Manage eazyBI in Large-Scale Jira Instances

Daina Tupule

Taking on new
challenges

Daina Tupule

Day in eazybi Cloud

50K
imports

120K
report executions

and growing ..

eazyBI is designed to run with many accounts

Do not guess - measure

Team account: support reporting
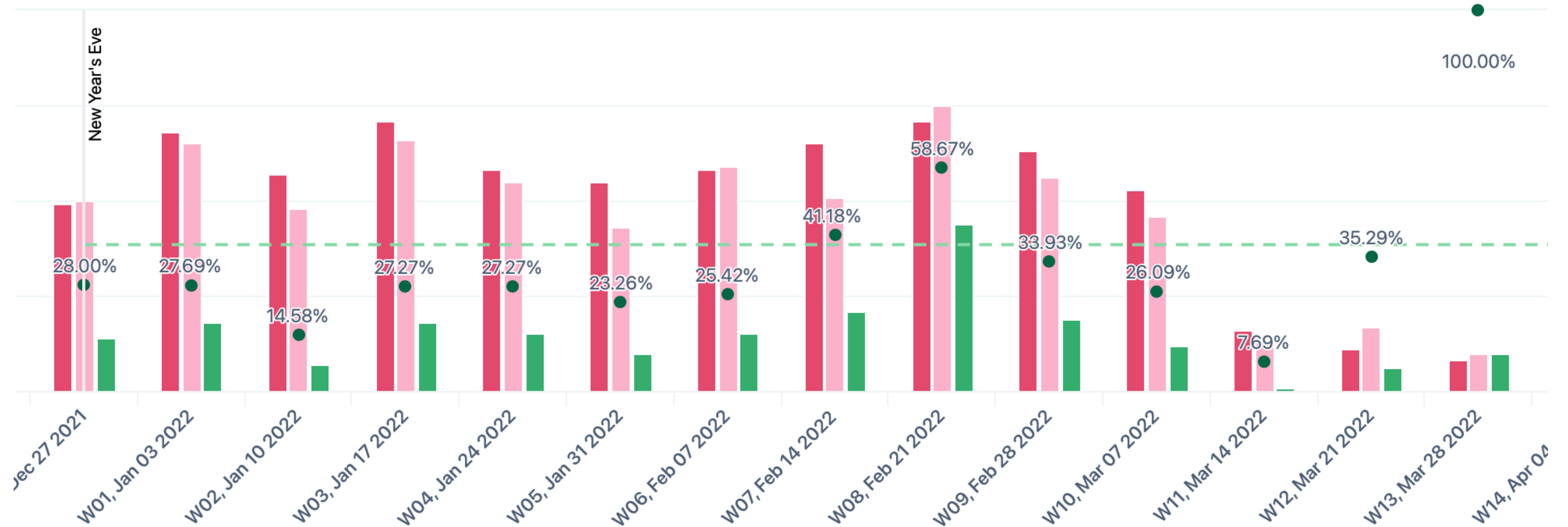
General account: usage statistics

All Cube Types ∨   All Report Locations ∨   All Accounts ∨

	% of all								
	Table	Bar chart	Line chart	Pie chart	Scatter chart	Timeline chart	Map chart	Gantt chart	Gauge
+ 2018	24%	42%	12%	10%	2%	3%	0%	0%	7%
+ 2019	27%	32%	16%	11%	1%	3%	0%	0%	9%
+ 2020	27%	36%	15%	10%	1%	5%	0%	1%	6%
+ 2021	33%	34%	12%	10%	0%	4%	0%	1%	5%
+ 2022	36%	34%	11%	8%	0%	4%	0%	1%	6%

TEAM ACCOUNT

- Small data set
- Better performance
- More freedom, rules suggested

GENERAL ACCOUNT

- Larger data set
- More expertise needed to get good working reports
- Strict rules to avoid haoss

Accounts

Account

Data

Reports

Users

Scope
Import projects and use JQL query to specify what information will be imported into the account

Details
Select custom fields, issue cycles, and apps - what details you would like to see and use for imported issues in your reports

Extend the data model *
eazyBI allows extending the data model with additional data import and custom javascript

Account

Data

Reports

Users

Extend the data model

Use **additional data import** to get more details for sprints, users, projects, etc.

Build custom hierarchies in Time, project, sprint, users, or other custom field dimensions

Define new account specific fields PLANNED

Override field behaviour (Jira fields / global customs) using **custom JavaScript code** and (PLANNED) setting a different behaviour for fields

Account

Data

Reports

Users

Use samples and default set of measures and calculations

eazyBI comes with a prebuilt set of measures, sample reports, and default calculations. Use our demo account examples for more inspiration

Template accounts

Build a set of **measures and reports** specific to your needs. They can give some common ground for all teams and could be used as building blocks in new reports

Extend the metrics *

Allow teams to create their own set of measures for specific needs

Account

Data

Reports

Users

Extend the metrics

Report specific measures
The primary choice when building any report.

Shared user defined measures
Move "proved report specific measures" as shared. Keep them organised. Consider using some **naming patterns**

Template reports and measures
Move shared reports and measures to template accounts if they could be reusable for other teams. Consistent naming is important there.

Account

Data

Reports

Users

Who can access data

eazyBI does not use Jira permissions to access data in reports. Set [users/groups](#) who can access the data per account

Dashboard viewer - read predefined reports published via dashboards
Viewer - read any report
User - create and edit own reports and measures
Report admin - create/edit any report, create shared user defined calculated members and measures
Data admin - create/edit data sources, import data
User admin - add users to the account
Owner - one user - responsible of the account, will receive error messages on imports

Tuning up

Settings > All Accounts

eaZyBI   Home   Source Data   Analyze   Dashboards   **System Admin**   Jira   DEMO ⌄   ⍰ ⌄   ⚙ ⌄   🐱 ⌄

All Accounts

| | | Search   or   Show recently created |

Search accounts by name, description, owner, source data type, "template" for template accounts, or "error" for source data in the error status.

Usage statistics *available for cloud and server/DC

				Source Data	Data size	Reports	Template (used in accounts)	Actions
(usage statistics)	INTERNAL	Adam Mint (admin)	2 years ago		📦 Size: 32.5 MB	**9 reports** 1 dashboard		⋯
Customer cases		Adam Mint (admin)	2 years ago	14 days ago	Default (every day)	Jira {REST:API} Excel CSV	📦 Size: 49.2 MB Files size: 25.7 KB Issues: 1860	**117 reports** 2 dashboards	⋯
DEMO		Adam Mint (admin)	2 years ago	about 15 hours ago	Default (every day)	Jira {REST:API}	📦 Size: 45.5 MB Issues: 1838	**167 reports** 14 dashboards	✓	⋯
DEMO - Insight		Adam Mint (admin)	about a year ago	about 17 hours ago	Default (every day)	insight	📦 Size: 3.5 MB	**16 reports**		⋯
DevOps		Adam Mint (admin)	about a year ago		No regular import	Jira	📦 Size: Issues: 132	**1 dashboard**		⋯

Template account

Usage statistics

Pages

> Time

Rows   Nonempty

> Account

Columns | > Measures | | > Report Location |

Table | Bar | Line | Pie | Scatter | Timeline | Map | Gantt | Gauge

⤢ | ↶ | ↷ | 💬 | ⤓ | Hide empty ⌄ | Total ⌄ | Freeze header

2022 ⌄

	Import Count	Report Executions + All Report Locations
Customer cases (3)	82	142
DEMO (4)	146	168
DEMO - Insight (45)	60	36
DevOps (53)		1
experimental (19)	18	51
Insight (6)	217	
Insight inheritance (7)	64	
Jira (52)	61	
Jira default (1)	307	142
Jira Service Management (8)	4	72
Jira uses cases (11)	241	25
Portfolio (10)	329	4
Source data - Custom data cubes (31)	1,502	
Source data - Jira custom cube (37)	181	

Tuning up

Optimize reports

Report execution setup

Imports execution setup

Detect the slow reports

Use **Usage statistics** to detect slow running reports. Check both the average execution time and the total report execution time.

Use best practices when creating reports

Check out the documentation page
Best practices for report creation

Profiling *

Use profiling to detect bottlenecks in the report execution. Function **Profiling** can help you detect measure working slow or part of the measure working slow.
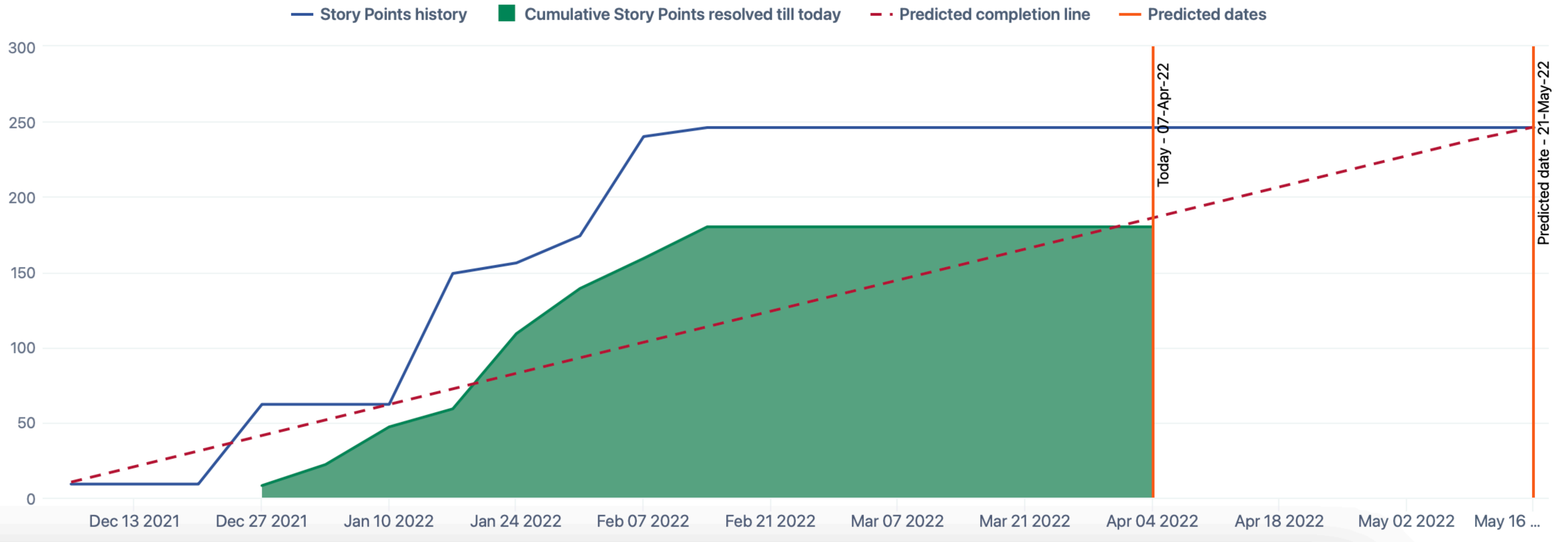
Report example for profiling

Table   Bar   Line   Pie   Scatter   **Timeline**   Map   Gantt   Gauge

⤢ ↺ ↻ 💬 ⤓ | Hide empty ⌄ | Time in version > 0 | All others | Total ⌄ | Line | Spline | Column | Area | Axes options ⌄ | Font size ⌄ | ⟳

Version report works for versions with some resolved issues with Story points and some unresolved issues with Story points. Usually used for unreleased Versions.
You should add Time members for Version period (dates between Version report start date and Planned completion date at least) to represent data over Time.

Unreleased ⌄

— Story Points history   ▇ Cumulative Story Points resolved till today   ┄ Predicted completion line   — Predicted dates

Today - 07-Apr-22

Predicted date - 21-May-22

Dec 13 2021   Dec 27 2021   Jan 10 2022   Jan 24 2022   Feb 07 2022   Feb 21 2022   Mar 07 2022   Mar 21 2022   Apr 04 2022   Apr 18 2022   May 02 2022   May 16 ...

Report example for profiling

```
Profiling(
    "name",
     <your formula goes here>
)
```

Optimize reports

Report execution setup

Imports execution setup

Define calculated member formula

[Measures].[Predicted completion line] =

```
1  --annotations.group=Predicted
2  Profiling(
3    "Predicted completion line" ,
4  CASE WHEN
5  -- apply for predicted period only
6  (DateinPeriod(
7  [Measures].[Version report Start Date],
8  [Time].CurrentHierarchyMember)
9  OR
10 DateBetween([Time].CurrentHierarchyMember.StartDate,
11 [Measures].[Version report Start Date],
12 [Measures].[Predicted completion date]))
```

Formatting   #,###.## Decimal ⌄

Read calculated members tutorial to learn about calculated member formulas. You can select members, operators and frequently used functions from sidebar to insert them into calculated member formula.

Root members

Issues created   Issues due

Issues resolved   Issues closed

Issues with due date

Issues last updated

Original estimated hours

Remaining estimated hours

Issues created count

Issues due count

Issues resolved count

Issues closed count   Hours spent

Issues with hours spent

Used only in this report. Convert to report specific.

Update   Delete   cancel

Report example for profiling

Tuning up

Optimize reports

Report execution setup

Imports execution setup

```
SumFunDef invoked 2663 times for total of 101ms.  (Avg. 0ms/invocation)
Cumulative Story Points resolved till today invoked 48 times for total of 2ms.  (Avg. 0ms/invocation)
version report start date invoked 2375 times for total of 2297ms.  (Avg. 0ms/invocation)
Predicted completion line invoked 48 times for total of 0ms.  (Avg. 0ms/invocation)
FilterFunDef invoked 10 times for total of 2355ms.  (Avg. 235ms/invocation)
MinMaxFunDef invoked 183 times for total of 6ms.  (Avg. 0ms/invocation)
Predicted completion date invoked 327 times for total of 1ms.  (Avg. 0ms/invocation)
Time in version invoked 1060 times for total of 2328ms.  (Avg. 2ms/invocation)
SqlStatement-SqlMemberSource.getMemberChildren invoked 2 times for total of 1ms.  (Avg. 0ms/
invocation)
SqlStatement-SqlTupleReader.readTuples [[Time.Weekly].[Day]] invoked 1 times for total of 16ms.
(Avg. 16ms/invocation)
SqlStatement-Segment.load invoked 7 times for total of 31ms.  (Avg. 4ms/invocation)
SqlStatement-SqlTupleReader.readTuples [[Time.Weekly].[Week]] invoked 1 times for total of 19ms.
(Avg. 19ms/invocation)


Execution time: 2539ms
```

Total time

Tuning up

Optimize
reports

**Report
execution setup**

Imports
execution setup

Child process
Creates a separate JVM child process to Jira JVM

Child process JVM memory
Make sure you have enough physical memory and designate resources accordingly. Increase when necessary

Query timeout
Increase query timeout if you have enough memory only. Make small increase.

Parallel query threads
Increase parallel query threads if your Jira server has enough CPU capacity to serve users running reports simultaneously

Tuning up

Optimize reports

Report execution setup

Imports execution setup

Frequency
Impacts the total import time the most. Set minimal frequency to limit this if needed

Allowed / disabled time
Allow or disable when imports could be run. For example, disable regular imports during working hours, enable manual imports only

Parallel imports
Increase the size of parallel imports if you have enough CPU capacity on Jira server

Profiling *
Enable if you have performance problems before sharing log files with us

Scalability action plan

Plan your accounts
Team accounts vs general accounts

Measure don't guess
Profiling, usage statistics, **repeat**

Scale the infrastructure
Resources - memory, CPU, and settings accordingly

Questions?

community.eazybi.com
support@eazybi.com

Thank you!