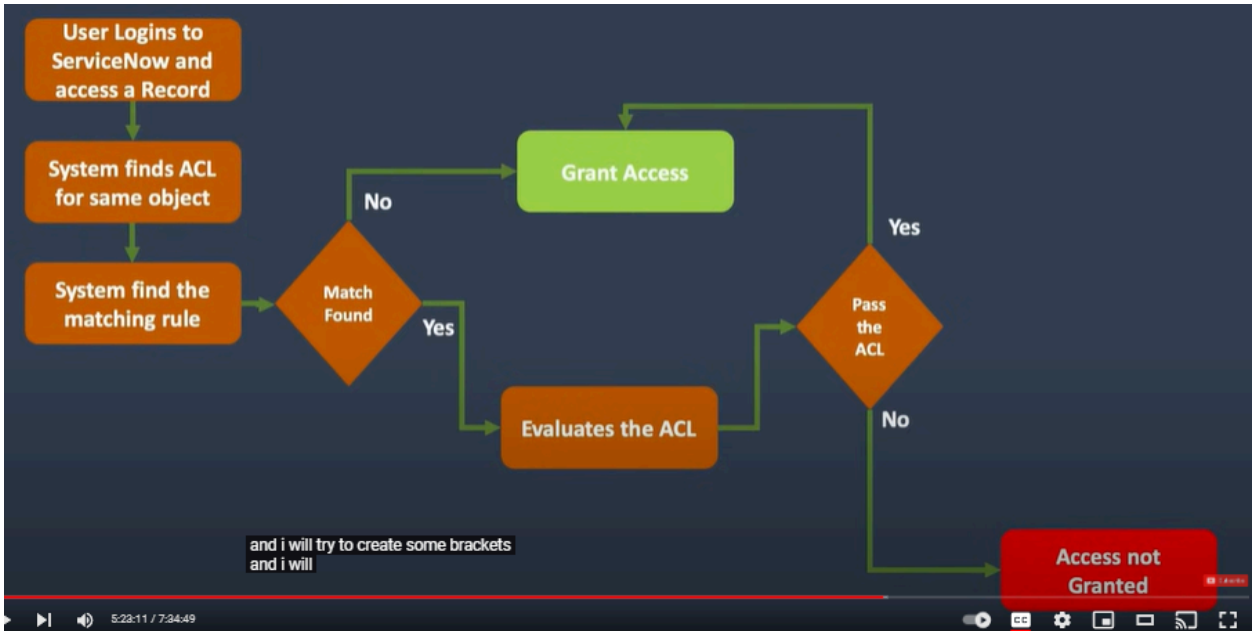


- Table types in Sn:
    a) Base: parent to another SN table; has no parent of its own
    b) Extended table: table extended from existing child table
    c) Core table: table included in SN baseline implementation
    d) Custom table: configured by the user
- Custom table naming uses scope: u_(global) and x_(application scope)
- Users can specify if table created must be made as a module
- Controls of table determine whether table is extensible, whether to create ACLs and a mandatory default user role
- Table application access> can read/write/update/delete, accessible by web services, allow configuration to table by other scopes
- Schema map> visual representation of the relationships between tables> access using tables and columns module

Access Control Lists:

- List of rules that control access to servicenow tables, records and fields
- Permission types—-> Login(authentication)/ Application and modules(governed by roles)/ records,fields and tables
- Access level controls help manage access to CRUD operations
- Servicenow specific operation restrictions
    a) Execute
    b) Edit ci_relations
    c) Save as template
    d) Report
    e) Personalize choice
- To add and configure ACL> use high security setting module/ system security>ACL module with **role elevation to security_admin**
- Acl form components:
    a) Type of ACL(mostly record)
    b) Operation(CRUD)
    c) Roles required for ACL
    d) Table name and ACL type(none, field or *)
    e) Script and conditions(advanced view)
- Order ACL evaluation> role/condition/script
- Admin override if enabled allows admin to skip ACLs
- ACL goes from Specific to general evaluation

- Types of ACL
    a) Table_name.none: access of operation listed given to all table records
    b) Table_name.field: access given to only listed table field
    c) table _name.*: applies to all fields of the table where a field level ACL is not applied(restrictive rule)



- As a system administrator> ACLs rules cannot be created as the new button on ACL list is not shown
- Upon new table creation, ACL created by default(CRUD), these ACL rules include the default table user **u_tablename_user** role. Hence this role is granted access to CRUD operations automatically; however during table creation user must enable create ACL option for the above to occur; user must now create ACL and default roles manually
- Example of ACL working:
    a) Assume to create a table say ACL demo, u_acl_demo_user default role has CRUD access along with admin
    b) Assuming another itil user has the u_acl_demo_user role, implying this user can perform CRUD operations on ACL demo table
    c) However we now create ACL demo.test2 read access for admin→ where test 2 is a field
    d) Now the itil user is unable to see the test 2 field of the ACL demo table

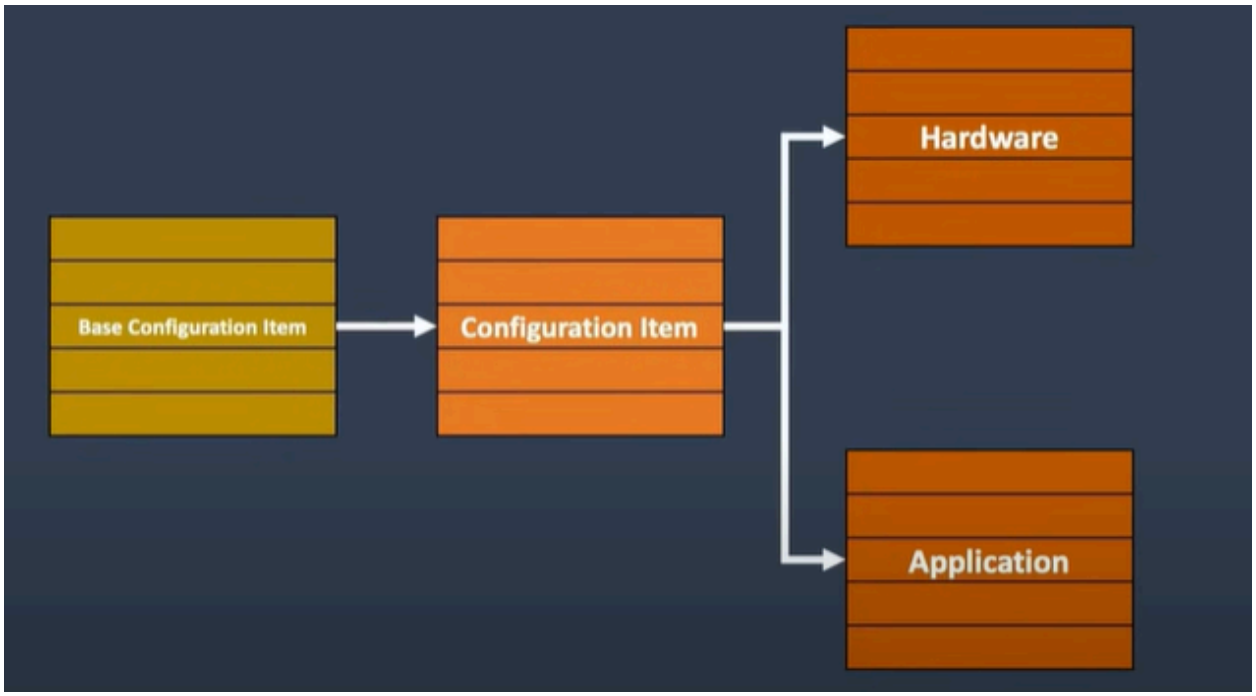
Data Import:

- Data import helps import KB articles, Active directory, HR data, Assets data into existing SN tables, thus reducing overhead of manually creating records
- Module used> System Import sets> **load data(allows import of only excel files)** or use import XML option available in list context menu
- Roles: **import_admin/ admin roles**
- Data source: file, jdbc, SQL data, attachment, LDAP
- Import set table: staging area to temporarily store imported data records
- Transform map: map data from staging area to target SN table
- Mapping Assist: used to create field maps automatically or manually
- Coalesce: select a field as key to prevent dupe entries: every time a key match is found in the staging table to the target table, the entry is the target table is updated(single field, multiple field, conditional)
- XML import→ choose any record and download its XML format, make any change to the XML, now on any tables list context menu, select import XML option
- **Define data sources in servicenow using data sources module**
- Imports can be **scheduled**
- Import logs showcase all import activities; after successful data imports delete the import set tables created; perform cleanup using **Cleanup or Scheduled Cleanup module**
- Data policies→ similar to UI policies to make fields mandatory/ read only but apply to the entire table rather than just the Ui form—————-> example: create a data policy to make a field mandatory during import, if that filed values does not exist, import does not occur
- Though the import is successfully run> check import logs to find that the records that do not abide by data policy were skipped

Configuration management Database(CMDB):

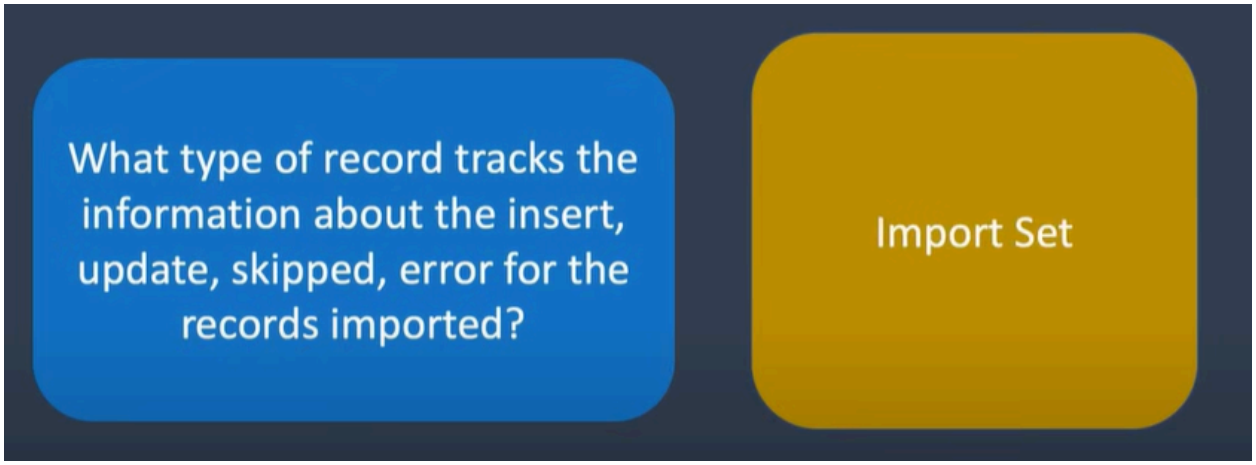
- Database that stores details about configuration items of an organization and relationships between them
- A CI is a tangible or intangible device or application
- Out of the box app to manage CI's is the **configuration application**
- CI form is a way of submitted necessary details to introduce a new configuration item

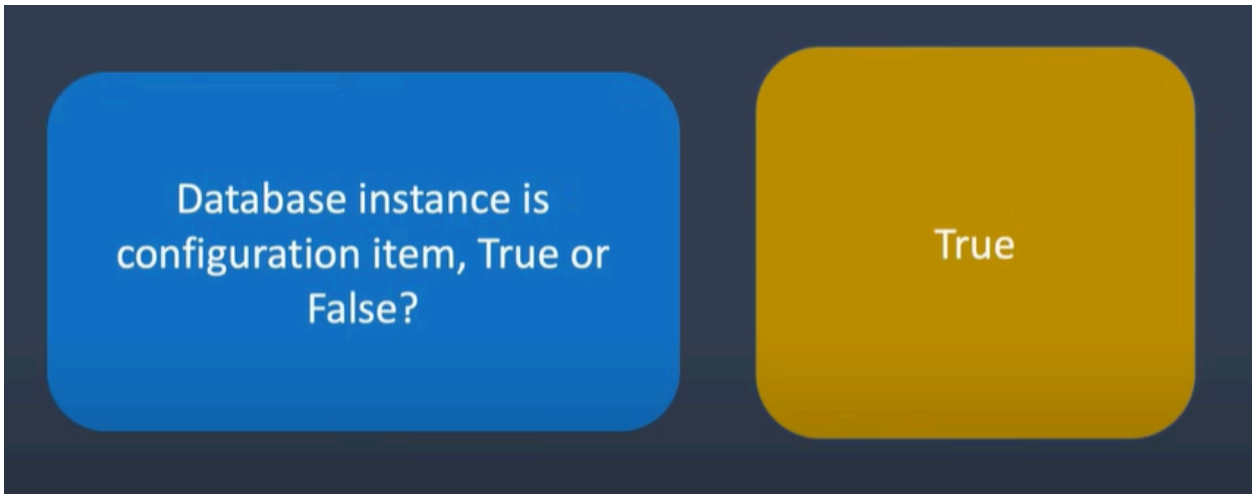
- CI form elements:
    a) Name
    b) Asset tag (unique identifier)
    c) Class (class to which the CI belongs to)
    d) Company (company that has this CI)
    e) Assigned to (primary person responsible of maintaining the CI)
    f) Configurations section> shows the configurations of the CI
    g) Related items: related CI to the current CI
- CI dashboard view: dynamic view about the health of the CI as calculated by certain defined metrics, related CI and tasks associated with the CI
- Key CMDB tables:
    a) Cmdb (base/ parent table for all cmdb data)
    b) Cmdb_rel_ci (table having data about CI relationships)
    c) Cmdb_ci (table storing data about all CI's owned by the organization)



- CMDB> used in incident, problem, change request tables
- CI dependency view> graphical representation / view of a CI and all other CI related to current record, also indicated any issue with the CI or any alerts
- CI class manager> displays the entire CI class available in the instance in tree format, showing all CI class definitions in a single place. Easier to view, visualize, modify and extend CI class
- To add new relationships between CMDB items, in related item section of CI form, click the "**+**" icon to add new CI relation in relationship editor
- Suggested relations suggest the relationships that a CI has to have ideally

- To graphically see the relationships between items> click on “**dependency view**” option on related items section





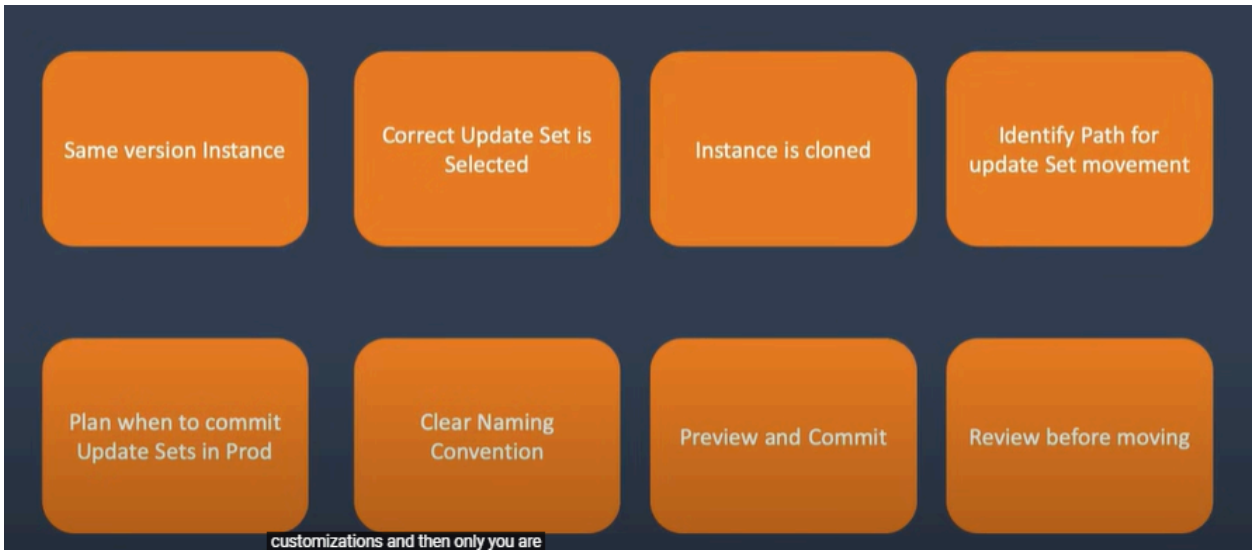
Integration in SN:

- Sn can share data with 3rd party apps or external services using integration concept
    a) SSO
    b) LDAP
    c) Monitoring
    d) Notifications
    e) Events
- What can be integrated:
    a) User login/ authentication using SSO
    b) User mgmt
    c) Incident, change and problem mgmt
    d) cmdb
- Ways to integrate to instance:
    a) Web services(SOAP and REST)
    b) LDAP
    c) Excel
    d) email
- Integrations with 3rd party apps also facilitated by **integration hub,** allows integration without coding, using flow designer concepts. The applications that have all necessary flows for integration purposes are called a **spoke.** Integrations vis the hub require license
- Enterprise licensing provides access to all spokes developed by servicenow

Update sets in SN:

- Update sets help transport changes and configurations made to an instance in non production environments, to the production environments
- Update sets help ensure consistent changes among all instances
- Admins can group a series of changes into a named update set and move these units to other systems for testing or deployment
- Update sets managed by **System Update sets** application
- Application modules:
    a) Update sources: details about sources from where update sets are pulled
    b) Update log: log information about updates made to an instance
    c) Local update sets: list of various update sets defined in the instance
    d) Retrieved update sets: updates sets that are retrieved on the instance
    e) Merger update sets: list that enables merging of existing update sets
- Recommended: use named update sets for propagating changes across instances over default

- Default update sets ensure to capture all changes made to an instance in a single place
- items captured in update sets:
    a) form/ list configurations
    b) Business rules
    c) scripts(server and client)
    d) Workflows
    e) Flows
    f) Script includes
    g) UI policy and actions
- Items not captured in update sets:
    a) Data
    b) Task Records
    c) Users
    d) Cmdb records
    e) System properties
    f) groups
- Applying update sets> r**etrieve>> preview>> commit**



Update set promotion>
Development> user acceptance testing> staging> production

- Update set tables:
    a) sys_update_Set
    b) sys_update_xml

Events in SN:

- Special log records that the system generates when a notable occurrence occurs
- Event generated by script(Business rules), workflow, flow, event queue scripting API(gs.eventQueue)
- For an event to occur> it must be defined and registered in the event registry
- Logging details about event> present in event log
- Event resultant> script action or notification
- Stats module> provides statistics of system activities that affects performance; also provides details about instance cluster and version